– Independent Work Report Fall 2020 –

Modelling Error Correction on Optical Interconnect Quantum Systems

Albert Kim
Adviser: Margaret Martonosi

Abstract

Recent efforts aim to design modular, scalable quantum computers with thousands of qubits. One such effort proposes to use photonic interconnects to link separate qubit registers in order to avoid distance challenges that eventually arise in QCCD computers. For a computer of this scale, we anticipate the implementation of fault-tolerant qubits, or quantum error correction. However, little work has been done to model and analyze the performance of optical interconnect quantum computers with fault-tolerant error correction. In this paper, we model error correction on a simple quantum computer linked with photonic interconnects using an error correcting schematic via a small surface code, Surface-17. We then aim to explore the approximate upper bounds of hardware-specific error constants that yield acceptable logical gate success rates. Additionally, we set groundwork for future comparisons of error correction related design choices and experiments through structured code.

1. Introduction

1.1. Quantum Computing

Quantum computing is an emerging field that leverages principles of quantum mechanics to manipulate information. That is, instead of bits we use qubits which can be in a superposition of the '0' and '1' state, and to perform computations we apply gates on the qubits. Through quantum computing, it becomes possible to efficiently solve some problems in fields ranging from cryptography to biology that would otherwise be intractable in a classical setting. The solutions to these problems require many qubits, for example, quantum advantageous use of Shor's factoring algorithm requires tens of thousands of qubits. However, quantum computers have only recently reached the 50-100 qubit milestone. It is thus the goal of many quantum computing researchers to close this gap between the capability of modern hardware and how many qubits are needed for useful algorithms.

1.2. Hardware

The current generation of functional quantum computer systems have fairly low qubit counts and high operational noise, and are therefore called Noisy Intermediate-Scale Quantum (NISQ) systems. Leading technologies for building these NISQ systems are superconducting qubit systems and trapped ion qubit systems. The latter is implemented by suspending ions within a space (trap) using electromagnetic fields. The ions are arranged in a linear chain and each store a qubit, and lasers are used to manipulate the ion states to implement gate-based computation. Figure 1 shows a trapped ion qubit system. Trapped ion qubits hold a number of advantages over superconducting qubits, including full connectivity, low error rate, and defect-free identical qubits. A recent study has shown that a trapped ion qubit system performs better than a superconducting qubit system of the same size [5]. However, trapped ion systems face scalability limitations due to gate implementation challenges in long ion chains.

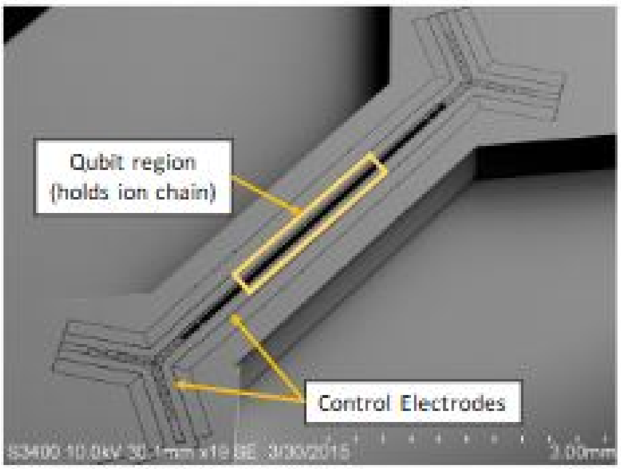

Figure 1: Scanning electron micrograph of the Sandia HOA2 trap. Figure from [6].

In an effort to push trapped ion systems to the 50-100 qubit scale, an architecture called the Quantum Charge Coupled Device (QCCD) was proposed [4]. Ions are held in several, separate traps and are shuttled through space as required. The first QCCD system with 4 qubits has already been built [3] making this a promising scalable architecture. However, QCCD systems face their own scalability limitations past 1000 qubits due to the increasing complexity of moving quantum information through the system and eventual limit to quantum processing speed [1]. In light of this, another, further scalable design for a quantum computer that uses optical interconnects has been proposed [1]. This architecture aims to link separate registers of trapped ion chains with photonic interfaces to avoid the complexity that scales with distance between the qubits. Figure 2 outlines the design of this system, and further information can be found at [1]. In an architecture of this scale, it will become necessary to have fault-tolerant qubits, thus it is important that development of this architecture is done with understanding in how well we expect error correction to perform.

