MTRN3020: Modelling & Control of Mechatronic Systems

Speed Controller Experiment

The University of New South Wales

Faculty of Engineering

School of Mechanical and Manufacturing Engineering

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

Scott Morris
z5165456
April 16, 2021

Contents

1 Introduction

A prime mover is a system that converts some form of energy into motive energy. Some examples of prime movers are gas turbines, steam engines or hydro turbines. Generators can be connected via a coupling to these prime movers to convert the motive energy into an output voltage. Generators are commonly connected to a number of different loads. For example, different houses with washing machines, dish washers can be the various loads connected to a generator. Furthermore there could be other loads such as factories. These loads can be connected and disconnected from the generator, causing the total load connected to the generator to change. If no other systems were considered, then the speed of the shaft of the prime mover would change as the load changes.

However, this is an undesirable situation as this can cause unpredictable behaviour of the system. Instead the speed of the shaft delivering energy to the generated should be constant and so the system exhibits predictable and manageable behaviour. In order to achieve this, there needs to be a system that can respond to the changes in load and either increase or decrease the speed of the shaft to correspond to this change in load.

Control systems are useful for this purpose. If the load changes, then the state of the shaft speed can be fed back and the controller can change the drive signal that is sent to the prime mover to maintain a near constant shaft speed. The design of this controller is vital as it can determine whether the system is unstable, responds quickly, has oscillations and can influence many other outcomes. Ragazzini's method can be implemented in order to produce a controller that achieves the desired outcomes.

2 Aim

The outcome of this experiment is to see whether a digital speed controller can be designed so that it outputs a voltage to the motor that allows a specified speed to be maintained despite load changes.

In order to achieve the aforementioned controller, Ragazzini's method will be learned and applied as shown in [1] and [2].

3 Experimental Procedure

In this experiment, the prime mover that is being used is an electric motor. The shaft of the electric motor is joined to a generator via a coupling. This generator is connected through wires to a variable load. The variable load was a set of 150 Ω resistors. This set up is replicated in diagrammatic form in *1*. The speed of the electric motor shaft is measured through an encoder. This encoder sends this data back to the control computer which implements a control system. The generated control signal is sent back into the electric motor.

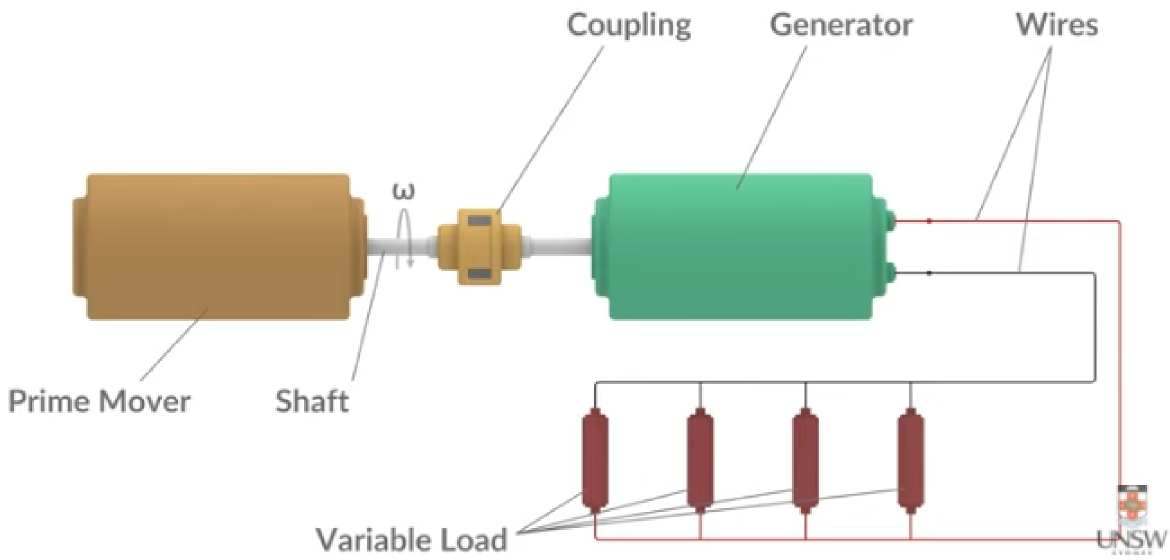

Figure 1: Diagram of experimental speed controller experiment [3]

There were two test conditions that were experimented with in order to test the validity of the controller. The sampling time for both tests was 8 seconds.

