ENSC 483 – Modern Control Systems

Project 2 – Inverted Pendulum

Spring 2016

April 20, 2016

Group 22

Lestley A. Gabo – 301170055

Steven Luu – 301150253

Steven Liu – 301086295

Abstract

An analysis of the behavior of an inverted pendulum system. The inverted pendulum system is analyzed and experimented on its controllability, observability, and stability. We will perform our experiment using MATLAB and then use the data to interpret our own inverted pendulum system in Simulink. Then we will test the performance of our experimental controller and observer on the Inverted Pendulum apparatus inside the lab.

Introduction

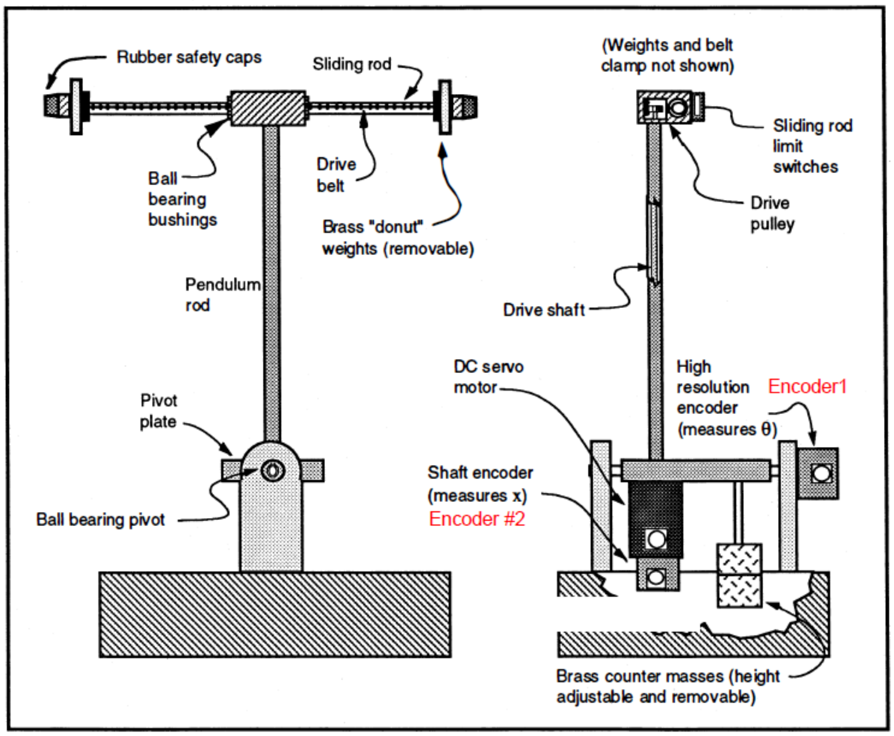

Figure 1: Inverted Pendulum Apparatus

What an inverted pendulum (Figure 1) does is it balances the pendulum rod with the sliding rod to keep it standing straight up. So a perfect inverted pendulum system would have the pendulum rod balance itself back to stand straight up in a vertical position after the sliding rod is pushed or pulled to the side. This inverted pendulum rod is very unstable because it has to compensate for the weight of the sliding rod, the mass of the weights, and gravity. Therefore a feedback controller is used to make the system stable.

This report contains the results of an experimentally created feedback controller. While creating the feedback system, the controllability, observability, and stability must be checked. The calculations done in MATLAB will then be used to simulate the inverted pendulum system in Simulink. A controller is created to improve the performance of the simulated system. A new system will then be designed using an observer. The conclusion will see if the new systems with a controller with or without an observer has a better performance.

1. Mathematical Model of the System

The system kinetic energy T(s) is:

$$\mathbf{T} = \frac{1}{2} m_1 v_{cg_1}^2 + \frac{1}{2} J_1 \dot{\theta}^2 + \frac{1}{2} m_2 v_{cg_2}^2 + \frac{1}{2} J_2 \dot{\theta}^2 \tag{1}$$

The variables v_{cg1} and v_{cg2} represent the inertial velocities of the respective member's center of gravity. While J_1 and J_2 represent the polar moments of inertia about the respective member's center of gravity. These are then constrained such that:

$$v_{cg_1} = \left| \underline{\dot{x}} + \underline{\dot{\theta}} \times \underline{l_{m1}} \right| \tag{2}$$

$$v_{cg_2} = l_c \dot{\theta}$$

We also have the equation:

$$v_{cg_1}^2 = \dot{x}^2 + \left(l_{m1} \dot{\theta}\right)^2 + 2\left(l_{m1} \dot{x} \dot{\theta} \cos\alpha\right) \tag{3}$$

