University of Colorado - Boulder

ASEN 2003

Dynamics Lab 6

Rotary Position Control

Author: Connor O'Reilly[a]

Author: Jason Le[b]

[a]107054811
[b]108972847

Professors:
Jay McMahon
Penina Axelrad

Lab Instructors:
Robert Hodgkinson
Josh Mellin

April 28, 2020

Contents

Although systems can remain static, some systems need to be moved as the system may be in an undesirable state. To move these systems to a desirable state, control systems can be added to move the system with the desired angle and/or velocity. In this lab, the group analyzed control gains on a rigid and flexible arm with a step function. Through this analysis, the gains for the desired state of the rigid and flexible arm were achieved. Equations that govern the control system are derived from differential equation techniques. The rigid and flexible models and experiments were then compared. The gains for the rigid arm are $K_{P\theta} = 19.41$ and $K_{D\theta} = -0.10$. The gains for the flexible arm are $K_{p\theta} = 10.40$, $K_{pd} = 1.50$, $K_{D\theta} = -40$, and $K_{Dd} = 1.50$. The model had faster controls due to the assumption of no friction. Also, a rigid arm has faster control than a flexible arm. This code was specific to the configuration of the rotating arm setup, but the methods used can be generalized to any second-order control system.

Nomenclature

J	=	moment of inertia
$K_{D\theta}$	=	derivative control gain applied to the hub angle rate
$K_{P\theta}$	=	proportional control gain applied to the hub angle measurement
K_g	=	motor gear ratio
K_m	=	proportional motor constant
R_m	=	spin rate of the motor shaft
θ_D	=	desired output angle
θ_L	=	angular position
V_{in}	=	input voltage

I. Theory

A closed-loop control system is a system which utilizes the open-loop system concept as its forward path but has at least one feedback loop or path between the input and output. Closed-loop control systems are designed to be autonomous systems that maintain the desired output value by comparison with the actual output value. The transfer function itself is a mathematical relationship between the system's input value and output value. In addition, because of this relationship, the transfer function also describes the behavior of the system. The derivation of the closed-loop system began with obtaining an equation for the input voltage $\mathbf{V_{in}}$. This was done by solving for the $\mathbf{V_{in}}$ term in the open-loop transfer function which is shown below.

$$V_{in} = \frac{\Theta_L s(s + K_g^2 K_m^2/(J * R_m))}{K_g K_m/(J * R_m)} \tag{1}$$

For a constant desired angle, the equation to determine the applied voltage from a PD controller is given as $V_{in} = V_{pd} = K_{p\theta}(\Theta_D - \Theta_L) - sK_{D\Theta}\Theta_L$. Substituting equation (1) into the the equation for the PD controller input voltage we can obtain the Closed-Loop Transfer function. Equations (2) and (3) respectively show the initial substitution and the final derivation for the Closed-Loop Transfer Function. $\frac{\Theta_L}{\Theta_D}$.

$$\frac{\Theta_L s(s + K_g^2 K_m^2/(J * R_m))}{K_g K_m/(J * R_m)} = K_{p\theta}(\Theta_D - \Theta_L) - sK_{D\theta}\Theta_L \tag{2}$$

$$\frac{\Theta_L}{\Theta_D} = \frac{K_{p\theta}(K_g K_m/(J * R_m))}{[s^2 + s[K_g^2 K_m^2/(J * R_m) + K_{D\theta} K_g K_m/(J * R_m)] + K_{p\theta} K_g K_m/(J * R_m)]} \tag{3}$$

Show below is the control block diagram with the derived transfer function

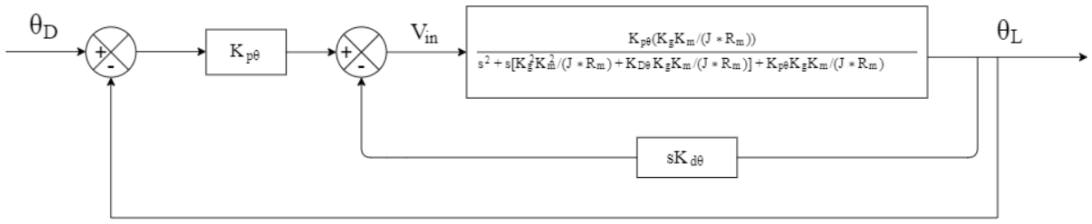

Figure 1: Control Block Diagram

Both the rigid and flexible arm used different methods to get the gains. Because the rigid arm has a traditional second order differential equation, the dynamic characteristics can be gained by mimicking the generalized second differential equation, $s^2 + 2\zeta w_n s + w_n^2$. This produces

$$w_n^2 = \frac{K_{p\theta} K_g K_m}{J R_m}$$

$$\zeta = \frac{K_g^2 K_m^2 + K_{D\theta} K_g K_m}{2\sqrt{K_{p\theta} K_g K_m J R_m}}$$

