ENPM 667: Control of Robotics Systems

Final Project

Sneha Nayak
(UID:115586284)
Arun Kumar Dhandayuthabani
(UID:116953924)

16 December 2019

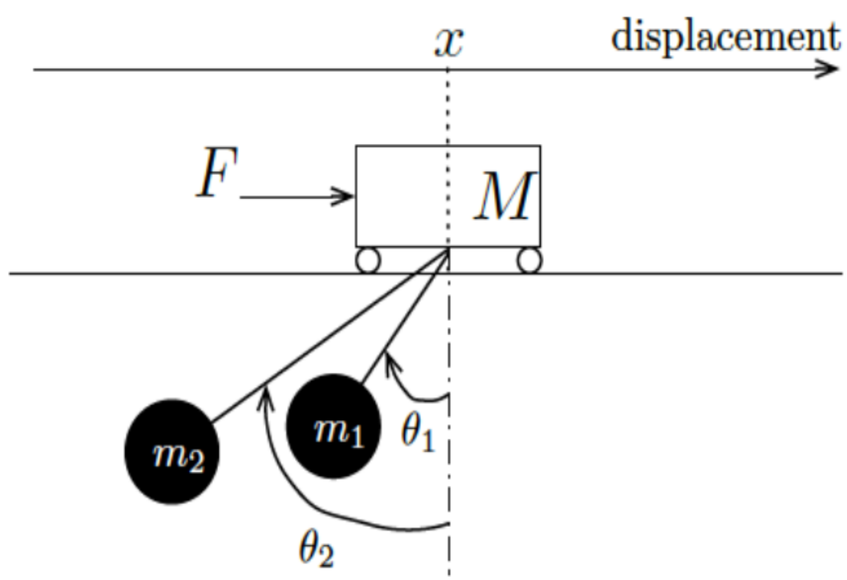
Content

Chapter 1

Problem Statement

Consider a crane that moves along a one-dimensional track. It behaves as a frictionless cart with mass M actuated by an external force F that constitutes the input of the system.

There are two loads suspended from cables attached to the crane. The loads have mass m1 and m2, and the lengths of the cables are l1 and l2, respectively. The following figure depicts the crane and associated variables used throughout this project.



Chapter 2

Component A

I. Equations of motion for the system and the corresponding nonlinear state-space representation.

Given below are the notations that we follow:

x : Position of displacement of the cart
x_1 : Position of Pendulum 1 in x- axis
x_2 : Position of Pendulum 2 in x- axis
y_1 : Position of Pendulum 1 in y- axis
y_2 : Position of Pendulum 2 in y- axis
l_1 : Length of Cable 1
l_2 : Length of Cable 2
m_1 : Mass of Pendulum 1
m_2 : Mass of Pendulum 2
M : Mass of Cart
F_{in} : Force applied on cart

$$x_1 = x - l_1 \sin\theta_1 \tag{1}$$
$$y_1 = l_1 \cos\theta_1 \tag{2}$$

$$x_2 = x - l_2 \sin\theta_2 \tag{3}$$
$$y_2 = l_2 \cos\theta_2 \tag{4}$$

Obtaining velocities with by differentiating (1), (2), (3) and (4) with respect to time,

$$\dot{x_1} = \dot{x} - l_1 \cos\theta_1 \dot{\theta_1} \tag{5}$$
$$\dot{y_1} = -l_1 \sin\theta_1 \dot{\theta_1} \tag{6}$$

$$\dot{x_2} = \dot{x} - l_2 \cos\theta_2 \dot{\theta}_2 \tag{7}$$
$$\dot{y_2} = -l_2 \sin\theta_2 \dot{\theta}_2 \tag{8}$$

Now we know that Lagrangian L = Kinetic Energy (KE) – Potential Energy (KE).

Therefore,

$$KE = \frac{1}{2} M\dot{x}^2 + \frac{1}{2} m_1(\dot{x} - l_1 \cos\theta_1 \dot{\theta}_1)^2 + \frac{1}{2} m_1(-l_1 \sin\theta_1 \dot{\theta}_1)^2 + \frac{1}{2} m_2(\dot{x} - l_2 \cos\theta_2 \dot{\theta}_2)^2 + \frac{1}{2} m_2(-l_2 \sin\theta_2 \dot{\theta}_2)^2$$

Simplifying the above,

$$KE = \frac{1}{2} (M + m_1 + m_2)\dot{x}^2 + \frac{1}{2} m_1\left(l_1\dot{\theta}_1\right)^2 + \frac{1}{2} m_2\left(l_2\dot{\theta}_2\right)^2 - \frac{1}{2} (m_1 l_1 \cos\theta_1 \dot{\theta}_1 + m_2 l_2 \cos\theta_2 \dot{\theta}_2)^2 \tag{9}$$

$$PE = -m_1 g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2 \tag{10}$$

$$L = \frac{1}{2} (M + m_1 + m_2)\dot{x}^2 + \frac{1}{2} m_1\left(l_1\dot{\theta}_1\right)^2 + \frac{1}{2} m_2\left(l_2\dot{\theta}_2\right)^2 - \frac{1}{2} (m_1 l_1 \cos\theta_1 \dot{\theta}_1 + m_2 l_2 \cos\theta_2 \dot{\theta}_2)^2 + (m_1 g l_1 \cos\theta_1 + m_2 g l_2 \cos\theta_2) \tag{11}$$

The equations of motion are given by,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = F \tag{12}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) - \left(\frac{\partial L}{\partial \theta_1}\right) = 0 \tag{13}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) - \left(\frac{\partial L}{\partial \theta_2}\right) = 0 \tag{14}$$

Partially differentiating L w.r.t $\dot{x}$ and then differentiating it with respect to time,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = \dot{x}\,(M + m_1 + m_2) - m_1 l_1\left(\cos\theta_1 \ddot{\theta_1} - \sin\theta_1 \dot{\theta_1}^2\right) - m_2 l_2(\cos\theta_2 \ddot{\theta_2} - \sin\theta_2 \dot{\theta_2}^2) \tag{15}$$

Partially differentiating L w.r.t $\dot{\theta}_1$ and then differentiating it with respect to time,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) = m_1 {l_1}^2 - m_1 l_1\left(\ddot{x}\cos\theta_1 - \sin\theta_1 \dot{\theta}_1\,\dot{x}\,\right) \tag{16}$$

Partially differentiating L w.r.t θ_1,

$$\frac{\partial L}{\partial \theta_1} = m_1 l_1 g \sin\theta_1 + m_1 l_1 \dot{x} \sin\theta_1 \dot{\theta}_1 \tag{17}$$

Partially differentiating L w.r.t $\dot{\theta_2}$ and then differentiating it with respect to time,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) = m_2 {l_2}^2 - m_2 l_2 (\ddot{x} \cos\theta_2 - \sin\theta_2 \dot{\theta_2} \dot{x}) \tag{18}$$

Partially differentiating L w.r.t θ_2,

$$\frac{\partial L}{\partial \theta_2} = m_2 l_2 g \sin\theta_2 + m_2 l_1 \dot{x} \sin\theta_2 \dot{\theta_2} \tag{19}$$

Now substituting (15) in (12) and simplifying, we get,

$$\ddot{x} = \frac{F + m_1 l_1 \left(\ddot{\theta_1} \cos\theta_1 - \sin\theta_1 {\dot{\theta_1}}^2\right) + m_2 l_2 \left(\ddot{\theta_2} \cos\theta_2 - \sin\theta_2 {\dot{\theta_2}}^2\right)}{M + m_1 + m_2} \tag{20}$$

