MECA 482 PROJECT

Control System Design of Furuta Pendulum

California State University, Chico

Sultan Alharbi

Ali Yousef

Antonio Figueroa

Khalid Alkendari

Brandon Burdick

May 18th, 2020

Introduction

In definition, .“The Furuta Pendulum (rotational inverted pendulum) consists of a driven arm which rotates in the horizontal plane and a pendulum attached to that arm which is free to rotate in the vertical plane”. [1]

In this project we are designing a control system of Furuta Pendulum by creating a mathematical model of it; to build an algorithmic test platform for the system. The project will be completely virtual. The Mathematical model and the controllers design will be demonstrated in MATLAB. Furthermore, the system will have a simulation with the control system and mathematical model by connecting Coppelia Sim to MATLAB

Project Description and Approach

Our goal is to design, build, and control a rotational inverted pendulum to be balanced upright. For a good start with our project, we began with a review of available researches about rotational inverted pendulum systems. Also, selected the project components, such as the motor and encoder. Furthermore, a mathematical model was developed. The system of dynamic equations rotational inverted pendulum can be derived by two methods:

1. Newtonian methods using free body diagrams
2. The Euler-Lagrange method

The Euler-Lagrange method derives the system equations by applying the Euler-Lagrange equation to the Lagrangian. Moreover, the Lagrangian is defined as the difference between potential and kinetic energy L=T-V and it simplifies the mathematical derivation greatly.

System Modeling

- Sample Furuta System from Quanser [2]

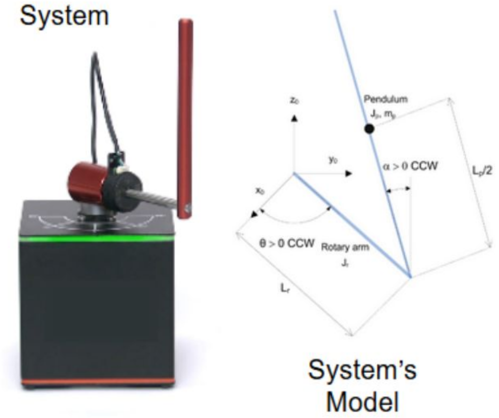


Figure 1: The System's Model

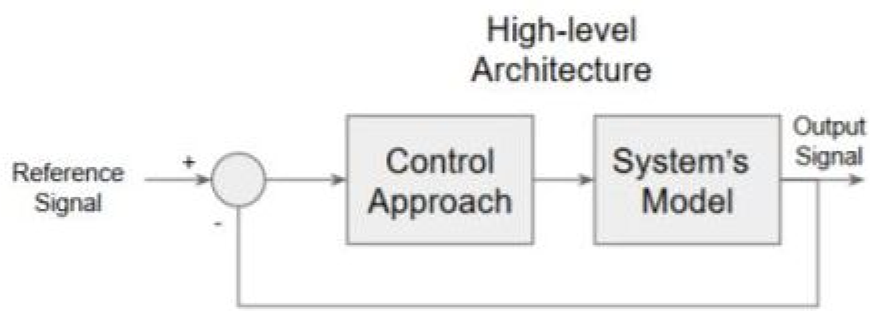


Figure 2: The theoretical control model of the Furuta Pendulum

- Sketches

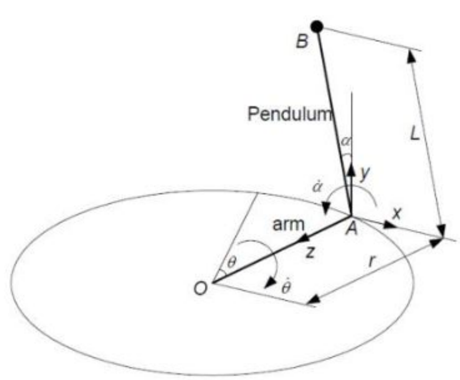


Figure 3: System Modelling Sketch

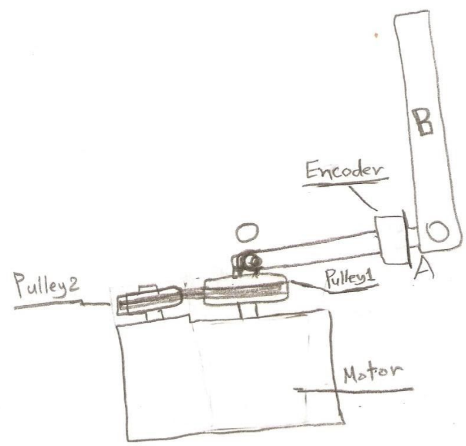


Figure 4: System Modeling Schematic

- System's selected components and parameters
  - Some values are estimated for simplicity and to set a close to reality parameters

