## Table of Contents



```
clear;clc;close all;

%Code by: Zhixing Yao
%Edited and Analysed by: Zak Reichenbach
```

# Experimental Data

```
path = "Flexible Arm Data/";
files = dir(path+"*.txt");
filename = files(randi([1,length(files)])).name;
data = load(path+filename);
time = (data(:,1)-data(1,1))./1000;
theta = data(:,2);
```

# Constants

```
Kg = 33.3; % gear ratio
Km = 0.0401; % motor constant
Rm = 19.2; % armature resistance [ohm]
J_hub = 0.0005;
J_load = .0015;
J = J_hub+J_load; % total inertia [kg*m^2]

L = .45; % link length [m]
Marm = 0.06; % link mass of stainless steel ruler [kg]
J_arm = 0.0041; % [kg*m^2]
Mtip = .050;  % [kg]
J_M = 0.01; % [kgm^2]
J_L = J_arm + J_M; % [kg*m^2]
fc = 1.8; % natural frequency [Hz]
K_arm = ((2*pi*fc)^2*(J_L)); % stiffness (Jl+Jm)

p1 = -Kg^2*Km^2/(J_hub*Rm);
q1 = K_arm/(L*J_hub);
r1 = Kg*Km/(J_hub*Rm);
p2 = Kg^2*Km^2*L/(J_hub*Rm);
q2 = -K_arm*(J_hub + J_L)/(J_L*J_hub);
r2 = -Kg*Km*L/(J_hub*Rm);
```

