METHODS FOR UNIVERSAL FAULT-TOLERANT QUANTUM COMPUTATION IN SMALL DEVICES

A Thesis
Presented to
The Academic Faculty

by

Colin Jacob Trout

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Chemistry and Biochemistry

Georgia Institute of Technology
May 2018

METHODS FOR UNIVERSAL FAULT-TOLERANT QUANTUM COMPUTATION IN SMALL DEVICES

Approved by:

C. David Sherrill, Committee Chair
Schools of Chemistry and Biochemistry,
Computational Science and Engineering
Georgia Institute of Technology

Joseph W. Perry
School of Chemistry and Biochemistry
Georgia Institute of Technology

Thomas M. Orlando
School of Chemistry and Biochemistry
Georgia Institute of Technology

Kenneth R. Brown, Advisor
Schools of Chemistry and Biochemistry,
Computational Science and Engineering,
and Physics
Georgia Institute of Technology

T.A. Brian Kennedy
School of Physics
Georgia Institute of Technology

Date Approved: 15 November 2017

To my family, colleagues, and all the friends I've met along the way

Acknowledgements

I would first like to thank my parents for providing me the love, support, and opportunity to pursue my goals and who taught me to work to my full potential. None of this was possible without them.

I would like to thank my advisor, Kenneth R. Brown, for his mentorship. Nobody else has had a greater impact on my growth as a scientist than you and I continue to learn how to do better science through our interactions and your example. I would also like to thank my professors from Saint Francis University for facilitating my growth; making graduate school a possibility. Two professors stand out particularly, Edward Zovinka and Rose Clark. I have not met professors more dedicated and invested in their student's success and personal growth than them. They helped mold my personal dream into a reality.

I would also like to thank my past and present committee members C. David Sherrill, Thomas M. Orlando, Joseph W. Perry, T.A. Brian Kennedy, and Rigoberto Hernandez for their guidance, suggestions, and time.

I've spent a majority of my Ph.D. at the Georgia Institute of Technology around lab members and friends. I would like to that thank them all including Rick Shu, Afsaneh Sadrolashrafi, Kisra Egodapitiya, Jyothi Saraladevi, Omid Khosravani, Yu Tomita, Grahame Vittorini, True Merrill, James Goeders, Chingiz Kabytayev, Ncamiso Khanyile, Mauricio Gutiérrez, René Rugango, Yon Choi, Smitha Janardan, Muyuan Li, Aaron Calvin, Eric Sabo, Natalie Brown, Pak Hong Leung, Bichen Zhang, Harrison Ansley, Rob Clark, and Mason Bogue. It was a pleasure meeting and working with you all.

During my graduate career, I have had the pleasure of attending two internships at the IBM T.J. Watson Research Center and the Johns Hopkins Applied Physics Laboratory. I would like to thank my supervisor Jay M. Gambetta for the oversight and my advisor Sergey Bravyi for his mentorship during my time at the Watson

Research Center. My continued collaboration with Dave Clader and Jeffrey Barnes during and extending beyond my time at the Applied Physics Laboratory has enriched my graduate career through provocative discussion and suggestions. None of these opportunities would have been possible without the help of my advisor, Ken Brown, facilitating these opportunities.

The connections I made with people during my time in Atlanta is what kept my spirits high throughout my Ph.D. It was a pleasure meeting you all: Alex Elder, Giovanni DeLuca, Kevin Benham, Mena Aioub, Ashley Channing, Stephen Shiring, Allison Geoghan Aiuob, Aaron McKee, Aida Demisse, Matt Cooper, Johanna Smeekens, Hailey Bureau, Brandon Bakr, Eric Drew, Osiris Guzman, Zach Duca, and Dom Sirianni. I am glad to have met you all and to consider you my friends. In addition to the friends I've made at school, I am fortunate enough to have a solid base of friends that remain close in spite of the distance: Mat Hollingshead, Cory Treese, Chris McDaniel, David Kozak, J.P. Mannsberger, Noah Langer, Cody Silay, Jason Ruppert, Pat Himes, Seth Burkert, Andy Holzer, Shane McCormick, Wil Fulton, Maya Rudolph, Matt Mason, Ashley Wilt, Kara Schroeder, Madeline Wolf, Meghan Drago, and Kaylee Treese. I am lucky to have you all in my life. I would like to thank one person in particular, Kailey Crutchfield, who has been nothing but supportive in this pursuit from the moment I met her. Our time in Georgia has been short but sweet and I am excited for the next chapter in our lives moving forward.

Contents

List of Figures

List of Tables

Summary

A reliable large-scale quantum computer would be able to solve problems in physics and chemistry exponentially faster than current classical processors. A large-scale quantum device has not been built because quantum systems are naturally sensitive to environmental influences which manifest as errors in memory and operations during computation. For a large-scale device to become a reality, protocols must be developed that reduce the influence of errors during computation in a manner that maintains scalability of the device. This scalability criteria requires the protocols developed to handle errors must be implemented in a way such that the size of the quantum system and number of operations grows in a tractable manner. Furthermore, the sources of errors must be modeled accurately for true assessments of the viability of these protocols.

In this dissertation, we present an investigation into methods of performing reliable quantum computation in the presence of errors in small quantum systems (≤ 50 qubits). These methods should be considered as software primitives used to built reliable large-scale quantum algorithms and quantum memories. These methods occur in two flavors: quantum error correction and fault-tolerant operations. For quantum error correction, we perform assessments of error correction in the presence of error sources indicative of ion trap quantum computers. For fault-tolerant operations, we investigate the quantum resource cost and efficacy of implementing various techniques for performing reliable operations that would allow for a quantum advantange in a large-scale device.

Chapter 1

Introduction

A quantum computer is a device engineered to utilize the complexity of a many-particle wavefunction for the purpose of solving computational problems [3–6]. Quantum algorithms are predicted to solve computational problems that are intractable classically (scale exponentially in time of and memory for computation) in polynomial time such as integer factorization [7], computing molecular energies in the full configuration interaction [8], solving linear systems of equations [9–11], boson sampling [12], and solving scattering probabilities in quantum field theories [13]. Other quantum algorithms show a non-exponential speedup over their classical counterparts solving problems such as unstructured database search [14] and numerical integration using Monte Carlo methods [15]. The computational space solvable by quantum algorithms has yet to be rigorously explored due to the absence of a working physical architecture.

Experimental implementations of small quantum algorithms in systems containing under 10 qubits have been exhibited in a variety of architectures [16–26]. However, realization of a large-scale algorithm showing a distinct quantum advantage would consist of hundreds or thousands of qubits. Implementing such an algorithm will require protocols that protect the quantum states from sources of decoherence and are robust to faulty operations during the computation. Quantum error correction is a viable method for protecting of quantum states from sources of decoherence and fault-tolerant operations are robust to faulty operations [27–30]. Error correction routines embed logical qubits into subspaces of a multi-qubit Hilbert space and uses active feedback to remove entropy from the system. Fault-tolerant operations bound the influence of errors during computation.

To achieve a clear path towards implementation of a quantum algorithm displaying a quantum advantage, we must answer the question: how do we realize error correction and fault-tolerance in near-term physical architectures? To realize error correction experimentally, one must implement protocols that demonstrate an encoded system with a decoherence time longer than that of the individual qubits composing the system. A variety of error correction protocols have been developed, but the success of such protocols completely rely on the efficacy of simulating the underlying physics of the experimental architecture, which is computationally hard classically. Because of this, approximations of the error sources that make the simulation tractable classically have been typically implemented. However, we are on the verge of controlling quantum systems consisting of $10-20$ qubits which will allow for the implementation of quantum error correction. The intricate connection between the modeling of physical error sources and the success of error correction has never been more paramount. Constructing physically representative error models and testing error correction protocols under the influence of such errors is the major focus of this thesis.

Unfortunately, error correction is not enough for a quantum advantage. Fault-tolerant operations must be employed to make the computation robust to sources of decoherence that allow for full control of the quantum space of the encoded information. Both error correction and fault-tolerance require additional quantum overhead, both in the number of operations and qubits. Therefore, implementing the hundreds or thousands of qubits required to achieve a quantum advantage *reliably* will require control over a substantially larger quantum system than just the hundreds or thousands of physical qubits. Showing a quantum advantage with a small fault-tolerant algorithm that requires billions of qubits is not useful practically. Near-term devices must be able to implement small-scale algorithms in a manner that maintains scalability. Therefore, protocols must be developed and resources assessments must be performed to assess the quantum resource cost of error correction and fault-tolerance. The goal is to show that fault-tolerant computation is practically feasible; an additional focus of this thesis.

This thesis is organized in the following manner. In chapter 2, we provide all of the relevant background in quantum error correction and fault-tolerance to proceed. In chapter 3, we discuss the modeling of ion trap error sources to assess the implementation of the surface code on a linear chain of atomic ions. In chapter 4, we discuss progress in reducing the quantum resource cost of achieving universal computation fault-tolerantly. The output distributions of error-correcting/detecting code success criteria for codes under influence of incoherent and coherent error sources are investigated in chapter 5. The thresholds for two topological quantum error-correcting codes, topological color codes and doubled color codes, are provided in chapter 6. A conclusion and outlook is provided in chapter 7.

Chapter 2

Quantum Information Background

2.1 Qubits and Quantum Gates

In classical computers, the elementary unit of information is a *bit* which can be in one of two states: 0 or 1. For a quantum computer, the rudimentary unit of information is a qubit which can be represented as the state of a spin-1/2 particle. Mathematically, a qubit is a normalized vector in $\mathbb{C}^2$ which, in the computational basis, is represented by the state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.1}$$

where $|\alpha|^2 + |\beta|^2 = 1$ and the computational basis states are $|0\rangle = [1\ \ 0]^\intercal$ and $|1\rangle = [0\ \ 1]^\intercal$. The measurement of a quantum bit, $|\psi\rangle$, in the computational basis results in a classical bit value, 0 or 1, with probability $|\alpha|^2$ or $|\beta|^2$, respectively. One qubit can be in a superposition of 2 states ($|0\rangle$ or $|1\rangle$). Two qubits can be in a superposition of 4 states ($|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$) where $|01\rangle$ represents the tensor product between two basis states: $|0\rangle \otimes |1\rangle$. An n-qubit system can be in a superposition 2^n states. The ability of quantum bits to be in a superposition of an exponential number of corresponding classical states, where each state has an amplitude *and* a phase, is at the core of why quantum algorithms have shown speedup over classical algorithms.

In the circuit model of quantum computation, the operations are unitary transformations and qubit measurements which can be represented by quantum circuits. Quantum circuits are pictorial representations of quantum logic, consisting of unitary gates and measurement, performed on a set of qubits. Quantum circuits are the quantum analogue to classical circuit diagrams. The state of quantum system is altered using gates. Some common single-qubit gates are the Pauli spin matrices:

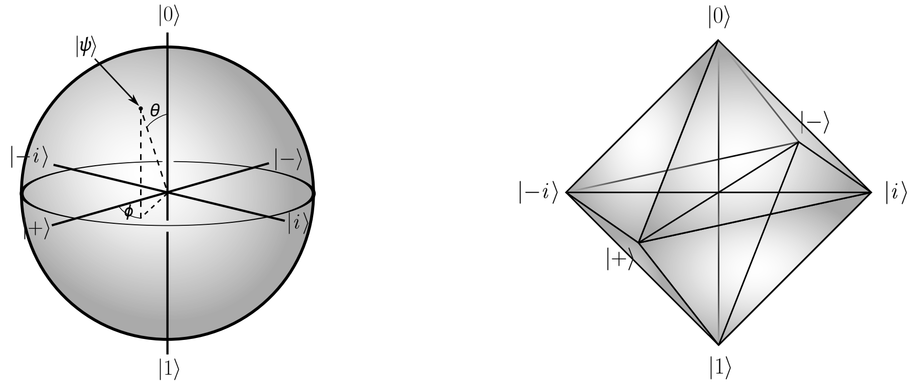
$$-\boxed{I}- \qquad I \equiv \sigma^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad -\boxed{X}- \qquad X \equiv \sigma^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$-\boxed{Y}- \qquad Y \equiv \sigma^2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad -\boxed{Z}- \qquad Z \equiv \sigma^3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

where the symbols on the left are the circuit model representations. More generally, quantum gates are unitary operators, $U^\dagger U = I^{\otimes n}$, where $\dagger$ indicates the Hermetian adjoint of the matrix U and $I^{\otimes n}$ is the n-fold tensor product of the single-qubit identity matrix I (above). In quantum algorithms, general unitary operations, U, acting on n qubits can be approximated up to any tolerance (δ) using a bound number of elementary gates known as a gate set [31]. A pure single-qubit state (ignoring the global phase which irrelevant), can be represented as a point on the Bloch sphere (figure 2.1a):

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.2}$$

where a single-qubit unitary gate, U, can be interpreted as a transformation from any point on the Bloch sphere to any other point on the Bloch sphere: $U : \theta, \phi \rightarrow \theta', \phi'$. This transformation can be efficiently approximated to arbitrary accuracy using a sequence of H (Hadamard) and T gates [31], which are of the form:



(a) The Bloch sphere

(b) Clifford transformations

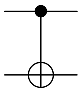
Figure 2.1: Visual representation of unitary transformations of a single-qubit state.

$$-\boxed{H}- \qquad H \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad -\boxed{T}- \qquad T \equiv \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{i} \end{pmatrix}$$

which is considered a *universal* gate set for constructing single-qubit unitary operations. While this is not the only single-qubit gate set that is universal, some gate sets cannot construct a general unitary transformation. Consider the single-qubit gates H and S (Phase):

$$-\boxed{H}- \qquad H \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad -\boxed{S}- \qquad S \equiv \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

which constructs the single-qubit Clifford group but cannot approximate a general single-qubit unitary transformation. The single-qubit Clifford operations are restricted to transformations about the chiral octahedron within the Bloch sphere [32] as shown in figure 2.1b. The Clifford group is of interest because many quantum error-correcting codes have "protected" unitary gates that correspond to Clifford transformations on the "protected" quantum states known as transversal gates (see section 2.4.1). Generating an n-qubit gate set performing universal and Clifford transformations on an n-qubit system requires one additional gate for each set, respectively: the controlled-NOT or controlled-X (termed $CNOT$) gate.



$$CNOT = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

where $|0\rangle\langle 0|$ indicates the outer product between the basis vectors: $|0\rangle$ and $\langle 0| = (|0\rangle)^\dagger$. The first qubit is the control qubit and the second qubit is the target qubit which is flipped in the computational basis $|0\rangle \leftrightarrow |1\rangle$ if the control qubit is in the $|1\rangle$ state. This two-qubit gate can generate entanglement between two qubits if the control qubit is in a superposition of the computational basis vectors. $CNOT$ operations between pairs of qubits in an n-qubit system and the application of

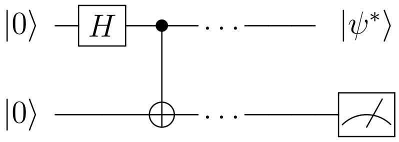


Figure 2.2: Quantum circuit generating entanglement between two qubits (left). Entangled qubits allow for completely correlated outcomes conditioned on measurement (right).

elements of the single-qubit universal/Clifford gate set on the n qubits generates a set of universal/Clifford gates over the 2^n dimensional state space of the qubits.

In addition to unitary gates, measurement serves as the information "read-out" during computation. We will adhere to a circuit model that implements strong projective measurement which projects a general quantum state into one state within a set of orthogonal states. Our choice will be the computational basis states and their associated projectors $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$ which constructs a complete measurement basis ($\sum_a P_a = I$). For a quantum state $|\phi\rangle$, the probability of measuring an outcome a and projecting into the state a is given by $\langle\phi|P_a|\phi\rangle$. For example, the probability of measuring a general single-qubit in the 0 state is $\langle\psi|0\rangle\langle 0|\psi\rangle = |\alpha|^2$ where $\langle 0|\psi\rangle$ is the inner product. The resulting state after the projective measurement is:

$$|\phi'\rangle = e^{i\phi}\frac{P_a|\phi\rangle}{\sqrt{\langle\phi|P_a|\phi\rangle}} \tag{2.3}$$

where the complex phase (again) is not of interest. This is known as the projection postulate of quantum mechanics.

An example quantum circuit constructed from elements above is shown in figure 2.2 that generates entanglement between two qubits followed by a measurement on the second qubit. Circuit diagrams are read from left to right starting with the input qubit state: $|00\rangle$. As shown above, single- and two-qubit gates have distinct representations which transform the input state: $H : |00\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$ then $CNOT : \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The meter indicates measurement and, in our case, it will always indicate measurement in the computational basis unless otherwise specified. The measurement of the second qubit will project the first qubit output state, $|\psi^*\rangle$, into one of two states, $|0\rangle$ or $|1\rangle$, conditioned on the measurement outcome of the second qubit. The output state $|\psi^*\rangle$ will have a 50% probability of being projected through measurement into either the $|0\rangle$ or $|1\rangle$ state.

2.2 Noise Models for Quantum Error Correction

Errors during quantum computation occur due to interactions between the computational quantum system and the environment. These errors can generate a non-unitary evolution on the quantum system's state [33] so it is more convenient to represent the quantum states in this context with density matrices, ρ. A general qubit state that is not necessarily pure ($|\alpha|^2 + |\beta|^2 \leq 1$), can be represented by a sum of the identity matrix and the Pauli matrices:

$$\rho = \frac{1}{2}\left(I + \vec{r} \cdot \vec{\sigma}\right) = \frac{1}{2}\left(I + r_x X + r_y Y + r_z Z\right) \tag{2.4}$$

where $\vec{r}$ is a real-valued vector indicating the degree of mixing of the qubit. For pure states described above, $|r| = 1$ and, for mixed states, $|r| < 1$. Returning to the Bloch sphere representation of a single-qubit state, pure states can be uniquely defined as a point on surface of the Bloch sphere and mixed states can be represented as points within the surface of the Bloch sphere. A general qubit state can be defined by the vector $\vec{r}$.

Because a quantum computer is not perfectly isolated, it is an open quantum system. Therefore, a useful perspective is to consider the composite system constructed by the quantum computer and it's environment. With a proper definition of the environment, the composite system can always be defined as an isolated system. The isolated composite system at time t is given by:

$$\rho_{CE} = U_{CE}(t)\, \rho_{CE}(0)\, U_{CE}^{\dagger}(t) \tag{2.5}$$

where $\rho_{CE}(0)$ is the initial state of the composite system and $U_{CE}(t)$ is the unitary time propagator of the system. After the time evolution, the quantum computer will generally become entangled with the environment. Therefore, the final state of the quantum computer is obtained by applying the partial trace over the degrees of freedom of the environment. If we assume that the initial state of quantum computer-environmental system is separable, that is $\rho_{CE}(0) = \rho_C(0) \otimes \rho_E(0)$, then the initial state of the environment can be expressed as $\sum_k \lambda_k |\phi_k\rangle\langle\phi_k|$. The state of the quantum computer as time t is then:

$$\rho_C(t) = \mathrm{Tr}_E(\rho_{CE}) = \sum_{i,k} \lambda_k \langle i|U_{CE}(t)|\phi_k\rangle \, \rho_{CE}(0) \, \langle\phi_k|U_{CE}^\dagger(t)|i\rangle \tag{2.6}$$

where $\{|i\rangle\}$ is an orthonormal basis for the environmental system. By defining $E_i(t) = \sum_k \sqrt{\lambda_k}\langle i|U_{CE}(t)|\phi_k\rangle$ the state of the quantum computer can be represented concisely as:

$$\rho_C(t) = \sum_i E_i(t) \, \rho_C(0) \, E_i^\dagger(t) \tag{2.7}$$

where $\{E_i\}$ are called Kraus operators. The above equation is known as the Kraus or operator-sum representation [33] of the quantum error channel. Dropping the time variables and the label C (from this point forward we will exclusively be considering the computational quantum system) gives the concise representation transformed computational state:

$$\varepsilon(\rho) = \sum_i E_i \, \rho \, E_i^\dagger \tag{2.8}$$

where the operators $\{E_i\}$ are not necessarily unitary. However, if $\sum_i E_i^\dagger E_i = I$ then the final state of the system and the environment can be represented by density matrices (positive semi-definite matrices with trace 1).

An example of a quantum error channel that can be represented with Kraus operators is the symmetric depolarizing error channel. The channel can be represented in the following manner:

$$\varepsilon = \begin{cases} E_0 = \sqrt{1-p}\, I \\ E_1 = \sqrt{p/3}\, X \\ E_2 = \sqrt{p/3}\, Y \\ E_3 = \sqrt{p/3}\, Z \end{cases} \tag{2.9}$$

This channel has an interpretation that, when applied to a single qubit, one of the Pauli matrices is applied to the state, each with a probability of $p/3$, and the state remains unchanged with probability $1-p$. Formally, this channel transforms the single-qubit density matrix like so:

$$\varepsilon(\rho) = (1-p)\,\rho + \frac{p}{3}\,X\rho X + \frac{p}{3}\,Y\rho Y + \frac{p}{3}\,Z\rho Z \tag{2.10}$$

This probabilistic interpretation of error channels has been traditionally used in many studies of quantum error-correcting codes. These error models have been a focus due to the ability to simulate many error channels of this type efficiently classically [34] and these error channels are exact when the an environmental state entangled with the computational system is orthogonal to all other states in the environment [35, 36]. This also serves as an accurate model for cases where the dynamics of the environment are faster than the dynamics of the computation.

Not all error channels admit the probabilistic interpretation defined above. A pertinent example of this is when the error results in a unitary evolution of the computational quantum state. These errors are often referred to as coherent errors are of particular interest due to their severe effect on error correction [37, 38]. An example of such an error is a coherent rotation of the qubit about the X axis by an angle θ:

$$RXC(\theta) = \exp(-i\,\theta X/2) \tag{2.11}$$

This error transforms the computational state as so:

$$\rho \xrightarrow{RXC(\theta)} e^{-i\,\frac{\theta}{2}X}\rho\, e^{+i\,\frac{\theta}{2}X} \tag{2.12}$$

Simulating errors of this type generally cannot be performed efficiently classically because they require tracking the evolution of the full computational Hilbert space (2^n states).

2.3 Classical Error Correcting Codes

To motivate quantum error correction, we will start will the concept of classical error correction. Error correction, both quantum and classical, uses some notion of redundancy to make the information more robust to errors. A simple classical error-correcting code is the classical *bit-flip code.* The code encodes the classical states, 0 and 1, into three copies of their respective states: $\overline{0} = 000$ and $\overline{1} = 111$ where the overline indicates an *encoded* state. This code is robust to classical noise that flips each bit independently with some probability p ($0 \leq p \leq 1$): the *bit-flip channel* [39]. If a single bit is under the influence of the bit-flip channel, then the information would be lost at a rate of p. Now consider a bit with bit-flip encoding under the influence of the same channel. Given the output state of 100 after application of the channel, we are faced with two choices of the initial state: $\overline{0}$ or $\overline{1}$ which would occur with probabilities $p\,(1-p)^2$ and $p^2(1-p)$, respectively. Choosing the outcome $\overline{0}$ given an output bit string s with $|s| = 1$, our choice will be incorrect at a rate of $3p^2(1-p)$. The total failure rate of the code from all outcomes is $3p^2(1-p) + p^3$ which suppresses the error rate of the encoded information relative to the unencoded information for this error channel below a certain error rate. This critical error rate, p_{th}, for the bit-flip channel and the bit-flip code is $1/2$. Below this critical rate, the bit-flip code has a lower failure rate than that of a single bit put through the same channel. Above this rate, encoding information with the bit-flip code fails at a higher rate than an unencoded bit. Note that this rate is completely dependent on the error-correcting code and the noise channel.

Formally, classical error correction can be viewed as embedding information into vectors within vector spaces constructed over the binary field $\mathbb{F}_2$ in a manner where the vectors embedding the information are sufficiently orthogonal that they remain distinguishable even in the presence of certain local transformations (single-bit errors for example). For the bit-flip code, the encoded states are vectors, $\overline{0} = (0\ \ 0\ \ 0)^T$ and $\overline{1} = (1\ \ 1\ \ 1)^T$, which are in the space $\mathbb{F}_2^3$. We will expand on this notion below.

2.3.1 Classical Linear Codes

A classical linear code can be defined as the null-space of an $(n-k) \times n$ parity-check matrix, H, or equivalently $\mathcal{C} = \{x \in \mathbb{F}_2^n \mid Hx = 0\}$. The distance d of the code is the number of bit-flip errors required to map one codeword to another. For example given a bit string under the influence of an error $x' = x + e$, we may determine the error syndrome by applying the parity-check matrix to the string giving $s = Hx' = He$. If the Hamming Weight of e is less than $(d-1)/2$, then map H between error configurations and syndromes is bijective and we can confidently apply a set of correction bit flips. Otherwise, faulty correction will be applied and the computation is corrupted. Equivalently, we may define a linear code as the row-space of an $k \times n$ generator matrix G such that $\mathcal{C} = \text{row}(G)$. Note that G is the dual of H which means that is the matrix of largest rank that obeys $HG^T = 0$ or, equivalently, $G = \text{Ker}\,(H)$.

An example of a classical error-correcting code is the Hamming $[7, 4, 3]$ code [40]. This code is an $[n, k, d]$ code encoding k bits of information into n classical bits with a distance of d between each of the k codewords. Therefore, this code can correct against single bit flips. The corresponding parity check and generator matrices for this code are:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{2.13}$$

Each row of the generator matrix is a codeword ($\mathbf{x}_i$) for the classically encoded information. Error correction is performed utilizing the parity check matrix. For all codewords, $H\mathbf{x}_i^T = (0 \quad 0 \quad 0)^T$. Given an single bit flip error (on bit 2 for instance), $\mathbf{e}_2 = (0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^T$ and will transform the first codeword (for example) in the following manner $\mathbf{y}_1 = \mathbf{x}_1 + \mathbf{e}_2 = (1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1)^T$. The application of the parity check matrix gives $H\mathbf{y}_1 = H(\mathbf{x}_1 + \mathbf{e}_2) = (0 \quad 1 \quad 0)^T$ which is known as the *error syndrome*. Because the Hamming weight of the error is $\leq (d-1)/2$, the syndrome can be used to reliably correct the error. For the Hamming code, the syndrome in this example corresponds to the second column of the parity check matrix, which is the bit requiring correction. The process of using the error syndrome to infer the underlying error in the encoded information and choosing a correction operation based on this inference is known as *decoding*.

Finally we need to introduce the concept of a *dual* code corresponding to classical linear error-correcting codes. Given the linear space of codewords for a classical error-correcting code $\mathcal{C} \in \mathbb{F}_2^n$, the associated codewords of the dual code linear code are $\mathcal{C}^\perp = \{x \in \mathbb{F}_2^n \,|\, \langle x|c\rangle = 0 \,|\, c \in \mathcal{C}\}$. The generator and parity check matrices for the original code, $\mathcal{C}$, construct the parity check and generator matrices, respectively, for the dual code.

2.4 Quantum Error Correcting Codes

Now, we shift focus on how to encode quantum information. Much like classical information can be seen as a choice of protected vectors within an n-bit vector space $\mathbb{F}_2^n$, quantum error correction can be viewed as embedding information within complex vectors spaces (known as Hilbert spaces in quantum mechanics) over an n-qubit space $\mathbb{C}_2^n$. While classical codes can be defined by the row space of binary matrices, quantum codes can compactly be defined by a set of mutually commuting observables (operators). The quantum code space is the space spanned by the eigenvectors of this set of observables; these codes are called *subspace stabilizer* *codes.* A canonical choice of operators are ones constructed from n-fold tensor products of the Pauli matrices (form a basis for all 2×2 complex matrices) which form a basis of $2^n \times 2^n$ complex matrices over the Hilbert space of the n qubits. The choice of observables is dictated by the *stabilizer formalism* [41], which utilizes group theory to compactly represent quantum states. It states that a subgroup S_n of the n-fold Pauli group over n qubits can uniquely represent a quantum state given that it satisfies the following conditions: 1.) S_n is an abelian group 2.) $-I$ is not an element of S_n. The elements of S_n are known as the *stabilizers* of the code and the quantum code space $\mathcal{C}$ ($\mathcal{C} \subset \mathbb{C}_2^n$) is the $+1$ eigenspace of the stabilizer group: $\mathcal{C} = \{\text{span}(\{\,|\psi_i\rangle\,\}) \,|\, s|\psi_i\rangle = +1|\psi_i\rangle \;\forall\; s \in S_n\}$ where the linear span is over all complex amplitudes ($\lambda_i \in \mathbb{C}$). The representation of the stabilizer group can be compact with the use of group theory in that (for groups constructed with elements of the Pauli group), only $\log_2(|S_n|)$ group generators are required to represent a stabilizer group of size $|S_n|$.

For example, consider the quantum analogue to the classical bit-flip code. In this code, the encoded basis states are $\left|\overline{0}\right\rangle = |000\rangle$ and $\left|\overline{1}\right\rangle = |111\rangle$ which form the basis vectors of the code space $\mathcal{C}$: $|\psi\rangle_L = \alpha\left|\overline{0}\right\rangle + \beta\left|\overline{1}\right\rangle$. The stabilizer group for this code is $S_3 \equiv \{III,\, ZZI,\, IZZ,\, ZIZ\}$ which can be represented by the group generators $\langle ZZI,\, IZZ\rangle$. Therefore, an n-qubit code has been compactly represented by a generating set of $n-1$ mutually commuting observables. Furthermore, it is sufficient to measure the generators of the stabilizer group to perform error correction. Errors that anticommute with the stabilizers will be detected by the stabilizer measurement and give insight into the correction required. To differentiate the encoded basis vectors within the code space, one more observable is necessary. This operator must commute with the stabilizer group but does not necessarily commute with every element of the group independently. Formally, this operator is within the *normalizer* of the stabilizer group $\mathcal{N}(S_n)$ and measurement of such an operator will differentiate the logical basis states. For the quantum bit-flip code, this operator is ZZZ which is known as the *logical operator* where $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are the $+1$ and -1 eigenstates of the logical operator, respectively. Note that this example encoded one logical qubit into three qubits giving a 2 dimensional code space but denser encoded schemes exist allowing for a code space of 2^{n-k} dimensions which would be stabilized by a generating group of size k. Constructing subspace stabilizer codes builds from the construction of classical linear error-correcting codes and is outlined below.

2.4.1 Calderbank-Shor-Steane Code Construction

There is a family of quantum error-correcting codes that are constructed from classical linear error-correcting codes known as Calderbank-Shor-Steane codes [42, 43]. This construction requires two classical linear error-correcting codes: C_1 with n, k_1, d_1 and C_2 with n, k_2, d_2. Given the condition that C_2 is a subcode of C_1 ($C_2 \subseteq C_1$ so $k_2 < k_1$), then the codes C_1 and $C_2^{\perp}$ can be used can be used to construct an $[\![n, k_1 - k_2, t]\!]$ *quantum* error-correcting code where $t = \min(d_1, d_2)$. The double brackets indicate that this is a quantum error-correcting code. The parity check matrices for C_1 and $C_2^{\perp}$ dictate the form of the Z-type and X-type stabilizers, respectively.

As an example, we will show how the $[7, 4, 3]$ Hamming code can be used to construct an error-correcting code known as the Steane code. The $[7, 4, 3]$ Hamming code will be the choice for C_1 and the dual code of the Hamming code, the $[7, 3, 4]$ code, will be chosen as C_2. However, $C_2^{\perp} = C_1$ which is again the $[7, 4, 3]$ code. A code with this property is called *self-dual.* These two classical linear codes are used to construct a $[\![7, 1, 3]\!]$ quantum error-correcting code. The parity check matrices for both codes is the parity check matrix for the classical Hamming code (equation 2.13) which construct the stabilizer generators of the stabilizer group of the code:

$$
\begin{aligned}
H(C_1) \rightarrow \langle \mathcal{S}_z \rangle &= \begin{pmatrix} Z & I & Z & I & Z & I & Z \\ I & Z & Z & I & I & Z & Z \\ I & I & I & Z & Z & Z & Z \end{pmatrix} \\
H(C_2^{\perp}) \rightarrow \langle \mathcal{S}_x \rangle &= \begin{pmatrix} X & I & X & I & X & I & X \\ I & X & X & I & I & X & X \\ I & I & I & X & X & X & X \end{pmatrix}
\end{aligned}
\tag{2.14}
$$

Generating the code space for the quantum error-correcting code can be done by the application of the elements of X-type stabilizer group ($\mathcal{S}_x$) to $n = 7$ qubits in the computational basis:

$$
|0\rangle_L = \frac{1}{\sqrt{|\mathcal{S}_x|}} \sum_{S \in \mathcal{S}_x} S|0\rangle^{\otimes n} = \frac{1}{\sqrt{|\mathcal{S}_x|}} \sum_{x \in \mathrm{row}(H(C_2^{\perp}))} |x\rangle \tag{2.15}
$$

where the subscript L denotes the *logical* qubit and $|\mathcal{S}_x|$ denotes the size of the stabilizer group. This results in the codeword:

$$
\begin{aligned}
|0\rangle_L = \frac{1}{\sqrt{2}^3} (&|0000000\rangle + |0001111\rangle + |0111100\rangle + |1101001\rangle + \\
&|1010101\rangle + |1011010\rangle + |0110011\rangle + |1100110\rangle)
\end{aligned}
\tag{2.16}
$$

and $|1\rangle_L = 1/\sqrt{|\mathcal{S}_x|} \sum_{x \in \mathrm{row}(H(C_2^{\perp}))} |x \oplus (2^n - 1)\rangle$ where $\oplus$ is the bitwise XOR. The lowest weight Pauli operator that changes the encoded computational basis state, $|0\rangle_L \leftrightarrow |1\rangle_L$, is the X logical operator $X_L = X_1 X_2 X_3$. The Z logical operator is $Z_L = Z_1 Z_2 Z_3$ and Y_L is (up to a global phase) obtained by the relation $Y_L = X_L Z_L$. Moving forward, any time we refer to error-correcting codes we will be referring to *quantum* error-correcting codes unless specified otherwise.

