NEXUS Qubit Analysis: Error Rates and Efficiencies

Grace Wagner,[1] Grace Batrud,[2] Dan Baxter,[1,2] and Arianna Colón Cesaní[2]
[1] *Fermi National Accelerator Laboratory*
[2] *Northwestern University*

(Dated: 26 April 2024)

Correlated qubit errors are a hindrance to quantum computing, but when using quantum chips as a particle detector, they can be a sign of energy deposits. Using a four transmon qubit chip in the NEXUS fridge at Fermilab, we have taken data to better understand the impact ionizing radiation has on the correlated errors. By running the experiment underground in the MINOS cavern, we reduce the amount of cosmic ray muons that can cause errors on the qubit (in addition to gammas from ionizing radiation), thus controlling the background of the experiment. To analyze this data, we need a error detection code that is robustly tested with understood efficiencies. In this work, we describe the process of getting these efficiencies using a rolling χ^2 error detection code.

FERMILAB-PUB-24-0182-STUDENT

I. INTRODUCTION

As described in Wagner *et al.*[1], correlated errors that are detrimental in quantum computing can be used in particle detection to identify energy deposits. An error on a quantum chip is any loss of information, either by decoherence or dephasing. We investigate the cause of these errors, specifically looking at the role gammas from ionizing radiation. This work focuses on the error detection code used to identify errors in the data.

A qubit error is any loss of information from a qubit chip. Unlike a classical chip, which can be in either the ground state $|0\rangle$ or excited state $|1\rangle$, a qubit can be in either state or neither. In addition to these states, there is also a phase aspect of the qubit state. A decoherence error is a loss of state when the qubit relaxes from the excited state, $|1\rangle$, to the ground state, $|0\rangle$. The time of relaxation is referred to as T_1.

There are also charge burst errors. These errors are caused by trapped charges in the bulk substrate altering the electric field of the qubit island, resulting in a loss of state. These charge burst errors are what we are investigating. In the data, these errors appear as 'jumps', due to the sudden discontinuity of the data. Thus, we will refer to these errors as 'charge jumps'.

Wilen *et al.*[2] stipulates that ionizing radiation and cosmic ray muons that are incident on the chip is one cause of these errors. In this work, we look to further quantize the effect of ionizing radiation and cosmic ray muons. To do so, we run the same experiment with the same four qubit chip from Wilen *et al.*[2] 100 meters underground, drastically reducing the cosmic ray muon rate.

II. METHODS AND DATA

In this section, we will explain the setup of the experiment and a description of the facilities, as well as how we created simulated data for testing the code, as well as details on how the point by point error detection code works.

This experiment is a continuation on the work done in Wilen *et al.*[2]. We use the same 4 transmon qubit chip. The goal of our experiment is to better understand the effect (or lack of effect) of ionizing radiation gammas and cosmic ray muons on qubit errors. We do so by controlling the radiation environment in which the dilution fridge is running. The NEXUS fridge is located in the MINOS cavern at Fermilab, over 100 meters below the surface. Surrounding the dilution fridge is a large, movable, 4 inch thick lead shield that fully encloses the fridge. This work uses two radiation configurations of the shield, the first being the shield closed (SC) configuration, which has the lowest incident rate, and the other being shield open (SO). Due to the rock overburden, the cosmic ray muon flux is decreased by 99%. Wilen *et al.*[2] suggests that the qubit errors are due to either muons or gammas. As we have significantly reduced the muon rate, we can investigate the role of gammas.

A. Simulated Data

To understand the efficiency of the error detection code, we use simulated Ramsey tomographic sweeps. In Ramsey tomography, a charge bias is applied to the qubits and measure the state population as a function of charge bias. We simulate this by generating a uniform distribution of jump sizes (the size of the jump is defined by the difference in the phase before the jump occurs, and the new phase after the jump) in the range of 0.01 e to 0.5 e. We also randomly select which sweeps will have jump(s) injected and where in the sweep the jump(s) are injected.

To inject a jump into the tomographic sweep, we start with a jump-less period template sweep, which is specific to each of the four qubits. Each tomographic sweep consists of 80 points, where, in a jump-less sweep, each point corresponds to 0.0125 e of bias charge. To inject the jumps, we up-sample the template to have 1265 points. In other words, for each 'real' data point, there are 16 interpolated points. At a randomly selected index in the up-sampled sweep, the phase is offset according to the jump size, and the sweep is completed with the new phase. If multiple jumps are injected into one sweep, then the process repeats for each injected jump. After all jumps are injected, the sweep is down-sampled back to 80 points.

