ENPM667 Project 2
Design of LQR and LQG Controllers
for a Crane Double Pendulum System

ADITYA VARADARAJ

(UID: 117054859)

SAURABH PRAKASH PALANDE

(UID: 118133959)

(SECTION: 0101)

M.Eng. Robotics (PMRO),

University of Maryland - College Park, College Park, MD - 20742, USA.

Date of Submission: 19th December, 2021

List of Figures

CONTENTS

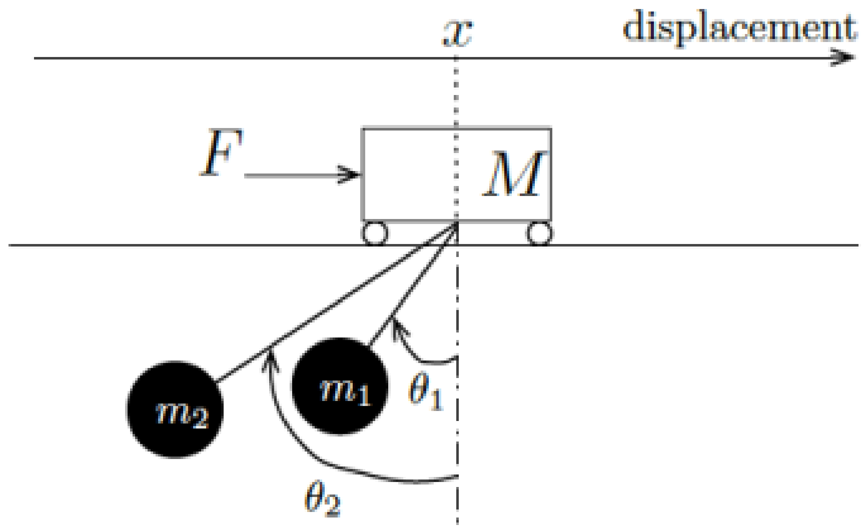


Fig. 1. Cart-Double Pendulum Crane System [1]

I. Question 1

A. Equations of motion and Nonlinear State Space Equation

From Fig. 1, we can write the position vectors as:

$$\begin{aligned}
r_1(t) &= (x - l_1 sin(\theta_1))\ i - (l_1 cos(\theta_1))\ j \\
r_2(t) &= (x - l_2 sin(\theta_2))\ i - (l_2 cos(\theta_2))\ j \\
\dot{r_1}(t) &= (\dot{x} - l_1 cos(\theta_1)\dot{\theta_1})\ i + l_1 sin(\theta_1)\dot{\theta_1}\ j \\
\dot{r_2}(t) &= (\dot{x} - l_2 cos(\theta_2)\dot{\theta_2})\ i + l_2 sin(\theta_2)\dot{\theta_2}\ j
\end{aligned} \tag{1}$$

Thus, we can write the Kinetic Energy, Potential Energy and Lagrangian as:

$$\begin{aligned}
K &= \frac{1}{2}M\dot{x}^2\ +\ \frac{1}{2}m_1(\dot{x} - l_1 cos(\theta_1)\dot{\theta_1})^2\ +\ \frac{1}{2}m_1(l_1\dot{\theta_1})^2 sin^2(\theta_1) \\
&\quad +\ \frac{1}{2}m_2(\dot{x} - l_2 cos(\theta_2)\dot{\theta_2})^2\ +\ \frac{1}{2}m_2(l_2\dot{\theta_2})^2 sin^2(\theta_2) \\
P &= Mgl_2\ +\ m_1 g(l_1 - l_1 cos(\theta_1))\ +\ m_2 g(l_1 - l_2 cos(\theta_2)) \\
L &= K\ -\ P,\ where,\ L\ is\ Lagrangian
\end{aligned} \tag{2}$$

Using Lagrangian method for x, we can write,

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

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

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

$$\frac{d}{dt}\frac{\delta L}{\delta \dot{x}} \ - \ \frac{\delta L}{\delta x} = F$$

$$(M + m_1 + m_2)\ddot{x} \ - \ m_1 l_1 cos(\theta_1)\ddot{\theta}_1 \ - \ m_2 l_2 cos(\theta_2)\ddot{\theta}_2 \ + \ m_1 l_1 sin(\theta_1)\dot{\theta}_1^{\ 2} \ + \ m_2 l_2 sin(\theta_2)\dot{\theta}_2^{\ 2} = F \tag{3}$$

