ENPM 667 FINAL PROJECT

Yash Savle
116740425

Pranav Jain
116694797

UNIVERSITY OF MARYLAND
COLLEGE PARK

Table of Contents

List of Figures

Introduction

The System used in this project is a crane undergoing translational motion. The crane moving in a single direction having mass M consists of two bobs suspended via cables of length l_1 and l_2 respectively. The mass of the bobs is m_1 and $m_{2.}$

In our project we first formulate the equations of motion for the system using Lagrangian mechanics in order to obtain the state-space representation of the system. As the system is non-linear, firstly we linearize the system around the equilibrium point at x= 0 θ_1 = 0 θ_2 = 0. And thus, obtain the state-space model for the linearized system.

The condition for controllability of the system is derived and for a given set of parameters the LQR controller is implemented on this system. By carrying out simulations for several iterations of the LQR values we obtain the optimal response. We simulate the LQR controller for Linear as well as non-linear system. And then verify its stability using Lyapunov's indirect method

For the Second part of the project, initially we check if he system is observable for different sets of output vectors namely, x(t),(θ1(t),θ2(t)),(x(t),θ2(t))or(x(t),θ1(t),θ2(t)). Then we simulate the Luenberger observer for both the linearized system and the original nonlinear system and obtain its response to initial conditions and unit step input. Lastly, we design an output feedback controller using LQG method.

Modelling of the system

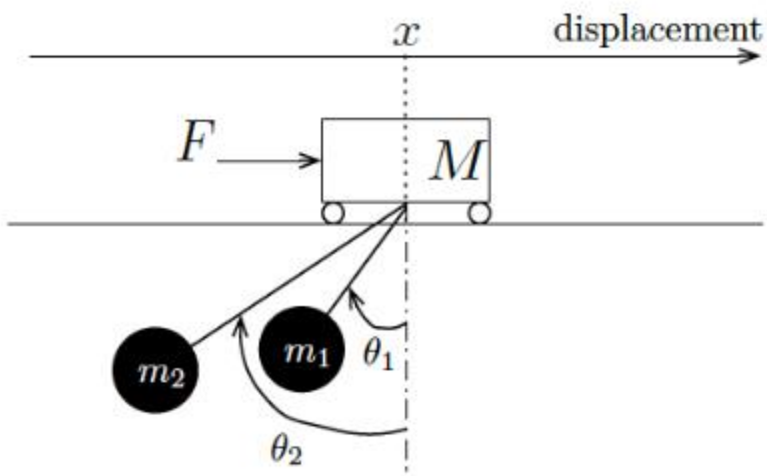

Figure 1: System

$$kE_{cart} = \frac{1}{2} M\dot{x}^2$$

$$KE_{bob1} = \frac{1}{2} m_1[(\dot{x} + v_1 \cos\theta_1)^2 + (v_1 \sin\Theta)^2]$$

$$V_1 = L_1\dot{\theta}_1$$

$$kE_{bob1} = \frac{1}{2} m_1 \left[\dot{x}^2 + 2\dot{x}l_1\,\dot{\theta}_1 \cos\theta_1 + l_1^2\dot{\theta_1}^2\right]$$

$$kE_{bob2} = \frac{1}{2} m_2 \left[\dot{x}^2 + 2\dot{x}l_2\,\dot{\theta}_2 \cos\theta_2 + l_2^2\dot{\theta_2}^2\right]$$

$$PE_{bob1} = m_1 g l_1 (1 - \cos\theta_1)$$

$$PE_{bob2} = m_2 g l_2 (1 - \cos\theta_2)$$

$$L = \sum KE - \sum PE$$

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

$$- m_1 g l_1 (1 - \cos\theta_1) \ - m_2 g l_2 (1 - \cos\theta_2)$$

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

$$\frac{\partial L}{\partial \dot{x}} = \ M\dot{x} + \ m_1\dot{x} + m_1 l_1 \dot{\theta}_1 \cos\theta_1 + m_2\dot{x} + m_2 l_2 \dot{\theta}_2 \cos\theta_2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = \ M\ddot{x} + \ m_1\ddot{x} + m_1 l_1 \left(\ddot{\theta}_1 \cos\theta_1 - \ \dot{\theta_1}^2 \sin\theta_1\right) + m_2\ddot{x} + m_2 l_2 \left(\ddot{\theta}_2 \cos\theta_2 - \ \dot{\theta_2}^2 \sin\theta_2\right)$$

$$\frac{\partial L}{\partial \dot{x}} = 0$$

$$\frac{\partial L}{\partial \dot{\theta}_1} \ = \frac{1}{2} m_1 \left(2\dot{x}\cos\theta_1 \ + 2{l_1}^2\dot{\theta_1}^2\right)$$

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

$$\frac{\partial L}{\partial \theta_1} = m_1\dot{x}l_1\dot{\theta}_1(-\sin\theta_1) - \ \frac{1}{2} m_1 (l_1 g \sin\theta_1)$$

$$\frac{\partial L}{\partial \dot{\theta_2}} = \frac{1}{2} m_2 \left(2\dot{x}\cos\theta_2 + 2{l_2}^2{\dot{\theta_2}}^2\right)$$

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

$$\frac{\partial L}{\partial \theta_2} = m_2\dot{x}l_2\dot{\theta_2}(-\sin\theta_2) - \frac{1}{2}m_2(l_2 g\sin\theta_2)$$

$$\ddot{x}(M + m_1 + m_2) + m_1\ddot{\theta_1}l_1\cos\theta_1 + m_2l_2\ddot{\theta_2}\cos\theta_2 - m_1{\dot{\theta_1}}^2 l_1\sin\theta_1 - m_2l_2\,{\dot{\theta_2}}^2\sin\theta_2 = F$$

$$\ddot{x}(m_1\cos\theta_1) + \ddot{\theta_1}(m_1l_1^2) - \dot{x}l_1m_1\dot{\theta_1}\sin\theta_1 + l_1m_1\dot{\theta_1}\sin\theta_1 + m_1(l_1g\sin\theta_1) = 0$$

$$\ddot{x}(m_2\cos\theta_2) + \ddot{\theta_2}(m_2l_2^2) - \dot{x}l_2m_2\dot{\theta_2}\sin\theta_2 + l_2m_2\dot{\theta_2}\sin\theta_2 + m_2(l_2g\sin\theta_2) = 0$$

$$\ddot{x}\left(\cos\theta_1 + \ddot{\theta_1}l_1^2 + l_1g\sin\theta_1\right) = 0$$

$$\ddot{x}\left(\cos\theta_2 + \ddot{\theta_2}l_2^2 + l_2g\sin\theta_2\right) = 0$$

X=[x θ_1 $\theta_2 \dot{x}$ $\dot{\theta_1}$ $\dot{\theta_2}$]^T

$\dot{X} = [\dot{x}\ \ \dot{\theta_1}\ \dot{\theta_2}\ \ddot{x}\ \ddot{\theta_1}\ \ddot{\theta_2}\ ]^T$

$$\ddot{x} = (F - \dot{\theta}_1^2(-m_1l_1\sin\theta_1) + \dot{\theta}_2^2 m_2\,l_2 sin\,\theta_2 - m_1 gsin\theta_1 cos\theta_1 - m2gsin\theta_2 cos\theta_1)/(M + m1 + m2 - m_1{cos\theta_1}^2 - m_2{cos\theta_2}^2)$$

So now we can write $\ddot{\theta_1}$ and $\ddot{\theta_2}$ in terms of $\ddot{x}$ and in the code we calculate $\ddot{x}$ first and then put it in the equation of $\ddot{\theta_1}$ and $\ddot{\theta_2}$

$$\ddot{\theta_1} = \frac{gsin\theta_1}{l1} - cos\theta_1\frac{\ddot{x}}{l1}$$

$$\ddot{\theta_2} = \frac{gsin\theta_2}{l1} - cos\theta_2\frac{\ddot{x}}{l2}$$

Linearization of the system

In order to linearize the above equations, we differentiate the state equation as follows