Also using other derived equations like $M_p = e^{\frac{-\zeta}{\sqrt{1-\zeta^2}}\pi}$ and $envelope = e^{-\zeta w_n t_s}$. These equations allow to solve for K_p and K_d for a given overshoot and settling time wanted. However the voltage used from the gains need to be check so that it does not go pass 10V as that is the limit of the system. For the flexible arm, the transfer function consist of a forth order differential equation. Unfortunately there is no way into solving these types of equation unless using advance techniques. Therefore $K_{p\theta}$, K_{pd}, $K_{D\theta}$, and K_{Dd} was iterated to fit a less than 10% overshoot, achieves 5% settling time in less than 1 seconds, and reduces tip deflection to less than 0.01m of residual vibrations.

II. Results and Analysis

After the closed-loop transfer function was derived for the rigid arm system, equation 3 along with the rigid arm design requirements were used together to determine acceptable gain values for $K_{p\theta}$ and $K_{D\theta}$. The requirements for rigid arm design were to compute gain values for the system to have less than 5% overshoot and achieve a 5% settling time in less than 0.15 seconds. Utilizing built-in Matlab functions included in the Matlab Control System toolbox several theoretical gain values were found to meet the design requirements. The values chosen were determined by solving for $K_{p\theta}$ and $K_{D\theta}$ using the equation in the Theory section.

$K_{p\theta}$	$K_{D\theta}$
19.41	-0.10

Table 1: Theoretical Gain values for Rigid Arm

After the theoretical values were computed, they were tested experimentally using the LabView VI software.

In Figure 2, the desired control, theoretical control, Matlab simulation, and experiment were compared. The theoretical model has a higher hub angle than the other experiment and Matlab simulation during the transition. This difference between the model, experiment, and Matlab simulation is probably due to friction. Friction would lower the speed the system would get to the desired Hub angle. That's why it would take longer for those two models to get to the desired angle. Another observation from the plot is that both the theoretical model and the experiment met the parameters of the overshoot and settling time limits but the Matlab simulation fell short. This is probably due to assumptions that were used in the Matlab simulation that dampen the system. For these $K_{p\theta} and K_{D\theta}$ values, there is almost no overshoot in all the plots.

As shown in the images above, there is a noticeable difference between the experimental and the theoretical model. A major factor for this is that the theoretical model does not account for friction between components or other forces acting on the system. In addition, it is possible that there is a delay with the response of the electronic components.
Control of the flexible arm system is complicated when compared to the rigid arm system. This was due to the inclusion of the flexible link. To combat this, more sensors were added along the arm to measure the deflection of the tip. The closed-loop transfer functions to determine tip deflection and the angular position of the flexible arm system were provided and are shown below.

$$\frac{\Theta_L}{\Theta_D} = \frac{K_1\left[r_1 s^2 + (q_1 r_2 - r_1 q_2)\right]}{s^4 + \lambda_3 s^3 + \lambda_1 s + \lambda_0} \tag{4}$$

$$\frac{D}{\Theta_D} = \frac{K_1\left[r_2 s^2 + (p_1 r_2 - r_2 p_2)s\right]}{s^4 + \lambda_3 s^3 + \lambda_1 s + \lambda_0} \tag{5}$$

The requirements for the flexible arm system differed from the rigid arm in which the theoretical gain values must allow the system to have less than 10%overshoot, achieve a 5% settling time in less than 1 second, and reduce tip deflection to less than 0.01 $[m]$ of residual vibrations.

Using equations 4 and 5 along with the inclusion of two new gains K_{pd} and K_{Dd} multiple values for $K_{p\theta}$, $K_{D\theta}$, K_{pd} and K_{Dd} were computed.

