Nicholas Jaber

April 27th 2022

ECE 621

Dr. Kenneth Brown

Quantum error correction analysis: CSS 9-qubit, Laflamme 5-qubit, repetition 5/9-qubit

While quantum computers may serve as an impactful evolution in NP computation, currently errors outstrip efforts to compute. Quantum computers are highly susceptible to errors, because they are a form of analog computation, meaning that errors aggregate more easily than in classical computaters [1]. In classical computers, a voltage threshold prevents small errors from aggregating. Additionally, the easy duplication of data allows for simple comparisons to identify and correct errors. These computers are very useful for serial tasks, but fail to scale adequately for a number of key emerging fields. Quantum computers fix this scaling issue, but must employ a host of extensive hardware and error correction techniques to limit and mitigate the increased presence of errors. Hardware and error correction teams work in parallel to develop reliable quantum computers, however in this paper we will focus on simulation and analysis of basic quantum error correction codes.

Quantum error correction codes use quantum information theory to inform a quantum circuit design, which mitigates the damage of errors. Depending on the type and specific implementation errors may be dominated by one or many different models. In this paper we will consider: thermal relaxation, amplitude damping, phase damping and depolarization error models. These error models were developed by Qiskit to mimic common error channels seen in real world quantum computers [1,2]. Thermal relaxation is the likelihood that over time a |1> state decays into a |0> state [1,2]. Amplitude damping is a less than π rotation on x and y gates such that repeated application causes the state to approach an equatorial state on the Bloch sphere [2]. Phase damping is similar to amplitude damping, but applied to z gates [2]. Depolarization is a series of small z rotations applied over time [2].

I am particularly interested in simulating small error correction codes that can correct any single pauli gate error. By comparing them against repetition codes with the same number of qubits, one could identify how much more effective advanced error correction codes are. I planned to check the (5,1,3) Laflamme code against the 5 qubit repetition code, non-corrected circuit, and the noiseless circuit. Additionally, I planned to check the (9,1,3) CSS (Calderbank Shor Steane) code against the 9 qubit repetition code, non-corrected circuit, and the noiseless circuit. Unfortunately, for both of these cases, the repetition code and non-corrected circuit implementation failed; more on this later. I chose the 5 and 9 qubit models to demonstrate their relative effectiveness. The 5 qubit code has the minimum number of qubits for single pauli correction, and the 9 qubit code, which is a standard in single logical qubit error correction [3,4]. CSS is the standard, because it can correct at a minimum 1 phase flip and 1 bit flip, while Laflamme has a minimum 1 phase or 1 bit flip error [4,5].

I modeled my CSS circuit on the circuit design shown in fig. 1c. This model is derived from the component single bit flip and single qubit phase repetition codes, shown in fig. 1a-b [3]. I modeled my Laflamme circuit on the design shown in fig. 1d-e. I simulated a number of different error probability factors. In my first attempt, I simulated an error probability factor starting at 1 and decreasing by a factor of 1.25 for 62 iterations, until the rate was lower than 10^{-6}. I simulated running each of these iterations 10^{6} times, referred to as 10^{6} shots. On my next attempt, I simulated an error probability factor starting at 1 and decreasing by a factor of 10 for 7 iterations, until the factor was lower than 10^{-6}. I simulated these circuits with 10^{8} shots. I chose a small step size low shot and large step size high shot methods to identify any shot size dependent effects. These conditions were picked somewhat arbitrarily to maximize usage of available computational resources and to seek a sufficiently large sample size to attempt to observe a correlation between error probability factor and divergence. I measure divergence as the number of simulations in the error corrected model which do not match the number of the noiseless model. Current quantum computers have an estimated minimum of a 10^{-3} error rate. I chose the range of error probability factors to emulate quantum computers 3 orders of magnitude better and worse than the current hardware minimum.