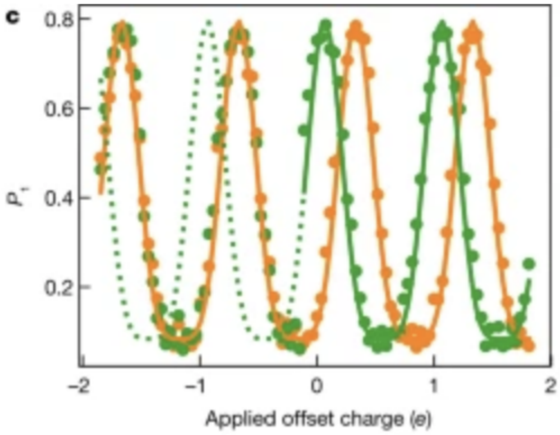

FIG. 1. Example of a Ramsey tomography sweep from Wilen *et al.*[2]; On the y axis is the Ramsey amplitude vs offset charge on the x axis. The orange points demonstrate a sweep with no errors, which is seen through the orange points following the orange line for the duration of the sweep. The following sweep (shown in green) has an error in the middle of the sweep. The green points do not match the dotted green line prior to error, instead only matching the solid green line after the error.)

B. Pt. by Pt. Error Detection Code

The point by point error detection code uses a rolling χ^2 to determine how well the data matches the template; when the data deviates from the template, resulting in a high χ^2 value, a jump is detected.

The first step is taking the χ^2 (Equation 1) of each of the 80 real points and each interpolated point of the template, as well as taking the combined χ^2_n (Equation 2). The combined χ_n^2 is the average χ^2 across the extended indices. From the combined χ_n^2, the minimum is selected. If the minimum χ_n^2 is greater than the least χ^2 threshold, then a jump is found. Before it is confirmed to be a jump, there are two more quality checks to confirm the jump as a true positive.

$$\chi^2 = \frac{x_i - x_\phi}{\sigma_\phi} \tag{1}$$

$$\chi_n^2 = \frac{1}{n}\sum_i^n \chi^2[i,] \tag{2}$$

Due to the rolling χ^2 nature of the jump finder, the code needs to have enough points to accurately find the χ^2 of the data to the template. Jumps at the beginning of the sweep are more difficult for the code to find, as there are fewer points than ideal to understand how well the data fits the template. To avoid a high instance of false positives at the beginning of a sweep, jumps are tagged after a set number of interpolated indices (out of 1265 points) have been checked in the code. This limit is referred to as the Jump Num. Limit in Table I.