$K_{p\theta}$	K_{pd}	$K_{D\theta}$	K_{Dd}
10.40	1.50	-40	1.5

In Figure 3, there is just one big difference between the theoretical model and the Matlab simulation. Just like the rigid arm, the theoretical model has a higher hub angle than the Matlab simulation. This is probably because the Matlab simulation has friction accounted for as friction would add damping to the system. Besides this difference the model and the Matlab simulation and to the same value almost at the same time. On the other hand in Figure 4, the deflection in the Matlab simulation is much larger than the theoretical model. This difference is probably due to the material properties of the flexible arm. The model probably had a more rigid arm material property than the Matlab simulation which cause the large difference between the two deflections.

With the rigid and flexible arm, there were many differences in the results and implementation. In the results, it took the flexible arm a longer time get to a lower desired hub angel than the rigid arm. This is due to the gains that had to be implemented. Because the flexible arm had a tip deflection to be accounted for, mainly $K_{p\theta}$ had to be lowered so that the arm doesn't move fast enough to make the tip break. This allows the rigid arm to go to higher desired hub angles at a faster rate than the flexible arm. From this analysis, controls on flexible structures are much slower than that of rigid structures. So flexible structures need to be in an environment where it doesn't need to have fast motion.

III. Conclusions and Recommendations

Differential Equations, control system theory, and MATLAB were used together to compare a theoretical model to an experiment of a rigid and flexible rotating arm. The rotating arm has a design limitation which limited the gains that the system can achieve. The code analyzed the results of a step control on the rotating arm and the resulting steady-state. Through manipulating the gain values to get the desired results, the experience gained was how impactful these gains are and how mathematically some gains can make the system unstable. With this experiment, control limitations are due to the structures themselves. For future experimental setup or procedures, performing the experiment would yield a better intuitive understanding of how impactful these gains are. If these adjustments were made, control systems can be better understood.

IV. References

[1] Introduction to Dynamics and Systems University of Colorado at Boulder, *ASEN 2003 Lab 6: Rotary Position Control Experiemtn*, Project outline document, University of Colorado, 2020.

[2] Introduction to Dynamics and Systems University of Colorado at Boulder, *ASEN 2003 Lab 6: Rotary Position Control*, Project outline Video, University of Colorado, 2020.

[3] Introduction to Dynamics and Systems University of Colorado at Boulder, *ASEN 2003 Lab 6: Rotary Position Control*, MATLAB Simulation, University of Colorado, 2020.

[4] Introduction to Dynamics and Systems University of Colorado at Boulder, *ASEN 2003 Lab 6: Rotary Position Control*, Experimental Data, University of Colorado, 2020.

V. Acknowledgments

Thank you to our Lab instructors Bobby Hodgkinson and Josh Mellin, along with all the TA's for help and support throughout the lab and everything they have done for us in this crazy time.

VI. Appendices

A. Closed Loop Transfer Function Derivation

Open Loop transfer function

$$\frac{\theta_L}{V_{in}} = \frac{K_g K_m/(JR_m)}{s(s + K_g^2 K_m^2/(JR_m))}$$

Solving for V_{in}

$$V_{in} = \frac{(\theta_L)(s)(s + K_g^2 K_m^2/(JR_m))}{K_g K_m/(JR_m)}$$

Substitute above equation into Laplace Transform of V_{in}

Laplace Transform: $V_{in} = V_{pd} = K_{p\theta}(\theta_D - \theta_L) - sK_{d\theta}\theta_L$

$$\theta_L\left(\frac{s(s+K_g^2K_m^2/(JR_m))}{K_gK_m/(JR_m)}\right) = K_{p\theta}(\theta_D - \theta_L) - sK_{d\theta}\theta_L$$

Solve for closed loop Transfer Function $\frac{\theta_L}{\theta_D}$

$$\theta_L\left(\frac{s(s+K_g^2K_m^2/(JR_m))}{K_gK_m/(JR_m)}\right) = K_{p\theta}\theta_D - \theta_L(K_{p\theta} + sK_{d\theta})$$

$$\theta_L\left(\frac{s(s+K_g^2K_m^2/(JR_m))}{K_gK_m/(JR_m)}\right) + \theta_L(K_{p\theta} + sK_{d\theta}) = K_{p\theta}\theta_D$$

$$\theta_L(s)(s+K_g^2K_m^2/(JR_m)) + \theta_L(K_{p\theta}+sK_{d\theta})(K_gK_m/(JR_m)) = K_{p\theta}\theta_D(K_g/K_m$$

