University of Maryland

Controls Final Project

Project Report

Linear Quadratic Regulator and Linear Quadratic Gaussian Controller Design

Pranali Desai
116182935

Nikhil Mehra
116189941

December 17, 2018

Contents

List of Figures

Question A

$$r_1(t) = (x - l_1 sin\theta_1)i - l_1 cos\theta_1 j$$

$$r_2(t) = (x - l_2 sin\theta_2)i - l_2 cos\theta_2 j$$

where x, θ_1 and θ_2 are functions of time

$$\dot{r}_1(t) = (\dot{x} - l_1\dot{\theta}_1 cos\theta_1)i + l_1\dot{\theta}_1 sin\theta_1 j$$

$$\dot{r}_2(t) = (\dot{x} - l_2\dot{\theta}_2 cos\theta_2)i + l_2\dot{\theta}_2 sin\theta_2 j$$

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

$$P.E = -mgl_1 cos\theta_1 - mgl_2 cos\theta_2$$

Lagrange Equation

$$L = K.E - P.E$$

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1(\dot{x} - l_1\dot{\theta}_1 cos\theta_1)^2 + \frac{1}{2}m_2(\dot{x} - l_2\dot{\theta}_2 cos\theta_2)^2 + \frac{1}{2}m_1(l_1\dot{\theta}_1 sin\theta_1)^2 + \frac{1}{2}m_2(l_2\dot{\theta}_2 sin\theta_2)^2 + mgl_1 cos\theta_1 + mgl_2 cos\theta_2 \quad (1)$$

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

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

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

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

Similarly for θ_1

$$\frac{\partial L}{\partial \dot{\theta}_1} = m_1(\dot{x} - l_1\dot{\theta}_1 cos\theta_1)(-l_1 cos\theta_1) + m_1(l_1\dot{\theta}_1 sin\theta_1)(l_1 sin\theta_1)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}_1} = -m_1\ddot{x}l_1 cos\theta_1 + m_1 l_1^2\ddot{\theta}_1 + m_1\dot{x}l_1\ddot{\theta}_1 sin\theta_1$$

$$\frac{\partial L}{\partial \theta_1} = m_1 l_1^2\dot{\theta}_1 - m_1\dot{x}l_1 cos\theta_1$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = -m_1\ddot{x}l_1 cos\theta_1 + m_1 l_1^2\ddot{\theta}_1 + m_1\dot{x}l_1\ddot{\theta}_1 sin\theta_1 - m_1 l_1^2\dot{\theta}_1 + m_1\dot{x}l_1 cos\theta_1 = 0 \quad (3)$$

Similarly for θ_2

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2(\dot{x} - l_2\dot{\theta}_2 cos\theta_2)(-l_2 cos\theta_2) + m_2(l_2\dot{\theta}_2 sin\theta_2)(l_2 sin\theta_2)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}_2} = -m_2\ddot{x}l_2cos\theta_2 + m_2l_2^2\ddot{\theta}_2 + m_2\dot{x}l_2\ddot{\theta}_2sin\theta_2$$

$$\frac{\partial L}{\partial \theta_2} = m_2l_2^2\dot{\theta}_2 - m_2\dot{x}l_2cos\theta_2$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = -m_2\ddot{x}l_2cos\theta_2 + m_2l_2^2\ddot{\theta}_2 + m_2\dot{x}l_2\ddot{\theta}_2sin\theta_2 - m_2l_2^2\dot{\theta}_2 + m_2\dot{x}l_2cos\theta_2 = 0 \tag{4}$$

From equation (3) and (4)

$$l_1\ddot{\theta}_1 = \ddot{x}cos\theta_1 - gsin\theta_1 \tag{5}$$

$$l_2\ddot{\theta}_2 = \ddot{x}cos\theta_2 - gsin\theta_2 \tag{6}$$

Putting (5) and (6) in (2)

$$(M + m_1 + m_2)\ddot{x} = m_1(\ddot{x}cos\theta_1 - gsin\theta_1)cos\theta_1 + m_2(\ddot{x}cos\theta_2 - gsin\theta_2)cos\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2 + F$$

$$\ddot{x}(M + m_1sin^2\theta_1 + m_2sin^2\theta_2) = F - m_1gcos\theta_1sin\theta_1 - m_2gcos\theta_2sin\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2 \tag{7}$$

Putting $\ddot{x}$ from (7) to (5) and (6)

$$l_1\ddot{\theta}_1 = cos\theta_1\frac{(F - m_1gcos\theta_1sin\theta_1 - m_2gcos\theta_2sin\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2)}{(M + m_1sin^2\theta_1 + m_2sin^2\theta_2)} - gsin\theta_1 \tag{8}$$

$$l_2\ddot{\theta}_2 = cos\theta_2\frac{(F - m_1gcos\theta_1sin\theta_1 - m_2gcos\theta_2sin\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2)}{(M + m_1sin^2\theta_1 + m_2sin^2\theta_2)} - gsin\theta_2 \tag{9}$$

Non-Linear State Space Representation

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = f(t, x(t), \dot{x}(t), \theta_1(t), \dot{\theta}_1(t), \theta_2(t), \dot{\theta}_2(t))$$

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = \begin{bmatrix} \frac{(F - m_1gcos\theta_1sin\theta_1 - m_2gcos\theta_2sin\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2)}{(M + m_1sin^2\theta_1 + m_2sin^2\theta_2)} \\ \frac{cos\theta_1(F - m_1gcos\theta_1sin\theta_1 - m_2gcos\theta_2sin\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2)}{l_1(M + m_1sin^2\theta_1 + m_2sin^2\theta_2)} - \frac{gsin\theta_1}{l_1} \\ \frac{cos\theta_2(F - m_1gcos\theta_1sin\theta_1 - m_2gcos\theta_2sin\theta_2 - m_1l_1\dot{\theta}_1^2sin\theta_1 - m_2l_2\dot{\theta}_2^2sin\theta_2)}{l_2(M + m_1sin^2\theta_1 + m_2sin^2\theta_2)} - \frac{gsin\theta_2}{l_2} \end{bmatrix} \tag{10}$$

where in RHS θ_1, θ_2 are functions of time

Question B

Let our state be

$$\begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta}_1(t) \\ \theta_2(t) \\ \dot{\theta}_2(t) \end{bmatrix} \tag{11}$$

