Designing Quantum Algorithms for Qubit Control Calibration

Peter Karalekas
Yale College, Department of Physics and Department of Computer Science

Robert Schoelkopf (Thesis Advisor) and Brian Vlastakis (Mentor)
Yale University, School of Engineering & Applied Science, Robert Schoelkopf Lab

(Dated: April 30, 2015)

One of the most important requirements for building a quantum computer is having complete control over qubits and quantum gates. That is, errors in qubit state preparation and measurement, as well as errors in the fidelity of single-qubit gates need to to be sufficiently low for quantum computation to be feasible. One of the most widely accepted methods of diagnosing errors in gate implementation is that of randomized benchmarking, a process in which a quantum circuit composed of randomly chosen, yet known, gates is applied to a qubit prepared in the ground state, and the final state is then measured. The randomness of the circuit allows for the extraction of an average error per gate independent of the individual gates themselves, effectively evaluating the gate implementation process as a whole. An additional method known as interleaved randomized benchmarking can be used in conjunction to separate out the errors due to individual gates. For my thesis, I wrote a pulse generation and simulation software for randomized benchmarking in the Python programming language, and was then able to run it on a physical quantum system composed of superconducting qubits. I hope this tool will prove useful as the lab works to improve pulse tuning methods and ultimately gate fidelities of their quantum operations.

I. INTRODUCTION

In classical computing, a problem is said to have an efficient solution if the solution can be determined in polynomial time. However, some of the most challenging problems in computer science—such as finding the prime factors of large numbers—as well as others in physics—such as simulating multi-particle quantum systems—have no known efficient solution using the best known classical algorithms. In theory, quantum computing offers ways to improve the efficiency of otherwise intractable problems in classical computing, and therefore has gained much interest in both the physics and computer science communities. One of the most famous quantum algorithms that has been developed since the advent of quantum computing is Shor's algorithm, which is able to perform integer factorization in polynomial time. The development of this algorithm revolutionized the study of quantum computing, pushing other physicists to develop additional quantum algorithms that offer drastic speedups over their classical counterparts. However, without the ability to physically implement a quantum computer, no amount of progress in the theory of quantum computing will be realized, because even the simplest instances of the known quantum algorithms require hundreds of qubits and an even larger number of quantum gates to perform computations.

In attempting to physically realize a scalable quantum computer, one of the most important challenges that scientists face is being able to produce quantum gates that have low error rates during operations regardless of the context. One approach to diagnosing errors in gate implementation is the method of process tomography, which can be used to establish the complete behavior of a single quantum gate. This method can give much useful information about a gate being used as the sole operation on a qubit, but does not tell how well it will work when being used as the kth gate of a computation. Additionally, process tomography has exponential time complexity as a function of the number of qubits, so it can become very time consuming very quickly.

For my thesis, I instead decided to use randomized benchmarking, another widely accepted method of diagnosing errors in quantum gates. Rather than describing the complete behavior of a single quantum gate, randomized benchmarking works by preparing a sequence of random gates, and gives information about the average error rate of the entire circuit. Thus, randomized benchmarking can be considered as a tool for determining general gate implementation as a function of sequence length. One of the earliest papers on randomized benchmarking is that of Knill *et al* from 2008, which describes an implementation on trapped ion qubits [1]. In addition, a paper by Chow *et al* from 2009 describes another implementation of randomized benchmarking, but on solid-state qubits [2]. Each description of randomized benchmarking has slightly different features, but the one I ended up implementing is equivalent to the protocol in the 2012 paper by Magesan *et al* [3]. In this version, the sequence is comprised of random Clifford gates, where the set of Clifford gates is $\{I, X_\pi, Y_\pi, X_{\pm\pi/2}, Y_{\pm\pi/2}\}$ and U_θ is a θ-rotation

