ACCURATE MODELING OF NOISE IN QUANTUM ERROR-CORRECTING CIRCUITS

A Thesis
Presented to
The Academic Faculty

by

Mauricio Gutiérrez Arguedas

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

Georgia Institute of Technology
December 2015

ACCURATE MODELING OF NOISE IN QUANTUM ERROR-CORRECTING CIRCUITS

Approved by:

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

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

Rigoberto Hernández
School of Chemistry and Biochemistry
Georgia Institute of Technology

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

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

Date Approved: 10 November 2015

To my parents, my brother, and the rest of my family.

ACKNOWLEDGEMENTS

In 2009, after getting my bachelor's degree from the University of Costa Rica, I decided to go abroad to pursue a Ph.D. This decision had both academic and personal reasons: I wanted to learn about science, but also about life. In these 5 years, I have had the immense fortune to learn about both.

First and foremost, I would like to thank my advisor Ken Brown. I really don't have enough words to express my gratitude towards him. He opened the doors of his group to me, even though I knew almost nothing about research, quantum mechanics, and programming. He gave me (and all the other lab members) the opportunity to grow academically and the freedom to choose the direction of growth. He introduced me into the fascinating worlds of quantum information and quantum error correction. From him I have learned what science is about. And yet, the two most valuable lessons I'll take from Ken are not even scientific; they're two philosophical maxims. "The important thing is to fail as soon as possible" (supposedly attributed to John Archibald Wheeler) and "All paths lead nowhere, so you have to choose a path with a heart" (adapted from Carlos Castaneda) will guide me for the rest of my life.

My committee members have been patient and always eager to help. They have made insightful suggestions during these years. I thank all of them: Dr. Sherrill, Dr. Hernández, Dr. Perry, and Dr. Kennedy.

I also want to thank all the former and current lab members. It was a pleasure to meet you all: Ricardo Viteri, Craig Clark, James Goeders, Grahame Vittorini, True Merrill, Yu Tomita, Ncamiso Khanyile, Chingiz Kabytayev, Rick Shu, Colin Trout, Smitha Janardan, René Rugango, Aaron Calvin, Harrison Ansley, Brian McMahon, Mudit Sinhal, Aaron Buikema, Mehfouz Jalal, Josh Job, Pradeep Sarvepalli, Spencer Nichols, Conner Herndon, Aaron Moon, Muyuan Li, Catherine Wen, Rob Clark, Jenny Moody, Mason Bogue, JP Addison, Karl Burkhardt, John Gray, Alex Popescu, Caitlin Morris, Naveena Karusala, Jeffrey Chiu, Yon Choi, Chien-Yuan Chang, Samuel Musser, and Afsaneh Sadrolashrafi. Throughtout these years, I have had the great pleasure to mentor several undergrad and high-school students. I have learned from each one of them: Daniel Guilak, Michael Liang, Jared Huggins, Reuben Bryant, Yubo Su, Conor Smith, Livia Lulushi, Alonzo Hernandez, and Silas Fradley. I have also been lucky enough to collaborate with amazing people: Kelly Stevens, Greg Mohler, Andre Van Rynbach, Muhammad Ahsan, Lukas Svec, and Alex Vargo.

I would like to thank all the friends I have made along the way: Sergio and Jessie Paniagua, Arnoldo Castro, Alfredo Valverde, Darío Mena, Mauricio Bedoya, Anjaya Shrestha and his triple A team, Matt Golob, Ben Buchanan, Pierre Valdez, Vikram Atre, Kait Stanhope, Andrea Catalano, Daniel Mahoney, Khanh Do, Stella Huang, Aida Demissie, Aaron McKee, Loice Chingoza, Josh Méndez, Isaac Gallego, Christos Samaras, Anosh Daruwalla, Luis Hasbun, Muzi Ginindza, Allan Martell, Taisha Alonzo, and Deana Brown.

I was able to come to the U.S. thanks to supportive people back home: John Ogilvie, don Julio Mata, Mavis Montero, Alice Perez, Guy Lamoureux, Cristian Campos, and Carlos Sandoval. John Ogilvie was the one who gave me the motivation to study abroad. Thanks, John! Some friends in Costa Rica have been very important for me these years, especially during the hard periods: Cascos, Perre, Hugo, Kath, Jessica, Diego, Bryan, Tati, Chus. Pura vida! My family has always supported me and given me good vibes when I needed them: my parents, Beto, Silvia, Alvaro, Emma, Laura, the barvenos, Marco, Louise, and everybody else.

Finally, I would like to thank Mbali, who, like the dawn, has brought light to my day.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Microprocessor-based computers have revolutionized the world. Problems that seemed hopelessly daunting a century ago are now addressable with the aid of computer simulation, from modeling safe antiseismic structures to sequencing an organism's genome and designing new materials and drugs. Yet, despite the continuous increase in computational power and the proliferation of computer clusters, supercomputers, and other forms of massively parallel computing, there are important problems that remain hard. In most cases, the algorithms to solve these problems exist, but the memory and time to compute them scale exponentially with the problem size; the intractability wall is hit rather quickly.

Arguably, in the fields of physics and chemistry, the most significant of these intractable problems is the simulation of quantum mechanics. The mathematical framework needed to accurately model quantum phenomena like superposition and entanglement requires computational resources that scale exponentially with the size of the quantum system [96]. In the 1980s, it was proposed that this exponential roadblock could be overcome by using a controllable quantum system to simulate the system of interest [38]. Moreover, the controllable quantum system could be also programmed to achieve a general-purpose computing device, a quantum computer [15, 16, 31]. The conjecture that a quantum system could efficiently simulate the dynamics of another quantum system was proved right in the next decade [67]. However, it was the discovery that a quantum computer could factor integer numbers in polynomial time [89] what truly sparked widespread interest in quantum information processing, given its potential to crack RSA cryptographic systems [86].

In the last two decades, other problems that do not have a known efficient solution on a classical computer have proved to be efficiently solvable on a quantum computer. These include calculating the energy of atoms and molecules in the full configuration interaction (FCI) [12], boson sampling [1], computing scattering probabilities in quantum field theory [58], and solving a linear system of equations [54]. Quantum computers would also allow for a considerable speedup, though not an exponential one, in other problems, like searching an unstructured database [50] and calculating integrals numerically with Monte Carlo methods [3]. A large-scale quantum computer would definitely be a major technological breakthrough, with amazing potential applications in several fields. Chemistry is one of these fields: apart from the efficient calculation of FCI molecular energies, quantum algorithms have been proposed to simulate chemical dynamics [60] and solve protein folding optimization problems [61]. There are experimental demonstrations of the calculation of energies for several small molecules [66, 103, 34] and other problems of interest for the physical chemical community, such as the simulation of frustrated spin systems [107, 57]. However, all of these demonstrations have been proof-of-principle experiments: so far, no quantum computer is large enough to solve a problem currently unfeasible in a classical computer.

The main obstacle to the construction of a scalable quantum computer is the presence of noise. Complete isolation from the environment is not only impossible, but also unwanted. The question of how to do quantum information processing reliably in the presence of noise is a very active area of research. So far, the most general solution to this problem has been the use of quantum error-correcting codes (QECCs) and fault-tolerant (FT) procedures to perform the quantum computation [88, 22, 33]. In this scheme, the whole computation is performed with quantum information encoded in a QECC and with FT operations that limit the degree to which the information is corrupted [77].

One of the most important results in the field of FT quantum computation is the quantum threshold theorem [5, 63, 65, 85, 8], which states that if noise acts independently on each qubit and its occurence is below a certain threshold value, then it is possible to perform an efficient quantum computation with an arbitrarily small failure probability. The threshold has been proved for several noise models [6, 76, 84, 39]. This threshold value will depend not only on the particular QECC, but also on the methods used to detect and correct the errors, and to prepare the encoded states. Therefore, there is plenty of work to be done before a scalable quantum computer is built. We need accurate estimates of the threshold values for different error correction protocols in order not only to compare them, but also to give experimentalists an estimate of the gate fidelities needed to achieve overall fault tolerance.

This information requires simulations of error-correcting quantum circuits, often involving hundreds of qubits. The exponential cost of exact full-density-matrix simulations has led researchers to focus almost exclusively on Clifford circuits, in which the allowed states throughout the circuit correspond to a small subset of all possible quantum states that can be efficiently stored in and manipulated by a classical computer. Fortunately, the error detection and correction steps in most QECCs are described by Clifford circuits. However, it is still not entirely clear how to model noise. In most cases, the noise affecting a quantum computer is not accurately modeled by a stabilizer operator, so a large-scale simulation of a quantum circuit will imply that we cannot model noise exactly.

Despite its importance, there has been very little focus on this topic. It is often claimed in the introductions of papers in the field that the exact nature of the noise, although perhaps important at the single-qubit level, becomes practically irrelevant in the limit of large circuit size, and so we can just employ a stabilizer error model in our simulations. However, there has not been any rigorious theoretical or numerical proof that this is in fact true, and so it is still not completely clear whether or not this represents a major problem. This has been the main question that has motivated my graduate work. We have focused on determining how the performance of a quantum error-correcting protocol depends on the exact nature of the error. We have also worked on improving the way we obtain stabilizer approximations to realistic error models, in order to bridge the gap between a description that is exact and one that is efficient.

This thesis is organized as follows. In Chapter 2, we present an introduction to quantum information. In Chapter 3, we describe classical and quantum error correction. In Chapter 4, we introduce the expanded Clifford error approximations to non-Clifford channels. In Chapter 5, we use these approximations to determine the performance of the expanded Clifford channels in estimating important parameters of a QECC when the noise affecting the system is incoherent. In Chapter 6 we extend this analysis to coherent errors. In Chapter 7, we describe the implementation of a Python-based toolset to simulate quantum circuits.

CHAPTER II

QUANTUM INFORMATION BACKGROUND

2.1 Qubits and quantum gates

A quantum bit or qubit can be implemented on very different physical systems, including spin states of nuclei [102, 34], internal states of trapped ions [25, 73], photons [23], semiconductors [53, 13], and superconducting circuits [69, 32]. In all of these systems, two well-defined states are chosen to serve as the computational basis for the qubit. Mathematically, regardless of the physical implementation, a qubit can be represented as the state of a spin-$1/2$ particle. The state of the qubit is a normalized vector in $\mathbb{C}^2$. The computational basis is given by:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad , \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1}$$

The state of the qubit can be any normalized linear superposition of the computational basis:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad , \quad |\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

In the circuit model of quantum computing, the evolution of the quantum system is represented by a sequence of unitary operations known as gates. For a single qubit, the most common gates are the Identity:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{3}$$

and the 3 Pauli matrices:

$$\sigma_X = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad \sigma_Y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad , \quad \sigma_Z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{4}$$

Other common gates are the phase (S), the T, and the Hadamard (H) gates:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad , \quad T = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{i} \end{pmatrix} \quad , \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} . \tag{5}$$

Any single qubit unitary can be efficiently approximated with arbitrary accuracy as a sequence of the T and the H gates [29]. To obtain the overall state of a collection of qubits, we compute the tensor product of the individual qubits. The two most common multiqubit gates are the controlled-Z and the controlled-X or CNOT, which apply a conditional Z or X gate on the target qubit depending on the state of the control qubit. If the control qubit is in state $|0\rangle$, the target qubit is not changed. On the other hand, if the control qubit is in state $|1\rangle$, a Z or X gate is applied to the target qubit. These gates are entangling, because if the initial state of the control qubit is a superposition of $|0\rangle$ and $|1\rangle$, then the final state of the two qubits will be entangled. The control-Z is given by:

$$CZ \; = \; |0\rangle\langle 0| \otimes I \; + \; |1\rangle\langle 1| \otimes Z \; = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} . \tag{6}$$

The control-X is given by:

$$CX \; = \; |0\rangle\langle 0| \otimes I \; + \; |1\rangle\langle 1| \otimes X \; = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{7}$$

In both of these cases, the first qubit is the control and the second qubit is the target.

If we ignore the global phase, the state of a single qubit can be conveniently expressed as:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle . \tag{8}$$

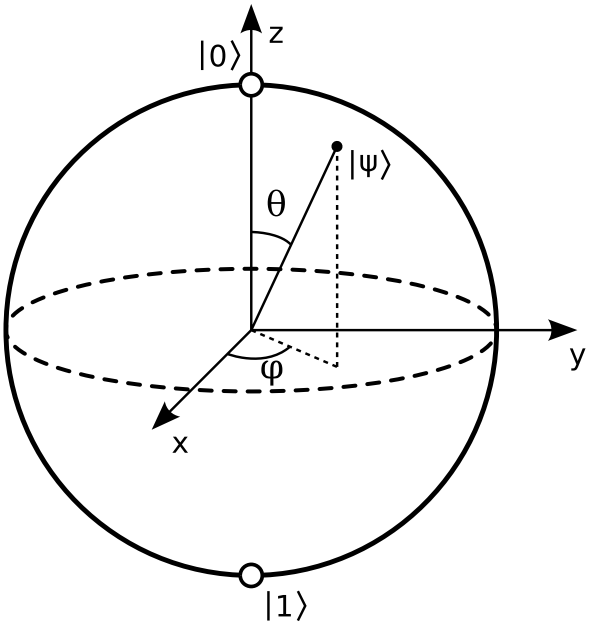

Figure 1: The Bloch sphere constitutes a convenient graphical representation of a qubit. The computational states are located on the poles of the sphere. The angle θ parametrizes the relative weight of each computational state in the superposition, while ϕ defines the relative phase. Pure states are located on the surface of the sphere, while mixed states are inside of it. The Pauli matrices define Cartesian coordinates. Image taken from https://commons.wikimedia.org.

Moreover, if the state of the qubit is not necessarily pure, it can be compactly represented as a sum of the Identity and the Pauli matrices:

$$\rho \;=\; \frac{1}{2}\left(I + \vec{r}\cdot\vec{\sigma}\right) \;=\; \frac{1}{2}\left(I + r_x X + r_y Y + r_z Z\right). \tag{9}$$

The vector of real numbers $\vec{r}$ parametrizes the degree of mixture of the qubit. For pure states, $|\vec{r}| = 1$. For mixed states, $|\vec{r}| < 1$. The maximally mixed state has $\vec{r} = 0$. These expressions allow for a concise graphical representation of a qubit: the Bloch sphere, presented in Figure 1. Up to an ignored global phase, every pure single qubit state is uniquely represented by a point on the surface of the Bloch sphere, while mixed states are points inside of it. A point on the surface of the Bloch sphere can be specified by the angles θ and ϕ. A general state can be specified by the vector $\vec{r}$.

2.2 Simulation of stabilizer circuits

Although the simulation of a general quantum circuit is inefficient in a classical computer, a subgroup of quantum circuits known as stabilizer circuits can actually be simulated in polynomial time. This is the Gottesman-Knill theorem and it is one of the most important results in quantum error correction [77].

The idea behind the efficient simulation of this subgroup of quantum circuits is based on the so-called *stabilizer formalism* [46]. In this theory, if an operator maps a given quantum state vector to itself, then the operator is said to stabilize that state. In other words, a state stabilized by an operator is an eigenvector of that operator with eigenvalue equal to 1. Consider the following examples. The state $|0\rangle$ is stabilized by Z, while $|1\rangle$ is stabilized by $-Z$, where Z denotes the Pauli Z matrix. The two-qubit state $|0\rangle \otimes |0\rangle = (1,0,0,0)^T$, usually abbreviated by $|00\rangle$, is stabilized by $\{Z \otimes I, I \otimes Z\}$, while $\frac{|00\rangle+|11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}(1,0,0,1)^T$ is stabilized by $\{Z \otimes Z, X \otimes X\}$, where X denotes the Pauli X matrix. Notice that the stabilizers of a particular state will always be commuting, precisely because they stabilize the same state. Any n-qubit state can be completely described by a set of n independent stabilizers. Conversely, any set of n independent commuting operators will simultaneously stabilize a unique n-qubit state, up to a global phase.

In the *stabilizer formalism*, a quantum state is represented by its stabilizers, rather than by its state vector. In the simulation of a quantum circuit, an initial quantum state is sequentially transformed by the unitary gates that make up the circuit. This can either be done by keeping track of the state vector at each point in the circuit or equivalently by keeping track of the stabilizers. When a vector state $|\psi\rangle$ is transformed by a unitary gate U to the state $U|\psi\rangle$, each stabilizer S_i is transformed to $US_iU^\dagger$.

Quantum states stabilized by elements of the Pauli group are known as stabilizer states. In the 1-qubit case, there are 6 stabilizer states: the $+1$-eigenvalued vectors of the operators $\pm X$, $\pm Y$, and $\pm Z$. For n qubits, it is the group generated by all the possible n-tensor products of the Pauli matrices. The memory required to describe a stabilizer state scales polynomially with the number of qubits [2], in contrast to the exponential scaling of a general quantum state. This implies that a quantum circuit in which the initial state is a stabilizer state and each gate transforms the state to another stabilizer state can be simulated in polynomial time. Therefore, we are restricted to quantum circuits composed exclusively of operations that map stabilizer states to stabilizer states or, from the *stabilizer formalism* point of view, operations that map a Pauli element to another Pauli element: $UP_iU^\dagger = P_j$, where P_i and P_j are two elements of the Pauli group. In general, the group of all these operations is known as the Clifford group. Additionally, the state preparations and the measurements can only be done in the Pauli basis.

2.3 Errors in quantum information and simulation of noisy quantum circuits

Errors during a quantum computation arise from unwanted interactions with the environment. In general, these interactions will produce a non-unitary evolution on the system's quantum state [77], so it is more useful to work with the density matrix, ρ, instead of the state vector, $|\psi\rangle$. Because the quantum computer is an open quantum system, it is convenient to consider the combined system that it forms with the environment. Without loss of generality, together they form an isolated system, described by the joint initial state $\rho_{CE}(0)$, where C refers to the computer subsystem and E refers to the environment. The composite computer-environment state after some time t is given by:

$$\rho_{CE}(t) = U_{CE}(t)\rho_{CE}(0)U^\dagger_{CE}(t), \tag{10}$$

where $U_{CE}(t)$ refers to the unitary time propagator for the joint system. In general, after such an evolution, the computer will become entangled with the environment and its final state is found by applying a partial trace operation over the environmental degrees of freedom. Assuming that initially the computer and the environment are uncorrelated, we can write $\rho_{CE}(0) = \rho_C(0) \otimes \rho_E(0)$. The initial state of the environment can be expressed as $\sum_k \lambda_k |\phi_k\rangle\langle\phi_k|$. Therefore, after tracing out the environment, the state of the computer is given by:

$$\rho_C(t) = \sum_{i,k} \lambda_k \langle i|U_{CE}(t)|\phi_k\rangle \rho_C(0) \langle \phi_k|U^\dagger_{CE}(t)|i\rangle, \tag{11}$$

where $\{|i\rangle\}$ is an orthonormal basis for the environment Hilbert space. More compactly, we can express the above equation as:

$$\rho_C(t) = \sum_i E_i(t)\rho_C(0)E_i^\dagger(t), \tag{12}$$

with $E_i(t) = \sum_k \sqrt{\lambda_k}\langle i|U_{CE}(t)|\phi_k\rangle$. The operators $\{E_i\}$ are known as the Kraus operators and equation 12 is known as the Kraus or operator-sum representation of a quantum channel. From now on, we will eliminate the subscript C, as we will always refer to the state of the quantum computer, and the time parameter, and represent the transformation by ε:

$$\varepsilon(\rho) = \sum_i E_i \rho E_i^\dagger \tag{13}$$

The operators $\{E_i\}$ will in general not be unitary. However, if $\sum_i E_i^\dagger E_i = I$, then the transformation ε is completely positive, which implies that the final states of both the system and the environment can still be described by positive semi-definite matrices of trace 1, density matrices. Throughout this document, we will only deal with completely positive transformations. Nevertheless, we should note that not all quantum maps need to be completely positive and indeed important physical processes are described by non-completely positive maps.

To further understand how an error channel described in the Kraus representation transforms a quantum state, consider the following example. A map is defined by:

$$\varepsilon = \begin{cases} E_0 = \sqrt{1-p}I \\ E_1 = \sqrt{p}X \end{cases} \tag{14}$$

with $0 \leq p \leq 1$. The effect of this operation on a state ρ is:

$$\varepsilon(\rho) = (1-p)\,\rho + p\,X\rho X \tag{15}$$

which can be interpreted as a channel in which there is a probability p that the state will undergo an X transformation, and a probability of $1 - p$ that the state will remain unchanged. This "probabilistic" interpretation of an error channel is fundamental for the simulation of noisy quantum circuits. For example, if the noise affecting the quantum computer is the error channel mentioned above, then to perform the simulation, we would insert after each gate in the quantum circuit an X gate with a probability of p and compare the output of this circuit with the output of the error-free circuit. This would be done multiple times to obtain representative averages.