Linearizing equation (12) at equilibrium point $x = 0$, $\theta_1 = 0$ and $\theta_2 = 0$, considering $sin\theta \approx \theta$ and $cos\theta \approx 1$

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = \begin{bmatrix} \frac{F}{M} - \frac{m_1 g \theta_1}{M} - \frac{m_2 g \theta_2}{M} \\ \frac{F}{Ml_1} - \frac{m_1 g \theta_1}{Ml_1} - \frac{g\theta_1}{l_1} - \frac{m_2 g \theta_2}{Ml_1} \\ \frac{F}{Ml_2} - \frac{m_2 g \theta_2}{Ml_2} - \frac{g\theta_2}{l_2} - \frac{m_1 g \theta_1}{Ml_2} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{m_1 g}{M} & -\frac{m_2 g}{M} \\ 0 & -\frac{m_1 g}{Ml_1} - \frac{g}{l_1} & -\frac{m_2 g}{Ml_1} \\ 0 & -\frac{m_1 g}{Ml_2} & -\frac{m_2 g}{Ml_2} - \frac{g}{l_2} \end{bmatrix} \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{M} \\ \frac{1}{Ml_1} \\ \frac{1}{Ml_2} \end{bmatrix} F \tag{13}$$

Using state (11)

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \\ \dot{\theta}_1(t) \\ \ddot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{m_1 g}{M} & 0 & -\frac{m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{m_1 g}{Ml_1} - \frac{g}{l_1} & 0 & -\frac{m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{m_1 g}{Ml_2} & 0 & -\frac{m_2 g}{Ml_2} - \frac{g}{l_2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} F \tag{14}$$

Using Lyapunov's Indirect Method

$$Jacobian = \begin{bmatrix} \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \theta_1} & \frac{\partial \ddot{x}}{\partial \dot{\theta}_1} & \frac{\partial \ddot{x}}{\partial \theta_2} & \frac{\partial \ddot{x}}{\partial \dot{\theta}_2} \\ \frac{\partial \ddot{\theta}_1}{\partial x} & \frac{\partial \ddot{\theta}_1}{\partial \dot{x}} & \frac{\partial \ddot{\theta}_1}{\partial \theta_1} & \frac{\partial \ddot{\theta}_1}{\partial \dot{\theta}_1} & \frac{\partial \ddot{\theta}_1}{\partial \theta_2} & \frac{\partial \ddot{\theta}_1}{\partial \dot{\theta}_2} \\ \frac{\partial \ddot{\theta}_2}{\partial x} & \frac{\partial \ddot{\theta}_2}{\partial \dot{x}} & \frac{\partial \ddot{\theta}_2}{\partial \theta_1} & \frac{\partial \ddot{\theta}_2}{\partial \dot{\theta}_1} & \frac{\partial \ddot{\theta}_2}{\partial \theta_2} & \frac{\partial \ddot{\theta}_2}{\partial \dot{\theta}_2} \end{bmatrix}$$

$$A_F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{m_1 g}{M} & 0 & -\frac{m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{m_1 g}{Ml_1} - \frac{g}{l_1} & 0 & -\frac{m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{m_1 g}{Ml_2} & 0 & -\frac{m_2 g}{Ml_2} - \frac{g}{l_2} & 0 \end{bmatrix} \tag{15}$$

Question C

The condition for system to be Controllable.

```
syms m1 m2 m l1 l2 g


%% A matrix
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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];


%% B matrix
B=[0; 1/m; 0; 1/(m*l1); 0; 1/(m*l2)];


%% Check for Controlability
co1=[B A*B A^2*B A^3*B A^4*B A^5*B]
simplify(co1)
rank(co1)
%The rank of the matrix is 6.
%Thus we see that controllability decreases only
%when l1 = l2 by comparing rows for linear
%independance.Thus the condition for it to be controllable is l1
%cannot be equal to l2. This makes sense in physical
%system as the pendulumms will collide if they have the same lenghth.
```

Question D

```
%% LQR For linearized system
%% Parameters already given
m=1000;
l1=20;
l2=10;
g=9.8;
m1=100;
m2=100;
%% defining the state 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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];
B=[0; 1/m; 0; 1/(m*l1); 0; 1/(m*l2)];
C=[1 0 0 0 0 0; 0 0 1 0 0 0;0 0 0 0 1 0];
D=0;
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'}; %states
inputs = {'F'};    % intput
outputs = {'x' ;'p1';'p2'};  % output
%% defining the lqr inputs
Q = [1000 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 1000000 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 1000000 0;
    0 0 0 0 0 0]       %LQR Input
R = 0.1 ;              %LQR Input
K = lqr(A,B,Q,R)       %Gain Calculation from LQR
%% applying the lqr gain K
Ac = [(A-(B*K))];
Bc = [B];
Cc = [C];
Dc = [D];
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};  %states
inputs = {'f'}; % intput after lqr
outputs = {'x' ;'p1';'p2'}; % output after lqr
sys_cl = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,
'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
f = 50*ones(size(t));
[y,t,x]=lsim(sys_cl,f,t); %simulates response
```

```
%% plotting the response
[AX,H1,H2] = plotyy(t,y(:,1),[t,t],[y(:,2),y(:,3)],'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)');
set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)');
title('Step Response with LQR Control');
```