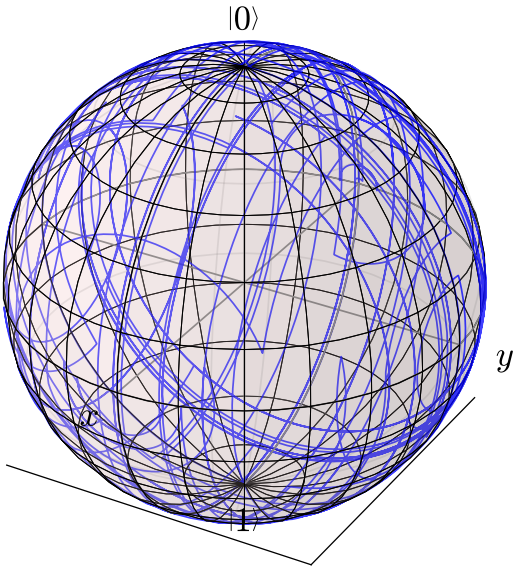

FIG. 1. Bloch sphere simulation showing the effect of a pulse train on a qubit in the ground state. The blue line shows the trajectory of a qubit state over the course of a single randomized pulse train.

around the U-axis. In addition, a final gate is added such that the entire sequence composes to the identity operation, and then the state can be measured and compared to the qubit's initial state.

II. SIMULATION TECHNIQUES

Before I could implement a randomized benchmarking simulation, I had to first get a handle on the tools at my disposal for designing quantum simulations. Using the QuTiP simulation and pulse sequence libraries developed by members of the lab, I learned how to set up a single-qubit simulation environment and introduce time-dependent driving terms. To set up the system, we must first define its starting Hamiltonian, which defines the evolution of the quantum system. For a two-level system this equation is represented by 2x2 unitary operators called Pauli matrices. However, for physical quantum systems like superconducting qubits, a two-level system is only an approximation of reality. This is one of the major difficulties in determining a physical qubits response to a time-dependent drive–although we want to represent physical qubits as merely a ground and excited state, we are in fact dealing with a multi-level quantum system. One way to combat this problem is by numerically simulating an approximation of the physical Hamiltonian, and a superconducting qubit is well approximated by an anharmonic quantum oscillator. By altering the energy levels of the various photon number states, we can create a set of non-uniform energy differences between the states. These non-uniform energy differences in effect decrease the probability that a driving pulse will result in an undesirable photon number state (one that is higher than the first excited state).

After understanding how to prepare the quantum system, my next goal was to investigate the possible operations I could perform on the system. When dealing with qubits, it is convention to use what is known as the Bloch sphere to visualize a qubit's state. The Bloch sphere is a three-dimensional representation of all the possible superposition states of a qubit, with the north and south poles representing the excited and ground states of the system. The vector from the center of the sphere to a point on the surface represents the current state of the qubit, and being able to correctly manipulate this state is a critical step in building a system capable of performing quantum computations. To manipulate a physical qubit, one can alter its state by applying quantum gates that act as rotations around the Bloch sphere. The lab uses microwaves pulses to apply these quantum gates to transmon qubits, where a π pulse rotates the state 180 degrees around the Bloch sphere, and a $\pi/2$ pulse rotates the state 90 degrees. However, because quantum systems are extremely sensitive to errors, these pulses do not always result in the desired state, thus motivating my goal of implementing randomized benchmarking to diagnose these errors.

III. RANDOMIZED BENCHMARKING

A. Standard Randomized Benchmarking

Implementing a randomized benchmarking simulation begins with the selection of a sequence of pulses to form the randomized pulse train. These pulses are chosen at random from the set of Clifford pulses $\{I, X_\pi, Y_\pi, X_{\pm\pi/2}, Y_{\pm\pi/2}\}$, where the first is equivalent to no pulse, the next two are π pulses, and the last two are $\pi/2$ pulses. Represented in matrix form, the first pulse is the identity operator, the second is the Pauli operator σ_x, the third is the Pauli operator σ_y, and the final two are:

$$X_{\pm\pi/2} = (I \pm i\sigma_x)/\sqrt{2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \pm i \\ \pm i & 1 \end{pmatrix}$$

$$Y_{\pm\pi/2} = (I \pm i\sigma_y)/\sqrt{2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \pm 1 \\ \mp 1 & 1 \end{pmatrix}$$