In addition to encoded Pauli operations, there are additional encoded operations that preserve the encoding of the Steane code, known as transversal operations. Because the Steane code was constructed from a self-dual linear error-correcting code, the X-type and Z-type stabilizer generators have the same structure. Therefore, the application of the operator $H^{\otimes 7}$ preserves the stabilizer group of the code under this Clifford transformation, $H^{\otimes 7} \mathcal{S}_x (H^{\otimes 7})^{\dagger} = \mathcal{S}_z$, which naturally preserves the encoded and is there a transversal operation. This is the case for all self-dual quantum error-correcting codes. The Phase gate:

$$S = R_z\left(\frac{\pi}{2}\right) = \begin{pmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = e^{-i\frac{\pi}{4}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} \tag{2.17}$$

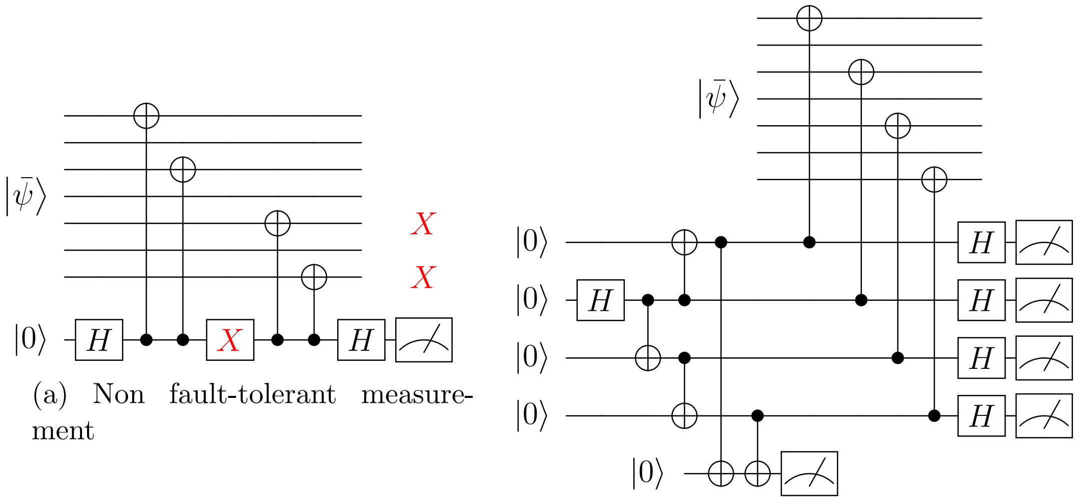
is also a transversal gate. Consider the action of the gate on the logical state $|0\rangle_L$:

$$S^{\otimes 7}|0\rangle_L = \sum_{x \in \mathrm{row}(H(C_2^{\perp}))} S^{\otimes 7}|x\rangle = \sum_{x \in \mathrm{row}(H(C_2^{\perp}))} e^{i\frac{\pi}{2}|x|}|x\rangle = |0\rangle_L \tag{2.18}$$

where $|x|$ is the Hamming weight of the binary string. Note that this occurs because, for the elements of the logical state, $|x| \bmod 4 = 0$. Codes with this property are known as *doubly even* codes and admit a transversal phase gate. The final transversal operation is the $CNOT$ gate. Again, we will use the structure of the stabilizer group to understand why. Consider the product space between two logical qubits encoded with the Steane code (a Hilbert space over 14 qubits). Then a generating set for the stabilizer group for the product space is the following: $\langle \mathcal{S}_x \otimes I^{\otimes 7}, \mathcal{S}_z \otimes I^{\otimes 7}, I^{\otimes 7} \otimes \mathcal{S}_x, I^{\otimes 7} \otimes \mathcal{S}_z \rangle$. The $CNOT$ transforms the elements of the stabilizer group according to the Clifford transformation $\{XI, IX, ZI, IZ\} \rightarrow \{XX, IX, ZI, ZZ\}$. The transversal $CNOT$ is

$$\overline{CNOT} = \bigotimes_{i=1}^{n} CNOT(i, i+n) \tag{2.19}$$

where the line over the operator indicates an encoded operation and n is the number of qubits in one of the encoded states ($n = 7$ in this example). This operator will transform the stabilizer group of the product system as $\langle \mathcal{S}_x \otimes \mathcal{S}_x, \mathcal{S}_z \otimes I^{\otimes 7}, I^{\otimes 7} \otimes \mathcal{S}_x, \mathcal{S}_z \otimes \mathcal{S}_z \rangle$ which is an equivalent generating group for the two-encoded qubit stabilizer group. Given that all CSS codes have a stabilizer groups with this structure, this family of codes admits a transversal $CNOT$. Note that the transversal gates allowed for the Steane code correspond to the Clifford group, which is not a universal gate set. Any subspace stabilizer code that has geometrically local stabilizer generators cannot admit a transversal, universal gate set [34]. To ob-



(a) Non fault-tolerant measurement

(b) Fault-tolerant measurement

Figure 2.3: Circuits implementing the measurement of the $XIXIXIX$ stabilizer of a Steane encoded qubit, $|\bar{\psi}\rangle$. (a) Circuit implementing a non fault-tolerant stabilizer measurement as a single X error on the ancilla will propagate to an uncorrectable error on the data for a distance-3 code. (b) Shor style syndrome measurement [1] that is robust to the hook error.

tain universality in subspace stabilizer codes, other methods must be implemented which is the topic of chapters 4 and 6.4.

To conclude this section, we need to discuss how to perform error correction. We cannot directly measure the state of encoded information without collapsing it to a classical state, which removes the quantum-ness of the information. Instead, we must use entangling gates and additional qubits (called *ancillary* or *ancilla qubits*) to measure specific degrees of freedom of the information in a manner that does not collapse the wavefunction. Errors are detected due to the manner which the $CNOT$ gate transforms the error. As we have seen above $CNOT : \{XI, IX, ZI, IZ\} \rightarrow \{XX, IX, ZI, ZZ\}$ so the entangling gate can be used to propagate the error to the ancillary state for measurement. The wiring of the control and target qubit indicates whether an X-type or Z-type error will propagate to an ancillary qubit which can then be measured. The relation $Y = iXZ$ allows for Y-type errors to be corrected as well. Measurements must be performed in a manner that preserves the logical state, which indicates that it must be an eigenstate of any operator measured. Therefore, the operators that will be measured will be the stabilizers themselves. Figure 2.3a shows how to measure the $XIXIXIX$ stabilizer of the Steane code which will be read out as a classical bit, 1, if there is a Z-type error that anticommutes with the stabilizer; otherwise, 0 is read out. The Z-type stabilizer with this form is measured with a similar circuit with the direction of the $CNOT$s flipped and removing the Hadamard (H) gates from the ancillary bit ($|0\rangle$). The collection of stabilizer measurements gives again the *error* syndrome that (similar to the classical codes used to construct the quantum code) is used to decipher a correction operation. For the quantum code, there are two parity check matrices $H(C_1)$ and $H(C_2^\perp)$ which can be used to correct X and Z errors, respectively.

For error correction to prevail, errors that occur must not spread in a malignant manner; i.e. do not create logical errors during computation. If a code prevents this malignant spreading of errors during computation, the computation is said to be *fault-tolerant* which will reflect in the lower error rate of the encoded operations relative to an unencoded operations at low enough physical error rates, $p_L < p_{th}$ [44–49]. This critical error rate, p_{th} is known as the error threshold. Similar to the classical bit flip code example, encoding information in a quantum error-correcting code fails a lower rate relative to an unencoded qubit only below this threshold; otherwise, encoded information fails at a higher rate than a single qubit. Again, this threshold is completely dependent on the error-correcting code and noise model. To illustrate what is meant by a malignant error, consider the case in figure 2.3a where an X error occurs on the ancillary qubit that propagates to two X errors on the data; known as a "hook error". This error will not be detected by the stabilizer measurement but it will also be detected by the other two Z stabilizers indicating a correction operation of X_6. In this case, a single-qubit error has now become three errors in the code, which (modulo stabilizers) has actually applied a logical operator to the code: a *logical error*. At this time, the user has lost track of the state of the encoded information and so the computation is ruined. This error has the encoded information failing at a rate equivalent to an unencoded qubit and is therefore not fault-tolerant so a protocol must be developed to make the computation robust to these errors. One method is to use a many-qubit ancillary state in a superposition state (known as a *cat* state) to control hook errors; an implementation of Shor style ancilla [1]. Verification of the cat state with a fifth ancillary qubit and measurements of the many-qubit state can be used to decipher when hook errors occur and allow for them to be corrected. The Shor method of stabilizer measurement is one of many fault-tolerant methods of measuring stabilizers [1, 50–53]. They all share one trait: implementation requires an increase in the number of ancillary qubits/gates necessary to perform robust computation. The trade off between implementing robust routines versus quantum resource (qubits in this case) efficiency is discussed in part in chapter 4.

2.4.2 Topological Quantum Error Correcting Codes

A class of quantum error-correcting codes of interest due to their high error thresholds are topological quantum error-correcting codes. These codes are constructed by a cellulation of a surface. The choice of cellulation is a representation of the stabilizer group and properties of the surface dictate the density of encoding (number of logical qubits per surface) of the code. Formally, boundary maps of the homology and cohomology groups of the surface cellulation define the qubits, stabilizer group, and logical operators. The number of logical operators (density of encoding) is dictated by the genus of the surface. An example of such a code is the surface code, which is constructed from a square lattice cellulation of a plane with 2 types of boundaries as shown in figure 2.4. Edges represent qubits. Vertices and faces represent stabilizer generators of the code. The parity check matrix of the classical code C_1 ($C_2^{\perp}$) constructing the surface code can be generated by an $n \times m$ matrix where the columns correspond to edge labels and the rows correspond to face (vertex) labels where 1 values occur when an edge is contained with the face

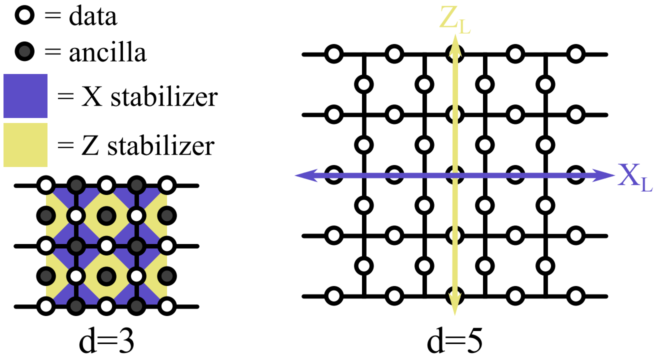


Figure 2.4: Representation of two members of the surface code family. (a) The distance-3 surface code fully labeled. (b) Equivalent representation of a distance-5 surface code where the labels and ancilla are implied. The logical operators highlighted are any weight d string (modulo stabilizers) spanning from two equivalent boundary types.

(vertex). The construction of the code space then naturally follows from section 2.4.1. The normalizer of the stabilizer group (the logical operators) for surface codes has an intuitive representation as the minimum weight Pauli operator that spans from two different boundaries of the same type; that is, boundaries that terminate with the same stabilizer type as shown in figure 2.4. Therefore, the distance of the code (for a square lattice) is the number of data qubits spanning across a boundary.

To understand error correction on the surface code, we will introduce the concept of error chains. Errors will accumulate on the surface code. Due to the regular tiling of the stabilizer generators on the lattice, the endpoints of the error strings are all that is detected by the stabilizer measurements; giving the error syndrome. The error strings can have endpoints at two stabilizers or endpoints at a stabilizer and a boundary of the same type (X-type or Z-type). Error chains must be undone or converted into loops within the lattice, which are products of stabilizer generators, for error correction. If an error chain plus the correction operation results in an error string connecting the boundaries, then a logical error occurs and the computation is corrupted. Intuitively, a string of $(d-1)/2$ errors from a boundary can be reliably corrected back to the string-terminating boundary, but a $(d-1)/2+1$ error chain would be inferred an a $(d-1)/2$ error from the opposing boundary invoking a logical error after correction. Similarly, a length $(d-1)/2$ internal string will be corrected, but a $(d-1)/2+1$ internal string would be corrected as two correction strings from each boundary with a total distance of $(d-1)/2$ between both correction strings. Classical decoding algorithms have been developed to infer the underlying error chains given the syndrome ranging between optimal and computationally efficient.

2.5 Monte Carlo Sampling of Clifford Circuits

Due to the exponential scaling of classical variables required to track the time evolution of a full quantum state, classical simulations of quantum algorithms extending beyond around 20 qubits is difficult. However, most quantum algorithms and error correction routines require ≥ 20 qubits. Luckily, there is a subclass of quantum circuits that are efficiently simulable classically allowing for assessment of larger scale algorithms and error correction protocols; given by the Gottesman-Knill theorem [34]. This subclass of quantum circuits contain Clifford gates and Pauli basis measurements, often known as stabilizer simulations because they restrict to transformations from/to elements within the Pauli group. For a majority of this thesis, we relied on Monte Carlo sampling of Clifford circuits to obtain error rates for various quantum error-correcting codes.

Chapter 3

Simulating the performance of a distance-3 surface code in a linear ion trap

This chapter contains results from:

Colin J. Trout, Muyuan Li, Mauricio Gutierrez, Yukai Wu, Sheng-Tao Wang, Luming Duan, Kenneth R Brown, ”Simulating the performance of a distance-3 surface code in a linear ion trap”, (submitted: NJP-107838).

Experimental implementations of small quantum algorithms in systems containing under 10 qubits have been exhibited in a variety of architectures [16–26]. However, quantum error correction requires 10-20 qubits for encoding schemes with depolarizing error thresholds are on the order of experimental error rates obtainable in the foreseeable future. An enticing selection for an error correction protocol is the surface code [54] which exhibits an error correction threshold in the circuit model between $0.5\% - 1\%$ for depolarizing Pauli noise [55–58]. Here we examine the distance-3 surface code implemented with nine data qubits and eight ancilla qubits, known as surface-17 [2, 59, 60]. In principle only a single ancilla qubit could be used over and over, but the gains from parallelism are even apparent in studies comparing 8 ancilla qubits to 6 ancilla qubits [2]. With 10-20 qubits, a number of QEC codes can be implemented fault-tolerantly including the 5-qubit code [61], Steane [[7,1,3]] [62, 63], Bare [[7,1,3]] [64], the Bacon-Shor [[9,1,3]] [65, 66], or the twisted surface [67] code. We chose to study the surface code because the memory pseudothreshold is superior to the 5-qubit code, the Steane code, and the Bare code and comparable to the Bacon-Shor and twisted surface code [67].

Our choice of a physical architecture is a linear chain of ultracold trapped atomic ions. Atomic ions have proven to be high-fidelity qubits for quantum information processing. Single-qubit gates are performed by the application of electromagnetic radiation with lasers [68] or microwaves [69, 70]. Two-qubit gates are performed by conditionally exciting the coupled motion of ions in the chain dependent on the ion's internal states [71–74]. These normal modes of motion are nonlocal allowing interactions between any ions in the chain without requiring additional overhead from moving information through local couplings [75] or storing qubits in auxiliary states [76]. This arbitrary connectivity without altering the intrinsic nature of the qubit adds modularity at the hardware level which relaxes software constraints on compilation when building up high-level algorithms from the hardware primitives [26, 77].

Atomic ion experiments have already demonstrated classical error correction [78, 79], encoding logical states for quantum error correction [80], and fault-tolerant quantum error detection [81]. Single-qubit operations in this system have displayed error rates below the surface code pseudothreshold [82, 83] reported from Tomita and Svore [2]. Two-qubit gate fidelities are limited by a number of factors including spontaneous Raman scattering during gates and residual entanglement between the internal state and the motional modes of the ion. Compensation pulses have been developed with a predicted error rate due to scattering of 10^{-4} [84] and control sequences have been implemented exhibiting single- and two-qubit gate fidelities of 99.9% using the hyperfine ground states of trapped ^{9}Be^{+} [85] and ^{43}Ca^{+} [86] ions, respectively. However, quantum control applied to a scaled-up five-ion chain consisting of ^{171}Yb^{+} qubits currently exhibits two-qubit error rates of 2% [26].

This study provides an assessment of the feasibility of implementing the surface-17 error-correcting code on a linear trap holding a chain of ^{171}Yb^{+} ions. Furthermore, this study provides target fidelities for experimentalists to realize error correction with the 17-qubit surface code. This chapter is structured in the following manner. First, we explain the 17-qubit surface code and how atomic ions are implemented as qubits. Following that, the ion trap architecture will be defined and a map between the linear ion chain and the two-dimensional surface code is provided. The remainder of the paper will focus on error correction. Efficiently simulable models of ion trap error sources will be outlined followed by an examination of results from decoding methods tailored for such errors.

3.1 The 17-Qubit Surface Code

The 17-qubit surface code is shown in figure 3.1a [60]. It is a distance-3 surface code that can be implemented in parallel with less qubits ($n = 17$) than the standard surface code ($n = 25$). It is constructed from removing the corner data qubits and stabilizers that act upon them from the original surface code shown in figure 2.4. The white (black) circles represent data (ancilla) qubits and the dark (light) faces of the lattice dictate the X-type (Z-type) stabilizer generators of the code. These codes use single-qubit ancilla for fault-tolerant measurement of the stabilizer generators, which reduces resource requirements compared to other fault-tolerant protocols. The scheduling of the two-qubit gates following an N-like pattern about the face of a weight-4 Z stabilizer directs hook errors in a direction perpendicular to the direction of logical Z operator as shown in figures 3.1b and 3.1c. This error is equivalent to a single-qubit error *from the perspective of the Z logical operator*, thus retaining fault-tolerance during syndrome measurement.

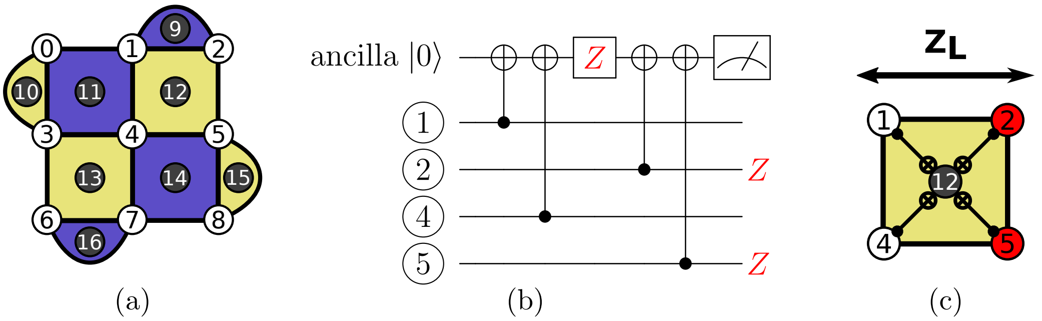


Figure 3.1: (a) The 17-qubit surface code. (b) Typically nonfault-tolerant stabilizer measurement. (c) The ancillary error from (b) propagates in a direction perpendicular to the direction of the logical operator which is equivalent to a single-qubit error *from the perspective of the logical operator* providing fault-tolerance. All images were adapted from Tomita and Svore [2].

Many other error correction routines require the use of many-qubit ancillary states to ensure fault-tolerance, which increases the number of qubits required for error correction [1, 50–53].

3.2 Atomic Ions as Qubits

Qubits can be encoded into either optical states [87], Zeeman states [88, 89], or hyperfine states of ions [85, 86, 90]. For this study, information was stored in the hyperfine "clock" states of $^{171}\mathrm{Yb}^{+}$. Single-qubit gates are performed by the application of laser fields [68] or microwave radiation [69, 70] to manipulate the hyperfine states of trapped $^{171}\mathrm{Yb}^{+}$ ($^{2}\mathrm{S}_{1/2}|F=0;m_F=0\rangle \leftrightarrow {}^{2}\mathrm{S}_{1/2}|F=1;m_F=0\rangle$ transition) which can drive arbitrary single-qubit rotation gates. High-fidelity, fast two-qubit gates are performed by the application of counter-propagating laser fields achieving entanglement through the coupling of the internal states with the motional modes of the ion crystal through a method known as the Mølmer-Sørensen gate which engineers an XX entangling gate [26, 91, 92]. Controlled-NOT ($CNOT$) gates can be built from Mølmer-Sørensen gates and available single-qubit rotations [93]. State initialization and measurement are performed by applying laser beams resonant with the $^{2}\mathrm{S}_{1/2} \leftrightarrow {}^{2}\mathrm{P}_{1/2}$ transition. For $|0\rangle$ state preparation, qubits are optically pumped out of the $^{2}\mathrm{S}_{1/2}|F=1\rangle$ state into the $^{2}\mathrm{P}_{1/2}|F=1\rangle$ manifold which, with high probability, falls into the $^{2}\mathrm{S}_{1/2}|F=0\rangle$ state [94–96]. For measurement in the Z-basis, a $^{2}\mathrm{S}_{1/2}|F=1\rangle \leftrightarrow {}^{2}\mathrm{P}_{1/2}|F=0\rangle$ cycling transition is induced where the discrepancy between scattered photon counts of the qubit states serves as readout [94–96].

To perform error correction with the surface code, the required operations are single-qubit gates (H), two-qubit gates ($CNOT$), state initialization ($|0\rangle$ state), and measurement (Z-basis). Note that the state preparation and measurement processes scatter photons that should not interact with surrounding ions. This requirement introduces an additional operation, ion shuttling [97–102], which will

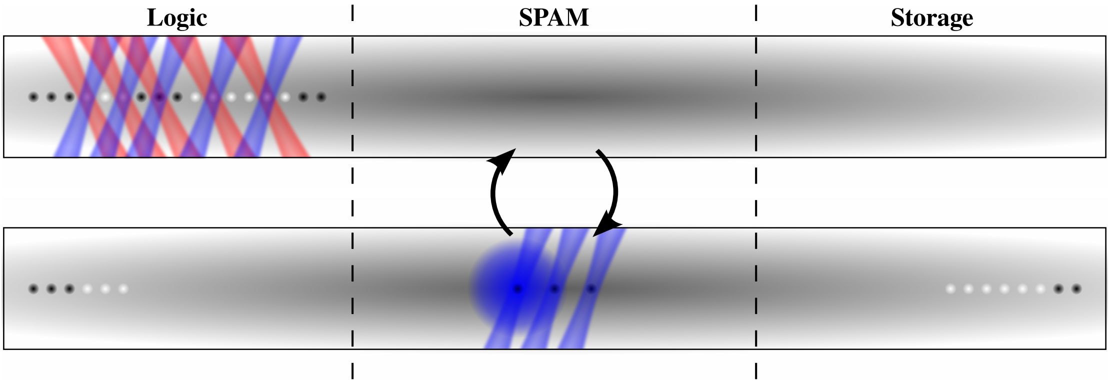


Figure 3.2: Ion trap architecture and syndrome measurement operations. Ions are trapped in a mixed data/ancilla qubit arrangement labeled by white/black dots (see figure 3.1a). There are three zones: a Logic, SPAM, and Storage zone. (Top) The Logic zone where qubit gates are applied. (Bottom) The SPAM zone where state preparation and measurement is performed, scattering photons. The Storage zone serves to sufficiently distance qubits from the SPAM zone.

be used to separate qubits in memory from the scattered photons during measurement/preparation.

3.3 Mapping the Surface Code to an Ion Chain

For a first generation implementation of a logical qubit consisting of atomic ions, a trapped linear chain of ions was favored over two-dimensional architectures due to technological challenges in the latter that result in issues such as additional ion heating through shuttling junctions in traps [97, 98, 101, 102], high idle ion heating rates [103], and single-ion addressing/readout issues in two-dimensional trap layouts. The linear trap is composed of at least three zones: a Logic, State Preparation and Measurement (SPAM), and Storage zone (figure 3.2). Ion shuttling across the axial direction of the trap allows for the 17-ion chain to be arbitrarily split into three separate linear chains of ions inhabiting each of the three zones. The Logic zone is where all single- and two-qubit gates are applied. The central SPAM zone is where state preparation and measurement operations are performed. The Storage zone serves the purpose that its name implies and is required due to the geometric constraint of having the ions confined in a linear chain. Such an implementation begs the question: how should the qubits in the surface code be assigned to the linear chain of ions? We are particularly interested in configurations that minimize the gate times (errors) of the error correction circuit. To proceed, we must first discuss two-qubit gates.

For computation of the two-qubit gate times, current gate protocols [26] and motional decoupling techniques [104] were modeled; the latter of which contributes significantly to the distance dependence of the gate times. The calculation of the gate time of an ion pair is outlined below. In the weak trap limit, a Paul trap can be well approximated by a pseudo-harmonic potential (see e.g. Ref. [105]). Here we consider ions in a linear Paul trap along the z direction ($\omega_z \ll \omega_x$, ω_y). With a harmonic trap potential, the spacing between ions in the chain will be nonuniform, which can lead to undesired transition into a zigzag shape [106, 107], as well as the difficulty in cooling many low frequency modes. To overcome this problem, an additional quartic potential can be added to the z direction [108] giving the total potential energy:

$$V = \sum_i \left(-\frac{1}{2}\alpha_2 \, z_i^2 + \frac{1}{4}\alpha_4 \, z_i^4 \right) + \sum_{i<j} \frac{e^2}{4\pi\epsilon_0 \, |z_i - z_j|} \tag{3.1}$$

where α_2, $\alpha_4 > 0$ are two parameters characterizing the strength of the quadratic and the quartic potentials. The ion configuration is then fully determined by a length unit $l_0 \equiv \left(e^2/4\pi\epsilon_0\alpha_2\right)^{1/3}$ and a dimensionless parameter $\gamma_4 \equiv \alpha_4 l_0^2/\alpha_2$. For $N = 17$ $^{171}\mathrm{Yb}^+$ ions, we choose $\gamma_4 = 0.86$ to minimize the relative standard deviation of the ion spacings. An average ion distance of about $8.2\,\mu$m can then be realized by setting $l_0 = 25\,\mu$m.

The two-qubit entangling gate is implemented with a spin-dependent force on the two ions via the transverse collective modes. For example, we can use the transverse modes in the x direction whose k-th normalized mode vector is denoted as $\mathbf{b}_j^k$ with a mode frequency ω_k where the index j runs over all ions ($j = 1, 2, \cdots, N$). The creation and annihilation operators corresponding to this collective mode are denoted as $\hat{a}_k^\dagger$ and $\hat{a}_k$ respectively. The transverse trap frequency is set to a typical value $\omega_x = 2\pi \times 3\,\mathrm{MHz}$ and the temperature is set to $k_B T = \hbar\omega_x$ giving an average phonon number of $\bar{n} \approx 0.5$ for each transverse mode. This can be easily achieved with a Raman sideband cooling. The spin-dependent forces are generated by counter-propagating laser beams on the two ions that we choose to entangle. The Hamiltonian, in the interaction picture, can be represented as:

$$H_I = \hbar \sum_j \tilde{\Omega}_j \sum_k \eta_k \mathbf{b}_j^k \sin\mu t \left(\hat{a}_k e^{-i\omega_k t} + \hat{a}_k^\dagger e^{i\omega_k t}\right) \hat{\sigma}_j^x \tag{3.2}$$

where we further define the Lamb-Dicke parameter $\eta_k \equiv \Delta k\sqrt{\hbar/2m\omega_k}$, Δk is the difference in the wavevectors of the counter-propagating Raman beams, μ is the two-photon detuning, and $\hat{\sigma}_j^x$ is the $\hat{\sigma}^x$ Pauli matrix on ion j. For the $^{171}\mathrm{Yb}^+$ qubit transitions, the laser beams have wavelengths around $\lambda = 355\,\mathrm{nm}$ [109] and for counter-propagating pairs $\Delta k = 2k$, hence the Lamb-Dicke parameter $\eta_k \approx 0.111$. In the above equation, $\tilde{\Omega}_j$ is the effective Rabi frequency of the Raman transition pairs ($\tilde{\Omega}_j \approx \Omega_1\Omega_3/\Delta = \Omega_1\Omega_2/\Delta$ where Δ is the single-photon detuning from the excited state). From now on we will drop the tilde notation for simplicity. Note that one of the laser beams contains two frequency components and we assume that the two Raman transition pairs have the same effective Rabi frequency Ω_j, opposite detunings $\pm\mu$, and opposite wavevector differences $\pm\Delta k$. This is known as the phase-insensitive geometry [110].

The time evolution under the above Hamiltonian can be written as [108, 110]:

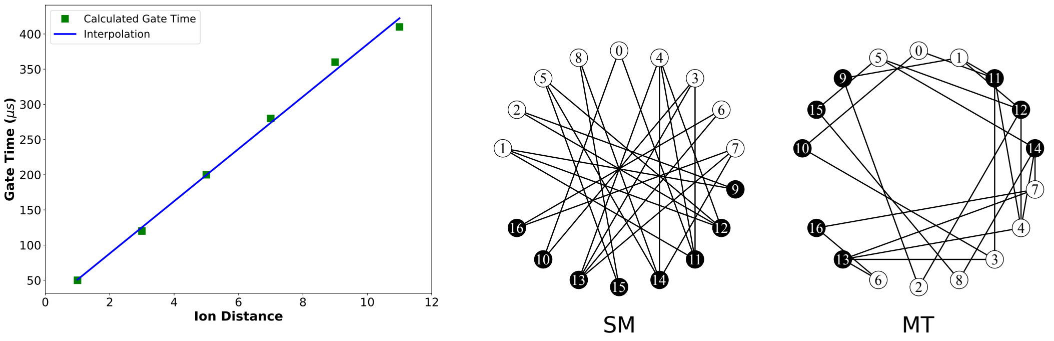
$$\hat{U}_I(\tau) = \exp\left(i\sum_j \hat{\phi}_j(\tau)\hat{\sigma}_j^x + i\sum_{i<j}\Theta_{ij}(\tau)\hat{\sigma}_i^x\hat{\sigma}_j^x\right) \tag{3.3}$$

where $\hat{\phi}_j(\tau) = -i\sum_k[\alpha_j^k(\tau)\hat{a}_k^\dagger - \alpha_j^{k*}(\tau)\hat{a}_k]$. The parameters α_j^k and Θ_{ij} are purely numbers related to the phase space displacement of the motional states after the gate and angle of the entanglement gate, respectively. For the following calculations, we assume that Ω_j is the same for both ions and we divide it into segments with equal durations; that is, a piecewise constant $\Omega(t)$. With a suitable choice of detuning μ, gate time τ, and Rabi frequency $\Omega(t)$, we can suppress all the $\alpha_j^k(\tau)$ terms and realize an ideal entangling gate $e^{\pm i\pi\hat{\sigma}_i^x\hat{\sigma}_j^x/4}$ with high fidelity. Here, we focus on the intrinsic gate infidelity caused by the residual coupling to multiple phonon modes after the entangling gate. Other noise sources from technical control errors are not included for this calculation.

We fit the two-qubit gate times as a result of the ion distance (figure 3.3a) to a linear function yielding:

$$t_g = 10 + 38d \tag{3.4}$$

where t_g is the gate time (μs) and d is the ion distance. Single-qubit gates can be performed in parallel with a gate time of 10 μs. The ion splitting time was assigned to be 100 μs between neighboring zones [99, 100, 111]. This time is built from an assumption of a 200 kHz lowest axial frequency implying that splitting/merging of subsets of ions in the ion chain can occur at a rate almost at this frequency. The remaining 95 μs allows for the chains to be separated by a distance of 665 μm assuming a 50 kHz update rate in the transport waveforms [102], which is excellent separation between the detection lasers and the data qubits. We assume operations can happen in parallel allowing subchains to be shuttled away from the detection



(a) Gate times with respect to ion distance

(b) Qubit to ion maps

Figure 3.3: (a) Two-qubit gate times calculated as a function of the ion distance in the linear chain. Extrapolation of this data was used to calculate the time required to measure the error syndrome. (b) Ion chains mapped to a circle. The nodes correspond to the qubit numbers in figure 3.1a. The labels refer to different optimizations with configurations shown in figure 3.1.

Table 3.1: Trap operation times and ion arrangements optimized for an array of parameters. The first letter of the label refers to S=separate and M=mixed arrangements of data and ancilla qubits. The second letter of the label refers to the parameter minimized with M=maximum distance between entangled ions, A=average distance between entangled ions, and T=parallel total gate time. All values are reported in microseconds and the numbers in roman and *italics* refer to the gate time of the operations performed in serial and parallel, respectively. Parallel operations allow for two simultaneous two-qubit gates exciting the independent x and y radial modes and fully parallel single-ion operations. Single-qubit gates, parallel measurement/state preparation, and shuttling between neighboring zones require 10 μs, 100 μs, and 100 μs (5 μs split and 95 μs shuttle time), respectively.

Label	Logic	Shuttle	SPAM	Total	Ion Ordering
SM	7650 *3920*	200	100	7950 *4220*	1 2 5 8 0 4 3 6 7 **9** **12** **11** **14** **15** **13** **10** **16**
SA	7240 *4140*	200	100	7640 *4440*	0 2 6 8 1 4 3 7 5 **11** **12** **10** **15** **13** **14** **9** **16**
MM	3080 *1690*	1200	500	4780 *3390*	5 **15** 2 **12** **14** **9** 8 1 4 7 **11** 3 **13** **16** 0 **10** 6
MA	2300 *1170*	1800	800	4900 *3770*	2 **9** 1 **12** 5 **15** 8 **14** 4 **11** 0 **10** 3 **13** 7 **16** 6
MT	4300 *2320*	700	300	5300 *3320*	**10** **15** **9** 5 0 1 **11** **12** **14** 7 4 3 8 2 6 **13** **16**

zone, parallel joining and splitting operations, and three way ion chain splitting. The measurement time was also fixed to 100 μs which is a lax requirement on the experimental apparatus and will allow for high fidelity state detection [83].