Now substituting (16) and (17) in (13) and simplifying, we get,

$$\ddot{\theta_1} = \frac{\ddot{x} \cos\theta_1 - g \sin\theta_1}{l_1} \tag{21}$$

Now substituting (18) and (19) in (14), we get,

$$\ddot{\theta_2} = \frac{\ddot{x} \cos\theta_2 - g \sin\theta_2}{l_2} \tag{22}$$

Substituting (21) and (22) in (20) and simplifying, we get,

$$\ddot{x} = \frac{F - g(m_1 l_1 \sin\theta_1 \cos\theta_1 - m_2 l_2 \sin\theta_2 \cos\theta_2) - m_1 l_1 \sin\theta_1 {\dot{\theta_1}}^2 - m_2 l_2 \sin\theta_2 {\dot{\theta_2}}^2}{M + m_1 \sin{\theta_1}^2 + m_2 \sin{\theta_2}^2} \tag{23}$$

Writing the above equations (21, 22, 23) in non-linear state space form,

$$X = \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta_1} \\ \theta_2 \\ \dot{\theta_2} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

Now representing $\dot{X}$ of the form $\dot{X}(t) = F(X(\text{t}), \text{U(t)})$,

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} x_2 \\ \dfrac{F - g(m_1 l_1 \sin\theta_1 \cos\theta_1 - m_2 l_2 \sin\theta_2 \cos\theta_2) - m_1 l_1 \sin\theta_1 \dot{\theta}_1^{\,2} - m_2 l_2 \sin\theta_2 \dot{\theta}_2^{\,2}}{M + m_1 \sin{\theta_1}^2 + m_2 \sin{\theta_2}^2} \\ x_4 \\ \dfrac{\ddot{x}\cos\theta_1 - g\sin\theta_1}{l_1} \\ x_6 \\ \dfrac{\ddot{x}\cos\theta_2 - g\sin\theta_2}{l_2} \end{bmatrix}$$

And representing Y of the form $Y(t) = H(X(\mathrm{t}), \mathrm{U(t)})$,

$$Y = CX$$

C can be represented as,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

II. Linearization.

Here we are required to linearize the system around the equilibrium point specified by x = 0 and θ_1= θ_2 = 0 and write state-space representation of the linearized system. We can do this by computing the Jacobian of the F in order to find A_F, B_F, C_F, D_F which are of the form,

$$A_F = \nabla_x | F(X, \mathrm{U}), \qquad B_F = \nabla_u | F(X, \mathrm{U})$$
$$C_H = \nabla_x | H(X, \mathrm{U}), \qquad D_H = \nabla_u | H(X, \mathrm{U})$$

The code to linearize the above system is in the file linearization.m.

```
clc;
clear all;
syms F M m1 m2 l1 l2 x1 x2 x3 x4 x5 x6 g real

%Representing the non-linear state-space equations
D=(M+(m1*((sin(x3))^2))+(m2*((sin(x5))^2)));
x_d = ((F - g*((m1*sin(x3)*cos(x3))- (m2*sin(x5)*cos(x5)))...
- (m1*l1*sin(x3)*(x4^2))- (m2*l2*sin(x5)*(x6^2)))/D);
theta1_double_dot = (((x_d * cos(x3))-g*(sin(x3)))/l1);
theta2_double_dot = (((x_d * cos(x5))-g*(sin(x5)))/l2);
f1 = x2;
f2 = x_d;
f3 = x4;
f4 = theta1_double_dot;
f5 = x2;
f6 = theta2_double_dot;

%Find Jacobian of Matrix A
jacob_A = [ diff(f1,x1) diff(f1,x2) diff(f1,x3) diff(f1,x4) diff(f1,x5) diff(f1,x6);
diff(f2,x1) diff(f2,x2) diff(f2,x3) diff(f2,x4) diff(f2,x5) diff(f2,x6);
diff(f3,x1) diff(f3,x2) diff(f3,x3) diff(f3,x4) diff(f3,x5) diff(f3,x6);
diff(f4,x1) diff(f4,x2) diff(f4,x3) diff(f4,x4) diff(f4,x5) diff(f4,x6);
diff(f5,x1) diff(f5,x2) diff(f5,x3) diff(f5,x4) diff(f5,x5) diff(f5,x6);
diff(f6,x1) diff(f6,x2) diff(f6,x3) diff(f6,x4) diff(f6,x5) diff(f6,x6) ];

%Find Jacobian of Matrix B w.r.t Input i.e. F
jacob_B = [ diff(f1,F); diff(f2,F); diff(f3,F); diff(f4,F); diff(f5,F); diff(f6,F)];

%Substituting values around equilibrium point for A
jacobian_A =subs(jacob_A, [x1 x2 x3 x4 x5 x6],[ 0 0 0 0 0 0]);
%Substituting values around equilibrium point for B
jacobian_B = subs(jacob_B, [x1 x2 x3 x4 x5 x6],[ 0 0 0 0 0 0]);
```

We find the Jacobian by finding the partial differentiation of each term of F with respect to its corresponding x (in the case of A_F) or u values (in the case of B_F). The code generates A_F and B_F which are then found to be,

$$A_F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m_1 g}{M} & 0 & \frac{-m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-(m_1 g + Mg)}{Ml_1} & 0 & \frac{-m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{Ml_2} & 0 & \frac{-(m_2 g + Mg)}{Ml_2} & 0 \end{bmatrix}, \quad B_F = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix}$$

$$C_H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_H = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Therefore, the linearized State Space Equation, which is of the form

$$\dot{X} = A_F\, X + \, B_F\, U$$
$$Y = C_H\, X + \, D_H\, U$$

Where $U = F_{in}$ here, is as follows,

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m_1 g}{M} & 0 & \frac{-m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-(m_1 g + Mg)}{Ml_1} & 0 & \frac{-m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{Ml_2} & 0 & \frac{-(m_2 g + Mg)}{Ml_2} & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} F_{in}$$

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} F_{in}$$

III. Controllability

This state space model that we obtained above is a Linear Time In-varying System, since we linearized the system about its equilibrium point and therefore, we can use the below condition to check if a system is controllable or not:

$$rank\left[B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B \quad \dots A^{n-1}B\right] = n$$

In our case since dimensionality of A is equal to 6,

$$rank\left[B_F \quad A_FB_F \quad {A_F}^2B_F \quad {A_F}^3B_F \quad {A_F}^4B_F \quad {A_F}^5B_F\right] = 6$$