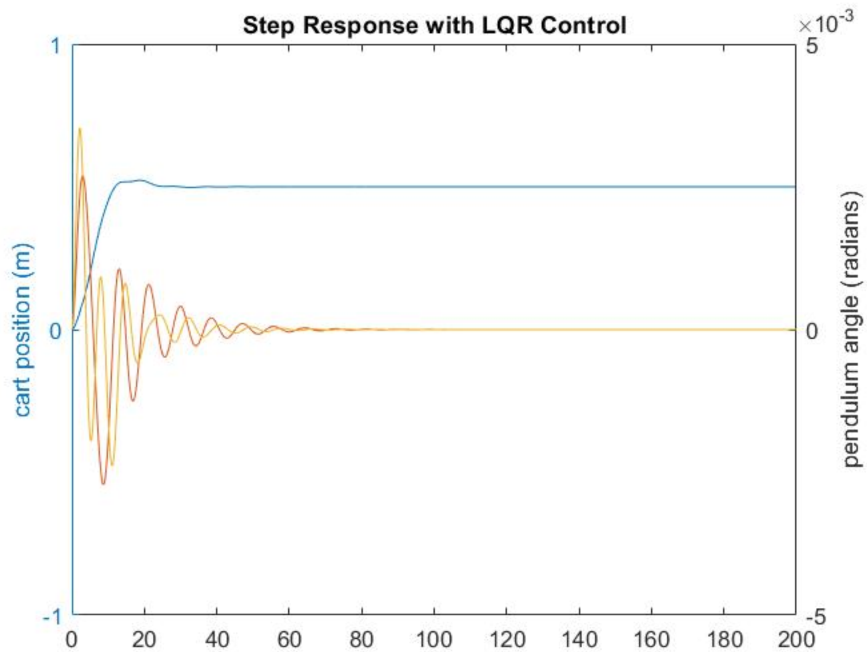

Figure 1: LQR Control on Linearized model

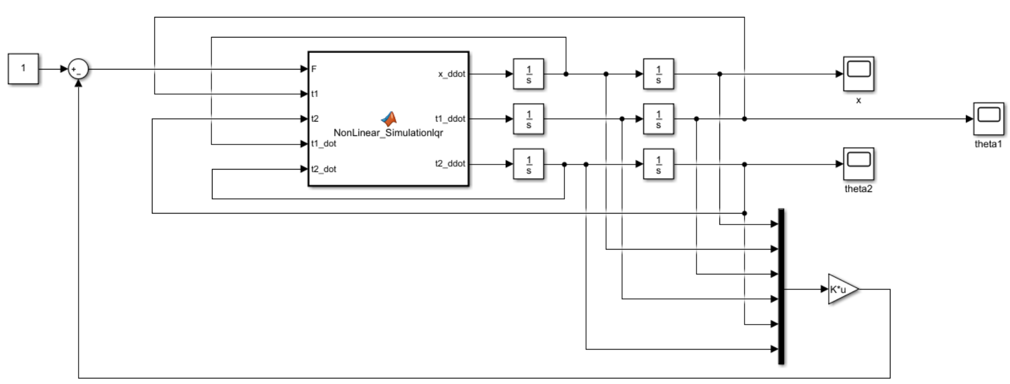

Figure 2: LQR Model for non-linear system

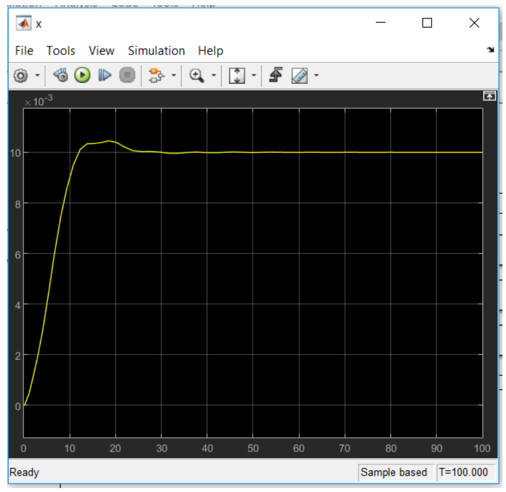

Figure 3: LQR response for cart-position vs time

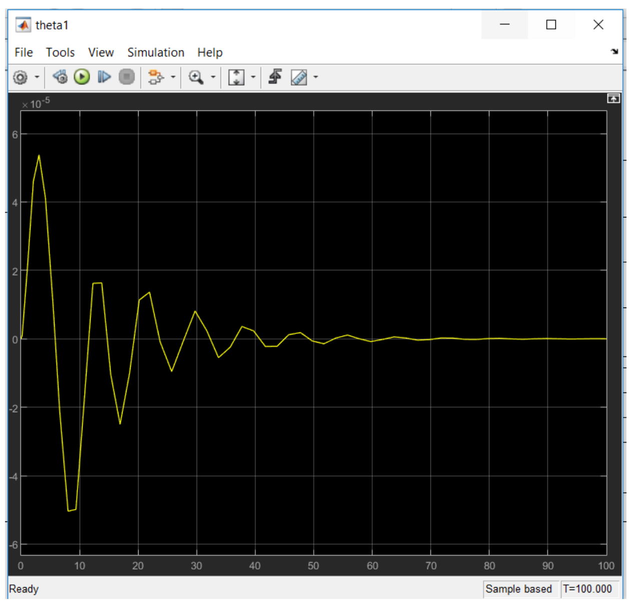

Figure 4: LQR response for pendulum angle(θ_2) vs time

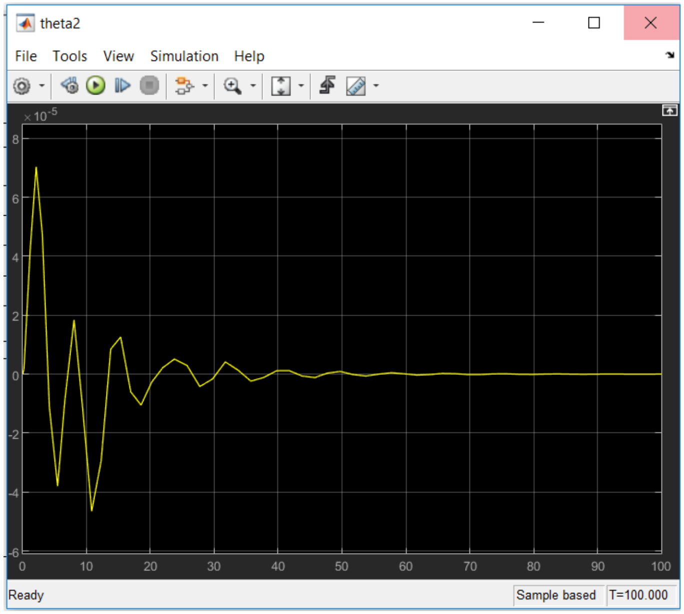

Figure 5: LQR response for pendulum angle(θ_2) vs time

By Lyapunov's Indirect method the system is atleast locally stable, because the real part of eigen values(found in MATLAB code) are negative(left half place).