$$\theta_L(s)(s+K_g^2K_m^2/(JR_m)) + \theta_L(K_{p\theta}+sK_{d\theta})(K_gK_m/(JR_m)) = K_{p\theta}\theta_D(K_gK_m/(JR_m))$$

$$\theta_L(s)(s+K_g^2K_m^2/(JR_m)) + \theta_L(K_{p\theta}+sK_{d\theta})(K_gK_m/(JR_m)) = K_{p\theta}\theta_D(K_g/K_m$$

$$\theta_L(s)(s+K_g^2K_m^2/(JR_m)) + \theta_L(K_{p\theta}+sK_{d\theta})(K_gK_m/(JR_m)) = K_{p\theta}\theta_D(K_gK_m/(JR_m))$$

$$\theta_L(s^2 + sK_g^2K_m^2/(JR_m) + K_{p\theta}K_gK_m/(JR_m) + sK_{d\theta}K_gK_m/(JR_m)) = K_{p\theta}\theta_D(K_gK_m/(JR_m))$$

$$\frac{\theta_L}{\theta_D} = \frac{K_{P\theta}\,\theta_D\,(K_g K_m/(JR_m))}{(s^2 + s(K_g^2 K_m^2/(JR_m) + K_{D\theta} K_g K_m/(JR_m)) + K_{P\theta} K_g K_m/(JR_m))}$$

Characteristic Polynomial of second order system

$$s^2 + 2\zeta\omega_n s + \omega_n^2$$

Denominator of closed loop transfer function

$$(s^2 + s(K_g^2 K_m^2/(JR_m) + K_{D\theta} K_g K_m/(JR_m)) + K_{P\theta} K_g K_m/(JR_m))$$

$$2\zeta\omega_n s = s(K_g^2 K_m^2/(JR_m) + K_{D\theta} K_g K_m/(JR_m)) \quad (*)$$

looking at the denominator we can see $\omega_n^2 = \frac{K_{P\theta} K_g K_m}{JR_m}$

Substituting into eqn 1

$$2\zeta\sqrt{\frac{K_{P\theta} K_g K_m}{JR_m}} = \frac{K_g^2 K_m^2 + K_{D\theta} K_g K_m}{JR_m}$$

$$2\zeta = \frac{(K_g^2 K_m^2 + K_{D\theta} K_g K_m)\sqrt{JR_m}}{JR_m(\sqrt{K_{P\theta} K_g K_m})}$$

$$\zeta = \frac{(K_g^2 K_m^2 + K_{D\theta} K_g K_m)}{2\sqrt{JR_m K_{P\theta} K_g K_m}}$$

B. Figures

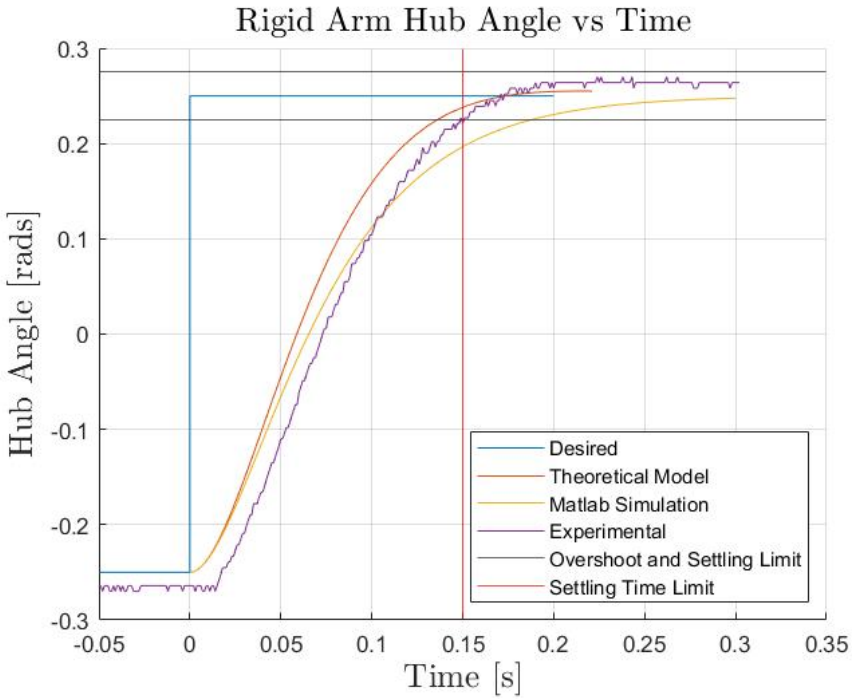

Figure 2: The black horizontal lines are the 5% settling and overshoot limits. The vertical red line is the 0.15 second settling time limit

