UNIVERSITY OF NEW SOUTH WALES

MTRN3020

Modelling and Control of

Mechatronic Systems

# Laboratory Experiment 2

---

## Design and Implementation of a Speed Controller

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

By Aidan O'Brien

z3324494

26 September 2014

# 1. Introduction

This laboratory experiment involved designing and then implementing a speed controller for an electric motor generator system. It uses the direct analytical method, also called Ragazzini's method to design the controller. The calculations form a difference equation that can then be implemented. The system is analysed for constants, which are then used to form $G_C(z)$, the controller transfer function, based upon given variables.

After the controller has been developed, experimental data is gathered from the motor generator system. This data is used for the remainder of the experiment to confirm that the simulation model is correct. Part A of the experiment is the design verification process, where a Simulink model is created, with the controller's transfer function applying to the generator. Set speed changes from 0 to 1000 and then 2000 rpm are made with no load added to the generator.

Part B shows the result of the controller when loads are applied to the system, through simulated resistors in Simulink. This disturbance rejection has the controller ensuring the correct speed is achieved when the load levels change.

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 speed controller for an electric motor generator system using the direct analytical method. The speed controller ideally will attain and/or maintain the set speed with minimal disruption due to loads applied.

# 3. Experimental Design Procedure

The procedure for the experiment involved developing the controllers transfer function, then we gathered the data upon the motor generator. The first dataset retrieved was two steps of 1000 rpm each. This experimental dataset is then utilised in the design verification stage, to ensure the Simulink model is correct. Afterwards, a second dataset is created, where a load is applied to the generator. The only changes to the Simulink model should be the load (to match those of the experiment) and the input RPM, as it will not be doing two steps. After the data is retrieved, the Simulink model is generated and tested. The idealised block diagram is given in the experiment assignment and is shown in figure 3.1.

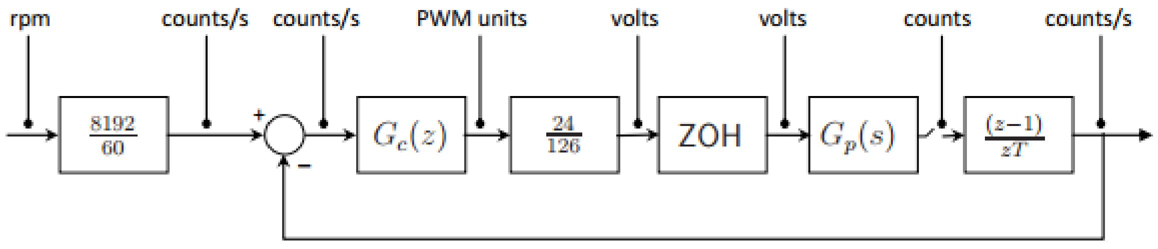


Figure 3.1. Idealised Block Diagram. From: MTRN3020 assignment specification.

This block diagram is broken down into the component parts for the plants controller.

# 4. Controller Design Calculation

The first stage of finding the controller design was to discover the gain (A) and time constant (τ) for the motor. This was taken from the noload.m data file. Utilising the following MATLAB code, shown in snippets 1 and 2, it was possible to gather the values.

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

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

Dividing the gain by the voltage applied (24V) resulted in a value of 31688 and a tau of 38 ms. Snippets 3 through to 7 show the code utilised to generate the graph comparing the theoretical responses.

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

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

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

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

These snippets then generate the graph shown in figure 4.1, which results in a nearly perfect alignment between the theoretical response and the experimental response.

