# Testing the Jump Finding Code for NEXUS Qubit Analysis

G. Wagner,[1] G. Batrud,[1,2] D. Baxter,[1,2] A. H. Colón Cesaní,[2] and E. Figueroa-Feliciano[2,1]
[1)]*Fermi National Accelerator Laboratory*
[2)]*Northwestern University*

(Dated: 7 December 2023)

At Fermilab, the NEXUS dilution refrigerator is repeating the same experiment from Wilen *et al.* [1] underground to further investigate the effect of ionizing radiation and cosmic ray muons on qubit decoherence. This refrigerator is 100 m underground in the MINOS cavern to significantly lower the incident rate of cosmic ray muons, meaning the majority of incidents is due to ionizing radiation. To gain a better understanding of how the rate of ionizing radiation affects the qubits, we use 4 radiation configurations to collect data from the chip. In this work, we examine the efficiency of our jump detection code to find jumps of various sizes. We do so using simulated data with a known number of injected jumps of known sizes. These jumps are charge offsets of the qubit, thought to be due to quasiparticle poisoning. To test this, we run the jump finding code on an array of simulated data in which two main parameters, the smoothing factor and the threshold at which a charge offset is considered a jump, are changed to test an array of these parameters. We then perform a $\chi^2$ analysis to find the best combination of these two parameters.

## I. INTRODUCTION

In quantum computing applications, correlated errors are the end of the line. These errors cannot be rectified using error correction after the fact. Alternatively, when using quantum chips for particle detection, the opposite is the case, since correlated errors can be used 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.* [1] 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.* [1] 100 meters underground, drastically reducing the cosmic ray muon rate.

The errors we are investigating are correlated charge jump errors. There are two parts of this type of error: charge jump errors and correlated errors. To understand what is so interesting about the combination of these two, we will first look at the two parts separately. As previously stated, a qubit error is any loss of information that was stored on the chip. In a classical computer bit, which has two possible states, $\langle 0|$ and $|1\rangle$, an error occurs when the bit 'flips' incorrectly. There is only one axis for the bit to flip over, therefore there is only one kind of error. In a qubit, there are two axes, meaning more types of errors. We are interested in two of these: decoherence and dephasing. A decoherence error is when the qubit relaxes from the excited state, $|1\rangle$, to the ground state, $\langle 0|$. The second type of error mentioned, a dephasing error, is any loss of phase. A charge jump error (which is described in more detail in Section II) is related to decoherence or dephasing and are caused by the same underlying processes. Ultimately, we will use charge jump error analysis to further understand the effects of these processes.

Next, we will look at correlated errors. Correlated errors are errors that are related spatially and temporally (they are related by both time and space). It means that multiple errors are not independent of each other and therefore cannot be resolved with error correction. These correlated errors do make quantum computing difficult, but they are helpful when using qubits as particle detectors.

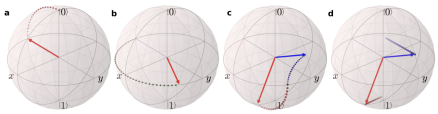


FIG. 1. Image from O'Malley 2016; Bloch sphere with arrows demonstrating the process of decoherence

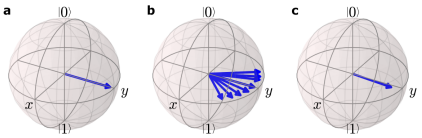


FIG. 2. Image from O'Malley 2016; Bloch sphere with arrows demonstrating the process of decoherence

McEwen *et al.* [2] explores the impact cosmic ray muons have on qubit chips, showing how errors can propagate out after a muon passes through the chip. Specifically, McEwen *et al.* [2] shows that this propagation of errors can be seen in time and space, that the error rates 'ripple' outward. It is because of this ripple that we can use qubits in particle detection. These 'ripples' happen when quasiparticles are incident on the qubit. An example of a quasiparticle is a phonon, which is the quantization of vibrations, much like a photon is the quantization of light. The difference between a phonon and a photon is that a photon is a vibration in the electric field, whereas a phonon is a vibration of particles. Another example of a relevant quasiparticle is a broken Cooper pair electron. This type of quasiparticle is relevant since we are using superconducting qubits. A Copper pair is a pair of electrons that are bonded at extremely low temperatures. The presence of these pairs of electrons give superconductivity its characteristic, extremely low resistance. When these pairs of electrons break, the newly separated electron is now a quasiparticle.

