1. Move the load shaft by hand. Does it feel like a rotary mass, spring, or damper? Is this what you expected based on the model we developed? If not, what seems to be missing in the model?

When moving the load shaft by hand, it feels like a rotary mass with some damping. After releasing it, it spins for approximately one half of a second before stopping.

Revisiting the system model from Lecture 5 in Eqn. 1, the system can be divided into three parts: the virtual spring K relative to θ_R, the virtual damper B relative to θ_R, and the virtual damper B_0 relative to 0. With the system off, there should be no resistance to the movement of the output shaft. Even though the movement of the motor rotor would cause a back EMF voltage and therefore a resistive torque when the system is powered on (represented by k_B in B_0), this is not the case in the unpowered system because the motor circuit is open. However, there is still some damping occurring that causes the output shaft to stop rotating. This is most likely due to unmodeled friction in the system, possibly between each gear, between the shaft and the bearings, or between the shaft and the resistor.

$$
\begin{aligned}
\tau_{ext} &= J_{eq}\ddot{\theta}_L + K + B + B_0 \\
\text{virtual spring } K &= \frac{Nk_\tau G_P(\theta_R - \theta_L)}{R_m} \\
\text{virtual damper } B &= \frac{Nk_\tau G_D(\dot{\theta}_R - \dot{\theta}_L)}{R_m} \\
\text{virtual damper } B_0 &= \frac{Nk_\tau k_B \dot{\theta}_L}{R_m}
\end{aligned} \tag{1}
$$

2. Turn on the electronics module and move the arm by hand again. Does it feel any different? What is causing the difference?

Turning on the electronics module in the system does change the feel and behavior of the output shaft. Specifically, the output shaft feels harder to move and feels more damped. This can be explained by revisiting Eqn. 1 and the motor circuit diagram from Lecture 2 in Fig. 1.

Now that the circuit is powered, when the motor is turned the back EMF voltage increases proportional to k_B in the virtual damper B_0. As the back EMF voltage increases the voltage drop across the motor must remain at 0 (because there is no input), so resistance to movement of the output shaft also increases due to torque generated by the back EMF voltage.

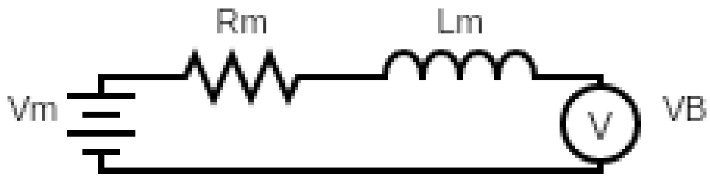

Figure 1: Motor Circuit Diagram.

3. Open Labview and set $G_P = -1.0$ and $G_D = 0.0$. Increase the proportional feedback gain. What does this feel like when you manipulate the output shaft by hand: a change in mass, damping, or stiffness?

After setting the proportional gain to -1, the output shaft feels much stiffer than in part 2. It is harder to push, and the farther it is pushed the faster it moves to return to the reference angle. Revisiting Eqn. 1, we see that G_P is in the spring term K, corresponding to an increase in spring stiffness as the magnitude of G_P is increased.

4. Increase the derivative gain. What does this feel like?

While the increased stiffness is due to the increase in the magnitude of the proportional gain, an increase of the derivative gain increases the damping of the output shaft. As the output shaft is pushed faster, the resistance increases proportionally to the rotational speed of the shaft. Since G_D is present only in the virtual damper term B, this observation is in correspondence with the system model in Eqn. 1.

5. What happens if you use a positive proportional gain? Explain what is happening.

Setting the derivative gain equal to 0 and the proportional gain positive, the output shaft becomes easier to move and the system should become unstable. Fig. 2 shows the poles for the transfer function derived in Experiments 1 and 2 when G_P is positive. One pole has a positive real component, and as such the theoretical system is unstable.

On the hardware, for small values of G_P and θ_L the output will move away from the reference angle and then stop. This is most likely due to the unmodeled friction discussed in part 1 of this Experiment. However, once the proportional gain is increased enough the physical system does become unstable and will continue to move away from the reference angle until it hits the software dead zone, and power to the motor is cut.