Similarly, applying lagrangian method for θ_1, we can write:

$$\begin{aligned}
&\frac{\delta L}{\delta \dot{\theta}_1} = -m_1(\dot{x} - l_1 cos(\theta_1)\dot{\theta}_1) l_1 cos(\theta_1) \ + \ m_1 l_1^2 \dot{\theta}_1 sin^2(\theta_1) \\
&\frac{d}{dt}\frac{\delta L}{\delta \dot{\theta}_1} = -m_1\ddot{x} l_1 cos(\theta_1) \ + \ m_1\dot{x} l_1 sin(\theta_1)\dot{\theta}_1 \ + \ m_1 l_1^2 \ddot{\theta}_1 \\
&\frac{\delta L}{\delta \theta_1} = m_1 l_1 sin(\theta_1)(\dot{x}\dot{\theta}_1 - g) \\
&\frac{d}{dt}\frac{\delta L}{\delta \dot{\theta}_1} \ - \ \frac{\delta L}{\delta \theta_1} = 0 \\
&- m_1\ddot{x} l_1 cos(\theta_1) + m_1 l_1^2 \ddot{\theta}_1 + m_1 g l_1 sin(\theta_1) = 0
\end{aligned} \tag{4}$$

Similarly, by using Lagrangian Method on θ_2, we get:

$$\begin{aligned}
&\frac{d}{dt}\frac{\delta L}{\delta \dot{\theta}_2} \ - \ \frac{\delta L}{\delta \theta_2} = 0 \\
&- m_2\ddot{x} l_2 cos(\theta_2) + m_2 l_2^2 \ddot{\theta}_2 + m_2 g l_2 sin(\theta_2) = 0
\end{aligned} \tag{5}$$

By rearranging terms in equations Eq. 4 and Eq. 5, we can write:

$$\begin{aligned}
m_1 l_1 \ddot{\theta}_1 &= m_1\ddot{x} cos(\theta_1) - m_1 g sin(\theta_1) \\
m_2 l_2 \ddot{\theta}_2 &= m_2\ddot{x} cos(\theta_2) - m_2 g sin(\theta_2)
\end{aligned} \tag{6}$$

Plugging equations Eq.6 into Eq.3, we get:

$$\begin{aligned}
\ddot{x} = \left[\frac{1}{M + m_1 sin^2(\theta_1) + m_2 sin^2(\theta_2)}\right] &(F \ - \ m_1 g sin(\theta_1) cos(\theta_1) \ - \ m_2 g sin(\theta_2) cos(\theta_2) \\
&- \ m_1 l_1 sin(\theta_1)\dot{\theta}_1^{\ 2} \ - \ m_1 l_1 sin(\theta_1)\dot{\theta}_1^{\ 2})
\end{aligned} \tag{7}$$

Substituting this value into Eq. 6, we get:

$$\ddot{\theta}_1 = \left[\frac{cos(\theta_1)}{l_1(M + m_1 sin^2(\theta_1) + m_2 sin^2(\theta_2))}\right](F \; - \; m_1 g sin(\theta_1)cos(\theta_1) \; - \; m_2 g sin(\theta_2)cos(\theta_2)$$
$$- \; m_1 l_1 sin(\theta_1)\dot{\theta_1}^2 \; - \; m_1 l_1 sin(\theta_1)\dot{\theta_1}^2) \; - \; \frac{g sin(\theta_1)}{l_1}$$
$$\ddot{\theta}_2 = \left[\frac{cos(\theta_2)}{l_2(M + m_1 sin^2(\theta_1) + m_2 sin^2(\theta_2))}\right](F \; - \; m_1 g sin(\theta_1)cos(\theta_1) \; - \; m_2 g sin(\theta_2)cos(\theta_2)$$
$$- \; m_1 l_1 sin(\theta_1)\dot{\theta_1}^2 \; - \; m_1 l_1 sin(\theta_1)\dot{\theta_1}^2) \; - \; \frac{g sin(\theta_2)}{l_2} \tag{8}$$

