1 Introduction

Team HASAA attempted to implement both the Optimal control and Automated closed-loop control (aka Learning control) approaches for solving the challenge. Ultimately, we settled on using the Automated closed-loop approach. However, it was a valuable experience for our team to delve deeper into the physics outlined in the application notes that were given in the challenge documentation. We thoroughly enjoyed the challenge. We begin this document by outlining some of the physics of superconducting qubits.

$$\frac{H_{ideal}(t)}{\hbar} = \frac{1}{2}\Omega(t)b + \frac{1}{2}\Omega^*(t)b^\dagger \tag{1}$$

In equation 1, we show the *ideal* qubit Hamiltonian with no error processes under the action of a microwave pulse. We use this as the basis for of our mathematical model and then we consider further error terms.

$$H(t) = \frac{\Omega(t)}{2}\left(\sigma_x + Q_{\text{unknown}}\right) \tag{2}$$

$$= H_{ideal}(t) + \frac{\Omega(t)}{2}Q_{\text{unknown}} \tag{3}$$

In equation 2, $\Omega(t)$ is the control pulse Rabi rate and $Q_{unknown}$ is the unknown error operator, combining all the error terms. This Hamiltonian represents a more realistic model of a qubit, albeit not detailed. In order to optimize these pulses to achieve a high fidelity *NOT*-gate and *H*-gate, we estimate the errors.
There are some leading source of errors for a superconducting qubit like dephasing errors, state preparations and measurement (SPAM) error, amplitude errors etc. Dephasing causes decoherence and this error can partially be addressed with the optimization of bandlimited pulses.

2 Optimal Control

Optimal control involves constructing a system Hamiltonian for the qubit with appropriate error terms and specific parameters. One must have a detailed model of the system in order to use this approach. We tried expanding on the model given for the "*more realistic superconducting qubit*". However, in practice restructuring the code for a new Hamiltonian became a big challenge and we eventually decided that Optimal control would require more time than we had, so we opted to switch to the Automated closed-loop approach, since the other half of our team had achieved more promising results in the time frame we had allotted. If we had continued with Optimal control our approach would have been as follows:

1. Get an accurate model of the "more realistic superconducting qubit" working using the system identification method outlined in the application notes and parameterizing the error channels in the Hamiltonian. Trial pulses would be sent to the local qubit to test and optimize our system. This step was primarily to validate the approach before moving to the more complex "qubit in the cloud".

2. Create an accurate model of the "qubit in the cloud". Use the system identification method on one of the templates that we prepared based on the findings that came from step 1 to characterise the control lines and identify the constant terms of the Hamiltonian.

3. Use the model derived in step 2 and optimize it using one of the worked examples described in the Optimize control user guide.

The main outcome from having attempted to work with Optimal control has been the invaluable learning experience of the physics and mathematics of quantum systems that we would not have been able to get if we decided to only stick with the Learning control approach due to the way machine learning models deprecate the need to exactly model the qubit Hamiltonian error terms.

3 Closed-loop automated control

The idea behind the closed-loop automated control is the use of BOULDER OPAL's toolkit to enable a closed feedback loop, which optimizes the parameters of the control pulse. This is easier to implement because the