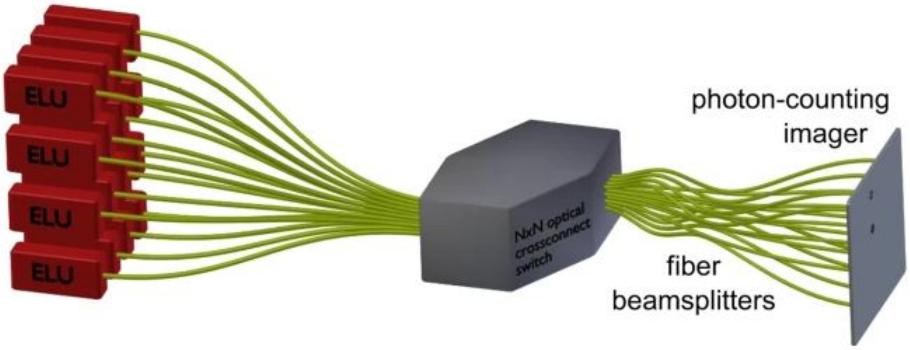

Figure 2: Modular distributed quantum computer. Several ELU (elementary logic unit) modules, each in control of 50-100 qubits, are connected via a photonic network. Figure from [1].

Optical interconnect quantum systems will necessarily implement fault-tolerant error correction. However, little work has been done to explore how well error correction would work on such a system. The goal of my Independent Work project was to model error correction on an elementary version of an optical interconnect system then observe how it affects logical gate success rate in order to obtain hardware benchmarks to potentially guide future development of optical interconnect quantum systems. In addition, I set the groundwork for future comparisons of error correction related design choices and experiments via code.

2. Approach

Using Cirq, an open source Python framework for simulating NISQ computers, we first model a downscaled version of an optical interconnect quantum system, i.e. a system with a given number of traps, each controlling only a few qubits, which are linked through a photonic interface. We then implement a small surface code, namely Surface-17, and use it to encode logical qubits in our system model. Finally, we simulate several runs of the logical CNOT gate and measure the success rate against certain hardware error parameters. This will be informative of the approximate range of constants that we need to reach acceptable logical gate fidelities.

3. Implementation

3.1. Surface Code

To protect quantum information from faults that may occur due to the environment, we can use quantum error correcting codes. In this project, we implement error correction based on a small surface code, namely Surface-17 which encodes 9 data qubits and 8 ancilla qubits into a single logical qubit, and is the smallest useful realization of a surface code [7]. The data qubits are arranged in a 3 by 3 lattice labelled 9 to 17 in Figure 3. The code is defined by a set of stabilizer operations (implemented by Hadamard and CNOT gates) that projects error into the ancilla qubits (in order to avoid destroying information in the data qubits), and can detect any single qubit error on the 9 data qubits. Information we get from measurement of the ancilla qubits is known as the syndrome, and extracting the most likely logical state from the syndrome is called decoding. For larger surface codes, decoding typically requires the use of sophisticated algorithms, but for the purposes of our project, because the surface code is small, it suffices to use a simple process based on look-up tables to perform decoding. A more detailed explanation of Surface-17 can be found here [7].

To implement our surface code, we create a circuit that initializes the 17 qubits, performs the projection of the error into the ancilla qubits, and measures the ancilla qubits. We then use the measurements to find via a lookup table the corresponding ’correction’ which is in the form of a

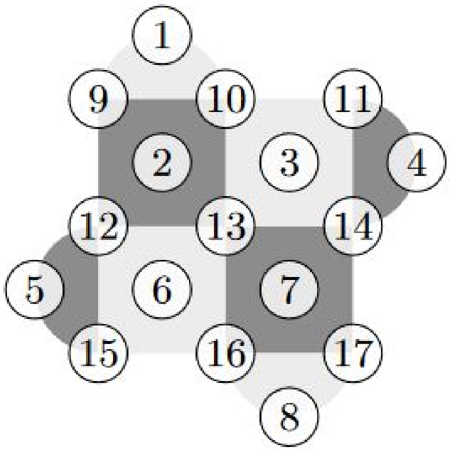

Figure 3: Ancilla qubits labelled 1-8, data qubits labelled 9-17. The positions of the ancilla qubits are representative of the data qubits from which errors are detected. Figure from [7].