The drawback of this approach is that not all error channels can be simulated efficiently. In the first place, some error channels do not allow this "probabilistic" interpretation. The quintessential example of this is the amplitude damping channel (ADC), which describes the dissipation of energy from a 2-state quantum system to its environment:

$$\text{ADC} = \begin{cases} E_0 = |0\rangle\langle 0| \;+\; \sqrt{1-\gamma}\,|1\rangle\langle 1| \\ \\ E_1 = \sqrt{\gamma}\,|0\rangle\langle 1| \end{cases} \tag{16}$$

In this expression γ is the damping strength and is defined by $1 - e^{-\Gamma t}$, where Γ is the damping rate ($\Gamma > 0$). This error channel is present in every quantum computer implementation. Despite its simple form, it is impossible to rerrange it in a way that allows a "probabilistic" interpretation.

In the second place, some error channels that do have a "probabilistic" interpretation involve operators that are not part of the Clifford group, so their simulation is not possible with the stabilizer approach explained above. An example of this class of errors is given by a probabilistic rotation about a non-Clifford axis:

$$\text{Pol}_{\phi}\text{C} = \begin{cases} E_0 = \sqrt{1-p}\,I \\ \\ E_1 = \sqrt{p}\,[\cos(\phi)X + \sin(\phi)Y] \end{cases} \tag{17}$$

For $\phi \neq \frac{n\pi}{4}$, with n an integer number, the second operator of this channel is outside the Clifford group.

2.4 From the operator-sum to the process matrix representation

Although elegant and concise, a channel expressed in the operator-sum form has the disadvantage of the freedom of representation. A particular quantum channel is not uniquely determined by a set of Kraus operators; instead, (infinitely) many sets of Kraus operators represent the same transformation. To see this, it is useful to go back to equation 11. Notice that we obtained the operators $E_i(t)$ by tracing out the environment in terms of an orthonormal basis $\{|i\rangle\}$. The selection of this basis is completely arbitrary and different choices will generate different Kraus operators.

When characterizing a quantum channel, people in the community prefer to use the χ or process matrix [70, 24, 87, 37]. This representation is very useful since, once a basis for the matrix has been chosen, the freedom of the Kraus operators disappears. It is also a very amenable representation to be used when determining a channel from experimental data [24, 49, 18].

The idea behind the process matrix is simple. If we start with the Kraus representation of a quantum channel ε:

$$\varepsilon(\rho) = \sum_i E_i \rho E_i^\dagger, \tag{18}$$

we first select a complete basis for the space of operators $\{E_i\}$. For n qubits, the space of operators has dimension $d = 4^n$. We require the basis set to the orthonormal, and define orthonormality with respect to the Hilbert-Schmidt or trace inner product:

$$(A, B) = tr(A^\dagger B) \tag{19}$$

for linear operators A and B. The most intuitive basis set that comes to mind is given by $\{|j\rangle\langle j'|\}$, where $\{|j\rangle\}$ is a complete basis set for the n-qubit vector (ket) space, for example, the computational basis set. However, it turns out that experimentally it is more useful to work with the Pauli basis. In terms of the Hilbert-Schmidt inner product, the operators in the 1-qubit Pauli basis are all mutually orthogonal. Since the cardinality of this set is the same as the dimension of the 1-qubit operator space, it follows that the set is complete, *i.e.*, every 2×2 matrix can be expressed as a linear combination of them. The n-qubit Pauli matrices are constructed as n-tensor products of the 1-qubit matrices and they also form a complete, orthogonal basis set for their respective operator space. In order to satisfy the normality condition, we multiply every matrix in the set by $1/\sqrt{2^n}$.

Once we have chosen our operator basis set, we can express the Kraus operators of a given channel as linear combinations of the basis elements:

$$E_i = \sum_m \alpha_{im} P_m, \tag{20}$$

where E_i is a Kraus operator and $\{P_m\}$ is an operator basis set. We have chosen to represent the basis elements with a P, since our basis of choice is Pauli, but this holds for any other basis set. We can now substitute E_i on equation 18 to obtain:

$$\varepsilon(\rho) = \sum_i \Big(\sum_m \alpha_{im} P_m\Big) \rho \Big(\sum_n \alpha_{in}^* P_n^\dagger\Big) \tag{21}$$

After rearranging, we obtain:

$$\varepsilon(\rho) = \sum_{mn} \Big(\sum_i \alpha_{im} \alpha_{in}^*\Big) P_m \rho P_n^\dagger, \tag{22}$$

which can be simplified to:

$$\varepsilon(\rho) = \sum_{mn} \chi_{mn} P_m \rho P_n^\dagger, \tag{23}$$

where the sum $\sum_i \alpha_{im}\alpha_{in}^*$ has been grouped in the scalar χ_{mn}. Since we have two indices defining these numbers, a convenient way to store them is with a matrix, which is known as the χ or process matrix. For an n-qubit quantum channel, we have 4^n elements in our basis set and, therefore, the χ matrix is $4^n \times 4^n$. Notice that once we have fixed the operator basis set, then the process matrix uniquely defines a quantum channel. If we choose an orthonormal basis, the process matrix has some extra properties:

- It is Hermitian: $\chi_{mn} = \chi_{nm}^*$.
- It is positive semi-definite: all its eigenvalues are greater than or equal to zero.
- The diagonal elements are all real and add up to 2^n.

2.5 *Distance measures*

In quantum information, we are often interested in comparing two quantum states or two quantum channels and quantify how different they are. To compare states, the two most widely used measures are the trace distance and the fidelity. The trace distance between two quantum states of equal dimension with density matrices ρ and σ is given by:

$$D^{\text{tr}}(\rho,\sigma) = \frac{1}{2} tr|\rho-\sigma| \quad , \quad \text{with} \quad |A| = \sqrt{A^\dagger A}. \tag{24}$$

The trace distance is 0 if $\rho = \sigma$ and 1 if the states are orthogonal. Another measure to quantify the difference between two quantum states is the fidelity:

$$F(\rho,\sigma) = tr\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}}. \tag{25}$$

In the special case when one of the states is pure, $\rho = |\psi\rangle\langle\psi|$, the fidelity is given by:

$$F(|\psi\rangle,\sigma) = tr\sqrt{|\psi\rangle\langle\psi|\sigma|\psi\rangle\langle\psi|} = \sqrt{\langle\psi|\sigma|\psi\rangle}, \tag{26}$$

which is simply the square root of the overlap between the two states. The fidelity ranges from 0 for two orthogonal states to 1 for identical states.

There are also various distance measures to compare the difference between quantum channels. Throughtout this thesis, we employ the normalized Hilbert-Schmidt distance [48]. For a channel ε_1 with an associated process matrix χ_1 and another channel ε_2 with an associated process matrix χ_2, the Hilbert-Schmidt distance is given by:

$$D^{\text{HS}}(\chi_1,\chi_2) = \frac{1}{2N^2}\|\chi_1-\chi_2\|^2_{\text{HS}} \tag{27}$$

$$\|A\|_{\text{HS}} = \sqrt{Tr(A^\dagger A)}, \tag{28}$$

where N is the dimension of the Hilbert space upon which the channels act. Throughtout this thesis, we will deal mainly with 1-qubit channels, so $N = 2$. This distance ranges from 0 for identical channels to 1 for orthogonal channels. Although not used in this thesis, other distance measures exist [62].

CHAPTER III

ERROR CORRECTING CODES

Error detection and correction encompass a wide variety of techniques used to suppress errors that occur during the handling and communication of digital information [56]. One of the most promising and widely studied of these techniques is the use of error correcting codes. In these schemes, the digital information is encoded by employing some redundancy, which protects the information from being corrupted. It is useful to begin our description with classical error correcting codes, since historically they were developed first and they operate more intuitively than their quantum counterparts. Throughout this chapter, as an illustrative example, we will focus on the triple modular redundancy code, also known as the bit flip code (BFC).

3.1 Classical error correcting codes

Classical error correcting codes (CECC) are typically used in computer science and telecommunication when transmitting digital information through a noisy channel. Imagine the following situation. We are trying to send 1 bit of information through a communication channel that randomly flips the bit with a probability p $(0 \leq p \leq 1)$. To reduce the probability of a flip, instead of sending that single bit, we prepare 2 copies of it, and send the 3 bits through the channel. We are assuming that noise acts independently on each bit and there are no time or spatial correlations. The receiver can then measure each bit and via majority voting determine what the original bit must have been. This scheme allows us to correct a single flip, while 2 or 3 flips remain uncorrectable. Assuming that the probablity of a flip is the same on every bit, this means that the probabability of error is equal to $p_e = 3p^2(1-p) + p^3$. Notice that $p_e < p$ when $p < 1/2$: if the probability of a flip is less than a half, employing the BFC results in a lower error probability than sending the bare bit. In contrast, if $p > 1/2$ encoding the information is actually more harmful.

3.1.1 Formal definitions and linear codes

With the previous example in mind, we introduce CECCs in a more rigorous way. In information theory, it is customary to refer to a k-bit string as a word of length k. A word can be seen as a vector in the space $\mathbb{Z}_2^k$. If our original message is a word of length k, to protect the information, we can employ a coding function $f : \mathbb{Z}_2^k \longrightarrow \mathbb{Z}_2^n$, and instead send a word of length n. For the scheme to be able to correct errors, then $k < n$. This coding function defines a CECC that encodes k bits of information into n physical bits. Every element in the image of f is referred to as a codeword. In the BFC, $k = 1$ and $n = 3$ and the two codewords are $(0, 0, 0)^T$, the image of 0, and $(1, 1, 1)^T$, the image of 1. Other important concepts when dealing with error-correcting codes are the Hamming weight, the Hamming distance, and the distance of a code. The Hamming weight refers to the number of non-zero elements of a word. The Hamming distance between two words is the Hamming weight of their difference (subtraction modulo 2). Finally, the distance of a code is defined as the minimum Hamming distance between any two codewords. The distance of the BFC is 3. Notice that a distance-3 code can perfectly correct a single error. It can also detect, but not correct, 2 errors. In general, a distance-d code will be able to correct up to $\lfloor (d-1)/2 \rfloor$ errors. Therefore, codes with odd distance values are preferred since they are more economical. Codes are often specified in terms of their important properties: $[[n, k]]$ or $[[n, k, d]]$. Using this nomenclature, the BFC would be a $[[3, 1, 3]]$ code.

In linear codes, the coding function corresponds to a matrix multiplication. This has the advantage that the encoding can be done efficiently. It also results in the property that any linear combination of codewords is also a codeword. Throughout this thesis, we will deal exclusively with linear codes. However, nonlinear codes exist [75, 106]. Notice that the BFC is a linear code. There are various ways to define a linear code $\mathcal{C}$:

1. By the set of all the 2^k codewords x that are images of the coding function f.

2. By the coding function itself, which in this case is known as the generator matrix G. We define G to be a $n \times k$ matrix, in order for the codewords to be column vectors of length n. In many sources, G is defined as a $k \times n$ matrix. We prefer the former format, as we can just apply standard matrix multiplication to obtain the codewords. The generator matrix for the BFC is given by:

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{29}$$

3. By a $(n-k) \times n$ matrix known as the parity check matrix H. The code $\mathcal{C}$ is defined as the null space of H, i.e., the set of all the words $x \in \mathbb{Z}_2^n$ that satisfy the condition $Hx = 0$, where 0 is a column vector. The parity check matrix and the generator also satisfy the condition $HG = 0$. The parity check matrix for the BFC is given by:

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \tag{30}$$

The parity check matrix is used to detect and correct errors easily. We encode the word x as $y = Gx$ and an error e corrupts the codeword to $y' = y + e$. Since $Hy = 0$ for all codewords and the code is linear, $Hy' = He$. The result Hy' is known as the error syndrome. For a distance-3 code, if at most 1 error occurred, then the error syndrome will unequivocally specify which bit to fix. Returning to our already familiar code BFC, we can check that:

$$Hy' = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \text{if} \quad e = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$Hy' = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{if} \quad e = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$Hy' = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{if} \quad e = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$Hy' = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \text{if} \quad e = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

3.2 Quantum error-correcting codes

Correcting quantum information is in general much harder than classical information for several reasons. In the first place, it is impossible to create copies of a general unknown quantum state. This is known as the no-cloning theorem [105]. We can measure the state and determine its complex amplitudes, but this would destroy it. Therefore, our scheme has to be more sophisticated than just generating copies of the quantum state $|\psi\rangle$ and sending them through the channel. What we need to do in order to encode is to map the complex amplitudes to a state with more redundancy and more protection. We select codewords for the computational basis of the qubits we wish to encode and use them to express the quantum state to be encoded. For example, if we wish to encode a single qubit, we select an appropriate codeword for $|0\rangle$, which we label as $|0_L\rangle$ and call *logical 0*. Likewise, we select a codeword for $|1\rangle$, which we label as $|1_L\rangle$ and call *logical 1*. So, to encode a general 1-qubit quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, instead of copying it, we generate the state $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$.

QECCs are also specified by the nomenclature $[[n, k, d]]$. In this case, the code distance is given by the minimum number of errors to map one codeword to another. The quantum version of the BFC (QBFC) has codewords $|0_L\rangle = |000\rangle = (1, 0, 0, 0, 0, 0, 0, 0)^T$ and $|1_L\rangle = |111\rangle = (0, 0, 0, 0, 0, 0, 0, 1)^T$. The distance of the QBFC is still 3. Just like for CECCs, we can define QECCs in several alternative but equivalent ways:

1. As a 2^k-dimensional subspace of an 2^n-dimensional Hilbert space spanned by the

codewords $\{|x_L\rangle\}$. For the QBFC, this would be the 2-dimensional subspace spanned by $\{|000\rangle, |111\rangle\}$. Other QECCs are defined as a subsystem of a subspace of a larger Hilbert space. These so-called operator or subsystem codes [19, 78, 14] offer interesting advantages over subspace codes, but they will not be analyzed in this thesis.

2. As an encoding function $f : (\mathbb{C}^2)^{\otimes k} \longrightarrow (\mathbb{C}^2)^{\otimes n}$. This function is specified by a $2^n \times 2^k$ generator matrix $G = \sum_x |x_L\rangle\langle x|$. For the QBFC, $G = |000\rangle\langle 0| + |111\rangle\langle 1|$. It is also common to turn G into a unitary operation U_G by including $(n-k)$ extra physical qubits initialized in the state $|0\rangle$ before the encoding. In this way the encoding is defined by a square matrix.

3. As a set $\mathcal{A}$ of $(n-k)$ commuting independent operators that stabilize the vectors in the codespace. This is the most common way to think about QECCs. In the case of stabilizer codes, these are n-qubit Pauli operators. Notice that $\mathcal{A}$ only includes the generators of the stabilizers, which form a group $\mathcal{S}$. For the QBFC, $\mathcal{A} = \{ZZI, IZZ\}$.

The stabilizer generators of the QBFC closely resemble the parity check matrix of its classical counterpart. Indeed, just like the parity check matrix in the classical setting, the stabilizers determine the error syndrome. As long as the number of errors is within the correctable scope, the error syndrome will unequivocally determine which error occurred. The procedure consists of measuring each stabilizer generator. Each measurement outcome will indicate whether that particular operator commutes or anticommutes with the error. Notice that since Pauli operators are unitary and Hermitian, their eigenvalues will be exclusively $+1$ and -1. Furthermore, since these operators are traceless too, they will have the same number of $+1$ and -1 eigenvalues. This results in a nice interpretation of stabilizer codes. Since the $(n-k)$ generators commute, they have the same set of (orthogonal) eigenvectors. Each generator will have 2^n eigenvectors: 2^{n-1} with eigenvalue $+1$ and 2^{n-1} with eigenvalue -1. We have 2^{n-k} different combinations of ± 1 eigenvalues. Exactly $2^{n-(n-k)} = 2^k$

L	X_1
$\alpha \lvert 000\rangle + \beta \lvert 111\rangle$	$\alpha \lvert 100\rangle + \beta \lvert 011\rangle$
$\alpha \lvert 010\rangle + \beta \lvert 101\rangle$	$\alpha \lvert 001\rangle + \beta \lvert 110\rangle$
X_2	X_3

Figure 2: The 4 subspaces of the bit flip code (BFC). L refers to the logical subspace, or codespace, where the encoded qubit lives. The other 3 subspaces are labeled by the error that maps them back to the logical subspace.

will share the same combination. So, the stabilizer generators split the 2^n-dimensional Hilbert space into 2^{n-k} orthogonal subspaces of dimension 2^k, each one uniquely labeled by the eigenvalues. One of these subspaces is chosen as the logical subspace or codespace, where the quantum information is encoded. For convenience, this is usually the one with the all $+1$ eigenvalues. Each of the other subspaces will correspond to a unique correctable error. Finally, notice that by measuring the stabilizer generators, we gain no knowledge about the amplitudes α and β, so the coherence of the superposition is maintained. Figure 2 illustrates the 4 subspaces of dimension 2 that result from the QBFC.

At the single bit level, the only possible error is a flip. At the single qubit level, the number of possible error is in principle infinite. However, the act of measuring the stabilizer generator effectively discretizes these continuous errors. In the first place, as with CECCs, any QECC will have a set of correctable and a set of uncorrectable errors. For a quantum code with a projector P onto the logical subspace, the set of correctable errors is given by $\{E_i\}$ if and only if:

$$PE_i^\dagger E_j P = \alpha_{ij} P, \tag{31}$$

where α is a Hermitian matrix. The interested reader can study the proof on [77]. Notice that for the QBFC, the set of correctable errors is $\{III, XII, IXI, IIX\}$. Of course, III corresponds to no error, but we include it because it satisfies equation 31.

The real power of QECCs comes from another fundamental result. Imagine we have a quantum code whose set of correctable errors is given by $\{E_i\}$. We construct another set $\{F_j\}$ formed by linear combinations of the operators E_i: $F_j = \sum_i m_{ji} E_i$ for some matrix m_{ji} of complex numbers. Quite amazingly, the operators in the set $\{F_j\}$ are also correctable errors. Once more, the interested reader can study the proof of this important theorem in [77].

For the QBFC, this result has important implications. Imagine the operator $F_0 = \cos(\theta/2) III - i \sin(\theta/2) XII$, for some arbitrary θ. This corresponds to a coherent rotation about the XII axis (the X axis on the first qubit) by an angle θ. The result on the previous paragraph implies that this error is correctable for every real value of θ. The key insight is that, although this error maps the state of the logical qubit to a superposition of vectors on different subspaces, the measurement of the stabilizers collapses that superposition to the codespace with probability $\cos^2(\theta/2)$ and to the X_1-subspace with probability $\sin^2(\theta/2)$. In other words, measuring the stabilizers discretizes a continuous error to one of the errors the code is meant to correct, as long as the error can be expressed as a linear combination of the original correctable errors.

Despite its usefulness in illustrating the properties codes, the QBFC is not a good QECC since it cannot correct any 1-qubit error. In particular it cannot correct a phase or Z error. Numerous codes have been developed that can handle any error, not just bit flip. One of the earliest and most widely studied is the $[[7, 1, 3]]$ Steane code [91]. The stabilizer generators of this code are:

$$\{IIIXXXX, IXXIIXX, XIXIXIX, IIIZZZZ, IZZIIZZ, ZIZIZIZ\}. \tag{32}$$

It is straightforward but tedious to check that the Steane code can correct any Pauli error on a single qubit. Since any 1-qubit operation can be expressed as a linear combination of the Pauli matrices and the Identity, it follows that the Steane code can correct any 1-qubit error. The Steane code is employed on Chapters 5 and 6 to study the properties of several error approximations in the context of error correction.

CHAPTER IV

IMPROVING THE APPROXIMATION OF NON-STABILIZER ERRORS

This chapter is based on the following paper:

Mauricio Gutiérrez, Lukas Svec, Alexander Vargo, and Kenneth R. Brown, "Approximation of realistic errors by Clifford channels and Pauli measurements", Phys. Rev. A **87**, 030302(R) (2013).

As explained in the previous chapter, a quantum computation is classically tractable as long as the state of the system remains stabilized by Pauli operators at every time step. If we start with a Pauli state, then the time evolution has to be given by Clifford operators and measurements in a Pauli basis. This includes not just the desired operations in the circuit, but the noise as well. Of course, in general, the noise will not be described as a unitary operation, but at least it has to be representable as a Kraus channel formed by Clifford operators and Pauli measurements.

A standard error model used in simulations of quantum circuits is the Pauli channel (PC). In this model, a Pauli operator, chosen from a probability distribution, is applied at every possible error position [93, 64, 30, 26]. The Kraus representation of the PC corresponds to:

$$\text{PC} = \begin{cases} E_0 = \sqrt{1-p_1-p_2-p_3}\, I \\ E_1 = \sqrt{p_1}\, X \\ E_2 = \sqrt{p_2}\, Y \\ E_3 = \sqrt{p_3}\, Z, \end{cases} \tag{33}$$

where p_1, p_2, and p_3 correspond to the probabilities of applying a Pauli X, Y, or Z operation, respectively. The PC serves as a good approximation to some common laboratory processes such as dephasing. It also serves as a good approximation for most error process that lead to a steady-state in which the system becomes maximally mixed. These are unital channels that map completely mixed states to completely mixed states ($\varepsilon(I) = I$).

