School of Mechanical and Manufacturing Engineering
Faculty of Engineering
University of New South Wales

Laboratory Experiment 2 – Design and Implementation of a Speed Controller

MTRN3020 - Modelling and Control of Mechatronic Systems

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

Gajendra Jayasekera
z5260252
12/04/2022

1. Introduction

This laboratory experiment aims to design and implement a speed controller in an electric motor-generator system to maintain a set speed. The controller will be designed using the direct analytical method or Ragazzini’s method [1]. This method will give the controllers transfer function which will be put into a difference equation form to obtain the coefficients required for the laboratory experiment. Part A of the experiment will verify the designed speed controller is correct. This will be performed using a Simulink model using the designed controller and checking that the steady-state error is 0 and the time constant matches the specified value in Table 1. Part B of the experiment will analyse the disturbance rejection of the controller. This will be performed using the previously created Simulink model and ensuring the controller maintains the specified speed when the load on the system varies. Both simulated results from Parts A and B will be compared against the experimental data to determine the validity of the simulations and designed controller and draw parallels to real-world applications.

Table 1. Design Parameters

Design Parameter	Desired Time Constant (s)	Sampling Time (s)
Value	0.045	0.006

2. Aim

This experiment aims to design and model a controller using the direct analytical design method or Ragazzini’s method to control the speed of a motor-generator system. This controller will aim to generate a signal that will attain and/or maintain the speed of the motor at a specified value, regardless of the loads experienced by the generator. The implementation of the experiment and the simulation of the model through MATLAB’s Simulink will produce experimental and simulation results to analyse and discuss.

3. Experimental Procedure

Using the direct analytical method or Ragazzini's method, calculate the transfer function for the speed controller. Convert the transfer function into a difference equation form to obtain the coefficients of the controller. Using the calculated coefficients, run the experimental simulation for Part A and observe the change in the motor speed from 1000 rpm to 2000 rpm under no-load and save the results. Once complete, enter your zID and calculated coefficients and run the experimental simulation for Part B. Observe the change in the motor speed under varying loads and save the results. Note that the loads used are calculated using the hexadecimal of your zID.

Model the experimental setup in MATLABs Simulink using the provided design parameters and designed controller. Simulate Part A and B of the experiment using the designed model and respective resistor loads. Save the simulation results for both parts and superimpose the experimental and simulation results for Parts A and B on separate graphs and compare the compatibility of the results.

4. Controller Design Calculation

The design of the speed controller was based on the block diagram seen below in Figure 1.

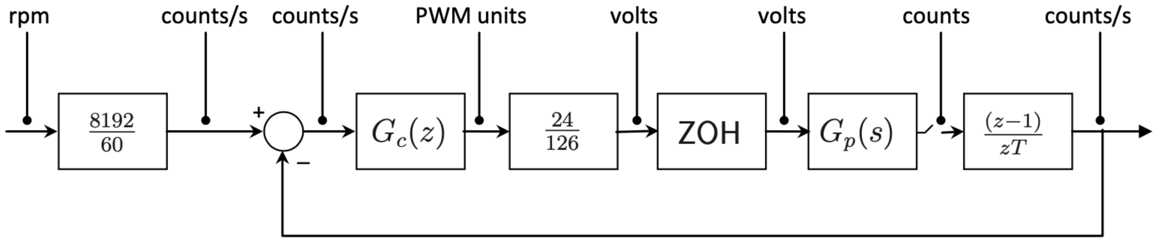

Figure 1. Speed Control Block Diagram [1]

By plotting motor speed versus time, the response of the motor when an input voltage of 24V was applied under no-load was graphed. Through first-order response fitting to the motor response as seen in Figure 2, a transfer function was obtained that related the applied voltage to the speed in counts/second.