single qubit gate on each data qubit. Then, we create another circuit that runs the relevant correction gates on the data qubits, and measures the data qubits. Finally, we use the secondary measurements and another lookup table to retrieve our final logical measurement which is ’0’ or ’1’. We go through these steps whenever we want to measure the logical state of a logical qubit. Note that all gates that are executed in this process are Clifford gates - therefore by the Gottesman-Knill theorem, simulation can be done in polynomial time. This makes our surface code simulation and all of our future experiments feasible. To model our fault-tolerant system, for simplicity, we model each ELU in our optical interconnect quantum computer as an ion trap in control of 17 qubits. Then through our surface code, we encode all the qubits in a single ELU to a single logical qubit, so that each ELU corresponds to a logical qubit. Because it has been observed that error does not change significantly with changes in qubit state [7], we consider only the ’0’ or ’1’ initial logical states. In addition, whenever we refer to the execution of a logical gate on a logical qubit, we are referring to a gate that is applied to all data qubits in the logical qubit.

3.2. Noise and Error Models

Any environmental effects to our qubits that are outside our control must be treated as noise. Simulating realistic noise is difficult, so in this project we make strong assumptions about the nature of the environment and how it will affect our qubits. We consider three types of error in reference to a related paper [7] that can occur as a result of noise - bitflip, phaseflip, and depolarization error. Depolarization erases all information stored in a qubit, that is, randomizes its state at a given probability. Unfortunately, because we can only use Clifford gates, we are unable to utilize Cirq's implementation of a depolarizing channel, so instead we interpret depolarization as either a Hadamard or X gate occurring at random chance. We set the probability of each error occurring to be the same, that is,

$$P(\text{bitflip error}) = P(\text{phaseflip error}) = P(\text{depolarization})$$

with them occurring mutually exclusively. Because this is not modelled on a specific piece of hardware, the aforementioned issue with depolarization, and lack of consideration for time evolution, this model is likely not representative of realistic noise. However, even with arbitrary choice of probability, it suffices to demonstrate the effectiveness of our surface code.

In addition to noise, we expect errors to occur on the system upon runs of 2-qubit or entangling gates. These errors are hardware-specific, and possess structured models. For this project, we experiment with error models based on notes by Dripto Debroy - a chain-specific error model and a gate teleportation error model [2]. In particular, these models describe what errors we expect to occur upon execution of a CNOT gate. Chain-specific error occurs with probability that scales with distance between the control and target qubits, and gate teleportation error occurs upon applying a 2-qubit gate between qubits in different traps. Specifically, upon applying a CNOT gate there is a

- αd^3 probability of a ZX error to occur for ions on the same chain
- $\alpha d_1^3 + \alpha d_2^3$ probability of a ZX error to occur for ions on different chains
- $P_{dark}/2$ probability for a Z to be applied to the control instead of the CNOT
- $P_{dark}/2$ probability for nothing to happen instead of the CNOT

where α and P_{dark} are constants that affect the probabilities of chain-specific error and gate teleportation error respectively. Thus we expect the occurrences of these errors to fall under two cases: error from CNOT gates on ions in the same trap and error from CNOT gates on ions in separate traps. See figures 4 and 5.

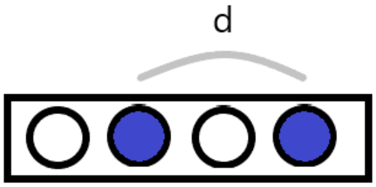

Figure 4: A CNOT gate on two ions (colored blue) in the same chain has a chance of inducing a chain-specific error with probability proportional to d^3.

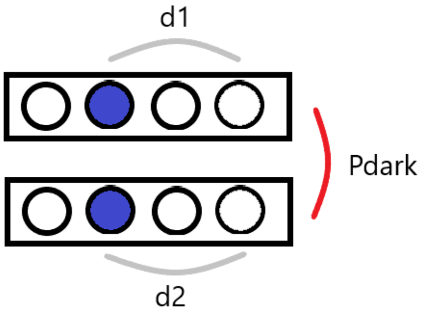

Figure 5: A CNOT gate on two ions (colored blue) in separate chains has a chance of inducing either or both a chain-specific error and a gate teleportation error.

3.3. Experiments

In this paper, we design some experiments to examine how our noise and error models affect logical state and gate error rates. First, we initialize a value for the noise probability (0.01), then repeat initialization and measurement of a single logical qubit 1000 times and record the logical error rate. We iterate through these steps 100 times, and for each iteration, we increment the noise probability (the probability that one of the three aforementioned types of noise occurs) by 0.01, and observe how well our logical qubit tolerates noise.

