LAB EXPERIMENT II

DESIGN OF A SPEED CONTROLLER

MTRN3020

Modelling and Control of Mechatronic Systems

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

Cameron Murray

Z3417671

5th October 2017

1. INTRODUCTION

This experiment refers to the design and operation of a speed governing controller of a motor existing under a variable load. The task was to be approached by following a systematic design procedure, namely the Ragazzini method, and employing the multidomain dynamical system analysis software Simulink in formulating a consistent block diagram. Implementing the design output with a real life apparatus allowed a comparison of said experimental controller design data against predicted controller performance, to give students feedback on how successfully their controllers performed.

Students were obligated to derive their personalised speed controller from given time constant/sample times before attending a laboratory session. The controller was inserted into software to calibrate and direct a generator rig/electric motor system by way of exploiting a 15 resistor bank to adjust the motor loading. After performing the experiment twice (once with disturbance rejection, once without), two data sets were logged and distributed for simulation in Simulink to help analyse any theoretical consistencies by applying specific/unique loading patterns.

2. AIM

The aim of this experiment is to verify whether an analytically formulated speed controller functions correctly in handling and rejecting disturbances resulting from a variable motor/generator load on a real life rig implementation, and to compare these results to a theoretically based performance.

3. THE EXPERIMENTAL PROCEDURE

Following the derivation of a unique speed controller transfer function from given time constant/sample times, the coefficients acquired from forming the corresponding final difference equation can be input into the control computers UI to calibrate and drive the motor/generator system.

The procedure can be abstracted into a series of two sections. The first of which (A) is performed to ensure the validity of the design procedure, namely under inspection if a 1^{st} order response (FOR) produces a time constant aligning with the given unique time constant design-parameter and a zero steady state error (SSE) is produced by the actual response, then the controller design can be considered valid. After confirming validity and to produce a plot, the initial run was performed with the system running at 1000rpm, and modulated to 2000rpm by inducing a step of 1000rpm. As noted, a result of using a correct controller, the step shift from 1-2krpm will produce a FOR that is consistent with design parameters. The zero SSE will be detectable at both rotational velocities (1-2krpm). The data will then be plotted and the accuracy of the design can be verified by inspection

The second section (B) refers to disturbance handling in response to loading. The quantity of active resistors connected to the generator affects the loading placed on the motor. When performing this section, eight run’s with 200 successive values are recorded with runs 1 and 8 having zero active resistor loadings, whilst the remaining six runs run take varying active resistor values determined from student numbers. As mentioned, the controllers design success can be validated by inspecting how it responds to variations in velocity; if the controller bounces back to a chosen definite speed, the controller can be deemed correct.

4. CONTROLLER DESIGN CALCULATIONS

The unique design tau time is 49ms. The sample time is 9ms.

Start by attaining a FOA to the zero load/open loop results:
By utilizing the `lsqcurvefit` function MATLAB offers, we can fit a curve to the zero load data to find the values of τ and A.:

$$y(t) = A\left(1 - e^{-\frac{t}{\tau}}\right),$$

```
x = lsqcurvefit(@myfun,[750000,0.04],time,speed)

A = 7.5809e+005;

tau = 0.0372;
```

By substituting in these values and introducing an integrator, the voltage/counts transfer function is determined as

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

$$A = \frac{7.5809 \times 10^5}{24} = 31587$$

$$Gp(s) = \frac{24A}{4.669s^2 + 126\tau s} \quad OR \quad = \frac{758090}{24} . \frac{24}{126} . \frac{1}{s(1 + 0.03702s)}$$

$$Gp(s) = \frac{6016.5873}{s(1 + 0.0367s)}$$

By creating appropriate variables and utilizing the following MATLAB functions finds a discrete version of the continuous TF, which allows for *Gp(z)*. to be formed:

```
num = 6016.5873

den = [tau 1 0];

    = [0.0372 1 0];

sampleT = 0.009;

[numd, dend] = c2dm(num,den,sampleT,'zoh');
```