We can check this using our code in controllability.m

```
clc;
clear all;
syms m1 m2 M l1 l2 g
A= [0 1 0 0 0 0,
0   0 -m1*g/M 0 -m2*g/M 0,
0   0 0 1 0 0,
0   0 (-g*(m1+M))/(M*l1) 0 (-m2*g)/(M*l1) 0,
0   0 0 0 0 1,
0   0 (-m1*g)/(M*l2) 0 (-g*(m2+M))/(M*l2) 0];

B= [0 1/M 0 1/(M*l1) 0 1/(M*l2)]';

C = [B A*B (A*A)*B (A*A*A)*B (A*A*A*A)*B (A*A*A*A*A)*B];
det(C);
rankC = rank(C);

% m1 = m2
c_1 = subs(C,m1,m2);
rankC_m = rank(c_1);
if rank(c_1) == rank(C)
    fprintf('case 01: m1 = m2 -- Controllable System\n\n');
else
    fprintf('case 01: m1 = m2 -- UnControllable System\n\n')
end

% l1 = l2
c_2 = subs(C,l1,l2);
rankC_l = rank(c_2);
if rank(c_2) == rank(C)
fprintf('Case 2: l1 = l2 -- Controllable System\n\n');
else
fprintf('Case 2: l1 = l2 -- Uncontrollable System\n\n')
end
```

```
% ml = M
c_3 = subs(C,m1,M);

rankC_M = rank(c_3);
if rank(c_3) == rank(C)
fprintf('case 3: ml = M -- Controlable system\n\n' );
else
fprintf('case 3: ml = M -- Uncontrolable system\n\n');
end

%case 04: m2 = M
c_4 = subs(C,m2,M);
rankC_M2 = rank(c_4);
if rank(c_4) == rank(C)
fprintf('case 4: m2 = M -- Controlable system\n\n');
else
fprintf('case 4: m2 = M -- Uncontrolable system\n\n' );
end
```

Here we check the controllability following cases:

Case 1: $m_1 = m_2$
Case 2: $l_1 = l_2$
Case 3: $m_1 = M$
Case 4: $m_2 = M$

Output

```
case 01: ml = m2 -- Controllable System

Case 2: l1 = l2 -- Uncontrollable System

case 3: ml = M -- Controlable system

case 4: m2 = M -- Controlable system
```

We can look at the above results and clearly state that the system is not controllable in Case 2, when $l_1 = l_2$. All other conditions give out a full rank of 6 while Case 2 gives out a rank 4, which means that mass of the cart, mass of the pendulum 1 and mass of pendulum 2 can all be equal to each other, but the length of cable of Pendulum 1 cannot be equal to the length of cable of Pendulum 2.

IV. LQR Controller when M = 1000Kg, $m_1 = m_2$ = 100Kg, l_1 = 20m and l_2 = 10m

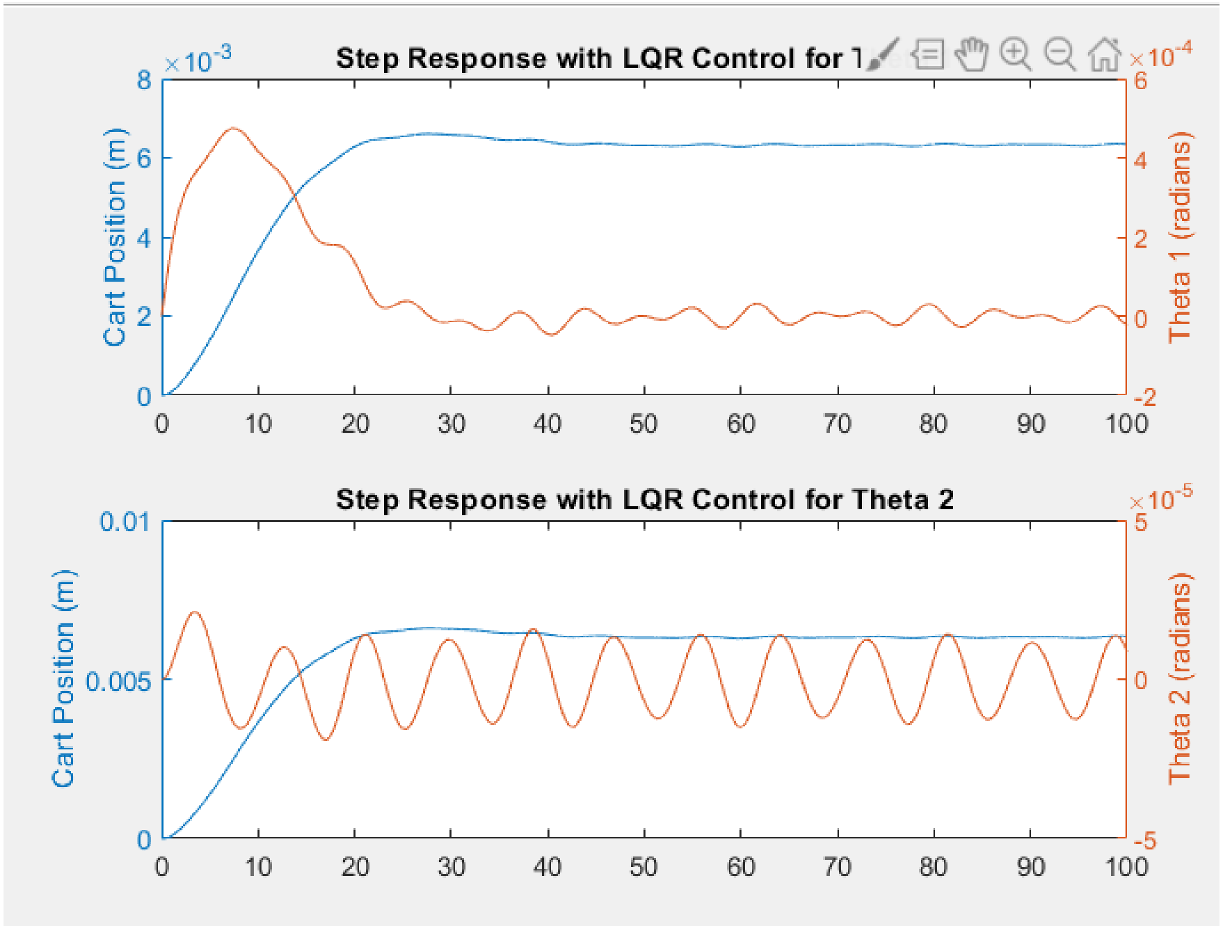
- Linearized System – Code for this is in lqr_lin.m

```
clc;
clear all;
M = 1000;
m1  =100;
m2  = 100;
l1 = 20;
l2 = 10;
g = 9.81;

A=  [0  1 0 0 0 0,
    0 0 -m1*g/M 0 -m2*g/M 0,
    0 0 0 1 0 0,
    0 0 (-g*(m1+M))/(M*l1)  0   (-m2*g)/(M*l1)  0,
    0 0 0 0 0 1,
    0 0 (-m1*g)/(M*l2)  0   (-g*(m2+M))/(M*l2)  0];
B=  [0  1/M 0   1/(M*l1)    0   1/(M*l2)]';
C=  [1  0 0 0 0 0,
    0 1 0 0 0 0,
    0 0 1 0 0 0,
    0 0 0 1 0 0,
    0 0 0 0 1 0,
    0   0   0   0 0 1] ;
D=  [0  0   0   0   0   0]';

% LQR Controller
Q = C'*C;
R = 0.001;
K = lqr(A,B,Q,R);
plotting(A,B,C,D,K);

%For different Value of Q.
Q= 10*[10 3 2 5 2 3,
    3 100 5 2 2 3,
    2 4 5 2 2 4,
    5 7 6 5 6 6,
    2 5 2 5 100 6,
    3 7 3 5 4 10];
Q=(Q+Q')/2;
K = lqr(A,B,Q,R);
plotting(A,B,C,D,K);
```