In nature it is also common to encounter interactions with the environment that lead to non-unital error channels in which the maximally mixed states are not a fixed point of the error process. One example is the amplitude damping introduced in the previous chapter. If an error channel is far from unital, then simulating it with Pauli errors gives large approximation errors, thus making it hard to extract useful results.

We have gone beyond simulating errors with the conventional PC. Rather than restricting to Pauli errors, we allow any subset of efficiently simulable gate errors to occur. In particular, we look at subsets generated by including all Clifford group operators and/or Pauli measurements to the PC. We have found that adding Clifford errors and/or measurement errors always results in more accurate approximations and significant improvements for most error channels.

We begin by considering all error channels that can be simulated efficiently within the stabilizer formalism. Then, we discuss the method by which we create an error channel that approximates a target channel.

4.1 Efficiently simulable error processes

The Clifford group for n-qubits can be generated from CNOTs and the 1-qubit Clifford gates, which preserve the symmetry of the chiral octahedron [100]. As error channels, the probabilistic application of 1-qubit Clifford operators can be represented by the following Kraus operators:

- Identity

  $E_0 = \sqrt{p_0}I$

- Pauli operators

  $E_i = \sqrt{p_i}\sigma_i$

- S-like operators

  $E_{S,\pm j} = \sqrt{p_{S,\pm j}} \exp(-i\frac{\pi}{4}(\pm\sigma_j))$

- Hadamard-like operators

  $E_{j,\pm k} = \sqrt{p_{j,\pm k}} \exp(-i\frac{\pi}{2}\frac{1}{\sqrt{2}}(\sigma_j \pm \sigma_k))$ for $k > j$

- Rotations about the face centers

  $E_{\vec{F}} = \sqrt{p_{\vec{F}}} \exp(-i\frac{\pi}{3}\sigma_{\vec{F}})$, where $\sigma_{\vec{F}} = \vec{F} \cdot \vec{\sigma}$ and $\vec{F}$ is the unit vector from the origin to one of the eight faces of the 1-qubit Clifford octahedron [100].

One can create an error process which is the weighted random application of these 24 unitary operators. We call this class of error models the Clifford Channel (CC) [68]. Most simulations of error correction circuits have used Pauli depolarizing channel (PC), which is a subset of CC consisting of only the random application of I or Pauli operators.

The stabilizer formalism also allows for efficient simulation of non-unital operations involving Pauli measurements and, optionally, conditional application of Clifford gates based on those measurements. We have focused on the set of operators that corresponds to measuring a Pauli operator and then conditionally applying a Pauli matrix such that all states map to the same state. We call these channels measurement induced translations. For each eigenstate, $|\lambda\rangle$, of a Pauli operator, we define the channel $\mathcal{E}_\lambda$ by two Kraus operators: $E_{\lambda 0} = |\lambda\rangle\langle\lambda|$ and $E_{\lambda 1} = |\lambda\rangle\langle\lambda^{\perp}|$. Notice that the effect of these two operators is to discard the state and replace it by $|\lambda\rangle$. We add these channels to our model with probability p_λ. The effect on a state, when represented on the Bloch sphere, is to translate it toward $|\lambda\rangle$. This allows us to generate non-unital error channels that can be efficiently simulated. The extended models including measurement are labeled PMC and CMC. Table 1 describes the content of each channel class in terms of the underlying channel set and the number of free parameters.

- Measurement-induced translations

  $E_{|f\rangle\langle f|} = \sqrt{p_{|f\rangle}}|f\rangle\langle f|$

$$E_{|f\rangle\langle f^{\perp}|} = \sqrt{p_{|f\rangle}}|f\rangle\langle f^{\perp}|$$

Notice that the effect of these two operators is to discard the state with a probability of $p_{|f\rangle}$ and replace it by $|f\rangle$. The effect on a state, when represented on the Bloch sphere, is to translate it toward $|f\rangle$. To ensure trace preservation, we set $p_0 = \sqrt{1 - \sum_a p_a}$ where a sums over all other operators.

Table 1: Four error models compatible with the stabilizer formalism.

Channel Class	Channel Set	Parameters
PC	$\{I, \sigma_i\}$	3
PMC	$\{I, \sigma_i, \mathcal{E}_\lambda\}$	9
CC	$\{I, \sigma_i, S_v, H_e, R_f\}$	23
CMC	$\{I, \sigma_i, S_v, H_e, R_f, \mathcal{E}_\lambda\}$	29

4.2 Evaluating the approximations

To study how closely our error models approximate target error channels, we compute the distance between the process matrix of our error model and the process matrix of the target error. For an error model with n operators (including the identity), this distance is a function of the $n - 1$ linearly independent probabilities associated with the operators. As a distance measure we employ the normalized Hilbert-Schmidt distance, introduced in Chapter 2. After calculating the distance, we then minimize it over the $n - 1$ independent variables.

As our goal is to understand for which cases this error model would be an appropriate approximation, we want our model to be an upper bound to the error induced on the system. Therefore, we perform the distance minimization with the constraint that the fidelity between the identity channel and our error model is not greater than the fidelity between the identity channel and the target error. This constraint ensures that our approximation will not underestimate the real target error.

$$F(I, \text{Target}) \geqslant F(I, \text{Model}) \tag{34}$$

The fidelity can either be an average fidelity:

$$F_{av}(V,K) = \frac{1}{N^2}\sum_i |\mathrm{Tr}(V^\dagger K_i)|^2 \tag{35}$$

where $\{K_i\}$ are the Kraus operators of the error channel K and V is a unitary transformation, or a worst-case fidelity:

$$F_w(V,K) = \min_{\rho\,\epsilon\,D}\sum_i |\mathrm{Tr}(V^\dagger K_i \rho)|^2 \tag{36}$$

where in this case the fidelity is minimized over all the density matrices ρ. The minimization was performed with Python's sequential least squares programming minimization subroutine.

We use the Hilbert-Schmidt distance for most of the analysis here due to ease of computation, but the method works for any distance measure or constraint [44]. In most cases, the worst case fidelity constraint would be appropriate for calculating lower bounds on error correction thresholds. For certain cases, such as Pol_ϕC, the two constraints give the same results.

4.3 Approximating the Amplitude Damping Channel (ADC)

Figure 3 shows the results of the approximation of the ADC by the error models introduced in Section 4.1 with the average fidelity constraint. Each one of the 200 points corresponds to a numerical minimization for a particular damping strength. After fitting these points and then solving symbolically, for both the PC and the CC the distance between the ADC and the best approximation was found to be $D_P = \gamma^2/8$, where γ is the damping strength. This means that as the non-unital character of the ADC becomes more pronounced, the unital error models give less an accurate approximation. The larger repertoire of operators in the Clifford group does not improve the approximation obtained with only Pauli operators.

On the other hand, the addition of the measurement-induced translations considerably improves the approximation. In this case, the distance between the approximation and the ADC is given by $D_m = (\gamma-1)(\gamma+2\sqrt{1-\gamma}-2)/8$, and the PMC and CMC significantly

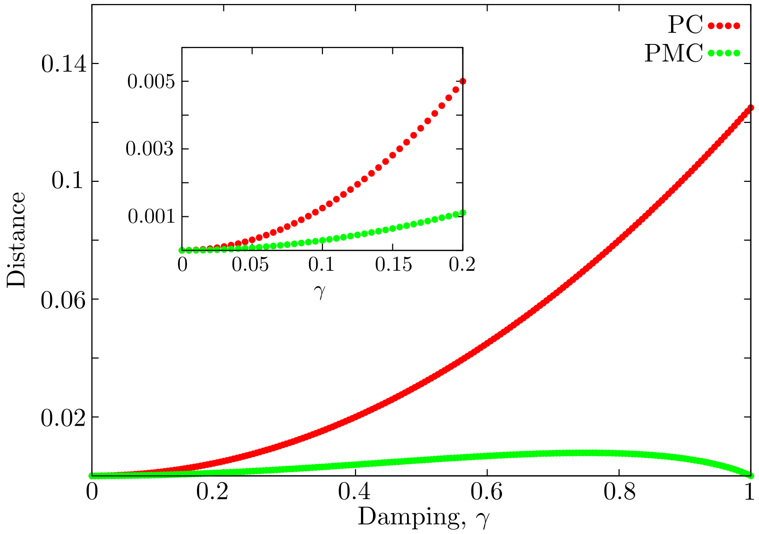

Figure 3: The results for the CC and CMC are exactly the same as the results for PC and PMC, respectively, since the additional Clifford operators do not improve the approximation. The inset figure, a zoomed version of the same plot, gives an idea of how fast an error model without measurement-induced translations becomes an inaccurate approximation compared to an error model that includes them. For small values of γ both distances scale quadratically.

outperform the models without measurement for $\gamma > 0.05$. The PMC and CMC can match the ADC perfectly only for $\gamma = 0$, which corresponds to the trivial case, and $\gamma = 1$, which corresponds to a measurement that is actually part of the operator repertoire of our error model. Interestingly, despite the large amount of operators in the CMC error model, the best approximation only employs the identity and the translation towards $|0\rangle$ and it is given by:

$$\mathrm{PMC}_{\mathrm{ADC}} = \begin{cases} E_0 = \sqrt{1-p_m}\, I \\ E_1 = \sqrt{p_m}\, |0\rangle\langle 0| \\ E_2 = \sqrt{p_m}\, |0\rangle\langle 1| \end{cases} \tag{37}$$

with $p_m = (1 + \gamma - \sqrt{1-\gamma})/2$. It is also noteworthy that, for small γ values,

$$D_m = \frac{\gamma^2}{32} - \frac{\gamma^3}{64} + \mathcal{O}(\gamma^4), \quad \text{while} \quad D_P = \frac{\gamma^2}{8}. \tag{38}$$

Although the measurement operators improve the approximation even for small γ values, both methods have a quadratic dependence on γ.

When the constraint is changed from the average fidelity to the worst fidelity, then the PC and CC approximations have a distance of

$$D_{P,w} = \frac{2\gamma^2 - 3\gamma + 2 + 2\gamma\sqrt{1-\gamma} - 2\sqrt{1-\gamma}}{4}, \tag{39}$$

while the PMC and CMC have a distance of $D_{m,w} = 2D_m$. Both of these cases result in larger distances than the ones with the average fidelity approximation and the difference between models with and without measurements is even more pronounced.

The results obtained by the average fidelity and the worst fidelity contraints are best illustrated in Figure 4. Here we examine, for $\gamma = 0.25$, the closest PC (a) and PMC (b) approximation assuming either one of the two constraints. The figure shows a cross section of the Bloch sphere and its transformation by the ADC and the closest approximate channel with either the average fidelity constraint (red) or the worst fidelity constraint (blue). Notice that for these error channels the deformed Bloch sphere is still symmetric with respect to rotations around z, so a cross section is enough to visualize the whole process.

The approximation using the worst fidelity constraint guarantees that the largest distance between any input and the target channel output will be less than the largest distance between any input and the approximate channel output. In this case, for both the ADC and its approximations the largest discrepancy between input and output occurs when the initial state is $|1\rangle$. Notice that for the PMC approximation this constraint also guarantees that for all inputs the approximate channel outputs are further from the input than for the target channel. This is pictorially represented in Figure 4(b), where the blue curve is always *inside* the green curve or further away from the initial states (black curve). For the PC, however, this is not the case, as Figure 4(a) shows. Here the blue curve lies *outside* the green curve for some points. Indeed, if we use a unital channel to approximate a non-unital one, it is impossible to satisfy the condition that for every input the approximate channel output will be further from the input than for the target channel. Simply consider the maximally mixed state, which is mapped to itself by a unital channel, but mapped to a different state by a non-unital one.

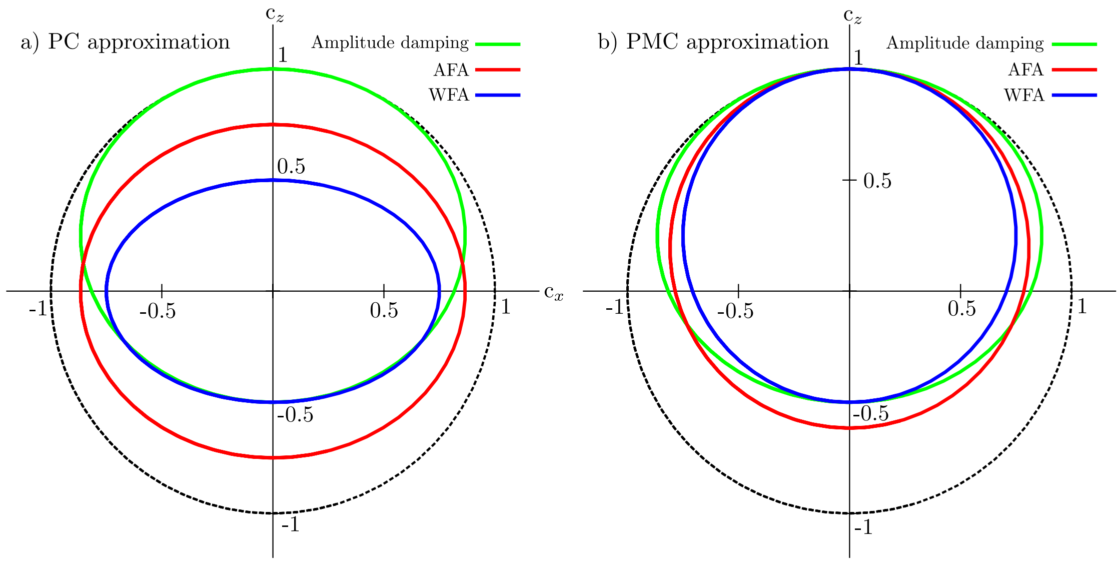

Figure 4: AFA stands for average constraint approximation, while WFA stands for worst constraint approximation. a) Channels without measurement operators. b) Channels with measurement operators. For both cases, $\gamma = 0.25$.

4.4 Approximating the Polarization along an axis in the X-Y plane Channel (Pol$_\phi$C)

Figure 5 shows the results of the approximation of the Pol$_\phi$C by the error models introduced earlier. Once again, each one of the 200 points corresponds to a numerical minimization. Because of the unital nature of this channel, it is the addition of the Clifford operators rather than the measurement operators that improve the approximation. For both the PC and the PMC, the distance between Pol$_\phi$C and the best approximation was found to be:

$$D_P = \frac{1}{4}p^2 \sin^2(2\phi). \tag{40}$$

When the Clifford operators are included in the approximate channel, the new distance is reduced to:

$$D_C = \frac{3}{28}p^2(\sin(2\phi) + \cos(2\phi) - 1)^2 \tag{41}$$

for $0 \leq \phi \leq \pi/4$ and for $p < 0.9$. The minimum distance is given by this function only when $p \leq 7/(6+\sqrt{2}) \approx 0.944$. Because we are interested in small errors, we will not deal with the $p > 7/(6+\sqrt{2})$ case. Furthermore, the expression in equation 41 is only valid for $0 < \phi < \pi/4$. For other intervals, the distance is the same expression translated by the corresponding amount. For example, for the $\pi/4 < \phi < \pi/2$ interval, the distance is:

$$D_C = \frac{3}{28} p^2 (\sin[2(\phi - \pi/4)] + \cos[2(\phi - \pi/4)] - 1)^2. \tag{42}$$

At the worst point of the CC (which in this interval occurs at $\phi = \pi/8, 3\pi/8$), the PC is 6.8 times worse. Notice that not only the distance is decreased; the period of the distance function is also reduced from $\pi/2$ to $\pi/4$, because between every two Pauli axes there is a Clifford axis.

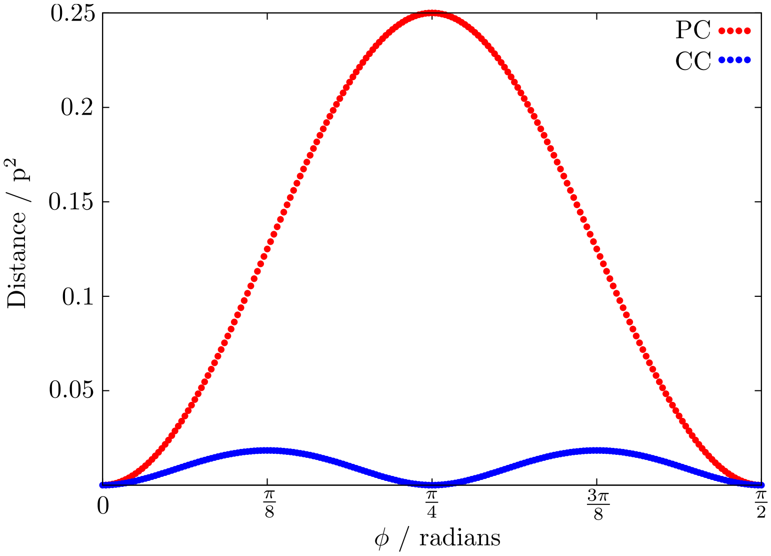

Figure 5: The results for PMC and CMC are the same as the results for PC and CC, respectively, since the channel is unital and the approximation does not benefit from the measurement-induced translation operators. The distances scale quadratically with p, so the results are normalized by p^2.

Once again, despite the large amount of operators in the CMC, the best approximation uses a small number of them: the identity and the two axes closest to the polarization axis. If we only employ Pauli axes, the best approximation is:

$$\mathrm{PC}_{\mathrm{Pol}_\phi \mathrm{C}} = \begin{cases} E_0 = \sqrt{1-p}\, I \\ E_1 = \sqrt{p}\, \cos(\phi)\, X \\ E_2 = \sqrt{p}\, \sin(\phi)\, Y, \end{cases} \tag{43}$$

where ϕ and p are the same as in Equation (2). If we employ the whole Clifford group, the best approximation is given by:

$$\mathrm{CC}_{\mathrm{Pol}_\phi \mathrm{C}} = \begin{cases} E_0 = \sqrt{1-p_1-p_2}\, I \\ E_1 = \sqrt{p_1}\, X \\ E_2 = \sqrt{p_2}\, H_{XY} = \sqrt{p_2/2}(X+Y) \end{cases} \tag{44}$$

where $p_1 = p(3 + 4\cos(2\phi) - 3\sin(2\phi))/7$, and $p_2 = p(3 - 3\cos 2\phi + 4\sin(2\phi))/7$. Finally, as mentioned before, for this error channel there is no difference between the results obtained with either fidelity constraint.

4.5 *Approximating Random Error Channels*

We have seen that the addition of the measurement-induced translations and the Clifford operators improves the approximation of two specific error channels. To determine how the method works for general errors, we generated 1000 random process matrices and computed the distance of the best approximation that each one of the 4 approximate channels could make. For the 1-qubit case, a process matrix is a 4×4 Hermitian positive matrix M with 4 constraints in the normalized Pauli basis: $tr(M)$ = 2, Re(M_{01}) = -Im(M_{23}) , Re(M_{02}) = Im(M_{13}) , and Re(M_{03}) = -Im(M_{12}). To generate this matrix we first create a 4×4 diagonal matrix D with real, positive diagonal entries that add to 2. We then create a 4×4 random unitary matrix U and apply this unitary transformation to D to obtain $M = UDU^\dagger$, which is positive with trace 2. We then enforce the last 3 constraints mentioned earlier and keep the random process if the matrix is still positive.

Figure 6 illustrates the distance between each random error channel and the best approximation as a function of the distance between the error channel and the identity channel. Notice that as the number of operators in the error models increases, both the mean and the median distance between each model and the random error decreases and the distributions become more compact, as summarized in Table 2. For this data, the improvement of adding either Clifford gates, CC, or measurement-induced translation operators, PMC, over PC is

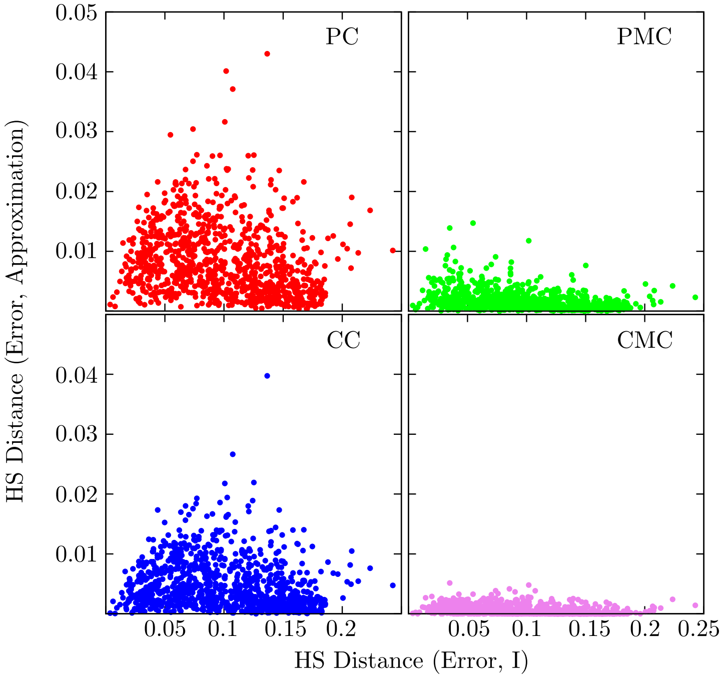

Figure 6: The slope of a line joining the origin and a point represents the distance of the best approximation to that error relative to the magnitude of the error.

comparable. The total set of operators leads to an order of magnitude improvement over the PC. In the case of the CMC, for the 1000 random channels tested, the number of non-zero parameters used in the approximations varied from 4 to 29 with a median of 12. This is in contrast to the ADC and the Pol$_\phi$C where only 1 and 3 parameters, respectively, are required due to the symmetry of the error channels.