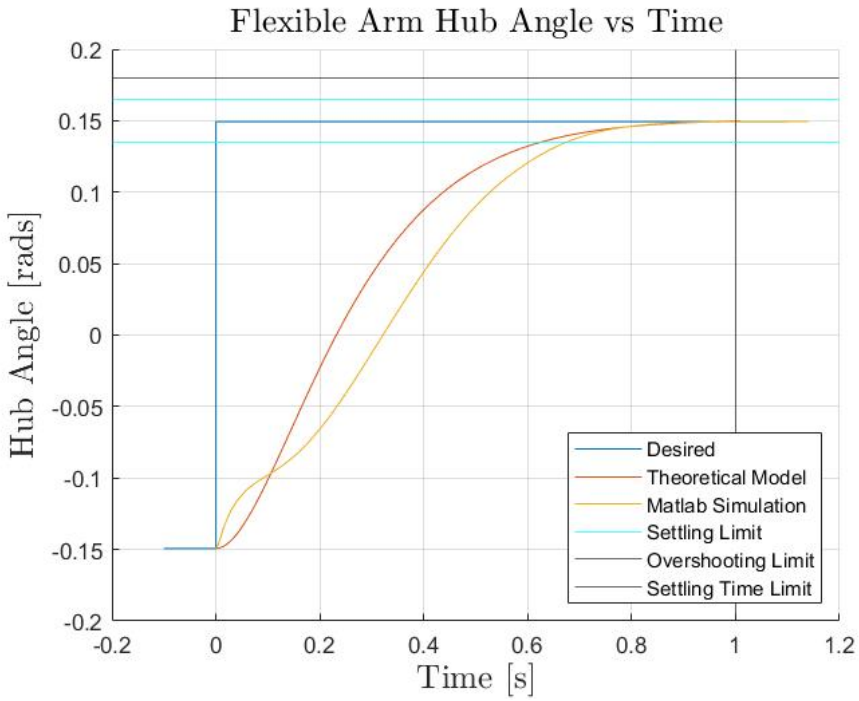

Figure 3: The cyan horizontal lines are the 5% settling limits and the black horizontal line is the overshoot limit. The vertical red line is the settling time limit of 1 seconds.

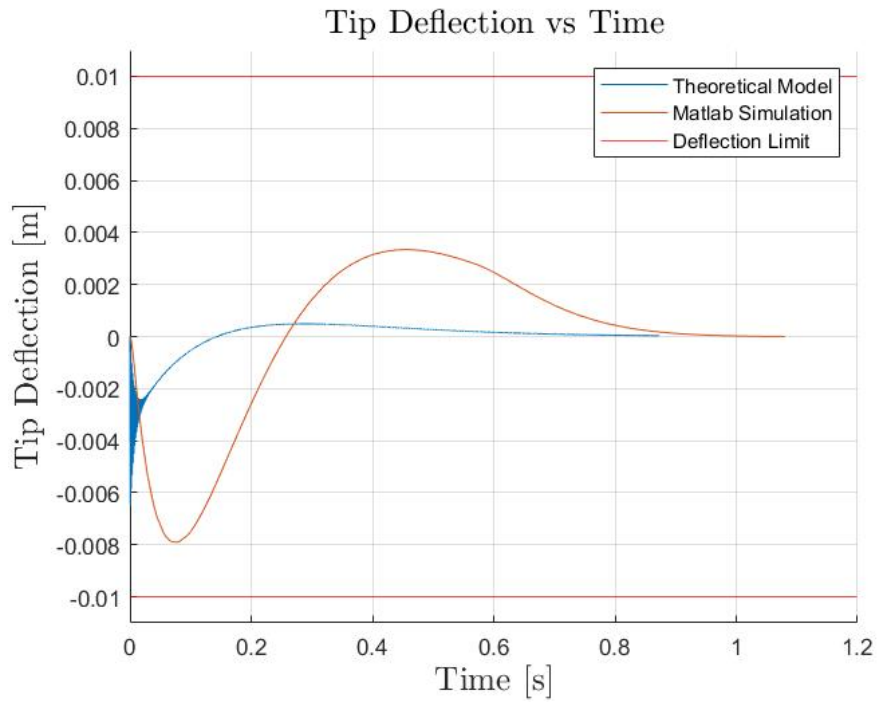

Figure 4: The red horizontal lines are the 0.01m limit of the flexible arm system.

