University of Waterloo

Faculty of Engineering

Department of Electrical and Computer Engineering

ECE 481: Lab 2

By Nathaniel Johnston, 20710059

July 20th, 2021

# Table of Contents

## Lab 2: Design of the Inner-Loop Controller and modelling of the Ball & Beam

### Introduction

In this lab, I use the emulation technique to design a continuous time controller and then discretize this controller to work in a closed loop sample-data system. This controller will be used to control the angle of a motor. This is one part of a system which will ultimately control the position of a ball on a beam which is connected to the motor.

A linearized model of the ball and beam system is also derived in this lab using basic physics and first principles.

### Inner-Loop Controller Design

The inner loop controller design involves using the emulation approach to first design a continuous time controller and then discretizing to get a discrete time controller. This controller will be used to control the angle of a motor.

#### Continuous Time Controller Design

My initial thought process for the controller was to simply use a proportional controller. With a proportional controller, I could ensure that the system would always be stable and I would be able to mee the overshoot and settling time specifications. The steady state tracking specification would naturally be satisfied since there is an integrator in the plant. However, after some experimentation with proportional controllers, it quickly became apparent that a proportional controller would not be able to satisfy all these specifications while also staying between +/- 6V output.

I knew that the controller below would be able to give me the specifications I needed for overshoot and settling time, but I was unsure of if it would be able to meet the +/- 6V output requirement.

$$C_1(s) = \frac{k(s+a)}{\tau s + 1}$$

After playing around with values using the control system designer, I found a few controllers which worked using the form above, however when I went to discretize the controller, I did not get similar performance at a reasonable sampling rate. This was likely partially due to the fact that I did not add in the T/2 time delay in the plant model. The reason this was not added in was because I had not yet learned about this trick when designing the controller.

The final controller I designed for this system is shown below as $C_1(s)$

$$C_1(s) = \frac{2s + 160}{s + 32}$$

This is a lead compensator and was designed using Matlab's control system designer tool. I modified the locations of the pole and zero until I got something reasonable and they would simulate this continuous time controller and its discrete time counterpart in the overall system. This controller meets all specifications and results in a discrete time controller which also meets the specifications at a reasonable sampling frequency.

Figure 1 shows the step response of the continuous time system with the applied step starting from -0.7 and rising to 0.7 at time t = 1 second. Figure 2 shows the output of the controller (input voltage to the motor) when the same step is applied. As can be seen from the figures, they meet all the required specifications.

Note that my settling time requirement was 0.37 seconds.