The underlying connection graph of a trapped linear ion chain is a fully connected graph [112]. Therefore, there are many ways to map the surface-17 code to the linear ion chain. With this relationship between ion distance and gate times, we screened for the optimal ion chain configurations using a simulated annealing algorithm that minimized several parameters of interest. Three parameters were minimized: the maximum ion distance between entangled ions (M), the average ion distance between entangled ions (A), and the total time for one round of syndrome measurement in parallel (T) corresponding to the second letter in the labels in figure 3.1. In addition, the optimizations were performed with constraint that the data and ancilla qubits are separate (S) and are allowed to be mixed together (M) corresponding to the first letter in the labels in figure 3.1. The corresponding connection graphs for two optimized chains (SM and MT) are shown in figure 3.3b. The time required to measure the error syndrome for different optimized configurations are shown in figure 3.1. The parallel operation times are shown in italics in figure 3.1 allowing two simultaneous entanglement gates to be performed on two independent pairs of ions by exciting the x and y radial modes, respectively, for each pair. For the calculations below, we chose the ion chain configuration that gives the minimal total syndrome measurement time (serial or parallel), MT.

3.4 Modeling Ion Trap Error Sources

This section provides the components for building up a noise model representative of an ion trap architecture. A graphical representation of the full ion trap error model is shown in figure 3.4.

3.4.1 Circuit-Level Depolarizing Error Model

The depolarizing error model is a standard error model used in simulations of quantum error-correcting codes. After the application of each gate in the quantum circuit implemented to measure the stabilizers, an element is sampled from the one-qubit (two-qubit) Pauli group and applied after each single-qubit (two-qubit) gate. The one- and two-qubit Kraus channels are of the form:

$$\begin{aligned} E_{1,d} &= \left\{\sqrt{1-p}\,I,\ \sqrt{\frac{p}{3}}X,\ \sqrt{\frac{p}{3}}Y,\ \sqrt{\frac{p}{3}}Z\right\} \\ E_{2,d} &= \left\{\sqrt{1-p}\,II,\ \sqrt{\frac{p}{15}}IX,\ \sqrt{\frac{p}{15}}IY,\ \sqrt{\frac{p}{15}}IZ,\ \sqrt{\frac{p}{15}}XX,...,\ \sqrt{\frac{p}{15}}ZZ\right\} \end{aligned} \tag{3.5}$$

where p is the *error rate* of the error channel. The stabilizer circuits in this work are built using only the $CNOT$ as the two-qubit gate. The circuit-level implementation of this error model allows for errors on both the data and ancilla qubits, which translate into errors in the measurement of stabilizers during syndrome extraction. Furthermore, preparation and measurement errors are modeled by the application of a single-qubit depolarizing error channel after preparation gates and before measurement. This model will serve as a baseline error model for assessment of error correction.

3.4.2 Coherent Over-Rotation of the Mølmer-Sørensen Gate

The Mølmer-Sørensen (MS) gate [91, 92] was chosen as the entangling gate due to its faster gate times and higher gate fidelities relative to other entangling gate schemes [26]. The MS gate uses a bichromatic laser field to induce a two-photon transition that couples the $|00\rangle \leftrightarrow |11\rangle$ and $|10\rangle \leftrightarrow |01\rangle$ qubit states. The MS gate induces a transition with a bichromatic laser tuned close to the upper and lower motional sideband of a qubit transition [91, 92]. In the computational basis, the unitary operator associated with the Mølmer-Sørensen gate is:

$$XX(\chi) = \begin{pmatrix} \cos(\chi) & 0 & 0 & -i\sin(\chi) \\ 0 & \cos(\chi) & -i\sin(\chi) & 0 \\ 0 & -i\sin(\chi) & \cos(\chi) & 0 \\ -i\sin(\chi) & 0 & 0 & \cos(\chi) \end{pmatrix} \tag{3.6}$$

where the parameter χ depends on the gate time applied to the specific ion pair [26]. The absolute value of the angle, $|\chi|$, may be set to any real number between 0 and $\pi/2$ by varying the power of the laser in the experiment [26]. The sign of χ is dependent on the laser detuning which is chosen from normal modes of the ion pair [26]. The Mølmer-Sørensen unitary implemented during the $CNOT$ can equivalently be written as:

$$XX(\chi) = \exp(-i\,\chi\,XX) = \cos(\chi)\; II - i\sin(\chi)\; XX \tag{3.7}$$

where we attempt to assign χ as $\pi/4$ with the laser field. However due to experimental error, a small over-rotation (with angle α) may be applied about the XX axis with the real gate applied in equation 3.7 having an angle of $\chi+\alpha$. This error will be simulated by a probabilistic error channel of the form:

$$E_{2,xx} = \left\{\sqrt{1-p_{xx}}\,II,\ \sqrt{p_{xx}}\,XX\right\} \tag{3.8}$$

where the probability of the channel above is a function of the over-rotation angle. For example, one possible relation between p_{xx} and α is obtained by the Pauli twirled approximation, which results in $p_{xx} = \sin^2(\alpha)$ [113]. It is also possible to choose p_{xx} such that the Pauli approximation to the over-rotation satisfies additional constraints [114, 115]. Furthermore, the single-qubit rotation gates in the circuit can also suffer over-rotations typically to a much less degree. The over-rotations can be modeled in an analogous way giving three distinct gate-dependent error channels:

$$\begin{aligned} E_{1,x} &= \left\{\sqrt{1-p_x}\, I,\ \sqrt{p_x}\, X\right\} \\ E_{1,y} &= \left\{\sqrt{1-p_y}\, I,\ \sqrt{p_y}\, Y\right\} \\ E_{1,z} &= \left\{\sqrt{1-p_z}\, I,\ \sqrt{p_z}\, Z\right\} \end{aligned} \tag{3.9}$$

which are applied after every single-qubit rotation gate $R_X(\theta)$, $R_Y(\theta)$, and $R_Z(\theta)$, respectively. For simulations, the error rates for the single-qubit gates are a factor of 10 lower than those corresponding to two-qubit gates; representing observed single- and two-qubit gate fidelities [85, 86].

3.4.3 Motional Mode Heating

In addition to control errors, the applied field from the Mølmer-Sørensen gate can result in motional heating of the ions, which impacts the fidelity of the two-qubit gate. Modeling heating as a coupling of the motional states of the ions to an infinite temperature bath [116], Ballance et al. characterized the impact of motional heating on the error rate of a two-qubit entangling gate, ϵ_h, giving:

$$\epsilon_h = \frac{\dot{n} t_g}{2K} \tag{3.10}$$

where $\dot{n}$ is the average change in thermal occupation number of the gate mode, t_g is the gate time, and K is the number of loops in phase space traversed by the ions during the gate [86]. We chose to study the low K limit ($K = 1, 2$) of equation 3.10 modeling heating errors with the Kraus operators which are applied after every MS gate:

$$E_{2,h} = \left\{\sqrt{1-p_h}\, II,\ \sqrt{p_h}\, XX\right\} \tag{3.11}$$

where the probabilities are ion-dependent: $p_h = (r_{heat}) \times (t_{MS})$ where r_h is the heating rate and t_{MS} is the time of the Mølmer-Sørensen gate. It is important to note that this model is pessimistic with respect to ion heating, even in the low K limit, and the choice of coupling modes can increase K by $1-2$ orders of magnitude [117, 118].

3.4.4 Background Depolarizing Noise

For the stable "clock" states of the hyperfine qubits, errors arise from the application of gates. In addition to systematic over/under-rotations of the applied laser field, instabilities in the control of the qubits (laser field drifts, magnetic field fluctuations, etc.) can lead to stochastic error processes that we will model with a depolarizing error channel. One such natural stochastic process that has shown to be a contributing source of error is scattering during the application of the gate [84, 119]. To model the effects of spontaneous Raman and Rayleigh photon scattering, we will apply a single-qubit depolarizing channel (equation 3.5) after every qubit involved in a gate (single- or two-qubit gates).

3.4.5 Dephasing Errors

While the ions are located in the trap where the DC electric fields vanish, the ions may still be exposed to oscillating electric fields from blackbody radiation, laser fields, or motion around the field free point in the oscillating trap field [120]. The application of the oscillating electric field shifts the energy each of the states of the two-level qubit system by the AC Stark effect, which introduces dephasing errors in the applied gates. This effect is observed for both single- and two-qubit gates. We choose to model these dephasing errors as a single qubit channel of the form:

$$E_d = \left\{ \sqrt{1-p_d}\, I,\ \sqrt{p_d}\, Z \right\} \tag{3.12}$$

where each channel is applied to each qubit involved in single- and two-qubit

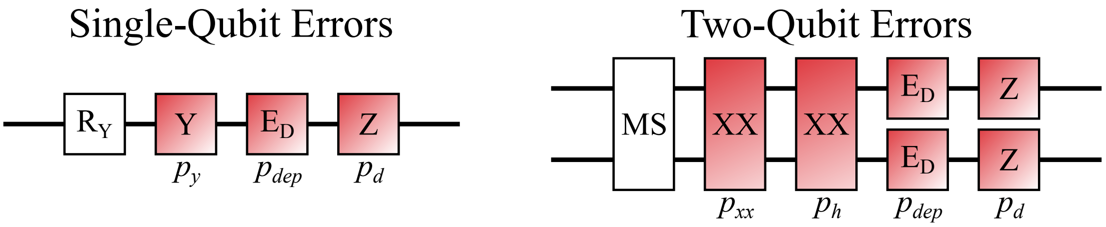


Figure 3.4: The ion trap error model implemented for the 17-qubit surface code simulations. Gate over-rotations about the applied gate axis and depolarizing errors have static error rates. Motional mode heating (XX errors) and single-qubit dephasing have error rates proportional to the gate time of the applied gate.

gates and $p_d = r_d \times t_g$ for each gate where r_d is the dephasing rate and t_g is the time of the applied gate. We make the approximation that single- and two-qubit dephasing errors occur at a constant rate. This is certainly not true in that the dephasing rates will be gate dependent between two-qubit gates and will likely not be at the same rate of single-qubit dephasing but, taking that single-qubit gates have higher fidelities relative to two-qubit gates, this serves as a pessimistic approximation which is consistent with our level of abstraction.

3.4.6 Ancilla Preparation and Measurement Errors

For the ion trap error model, measurement errors were modeled by a single-qubit depolarizing channel applied before the measurement with a probability equivalent to that of the single-qubit over-rotation errors of the single-qubit gates. Preparation errors were modeled with a single-qubit depolarizing channel applied immediately after the preparation of the state but with a probability equivalent to the background depolarizing channel. All states are prepared and measured in the $+Z$ basis, which can be performed with high-fidelity [121]. Note that this implementation is not ideal given that both state preparation and state readout rely on the same scattering processes. However, the preparation and measurement errors should not be the dominant source of failure in the simulations consistent with single-qubit gate, preparation, and readout fidelities of $\geq 99.9\,\%$ [121]. Furthermore, state preparation/measurement is a high-fidelity operation (relative to two-qubit gates) so the inflated state preparation errors will give a pessimistic simulation of the fault-tolerance of the surface code on ion traps relative to the physical architecture. These claims are reinforced in section 3.5.4.

3.5 Error Correction for Ion Trap Errors

To perform error correction on the surface code, classical decoding algorithms have been developed to determine the most appropriate correction operation to perform given the limited information about the encoded state from the syndrome. Various decoders are available that trade-off classical efficiency and observed error threshold. We apply a few decoders for error correction on the surface code below and discuss their performance. For all simulations, we implemented a Monte Carlo simulation of the surface code using an importance sampling method described in Ref. [64].

3.5.1 Integration into Ion Trap Hardware

When choosing a decoding method to integrate into a physical architecture, there is much to consider that extends beyond the (pseudo)threshold. Processing, memory, and runtime requirements of the decoder play a role in the feasibility of implementing error correction with an experimental control system.

Lookup Table Decoder

The simplest decoder is a lookup table that maps a syndrome configuration to the lowest weight Pauli error corresponding to the syndrome. We may represent an error configuration $\mathbf{e}$ as a binary (row) vector $\mathbb{F}_2^{18}$ where the first/last 9 elements of the vector correspond to X-type/Z-type errors on the data qubits; for instance:

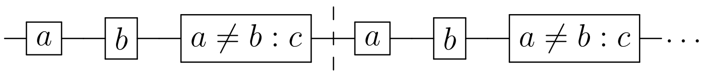
$$\mathbf{e}(2563) = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1] = X_6Z_7Y_8 \tag{3.13}$$

Given two matrices, $H(C_1)$ and $H(C_2^{\perp})$, that correspond to the binary representation of the Z-type and X-type stabilizers, respectively; one may define a mapping matrix T between error configurations $\mathbf{e}$ and binary syndrome (column) vectors $\mathbf{s}$:

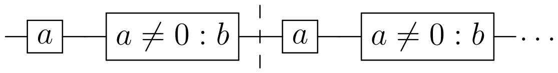
$$T = \begin{pmatrix} H(C_1) & 0 \\ 0 & H(C_2^{\perp}) \end{pmatrix} \tag{3.14}$$

Iterating over all elements of $\mathbb{F}_2^{18}$ and applying T, we constructed a lookup table $\text{Tab}[s] = \lfloor e \rceil$ where $\lfloor e \rceil = \min_s(|e|)$ is the minimum weight error configuration corresponding to the syndrome string s. With a slight abuse of notation, we denote $|\cdot|$ as the hamming weight of the error string e with the caveat that Y-type errors are evaluated as the same weight as X and Z-type errors. All of the rules of the full lookup table ($\text{Tab}[s]$) can be constructed with two 16-element tables, each with keys corresponding to the X-type and Z-type stabilizer measurements, respectively.

For circuit-level noise, the lookup table above is not sufficient for fault-tolerance. A set of syndrome processing rules must be imposed to ensure that measurement errors do not result in faulty corrections that introduce errors onto the data qubits. An example of a typical set of rules is shown below (a, b, and c are syndrome outcome strings):



where two rounds of stabilizer measurement are performed and, if the first two measurement outcomes disagree, a third round of stabilizer measurement is performed. Correction is applied based upon the final measurement performed. We chose to employ a different set of fault-tolerant syndrome processing rules that can, on average, reduce the depth of the circuit required to perform a fault-tolerant correction by one round of stabilizer measurement. The routine:



performs one round of stabilizer measurements and performs a correction based on the following round of stabilizer measurements (b) only if the first round was non-trivial ($a \neq 0$). These two sets of rules yield equivalent results for the 17-qubit surface code under circuit-level depolarizing noise.

Minimum Weight Perfect Matching

For topological codes, minimum weight matching algorithms have been shown to be a useful heuristic technique for performing error correction [122–124]. For the distance-3 surface code, the minimum weight perfect matching rules can be encoded into a lookup table that presents a correction operation based on three rounds of syndrome measurement (for circuit-level depolarizing noise) [2].

Decoder Performances

Figure 3.2 shows the performance of the two lookup table style decoders, standard lookup and matching rule derived lookup, considered for implementation in a near-term experimental quantum error correction routine. Lookup table decoders were chosen for their easy integration into existing ion trap experimental controls which have restricted logic/memory available versus other techniques, such as maximum likelihood [125, 126] or deeper memory step matching algorithms [122–124] for example, which would require additional processing power to implement/integrate into an experiment. The lookup table decoder was favored over the matching table because of its requirement for one less round of stabilizer measurement to perform fault-tolerant error correction with a comparable level-1 pseudothreshold to the matching table (figure 3.2). Because current estimates of the syndrome extraction indicate it is relatively slow (figure 3.1), the ability to choose a correction fault-tolerantly from a minimal number of experimental operations is important to maintain coherence of the encoded information. The lookup table was implemented in all further simulations because of ease of integration into ion trap controls while requiring at most two syndrome measurements to fault-tolerantly

Table 3.2: Performance of the two lookup table style decoders considered for implementation into a near-term quantum error correction experiment. Lookup table style decoders were chosen due to their easy integration into the control software of an ion trap system.

Decoder	**Level-1 Pseudothreshold**	**Computational Time (s)**
Lookup Table	3.0×10^{-3}	1.1×10^{-7}
Matching (table)	5.5×10^{-3}	1.43×10^{-6}

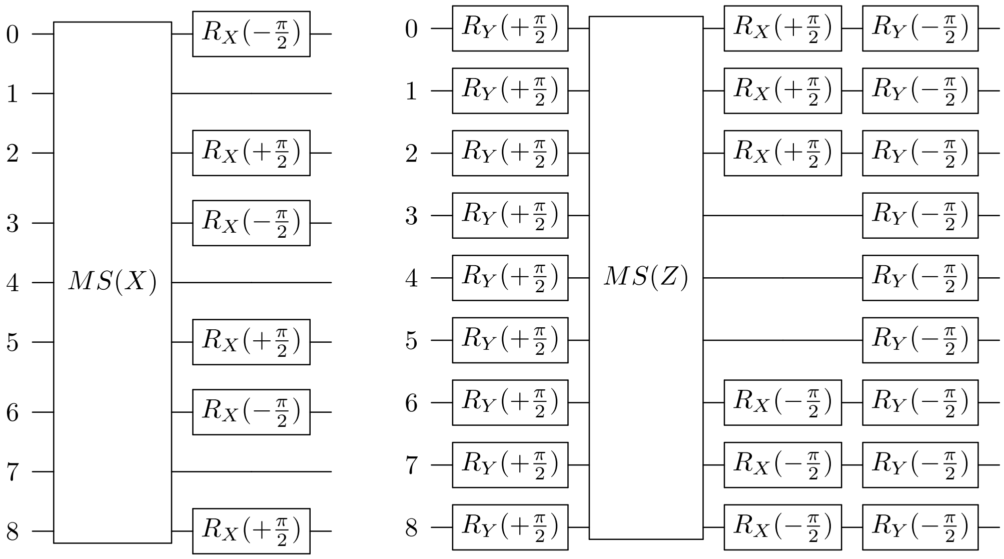
perform error correction.

3.5.2 Error Correction on Ion Traps

Now that a fast, light memory, high-performance decoder has been identified, we will switch attention to using such a method to apply error correction on the 17-qubit surface code under the influence of ion trap errors. First, we must map the abstract quantum circuit used for error correction in the surface code to a circuit that implements gates that would be available in an ion trap quantum computer; specifically single-qubit rotations and Mølmer-Sørensen gates. Next, we will discuss the influence of the individual ion trap error sources (outlined in section 3.4) on the fault-tolerance of the surface code mapped to a linear ion chain highlighting the experimental parameter regimes which would allow for fault-tolerance for the surface code implementation. Finally, we analyze the error subset probabilities from the importance sampling simulations to understand the roles of the competing error sources and gain insight into the error sources that are most influential/detrimental to the error-correcting properties of the code.

3.5.3 Surface-17 Syndrome Extraction Circuit Gate Compilation

The two-qubit gates in the syndrome extraction circuit for the 17-qubit surface code must be decomposed into single-qubit rotation gates and two-qubit Mølmer-Sørensen gates. In addition, Hadamard gates are required during the measurement of the X-type stabilizers which can be decomposed into rotation gates in two equivalent ways: $H \equiv R_Y\left(-\frac{\pi}{2}\right) R_X\left(\pi\right)$ or $H \equiv R_X\left(-\pi\right) R_Y\left(\frac{\pi}{2}\right)$. Note that the



(a) X-type stabilizers

(b) Z-type stabilizers

	MS(X)				MS(Z)		
GATE	s	ID1	ID2	GATE	s	ID1	ID2
PREP		16		PREP		10	
XX	+	6	16	XX	−	0	10
XX	−	7	16	XX	+	3	10
MEAS		16		MEAS		10	
PREP		11		PREP		12	
XX	+	0	11	XX	−	1	12
XX	−	1	11	XX	−	4	12
XX	+	3	11	XX	+	2	12
XX	−	4	11	XX	+	5	12
MEAS		11		MEAS		12	
PREP		14		PREP		13	
XX	+	4	14	XX	−	3	13
XX	−	5	14	XX	−	6	13
XX	+	7	14	XX	+	4	13
XX	−	8	14	XX	+	7	13
MEAS		14		MEAS		13	
PREP		9		PREP		15	
XX	+	1	9	XX	−	5	15
XX	−	2	9	XX	+	8	15
MEAS		9		MEAS		15	

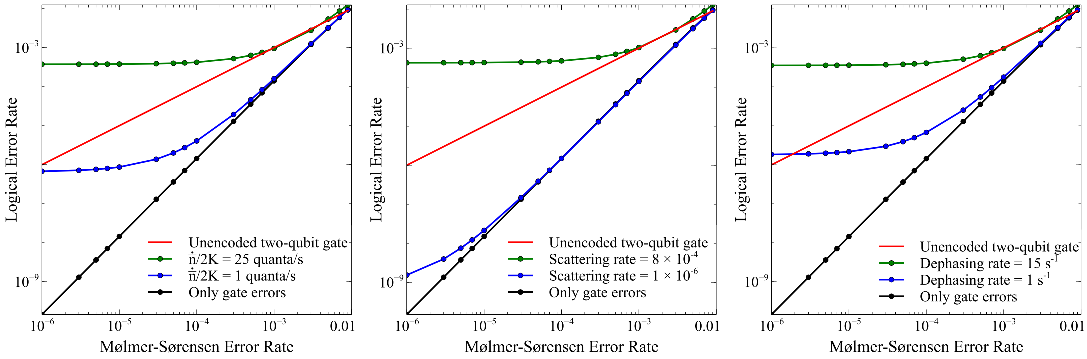
Figure 3.5: (Top) The syndrome extraction circuit for the 17-qubit surface code compiled with Mølmer-Sørensen entangling gates and single-qubit ion trap operations where the ancillary qubit wires have been suppressed. (Bottom) The primitive gate operations compiling the $MS(X)$ and $MS(Z)$ gates above. The values ID1 and ID2 correspond to the qubit indices to which the gate is applied defined in figure 3.1a. The PREP gate projects ancillary qubits into the $|0\rangle$ state and all MEAS gates are Z-basis measurements (see section 3.4.6). The XX gates are Mølmer-Sørensen gates. The parameter s which is dictated by the sign experimental interaction parameter was taken as a free parameter during compilation which we show explicitly.

implementation of the rotation gates constructing the $CNOT$ gate allows for some freedom in the direction of the rotation which can be used to reduce the number of primitive gates (an outline of the ion trap compilation techniques can be found in [93]). The parameter $s \in \{+1, -1\}$ in the circuit dictated by the sign of the interaction parameter χ between two ions which is determined by the experimental apparatus. At our layer of abstraction, the value of s is left as a free parameter. Applying such a compilation method allowed for the reduction of the number of single-qubit gates from 48 in the naive implementation to 30 in the compiled circuit; the number of entangling gates cannot be reduced in the error correction routine leaving 24 Mølmer-Sørensen gates as well. A representation of the compiled syndrome extraction circuit is shown in figure 3.5 where the ancillary wires have been suppressed. This circuit was used for all further results.

3.5.4 Single Error Source Dominant Effects

In this section, we characterize the influence of the error sources in the limit where each error type is the dominant source of the error. Therefore the simulations that generate the following pseudothresholds will have varying single- and two-qubit error rates (remember that $p_x = p_y = p_z = p_{xx}/10$) and constant heating, depolarizing, or spin dephasing error rates during simulations. Our goal is to find a parameter range under which, again in this limit of a dominant error source, fault-tolerant retention of the encoded information would be possible. In all instances, a two-qubit gate fidelity of $\geq 99.9\%$ and an error source error rate below a critical rate is necessary to allow for fault-tolerance (see figure 3.6). We discuss those critical rates for each error source below.

Ion heating was characterized by a parameterized representation of the heating rate $\dot{\bar{n}}/2K$ where $\dot{\bar{n}}$ is the heating (in quanta/s) of the gate motional mode and K is the number of loops in phase space traversed by the Mølmer-Sørensen gate. As shown in figure 3.6a, fault-tolerance is not achieved at a heating rate above 25



(a) Ion heating (b) Depolarizing Noise (c) Spin dephasing

Figure 3.6: The influence of ion trap error sources on the fault-tolerance of the 17-qubit surface code. For each plot, only the labeled error source was introduced in the simulation in addition to gate errors. To achieve fault-tolerance, a two-qubit gate fidelity of $\geq 99.9\%$ and an error in the gate from the specific error source below a critical value (green curves) is required.

quanta/s which corresponds to a motional mode heating rate during the gate of 100 and 200 quanta/s for $K = 2$ and $K = 4$, respectively. A heating rate ($\dot{\bar{n}}$) of about 58 quanta/s has been observed for a single $^{9}\mathrm{Be}^{+}$ ion on a room temperature surface trap [127] and a silicon based trap in a cryogenic environment used to trap individual $^{40}\mathrm{Ca}^{+}$ ions exhibited heating rates as low as 0.33 quanta/s (0.6(2) quanta/s on average) [128]. Note that macroscopic traps exhibit significantly lower heating rates relative to surface traps; for instance a single trapped $^{111}\mathrm{Cd}^{+}$ ion exhibited a heating rate of 2.48 quant/s for a room temperature macroscopic trap [129]. However, additional difficulties arise for macroscopic traps in engineering a system that allows for ion separation, addressing, and detection required for an error correction protocol. Also, the use of sympathetic cooling ions has been shown to reduce motional mode heating effects on T_2^* [130]; a method which could reduce the heating rates of the idle computational qubits during the error correction routine.

The depolarizing error channel was applied to simulate stochastic error processes. One such process of interest is spontaneous Raman and Rayleigh scattering, which result in single- and two-qubit gate errors. Figure 3.6b displays an upper limit on the scattering rate (per-gate) of 8×10^{-4} to allow for fault-tolerance when scattering errors dominate. Ozeri et al. have shown that gate errors due to Raman scattering to occur at a rate less than 10^{-4} for single-qubit gates but two-qubit gates have scattering rates on the order of 10^{-2} for their experimental setup for various species of trapped ions [84]. These achieved scattering rates are still above theoretical lower bound on the scattering rates for single- and two-qubit gates for $^{171}\mathrm{Yb}^{+}$ by 3 and 7 orders of magnitude, respectively [84], showing potential for improvement especially in the two-qubit scattering case. Rayleigh scattering errors are less substantial, resulting in error rates per gate orders of magnitude below the Raman scattering error rates for heavy ions such as $^{171}\mathrm{Yb}^{+}$ [84].

Spin dephasing was modeled using a model that assumed a constant dephasing rate that scaled linearly with the time of the applied gate. The upper bound on the error rate (figure 3.6c) corresponds to a dephasing rate of $15\,s^{-1}$. These values are related to T_2^* [130]. Note that the use of magnetic clock transitions [121, 131], decoherence free subspaces [132], or sympathetic cooling ions [130] during computation has been observed to increase the T_2 coherence times of the qubits to the order of seconds. A 5-qubit system that has implemented small quantum algorithms [26] and the $[\![4,2,2]\!]$ error detection code [81] with hyperfine qubits exhibits a T_2^* of $\approx 0.5\,s$ [77], but further magnetic field stabilization could improve this as shown in [130] which exhibits over a 10 minute coherence time for trapped $^{171}\mathrm{Yb}^{+}$ ions.

3.5.5 Competing Error Sources: Dominant Errors

To characterize the dominant error sources contributing to the logical error rate in the 17-qubit surface code in the case where multiple error sources are competing, we take advantage of the importance sampling technique. We will briefly outline the importance sampling method; highlighting the use of error subsets that will be independently analyzed to gain insight into the effect of the error source on the logical error rate of the encoded state. This will then be followed by an analysis of the statistically significant error subsets, which will be used to characterize the most malignant errors contributing to the failure rate of the error-correcting circuit.

Importance Sampling

This method is an adaptation of the method from [64] but extended to the case where multiple error sources are available during the simulation. The method relies on approximating the logical error rate as a sum of statistically weighted logical error rates of error subsets. For low enough physical error rates, few subsets need to be sampled in order to obtain an accurate approximation, which makes the approach considerably more efficient than the standard direct Monte Carlo sampling. The subsets are indexed according to the number of errors present in the circuit. For instance for the standard depolarizing error model, the subsets would be indexed according to the number of single- and two-qubit errors present in the circuit. Sampling error configurations corresponding to the number of errors for this subset and calculating the fraction of configurations resulting in a logical error gives an effective subset error rate $A_{s,t}$. Multiplying this subset logical error rate by the total statistical weight of the error subset will provide the subset's contribution to the total logical error rate. Computing the statistical weight is done as so:

$$W_{s,t} = \left(\begin{array}{c} n_s \\ s \end{array} \right) p_s^{|e|}(1-p_s)^{n_s-|e|} \left(\begin{array}{c} n_t \\ t \end{array} \right) p_t^{|e|}(1-p_t)^{n_t-|e|} \tag{3.15}$$

where s and t are the number of single- and two-qubit errors in the circuit being considered, respectively. These are also the indices of the subset. The values n_s and n_t are the number of single- and two-qubit fault-points in the circuit, respectively. The values p_s and p_t are the single- and two-qubit error channel probabilities and $|e|$ denotes the weight of the error. Estimating the logical error rate then constitutes calculating the following sum:

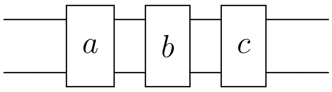


Figure 3.7: Circuit containing three two-qubit gates, labeled a, b, and c, with error rates p_a, p_b, and p_c, respectively.

$$p_L \approx \sum_{(s,t)}^{W_{s,t}<\lfloor W \rfloor} W_{s,t}\, A_{s,t} \tag{3.16}$$

where subsets with statistical weights below a chosen cutoff value, $\lfloor W \rfloor$, are omitted from the sum. Note that, with this method, the sampling of each error subset only needs to be performed once to generate a logical error curve.

We altered the method above to handle situations where errors of equivalent types have different error rates; such is the situation for our ion heating and dephasing error models with ion dependent gate times, which influence the error rate per qubit. To motivate this point, consider the quantum circuit in figure 3.7. This circuit contains three two-qubit gates, a, b, and c, with different error rates p_a, p_b, and p_c, respectively. The weight of the $(0, 2)$ subset would then be:

$$p_a p_b(1 - p_c) + p_a p_c(1 - p_b) + p_b p_c(1 - p_a) \tag{3.17}$$

so the two-qubit subset calculation requires one more ingredient: we need to sum over all n-tuple error configurations (f_n) during the subset weight calculation:

$$W_n = \sum_{n=|e|}^{e\in f_n} \prod_{k\in e} p_k \prod_{j\notin e}(1 - p_j) \tag{3.18}$$

When we adapt this approach to heating errors in an ion trap circuit, we get the following calculation of the subset weight:

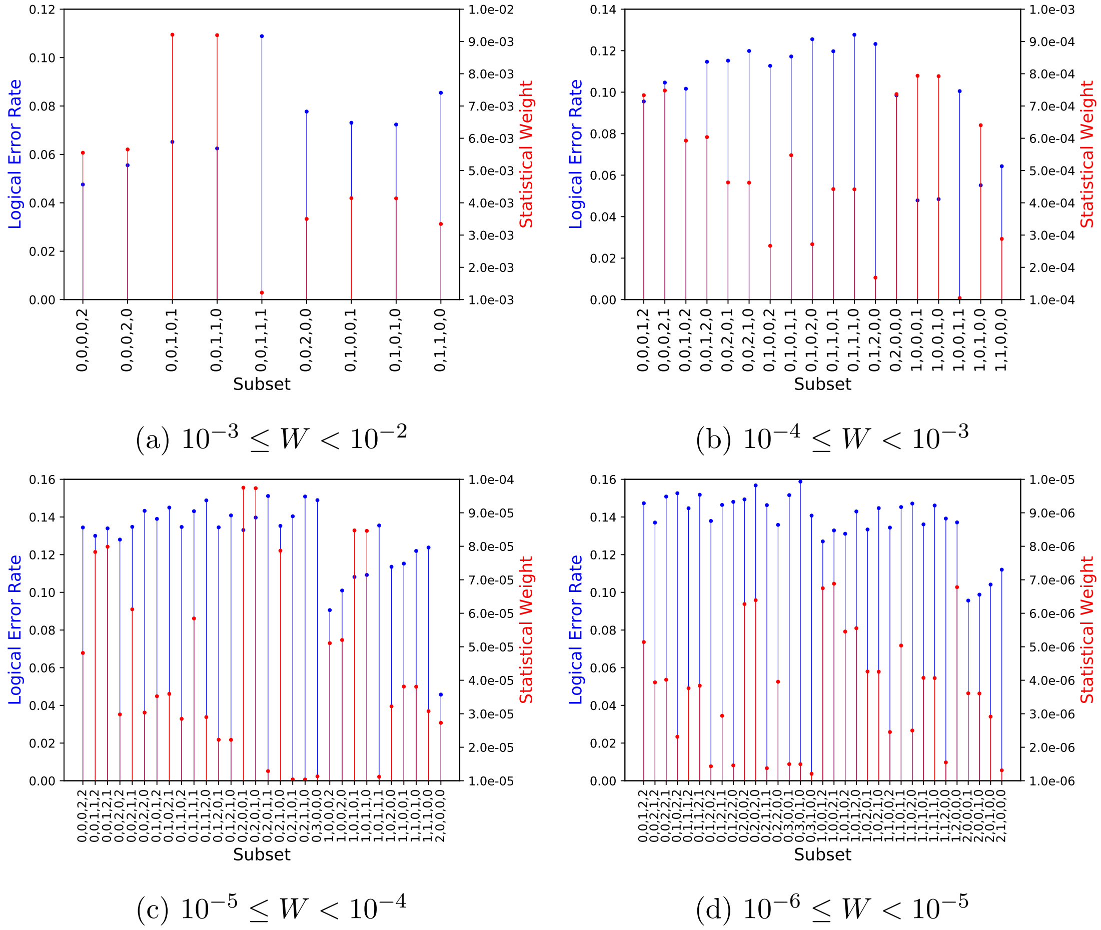
$$\left(\begin{array}{c} n_s \\ s \end{array}\right) p_s^{|e|}(1-p_s)^{n_s-|e|} \left(\begin{array}{c} n_t \\ t \end{array}\right) p_t^{|e|}(1-p_t)^{n_t-|e|} \sum_{n=|e|}^{e\in f_n} \prod_{h\in e} p_h \prod_{\not{h}\notin e}(1 - p_{\not{h}}) \tag{3.19}$$

where p_h and $p_{\not{h}}$ are the individual error rates of the heating error channels for each two-qubit configuration on which an heating error is and is not applied in the simulation, respectively. We have taken into account the influence of the different rates for the calculation of the subset weights, but this also has an influence on the sampled subset logical error rates as certain error configurations will be more probable than others. Because the heating error rates are linearly proportional to the gate times in our error model, we have chosen to sample heating error configurations from a gate time weighted distribution of error configurations giving a corresponding logical error rate of $A_{s,t,h}$. With the new subset weights and subset logical error rates, the estimation of the total logical error rate naturally extends from equation 3.16. Note that heating adds an extra subset label: (s,t,h). The indices s, t, and h represent the number of single-qubit gate, two-qubit gate, and heating errors sampled, respectively.