Plugging in (2) into (1) we get:

$$\begin{aligned} \mathbf{T} &= \frac{1}{2}\left(J_1 + J_2 + m_1\left(l_o^2 + x^2\right) + m_2 l_c^2\right)\dot{\theta}^2 + \frac{1}{2} m_1 \dot{x}^2 + m_1 l_o \dot{x} \dot{\theta} \\ &= \frac{1}{2} J_o(x) \dot{\theta}^2 + \frac{1}{2} m_1 \dot{x}^2 + m_1 l_o \dot{x} \dot{\theta} \end{aligned} \tag{4}$$

The potential energy, V, is derived below in equation 5.

$$\begin{aligned} \mathbf{V} &= m_1 g l_{m1} \cos(\theta + \alpha) + m_2 g l_c \cos\theta \\ &= m_1 g l_{m1}\left(\cos\theta \cos\alpha - \sin\theta \sin\alpha\right) + m_2 g l_c \cos\theta \\ &= m_1 g\left(l_o \cos\theta - x \sin\theta\right) + m_2 g l_c \cos\theta \end{aligned} \tag{5}$$

There is also Lagrange's equation in the form:

$$\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{T}}{\partial \dot{q}_i}\right) - \frac{\partial \mathbf{T}}{\partial q_i} + \frac{\partial \mathbf{V}}{\partial q_i} = Q_i \qquad (6)$$

The Lagrange equation above is used to find the nonlinear equations of motion. Q_i is the generalized force while q_i is the generalized coordinate. For equation 7 below is formed from choosing q_1=x and q_2= θ.

$$\begin{aligned} &m_1\ddot{x}+m_1 l_o\ddot{\theta} - m_1 x\dot{\theta}^2 - m_1 g \sin\theta = F(t) \\ &m_1 l_o\ddot{x}+J_o\ddot{\theta}+2m_1 x\dot{x}\dot{\theta} - (m_1 l_0+m_2 l_c)g\sin\theta - m_1 g x\cos\theta = 0 \end{aligned} \qquad (7)$$

2. Linearized Model - Equilibrium Points

The equilibrium points $\theta = \theta_e$ and $x = x_e$ are found from the equations 6 and 7 above. When the F(t) is motionless then we get the values:

$$\theta_e = 0,\ x_e = 0 \qquad (8)$$

The linearized approximation of the system is derived from the Taylor's series expansion

$$\begin{aligned} &m_1\ddot{x}+m_1 l_o\ddot{\theta} - m_1 g\theta = F(t) \\ &m_1 l_o\ddot{x}+J_{o_e}\ddot{\theta} - (m_1 l_0+m_2 l_c)g\theta - m_1 g x = 0 \end{aligned} \qquad (9)$$

where J_{oe} equals $J_o\,|(x=x_e,\ \theta = \theta_e)$.

3. Linearized Model - State-Space

The state space equation is shown below:

$$\begin{aligned} \dot{x}&=Ax+Bu \\ y&=Cx+Du \end{aligned}$$

We have the values for the variables of the state space equation are shown in Table 1 below.

$$\boldsymbol{x} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ m_2 l_c g/J^* & 0 & m_1 g/J^* & 0 \\ 0 & 0 & 0 & 1 \\ (J^* - m_2 l_o l_c)\, g/J^* & 0 & -m_1 l_o g/J^* & 0 \end{bmatrix},$$

$$B = \frac{1}{J^*}\begin{bmatrix} 0 \\ -l_o \\ 0 \\ J_{oe}/m_1 \end{bmatrix}, \qquad C = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & C_4 \end{bmatrix}$$

Table 1: Given values for the inverted pendulum experiment.

Parameter	**Value**	**Description**
l_o	0.330 (m)	Length of pendulum rod from pivot to the sliding rod T section
m_1	to be determined (kg)	Mass of the complete sliding rod including all attached elements.
m_{1o}	0.103 (kg)	Mass of the sliding rod with belt, belt clamps, and rubber end guards
m_{w1}	0.110 (kg)	Combined mass of both of the sliding rod brass "donut" weights
m_2	to be determined (kg)	Mass of the complete assembly minus m1
m_{w2}	1.000 (kg)	Mass of brass balance weight
m_{2o}	0.785 (kg)	Mass of the complete moving assembly minus m1 and mw2
l_{co}	0.071 (m)	Position of c.g. of the complete pendulum assembly with the sliding rod and balance weight removed
J_o^*	0.0246 (kg-m2)	[Joe-m1lo2] evaluated when mw2=0.
t	0.016 (m)	Thickness of plate
l_t	0.07 (m)	or 7.0 cm, "Plant #2"
l_b	0.132 (m)	From bottom of weight to bottom