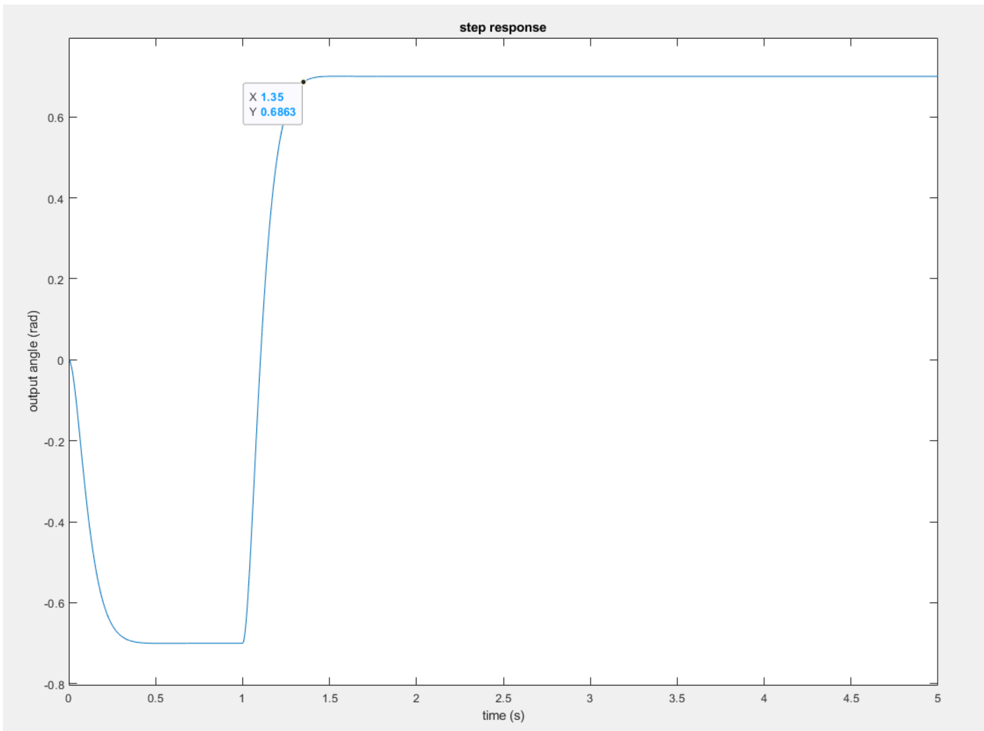


*Figure 1. Continuous time controller step response*

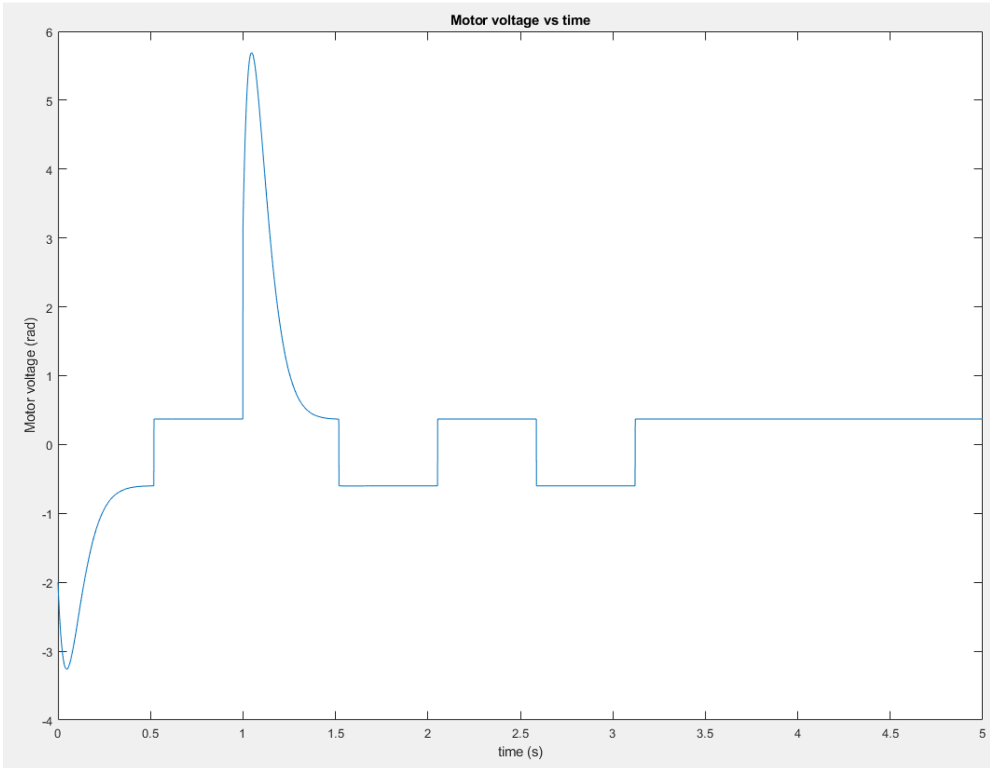


*Figure 2. Continuous time controller motor voltage*

## Discrete Time Controller Design

The continuous time controller, $C_1(s)$, resulted in the following discrete time controller.

$$D_1[z] = \frac{2z - 0.6307}{z - 0.7261}$$

This controller was found by discretizing using the SCH rule with a sampling frequency of 100 Hz (sampling period of 0.01 seconds). This sampling rate should be more than adequate since the bandwidth of the overall system  is with the continuous time controller is roughly 1.3 Hz. The bandwidth was estimated by using the gain crossover frequency of C(s)P(s). Using the rule of thumb that the sampling frequency should be at least 10 times the system bandwidth, the minimum sampling frequency required would be 13 Hz. The reason for the faster sampling rate is because the discrete time controller acts more similar to the continuous time controller at a faster sampling rate, and a 100 Hz sample and hold is perfectly feasible given current technology.