Future work that intends to use qubit errors will require a robustly tested jump detection code that is able to accurately

| Radiation Configuration | Rate (errors second$^{-1}$) |
|---|---|
| Shield Closed, No Source | 7.211 |
| Shield Open, No Source | 7.439 |
| Shield Closed, Barium - 133 | 7.875 |
| Shield Closed, Cesium - 137 | 8.986 |

TABLE I. The four radiation configurations used, listed in order from lowest rate to highest rate. The lead shield is 4 inches thick and surrounds the chip on all sides. These values were found with an older jump detection code, where any size charge offset is considered a jump. The jump detection code explored in this work looks to improve these numbers.

discern between small jumps and noise in the data. This works introduces a method to verify the efficiency of jump detection codes.

## II. METHODS & DATA

A charge jump is a measurement of a change in the electric field near the qubit island due to the presence of extra charge. We observe this in our experiment by scanning over an applied voltage at the qubit and noticing a discontinuity in this behavior over the course of the scan. This is different from a decoherence error, but the two rely on the same underlying processes. Wilen *et al.* [1] characterizes the 4-qubit chip we are working with. They start by preforming Ramsey tomography simultaneously on the four qubits to get a time series of fluctuating offset charge. Wilen *et al.* [1] examines charge jumps in qubits that are spatially correlated. These correlated jumps are caused by $\gamma$-ray absorption in the substrate, which releases phonons in the substrate, which impacts multiple qubits. Absorption of cosmic ray muons can also have this effect, but by conducting this experiment underground, we drastically reduce the muon rate. We also use a lead shield and two radioactive sources to further quantify the effect of $\gamma$-rays on the qubits.

### A. Real Data

To investigate the impact of quasiparticles on the qubit and isolate the chip from background radiation, the experiment is done in the NEXUS clean room in the MINOS cavern at Fermilab, more than 100 meters underground. In the dilution fridge is a chip with four transmon qubits, with a different distance between each the qubits. This chip is the same one used in Wilen *et al.* [1]. We collect $T_1$ measurements using Ramsey Tomography from the qubit chip while the dilution fridge is in four different radiation configurations, consisting of a lead shield being open or closed and the presence (or lack) of a radioactive source. The configurations and the radiation rate are listed in Table I, in descending order.

### B. Simulated Data

To test the jump detection code, we used simulated data. To do so, we start by making a no jump template using real data from the shield closed, no source configuration. First, we up-sample the number of points in each sweep, from 80 to 1265 points. We then take the mean across the first 20 voltage sweeps (after confirming there are no jumps) to get a jump-less template. Once we have the jump-less template, we can inject jumps. Since we are interested in testing how efficiently the jump finding code detects jumps of many sizes, we inject the maximum number of jumps, once every other scan (since the detection code requires a jump-less scan after a detected jump to 'reset' the jump-less template). The location in the voltage sweep where the jump is injected is randomized to test the code's ability to find jumps at any location in the voltage sweep.

### C. Jump Detection Code

First, we either make or find a no jump template. If we make a no jump template, the process is the same as making the no jump template for the simulated data. If we find a no jump template, we start examining the first voltage sweep(s) to ensure there are no jumps. If there are none, we use this scan. This code is currently set up so that it needs at least 1 no jump scan at the start and a no jump scan after a jump to get a new template; next, subtract the no jump template from the data, which will result in a view of the scan that isolates the noise and jumps (anything that is different from the no jump scan) each of the 4 qubits have a distinctive scan pattern, so we can use this pattern across all the scans for each respective qubit. Also, we can use the shield closed no source configuration to get the starting no jump scan for each qubit. After subtracting the template from the real scan data, we apply a smoothing filter. This is the first of the two tunable parameters we're interested in.

We smooth the data using a 1D uniform filter. The smoothing parameter is the size of the window used in the filter. As we increase the smoothing factor, the rolling average becomes more and more flat. In terms of smoothing out noise, this is helpful, but over smoothing can result in fewer jumps being detectable, as seen in the left plot of Figure 3. Once the data is smoothed, we define a jump threshold. This is the threshold at which, when the rolling average crosses it, it is considered a jump.