For the following experiments we aim to obtain data related to 2-qubit gate errors, and since we have not modelled real architecture we may set noise to zero. We initialize a value for α (found by trial and error) and go through the following steps:

1. Initialize two logical qubits, each in separate ELU's or traps. The first logical qubit is in either the '0' or '1' state at random chance while the second is in the '0' state.
2. Perform a logical CNOT gate on the two logical qubits 10,000 times, and record the gate error rate.
3. Increment α by a certain value (found by trial and error).

4. Iterate previous steps 100 times.

The idea is to give us a view on how α affects the logical CNOT error rate. We also perform an identical experiment for P_{dark}. For both experiments, we are particularly interested in the values of the constants that yield 99% and 99.9% logical CNOT success rates.

Finally, we would like to compare the implementations of two logical qubits in separate traps and two logical qubits together in one ion chain. Specifically, we want to compare the values of α in both systems that yield 99% and 99.9% logical CNOT success rates. The purpose of this is to verify that optical interconnect systems are more forgiving with respect to the chain-specific error constant α.

4. Results

4.1. Logical qubit performance in noisy environment

Figure 6 shows the plot of logical state success rate against noise probability from the first experiment.

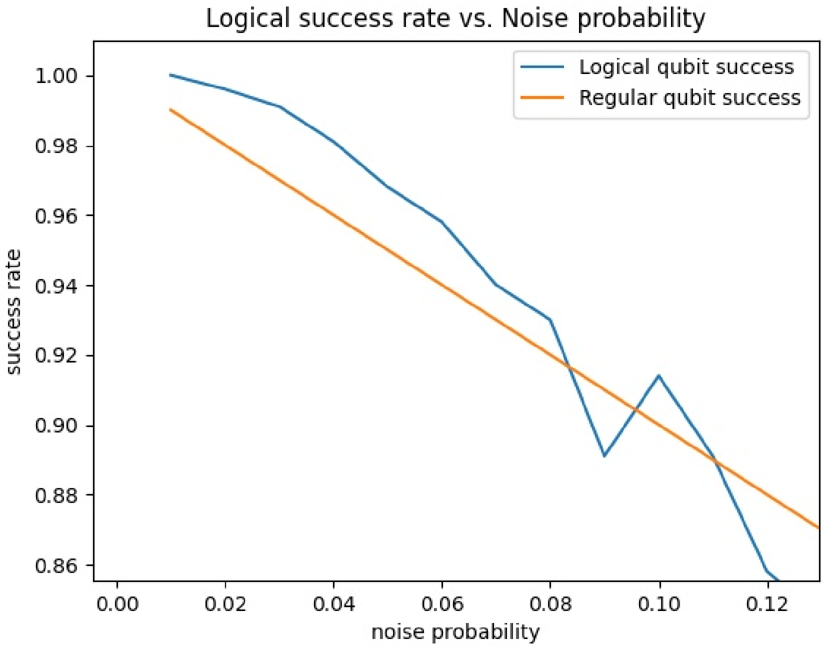

Figure 6: Logical qubit performance in noisy environment

We can see that for sufficiently low values of noise probability, measurement of the logical qubit returns the correct state more often than measurement of a single qubit. The logical qubit begins to perform worse than a regular qubit at around 0.08 noise probability. This is an anticipated result since Surface-17 is able to detect and correct only single qubit errors - as the noise probability exceeds a certain threshhold, multiple qubit errors occur more often, and as a result the surface code performs worse. For low noise, the surface code appears to increase the correct measurement rate by 2% to 3%. In general, it appears that the lower the noise probability, the higher percentage of the noise that the surface code is able to tolerate. We may conclude that our implementation of Surface-17 successfully detects and corrects noise to a significant extent.

4.2. Logical CNOT gate performance

Figures 7 and 8 show the plots of logical CNOT gate error rate against α, the constant affecting the distance dependent chain-specific error. In the absence of noise and gate teleportation error, a 99% and 99.9% logical CNOT success rate is realized at $\alpha \approx 5 \times 10^{-5}$ and $\alpha \approx 1.5 \times 10^{-5}$ respectively.