Recall that the ion trap error model from section 3.4 contains 5 distinct error sources. Therefore, we extended the concepts from equations 3.16 and 3.19 to calculate the logical error rate of the 17-qubit surface code under the influence of single-qubit gate, two-qubit gate, ion heating, background depolarization, and dephasing errors. The analysis below will include 5 index subsets ordered with the indices listing the number of single-qubit gate, two-qubit gate, heating, background depolarization, and dephasing errors sampled in the circuit; in that order.

Competing Error Sources: Sampling Subset Analysis

For the importance sampling simulations of the 17-qubit surface code, a subset weight cutoff of $\lfloor W \rfloor > 10^{-6}$ was used and $30,000$ samples were collected for the calculation of each subset's logical error rate $A_{s,t,h,dep,z}$. This weight cutoff corresponds to events expected to be sampled at least once out of a million randomly sampled events, which is sufficient for near-term error correction experiments. To calculate the statistical weights of the subsets, a single-qubit gate error rate ($p_y = p_x = p_z$), two-qubit gate error rate (p_{xx}), rate of heating (r_{heat}), background depolarizing noise error rate (p_{dep}), and rate of dephasing (r_d) of 10^{-4}, 10^{-3}, 25



(a) $10^{-3} \leq W < 10^{-2}$

(b) $10^{-4} \leq W < 10^{-3}$

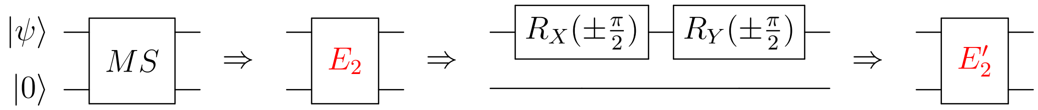
(c) $10^{-5} \leq W < 10^{-4}$

(d) $10^{-6} \leq W < 10^{-5}$

Figure 3.8: The subset logical error rates and subset statistical weights above the cutoff of 10^{-6}. The data is separated into four plots according to the order of magnitude of the subset statistical weights, which are plotted in red. The logical error rates for the subsets are plotted in blue. Note that the product of the subset weight and its corresponding logical error rate dictates the subset's contribution to the total logical error rate of the code. For calculation of the statistical weights of the subsets, the error rates for single-qubit gates ($p_y = p_x = p_z$), two-qubit gates error (p_{xx}), ion heating (r_{heat}), depolarizing noise (p_{dep}), and dephasing (r_d) were 10^{-4}, 10^{-3}, 25 quanta$/s$, 8×10^{-4}, and 15 s^{-1}, respectively.

quanta$/s$, 8×10^{-4}, and 15 s^{-1} was chosen, respectively, which corresponds to the error rates that exhibit a logical error rate equal to the two-qubit gate error rate (see the green curves in figure 3.6). The logical error rates and statistical error weights calculated for each subset are presented in figure 3.8. The goal of this analysis is to parse out situations where certain error sources show dominant contribution to the failure rate of the quantum error-correcting circuit.

The logical error rates for each of the subsets sampled are shown in blue in figure 3.8. The error subsets containing two-qubit gate or heating errors tend to have higher logical error rates than other subsets containing comparable number of errors. This occurs due to the ability of errors of this type to generate measurement faults in the circuit. The Mølmer-Sørensen gate transforms single-qubit data errors in the following manner: $ZI \leftrightarrow YX$ and $YI \leftrightarrow -ZX$ where the data and ancilla qubit errors are the first and second elements, respectively. A two-qubit gate or heating error makes preexisting errors undetectable which is a particularly malignant case. The tendency towards measurement errors in the ion trap error model indicates that implementing a decoder that makes a correction based on more syndrome measurement rounds may show an above average performance boost in error correction. Error subsets containing single-qubit gate errors tend to have lower logical error rates that other subsets with comparable number of errors. To understand why this is the case, we explore the effect the errors have on error correction. Figure 3.9 shows how single-qubit gate errors transform preexisting



(a) Entanglement gate faults are transformed by single-qubit rotations gates and their errors.

E_2		E_2'	E_2		E_2'
II	→	ZI	YY	→	YY
XI	→	II	YZ	→	YZ
YI	→	YI	ZX	→	XX
ZI	→	XI	ZY	→	XY
XX	→	IX	ZZ	→	XZ
XY	→	IY	IX	→	ZX
XZ	→	IZ	IY	→	ZY
YX	→	YX	IZ	→	ZZ

(b) $R_X\left(\pm\frac{\pi}{2}\right)$ Gate Error: X

E_2		E_2'	E_2		E_2'
II	→	YI	YY	→	ZY
XI	→	XI	YZ	→	ZZ
YI	→	ZI	ZX	→	IX
ZI	→	II	ZY	→	IY
XX	→	XX	ZZ	→	IZ
XY	→	XY	IX	→	YX
XZ	→	XZ	IY	→	YY
YX	→	ZX	IZ	→	YZ

(c) $R_Y\left(\pm\frac{\pi}{2}\right)$ Gate Error: Y

● Corrected ● Single Data Error ● Flagged Error ● Measurement Error

Figure 3.9: The transformation of an existing two-qubit Pauli error (ignoring the phase) after a single-qubit gate error on wires that contain an $R_X\left(\pm\frac{\pi}{2}\right)$ and an $R_Y\left(\pm\frac{\pi}{2}\right)$ gate is shown in (b) and (c). The first and second element of the Pauli error corresponds to the error on the data and ancilla qubit, respectively. There are two types of single-qubit over-rotation errors, X and Y, which transform Pauli errors according to (b) and (c), respectively. Applying the transformation $II \leftrightarrow ZI$ on the E_2' values in (c) give the resulting error on wires containing only $R_Y\left(\pm\frac{\pi}{2}\right)$ gate errors.

errors in the circuit. The particularly malignant case in when there is a measurement error, which can introduce errors into the code. For each single-qubit fault point, there are only two elements of the two-qubit Pauli group that are transformed in a manner that would result in a measurement error. Actually, half of the elements of this group result in single-qubit errors (or no errors) on data that can be readily decoded in the following step of stabilizer measurement (see figure 3.9). The remaining errors are detectable but not necessarily corrected properly (this depends on the location that the fault occurs). However, these errors do alert the decoder to the location of an error on the code which is favorable and the faulty correction on these qubits will not propagate errors in a malignant manner given the next round of stabilizer measurement is correct. Take note that one of the malignant errors transformed in figure 3.9b ($R_X(\pm\frac{\pi}{2})$ gate error) is XX which is the form of the two-qubit gate and heating errors. Therefore, compiling out the single-qubit gates $R_X(\pm\frac{\pi}{2})$ gates seems to have also boosted the efficiency of the decoder to decode two-qubit gate and heating errors in addition to the obvious performance boost from less fault points in the compiled circuit. Another alarming malignant configuration in figure 3.9c is the ZX error which is the result of the Mølmer-Sørensen transformation of YI (a single-qubit data error). However, this fault requires two single-qubit Y errors which has a low statistical weight of occurrence (see figure 3.8).

The subset statistical weights (probabilities of occurrence) are shown in red in figure 3.8. These statistical weights give insight into the likelihood of sampling particular error events. Recall that the subset's contribution to the total logical error rate of the code (used to generate the pseudothreshold plots in 3.6) is the product of the subset weight and subset logical error rate (see section 3.5.5). Only ten points above the subset weight of 10^{-3} (figure 3.8a) have significant contribution to the total logical error rate; that is, this small collection of subsets can be used to completely recreate the pseudothreshold plots in 3.6. Actually, the two subsets,

$(0,0,1,0,1)$ and $(0,0,1,1,0)$, have the largest contribution to the encoded logical error rate and bound the logical error rate to $p_L \approx \sum A_{s,t,h,dep,z} \times W \approx 1 \times 10^{-3}$, which corresponds to a two-qubit gate fidelity of 99.9% (recall that the gate error rate for calculation of the subset weights was 10^{-3}). This essentially recreates our calculation of a 99.9% two-qubit gate fidelity for fault-tolerance that used more subsets. Therefore, changes in the statistics of the dominating subsets have significant influence on the observed pseudothreshold of the quantum error-correcting code and can be considered when implementing a decoding algorithm. This also illustrates the concern that a success metric such as the (pseudo)threshold only represents the mean statistics of an underlying error model [133].

Chapter 4

Magic state distillation and gate compilation in quantum algorithms

This chapter contains results from:

Colin J. Trout and Kenneth R. Brown. *Int. J. Quantum Chem.* **115** (19), 12961304.

Transversal operations are inherently fault-tolerant and robust to the faulty operations constituting the computation at physical error rates below their fault-tolerant threshold. Furthermore, the qubit resource cost for transversal operations is simply the cost of encoding. However, the Eastin-Knill theorem states that no subspace stabilizer code with local stabilizer generators can admit a universal transversal gate set [134]. This is significant because quantum computation with a non-universal gate shows no advantage over the classical model of computation. An example of this is the Gottesman-Knill theorem that shows that Clifford circuits can be simulated efficiently on classical computers [34]. To obtain universal fault-tolerant operations, additional protocols are required that require the use of ancillary encoded states.

Traditionally, the quantum resource cost of such protocols is large. Distillation of the surface code requires between $10^6 - 10^{13}$ qubits, depending on the desired precision, to generate one non-transveral fault-tolerant gate [135], which is not feasible practically. Substantial progress in reducing this overhead has been achieved over the past decade. This progress comes in two flavors: more qubit-efficient generation of non-transversal fault-tolerant encoded gates and more non-transversal gate efficient approximations of general unitary transformations. We outline this recent progress in this chapter and show the resource reduction of these techniques. The chapter is outlined as follows. First, we discuss the machinery required to understand how to obtain universal fault-tolerant operations. Then, we discuss methods of obtaining a non-transversal gate for many quantum error correction routines, the T gate. A discussion of obtaining T gate efficient decompositions of general unitary gates follows. Finally, hybrid techniques of using an expanded the gate set to loosen restrictions on gate decomposition is examined.

4.1 Quantum Gates and Teleportation

Two important groups of unitary operations are the Pauli group and the Clifford group. The Pauli group, $\mathcal{P}$, is generated by multiplication of individual Pauli matrices on each qubit. The Clifford group, $\mathcal{C}$, is the group of unitary transformations that maps the Pauli group to itself; for $\hat{g} \in \mathcal{C}$, $\hat{g}\mathcal{P}\hat{g}^\dagger = \mathcal{P}$. On a single-qubit, the Clifford transformations correspond to the symmetry elements of the chiral octahedron (the point group O) [32] shown in figure 2.1b. The Clifford group can be generated by the single-qubit operators H, S, and the two-qubit operator $CNOT$. To obtain an arbitrary qubit rotation (a universal gate set), an additional single-qubit gate is required. Almost any pair of rotations will work with the exception of two rotations that are symmetry elements of the same point group. The canonical choice in quantum information is H and $T = R_Z(\pi/4)$ also referred to as the $\pi/8$ gate. The T gate, which does not preserve the chiral octahedron, plus the Clifford gates is a common universal gate set. Note that $T^4 = S^2 = Z$. It has long been known that any rotation can be efficiently simulated using these gates [31] but, as we discuss in this chapter, recent work has drastically reduced the cost.

Imagine a situation where qubit 1 can only have Clifford gates act on it (including the $\widehat{\text{CNOT}}$) and qubit 2 can have Clifford gates and rotations around the z-axis by an arbitrary angle $R_Z(\theta)$. We can effectively perform $R_Z(\theta)$ on the first qubit by using the important quantum primitive of gate teleportation [136]. The procedure, depicted in figure 4.1a, works as follows: qubit 1 is in an arbitrary state. Qubit 2 is prepared in the state $|+\rangle = 1/\sqrt{2}\,(|0\rangle + |1\rangle)$ and the rotation

$R_Z(\theta)$ is applied. We will refer to this state as $|A(\theta)\rangle = (e^{-i\frac{\theta}{2}}|0\rangle + e^{i\frac{\theta}{2}}|1\rangle)$. Next $\hat{\mathrm{CNOT}}(1,2)$ is applied and then qubit 2 is measured in the computational basis. The net effect is as follows: if qubit two is measured in state $|0\rangle$, then $R_Z\,(\theta)$ was applied to qubit 1. If qubit two is measured in state $|1\rangle$, then $R_Z\,(-\theta)$ was applied to qubit 2. In this case, we can gamble and attempt to apply $R_Z(2\theta)$ and continue this procedure until success [137].

If $\theta = \pi/4$, we do not have to gamble: $|A(\pi/4)\rangle = 1/\sqrt{2}(e^{-i\frac{\pi}{8}}|0\rangle + e^{i\frac{\pi}{8}}|1\rangle)$. If we measure qubit 2 to be in the state 1, we have performed the undesired rotation $R_z(-\pi/4)$: $|\psi\rangle = 1/\sqrt{2}(e^{+i\frac{\pi}{8}}\alpha|0\rangle + e^{-i\frac{\pi}{8}}\beta|1\rangle)$. This can be corrected deterministically by applying the Clifford group operator $R_Z(\pi/2) = S$ to the first qubit (see figure 4.1b). We can then use our ability to perform $R_Z(\pi/4)$ to create a deterministic circuit for implementing $R_Z(\pi/8)$. This process can continue and it is often useful to consider gates that are of the form $R_Z(\pi/2^k)$ as well.

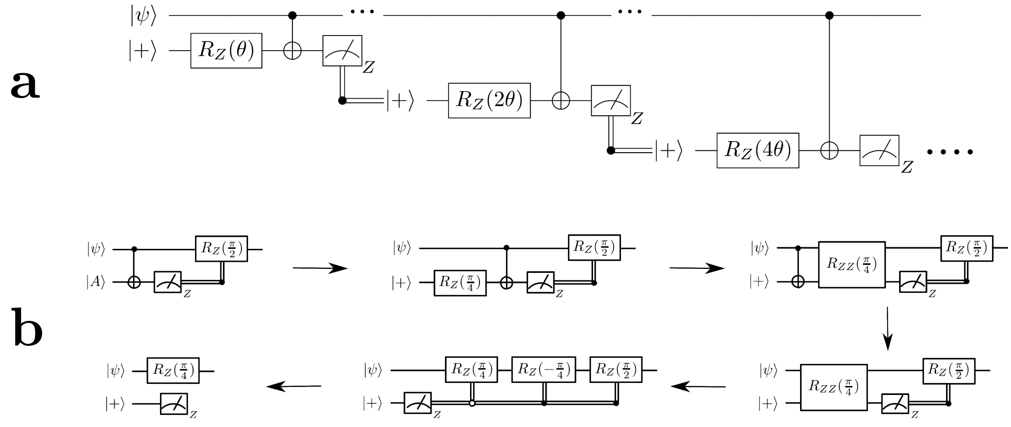


Figure 4.1: Gate teleportation for the implementation of general rotations, $R_Z(\theta)$, and the deterministic implementation of the rotation $R_Z(\pi/4)$. Horizontal double black lines indicate classical bits. Vertical lines represent control with a black (white) dot signifying control on the bit being 0 (1). (a) Circuit for the implementation of the gate $R_Z(\theta)$. If the measurement outcome is 0, then the desired gate was applied and the circuit terminates. If the outcome of the measurement is 1, the rotation was applied in opposite direction and a correction rotation, $R_Z(2\theta)$, may be attempted in a similar manner. We may continue to apply these corrections until we get the desired measurement. (b) Deterministic teleportation of the rotation, $R_Z(\pi/4)$, to the input register $|\psi\rangle$ from the utilization of magic state, $|A\rangle$. The first circuit (top left) is the teleportation circuit. Through quantum circuit identities, we show how this circuit results in a deterministic gate teleportation.

4.2 Fault-Tolerant Universal Gate Sets

For an important family of codes the Clifford gates are transversal and the costly gate is the T gate [138]. This gate can be implemented by teleportation if the magic state $|A(\pi/4)\rangle = T|+\rangle$ can be prepared following figure 4.1b. At first glance this seems like an avoidance of the problem. Clearly to generate $|A(\pi/4)\rangle$, we need to apply T. However, we can use a different code family to build a distillation circuit that allows us to efficiently generate high-quality $|A(\pi/4)\rangle$ states from multiple noisy copies.

4.2.1 T Gate Distillation

Bravyi and Kitaev developed the first method for T gate distillation based on the $[\![15,1,3]\!]$ quantum Reed-Muller code to distill the magic state $|A(\pi/4)\rangle$ [139]. This family of codes are *triply even* codes which admit a transversal T gate. This code takes 15 noisy ancillary magic states, with error rate p, and encodes them into 1 qubit of information with a failure in the distillation circuit when 3 or more errors occur during the encoding. By performing the error check measurements and corrections in the quantum error-correcting routine, one may iteratively project the noisy set of magic states onto one higher fidelity magic state, with an output error rate $\mathcal{O}(p^3)$, resulting in a noisy to distilled state ratio of 15-to-1 [139].

The distillation circuit for this protocol is depicted in figure 4.2. Notice the circuit employs only Clifford operations except for the input magic states used for deterministic T gate teleportation as in Fig. 4.1b. Further rounds of distillation can be performed on distilled states to suppress error rates up to an arbitrary precision and, thus, allows for fault-tolerant implementation of the non-Clifford T gate.

Meier, Eastin, and Knill (MEK) utilized a $[\![10,2,2]\!]$ error detection routine within the distillation circuit [140]. This code has a higher ratio of noisy to

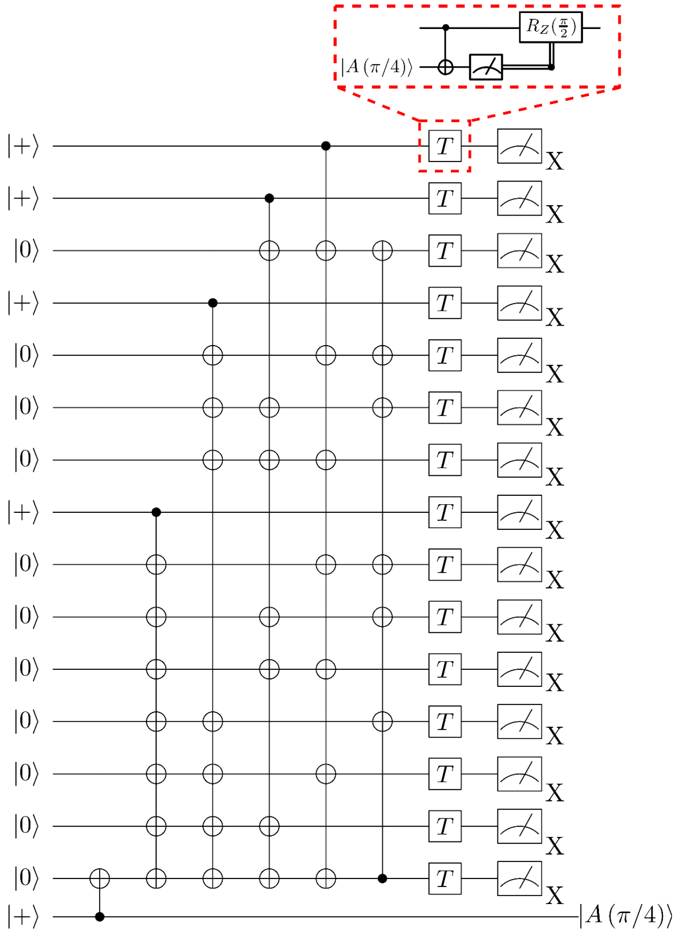


Figure 4.2: The Bravyi-Kitaev 15-to-1 magic state distillation protocol. The input to the circuit is 16 easy to prepare input states, $|0\rangle$ and $|+\rangle$, and 15 low-quality magic states with error p, and the output is 1 magic state with an output error rate of $\mathcal{O}\left(p^3\right)$. It implements 15 deterministic T gate teleportations (figure 4.1b).

distilled states of 10-to-2 at the cost of a reduction in error suppression of the distilled states of $\mathcal{O}\left(p^2\right)$. Dependent on the target error rate for the distilled state, the 10-to-2 technique can more efficiently produce distilled states than the 15-to-1 protocol. There is a regime of low input error rates for which the quadratic suppression of errors from 10-to-2 technique is sufficient and extra resource cost for cubic error suppression via the Bravyi-Kitaev distillation is deemed unnecessary. For high-initial errors and low target errors, initial implementation of the 15-to-1 protocol for maximum error suppression followed by the 10-to-2 protocol to minimize ancilla overhead can result in a favorable resource scaling relative to using a single protocol [140].

Bravyi and Haah introduced a method with a ratio of noisy to distilled states of $3k+8$ to k for $|A(\pi/4)\rangle$ with output error rates comparable to that of Meier-Eastin-Knill, $\mathcal{O}(p^2)$ [141]. Furthermore, they introduced a systematic method of generating distillation protocols by realizing an equivalence between a set of quantum error-correcting codes that admit a transversal T gate and a family of matrices known as triorthogonal matrices. The triorthgonal matrix G defines a classical linear error correcting code space $\mathcal{G}$ where, adhering to a CSS-like construction of stabilizer codes, the X stabilizer generators are defined by the even weight rows of G, G_0, and the Z stabilizer generators are constructed from $\mathcal{G}^{\perp}$. Therefore, generating new distillation routines translates into generating new triorthogonal matrices. As with the 10-to-2 distillation, the utility of this technique is substantial at lower error rates and, therefore, admits serial use of distillation protocols due to constraints on the input error regime in which the resource gains of this technique outweighs the gain in output state precision [141].

4.3 Gate Decomposition Techniques and Minimizing Non-Clifford Rotations

We will now present three general methods for generating fault-tolerant arbitrary unitary transformations. The first method uses a minimal, universal, elementary gate set (typically Clifford operations and T gates) and approximates arbitrary rotations through successive application of gates from this set. These methods minimize the number of T gates to reduce the overhead due to distillation. The second method, known as complex distillation, incorporates an overcomplete gate set which is composed of Clifford operations and incremental rotations about the x, y, or z axis on the Bloch sphere to construct arbitrary rotations. The third method is sequential probabilistic application of gates through teleportation. Table 4.1 illustrates the decrease is resource costs from the progress in these techniques.

4.3.1 Optimization with a Minimal Gate Set

Gate decomposition techniques critically rely on a fundamental result in quantum computing known as the Solovay-Kitaev theorem. Given a set of single-qubit gates (e.g., H, T, and $T^\dagger$) that generate a dense set of unitary gates (i.e. compose a symmetry group isomorphic to $SO(3)$), it is possible to approximate any unitary gate to an arbitrary precision (δ), by an efficient number of operations, $\propto \log^c(1/\delta)$ where c is a constant [33]. The challenge is how to compile these elementary gates into accurate approximations of arbitrary unitary operations.

We will now outline the progress in methods of approximating arbitrary unitary rotations from a set of Clifford and T gates. The Solovay-Kitaev algorithm presents a step-wise procedure for generating arbitrary gates [142]. First, a library is generated of all rotations up to a sequence length of elementary gates l. The target rotation is compared to the closest rotation in the library. The difference between these rotations is a small rotation that can be generated from the library taking advantage of the non-commutivity of rotations. This can be done in a recursive manner to generate arbitrarily accurate gates without ancillary qubits. Further optimization methods have been implemented to the Solovay-Kitaev algorithm with improvements that reduce the level of recursion required for the calculation of the sequences [143]. The final elementary gate count also depends critically on the size of the library as demonstrated by Bocharov and Svore [144].

The best gate counts will occur for infinitely large libraries. However, this approach becomes impractical for high-precision approximations. For moderate precision, Fowler found optimal sequences by a "brute force" space search that utilizes information about previously applied gate sequences to selectively screen for the next gate sequence [143, 145]. The precision at which this method can approximate gates is limited as the space search is exponential in gate sequence length, but this result was nevertheless incredibly insightful as it illuminated the significant gap in gate sequence length between optimal decompositions and the Solovay-Kitaev result.

An alternate approach for developing an optimal decomposition required the use of ancillary states that could offer small corrections to the approximated state through a method known as phase kickback [146]. From these methods, a representation developed where the problem is mapped to a unit field and optimal gate sequences that can be obtained without the use of ancillary states [146–149]. Nondeterministic methods of approximating such rotations have also been developed which make use of a minimal set of gates and ancillary states which are measured to iteratively project the input state to the required state within a given precision [150–153].

Significant progress has recently been sparked by work performed by Kluichnikov, Maslov, and Mosca (KMM) [154]. This work presented a rigorous proof that any single-qubit unitary operator can be *exactly* decomposed into a set of single-qubit Clifford and T gates, with one $|0\rangle$ ancilla state, if and only if the matrix elements of the unitary operator belong to a special algebraic group and fulfill constraints on the assignment of the matrix elements to guarantee unitarity. Furthermore, it was shown that this decomposition was efficient. Giles and Selinger extended this proof to include n-qubit unitary transformations and eliminated the need for ancillary qubits in the exact decompositions [155]. This result is powerful as it added intuition into the search space of potential circuits resulting in a deviation from random circuit search methods.

The question remained: given a desired rotation within a given precision, how does one find a rotation with an exact decomposition that lies within that precision? Furthermore, can one find an exact unitary decomposition, within that precision, which is minimal in the use of the costly T gate? Various algorithms for searching the space of unitaries with exact decompositions have been implemented for the generation of these circuits [146–149]. These methods differ from the search methods implemented by Fowler in that, instead of searching over gate sequences, they search for the entries in the single-qubit unitary that simultaneously satisfy the norm equation to guarantee unitarity and are members of an algebraic ring, the cyclotomic integers, to ensure the existence of an exact decomposition. Initial attempts at these optimizations resulted in decompositions that did [146] and did not [147] require the use of a few ancillary $|0\rangle$ states in the circuit. Additional algorithms resulted in an optimal solution in the number of T gates with ancilla [148], and Ross and Selinger recently showed optimal solutions in gate decompositions that are exact and require no ancillary qubits [149]. Ironically, the efficiency of the decomposition depends on the ability to factor numbers, a task at which quantum computers excel [156]. Fortunately, probabilistic versions of this search algorithm can be implemented in polynomial classical runtime and can achieve the third-to-optimal decomposition sequence, with T gate counts comparable to the optimal solution [149].

4.3.2 The Utility of an Expanded Gate Set from Complex State Distillation

Complex distillation protocols [157–159] utilize an overcomplete elementary gate set consisting of Clifford operations and a set of rotational states about a given axis. These rotations about an axis are produced by a distillation protocol that takes magic states, such as those constructed from the routines above, as an input and generates a "harder" rotation. While employing subsequent distillation procedures on distilled states appears to lower efficiency, the rationale behind these techniques is founded in the fact that the minimization of the number of ancilla required per magic state does not necessarily imply more cost-efficient computation as the above protocols are ignorant to the compiling cost of the computation; an issue addressed by these complex distillation techniques. Indeed, there is gate number threshold above which generating a "difficult" rotation directly may benefit over successive applications of a set of cheaper rotations when approximating an arbitrary rotation. As an example of how this may reduce a gate sequence length for an arbitrary desired rotation, if one has access to the rotations $R_Z(\pi/2^k)$, one can approximate any arbitrary $R_Z(\theta)$ by digital compilation. This can be especially useful for algorithms that require on-the-fly compilation of unitary gates based on measurement of qubits [159, 160].

These protocols differ in the expanded gate set used and the distillation used to generate the expanded magic states. What they do hold in common is that the expanded set of rotations about an axis are generated recursively, ($R(\theta_1) \rightarrow R(\theta_2) \rightarrow ... \rightarrow R_Z(\theta_k)$), with finer rotations requiring more distillation steps for all techniques and these rotations are applied to data via gate teleportation.

Duclos-Cianci and Svore [157] use an elementary gate set that includes Clifford gates and R_Y rotations. The resource states used in the teleportation were constructed from distillation of $|H_i\rangle$ magic states, $|H_i\rangle = R_Y(2\theta_i)|0\rangle$ where θ_i is defined by $\cot\theta_i = \cot^{i+1}(\pi/8)$. Note that $|H_0\rangle = |+\rangle$). An additional routine

Table 4.1: Resource states required for non-Clifford gates for the various methods discussed. Two target gate precisions (δ) are provided: 10^{-2} (left) and $\delta = 10^{-20}$ (right) except for the Fowler (†), PARs (††), and KMM (†††) which are evaluated at a precision of 10^{-6}, 10^{-5}, and 10^{-15}, respectively. Lower bounds on the floating-point method are given as the T gate costs are angle dependent ($\theta = a \times 10^{-\gamma}$). The superscripts denote the categories (a), (b), and (c) corresponding to sections 4.3.1, 4.3.3, and, 4.3.2, respectively. The estimates above assume the availability of sufficiently precise magic states. This assumption increases the advantage of complex distillation techniques. The Ladder State value relative to CISC and DCP is inflated as the number includes the error-free distillation cost of the higher-order states. The advantage of DCP over CISC is in the distillation procedures for noisy magic states (see text). The scaling factors s_1 and s_2 depend on the specific algorithm but are $\gtrsim 1$.

Method	**$\hat{\text{T}}$ Gate Count**	**Non-Clifford States**	
Solovay-Kitaev^a [142]	$s_1 \left(\log^{3.97}(1/\delta)\right)$	86	8.0×10^5
Fowler Search^a [144, 145]	$2.95 \log_2(1/\delta) + 3.75$	1.3×10^2	$3.4 \times 10^{2\,\dagger}$
PARs^b [150]	$s_2 (\log(1/\delta))$	6.0×10^2	$1.4 \times 10^{3\,\dagger\dagger}$
KMM^a [148]	$3.067 \log_{10}(1/\delta) - 4.322$	88	$8.2 \times 10^{2\,\dagger\dagger\dagger}$
Floating-Point^b [151]	$8 \log_2(1/\delta) + 1.14 \log_2(10^\gamma)$	$> 2.9 \times 10^2$	$> 2.9 \times 10^3$
Ladder States^c [157]	-	1.1×10^2	4.3×10^3
CISC^c [158]	-	7	34
RS^a [149]	$3 \log_2(1/\delta) + \log_2(\log_{10}(1/\delta))$	1.1×10^2	1.1×10^3
RUS^b [137, 152]	$1.15 \log_2(1/\delta)$	42	4.2×10^2
DCP^c [159]	-	7	34
PQF^b [153]	$\log_2(1/\delta) + \log_{10}(\log_{10}(1/\delta))$	38	3.7×10^2

utilizing a set of additional rotations $\{|\psi_i^0\rangle, |\psi_i^1\rangle, |\psi_i^2\rangle\}$ generated Clifford circuits with $|H_i\rangle$'s as an input creates a larger, denser set of rotations partitioning the xz plane. The work presented a systematic method of generating dense states about a plane on the Bloch sphere. The rotations generated by the magic states are not uniformly partitioned and although they can be combined to generate any angle, there is no natural decomposition.

Landahl and Cesare modified the previous approach to complex distillation techniques in two ways: first they addressed the non-uniform partitioning of the yz plane by performing $R_Z\left(\pi/2^k\right)$ rotations and also implemented a distillation routine with a "top down" recursion [158]. In the "top down" recursion method, additional $R_Z\left(\pi/2^{k-1}\right)$ input states are needed if the $R_Z\left(\pi/2^k\right)$ teleportation is faulty and even less $R_Z\left(\pi/2^{k-2}\right)$ states are required as they are applied upon consecutive failures of the previous two teleportation attempts which probabilistically reduces the number of traditionally distilled states required for gate teleportation because the teleportation must fail $k-3$ times before a $T|+\rangle$ magic state must be prepared [157, 158]. The technique used shortened Reed-Muller codes that require input magic states of increasing quality to achieve the smallest rotations.

Duclos-Cianci and Poulin overcame the limits of using Reed-Muller codes by the implementation of a modified 10-to-2 distillation circuit [140] to generate magic states $|Y_k\rangle = ZSH\big|A(2\pi/2^k)\big\rangle$ for $R_Y\left(\pi/2^k\right)$ rotations. Conditional on the measurement, the distillation procedure transforms two noisy $|Y_k\rangle$ states into two quadratically improved $|Y_k\rangle$ states. The method requires Clifford operations and the application of a parity measurement, which consumes 16 magic states of the type $|Y_3\rangle$ and 1 magic state of the type $|Y_{k-1}\rangle$ [159]. The similarity of the distillation for all k allows the procedure to work with magic states of a fixed initial accuracy regardless of k. For an input accuracy of 1%, $|0\rangle$, a Clifford state, serves as a sufficiently accurate input state. The authors find a slightly lower resource overhead in terms of non-Clifford input states when substituting $|0\rangle$ for input $|Y_k\rangle$

when $k > 8$ than for $k > 3$. This method has comparable resource costs relative to T gate only compilation methods for arbitrary rotations, but substantial savings for rotations by angles $2\pi/2^k$.