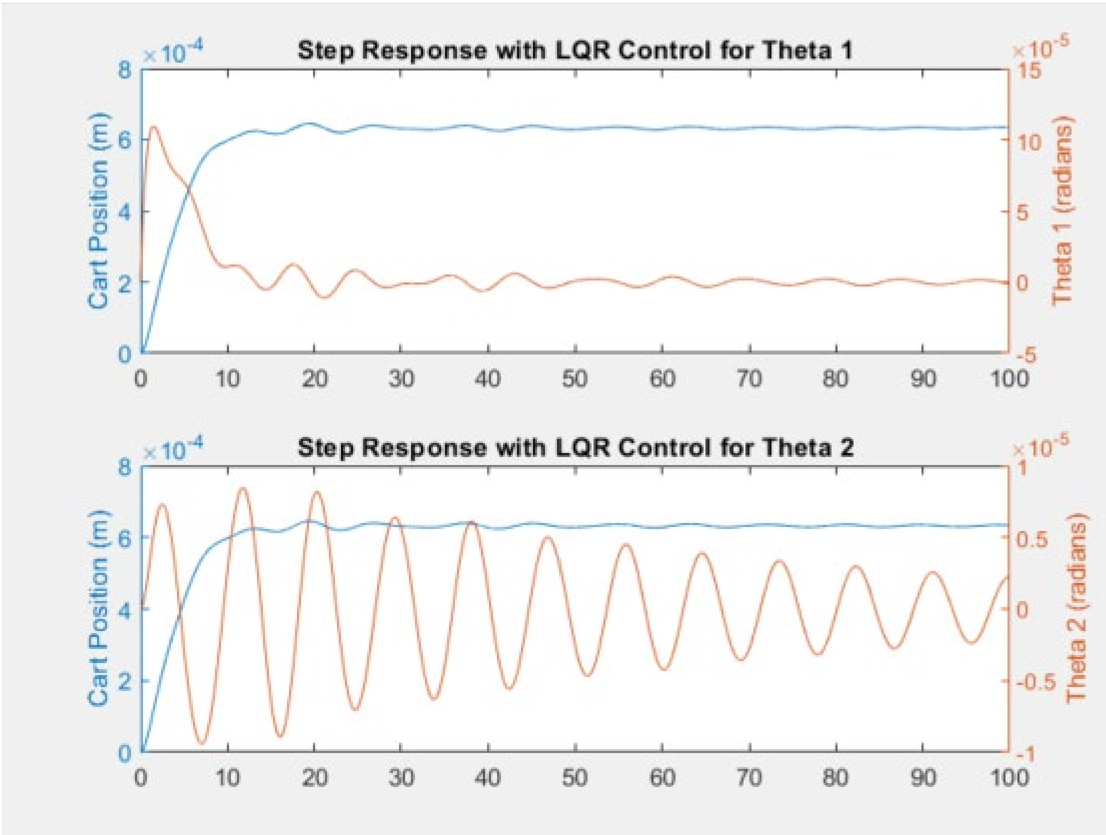
```
% Lyapunovs indirect method
vals = eig(A-B*K);
if (vals<0)
     fprintf('All Eigen Values in the left half plane and System is stable')
else
    fprintf('System Unstable')
end


function plotting(A,B,C,D,K)
[y,t]=stepSim(A,B,C,D,K);

% Closed Loop System Plot
figure;
subplot(2,1,1)
[AX] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','Cart Position (m)')
set(get(AX(2),'Ylabel'),'String','Theta 1 (radians)')
title('Step Response with LQR Control for Theta 1')
subplot(2,1,2)

[AX] = plotyy(t,y(:,1),t,y(:,3),'plot');
set(get(AX(1),'Ylabel'),'String','Cart Position (m)')
set(get(AX(2),'Ylabel'),'String','Theta 2 (radians)')
title('Step Response with LQR Control for Theta 2')
end

function [y,t] = stepSim(A,B,C,D,K)
Ac = A-B*K;

%Closed loop System
stateSpaceModelClosedLoop = ss(Ac,B,C,D);
t = 0:0.01:100;
r = 0.2*ones(size(t));
[y,t]=lsim(stateSpaceModelClosedLoop,r,t);
end
```

The outputs are as follows for Different values of Q:

Case 1: C'*C



Case 2: Q = (Q+Q')/2



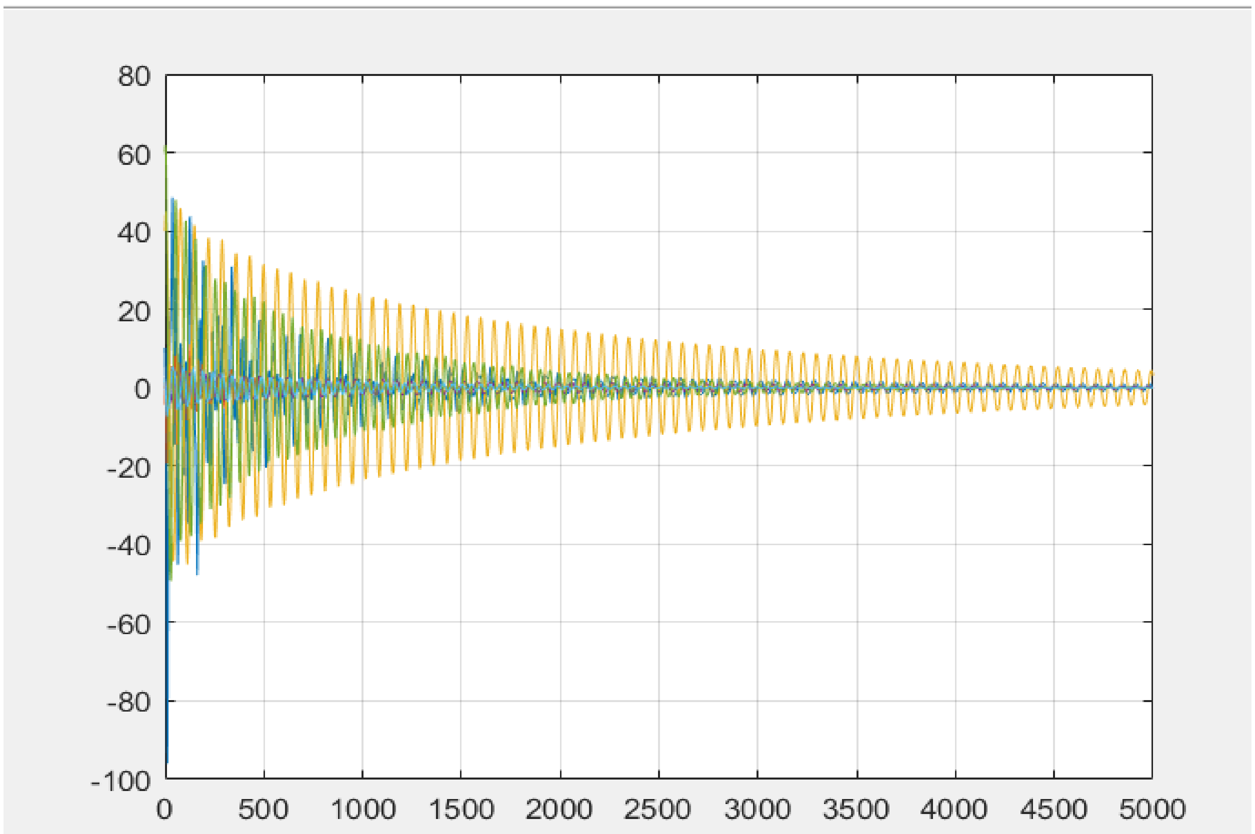
Step Response with LQR Control for Theta 1
$\times 10^{-4}$
Cart Position (m)
$\times 10^{-5}$
Theta 1 (radians)
Step Response with LQR Control for Theta 2
$\times 10^{-4}$
Cart Position (m)
$\times 10^{-5}$
Theta 2 (radians)

