UNIVERSITY OF NEW SOUTH WALES

MTRN3020

Modelling and Control of

Mechatronic Systems

Laboratory Experiment 3

Design and Implementation

of a Position Controller

I verify that the contents of this report are my own work.

By Aidan O'Brien

z3324494

17 October 2014

1. Introduction

The third laboratory experiment built upon the work of previous labs and involved designing and implementing a position controller. This was done by first creating a speed controller loop by the direct analytical method for the internal velocity feedback loop, then using the root locus method to create the external position controller feedback loop. It required steady state errors to be zero in both the speed and position controllers. The constants used in the final control loop are based upon the given variables for the laboratory and

The experimental data was gathered after the controller was designed. The experimental setup involved having a motor driving a revolute joint via a gear box. At the top of this setup was an arm that could be extended or retracted. The change in length of the arm allows for a change in inertia at the revolute joint.

Part A of this experiment involves design verification of the controller. A simulink model was created with the controller function and a transfer function for the plant itself. It will verify if the design has been carried out correctly. Set position changes are made at regular intervals.

Part B shows how the controller responds when applied to a slightly different system, to test the effect of uncertainties in the model. The arm length is changed to alter the moment of inertia, which shows whether the controller can be designed without knowing exact system parameters.

Finally comparisons between experimental data and simulation data are made, to judge the validity of the simulation.

2. Aim

The purpose of this experiment was to design and implement a position controller for a motor that is measured by an encoder, using a combination of the direct analytical and root-locus methods. The arm connected to the motor is capable of multiple lengths which subjects the motor to various inertial loads. The arm will ideally attain set positions with zero steady state error, and changes in arm length will be adapted for within the controller.

3. Experimental Design Procedure

The procedure for the experiment involved first developing the speed controllers transfer function. Using data gathered from the first order response of the motor according to voltage input, enables us to find the transfer function for the plant. Once we find the plant information, we are able to design the speed controller. From this, we then use the root locus method to develop the position controller, thanks to the gain and breakaway point. Once this controller has been developed, we then applied it to the physical system, which set positions at certain time intervals (approximately ever 2.4 seconds) and recorded the experimental data. This data is used to ensure the Simulink model is correct. Afterwards, a second dataset is created, by changing the length of the arm, to allow us to check if the controller is capable of dealing with an uncertain model of a different system. The simplified Simulink model is shown in Fig 3.1.

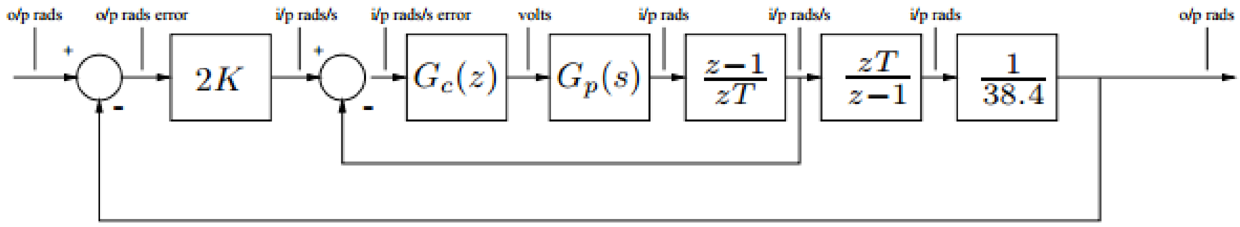

Figure 3.1. Simplified Block Diagram. From: MTRN3020 assignment specification.

This block diagram shows the simplified blocks for the Simulink model. The differentiators and integrators are included along with the gain and gear ratio.

4. Controller Design Calculation

The first stage of finding the controller, was to find the open loop response, by fitting a curve to the experimental data. This found the gain (A) and time constant (τ) for the motor. This was taken from the ROT335.m data file. Utilising the same code as used in experiment 2, and repeated here in snippets 1 and 2 for reference, it was possible to discover the values.

$$A = max(TEST(:,3))/24*(2*pi)/8192; \tag{1}$$

$$tau = TEST(max(find(TEST(1:250,3) < max(TEST(:,3))*(2/3))),1)/1000; \tag{2}$$

Once the gain was divided by the applied voltage (24V) and converted from the counts, a value of 8.96616 was found. The tau value was 110ms. The code utilised to generate the graph that compares the theoretical responses with the experimental data is shown in snippets 3 to 6.