First, the motor was run at a constant speed of 1000rpm with no loads applied. Then, the desired speed was set to 2000rpm, and the response of the system was recorded.

Second, the desired speed of the motor shaft was set to 2000rpm and the loads connected to the generator were varied. The output speed of the motor shaft was recorded.

4 Controller Design Calculations

By fitting a general first order response to the expected response of the system with no load, the plant of the system can be estimated. *Figure 2* shows the response with the desired parameters.

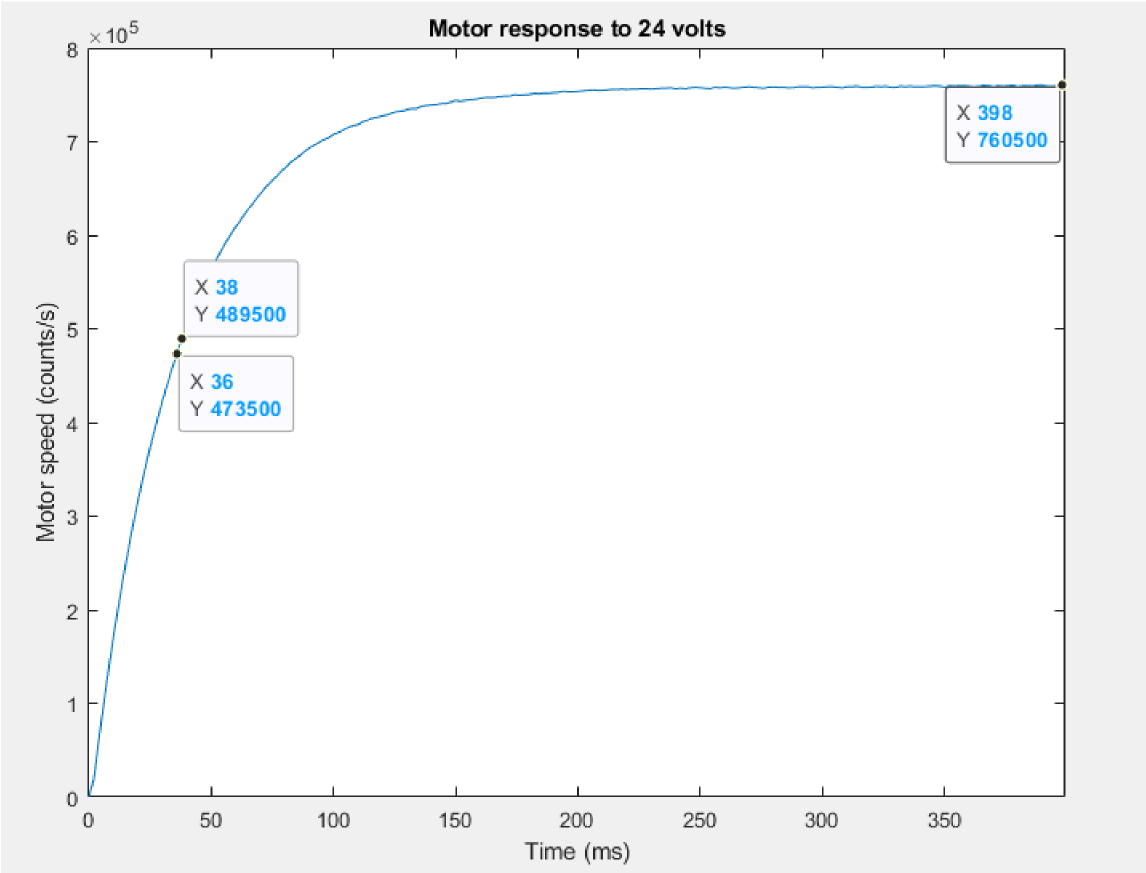

Figure 2: MATLAB plot of motor response to a 24 volt input [4]