C. Team Participation

	Derivations	Code	Write Up
Connor O'Reilly	2	1	2
Jason Le	2	2	1

Initials: CO (Connor) JL (Jason)

D. Matlab Code

```
%% ASEN 2003: Rotary Position Control Experiment %%
% Author: Jason Le and Connor Reilly
% Date created: April 7, 2020
% Date modified: April 26, 2020
% Purpose: Analysis Controls Systems on a rotating arm with different gains
% Inputs: Given variables and experiment data
% Outputs: Plots on the control system results
% Assumptions: Frictionless

%% Housing Keeping
clear
close all
clc

%% Parsing
%load for rigid data and fixing it
load('RigidExperiment.mat')
data = readtable('Experiment.txt');
```

```
ExperimentT = (data.Var1(2752:3150)*0.001)-1132.5;
ExperimentH = data.Var2(2752:3150);
ExperimentSimT = (time(1001:1301,1)*0.001)-1;
ExperimentSimH = (time(1001:1301,2));

%load for flexible data and fixing it
load('FlexibleExperiment.mat')
data2 = readtable('FlexibleExperiment.txt');
FExperimentT = (data2.Var1(933:1570)*0.001)-4785.020;
FExperimentH = data2.Var2(933:1570);
FExperimentD = data2.Var3(933:1570);
FExperimentSimT = (time2(894:1000,1)*0.001)-15;
FExperimentSimH = (time2(894:1000,2));
FExperimentSimD = -(time2(1:93,3));
FExperimentSimT2 = (time2(1:93,1)*0.001);

%% Rigid Arm Variables
ts = 0.04; %Setting time
os = 0.01; %Overshoot
%These varibles are given
Kg = 33.3;
Km = 0.0401;
Rm = 19.2;
Jh = 0.0005;
Jl = 0.0015;
J = Jh + Jl;
[Kp,Kd] = my_func(ts,os,Kg,Km,J,Rm);
rigid_values = [Kp,0,Kd,0];
%% Rigid Computing
%These section uses functions by matlab
n1 = (Kp*Kg*Km)./(J*Rm);
d2 = ones(1,numel(Kp));
d1 = ((((Kg^2)*(Km^2))./(J*Rm))+((Kd*Kg*Km)./(J*Rm)));
d0 = ((Kp*Kg*Km)./(J*Rm));
numrigid = n1';
denrigid = [d2;d1;d0]';
%Plot of rigid arm
figure()
hold on
%This data is for desired
Rigid_desiredH = [-0.25 -0.25 0.25 0.25];
Rigid_desiredT = [-0.05 0 0 0.2];
plot(Rigid_desiredT,Rigid_desiredH)
for i = 1:numel(Kp)
    sysTF = tf(numrigid(i,:),denrigid(i,:));
    [x_rigid,t_rigid] = step(sysTF);
    x_rigid = (x_rigid*0.5)-0.25;
    plot(t_rigid,x_rigid)
end
plot(ExperimentSimT,ExperimentSimH)
plot(ExperimentT,ExperimentH)
xlabel('Time [s]','fontsize',15,'interpreter','latex')
ylabel('Hub Angle [rads]','fontsize',15,'interpreter','latex')
title('Rigid Arm Hub Angle vs Time','fontsize',15,'interpreter','latex')
grid on
```

```
yline(0.275)
xline(0.15,'r')
yline(0.225)
xlim([-0.05 0.35])
legend('Desired','Theoretical Model','Matlab ...
    Simulation','Experimental','Overshoot and Settling Limit','Settling ...
    Time Limit','Location','SouthEast')
hold off

%% Flexible Arm Variables
%These variables are provided
fc = 1.8;
Jh = 0.0005;
Jarm = 0.0040;
Jm = 0.01;