Question E

```
syms m1 m2 m l1 l2 g
%% A matrix
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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];

%% Calculating C matric for different outputs
C1=[1 0 0 0 0 0];                  % C matrxi for X as output
C2=[1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];                  % C matrxi for X,theta1,theta2 as output
C3=[0 0 1 0 0 0; 0 0 0 0 1 0];     % C matrxi for theta1,theta2 as output
C4=[1 0 0 0 0 0; 0 0 0 0 1 0];     % C matrxi for X,theta2 as output
%% Check for Observability of output - X
co1=[C1 ;C1*A; C1*A^2; C1*A^3; C1*A^4; C1*A^5];
rank(co1)
%Thus it is observable since rank is 6.
%% Check for Observability of output - X,theta1,theta2
co2=[C2 ;C2*A; C2*A^2; C2*A^3; C2*A^4; C2*A^5];
rank(co2)
%Thus it is observable since rank is 6.

%% Check for Observability of output - theta1,theta2
co3=[C3 ;C3*A; C3*A^2; C3*A^3; C3*A^4; C3*A^5];
rank(co3)
%Thus it is not observable since rank is 4.

%% Check for Observability of output - X,theta2
co4=[C4 ;C4*A; C4*A^2; C4*A^3; C4*A^4; C4*A^5];
rank(co4)
%Thus it is observable since rank is 6.
```

Question F

Simulation of best observer for Output Vector x(t)

```
%% Best Observer for X as output
%% Parameters already given
m1=100;
m2=100;
m=1000;
l1=20;
l2=10;
g=9.8;
%% State 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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];
B=[0; 1/m; 0; 1/(m*l1); 0; 1/(m*l2)];
C=[1 0 0 0 0 0];
D=0;
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'}; %states
inputs = {'F'};                                     % intput
outputs = {'x'};                                    % output
%% Finding the gain matrix from LQR
Q = [1000 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 1000000 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 1000000 0;
    0 0 0 0 0 0]  %LQR Input
R = 0.1 ;                                          %LQR Input
K = lqr(A,B,Q,R)                  %Gain Calculation from LQR
Ac = [(A-(B*K))];
Bc = [B];
Cc = [C];
Dc = [D];
l=eig(Ac)                                             %eigenvalues of cloed loop
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};   %states
inputs = {'f'};                                       % intput after lqr
outputs = {'x'};                                      % output after lqr
sys_c2 = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
```

```
f = 50*ones(size(t));
[y,t,x]=lsim(sys_c2,f,t);  %simulates response
plot(t,y(:,1));
%% Finding 'best' observer matrix
P = 10*[l.']                         %finding the best poles
L = place(A',C',P)'     %Values of observer are found using  pole placment
Al = [(A-(L*C))];
Bl = [B];
Cl = [C];
Dl = [D];
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};  %states
inputs = {'f'};                                       % intput for observer
outputs = {'x'};                                      % output after observe
sys_c2 = ss(Al,Bl,Cl,Dl,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
f = 50*ones(size(t));
[y,t,x]=lsim(sys_c2,f,t);  %simulates response
plot(t,y(:,1));            %plotting the response
title('Best Observer');
```

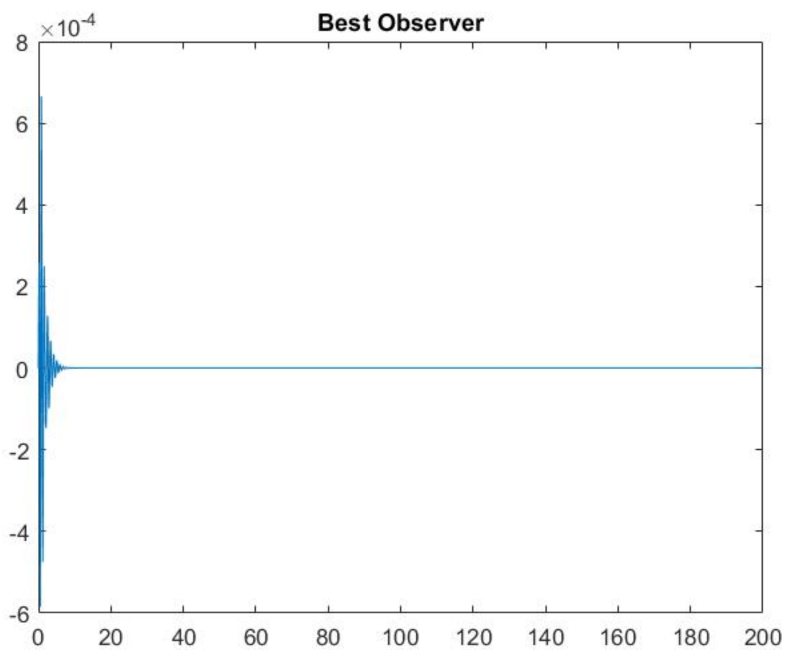

Figure 6: Output Vector x(t) :Best Luenberger

Simulation of best observer for Output Vector x(t),$\theta_2(t)$

```
%% Best Observer for X,theta2 as output
%% Parameters already given

m1=100;
m2=100;
m=1000;
l1=20;
l2=10;
g=9.8;
%% State 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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];
B=[0; 1/m; 0; 1/(m*l1); 0; 1/(m*l2)];
C=[1 0 0 0 0 0; 0 0 0 0 1 0];
D=0;
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'}; %states
inputs = {'F'};                                     % intput
outputs = {'x' ;'p2'};                            % output
%% Finding the gain matrix from LQR
%%
Q = [1000 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 1000000 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 1000000 0;
    0 0 0 0 0 0]                        %LQR parameters
R = 0.1 ;                               %LQR Input
K = lqr(A,B,Q,R)                        %Gain Calculation from LQR
Ac = [(A-(B*K))];
Bc = [B];
Cc = [C];
Dc = [D];
l=eig(Ac)                                        %eigenvalues of closed loop
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};    %states
inputs = {'f'};                                        % intput for lqr
outputs = {'x' ;'p2'};                                 % output after lqr
sys_cl = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
```

```
f = 50*ones(size(t));
[y,t,x]=lsim(sys_cl,f,t);   %simulates  response
[AX,H1,H2] = plotyy(t,y(:,1),[t,t],[y(:,2)],'plot');
%% Finding 'best' observer matrix
%%
P = 10*[l.']
L = place(A',C',P)'
Al = [(A-(L*C))];
Bl = [B];
Cl = [C];
Dl = [D];
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};   %states
inputs = {'f'};                                     % intput for observer
outputs = {'x' ;'p2'};                              % output after obserever
sys_c2 = ss(Al,Bl,Cl,Dl,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
f = 50*ones(size(t));
[y,t,x]=lsim(sys_c2,f,t); %simulates response
[AX,H1,H2] = plotyy(t,y(:,1),[t,t],[y(:,2)],'plot');%plotting the response
title('Best Observer');
```