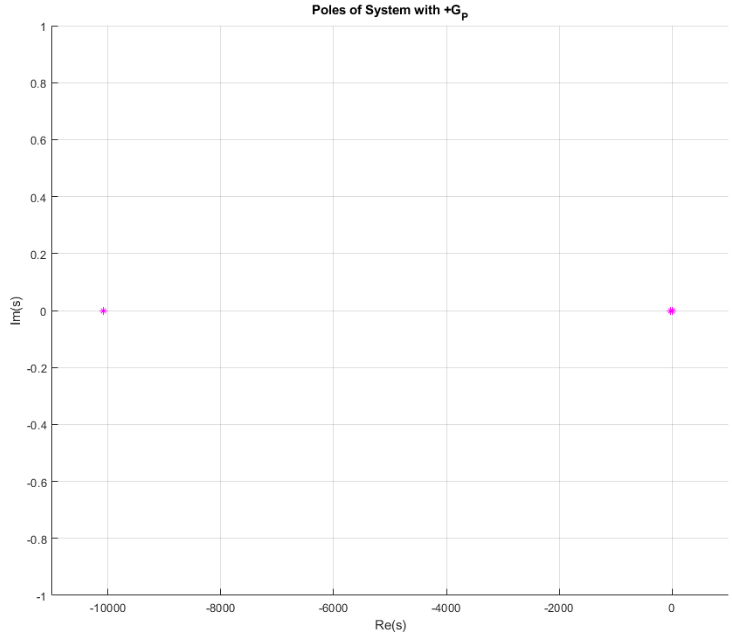

Figure 2: Poles of System when G_P is Positive.

6. What happens if you use a positive derivative gain? Explain this as well.

Like with setting the proportional gain to a positive value, setting the derivative gain to a positive value makes the system BIBO unstable. Fig. 3 shows the poles for the transfer function when G_D is positive. While none of the poles have a positive real component, one of the poles sits at 0, making the system not BIBO stable.

When moving the output shaft, the shaft seems much lighter, like the moment of inertia of the system has actually decreased. The faster the output shaft is moved the more pronounced this effect is, with the shaft seeming to accelerate away from the reference angle. Referring back to Eqn. 1, a positive derivative gain will produce an effect opposite to that of a damper, which matches our observations.

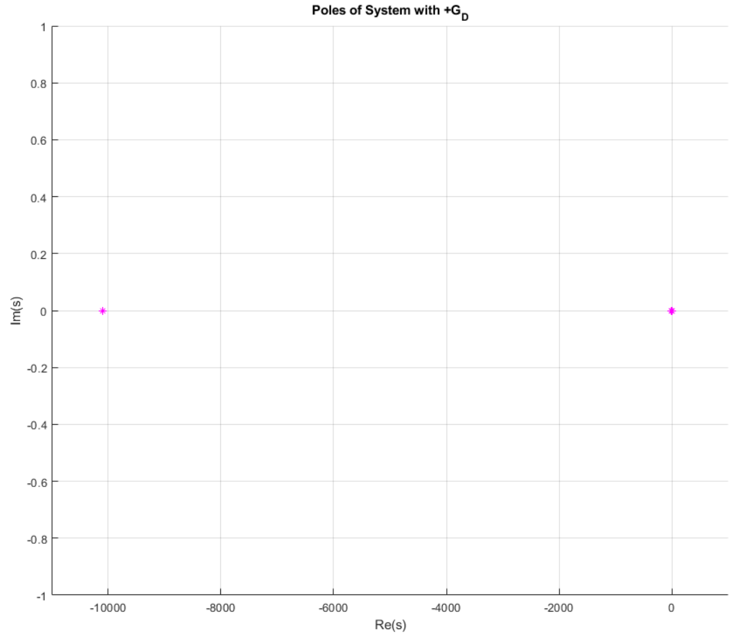

Figure 3: Poles of System when G_D is Positive.

7. Repeat part 5 of Experiment 1 on the hardware. Record this step response and compare with the simulation from Experiment 1.