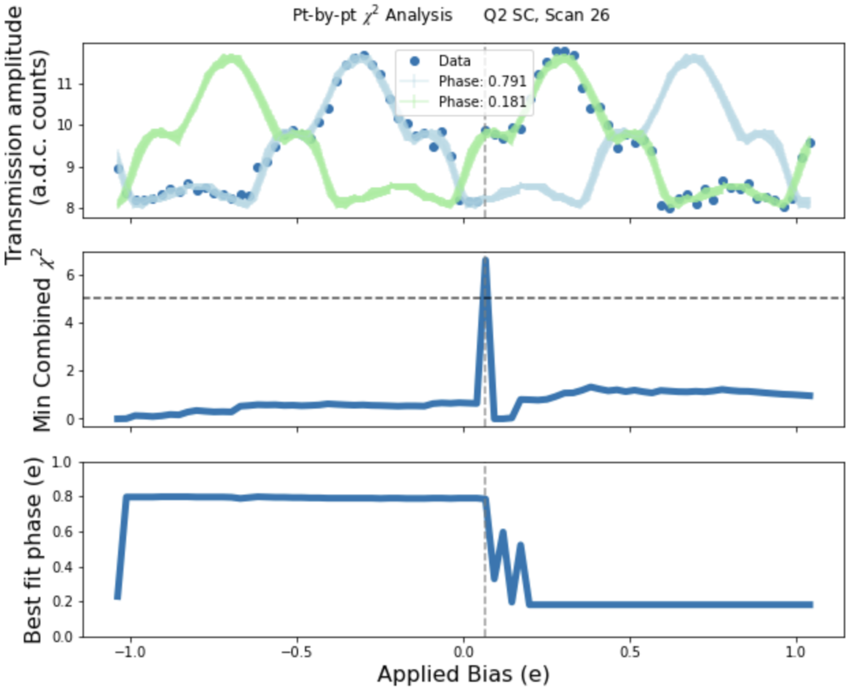

FIG. 2. An example of an error (also referred to as a jump) that is detected by our error detection code. In the upper plot, which shows the tomographic sweep, prior to the error, the data (blue points) follows the blue line. After the error, the data follows the green line. The middle plot shows the minimum combined χ^2. The lower plot illustrates the difference of phase before and after the error.

Once a jump is found, the phase before the jump is noted and saved, as well as the phase following the jump. Another check to ensure the jump tagged is real is to confirm that the pre-jump phase subtracted from the phase after the jump is greater than the Phase Dif. Limit parameter. The pre-jump phase is determined by calculating what phase of the template best matches the sweep before the jump is detected. We use the Back Off

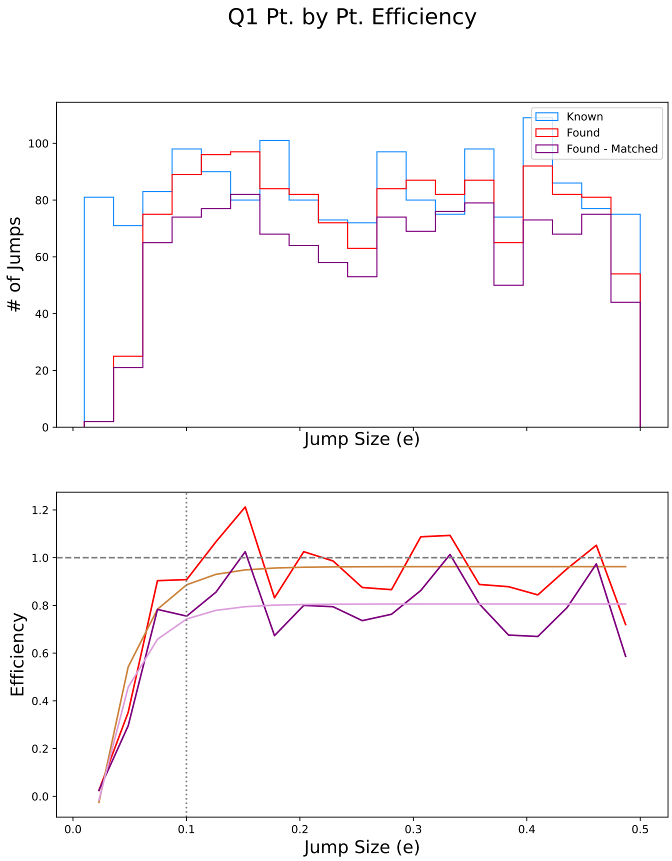

FIG. 3. Example of an efficiency plot for a single set of simulated data on qubit 1; The upper plot is a histogram that shows how many jumps were injected (blue), how many jumps were found in total (red) and how many jumps were found and matched (purple). The lower plot divides the number of found jumps by the number of injected jumps per bin. This is once again done for all found jumps (red) and for matched jumps (purple). The error detection method utilized is ideal for jump with a size greater than 0.1 e, so we are interested in the efficiency above that threshold, which is denoted by the grey vertical dotted line.

parameter to determine when before the jump we should determine the pre-jump phase. We calculate the post jump phase by selecting the phase of the template that best fits the phase of the sweep following the jump.

After we have run the simulated data through the error detection code, we want to show how well the error detection code works as a function of jump size. To do so, we bin the jumps found and jumps injected by jump size, and divide the number of found jumps by the number of injected jumps per bin. We then fit an analytic curve to the resultant plot to understand the efficiency for jumps with a size greater than 0.1 e. This is shown in Figure 3.