This controller performs similarly to the continuous time controller as shown in figures 3 and 4. In addition, it can be seen that all design requirements have been met with this controller.

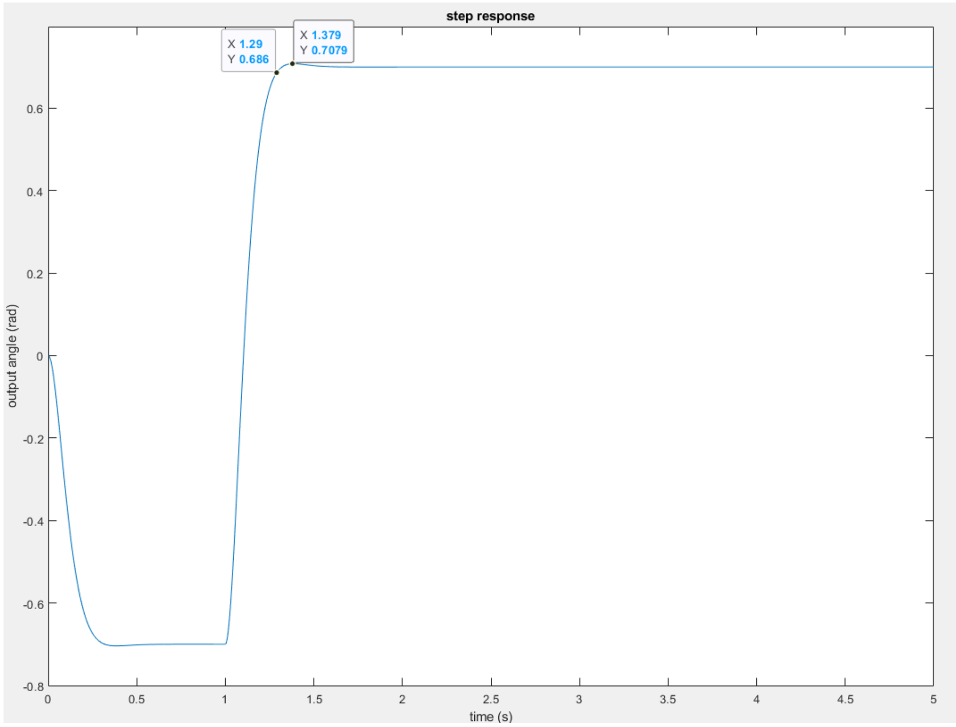


*Figure 3. Discrete time controller step response*

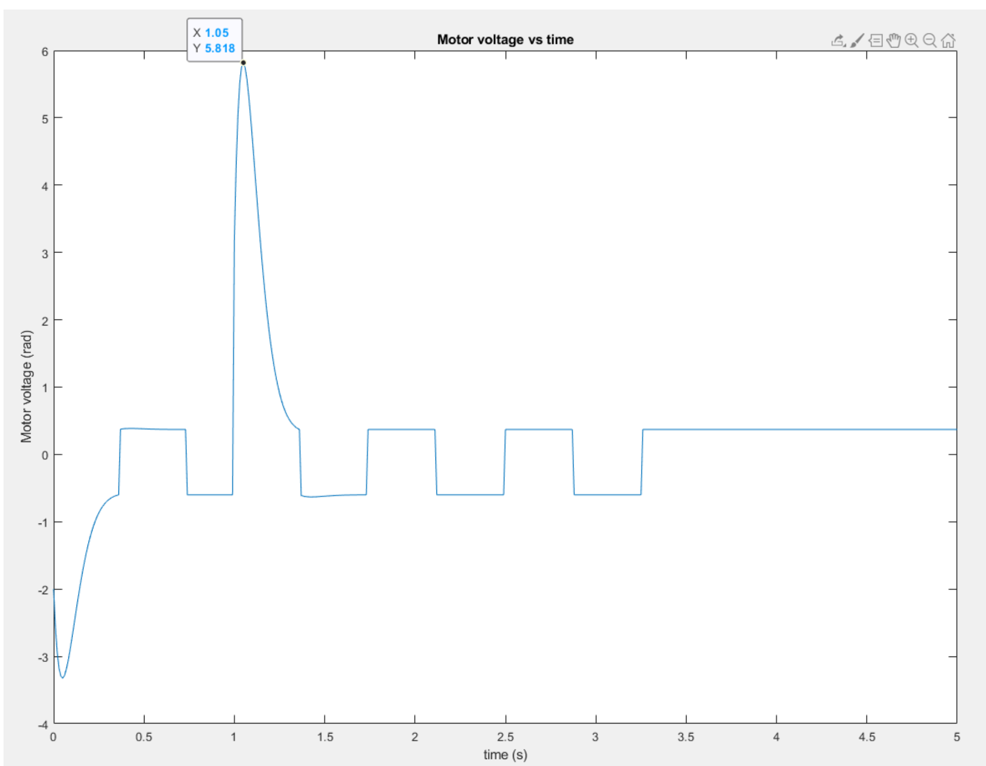


*Figure 4. Discrete time controller motor voltage*

### Time Domain Equation for $D_1[z]$

Converting this discrete time controller to the time domain, we get the following difference equation where e[k] is the input to the controller and v[k] is the output.