The roots of numd produce the numerator of A(z) likewise the roots of dend help form the denominator:

$$A(z) = \frac{6.0769(z + 0.92218366)}{(z - 1)(z - 0.7842)}$$

Gp(z) can then be established by immediate substitution as:

$$Gp(z) = A(z)\frac{(z - 1)}{zT}$$

$$Gp(z) = \frac{6.0769(z + 0.92218366)}{0.005(z)(z - 0.7842)}$$

Utilizing the personalized time constant value provided on moodle and requiring a zero SSE, which is unity DC gain, then $F(z) = 1$. As the zero has the potential to result in ringing, the numerator needs to take in its value. Since $s = -1/\tau_d$, the placement is then e^{-T/τ_d}. To form F(z) we use:

$$F(z) = \frac{(1 - e^{-\frac{T}{\tau_d}})}{(1 - z_1)} \frac{(z - z_1)}{(z - e^{-\frac{T}{\tau_d}})}$$

Hence:

$$F(z) = \frac{bo\,(z - z1)}{z(z - e^{T/Td})}$$

And so due to the ringing caused by the zero in the numerator of Gp(z):

$$F(z) = \frac{bo\,(z + 0.92218366)}{z(z - 0.8222982)}$$

If F(1) = 1*:*

$$F(1) = 1 = \frac{bo\,(1.92218366)}{(0.1777018)} \quad , \quad bo = \frac{1(0.1777018)}{(1.92218366)} = 0.0924478777$$

$$F(z) = \frac{0.092448\,(z + 0.92218366)}{z(z - 0.8222982)}$$

To calculate Gc(z):

$$Gc(z) = \frac{1}{Gp(z)} \frac{F(z)}{(1 - F(z))}$$

$$Gc(z) = \frac{0.005\,\text{[redacted]}}{6.0769\,\text{[redacted]}} \frac{(z - 0.7842)}{\text{[redacted]}} \text{[redacted]} \frac{(0.07397)(z)\,\text{[redacted]}}{z(z - 0.8222982) - 0.092448(z + 0.92218)}$$

$$\boxed{Gc(z) = \frac{(0.0001369)(z^2 - 0.7842z)}{z^2 - 0.9147461z - 0.0852536}}$$

Where the roots of the numerator are 0 and 0.7842.

The roots of the denominator are 1.000 and -0.0852536

1. SIMULINK BLOCK DIAGRAM

With the intention of generate a model to relate to the real life data, Simulink can be utilized to produce a block diagram of the motor/generator system. Taking the values from the the appendix in the experiment guidelines, the proceeding BD as seen in Figure 1 was prepared:

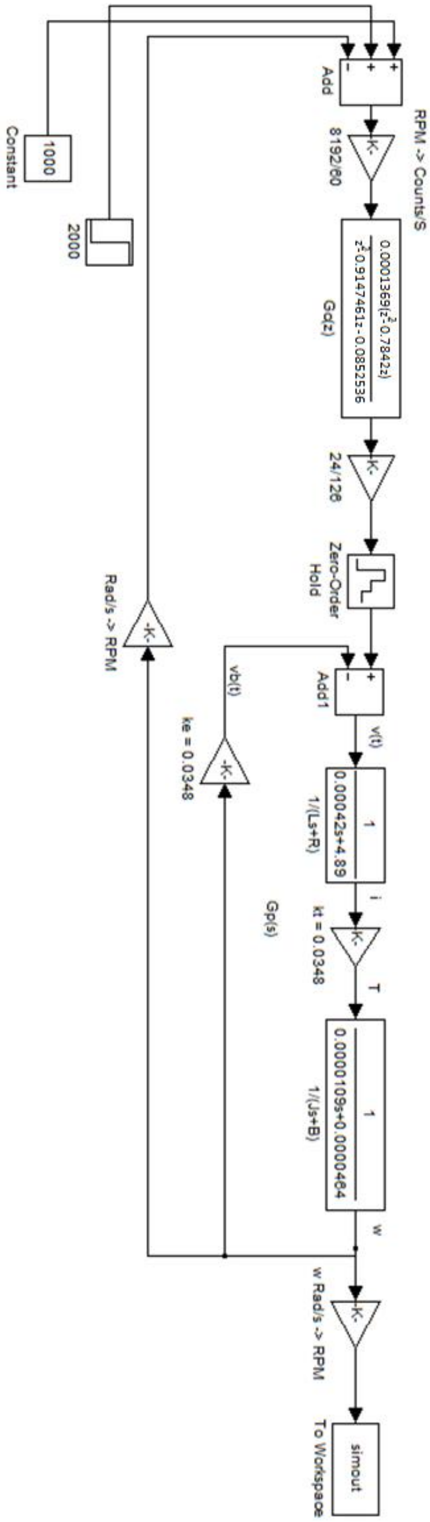

Figure 1: Simulink BD of the system/experiment.