Table 2: Summary of the approximations obtained with each of the 4 error models.

Channel	Distance mean	Distance median	Distance variance
PC	1.7×10^{-2}	1.4×10^{-2}	1.4×10^{-4}
PMC	3.4×10^{-3}	2.4×10^{-3}	1.1×10^{-5}
CC	9.8×10^{-3}	7.5×10^{-3}	7.0×10^{-5}
CMC	1.1×10^{-3}	4.2×10^{-4}	2.2×10^{-6}

CHAPTER V

COMPARISON OF THE STEANE CODE THRESHOLD FOR EXACT AND APPROXIMATE ERRORS

This section is based on the following paper:

Mauricio Gutiérrez and Kenneth R. Brown, "Comparison of a quantum error-correction threshold for exact and approximate errors", Phys. Rev. A **91**, 022335 (2015).

The threshold theorem of quantum error correction promises the accurate implementation of arbitrary size quantum algorithms if the underlying physical errors are below certain values. The error thresholds depend strongly on the specific quantum error correcting code, how errors are detected and fixed [82, 45, 94], and what errors are assumed [80, 41, 99, 59]. Most codes have been designed to fix random Pauli errors and error correction procedures can be simulated efficiently using the stabilizer formalism [47, 2]. As explained in Chapter 4, a broader class of errors including Clifford operations [68] and Pauli measurements [52] can also be included in this formalism.

Here we examine whether these improved approximations also lead to more accurate threshold estimates. Specifically, we calculate the level-1 pseudo-threshold for the Steane [[7,1,3]] code [91] for two non-stabilizer errors, amplitude damping and a depolarization channel along a magic-state axis, and compare the exact solution to approximations based on Pauli errors or Clifford and Pauli measurement errors. The Steane code has been well studied theoretically [93, 95, 71, 98, 104, 4, 11] and a logically encoded state has been recently demonstrated experimentally [79]. The code is small enough to allow for exact simulation similar to recent work on distance-3 surface codes, which compared a realistic error model corresponding to T_1 (amplitude damping) and T_2 (dephasing) processes and an approximate Pauli error model based on twirling [99].

In addition to the pseudo-threshold, we are interested in two other qualities of the approximation, the accuracy and the honesty. The accuracy is a measure of how close is the state generated by the approximate evolution to the state generated by the exact evolution. We describe an approximation as honest if the final state after the approximate evolution is further from the initial state than the final state after the exact evolution. In other words, an approximation is honest if it upper-bounds the error of the exact evolution. As pointed out by Puzzuoli *et al.* the composition of honest approximations is not necessarily honest [83]. This implies that an approximation that is honest at the 1-qubit physical level might lead to a dishonest representation of the overall error produced on the system. As our goal is to employ our approximate channels to infer the performance of error-correcting strategies under realistic non-stabilizer noise, we need to be sure that they compose in an honest fashion. We provide numerical evidence that, in the context of an error-correcting circuit, an honest approximation at the physical level remains honest at the logical level. Furthermore, we show that, for the error models studied, physically dishonest approximations based on the Pauli channel might lead to approximations at the logical level that are both approximately honest and very accurate, in agreement with similar results obtained by Geller and Zhou [43]. This suggests that it might not be necessary for the approximations to be honest at the physical level.

This chapter is organized as follows. First, we review the important concepts of honesty and accuracy of an approximate channel. Next, we explain our procedure for calculating the pseudo-thresold. Finally, we present our results.

5.1 Honesty and accuracy at the physical and logical levels

For each approximate channel, we study two properties: honesty and accuracy. An approximate channel is honest if it does not underestimate the detrimental effect of the target error channel. The accuracy of an approximate channel refers to how closely it can mimic the effect of the target channel on an initial state. More explicitly, if a target error channel E

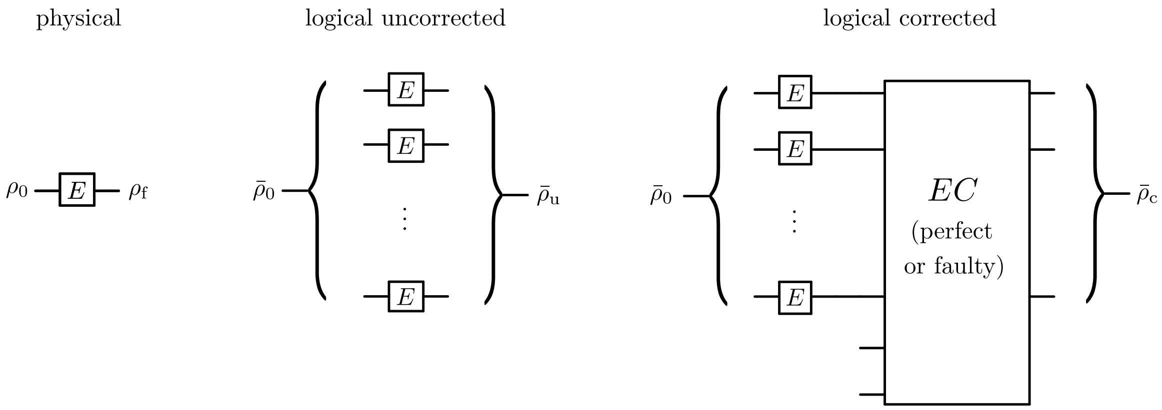

Figure 7: We focus on the physical level and the logical level before and after error correction. The logical initial state is encoded without errors and then errors are applied. The preparation of the ancillary cat state in the faulty EC is error-free as well, as shown in Fig. 8.

maps a pure state ρ to $E(\rho)$ and an approximate channel A maps the same state to $A(\rho)$, then A is honest if

$$D^{\mathrm{Tr}}(\rho, E(\rho)) \leq D^{\mathrm{Tr}}(\rho, A(\rho)) \tag{45}$$

for every pure state in the initial physical or logical space. The accuracy is measured by the average trace distance between the resulting states:

$$\Big\langle D^{\mathrm{Tr}}(E(\rho), A(\rho)) \Big\rangle. \tag{46}$$

Notice that for both properties, our measure of choice is the trace distance. A good approximate channel will be honest (or as least dishonest as possible) and as accurate as possible, not only at the physical level, but also at the logical levels. We distinguish 4 different scenarios to compare honesty and accuracy: (a) the physical (1-qubit) level, (b) the uncorrected logical level, (c) the logical level with perfect EC, and (d) the logical level with faulty EC, as depicted in Fig. 7.

For each target non-Clifford error channel, we study two kinds of approximations: (a) the Pauli channels (PC), which employ only 1-qubit Pauli operators, and (b) the expanded channels or Clifford+measurements channels (CMC), which include all the 1-qubit Clifford operators and the measurement-induced translations [52]. In turn, each kind of approximation is performed with the average fidelity constraint ("a") and the worst trace distance constraint ("w"), resulting in four approximate channels. Notice that the unconstrained PC is equivalent to the Pauli Twirled Approximation [68, 83, 43], the channel obtained by removing the off-diagonal elements from the target channel's process matrix in the Pauli basis [24]. We also analyze the completely isotropic Pauli channel or depolarizing channel (DC), the most common error model used when calculating thresholds. In this paper we are comparing single qubit error channels and we only use the single qubit depolarizing channel. This channel is a version of the PCa where the coefficients corresponding to each Pauli matrix are forced to have the same value. This error model serves as a reference. The approximations are summarized in Table 3.

Table 3: Summary of the various target and approximate channels.

Channel	Complete name	Honesty constrained
ADC	amplitude damping	–
Pol$_\phi$C	polarization along non-Clifford axis	–
PCa	Pauli	no
PCw	Pauli	yes
CMCa	Clifford+measurements	no
CMCw	Clifford+measurements	yes
DC	Depolarizing channel	no

5.2 Calculation of the pseudo-threshold

Our objective with respect to the pseudo-threshold is twofold. On the one hand, we want to study how sensitive a QECC's threshold is to the noise model. On the other hand, we want to determine if the thresholds obtained with our expanded error models approximate the realistic threshold more accurately than the PC.

5.2.1 Procedure to compute the level-1 pseudo-threshold

Because our target error models are non-stabilizer, we perform exact (full density matrix) simulations of quantum error correction (QEC) circuits up to the first level of encoding. We calculate a particular QEC code's level-1 pseudo-threshold under a given error channel in the following way:

1. Run the physical circuit:
    (a) Choose an initial 1-qubit pure state, $|\psi\rangle$.
    (b) Apply the selected error channel.
    (c) Compute the fidelity between the initial and final states.
2. Run the logical circuit:
    (a) Encode the initial state using the selected QECC.
    (b) Apply the error channel to each physical qubit.
    (c) Perform EC.
    (d) Compute the fidelity between the initial and final logical states.

We are interested in how much the final state is affected by errors which are uncorrectable by the selected QECC. Therefore, for the case with faulty corrections, we perform one round of perfect EC before computing the fidelity. This has the effect of eliminating correctable errors which happened during or after the faulty EC. The process of performing a round of perfect EC and then computing the fidelity can also be viewed as computing an error-corrected fidelity:

$$F_{\text{EC}}(|\psi_L\rangle, \rho_L) = \sqrt{\sum_i \langle\psi_L|E_i^\dagger P_i^\dagger \rho_L P_i E_i|\psi_L\rangle}, \tag{47}$$

where $|\psi_L\rangle$ is the initial logical state and ρ_L is the final logical state, which, in general, will not be pure. The set $\{E_i\}$ consists of all error operators which the QECC is designed to correct, while $\{P_i\}$ is the set of projectors to the subspaces associated with each error. For the Steane [[7,1,3]] code, the set $\{E_i\}$ is formed by the 64 Pauli operators formed by all possible combinations of X and Z errors acting independently on at most one qubit and includes the Identity operator for the case of no errors.

3. Repeat steps (1) and (2) for various noise strengths to obtain fidelities for the physical and logical circuits. The threshold is given by the first intersection between the two curves.

4. Repeat this procedure for several initial states to obtain an average threshold. For the perfectly corrected case, we select 80 initial points uniformly distributed on the Bloch sphere. For the faultily corrected case, we select 20.

Notice that our logical unit consists of a logical Identity gate, which is always faulty, and an EC step, which may or may not be faulty. This logical unit is often referred to as a simple rectangle, in contrast to an extended rectangle, where the logical gate is inserted between two EC steps [10, 26]. The qualitative trends of the resulting pseudo-thresholds should remain unchanged between a simple and an extended rectangle.

5.2.2 Methods of error correction

The EC step is performed by measuring the stabilizer generators and later correcting any detected errors. We distinguish between the error-free EC, which results in a code-capacity pseudo-threshold, and the faulty EC, which results in the more realistic circuit-based pseudo-threshold. The faulty EC is built by inserting an error channel after each gate in the original circuit. As the Steane code will be the focus of our analysis, consider, for example, the measurement of the stabilizer $IIIXXXX$, as depicted in Fig. 8. The error-free EC step would consist of circuits analogous to (a) for each stabilizer generator. On the other hand, in the faulty EC regime, each stabilizer generator would be measured as shown in (b). Each

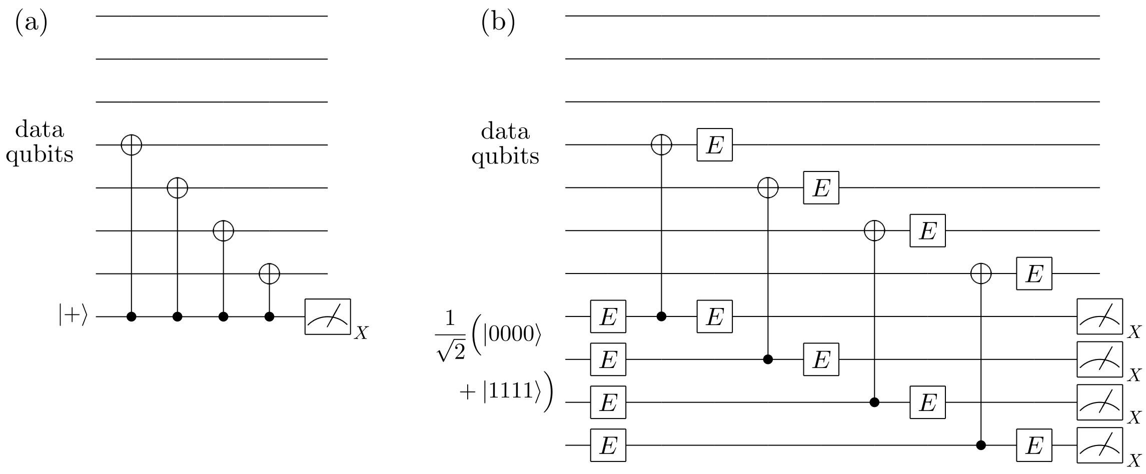

Figure 8: In the former case, we only need to employ 1 ancillary qubit. Notice that the ancillary qubit starts in the $|+\rangle$ state and the measurement is performed in the X basis. In the faulty EC regime, in order to make the procedure fault tolerant, we employ 4 ancillary qubits initialized in a cat state [33]. We then measure each ancillary qubit in the X basis and compute their parity to extract the outcome.

stabilizer measurement is then repeated and the syndrome is compared to the one in the previous round. If there is a disagreement between these two, a third round of stabilizer measurements is performed and its syndrome is selected as the definitive one.

5.3 Honesty and accuracy of the approximations

By construction, the "w" approximations are honest at the physical (1-qubit) level, provided that the initial state is pure. In our previous work we also determined that, when approximating a general non-Clifford channel at the physical level, the expanded channels are more accurate than the Pauli channels. Before computing the level-1 pseudo-thresholds for different approximations, we first examine if the honesty of the "w" approximations and the greater accuracy of the expanded channels were maintained at the logical level.

5.3.1 Amplitude Damping Channel (ADC)

For the physical, logical uncorrected, and logical with perfect EC levels, we have selected 80 initial states uniformly distributed over the Bloch sphere surface. For the logical faultily corrected level, we have selected 20 points, because the simulations involve 3 extra qubits and consequently take an exponentially longer time. We have computed the trace distance between each initial state and the resulting final state after the ADC and its approximations. The average distances are shown in the first row of Fig. 9 as a function of the damping strength, γ. Likewise, we have computed the trace distance between each final state after the ADC and each final state after every approximate channel. The average distances are presented in the second row of Fig. 9. The behavior in the limit of small damping strength ($\gamma \to 0$) is summarized in Tables 4 and 5. In this limit, it is useful to Taylor-expand the distances in terms of the noise strength and compare the coefficients of the leading order terms. Expectedly, for the corrected logical cases the linear term is suppressed and the leading order is quadratic. For the physical and uncorrected logical cases, the leading order is linear. At the logical level with faulty EC, simulations were only carried out at low damping strengths $\left(\gamma \in [10^{-5}, 10^{-3}]\right)$, which is the pertinent region for the pseudo-threshold computation.

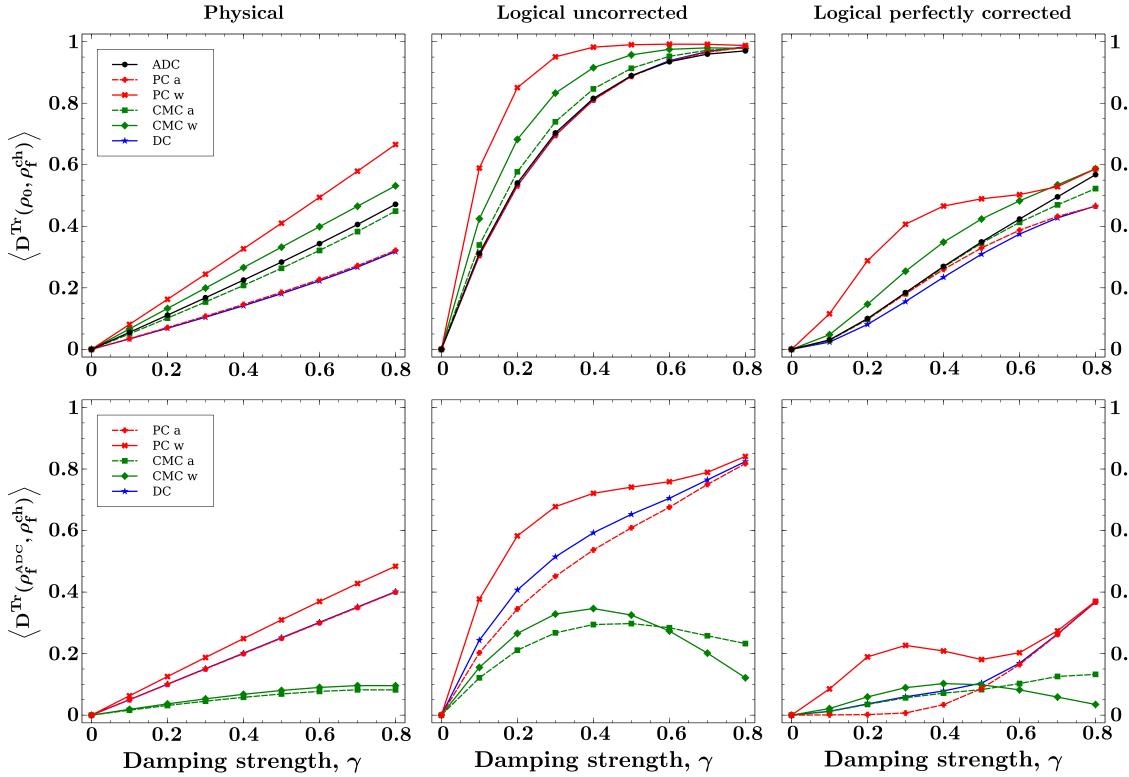

Figure 9: Distances used to assess the honesty (top row) and accuracy (bottom row) of the approximate channels to the ADC at various levels.

Table 4: Honesty of the approximate channels to the ADC in the limit of small damping strength. Standard deviations below 10^{-9} are not reported.

Channel	Physical	Logical uncorrected	Logical perfectly corrected	Logical faultily corrected
	$\mathrm{D}^{\mathrm{Tr}}/\gamma$	$\mathrm{D}^{\mathrm{Tr}}/\gamma$	$\mathrm{D}^{\mathrm{Tr}}/\gamma^2$	$\mathrm{D}^{\mathrm{Tr}}/(10^2\gamma^2)$
ADC	$0.55(27)$	3.62	$3.76(96)$	$8.0(1.8)$
PCa	$0.347(79)$	3.50	$3.76(96)$	$7.8(1.8)$
PCw	$0.81(12)$	8.35	$18.5(3.5)$	$37.7(8.0)$
CMCa	$0.50(18)$	4.00	$3.48(45)$	$6.3(1.2)$
CMCw	$0.66(24)$	5.33	$6.19(80)$	$11.3(2.2)$
DC	0.333	3.50	$2.75(36)$	$4.95(96)$

Table 5: Accuracy of the approximate channels to the ADC in the limit of small damping strength. Standard deviations below 10^{-9} are not reported.

Channel	Physical	Logical uncorrected	Logical perfectly corrected	Logical faultily corrected
	$\mathrm{D}^{\mathrm{Tr}}/\gamma$	$\mathrm{D}^{\mathrm{Tr}}/\gamma$	$\mathrm{D}^{\mathrm{Tr}}/\gamma^2$	$\mathrm{D}^{\mathrm{Tr}}/(10^2\gamma^2)$
PCa	0.500	2.41	$7(12) \times 10^{-6}$	$0.123(28)$
PCw	$0.63(26)$	4.94	$14.8(2.6)$	$29.8(6.2)$
CMCa	$0.166(60)$	1.35	$1.61(44)$	$2.15(74)$
CMCw	$0.194(60)$	1.75	$3.05(94)$	$3.7(1.1)$
DC	$0.505(97)$	2.92	$1.68(69)$	$3.2(1.2)$

Notice that at the physical level in the first row of Fig. 9, the “w” approximations result in curves that are above the target ADC by construction, while the “a” approximations produce curves below it. This behavior is also present in the small noise strength limit, as can be seen by the magnitudes of the linear coefficients (Table 4): PCa $<$ CMCa $<$ ADC $<$ CMCw $<$ PCw. Likewise, the accuracies of the CMC approximations are much better than that of the PC approximations (Table 5). In the $\gamma \to 0$ limit, the CMC approximations are ≈ 3 times more accurate.