The data structure that we developed is as follows. control.py stores the main commands and refers to the various component processes. Each of these components employs functions stored in common_func.py. The components are noiseless.py, CSS9.py, and Laflamme5.py. noiseless.py generates circuits, applies a series of arbitrary gates, and simulates running this circuit many times. The count data is collected and stored in a dictionary which is passed up to control.py and then written to a txt file. control.py commands process.py to read data from the txt files and generate graphs. A similar design is employed for the remaining constituents, the major difference being the application of noise models, and specific error correction encoding and decoding.

Through analysis of the figures generated in analysis.py, we have generated fig. 2. As seen in fig. 2, I was unable to find a trend in these graphs despite their large shot sizes. I was surprised to see how error model and error probability factor agnostic my results were. I intended to effectively distribute many types of arbitrary gates to limit error model dependence, but I am skeptical of the lack of error probability factor dependence. One can identify a consistently much higher divergence in 5 qubit Laflamme code than in 9 qubit CSS code. If the CSS and Laflamme error correction schemes were similar in effect, the magnitude of fig. 2b,d,f,h should be $2^{4 \text{ qubits}}$ = 16 times higher than fig. 2a,c,e,g. I expected the 9 qubit code to more effectively suppress error, but I did not anticipate a nearly 10^5 decrease in error suppression when switching to 5 qubit Laflamme. Even when accounting for the different number of qubits between these schemes, CSS is still 10^3 - 10^4 times more effective. This is seen by comparing the magnitude of the state change population for fig. 2a-b and fig. 2e-f.

With the addition of sufficiently more advanced data structures and computational power, I would be able to simulate much higher shot counts. Doing so, we would ideally be able to identify a decrease in the correctability of the model as the error probability factor increases. I ran into some issues implementing the Qiskit model for repetition codes. I initially planned to implement repetition codes in a similar manner as CSS and Laflamme. I also attempted to implement a noisy model to indicate how much noise was affecting the system. Both of these attempts failed for an unknown reason. They would produce the exact same result as their respective noiseless channels regardless of noise model.

I expected to see similarly poor performance when the error probability factor was much higher. Based on the results shown in fig. 2, there seems to be no case in which these error correction schemes operate with similar effect. One should always employ CSS rather than Laflamme. I hoped to identify if a 9 qubit repetition code would outperform a 5 qubit Laflamme code. This would've shown me whether or not the number of qubits is the predominant factor in code efficacy. This will remain an avenue for future investigation.