To be determined values are shown below including the equations of variables l_{w2}, l_c, J_{oe} and J^*.

l_{w2} = -(t+l_t+l_b)/2;
l_c = (m_{w2}*l_{w2}+m_{2o}*l_{co})/m_2;
J_{oe} = J_o *+(m_1*$(l_o)^2$)+(m_{w2}*$(l_{w2})^2$);
J^* = J_{oe}-(m_1*(lo)^2);

The equations above were coded into MATLAB. After substituting the values into the system, we then have our values for A, B, C, and D of our state space equations:

A =

0	1.0000	0	0
-14.3233	0	57.2772	0
0	0	0	1.0000
14.5367	0	-18.9015	0

B =

0
-9.0458
0
7.6800

C =

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

D =

0
0
0
0

4. Transfer Function

```
>> transferfunction
transferfunction =
  From input to output...
             -9.046 s^2 + 1.607e-14 s + 268.9
   1:  ---------------------------------------------------
     s^4 - 2.22e-15 s^3 + 33.22 s^2 - 5.684e-14 s - 561.9

                  -9.046 s^3 + 268.9 s
   2:  ---------------------------------------------------
     s^4 - 2.22e-15 s^3 + 33.22 s^2 - 5.684e-14 s - 561.9

                    7.68 s^2 - 21.49
   3:  ---------------------------------------------------
     s^4 - 2.22e-15 s^3 + 33.22 s^2 - 5.684e-14 s - 561.9

             7.68 s^3 - 3.411e-15 s^2 - 21.49 s
   4:  ---------------------------------------------------
     s^4 - 2.22e-15 s^3 + 33.22 s^2 - 5.684e-14 s - 561.9
```

5. Controllability and Observability

We check if the system is controllable by finding out if the controllability matrix is full row rank. Also, we check if the system is observable by finding out if the observability matrix is full column rank. From the results shown below, **the system is both controllable and observable**.

```
CM = ctrb(sys);
OM = obsv(sys);
rankCM = rank(CM);
rankOM = rank(OM);

>> CM
CM =
        0     -9.0458         0    569.4523
  -9.0458           0  569.4523          0
        0      7.6800         0   -276.6586
   7.6800           0 -276.6586          0

>> rankCM
rankCM =  4

>> OM
OM =
  1.0e+03 *
   0.0010        0         0         0
        0   0.0010         0         0
        0        0    0.0010         0
        0        0         0    0.0010
        0   0.0010         0         0
  -0.0143        0    0.0573         0
        0        0         0    0.0010
   0.0145        0   -0.0189         0
  -0.0143        0    0.0573         0
        0  -0.0143         0    0.0573
   0.0145        0   -0.0189         0
        0   0.0145         0   -0.0189
        0  -0.0143         0    0.0573
   1.0378        0   -1.9030         0
        0   0.0145         0   -0.0189
  -0.4830        0    1.1899         0

>> rankOM
rankOM =  4
```

6. Internal and External Stability

```
>> eigenvalues
eigenvalues =
  -0.0000 + 6.7497i
  -0.0000 - 6.7497i
  -3.5119 + 0.0000i
   3.5119 + 0.0000i
```

The eigenvalues of the system are not all in the left side of the s-plane. They are scattered like a cross shape. One eigenvalue is on the left side of the s-plane while another one is the same value but on the opposite side of the s-plane. The other two are on the imaginary plane one on the top and its opposite in the bottom. This means that the system is not BIBO stable. Also, since not all of the eigenvalues of A do not have negative real parts the system is not asymptotically stable. **We can then say the system is not internally stable and not externally stable.**

7. Minimal Realization

The system is already both controllable and observable. Therefore it **is already in its minimum realization form**. We can prove this by using the code below and as expected, the minimum realized system, sysr, is exactly the same as the original system.

```
>> sysr = minreal(sys)
sysr =
 a =
        x1      x2      x3      x4
  x1     0       1       0       0
  x2  -14.32     0     57.28     0
  x3     0       0       0       1
  x4   14.54     0     -18.9     0

 b =
        u1
  x1     0
  x2  -9.046
  x3     0
  x4   7.68

 c =
     x1  x2  x3  x4
  y1  1   0   0   0
  y2  0   1   0   0
  y3  0   0   1   0
  y4  0   0   0   1

 d =
     u1
  y1  0
  y2  0
  y3  0
  y4  0
```