At the three logical levels, the “w” approximations are honest for every damping strength. This is true not just on average, but for every initial state considered. This is an important result, as it means that we can safely use the “w” approximations as a substitute of the ADC when determining codes’ thresholds or other error-correcting properties. Remarkably, the dishonesty of the PCa is greatly reduced from the physical to the logical levels in the limit of small γ. Its error is below the honesty cutoff by 36% at the physical level but by less than 2% for both corrected logical levels and well within the deviation in the distance. In contrast, the dishonesty of the CMCa is not improved at the logical levels and is below the honesty cutoff by $8 - 20\%$ for all cases.

The variation of the accuracy from level to level shows an interesting behavior. For both levels where the effect of the errors is linear (physical and uncorrected logical), in general the CMC channels and the "a" approximations are more accurate than the PC channels and the "w" approximations, respectively. This is seen by the magnitudes of the linear coefficients (Table 5): CMCa $<$ CMCw $<$ PCa $<$ PCw. At the logical level with perfect EC, this intuitively expected behavior is seen only for high damping strengths ($\gamma > 0.5$) (see Fig. 9). Surprisingly, for lower damping strengths, the most accurate approximation is given by the unconstrained PC, as can be observed by the suppresion of the second order terms in the accuracy (Table 5). This behavior is particularly pronounced at the logical level with perfect EC, where the second order terms for the PCa and ADC are practically indistinguishable.

5.3.2 Polarization along a non-Clifford Axis Channel (Pol$_\phi$C)

We perform an analogous analysis for our second target error channel: the polarization along a non-Clifford axis on the XY plane of the Bloch sphere. We select the axis forming an angle $\phi = \pi/8$ with respect to the X axis, as this is the angle for which the expanded error models perform the worst [52]. Once again, we have selected 20 initial states for the faultily corrected level and 80 points for all other levels. We have computed the trace distance between each one of them and the resulting final state after the Pol$_{\pi/8}$C and its approximations. The average distances are shown in the first row of Fig. 10 as a function of the noise strength, p. Likewise, we have computed the trace distance between each final state after the Pol$_{\pi/8}$C and each final state after every approximate channel. The average distances are presented in the second row of Fig. 10. The behavior in the limit of small noise strength ($p \to 0$) is summarized in Tables 6 and 7. As for the ADC, at the physical and uncorrected logical levels, the leading order is linear. At the corrected logical levels, the leading order is quadratic.

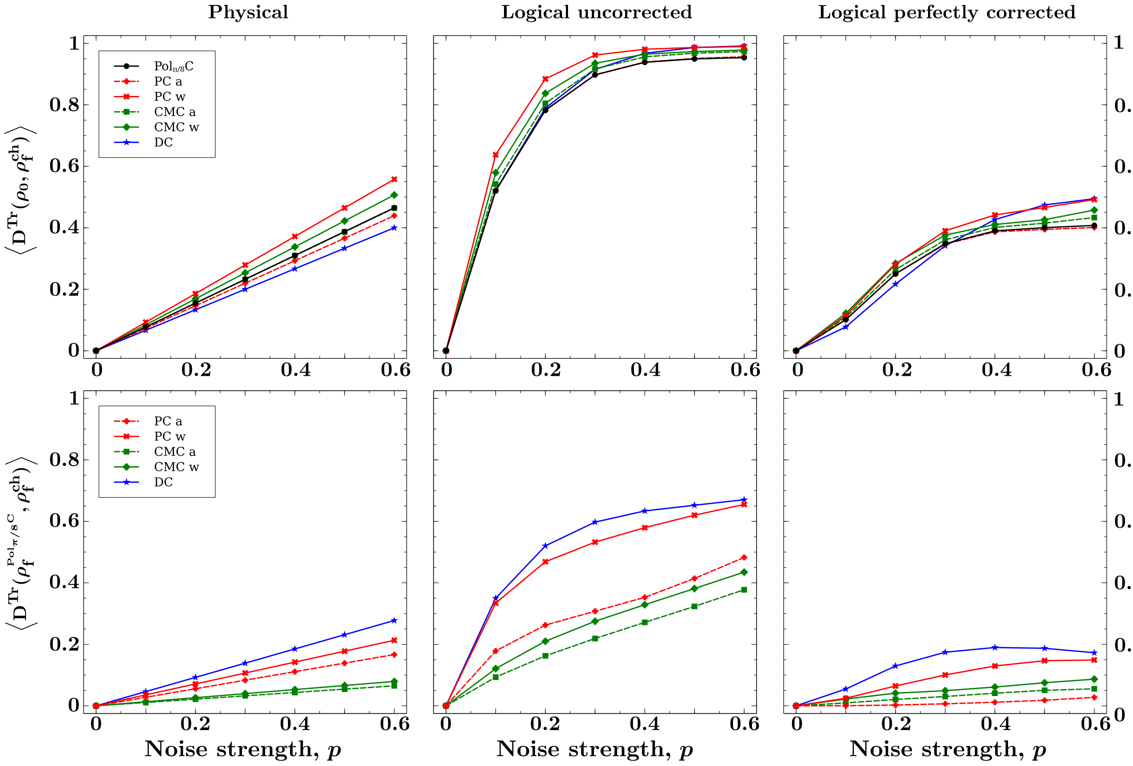

Figure 10: Distances used to assess the honesty (top row) and accuracy (botom row) of the approximate channels to the Pol$_{\pi/8}$C at various levels.

As observed on the ADC, the "w" approximations are honest at every level and for every noise strength. This holds in the average case and also for each initial state considered. Interestingly, the CMCa becomes honest on average and the PCa average distances are indistinguishable from honest.

Notice that, just like for the ADC, at the physical and uncorrected logical levels, the CMC channels and the "a" approximations are more accurate than their counterparts PC and "w", respectively. This can be seen by the magnitudes of the linear coefficients (Table 7): CMCa $<$ CMCw $<$ PCa $<$ PCw. At the physical level, and in the $p \to 0$ limit, the CMC approximations are ≈ 3 times more accurate than the PC approximations. At the

Table 6: Honesty of the approximate channels to the $\mathrm{Pol}_{\pi/8}$C in the limit of small noise strength. Standard deviations below 10^{-9} are not reported.

Channel	Physical	Logical uncorrected	Logical perfectly corrected	Logical faultily corrected
	$\mathrm{D}^{\mathrm{Tr}}/p$	$\mathrm{D}^{\mathrm{Tr}}/p$	$\mathrm{D}^{\mathrm{Tr}}/p^2$	$\mathrm{D}^{\mathrm{Tr}}/(10^3 p^2)$
$\mathrm{Pol}_{\pi/8}$C	0.78(24)	7.00	16.2(4.7)	2.22(72)
PCa	0.73(20)	7.00	16.2(4.7)	2.22(72)
PCw	0.93(18)	9.47	18.5(4.9)	2.71(77)
CMCa	0.77(22)	7.41	17.8(5.1)	2.67(83)
CMCw	0.84(23)	8.17	20.8(5.9)	3.12(96)
DC	0.667	7.00	11.0(1.4)	1.82(35)

Table 7: Accuracy of the approximate channels to the $\mathrm{Pol}_{\pi/8}$C in the limit of small noise strength. Standard deviations below 10^{-9} are not reported.

Channel	Physical	Logical uncorrected	Logical perfectly corrected	Logical faultily corrected
	$\mathrm{D}^{\mathrm{Tr}}/p$	$\mathrm{D}^{\mathrm{Tr}}/p$	$\mathrm{D}^{\mathrm{Tr}}/p^2$	$\mathrm{D}^{\mathrm{Tr}}/(10^2 p^2)$
PCa	0.278(82)	2.47	$6.1(7.7) \times 10^{-6}$	$5.8(1.6) \times 10^{-5}$
PCw	0.36(21)	4.95	3.8(1.1)	6.2(1.6)
CMCa	0.108(25)	1.09	1.76(41)	4.7(1.0)
CMCw	0.132(43)	1.37	4.6(1.1)	9.2(2.4)
DC	0.46(11)	4.67	9.2(1.3)	9.7(1.5)

corrected logical levels, the most accurate approximation is once again given by the PCa. Surprisingly, this behavior holds even up to high noise strengths ($p = 0.6$). In the low noise limit, and at the corrected logical levels, the second order terms are practically suppressed. In this limit, the PCa is on average more accurate than the CMC channels by a factor of 10^5.

5.3.3 High accuracy of the unconstrained PC

For both the ADC and the $\text{Pol}_{\pi/8}$C, the unconstrained PC results in approximations that are honest (or almost honest) and extremely accurate at the logical corrected levels. In the limit of small error, this is very evident by comparing the quadratic coefficients of the PCa to the other approximations (See Tables 5 and 7.). The high accuracy of the unconstrained PC in the context of EC has previously been observed. Geller and Zhou found very good agreement between the PCa or Pauli twirled approximation and two different realistic noise models when correcting a Bell state [43]. Likewise, Puzzuoli *et al.* observed great accuracy of the PCa when correcting a Choi state encoded in the [[5,1,3]] code [83]. As clearly explained in Ref. [83], after (perfect) EC, the process matrix elements corresponding to Pauli error strings that result in different syndromes become zero. Intuitively, we can say that the "non-Pauli" advantage of the expanded approximations at the physical level gets "washed away" after EC.

5.4 Level-1 pseudo-thresholds

We perform the simulations of two different scenarios: (1) one with perfect EC, which results in a relatively high code-capacity pseudo-threshold and (2) one with faulty EC, which results in a more realistic circuit-based pseudo-threshold. Apart from the average pseudo-threshold, we also calculate the root mean square difference (RMS) between the pseudo-thresholds given by the target non-Clifford channel and the ones predicted by each approximate channel:

$$\text{RMS} = \sqrt{\langle(\text{p}_{th}^{\text{channel}} - \text{p}_{th}^{\text{approx}})^2\rangle}. \tag{48}$$

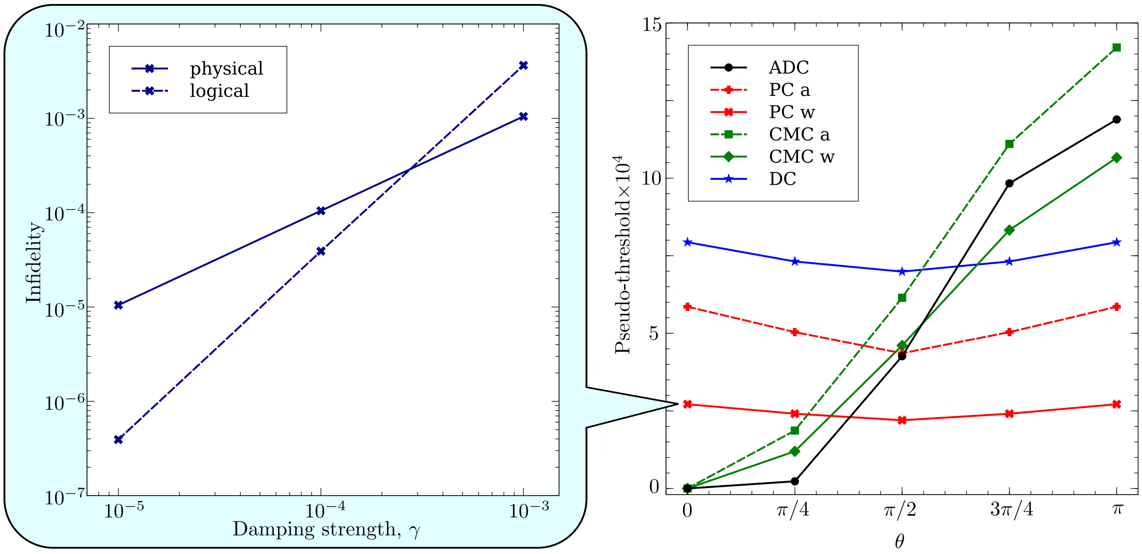

Figure 11: Variation of the circuit-based level-1 pseudo-threshold as a function of the angle θ for the ADC and its approximations. Each point corresponds to the pseudo-threshold averaged over different values of the angle ϕ for the same angle θ. The initial state is given by $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$. For the ADC and its expanded approximations, the pseudo-threshold depends strongly on the initial state. The zoom-in figure on the left shows how the pseudo-threshold is computed for a particular point on the Bloch sphere, namely by finding the first intersection between the physical and logical error $(1 - F)$.

The RMS quantifies the accuracy of each approximate channel to estimate the pseudo-threshold of the target channel. We calculate the RMS because comparing only the average values does not account for any cancellation of errors. A certain approximate channel can do a very poor job at approximating the pseudo-threshold for every initial state, but result in an average that is close to the target's average.

The results for the ADC and the Pol$_{\pi/8}$C are summarized in Tables 8 and 9, respectively. For the ADC, the pseudo-thresholds are expressed in terms of the damping strength, γ, while for the Pol$_{\pi/8}$C, they are expressed in terms of the noise strength, p. In both cases, the standard deviation of the pseudo-thresholds is included inside parentheses. Notice that the code-capacity pseudo-thresholds are about 3 orders of magnitude higher than the circuit-based ones. The latter ones are on the range expected for the Steane code [92]. Although the code-capacity pseudo-thresholds are unrealistically high, they show similar trends with respect to their circuit-based counterparts.

Table 8: Thresholds for the Steane code under the ADC and its Pauli and expanded approximations. ADC/PCa uses ADC at the physical level and PCa at the logical level.

Channel	Code capacity		Circuit-based	
	$\langle\gamma_{th}\rangle$	RMS	$\langle\gamma_{th}\rangle \times 10^4$	RMS $\times 10^4$
ADC	$0.18(17)$	–	$4.8(4.2)$	–
PCa	$0.132(38)$	0.171	$4.8(1.4)$	3.91
PCw	$0.061(43)$	0.204	$2.36(60)$	4.69
CMCa	$0.19(17)$	0.0498	$6.4(4.2)$	1.67
CMCw	$0.15(14)$	0.0644	$4.8(3.1)$	1.12
DC	$0.162(22)$	0.165	$7.2(1.4)$	4.60
ADC/PCa	$0.30(37)$	0.255	$4.9(4.2)$	0.101

Table 9: Thresholds for the Steane code under the Pol$_{\pi/8}$C and its Pauli and expanded approximations. Pol$_{\pi/8}$C/PCa uses Pol$_{\pi/8}$C at the physical level and PCa at the logical level.

Channel	Code capacity		Circuit-based	
	$\langle p_{th}\rangle$	RMS	$\langle p_{th}\rangle \times 10^4$	RMS $\times 10^4$
Pol$_{\pi/8}$C	$0.14(24)$	–	$3.5(1.5)$	–
PCa	$0.086(74)$	0.238	$3.10(26)$	1.53
PCw	$0.078(16)$	0.237	$3.46(35)$	1.48
CMCa	$0.11(21)$	0.112	$3.09(85)$	0.816
CMCw	$0.09(14)$	0.169	$2.91(76)$	0.991
DC	$0.083(12)$	0.244	$3.92(64)$	1.60
Pol$_{\pi/8}$C/PCa	$0.14(25)$	0.0255	$3.5(1.5)$	1.19×10^{-3}

As can be seen from Tables 8 and 9, in general, the standard deviations of the pseudo-thresholds about its average values are high, especially for the target non-Clifford channels and its expanded approximations. Despite the fact that we consider relatively few initial states (80 and 20 for the code-capacity and circuit-based cases, respectively), the high variances are not a consequence of the small sample sizes. In fact, when reducing the sample size to only 6 initial points (the 6 Pauli states), the variances increase only slightly, and for some channels do not increase at all. Instead, the high variances are due to the extreme sensitivity of the pseudo-threshold to the initial state. As an illustrative example, consider Fig. 11. The plot to the right shows how the circuit-based pseudo-threshold of the ADC and its approximations varies as the initial state changes from $|0\rangle$ to $|1\rangle$ as a function of the angle θ. Notice, that the pseudo-threshold for the ADC and its expanded approximations is particularly sensitive to the initial state, ranging from 0, when the initial state is the fixed point of the ADC, to $\sim 10^{-3}$.

It is interesting that the Pauli channels always result in pseudo-thresholds that are lower than the real ones. This trend has also been observed by Tomita and Svore on the surface code [99] and suggests that anisotropic Pauli channel approximations to realistic noise models are pessimistic. The CMCw approximations also result in lower pseudo-thresholds. This is in contrast to the isotropic Pauli channel approximation (DC) that yields optimistic pseudo-thresholds.

The CMCs give more accurate pseudo-threshold estimates than the PCs, as can be seen by comparing their RMS values. Although we might expect the “a” channels to result in better approximations than the “w” channels, in general this is not the case. The most important variation is between the CMCs and the PCs. In general, however, the “w” channels result in lower pseudo-thresholds than the “a” channels, which implies that honest approximations at the physical level do a good job at giving conservative estimates of the threshold. Finally, we notice that the circuit-based pseudo-thresholds are quite comparable yielding pseudo-thresholds within a factor of two for all of the error models. The DC model representing isotropic depolarizing noise yields the worst results.

In the previous section, we noticed that the PCa, one of the simplest approximations at the physical level, and one that is not even honest, results in very accurate and practically honest approximations at the corrected logical levels. In the context of our level-1 pseudo-threshold estimation, this result suggests that we can take a different strategy. Instead of using the approximate channel at both the physical and logical level to calculate the pseudo-threshold, we can use the target channel at the physical level and the PCa approximation at the logical level. More generally, we can simulate the realistic noise model in an exact way whenever it is feasible, and in the encoded cases just use the PCa. If we take this approach, we obtain more accurate state by state pseudo-thresholds for the circuit-based case as seen in Tables 8 and 9.

CHAPTER VI

APPROXIMATIONS TO COHERENT ERROR CHANNELS

As seen in Chapter 5, the PCa provides an extremely accurate (and almost honest) approximation to the ADC and the Pol$_\phi$C at the logical corrected levels. However, it has been suggested that this behavior only occurs when the target channel is non-unitary or incoherent [83]. The two realistic channels that we have focused on in the previous chapters are of this kind. Therefore, we are interested in studying the honesty and accuracy of the Pauli and expanded channels when approximating a coherent channel. Just like in Chapter 5, we are also interested in estimating the level-1 pseudo-threshold. Finally, we want to understand why the PCa is so good at the logical corrected levels for incoherent channels and how it performs for unitary errors.

As a model unitary error we select a rotation about the Bloch sphere's Z axis by an angle θ:

$$\text{RZC} = \exp(-i\theta Z/2) = \cos(\theta/2)I - i\sin(\theta/2)Z \tag{49}$$

This channel can arise as an over-rotation when applying a rotation during a quantum computation, due to a miscalibration of the laser intensity or an extra long pulse duration. The angle θ parametrizes the error strength.

6.1 Analysis in the limit of low error rates

Our analysis is based on the effective 1-qubit process matrix for the whole circuit, including the encoding, occurrence of error, syndrome measurement, error correction, and decoding. This strategy is motivated by the observation that in our simulation scheme the final state of the quantum circuit is always completely localized on the logical codespace, so that the overall circuit can be compactly represented by a 1-qubit process matrix.

For the perfect EC, after the stabilizer measurement and correction, it is evident that the final state will live in the logical codespace. In general, this will not be the case when the EC is faulty, since errors during the measurement of the stabilizers will cause the logical state to not be projected perfectly onto the code subspaces. However, in our previous analysis [51], after faulty EC we always perform one round of perfect EC, to account exclusively for uncorrectable errors. This has the effect of completely projecting the final state onto the codespace.