B. Linearization of Nonlinear System

We can write the state-space equation as:

$$\dot{X} = F(X(t), U(t))$$
$$Using \ Jacobian \ Linearization \ we \ can \ write:$$
$$A_F = \nabla_x | F(X(t), U(t))$$
$$B_F = \nabla_u | F(X(t), U(t))$$

$$A_F = \begin{bmatrix} \frac{\delta f_1}{\delta x} & \frac{\delta f_1}{\delta \dot{x}} & \frac{\delta f_1}{\delta \theta_1} & \frac{\delta f_1}{\delta \dot{\theta_1}} & \frac{\delta f_1}{\delta \theta_2} & \frac{\delta f_1}{\delta \dot{\theta_2}} \\ \frac{\delta f_2}{\delta x} & \frac{\delta f_2}{\delta \dot{x}} & \frac{\delta f_2}{\delta \theta_1} & \frac{\delta f_2}{\delta \dot{\theta_1}} & \frac{\delta f_2}{\delta \theta_2} & \frac{\delta f_2}{\delta \dot{\theta_2}} \\ \frac{\delta f_3}{\delta x} & \frac{\delta f_3}{\delta \dot{x}} & \frac{\delta f_3}{\delta \theta_1} & \frac{\delta f_3}{\delta \dot{\theta_1}} & \frac{\delta f_3}{\delta \theta_2} & \frac{\delta f_3}{\delta \dot{\theta_2}} \\ \frac{\delta f_4}{\delta x} & \frac{\delta f_4}{\delta \dot{x}} & \frac{\delta f_4}{\delta \theta_1} & \frac{\delta f_4}{\delta \dot{\theta_1}} & \frac{\delta f_4}{\delta \theta_2} & \frac{\delta f_4}{\delta \dot{\theta_2}} \\ \frac{\delta f_5}{\delta x} & \frac{\delta f_5}{\delta \dot{x}} & \frac{\delta f_5}{\delta \theta_1} & \frac{\delta f_5}{\delta \dot{\theta_1}} & \frac{\delta f_5}{\delta \theta_2} & \frac{\delta f_5}{\delta \dot{\theta_2}} \\ \frac{\delta f_6}{\delta x} & \frac{\delta f_6}{\delta \dot{x}} & \frac{\delta f_6}{\delta \theta_1} & \frac{\delta f_6}{\delta \dot{\theta_1}} & \frac{\delta f_6}{\delta \theta_2} & \frac{\delta f_6}{\delta \dot{\theta_2}} \end{bmatrix}, \qquad B_F = \begin{bmatrix} \frac{\delta f_1}{\delta F} \\ \frac{\delta f_2}{\delta F} \\ \frac{\delta f_3}{\delta F} \\ \frac{\delta f_4}{\delta F} \\ \frac{\delta f_5}{\delta F} \\ \frac{\delta f_6}{\delta F} \end{bmatrix} \tag{9}$$

After Linearizing about equilibrium point, i.e., $\begin{bmatrix} x \\ x_dot \\ \theta_1 \\ \dot{\theta_1} \\ \theta_2 \\ \dot{\theta_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, we get the State Space Equation as:

$$
\begin{aligned}
\dot{X}(t) \;&=\; A_F X(t) \;+\; B_F U(t) \\
A_F &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m_1 g}{M} & 0 & \frac{-m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & (-g - \frac{m_1 g}{M})\frac{1}{l_1} & 0 & \frac{-m_2 g}{M l_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{M l_2} & 0 & (-g - \frac{m_1 g}{M})\frac{1}{l_1} & 0 \end{bmatrix} \\
B_F &= \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M l_1} \\ 0 \\ \frac{1}{M l_2} \end{bmatrix}
\end{aligned}
\tag{10}
$$

$$
Substituting\ M = 1000kg,\ m_1 = m_2 = 100kg, l_1 = 20m, l_2 = 10m,\ we\ get:
$$

$$
A_F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -11/20 & 0 & -1/20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1/10 & 0 & -11/10 & 0 \end{bmatrix},\ B_F = \begin{bmatrix} 0 \\ \frac{1}{1000} \\ 0 \\ \frac{1}{20000} \\ 0 \\ \frac{1}{10000} \end{bmatrix}
$$