Denoting the set of Clifford pulses as $\mathcal{C}$, the jth Clifford pulse in our randomized pulse train will be written as C_j, where $C_j \in \mathcal{C}$. Then, the entire randomized pulse train is represented by $\bigcirc_{j=1}^{m} C_j$ where m is the desired pulse train length and $\bigcirc$ represents the composition of gates. But, there is still one final operation to be done. In simulation, we can track the evolution of a state vector, but in physical systems that would constitute measurement and therefore the collapse of our quantum system. Thus in experiment, we must choose what basis upon which to measure our qubit's state when we perform our final measurement. Rather than try to change this basis in accordance with the expected outcome of the randomized pulse train, we add one final correcting pulse to the train that should bring our quantum state back into an eigenstate of the z-basis. The z-basis is desirable because it is the basis in which our qubit's initial state was prepared (the ground state), and therefore it is easy to figure out if our qubit ended up where we predicted it to. Thus, if we say that $U = \bigcirc_{j=1}^{m} C_j$, we can then think of the correcting pulse as U^{-1}, effectively forcing the composition of the randomized pulse train and correcting pulse to be equivalent to the identity operator I. Finally, with the correcting pulse C_{m+1} added, the entire randomized benchmarking sequence S_{RB} can be written as:

$$S_{RB} = C_{m+1} \circ (\bigcirc_{j=1}^{m} C_j)$$

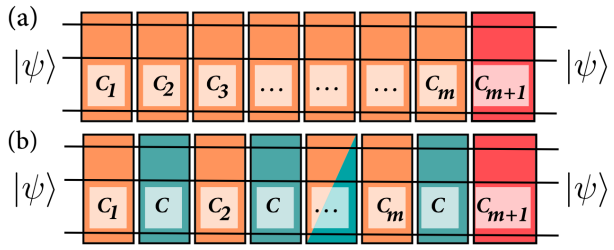

FIG. 2. Pulse train schemes for standard (a) and interleaved (b) randomized benchmarking protocols. The target pulse C (green) is interleaved between random pulses C_j (orange) selected from the Clifford pulses $\mathcal{C}$. A final correcting pulse C_{m+1} (red) is added to make the composition of the entire sequence equivalent to the identity operation. (Reprinted from Magesan *et al* [3]).

B. Interleaved Randomized Benchmarking

Like standard randomized benchmarking, interleaved randomized benchmarking is a protocol that uses a sequence of randomized pulses. But, in addition to these pulses, a target pulse C is interleaved between each randomized pulse. C, like the other pulses, is chosen from the set of Clifford pulses $\mathcal{C}$, and performing interleaved randomized benchmarking in conjunction with standard randomized benchmarking is a way to hone in on the errors contributed by a specific pulse to the entire pulse train. Other than the introduction of the C pulse, every other aspect of the protocol is identical to standard randomized benchmarking, so the entire interleaved randomized benchmarking sequence S_{IRB} can be written as:

$$S_{IRB} = C_{m+1} \circ (\bigcirc_{j=1}^{m} [C \circ C_j])$$

IV. DATA ANALYSIS

A. Standard Randomized Benchmarking

Applying the randomized benchmarking sequence S_{RB} to the initial state vector of a qubit $|\psi\rangle$ would theoretically result in no change to the state vector in the event of an error-free quantum system. However, the purpose of randomized benchmarking is to diagnose the possible errors that could arise during a sequence of quantum operations. Denoting the final state vector that is measured after the application of S_{RB} as $|\psi'\rangle$, we measure the probability that these states are different. This probability, called the "survival probability" or "excited state probability," would be 1 for each sequence in the ideal case. Averaging the probability over k sequences of the same length m, we then determine the sequence fidelity $F_{\text{seq}}(m)$. Performing this protocol for a number of different sequence lengths (we denote this set of lengths as M), we can then fit the data to the zeroth-order exponential decay model:

$$F_{\text{seq}}(m) = Ap^m + B$$

This fit gives the depolarizing parameter p, which is equivalent to our metric for gate fidelity. And, from p, it is trivial to calculate the average error rate per gate $e = 1 - p$.

B. Interleaved Randomized Benchmarking

In order to perform the analysis for interleaved randomized benchmarking, we must first perform standard randomized benchmarking. Then, using the same number k of sequences per length, and the same set of sequence lengths M as the standard randomized benchmarking instance, we build the interleaved randomized benchmarking sequences S_{IRB} and apply them to our qubit, once again de-