Like with the smoothing factor, this parameter plays a large role in deciding whether or not a jump is detected. The rolling average must cross the threshold both positively and negatively to be considered a jump, so in cases where the rolling average only crosses one of these two, no jump is detected. An example of this is shown in the right plot of Figure 4.

To test the efficiency of the code at detecting jumps of various sizes, we run the code on simulated data with a known number of jumps of a known size. Included in the code is a check, which notes if a jump was injected in the scan, and weather or not a jump was detected. We have four possible

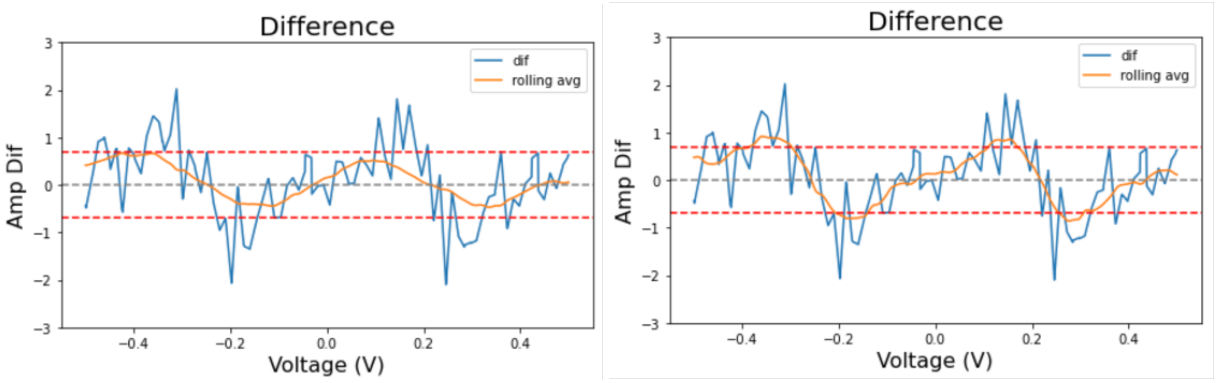


FIG. 3. Example of two smoothing factors. The plot on the left has a smoothing factor of 8 and the plot on the right has a smoothing factor of 4. The plots are of the same scan, with the same jump threshold (set at 0.7). There is a jump injected in this scan, but it is only detected with the smoothing factor set to 4. These plots demonstrate how over-smoothing can result in a false negative.

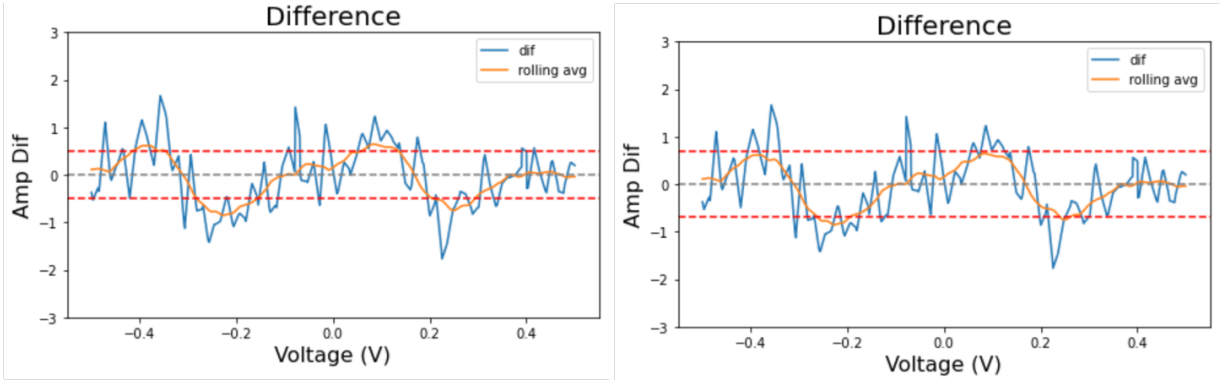


FIG. 4. Example of two jump thresholds. The plot on the left has a threshold set at 0.5 and the plot on the right has a threshold set at 0.7. This is the same scan (with a smoothing factor of 4). There is a jump injected in this scan, but with the higher jump threshold (right), it is not detected.

outcomes for this check: true positives, where a jump was injected and detected; true negatives, where no jump was injected and no jump was detected; false positives, where no jump was injected but a jump was detected; and false negatives, where a jump was injected but no jump is detected.