$$t = (0:0.008:0.400); \tag{3}$$

$$y = 24*A*(1-\exp(-t/tau)); \tag{4}$$

$$plot(TEST(:,1)/1000, TEST(:,3),'r'); \tag{5}$$

$$plot(t, y,'b'); \tag{6}$$

The generated graph is shown in figure 4.1, which results in a close alignment between the theoretical response and the experimental response, after the initial disturbance died down

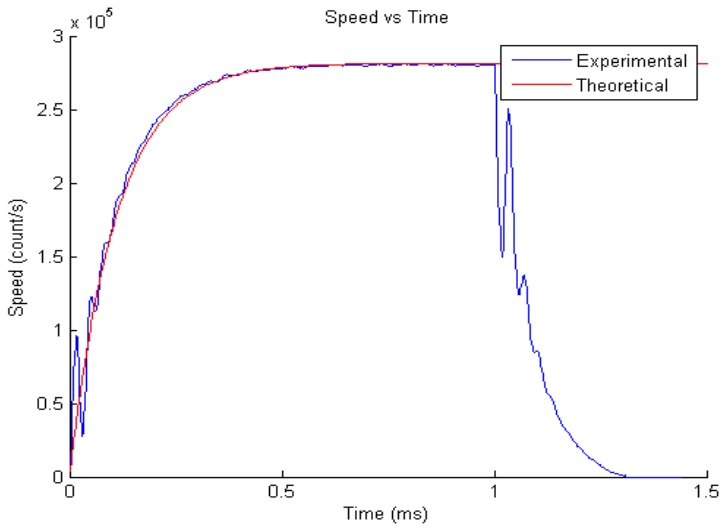

Figure 4.1. Graph of Speed vs Time of Experimental vs Theoretical responses.

Once tau and A were found, it was possible to directly form the plant's transfer function in the s-domain as shown in equation 4.1.

$$G_p(s) = \frac{A}{s(1+\tau s)} = \frac{8.99158}{s(1+0.110)} \tag{4.1}$$

Using the properties of c2dm in MATLAB, it was possible to directly find the discrete time transfer function of the plant. z1, p1 and C were all found in the same method as experiment 2. Snippets 7 through 9 show the formation of the plant's transfer function, while snippets 10 and 11 solve for the controllers transfer function. A necessary part of solving for the controller is B0, which is shown in equation 4.2.

$$numGs = [A]; \tag{7}$$

$$denGs = [tau\,1\,0]; \tag{8}$$

$$[numGz, denGz] = c2dm(numGs, denGs, T, 'zoh'); \tag{9}$$

$$Gz = tf(numGz, denGz, T); \tag{10}$$

$$B_0 = \frac{1-e^{\frac{-T}{\tau_d}}}{1-z1} = \frac{1-e^{\frac{-0.008}{0.048}}}{1+0.97605} = 0.077689 \tag{4.2}$$

$$e = \exp(-T/tau_d) \tag{10}$$

$$Gc = tf((T*B_0/C)*[1-p_1 0],[1,-(B_0+e),B_0*z_1],T) \tag{11}$$

This solves the transfer function of the controller directly by utilising MATLAB's inbuilt tf function and the output is shown in figure 4.2.

```
          0.2433 z^2 - 0.2262 z
Gc =  -------------------------
        z^2 - 0.9242 z - 0.07583

Sample time: 0.008 seconds
Discrete-time transfer function.
```

Figure 4.2. MATLAB output showing the controller's transfer function.

The controller must meet certain conditions to ensure that it is effective. Firstly, it must be stable, so any poles to the right of the axis must be cancelled out by the controller. The controller is also bound by the causality constraint, which involves making the controller depend solely on past and present information. This is absorbed by the arrangement of F(z) as given in the assignment specification. This solves the velocity controller. It is necessary to solve for the external position control loop.

The external loop was developed by using the root locus method. Code snippets 12 to 15 show how the plot was developed. Once the plot was displayed in MATLAB, it was a matter of finding the breakaway point and the gain value at that point.

$$numGp2=(2*b0*T/34.8)*[1-z1]; \quad (12)$$

$$denGp2=[1(-1-pD)\,pD]; \quad (13)$$

$$Gp2=tf(numGp2, denGp2, T); \quad (14)$$

$$rlocus(Gp2) \quad (15)$$

Figure 4.3 displays the root locus graph generated by the above code snippets.