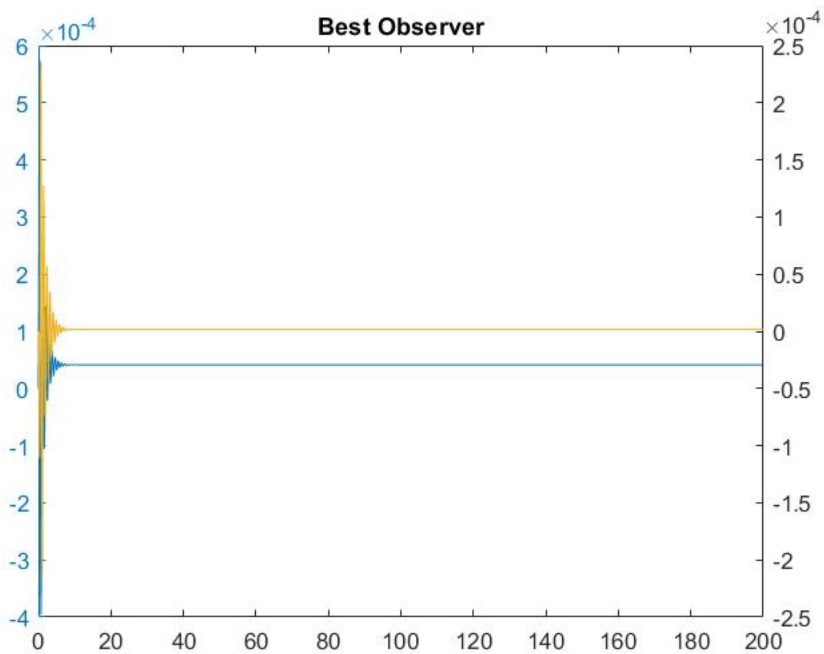

Figure 7: Output Vector x(t),$\theta_2(t)$:Best Luenberger

Output Vector x(t),$\theta_1(t)$,$\theta_2(t)$:Best Luenberger

```
%% Best Observer for X,theta1,theta2 as output
%% Parameters already given

m1=100;
m2=100;
m=1000;
l1=20;
l2=10;
g=9.8;
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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];
B=[0; 1/m; 0; 1/(m*l1); 0; 1/(m*l2)];
C=[1 0 0 0 0 0; 0 0 1 0 0 0;0 0 0 0 1 0];
D=0;
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};  %states
inputs = {'F'};                                      % intput
outputs = {'x' ;'p1';'p2'};                        % output
%% Finding the gain matrix from LQR
Q = [1000 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 1000000 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 1000000 0;
    0 0 0 0 0 0]      %LQR Input
R = 0.1 ;                     %LQR Input
K = lqr(A,B,Q,R)             %Gain Calculation from LQR
Ac = [(A-(B*K))];
Bc = [B];
Cc = [C];
Dc = [D];
l=eig(Ac)                                            %eigenvalues of
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};  %states
inputs = {'f'};                       % intput for lqr
outputs = {'x';'p1';'p2'};      % output after lqr
sys_cl = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
f = 50*ones(size(t));
[y,t,x]=lsim(sys_cl,f,t); %simulates response
[AX,H1,H2] = plotyy(t,y(:,1),[t,t],[y(:,2),y(:,3)],'plot');
%% Finding 'best' observer matrix
```

```
%%
P = 10*[l.']                          %finding the best poles
L = place(A',C',P)' %Values of observer are found using  pole placment
Al = [(A-(L*C))];
Bl = [B];
Cl = [C];
Dl = [D];
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};  %states
inputs = {'f'};                                       % intput for observer
outputs = {'x';'p1';'p2'};                          % output after observer
sys_c2 = ss(Al,Bl,Cl,Dl,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
f = 50*ones(size(t));
[y,t,x]=lsim(sys_c2,f,t); %simulates response
[AX,H1,H2] = plotyy(t,y(:,1),[t,t],[y(:,2),y(:,3)],'plot'); %Plotting the response
title('Best Observer');
```