- Non-Linearized System– Code for this is in lqr_nonlin.m

```
clc;
clear all;
%output variable
y0 = [10; 2; 40; 2; 60; 2]
tspan = 0:0.01:5000;%timespan

[t1,y1] = ode45(@nonlinear,tspan,y0); %use ode45 function

plot(t1,y1)%plot
grid on


function dydt = nonlinear(t,x)

M = 1000;
m1  =100;
m2  = 100;
l1 = 20;
l2 = 10;
g = 9.81;
dydt =zeros(6,1);
A=  [0  1 0 0 0 0,
    0 0 -m1*g/M 0 -m2*g/M 0,
    0 0 0 1 0 0,
    0 0 (-g*(m1+M))/(M*l1)  0   (-m2*g)/(M*l1)  0,
    0 0 0 0 0 1,
    0 0 (-m1*g)/(M*l2)  0   (-g*(m2+M))/(M*l2)  0];
B=  [0  1/M 0   1/(M*l1)    0   1/(M*l2)]';
C=  [1  0 0 0 0 0,
    0 1 0 0 0 0,
    0 0 1 0 0 0,
    0 0 0 1 0 0,
    0 0 0 0 1 0,
    0   0   0   0 0 1] ;
```

```
% LQR Controller
Q = 100*(C'*C);
R = 0.001;
[K,~,~] = lqr(A,B,Q,R);
F=-K*x;


D=(M+(m1*((sind(x(3)))^2))+(m2*((sind(x(5)))^2)));
x_d = ((F - g*((m1*sind(x(3))*cosd(x(3)))- (m2*sind(x(5))*cosd(x(5))))...
- (m1*l1*sind(x(3))*((x(4))^2))- (m2*l2*sind(x(5))*((x(6))^2)))/D);
theta1_double_dot = (((x_d * cosd(x(3)))-g*(sind(x(3))))/l1);
theta2_double_dot = (((x_d * cosd(x(5)))-g*(sind(x(5))))/l2);

dydt (1) = x(2);
dydt (2)= x_d;
dydt (3)= x(4);
dydt (4)= theta1_double_dot;
dydt (5)= x(6);
dydt (6)= theta2_double_dot;
end
```

Output



Chapter 3

Component B

V. Observability

For a system to be observable, the observability matrix should satisfy the following:

rank = $[C^T \quad A^T C^T \quad (A^T)^2 C^T \ \ldots\ldots\ (A^T)^{n-1} C^T]$ = n

rank = $[C^T \quad A^T C^T \quad (A^T)^2 C^T \quad (A^T)^3 C^T \ \ldots\ldots (A^T)^5 C^T] = 6$

Here the matrices are:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m_1 g}{M} & 0 & \frac{-m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-(m_1 g + Mg)}{Ml_1} & 0 & \frac{-m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{Ml_2} & 0 & \frac{-(m_2 g + Mg)}{Ml_2} & 0 \end{bmatrix}$$

We find 4 cases of C. Choosing one C at a time and checking the rank:

```
syms M m1 m2 g l1 l2
A=[0 1 0 0 0 0;0 0 -m1*g/M 0 -m2*g/M 0;0 0 0 1 0 0;0 0 -(M*g-m1*g)/M*l1 0 -m2*g/M*l1 0;0 0 0
0 0 1;0 0 -m1*g/M*l2 0 -(M*g-m2*g)/M*l2 0];
D=transpose(A);
%C=[1 ;0 ;0 ;0 ;0 ;0 ];%for x(t)
%C=[0 0;0 0;1 0;0 0;0 1;0 0];%for theta1(t),theta2(t)
%C=[1 0;0 0;0 0;0 0;0 1;0 0];%for x(t),theta2(t)
C=[1 0 0;0 0 0;0 1 0;0 0 0;0 0 1;0 0 0];%for x(t),theta1(t),theta2(t)
K=[C D*C D^2*C D^3*C D^4*C D^5*C];
E=rank(K);
disp(K)
disp(E)
```

- Code for this is in obs_code_proj.m

For x(t), rank = 6 [observable].
For (theta1(t); theta2(t)), rank = 4 [not observable]
For (x(t); theta2(t)), rank = 6 [observable.]
For (x(t); theta1(t); theta2(t)), rank = 6 [observable.]