C. Controllability Conditions

Since the system is linearized and time-invariant system, we can use the rank condition. The rank matrix is $rank[B\ AB\ A^2B\ A^3B\ A^4B\ A^5B]$ since n=6. For system to be controllable, rank should be 6, i.e., there should be 6 linearly independent column vectors. We can calculate the matrices using Sympy in Python as:

$$B = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} \quad AB = \begin{bmatrix} \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \\ 0 \end{bmatrix}$$

$$A^2B = \begin{bmatrix} 0 \\ \frac{1}{M}(-\frac{m_2 g}{Ml_2} - \frac{m_1 g}{Ml_2}) \\ 0 \\ \frac{1}{Ml_1}(\frac{-g - \frac{m_1 g}{M}}{l_1} - \frac{m_2 g}{Ml_2}) \\ 0 \\ \frac{1}{Ml_2}(\frac{-g - \frac{m_2 g}{M}}{l_2} - \frac{m_1 g}{Ml_1}) \end{bmatrix} \quad A^3B = \begin{bmatrix} \frac{1}{M}(-\frac{m_2 g}{Ml_2} - \frac{m_1 g}{Ml_2}) \\ 0 \\ \frac{1}{Ml_1}(\frac{-g - \frac{m_1 g}{M}}{l_1} - \frac{m_2 g}{Ml_2}) \\ 0 \\ \frac{1}{Ml_2}(\frac{-g - \frac{m_2 g}{M}}{l_2} - \frac{m_1 g}{Ml_1}) \\ 0 \end{bmatrix} \tag{11}$$

Similarly, A^4B and A^5B also have different alternate columns filled up with the same elements. We can see that we have atleast 4 linearly independent column vectors. The condition on M, l_1, l_2, m_1, m_2 can be obtained by constraining the 3-vector containing non-zero rows of A^4B to be along the cross product of a 3-vector containing non-zero rows of B with a 3-vector containing non-zero rows of A^2B.

D. LQR (Linear Quadratic Regulator) Controller Design

We have considered the initial conditions as $[0, 0, 0, 0, \frac{\pi}{2}, 0]$.

1) LQR for Linearized System: First, we check whether the system is controllable or not using rank condition. We consider the output vector as $[x,\ \theta_1,\ \theta_2]^T$. We use the rank function in MATLAB. If the matrix $[B\ AB\ A^2B\ A^3B\ A^4B\ A^5B]$ is of rank 6 (full rank), then system is controllable. Else it is uncontrollable.

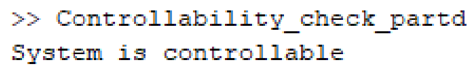


Fig. 2. Controllability Check

From the code shown below, we get the output as shown in Fig. 2

Code for Controllability Check:

```
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];

%Checking controllability of the system
contrallibility_matrix = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
if rank(contrallibility_matrix)==6
    disp('System is controllable')
else
    disp('System is not controllable')
end
```

Now, we can use LQR to get the system output. The output from the code given below is as shown in Fig. 3.

Code for LQR for Linearized System:

```
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
```