Description	Symbol	Value
Pulley1 + arm mass	m_o	0.05 kg
Pulley1 radius	r_o	0.04 m
Arm length	l_o	0.1 m
Pulley1 rod length	*l_rod*	0.120 m
Pully1 rod mass	m_{rod}	0.015 kg
pendulum length	*l_p*	0.30 m
pendulum mass	m_p	0.05 kg
Encoder mass	m_e	0.02 kg
Motor Resistance	R	12.50 Ohm
Motor Constants	K	0.2751

Mathematical Model

The mathematical models allow us to design, test and develop controllers. We will demonstrate our mathematical model using:

- Free Body Diagram (See Figure2)
- Lagranian Formulation.

First, we will define the dynamic system to get an accurate mathematical model. Using the parameters above to set the mathematical model, we have,

$$\overline{r}_{B/A} = \frac{1}{2}l_2(-sin\theta_2\widehat{i} + cos\theta_2\widehat{j}) \Rightarrow \overline{v}_{B/A} = \dot{\overline{r}}_{B/A} = \frac{1}{2}l_2(-cos\theta_2\widehat{i} - sin\theta_2\widehat{j})$$

$$\overline{v}_{A/O} = \dot{\theta}_1 l_1\widehat{i}$$

$$\overline{v}_{B/O} = \overline{v}_{B/A} + \overline{v}_{A/O} = (\dot{\theta}_1 l_1 - \frac{1}{2}l_2\dot{\theta}_2 cos\theta_2)\widehat{i} - \frac{1}{2}l_2\dot{\theta}_2 sin\theta_2\widehat{j}$$

Select point O the datum for potential energy. So, the potential energies are:

$$V_1 = 0,\ V_2 = m_2 g(\frac{1}{2}l_2 cos\theta_2),\ V = V_1 + V_2$$

Thus, the arm's kinetic energy consist of the rotational kinetic energy of the large pulley, the rotational kinetic energy of the copper rod, and the translational kinetic energy of the 1250 CPR encoder (model the encoder as a point mass):

$$T_1 = \frac{1}{2}I_1\dot{\theta}_1^2 + \frac{1}{2}I_c\dot{\theta}_1^2 + \frac{1}{2}m_e v_{A/O}^2 = \frac{1}{2}\cdot\frac{1}{2}m_1 r_1^2\cdot\dot{\theta}_1^2 + \frac{1}{2}\cdot\frac{1}{3}m_c l_c^2\cdot\dot{\theta}_1^2 + \frac{1}{2}m_e l_1^2\dot{\theta}_1^2$$

Kinetic energy for link 2 (the pendulum) is

$$T_2 = \frac{1}{2}I_2\dot{\theta}_2^2 + \frac{1}{2}m_2 v_{B/O}^2 = \frac{1}{2}\frac{1}{12}m_2 l_2^2\dot{\theta}_2^2 + \frac{1}{2}[(\dot{\theta}_1 l_1 - \frac{1}{2}l_2\dot{\theta}_2 cos\theta_2)^2 + (-\frac{1}{2}l_2\dot{\theta}_2 sin\theta_2)^2]$$

Total kinetic energy:

$$T = T_1 + T_2$$

Lagrangian:

$$L = T - V$$

Ignore friction of the encoder shaft, we have equations of motions:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = \tau$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} = 0$$

And we have

$$T = Ki$$

Because we ignore motor inductance, we have

$$V = Ri + K\dot{\theta}_1$$

The above two equations will give us

$$\tau = \frac{K}{R}V - \frac{K^2}{R}\dot{\theta}_1$$

The input for the system is voltage, thus

$$u = V$$

Select states

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

Controller Design

1. Full State Feedback Controller

The state space model created from the system's modeling. We used MATLAB to design a full state feedback controller. See (Appendix A).