4.3.3 Non-Deterministic Application of Rotations

The third set of methods can be thought as random-walk with correction teleportation techniques. Teleportation is attempted with ancillary states that are measured to detect the success of the application of the gate, similar to the teleportation scheme shown in figure 4.1a. If the gate fails, recursive corrections, such as those we have seen from complex distillation techniques, can be applied until the desired gate is applied. An early example of this technique was shown by Jones et al., which implemented programmable ancilla rotations (PARS) [150]. This method required non-Clifford ancillary states that were expensive to generate. Recent methods have eliminated the need for these difficult ancillary states to non-deterministically achieve arbitrary rotations [151, 152]. The most recent non-deterministic technique by Bacharov et al. utilizes the knowledge of previous rotation attempts in a manner that probabilistically guarantees the termination of the circuit after a finite set of gates [153].

Chapter 5

Quantum error correction failure distributions: Comparison of coherent and stochastic error models

This chapter contains results from:

Jeff P. Barnes, **Colin J. Trout**, Dennis Lucarelli, and B. D. Clader, *Phys. Rev. A* **95**, 062338 and unpublished work performed during an internship at the Johns Hopkins Applied Physics Laboratory.

Coherent errors are of interest due to their deleterious affect on quantum error correction [37, 38]. They are errors that act as unitary evolutions on the computational quantum state. These errors cannot map to a stochastic error channel because they require the coherent evolution of multiple interfering error pathways. At the physical level, incoherent approximations to coherent channels can be simulated efficiently classically [34] but cannot capture the full dynamics of the coherent evolution [113]. This is significant because threshold theorems for general noise sources require a grasp on the error spread during computation [44]. The study of the extent of the influence of coherent errors on fault-tolerance is ongoing [113, 161] partially due to the difficulty in simulating such error channels for large systems.

An additional difficultly arises when attempting to determine fault-tolerance experimentally for coherent errors. Typically, the figure of merit measured in experiments is the average gate fidelity, which can be used to calculate an average error rate for the process. However, difficulties can arise when relating the error rate to the Diamond distance; a metric useful for determining fault-tolerance [37, 38, 162]. For incoherent errors, the average error rate and the Diamond distance are nearly equivalent allowing for fault-tolerant thresholds to be calculated straightforwardly. For coherent errors, this is not the case because the average error rate only loosely bounds the value of Diamond distance [37]. Therefore if one was given gate infidelities from processes containing coherent errors, the inferred fault-tolerance threshold for the computation may range over several orders of magnitude.

In this chapter, we explore the output statistics of certain failure metrics of error detection/correction codes under the influence of stochastic Pauli and the coherent pulse-area error channels. This chapter is different than the other sections in that we performed a full density matrix simulations of the quantum codes for the following analysis. Two codes were examined: the $[\![4,2,2]\!]$ error detection and 13-qubit rotated surface code. First, the error models are outlined. Then, the failure metrics and results for the error detection code and 13-qubit surface code are examined.

5.1 The Pulse Area Error Model

First, we introduce the pulse area noise model. This model emulates control errors that arise from imperfect unitary gates during computation. A perfect unitary gate is transformed into an imperfect gate by the following:

$$U = \exp\left(-\frac{i}{\hbar}\left[\hbar\omega G\right]\frac{\pi}{2\omega}\right) \quad \rightarrow \quad U' = \exp\left(-\frac{i}{\hbar}\left[\hbar(1+r)G\right]\frac{\pi}{2}\right) \tag{5.1}$$

where $r \in [-1,\, 1]$ indicates the degree of under/over-rotation of the gate and is sampled from a uniform distribution. An example of such a control error is a jitter in the field amplitude during the application of a gate to a trapped atomic ion. The time integrated field amplitude applied the trapped atomic ion will therefore deviate from its target value resulting in a coherent over/under-rotation of the applied gate where the imprecision angle is random.

5.2 Failure Metrics for Quantum Error Correction

The following analysis relies on calculating the output wavefunctions overlap with the the Hilbert space of correctable/detectable errors for the quantum codes. The correctable spaces are built from codeword space and the correctable space relative to the input wavefunction. Generally, the space build from the logical codewords for a distance-3 code is:

$$\begin{aligned}&\mathcal{S}_L = \{|0\rangle_L,\, |1\rangle_L\}\\&\mathcal{S}_{L+1} = \mathcal{S}_L \cup \{X_i|0\rangle_L,\, X_i|1\rangle_L,\, Y_j|0\rangle_L,\, Y_j|1\rangle_L,\, Z_k|0\rangle_L,\, Z_k|1\rangle_L\}\end{aligned} \tag{5.2}$$

and the spaces built from the initial wavefunction is of the form:

$$\begin{aligned}&\mathcal{S}_\psi = \{|\Psi_0\rangle\}\\&\mathcal{S}_{\psi+1} = \mathcal{S}_\psi \cup \{X_i|\Psi_0\rangle,\, Y_j|\Psi_0\rangle,\, Z_k|\Psi_0\rangle\}\\&\mathcal{S}_{\psi+2} = \mathcal{S}_{\psi+1} \cup \{X_i Z_k|\Psi_0\rangle\}\ \forall\, i \neq k\end{aligned} \tag{5.3}$$

where $\psi+2$ is the full spaces of correctable errors for a distance-3 code. Note that an $L+2$ space would span the whole Hilbert space for a distance-3 code. Therefore, the codeword criteria is pessimistic because $\mathcal{S}_{L+1}$ does not include correctable $X_i Z_k$ $(i \neq k)$ errors. We are interested in the failure of the error correction routine. We will provide six failure metrics for analysis:

$$\begin{aligned}&P^f_{\text{fail}} = 1 - \sum_{s \in \mathcal{S}_f} |\langle s|\Psi(t)\rangle|^2\\&P_{\text{code}} = \sum_{s \in \mathcal{S}_L} |\langle s|\Psi(t)\rangle|^2\\&\mathcal{F}^2 = |\langle \Psi(t)|\Psi_0\rangle|^2\end{aligned} \tag{5.4}$$

where f is an element of $\{L+1,\, \psi+1,\, \psi+2\}$. P_{code} is the projection onto the codeword space and $\mathcal{F}$ is the fidelity of the process. It will be useful to consider cases where the error rotates the code partially outside the codespace. This is an incorrectable error. The metric:

$$\mathcal{F}^2/P_{\text{code}} = \frac{|\langle \Psi_0|\Psi(t)\rangle|^2}{|\langle 0_L|\Psi(t)\rangle|^2 + |\langle 1_L|\Psi(t)\rangle|^2} \tag{5.5}$$

normalizes the part of the output wavefunction that remains in the codespace. If this value is < 1, then the Pauli correction used in standard error correction will never fix the state. We will discuss the error spaces of the quantum error correcting codes in detail below.

5.3 The $[\![4,2,2]\!]$ Error Detection Code

The four-qubit code is a quantum error detection code where the logical subspace is defined by the stabilizer generators:

$$\mathcal{S} = \langle XXXX,\, ZZZZ \rangle \tag{5.6}$$

This leaves two quantum degrees of freedom that are used to define two logical qubits with logical operators exhibiting appropriate commutation relations with respect to the stabilizer group and one another:

$$\begin{array}{cc} \bar{X}_a = XXII & \bar{X}_b = XIIX \\ \bar{Z}_a = ZIIZ & \bar{Z}_b = ZZII \end{array} \tag{5.7}$$

Note that only one of the qubits can be prepared fault-tolerantly and will hold information and the second qubit will be an extra degree of freedom or "gauge qubit". This is sufficient information to construct the logical states. A quick inspection of the stabilizer group and one may notice that a state stabilized by this group is a four-qubit cat state which is our effective $|0\rangle_L$ state. A basis for the logical subspace is the following:

$$\begin{aligned} |0,0\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|0000\rangle + |1111\rangle\right) \\ |1,0\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|1100\rangle + |0011\rangle\right) \\ |0,1\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|1001\rangle + |0110\rangle\right) \\ |1,1\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|0101\rangle + |1010\rangle\right) \end{aligned} \tag{5.8}$$

where the two values separated by commas refer to the state of the first (logical) and second (gauge) encoded qubits, respectively. We will choose to make the first

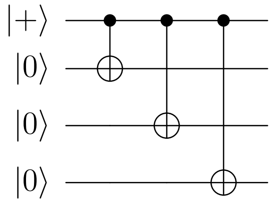


Figure 5.1: Fault-tolerant preparation of the $|0,0\rangle_L^1$ logical state. In this scheme, only the first logical qubit (qubit a) is fault-tolerantly prepared.

logical qubit fault-tolerant with the encoding circuit shown in figure 5.1. Note that no single X error scheduled at any time in the circuit will propagate malignantly *from the perspective of logical qubit a* and propagating Z errors act upon pairs of qubits in the $|0,0\rangle_L$ state which does nothing to the encoded state giving us a fault tolerant encoding scheme.

The four-qubit code is a distance-2 code so it cannot correct errors but we can detect when an error occurs. We can see this clearly both my examination of the logical operations (eqn. 5.7) and the logical states (eqn. 5.8) of the four-qubit code. The action of single qubit X errors may scramble the information just enough to lose distinction between the $\bar{X}_1$ and $\bar{X}_2$ operators or, equivalently, lose distinction between two logical basis states.

5.3.1 Success Criteria for the $[\![4,2,2]\!]$ Error Detection Code

We will now discuss the success criteria for an error *detection* code. Because the role of the code is to detect errors, a failure is a case where errors occur that were not detected by the code. This perspective adapts itself well to a failure criteria where we compute the probabilities of falling into an *undetectable* error subspace. For the Steane Code two subspaces of interest would be:

$$\begin{aligned} \mathcal{S}_L &= \{|0\rangle_L,\, |1\rangle_L\} \\ \mathcal{S}_{L+1} &= \Big\{P|\Phi\rangle \Big| |\Phi\rangle \in \mathcal{S}_L,\ P \in \mathcal{P}_n \text{ s.t. } |P| \leq 1\Big\} \end{aligned} \tag{5.9}$$

where $\mathcal{P}_n$ denotes the Pauli group of size n and $|P|$ is the weight of the operator, i.e. number of non-identity elements composing the operator. This is a common metric used for CHP-like simulations of quantum codes. For wavefunction simulations, the codespace in $\mathcal{S}_L$ is replaced by an initial wavefunction $|\Psi_0\rangle$ giving the following criteria:

$$\begin{aligned}\mathcal{S}_\psi &= \{|\Psi_0\rangle\}\\ \mathcal{S}_{\psi+1} &= \Big\{P|\Psi_0\rangle\Big|P \in \mathcal{P}_n \text{ s.t. } |P| \leq 1\Big\}\end{aligned} \tag{5.10}$$

for partitioning of the error space. Error subspace degeneracy is relevant for the $[\![4,2,2]\!]$ code. We therefore deviate from the natural partitioning of the space shown above in equations 5.9 and 5.10 for assessment of the $[\![4,2,2]\!]$ error detection routine. The four-qubit detection code is small enough ($2^4 = 16$ degrees of freedom) that we can illustrate this in full detail. Our logical space will consist of the qubit that we have chosen to be fault tolerant:

$$\mathcal{S}_L = \left\{|0,0\rangle_L = \frac{1}{\sqrt{2}}\left(|0000\rangle + |1111\rangle\right), |1,0\rangle_L = \frac{1}{\sqrt{2}}\left(|1100\rangle + |0011\rangle\right)\right\} \tag{5.11}$$

where, naturally, $|\mathcal{S}_L| = 2$. Through symmetry of the codewords, there is a equivalence between the states generated by the set of errors $\{X_0, X_2\}$ and $\{X_1, X_3\}$. Furthermore, the set of Z errors is fully degenerate: that is, a Z error acting on *any* qubit results in the same final state. A comprehensive list of the detectable error states is:

$$\begin{aligned}
X_0|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0001\rangle + |1110\rangle\right) & X_0|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1101\rangle + |0010\rangle\right)\\
X_2|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0100\rangle + |1011\rangle\right) & X_2|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1000\rangle + |0111\rangle\right)\\
Z_0|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0000\rangle - |1111\rangle\right) & Z_0|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1100\rangle - |0011\rangle\right)\\
X_0Z_0|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0001\rangle - |1110\rangle\right) & X_0Z_0|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1101\rangle - |0010\rangle\right)\\
X_2Z_0|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0100\rangle - |1011\rangle\right) & X_2Z_0|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1000\rangle - |0111\rangle\right)
\end{aligned} \tag{5.12}$$

leaving only 4 states outside of the detectable error space. A quick observation shows that the remaining states are those where we have flipped the gauge degree

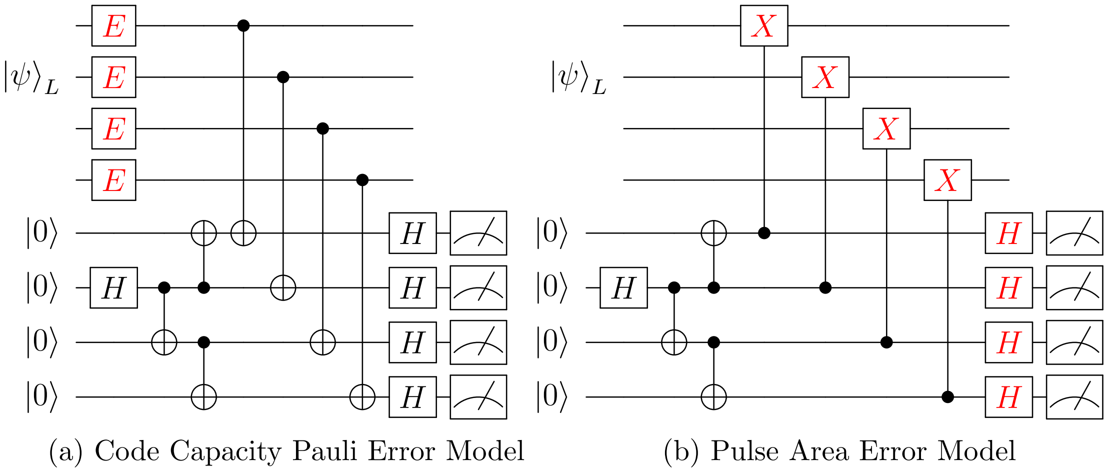
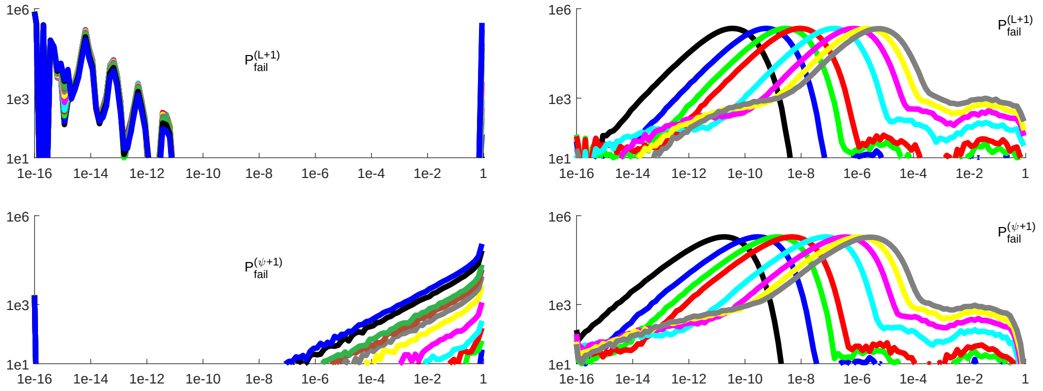


Figure 5.2: Fault-tolerant measurement of the Z (X) stabilizer in the $[\![4,2,2]\!]$ error detection code under the influence of the code capacity Pauli (pulse area) noise model in (a) ((b)).

of freedom. From the perspective of the fault-tolerant qubit, this is equivalent to an error but it is not detected because the logical operator $\bar{X}_b = X_0X_3$ commutes with the stabilizers. We can now build up the four malignant states as the following:

$$\begin{aligned} \bar{X}_b|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1001\rangle + |0110\rangle\right) & \bar{X}_b|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0101\rangle + |1010\rangle\right) \\ \bar{X}_bZ_0|0,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|1001\rangle - |0110\rangle\right) & \bar{X}_bZ_0|1,0\rangle_L &= \frac{1}{\sqrt{2}}\left(|0101\rangle - |1010\rangle\right) \end{aligned} \tag{5.13}$$

where our failure is measured by the overlap of the wavefunction, after an error detection round, with these four undetectable error states. We will follow the previous convention of referring to the failure probabilities as P_{fail}^{L+1} and $P_{\text{fail}}^{\psi+1}$ but with the knowledge that we are considering the aforementioned degenerate subspaces, respectively. Also, note that since there is no correction step in the error detection circuit, the quantities P_{code} and $\mathcal{F}^2$ provide us only the information of whether our noise model sampled an error configuration or not which is not useful to us.



(a) Pauli error model

(b) Pulse area error model

Figure 5.3: Failure criteria distributions for the $[\![4,2,2]\!]$ error detection code under the influence of the Pauli (a) and pulse area (b) noise models. Each distribution contains 3×10^6 samples per error rate. (a) Error rates of $p = 0.0001$, 0.00025, 0.0005, 0.00075 0.001, 0.002, 0.004, 0.006, 0.008, 0.01, 0.02, 0.03 were run. (b) Noise strengths of $\sigma = 0.0025$, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, 0.05 were simulated.

5.3.2 Simulation of the $[\![4,2,2]\!]$ Error Detection Code

We implemented a fault-tolerant syndrome extraction routine for error detection implementing measurements shown in figure 5.2. Fault-tolerance required repeated stabilizer measurements until the syndrome in consecutive steps were in agreement and the use of Shor ancilla [1]. At the beginning of each run, a random vector of the form $\alpha|0\rangle_L + \beta|1\rangle_L$ (where α, $\beta \in \mathbb{C}$) is chosen as the initial state and, for this code, a four-qubit cat state is perfectly prepared (no ancilla verification or decoding required). For simulations with the Pauli error model, a random Pauli error was inserted at each of the data qubits (figure 5.2a) before the measurement of each stabilizer and all operations on the ancilla are perfect. In the simulations under the influence of the pulse area error model, a sampled over/under-rotation is applied to all data-ancilla entangling gates and pre-measurement gates applied to the cat states (figure 5.2b). Histograms of the failure criteria (section 5.2) for different error rates/strengths were accumulated and are shown below.

The Pauli Error Model

A circuit implementing fault-tolerant error detection was run 3×10^6 times for each value of the error rate, p: 0.0001, 0.00025, 0.0005, 0.00075 0.001, 0.002, 0.004, 0.006, 0.008, 0.01, 0.02, and 0.03. The histograms generated for both failure criteria are shown in Fig. 5.3a. The $P_{\text{fail}}^{(L+1)}(p)$ criteria is binomially distributed (float64 accuracy). By plotting the fraction of trials with $P_{\text{fail}}^{(L+1)}(p) = 1$, an error detection pseudothreshold of $p_{th} \approx 0.003$ was observed (Figure 5.4). Note that the detection pseudothreshold does not represent the probability that the data stays within the logical codeword space, it represents the success of the error detection circuit: the probability that errors have occurred on the data qubits that were not detected by the syndrome measurement.

We observe a spread in the failure metric $P_{\text{fail}}^{(\psi+1)} > 10^{-8}$ similar to that of the Steane code [133] but with a smaller tail in the distribution relative to the Steane results. We claim that the tighter distribution of $P_{\text{fail}}^{(\psi+1)}$ criteria is a consequence of the degeneracy of the error space of the code. Recall that the error detection code takes no action to return the wavefunction back to the logical space. One may then expect that the action of the error model should allow the uncorrected wavefunction to freely sample a larger span of Hilbert space relative to a code that implements active error correction translating into a larger degree of variation in the computed overlap of the output state with the error subspaces; yet, the converse is true. This appears to be a consequence of the degenerate structure of the error subspace $\mathcal{S}_{\psi+1}$ where Pauli errors on different data qubits tend to transform the codewords in an equivalent manner. An intuitive interpretation of this phenomena can be formulated in terms of diffusion of the complex vector representing the wavefunction throughout Hilbert space. Given some initial position of the complex vector within the codeword space, the action of the error model serves to drive the vector outside of the regime of the codeword space. Sometimes, the vector is driven outside of the codeword space but into the space ($\mathcal{S}_{\psi+1}$) (a successful error detection), but other times it does not giving a failure in the error detection routine. For the Steane code, all single-qubit Pauli errors have a distinct action on the codeword qubits; each of which can be viewed as a distinct diffusion pathway away from the codespace. However, the degenerate space of the $[\![4,2,2]\!]$ codespace under the action of single-qubit Pauli errors decreases the number of diffusion pathways away from the codespace; a funneling behavior of sorts. It is therefore the symmetry of the codeword space under the influence of errors that acts to localize the distribution of output wavefunctions (for the Pauli error model) relative to a code devoid of such symmetries. We will provide further evidence of this claim in section 5.5.2 and discount this behavior as simply a consequence of the smaller Hilbert space of the $[\![4,2,2]\!]$ code.

An unexpected result of the four-qubit detection code is that the metric $P_{\text{fail}}^{(\psi+1)}$ is binomially distributed with a sharp peak at $P_{\text{fail}}^{(\psi+1)} \leq 10^{-16}$ similar to the behavior of the $\mathcal{P}_{\text{fail}}^{(L+1)} = 1$ peak. This is a consequence of the small size of the undetectable error subspace (the complement of $\mathcal{S}^{(\psi+1)}$) relative to the detectable error subspace, $\mathcal{S}_{(\psi+1)}$. Recall from section 5.3.1 that the overlap of only four states (equation 5.13) contribute to the $P_{\text{fail}}^{(\psi+1)}$. The large span of the detectable error space relative to the undetectable space results in an "on-off" success criteria of the code similar to the failure criteria for $P_{\text{fail}}^{(L+1)}$ which fails only if the output state is a non-stabilizer state. This claim is supported by then overlap of the thresholds generated from the failure metrics $\mathcal{S}_{L+1}$ and $\mathcal{S}_{\psi+1}$. One may view this case as approaching the limit where $|\mathcal{S}^{(\psi+1)}| \rightarrow |\mathcal{H}^n|$ where $\mathcal{H}^n$ is the total Hilbert space of the data qubits.

The Pulse Area Noise Model

For the assessment of the four-qubit error detection code under the influence of pulse area errors, 3×10^6 simulations of the error detection routine were performed

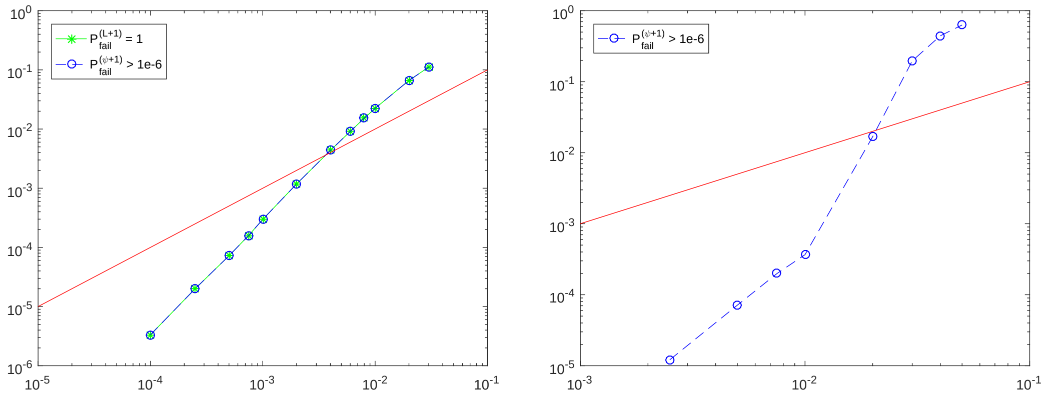


Figure 5.4: Error detection thresholds for the $[\![4,2,2]\!]$ error detection code for Pauli (left) and pulse area (right) errors. A logical failure was counted when $P_{\text{fail}}^{L+1} = 1$ or $P_{\text{fail}}^{\psi+1} > 10^{-6}$ for the respective curves. The red line is the $P_{\text{fail}} = p$ curve.

at each error strength, σ: 0.0025, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, and 0.05. The two failure metric distributions are shown in figure 5.3b. The distributions of the failure criteria are again quite broad, particularly at high error rates. Many of the trials appear to follow a log-normal distribution with some tailing, especially at higher error strengths, that deviates from such a curve. Setting a $P_{\text{fail}}^{(\psi+1)} > 10^{-6}$ failure criteria allowed for a pseudothreshold to be generated for the four-qubit detection code under the influence of the pulse area error model. A threshold value of $\sigma_{th} \approx 0.2$ was obtained. Again, a more sigmoidal shape to the pulse-area threshold was observed relative to the Pauli error model.

An interesting aspect of the histograms is the consistency between the two failure criteria; up to some discrepancies in the tailing behavior near the floating point precision of the calculation. Recall that the histograms for the Steane code showed distinctly different properties between the two types of failure criteria, $P_{\text{fail}}^{(L+1)}$ and $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)}$, including peak center location, distribution breadth, and even general distribution shape for certain cases. It is shown later that this behavior is not a consequence of the peculiarities of the $[\![4,2,2]\!]$ code but, again, attributed to the codeword/space symmetries. It is also important to note that this consistency between the failure metrics occurs at large sample sizes (a large sampling of initial logical states). The output wavefunction of the error detection routine has the ability to report quite different $P_{\text{fail}}^{(L+1)}$ and $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)}$ with a difference up to an order of magnitude. Why this consistency in failure criteria for pulse-area errors occurs in codes with degenerate error spaces but not for a code such as the Steane code is still an open question.

5.4 Concatenation of the $[\![4,2,2]\!]$ Code

Now, we will discuss a fault-tolerant concatenation protocol for encoding the four-qubit code error detection code. The goal is to have the logical state at the higher level of encoding to be of the following form:

$$|0,0;0,0\rangle_L^2 = \frac{1}{\sqrt{2}}\left((|0,0\rangle_L^1)^{\otimes 4} + (|1,1\rangle_L^1)^{\otimes 4}\right) \tag{5.14}$$

where the superscript on the kets denotes the level of encoding. The first and second labels, separated by a semicolon, in $|0,0;0,0\rangle_L^2$ refer to the collective state of fault-tolerant and gauge qubits of the lower level code, respectively. This labeling is made clearer with an example using the logical operators below. To show the full logical state, we will use an abbreviated version of the level-1 encoded states:

$$\begin{aligned}
|0,0\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|\bar{0}\rangle + |\bar{15}\rangle\right)\\
|1,0\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|\bar{12}\rangle + |\bar{3}\rangle\right)\\
|0,1\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|\bar{9}\rangle + |\bar{6}\rangle\right)\\
|1,1\rangle_L &= \tfrac{1}{\sqrt{2}}\left(|\bar{5}\rangle + |\bar{10}\rangle\right)
\end{aligned} \tag{5.15}$$

where the values within the kets are the integer representations of the binary strings that represent the original basis states (equation 5.8). The second level logical state is therefore:

$$\begin{aligned}
|0,0;0,0\rangle_L^2 = \frac{1}{\sqrt{2}^5}&\left[\left(|\bar{0}\,\bar{0}\,\bar{0}\,\bar{0}\rangle + |\bar{0}\,\bar{0}\,\bar{0}\,\bar{15}\rangle + |\bar{0}\,\bar{0}\,\bar{15}\,\bar{0}\rangle + ... + |\bar{15}\,\bar{15}\,\bar{15}\,\bar{15}\rangle\right)\right.\\
&\left.+\left(|\bar{5}\,\bar{5}\,\bar{5}\,\bar{5}\rangle + |\bar{5}\,\bar{5}\,\bar{5}\,\bar{10}\rangle + |\bar{5}\,\bar{5}\,\bar{10}\,\bar{5}\rangle + ... + |\bar{10}\,\bar{10}\,\bar{10}\,\bar{10}\rangle\right)\right]
\end{aligned} \tag{5.16}$$

the sequences in the logical states can be viewed as the group of all length 4 binary strings where we map $0 \to \bar{0}$ and $1 \to \bar{1}\bar{5}$ for the term in the first set of parentheses and define a similar map for the states $\bar{5}$ and $\bar{1}\bar{0}$ for the states in the second set of parentheses.

We can define the stabilizer group and corresponding logical operators recursively. The stabilizer generators for the level two code are the following weight 16 operators:

$$\begin{aligned}\mathcal{S}^2 = \langle &\mathcal{S}_X^1 \otimes \bar{I} \otimes \bar{I} \otimes \bar{I},\ \bar{I} \otimes \mathcal{S}_X^1 \otimes \bar{I} \otimes \bar{I},\ \bar{I} \otimes \bar{I} \otimes \mathcal{S}_X^1 \otimes \bar{I},\ \bar{I} \otimes \bar{I} \otimes \bar{I} \otimes \mathcal{S}_X^1, \\ &\mathcal{S}_Z^1 \otimes \bar{I} \otimes \bar{I} \otimes \bar{I},\ \bar{I} \otimes \mathcal{S}_Z^1 \otimes \bar{I} \otimes \bar{I},\ \bar{I} \otimes \bar{I} \otimes \mathcal{S}_Z^1 \otimes \bar{I},\ \bar{I} \otimes \bar{I} \otimes \bar{I} \otimes \mathcal{S}_Z^1\rangle\end{aligned} \tag{5.17}$$

where $\bar{I} = I^{\otimes 4}$ and the X and Z stabilizers for the level-1 encoding are depicted above as $\mathcal{S}_X^1$ and $\mathcal{S}_Z^1$, respectively. Note that we have increased our degrees of freedom and therefore need a larger number of logical operators that act upon the space. This code incorporates 16 qubits and contains 8 stabilizers, leaving 8 logical degrees of freedom that can be controlled with logical operators. The total group of logical operators, written in a suggestive manner, is shown in figure 5.5. The first four operators are written in a recursive form shown by the labels $\bar{X}_{A,a}^2$ where capital A denotes the pattern of the operator ($a \to X{\otimes}X{\otimes}\bar{I}{\otimes}\bar{I},\ Z{\otimes}\bar{I}{\otimes}\bar{I}{\otimes}Z$) and lowercase a dictates the type of operator place within that pattern. For instance, $\bar{X}_{A,b}^2 = \bar{X}_b^1 \otimes \bar{X}_b^1 \otimes \bar{I} \otimes \bar{I}$ using the b-type logical operators in the pattern of the a-type operators in equation 5.7. Note that only the level-1 logical qubit a is fault-tolerant in our procedure. This means that only operators that incorporate a-type level-1 logical operators are fault tolerant. The fault tolerant operators are shown above the dashed line in figure 5.5. With these operators, we can obtain any of the logical basis states. For instance, we can obtain the state $|1,0;0,1\rangle$ by the application of $\bar{X}_{A,a}^2$ and $\bar{X}_{B,b}^2$ to the state $|0,0;0,0\rangle_L^2$ giving:

$$\begin{aligned}|1,0;0,1\rangle = \frac{1}{\sqrt{2}}\big(&|1,1\rangle_L^1 \otimes |1,0\rangle_L^1 \otimes |0,0\rangle_L^1 \otimes |0,1\rangle_L^1 \\ &+|0,0\rangle_L^1 \otimes |0,1\rangle_L^1 \otimes |1,1\rangle_L^1 \otimes |1,0\rangle_L^1\big)\end{aligned} \tag{5.18}$$

$$
\begin{array}{ll}
\bar{X}^2_{A,a} = \bar{X}^1_a \otimes \bar{X}^1_a \otimes \bar{I} \otimes \bar{I} & \bar{X}^2_{B,a} = \bar{X}^1_a \otimes \bar{I} \otimes \bar{I} \otimes \bar{X}^1_a \\
\bar{Z}^2_{A,a} = \bar{Z}^1_a \otimes \bar{I} \otimes \bar{I} \otimes \bar{Z}^1_a & \bar{Z}^2_{B,a} = \bar{Z}^1_a \otimes \bar{Z}^1_a \otimes \bar{I} \otimes \bar{I} \\
\bar{X}^2_{C} = \bar{X}^1_a \otimes \bar{I} \otimes \bar{I} \otimes \bar{I} & \bar{X}^2_{D} = \bar{I} \otimes \bar{I} \otimes \bar{X}^1_a \otimes \bar{I} \\
\bar{Z}^2_{C} = \bar{Z}^1_a \otimes \bar{I} \otimes \bar{I} \otimes \bar{I} & \bar{Z}^2_{D} = \bar{I} \otimes \bar{I} \otimes \bar{Z}^1_a \otimes \bar{I} \\
\hdashline
\bar{X}^2_{A,b} = \bar{X}^1_b \otimes \bar{X}^1_b \otimes \bar{I} \otimes \bar{I} & \bar{X}^2_{B,b} = \bar{X}^1_b \otimes \bar{I} \otimes \bar{I} \otimes \bar{X}^1_b \\
\bar{Z}^2_{A,b} = \bar{Z}^1_b \otimes \bar{I} \otimes \bar{I} \otimes \bar{Z}^1_b & \bar{Z}^2_{B,b} = \bar{Z}^1_b \otimes \bar{Z}^1_b \otimes \bar{I} \otimes \bar{I} \\
\bar{X}^2_{C} = \bar{X}^1_b \otimes \bar{I} \otimes \bar{I} \otimes \bar{I} & \bar{X}^2_{D} = \bar{I} \otimes \bar{I} \otimes \bar{X}^1_b \otimes \bar{I} \\
\bar{Z}^2_{C} = \bar{Z}^1_b \otimes \bar{I} \otimes \bar{I} \otimes \bar{I} & \bar{Z}^2_{D} = \bar{I} \otimes \bar{I} \otimes \bar{Z}^1_b \otimes \bar{I}
\end{array}
\tag{5.19}
$$

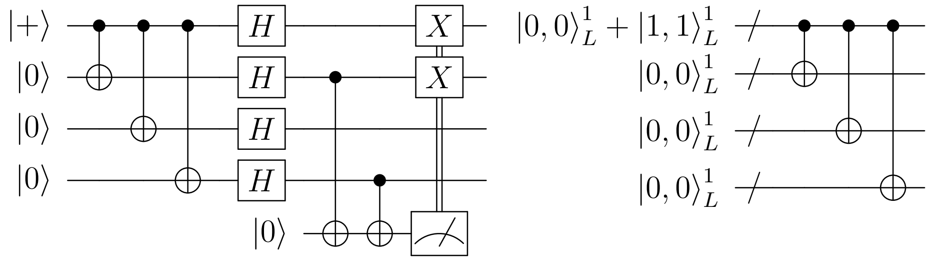
Figure 5.5: Logical operators for the second level of encoding for the four-qubit error detection code. The four logical qubits with double subscript labels are defined recursively from the forms of the level-1 logical operators (figure 5.7). Note that only the qubits above the dashed line have been prepared fault-tolerantly.

and one can see that the product state of the fault-tolerant qubits of the level-1 code mimic that of the $|1,0\rangle^1_L$ (equation 5.8) state (first label before the semicolon) and similarly for the gauge qubits which are represented by the second label.