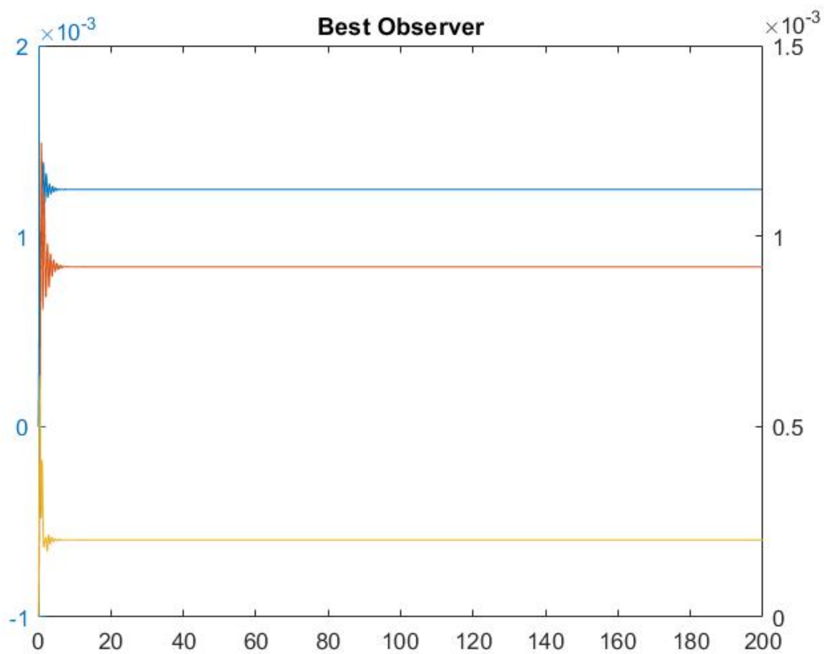

Figure 8: Output Vector x(t),$\theta_1(t)$,$\theta_2(t)$:Best Luenberger

Simulation of best observer nonlinear system

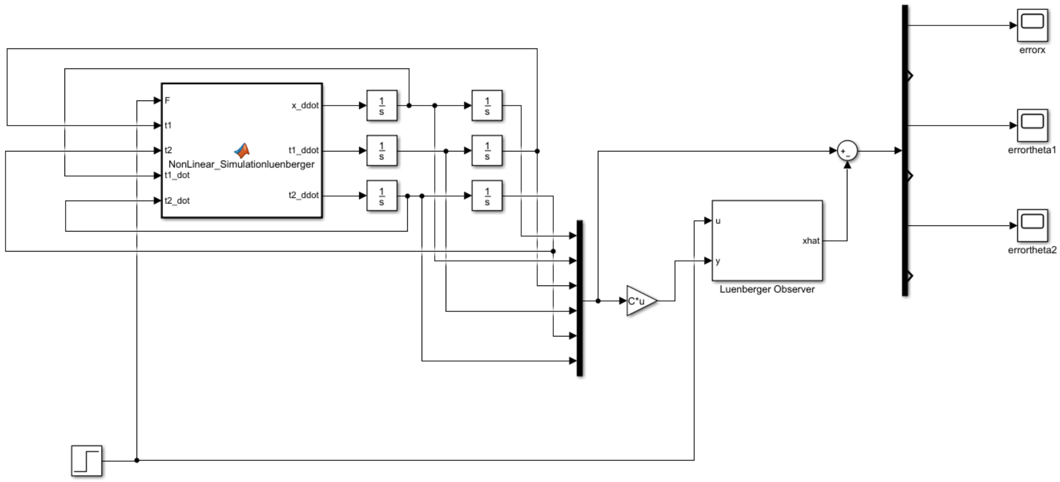

Figure 9: Simulation of best observer nonlinear system

Simulation of best observer nonlinear system: x(t)

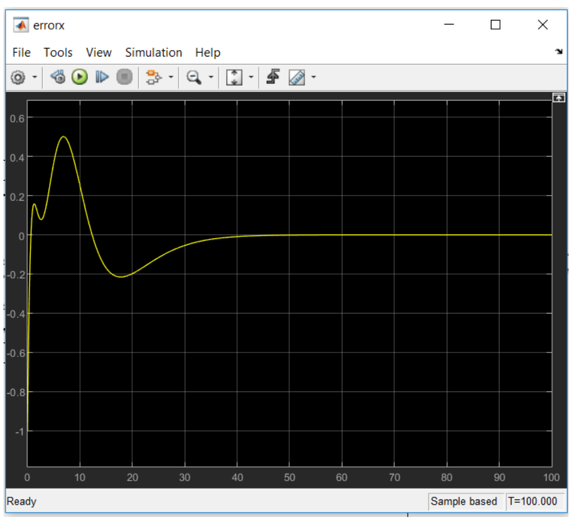

Figure 10: Simulation of best observer nonlinear system: x(t)

Simulation of best observer nonlinear system: x(t),$\theta_2(t)$

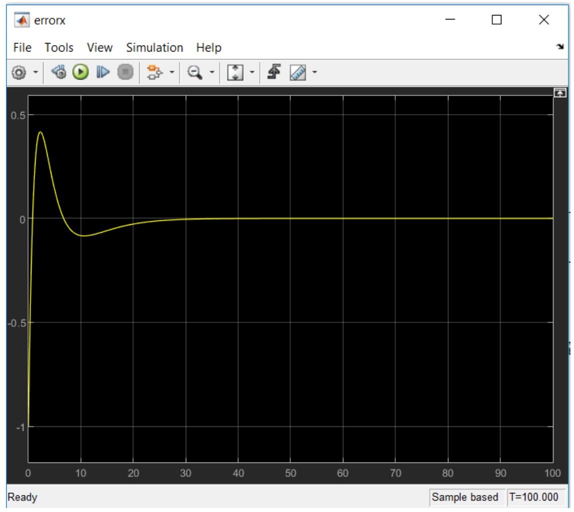

Figure 11: Simulation of best observer nonlinear system: x(t)

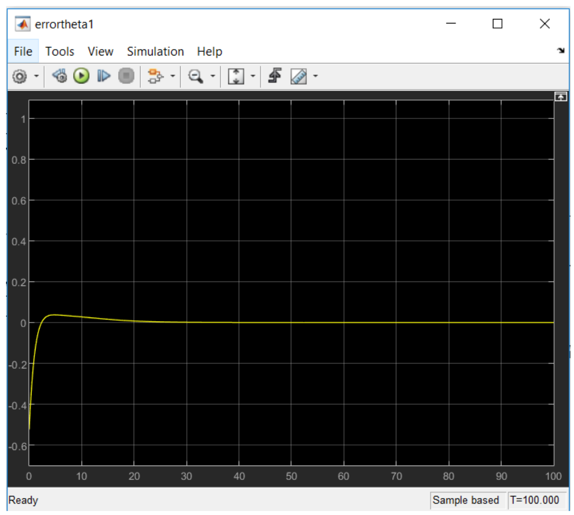

Figure 12: Simulation of best observer nonlinear system: $\theta_2(t)$

Simulation of best observer nonlinear system: x(t),$\theta_1(t)$,$\theta_2(t)$

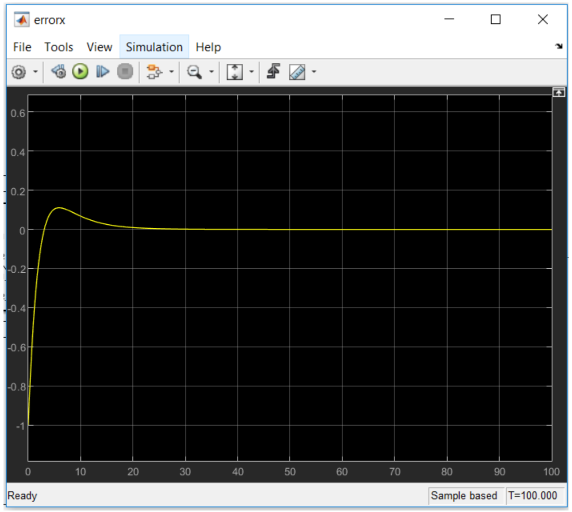

Figure 13: Simulation of best observer nonlinear system: x(t)

