MULT90063 Introduction to Quantum Computing

Lab Session 7

7.1 Introduction

Welcome to Lab 7 of MULT90063 Introduction to Quantum Computing.

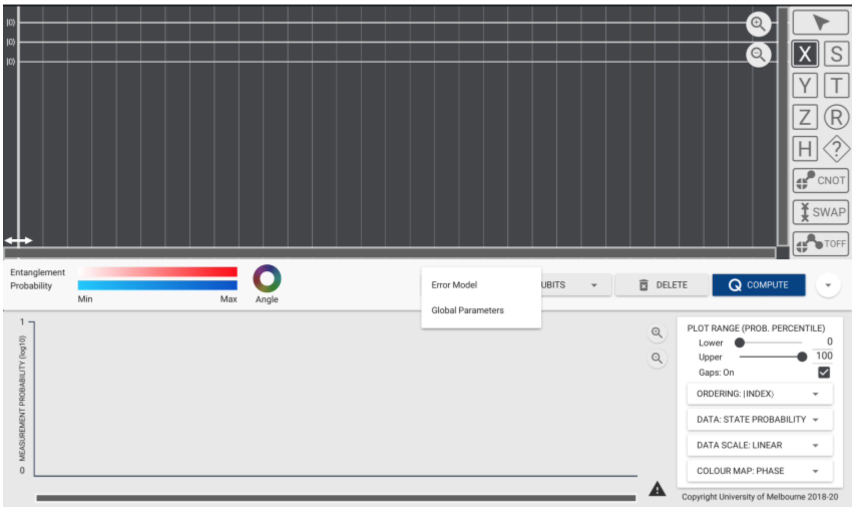
The purpose of this lab session is to:

- understand and implement rotation errors in quantum circuits
- understand and implement quantum supremacy (QS) circuits
- implement rotations errors in QS circuits and understand their effect

7.2 Rotation errors in the QUI

One of the key goals of this subject is to provide students with experience in programming an actual quantum computer. However, in the real-world the digital logic of qubits is prone to "errors" due to noise in the control systems and/or the immediate environment of the qubit itself – in order to understand and appreciate the results when we access quantum computer hardware we need to study these effects. While the quantum computer simulator powering the QUI is effectively a pristine qubit environment (we have set the errors to zero!), we can introduce such effects systematically and investigate how quantum gate errors affect the output of quantum circuits. Consideration of quantum errors is particularly important when we study quantum supremacy circuits as the real world is not pristine, and the effect of errors on quantum gates must be taken into account in the determination of the quantum supremacy point.

In what follows, we will represent rotation errors around the cartesian axes in the QUI using an error model. You can set the error model under the "Settings" drop-down in QUI:



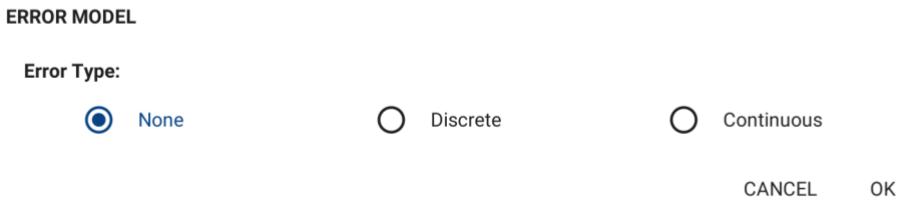
ORDERING: |INDEX⟩

DATA: STATE PROBABILITY

DATA SCALE: LINEAR

COLOUR MAP: PHASE

When you select “Error Model” you are given a choice between no errors, a discrete or continuous error model:



ERROR MODEL

Error Type:

● None ○ Discrete ○ Continuous

We will first consider control errors, which can be modelled by a continuous error model. We saw in lectures that a Z-rotation error (or just “Z-error”) can be considered an unwanted gate δZ defined as:

$$\delta Z \equiv \begin{pmatrix} e^{-i\epsilon/2} & 0 \\ 0 & e^{i\epsilon/2} \end{pmatrix}$$

where the level of error is governed by the angle ε (assumed to be small). We can implement this effective error gate in the QUI as follows:

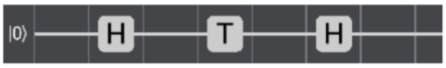
$$R_Z(\epsilon) = e^{i\theta_g}\big|_{\theta_g=0}\left(I\cos\frac{\epsilon}{2} - iZ\sin\frac{\epsilon}{2}\right) = \begin{pmatrix} e^{-i\epsilon/2} & 0 \\ 0 & e^{i\epsilon/2} \end{pmatrix}$$

Similarly for X-error and Y-error gates so we have for the “Pauli” continuous rotation error gates:

$$\delta X = R_X(\epsilon)\,, \quad \delta Y = R_Y(\epsilon)\,, \quad \delta Z = R_Z(\epsilon)$$

with global phases zero. We could define more general errors, but these will suffice. These are modelled in QUI as continuous errors.

Exercise 7.2.1 Let’s go back to our favourite 3-gate (interference) example, HTH, and examine the effect of rotation errors on the circuit:



a) Z-errors. Program the HTH_ circuit in the QUI. Apply an error model with angles having a mean of zero and a standard deviations of ε around the z-axis. Notice that when you run the circuit with a non-zero error model, the location of errors are indicated in your circuit. Gather statistics about the outcomes which would be measured after this circuit, by running this circuit by running the circuit many times, and averaging over the outcome probabilities reported by QUI for the zero state and one state.

Compare the effect of running the the “pristine” no-error output to those with increasingly large error by filling in the table below:

Table 1: The effect of Z-errors on HTH

ε	Output state $\lvert\psi\rangle = a_0\lvert 0\rangle + a_1\lvert 1\rangle$	Output state change	Output state relative change
None	$p_0\rvert_{\text{exact}} = 0.854$ $p_1\rvert_{\text{exact}} = 0.146$	$\Delta p_0 = 0$ $\Delta p_1 = 0$	$\frac{\Delta p_0}{p_0\rvert_{\text{exact}}} = 0$

			$\frac{\Delta p_1}{p_1\|_{\text{exact}}} = 0$
0.1 π			
0.2 π			
0.5 π			
π			

Look at the data in the table – what do you conclude?

b) X-errors. Repeat for the case of x-errors, and examine the effect of different error rates.

Table 1: The effect of X-errors on HTH

ε	Output state $\|\psi\rangle = a_0\|0\rangle + a_1\|1\rangle$	Output state change Compared to	Output state relative change
None	$p_0\|_{\text{exact}} = 0.854$ $p_1\|_{\text{exact}} = 0.146$	$\Delta p_0 = 0$ $\Delta p_1 = 0$	$\frac{\Delta p_0}{p_0\|_{\text{exact}}} = 0$ $\frac{\Delta p_1}{p_1\|_{\text{exact}}} = 0$
0.1 π			
0.2 π			
0.5 π			
π			

Compare these results with those you found in part (a).

7.3 Effect of rotation errors in quantum circuits

Obviously, not all errors have the same effect – it will depend on the type of error and where it occurs in the circuit. The effect of the error also depends on the circuit itself. We will now investigate how the outputs of some of our previously coded quantum algorithms (hopefully saved!) are affected by such rotation errors.

Exercise 7.3.1 Load the file "Lab 5 Grover 3 oracle marks 5", run it and familiarise yourself with how it works and the output. Set the circuit to compute the first two iterations only – to check: the probability of finding the target state 5=101 is 94.5%.

Add a continuous error model, with different levels standard deviation in the angle, and examine the resulting states, determine the probability of success of the algorithm:

ε (z-error)	Probability of finding the marked state
None	94.5%
0.01 π	
0.05 π	
0.10 π	
π	

Repeat for Z and Y errors independently (also at the ε = 0.1π level).

Exercise 7.3.3 Repeat for the 3-qubit QFT adder circuit. What is the maximum error level you would tolerate in the adder?