$$\frac{V[z]}{E[z]} = \frac{2z - 0.6307}{z - 0.7261}$$

$$V[z](z - 0.7261) = E[z](2z - 0.6307)$$

$$v[k+1] - 0.7261v[k] = 2e[k+1] - 0.6307e[k]$$

$$v[k] = 0.7261v[k-1] + 2e[k] - 0.6307e[k-1]$$

## Modelling of the Ball and Beam System

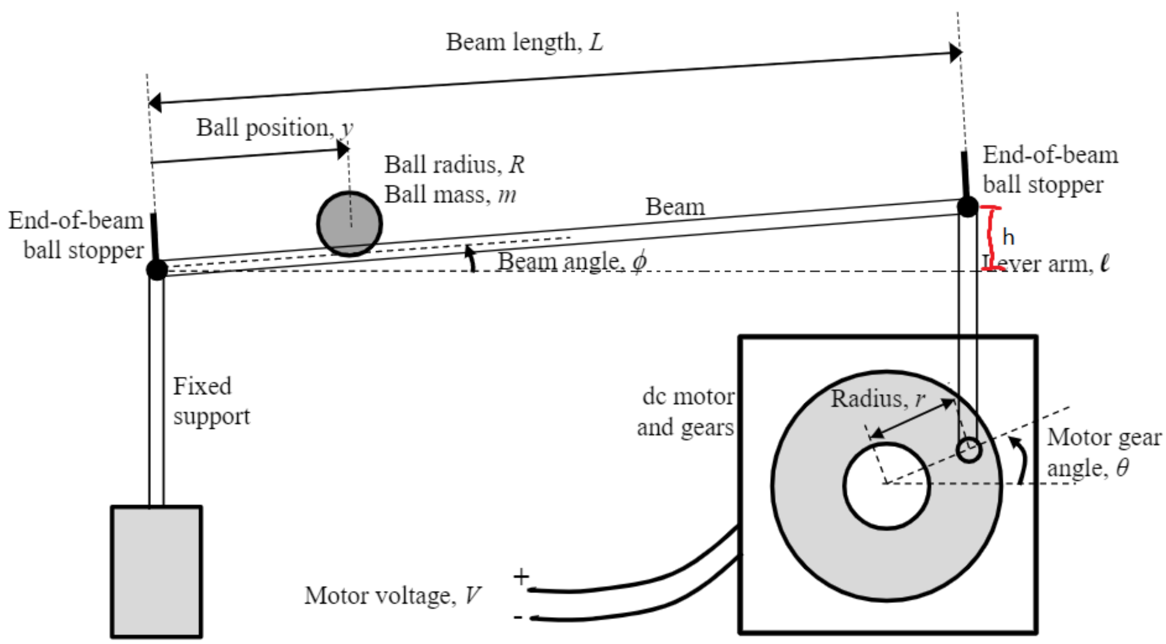


*Figure 5. ball-beam setup*

The height difference between the left and right ends of the beam shown in figure 5 is denoted as h and can be calculated as shown, assuming that the lever arm stays perfectly upright.