VI. Luenberger Observer

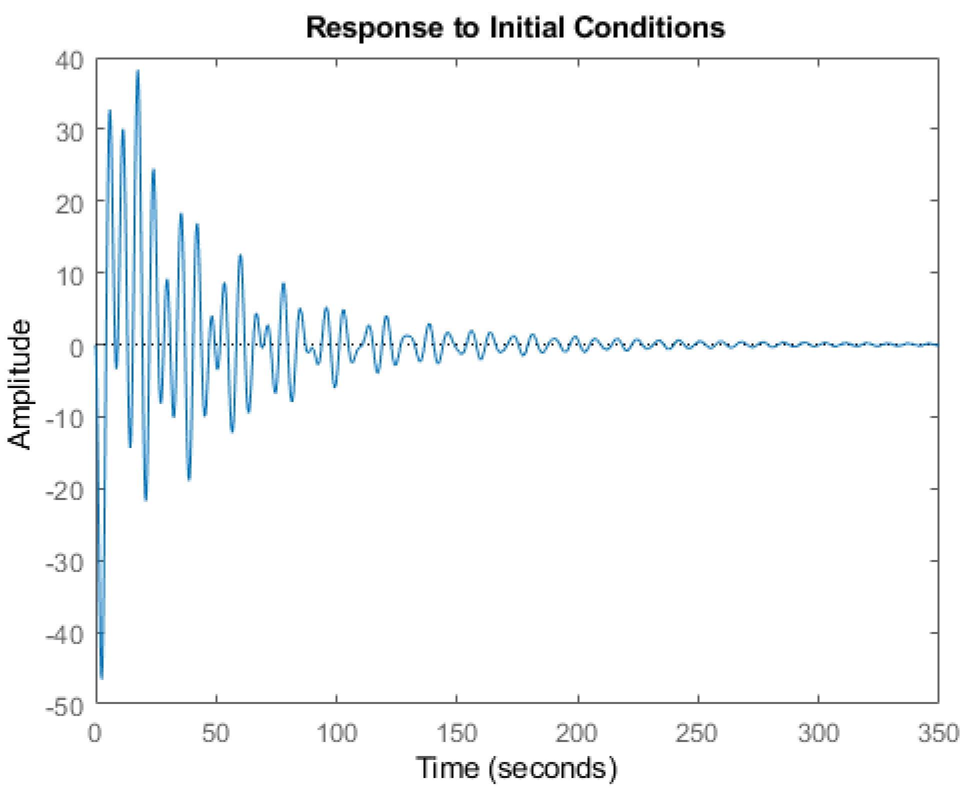
- Code for this is in luenberger_obs.m

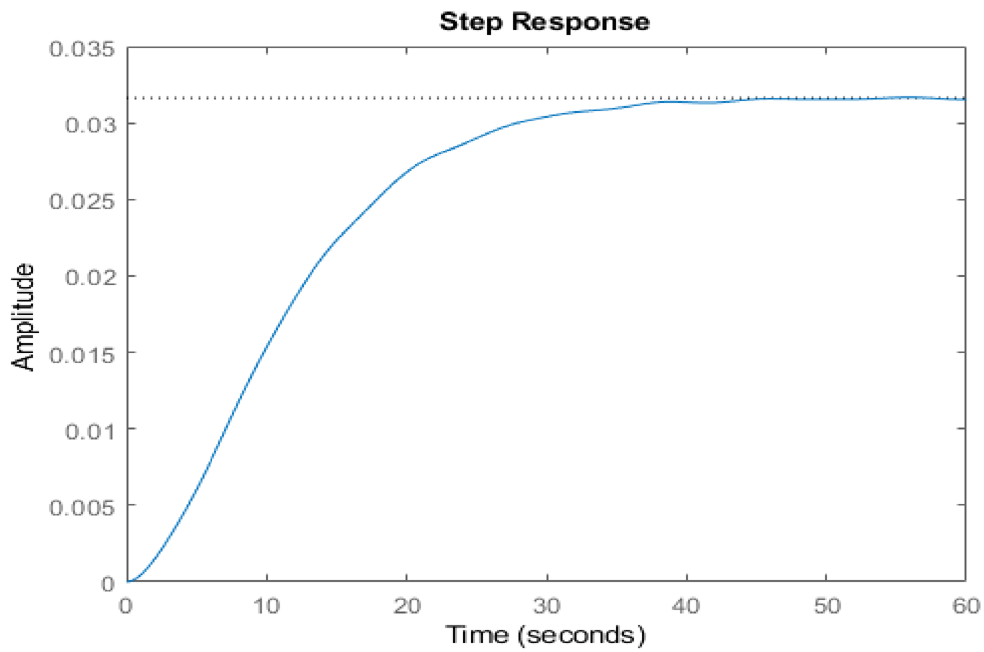
```
syms m1 m2 l1 l2 g M

m1 = 100;
m2 = 100;
M = 1000;
l1 = 20;
l2 = 10;
g = 9.8;

poles=[-1;-2;-3;-4;-5;-6];

%Initial Conditions
x0=[0,0,10,0,30,0,0,0,0,0,0,0];

%Matrices
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
Q=[10 0 0 0 0 0;
```

```
   0 700 0 0 0 0;
   0 0 1000 0 0 0;
   0 0 0 800 0 0;
   0 0 0 0 900 0;
   0 0 0 0 0 1000];
R=0.01;
%  C1, C3 and C4 are observable
C1 = [1 0 0 0 0 0];  %Corresponding to x(t)
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %corresponding to x(t) and theata2(t)
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %corresponding to x(t), theta1(t) and theata2(t)
D = 0;

K=lqr(A,B,Q,R);
L1 = place(A',C1',poles)'
L3 = place(A',C3',poles)'
L4 = place(A',C4',poles)'

Aq1 = [(A-B*K) B*K;
        zeros(size(A)) (A-L1*C1)];
Bq = [B;zeros(size(B))];
Cq1 = [C1 zeros(size(C1))];

Aq3 = [(A-B*K) B*K;
        zeros(size(A)) (A-L3*C3)];
Cq3 = [C3 zeros(size(C3))];

Aq4 = [(A-B*K) B*K;
        zeros(size(A)) (A-L4*C4)];
Cq4 = [C4 zeros(size(C4))];

sys1 = ss(Aq1, Bq, Cq1, D);%x(t)
figure
initial(sys1,x0)
figure
step(sys1)
sys3 = ss(Aq3, Bq, Cq3, D);%x(t) and theta1(t)
figure
initial(sys3,x0)
figure
step(sys3)
sys4 = ss(Aq4, Bq, Cq4, D);%x(t),theta1(t) and theta2(t)
figure
initial(sys4,x0)
figure
step(sys4)

grid on
```

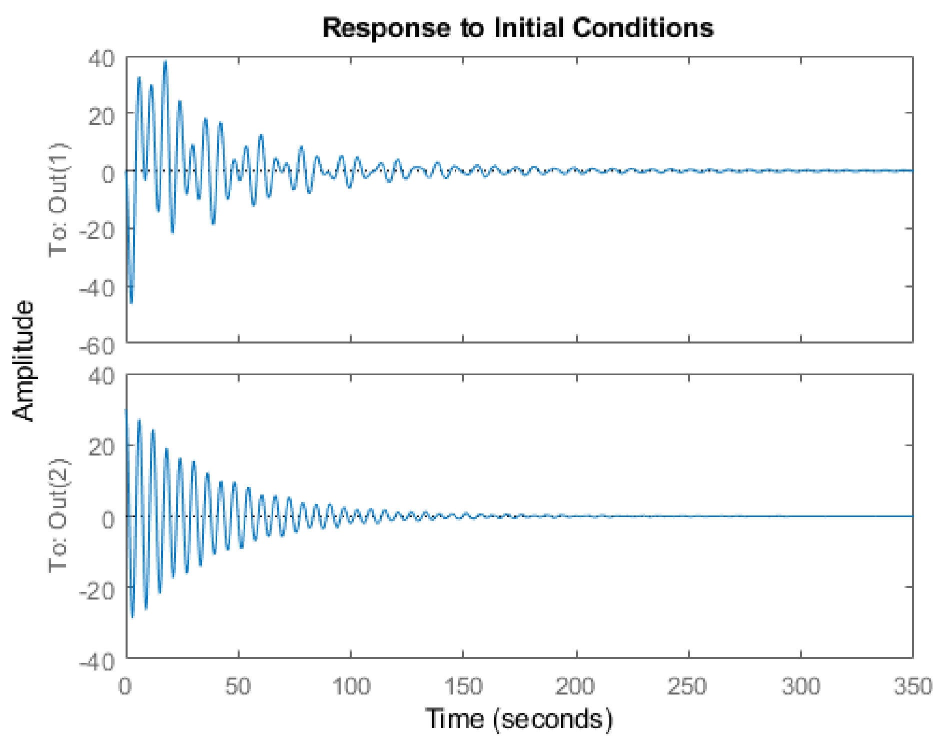
```
L1 =

   1.0e+03 *
```

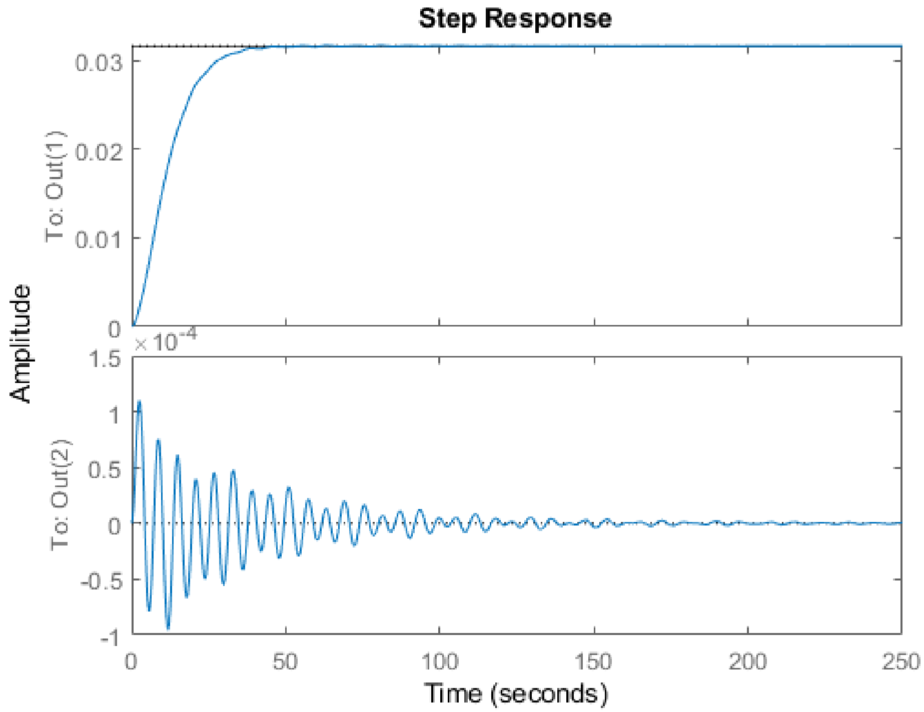
```
    0.0210
    0.1734
   -2.9329
    0.0792
    2.2176
   -1.4496


L3 =

   13.0743   -0.8243
   56.2564   -8.4778
  -89.1734   19.7841
  -20.0624   10.9530
    0.3520    7.9257
    3.4792   13.2136


L4 =

    8.5631   -0.8851    0.0000
   17.5219   -4.9474   -0.9800
   -0.9140    9.4369   -0.0000
   -4.1173   20.9390   -0.0491
    0.0000   -0.0000    3.0000
    0.0000   -0.0980    0.9220
```