F1=$\dot{x}$

F2=$\dot{\theta}_1$

F3=$\dot{\theta}_2$

F4=$\ddot{x}$

F5=$\ddot{\theta}_1$

F6= $\ddot{\theta}_2$

$A = \left[\frac{\partial F_i}{\partial x_j}\right]$ where i goes from 1 to 6 and j goes from 1 to 6 respectively

$$\frac{\partial F_4}{\partial \theta_1} = -\frac{m_1 g}{M}$$

$$\frac{\partial F_4}{\partial \theta_2} = -\frac{m_2 g}{M}$$

$$\frac{\partial F_5}{\partial \theta_1} = \frac{-(M + m_1)g}{M\, l_1}$$

$$\frac{\partial F_5}{\partial \theta_2} = \frac{-m_2 g}{M\, l_1}$$

$$\frac{\partial F_6}{\partial \theta_1} = \frac{-m_1 g}{M\, l_2}$$

$$\frac{\partial F_6}{\partial \theta_2} = \frac{-(M + m_2)g}{M\, l_2}$$

All other derivative terms are zero hence they are not typed in for convenience

$B = \left[\frac{\partial F_i}{\partial u}\right]$

$$B = \left[0\ 0\ 0\, \frac{1}{M}\, \frac{1}{M l_1}\ \frac{1}{M l_2}\right]$$

We get A as

$$A=\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & \frac{\partial F_4}{\partial \theta_1} & \frac{\partial F_4}{\partial \theta_2} & 0 & 0 & 0\\ 0 & \frac{\partial F_5}{\partial \theta_1} & \frac{\partial F_5}{\partial \theta_2} & 0 & 0 & 0\\ 0 & \frac{\partial F_6}{\partial \theta_1} & \frac{\partial F_6}{\partial \theta_2} & 0 & 0 & 0\end{bmatrix}$$

$$A=\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & -\frac{m_1 g}{M} & -\frac{m_2 g}{M} & 0 & 0 & 0\\ 0 & \frac{-(M+m_1)g}{M\, l_1} & \frac{-m_2 g}{M\, l_1} & 0 & 0 & 0\\ 0 & \frac{-m_1 g}{M\, l_2} & \frac{-(M+m_2)g}{M\, l_2} & 0 & 0 & 0\end{bmatrix}$$

Controlability of the system

For finding the condition for the system to be controllable

$$ctrb = [B\ AB\ A^2B\ A^3B\ A^4B]$$

Doing this on MATLAB and taking determinant of the matrix

We further simplify the determinant to obtain

$$D=-g^6 (l_1 - l_2)^2 / M^6 l_1^6 l_2^6$$

Our system is controllable if the Rank of Matrix is 6, which will be only possible for

When determinant of D=0

Hence the system will be only controllable if

$$g^6 (l_1 - l_2)^2 / M^6 l_1^6 l_2^6 = 0$$

$$\therefore l_1 \neq l_2$$

LQR Controller Design

Now for given set of parameters M= 1000Kg, m1=m2= 100Kg, l1= 20m and l2= 10m we will design an LQR controller using matlab. We use the lqr() function in MATLAB for calculating the gain matrix values, This function uses the Algebraic Ricatti equation to find the Matrix K for given set of poles and Matrix A, Q,R. We need to select the values of Q,R,P such that we minimize the cost function

$$\tilde{J} = \frac{1}{2}(x^T Qx + u^T Ru)\, dt + \frac{1}{2}\, x^T(t)Px(t)$$

First we check he controllability of the systems for above values, And as the system is determined to be controllable we carry out iterations to select the values of LQR. Then we plot it's response to arbitary initial conditions `x_0= [ 5 ; 0.1; 0.2 ; 0 ; 0; 0];`

The Matlab code used for the above calculations is as following:

```
clc
clear
syms m1 m2 M l1 l2 x t1 t2 xdot t1dot t2dot F g
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
x_0= [ 5 ; 0.1; 0.2 ; 0 ; 0; 0];
C=eye(6);
D=[0];
A=[0  0  0  1  0  0
    0  0  0  0  1  0
    0  0  0  0  0  1
    0  -m1*g/M  -m2*g/M  0  0  0
    0  -g*(M+m1)/M/l1  -g*m2/M/l1  0  0  0
    0  -g*m1/M/l2  -g*(M+m2)/M/l2  0  0  0];
B=[0  0  0  1/M  1/M/l1  1/M/l2]';
Cont=([B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B]);
rank(Cont)
d=((det(Cont)))

Q=[1  0  0  0  0  0;0  1  0  0  0  0;  0  0  100  0  0  0;  0  0  0  1000  0  0;  0  0  0  0  150  0;  0  0
0  0  0  1500]
R=0.001;
K=lqr(A,B,Q,R);
eig(A-B*K)
sys=ss(A-B*K,B,C,D)
initial(sys,x_0)
```