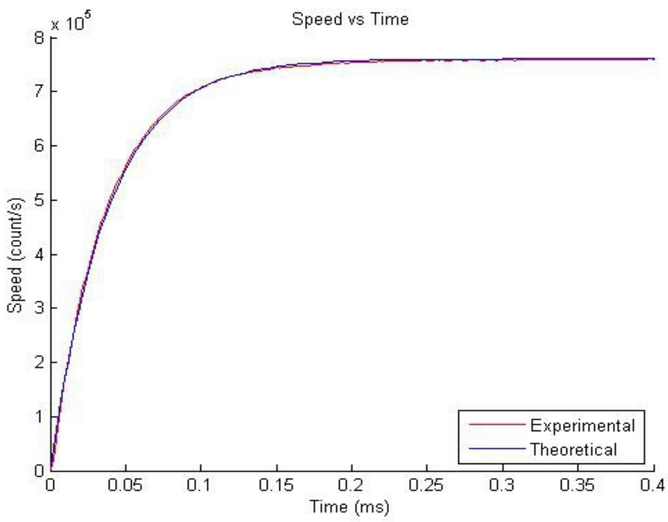


Figure 4.1. Graph of Speed vs Time of Experimental vs Theoretical responses.

Using the code snippet given, it was also possible to discover p1, z1 and C. Which corresponds to the first pole, first zero and a constant. The given and discovered values are listed in Appendix A. Utilising these values, it's possible to find the numerical values of Gp(s), as shown in equation 4.1.

$$G_p(s)=\frac{C(z-z1)}{T\cdot z(z-p1)}=\frac{593.01(z+0.93225)}{z(z-0.81016)} \tag{4.1}$$

Since z1 is between 0 and -1, it is ringing and requires us to utilise the version of F(z) where z1 is ringing as given in the design specification. Calculating B0 as shown in equation 4.2 allows for the direct solving of the controllers transfer function as shown in snippets 7 and 8.

$$B_0=\frac{1-e^{\frac{-T}{\tau_d}}}{1-z1}=\frac{1-e^{\frac{-0.008}{0.038}}}{1+0.93225}=0.098249 \tag{4.2}$$