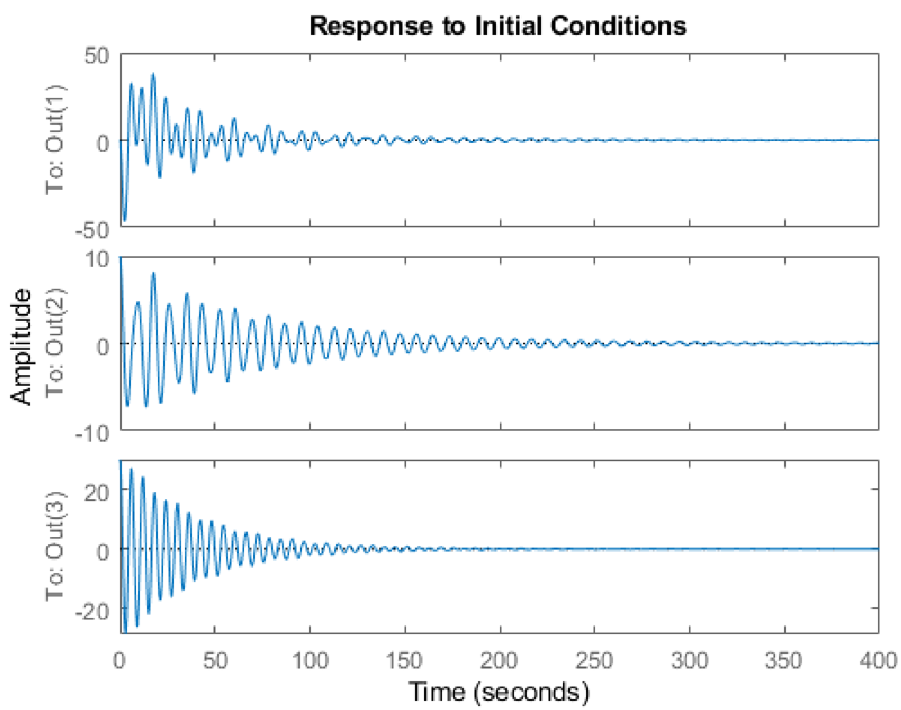

Step Response
Amplitude
Time (seconds)
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
10
20
30
40
50
60

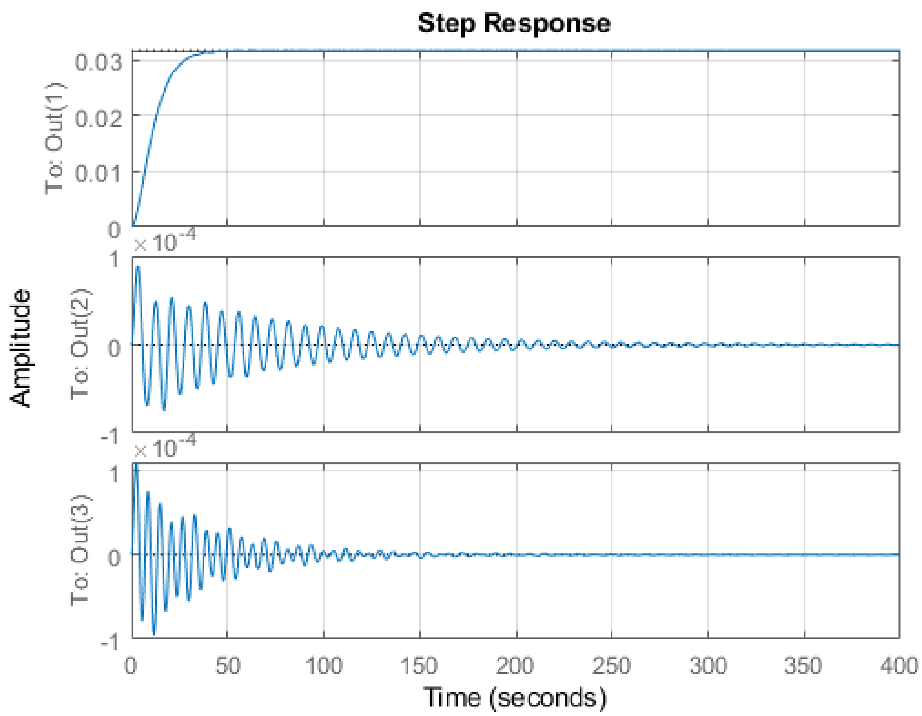
- For x(t)


Response to Initial Conditions
Amplitude
To: Out(1)
To: Out(2)
Time (seconds)
40
20
0
-20
-40
-60
40
20
0
-20
-40
0
50
100
150
200
250
300
350



- For x(t) and theta2(t)





- For x(t) ,theta1(t),theta2(t)