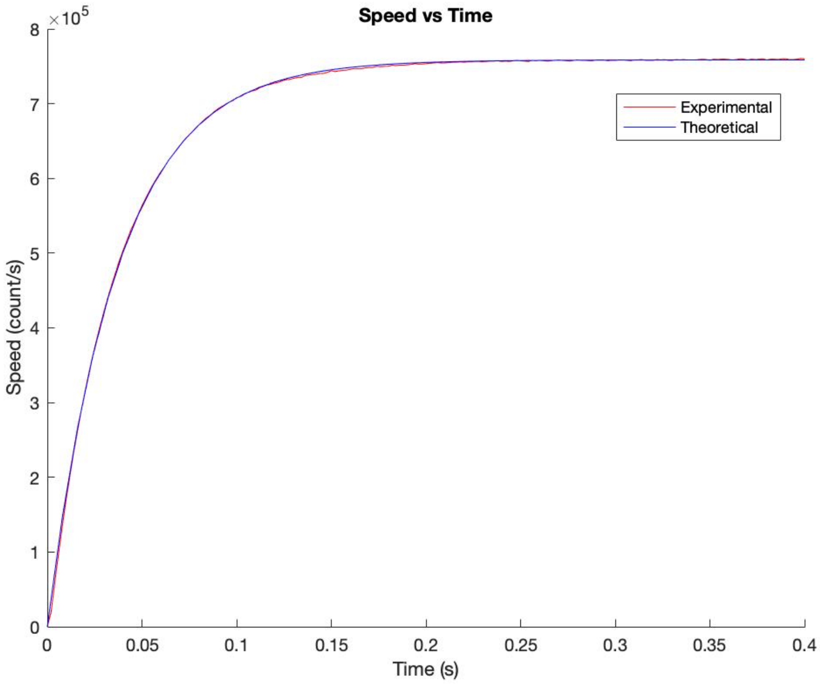

Figure 2. Experimental and Theoretical Response Plot for Speed vs Time of a Motor

$$G_{p1}(s) = \frac{A}{1 + \tau s} = \frac{31607.5000}{1 + 0.0370s} \tag{1}$$

Since the encoder used in the experiment samples the counts per second and not speed, an integrator was applied to the transfer function to relate applied voltage to the speed in counts/second.

$$G_p(s) = \frac{31607.5000}{s(1 + 0.0370s)} \tag{2}$$

By combining all the blocks except $G_c(z)$ in Figure 1, the plant transfer function is obtained as follows.

$$G_p(z) = \mathcal{Z}\left[\frac{24}{126}\frac{(1 - e^{-st})}{s}\frac{31607.5000}{s(1 + 0.0370s)}\frac{(z - 1)}{zT}\right] \tag{3}$$

This can be broken into two parts,

$$G_p(z) = G(z)\frac{(z - 1)}{zT} \tag{4}$$

To determine $G(z)$, MATLAB's $\mathrm{c2dm}$ function was used with a sampling time of 0.006 seconds as required in Table 1.

$$G(z) = \frac{2.7768(z + 0.9474)}{(z - 1)(z - 0.8503)} \tag{5}$$

Substituting (5) into (4), $G_p(z)$ is now,

$$G_p(z) = \frac{462.7963(z + 0.9474)}{(z - 0.8503)} \tag{6}$$

$F(z)$ is obtained using the following equation,

$$F(z) = \frac{b_0}{z - z_p} \tag{7}$$

The desired time constant is 0.045 seconds as seen in Table 1. This can be used to calculate the desired pole in the z-plane.

$$z_p = e^{sT} = e^{\frac{-0.006}{0.045}} = 0.8752 \tag{8}$$

Hence, $F(z)$ is now,

$$F(z) = \frac{b_0}{z - 0.8752} \tag{9}$$

The plant obtained in (6) has a zero at $z = -0.9474$ which is stable but ringing. To eliminate the ringing, the ringing zero of the plant must be embedded in the numerator of $F(z)$.

$$F(z) = \frac{b_0(z + 0.9474)}{z - 0.8752} \tag{10}$$

To satisfy the causality constraint, an additional z is placed in the denominator.

$$F(z) = \frac{b_0(z + 0.9474)}{z(z - 0.8752)} \tag{11}$$

Applying the zero steady-state requirement, $F(1) = 1$, b_0 is found, and $F(z)$ is now,

$$F(1) = \frac{b_0(1 + 0.9474)}{1(1 - 0.8752)} = 1 \tag{12}$$

$$b_0 = 0.06409 \tag{13}$$

$$F(z) = \frac{0.06409(z + 0.9474)}{z(z - 0.8752)} \tag{14}$$

The transfer function of the controller, $G_c(z)$, is obtained using the expression,

$$G_c(z) = \frac{1}{G_p(z)} \frac{F(z)}{\left(1 - F(z)\right)} \tag{15}$$

Hence, substituting (6) and (14) into (15), $G_c(z)$ is obtained as,

$$G_c(z) = \frac{0.0001z^2 - 0.0001z}{z^2 - 0.9393z - 0.0607} \tag{16}$$

The speed controllers transfer function can be put into a difference equation form to obtain the coefficients of the controller.

$$\frac{M(z)}{E(z)} = \frac{0.0001 - 0.0001z^{-1}}{1 - 0.9393z^{-1} - 0.0607z^{-2}} \tag{17}$$

$$m(k) = 0.9393m(k-1) + 0.0607m(k-2) + 0.0001e(k) - 0.0001e(k-1) \tag{18}$$

5. Simulink Block Diagram

The system used during the experiment is modelling using MATLAB's Simulink. The block diagram is shown below in Figure 3.