The feedback gains are determined by using LQR method (linear quadratic regulator). The Results shows the feedback gains K and the system eigenvalues. (See Appendix A)

K =

-23.1455 150.1343 -6.9158 16.2056

EIG =

-6.8096 + 2.8245i
-6.8096 - 2.8245i
-30.0063 +27.4181i
-30.0063 -27.418

The results of the simulation:

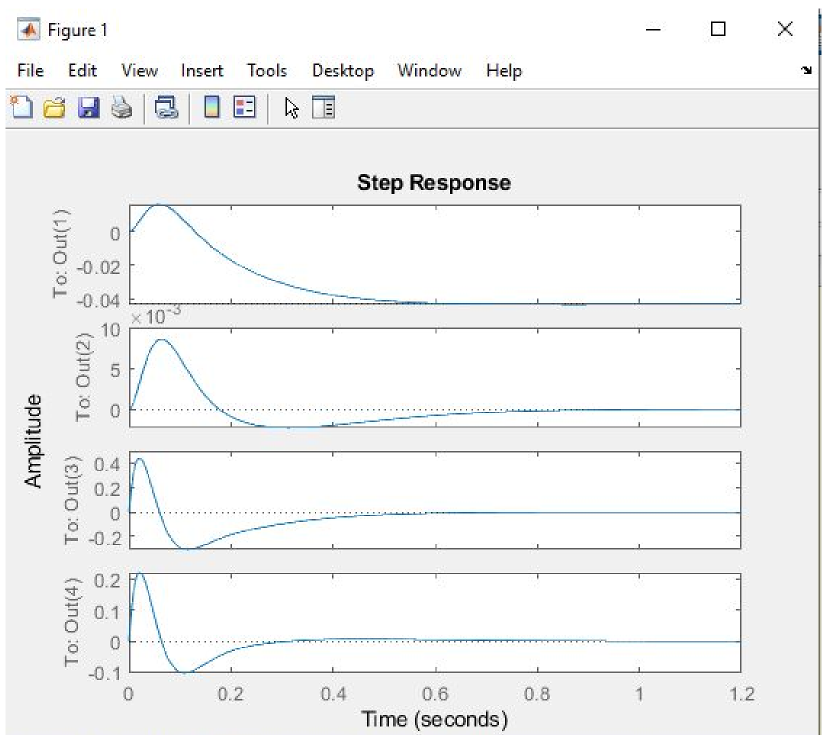


Figure 4: Full State Feedback Controller Step Response

2. PID Controller
    a. By setting the system's parameters and using MATLAB PID controller code we got the following results.
    b. The variables:

i. K_p, K_d,K_i represent the K Gains of the system. The values of these Gains tune the system's performance.