# Transfer Function

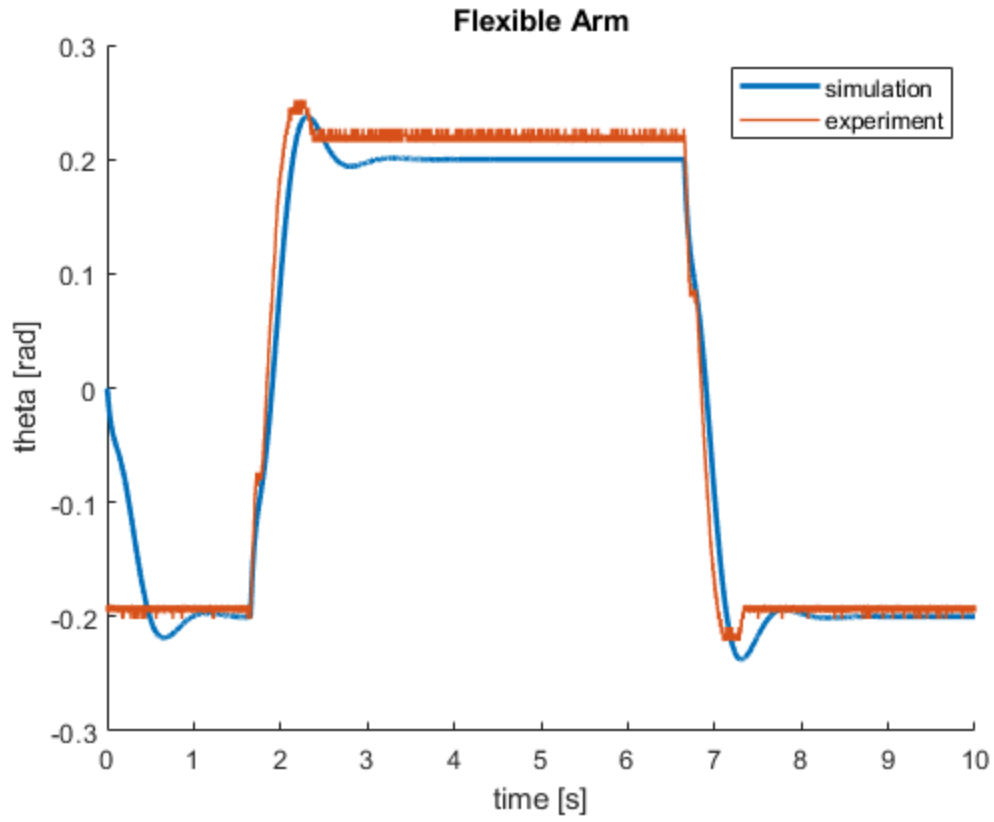
```
fpara = strsplit(filename,"_");
```

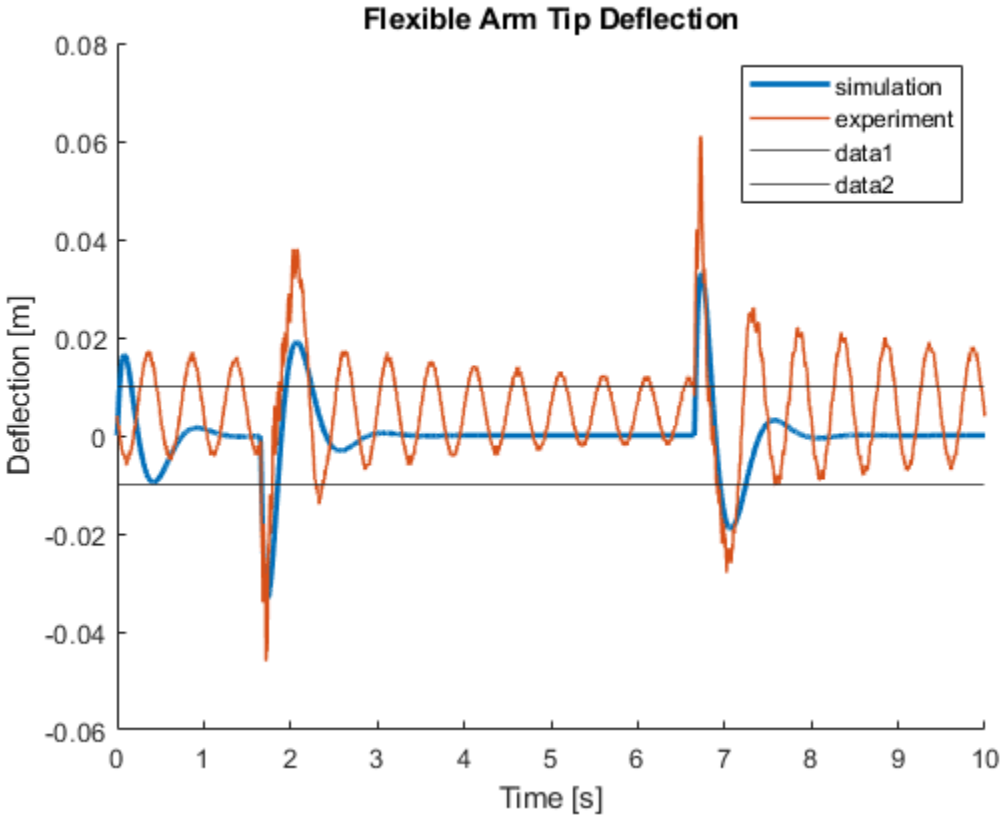
```
K1 = 10 ;      % str2double(fpara{2});  %hub P
K2 =  -2.5;     % str2double(fpara{3}); %Defletion P
K3 = .8;       %str2double(fpara{4});  %Hub D
K4 =  .5 ;    % str2double(strrep(fpara{5},".txt","")); %Deflection D
lambda0 = K1*(q1*r2-q2*r1);
lambda1 = p1*q2-q1*p2+K3*(q1*r2-r1*q2)+K2*(p2*r1-r2*p1);
lambda2 = -q2+K1*r1+K2*r2+K4*(p2*r1-r2*p1);
lambda3 = -p1+K3*r1+K4*r2;

T = time;
U = data(:,6);

num1 = [K1*r1, 0, K1*(q1*r2-r1*q2)];
den1 = [1, lambda3, lambda2, lambda1, lambda0];
sysTF1 = tf(num1,den1);
[thetaSim,timeSim,~] = lsim(sysTF1,U,T);


num2 = [K1*r2, K1*(p2*r1-r2*p1), 0];
den2 = [1, lambda3, lambda2, lambda1, lambda0];
sysTF2 = tf(num2,den2);
[tipSim,tipTimeSim,~] = lsim(sysTF2,U,T);


s = stepinfo(sysTF1,'SettlingTimeThreshold',.05);

s2 = stepinfo(sysTF2,'SettlingTimeThreshold',.05);
```

# Plot and Compare

```
figure(1);
hold on;
plot(timeSim,thetaSim,"LineWidth",2);
plot(time,theta,"LineWidth",1);
title("Flexible Arm");
legend("simulation","experiment");
ylabel("theta [rad]");
xlabel("time [s]");

figure;
hold on;
plot(tipTimeSim,tipSim,"LineWidth",2);
plot(time,data(:,3),"LineWidth",1);
title("Flexible Arm Tip Deflection");
legend("simulation","experiment");
xlabel("Time [s]");
ylabel("Deflection [m]");
yline(.01)
yline(-.01)
fprintf('K1 (Proportional Hub Angle):%f K2 (Proportional Deflection):
%f \nK3 (Derivative Hub Angle):%f    K4 (Derivative Deflection):%f
\n',K1,K2,K3,K4)
```

```
ans =

  ConstantLine with properties:

    InterceptAxis: 'y'
            Value: 0.0100
            Color: [0.1500 0.1500 0.1500]
        LineStyle: '-'
        LineWidth: 0.5000
            Label: ''
      DisplayName: ''

  Use GET to show all properties


ans =

  ConstantLine with properties:

    InterceptAxis: 'y'
            Value: -0.0100
            Color: [0.1500 0.1500 0.1500]
        LineStyle: '-'
        LineWidth: 0.5000
            Label: ''
      DisplayName: ''

  Use GET to show all properties

K1 (Proportional Hub Angle):10.000000 K2 (Proportional
 Deflection):-2.500000
K3 (Derivative Hub Angle):0.800000    K4 (Derivative
 Deflection):0.500000
```


Flexible Arm
simulation
experiment
theta [rad]
time [s]
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
1
2
3
4
5
6
7
8
9
10


Flexible Arm Tip Deflection
simulation
experiment
data1
data2
Deflection [m]
Time [s]
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
0
1
2
3
4
5
6
7
8
9
10

*Published with MATLAB® R2019a*