The hardware was slightly misconfigured for this part. Instead of using a step input that held it's position for long enough for the system to stabilize, the input was a 2 Hz square wave with an amplitude of 0.4 rad. However, there is still some comparison that can be done. Part 5 of Experiment 1 is modified to use a step input with amplitude 0.8 rad instead, and the results of the simulation and the experiment are plotted in Fig. 4. It is apparent that the hardware does not perfectly follow the simulation, which could be due to the extra friction discussed in Problem 1. It could also be due to moments of inertia in the system that are unmodeled.

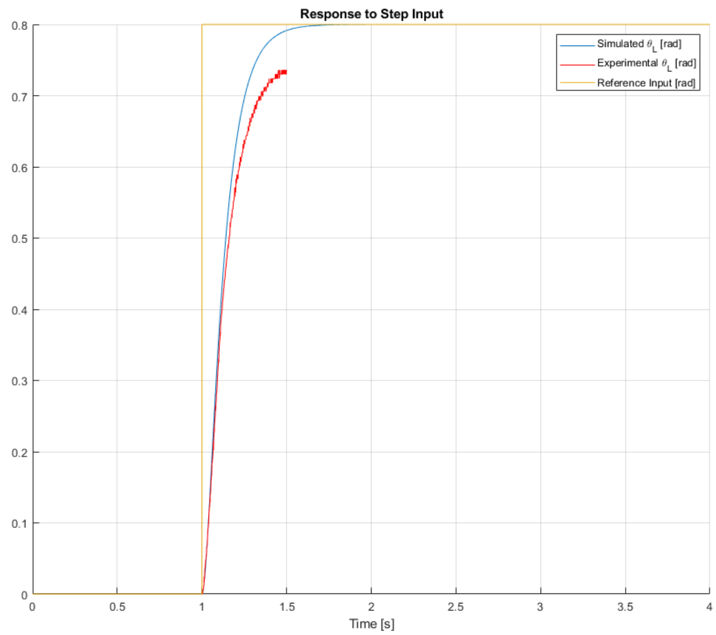

Figure 4: Step Input Response of the System.

8. Repeat part 6 of Experiment 1 on the hardware. Compare these sinusoidal responses with the simulations from Experiment 1.

For this problem, the data recording was not started until after the output shaft had already started moving, so the initial resistance to movement of the shaft is not captured in the plot. The response to the 0.2 Hz signal is shown in Fig. 5, and to the 2 Hz signal is shown in Fig. 6. In the 0.2 Hz example, the experimental data does not reach the full amplitude of the signal, and is less than both the input signal and the simulation. In the 2 Hz example, it also performs worse than the simulation, only reaching just over 0.2 rad.

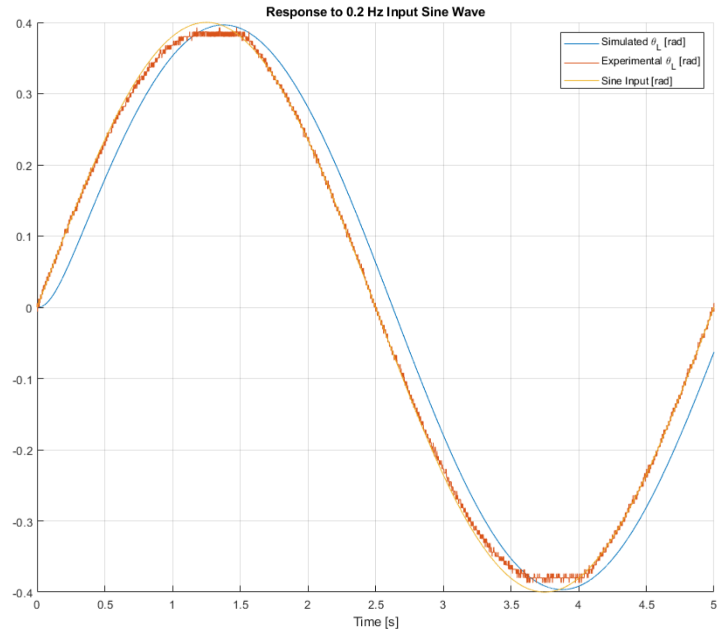

Figure 5: Response of the System to a 0.2 Hz Sine Wave.