8) Simulate and Plot the Open-Loop

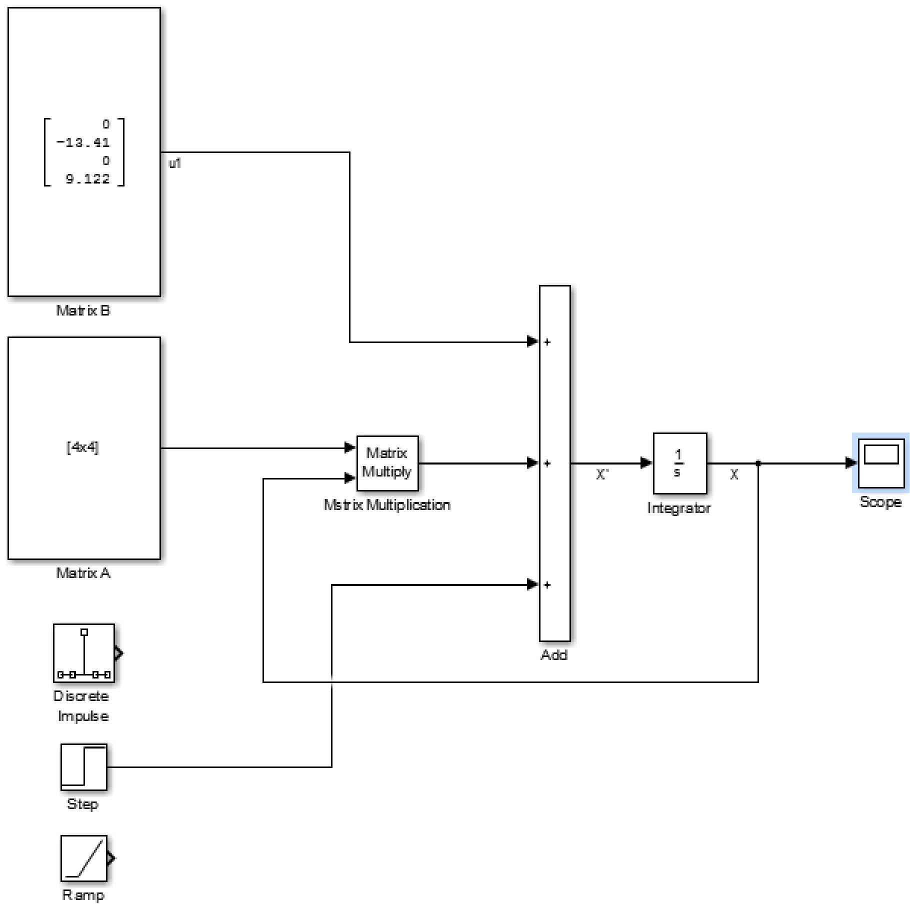

Figure 2: Open-loop Simulink simulation

The simulated Figures are shown below in Figures 3 and 4.