We are particularly interested in the low error rate limit, where it is appropriate to Taylor-expand each entry in the process matrix in terms of powers of the error rate. This helps us visualize in a very clear way which terms are more important in determining the relevant characteristics of a given error channel. As an example, consider a 1-qubit coherent error channel consisting of a rotation about the X axis by an angle θ:

$$R_X(\theta) = \exp(-i\theta X/2) \tag{50}$$

In the normalized Pauli basis, the process matrix for this channel is:

$$\begin{pmatrix} 2\cos^2(\theta/2) & i\sin(\theta) & 0 & 0 \\ -i\sin(\theta) & 2\sin^2(\theta/2) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In the small error limit ($\theta \to 0$), this becomes:

$$\begin{pmatrix} 2 - \theta^2/2 + \mathcal{O}(\theta^4) & i\theta + \mathcal{O}(\theta^3) & 0 & 0 \\ -i\theta + \mathcal{O}(\theta^3) & \theta^2/2 + \mathcal{O}(\theta^4) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The PCa approximation to this channel is given by the diagonal entries of its process matrix:

$$\begin{pmatrix} 2\cos^2(\theta/2) & 0 & 0 & 0 \\ 0 & 2\sin^2(\theta/2) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

which corresponds to a channel where the qubit is flipped with a probability $p_x = \sin^2(\theta/2)$:

$$\mathrm{PC}_{\mathrm{RXC}} = \begin{cases} \sqrt{1-p_x}\, I \\ \sqrt{p_x}\, X \end{cases} \tag{51}$$

For illustrative purposes, imagine the situation where we use the QBFC introduced in Chapter 3. We perfectly encode our qubit, then 3 independent error instances happen, 1 on each qubit, and finally we measure the stabilizer generators, correct, and decode. If the individual errors correspond to flips with probability p_x, the effective channel for the whole circuit is given by:

$$\mathrm{PC}^l = \begin{cases} \sqrt{(1-p_x)^3 + 3(1-p_x)^2 p_x}\, I \\ \sqrt{3(1-p_x)p_x^2 + p_x^3}\, X \end{cases} \tag{52}$$

This channel is still a probabilistic application of an X operator. The first Kraus operator corresponds to the situation where either no flip or 1 flip occurred. The second Kraus operator accounts for the case where 2 or 3 flips occurred, thus causing a logical X error. If $p_x = \sin^2(\theta/2)$, this channel's reduced process matrix is:

$$\begin{pmatrix} 2\cos^4(\theta/2)\left(1+2\sin^2(\theta/2)\right) & 0 \\ 0 & 2\sin^4(\theta/2)\left(1+2\cos^2(\theta/2)\right) \end{pmatrix}$$

Here we have only focused on the first 2 rows and columns of the 1-qubit process matrix. (All the other entries are 0.) On the other hand, if the 3 independent errors are coherent rotations about the X axis by an angle θ, the effective reduced process matrix for the whole circuit is:

$$\begin{pmatrix} 2\cos^4(\theta/2)\left(1+2\sin^2(\theta/2)\right) & -i\left[(1-\sqrt{3})/8\right]\sin^3(\theta) \\ i\left[(1-\sqrt{3})/8\right]\sin^3(\theta) & 2\sin^4(\theta/2)\left(1+2\cos^2(\theta/2)\right) \end{pmatrix}$$

Interestingly, for a circuit where the errors are the best PCa approximation to the coherent channels, the effective process matrix for the whole circuit still matches the diagonal entries perfectly. However, it is completely unable to match the off-diagonal entries. In the limit of small error, this becomes:

$$\begin{pmatrix} 2-(3/8)\theta^4+\mathcal{O}(\theta^6) & -i\left[(1-\sqrt{3})/8\right]\theta^3+\mathcal{O}(\theta^5) \\ i\left[(1-\sqrt{3})/8\right]\theta^3+\mathcal{O}(\theta^5) & (3/8)\theta^4+\mathcal{O}(\theta^6) \end{pmatrix}$$

At the physical level, the process matrix for the $R_X(\theta)$ channel has diagonal entries proportional to θ^2 and off-diagonal entries proportional to θ. At the logical level with perfect EC, the leading orders get supressed and the effective process matrix now has diagonal entries proportional to θ^4 and off-diagonal ones proportional to θ^3. In this case, the PCa underestimates the magnitude of the error by 1 power of θ both at the physical and logical levels.

6.2 Effective process matrices for the ADC and the RZC

We have followed the same procedure explained in Chapter 5 to compute the final states after error correction with the Steane code. Because of the size of the density matrices and the time it takes to cover all the possible syndrome branches, we are unable to obtain a symbolic expression for the effective 1-qubit process matrices. Instead, we use quantum process tomography to reconstruct the numerical process matrix for various error strengths and subsequently fit each entry to a polynomial to determine the leading order and its coefficient.

6.2.1 ADC

Table 10 summarizes the results for the ADC and its approximations at the physical level. The values for error magnitude and inaccuracy are the same ones presented in Chapter 5 and are included just for visualization purposes. At the physical level, the entries of the ADC process matrix are all linear in γ except for the χ_{zz} term, which is quadratic. Consequently, the error magnitude is linear. The PCa matches the diagonal entries perfectly. In order to not underestimate the magnitude of the error, the PCw results in a pessimistic estimate with all diagonal entries being linear. Both of these channels only have access to the diagonal entries. In contrast, the CMC channels have access to off-diagonal entries in the process matrix. This gives them the ability to result in more accurate approximations, as illustrated by their lower inaccuracy values. However, in this case, they give a linear estimate to the ZZ entry.

Tables 11 and 12 summarize the results for the ADC and its approximations at the logical levels, perfectly and faultily corrected, respectively. Several interesting changes occur after error correction. First of all, notice that all the linear terms have disappeared, which confirms that the Steane code's correcting procedure is indeed successful in suppressing single errors. In the ADC, the diagonal entries that were linear at the physical level become quadratic. The quadratic entry becomes cubic at the perfectly corrected level, but remains quadratic at the faultily corrected level. However, the linear off-diagonal entries become cubic. This illustrates why at low error rates the PCa is more accurate than the expanded approximations. The PC is more constrained; it does not have access to the off-diagonal entries, which results in inaccuracy at the physical level. However, at the logical level the errors on the off-diagonal entries are suppressed by an extra order of magnitude. The real error is given by the diagonal terms, which the PCa estimates much better than the CMCs, so the advantage of the latter channels is lost. Moreover, for practical purposes, the PCa is honest, since its dishonesty comes at third order for the perfect correction and at a low second order for the faulty correction.

Table 10: ADC and approximations at physical level. Only the leading orders are shown. Empty entries are equal to 0.

Channel	Process matrix	Error magnitude	Inaccuracy
ADC	$\begin{pmatrix} 2-\mathcal{O}(\gamma) & & & \gamma/2 \\ & \gamma/2 & -i\gamma/2 & \\ & i\gamma/2 & \gamma/2 & \\ \gamma/2 & & & \gamma^2/8 \end{pmatrix}$	$0.55(27)\gamma$	
PCa	$\begin{pmatrix} 2-\mathcal{O}(\gamma) & & & \\ & \gamma/2 & & \\ & & \gamma/2 & \\ & & & \gamma^2/8 \end{pmatrix}$	$0.347(79)\gamma$	0.500γ
PCw	$\begin{pmatrix} 2-\mathcal{O}(\gamma) & & & \\ & 1.047\gamma & & \\ & & 1.047\gamma & \\ & & & 0.2915\gamma \end{pmatrix}$	$0.81(12)\gamma$	$0.63(26)\gamma$
CMCa	$\begin{pmatrix} 2-\mathcal{O}(\gamma) & & & 3\gamma/8 \\ & 3\gamma/8 & -i3\gamma/8 & \\ & i3\gamma/8 & 3\gamma/8 & \\ 3\gamma/8 & & & 3\gamma/8 \end{pmatrix}$	$0.50(18)\gamma$	$0.166(60)\gamma$
CMCw	$\begin{pmatrix} 2-\mathcal{O}(\gamma) & & & \gamma/2 \\ & \gamma/2 & -i\gamma/2 & \\ & i\gamma/2 & \gamma/2 & \\ \gamma/2 & & & \gamma/2 \end{pmatrix}$	$0.66(24)\gamma$	$0.194(60)\gamma$

Table 11: ADC and approximations at logical level with perfect EC. Only the leading orders are shown. Empty entries are equal to 0.

Channel	Process matrix	Error magnitude	Inaccuracy
ADC	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & -1.780\gamma^3 \\ & 7.875\gamma^2 & i1.750\gamma^3 & \\ & -i1.750\gamma^3 & 2.625\gamma^2 & \\ -1.780\gamma^3 & & & 3.937\gamma^3 \end{pmatrix}$	$3.76(96)\gamma^2$	
PCa	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & \\ & 7.875\gamma^2 & & \\ & & 2.625\gamma^2 & \\ & & & 3.939\gamma^3 \end{pmatrix}$	$3.76(96)\gamma^2$	$7(12) \times 10^{-6}\gamma^2$
PCw	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & \\ & 34.54\gamma^2 & & \\ & & 11.51\gamma^2 & \\ & & & 7.302\gamma^2 \end{pmatrix}$	$18.5(3.5)\gamma^2$	$14.8(2.6)\gamma^2$
CMCa	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & -0.7383\gamma^3 \\ & 4.430\gamma^2 & i0.7383\gamma^3 & \\ & -i0.7383\gamma^3 & 1.477\gamma^2 & \\ -0.7383\gamma^3 & & & 4.430\gamma^2 \end{pmatrix}$	$3.48(45)\gamma^2$	$1.61(44)\gamma^2$
CMCw	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & -1.750\gamma^3 \\ & 7.875\gamma^2 & i1.750\gamma^3 & \\ & -i1.750\gamma^3 & 2.625\gamma^2 & \\ -1.750\gamma^3 & & & 7.875\gamma^2 \end{pmatrix}$	$6.19(80)\gamma^2$	$3.05(94)\gamma^2$

Table 12: ADC and approximations at logical level with faulty EC. Only the leading orders are shown. Empty entries are equal to 0.

Channel	Process matrix	Error magnitude$/10^2$	Inaccuracy$/10^2$
ADC	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & -212.0\gamma^3 \\ & 1585\gamma^2 & i212.0\gamma^3 & \\ & -i212.0\gamma^3 & 180.0\gamma^2 & \\ -212.0\gamma^3 & & & 495.0\gamma^2 \end{pmatrix}$	$8.0(1.8)\gamma^2$	
PCa	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & \\ & 1567\gamma^2 & & \\ & & 180.0\gamma^2 & \\ & & & 491.0\gamma^2 \end{pmatrix}$	$7.8(1.8)\gamma^2$	$0.123(28)\gamma^2$
PCw	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & \\ & 7081\gamma^2 & & \\ & & 790.0\gamma^2 & \\ & & & 3023\gamma^2 \end{pmatrix}$	$37.7(8.0)\gamma^2$	$29.8(6.2)\gamma^2$
CMCa	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & -89.42\gamma^3 \\ & 988.3\gamma^2 & i89.42\gamma^3 & \\ & -i89.42\gamma^3 & 101.3\gamma^2 & \\ -89.42\gamma^3 & & & 770.6\gamma^2 \end{pmatrix}$	$6.3(1.2)\gamma^2$	$2.15(74)\gamma^2$
CMCw	$\begin{pmatrix} 2-\mathcal{O}(\gamma^2) & & & -211.9\gamma^3 \\ & 1757\gamma^2 & i211.9\gamma^3 & \\ & -i211.9\gamma^3 & 180.1\gamma^2 & \\ -211.9\gamma^3 & & & 1370\gamma^2 \end{pmatrix}$	$11.3(2.2)\gamma^2$	$3.7(1.1)\gamma^2$

6.2.2 RZC

As for the ADC and the Pol$_\phi$C, we have selected 20 initial states for the faultily corrected level and 80 points for all other levels. We have computed the trace distance between each one of them and the resulting final state after the RZC and its approximations. The average distances are shown in the first row of Fig. 12 as a function of the rotation angle, θ. Likewise, we have computed the trace distance between each final state after the RZC and each final state after every approximate channel. The average distances are presented in the second row of Fig. 12. The behavior in the limit of small noise strength ($\theta \to 0$) is summarized in Tables 13, 14, and 15.

The RZC has very different characteristics from the ADC. Table 13 summarizes the results for the physical level. The RZC has a quadratic term along its diagonal and a linear one on off-diagonal entries. The PCa matches the diagonal entries perfectly, but this means that it predicts a quadratic error, when in reality is linear. It underestimates the real error by

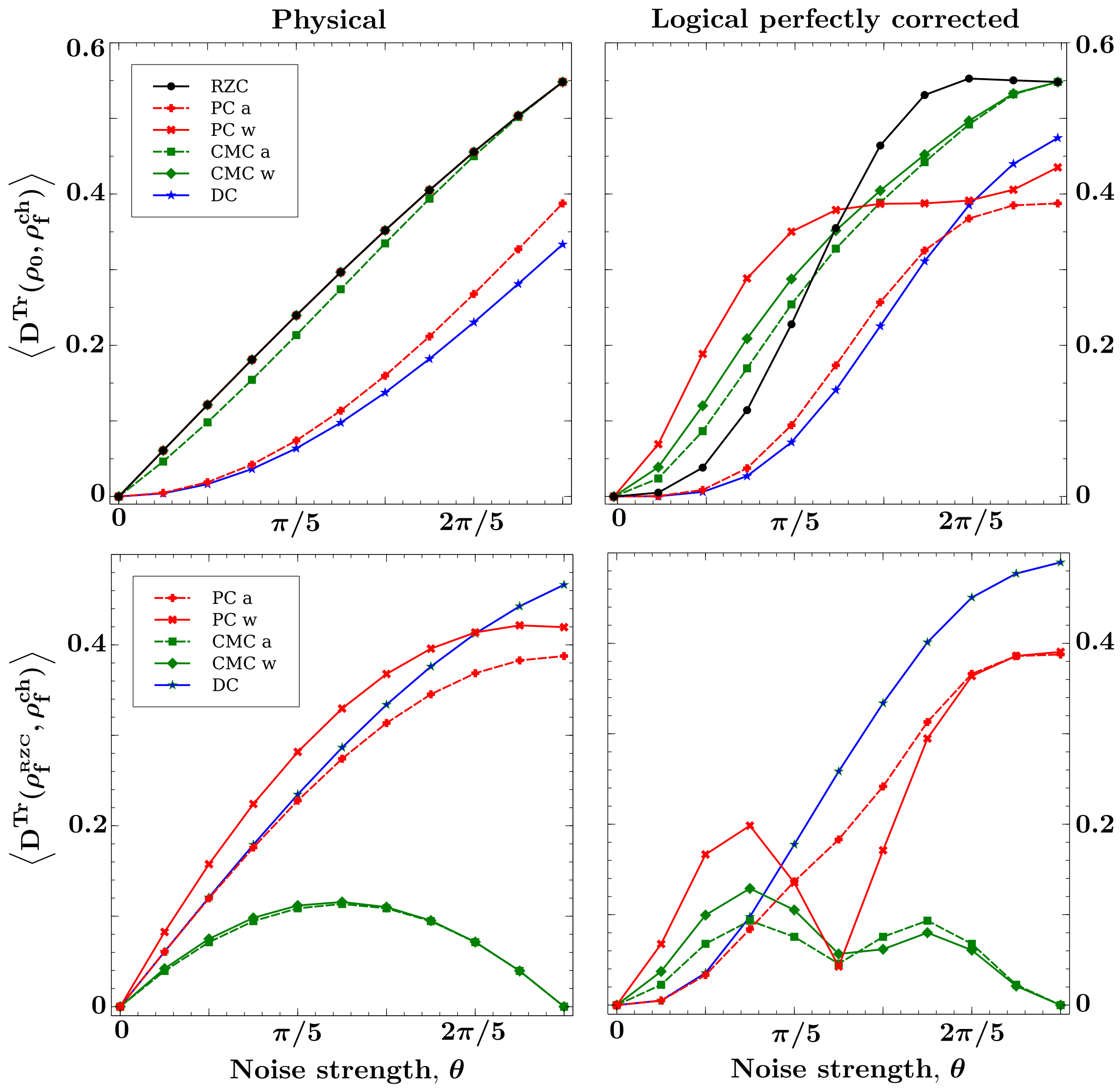

Figure 12: Distances used to assess the honesty (top row) and accuracy (bottom row) of the approximate channels to the RZC at various levels.

one order of magnitude, making it very dishonest. The CMCa is dishonest too, but not by one order. As observed on the incoherent channels, at the physical level both constrained approximations are honest by construction. Interestingly, their error magnitude is exactly the same as the RZC error magnitude: they saturate the honesty limit.

Just like for the other channels, at the physical level, the CMC channels and the "a" approximations are more accurate than their counterparts PC and "w", respectively. This can be seen by the magnitudes of the linear coefficients (Table 13): CMCa $<$ CMCw $<$

PCa < PCw. At the corrected logical levels, the most accurate approximation is once again given by the PCa. However, in contrast to the approximations to the incoherent channels, the PCa is severely dishonest. At these levels, the diagonal quadratic entries becomes quartic. The off-diagonal linear entries become cubic. The PCa is more accurate than the other channels, but only because it severely underestimates the magnitude of the error. It is not a good approximation.

Table 13: RZC and approximations at physical level. Only the leading orders are shown. Empty entries are equal to 0.

Channel	Process matrix	Error magnitude	Inaccuracy
RZC	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & & & i\theta \\ & & & \\ & & & \\ -i\theta & & & \theta^2/2 \end{pmatrix}$	$0.38(14)\theta$	
PCa	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & & & \\ & & & \\ & & & \\ & & & \theta^2/2 \end{pmatrix}$	$0.188(72)\theta^2$	$0.38(14)\theta$
PCw	$\begin{pmatrix} 2-\mathcal{O}(\theta) & & & \\ & & & \\ & & & \\ & & & \theta \end{pmatrix}$	$0.38(14)\theta$	$0.53(20)\theta$
CMCa	$\begin{pmatrix} 2-\mathcal{O}(\theta) & & & i\theta/2 \\ & & & \\ & & & \\ -i\theta/2 & & & \theta/2 \end{pmatrix}$	$0.27(10)\theta$	$0.27(10)\theta$
CMCw	$\begin{pmatrix} 2-\mathcal{O}(\theta) & & & i\theta/\sqrt{2} \\ & & & \\ & & & \\ -i\theta/\sqrt{2} & & & \theta/\sqrt{2} \end{pmatrix}$	$0.38(14)\theta$	$0.29(11)\theta$

6.3 Level-1 pseudo-thresholds for the RZC and its approximations

Table 16 summarizes the pseudo-threshold results for the RZC and its approximations. The more realistic circuit-based thresholds show interesting trends. The PCa, which results in

Table 14: RZC and approximations at logical level with perfect EC. Only the leading orders are shown. Empty entries are equal to 0.

Channel	Process matrix	Error magnitude	Inaccuracy
RZC	$\begin{pmatrix} 2-\mathcal{O}(\theta^4) & & & i3.500\,\theta^3 \\ & & & \\ & & & \\ -i3.500\,\theta^3 & & & 7.881\,\theta^4 \end{pmatrix}$	$1.32(50)\theta^3$	
PCa	$\begin{pmatrix} 2-\mathcal{O}(\theta^4) & & & \\ & & & \\ & & & \\ & & & 2.622\,\theta^4 \end{pmatrix}$	$0.99(38)\theta^4$	$1.32(50)\theta^3$
PCw	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & & & \\ & & & \\ & & & \\ & & & 10.50\,\theta^2 \end{pmatrix}$	$4.0(1.5)\theta^2$	$4.0(1.5)\theta^2$
CMCa	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & & & i0.4375\,\theta^3 \\ & & & \\ & & & \\ -i0.4375\,\theta^3 & & & 2.625\,\theta^2 \end{pmatrix}$	$0.99(38)\theta^2$	$0.99(38)\theta^2$
CMCw	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & & & i1.237\,\theta^3 \\ & & & \\ & & & \\ -i1.237\,\theta^3 & & & 5.250\,\theta^2 \end{pmatrix}$	$1.98(75)\theta^2$	$1.98(75)\theta^2$

Table 15: RZC and approximations at logical level with faulty EC. Only the leading orders are shown. Empty entries are equal to 0.

Channel	Process matrix	Error magnitude$/10^2$	Inaccuracy$/10^2$
RZC	$\begin{pmatrix} 2-\mathcal{O}(\theta^4) & i557.7\,\theta^3 \\ -i557.7\,\theta^3 & 7859\,\theta^4 \end{pmatrix}$	$2.07(79)\theta^3$	
PCa	$\begin{pmatrix} 2-\mathcal{O}(\theta^4) & \\ & 206.0\,\theta^4 \end{pmatrix}$	$0.78(30)\theta^4$	$2.07(79)\theta^3$
PCw	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & \\ & 826.0\,\theta^2 \end{pmatrix}$	$3.2(1.2)\theta^2$	$3.2(1.2)\theta^2$
CMCa	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & i69.90\,\theta^3 \\ -i69.90\,\theta^3 & 206.1\,\theta^2 \end{pmatrix}$	$0.80(31)\theta^2$	$0.80(31)\theta^2$
CMCw	$\begin{pmatrix} 2-\mathcal{O}(\theta^2) & i198.2\,\theta^3 \\ -i198.2\,\theta^3 & 412.4\,\theta^2 \end{pmatrix}$	$1.61(61)\theta^2$	$1.61(61)\theta^2$

a severely dishonest approximation, especially at the faultily corrected level, generates a very large threshold value. This is quite dangerous for fault tolerant purposes. On the other hand, all the other approximations generate pessimistic threshold values, due to the pessimistic error magnitude estimate. This is better for fault tolerant purposes.