We now have a 16-qubit error detection code. A quick inspection of equation 5.16 shows that the minimum distance between codewords is still 2 as we cannot reliably decode two-qubit errors within a given code block. Therefore, the concatenated four-qubit code is a $[\![16, 8, 2]\!]$ error detection code where only four logical qubits are fault-tolerant. However, we have gained the ability to detect some multi-qubit error configurations across separate code blocks through the concatenation so our space of detectable errors has grown relative to the level-1 encoding.

5.4.1 Fault-Tolerant Level-2 Encoding

We now outline a fault-tolerant scheme for encoding a level-2 four-qubit error detection code. To move foreward, we must note that the $[\![4,2,2]\!]$ is a self-dual CSS code; allowing for transversal Hadamard and $CNOT$ gates. We show how to gen-



(a) Preparation of $\frac{1}{\sqrt{2}}\left(|0,0\rangle_L^1 + |1,1\rangle_L^1\right)$ (b) Fault-tolerant preparation of $|0,0\rangle_L^2$

Figure 5.6: Machinery required for the fault-tolerantly encode a second level error detection code. (a) First, a plus-like state must be produced by a transversal Hadamard and measurement-based correction. (b) Transversal CNOTs can then be used to encode the $|0,0\rangle_L^2$ state (eqn. 5.14) using the plus-like state and three $|0,0\rangle_L^1$ states.

erate a level-2 fault-tolerant encoding circuit using the aforementioned transversal gates and measurement below.

There are two parts to the encoding circuit: generating a plus-like logical state and entangling the plus-like state with ancillary states to generate $|0,0\rangle_L^2$ which must be performed fault-tolerantly. The plus-like state is of the form: $1/\sqrt{2}(|0,0\rangle_L^1 + |1,1\rangle_L^1)$. We will start with the creation of the plus-like logical state. I've referred to this as a "plus-like" state because it would be misleading to refer to it as the $|+,+\rangle_L^1$ state. This is illustrated by the action of the transversal Hadamard gate on the $|0,0\rangle_L^1$:

$$|+,+\rangle_L^1 = H^{\otimes 4}|0,0\rangle_L^1 = \frac{1}{2}\left(|0,0\rangle_L^1 + |0,1\rangle_L^1 + |1,0\rangle_L^1 + |1,1\rangle_L^1\right) \tag{5.20}$$

which shows the true plus state of the encoded four-qubit code. However, given $|+,+\rangle_L^1$ we can use measurement to probabilistically project to the appropriate subspace to obtain the plus-like state. Specifically, the operator $\bar{X}_a^1\bar{X}_b^1 = IXIX$ allows us to discriminate between the even and odd parity states in equation 5.20. So, to generate the state $1/\sqrt{2}(|0,0\rangle_L^1 + |1,1\rangle_L^1)$ we simply encode four physical qubits into the $|0,0\rangle_L^1$ state (figure 5.1), apply a transversal Hadamard gate to this state, and then apply a measurement-based correction if $\hat{M}_{\bar{X}_a^1\bar{X}_b^1} = 1$. This measurement outcome requires us to flip either the gauge or logical qubit to get the desired state. This process is illustrated in figure 5.6a. The second step is to entangle the plus-like state with three $|0,0\rangle_L^1$ states using an encoding circuit analogous to the $|0,0\rangle_L^1$ state preparation circuit except all gates are now transversal in the level-2 encoding circuit (figure 5.6b). We will now show one step of the entanglement circuit: the first CNOT operation. At the beginning of the circuit, the two qubit logical space looks like the following:

$$\frac{1}{\sqrt{2}}\left(|0,0\rangle_L^1+|1,1\rangle_L^1\right)\otimes|0,0\rangle_L^1=\frac{1}{\sqrt{2}^3}\left(|\bar{0}\rangle+|\overline{15}\rangle+|\bar{5}\rangle+|\overline{10}\rangle\right)\otimes\left(|\bar{0}\rangle+|\overline{15}\rangle\right) \tag{5.21}$$

The simplest way to understand the action of the transversal CNOT is to define it's mapping of the basis states of the two qubit logical space. The mapping is the following:

$$\mathrm{CNOT}(1,2)\left[|\bar{i}\rangle\otimes|\bar{j}\rangle\right]=\begin{cases}|\bar{i}\rangle\otimes|\bar{j}\rangle & \text{if } \bar{i}=\bar{0}\\ |\bar{i}\rangle\otimes|\bar{j}\oplus\overline{15}\rangle & \text{if } \bar{i}\neq\bar{0}\end{cases} \tag{5.22}$$

which, regarding the state in equation 5.21, shows that the action of the transversal NOT simply maps the basis states in the following way: $|\bar{0}\rangle \leftrightarrow |\overline{15}\rangle$ and $|\bar{5}\rangle \leftrightarrow |\overline{10}\rangle$. Applying this map to the initial state two logical qubit state, we get:

$$\begin{aligned}\mathrm{CNOT}(1,2)\left[\frac{1}{\sqrt{2}}\left(|0,0\rangle_L^1+|1,1\rangle_L^1\right)\otimes|0,0\rangle_L^1\right]=&\frac{1}{\sqrt{2}^3}\left(|\bar{0}\,\bar{0}\rangle+|\bar{0}\,\overline{15}\rangle+|\overline{15}\,\overline{15}\rangle+|\overline{15}\,\bar{0}\rangle\right.\\ &\left.+|\bar{5}\,\bar{5}\rangle+|\bar{5}\,\overline{10}\rangle+|\overline{10}\,\overline{10}\rangle+|\overline{10}\,\bar{5}\rangle\right)\end{aligned} \tag{5.23}$$

from this, it should be fairly clear how we are able to generate the state $|0,0\rangle_L^2$ shown in equation 5.16 . Because we generated states labeled by all permutations of pairs of $\{\bar{0},\,\overline{15}\}$ and $\{\bar{5},\,\overline{10}\}$, any additional $|0,0\rangle_L^1$ state transversally entangled (CNOT) with the first logical qubit will generate states represented by labels of all triples, quadruples, ect. of $\{\bar{0},\,\overline{15}\}$ and $\{\bar{5},\,\overline{10}\}$ which is exactly what we need in equation 5.16.

Now, let us recap the whole process. First, one fault-tolerantly prepares four $|0,0\rangle_L^1$ states. A transversal Hadamard is applied to one of the logical qubits and correction is applied fault-tolerantly based on the measurement outcome of $\bar{X}_a^1\bar{X}_b^1$

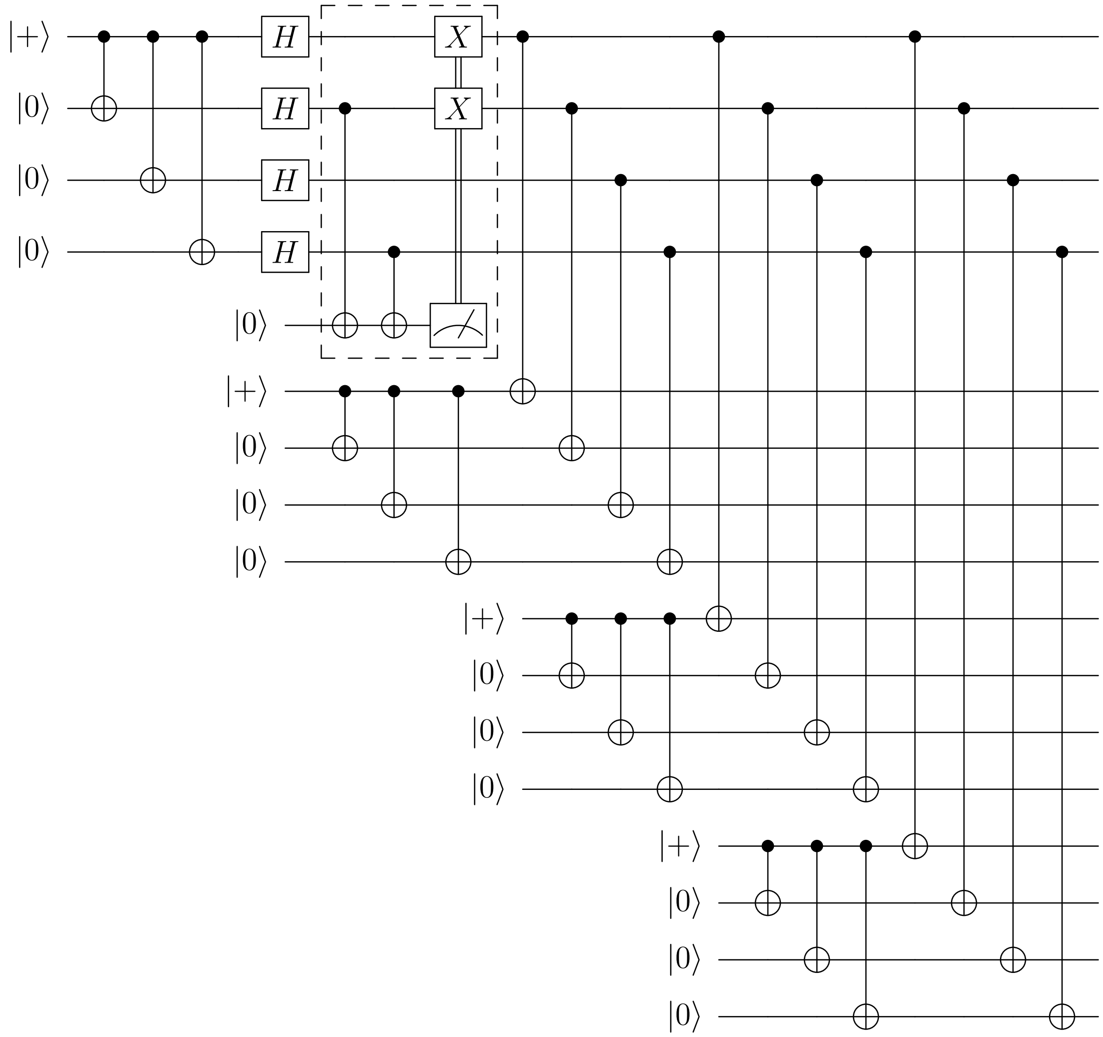


Figure 5.7: Total circuit encoding the level-2 error detection code. The dashed box indicates a fault-tolerant measurement (which implies multiple measurements) of the gauge operator which will be used for correction.

generating a plus-like state. The plus-like state and the other three $|0,0\rangle_L^1$ states are run through an encoding circuit similar to a transversal cat state preparation resulting in a level-2 encoded four-qubit error detecting code. The full encoding circuit is show in figure 5.7.

5.5 The Tilted 13-Qubit Surface Code

The 13-qubit surface code is shown in figure 5.8. It has the same stabilizer generators and code space as the 17-qubit rotate surface code (shown in figure 3.1a). It differs from the 17-qubit code in that it only uses 4 ancilla qubits to measure

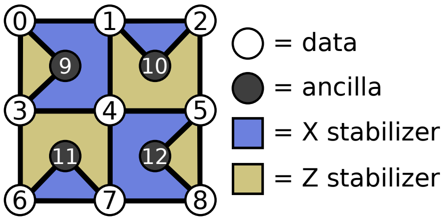


Figure 5.8: The rotated-13 (distance-3) surface code. Labels are consistent with figures 2.4 and 3.1a.

the stabilizers. Therefore, each ancilla qubit is used to measure two stabilizers: one weight-4 stabilizer and one weight-2 stabilizer. The larger depth circuit for measuring the stabilizer generators results in a lower pseudothreshold than the 17-qubit code for circuit-level depolarizing noise [2].

5.5.1 The Space Degeneracy of the Tilted 13-Qubit Surface Code

Now to discuss the degeneracy of the spaces $\mathcal{S}_{(L+1)}$, $\mathcal{S}_{(\psi+1)}$, and $\mathcal{S}_{(\psi+2)}$ for the 13-qubit surface code. First, the logical state is generated utilizing the $H(C_2^\perp)$ parity-check matrix constructing the X stabilizers of the surface code resulting in the basis states:

$$\begin{aligned}
|0\rangle_L = \frac{1}{\sqrt{2}^4}\Big(&|000000000\rangle + |110110000\rangle + |011000000\rangle + |101110000\rangle + \\
&|000000110\rangle + |110110110\rangle + |011000110\rangle + |101110110\rangle + \\
&|000011011\rangle + |110101011\rangle + |011011011\rangle + |101101011\rangle + \\
&|000011101\rangle + |110101101\rangle + |011011101\rangle + |101101101\rangle\Big)
\end{aligned} \tag{5.24}$$

$$|1\rangle_L = X^{\otimes 9}|0\rangle_L$$

which we can again use to construct our logical code space as in equation 5.9. Note that state $|1\rangle_L$ is the $|0\rangle_L$ state under the action of the bitwise-NOT on every binary string within it's sum. Due to this relation, there is an equivalence in the action of single-qubit errors to both $|0\rangle_L$ and $|1\rangle_L$ with respect to the degeneracy of the single-error spaces. For single-qubit errors, there is equivalence of the action of X errors, $X_1|0\rangle_L = X_2|0\rangle_L$ and $X_6|0\rangle_L = X_7|0\rangle_L$, and also degeneracies for Z errors, $Z_0|0\rangle_L = Z_3|0\rangle_L$ and $Z_5|0\rangle_L = Z_8|0\rangle_L$. Y errors are completely non-degenerate.

We can therefore construct the failure criteria pertaining to the logical codeword space in the following manner:

$$\begin{aligned}
&\mathcal{S}_L = \{|0\rangle_L,\, |1\rangle_L\} \\
&\mathcal{S}_{L+1} = \mathcal{S}_L \cup \{X_i|0\rangle_L,\, X_i|1\rangle_L,\, Y_j|0\rangle_L,\, Y_j|1\rangle_L,\, Z_k|0\rangle_L,\, Z_k|1\rangle_L\}
\end{aligned} \tag{5.25}$$

where the indices i, j, k run over all data qubit indices except for the cases where $i = 2, 7$ and $k = 3, 8$. Similarly, for the criteria more natural for the purpose of wavefunction simulations the spaces can be constructed like so:

$$\begin{aligned}
&\mathcal{S}_\psi = \{|\Psi_0\rangle\} \\
&\mathcal{S}_{\psi+1} = \mathcal{S}_\psi \cup \{X_i|\Psi_0\rangle,\, Y_j|\Psi_0\rangle,\, Z_k|\Psi_0\rangle\} \\
&\mathcal{S}_{\psi+2} = \mathcal{S}_\psi \cup \{X_i|\Psi_0\rangle,\, Z_k|\Psi_0\rangle,\, X_iZ_k|\Psi_0\rangle\} \text{ s.t. } i \neq 2, 7;\; k \neq 3, 8
\end{aligned} \tag{5.26}$$

where, again, the indices run over all qubits except for special cases. The assessment of the surface code will incorporate computing the overlap of output wavefunctions with the aforementioned correctable error spaces. Because the surface code is an error *correcting* code, the success criteria P_{code} and fidelity ($\mathcal{F}^2$) will be of interest as well.

5.5.2 Simulation of the 13-Qubit Tilted Surface Code

There is some freedom to how one can schedule the syndrome extraction routine. Because of this, nine variations in the schedule of stabilizer measurement have been simulated, labeled A-D. These routines (A-D) are “single-shot” detection and correction cycles where three rounds of stabilizer measurement are performed and decoding/correction is performed once which are shown schematically in figure 5.9 under the appropriate labels. The first three of these are different permutations of X and Z stabilizer measurement (labeled M_X and M_Z) and error correction, C_X and C_Z. Note that M_X (M_Z) denotes the measurement of **all** X (Z) stabilizers at the given time step. Schedule D is unique in that the circuit is scheduled according to the ancilla measured and not the stabilizer type. Equivalently, this

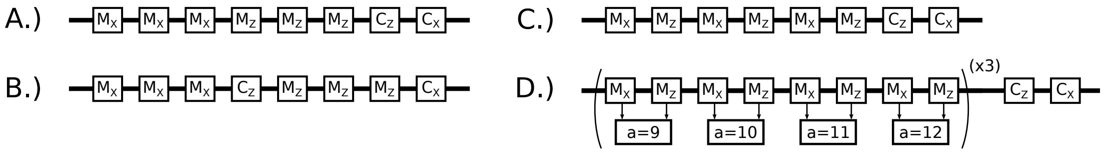


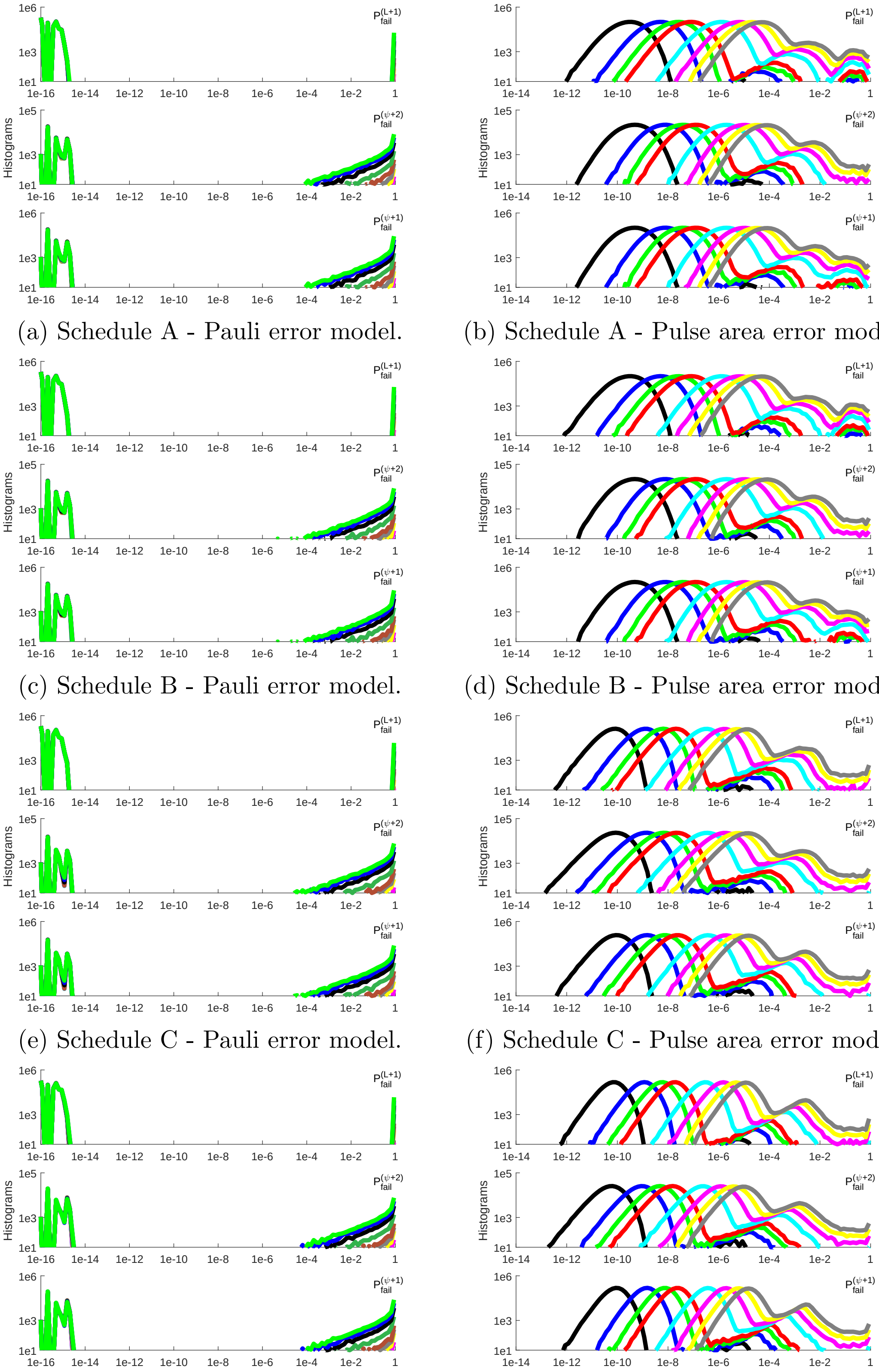
Figure 5.9: Stabilizer measurement schedules studied. The symbols M and C denote stabilizer measurement and error correction, respectively, of the type indicated by their subscripts. Schedules A-D were implemented in "single-shot" error correction routines which were used to generate the data in figures 5.10, 5.14, 5.12, and 5.15.

can be viewed as measurement scheduled by the four quadrants of the surface code with quadrant scheduling dictated by the ancillary qubit labels. A variety of syndrome extraction routines were incorporated to gain insight into the role of the circuit schedule on the effect of the error model, particularly in the presence of coherent errors. For simulation, a logical input state is generated, stabilizer measurement is performed with perfect ancilla preparation, and the failure metrics of the error-corrected state are calculated.

The Pauli Error Model

The "single-shot" error correction routines were each run 1×10^6 times at each error rate $p = 0.000001, 0.0000025, 0.000005, 0.0000075, 0.00001, 0.000025, 0.00005, 0.000075, 0.0001, 0.0002, 0.0004, 0.0006, 0.0008$. The histograms generated from the surface code simulations, in figure 5.10, contain the same properties as the four-qubit error detection code in figure 5.3. Because this circuit implements error correction, the tail in the distributions of $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)} > 10^{-8}$ for the surface are smaller than those of four-qubit error detection code and the Steane code. This smaller spread in criteria is due to the aforementioned "funneling" behavior of the degenerate space plus the correction operations that drive the wavefunction to specific regimes of the Hilbert space.

Also note that there is no noticeable discrepancy between the distributions of $P_{\text{fail}}^{(\psi+1)}$ and $P_{\text{fail}}^{(\psi+2)}$ which is due to the fact that the surface code is a CSS code, i.e. a code that partitions the subspaces stabilized by the X and Z stabilizers



(a) Schedule A - Pauli error model.

(b) Schedule A - Pulse area error model.

(c) Schedule B - Pauli error model.

(d) Schedule B - Pulse area error model.

(e) Schedule C - Pauli error model.

(f) Schedule C - Pulse area error model.

(g) Schedule D - Pauli error model.

(h) Schedule D - Pulse area error model.

Figure 5.10: Histograms of the failure criteria for the four "single shot" error correction schedules. 1×10^6 samples were accumulated per error rate. Pauli error rates: p = 0.000001, 0.0000025, 0.000005, 0.0000075, 0.00001, 0.000025, 0.00005, 0.000075, 0.0001, 0.0002, 0.0004, 0.0006, 0.0008. Pulse area error strengths: σ = 0.0025, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, 0.05.

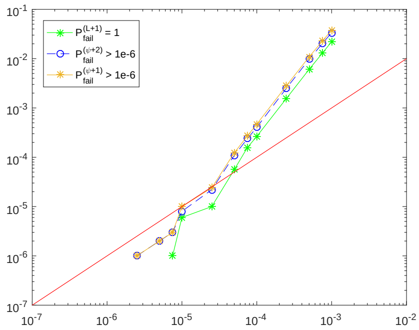


Figure 5.11: Threshold obtained when only the $|0\rangle_L$ state is generated as the initial state of the quantum circuit simulation using Schedule C. The threshold of $\approx 4 \times 10^{-5}$ is equivalent to a reported threshold value of $\approx 1.2 \times 10^{-4}$ in the language of [2].

independently, and $\mathcal{S}_{\psi+1} \subset \mathcal{S}_{\psi+2}$. There are no noticeable discrepancies in the histograms between the four different error correction schedules when correcting Pauli errors (figure 5.10). Figure 5.14 shows the deviation between the two types of failure criteria, $P_{\text{fail}}^{(L+1)}$ and $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)}$ on a per-run basis. The distributions have a similar profile for all circuit schedules (A-D) for the Pauli error model. It also clearly shows that the success criteria $P_{\text{fail}}^{(L+1)}$ has more of a tendency to overestimate the failure value of the wavefunction simulation relative to the wavefunction criteria in the cases when the two types of failure criteria disagree. Note, however, that the codeword criterion records a higher threshold relative to the wavefunction failure metric (Figure 5.12) indicating that this overestimate occurs in a statistically small number of cases relative to the entire sample.

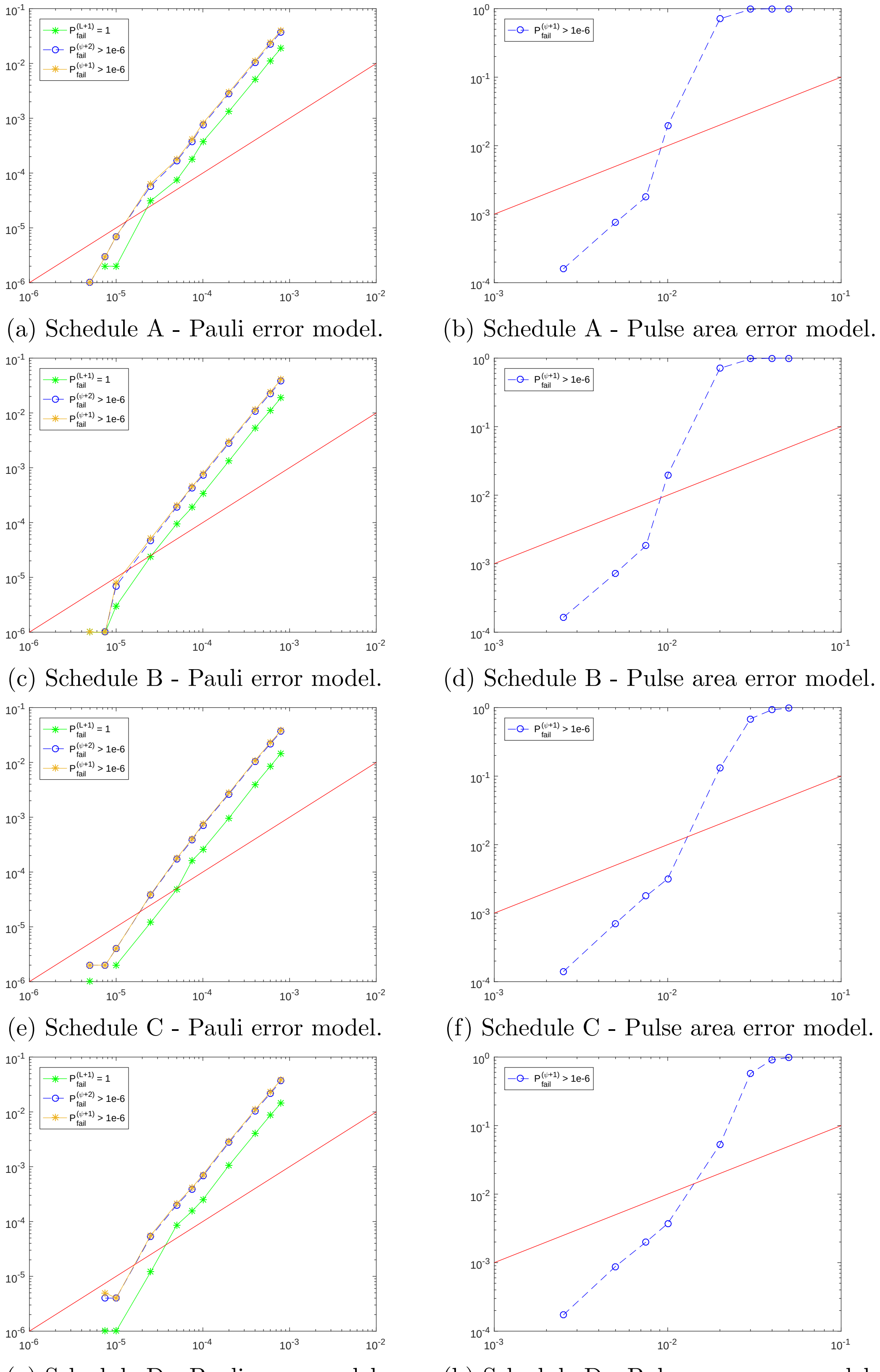
A pseudothreshold of $\approx 3 \times 10^5$ was observed for all four "single-shot" error detection routines by plotting the logical error rate versus the physical error rate. All thresholds for the surface code are shown in figure 5.12. Note that our reported pseudothresholds appear to be below the value reported in [2] of 3×10^{-4}. The simulations in [2] implement $|0\rangle_L$ as in input wavefunction always while the simulations in figure 5.12 randomly sample a vector in the logical codeword space $1/\sqrt{2}\left(\alpha|0\rangle_L + \beta|1\rangle_L\right)$ which is taken as the input wavefunction in the simulations. Also, there is a discrepancy between the labeling of the error rates between this study and the study in [2] where our recorded values of p is equivalent to $p/3$ in [2].

Figure 5.11 provides the threshold (schedule C) for all three failure criteria for simulations where the initial state was always initialized in $|0\rangle_L$. The pseudothreshold obtained using the $P_{\text{fail}}^{(L+1)}$ criteria resulted in value of $\approx 4 \times 10^{-5}$ which, in the language of [2], is reported as $p_{th} \approx 1.2 \times 10^{-4}$; a comparable threshold value. As observed for the Steane code, there is a deviation between the thresholds obtained between the two types of failure metrics: $P_{\text{fail}}^{(L+1)}$ and $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)}$. However, the deviation in thresholds for the Steane code spanned over an order of magnitude while the surface code deviations are smaller.

For completeness, the fidelity squared, P_{code}, and $\mathcal{F}^2/P_{\text{code}}$ criteria for all of the "single-shot" error correction routines are shown in figure 5.15. The overlap with the logical basis states, P_{code}, is binomially distributed (up to floating point precision) between two values: 0 and 1. The fidelity ($\mathcal{F}^2$) is binomially distributed as well but a distribution of values at $0 \leq \mathcal{F}^2 < 0.999$ is observed while $\mathcal{F}^2 = 1$ exhibits a sharp peak. The distribution in the regime $0 < \mathcal{F}^2 < 0.999$ can be attributed to cases where either three-qubit errors have occurred or where a two-qubit error was incorrectly decoded and corrected leading to a logical error. In these cases, there is a swapping between codewords in the $|0\rangle_L$ and $|1\rangle_L$ states in $|\Psi(t)\rangle$ resulting in a contribution of $\alpha * \beta$ cross-terms in the calculations of $\mathcal{F}^2$ which generates a distribution of fidelity values because α and β values are drawn from a random distribution. The metric $P_{\text{code}}/\mathcal{F}^2$ corroborates this claim, recording values < 1.0, indicating that the output wavefunctions of the simulations are rotated versions (in those cases there is an angle swap $\alpha \leftrightarrow \beta$) of $|\Psi(0)\rangle$ within the codeword space.

5.5.3 The Pulse Area Model

Surface code simulations using schedules A-D performed at the following error strengths, σ: 0.0025, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, 0.05 (1×10^6 samples per error rate). Broad distributions for the failure criteria are again observed as



(a) Schedule A - Pauli error model.

(b) Schedule A - Pulse area error model.

(c) Schedule B - Pauli error model.

(d) Schedule B - Pulse area error model.

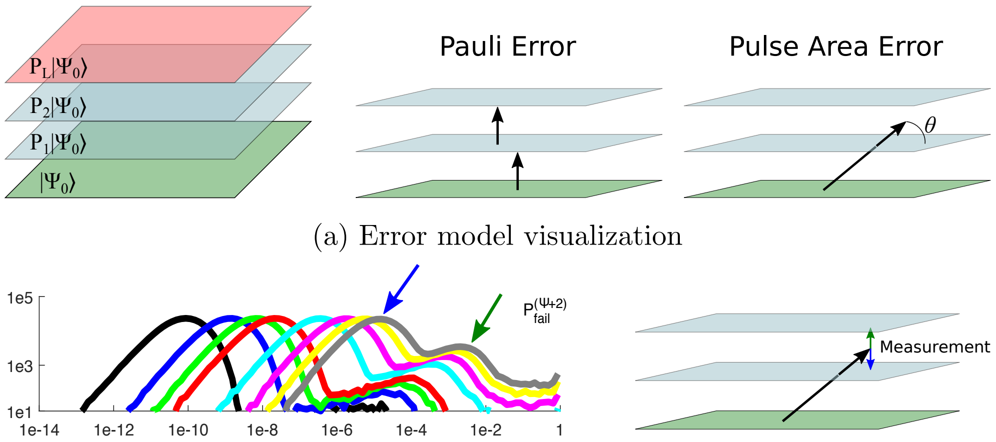
(e) Schedule C - Pauli error model.

