NEXUS Qubit Analysis: Jump 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 hinderance 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. To analyze this data, we need a jump 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 jump detection code.

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 qubit (a quantum chip, as opposed to a classical chip and bit) is any loss of information. 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 jump detection code works.

This experiment is an 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 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). Between the lead shield and underground location, the cosmic ray muon rate is decreased by a factor of . 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 jump detection code, we use simulated Ramsey tomographic sweeps. In Ramsey tomography, a charge bias is applied to the qubits and the phase is read out. 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.

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 scan, 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 scan, 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.

B. Pt. by Pt. Jump Detection Code

The point by point jump 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^2_n is the average χ^2 across the extended indices. From the combined χ^2_n, the minimum is selected. If the minimum χ^2_n 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.

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

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

Due to the rolling χ^2 nature of the jump finder, the code needs to have enough points to accurately find the χ^2. Jumps at the beginning of the scan are more difficult for the code to find, as there are fewer points that ideal to understand how well the data fits the template via χ^2s. 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.

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 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 jump detection code, we want to show how well the jump 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 (add image) to understand the efficiency for jumps with a size greater than 0.1 e.

C. Index Matching

After running the jump 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

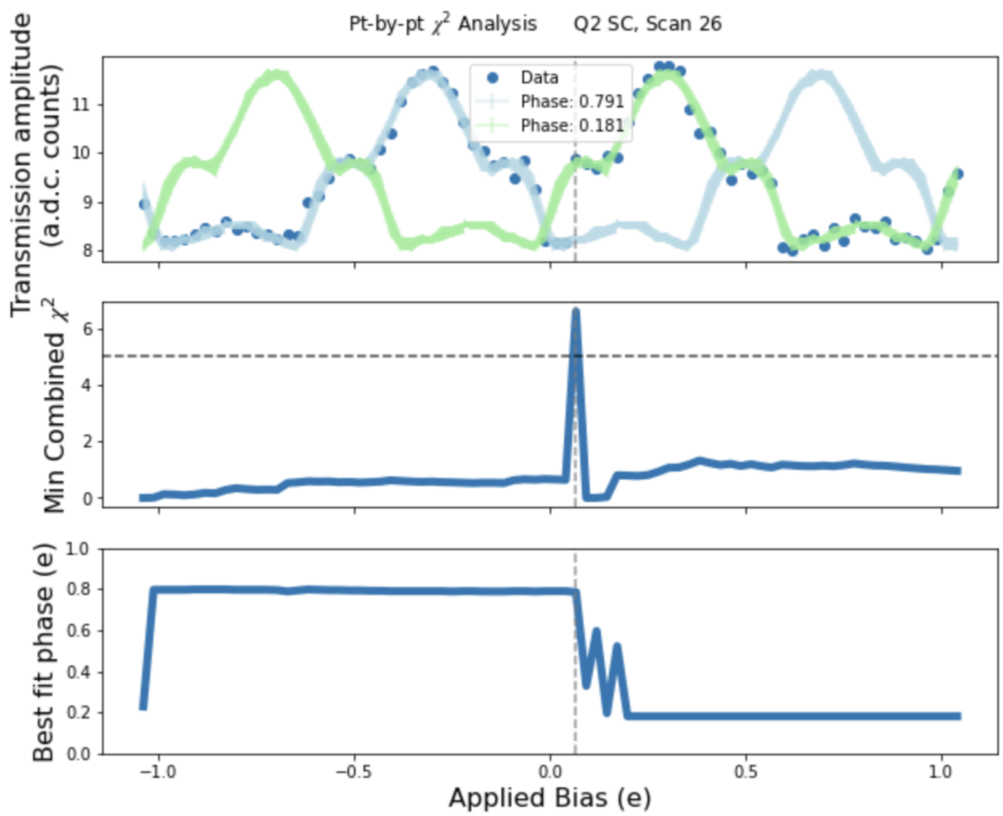

FIG. 1. Example of a found jump using the Jump Detection Code

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

	Q1	Q2	Q3	Q4 SC	Q4 SO
Least χ^2 Threshold	5	5	7	10	8
Jump Num. Limit (e)	0.017	0.017	0.020	0.017	0.018
Phase Dif. Limit (e)	0.025	0.025	0.024	0.025	0.020
Back Off (e)	0.0016	0.0016	0.0032	0.0016	0.0016
Combined χ^2 Cut	5	5	3	3	5

where jumps were found and another of where jumps were injected, we subtract the found 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 (greater than 45 seconds).