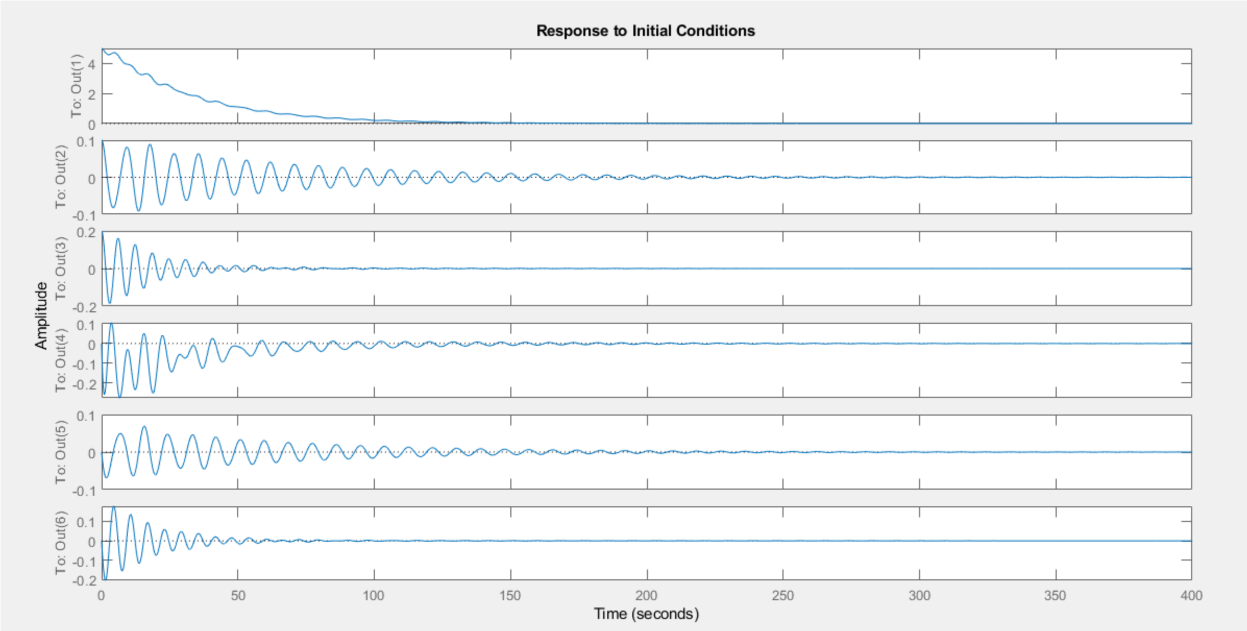

Figure 2: Linear Response to Initial Conditions

For nonlinear System we use ode45 function to solve the equations which is done as follows

```
clc
clear

x_0= [ 5 ;0.10;0.2;0 ;0;0];

t=0:0.01:100;

[t,x]=ode45(@diffrential_eqn,t,x_0)
plot(t,x);


function dx=diffrential_eqn(t,x)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
C=eye(6);
D=[0];
A=[0  0  0  1  0  0
    0  0  0  0  1  0
    0  0  0  0  0  1
    0  -m1*g/M  -m2*g/M  0  0  0
    0  -g*(M+m1)/M/l1  -g*m2/M/l1  0  0  0
    0  -g*m1/M/l2  -g*(M+m2)/M/l2  0  0  0];
B=[0  0  0  1/M  1/M/l1  1/M/l2]';
Q=[1  0  0  0  0  0;0  1  0  0  0  0;  0  0  1  0  0  0;  0  0  0  60  0  0;  0  0  0  0  150  0;  0  0  0  0
0  150];
R=0.0001;
K=lqr(A,B,Q,R);
```

```
eig(A-B*K);
u=-K*x;
dx = zeros(6,1);
dx(1)= x(4);
dx(2)= x(5);
dx(3)= x(6);
dx(4)= -((-u) + m1*l1*x(5)^2*sin(x(2)) + m1*g*sin(x(2))*cos(x(2))+
m2*l2*x(6)^2*sin(x(3))+m2*g*sin(x(3))*cos(x(3)))/(M+m1*sin(x(3)^2)+
m2*sin(x(3)^2));
dx(5)= -((-u) +(M+m1)*g*sin(x(2)) + m1*l1*sin(x(2))*cos(x(2))*x(5)^2 +
m2*l2*sin(x(3))*cos(x(3))*x(6)^2 + m2*g*sin(x(3))*cos(x(2)-x(3)))/(( M+
m1*sin(x(2)^2)+ m2*(sin(x(3)^2)))*l1);
dx(6)= -((-u) + m1*l1*sin(x(2))*cos(x(3))*x(5)^2 +m1*g*sin(x(2))*cos(x(2)-
x(3)) + (M+m1)*g*sin(x(3))+m2*l2*x(3)^2*sin(x(3))*cos(x(3)))/(( M+
m1*sin(x(3)^2)+ m2*(sin(x(3)^2)))*l2);
end
```

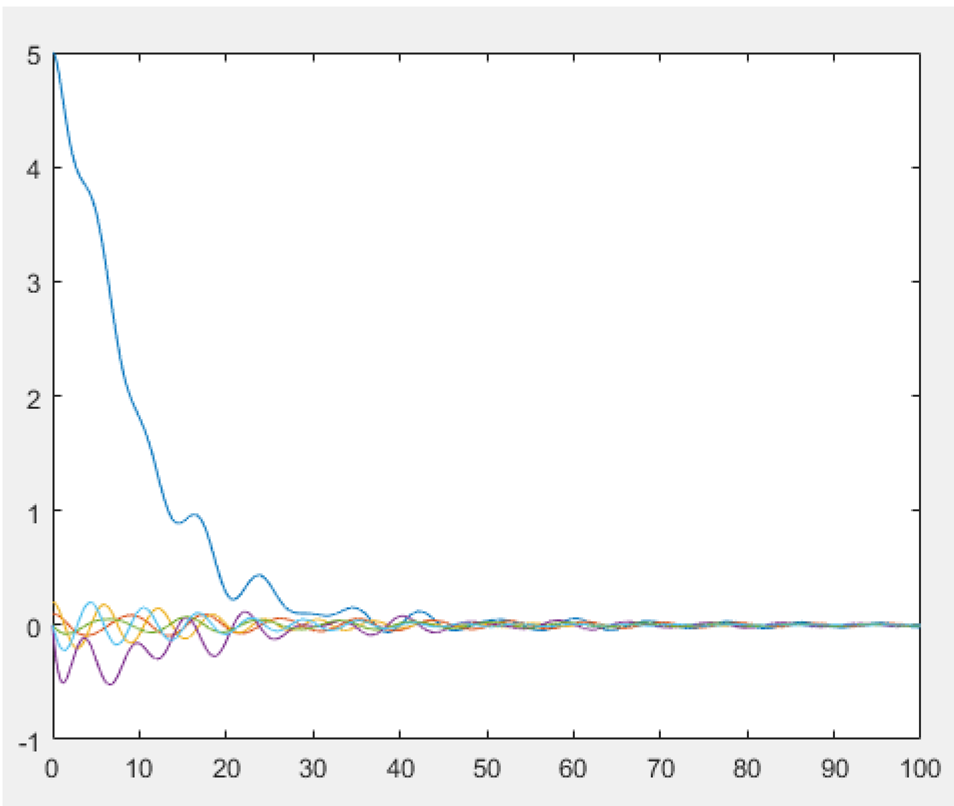

Figure 3: Non-Linear Response to Initial Conditions

To check the stability of the system by using Lyapunov's indirect method we have used the `eig(A-B*K)`

We get the following values

-0.1319 + 0.0000i

-0.0590 + 1.0205i

-0.0590 - 1.0205i

-0.0239 + 0.7158i

-0.0239 - 0.7158i

As all the real parts of these poles are negative the system is stable.

Observability

Now we check if the system is observable for given set of output vectors, by checking the Rank of matrix C we can find if the system is observable or not.

```
M  = 1000;        % M      mass of cart
m1 = 100;         % m1     mass of bob 1
m2 = 100;         % m2     mass of bob 2
l1 = 20;          % l1     length of link of first pendulum
l2 = 10;          % l2     length of link of second pendulum
g  = 9.8;         % g

A =[0 0 0 1 0 0 ; 0 0 0 0 1 0;0 0 0 0 0 1; 0 (-m1*g/M) (-m2*g/M) 0 0 0;
    0 (-g*(M+m1)/(M*l1)) (-m2*g/M*l1) 0 0 0 ; 0 (-m1*g/M*l2) (-
g*(M+m2)/(M*l2)) 0 0 0 ]
% Observability for X(t)
C_1=[1 0 0 0 0 0];
O_1 = [C_1;C_1*A;C_1*A^(2);C_1*A^(3);C_1*A^(4);C_1*A^(5)];
if rank(O_1)==6;
    disp('The system is observable for output x(t)')
    else
    disp('The system is not observable for output x(t)')
end

% Observability for t_1,t_2
C_2=[0 1 0 0 0 0; 0 0 1 0 0 0];
O_2 = [C_2;C_2*A;C_2*A^(2);C_2*A^(3);C_2*A^(4);C_2*A^(5)];
if rank(O_2)==6;
    disp('The system is observable for output t_1,t_2')
else
    disp('The system is not observable for output t_1,t_2')
end

% Observability for x(t),t_2
C_3=[1 0 0 0 0 0; 0 0 1 0 0 0];
O_3 = [C_3;C_3*A;C_3*A^(2);C_3*A^(3);C_3*A^(4);C_3*A^(5)];
if rank(O_3)==6;
    disp('The system is observable for output x(t),t_2')
```

```
else
    disp('The system is not observable for output x(t),t_2')
end

% Observability for x(t), t_1, t_2
C_4=[1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 1 0 0 0];
O_4 = [C_4;C_4*A;C_4*A^(2);C_4*A^(3);C_4*A^(4);C_4*A^(5)];
if rank(O_4)==6;
    disp('The system is observable for output x(t),t_1, t_2')
else
    disp('The system is not observable for output x(t),t_1, t_2')
end
```

Output:

The system is observable for output x(t)

The system is not observable for output t_1,t_2

The system is observable for output x(t),t_2

The system is observable for output x(t),t_1, t_2

Luenberger Observer Design

The state space expression for the system along with Luenberger observer is given as

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - Bk & Bk \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \mathrm{e} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} F$$

We use this function and provide C as an input to check whether system is observable or not and what is its response to initial conditions specified and to step input. Then based on the K value obtained from LQR simulation we can use he place function to find the observer gain matrix L.
The poles for the controller design are chosen at `p=[-5;-2;-1;-9;-6;-7]`

```
function luenberger(C,type)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
%C=eye(6);
D=[0];
A=[0 0 0 1 0 0
    0 0 0 0 1 0
    0 0 0 0 0 1
    0 -m1*g/M -m2*g/M 0 0 0
    0 -g*(M+m1)/M/l1 -g*m2/M/l1 0 0 0
    0 -g*m1/M/l2 -g*(M+m2)/M/l2 0 0 0];
B=[0 0 0 1/M 1/M/l1 1/M/l2]';
Q=[1 0 0 0 0 0;0 1 0 0 0 0; 0 0 1 0 0 0; 0 0 0 60 0 0; 0 0 0 0 150 0; 0 0 0 0
0 150];
R=0.0001;
p=[-5;-2;-1;-9;-6;-7]';
x_0= [ 5 ; 0; 0.1 ; 0 ; 0.2; 0;0;0;0;0;0;0];
Kr=lqr(A,B,Q,R);
eig(A-B*Kr);
Kf=place(A',C',p);
Kf=Kf';
eig(A-Kf*C);
%system=ss([(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)],[B;zeros(size(B))],[C
zeros(size(C))],[0]);
%initial(system,x_0)
x_0= [ 5 ; 0.1; 0.2 ; 0 ; 0;0;0;0;0;0;0;0];

if type==1
%system=ss((A-Kf*C),[B Kf],[C],[0]);
%initial(system,x_0)
system=ss([(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)],[B;zeros(size(B))],[C
zeros(size(C))],[0]);
initial(system,x_0)
end

if type==2
%system=ss((A-Kf*C),[B Kf],[C],[0]);
%step(system)
system=ss([(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)],[B;zeros(size(B))],[C
zeros(size(C))],[0]);
step(system)

end
```