(f) Schedule C - Pulse area error model.

(g) Schedule D - Pauli error model.

(h) Schedule D - Pulse area error model.

Figure 5.12: Pseudothresholds for the “single shot” error correction schedules. Pauli error rates: p = 0.000001, 0.0000025, 0.000005, 0.0000075, 0.00001, 0.000025, 0.00005, 0.000075, 0.0001, 0.0002, 0.0004, 0.0006, 0.0008. Pulse area error strengths: σ = 0.0025, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, 0.05.



(a) Error model visualization

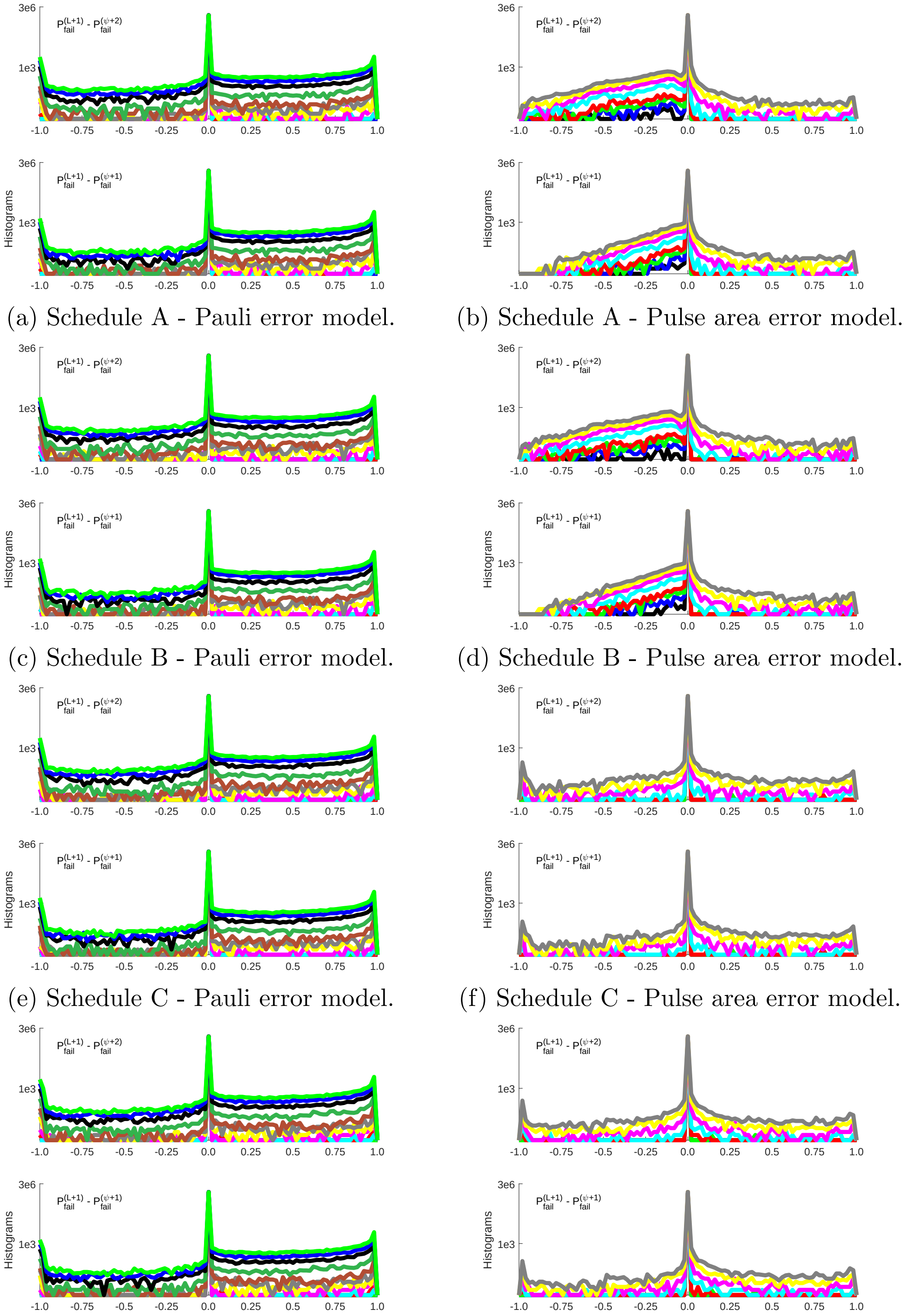
(b) The effect of measurement on the coherent errors

Figure 5.13: a.) Visualization of the action of both the Pauli and pulse area model on the input wavefunction. (left) One may use a set of planes the represent the Pauli operator distance away from applying a logical operator to the initial state defined by $P_1|\Psi_0\rangle(=\mathcal{S}_{\psi+1})$, $P_1|\Psi_0\rangle(\subset \mathcal{S}_{\psi+1})$, and $P_L|\Psi_0\rangle$ (logical failure). The action a Pauli error (middle) serves as a direct map between the states defined by the aforementioned planes. Coherent errors (right) however do not necessarily map directly between planes but can generate a superposition between the states contained within two separate planes, which can be visualized as a rotation out of the plane of the initial state. b.) The action of measurement after application of the coherent error serves as a projector on any element of the codewords that is in the aforementioned superposition state between the two planes. Sampling of the output states will provide the relative amplitudes of superposition states dependent on the strength of coherent error represented by the angle of rotation outside of a given plane in our visualization, which can serve to explain the emerging bimodal behavior of the data with increasing error strength.

shown in figure 5.10 and appear to follow a log-normal distribution in a majority of the cases. Circuit schedules A and B separate the measurement of the X and Z stabilizers (figure 5.9) and circuit schedules C and D interleave the X and Z stabilizer measurement; these two types of syndrome extraction provide different behaviors outside the log-normal distribution-like regime of the histograms. For schedules A and B, there is a discrepancy between the histograms utilizing the failure criteria $P^{(L+1)}_{\text{fail}}$ and $P^{(\psi+2)}_{\text{fail}}$ or $P^{(\psi+1)}_{\text{fail}}$ (figure 5.10). The interleaved syndrome extraction routines (C and D) show consistency in the histograms for all three failure criteria. Regardless of the syndrome extraction routine, the bulk of the samples do not incur a logical error, i.e. the output wavefunction is not a state completely orthogonal to the input wavefunction. Furthermore, the profiles of the histograms follow a 2-3 peak behavior depending on the syndrome extraction routine. Figure

5.13 contains a schematic explanation of this multi-peak behavior observed by the pulse area model. If one views the space spanned by the initial basis vectors and each of the spaces spanned by the application of one $(P_1|\Psi_0\rangle/\mathcal{S}_{\psi+1})$, two ($P_2|\Psi_0\rangle \subset \mathcal{S}_{\psi+2})$, and three Pauli operators $(P_L|\Psi_0\rangle$, logical operator applied mod stabilizers) to the initial basis space as stacked planes with some defined distance from one another defined by the weight of the Pauli operator required to map from one plane to another (Figure 5.13a), we can view action of both the Pauli and pulse area noise model on the codespace in an intuitive way. The action of a Pauli error serves as a direct map between the aforementioned planes. The pulse area model (and coherent errors for that matter) can be visualized as a rotation of the state off of the current plane towards another plane with the magnitude of the angle off the plane being defined by the strength of the coherent error as shown in figure 5.13a which signifies the generation a superposition between the elements of the two planes. The magnitude of the angle off of the plane is representative of the strength of the coherent error. Measurement of the ancillary state (stabilizer) then serves to project the kets in the superposition constructing the basis states to one of the planes that the state is rotated between with a probability proportional to the distance of the state/magnitude of the angle off of one of the planes. This appears to the be the cause of the emergence of the multi-peak behavior with increasing error strength in the histograms for the coherent errors in figure 5.10. The histograms are sampling outputs from projections of the kets in the superposition of the codewords to separate planes giving the multi-peak behavior (figure 5.13b) with the relative magnitude of these peaks containing an error strength dependence that trends to higher failure values at higher error strengths (a larger angle of rotation away from the "no error plane").

The plot of the deviations between $P_{\text{fail}}^{(L+1)}$ and $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)}$ (per-run) in figure 5.14 provides some insight into this behavior. The deviations in the schedules A and B are weighted more heavily towards the regime $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)} > P_{\text{fail}}^{(L+1)}$



(a) Schedule A - Pauli error model.

(b) Schedule A - Pulse area error model.

(c) Schedule B - Pauli error model.

(d) Schedule B - Pulse area error model.

(e) Schedule C - Pauli error model.

(f) Schedule C - Pulse area error model.

(g) Schedule D - Pauli error model.

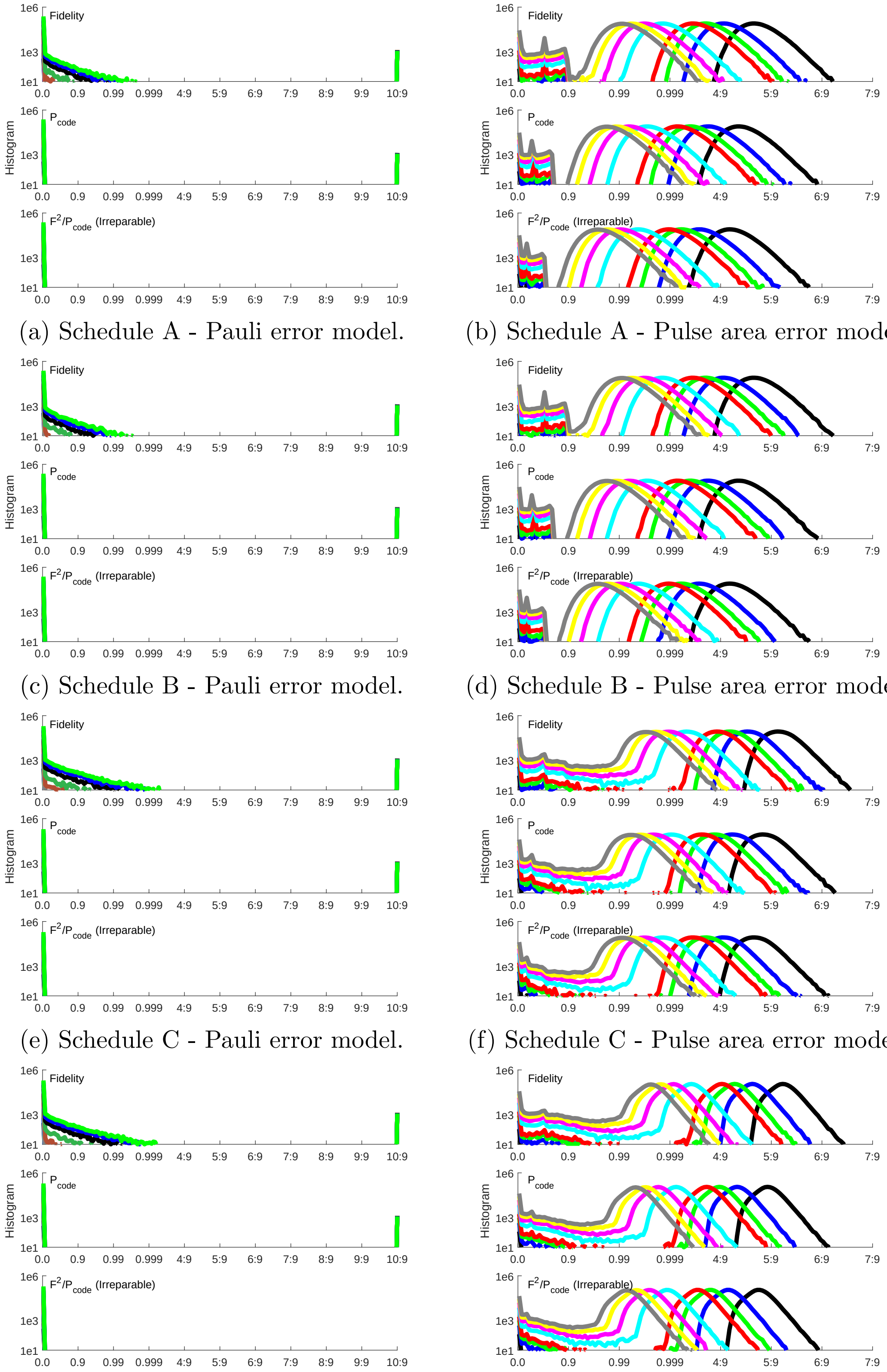
(h) Schedule D - Pulse area error model.

Figure 5.14: Histograms of the deviation between the two types of failure criteria for the four "single shot" error correction schedules. 1×10^6 samples were accumulated per error rate. Pauli error rates: $p = 0.000001$, 0.0000025, 0.000005, 0.0000075, 0.00001, 0.000025, 0.00005, 0.000075, 0.0001, 0.0002, 0.0004, 0.0006, 0.0008. Pulse area error strengths: $\sigma = 0.0025$, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, 0.05.

resulting in the noticeable deviations between the histograms for the two types of failure criteria. In comparison, the deviations for schedules C and D are more symmetrically distributed about $P_{\text{fail}}^{(\psi+1)}/P_{\text{fail}}^{(\psi+2)} = P_{\text{fail}}^{(L+1)}$. Furthermore, the deviation distributions for schedules C and D have remarkably similar profiles to the deviations obtained from the same circuits under the influence of the Pauli error model, which provides small evidence that the interleaved syndrome extraction routines exhibit more Pauli-like output statistics relative to the non-interleaved techniques. This is quite intuitive as the interleaved measurement does no allow for one type of coherence to commute through the first three rounds of syndrome extraction like it can for the non-interleaved techniques. These coherences can slowly build up after syndrome extraction rounds and, in such cases, increase the error in the output wavefunction ($P_{\text{fail}} > 10^{-3}$ regime) which is illustrated by the additional bump in sampling of histograms in figure 5.10b and d relative to the histograms from 5.10f and h.

By defining a failure criteria of $P_{\text{fail}}^{(\psi+1)} > 10^{-6}$, pseudothresholds were obtained for the surface code under the influence of the pulse-area error model as shown in figure 5.12. Consistent pseudothresholds in the range $8 \times 10^{-3} < p_{th} < 0.5 \times 10^{-2}$ were obtained for all circuit schedules. The sigmoidal shape of the threshold was again observed. It is particularly interesting that the pseudothresholds for the pulse-area noise model appear to be insensitive to the type of error correction routine implemented for this criteria with reported pseudothreshold values near 1×10^{-2} for the Steane, $[\![4, 2, 2]\!]$, and surface code simulations. This may hint that the criteria $P_{\text{fail}}^{(\psi+1)} > 10^{-6}$ is not appropriate for assessment of the failure of an error correction/detection routine. This metric appears to be similar to a "fidelity-like" assessment (a metric like $\langle 1 - \mathcal{F} \rangle$ of the success of a code, which has been shown to deviate greatly from standard metrics implemented for unitary errors such as the diamond distance, $D_{\diamondsuit}$, and the trace distance, D_{tr} [37, 38, 113, 162].

Figure 5.15 displays the histograms of the failure metrics P_{code}, $\mathcal{F}^2$, and $P_{\text{code}}/\mathcal{F}^2$



(a) Schedule A - Pauli error model.

(b) Schedule A - Pulse area error model.

(c) Schedule B - Pauli error model.

(d) Schedule B - Pulse area error model.

(e) Schedule C - Pauli error model.

(f) Schedule C - Pulse area error model.

(g) Schedule D - Pauli error model.

(h) Schedule D - Pulse area error model.

Figure 5.15: Histograms of the code space failure criteria for the four "single shot" error correction schedules. 1×10^6 samples were accumulated per error rate. Pauli error rates: $p =$ 0.000001, 0.0000025, 0.000005, 0.0000075, 0.00001, 0.000025, 0.00005, 0.000075, 0.0001, 0.0002, 0.0004, 0.0006, 0.0008. Pulse area error strengths: $\sigma =$ 0.0025, 0.005, 0.0075, 0.01, 0.02, 0.03, 0.04, 0.05.

for the pulse-area noise model applied to the surface code. Note that no output wavefunction landed completely back into the codeword space, i.e. $P_{\text{code}} = \mathcal{F}^2 \neq 1$ for all simulations. Again, there are noticeable differences in the results of the circuits implementing schedules A and B versus C and D. All schedules have similar profiles in the "log-normal-like" sections of the distributions but with a noticeable shift to higher fidelity/P_{code} values for the center of the distributions with schedules C and D. This may be due an ability of the interleaved measurement routines to break up coherences more effectively than a syndrome extraction routine that repeats one type of stabilizer measurement allowing for coherent errors of certain types to commute and potentially build in amplitude through the successive steps of syndrome extraction but this is simply conjecture. The distinction in the shape of the distributions for the two types of stabilizer measurement routines occurs outside of the log-normal-like regime of the distributions. For schedules A and B, there is a noticeable spike in the number of samples with $\mathcal{F}^2/P_{\text{code}} \approx 0.95$ and a drop in output wavefunctions with $\mathcal{F}^2/P_{\text{code}} \approx 0.90$. As shown by the deviations in figure 5.14, schedules A and B result in output wavefunctions that don't appear to symmetrically sample about the codeword space which appears to manifest itself in the non-uniform distribution of the computed fidelity and codeword overlap metrics. For the interleaved stabilizer measurement routines, there is a generally flat distribution of $\mathcal{F}^2/P_{\text{code}}$ values in the equivalent regime much like the results of the Steane code. This my be attributed to the symmetry of the output wavefunction sampling about the codeword space as shown in figure 5.14. It would be particularly interesting to be able to determine which output states are not sampled by schedules A and B to gain insight into how the error correction routine influences the action of the coherent error on the input wavefunction.

Chapter 6

Maximum Likelihood Decoding of Topological Codes

The following chapter contains unpublished work that was performed at Georgia Institute of Technology and the T.J. Watson Research Center (IBM) during an internship for the 4.6.12 color code and doubled color code work, respectively.

While the surface code has been shown to be a robust platform for error correction. Universal fault-tolerant quantum computation with the surface code encounters some difficulties. Particularly, the restricted transversal gate set of the surface code requires magic state distillation for both the S and T gate for universality which, as we have seen in chapter 4, is costly in quantum resources. Therefore, other error-correcting codes may be a better platforms for fault-tolerant computation in practice.

Color codes are an intriguing alternative to surface codes. They have error thresholds for depolarizing noise near the surface code thresholds but also allow a transversal implementation of the Clifford group. Note that not all members of the family of color codes have been investigated; the 4.6.12 family of color codes remained to be studied. In this work, we build up tools to perform such a study. In additional to standard color codes, a family of subsystem codes built from color codes, known as doubled color codes, was investigated as well. This subsystem code is of interest due to its ability to achieve a universal fault-tolerant gate set with gauge fixing opposed to magic state distillation.

The outline of this chapter is the following. First, we discuss the error model for computation. We discuss the decoding technique used: the optimal Maximum Likelihood Decoding algorithm. Finally, thresholds for the 4.6.12 color code and doubled color codes are provided.

6.1 Error Model

We enforce the following conditions for the simulations:

- Syndromes are measured at integer time steps $t = 1, 2, ..., L$.
- Each measurement results in a wrong outcome with probability p_{meas}.
- Memory errors occur at half-integer time steps $t - 1/2$.
- Only X-type errors with error rate p_{mem}.
- The initial state, $t = 0$, has no errors.
- The final state, $t = L$, has no syndrome errors.

The perfect measurement of the final state is required to evaluate the success of decoding at time $t = L$ and can be justified as the measurement of every qubit in the code after some memory time $t = 0$ to $t = L - 1$. The restriction to only X-type errors is justified because color codes are CSS codes and, therefore, X and Z errors may be detected and corrected independently. Note that this error model differs slightly to the standard depolarizing channel in that Y errors occur with probability p^2. For simulations where $p_{meas} = 0$, the error model applied is the code capacity error model. Otherwise, the model applied is the phenomenological error model where measurement errors occur but ignore circuit error propagation.

6.2 Maximum Likelihood Decoding

6.2.1 Background and Notation

The ML decoder takes the measured stabilizer syndromes as an input and outputs a candidate error C such that (i) C agrees with the final measured syndrome and (ii) the probability that C is equivalent (modulo some stabilizer operators) to the actual accumulated error is maximal, conditional on the observed stabilizer syndrome. This decoder relies on tracking error coset probabilities during the computation. We discuss an implementation of the ML decoder below.

Let $E \in \mathbb{F}_2^n$ be an arbitrary X-type error where the error is represented by a binary vector where 1's represent the application of the error to the qubit i, indexed by the bit E_i in E. The goal is to show that there is a stabilizer S exists such that an error $E' = E + S$ has support only on a subset of c bits representing the code. The error E' shall be referred to as the *reduced error* corresponding to the full error configuration E and corresponds to an error coset formally. Note that any additions involving binary vectors, such as E, with be assumed to be additions modulo two throughout the document.

Define a binary matrix R of size $n \times c$ such that the b-th column of the matrix corresponds to the reduced error corresponding to the single error with support on the b-th bit of the full error. By assumption above, a full error E appears with probability $p_{mem}^{|E|}\left(1 - p_{mem}\right)^{n-|E|}$, where p_{mem} is the memory error rate. The reduced errors are pulled from a full error distribution, $\pi(e)$ in the following manner:

$$P[E'] = \sum_{E:ER=E'} p_{mem}^{|E|}\left(1 - p_{mem}\right)^{n-|E|} \text{ , where } E' \in \mathbb{F}_2^c \tag{6.1}$$

where E and E' are row vectors. Because the reduced errors are equivalent modulo some set of stabilizer operators, they have the same syndromes and result in the same decoding outcome. From this point on, we will consider only the reduced error model such that errors only occur on a subset of c qubits of the code with error $E \in \mathbb{F}_2^c$ that occurs with probability $P[E]$.

Consider a given time step t. Let $s_a \in \mathbb{F}_2$ be the stabilizer syndrome of stabilizer S_a at the given time step. The stabilizer syndrome is $s = (s_1, s_2, ..., s_N) \in \mathbb{F}_2^N$ where N is the number of stabilizers of the code. Define a mapping matrix Q which maps the stabilizers to a subset of the stabilizer group that properly decodes errors on the subset of qubits (the reduced errors) of the color code or, equivalently $s = s'Q$ where s' is the new set of stabilizers. As listed above, each syndrome error occurs with probability $p_{meas}^{|e|}\left(1-p_{meas}\right)^{N-|e|}$ where, in this case, $e \in \mathbb{F}_2^N$ is the syndrome measurement error. Then, a syndrome measurement error, e', occurs with probability:

$$P_{syn}[e'] = \sum_{e:eQ=e'} p_{meas}^{|e|}\left(1-p_{meas}\right)^{N-|e|} \tag{6.2}$$

We now have sufficient information to describe the maximum likelihood decoding algorithm

6.2.2 Maximum Likelihood Decoding Algorithm

Below we will discuss the decoding algorithm and then finish with a workflow of the algorithm. Let $E'(t)$ be the reduced error that occurs at time step $t-1/2$ where $t \in \{1, 2, ..., L\}$. We define a full error history as:

$$\hat{E}' = \{E'(1), E'(2), ..., E'(L)\} \tag{6.3}$$

Then, define the accumulated reduced error at time step t in the following manner:

$$F(t) = \sum_{u=1}^{t} E'(u) \tag{6.4}$$

Note that, by assumption, $t = 0$ contains no errors and the final syndrome measurement provides the final syndrome $F(L)$ with perfect measurements. Let $\mathcal{O}$ and $\mathcal{E}$ be the sets of the odd-weight and even-weight reduced errors. Each set $\mathcal{O}$ and $\mathcal{E}$ contains a unique reduced error with a given syndrome. Therefore, it is our job to infer whether $F(L) \in \mathcal{O}$ or $F(L) \in \mathcal{E}$.

Let $s(t)$ be the syndrome at time t and $\hat{S} = \{s(1), s(2), ..., s(L)\}$ be the full syndrome history. Now, define a $c \times N'$ binary matrix, A, such that the i-th row of A is the stabilizer syndrome caused by a single-qubit X error on qubit i of the reduced set of qubits, $i = 1, 2, ..., c$. If the syndrome measurements were noiseless, then the stabilizer syndrome observed at time step t would be $F(t)A$. So, a pair of $\hat{E}, \hat{S}$ occur with the following conditional probabilities:

$$\Pr\left[\hat{E}|\hat{S}\right] = \prod_{t=1}^{L} P\left[E(t)\right] \prod_{t=1}^{L} P_{syn}\left[s(t) + F(t)A\right] \tag{6.5}$$

Then define the probabilities:

$$P_{odd}(\hat{S}) = \sum_{\hat{E}:F(L)\in\mathcal{O}} \Pr\left[\hat{E}|\hat{S}\right] \quad \text{and} \quad P_{even}(\hat{S}) = \sum_{\hat{E}:F(L)\in\mathcal{E}} \Pr\left[\hat{E}|\hat{S}\right] \tag{6.6}$$

The decoder then decides the $F(L) \in \mathcal{O}$ if $P_{odd} \geq P_{even}$ and $F(L) \in \mathcal{E}$ otherwise. To show how to compute the probabilities in Eqn. (6.6), we substitute $E(t) = F(t) + F(t-1)$ in Eqs. (6.5,6.6) to get:

$$P_{odd}(\hat{S}) = \sum_{F(1),\ldots,F(L):F(L)\in\mathcal{O}} P\left[F(t) + F(t-1)\right] \prod_{t=1}^{L} P_{syn}\left[s(t) + F(t)A\right] \tag{6.7}$$

where the sum runs over all L-tuples of binary vectors $F(t) \in \mathbb{F}_2^c$ such that $F(0) = 0^c$ and $F(L) \in \mathcal{O}$. Define a real-valued matrix Γ of size $2^c \times 2^c$ such that the rows and columns of Γ are labeled by the binary vectors $f, g \in \mathbb{F}_2^c$ and

$$\Gamma_{f,g} = P\left[f + g\right] \tag{6.8}$$

Define an additional binary matrix $\Delta^{(t)}$ of size $2^c \times 2^c$ such that

$$\Delta^{(t)}_{f,f} = P_{syn}\left[s(t) + fA\right] \tag{6.9}$$

this results in the following probabilities

$$\begin{aligned} P_{odd}(\hat{S}) &= \langle\mathcal{O}|\Gamma\Delta^{(L-1)}\Gamma\ldots\Gamma\Delta^{(2)}\Gamma\Delta^{(1)}\Gamma\big|0^c\rangle \\ P_{even}(\hat{S}) &= \langle\mathcal{E}|\Gamma\Delta^{(L-1)}\Gamma\ldots\Gamma\Delta^{(2)}\Gamma\Delta^{(1)}\Gamma\big|0^c\rangle \end{aligned} \tag{6.10}$$

where

$$|\mathcal{O}\rangle \equiv \sum_{x\in\mathcal{O}} |x\rangle \quad \text{and} \quad |\mathcal{E}\rangle \equiv \sum_{x\in\mathcal{E}} |x\rangle \tag{6.11}$$

At this point, computation of P_{odd} and P_{even} required matrix-vector multiplications

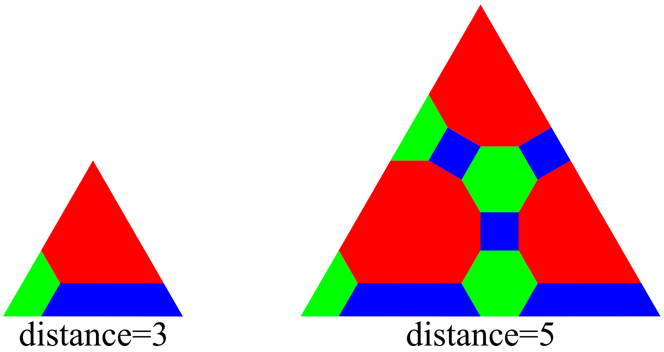


Figure 6.1: First two members of the 4.6.12 topological color code family.

for matrices of size $2^c \times 2^c$ and, therefore, would require $2^{2c}L$ arithmetic operations. The application of Walsh-Hadamard transform, H, on Γ such that we replace Γ with $\hat{\Gamma} = H\Gamma H$ which is diagonal (just like Δ) and can be represented as a vectors altering the computation of the probabilities:

$$P_{odd}(\hat{S}) = \left\langle \mathcal{O}|H\hat{\Gamma}H\Delta^{(L-1)}H\hat{\Gamma}H...H\hat{\Gamma}H\Delta^{(2)}H\hat{\Gamma}H\Delta^{(1)}H\hat{\Gamma}H\middle|0^c\right\rangle \tag{6.12}$$

where P_{even} is calculated analogously. This alters the computation from matrix-vector operations to vector-vector operations with running time $O(Lc2^c)$.

The logical error rate is the probability of a decoding failure per time step. More formally, imagine that one could generate random errors and collect noisy syndrome for an infinite time interval. Let τ be the first time step where the MLD would fail to correct the error if it were applied over the interval $[0, \tau]$. Here, we assume that the decoder has knowledge of the noiseless syndrome of the accumulated error. The random variable τ can be interpreted as the memory time of reliable computation. One expects an exponential decay

$$\Pr[\tau \geq t] \sim \exp(-ct) \tag{6.13}$$

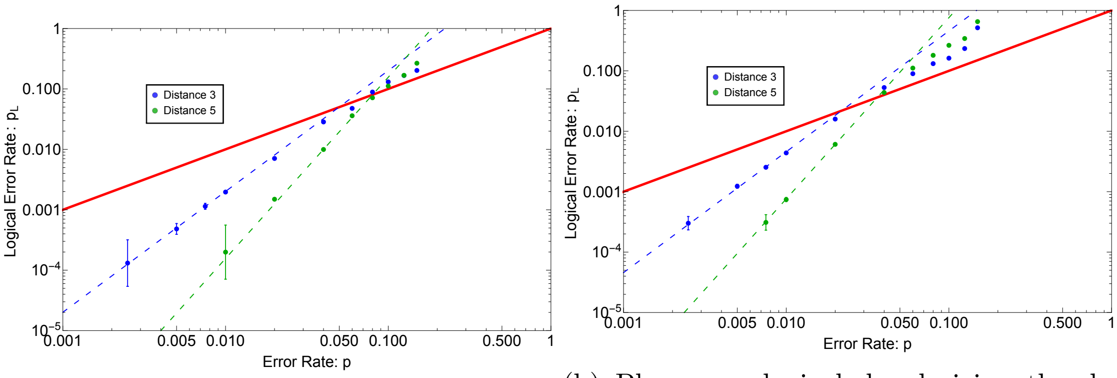
for some constant $c > 0$. The formal definition of the logical error rate, p_l, is then

$$p_l = -\lim_{t\to\infty} \frac{\log[\Pr[\tau \geq t]]}{t} \tag{6.14}$$

In our case for the MLD, we run the simulation for a number of steps $\tau = L$ where L is the first instance of failed decoding. We repeat this process to obtain a set of memory times which are fit to the exponential distribution to obtain the logical error rate, c from (6.13).

6.3 The 4.6.12 Color Code

Up until this point, we have discussed only the surface code when referring to topological quantum computation. Another class of topological subspace stabilizer codes are topological color codes. These codes are constructed from regular or semi-regular cellulations of the surface that are represented as trivalent graphs [163]. In this cellulation, all vertices represent qubits. Again, the cellulation defines the stabilizer group but these codes are self-dual so the faces of the cellulation define both the Z-type and X-type stabilizer generators. See figure 6.1 for the first two members of the 4.6.12 color code family. Now that we have a three-colorable graph, a planar layout of the code will be a triangle with the three corners of the triangle terminating each at a stabilizer of a different color (red, green, and blue in our case). An edge of the code has a boundary color type of the stabilizers that do not terminate at the boundary. The length of each boundary (in number of qubits) again has the intuitive representation of the distance of the code. The X and Z logical operators span from *any* two corners of different color types or from one corner to a boundary of equivalent color type. These codes are of interest because they can implement the full Clifford group transversally which the surface code cannot do. Indeed, upon inspection you will notice that the distance-3 code is simply a graphical representation of the stabilizer group for the Steane code (see equation 2.14).



(a) Code capacity depolarizing threshold

(b) Phenomenological depolarizing threshold

Figure 6.2: Error thresholds for the 4.6.12 topological color code.

6.3.1 Error Thresholds for the 4.6.12 Color Code

The error threshold for the 4.6.12 topological color code was 10.3% and 3.1% for the code capacity and phenomenological error models, respectively. Note that these values are consistent with code capacity [164–167] and phenomenological error thresholds [167, 168] of color codes built from the 4.8.8 and 6.6.6 cellulations of the surface. The code capacity threshold for the 4.6.12 color code is slightly lower than the threshold for the surface code of 10.9% [125]. The phenomenological threshold is also slightly lower than the surface code threshold of 3.3% [169].

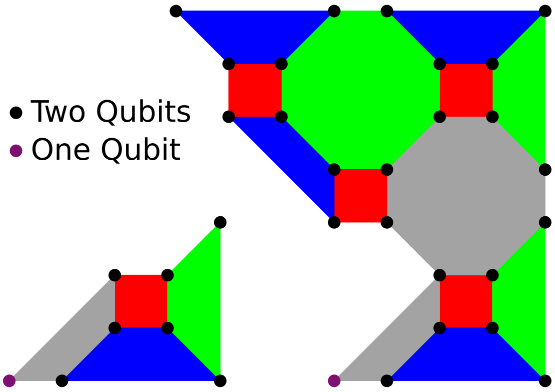
6.4 Doubled Color Codes

Doubled color codes are subsystem stabilizer codes that allow for universal, fault-tolerant quantum computation through a method known as gauge-fixing. This method uses the gauge degrees of freedom of the subsystem code to fault-tolerantly convert between the code spaces of two codes which, together, have a universal gate set. We will start with discussing subsystem codes building from the background of subspace stabilizer codes. Then we will discuss the doubled color code family of codes followed by the phenomenological error threshold for the doubled color code.