7. PART A – 1000RPM TO 2000RPM

After using Simulink to create the corresponding block diagram, the step function (SF) used to alternate the speed of the motor can be input. No different to the 2nd run, the 1000rpm step function used to take the max rpm to 2000 will be couple with an initial step to 1000rpm. The SF in Simulink can be seen below in Fig2.

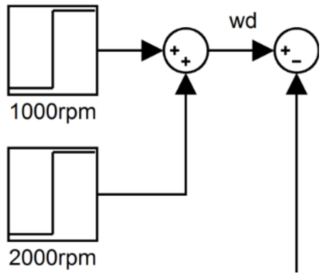

Figure 2: Simulink step function.

Since we are working with a couple of SF blocks they will act independently with the 1st block in Fig2 switching on at t = 0 and the 2nd block will switch on concurrently with when the step occurs in the experimental data record (it will step up to 2000rpm as it only appends an extra 1000rpm). After the SF block values are switched on and the values combined, the simulation can be run to produce a value array. To interpolate the data so the array lengths are compatible for plotting (equal length to time vector), we use `interp1`. After finalizing the array the comparison can be made by simultaneously plotting the two data sets as shown below.

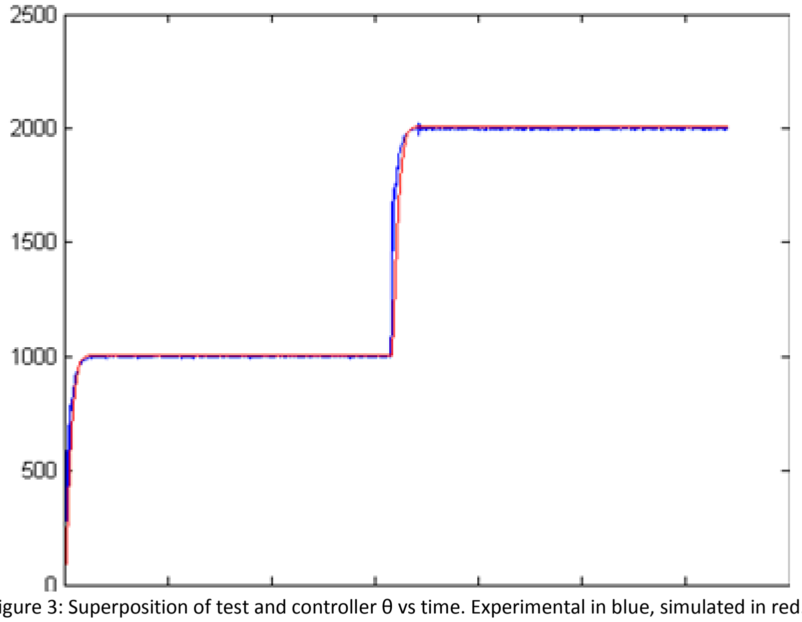

Figure 3: Superposition of test and controller θ vs time. Experimental in blue, simulated in red.

Figure 3. Validates the design as correct, as both the simulated design's results and tested results are closely related following the same trends, i.e. some stationary error is exhibited, both experience a fast rise time with minimal noticeable overshoot.

8. PART B – LOAD CHANGES

To incorporate the motor loading variations introduced earlier (sec 3.), the block diagram will need to be augmented with a load adder. Utilising the `dec2hex(on student number 3417671)` function in MATLAB yields hex number 342647, this hex value will be used to introduce the value of resistors (each digit signifying the amount of active resistors) per each 200 data size. As noted earlier, but varying the number of active resisters, the motor load can be manipulated to simulate the variable load in the demonstration.

The LoadSim block allows us to connect a 'Load Torque' to the motor's speed.