The values for L are

Kf =

1.0e+04 *

0.0030

-1.4599

1.2575

0.0350

-0.1952

-0.3532

Outputs:

For X(t)

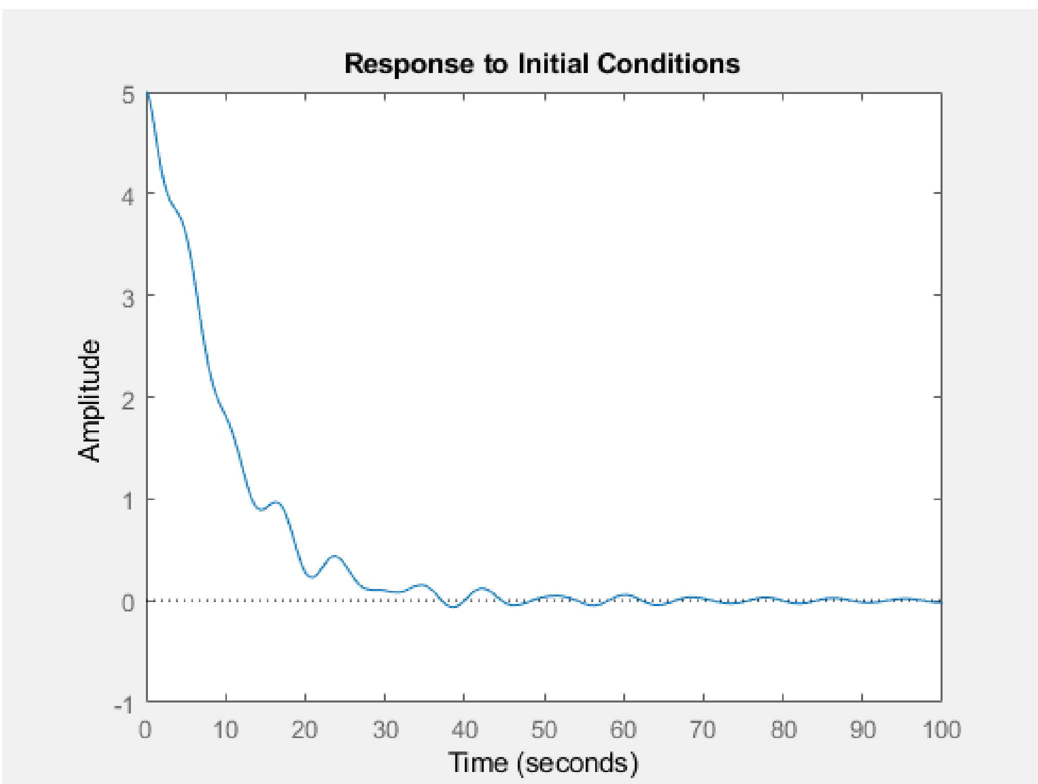

Figure 4: Initial Conditions Response for X(t)

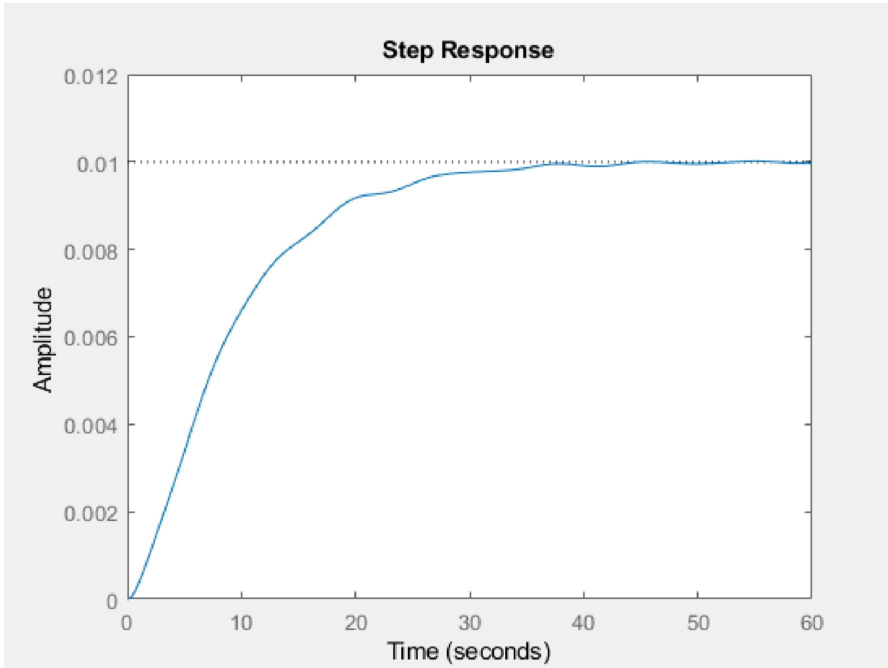

Figure 5 : Sep response for X(t)