The threshold is given by the intersection between 2 curves: the error magnitude at the physical level and the error magnitude at the logical level. With this in mind, there are different cases for the threshold estimation by approximate channels. A lower bound to the exact threshold will be given by a channel that is exact or dishonest at the physical level and honest at the logical level. This is the case for the CMCa approximation to the RZC. As an unconstrained channel, it is dishonest at the physical level. However, since it predicts the error magnitude of the RZC to be quadratic, it is honest at the logical levels. In the next section, it will become clear that this will hold for a rotation about any axis, not just Z. Indeed, when approximating a coherent channel, the CMCa will result in a lower bound to the exact threshold.

On the other hand, an upper bound to the exact threshold will be given by a channel that is exact or honest at the physical level and dishonest at the logical levels. This is the case for the PCa approximation to the RZC: although it is dishonest at the physical level, the dishonesty is more severe at the logical faultily corrected level. Therefore, in general the PCa will give pessimistic threshold estimates for incoherent channels and optimistic ones for coherent channels.

In contrast to the cases analyzed in Chapter 5, modeling the error like the RZC at the physical level and like the PCa at the logical level does not result in a more accurate estimate, since the fidelity of the RZC and the PCa at the physical level is exactly the same. In general, what can be done is model the noise exactly at the physical level and then use an honest channel at the logical level (any "w" approximation) to obtain a lower bound. To obtain an upper bound, we can use the PCa, which is always dishonest at the logical levels.

Table 16: Thresholds for the Steane code under the RZC and its Pauli and expanded approximations. RZC/PCa uses RZC at the physical level and PCa at the logical level.

Channel	Code capacity		Circuit-based	
	$\langle\theta_{th}\rangle$	RMS	$\langle\theta_{th}\rangle \times 10^3$	RMS $\times 10^3$
RZC	0.6552	–	$7.773(4)$	–
PCa	0.5140	0.1412	$49.582(2)$	41.8
PCw	0.1293	0.5259	$1.2166(1)$	6.56
CMCa	0.3873	0.2679	$2.400(1)$	5.37
CMCw	0.3208	0.3344	$4.5329(7)$	3.24
RZC/PCa	0.5140	0.1412	$49.582(1)$	41.8

6.4 Dishonesty of the PCa for coherent errors

In this last section, we want to present a short argument as to why the PCa will always be dishonest by one order of magnitude when approximating a coherent error. Up to a global phase, any unitary 1-qubit transformation can be represented as:

$$\exp(i\theta\hat{n}\cdot\vec{\sigma}/2) = \cos(\theta/2)I - i\sin(\theta/2)\left(n_xX + n_yY + n_zZ\right) \tag{53}$$

The process matrix associated to this transformation is:

$$\begin{pmatrix} 2\cos^2(\theta/2) & in_x\sin(\theta) & in_y\sin(\theta) & in_z\sin(\theta) \\ -in_x\sin(\theta) & 2n_x^2\sin^2(\theta/2) & 2n_xn_y\sin^2(\theta/2) & 2n_xn_z\sin^2(\theta/2) \\ -in_y\sin(\theta) & 2n_xn_y\sin^2(\theta/2) & 2n_y^2\sin^2(\theta/2) & 2n_yn_z\sin^2(\theta/2) \\ -in_z\sin(\theta) & 2n_xn_z\sin^2(\theta/2) & 2n_yn_z\sin^2(\theta/2) & 2n_z^2\sin^2(\theta/2) \end{pmatrix}$$

It becomes clear that the off-diagonal entries of the first row and column are linear in θ, whereas all the other are quadratic in θ. The PCa will always be an extremely dishonest approximation to a channel of this form. Furthermore, after EC, the first term in the polynomial expansion gets supressed. Because the sine function only has odd powers, the linear entries become cubic. The quadratic entries become quartic. The same argument is applicable for multi-qubit coherent channels.

CHAPTER VII

PYTHON TOOLS FOR THE GENERATION AND SIMULATION OF FAULT TOLERANT CIRCUITS

We have worked on multiple collaborative projects aimed at estimating the resources that will be needed to perform large-scale fault-tolerant quantum algorithms on several quantum architectures, our group has developed a set of Python tools to realistically simulate quantum circuits. A diagram of the toolset developed for the MUSIQC program is presented in Figure 13. MUSIQC stands for Modular Universal Scalable Ion-Trap Quantum Computer. The aim of the program is to investigate new hierarchical designs to build a scalable quantum computer based on trapped ions [74] . In this proposed quantum computer architecture, the building blocks of the computer will be ion traps holding ultra-cold ions. Two hyperfine states are chosen to store the quantum information. Within a trap, the states of the qubits can be coupled through the collective motional modes of the ions [25, 90]. Qubits in different traps can be entangled via photonic interconnections [35, 55].

The Performance Simulator of Fault-Tolerant Quantum Circuits is composed of various modules, represented as white rectangles in Figure 13. Each module can also be used as a standalone tool. The whole toolset works in the following way:

1. In the *algorithms* module, a particular quantum algorithm is compiled in the circuit model as a sequence of quantum gates. At this point, the circuit is unencoded, in the sense that no QECC has been used to encode the physical gates into fault-tolerant logical gates. So far, the only quantum algorithm hard-coded is the non-recursive Bernstein-Vazirani algorithm [17]. Given the circuit building tools we have implemented, it is straight-forward to include more quantum algorithms.

2. The *fault tolerant* module generates fault-tolerant logical gates specific to a given

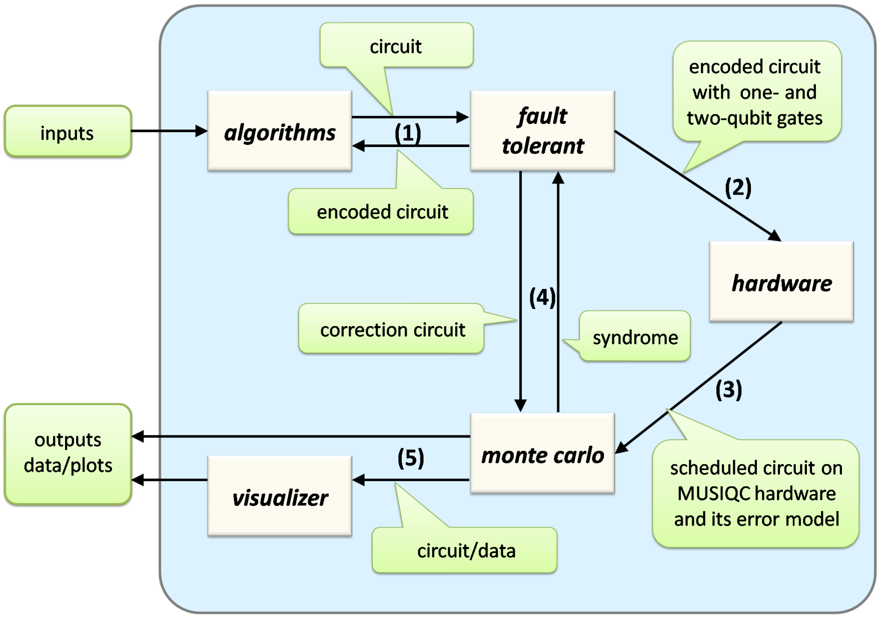

Figure 13: Diagrammatic flow within the Performance Simulator of Fault-Tolerant Quantum Circuits.

QECC and correction scheme. It takes an unencoded quantum circuit, converts each gate in the circuit to its logical version and adds an error correction step after it. So far, only the Steane [[7,1,3]] [91] and the Bacon-Shor [[9,1,3]] [9, 14] codes have been implemented, but the capabilities exist to add more QECCs. There are three implemented options for the error correction step: Shor's, Steane's, and Knill's. The user can choose any of the three.

3. The *hardware* module takes in an architecture-agnostic quantum circuit and translates it into a set of operations to be performed in a realistic quantum architecture. The *hardware* module also assigns error rates and time durations to each gate depending on the way it is to be implemented in that particular quantum computer. For the MUSIQC program, the quantum computer of choice is one based on ion traps. However, the high modularity of the tool allows the analysis of a circuit mapped to any quantum computer architecture. The *hardware* module was the only tool developed outside of our group [7, 101].

4. The *Monte Carlo* module is where the error analysis of the quantum circuit happens. Although its name suggests that the error analysis is based exclusively on Monte Carlo methods, this is really only one of three alternatives: the exact simulator, the fault-path tracer, and the Monte Carlo sampler. These methods are explained in the next section. The error analysis tool can obtain circuits from the *hardware* module or directly from the *fault tolerant* module depending on whether the user wants to study scheduled or architecture-agnostic circuits, respectively. Finally, notice that during the error analysis, the *monte carlo* module communicates with the *fault tolerant* one to obtain information on how to correct the detected errors.

5. The *visualizer* module allows the user to obtain a graphical representation of a particular quantum circuit instance. The circuit is typically displayed as an html file that the user can open in any web browser. Additionally, there is a hierarchical structure that allows the user to click on a particular logical gate to visualize the underlying circuit. This module is mostly used for debugging purposes.

7.1 The error analyzer

As mentioned previously, the error analysis can be performed with 3 different techniques: exact simulation, fault-path tracing, and Monte Carlo sampling.

7.1.1 Exact simulation

The exact simulator is based on the exact full density matrix simulation of the whole quantum circuit. The state of the system is evolved in time by explicitly applying every quantum gate in the circuit. The state of the system can be pure or mixed and the tool can apply unitary operations and non-unitary channels, which typically represent errors. In contrast to the other error analysis methods, the exact simulator is not restricted to a particular set of gates; it can apply any quantum channel. However, the exact nature of the simulation implies that its memory and time scale exponentially with respect to the number of qubits.

Circuits involving more than 14 qubits are unfeasible to analyze.

An extra feature of the exact simulator is that it can handle all the branching possibilities that occur during the error detection steps. Traditionally, Monte Carlo methods have been used to analyze quantum circuits because of the random nature, not only of the errors, but also of the action of the error detection schemes, which probabilistically project the state of the data qubits to one of the subspaces defined by the QECC of choice. Our exact simulator keeps track of all the branching possibilities and their relative probabilities, and averages them at the end of the simulation. This adds an extra source of exponential scaling, as each branch bifurcates when a stabilizer is measured. (The maximal number of branches is 2^s, where s is the number of generators in the stabilizer group, but each generator might have to be measured more than once in order to achieve fault tolerance.) This overhead is handled by parallelizing the evolution of each branch on independent computer cores. We have implemented a desktop version that uses Python's *multiprocessing* module and a cluster version based on MPI [27, 28]. The exact simulator was used to generate the results presented in Chapter 5.

7.1.2 Fault-path tracing

The fault-path tracer is based on the work of Dennis and co-workers [30] and Aliferis and co-workers [8, 10]. This technique uses combinatorics and error propagation rules to trace error paths along the circuit. Our version is based on a modification proposed by Tomita [97], in which the tracing of the errors is done in reverse order, from the final qubits to the initial ones. This increases the efficiency of the algorithm because it only focuses on uncorrectable errors at the end of the circuit and traces them back to determine all the possible fault points or error sources that could have caused such an output. A somewhat related method has been proposed to compute the exact failure probability of a surface code in the asymptotic limit of small error rates [40].

The main advantage of this method is that it is much faster than the exact simulator and, in contrast to the Monte Carlo sampler, its cost does not depend on the error rate. For an arbitrary circuit, its cost scales exponentially with the number of fault points, which depends on the number of entangling gates in the circuit. The fault-path tracer was used to analyze two different error-correcting strategies of the Steane [[7,1,3]] code on a model ion-trap architecture [98].

7.1.3 Monte Carlo sampling

Monte Carlo simulation is currently the method of choice to analyze the effect of errors on quantum circuits [26, 9, 64, 72, 42]. In most cases, the goal of the error analysis is to estimate the probability of failure of a particular quantum algorithm, QECC, or error-correcting strategy. An analytical expression for the failure probability in terms of the physical error rates and other circuit and architecture parameters is possible in principle, but intractable for all but a few trivial cases. For such a reason, Monte Carlo methods are used to sample the set of error configurations, collect representative statistics, and obtain an accurate estimate to the probability of failure. To avoid the exponential cost of simulating general quantum circuits, the simulations involve stabilizer circuits [46, 2]. This reduces the time scaling from exponential to quadratic in the number of qubits. A major drawback of simulating only stabilizer circuits is that we are limited to algorithms and error models based on Clifford gates and Pauli measurements. This was, of course, one of the main motivations for the work presented on this thesis.

Our Monte Carlo simulator tool consists of a Python wrapper to a C-based stabilizer circuit simulator called CHP [2]. CHP stands for CNOT-Hadamard-Phase, the most commonly used primitive gates to generate the whole Clifford group. It was not developed by our group, but it is free software. CHP can efficiently simulate stabilizer circuits, but cannot handle classical decoding of error syndromes and error correction. These tasks are performed by the Python wrapper.

The current algorithm works in the following way. The user inputs the error-free circuit and the error information, which contains the probability that a particular error would occur after each gate. To select an error configuration, we traverse the circuit gate by gate. For every gate, we select a random number between 0 and 1 and, based on its value with respect to the error probability for that particular gate, we decide whether or not to add an error. If the random number is less than than error probability, we insert an error. In order to decide which error to add, we follow a similar procedure. We then move to the next gate in the circuit and perform the same process. Then, we run the whole circuit and compare its output to the error-free output. A failure will be defined by the two outputs being different. Finally, we do this enough times to collect representative statistics.

A significant disadvantage of this algorithm is that the number of necessary samples depends on the error rate. The great majority of the circuits we analyze involve error correction with a distance-3 code. By construction, these circuits are able to perfectly correct any 1-qubit error, which means that, in order to observe a failure, at least 2 errors need to occur. If the average error rate per gate is ϵ, this implies that the number of error configurations we need to sample scales like $\mathcal{O}(1/\epsilon^2)$. For higher distance codes, which can perfectly correct more than 1 error, the scaling is even worse. Therefore, with the current Monte Carlo algorithm we cannot access the regime of low error rates or large code distances. This is main reason for the development of alternative efficient approaches like the fault-path tracer, discussed in Subsection 7.1.2. Recently, however, a new Monte Carlo algorithm has been proposed and implemented to more efficiently simulate rare events in surface codes [21].

We have recently modified our algorithm for concatenated codes to make it more efficient and accesible to the low error regime. The modification is based on dividing the set of error configurations into subsets depending on the number of errors that occur. Analytical expressions for the probabilities of the various subsets can be obtained relatively easily. These depend on the error model and the particular circuit. We then let the user decide which subsets to sample. For example, if we have an error-correcting circuit based on a distance-3 code, the user can decide to sample only the subsets with 2 or more errors. The average failure probability numerically calculated after Monte Carlo sampling these subsets is then normalized by the analytical expression computed previously to obtain the overall failure probability.

In the modified version of the algorithm, we do not traverse the circuit gate by gate and probabilistically decide whether or not to add an error after each operation. Instead, the first step is to determine how many errors will be added to the circuit. This is decided based on the subsets that the user wants to focus on and the analytical expressions for the probabilities of each subset. Then, we select which gates to add errors after and run the circuit. Finally, we repeat this to collect statistics. Notice that, in the case of error-correcting circuits based on a distace-3 code, the scaling of the necessary number of samples on the sqaure of the inverse error rate $(1/\epsilon^2)$ vanishes if the user chooses to focus on the subsets with 2 or more errors. However, for low error rates, it will still be unlikely to sample a configuration with 3 or more errors, so the failure probability will be dominated by 2-error events. To gain accuracy, the user can carry independent runs on the 2-error subset, the 3-error subset, and so on, and then calculate a weighted average. We are currently running simulations with the modified version of the algorithm to benchmark it to the original version.

All the software described in this chapter can be found at:
http://ww2.chemistry.gatech.edu/brownlab/node/1082.

CHAPTER VIII

CONCLUSIONS

The construction of a scalable quantum computer will require an enormous experimental and theoretical effort. On the theory side, it is particularly important to study numerous error-suppressing strategies, in order to select the most effective ones. The assessment of these strategies requires classical simulations of large-scale quantum circuits. Due to the prohibitive exponential scaling of general circuits, simulations are restrained to specific subsets of quantum operations.

The current work has contributed to reduce the gap between accuracy and efficiency in the modeling of noise in these simulations of quantum error-correcting circuits. We introduced an extension to the traditional random Pauli error model, which leads to accurate, yet computationally tractable, descriptions of realistic error models at the single-qubit level.

At the logical error-corrected level, the behavior of these approximate channels depends strongly on whether the error is incoherent or coherent. For the former, the anisotropic Pauli channel exhibits an extreme accuracy that makes it a suitable approximation to be used in simulations. It also results in pessimistic values for a quantum error-correcting code's (QECC) pseudo-threshold. For coherent errors, the anisotropic Pauli channel results in bad approximations due to its high dishonesty. In this case, it would be more advisable to use one of the expanded channels to model the realistic noise. The study of honesty and accuracy at the logical error-corrected levels is useful to determine which channels can be employed to generate lower and upper bounds to a QECC's threshold under the influence of a specific error model.

8.1 Future directions

There are plenty of interesting open problems that stem from the results described on this work. We present a brief survery of several of these future directions.

8.1.1 Exact analysis of other error correcting protocols and QECCs

Extending the exact analysis, in particular at low error rates, to other correction protocols and codes will provide more insight into the error-supression effectiveness of different strategies. In the short term, it would be interesting to obtain the 1-qubit process matrix for a Steane code error correction step using the Steane method of error correction [93]. It is known that this method results in a slightly higher threshold value than the stabilizer measurement method employed in Chapter 5. This means that, despite being the same code, the 1-qubit process matrices would be different. However, will they have the same leading orders? How different will the coefficients of the leading orders be? It would also be interesting to explore other codes, like the Bacon-Shor $[[9,1,3]]$ code [14, 9], and determine the 1-qubit process matrix for them. The Knill method of error correction [64] is computationally too demanding to simulate for the Steane code, since it involves 21 qubits.

8.1.2 Real threshold simulations with the expanded error channels

In general, the point of intersection between the error curves at the physical level and the first logical level will be an upper bound to the real threshold. The points of intersection of error curves for higher levels of concatenation (or larger lattices in the case of topological codes) depend on the architecture characteristics and the decoding algorithm. The Monte Carlo tools described in Chapter 7 can be applied to perform stabilizer simulations of several concatenated QECCs, including the Steane $[[7,1,3]]$ and Bacon-Shor $[[9,1,3]]$. It would be interesting to study the honesty and accuracy of the Pauli and expanded approximations at higher levels of concatenation to determine whether or not the same relative pseudo-threhsold magnitudes between the channels hold. In particular, for coherent errors, will the unconstrained PC always result in an upper bound and the unconstrained CMC in a lower bound? We are currently working on incorporating the measurement-induced translations into CHP and finishing implementing the efficient sampler. We will also build a module to perform decoding at various levels of concatenation, including soft decoding [82, 45], in which the classical algorithm used to interpret the error syndromes includes communication between the different levels.

8.1.3 Simulation at higher levels of concatenation using the effective 1-qubit process matrices

The 1-qubit process matrices will allow us to simulate higher levels of concatenation "semi-accurately" in the following way. We obtain the effective 1-qubit process matrix for a circuit step consisting of errors on the physical qubits and subsequent error correction. At the next level of encoding, we can treat the circuit as composed of perfect physical gates followed by an error given by the 1-qubit process matrix obtained previously. As long as the errors are independent, this strategy would provide an accurate means to simulate circuits at high levels of concatenation. Of course, we cannot implement soft decoding with this technique, since there will not be communication between the different levels of concatenation.

8.1.4 Approximate channels' optimizations at the logical level

As explained in Chapter 5, the determining factor in the effectiveness of the approximated channels is not the accuracy at the physical level, but the accuracy at the logical error-corrected levels. It is likely that by performing the distance minimization at the logical level we can obtain approximations that are even more accurate than the ones obtained so far. This would be demanding computationally. It will be intractable for faulty error correction, in which we use ancillary qubits, but will still be manageable for perfect error correction.

8.1.5 Connections to other problems in quantum information

The future directions summarized previously are logical extensions to the work presented on this thesis. We are also interested in exploring connections that the current results might have to other problems in quantum information. One of them is the randomization of coherent errors. Is it possible to transform coherent errors into incoherent by a clever application of the stabilizer measurements or other form of error correction? If the errors become random, then the Pauli twirling approximation will not be as dishonest as it is for coherent errors.

Another interesting quesion is the potential application of our results to randomized benchmarking, the technique of choice to characterize quantum gates experimentally. Randomized benchmarking is performed by twirling the desired gate using operators in the Clifford group [70, 37]. Since at the logical levels the important error properties are retained by the Pauli channel, rather than the Clifford ones, it might be possible to perform randomized benchmarking with Pauli gates only.