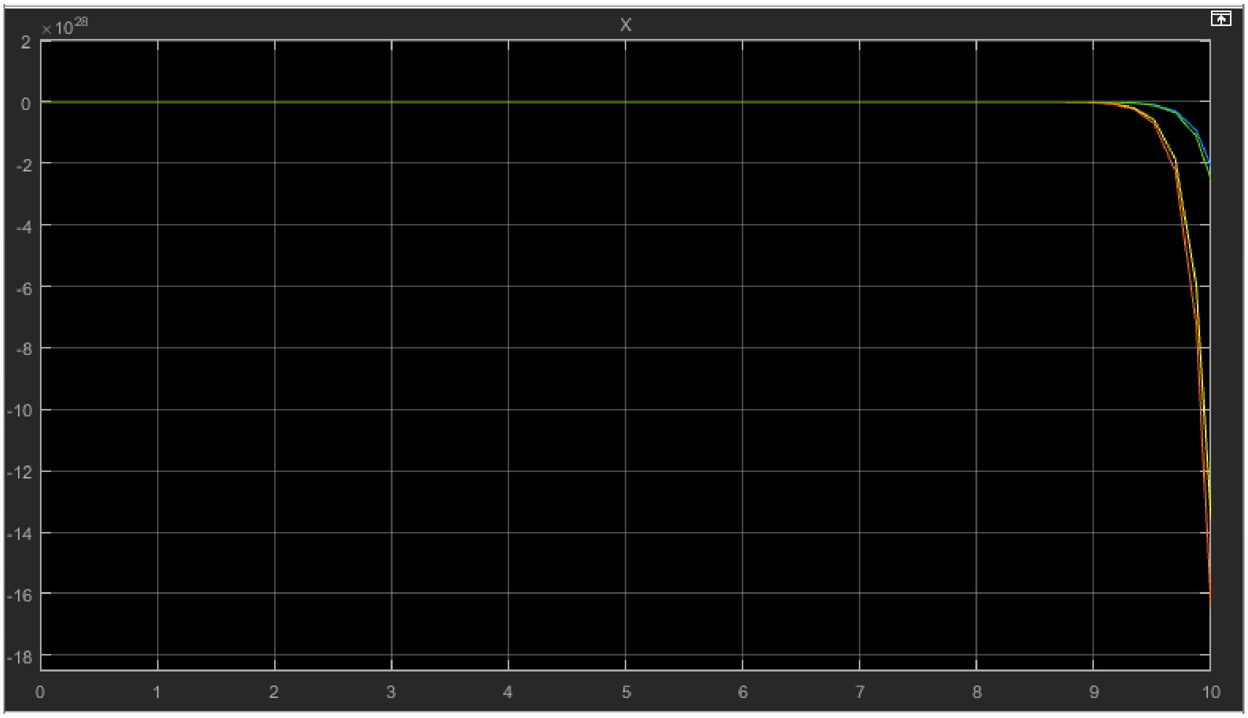

Figure 3: Open Loop step Response

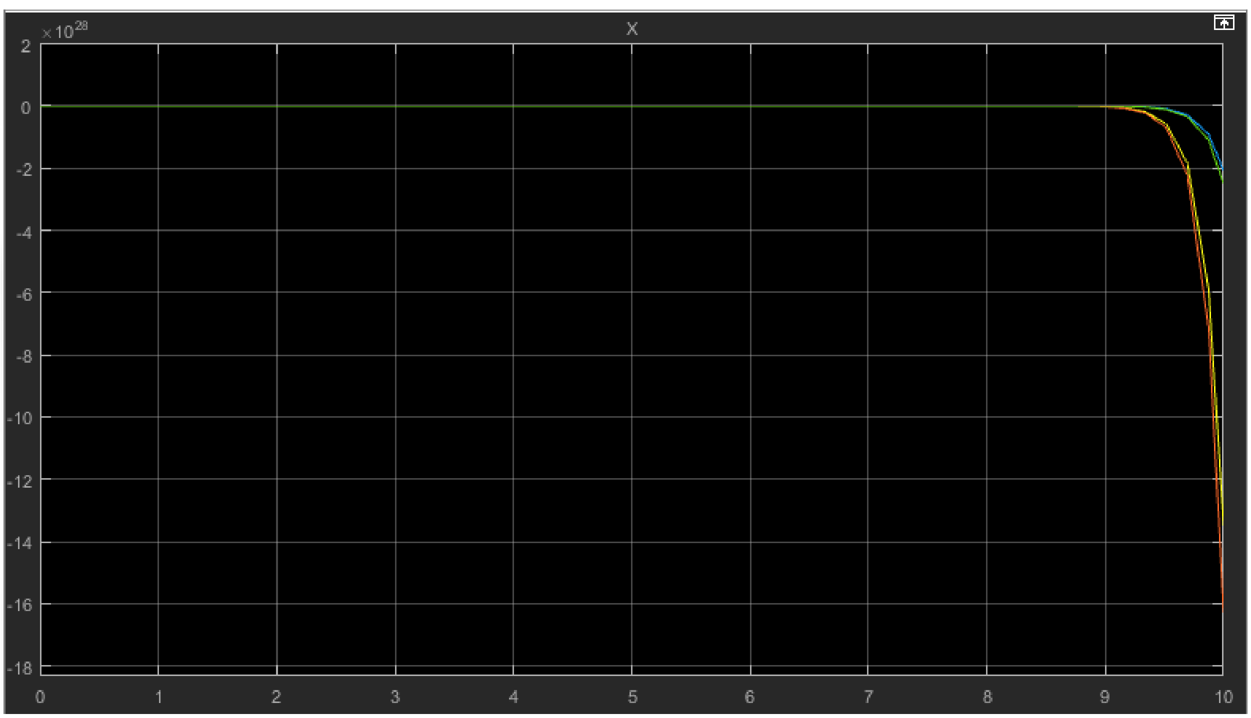

Figure 4: Open Loop Ramp Response

9) Desired Closed-Loop Poles That Improve the Transient and Steady State Response

We choose our poles to be

$$p = [-7+2i \quad -7-2i \quad -4+4i \quad -4-4i]$$

Using $acker(A, B, p)$ we get

$$k = [-0.245 \quad -1.6952 \quad -0.0274 \quad -0.0804]$$

We can calculate $\bar{A}$ using $\bar{A} = A - B * k$

$$\bar{A} = \begin{bmatrix} 0 & 50.54 & 0 & -6.8680 \\ -2.2849 & -22.7331 & -0.3668 & -1.0777 \\ 0 & 84.94 & 0 & -28.03 \\ 2.2345 & 15.4639 & 1.2495 & 0.7331 \end{bmatrix}$$

As we set $v = 1$ the K vector is the same as k. The simulation results as shown below. From the figure we can see that the system is stable after 2.5 seconds.