For x(t),t_2

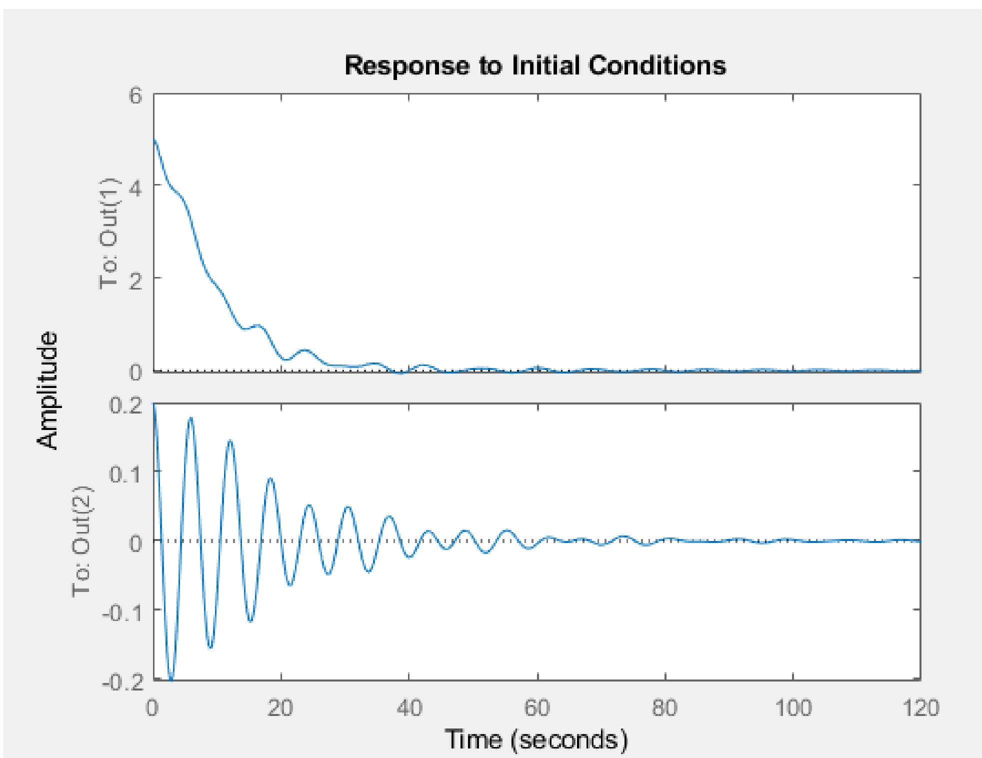

Figure 6: Initial Conditions Response for x(t),t_2

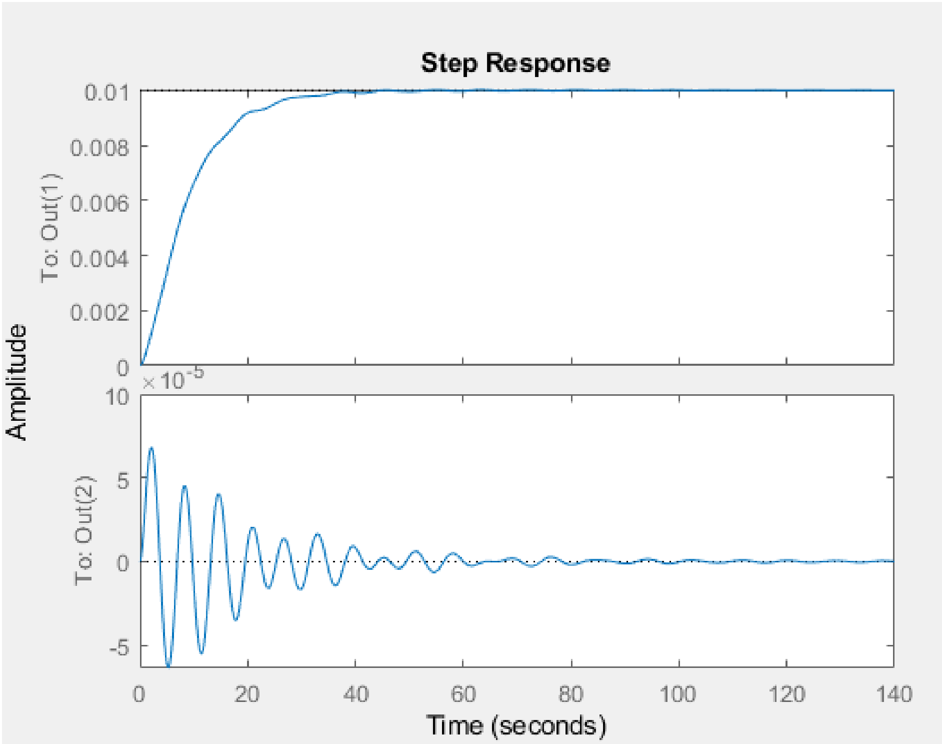

Figure 7 : Step Response for x(t), t_2

For x(t), t_1, t_2

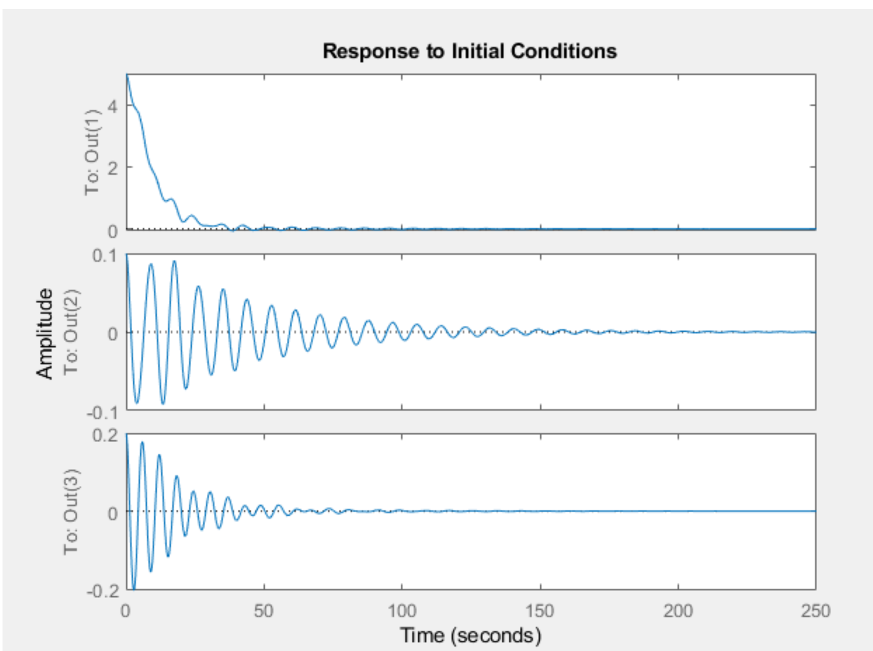

Figure 8 : Initial Condition Response for x(t),t_1, t_2

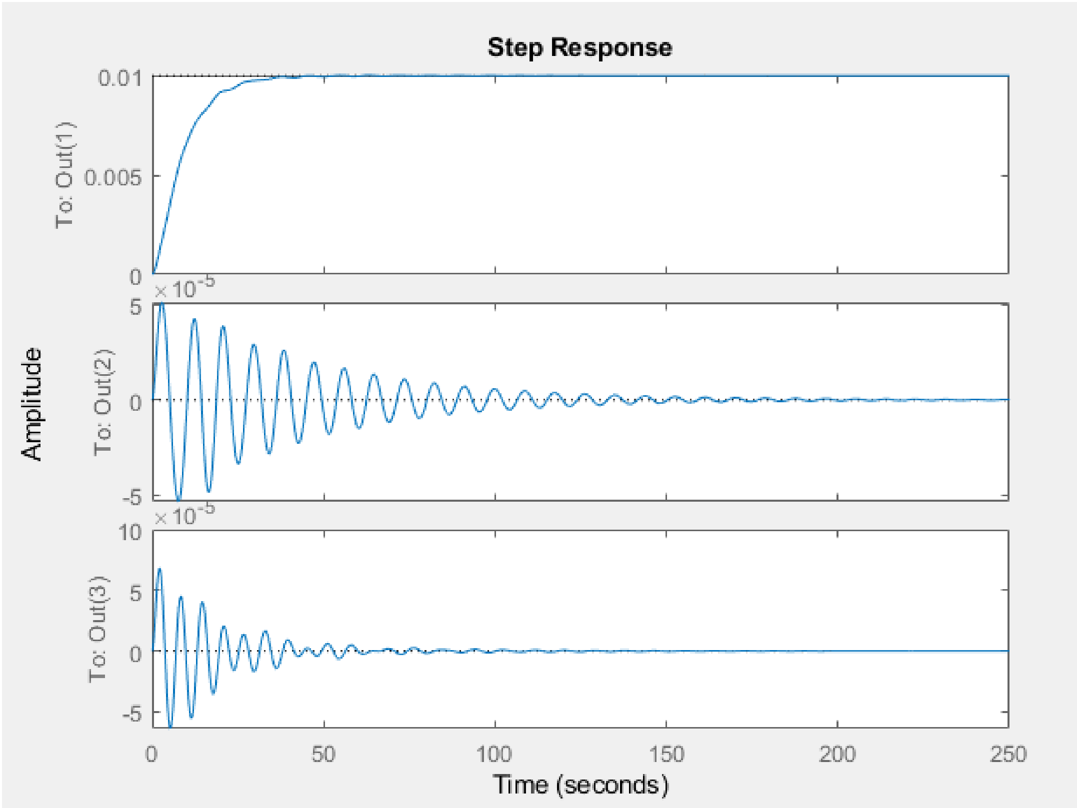

Figure 9 : Step Response for x(t), t_1, t_2

Luenberger Non-Linear

The following code is written to simulate the nonlinear response of the luenberger type observer. We select the observer such that the sensors can read the 1st two states that is x,θ_1.