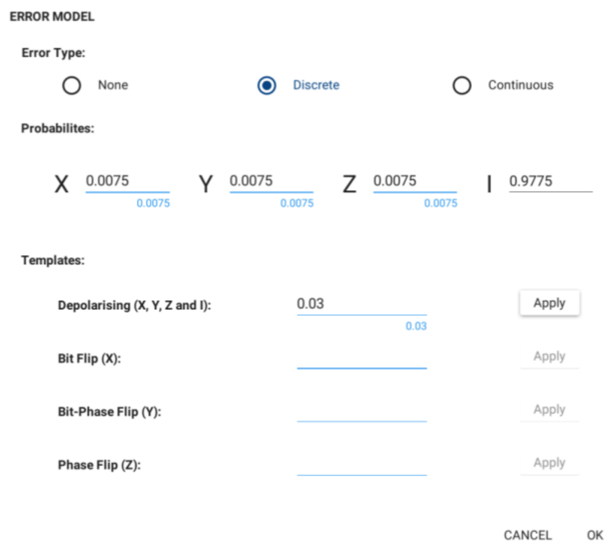
7.4 Types of errors: control precision and decoherence

The ways in which qubits and quantum gates can be disrupted is still an area of active research. Broadly, a quantum computer can be affected by imprecise control, or by stray interactions with the environment (i.e. decoherence), or both. In the previous sections the inclusion of rotation errors were closely related to the sorts of effects produced by control errors. Here we will look at the other type of error – those produced by decoherence. Typically, these are modelled as a complete "Pauli" gate (ε = π) occurring at random in the circuit with some probability p per time step. If we have a circuit with M possible error locations the total number of errors N_e that can occur for each run is given by the product $N_e = p\, M$. We interpret this as each time we run the quantum circuit we effectively have N_e extra X, Y or Z gates occurring at random locations, and which are different every time we run the circuit.

In the QUI this is implemented as a discrete error model.

Exercise 7.4.1 Load the three qubit Grover's search. In this algorithm, the number of error locations is 43. If we set the probability of an error per time step to be 3% (typically where the hardware is at), then the total number of errors in our circuit will be roughly 0.03x43~1. So, each time we run the Grover circuit an error in the form of an extra X, Y or Z gate will occur somewhere in the circuit.

Choose the discrete error model, and implement "Depolarising" noise, with a probability of 3% error, as shown below:



ERROR MODEL

Error Type:

○ None ● Discrete ○ Continuous

Probabilites:

X 0.0075 (0.0075) Y 0.0075 (0.0075) Z 0.0075 (0.0075) I 0.9775

Templates:

Depolarising (X, Y, Z and I): 0.03 (0.03)

Bit Flip (X):

Bit-Phase Flip (Y):

Phase Flip (Z):

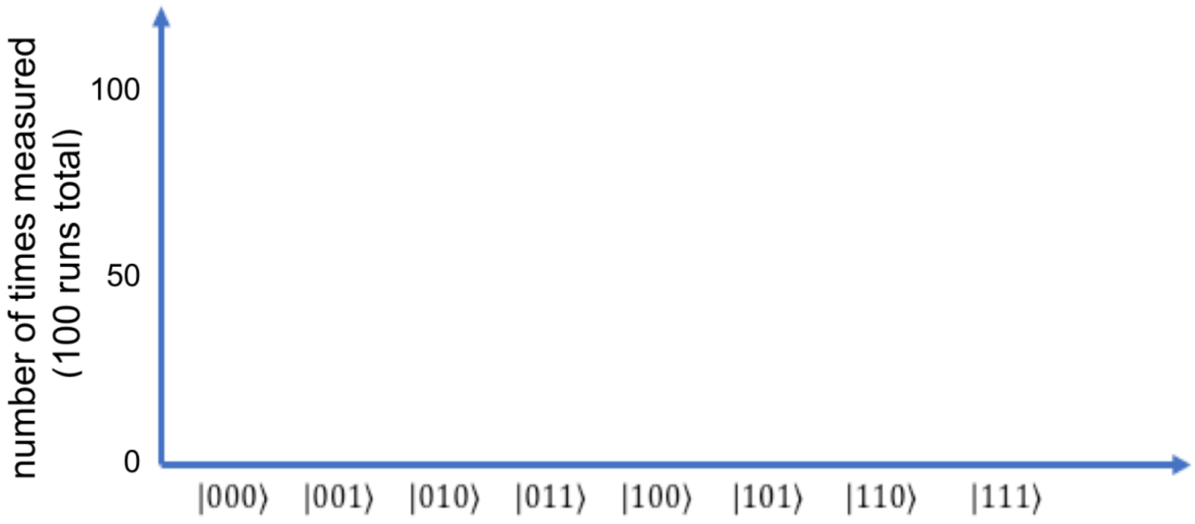
Run the circuit multiple times, and record the outcome of each run:

Table of results:

Measurement outcome	Record 𝍸…etc	Total
000		
001		
010		
011		
100		
101		
110		
111		

Now plot the results in the histogram below.