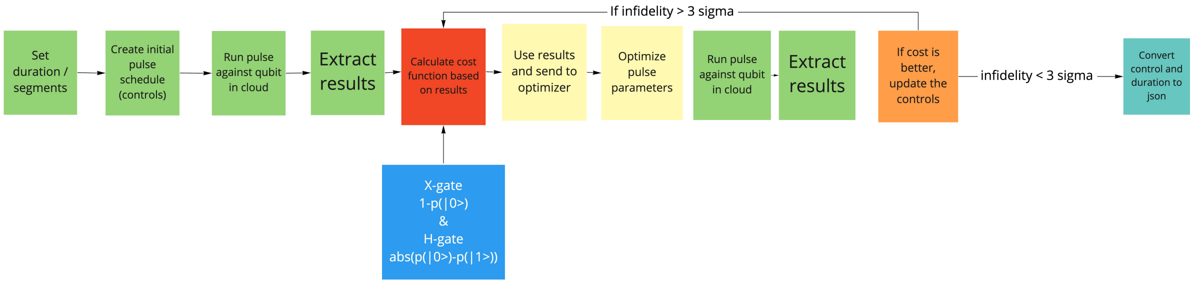

Figure 1: This is a Miro board explaining the algorithm flow. Here we show how the cost function is minimized by executing a for loop until the $<= 3\sigma$ condition is satisfied. For the **NOT-gate** & **H-gate**, the cost function is calculated based on the results from the cloud qubit. Then the simulation results are optimized with gaussian processing to obtain new pulse parameters. This process is looped and is contingent on the cost function decreasing.

machine learning model absolves us of the need to characterize the control lines and identify the constant terms of the Hamiltonian. As in equation 2, we can see the error terms in the Hamiltonian are left undefined and the machine learning model is powerful enough to converge even without knowing anything about the system. We have successfully deployed the closed-loop control method and we outline it with the Miro board shown above.
Internally we organize the results into a proper input format and then call the automated closed-loop optimizer and obtain the next set of test points. We record the best results after every round of experiments. We then export to json the best results. Ideally, we observe that we have chosen the right numbers for duration and segmentation. This will provide us with an optimization loop that will obtain the functioning gate. In our work we implemented a function that calculates infidelities (e.g. a cost function) for both the NOT-gate and the H-gate. As specified earlier, we repeated the process of running the pulse against the qubit in the cloud and extracting the results. We describe the process and challenges that we had in our loop in more detail below.

3.1 Sending pulses to the qubit in the cloud

After creating an initial pulse and getting results we used the closed-loop optimizer from BOULDER OPAL to generate the next set of pulses to be sent to the cloud qubit. Effectively, with each iteration we are minimizing the unknown error terms in Equation (3) for the *NOT- gate*. We continue this process until we are within 3 standard deviations i.e. (cost function $<= 3\sigma$).

3.2 Duration altering for NOT gate

Duration was first set at $t_{init} = \frac{\pi}{\Omega}$, where Ω is the maximum Rabi rate obtained from the challenge application note. We then manually adjusted the duration to minimize it. Initially, we set duration to 300 ns, since it was the maximum duration given by the mentors. We then proceeded to lower the duration with a divide-and-conquer approach. We reached an optimal duration of 60 ns. We also tried using different segment counts. We found that shorter segments worked better for us than longer segments, so we probed until we got down to 8 segments. For the **NOT-gate** we used a cost function of:

$$\text{Cost function} = 1 - p\left(|1\rangle\right) \tag{4}$$

where $p\left(|1\rangle\right)$ is the probability of the state being in the excited state.
Finally, we tested with multiple repetitions of the NOT-gate in a single experiment to see the effect on infidelity.

3.3 H gate

The Hadamard gate is a single qubit gate which when applied on $|0\rangle$ gives the $|+\rangle$. In order to implement this, we created the cost function by assuming that the ideal state is when the difference between the populations of the ground and excited state are minimized. So, the formula of : Cost function $= \|p(|0\rangle) - p(|1\rangle)\|$. Nevertheless, by minimizing this cost function, we would be optimizing for *any* superposition state. Therefore, we would not be able to tell if we specifically had the $|+\rangle$.
We got it to a superposition, however we are not sure if it is the $|+\rangle$ state. Therefore if we were to continue this project we would apply our supposed H gate and then our controlled NOT gate to check if it is in exactly the $|+\rangle$ state, as the $|+\rangle$ state is an eigenstate of the NOT-gate.

If we had more time, we would have liked to use the pulse calibration method outlined in a Q-CTRL application note to obtain a better approximation to the Ω of the simulated qubit. We would then see if robust control allowed us to infer the state trajectory. We started this process and found difficulty with configuring the notebook to connect to the cloud qubit instead of an IBM machine as it was originally in the application note.