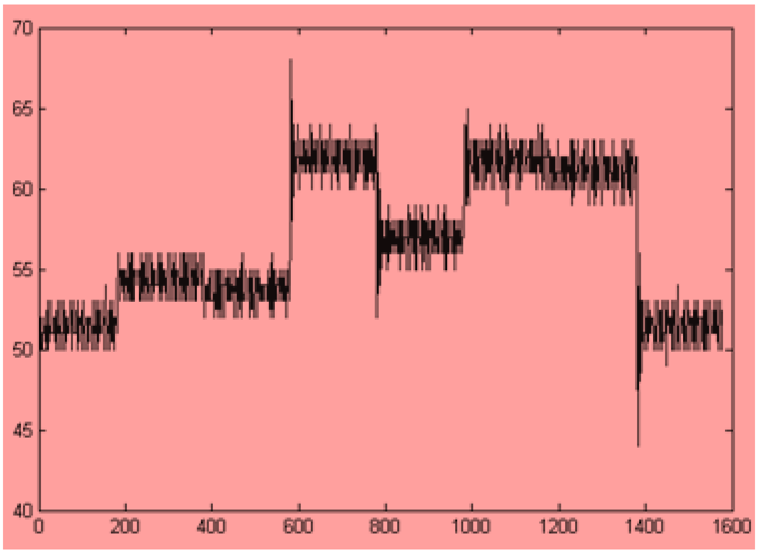

Figure 4: Resister loads in the pattern: 0-3-4-2-6-4-7

The "loadsim" block allows for the addition of a "Load Torque" (LT) to simulate the induced resistor loading. The actual manipulation of the load can be done by creating a summed series of step functions, where inside this resistor bank the appropriate active resistors can be modulated by sequential step functions (Figure5). The block itself will work behind the scenes to produce this "Load Torque" value by controlling the resistor activations, these being guided by the user inputted parameters omega and active resistors. The LT is utilized to establish the motor torque differencing for acceleration inertia

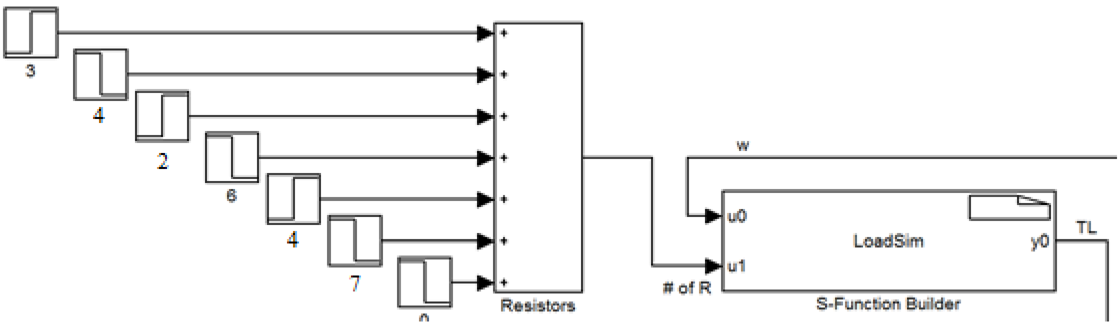

Figure 5: The summated resistor step files being fed into the LoadSim block.

The diagram below (Fig.6) shows the result of appending the resistor bank to the LoadSim block in Fig.5 superimposed on the test results to allow a comparative study between predicted and experimental motorcontrol data.