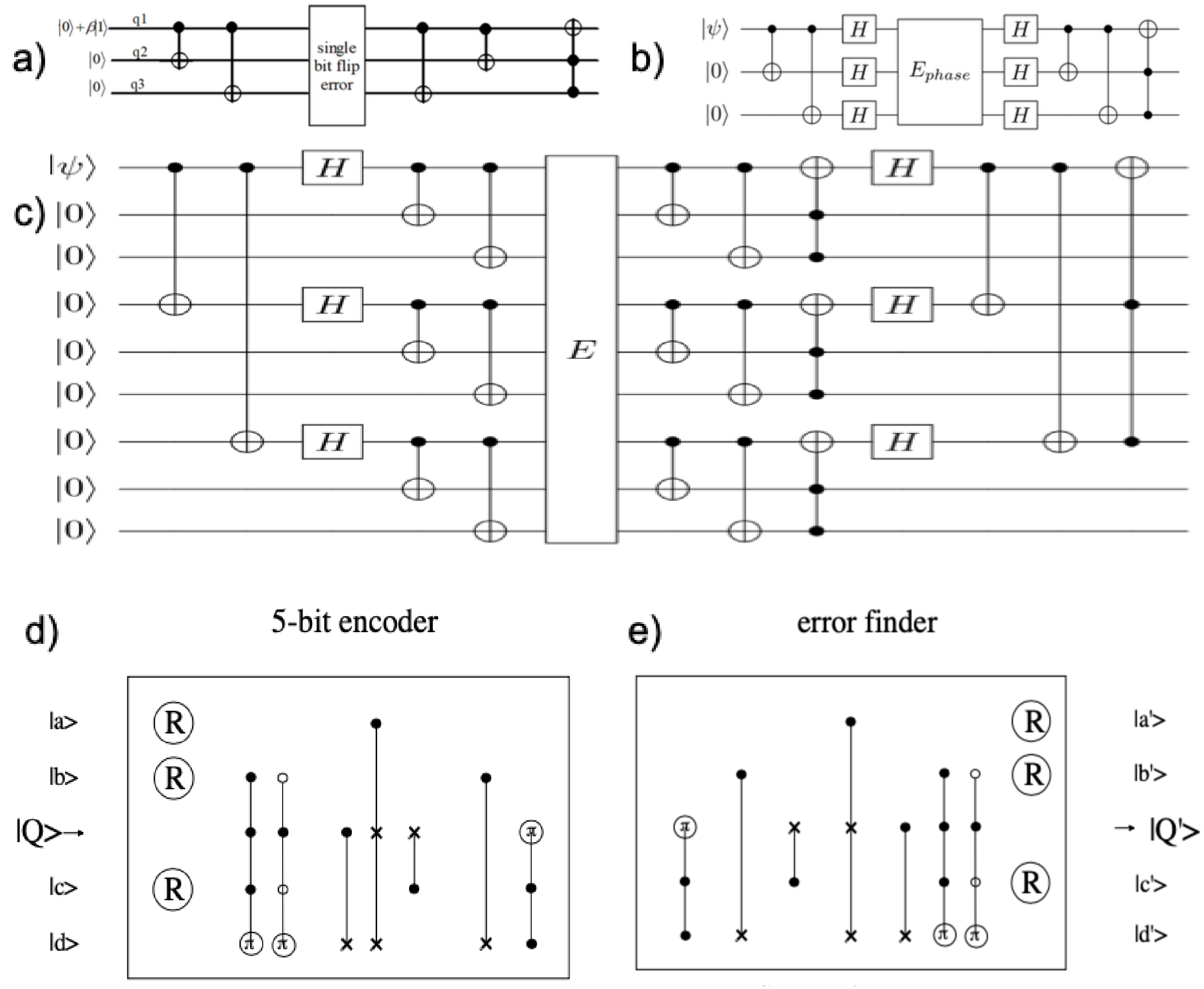

Figure 1 a-c uses traditional quantum circuit diagram design [3]. d-e R represents Hadamard gate, π represents a pauli z with control on full dots and inverse control on empty dots, x represents pauli x with the same control notation [4]. a) Encoder and decoder for 3 qubit repetition code, protects against single bit flip error [3]. b) Encoder and decoder for 3 qubit repetition code, protects against single phase error [3]. c) Encoder and decoder for 9 qubit CSS code, protects against at least 1 phase and 1 bit flip [3]. d) Encoder for 5 qubit Laflamme code, protects against at least 1 phase or 1 bit flip [4]. e) Decoder for 5 qubit Laflamme code [4].