Relating the voltage applied to the output speed results in the following transfer function shown in (1), with the final motor speed divided by 24 to account for the fact that 24 volts was applied to the system.

$$G_{p1}(s) = \frac{31689}{1 + 0.0375s} \tag{1}$$

However, the encoder samples counts per seconds and not speeds. Therefore we need to apply an integrator to the transfer function.

$$G_p(s) = \frac{31689}{s(1 + 0.0375s)} \tag{2}$$

However, the conversion from analog to digital and back to analog needs to be considered in the transfer function of the plant.

$$G_p(z) = \mathcal{Z}[\frac{24}{126}\frac{(1-e^{-sT})}{s}\frac{31667}{s(1+0.0375s)}\frac{(z-1)}{zT}] \tag{3}$$

$$G_p(z) = 160848.2540\mathcal{Z}[(1-e^{-sT})\frac{1}{s^2(s+26.6667)}\frac{(z-1)}{zT}] \tag{4}$$

To determine the Z transform of this equation, the MATLAB function 'c2dm' can be used with the sampling time of 0.008 seconds.

$$G_p(z) = \frac{4.7997(z+0.9313)}{(z-1)(z-0.8079)}\frac{(z-1)}{0.008z} \tag{5}$$

$$G_p(z) = \frac{599.9625(z+0.9313)}{z(z-0.8079)} \tag{6}$$

It is given that the desired time constant is 0.044 seconds. From this we can calculate the desired pole in the s-plane.

$$s = \frac{-1}{0.044}$$

$$s = -22.7273$$

Now, the desired pole in the z domain can be calculated.

$$z_p = e^{sT}$$

$$z_p = 0.8338$$

Hence, the final controller equation can be obtained using the following equation.

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

However, the plant has a zero at z=-0.9313 which is stable as it is inside the unit circle, but will cause ringing in the system. Therefore, this must be absorbed by in the numerator of F(z).

$$F(z) = \frac{b_0(z+0.9313)}{(z-z_p)} \tag{8}$$

However, due to the causality constraint, the F(z) must have a pole-zero deficiency equal to (or greater than) the plant, i.e. the denominator of F(z) must have pole-zero deficiency of 1. So a delay is introduced to the controller.

$$F(z) = \frac{b_0(z + 0.9313)}{z(z - 0.8338)} \tag{9}$$

The zero steady state requirement is applied.

$$F(1) = \frac{b_0(z + 0.9313)}{1(1 - 0.8338)} = 1 \tag{10}$$

$$b_0 = 0.08606 \tag{11}$$

$$F(z) = \frac{0.08606(z + 0.9313)}{z(z - 0.8338)} \tag{12}$$

So the transfer function of the controller can be obtained using the following expression.

$$G_c(z) = \frac{1}{G_p(z)} \frac{F(z)}{1 - F(z)} \tag{13}$$

$$G_c(z) = \frac{0.08606z^5 - 0.06114z^4 - 0.07361z^3 + 0.05399z^2}{600z^5 - 493.4z^4 - 567.8z^3 + 423.9z^2 + 37.34z} \tag{14}$$

$$G_c(z) = \frac{0.0001434z(z - 0.8079)}{(z - 1)(z + 0.0801)} \tag{15}$$

Therefore the transfer function can be written.

$$\frac{M(z)}{E(z)} = \frac{0.0001 - 0.0001z^{-1}}{1 - 0.9199z^{-1} - 0.0801z^{-2}} \tag{16}$$

This can be put into a difference equation form to obtain the coefficients of the controller which match very closely the provided coefficients for the same controller.

$$m(k) = 0.9199m(k-1) + 0.0801m(k-2) + 0.0001e(k) - 0.0001e(k-1) \tag{17}$$

5 Simulink Block Diagram

The system can be modelled via a block diagram. The block diagram was produced using Simulink and can be seen in *Figure 3*.