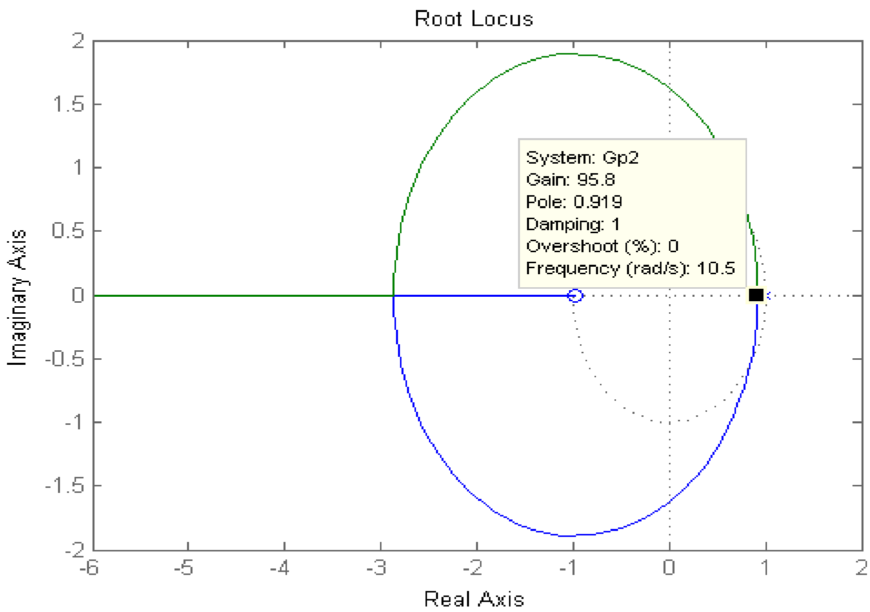

Figure 4.3 Root Locus of of the control system.

As can be seen in the root locus graph, the gain was 95.8 at the pole 0.919. The pole location relates to a time constant of 94.7ms. When placed into the simulink diagram and feedback loop, we need 2K, which is 95.8x2 = 191.6.

5. Simulink Block Diagram

As shown in figure 5.1, the Simulink block diagram was formed to provide outputs back to MATLAB.

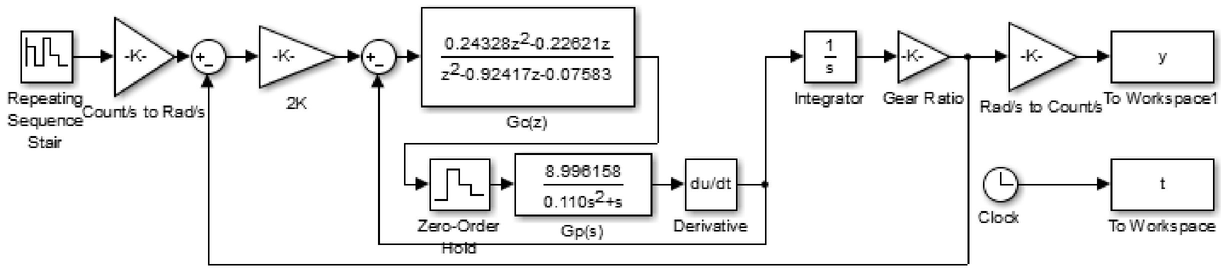

Figure 5.1. Simulink Diagram of Speed controller and Plant Transfer functions with external position control loop.

The Simulink block diagram is only slightly different to the simplified diagram given in the assignment outline. The inclusion of the ZOH to allow for operation in the s-domain and the conversion to and from rad/s are the only major changes.

The Gc(z) controller is the one developed in section 4, using Ragazzini's method. The plant's transfer function was found during the development of the control system. The outer loop was developed with the root locus method.

6. Design Verification

The Simulink block diagram's design was verified by comparing the output with the experimental data for the same length of the arm. The testing method involved spinning the arm to a certain number of counts, which is measured by the encoders in the system. The graph in figure 6.1 shows the comparison of the simulation output and the experimental data. The encoder counts have been scaled by the gearbox amount.