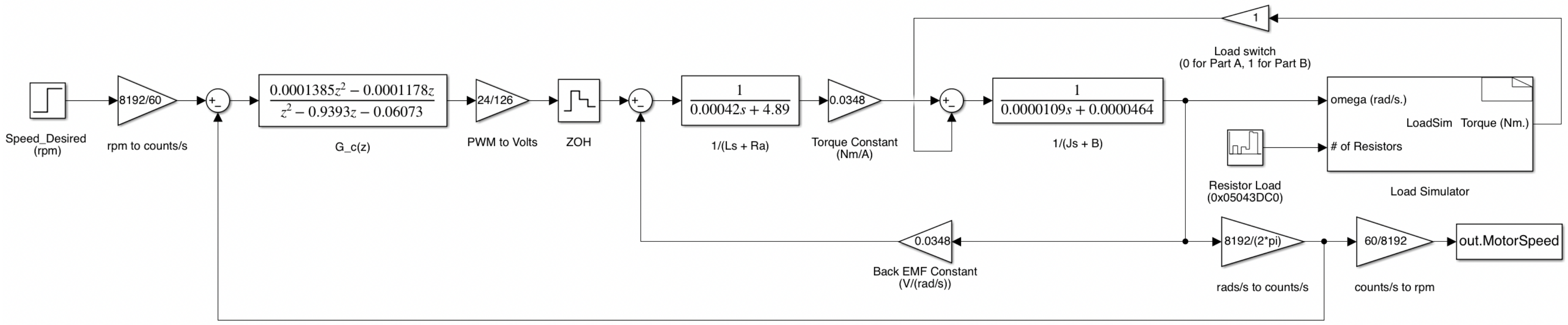

Figure 3. MATLAB Simulink Block Diagram for Speed Controller Experiment

The model takes in the desired motor speed in rpm as a step input. It is converted from rpm to counts/s before the error is calculated at the summation node through the input and a feedback loop. This is passed into the speed controller designed in the above section, Controller Design Calculation, before it is converted from PWM to volts and passed into a Zero Order Hold. The voltage is passed through a second summation node and transfer function to output the motor current. It is multiplied by the torque constant to calculate the motor torque and passed through a third summation node and transfer function to calculate the shaft speed in rad/s. The shaft speed is passed through several blocks. Firstly, it is used as an input into the given load simulator block which additionally takes the resistor load calculated from the hexadecimal conversion of the zID (5260252 = 0x05043DC0) to calculate the torque which is passed into the previously mentioned third summation node. This output has a switch to turn on and off the torque for the respective parts of the experiment. Similarly, the shaft speed is converted to volts using the back EMF constant and counts/s and fed back into the second and first summation nodes respectively. Finally, it is converted from rads/s to rpm and gives the desired output of motor speed.

6. Part A – Design Verification Results and Discussion

The superposition of the experimental and Simulink plot for Part A is seen below in Figure 4.