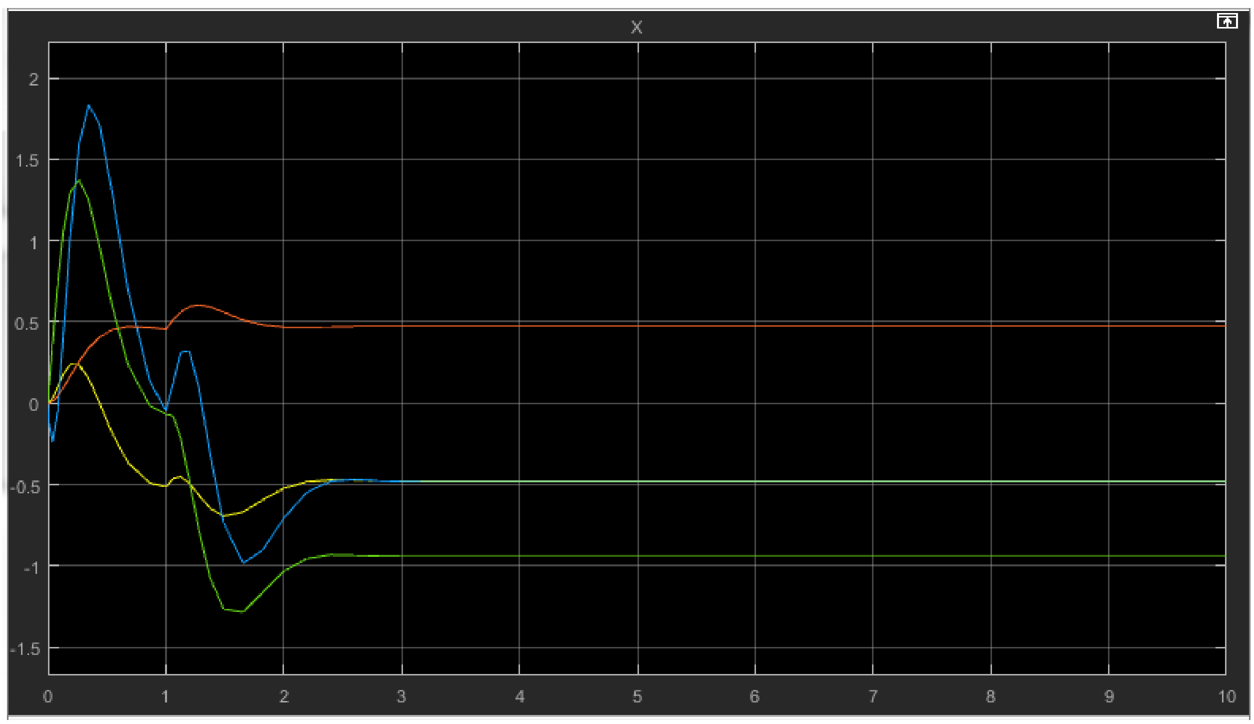

Figure 5: Closed Loop Response

10. Desired Observer Poles Faster Than the Closed-Loop

Desired Observer Poles:

$$Ob_{poles} = 3 * p$$

We find L using the Matlab Function $L = place(A, C, Ob_{poles})$

This result in

$$L = \begin{bmatrix} 21 & 56.54 & 0 & -6.8680 \\ -5 & 21 & 0 & 0 \\ 0 & 84.94 & 12 & -16.03 \\ 0 & 0 & -11 & 12 \end{bmatrix}$$

We can calculate $\bar{A}_2$ using $\overline{A}_2 = A - L * C$

$$\bar{A}_2 = \begin{bmatrix} -21 & -6 & 0 & 0 \\ 6 & -21 & 0 & 0 \\ 0 & 0 & -12 & -12 \\ 0 & 0 & 12 & -12 \end{bmatrix}$$

Simulation results are shown below and we see that the system with the observer becomes steady at roughly 1.3 seconds.