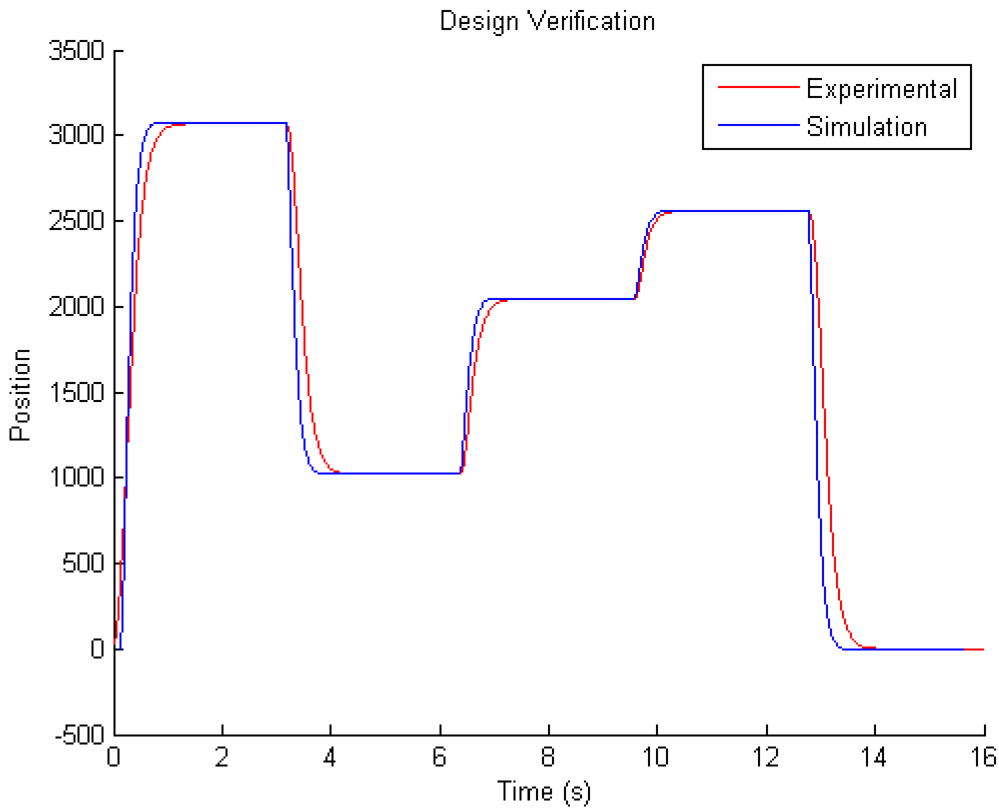

Figure 6.1. Design Verification Graph compared to ROT335 experimental data.

The figure shows a close parallel between the simulation data and the experimental data. However there's a distinctly faster rise time in the simulation data. This is most likely due to saturation of the real set-up, which will be discussed in the conclusion. The steady state error is extremely close to zero.

7. Effect of Modelling Uncertainties

The modelling of uncertainties is a test for the controller to see how effective it is at adapting to different plant set-ups. When the plant changes, the controller can still be useful if it adapts to the changes well. The method we used to test this was putting the unchanged controller on a different plant. This was performed by changing the arm length of the machine and then running it and seeing it reached zero steady state error. However, we had to find the different plant transfer function, the same as was performed in section 4. Figure 7.1 shows the graph of the first order response.