C. Index Matching

After running the error detection code, we checked that the jumps found were detected within a reasonable time frame after the jump was injected. Using a list of the sweep indices where jumps were found and another of where jumps were injected, we subtract the found

TABLE I. Parameters for each Qubit for Pt. by Pt. Error Code

	Q1	Q2	Q3	Q4 SC	Q4 SO
Least χ^2 Threshold	5	5	7	10	8
Jump Num. Limit (e)	0.035	0.035	0.044	0.037	0.041
Phase Dif. Limit (e)	0.053	0.053	0.053	0.057	0.046
Back Off (e)	0.003	0.003	0.007	0.004	0.004
Combined χ^2 Cut	5	5	3	3	5

indices from the known (injected) indices. The resulting list is used to remove any false positives (pairs where the difference is negative, meaning a jump was detected before any were injected) or where too much time has passed between when the jump was injected and found. The size of this window varies by qubit. The purple histogram and curve in Figure 3 includes just these correctly tagged jumps.

III. ANALYSIS

To calculate the efficiency of the error detection code, we repeat the whole process, from the start (creating the simulated data), though the end (index matching the found jumps to the injected jumps and fitting the analytical curve) 75 times for each qubit, and twice for qubit 4.

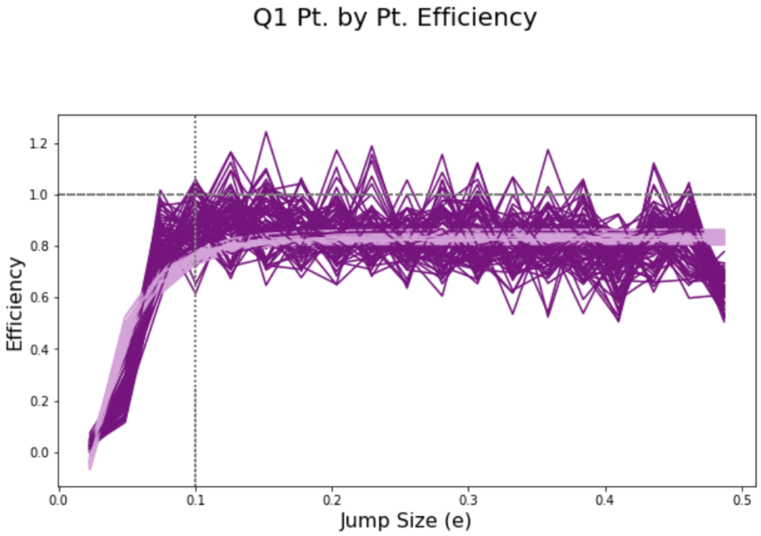

FIG. 4. Plot of all of the efficiency curves from 75 runs with qubit 1. The Gaussian in Figure 5 are fit to the distributions of the violet lines for each respective qubit.

We use two templates for Q4 due to the quality of the data. The amplitudes of the SC data varies more than it does in other qubits; using this template leads to much lower efficiency. The SO (shield open) template is more representative of the data for the SO configuration. Using the best fit curve of the matched jumps (violet) from the lower plot of Figure 3 (referred to as the efficiency curve) for jump sizes greater than 0.1 e^2 of the found jumps divided by injected jumps for bins of jump sizes, we get the spread of the asymptotes of the best fits. This gives is the mean μ and standard deviation σ of the efficiencies. We then divide the raw qubit rates by the found efficiencies to get the actual rates. Uncertainties on the efficiency are then included as systematic uncertainties in the final rates but not reported here as the rate analysis is ongoing..