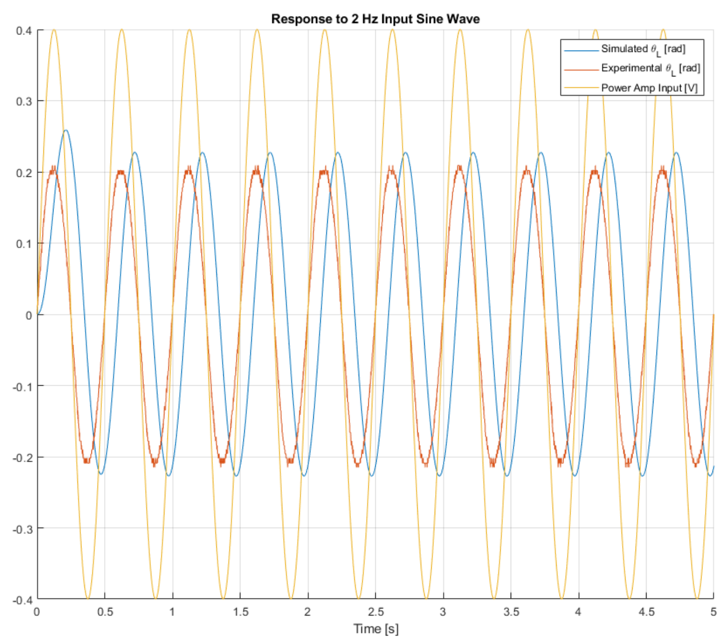

Figure 6: Response of the System to a 2 Hz Sine Wave.