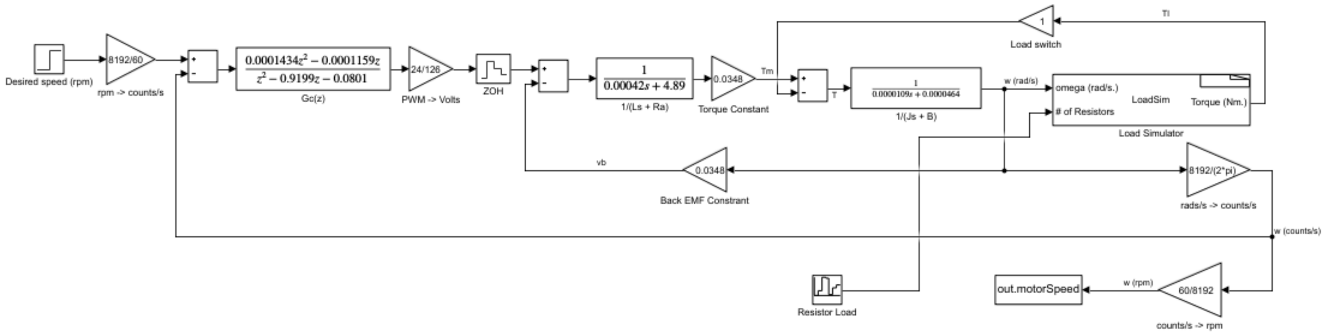

Figure 3: MATLAB Simulink Block Diagram of Speed Controller Experiment [4]

First, the desired motor speed (in rpm) is given as a step input and converted to counts/s. The error is then calculated through a feedback parameter. This is put into the controller designed above and multiplied by a gain to convert to volts. As this value is discretised, a Zero Order Hold must be applied. The motor current, torque and finally shaft speed (in rads/s) are obtained using the equations supplied in [2]. This needs to be converted to counts/s to be fed back to calculate the error in the system and is also converted to rpm as an output. Also in this diagram, a Resistor Load is applied which is a sequence of values corresponding to the number of resistors used to calculate the load torque. A switch is connected to the Load Torque in order to turn off the influence of the loads for the first part of the experiment and switch them on the the second part.

6 Part A

Both the simulated and experimental results for the system in response to a step input changing from 0 to 1000 to 2000 rpm within 5 seconds are shown in *Figure 4*.