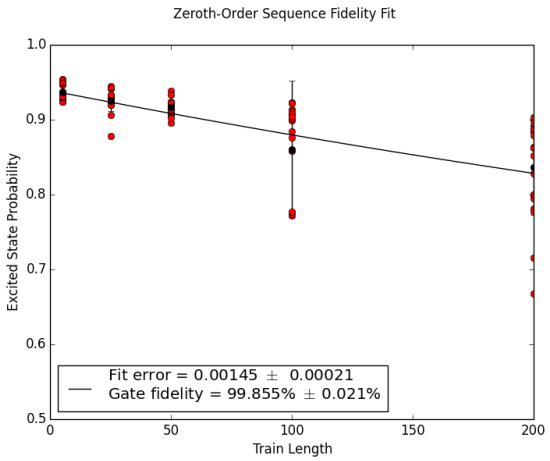

FIG. 3. Plot of a standard randomized benchmarking run with the set of train lengths $M = \{5, 25, 50, 100, 200\}$, the number of sequences per length $k = 20$, and 5000 averages per sequence. The spread of the excited state probability values is quantified using error bars, and the data is fit to a zeroth-order exponential decay model. From this fit, we can extract the gate fidelity $p = 99.855\% \pm 0.021\%$.

termining the survival probability for each sequence. We then average the probabilities over the k sequences per length m to determine the interleaved sequence fidelity $F'_{\text{seq}}(m)$. Similarly, we perform this protocol for every sequence length $m \in M$ and then fit the data to the zeroth-order exponential decay model once again. This time, we end up with the interleaved gate fidelity p', which allows us to calculate the corresponding average error per gate $e' = 1 - p'$. Finally, we can then extract the error e_C contributed by the target pulse C by the simple formula $e_C = e' - e$, where e was determined in the standard randomized benchmarking instance.

V. EXPERIMENTAL RESULTS

After completing my pulse generation and simulation software, I spent the remainder of my time in the lab running the software on physical qubits. Using the randomized benchmarking protocol and data analysis techniques described above, I was able to extract some meaningful data from experimentally implementing my randomized benchmarking software.

A. Standard Randomized Benchmarking

To learn more about the errors in the lab's pulse tuning methodology as a whole, I began by performing standard randomized benchmarking. Using the set of train lengths $M = \{5, 25, 50, 100, 200\}$ with $k = 20$ sequences per length and 5000 averages per sequence, I was able to produce the plot in figure 3. The plot features the zeroth-order exponential decay fit, along with error bars to quantify the spread of excited state probability values for a given train length. As is shown on the plot, the gate fidelity p determined by the fit is $p = 99.855\% \pm 0.021\%$, meaning that the average error per gate e can be calculated to be $e = 1 - p = 0.145\% \pm 0.021\%$. By itself, this number gives us an average error per gate, but cannot tell us what the source of these errors are.

B. Interleaved Randomized Benchmarking

Having performed standard randomized benchmarking, I then set out to try to gain more information about the specific sources of error in the randomized pulse trains. By interleaving the identity operator in our gate sequence, which is equivalent to a "wait" operation in the pulse train, we can implement something that is called "decoherence randomized benchmarking." This protocol is aptly named, because by determining the gate fidelity of a decoherence randomized benchmarking run and comparing it to an identically constructed standard randomized benchmarking run, we can learn more about the breakdown of errors between qubit decoherence and imperfect pulse tuning. In addition to the wait pulse, any of the Clifford pulses in $\mathcal{C}$ can be the target pulse C in an interleaved randomized benchmarking run. Using the same M, k, and number of averages per sequence as the standard randomized benchmarking run described in the previous subsection, I was able to extract the interleaved gate fidelities p' for a number of different pulses. Then, the corresponding e_C value for each interleaved pulse C can be calculated by the difference $e_C = p - p'$, where $p = 99.855\% \pm 0.021\%$ from the standard randomized benchmarking run (see table I).

C	p'	e_C
I	$99.833\% \pm 0.018\%$	$0.022\% \pm 0.039\%$
$X_{+\pi/2}$	$99.630\% \pm 0.041\%$	$0.225\% \pm 0.062\%$
$Y_{-\pi/2}$	$99.736\% \pm 0.030\%$	$0.119\% \pm 0.051\%$
X_{π}	$99.270\% \pm 0.125\%$	$0.585\% \pm 0.146\%$
Y_{π}	$99.157\% \pm 0.125\%$	$0.698\% \pm 0.146\%$

TABLE I. Error per gate contributions due to various interleaved Clifford pulses. To determine the value of e_C for each Clifford pulse C, we calculate the difference $p - p'$ of the standard and interleaved gate fidelties.