III. ANALYSIS

To calculate the efficiency of the jump 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) 75 times for each qubit. Using the best fit 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 multiply the found efficiencies by the raw qubit rates to get the actual rates.

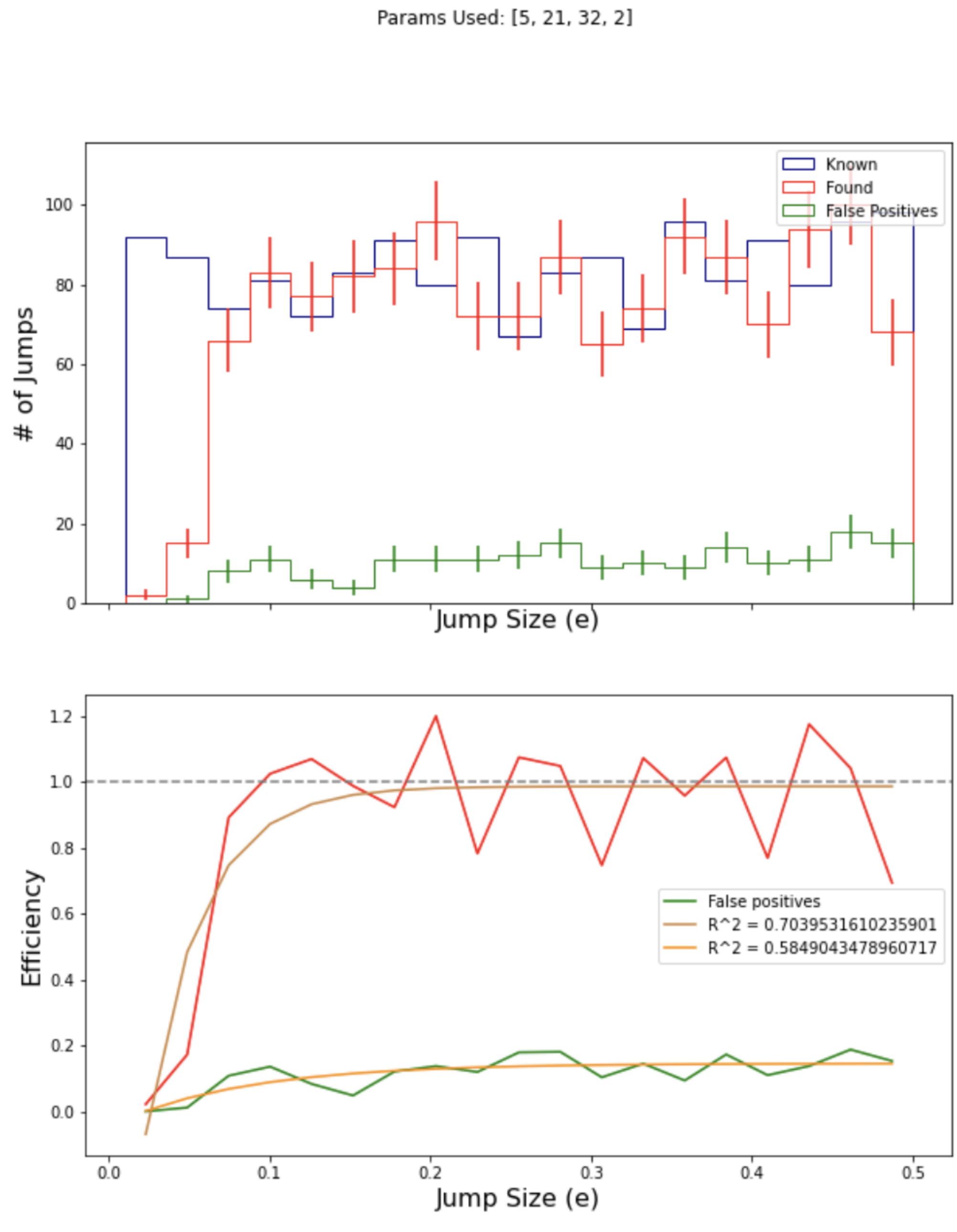

FIG. 2. Example of an efficiency plot for a single run

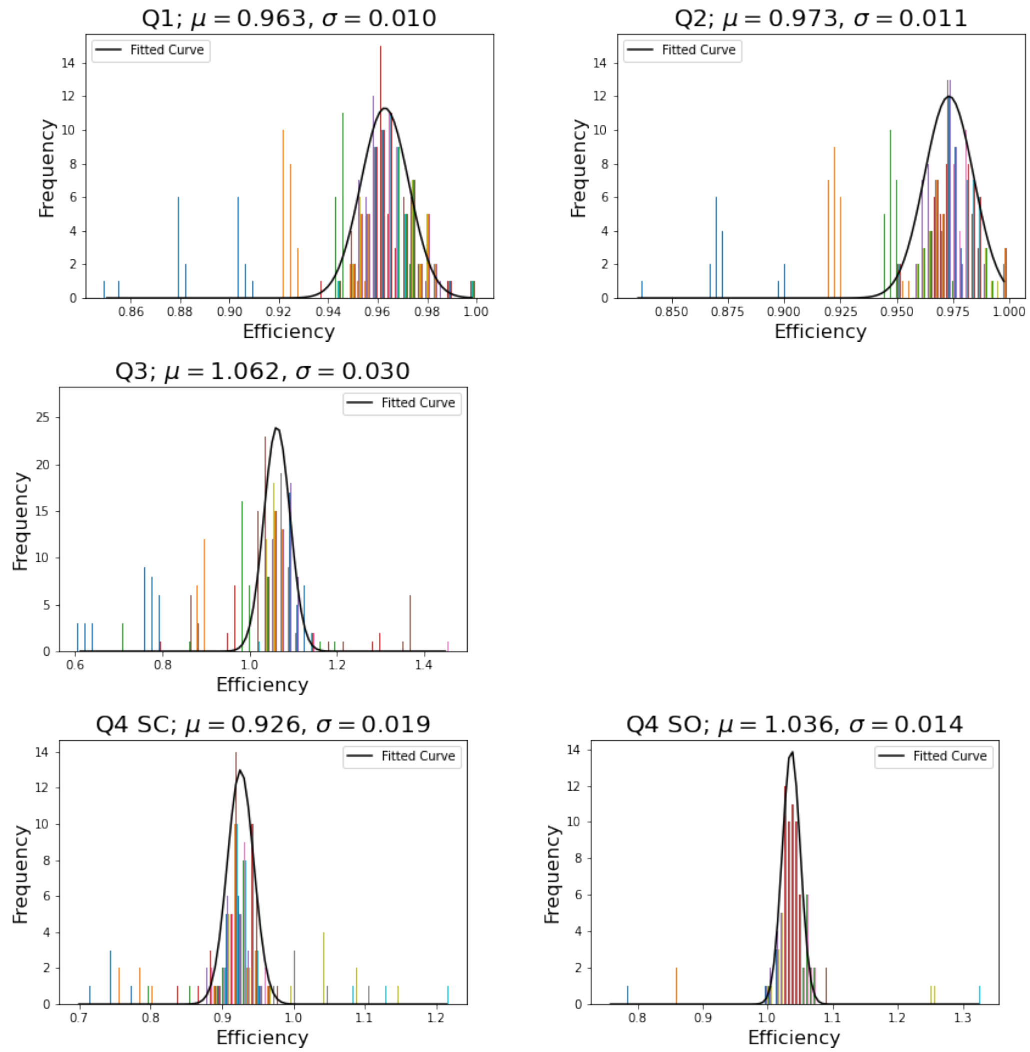

TABLE II. Efficiency Adjusted Rates for each qubit

	Q1	Q2	Q3	Q4 SC	Q4 SO
Efficiencies	0.963 ± 0.01	0.973 ± 0.01	1.062 ± 0.03	0.926 ± 0.02	1.036 ± 0.01
Raw Rates (mHz)	$0.165\pm$	$0.152\pm$	$0.177\pm$	$0.127\pm$	$0.416\pm$
Efficiency Adjusted Rates	$0.158\pm$	$0.147\pm$	$0.188\pm$	$0.118\pm$	$0.435\pm$

IV. CONCLUSION

In this work, we have presented a method of detecting jump in Ramsey tomography sweeps, as well as understanding the efficiency of the jump detection code. Using these efficiencies, we can then adjust the raw rates of qubit errors to reflect the true rates.

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).