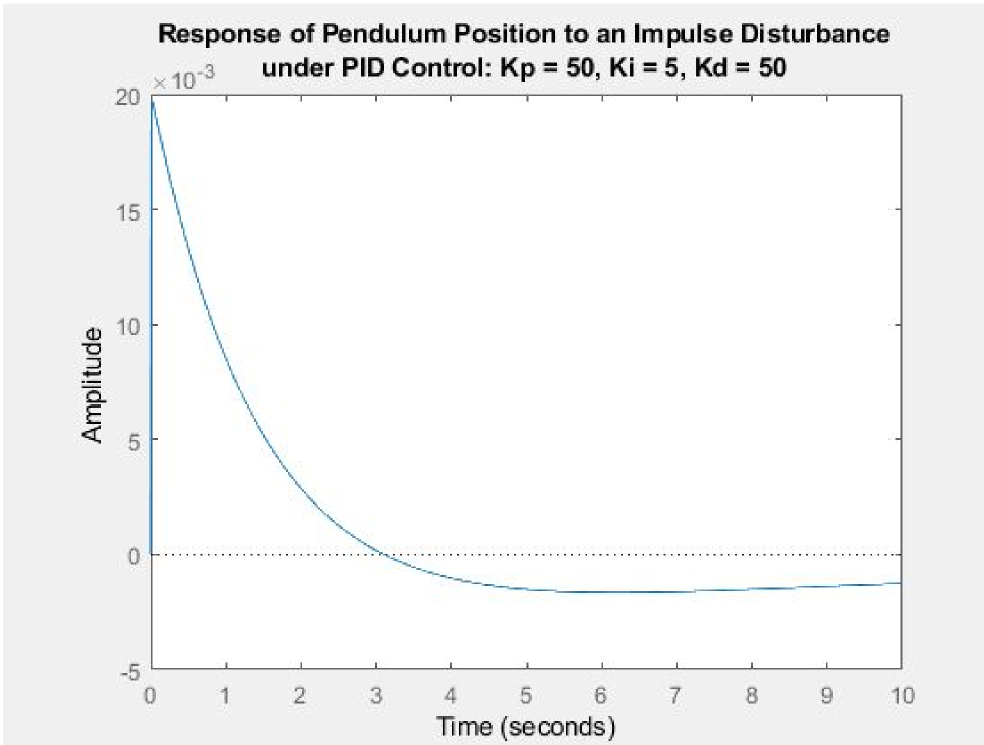


Figure 5: Response of Pendulum Position to an Impulse Disturbance under PID Control

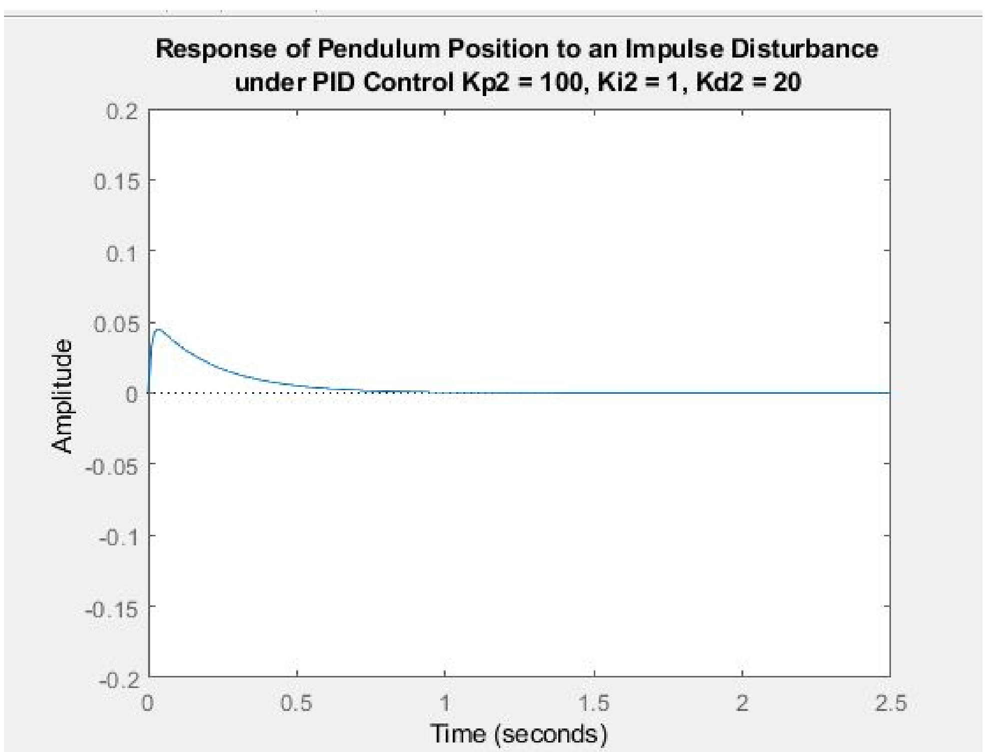


Figure 6: Response of Pendulum Position to an Impulse Disturbance under PID Control

V-REP: Furuta Pendulum designed using Coppelia

Due to difficulties we couldn't connect MATLABand Coppelia.