Given these two parameters, the jump detection code runs on a three dimensional array. The first is the number of jump sizes considered. For each jump size, we then have the number of consecutive scans. The last dimension is the number of points in each scan (this is the up-sampled number, 1265). We run the code across a range of jump sizes, starting with jumps with an offset charge of 0.0007 V and up to jumps with an offset charge of 0.25 V. For each scan, an array that sums all the outcomes is saved. Once the code is finished with all possible jump sizes, these arrays are plotted, resulting in an efficiency plot like Figure 5.

## III. ANALYSIS

In this work we have discussed two different parameters that play a prominent roll in the jump detection code: the smoothing factor and the jump threshold. We have explained how over smoothing and/or a too high threshold results in false negatives. One advantage of this method of jump detection is the ability to isolate noise, and ideally, detect small jumps that appear to be noise, while not mistaking noise for small jumps. We are ultimately interested in finding correlated jumps, and as these correlated jumps occur at the same time on qubits that are spatially close, the jump sizes can vary. Being able to detect the smallest of jumps will improve our capability to detect these correlated jumps. As the code is now, we see a greater number of false negatives for smaller sized jumps and more true positives for larger jump sizes. Remembering that a false negative is detecting a jump where no jump was injected, we find that with the test parameters as they are (smoothing = 4x and jump threshold at 0.5), the code falsely identifies noise and jumps. Future work will need to further investigate this, and find (if possible) how to resolve this.

## IV. CONCLUSION

Using simulated data, we are able to optimize parameters for the jump detecting code. We test parameters that increase the number of true positives and negatives, for jumps with

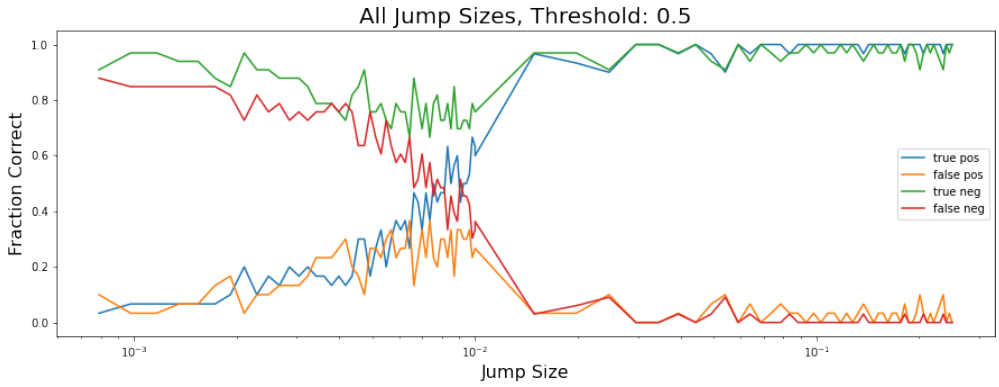


FIG. 5. Efficiency plot showing the number of each possible outcomes for 100 jump sizes. The smoothing factor is set at 4 and the jump threshold is 0.5

offset charges as small as 0.0007 V. Future work will include running a $\chi^2$ analysis to find the best combination of these two parameters. It will also involve running the jump detection code on real data and seeing how it preforms in all of the radiation configurations.

## 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] 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).

[2] M. McEwen, L. Faoro, K. Arya, A. Dunsworth, T. Huang, S. Kim, B. Burkett, A. Fowler, F. Arute, J. C. Bardin, A. Bengtsson, A. Bilmes, B. B. Buckley, N. Bushnell, Z. Chen, R. Collins, S. Demura, A. R. Derk, C. Erickson, M. Giustina, S. D. Harrington, S. Hong, E. Jeffrey, J. Kelly, P. V. Klimov, F. Kostritsa, P. Laptev, A. Locharla, X. Mi, K. C. Miao, S. Montazeri, J. Mutus, O. Naaman, M. Neeley, C. Neill, A. Opremcak, C. Quintana, N. Redd, P. Roushan, D. Sank, K. J. Satzinger, V. Shvarts, T. White, Z. J. Yao, P. Yeh, J. Yoo, Y. Chen, V. Smelyanskiy, J. M. Martinis, H. Neven, A. Megrant, L. Ioffe, and R. Barends, "Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits," Nature Physics **18**, 107–111 (2021).