Grover's algorithm (3 qubits, search on 5=101), prob error = 3%



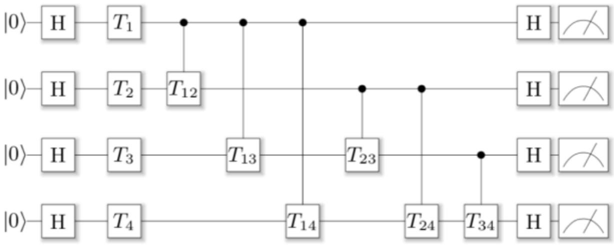
From these results estimate the probability of the algorithm producing a correct result in the presence of the 3% error rate.

To do: Now incorporate X, Y and Z errors at random (at various effective densities, and effective error angles) in some of the circuits you have programmed previously and examine the effect on the outputs. For example:

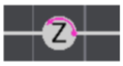
- Toffoli based adders
- Grover search
- QFT and QFT adders
- Baby Shor

7.5 Quantum supremacy – IQP circuits

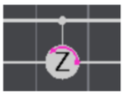
The basic IQP circuit scheme, and the relation to QUI gates, is shown below:



$$T_m = \cos\left(\frac{k_m\pi}{8}\right) I + i\sin\left(\frac{k_m\pi}{8}\right) Z_m \quad \to R_Z\left(\frac{-k_m\pi}{4}\right)\Bigg|_{\theta_g=0}$$