$$e=\exp(-T/tau_d) \tag{7}$$

$$Gc=tf((T*B_0/C)*[1-p_1\,0],[1,-(B_0+e),B_0*z_1],T) \tag{8}$$

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.

```
      0.0001657 z^2 - 0.0001342 z
Gc =  ---------------------------
        z^2 - 0.9084 z - 0.09159

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 is absorbed by the arrangement of F(z) as given in the assignment specification.

# 5. Simulink Block Diagram

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

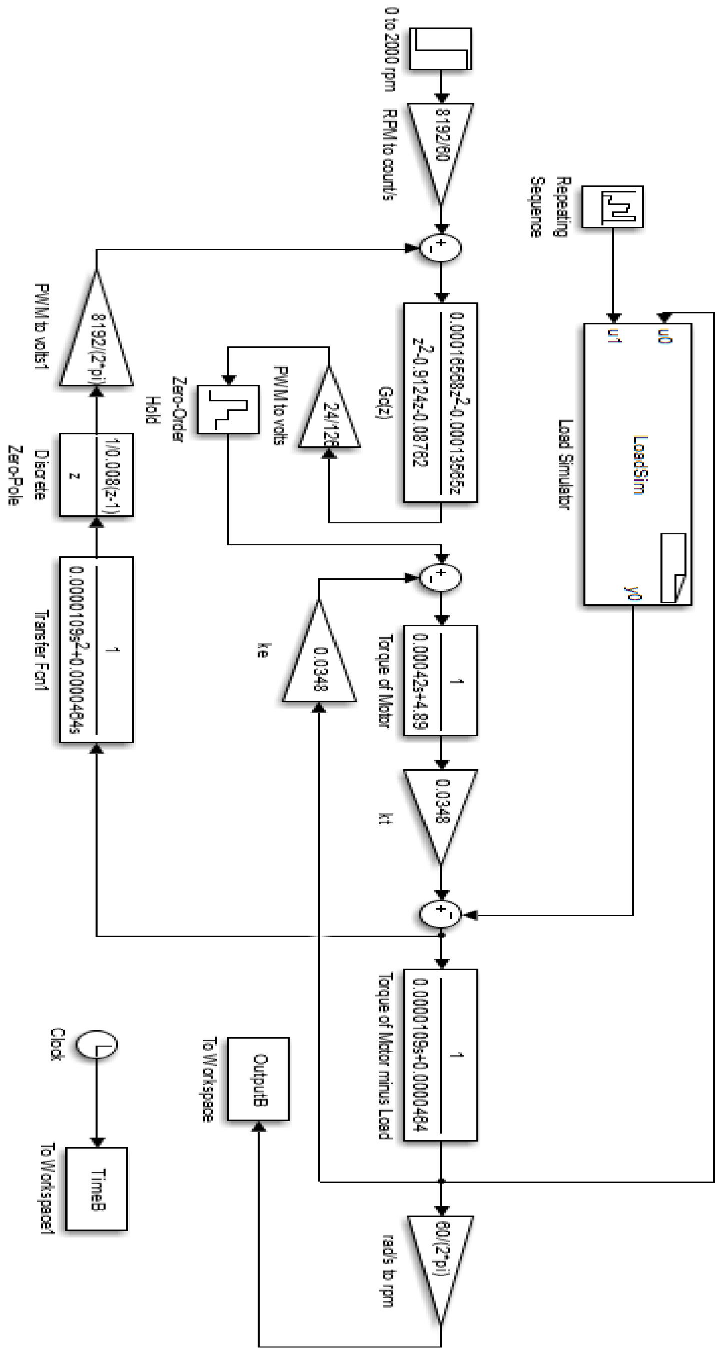


Figure 5.1. Working Simulink Diagram of Speed controller and Plant Transfer functions.

The Simulink block diagram is more complicated than the generic block diagram, to account for the different forces at work within the motor. $G_p(s)$ is extended out to include the various forms of torque.

The controller is the one developed in section 4, using Ragazzini's method. The motor's transfer function is a first order system, where the voltage is converted into torque. The systems overall dynamics also takes the form of a first order system. It is then converted back into rpm from the count/s before being returned as output.

# 6. Design Verification

To verify the Simulink block diagram's design, no load was applied to the simulation, with two step inputs, each of 1000 rpm. The first is active at 0 seconds and the second at 3.456 seconds. Shown below in figure 6.1 is the graph comparing the experimental data with the simulation data.