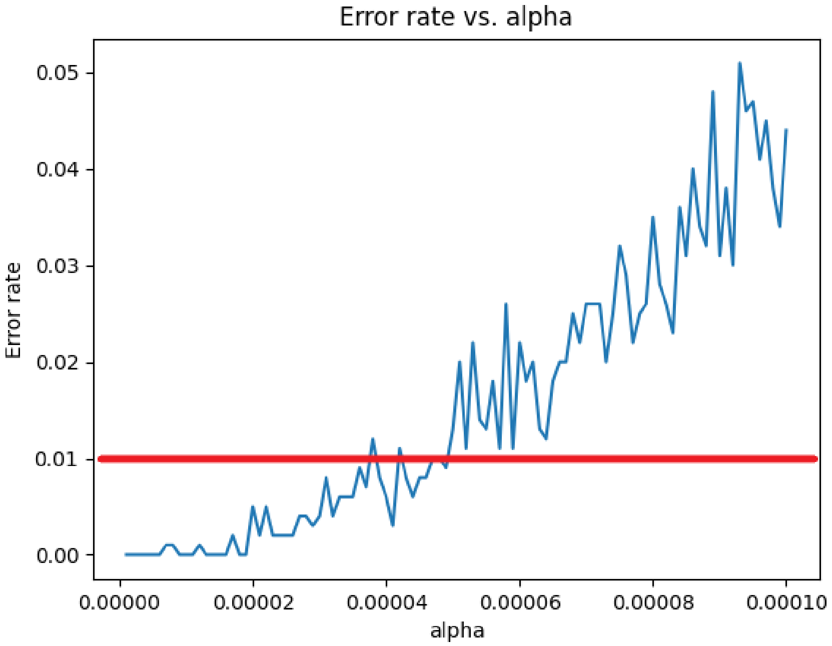

Figure 7: Logical CNOT gate performance in presence of chain-specific error. The red line represents 1% logical CNOT error rate.

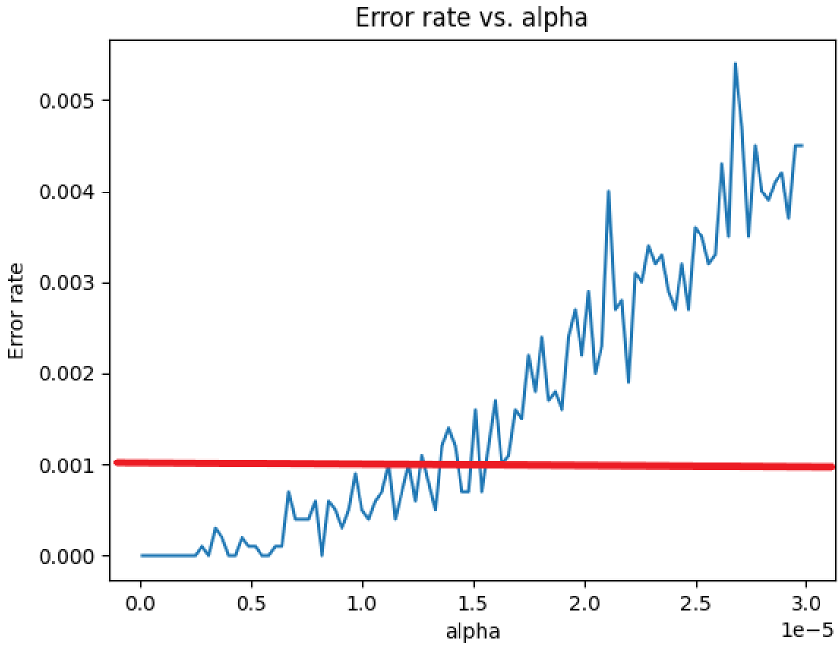

Figure 8: Logical CNOT gate performance in presence of chain-specific error. The red line represents 0.1% **logical CNOT error rate.**

Figures 9 and 10 show the plots of logical CNOT gate error rate against P_{dark}, the constant affecting the distance independent gate teleportation error. In the absence of noise and chain-specific error, a 99% and 99.9% logical CNOT success rate is realized at $P_{dark} \approx 4 \times 10^{-2}$ and $P_{dark} \approx 1.2 \times 10^{-2}$ respectively.