$$h = h_0 + r * \sin(\theta)$$

where $h_0$ is the value of h when θ is 0, this is assumed to be 0.

$$h = L * \sin(\phi)$$

Thus,

$$L * \sin(\phi) = r * \sin(\theta)$$

Linearizing about $\phi_0 = 0$ (which also implies $\theta_0 = 0$, under the assumption that $h_0 = 0$), we get:

$$a_1 \Delta\phi = a_2 \Delta\theta$$

$$a_1 = \frac{d}{d\phi}(L * \sin(\phi))|_{\phi_0} = L * \cos(\phi_0) = L$$

$$a_2 = \frac{d}{d\theta}(r * \sin(\theta))|_{\theta_0} = r * \cos(\theta_0) = r$$

Taking the Laplace transform of this linearized equation gives:

$$L\Phi(s) = r\Theta(s)$$

$$\frac{\Phi(s)}{\Theta(s)} = \frac{r}{L} = 0.05644$$

The value shown above is the value of $k_2$.

To find $k_3$, we start with the moment of inertia for solid sphere:

$$I = \frac{2}{5} mR^2$$

Torque equation:

$$\tau = I\alpha = F_{fr} d$$

Where $\alpha$ is the angular acceleration and $F_{fr}$ is the static friction force.

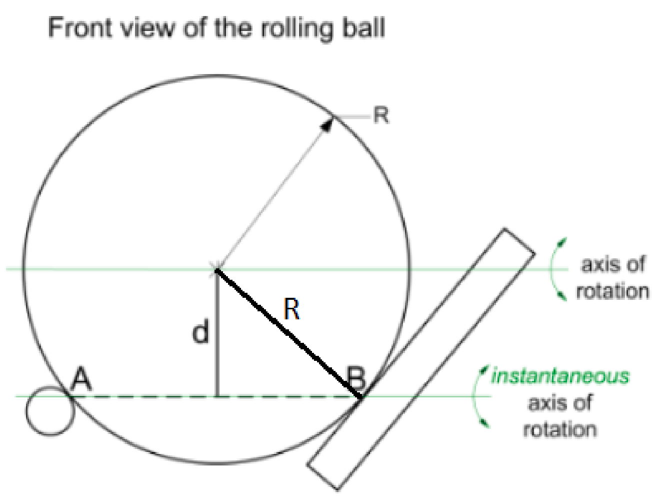


*Figure 6. Ball dimensions*

Here, d can be calculated by observing there is a right angle triangle with side lengths d, R, and AB/2. Using the Pythagorean theorem yields

$$d = \sqrt{R^2 - \left(\frac{AB}{2}\right)^2}$$

Static friction:

$$F_{fr} = mg * \cos(\phi) * \mu$$

This yields,

$$F_{fr}d = \frac{2}{5}mR^2\left(\frac{a}{d}\right)$$

$$mg * \cos(\phi) * \mu\sqrt{R^2 - \left(\frac{AB}{2}\right)^2} = \frac{2}{5}\frac{mR^2a}{\sqrt{R^2 - \left(\frac{AB}{2}\right)^2}}$$

$$g * \cos(\phi) * \mu * \left(R^2 - \left(\frac{AB}{2}\right)^2\right) = \frac{2}{5}R^2a$$

$$\mu = \frac{8R^2a}{5g(4R^2 - (AB)^2) * \cos(\phi)}$$

Applying Newton's 2nd law:

$$mg * \sin(\phi) - mg * \cos(\phi) * \mu = ma$$

$$g * \sin(\phi) - g * \cos(\phi) * \frac{8R^2a}{5g(4R^2 - (AB)^2) * \cos(\phi)} = a$$

$$g * \sin(\phi) - \frac{8R^2a}{5(4R^2 - (AB)^2)} = a$$

$$g * \sin(\phi) = a\left(1 + \frac{8R^2}{5(4R^2 - (AB)^2)}\right)$$

$$\frac{g}{1 + \frac{8R^2}{5(4R^2 - (AB)^2)}} * \sin(\phi) = a$$

The part multiplying the sin(ϕ) is a constant and will henceforth be referred to as c.