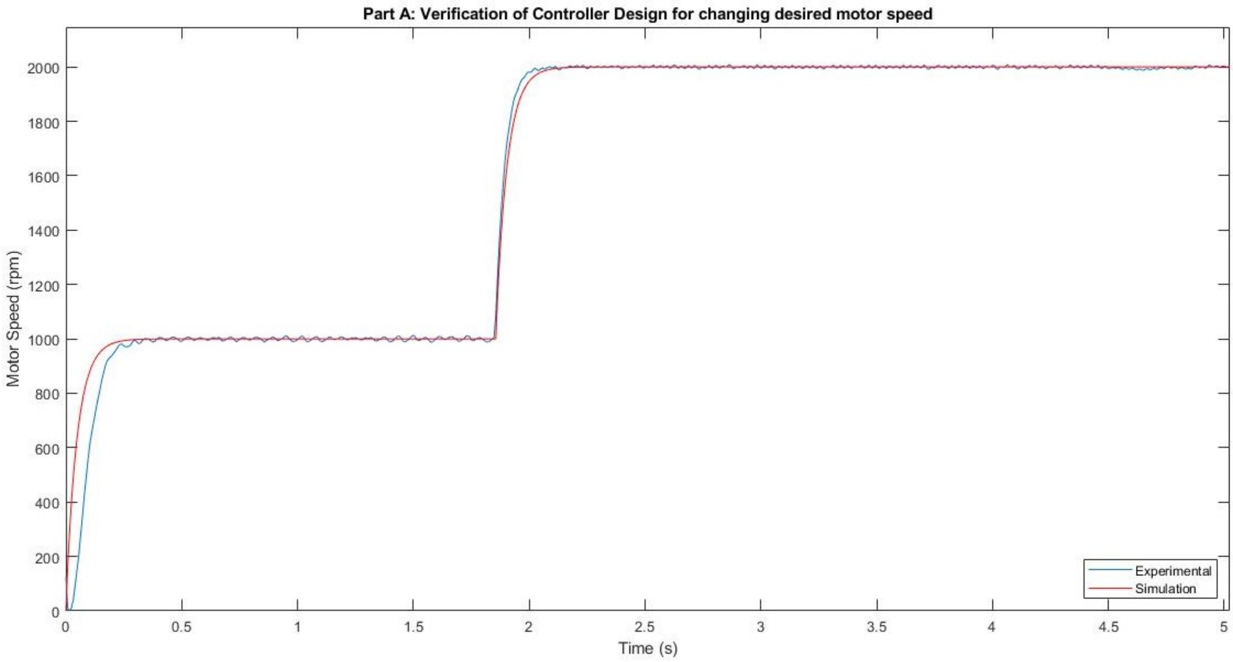

Figure 4: MATLAB plot of temporal experimental and simulated motor speed with designed controller and no load

Clearly, the experimental data and simulated data match closely, which indicates that the controller derived from the time constant was very effective. From a visual inspection of the graph, both changes in the experimental data occur within the desired 44ms time constant. Furthermore the accuracy, i.e. the steady state error of the response is very good, with the experimental data settling within 1% of the desired value as the system reaches steady state. However, there are some discrepancies between the two sets of data. First of all, there is some difference in the motor change timing, more so for the first change (0 to 1000rpm). This is to be expected as the timing of the change of the desired motor speed cannot be replicated accurately between the experimental and simulated data and so it appears that the change for the experimental data was delayed slightly more than in the simulation. Second, there is an oscillatory-like behaviour in the experimental data which is not present in the simulated data. This can be explained through several different factors and will be explained later in this report.

7 Part B

Simulated and experimental results for Part B are shown in *Figure 4*. The systems desired speed was set to 2000rpm and a number of loads were introduced to see how the system responded. The number of resistors, which were incrementally at 0.8 s intervals, was 4, 14, 13, 1, 6, 8. This was obtained by converting 5165416 into hexadecimal form. The number used should have been 5165456, however there was an error with entering the values for the controller and hence 5165416 was used.