Without any sort of error correcting schematic or chain-specific error, we expect that a regular CNOT on two qubits in different traps would have an error rate equivalent to P_{dark}. However, we see that for the logical CNOT, this is not the case. In particular, our surface code mitigates gate teleportation error by about 75% at $P_{dark} \approx 4 \times 10^{-2}$ and by about 92% at $P_{dark} \approx 1.2 \times 10^{-2}$. We can infer that the surface code is effective at tolerating gate teleportation error, and becomes more effective as P_{dark} decreases. This is also anticipated, as an occurrence of a gate teleportation error essentially is an occurrence of a single qubit error at each logical qubit. On the other hand, it is difficult to make a similar inference for the chain-specific error. Due to chain-specific error's dependency on distance and its potential to manifest itself in the process of decoding, it is challenging to deduce whether implementing the surface code successfully mitigates this type of error. Even so,

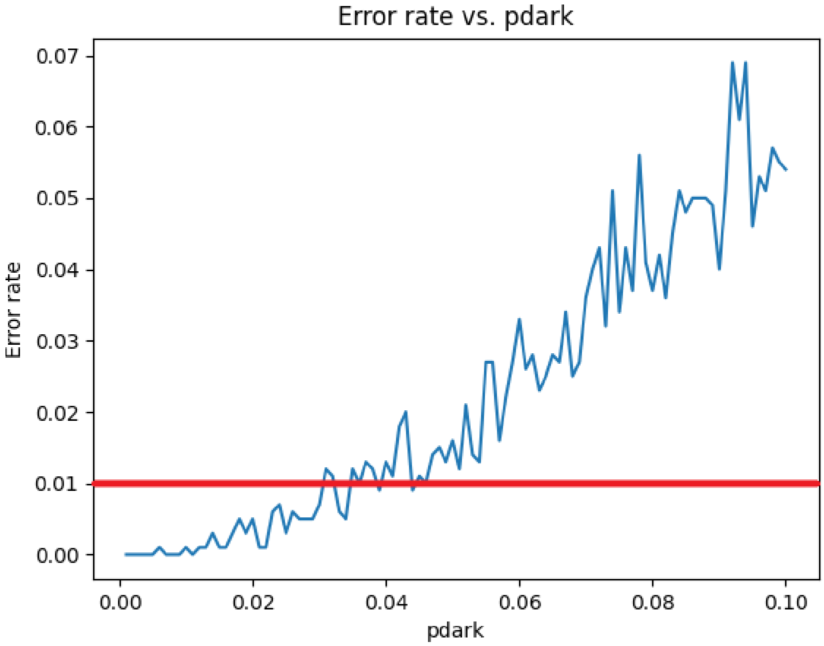

Figure 9: Logical CNOT gate performance in presence of gate teleportation error. The red line represents 1% logical CNOT error rate.

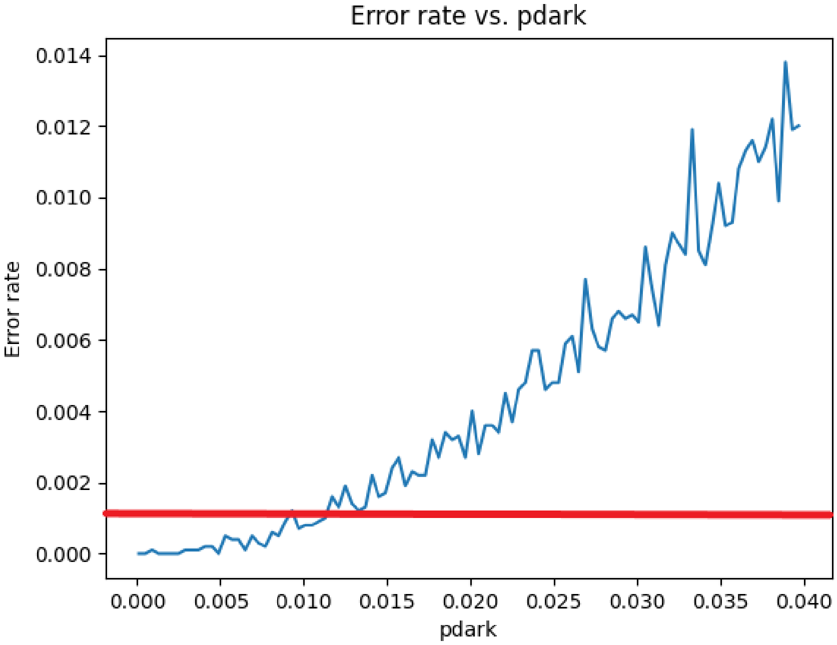

Figure 10: Logical CNOT gate performance in presence of gate teleportation error. The red line represents 0.1% logical CNOT error rate.

overall, the retrieved values of the constants are significant as upper bounds for yielding acceptable logical CNOT success rates.