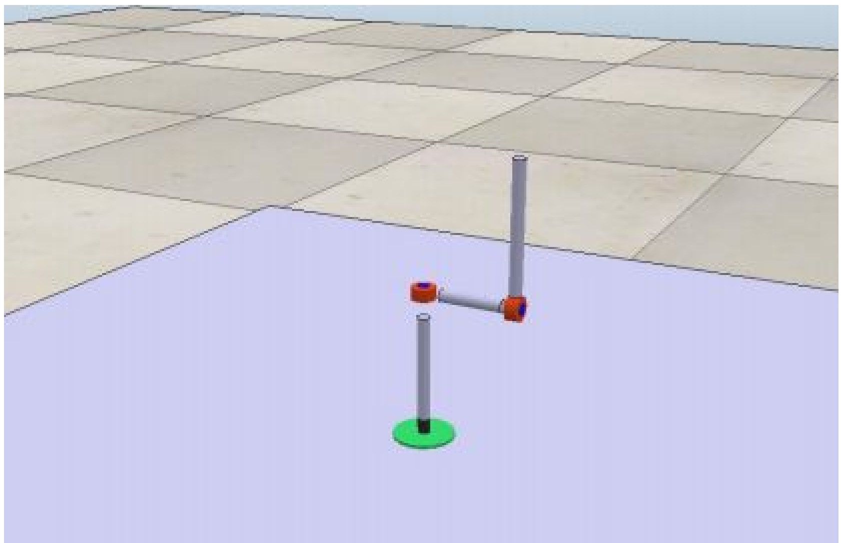


Figure 7: Image of the Furuta Pendulum designed using Coppelia (former V-REP)

REFERENCES

[1] Wikipedia, Furuta Pendulum, accessed by (05, 05, 2020) https://en.wikipedia.org/wiki/Furuta_pendulum

[2] Quanser Consulting Inc (2020), www.quanser.com accessed (11/05/2020)

Mathworks, (2019) World Wide Web link, www.mathworks.com/ accessed (11/5/2020)

Utkin, V., Gulder, J., SHI, J. (1999). Sliding Mode Control In Electromechanical Systems, Pudstow:T.J. International Ltd.

Zander, A, 2004, Vibrations 3012, Undergraduate course notes, School of Mechanical Engineering, The University of Adelaide.

APPENDIX A

The Matlab scripts for the inverted pendulum systems.

```
%% MECA 482, 05/18/2020, Group Ptoject, (Group ID# 13)
clear;close all;clc;
% System parameters
g=9.8; % gravity  [m/s^2]
ro=0.04; % Pulley1 radius [m]
lo=0.1; % Arm length [m]
```

```
l_rod=0.120; % Pulley1 rod [m]
l_p=0.30; % Pendulum length [m]
mo=0.05; % Pulley1 + arm mass [kg]
m_rod=0.015; % Pulley1 rod mass [m]
me=0.02; % Encoder mass [kg]
m_p=0.05; %Pendulum mass [m]
R=12.50; % Motor Resistance  [Ohm]
K=0.2751; % Motor Constant

% A matrix
A32 = 3*g*lo*m_p/(4*l_rod^2*m_rod+3*lo^2*(m_p+4*me)+6*mo*ro^2);
A33 = -12*K^2/(R*(4*l_rod^2*m_rod+3*lo^2*(m_p+4*me)+6*mo*ro^2));
A42 = 3*(g+9*g*lo^2*m_p/(4*l_rod^2*m_rod+3*lo^2*(m_p+4*me)+6*mo*ro^2))/(2*l_p);
A43 = -18*K^2*lo/(R*l_p*(4*l_rod^2*m_rod+3*lo^2*(m_p+4*me)+6*mo*ro^2));
A = [0 0 1 0; 0 0 0 1; 0 A32 A33 0; 0 A42 A43 0];

% B matrix
B3 = 12*K/(R*(4*l_rod^2*m_rod+3*lo^2*(m_p+4*me)+6*mo*ro^2));
B4 = 18*K*lo/(R*l_p*(4*l_rod^2*m_rod+3*lo^2*(m_p+4*me)+6*mo*ro^2));
B = [0; 0; B3; B4];

% C matrix
C = eye(4);

% D matrix
D = 0;

% LQR
Q = diag([1.5 6 0 0]);
R = 0.0028;
[K, S, EIG] = lqr(A, B, Q, R);
display(K);
display(EIG);

% system simulation
sys = ss(A,B,C,D);
sys_feedback = feedback(sys,K);
step(sys_feedback);
```