Jl = Jarm + Jm;
L = 0.45;
Ka = ((2*pi*fc).^2)./Jl;
p1 = -((Kg^2)*(Km^2))./(Jh*Rm);
p2 = ((Kg^2)*(Km^2)*L)./(Jh*Rm);
q1 = Ka./(L*Jh);
q2 = -(Ka*(Jh+Jl))./(Jl*Jh);
r1 = (Kg*Km)./(Jh*Rm);
r2 = -(Kg*Km*L)./(Jh*Rm);

%% Flexible Arm Computing
%These variables were chosen
Kp = 10.4;
Kd = 1.5;
Kpd = -40;
Kdd = 1.5;
K1 = Kp;
K2 = Kpd;
K3 = Kd;
K4 = Kdd;
l3 = -p1+(K3*r1)+(K4*r2);
l2 = -q2+(K1*r1)+(K2*r2)+(K4*((p2*r1)-(r2*p1)));
l1 = (p1*q2)-(q1*p2)+(K3*((q1*r2)-(r1*q2)))+(K2*((p2*r1)-(r2*p1)));
l0 = K1*((q1*r2)-(r1*q2));
%This section prepares to use matlab functions
fn3l = (K1*r1);
fn2l = zeros(1,numel(Kp));
fn1l = K1*((q1*r2)-(r1*q2));
fd4 = ones(1,numel(Kp));
fn3d = (K1*r2);
fn2d = K1*((p2*r1)-(r2*p1));
fn1d = zeros(1,numel(Kp));
%Matlab functions
num_flex_L = [fn3l;fn2l;fn1l]';
denflex = [fd4;l3;l2;l1;l0]';
num_flex_D = [fn3d;fn2d;fn1d]';
sysTF2 = tf(num_flex_L(1,:),denflex(1,:));
[x_flex,t_flex] = step(sysTF2);
x_flex = (x_flex*0.3)-0.1493;
sysTF3 = tf(num_flex_D(1,:),denflex(1,:));
```

```
[x_flexd,t_flexd] = step(sysTF3);
x_flexd = x_flexd * 100;
x_fdesireH = [-0.1493 -0.1493 0.1493 0.1493];
x_fdesireT = [-0.1 0 0 1.1];
%Plot of flexible arm
figure()
hold on
plot(x_fdesireT,x_fdesireH)
plot(t_flex,x_flex)
plot(FExperimentSimT,FExperimentSimH)
yline(0.1650,'c')
yline(0.18)
xline(1)
yline(0.1350,'c')
%yline(0.12)
ylim([-0.2 0.2])
xlabel('Time [s]','fontsize',15,'interpreter','latex')
ylabel('Hub Angle [rads]','fontsize',15,'interpreter','latex')
title('Flexible Arm Hub Angle vs Time','fontsize',15,'interpreter','latex')
grid on
legend('Desired','Theoretical Model','Matlab Simulation','Settling ...
    Limit','Overshooting Limit','Settling Time Limit','Location','SouthEast')
hold off
%Plot of deflection
figure()
hold on
plot(t_flexd,x_flexd)
plot(FExperimentSimT2,FExperimentSimD)
%plot(FExperimentT,FExperimentD)
xlabel('Time [s]','fontsize',15,'interpreter','latex')
ylabel('Tip Deflection [m]','fontsize',15,'interpreter','latex')
title('Tip Deflection vs Time','fontsize',15,'interpreter','latex')
grid on
ylim([-0.011 0.011])
yline(0.01,'r')
yline(-0.01,'r')
legend('Theoretical Model','Matlab Simulation','Deflection ...
    Limit','Location','NorthEast')
hold off

%% Function
function [Kp,Kd] = my_func(ts,os,Kg,Km,J,Rm)
%Purpose:Get Kp and Kd
%Input:Properties of the system and desire times and overshoot
%Output:Kp and Kd
Kp = zeros(1,numel(ts));
Kd = zeros(1,numel(ts));
for i = 1:numel(ts)
    zeta = sqrt((log(os(i))./pi).^2./(1+(log(os(i))./pi).^2));
    wn = log(ts(i))./(-zeta*0.15);
    Kp(i) = (wn.^2*J*Rm)./(Kg*Km);
    Kd(i) = ((zeta*2*sqrt(Kp(i)*Kg*Km*J*Rm))-((Kg.^2)*(Km.^2)))./(Kg*Km);
end
end
```