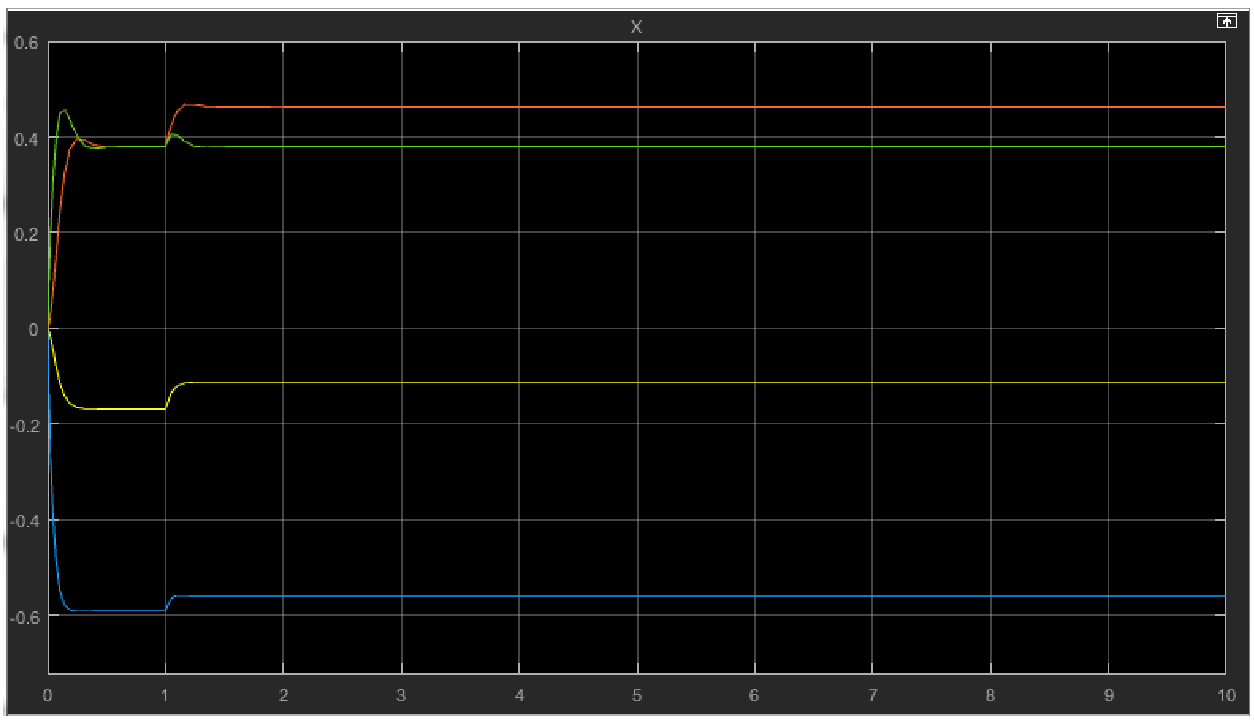

Figure 6: System with observer

11. Closed-loop System With Gain K From Step 9

Using the same method as in section 9 we get the p values shown below:

$$p = [-21 + 6i \quad -21 - 6i \quad -12 + 12i \quad -12 - 12i]$$

With these values we follow the same steps as before and get the simulation results as shown below.