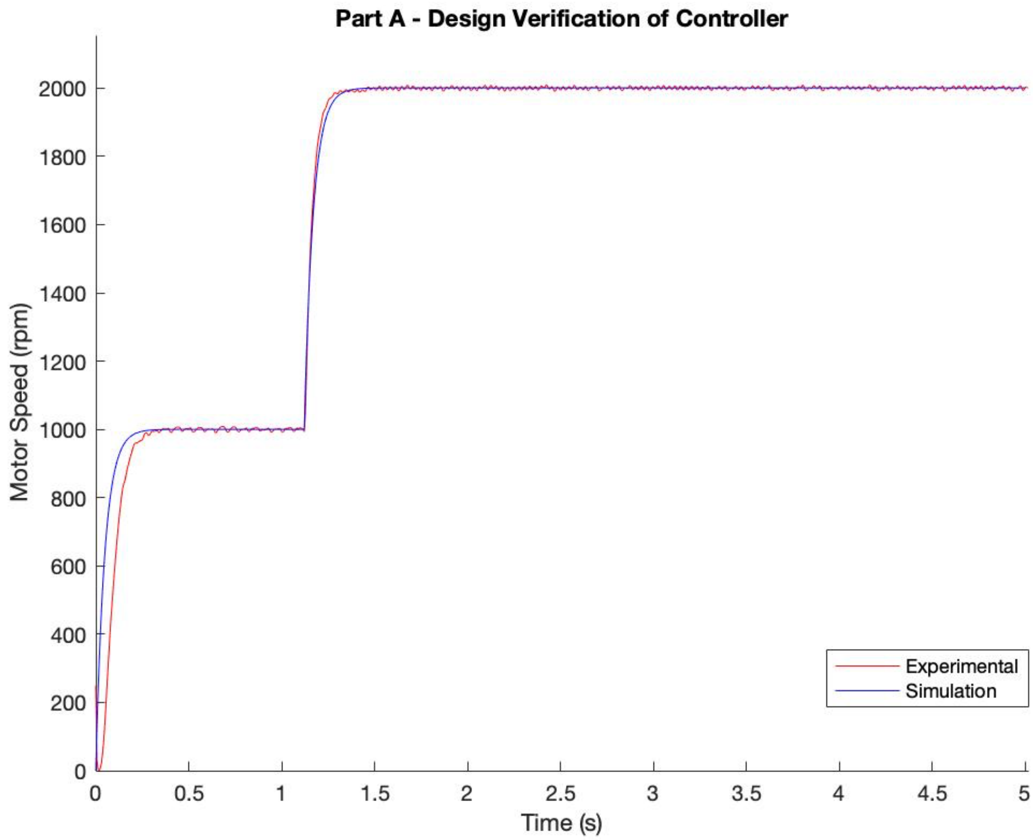

Figure 4. MATLAB Plot of Experimental and Simulated Motor Speeds for Designed Controller Under No Load

By visual inspection, the experimental and Simulink model data are seen to follow a similar plot. It is observed the desired 0.045 second time constant is achieved when the step change from 1000 rpm to 2000 rpm occurs and that the system exhibits accurate steady-state values at 1000 and 2000 rpm with the simulated values settling within 1-3% of the experimental values implying the designed controller is correctly designed.

There are however several deviations between the two data sets. The first is the experimental and simulated plots vary for the jump from 0 to 1000 rpm. This difference can be attributed to the simulated model not being able to accurately replicate the timing of the experimental setup. The second difference is the experimental data exhibits an oscillatory behaviour at steady state that is not present in the simulated plot. This difference can be attributed to several factors. Firstly, the oscillations seen could be the result of physical limitations such as inadequate damping to eliminate small vibrations. Secondly, the encoder used in the experimental setup may be prone to a measurement error where the counts/s are not accurately recorded. This would result in inaccurately calculated shaft speeds, which are fed back to the controller causing the slight oscillations seen in the experimental plot. The third possibility is that due to the computational delay between the setup and the computer, certain changes in the experimental setup are not recorded during processing. Hence, when the calculated control signal is sent to the system it is not the appropriate signal for the current state of the setup thus creating the oscillations observed. Finally, the oscillatory behaviour may be accounted for by the accuracy level of the coefficients used by the controller. The simulated controller used coefficients accurate to 4 decimal places whilst the experimental setup used coefficients accurate to 6-7 decimal places. The difference in accuracy levels introduces a small difference between the calculated signals for the experimental and simulated hence a possible reason for the oscillatory behaviour.

7. Part B – Disturbance Rejection Results and Discussion

The superposition of the experimental and Simulink plot for Part B is seen below in Figure 5 with a close up of the plot in Figure 6.