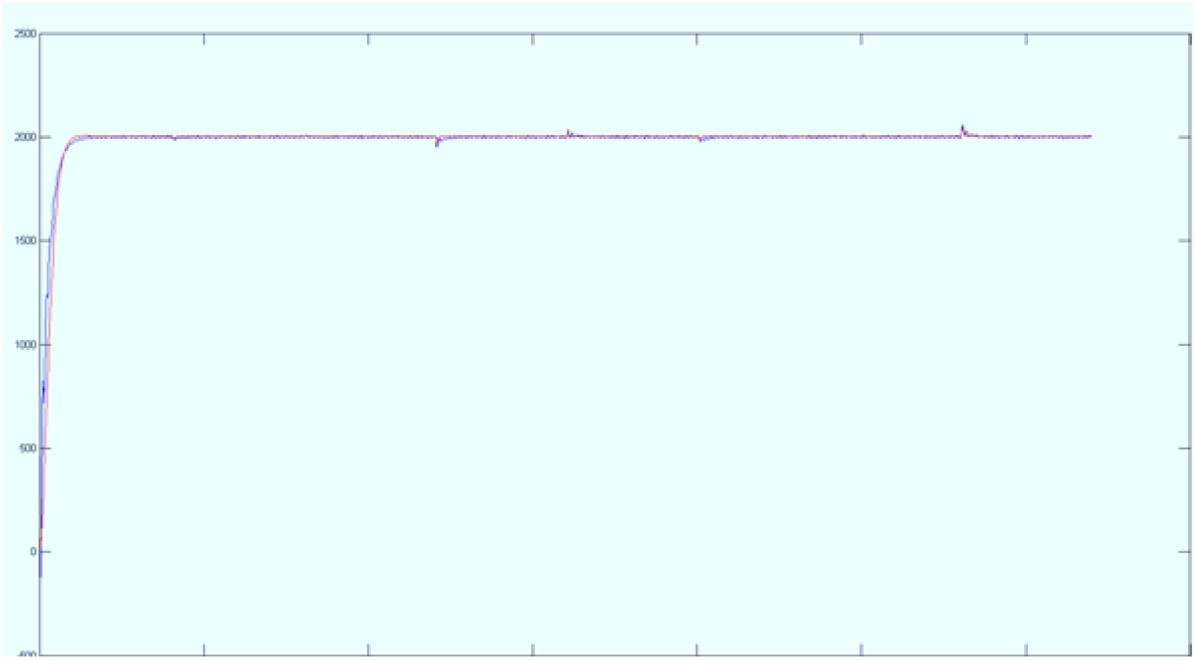

Figure 6: Experimental (blue) and simulated (red) load affected data superimposed.

Zooming in on two load-step regions in Fig 7.

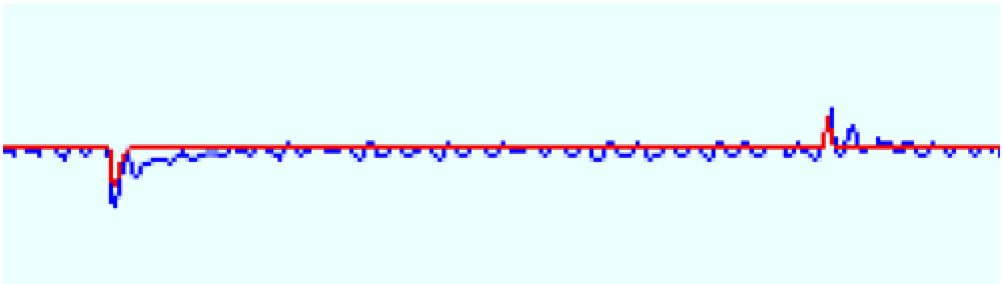

Figure 7: Close inspection of resistor shift from 2 - 13 and back down to 6.

It can be extrapolated that the model is well-suited with the test results, as the two superimposed plots react to the changes in resistor activation consistently.

9. CONCLUSION

Observing diagrams 7, 6 and 3, exhibits a relatively fast rise time whilst achieving the SS (steady state) of 1k/2krpm. The system has minimal to no overshoot and reveals symptoms of overdamping, both sought-after running-motor qualities. Even when taking into account a variable loading, the system design can be seen to correctly and quickly traverse to chosen velocity such that it can be attained as quickly as a 10th of a second. The relatively acceptable settling time of the system also concludes the success of the design controller.

There is apparent but not significant (Deviates a maximum of ±0.6% from rot. vel.) ringing affecting the system which appears controlled as its magnitude does not tend to escalate. The origin of the ringing can be derived as a result of the velocity sensor's precision. The ringing can be observed in Figure 8.

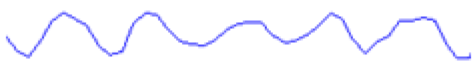

Figure 8: Ringing in Test data

There are several possible real life discrepancies that could cause differences between experimental and theoretical data values, most of which will affect the real plant parameter values. Back EMF coefficients may have differed marginally throughout the experiment. This inconsistency can definitely alter the plant TF (albeit not relatively significant. Other discrepancy origins include armature resistance and inductance, viscous damping, motor inertia, differing torque constant.

After analysis, and as expected from a correct controller design, the sim. design can be seen to describe the theoretical outcome of the system, thus outclassing the rig-run's data. This is gathered from the design having a less obvious SS line deviation characteristic and apparent fasters settling times (seen in Fig.7) and is likely due to internal frictions of the rig and inaccuracies induced from the resistor bank. Looking at the initial rise section for the controller, there is some differences. Fig.6 shows the controller designs plot has a steeper gradient than the testing data's plot. This can also be reduced to the occurrence of friction in the system due to large rot. speeds in the system.