Finally, the analysis of how coherent operators get transformed after error correction can provide us with alternative ways to implement non-transversal gates in QECCs. For a given QECC, it is known to be impossible to implement transversally all the gates needed to achieve universal quantum computation [36]. When performing quantum algorithms, the fault tolerant implementation of non-transversal gates is the most resource demanding part of the computation [42]. The study of alternatives to the current approaches to peform non-transversal gates is a very important problem in the community [20, 81, 11].

REFERENCES

[1] Aaronson, S. and Arkhipov, A., “The computational complexity of linear optics,” in *Proceedings of the 43rd annual ACM symposium on theory of computing*, STOC ’11, (New York, NY, USA), pp. 333–342, ACM, 2011.

[2] Aaronson, S. and Gottesman, D., “Improved simulation of stabilizer circuits,” *Phys. Rev. A*, vol. 70, p. 052328, 2004.

[3] Abrams, D. S. and Williams, C. P., “Fast quantum algorithms for numerical integrals and stochastic processes,” *arXiv:quant-ph/9908083*, 1999.

[4] Abu-Nada, A., Fortescue, B., and Byrd, M., “Relative performance of ancilla verification and decoding in the [[7,1,3]] Steane code,” *Phys. Rev. A*, vol. 89, p. 062304, 2014.

[5] Aharonov, D. and Ben-Or, M., “Fault-tolerant quantum computation with constant error,” in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, STOC ’97, (New York, NY, USA), pp. 176–188, ACM, 1997.

[6] Aharonov, D., Kitaev, A., and Preskill, J., “Fault-tolerant quantum computation with long-range correlated noise,” *Phys. Rev. Lett.*, vol. 96, p. 050504, 2006.

[7] Ahsan, M., *Architecture framework for trapped-ion quantum computer based on performance simulation tool.* PhD thesis, Duke University, 2015.

[8] Aliferis, P., *Level reduction and the quantum threshold theorem.* PhD thesis, California Institute of Technology, 2007.

[9] Aliferis, P. and Cross, A. W., “Subsystem fault tolerance with the Bacon-Shor code,” *Phys. Rev. Lett.*, vol. 98, p. 220502, 2007.

[10] Aliferis, P., Gottesman, D., and Preskill, J., “Quantum accuracy threshold for concatenated distance-3 codes,” *Quantum Inf. Comp.*, vol. 6, pp. 97–165, 2006.

[11] Anderson, J. T., Duclos-Cianci, G., and Poulin, D., “Fault-tolerant conversion between the Steane and Reed-Muller quantum codes,” *Phys. Rev. Lett.*, vol. 113, p. 080501, 2014.

[12] Aspuru-Guzik, A., Dutoi, A. D., Love, P. J., and Head-Gordon, M., “Simulated quantum computation of molecular energies,” *Science*, vol. 9, pp. 1704–1707, 2005.

[13] Awschalom, D. D., Bassett, L. C., Dzurak, A. S., Hu, E. L., and Petta, J. R., “Quantum spintronics: engineering and manipulating atom-like spins in semiconductors,” *Science*, vol. 339, pp. 1174–1179, 2013.

[14] Bacon, D., "Operator quantum error-correcting subsystems for self-correcting quantum memories," *Phys. Rev. A*, vol. 73, p. 012340, 2006.

[15] Benioff, P., "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *J. of Stat. Phys.*, vol. 22, pp. 563–591, 1980.

[16] Benioff, P., "Quantum mechanical models of turing machines that dissipate no energy," *Phys. Rev. Lett.*, vol. 48, pp. 1581–1585, 1982.

[17] Bernstein, E. and Vazirani, U., "Quantum complexity theory," *SIAM J. Comput.*, vol. 26, pp. 1411–1473, 1997.

[18] Blume-Kohout, R., "Optimal, reliable estimation of quantum states," *New J. Phys.*, vol. 12, p. 043034, 2010.

[19] Bravyi, S., "Stabilizer subsystem codes with spatially local generators," in *Information Theory Workshop (ITW), 2010 IEEE*, pp. 1–5, 2010.

[20] Bravyi, S. and Kitaev, A., "Universal quantum computation with ideal clifford gates and noisy ancillas," *Phys. Rev. A*, vol. 71, p. 022316, 2005.

[21] Bravyi, S. and Vargo, A., "Simulation of rare events in quantum error correction," *Phys. Rev. A*, vol. 88, p. 062308, 2013.

[22] Calderbank, A. R. and Shor, P. W., "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, 1996.

[23] Carolan, J., Harrold, C., Sparrow, C., Martn-Lpez, E., Russell, N. J., Silverstone, J. W., Shadbolt, P. J., Matsuda, N., Oguma, M., Itoh, M., Marshall, G. D., Thompson, M. G., Matthews, J. C. F., Hashimoto, T., OBrien, J. L., and Laing, A., "Universal linear optics," *Science*, vol. 349, pp. 711–716, 2015.

[24] Chuang, I. L. and Nielsen, M. A., "Prescription for experimental determination of the dynamics of a quantum black box," *J. Mod. Opt.*, vol. 44, pp. 2455–2467, 1997.

[25] Cirac, J. I. and Zoller, P., "Quantum computations with cold trapped ions," *Phys. Rev. Lett.*, vol. 74, pp. 4091–4094, 1995.

[26] Cross, A. W., DiVincenzo, D. P., and Terhal, B. M., "A comparative code study for quantum fault tolerance," *Quantum Inf. Comput.*, vol. 9, pp. 541–572, 2009.

[27] Dalcín, L., Paz, R., and Storti, M., "MPI for Python," *Journal of Parallel and Distributed Computing*, vol. 65, pp. 1108 – 1115, 2005.

[28] Dalcín, L., Paz, R., Storti, M., and D'Elía, J., "MPI for Python: Performance improvements and MPI-2 extensions," *Journal of Parallel and Distributed Computing*, vol. 68, pp. 655 – 662, 2008.

[29] Dawson, C. M. and Nielsen, M. A., "The Solovay-Kitaev algorithm," *Quantum Info. Comput.*, vol. 6, no. 1, pp. 81–95, 2006.

[30] Dennis, E., Kitaev, A., Landahl, A., and Preskill, J., "Topological quantum memory," *J. Math. Phys.*, vol. 43, p. 4452, 2002.

[31] Deutsch, D., "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proc. R. Soc. A*, vol. 400, pp. 97–117, 1985.

[32] Devoret, M. H. and Schoelkopf, R. J., "Superconducting circuits for quantum information: an outlook," *Science*, vol. 339, pp. 1169–1174, 2013.

[33] DiVincenzo, D. P. and Shor, P. W., "Fault-tolerant error correction with efficient quantum codes," *Phys. Rev. Lett.*, vol. 77, pp. 3260–3263, 1996.

[34] Du, J., Xu, N., Peng, X., Wang, P., Wu, S., and Lu, D., "NMR implementation of a molecular hydrogen quantum simulation with adiabatic state preparation," *Phys. Rev. Lett.*, vol. 104, p. 030502, 2010.

[35] Duan, L.-M. and Monroe, C., "*Colloquium*: Quantum networks with trapped ions," *Rev. Mod. Phys.*, vol. 82, pp. 1209–1224, 2010.

[36] Eastin, B. and Knill, E., "Restrictions on transversal encoded quantum gate sets," *Phys. Rev. Lett.*, vol. 102, p. 110502, 2009.

[37] Emerson, J., Silva, M., Moussa, O., Ryan, C., Laforest, M., Baugh, J., Cory, D. G., and Laflamme, R., "Symmetrized characterization of noisy quantum processes," *Science*, vol. 317, pp. 1893–1896, 2007.

[38] Feynman, R. P., "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, 1982.

[39] Fowler, A. G., "Proof of finite surface code threshold for matching," *Phys. Rev. Lett.*, vol. 109, p. 180502, 2012.

[40] Fowler, A. G., "Analytic asymptotic performance of topological codes," *Phys. Rev. A*, vol. 87, p. 040301, 2013.

[41] Fowler, A. G., "Coping with qubit leakage in topological codes," *Phys. Rev. A*, vol. 88, p. 042308, 2013.

[42] Fowler, A. G., Devitt, S. J., and Jones, C., "Surface code implementation of block code state distillation," *Sci. Rep.*, vol. 3, p. 1939, 2013.

[43] Geller, M. R. and Zhou, Z., "Efficient error models for fault-tolerant architectures and the Pauli twirling approximation," *Phys. Rev. A*, vol. 88, p. 012314, 2013.

[44] GILCHRIST, A., LANGFORD, N. K., and NIELSEN, M. A., "Distance measures to compare real and ideal quantum processes," *Phys. Rev. A*, vol. 71, p. 062310, 2005.

[45] GOTO, H. and UCHIKAWA, H., "Fault-tolerant quantum computation with a soft-decision decoder for error correction and detection by teleportation," *Sci. Rep.*, vol. 3, p. 2044, 2013.

[46] GOTTESMAN, D., *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.

[47] GOTTESMAN, D., *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.

[48] GRACE, M. D., DOMINY, J., KOSUT, R. L., BRIF, C., and RABITZ, H., "Environment-invariant measure of distance between evolutions of an open quantum system," *New J. Phys.*, vol. 12, p. 015001, 2010.

[49] GROSS, D., LIU, Y.-K., FLAMMIA, S. T., BECKER, S., and EISERT, J., "Quantum state tomography via compressed sensing," *Phys. Rev. Lett.*, vol. 105, p. 150401, 2010.

[50] GROVER, L. K., "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), pp. 212–219, ACM, 1996.

[51] GUTIÉRREZ, M. and BROWN, K. R., "Comparison of a quantum error-correction threshold for exact and approximate errors," *Phys. Rev. A*, vol. 91, p. 022335, 2015.

[52] GUTIÉRREZ, M., SVEC, L., VARGO, A., and BROWN, K. R., "Approximation of realistic errors by Clifford channels and Pauli measurements," *Phys. Rev. A*, vol. 87, p. 030302, 2013.

[53] HANSON, R. and AWSCHALOM, D. D., "Coherent manipulation of single spins in semiconductors," *Nature*, vol. 453, pp. 1043–1049, 2008.

[54] HARROW, A. W., HASSIDIM, A., and LLOYD, S., "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, p. 150502, 2009.

[55] HUCUL, D., INLEK, I. V., VITTORINI, G., CROCKER, C., DEBNAH, S., CLARK, S. M., and MONROE, C., "Modular entanglement of atomic qubits using photons and phonons," *Nat. Phys.*, vol. 11, pp. 37–42, 2015.

[56] HUMPHREYS, J. F. and PREST, M. Y., *Numbers, Groups, and Codes*. Cambridge, UK: Cambridge University Press, 2004.

[57] ISLAM, R., SENKO, C., CAMPBELL, W. C., KORENBLIT, S., SMITH, J., LEE, A., EDWARDS, E. E., WANG, C.-C. J., FREERICKS, J. K., and MONROE, C., "Emergence and frustration of magnetism with variable-range interactions in a quantum simulator," *Science*, vol. 340, pp. 583–587, 2013.

[58] JORDAN, S. P., LEE, K. S. M., and PRESKILL, J., “Quantum algorithms for quantum field theories,” *Science*, vol. 336, pp. 1130–1133, 2012.

[59] JOUZDANI, P., NOVAIS, E., TUPITSYN, I. S., and MUCCIOLO, E. R., “Fidelity threshold of the surface code beyond single-qubit error models,” *Phys. Rev. A*, vol. 90, p. 042315, 2014.

[60] KASSAL, I., JORDAN, S. P., LOVE, P. J., MOHSENI, M., and ASPURU-GUZIK, A., “Polynomial-time quantum algorithm for the simulation of chemical dynamics,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 105, pp. 18681–18686, 2008.

[61] KASSAL, I., WHITFIELD, J. D., PERDOMO-ORTIZ, A., YUNG, M.-H., and ASPURU-GUZIK, A., “Simulating chemistry using quantum computers,” *Annu. Rev. Phys. Chem.*, vol. 62, pp. 185–207, 2011.

[62] KITAEV, A. Y., SHEN, A. H., and VYALYI, M. N., *Classical and quantum computation*. Boston, MA, USA: American Mathematical Society, 2002.

[63] KITAEV, A. Y., “Quantum computations: algorithms and error corrrection,” *Russ. Math. Surv.*, vol. 52, p. 1191, 1997.

[64] KNILL, E., “Quantum computing with realistically noisy devices,” *Nature*, vol. 434, pp. 39–44, 2005.

[65] KNILL, E., LAFLAMME, R., and ZUREK, W. H., “Resilient quantum computation: error models and thresholds,” *Proc. R. Soc. A*, vol. 454, pp. 365–384, 1998.

[66] LANYON, B., WHITFIELD, J., GILLET, G., GOGGIN, M., ALMEIDA, M., KASSAL, I., BIAMONTE, J., MOHSENI, M., POWELL, B., BARBIERI, M., ASPURU-GUZIK, A., and WHITE, A., “Towards quantum chemistry on a quantum computer,” *Nat. Chem.*, vol. 2, pp. 106–111, 2010.

[67] LLOYD, S., “Universal quantum simulators,” *Science*, vol. 273, pp. 1073–1078, 1996.

[68] MAGESAN, E., PUZZUOLI, D., GRANADE, C. E., and CORY, D. G., “Modeling quantum noise for efficient testing of fault-tolerant circuits,” *Phys. Rev. A*, vol. 87, p. 012324, 2013.

[69] MARTINIS, J. M., NAM, S., AUMENTADO, J., and URBINA, C., “Rabi oscillations in a large Josephson-junction qubit,” *Phys. Rev. Lett.*, vol. 89, p. 117901, 2002.

[70] MEIER, A., *Randomized benchmarking of Clifford operators*. PhD thesis, University of Colorado at Boulder, 2013.

[71] METODI, T. S., THAKER, D. D., CROSS, A. W., CHONG, F. T., and CHUANG, I. L., “A quantum logic array microarchitecture: scalable quantum data movement and computation,” in *MICRO-38: Proc. 38TH Annual IEEE/ACM Int. Symp. on Microarchitecture*, pp. 305–316, 2005.

[72] MISHRA, P. and FOWLER, A. G., "Resource comparison of two surface code implementations of small angle Z rotations," *arXiv:1406.4948 [quant-ph]*, 2014.

[73] MONROE, C. and KIM, J., "Scaling the ion trap quantum processor," *Science*, vol. 339, pp. 1164–1169, 2013.

[74] MONROE, C., RAUSSENDORF, R., RUTHVEN, A., BROWN, K. R., MAUNZ, P., DUAN, L.-M., and KIM, J., "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Phys. Rev. A*, vol. 89, p. 022317, 2014.

[75] NENNO, R. B., "An introduction to the theory of nonlinear error correcting codes," Master's thesis, Rochester Institute of Technology, 1987.

[76] NG, H. K. and PRESKILL, J., "Fault-tolerant quantum computation versus Gaussian noise," *Phys. Rev. A*, vol. 79, p. 032318, 2009.

[77] NIELSEN, M. A. and CHUANG, I. L., *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge University Press, 2001.

[78] NIELSEN, M. A. and POULIN, D., "Algebraic and information-theoretic conditions for operator quantum error correction," *Phys. Rev. A*, vol. 75, p. 064304, 2007.

[79] NIGG, D., MÜLLER, M., MARTINEZ, E. A., SCHINDLER, P., HENNRICH, M., MONZ, T., MARTIN-DELGADO, M. A., and BLATT, R., "Quantum computations on a topologically encoded qubit," *Science*, vol. 345, p. 302, 2014.

[80] NOVAIS, E. and MUCCIOLO, E. R., "Surface code threshold in the presence of correlated errors," *Phys. Rev. Lett.*, vol. 110, p. 010502, 2013.

[81] PAETZNICK, A. and REICHARDT, B. W., "Universal fault-tolerant quantum computation with only transversal gates and error correction," *Phys. Rev. Lett.*, vol. 111, p. 090505, 2013.

[82] POULIN, D., "Optimal and efficient decoding of concatenated quantum block codes," *Phys. Rev. A*, vol. 74, p. 052333, 2006.

[83] PUZZUOLI, D., GRANADE, C. E., HAAS, H., CRIGER, B., MAGESAN, E., and CORY, D. G., "Tractable simulation of error correction with honest approximations to realistic fault models," *Phys. Rev. A*, vol. 89, p. 022306, 2014.

[84] RAUSSENDORF, R., HARRINGTON, J., and GOYAL, K., "Topological fault-tolerance in cluster state quantum computation," *New J. Phys.*, vol. 9, p. 199, 2007.

[85] REICHARDT, B. W., "Fault-tolerance threshold for a distance-three quantum code," in *Automata, Languages and Programming* (BUGLIESI, M., PRENEEL, B., SASSONE, V., and WEGENER, I., eds.), vol. 4051 of *Lecture Notes in Computer Science*, pp. 50–61, Springer Berlin Heidelberg, 2006.

[86] RIVEST, R., SHAMIR, A., and ADLEMAN, L., "Cryptographic communications system and method," 1983. US Patent 4,405,829.

[87] SHABANI, A., KOSUT, R. L., MOHSENI, M., RABITZ, H., BROOME, M. A., ALMEIDA, M. P., FEDRIZZI, A., and WHITE, A. G., "Efficient measurement of quantum dynamics via compressive sensing," *Phys. Rev. Lett.*, vol. 106, p. 100401, 2011.

[88] SHOR, P. W., "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, 1995.

[89] SHOR, P. W., "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, pp. 1484–1509, 1997.

[90] SØRENSEN, A. and MØLMER, K., "Quantum computation with ions in thermal motion," *Phys. Rev. Lett.*, vol. 82, pp. 1971–1974, 1999.

[91] STEANE, A. M., "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793–797, 1996.

[92] STEANE, A. M., "Space, time, parallelism and noise requirements for reliable quantum computing," *Fortsch. Phys.*, vol. 46, pp. 443–458, 1998.

[93] STEANE, A. M., "Overhead and noise threshold of fault-tolerant quantum error correction," *Phys. Rev. A*, vol. 68, no. 042322, p. 042322, 2003.

[94] SUCHARA, M., CROSS, A. W., and GAMBETTA, J. M., "Leakage suppression in the toric code," *Quant. Inf. Comp.*, vol. 15, pp. 1119–1123, 2015.

[95] SVORE, K. M., DIVINCENZO, D. P., and TERHAL, B. M., "Noise threshold for a fault-tolerant two-dimensional lattice architecture," *Quantum Inf. Comp.*, vol. 7, pp. 297–318, 2007.

[96] SZABO, A. and OSTLUND, N. S., *Modern quantum chemistry: introduction to advanced electronic structure theory*. New York: Dover Publications, 1989.

[97] TOMITA, Y., *Numerical and analytical studies of quantum error correction*. PhD thesis, Georgia Institute of Technology, 2014.

[98] TOMITA, Y., GUTIÉRREZ, M., KABYTAYEV, C., BROWN, K. R., HUTSEL, M. R., MORRIS, A. P., STEVENS, K. E., and MOHLER, G., "Comparison of ancilla preparation and measurement procedures for the Steane [[7,1,3]] code on a model ion-trap quantum computer," *Phys. Rev. A*, vol. 88, p. 042336, 2013.

[99] TOMITA, Y. and SVORE, K. M., "Low-distance surface codes under realistic quantum noise," *Phys. Rev. A*, vol. 90, p. 062320, 2014.

[100] VAN DAM, W. and HOWARD, M., "Tight noise thresholds for quantum computation with perfect stabilizer operations," *Phys. Rev. Lett.*, vol. 103, p. 170504, 2009.

[101] Van Rynbach, A., Ahsan, M., Mehta, A. C., Hussmann, J., and Kim, J., "A Quantum Performance Simulator based on fidelity and fault-path counting," *arXiv:1212.0845 [quant-ph]*, 2012.

[102] Vandersypen, L. M. K., Steffen, M., Breyta, G., Yannoni, C. S., Sherwood, M. H., and Chuang, I. L., "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, pp. 883–887, 2001.

[103] Wang, Y., Dolde, F., Biamonte, J., Babbush, R., Bergholm, V., Yang, S., Jakobi, I., Neumann, P., Aspuru-Guzik, A., Whitfield, J. D., and Wrachtrup, J., "Quantum simulation of helium hydride cation in a solid-state spin register," *ACS Nano*, vol. 9, pp. 7769–7774, 2015.

[104] Weinstein, Y. S., "Quantum error correction during 50 gates," *Phys. Rev. A*, vol. 89, p. 020301, 2014.

[105] Wooters, W. K. and Zurek, W. H., "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802–803, 1982.

[106] Zeng, F., *Nonlinear codes: representation, constructions, minimum distance computation and decoding*. PhD thesis, Universitat Autònoma de Barcelona, 2014.

[107] Zhang, J., Man-Hong, Y., Laflamme, R., Aspuru-Guzik, A., and Baugh, J., "Digital quantum simulation of the statistical mechanics of a frustrated magnet," *Nat. Commun.*, vol. 3, pp. 1–10, 2012.