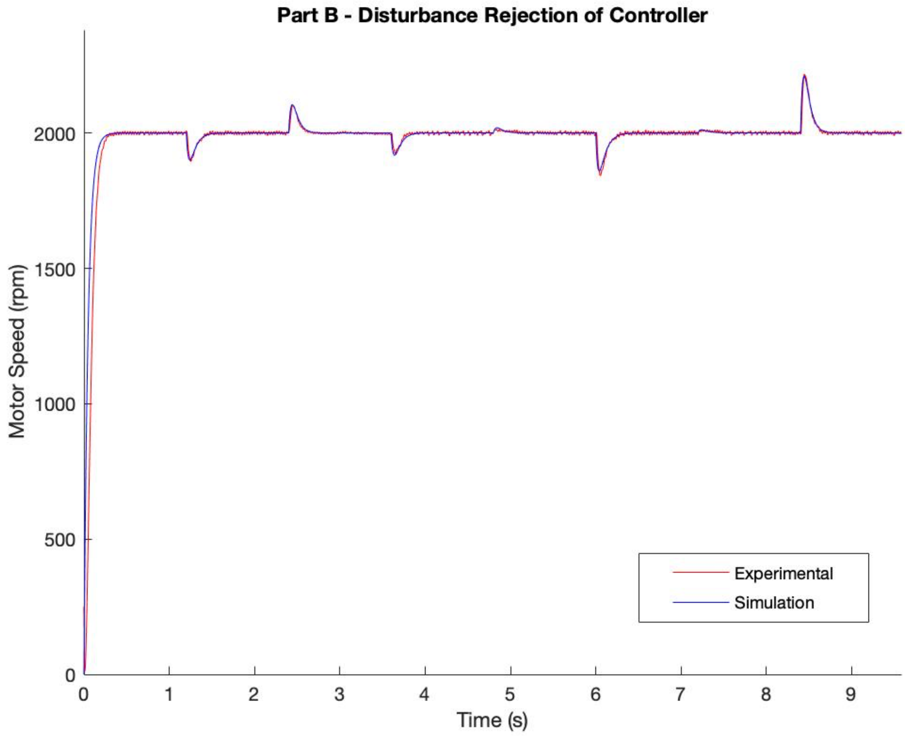

Figure 5. MATLAB Plot of Experimental and Simulated Motor Speeds for Designed Controller Under Varying Load

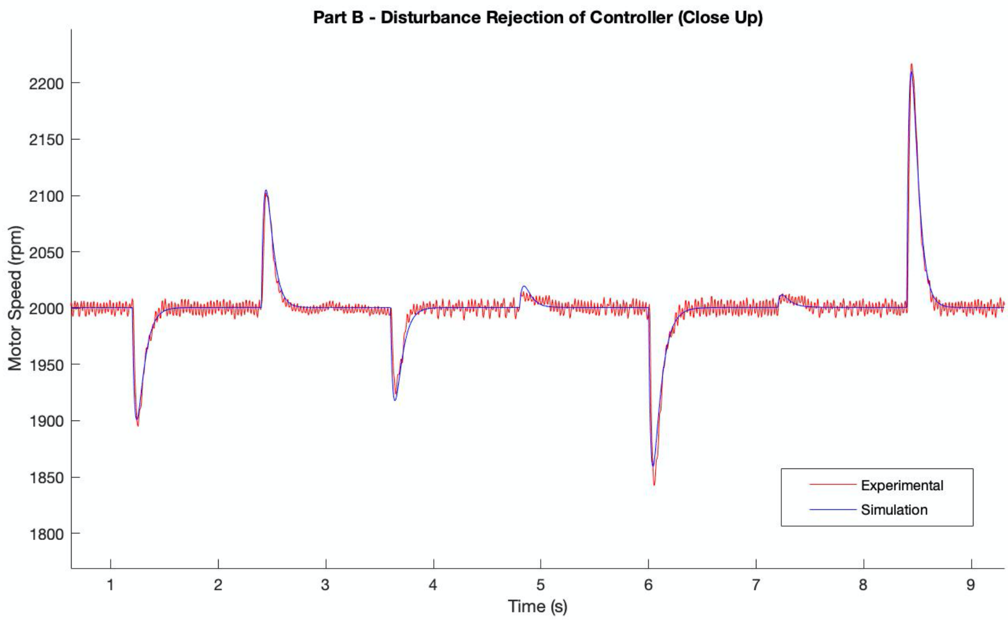

Figure 6. MATLAB Plot of Experimental and Simulated Motor Speeds for Designed Controller Under Varying Load (Close Up)

By visual inspection, the disturbance rejection of the simulated plot matches that of the experimental plot. The number of resistors active at 1.2-second intervals were 5, 0, 4, 3, 13, 12 and 0. The simulation follows the predicted behaviour, momentarily gaining speed when a load is removed and momentarily losing speed when a load is added [2]. This is seen when 5 resistors are activated at 1.2 seconds and then removed at 2.4 seconds. Despite the changing loads, the designed controller is able to successfully return to the specified speed of 2000 rpm within the specified time of 0.045 seconds.

There are some differences between the two plots. Similar to Part A, an oscillatory behaviour is observed in the experimental plot. As discussed in Part A this behaviour is due to either small vibrations in the experimental setup, encoder error, computational delay or the level of accuracy of the controller design or a combinate of the four.

8. Conclusion

In conclusion, the direct analytical method or Ragazzini's method was used to derive the transfer function for a speed controller cable of setting or maintaining a specified speed. This method also helped design a controller capable of achieving the desired response with an acceptable level of accuracy. Extrapolating this experiment to a real-world setting, the prime mover would be substituted for a gas, hydro or steam turbine and the resistors in parallel would be replaced by houses, companies, and public transport assuming the generator was connected to the national electricity grid. In this setting, the controller would be able to maintain or change to a required speed during peak and off-peak hours of usage as required to meet the needs of demand on the grid.

References

[1] J. Katupitiya, *Laboratory Experiment II : Design of a Speed Controller,* 2022.

[2] J. Katupitiya, *Laboratory Experiment II : Implementation of a Speed Controller.*