For the Non-Linear system, we don't use the linearized A matrix, but we solve the differential equations to obtain values of the states after giving them an initial trigger/input. The Input for the 1st six states is the same as the one provided in nonlinear lqr model, and the next 6 states are calculated as shown in the lqg_non linear.m function.

```
function luenberger_non_linear_runner()

x_0= [ 5 ;0.10;0.2;0 ;0;0;0;0;0;0;0;0];

t=0:1:50;

[t,x]=ode45(@luenberger_non_linear,t,x_0);

plot(t,x);
end


function dx=luenberger_non_linear(t,x)
syms x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 u;
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
C=zeros(6,6);C(1,2)=1;C(2,3)=1;
D=[0];
A=[0 0 0 1 0 0
    0 0 0 0 1 0
    0 0 0 0 0 1
    0 -m1*g/M -m2*g/M 0 0 0
    0 -g*(M+m1)/M/l1 -g*m2/M/l1 0 0 0
    0 -g*m1/M/l2 -g*(M+m2)/M/l2 0 0 0];
B=[0 0 0 1/M 1/M/l1 1/M/l2]';
Q=[1 0 0 0 0 0;0 1 0 0 0 0; 0 0 1 0 0 0; 0 0 0 60 0 0; 0 0 0 0 150 0; 0 0 0 0
0 150];
R=0.0001;
p=[-5;-2;-1;-9;-6;-7]';
x_0= [ 5 ; 0; 0.1 ; 0 ; 0.2; 0;0;0;0;0;0;0];
Kr=lqr(A,B,Q,R);

u=-Kr*x(1:6); % for step response u =1
Kf=place(A',C',p);
Kf=Kf';

sigma_dot=(A-Kr.*C)*x(7:12);
dx1(1)= x(4);
dx1(2)= x(5);
dx1(3)= x(6);
dx1(4)= -((-u) + m1*l1*x(5)^2*sin(x(2)) + m1*g*sin(x(2))*cos(x(2))+
m2*l2*x(6)^2*sin(x(3))+m2*g*sin(x(3))*cos(x(3)))/(M+m1*sin(x(3)^2)+
m2*sin(x(3)^2));
dx1(5)= -((-u) +(M+m1)*g*sin(x(2)) + m1*l1*sin(x(2))*cos(x(2))*x(5)^2 +
m2*l2*sin(x(3))*cos(x(3))*x(6)^2 + m2*g*sin(x(3))*cos(x(2)-x(3)))/(( M+
m1*sin(x(2)^2)+ m2*(sin(x(3)^2)))*l1);
dx1(6)= -((-u) + m1*l1*sin(x(2))*cos(x(3))*x(5)^2 +m1*g*sin(x(2))*cos(x(2)-
x(3)) + (M+m1)*g*sin(x(3))+m2*l2*x(3)^2*sin(x(3))*cos(x(3)))/(( M+
m1*sin(x(3)^2)+ m2*(sin(x(3)^2)))*l2);
dx1(7)=x(4)-x(10);
dx1(8)=x(5)-x(11);
dx1(9)=x(6)-x(12);
dx1(10)=dx1(4)-sigma_dot(4);
```

```
dx1(11)=dx1(5)-sigma_dot(5);
dx1(12)=dx1(6)-sigma_dot(6);


Acc=[(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)];
Bcc=[B;zeros(size(B))];
Ccc=[C zeros(size(C))];
Dcc=[0];
Qcc=eye(12)*10;
Rcc=0.01;
Ksyslqr=lqr(Acc,Bcc,Qcc,Rcc);
xcc=x;
u=-Ksyslqr*x;
Xdot=Acc*xcc+Bcc*u;
dx=Xdot;
dx=dx1';


end
```

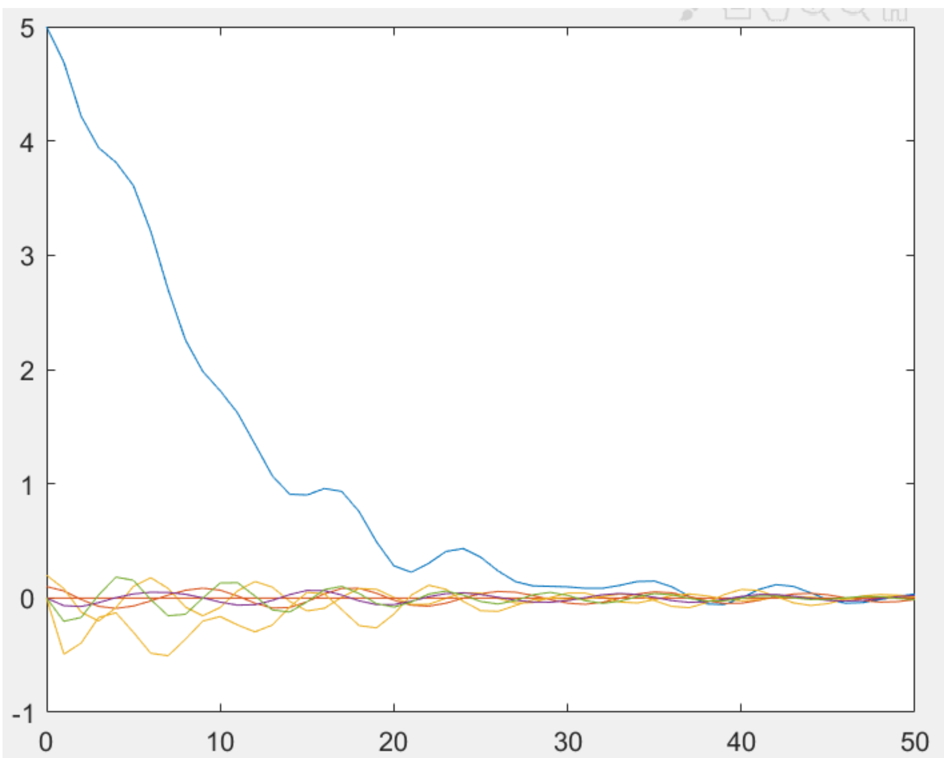

Figure 10 : Luenberger Observer Response for Non-Linear System

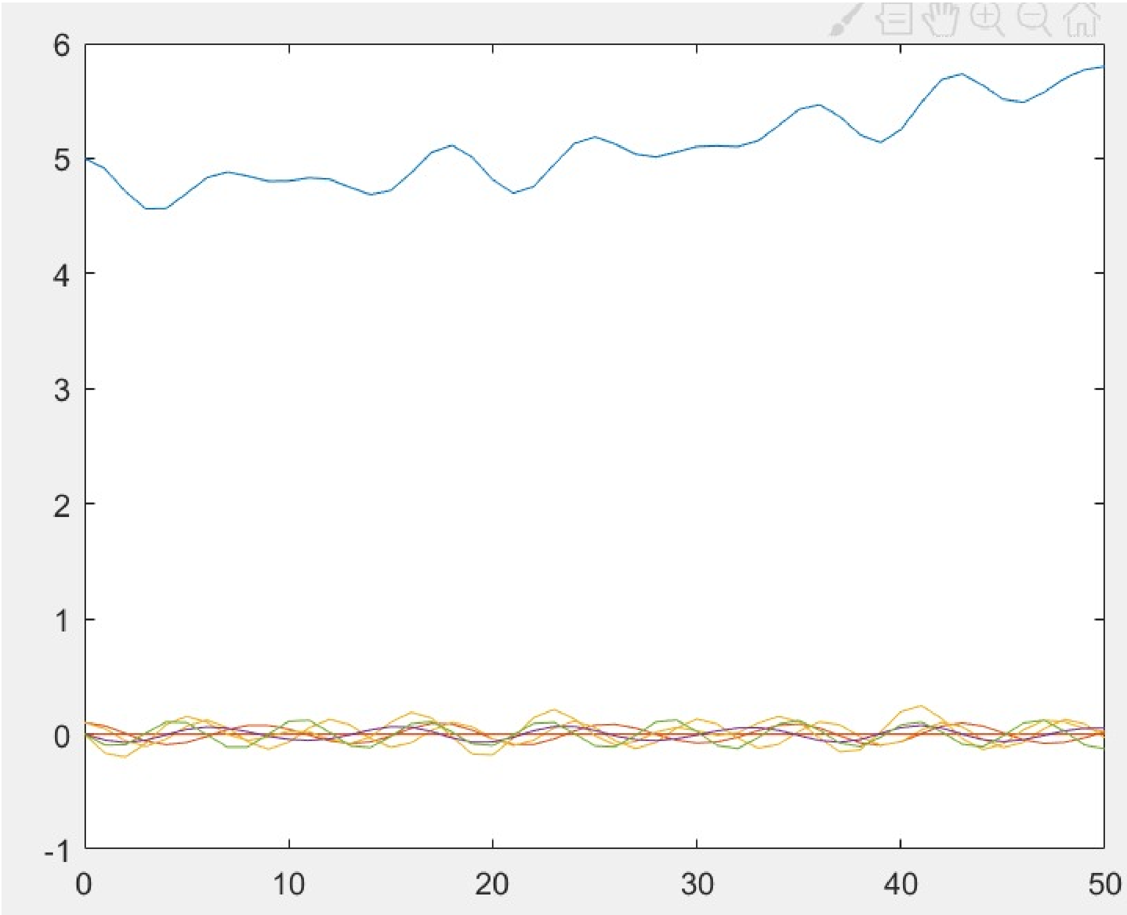

Figure 11: Step Response for Luenberger Non-Linear

LQG Controller

The Linear Quadratic Gaussian controller uses an LQR controller and Kalman Filter as a state estimator