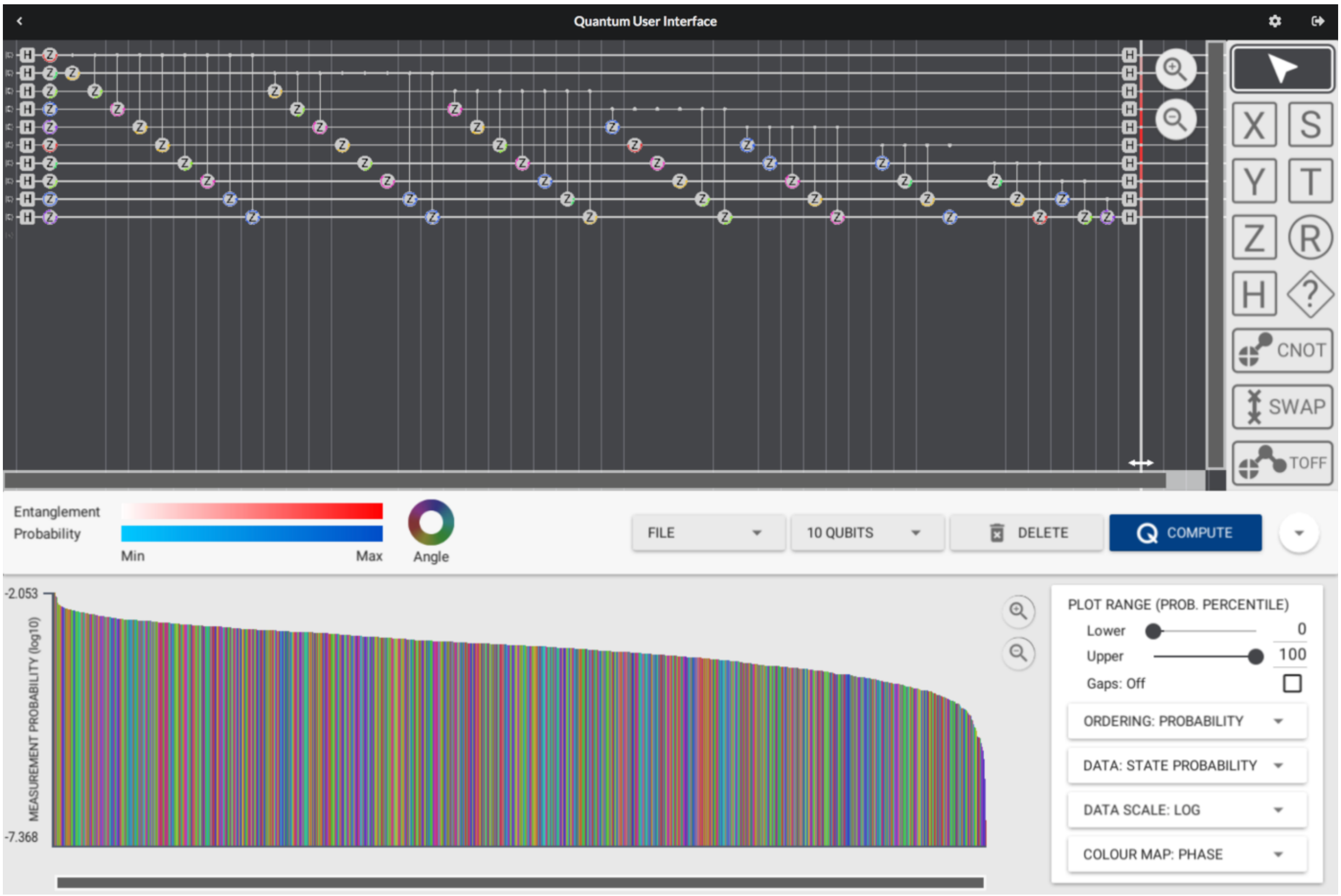


$$T_{mn} = \cos\left(\frac{k_{mn}\pi}{8}\right) I + i\sin\left(\frac{k_{mn}\pi}{8}\right) Z_m Z_n \to \boxed{R_Z\left(\frac{-k_{nm}\pi}{4}\right)\Bigg|_{\theta_g=0}}$$



$$\frac{-k_n\pi}{4}, \frac{-k_{nm}\pi}{4} \in \{0, -0.25, -0.50, -0.75, -1.00, -1.25, -1.50, -1.75\}\pi$$

Exercise 6.5.1 Code up a small-scale IQP circuit in the QUI (generalised from the schematic above) using random selections from the set of arguments of the R-gates. For the two-qubit gates, use controlled-z rotations with a random angle which is a multiple of π/4. Apply a discrete error model and observe the change in outcome as different errors are applied.



10 QUBITS

PLOT RANGE (PROB. PERCENTILE)

Lower 0

Upper 100

Gaps: Off □

ORDERING: PROBABILITY

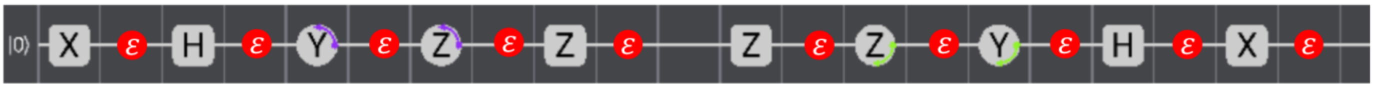
DATA: STATE PROBABILITY

DATA SCALE: LOG

COLOUR MAP: PHASE

7.5 Randomised benchmarking

We will now consider some randomised benchmarking examples using the QUI. Below is an example of a 5-gate sequence X-H-SQRT(Y)-S-Z (SQRT(Y) = RY($\pi/2$) etc) followed by its inverse. Notice the use of the R-gate to define some of the gates (why?). The schematic also indicates the error positions interleaved in the sequence.



Exercise 6.5.1 Program a random 5-gate sequence (with spaces as shown) using gates chosen from the set {X, Y, Z, H, RX($\pi/2$), RY($\pi/2$) RZ($\pi/2$)} and run to check it produces the |0⟩ state at the end (i.e. that the inverse sequence is correct). Choose a depolarising error model, with 3% probability and rerun the sequence. Run the sequence several times and determine the probability of measuring the |0⟩ state. Edit the gate sequence to produce another random 5-gate sequence (m=5), leaving the errors alone for simplicity (a static control error assumption). Run and record the probability of measuring the |0⟩ state. Keep repeating until you have enough statistics to calculate an average over the probabilities (i.e. $F_{m=5}$). Repeat all of that for an 8-gate sequence to obtain $F_{m=8}$. You've now got two data points (m=5 and m=8) to fit to the form given in the lectures:

$$F_m = A + Bf^m$$

i.e. determine *A* and *B* and *f* (assume m=0 implies A + B = 1). From this value of *f*, calculate “average fidelity” of the gates using the formula:

$$F_{av} = \frac{f+1}{2}$$