4.3. Ion chain layout comparison

Figures 11 and 12 show the plots of logical CNOT gate error rate against α within a single ion trap. That is, with 34 qubits in a single ion chain and two encoded logical qubits. In the absence of noise and gate teleportation error, a 99% and 99.9% logical CNOT success rate is realized at $\alpha \approx 2.4 \times 10^{-5}$ and $\alpha \approx 6 \times 10^{-6}$ respectively. This is in comparison with the respective values of $\alpha \approx 5 \times 10^{-5}$ and $\alpha \approx 1.5 \times 10^{-5}$ from the two trap system. We can see that the values for the single trap are lower. We can expect the same pattern as we increase the number of logical qubits. This result verifies that a single ion chain quantum system is less forgiving than an optical interconnect quantum system with respect to α for the chain-specific error.

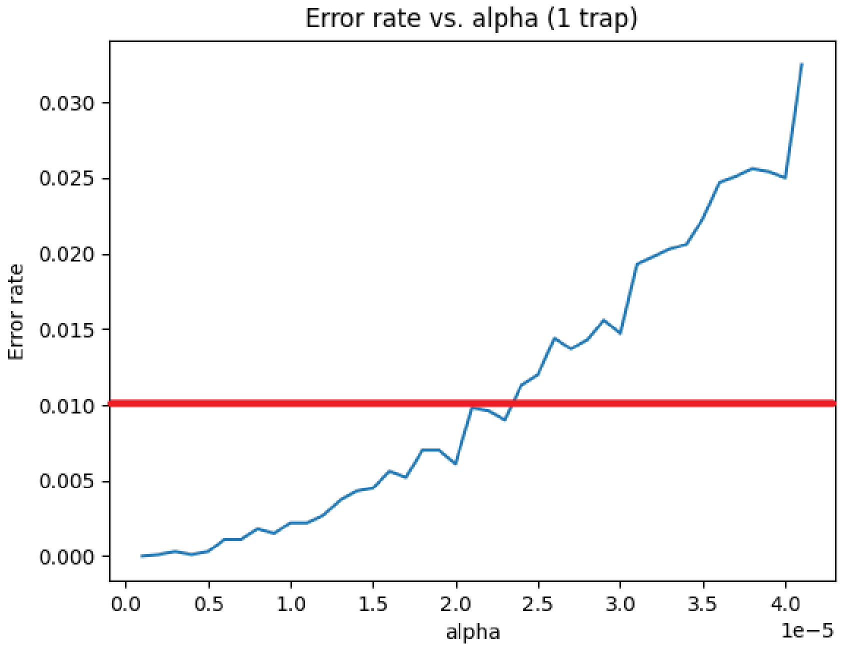

Figure 11: Logical qubit performance in noisy environment

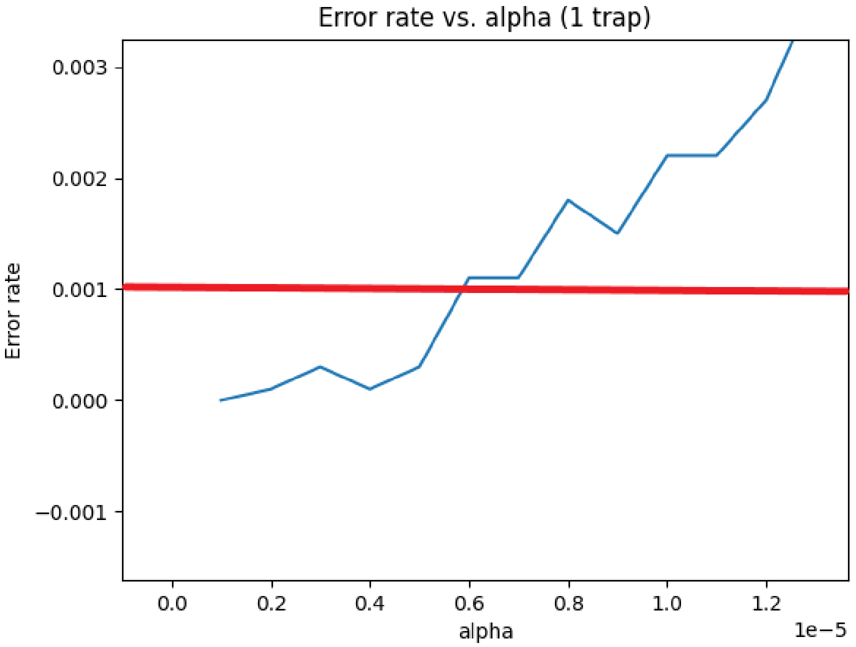

Figure 12: Logical qubit performance in noisy environment

5. Conclusion and Future steps

5.1. Conclusion

Our upper bounds for α and P_{dark} were gathered from a model of a simple optical interconnect quantum system with two ELU's. However, by design of the quantum computer, we may project to a system with several ELU's without significant changes to these upper bounds.This is the intended strength of this design. For that reason, our upper bounds for α and P_{dark} for 99% and 99.9% logical CNOT success rate also apply to a more generalized, much larger system, and could potentially be significant for those interested in developing early versions of this quantum computer.

5.2. Limitations

Our upper bounds, particularly the one for α, are most likely too low. This is because we do not consider the different possible qubit orientations within each ion chain in this project. Consider the logical CNOT that we apply, which consists of 9 cnot gates amongst 18 qubits distributed in two different traps. Depending on where the qubits are in the chain, the distance calculations for error would change, and thus the chain-specific error probability would change. However, we merely use an arbitrary orientation. This is a somewhat major limitation.

Another possible issue is that although a calculation for ’fidelity’ using the final states of qubits is often used in quantum computing literature, we instead use success rate of a logical qubit state or logical gate which uses the final measurements and is less informative . One reason this choice was made because our operations were limited to Clifford gates, so our qubits were never in a complex state. Thus, not as much information was lost from calculating success rate. Another reason was because it was unclear to me whether a value for logical fidelity would be representative of how well our logical qubit or gate has performed. As a result, it is possible that our calculations are not as representative of true performance as they could have been.