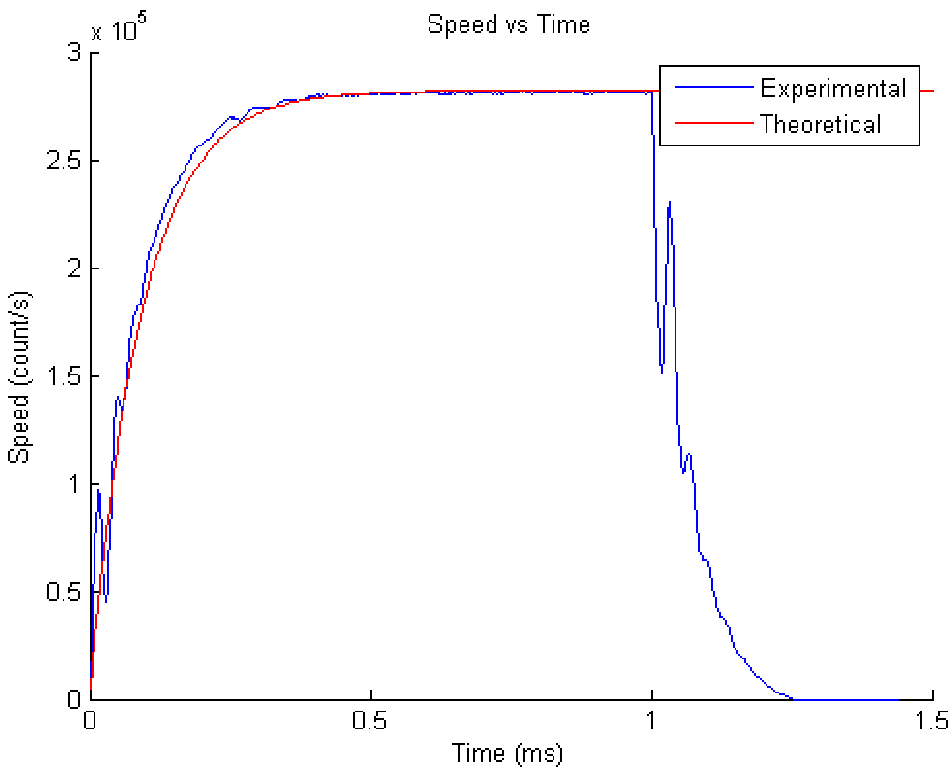

Figure 7.1 Graph of Speed. Theoretical vs Experimental Response, ROT225.

As this graph is extremely similar to that utilised to find the original plant, the same method was chosen to represent the plant's transfer function. Equation 7.1 shows the development of the different transfer function. Apart from the numbers changed in the plant's transfer function and the sequential step order, the Simulink diagram is unchanged from the one utilised in section 6, so isn't shown again.

$$G_p(s)=\frac{A}{s(1+\tau s)}=\frac{9.01214}{s(1+0.092)} \tag{7.1}$$

The tested output of the Simulink simulation verses the experimental data looks extremely similar at first glance, it rises faster, with a similar saturation of the real world implementation. However, closer inspection shows that there is a slight overshoot in the data. This is displayed in Figure 7.2.

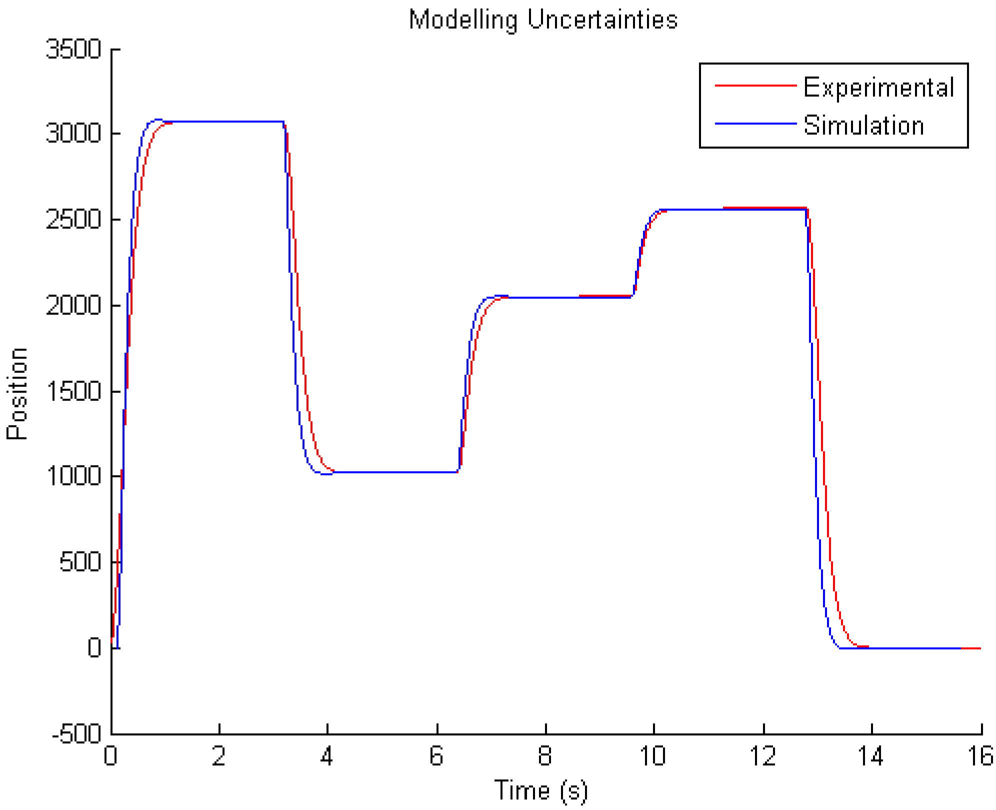

Figure 7.2. Uncertainty Modelling Graph vs ROT225 experimental data.