MATLAB Code

```
%% Experiment 3
% William Watkins
% 3 February 2022

%% Clean up
clear all
close all
clc

%% Constants/Properties
Jconv = 1/0.00034171718982094; %[g*cm^2 / lb*in^2]
Jc = 1*10^-7; % [kg*m^2 / g*cm^2]
Rm = 19.2;      % [Ohms]
Lm = 0.0019;    % [Henrys]
Ktau = 40.1*10^-3;  % [Nm/A]
Kb = 1/238 / (2*pi/60); % [V/(rad/s)]
rBig = 2.51; % [in]
rSmall = 0.79; % [in]
Ngearhead = 10;
Ngears = rBig/rSmall;
Jm = 12.5 * Jc; % [kg*m^2]
Jgearhead = 0.6 * Jc; % [kg*m^2]
N = Ngearhead * Ngears;

%% Computing Moments
Jtm = Jm; % [kg*m^2]
rhoAl = 0.097; % [lb/in^2]
h = 0.199; % [in]
wBeam = 0.504; % [in]
dBeam = 10; % [in]
sSquare = 1.75; % [in]
Vsq = sSquare^2 * h; % [in^3]
Vbeam = h * wBeam * dBeam; % [in^3]
mSquare = Vsq * rhoAl; % [lb]
mBeam = Vbeam * rhoAl; % [lb]
Jbeam = (1/12) * mBeam * (dBeam^2 + wBeam^2); % [lb*in^2]
Jsquare = (1/12) * mSquare * 2 * sSquare^2; % [lb*in^2]
z = dBeam/2 + sSquare/2; % [in]
JPA = Jbeam + mBeam*z^2; % [lb*in^2]
Jl = JPA + Jsquare; % [lb*in^2]
Jl = Jl * Jconv * Jc; % [kg*m^2]
Jeq = Jl + N^2 * Jtm; % [kg*m^2]

%% System Coefficients
s3 = (Jeq/(N*Ktau))*Lm;
s2 = (Jeq/(N*Ktau))*Rm;
s1 = N*Kb;
Gd_o = -0.1;
Gp_o = -10;
```

```
%% Problem 5

Gp = -0.1 * Gp_o;
Gd = 0;

p = [Jeq*Lm/(N*Ktau) (Jeq*Rm/(N*Ktau)) N*Kb-Gd -Gp];
r = roots(p);

figure('Position', [200 200 1000 800]);
hold on;
grid on;
plot(r, [0 0 0], '*','color','magenta');
xlabel('Re(s)');
ylabel('Im(s)');
title('Poles of System with +G_P')
xlim([-11000 1000])

%% Problem 6

Gp = 0;
Gd = -10*Gd_o;

p = [Jeq*Lm/(N*Ktau) (Jeq*Rm/(N*Ktau)) N*Kb-Gd -Gp];
r = roots(p);

figure('Position', [200 200 1000 800]);
hold on;
grid on;
plot(r, [0 0 0], '*','color','magenta');
xlabel('Re(s)');
ylabel('Im(s)');
title('Poles of System with +G_D')
xlim([-11000 1000])

%% Problem 7

load('Data/StepResponseP5');
TimeP7 = [(1:1:999)' / 1000; ((StepResponseP5(675:675+499,1) - ...
    StepResponseP5(675,1)) / 1000) + 1];
OutputP7 = [zeros(999,1); StepResponseP5(675:675+499,2)-...
    StepResponseP5(675,2)];

% From Experiment 1
Gp = -10; % [V/rad]
Gd = -0.1; % [V/(rad/s)]

simOut5 = sim('PD_Control_Step.slx');
thetaL5(1,:) = simOut5.yout{1}.Values.Data';
tSpan5(1,:) = simOut5.yout{1}.Values.Time';
thetaL5(2,:) = simOut5.yout{2}.Values.Data';
tSpan5(2,:) = simOut5.yout{2}.Values.Time';
figure('Position', [200 200 1000 800]);
hold on;
grid on;
```

```
plot(tSpan5(1,:),thetaL5(1,:))
plot(TimeP7, OutputP7,'r-')
plot(tSpan5(2,:),thetaL5(2,:))
legend({'Simulated \theta_L [rad]','Experimental \theta_L [rad]',...
    'Reference Input [rad]'},'Location','northeast')
title('Response to Step Input')
xlabel('Time [s]')
ylim([0 0.8])

%% Problem 8

% Part A
load('Data/SineResponseP6a');
TimeP6a = [((SineResponseP6a(124:124+5000,1) - SineResponseP6a(124,1)) /...
    1000)];
OutputP6a = [SineResponseP6a(124:124+5000,2)];

% From Exp 1
simOut6a = sim('PD_Control_Sine02.slx');
thetaL6a(1,:) = simOut6a.yout{1}.Values.Data';
tSpan6a(1,:) = simOut6a.yout{1}.Values.Time';
thetaL6a(2,:) = simOut6a.yout{2}.Values.Data';
tSpan6a(2,:) = simOut6a.yout{2}.Values.Time';
figure('Position', [200 200 1000 800]);
hold on;
grid on;
plot(tSpan6a(1,:),thetaL6a(1,:))
plot(TimeP6a, OutputP6a)
plot(tSpan6a(2,:),thetaL6a(2,:))
legend({'Simulated \theta_L [rad]','Experimental \theta_L [rad]',...
    'Sine Input [rad]'},'Location','northeast')
title('Response to 0.2 Hz Input Sine Wave')
xlabel('Time [s]')

% Part B
load('Data/SineResponseP6b');
TimeP6b = [((SineResponseP6b(210:210+5000,1) - SineResponseP6b(210,1)) /...
    1000)];
OutputP6b = [-SineResponseP6b(210:210+5000,2)];

% From Exp 1
simOut6b = sim('PD_Control_Sine2.slx');
thetaL6b(1,:) = simOut6b.yout{1}.Values.Data';
tSpan6b(1,:) = simOut6b.yout{1}.Values.Time';
thetaL6b(2,:) = simOut6b.yout{2}.Values.Data';
tSpan6b(2,:) = simOut6b.yout{2}.Values.Time';
figure('Position', [200 200 1000 800]);
hold on;
grid on;
plot(tSpan6b(1,:),thetaL6b(1,:))
plot(TimeP6b,OutputP6b)
plot(tSpan6b(2,:),thetaL6b(2,:))
legend({'Simulated \theta_L [rad]','Experimental \theta_L [rad]',...
    'Power Amp Input [V]'},'Location','northeast')
```

```
title('Response to 2 Hz Input Sine Wave')
xlabel('Time [s]')
```