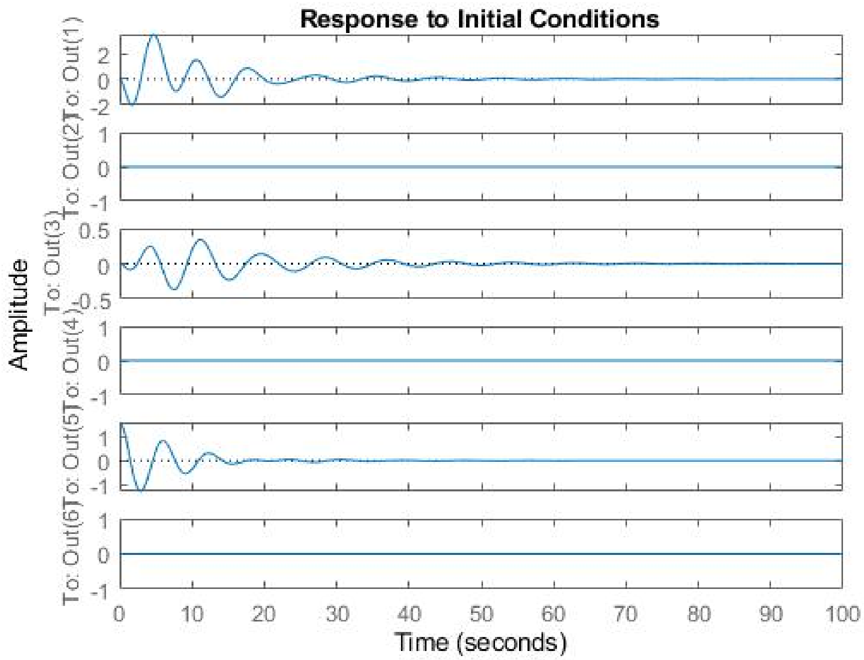


Fig. 3. Initial Conditions Response of Linearized System using LQR

```
     0  0  0  1  0  0;
     0  0  -11/20  0  -1/20  0;
     0  0  0  0  0  1;
     0  0  -1/10  0  -11/10  0];
disp("Eigenvalues of Linearized System without LQR Feedback:")
disp(eig(A));
B = [0; 1/1000; 0; 1/20000; 0; 1/10000];

C = [1 0 0 0 0 0;0 0 0 0 0 0; 0 0 1 0 0 0;0 0 0 0 0 0; 0 0 0 0 1
   0;0 0 0 0 0 0];
D = zeros(6,1);

Q = [1000  0  0  0  0  0;
     0  10  0  0  0  0;
     0  0  10000  0  0  0;
```

```
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
     0 0 0 0 0 100000];
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
X0 = [0; 0; 0; 0; 3.142/2; 0];
sys = ss((A-B*K), zeros(6,1), C,D);
disp("Eigenvalues after LQR Feedback:")
disp(eig(A-B*K));
initial(sys, X0);
```

We can use Lyapunov's Indirect Method to test the stability of the linearized system before and after LQR. Here, we check the eigenvalues of matrix A for uncontrolled system and $(A-BK)$ for LQR controlled system respectively. As we can see from Fig. 4, initially, the Lyapunov indirect method is inconclusive since real parts of eigenvalues are 0 and eigenvalues are on imaginary axis. Whereas, for the closed-loop system after applying LQR, we get the eigenvalues of $A-BK$ to be having negative real parts. Thus, the closed-loop system is atleast locally stable around equilibrium point after application of LQR.

```
>> LQR_Partd
Eigenvalues of Linearized System without LQR Feedback:
   0.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.7356i
   0.0000 - 0.7356i
   0.0000 + 1.0531i
   0.0000 - 1.0531i

Eigenvalues after LQR Feedback:
  -0.3529 + 0.3812i
  -0.3529 - 0.3812i
  -0.1699 + 1.0339i
  -0.1699 - 1.0339i
  -0.0616 + 0.7280i
  -0.0616 - 0.7280i
```

Fig. 4. Eigenvalues of A and $A-BK$

2) LQR for Original Nonlinear System: Now, for original Nonlinear system, we use the ode45 Ordinary Differential Equation Solver in MATLAB to apply LQR and find the initial conditions response of the system. Using the code below, we get the initial system response as shown in Fig. 5.