$$\mathrm{c} * \sin(\phi) = a$$

Assuming zero initial conditions, this is the same as

$$\ddot{y} = c * \sin(\phi)$$

Linearizing about $\phi_0 = 0$

$$\Delta\ddot{y} = a_1\Delta\phi$$

$$a_1 = \frac{d}{d\phi}(c * \sin(\phi))\bigg|_{\phi_0} = c * \cos(0) = c$$

Taking the Laplace transform and plugging in values to calculate c

$$s^2 Y(s) = c\Phi(s)$$

$$\frac{Y(s)}{\Phi(s)} = \frac{c}{s^2} = \frac{5.6884}{s^2}$$

Thus, c = $k_3$ = 5.6884.

This yields the system shown in figure 7.

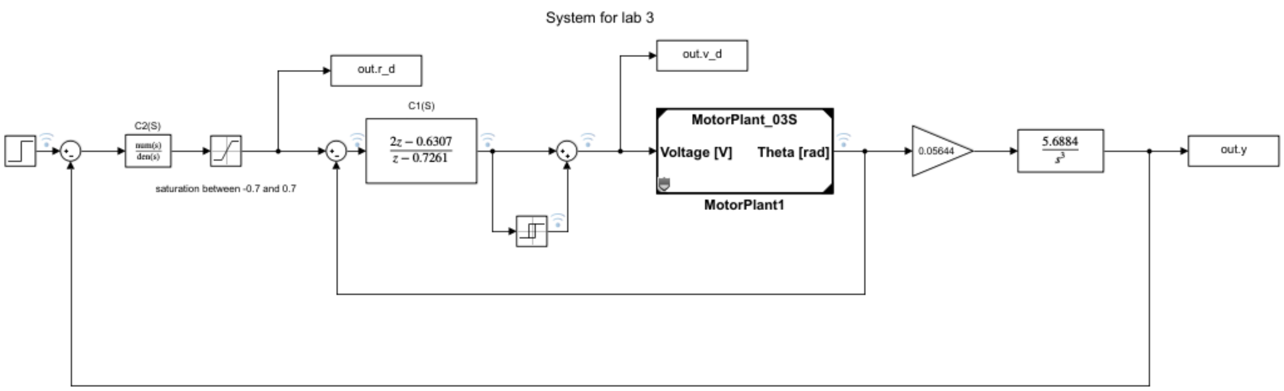


*Figure 7. Updated system control loop*

Note that biases do not need to be added in after the linearized transfer functions. This is because the operating point is when ϕ = θ = 0. So the biases are all 0 and do not make a difference to the system.

Also note that $C_2(S)$ (the block right after the first summer and before the saturator) has been placed in the feedback loop but has not been designed yet. The transfer function block for $C_2(S)$ is simply a placeholder.

## Conclusion

The final continuous time controller that was designed in this lab took the form of a lead compensator. When discretized using the SCH rule and an adequate sampling rate, a discrete controller was found with very similar performance to the continuous time controller.

A linearized model was found to describe the ball and beam system. This allowed me to construct the overall feedback system. The one thing left to do to fully control this system is to design another controller to control the position of the ball on the beam. This will be accomplished in the next lab.

I acknowledge and promise that:

(a) I am the sole author of this lab report and associated simulation files/code.

(b) This work represents my original work.

(c) I have not shared detailed analysis or detailed design results, computer code, or Simulink diagrams with any other student.

(d) I have not obtained or looked at lab reports from any other current or former student of ECE 484/481, and I have not let any other student access any part of our lab work.

(e) I have completely and unambiguously acknowledged and referenced all persons and aids used to help us with our work.

Signed N Johnston