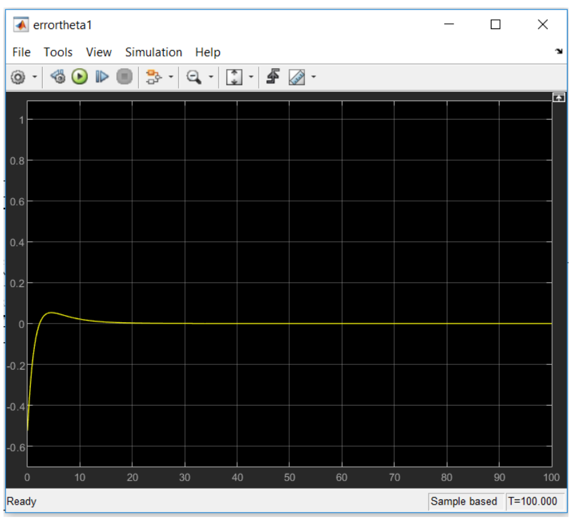

Figure 14: Simulation of best observer nonlinear system: $\theta_1(t)$

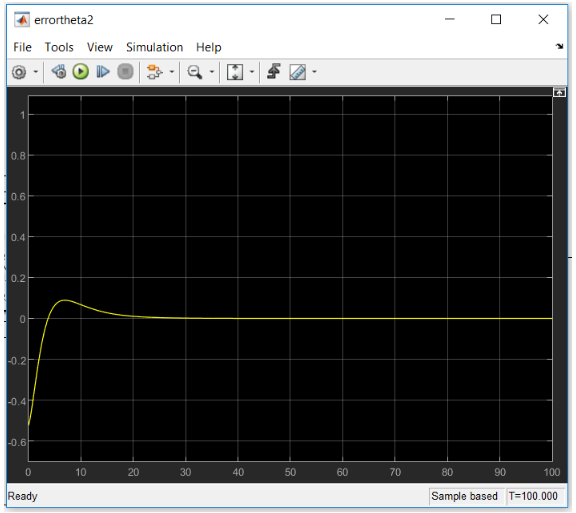

Figure 15: Simulation of best observer nonlinear system: $\theta_2(t)$

Question G

Linearized system : LQG

```
%% Best Observer for X as output
%% Parameters already given
m1=100;
m2=100;
m=1000;
l1=20;
l2=10;
g=9.8;
%% State 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 (-(m1*g)/(m*l1) -(g/l1)) 0 -(m2*g)/(m*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(m*l2) 0 (-(m2*g)/(m*l2) -(g/l2)) 0 ];
B=[0; 1/m; 0; 1/(m*l1); 0; 1/(m*l2)];
C=[1 0 0 0 0 0];
D=0;
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'}; %states
inputs = {'F'};                                     % intput
outputs = {'x'};                                    % output
%% Finding the gain matrix from LQR
Q = [1000 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 1000000 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 1000000 0;
    0 0 0 0 0 0]  %LQR Input
R = 0.1 ;                                          %LQR Input
K = lqr(A,B,Q,R)                   %Gain Calculation from LQR
Ac = [(A-(B*K))];
Bc = [B];
Cc = [C];
Dc = [D];
l=eig(Ac)                                      %eigenvalues of cloed loop
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};   %states
inputs = {'f'};                                       % intput after lqr
outputs = {'x'};                                      % output after lqr
sys_c2 = ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
```

```
f = 50*ones(size(t));
[y,t,x]=lsim(sys_c2,f,t);  %simulates response
plot(t,y(:,1));
%% Finding 'best' observer matrix
P = 10*[l.']                      %finding the best poles
L = place(A',C',P)'     %Values of observer are found using  pole placment
Al = [(A-(L*C))];
Bl = [B];
Cl = [C];
Dl = [D];
states = {'x' 'x_dot' 'p1' 'p1_dot' 'p2' 'p2_dot'};  %states
inputs = {'f'};                                      % intput for observer
outputs = {'x'};                                     % output after observe
sys_c2 = ss(Al,Bl,Cl,Dl,'statename',states,'inputname',
inputs,'outputname',outputs); %creates statesspace model
t = 0:0.1:200;
f = 50*ones(size(t));
[y,t,x]=lsim(sys_c2,f,t);  %simulates response
plot(t,y(:,1));            %plotting the response
title('Best Observer');
%% lQG for smallest vector.
Cn = [1 0 0 0 0 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K)
Ace = [(A-B*K) (B*K);
       zeros(size(A)) (A-L*C)];
Bce = [B*Nbar;
       zeros(size(B))];
Cce = [Cc zeros(size(Cc))];
Dce = [0];
states = {'x' 'x_dot' 'theta' 'theta1_dot'
    'theta2' 'theta2_dot' 'e1' 'e2' 'e3' 'e4' 'e5' 'e6'};
inputs = {'r'};
outputs = {'x'};
sys_est_cl = ss(Ace,Bce,Cce,Dce,'statename',states,
'inputname',inputs,'outputname',outputs);
t = 0:0.1:500;
r = 50*ones(size(t));
[y,t,x]=lsim(sys_est_cl,r,t);
plot(t,y)  %plotting lqg
title('LQG for Smallest Output Vector: x')
```