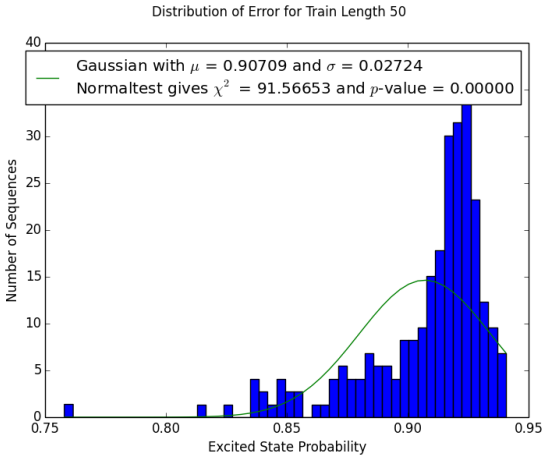

FIG. 4. Distribution of excited state probability for $m = 50$ and $k = 200$ with 5000 averages per sequence. The attempt at a Gaussian fit is in green, and the normality test of the data gives a p-value of 0.00000.

C. Distribution of Errors

In addition to analyzing the sequence fidelity fit of randomized benchmarking runs, and in turn extracting information about specific gate errors, I also investigated the distribution of excited state probability values for a given train length. Currently, not much is known about the behavior of such a distribution, but intuition says that it will not be Gaussian. The motivation for this intuition is that although the values for excited state probability are random, they are necessarily bounded above by the maximum probability value of 1. Therefore, I performed an additional standard randomized benchmarking run, but with the parameters slightly altered in order to determine whether or not the distribution is Gaussian. Using the set of train lengths $M = \{5, 25, 50\}$, the number of sequences per length $k = 200$, and 5000 averages per sequence, I was able to get rich distribution data for the various lengths. Then, I plotted the data for each train length in a corresponding histogram, tried to fit a Gaussian distribution to the data, and performed a normality test. As seen in figure 4, the distribution fits the data poorly, and the normality test gives a negligible p-value.

VI. CONCLUSION

Although more randomized benchmarking runs are necessary to validate the data, it seems that the e_C values can provide useful insight into the manifestation of errors in physical quantum systems. For example, the fact that the e_C value for the wait pulse is so small seems to indicate that the errors in the lab are weighted more heavily towards qubit decoherence rather than towards issues with pulse tuning. However, the difference in e_C values between the π and $\pi/2$ pulses also seems to indicate that of all the Clifford pulses, the π pulses are the largest sources of pulse tuning errors in a given sequence. In addition, the fact that a normality test of the excited state probability values gives a negligible p-value seems to affirm that the distribution is not Gaussian, and opens up possibilities for future research into the exact behavior of the distribution. Although my research leaves many questions unanswered, I hope that my pulse sequence generation and simulation software will serve as a useful tool in the future to improve the pulse tuning methods of the lab so that we can continue to make progress towards complete control over quantum systems and ultimately the realization of a scalable quantum computer.

[1] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. Randomized benchmarking of quantum gates. *Phys. Rev. A* **77**, 012307 (2008).

[2] J. M. Chow, J. M. Gambetta, L. Tornberg, Jens Koch, Lev S. Bishop, A. A. Houck, B. R. Johnson, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Randomized Benchmarking and Process Tomography for Gate Errors in a Solid-State Qubit. *Phys. Rev. Lett.* **102**, 090502 (2009).

[3] Easwar Magesan, Jay M. Gambetta, B. R. Johnson, Colm A. Ryan, Jerry M. Chow, Seth T. Merkel, Marcus P. da Silva, George A. Keefe, Mary B. Rothwell, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen. Efficient Measurement of Quantum Gate Error by Interleaved Randomized Benchmarking. *Phys. Rev. Lett.* **109**, 080505 (2012).

[4] A. Blais, J. Gambetta, A. Wallraff, D. I. Schuster, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Quantum Information Processing with Circuit Quantum Electrodynamics. *Phys Rev A* **75**, 032329 (2007).

[5] S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Circuit QED and engineering charge based superconducting qubits. *Phys. Scr. T* **137**, 014012 (2009).

[6] M. H. Devoret and R. J. Schoelkopf. Superconducting Circuits for Quantum Information: An Outlook. *Science* **339** 1169-1174 (2013).