6.4.1 Subsystem Codes

Subsystem codes perform quantum error correction through the use of quantum error-correcting subsystems. Consider a CSS code with a given stabilizer group $\mathcal{S}$ and a set of logical operators $\mathcal{L}$. Now consider a case where the structure of the stabilizer group over the n-qubit Hilbert space allowed for the existence of a set of Pauli operators that commute with the elements of the stabilizer group and the logical operators. These operators are called *gauge operators* and which form a group $\mathcal{G}$; the gauge group. The eigenspace of the gauge operators form a subspace of the n-qubit Hilbert space similar to that of subspace codes but not necessarily the +1 eigenspace of all gauge operators: $\mathcal{C}_G = \{\text{span}(\{\,|\psi_i\rangle\,\})\,|\,g|\psi_i\rangle = \pm 1|\psi_i\rangle\ \forall\ g \in \mathcal{G}\}$ where again we are dealing with complex values in the linear span. The code space is a subsystem of this gauge subspace where the stabilizers are constructed from tensor products of the gauge generators: $\mathcal{C}_S = \{\text{span}(\{|\psi_i\rangle\})\,|\,s|\psi_i\rangle = +1|\psi_i\rangle\ \forall\ \bigotimes_i g_i = s \in \mathcal{S}\}$. To perform error correction, the gauge generators are measured and the syndrome for the stabilizers is inferred from these measurements. This can reduce the number measurements required to perform error correction [170–172]. Furthermore, any error in the code that is a product of gauge generators acts trivially on the code space so these degrees of freedom are protected.

Furthermore, these gauge degrees of freedom have been utilized to fault-tolerantly convert between code spaces of two types of quantum error-correcting codes (given both codes satisfy certain properties) allowing for universal fault-tolerant quantum computation. The process of projectively measuring out of and reinitializing these gauge degrees of freedom into and out of the code space is a process known as gauge fixing. One such family of codes that perform this operations are doubled color codes.



(a) ⟦15, 1, 3⟧ and ⟦49, 1, 5⟧ codes.

Stabilizer Group

A	B	C
S	S	0
1	0	1
0	0	S_0

Gauge Group

A	B	C
S	0	0
Even	*Even*	0
L	0	L_0
0	0	G_0

(b) Stabilizer and gauge group.

Figure 6.3: Graphical and matrix representation of doubled color codes.

6.4.2 Doubled Color Code Family

Doubled color codes were constructed by Bravyi and Cross [173]. They are constructed from 3 codes: two color codes (4.8.8 color codes in our case) and a subsystem code. The doubled color code can be viewed as a set of stacked color codes (A and B) where the bottom color code shares a stabilizer with the subsystem code (C) as shown in figure 6.3a. The stabilizer groups and gauge groups for the family of doubled color codes can be defined recursively. The base code is the ⟦15, 1, 3⟧ code which is constructed from two distance-3 color codes stacked upon each other and the subsystem code is a bare qubit. The stabilizer group is constructed from the two stabilizer groups of color codes ($S_i \otimes S_i$ for $i \in \mathcal{S}$ of the color code), a parity operator between color code A (the "bottom" code) and code C ($\bigotimes_q Z_q$ for $q \in A, C$), and the stabilizer group of the subsystem code S_0 shown in figure 6.3b. Note that, up to a permutation of qubit labels, this base code is equivalent to the Reed-Muller ⟦15, 1, 3⟧ quantum error-correcting code which is a triply even code that admits a transversal T gate. The gauge group is constructed from the stabilizer group on code A, logical operators (L and L_0), the gauge group of the subsystem code G_0, and a set of operators labeled *Even*. This *even* set of operators are weight-4 Pauli operators connecting the edges of the "stacked" qubits in the doubled color code. The stacked qubits are labeled in black in figure 6.3a. This gauge group allows for fault-tolerant conversion between the doubled color code (transversal T) and a single color code (transversal Clifford) allowing for a universal gate set. A similar result was obtained by Poulin et al. where they fault-tolerantly converted between Steane and Reed-Muller codes [174] and O'Connor et al. proposed a similar stacked code construction independently [175]. The distance-5 color code is constructed from two stacked for distance-5 4.8.8 color codes and using the distance-3 doubled color code as the subsystem code.

6.4.3 Error Threshold for Doubled Color Codes

The error threshold for the doubled color code under phenomenological depolarizing noise was found to be 0.98%. As we have seen above, this value is a bit lower than standard color codes and the surface code. This is due to the number of gauge measurements required for this code relative to the standard color code. Bravyi et al. have since obtained an error rate in gauge fixing to fault-tolerantly apply Clifford and T gate circuits and have found a rate of 0.55% for the phenomenological noise model. While this value is an about an order of magnitude below that of the surface code, the resource cost of universality for the surface code is larger since it requires distillation.

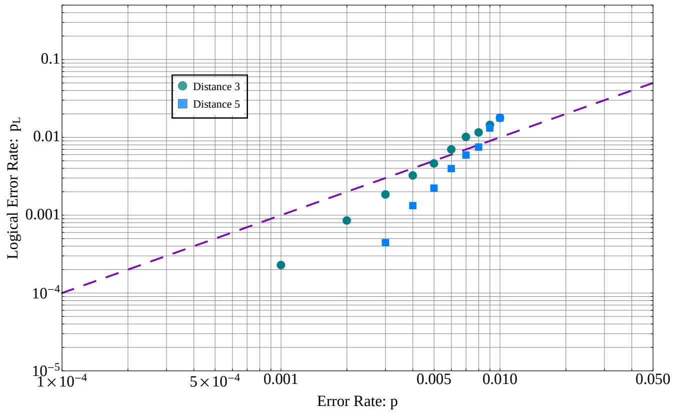


Figure 6.4: Phenomenological depolarizing threshold for the doubled color code.

Chapter 7

Future Directions

There are a plethora of extensions to the work provided above. Below is a non-exhaustive list.

7.1 Simulating the performance of a distance-3 surface code in a linear ion trap

A natural extension of this work is to build up a more experimentally accurate error model. This contains two components: constructing a physically representative error model as possible and modeling additional experimental sources of error. Many errors introduced during computation will have some coherent nature, which requires a full quantum simulation of the error correction routine. Therefore, removing the incoherent approximation to these error sources would provide a more accurate assessment of error correction. Simulation of the 17-qubit system would be computationally demanding, but quantum trajectory calculations of the 17-qubit surface code have been performed which, due to a clever circuit representation, allows for modeling with only 10 active qubits [176]. Additional sources of errors must be accurately modeled. Two particularly relevant error sources come to mind: ion heating due to shuttling and ion addressing errors. In our current error model, errors come strictly from gates and the ion chain has been optimized according to gate times. Indeed once errors from shuttling come into play, the ion chain optimization may change due to the interplay between the shuttling and gate errors. This gives an opportunity to reassess the ion chain optimization parameters as well. Indeed if the goal is to create a high fidelity logical qubit, then perhaps the ion chain optimization should involve a metric that reflects this goal.

Furthermore, subset information from the importance sampling method may be of use to improve error correction. For instance, this information about the failure rates *at the logical level* may be used to bias transition matrix elements of a maximum likelihood decoder to include information about the influence of error cosets on the code's performance instead of only considering the statistical weights of the error cosets [177]. Code considerations when optimizing the ion chain layout could serve to bound the effects of the most malignant error sources. Specifically, optimizing the ion chain to assign the most error-prone qubits to weight-2 stabilizers can reduce the influence of anisotropic incoherent error sources.

7.2 Magic state distillation and gate compilation in quantum algorithms

Since the assessment of distillation and gate compilation techniques; new methods have been developed. Two new techniques that should be appended to this analysis were developed by Campbell et al. [178, 179] and Haah et al. [180]. Campbell et al. constructed a composite gate synthesis-distillation routine (termed "synthillation") that effectively removes the T cost of gate synthesis relative to traditional distillation then synthesize techniques [178, 179]. Haah et al. constructed a new set of distillation protocols that approaches the (conjectured) asymptotically optimal resource counts for distillation using operations between clever choices of CSS codes with particular transversal gates [180]. This two-code distillation method removes the need to concatenate distillation protocols to achieve target error rates in T synthesis which substantially reduces resource costs. Also, similar assessments of the ancillary resource efficient flag-qubit method of fault-tolerantly measuring stabilizers [52, 53] should be performed to understand the overhead gain from such a technique.

An alternative to magic state distillation for obtaining a universal fault-tolerant gate set is gauge fixing. An understanding of the resource cost of gauge fixing relative to magic state distillation is an open question. Fair metrics for comparison of the techniques are required. Indeed, there is an interplay between the code implemented, the method used to obtain universality, and the error rates of the underlying operations. Distillation failure influences states ancillary to the encoded information while gauge fixing relies on successful error correction for the encoded gate to succeed. The efficacy of these two methodologies in different error rate regimes is particularly interesting.

7.3 Quantum error-correction failure distributions: Comparisons of coherent and stochastic error models

An open question remains of how the wavefunction failure criteria compares to other failure metrics such as the fidelity, trace distance, or diamond distance. Also, we have shown than the schedule of the syndrome extraction routine influences on the fringe statistics of the wavefunction failure criteria for the pulse area error model. Certain data even suggests that a X-type/Z-type interleaved measurement schedule can emulate Pauli-like error output statistics. An investigation into the effects of the syndrome schedule on the coherence of the error channel at the logical level is a particularly interesting direction due to the malignant effect of coherent errors on quantum error correction. The use of measurement as an avenue to bound the spread of the coherent error may also provide an avenue to apply proven threshold theorems to coherent error channels [44]. Bounding the coherence of the error may also provide tighter bounds for relating the diamond distance and the error rate for these cases as well [37, 162].

Non-Markovian noise models for quantum error correction are few an far between. By tweaking the sampling of the over/under-rotation parameter r, the pulse area error model could be used to model low-frequency time-dependent drifts in field amplitudes of applied gates; $r = A\sin(\omega t)$ for instance. Fitting these time-dependent errors within a circuit model formalism (opposed to a Hamiltonian simulation perspective) gives a straightforward path for assessing the effect of these control errors on quantum algorithms and understanding how compensation sequences can improve the computation at the algorithmic level.

The error detection routine implemented by Meier et al. [140] implements a subspace error detection code routine with a subspace that is a subsystem of the level-2 encoded subspace. With a proper choice of gauge and the use of additonal ancillary qubits to prepare both encoded qubits fault-tolerantly [181], gauge fixing techniques similar to those implemented in [174] could be used to fix between self-dual, doubly even, and triply even subsystems of the code space in equation 5.22. Therefore, level-2 error detection circuit could serve as a general purpose distillation circuit for H-type, S-type, and T-type magic states depending on the choice of gauge.

Finally, it's becoming more apparent that a single-valued metric for assessment of quantum processes may not be enough to fully characterize the underlying physics. The use of output statistics of the process may be useful due to the its natural overlap with how data is gathered experimentally. The comparison of output statistics of theoretically modeled and experimentally measured quantities can serve as a method of validating the accuracy of theoretical error model of the experimental apparatus. Finding connections between the output statistics of both methods could provide means of systematically constructing error models from experimental measurements as well.

7.4 Maximum likelihood decoding of topological codes

An obvious next step is to obtain an circuit-level depolarizing error threshold for the 4.6.12 topological color code. Tensor network representation of circuits may be used to compute the transition matrix elements efficiently for Maximum Likelihood Decoding. Unlike the surface code, color codes do not allow for fault-tolerant syndrome measurement with the use of single-qubit ancilla for small distance codes. Implementing a flag qubit method of syndrome extraction [52, 53] reduces the number ancillary resources required for fault-tolerant measurement of the stabilizers for the color code. Similar to a study performed by Tomita et al. [182], comparisons of the flag qubit technique to preexisting techniques to fault-tolerantly measure the stabilizers has been performed in [183] with the color code. Their results indicate a strong dependence of the error correction threshold on the error model of the idle qubits. Performing a full quantum simulation of a small distance color code under the influence idle-qubit error models indicative of ion trap and superconducting systems would shed light on the influence of the underlying physical system on the fault-tolerance of the flag-qubit technique.

Bibliography

[1] P. W. Shor (IEEE Computer Society Press, 1996), FOCS '96, pp. 56–65.

[2] Y. Tomita and K. M. Svore, Phys. Rev. A **90**, 062320 (2014).

[3] P. Benioff, Journal of Statistical Physics **22**, 563 (1980).

[4] P. Benioff, Physical Review Letters **48**, 1581 (1982).

[5] R. P. Feynman, Int. J. of Theor. Phys. **21**, 467 (1982).

[6] D. Deutsch (The Royal Society, 1985), vol. 400, pp. 97–117.

[7] P. W. Shor, in *Proceedings of the 35nd Annual Symposium on Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society Press, 1994), pp. 124–134.

[8] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).

[9] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[10] A. M. Childs, R. Kothari, and R. D. Somma, arXiv:quant-ph/1511.02306 (2015).

[11] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Commun. Math. Phys. (2017).

[12] S. Aaronson and A. Arkhipov, in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 2011), STOC '11, pp. 333–342.

[13] S. P. Jordan, K. S. M. Lee, and J. Preskill, Science **336**, 1130 (2012).

[14] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM Press, New York, NY, USA, 1996), STOC '96, pp. 212–219.

[15] D. S. Abrams and C. P. Williams, arXiv:quant-ph/9908083 (1999).

[16] N. Linden, H. Barjat, and R. Freeman, Chem. Phys. Lett. **296**, 61 (1998).

[17] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature **414**, 883 (2001).

[18] S. Gulde, M. Riebe, G. P. T. Lancaster, C. Becher, J. Eschner, H. Haffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt, Nature **421**, 48 (2003).

[19] E. Brainis, L. P. Lamoureux, N. J. Cerf, P. Emplit, M. Haelterman, and S. Massar, Phys. Rev. Lett. **90**, 157902 (2003).

[20] J. Chiaverini, J. Britton, D. Leibfried, E. Knill, M. D. Barrett, R. B. Blakestad, W. M. Itano, J. D. Jost, C. Langer, R. Ozeri, et al., Science **308**, 997 (2005).

[21] K. A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, and C. Monroe, Phys. Rev. A **72**, 050306 (2005).

[22] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, et al., Nature **460**, 240 (2009).

[23] F. Shi, X. Rong, N. Xu, Y. Wang, J. Wu, B. Chong, X. Peng, J. Kniepert, R. S. Schoenfeld, W. Harneit, et al., Phys. Rev. Lett. **105**, 040504 (2010).

[24] E. Martin-Lopez, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'Brien, Nat. Photonics **6**, 773 (2012).

[25] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016).

[26] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Nature **536**, 63 (2016).

[27] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[28] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[29] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).

[30] D. P. DiVincenzo and P. W. Shor, Phys. Rev. Lett. **77**, 3260 (1996).

[31] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Society, Boston, MA, USA, 2002).

[32] W. van Dam and M. Howard, Phys. Rev. Lett. **103**, 170504 (2009).

[33] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, New York, NY, USA, 2000).

[34] D. Gottesman (Cambridge, MA: International Press, 1999), Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics, pp. 32–43, longer version available at: arXiv:quant-ph/9807006.

[35] E. Knill, R. Laflamme, and W. H. Zurek, arXiv:quant-ph/9610011 (1996).

[36] E. Knill, R. Laflamme, and W. H. Zurek, Proc. Royal Soc. of London A **454**, 365 (1998).

[37] Y. R. Sanders, J. J. Wallman, and B. C. Sanders, New J. Phys. **18**, 012002 (2016).

[38] R. Kueng, D. M. Long, A. C. Doherty, and S. T. Flammia, Phys. Rev. Lett. **117**, 170502 (2016).

[39] C. E. Shannon, The Bell System Technical Journal **27**, 379423 (1948).

[40] R. W. Hamming, Bell System Technical Journal **26**, 147 (1950).

[41] D. Gottesman, Ph.D. thesis, California Institute of Technology (1997).

[42] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[43] A. Steane, Proc. Royal Soc. of London A **452**, 2551 (1996).

[44] D. Aharonov and M. Ben-Or, in *Proceedings of the 29th Annual Symposium on Theory of Computing* (1997), pp. 176–188.

[45] A. Kitaev, Russ. Math. Surv. **52**, 11911249 (1997).

[46] E. Knill, R. Laflamme, and W. Zurek, Proc. R. Soc. Lond. A **454**, 365 (1998).

[47] P. Aliferis, D. Gottesman, and J. Preskill, Quant. Inf. Comput. **6**, 97 (2006).

[48] P. Aliferis, Ph.D. thesis, California Institute of Technology (2007).

[49] B. W. Reichardt, in *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part I* (Springer-Verlag, Berlin, Heidelberg, 2006), ICALP'06, pp. 50–61.

[50] A. M. Steane, Phys. Rev. Lett. **78**, 2252 (1997).

[51] E. Knill, Nature **434**, 39 (2005).

[52] R. Chao and B. W. Reichardt, arXiv:quant-ph/1705.02329 (2017).

[53] R. Chao and B. W. Reichardt, arXiv:quant-ph/1705.05365 (2017).

[54] S. B. Bravyi and A. Y. Kitaev, arXiv:quant-ph/9811052 (2011).

[55] R. Raussendorf, J. Harrington, and K. Goyal, New J. Phys. **9**, 199 (2007).

[56] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).

[57] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).

[58] A. M. Stephens, Phys. Rev. A **89**, 022321 (2014).

[59] H. Bombin and M. A. Martin-Delgado, Phys. Rev. A **76**, 012305 (2007).

[60] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, New J. Phys. **14**, 123011 (2012).

[61] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, Phys. Rev. Lett. **77**, 198 (1996).

[62] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).

[63] A. M. Steane, IEEE Trans. Inf. Theory **45**, 2492 (1999).

[64] M. Li, M. Gutiérrez, S. E. David, A. Hernandez, and K. R. Brown, Phys. Rev. A **96**, 032341 (2017).

[65] D. Bacon, Phys. Rev. A **73**, 012340 (2006).

[66] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[67] T. J. Yoder and I. H. Kim, Quantum **1**, 2 (2017).

[68] D. Hayes, D. N. Matsukevich, P. Maunz, D. Hucul, Q. Quraishi, S. Olmschenk, W. Campbell, J. Mizrahi, C. Senko, and C. Monroe, Phys. Rev. Lett. **104**, 140501 (2010).

[69] C. Ospelkaus, U. Warring, Y. Colombe, K. R. Brown, J. M. Amini, D. Leibfried, and D. J. Wineland, Nature **476**, 181 (2011).

[70] U. Warring, C. Ospelkaus, Y. Colombe, R. Jördens, D. Leibfried, and D. J. Wineland, Phys. Rev. Lett. **110**, 173002 (2013).

[71] J. I. Cirac and P. Zoller, Phys. Rev. Lett. **74**, 4091 (1995).

[72] K. Mølmer and A. Sørensen, Phys. Rev. Lett. **82**, 1835 (1999).

[73] E. Solano, R. L. de Matos Filho, and N. Zagury, Phys. Rev. A **59**, R2539 (1999).

[74] G. Milburn, S. Schneider, and D. James, Fortschritte der Physik **48**, 801 (2000).

[75] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, et al., Nature **508**, 500 (2014).

[76] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016).

[77] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Proc. Natl. Acad. Sci. **114**, 3305 (2017).

[78] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, et al., Nature **432**, 602 (2004).

[79] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt, Science **332**, 1059 (2011).

[80] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt, Science **345**, 302 (2014).

[81] N. M. Linke, M. Gutiérrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, arXiv:quant-ph/1611.06946 (2016).

[82] E. Mount, C. Kabytayev, S. Crain, R. Harper, S.-Y. Baek, G. Vrijsen, S. T. Flammia, K. R. Brown, P. Maunz, and J. Kim, Phys. Rev. A **92**, 060301 (2015).

[83] R. Noek, G. Vrijsen, D. Gaultney, E. Mount, T. Kim, P. Maunz, and J. Kim, Opt. Lett. **38**, 4735 (2013).

[84] R. Ozeri, W. M. Itano, R. B. Blakestad, J. Britton, J. Chiaverini, J. D. Jost, C. Langer, D. Leibfried, R. Reichle, S. Seidelin, et al., Phys. Rev. A **75**, 042329 (2007).

[85] K. R. Brown, A. C. Wilson, Y. Colombe, C. Ospelkaus, A. M. Meier, E. Knill, D. Leibfried, and D. J. Wineland, Phys. Rev. A **84**, 030303 (2011).

[86] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, Phys. Rev. Lett. **117**, 060504 (2016).

[87] C. Roos, T. Zeiger, H. Rohde, H. C. Nägerl, J. Eschner, D. Leibfried, F. Schmidt-Kaler, and R. Blatt, Phys. Rev. Lett. **83**, 4713 (1999).

[88] H. P. Specht, C. Nolleke, A. Reiserer, M. Uphoff, E. Figueroa, S. Ritter, and G. Rempe, Nature **473**, 190 (2011).

[89] T. Ruster, C. T. Schmiegelow, H. Kaufmann, C. Warschburger, F. Schmidt-Kaler, and U. G. Poschinger, Appl. Phys. B **122**, 254 (2016).

[90] R. Noek, G. Vrijsen, D. Gaultney, E. Mount, T. Kim, P. Maunz, and J. Kim, Opt. Lett. **38**, 4735 (2013).

[91] A. Sørensen and K. Mølmer, Phys. Rev. Lett. **82**, 1971 (1999).

[92] A. Sørensen and K. Mølmer, Phys. Rev. A **62**, 022311 (2000).

[93] D. Maslov, New J. Phys. **19**, 023035 (2017).

[94] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof, J. Res. of the Natl. Inst. of Stand. Tech. **103**, 259 (1998).

[95] R. D. Cowan, *The Theory of Atomic Structure and Spectra* (University of California Press, 1981).

[96] A. M. Mårtensson-Pendrill, D. S. Gough, and P. Hannaford, Phys. Rev. A **49**, 3351 (1994).

[97] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. M. Amini, J. Britton, D. Leibfried, and D. J. Wineland, Phys. Rev. Lett. **102**, 153002 (2009).

[98] D. L. Moehring, C. Highstrete, D. Stick, K. M. Fortier, R. Haltli, C. Tigges, and M. G. Blain, New J. Phys. **13**, 075018 (2011).

[99] R. Bowler, J. Gaebler, Y. Lin, T. R. Tan, D. Hanneke, J. D. Jost, J. P. Home, D. Leibfried, and D. J. Wineland, Phys. Rev. Lett. **109**, 080502 (2012).

[100] A. Walther, F. Ziesel, T. Ruster, S. T. Dawkins, K. Ott, M. Hettrich, K. Singer, F. Schmidt-Kaler, and U. Poschinger, Phys. Rev. Lett. **109**, 080501 (2012).

[101] K. Wright, J. M. Amini, D. L. Faircloth, C. Volin, S. C. Doret, H. Hayden, C.-S. Pai, D. W. Landgren, D. Denison, T. Killian, et al., New J. Phys. **15**, 033004 (2013).

[102] G. Shu, G. Vittorini, A. Buikema, C. S. Nichols, C. Volin, D. Stick, and K. R. Brown, Phys. Rev. A **89**, 062308 (2014).

[103] R. C. Sterling, H. Rattanasonti, S. Weidt, K. Lake, P. Srinivasan, S. C. Webster, M. Kraft, and W. K. Hensinger, Nat. Commun. **5**, 3637 (2014).

[104] T. Choi, S. Debnath, T. A. Manning, C. Figgatt, Z.-X. Gong, L.-M. Duan, and C. Monroe, Phys. Rev. Lett. **112**, 190502 (2014).

[105] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland, Rev. Mod. Phys. **75**, 281 (2003).

[106] J. P. Schiffer, Phys. Rev. Lett. **70**, 818 (1993).

[107] D. H. E. Dubin, Phys. Rev. Lett. **71**, 2753 (1993).

[108] G.-D. Lin, S.-L. Zhu, R. Islam, K. Kim, M.-S. Chang, S. Korenblit, C. Monroe, and L.-M. Duan, EPL **86**, 60004 (2009).

[109] W. C. Campbell, J. Mizrahi, Q. Quraishi, C. Senko, D. Hayes, D. Hucul, D. N. Matsukevich, P. Maunz, and C. Monroe, Phys. Rev. Lett. **105**, 090502 (2010).

[110] P. J. Lee, K.-A. Brickman, L. Deslauriers, P. C. Haljan, L.-M. Duan, and C. Monroe, J. Opt. B **7**, S371 (2005).

[111] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, et al., arXiv:quant-ph/1705.02771 (2017).

[112] K. R. Brown, J. Kim, and C. Monroe, npj Quantum Inf. **2**, 16034 (2016).

[113] M. Gutiérrez, C. Smith, L. Lulushi, S. Janardan, and K. R. Brown, Phys. Rev. A **94**, 042338 (2016).

[114] D. Puzzuoli, C. Granade, H. Haas, B. Criger, E. Magesan, and D. G. Cory, Phys. Rev. A **89**, 022306 (2014).

[115] M. Gutiérrez, L. Svec, A. Vargo, and K. R. Brown, Phys. Rev. A **87**, 030302 (2013).

[116] Q. A. Turchette, Kielpinski, B. E. King, D. Leibfried, D. M. Meekhof, C. J. Myatt, M. A. Rowe, C. A. Sackett, C. S. Wood, W. M. Itano, et al., Phys. Rev. A **61**, 063418 (2000).

[117] R. Ozeri, W. M. Itano, R. B. Blakestad, J. Britton, J. Chiaverini, J. D. Jost, C. Langer, D. Leibfried, R. Reichle, S. Seidelin, et al., Phys. Rev. A **75**, 042329 (2007).

[118] D. Hayes, S. M. Clark, S. Debnath, D. Hucul, I. V. Inlek, K. W. Lee, Q. Quraishi, and C. Monroe, Phys. Rev. Lett. **109**, 020503 (2012).

[119] J. Gea-Banacloche, arXiv:quant-ph/0504187 (2005).

[120] A. D. Ludlow, M. M. Boyd, J. Ye, E. Peik, and P. O. Schmidt, Rev. Mod. Phys. **87**, 637 (2015).

[121] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, Phys. Rev. Lett. **113**, 220501 (2014).

[122] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Phys. Rev. Lett. **108**, 180501 (2012).

[123] J. Edmonds, Canad. J. Math. **17**, 449 (1965).

[124] J. Edmonds, J. Res. Nat. Bur. Stand. **69B**, 67 (1965).

[125] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).

[126] B. Heim, K. M. Svore, and M. B. Hastings, arXiv:quant-ph/1609.06373 (2016).

[127] D. A. Hite, Y. Colombe, A. C. Wilson, K. R. Brown, U. Warring, R. Jördens, J. D. Jost, K. S. McKay, D. P. Pappas, D. Leibfried, et al., Phys. Rev. Lett. **109**, 103001 (2012).

[128] M. Niedermayr, K. Lakhmanskiy, M. Kumph, S. Partel, J. Edlinger, M. Brownnutt, and R. Blatt, New J. Phys. **16**, 113068 (2014).

[129] L. Deslauriers, P. C. Haljan, P. J. Lee, K.-A. Brickman, B. B. Blinov, M. J. Madsen, and C. Monroe, Phys. Rev. A **70**, 043408 (2004).

[130] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L.-M. Duan, D. Yum, and K. Kim, arXiv:quant-ph/1701.04195 (2017).

[131] C. Langer, R. Ozeri, J. D. Jost, J. Chiaverini, B. DeMarco, A. Ben-Kish, R. B. Blakestad, J. Britton, D. B. Hume, W. M. Itano, et al., Phys. Rev. Lett. **95**, 060502 (2005).

[132] F. Schmidt-Kaler, S. Gulde, M. Riebe, T. Deuschle, A. Kreuter, G. Lancaster, C. Becher, J. Eschner, H. Häffner, and R. Blatt, J. Phys. B **36**, 623 (2003).

[133] J. P. Barnes, C. J. Trout, D. Lucarelli, and B. D. Clader, Phys. Rev. A **95**, 062338 (2017).

[134] B. Eastin and E. Knill, Phys. Rev. Lett. **102**, 110502 (2009).

[135] A. G. Fowler, S. J. Devitt, and C. Jones, Sci. Rep. **3**, 1939 (2013).

[136] D. Gottesman and I. L. Chuang, Nature **402**, 390 (1999).

[137] A. Paetznick and K. M. Svore, arXiv:quant-ph/1311.1074 (2013).

[138] A. G. Fowler, Phys. Rev. A **83**, 042310 (2011).

[139] S. Bravyi and A. Kitaev, Phys Rev A **71**, 022316 (2005).

[140] A. M. Meier, B. Eastin, and E. Knill, Quantum Inf Comp **13**, 195 (2013).

[141] S. Bravyi and J. Haah, Phys. Rev. A **86**, 052329 (2012).

[142] C. M. Dawson and M. A. Nielsen, Quant Inf Comp **6**, 81 (2006).

[143] T. T. Pham, R. Van Meter, and C. Horsman, Phys. Rev. A **87**, 052332 (2013).

[144] A. Bocharov and K. M. Svore, Phys. Rev. Lett. **109**, 190501 (2012).

[145] A. G. Fowler, Quantum Inf Comput **11**, 867 (2011).

[146] V. Kliuchnikov, D. Maslov, and M. Mosca, Phys. Rev. Lett. **110**, 190502 (2013).

[147] P. Selinger, Quantum Inf Comput **15**, 0159 (2015).

[148] V. Kliuchnikov, D. Maslov, and M. Mosca, IEEE Trans. Comput. **65** (2015).

[149] N. J. Ross and P. Selinger, arXiv:1403.2975 [quant-ph], (2014).

[150] N. C. Jones, J. D. Whitfield, P. L. McMahon, M. H. Yung, R. V. Meter, A. Aspuru-Guzik, and Y. Yamamoto, New J. Phys. **14**, 115023 (2012).

[151] N. Wiebe and V. Kliuchnikov, New J. Phys. **15**, 093041 (2013).

[152] A. Bocharov, M. Roetteler, and K. Svore, arXiv:quant-ph/1404.5320 (2014).

[153] A. Bocharov, M. Roetteler, and K. Svore, arXiv:quant-ph/1409.3552 (2014).

[154] V. Kliuchnikov, D. Maslov, and M. Mosca, Quantum Inf. Comput. **13**, 607 (2013).

[155] B. Giles and P. Selinger, Phys. Rev. A **87**, 032332 (2013).

[156] P. Shor, in *Proc 35th Symposium on Foundations of Computer Science* (1994), pp. 124–134.

[157] G. Duclos-Cianci and K. M. Svore, Phys. Rev. A **88**, 042325 (2013).

[158] A. J. Landahl and C. Cesare, arXiv:quant-ph/1302.3240 (2013).

[159] G. Duclos-Cianci and D. Poulin, arXiv:quant-ph/1403.5280 (2014).

[160] D. Kudrow, K. Bier, Z. Deng, D. Franklin, Y. Tomita, K. R. Brown, and F. T. Chong, in *ISCA'13* (2013), pp. 166–176.

[161] A. S. Darmawan and D. Poulin, Phys. Rev. Lett. **119**, 040502 (2017).

[162] J. J. Wallman, arXiv:quant-ph/1511.00727 (2015).

[163] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).

[164] H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, Phys. Rev. Lett. **103**, 090501 (2009).

[165] M. Ohzeki, Phys. Rev. E **80**, 011141 (2009).

[166] D. S. Wang, A. G. Fowler, C. D. Hill, and L. C. L. Hollenberg, Quant. Inf. Comp. **10**, 780 (2010).

[167] A. J. Landahl, J. T. Anderson, and P. R. Rice, arXiv:quant-ph/1108.5738 (2011).

[168] R. S. Andrist, H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, New J. Phys. **13**, 083006 (2011).

[169] T. Ohno, G. Arakawa, I. Ichinose, and T. Matsui, Nucl. Phys. B **697**, 462 (2004).

[170] D. Poulin, Phys. Rev. Lett. **95**, 230504 (2005).

[171] D. Bacon, Phys. Rev. A **73**, 012340 (2006).

[172] A. Klappenecker and P. Kiran Sarvepalli, arXiv:quant-ph/0604161 (2006).

[173] S. Bravyi and A. Cross, arXiv:quant-ph/1509.03239 (2015).

[174] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, Phys. Rev. Lett. **113**, 080501 (2014).

[175] T. Jochym-O'Connor and S. D. Bartlett, Phys. Rev. A **93**, 022323 (2016).

[176] T. E. O'Brien, B. Tarasinski, and L. DiCarlo, npj Quantum Inf. **3**, 39 (2017).

[177] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, arXiv:quant-ph/1708.08474 (2017).

[178] E. T. Campbell and M. Howard, Phys. Rev. A **95**, 022316 (2017).

[179] E. T. Campbell and M. Howard, Phys. Rev. Lett. **118**, 060501 (2017).

[180] J. Haah, M. B. Hastings, D. Poulin, and D. Wecker, Quantum **1**, 31 (2017).

[181] D. Gottesman, arXiv:quant-ph/1610.03507 (2016).

[182] Y. Tomita, M. Gutiérrez, C. Kabytayev, K. R. Brown, M. R. Hutsel, A. P. Morris, K. E. Stevens, and G. Mohler, Phys. Rev. A **88**, 042336 (2013).

[183] C. Chamberland and M. E. Beverland, arXiv:quant-ph/1708.02246 (2017).