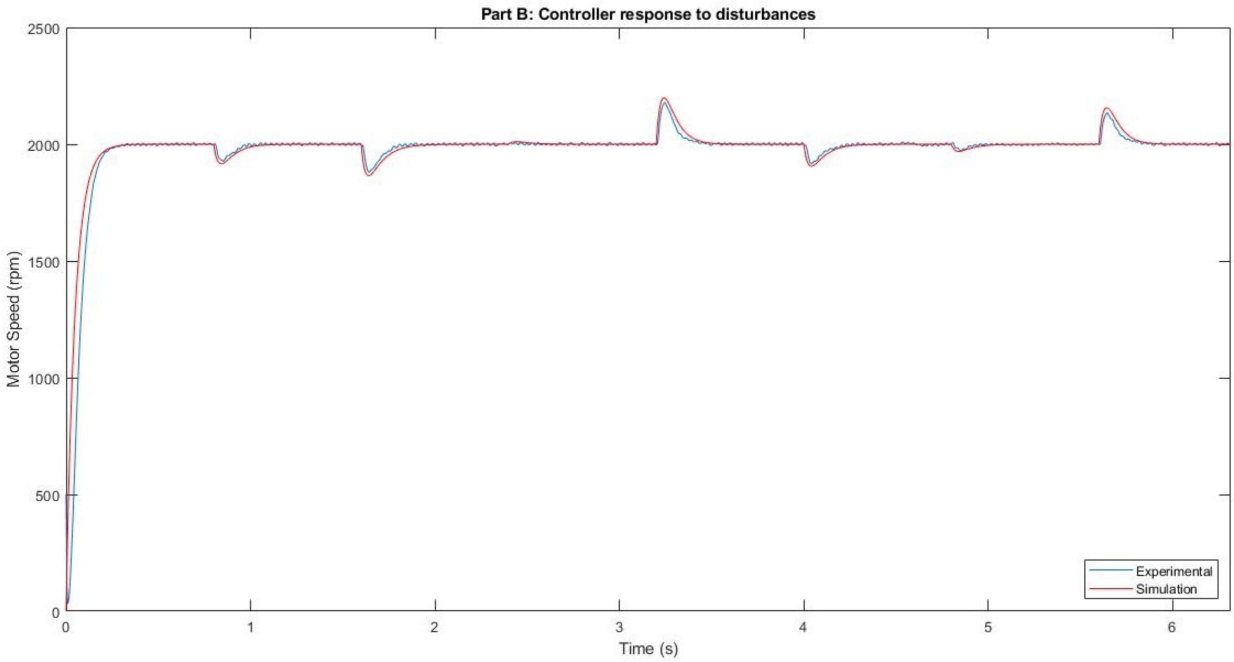

Figure 5: MATLAB plot of temporal experimental and simulated motor speed with designed controller and varying loads

As in the previous experiment, the simulated and experimental results followed each other very closely. With respect to the time constant, the experimental data responded within the specified time of 44 ms. The steady state error again was shown to be negligible as the system settled to within 1% of 2000 rpm each time the load was changed. Overall, the controller produced was therefore very successful for this situation as both the accuracy and speed of the response was within the design constraints. Again there are some slight differences between the experimental data and the simulated data, for example the oscillatory behaviour of the experimental data. Factors that have contributed to this will be explained in the following section.

8 Experimental and Simulation discrepancies

In both sections there are some differences between the simulated and experimental results. There are a few main contributing factors to these sources of error.

The computational delay between the system and the control computer introduced some error. The system is constantly evolving and there is inherent delay between the system feedback reaching the algorithm and then delay again as the algorithm output is applied to the system. During this delay, the system may change and render the feedback or output out of date and hence the signal sent to the system will not be perfectly accurate for that instant. This will require another correction that again occurs this same problem, causing more oscillation in the experimental data.

Another source of error is introduced when the numbers determined as inputs to the control system are rounded to a certain number of decimal places. The coefficients of the controller in this report were rounded to 4 decimal places. However, this will mean that the output of the algorithm again will not perfectly match the simulated system response.

Finally, the measurements of the encoder have an inherent error in them. The counts of the encoder cannot be assumed to be perfectly accurate and so the calculation of the motor shaft speed will be inaccurate to some degree. This will propagate throughout the system including the feedback. Hence, the experimental data will differ from the simulated results which dont suffer from this measurement innacuracy.

All these factors contribute to a real life system which behaves slightly different to the same theoretical system. However, there are also more less significant contributing events which are not mentioned in this report.

9 Conclusion

In conclusion, a direct analytical method can be applied to derive a controller that can manipulate the rotation speed of a motor. This approach can help to achieve a desired response speed and accuracy. It was revealed that the experimental data will inherently be less smooth due to the disturbances and noises in a practical setting. However, the overall characteristics of the response can be closely matched to a desired specification. The design methodology explained in this report can be applied in real life. Extending the example mentioned in the introduction, a generator is connected to a motor to produce an output voltage to a number of different sources which can be switched on and off. If the voltage to the motor is kept constant and the source pattern changes, then the speed of motor shaft will change. If we want to prevent this we can implement this direct analytical method and produce a controller that remains at a constant speed despite the load changes.

10 References

[1] J. Katupitiya, "Laboratory experiment ii : Design of a speed controller," 2021.

[2] ——, "Laboratory experiment ii : Implementation of a speed controller," 2021.

[3] (2019). "Mtrn3020 - introduction to the speed control experiment," UNSW, [Online]. Available: `https://www.youtube.com/watch?v=j9pv0LfVVXg`.

[4] Mathworks, "Matlab," 2021. [Online]. Available: `https://au.mathworks.com/`.