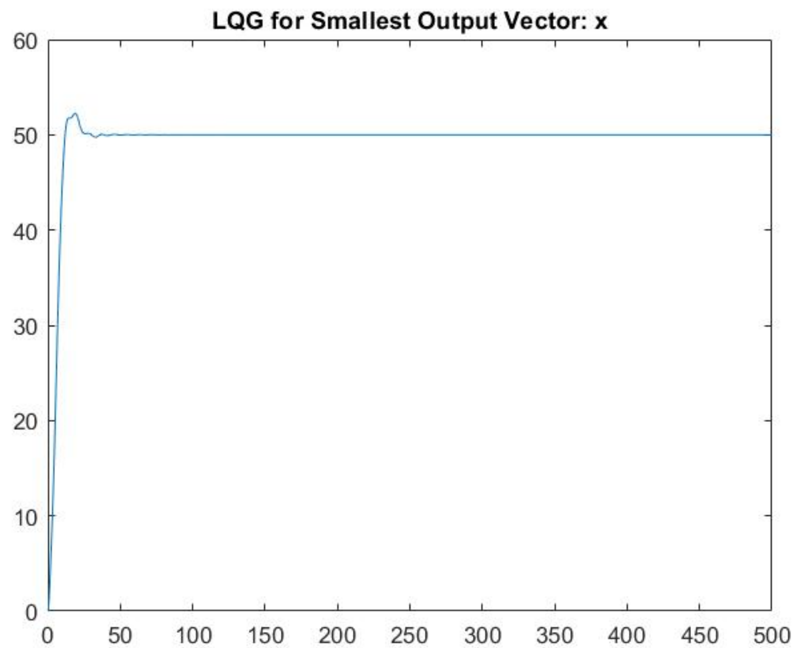

Figure 16: Smallest output vector x(t) : LQG for linearized model

Non-linearized system : LQG

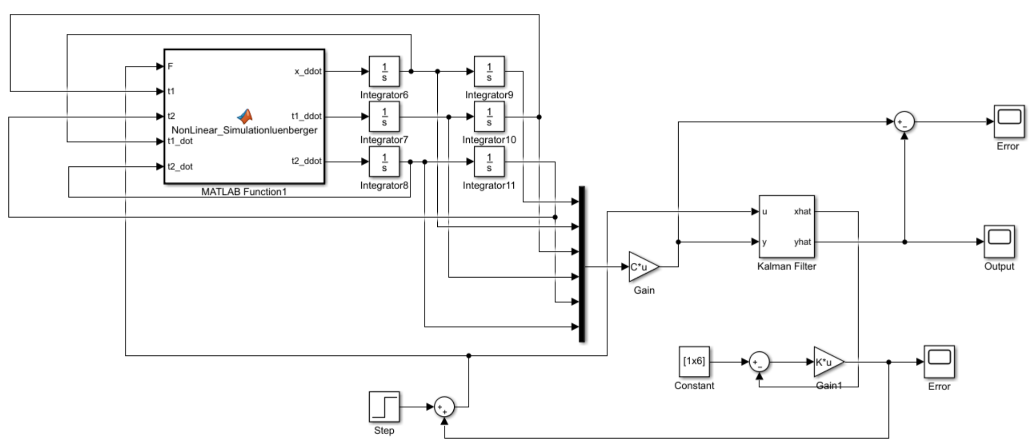

Figure 17: Simulink Model of LQG

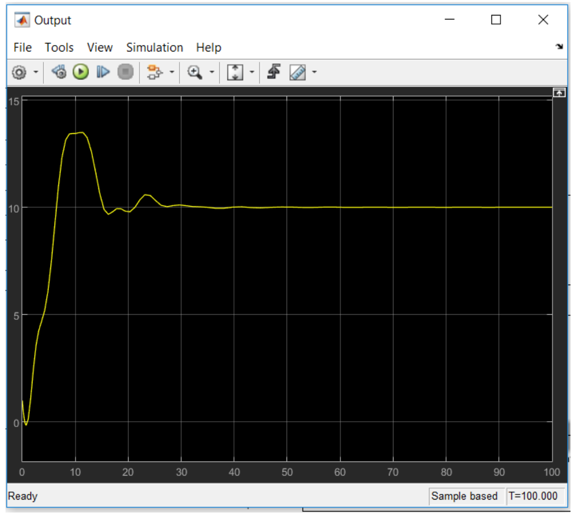

Figure 18: LQG output for smallest vector x(t)

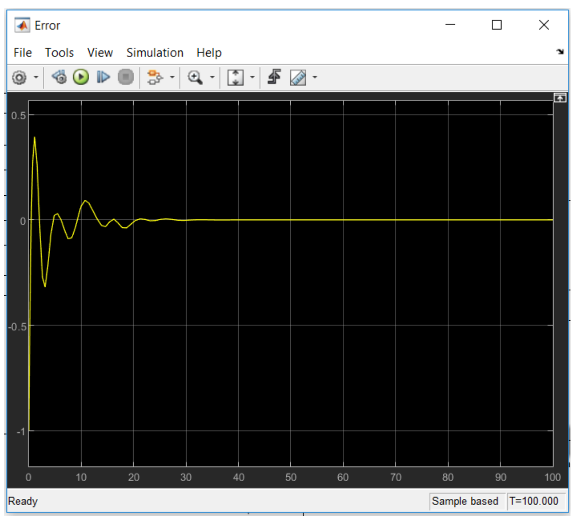

Figure 19: Reference tracking : Error

Reconfiguration of Controller to asymptotically track a constant reference on x(t)

Cost Function

$$CF_i = \int X^T(t)QX(t) + U_k^T(t)RU_k(t)$$

which changes to

$$CF_f = \int (X(t) - X_d)^T Q(X(t) - X_d) + (U_k(t) - U_\infty)^T R(U_k(t) - U_\infty) \tag{16}$$

We will get U_∞ from

$$AX_d + BU_\infty = 0 \tag{17}$$

and initially

$$U(t) = KX(t)$$

Now it changes to

$$U(t) = K(X(t) - X_d) + U_\infty \tag{18}$$

No, our design will not be able to reject constant forces, disturbances applied on the cart. In order to reject constant disturbances we need to augment our state with an integral term in it. This augmentation is a variant of LQR known as LQRI (Linear Quadratic Regulator with an Integral Term).