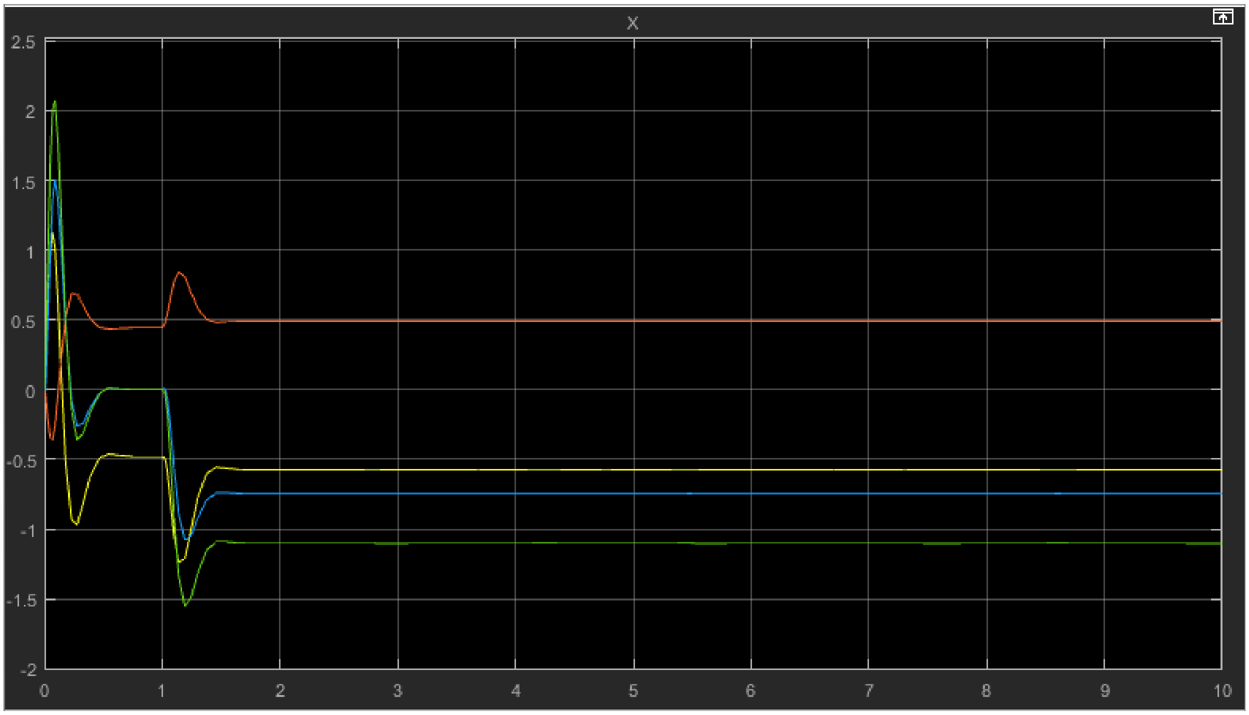

Figure 7: Closed Loop Response with 3*p

Comparing the results of the closed loop response with this figure we see that the system reaches stability faster than before but it has a much higher overshoot.

Conclusion

The inverted pendulum system was analyzed thoroughly. Following the derived mathematical model from the manual an inverted pendulum has been successfully created experimentally [1]. From the mathematical model we plugged in the values and created the linear state space equation of the system in MATLAB. Then simulated our experimental values for the controller and the observer. Our experimental desired closed-loop poles resulted in a steady state system of 2.5 seconds. However, the results of the closed-loop system using the states estimated by our own designed observer had a steady state of around 1.5 seconds. The closed-loop system designed by the observer was faster but it had higher overshoots. This small error can be due to the inverted pendulum system being unstable.

Reference:

[1] Educational Control Products, Manual For Model 505: Inverted Pendulum, Bell Canyon: ECP

Educational Control Products, 1994.

[2] Inverted Pendulum: State-Space Methods for Controller Design. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace

Appendix:

MATLAB Code

```
m1o = 0.103;
mw1 = 0.110;
m2o = 0.785;
mw2 = 1;
lo = 0.330;
lco = 0.071;
Jstaro = 0.0246;
g = 9.81;
t = 0.016; % thickness of plate
lt = 0.07; % or 7.0 cm, "Plant #2"
lb = 0.132; % bottom of weight to bottom

m1 = m1o+mw1;
m2 = m2o+mw2;


lw2 = -(t+lt+lb)/2;
lc = (mw2*lw2+m2o*lco)/m2;

Joe = Jstaro+(m1*(lo)^2)+(mw2*(lw2)^2);
Jstar = Joe-(m1*(lo)^2);


A = [0 1 0 0; (m2*lc*g)/Jstar 0 (m1*g)/Jstar 0; 0 0 0 1; (Jstar-
m2*lo*lc)*g/Jstar 0 -(m1*lo*g)/Jstar 0];

B = (1/Jstar)*[0;-lo;0;Joe/m1];

C = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];

D = 0;

sys = ss(A,B,C,D);
eigenvalues = eig(A); %poles
ev = eigenvalues;
transferfunction = tf(sys);
```

```
CM = ctrb(sys); %controller matrix
OM = obsv(sys); %observer matrix

rankCM = rank(CM); % rank of cm
rankOM = rank(OM); % rank of om

sysr = minreal(sys); %minimum realization

%{

%%lyapunov method***
F = [ -15  0 0 0; 0 -16 0 0 ; 0 0 -17 0; 0 0 0 -18];
G = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];

control_observer = ctrb(F,G);
newobsrank = rank(control_observer);

GC = G*C;
T = lyap(F,-A,GC);
H = T*B;

%}

p = [-7+2i -7-2j -4+4i -4-4i];
k = -acker(A,B,p);
Abar = A - B*k;
v = 1;

observerpole = 3*p;

L = place(A',C',observerpole)';
Abar2 = A - L*C;
```

Servo Code

```
; given values from manual [1]
Kf=0.0013
kx=50200
ka=2546
ks=32

; our own k values from experiment above
k1=-0.245
k3=-1.6952
k2=-0.0274
k4=-0.0804

past_effort = 0
control_effort=0
```

```
x1=0
x2=0
past_pos1=0
past_pos2=0

begin
x1=(enc1_pos)/(ks*ka)/kf
x2=(enc2_pos)/(ks*kx)/kf
past_pos1=x1
past_pos2=x2
control_effort=(-k1*x1-k2*x1-k3*x2-k4*x2)
past_effort=control_effort*kf
end
```

Acknowledgements

Steven Liu and Steven Luu did all of parts 8, 9, 10, and 11.

Lestley Gabo did all of parts 1-7 and the report.