There are minor differences in the shape of the model, in addition to the overshoot mentioned previously. However, the steady state error is minimal in this particular implementation.

Figure 7.3 shows a close up of the overshoot in the uncertainty modelling. However, it's possible to see the controller bringing the arm back into position once it has detected the overshoot.

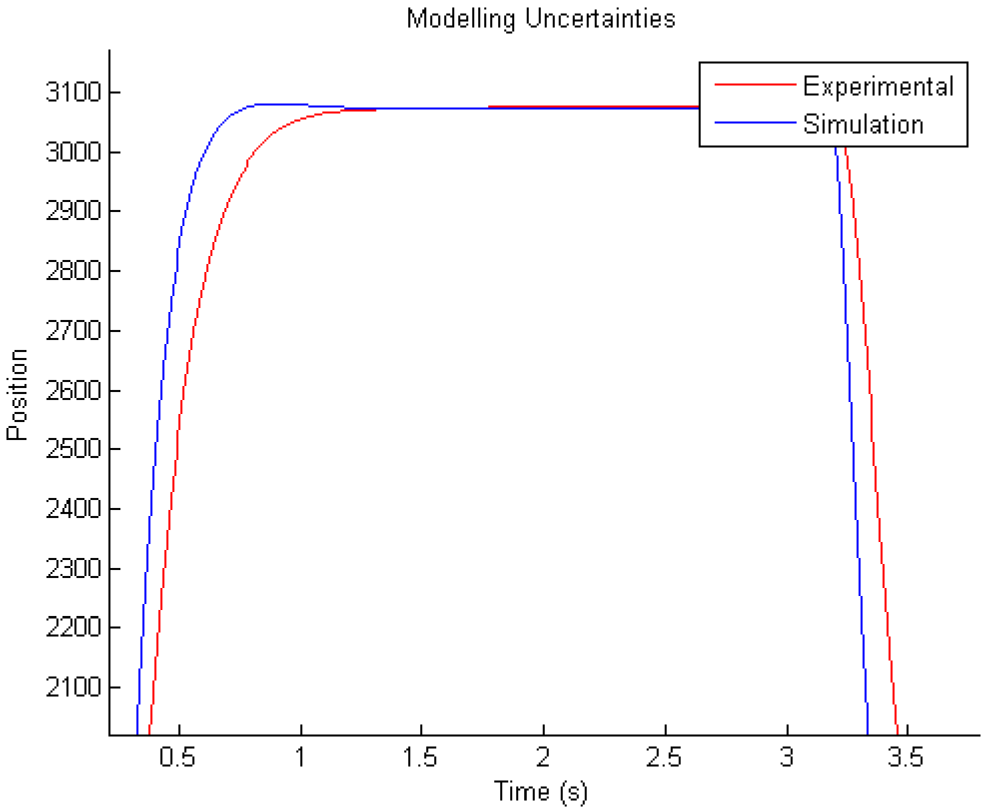

Figure 7.3. Uncertainty Modelling Close-up.

Each of the responses involved in this has a very similar shape between the system that the controller was designed for and the saturation caused the experimental data to not have a noticeable overshoot to the point.

8. Conclusion

The comparisons between the simulations and the experimental data are close matches, however the rise time in the simulation was much faster than the experimental data in every single case. There wasn't much noise that is noticeable in the measurements, however, there are reasons for inaccuracy. The major reason, and the most likely cause of the inaccuracies was the saturation of the real world system. There were no saturation limits programmed into the Simulink model, so they were unaccounted for. Subtle steps in the data that are possible to see when zoomed in are a result of the discrete time measurements. These discrete measurements ensure that the controller will not respond instantly to changes, this is most visible in the overshoot in the uncertainty modelling.

Overall, the controller performed extremely well, in modelling the ROT335 data once saturation is taken into account, however other possible sources of inaccuracy could have been caused by internal friction, air resistance or back emf, which was modelled in the previous experiment, from given values.

The difference in inertia was dealt with reasonably well in the uncertainty modelling stage, there was an extremely low steady state error, with only a small overshoot of the position, which the controller adapted to.

In conclusion, this experiment shows the controller is capable of controlling the system it was designed for and minor variations in the system. The simulation is a reasonably accurate representation of the real system to model inputs and effects, if saturation and forms of resistance were accounted for.