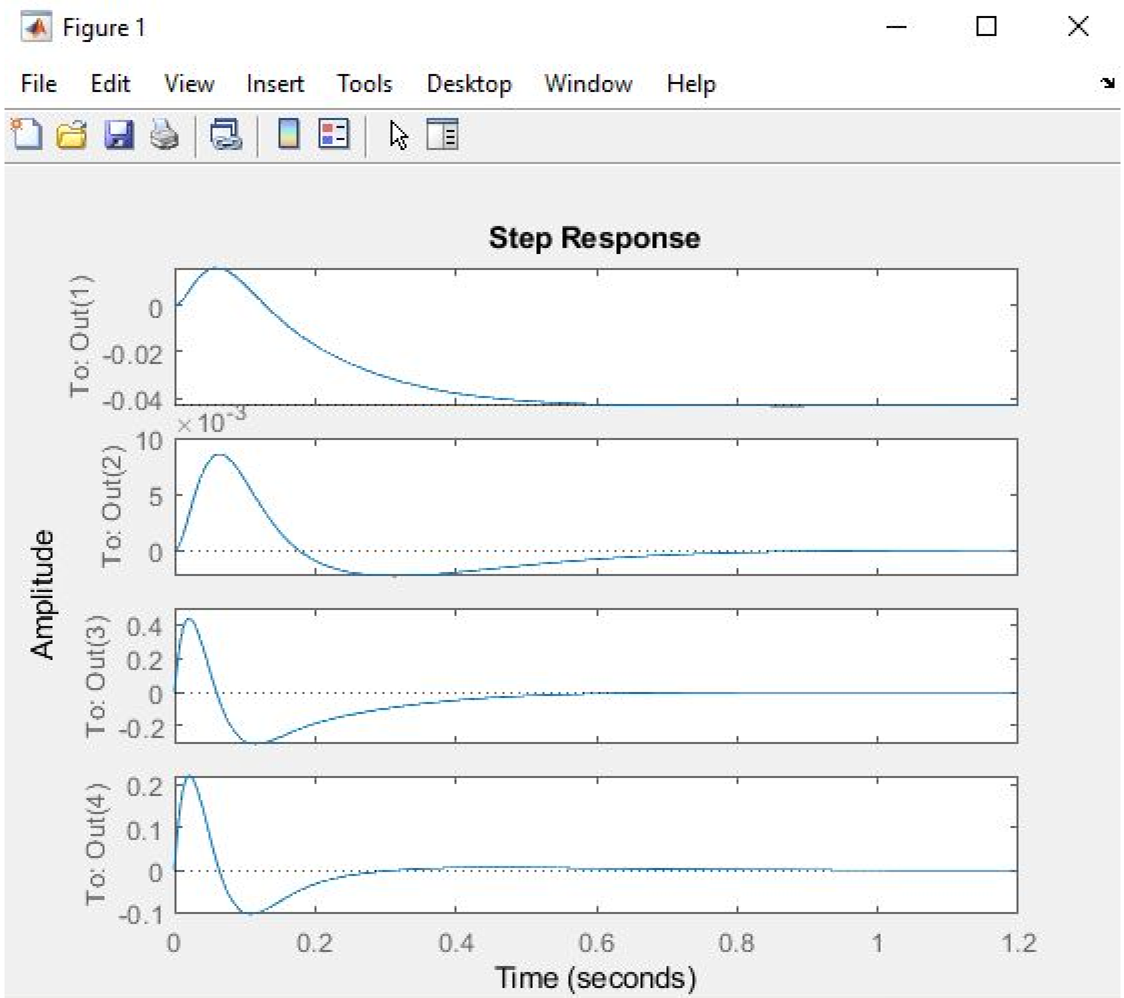
Results:

```
K =

  -23.1455  150.1343   -6.9158   16.2056
```

EIG =

-6.8096 + 2.8245i
-6.8096 - 2.8245i
-30.0063 +27.4181i
-30.0063 -27.4181i



```
%% MECA482, 05/18/2020,Group 13 Project,
% PID Controller Example
%% Parameters and attempt one

m_p = 0.5;
m = 0.5;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;
q = (m_p+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');
P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((m_p + m)*m*g*l)*s/q - b*m*g*l/q);

Kp = 50;
Ki = 5;
Kd = 50;
C = pid(Kp,Ki,Kd);
T = feedback(P_pend,C);

t=0:0.01:10;
impulse(T,t)
title({'Response of Pendulum Position to an Impulse Disturbance';'under PID Control: Kp = 50,
Ki = 5, Kd = 50'});

%% attempt two
Kp2 = 100;
Ki2 = 1;
Kd2 = 20;
C2 = pid(Kp2,Ki2,Kd2);
T2 = feedback(P_pend,C2);
t2=0:0.01:10;
impulse(T2,t2)
axis([0, 2.5, -0.2, 0.2]);
```

```
title({'Response of Pendulum Position to an Impulse Disturbance';'under PID Control Kp2 = 100, Ki2 = 1, Kd2 = 20'});
```