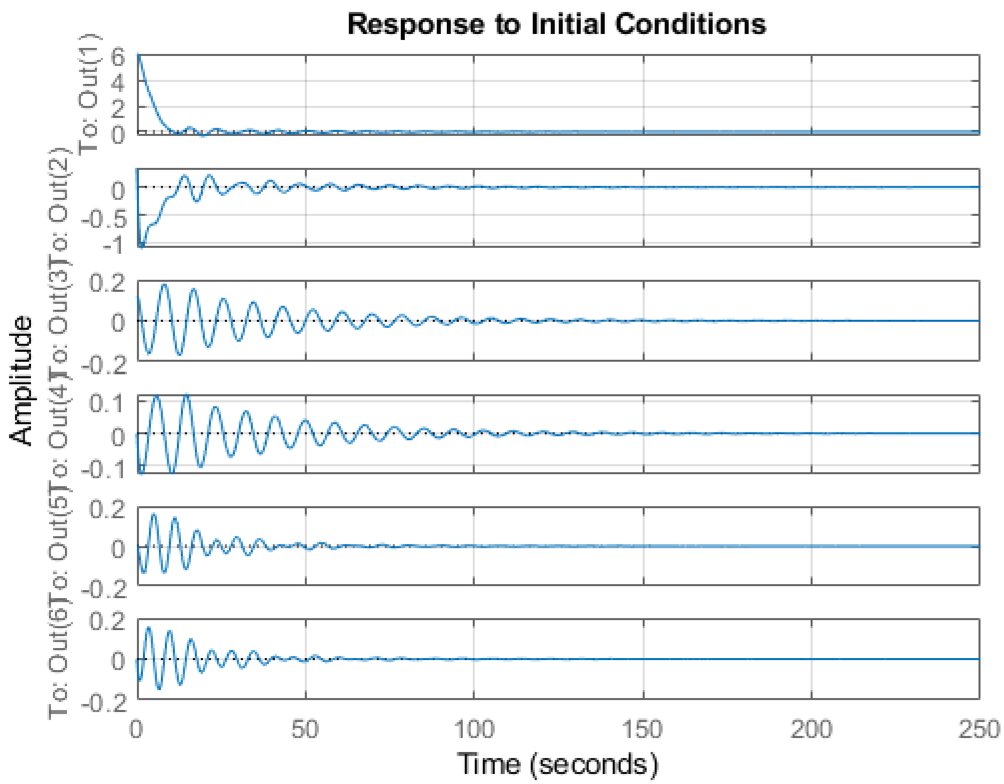
VII. LQG Controller

For linear LQG controller:

- Code for this is in lqg_linear.m

```
syms m1 m2 l1 l2 g M a


m1 = 100;
m2 = 100;
M = 1000;
l1 = 20;
l2 = 10;
g = 9.8;

a = 1;

%Initial Conditions
x0 =[6;0.35;0.12;0;0;0;0;0;0;0;0;0];
```

```
%Matrices
A=  [0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

Q=[10 0 0 0 0 0;
   0 70 0 0 0 0;
   0 0 100 0 0 0;
   0 0 0 80 0 0;
   0 0 0 0 90 0;
   0 0 0 0 0 100];

B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
C=eye(6);
%C = [1,0,0,0,0,0];
D = 0;
R = 0.0001;

K =lqr(A,B,Q,R);%LQR
vd=0.3*eye(6);%Noise
vn=1;%Noise
KF=lqr(A',C',vd,vn);%LQR with Kalman Filter
KF=KF';


if a == 1
    sys = ss([(A-B*K) B*K; zeros(size(A)) (A-KF*C)], [B;zeros(size(B))],[C zeros(size(C))], [0]);
    initial(sys,x0)%No step function
end
if a == 2
    sys = ss([(A-B*K) B*K; zeros(size(A)) (A-KF*C)], [B;zeros(size(B))],[C zeros(size(C))], [0]);
    step(sys)%With step function
end

grid on
```

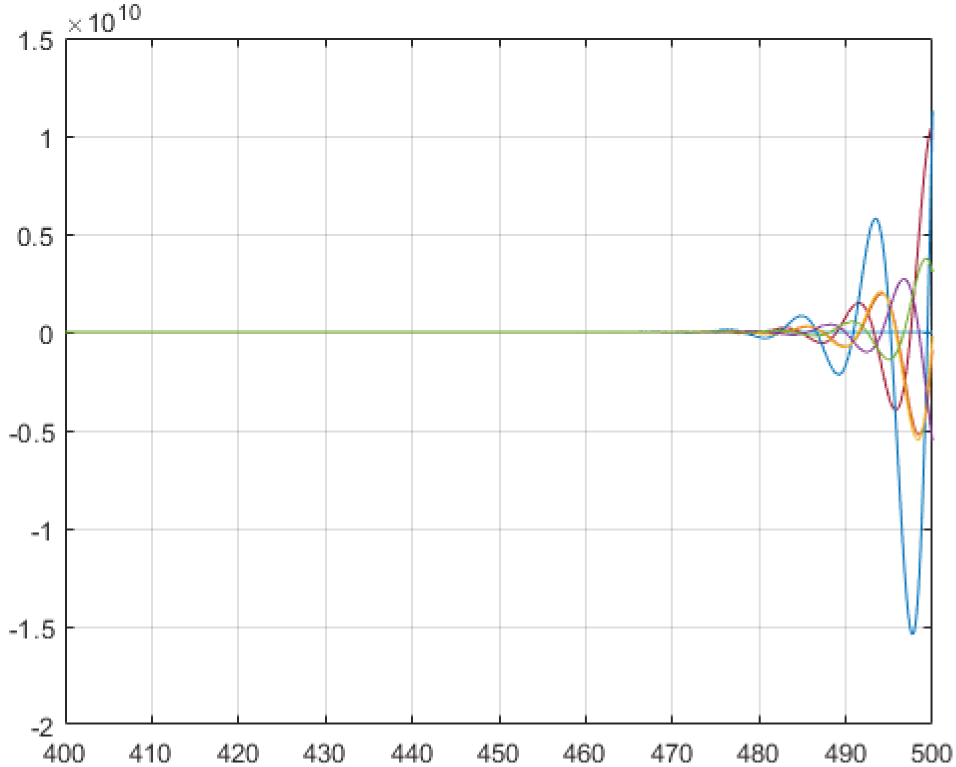


Non-Linear LQG Controller:

- Code for this is in lqg_nonlinear.m

```
clc;
clear all;
%output variable
y0 = [10;0.50;0.7;0;0;0;0;0;0;0;0;0];
tspan = 400:0.01:500;%timespan

[t1,y1] = ode45(@nonlinear_lqg,tspan,y0); %use ode45 function

plot(t1,y1)%plot
grid on

function dydt=nonlinear_lqg(t,x)
%syms m1 m2 l1 l2 g M F U

m1 = 100;
m2 = 100;
M = 1000;
l1 = 20;
l2 = 10;
g = 9.8;
```

```
%Matrices
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
Q=[10 0 0 0 0 0;
   0 700 0 0 0 0;
   0 0 1000 0 0 0;
   0 0 0 800 0 0;
   0 0 0 0 900 0;
   0 0 0 0 0 1000];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
C = [1 0 0 0 0 0];
D = 0;
R = 0.0001;

K =lqr(A,B,Q,R);
vd=0.3*eye(6);%Noise
vn=1;%Noise
KF=lqr(A',C',vd,vn);%lqg similar to lqr with addition of noises(Kalman Filter)
KF=KF';
F=-K*x(1:6);


error_dot=(A-KF*C)*x(7:12);
dydt=zeros(12,1);
D=(M+(m1*((sind(x(3)))^2))+(m2*((sind(x(5)))^2)));
x_d = ((F - g*((m1*sind(x(3))*cosd(x(3)))- (m2*sind(x(5))*cosd(x(5))))...
- (m1*l1*sind(x(3))*((x(4))^2))- (m2*l2*sind(x(5))*((x(6))^2)))/D);
theta1_double_dot = (((x_d * cosd(x(3)))-g*(sind(x(3))))/l1);
theta2_double_dot = (((x_d * cosd(x(5)))-g*(sind(x(5))))/l2);
dydt (1) = x(2);
dydt (2)= x_d;
dydt (3)= x(4);
dydt (4)= theta1_double_dot;
dydt (5)= x(6);
dydt (6)= theta2_double_dot;
dydt (7)= x(2)-x(10);
dydt (8)= dydt(2)-error_dot(2);
dydt (9)= x(4)-x(11);
dydt (10)= dydt(4)-error_dot(4);
dydt (11)= x(6)-x(12);
dydt (12)= dydt(6)-error_dot(6);
end
```



$\times 10^{10}$
1.5
1
0.5
0
-0.5
-1
-1.5
-2
400
410
420
430
440
450
460
470
480
490
500