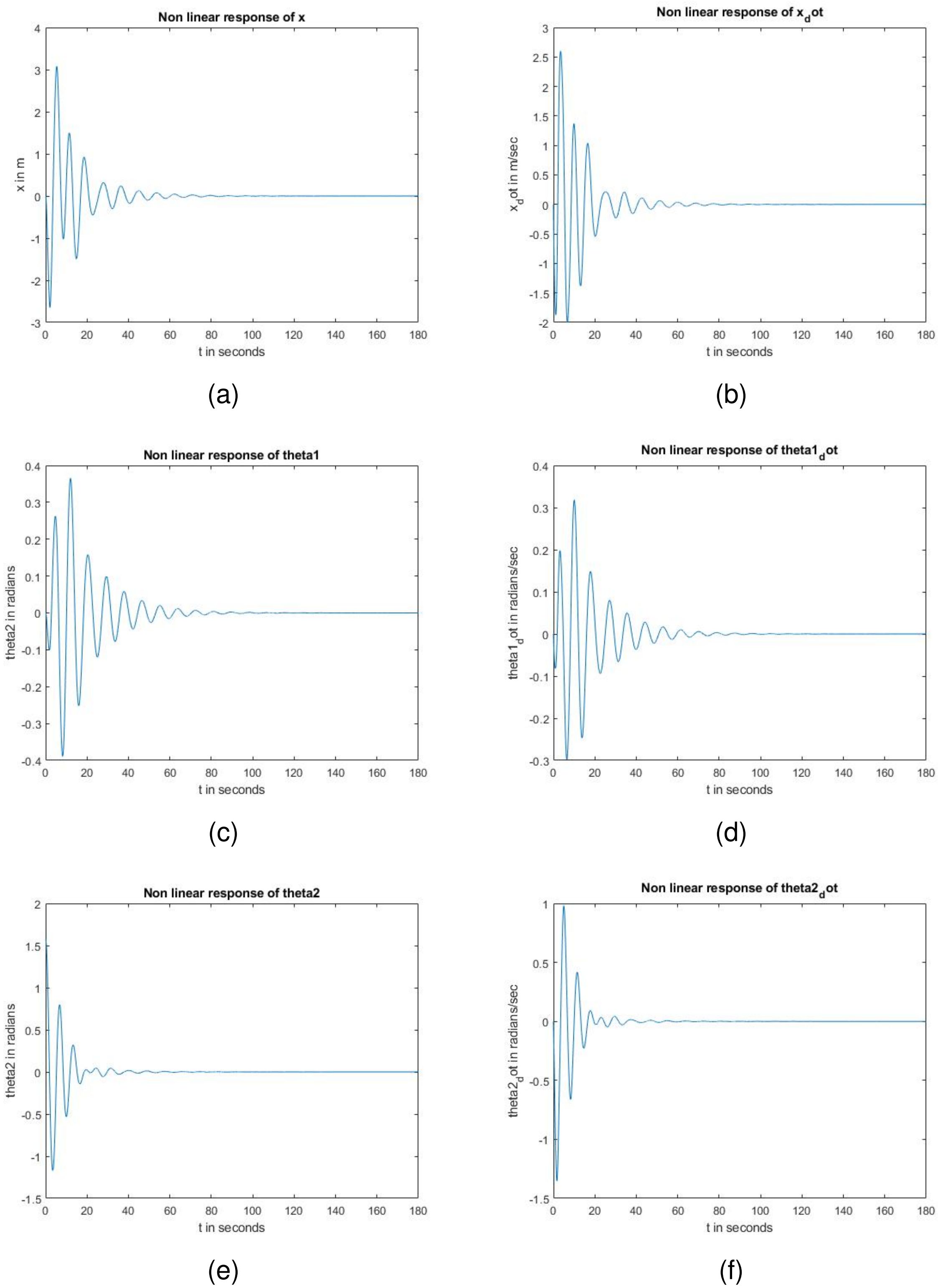


Fig. 5. LQR Initial Response of Original Nonlinear System: (a) x, (b) $\dot{x}$, (c) θ_1, (d) $\dot{\theta_1}$,(e) θ_2 and (f) $\dot{\theta_2}$