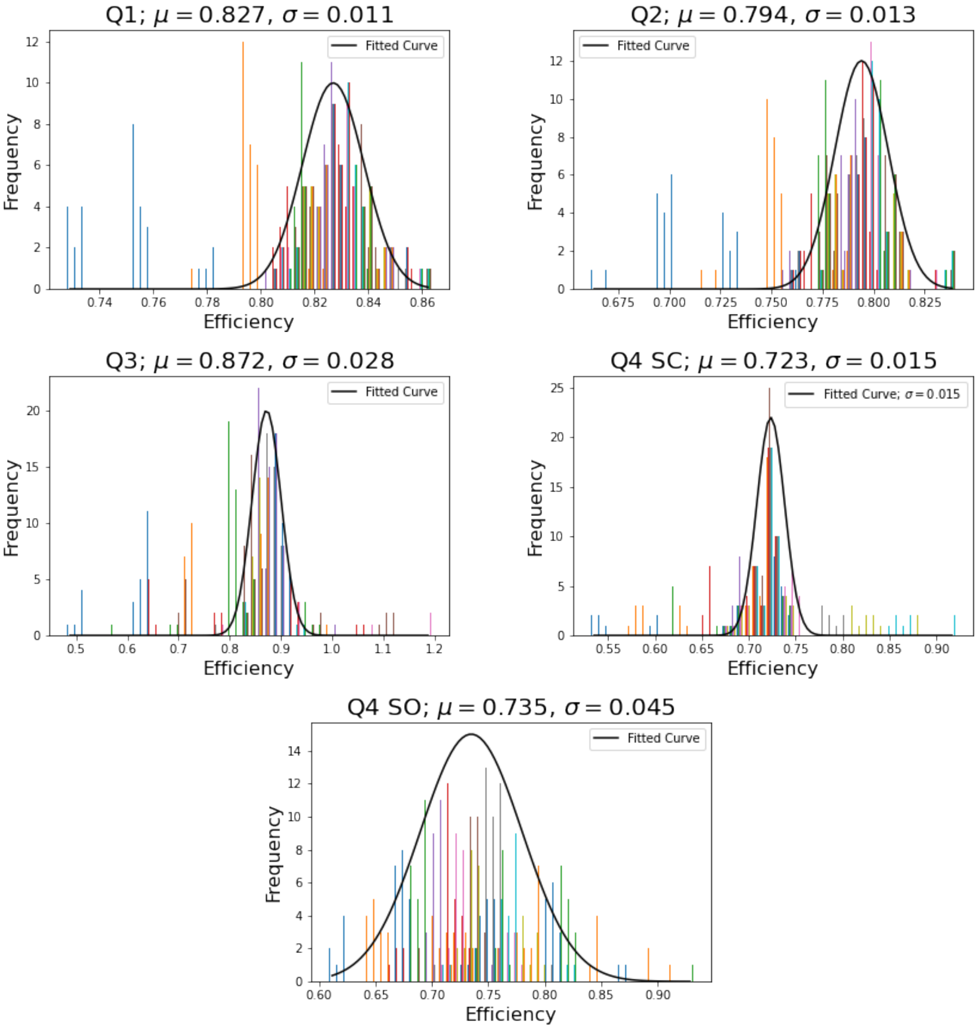

FIG. 5. Using the efficiency curves from all 75 runs for each qubit (for qubit 1, the violet lines from Figure 4), we then fit a Gaussian to the distribution of efficiencies of detecting jumps with a size greater than 0.1 e, to get the mean efficiency for each qubit, as well as the standard deviation.

TABLE II. Efficiency Adjusted Rates for each qubit

	Q1	Q2	Q3	Q4 SC	Q4 SO
Efficiencies	0.827 ± 0.01	0.794 ± 0.01	0.872 ± 0.03	0.723 ± 0.02	0.735 ± 0.05
Raw Rates (mHz)	0.17	0.15	0.17	0.12	0.51
Efficiency Adjusted Rates	0.20	0.19	0.16	0.16	0.51

IV. CONCLUSION

In this work, we have presented a method of detecting errors in Ramsey tomography sweeps, as well as understanding the efficiency of the error detection code. Each of the four qubits on the chip we used has unique characteristics, so using these efficiencies, we can then adjust the raw rates of qubit errors to reflect the true rates, as well as calculating the systemic errors for this method. The values in Table II are the efficiencies of the error detection code for errors with a size greater than 0.1 e and less than 0.5 e.

Future work will include a better understanding and modeling of the false positive rate as a function of jump size.

ACKNOWLEDGMENTS

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

[1] G. Wagner, G. Batrud, D. Baxter, A. H. Colón Cesaní, and E. Figueroa-Feliciano, "Testing the Jump Finding Code for NEXUS Qubit Analysis," .

[2] C. D. Wilen, S. Abdullah, N. A. Kurinsky, C. Stanford, L. Cardani, G. D'Imperio, C. Tomei, L. Faoro, L. B. Ioffe, C. H. Liu, A. Opremcak, B. G. Christensen, J. L. DuBois, and R. McDermott, "Correlated charge noise and relaxation errors in superconducting qubits," Nature **594**, 369–373 (2021).