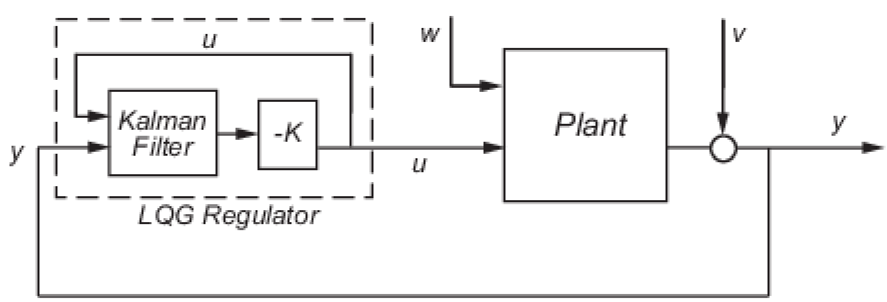

Figure 12 : Block diagram of LQG controller adopted from [1]

The state space equations are given as follows:

X'=Ax+Bu+Gw

Y=Cx+Du+Hw+v

For our case D and H are zero

We Set Vd=0.6 Vn=3 and value of L as obtained above, then by using following code we can formulate the LQG controller and simulate it for linear system. Similar to the lqr controller design we use the same lqr() function for calculating the gain matrix for LQG.

```
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
C=[1 0 0 0 0 0];
D=[0];
type=1;
A=[0 0 0 1 0 0
    0 0 0 0 1 0
    0 0 0 0 0 1
    0 -m1*g/M -m2*g/M 0 0 0
    0 -g*(M+m1)/M/l1 -g*m2/M/l1 0 0 0
    0 -g*m1/M/l2 -g*(M+m2)/M/l2 0 0 0];
B=[0 0 0 1/M 1/M/l1 1/M/l2]';
Q=[1 0 0 0 0 0;0 1 0 0 0 0; 0 0 1 0 0 0; 0 0 0 60 0 0; 0 0 0 0 150 0; 0 0 0 0
0 150];
R=0.0001;
x_0= [ 5 ; 0; 0.1 ; 0 ; 0.2; 0;0;0;0;0;0;0];
Kr=lqr(A,B,Q,R);
eig(A-B*Kr);
vd=0.3*eye(6);
vn=1;
Kf=lqr(A',C',vd,vn);
Kf=Kf';
eig(A-Kf*C);

x_0= [ 5 ; 0; 0.1 ; 0 ; 0.2;0;0;0;0;0;0;0];

if type==1

system=ss([(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)],[B;zeros(size(B))],[C
zeros(size(C))],[0]);
initial(system,x_0)
end

if type==2

system=ss([(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)],[B;zeros(size(B))],[C
zeros(size(C))],[0]);
step(system)
end
```

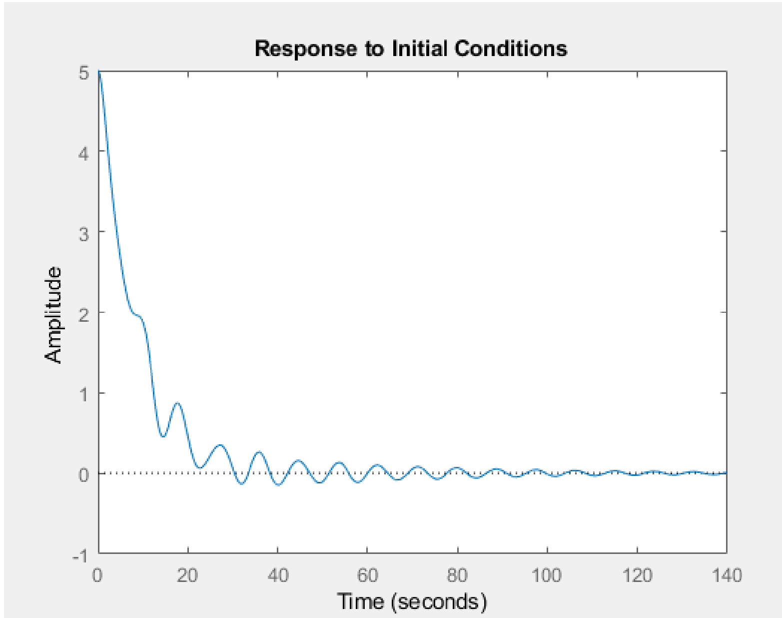

Figure 13: Response for LQG Controller

For Non-Linear System:

For the Non Linear system, we in lqg we don't use the linearized A matrix but we solve the differential equations to obtain values of the states after giving them a initial trigger/input. The Input for the 1st six states is the same as the one provided in non linear lqr model, and the next 6 states are calculated as shown in the lqg_non linear.m function

```
function lqg_non_linear_runner()
clc
clear
x_0= [ 5 ;0.10;0.2;0 ;0;0;0;0;0;0;0;0];

t=0:0.01:50;

[t,x]=ode45(@lqg_non_linear,t,x_0);
 plot(t,x)
end

function dx=lqg_non_linear(t,x)

M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
C=[1 0 0 0 0 0];% C=zeros(6,6);C(1,2)=1;C(2,3)=1;

D=[0];
A=[0 0 0 1 0 0
```

```
    0 0 0 0 1 0
    0 0 0 0 0 1
    0 -m1*g/M -m2*g/M 0 0 0
    0 -g*(M+m1)/M/l1 -g*m2/M/l1 0 0 0
    0 -g*m1/M/l2 -g*(M+m2)/M/l2 0 0 0];
B=[0 0 0 1/M 1/M/l1 1/M/l2]';
Q=[1 0 0 0 0 0;0 1 0 0 0 0; 0 0 1 0 0 0; 0 0 0 60 0 0; 0 0 0 0 150 0; 0 0 0 0
0 150];
R=0.0001;
% p=[-5;-2;-1;-9;-6;-7]';
% x_0= [ 5 ; 0; 0.1 ; 0 ; 0.2; 0;0;0;0;0;0;0];
Kr=lqr(A,B,Q,R);

vd=0.3*eye(6);
vn=1;
Kf=lqr(A',C',vd,vn);
Kf=Kf';
u=-Kr*x(1:6);

sigma_dot=(A-Kf*C)*x(7:12);
dx1(1)= x(4);
dx1(2)= x(5);
dx1(3)= x(6);
dx1(4)= -((-u) + m1*l1*x(5)^2*sin(x(2)) + m1*g*sin(x(2))*cos(x(2))+
m2*l2*x(6)^2*sin(x(3))+m2*g*sin(x(3))*cos(x(3)))/(M+m1*sin(x(3)^2)+
m2*sin(x(3)^2));
dx1(5)= -((-u) +(M+m1)*g*sin(x(2)) + m1*l1*sin(x(2))*cos(x(2))*x(5)^2 +
m2*l2*sin(x(3))*cos(x(3))*x(6)^2 + m2*g*sin(x(3))*cos(x(2)-x(3)))/(( M+
m1*sin(x(2)^2)+ m2*(sin(x(3)^2)))*l1);
dx1(6)= -((-u) + m1*l1*sin(x(2))*cos(x(3))*x(5)^2 +m1*g*sin(x(2))*cos(x(2)-
x(3)) + (M+m1)*g*sin(x(3))+m2*l2*x(3)^2*sin(x(3))*cos(x(3)))/(( M+
m1*sin(x(3)^2)+ m2*(sin(x(3)^2)))*l2);
dx1(7)=x(4)-x(10);
dx1(8)=x(5)-x(11);
dx1(9)=x(6)-x(12);
dx1(10)=dx1(4)-sigma_dot(4);
dx1(11)=dx1(5)-sigma_dot(5);
dx1(12)=dx1(6)-sigma_dot(6);



Acc=[(A-B*Kr) (B*Kr); zeros(size(A)) (A-Kf*C)];
Bcc=[B;zeros(size(B))];
Ccc=[C zeros(size(C))];
Dcc=[0];
Qcc=eye(12)*10;
Rcc=0.01;
Ksyslqr=lqr(Acc,Bcc,Qcc,Rcc);
xcc=x;
u=-Ksyslqr*x; % for step response u =1
Xdot=Acc*xcc+Bcc*u;
dx=Xdot;
dx=dx1';


end
```