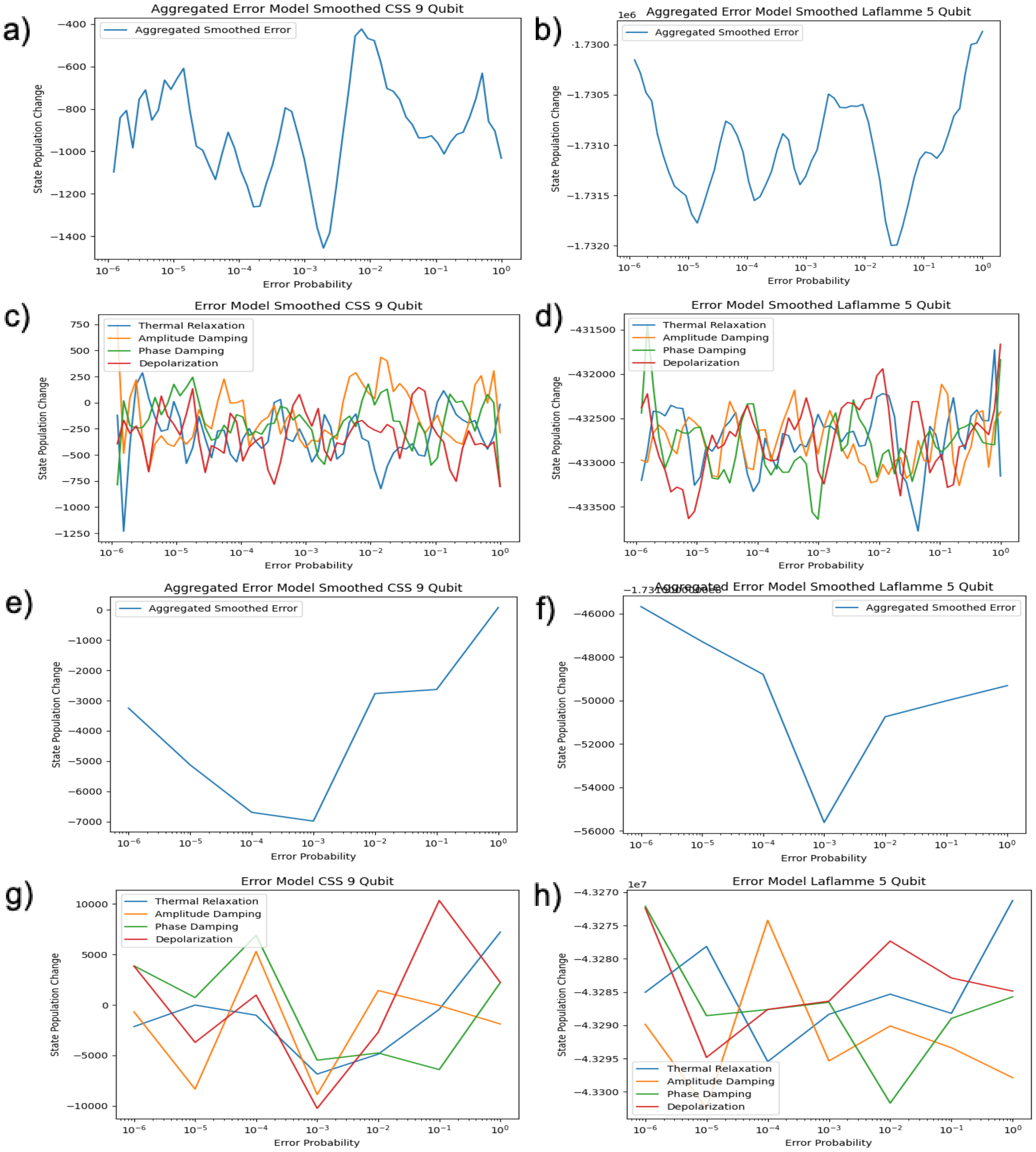

Figure 2 First column a,c,e,g corresponds to CSS 9 qubit code. Second column b,d,f,h corresponds to Laflamme 5 qubit code. First two rows a-d are a low shot number small step size model. Third and fourth rows e-h are a high shot number large step size model. First and third rows a-b,e-f show the aggregated (summed) and/or smoothed (averaged over 5 values) divergence from each error model. Second and fourth rows c-d,g-h show the individual smoothed divergence from their error model.

Bibliography

1. Krantz, P. *et al.* A quantum engineer's guide to superconducting qubits. *Appl. Phys. Rev.* **6**, 1–67 (2019).

2. Building Noise Models — Qiskit 0.36.1 documentation. Available at: https://qiskit.org/documentation/tutorials/simulators/3_building_noise_models.html . (Accessed: 26th April 2022)

3. Behera, B. K. & Panigrahi, P. K. The first three-qubit and six-qubit full quantum multiple error-correcting codes with low quantum costs. (2019). doi:10.13140/RG.2.2.18542.77129

4. Laz, J. P. & Zurek, W. H. Perfect quantum error correcting code. *Phys. Rev. Lett.* **77**, 198–201 (1996).

5. Cafaro, C. & Mancini, S. Quantum Error Correcting Codes and Correlated Noise Errors. *Computing* **374**, 2688–2700 (2010).