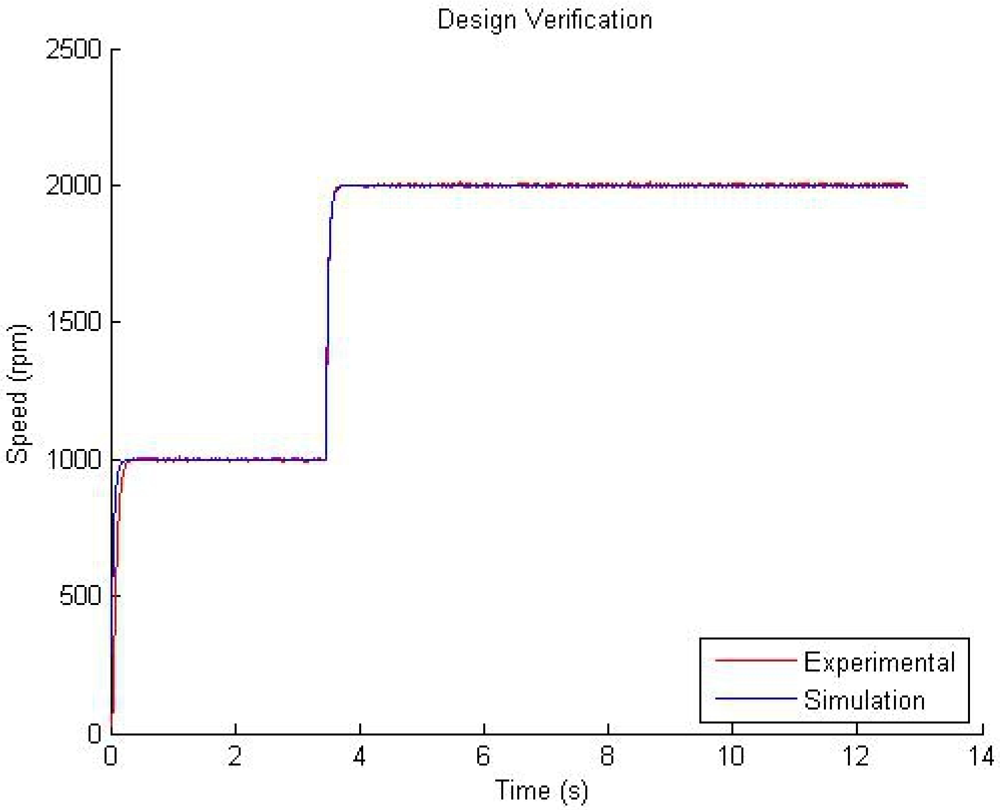


Figure 6.1. Design Verification Graph.

The figure shows that the model is extremely close to the experimental data. However the real data has noise in the readings, but the average reading is right in line with the simulation.

# 7. Disturbance Rejection

Disturbance rejection is where a controller can adapt to changing loads. The load for this experiment was generated utilising a number of resistors based upon the individual student number. This particular case was made by converting 3324494 into hexadecimal, which resulted in the combination of 32BA4E. With the changing of loads, the motor loses or gains speed depending on whether a load is added or removed respectively. This task is used to judge how sensitive the controller is to disturbances. Figure 7.1 shows the overall response to the disturbances.

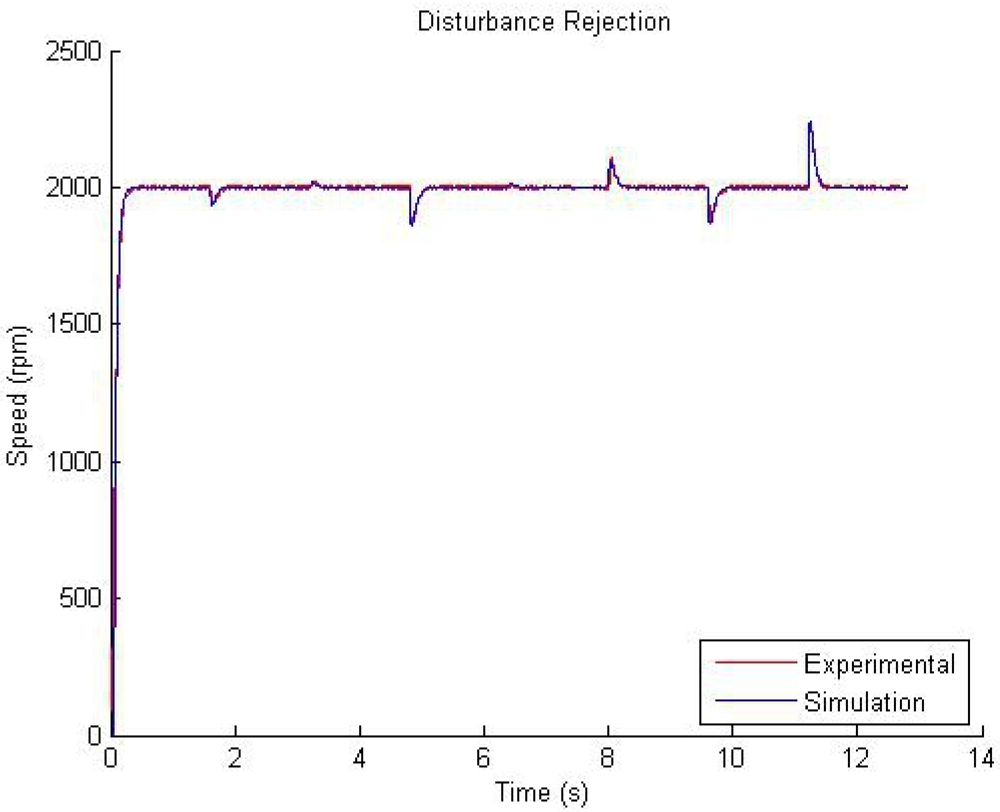


Figure 7.1. Disturbance Rejection Graph

The responses in the simulation map extremely close to the experimental data. Figure 7.2 is a close-up of the system regaining its set speed.

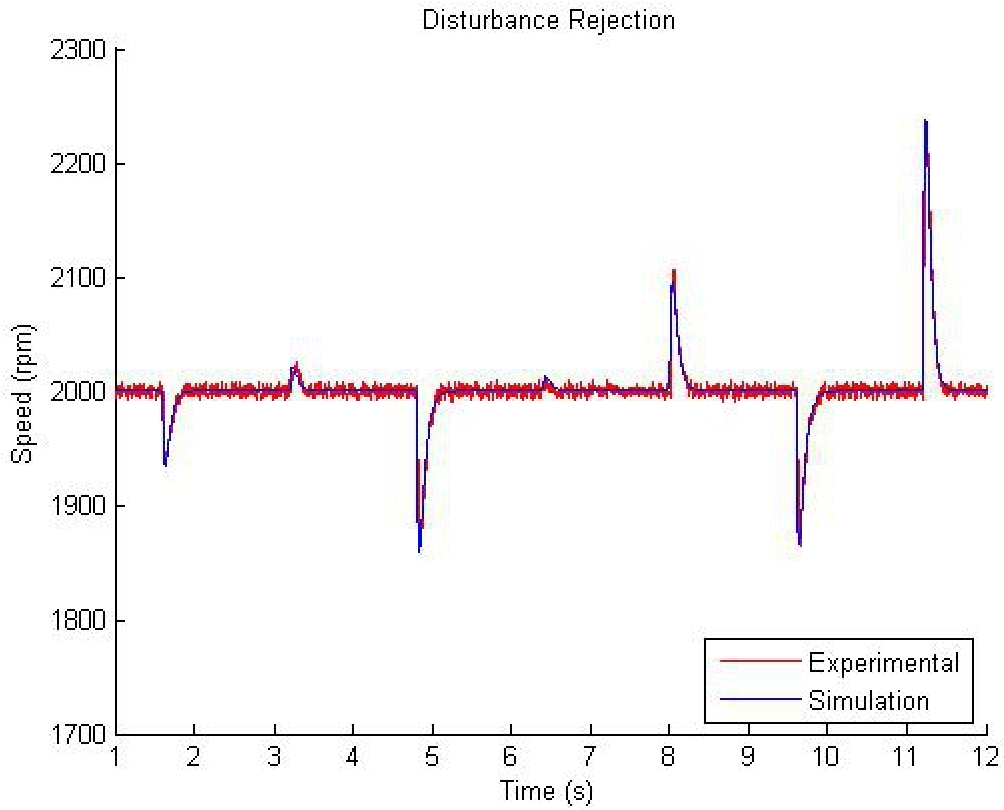


Figure 7.2. Disturbance Rejection Close-up.

As is shown, the simulation maps extremely close to the experimental data. The peaks and troughs match up almost perfectly. At no point was there a load large enough to prevent the motor from regaining the set speed.

Each of the response times has a time constant of 48ms as was assigned in the specification sheet, which shows the accuracy of this particular controller.

# 8. Conclusion

The comparisons between the simulations and experimental data match up extremely well. The major difference in the simulation is the leading rise in the design verification. This is most likely because of the start point of the simulation assuming starting exactly on 0 seconds, with the experimental data starting slower. The experimental data has a lot of noise present in the measurements, however the overall trend between simulation and experiment is very similar.

In conclusion, this experiment shows that the controller was well designed and that the simulation is an accurate representation of the real system that can be utilised to model inputs and effects.

# Appendix A – Table of Values

| **Name** | **Value** |
|---|---|
| Gain (A) | 31688 |
| Time Constant Model ($\tau$) | 38ms |
| Time Constant Given ($\tau$) | 48ms |
| Sampling Time (T) | 8ms |
| z1 | -0.93225 |
| p1 | 0.81016 |
| C | 4.74411 |