5.3. Future Work

As described in the limitations section, we consider only one, arbitrary orientation of the qubits in our quantum computer model. One direction we should take is to implement a mapping algorithm for the qubits that would allow us to find the highest possible upper bound of α given a circuit and logical qubits we want to run it on. Similar work has been done for QCCD computers here [6]. Another direction we should take is to repeat the above experiments using a different surface code such as Bacon-Shor-13 to encode the logical qubits. We could then make meaningful comparisons between systems with ELU’s encoded with Surface-17 and systems with ELU’s encoded with Bacon-Shor-13, then work out advantages and disadvantages of the two.

Finally, in the design paper [1], it mentions that each ELU is control of 50-100 qubits. QCCD quantum computers were designed to push trapped ion computers to that scale - therefore, it could be helpful to repeat our experiments on a model that makes each ELU a QCCD system rather than a single ion chain. If we were to go in this direction, we could make efficient use of tools developed in [6].

Appendix

Code

All relevant code is available in a public git repository.

`https://github.com/kalbert45/Surface-Codes`

Acknowledgements

I want to express gratitude to Prakash Murali and Dripto Debroy for taking their time to work closely with me on this Independent Work project. I would also like to acknowledge and thank my IW advisor, Margaret Martonosi, for leading the IW seminar and guiding me through my work. Thanks to these individuals I have had an enjoyable IW experience.

Honor Code

I pledge my honor that I have not violated the Honor Code.

Albert Kim

References

[1] K. R. Brown, J. Kim, and C. Monroe, "Co-designing a scalable quantum computer with trapped atomic ions," 2016. Available: https://doi.org/10.1038/npjqi.2016.34

[2] D. Debroy, "Connectivity project notes," 2020, unpublished.

[3] H. Kaufmann *et al.*, "Fast ion swapping for quantum-information processing," *Phys. Rev. A*, vol. 95, p. 052319, May 2017. Available: https://link.aps.org/doi/10.1103/PhysRevA.95.052319

[4] D. Kielpinski, C. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," *Nature*, vol. 417, no. 6890, pp. 709–711, Jun 2002. Available: https://doi.org/10.1038/nature00784

[5] N. M. Linke *et al.*, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017. Available: https://www.pnas.org/content/114/13/3305

[6] P. Murali *et al.*, "Architecting noisy intermediate-scale trapped ion quantum computers," 2020.

[7] A. Peter, D. Loss, and J. R. Wootton, "Benchmarks for approximate cnots based on a 17-qubit surface code," 2018.