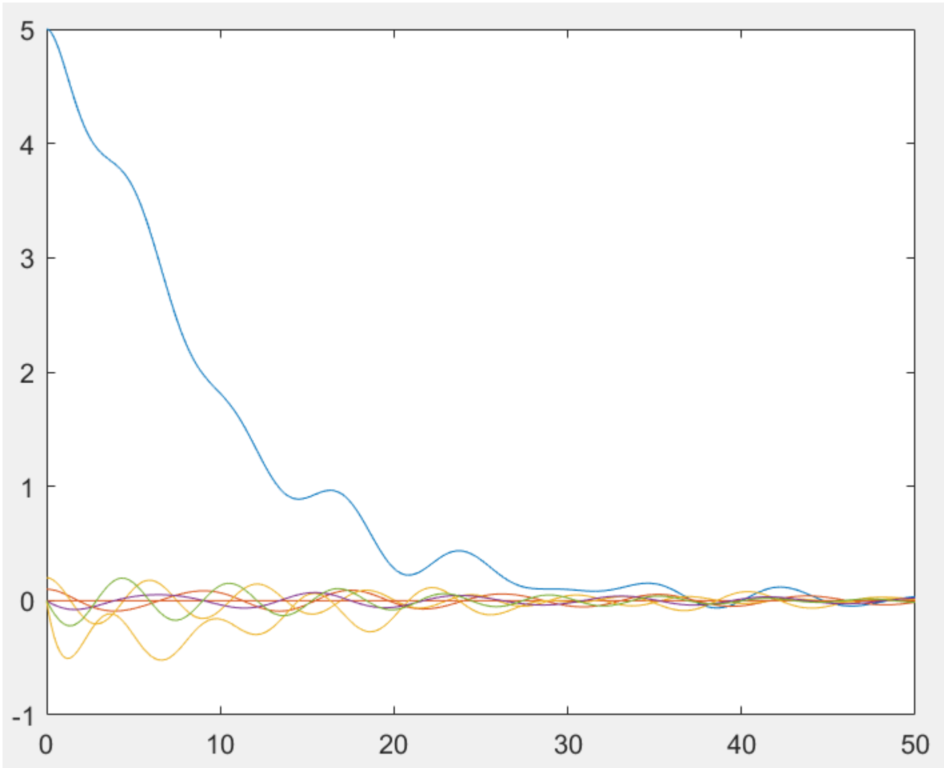

Figure 14: Response of Non-Linear LQG

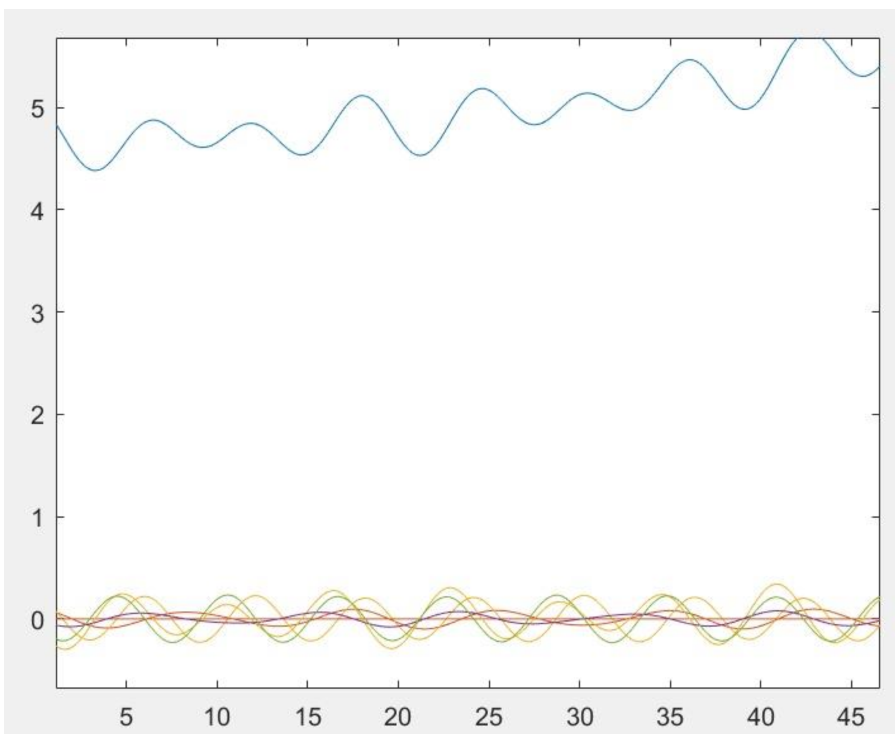

Figure 15: Step Response LQG Non-Linear

Reference Tracking

For constant tracking of the reference signal x_{ref} we design the controller such hat the steady state error is minimised to zero. First, we find out he error in state transition matrix form. Our system is non-linear hence for convenience let us consider it to be f(x,u)= f(x)+ b(x)u. This concept of Reference tracking was adapted from [2]

Hence, e= x_{ref} – x and v = u_{ref} – u. Taking derivative of this equation we obtain,

$$\dot{e} = \dot{x}_{ref} - \dot{x}$$

$$\dot{e} = f(x_{ref}) + b(x_{ref})u_{ref} - f(x) + b(x)u$$

$$= f(e+x) - f(x) + b(e+x)(v+u) - b(x)u$$

$$= F(e, v, x(t), u(t))$$

Now we Linearize this function at point e=0

$$\dot{e} \approx A(t)e + B(t)v$$

$$\text{where } A(t) = \frac{\partial f}{\partial e} \; and \; B(t) = \frac{\partial f}{\partial v}$$

Consider a state feedback controller v= K*x*e

Which gives us u = K(x_{ref} – x) + u

Where the gain K will help us reduce the tracking error.

Conclusion

In this project we formulated the Mathematical Model of a given system using Euler-Lagrangian equations, which help us in formulating the state space equations for the original Non-linear System. Then we Linearize the system to Implement LQR controller and Luenberger observer. In an LQR controller-based system all states of the system are observed as we can use a Kalman filter to estimate the state from output. In an LQG controller we introduce Gaussian noise and does not require all the states to be observed, we also minimize the expectation of cost function. From the results obtained through the simulations we can conclude that our design also rejects constant force disturbances applied to the cart.

References

[1] MathWorks, (2019). *Linear-Quadratic-Gaussian (LQG) Design (R2019b)*. Retrieved December 14, 2019 from https://www.mathworks.com/help/control/getstart/linear-quadratic-gaussian-lqg-design.html.

[2] R. M. Murray. Lecture 2 – LQR Control. Preprint, 2006. Available at https://www.cds.caltech.edu/~murray/courses/cds110/wi06/lqr.pdf.

[3] B. Friedland. Control System Design: An Introduction to State Space Methods. Dover, 2004.