Code for LQR on Original Nonlinear System :

```
y0 = [0; 0; 0; 0; 3.14/2; 0];
tspan = 0:0.01:180;%defining the timespan
[t1,y1] = ode45(@non_linear,tspan,y0);
figure
plot(t1,y1(:,1))
title('Non linear response of x')
xlabel('t in seconds')
ylabel('x in m')
figure
plot(t1,y1(:,2))
title('Non linear response of x_dot')
xlabel('t in seconds')
ylabel('x_dot in m/sec')
figure
plot(t1,y1(:,3))
title('Non linear response of theta1')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,4))
title('Non linear response of theta1_dot')
xlabel('t in seconds')
ylabel('theta1_dot in radians/sec')
figure
plot(t1,y1(:,5))
title('Non linear response of theta2')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,6))
```

```
title('Non linear response of theta2_dot')
xlabel('t in seconds')
ylabel('theta2_dot in radians/sec')

function dydt = non_linear(t,y)
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];

Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
     0 0 10000 0 0 0;
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
     0 0 0 0 0 100000];
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
F = -K*y;
dydt=zeros(6,1);
dydt(1) = y(2); %x_dot
dydt(2)=(F-1000*sin(y(3))*cos(y(3))-1000*sin(y(5))*cos(y(5))
   -(2000*(y(4)^2)*sin(y(3)))-(1000*(y(6)^2)*sin(y(5))))
   /(1000+100*((sind(y(3)))^2)+100*((sind(y(5)))^2));%x_ddot
dydt(3)= y(4); %theta 1D
dydt(4)= (dydt(2)*cos(y(3))-10*(sin(y(3))))/20'; %theta 1 ddot;
dydt(5)= y(6); %theta 2D
```

```
dydt(6)= (dydt(2)*cos(y(5))-10*(sin(y(5))))/10; %theta 2 ddot;
end
```

E. Observability

We have considered the initial conditions as $[0, 0, 0, 0, \frac{\pi}{2}, 0]$. To check the observability of the system, we check the controllability of (A^T, C^T) using Rank condition. There are 4 cases of output vectors in the question:

- $x(t)$
- $(\theta_1(t), \theta_2(t))$
- $(x(t), \theta_2(t))$
- $(x(t), \theta_1(t), \theta_2(t))$

The output of the observability code below is as shown in Fig. 6.

```
>> Observability_check_parte
System is observable for output vector x(t)
System is not observable for output vector theta1 and theta2
System is observable for output vector x and theta2
System is observable for output vector x, theta1 theta2
```

Fig. 6. Observability Check

Code for Observability Check :

```
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];
C1 = [1 0 0 0 0 0]; %for x as output
C2 = [0 0 1 0 0 0; 0 0 0 0 1 0];
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %for x and theta2 as output
```

```
C4 = [1 0 0 0 0 0 ;0 0 1 0 0 0; 0 0 0 0 1 0]; % for x, theta1 and
    theta2 as output

%Checking observability for output vector x(t)
O1 = [C1' A'*C1' A'*A'*C1' A'*A'*A'*C1' A'*A'*A'*A'*C1' A'*A'*A'*
   A'*A'*C1'];
if rank(O1)==6
    disp('System is observable for output vector x(t)')
else
    disp('System is not observable for output vector x(t)')
end
%Checking observability for output vector theta1 and theta2
O2 = [C2' A'*C2' A'*A'*C2' A'*A'*A'*C2' A'*A'*A'*A'*C2' A'*A'*A'*
   A'*A'*C2'];
if rank(O2)==6
    disp('System is observable for output vector theta1 and
       theta2')
else
    disp('System is not observable for output vector theta1 and
       theta2')
end
%Checking observability for output vector x and theta2
O3 = [C3' A'*C3' A'*A'*C3' A'*A'*A'*C3' A'*A'*A'*A'*C3' A'*A'*A'*
   A'*A'*C3'];
if rank(O3)==6
    disp('System is observable for output vector x and theta2')
else
    disp('System is not observable for output vector x and theta2
       ')
end
%Checking observability for output vector x, theta1 and theta2
```

```
O4 = [C4' A'*C4' A'*A'*C4' A'*A'*A'*C4' A'*A'*A'*A'*C4' A'*A'*A'*
    A'*A'*C4'];
if rank(O4)==6
    disp('System is observable for output vector x, theta1 theta2
        ')
else
    disp('System is not observable for output vector x, theta1
        and theta2')
end
```

F. Best Luenberger Observer

We find the Best Luenberger Observer Gain matrix L using a pole placement function. The pole we have assumed are $[-1, -2, -3, -4, -5, -6]$. The controller gain matrix K is found using LQR. We consider the observable cases, i.e.,

- Case 1: Output is $x(t)$
- Case 2: Output is $(x(t), \theta_2(t))$
- Case 3: Output is $(x(t), \theta_1(t), \theta_2(t))$

1) For Linearized System: The output of the code given below, i.e., Initial and Step response of the system is given in Fig. 7 and Fig. 8.

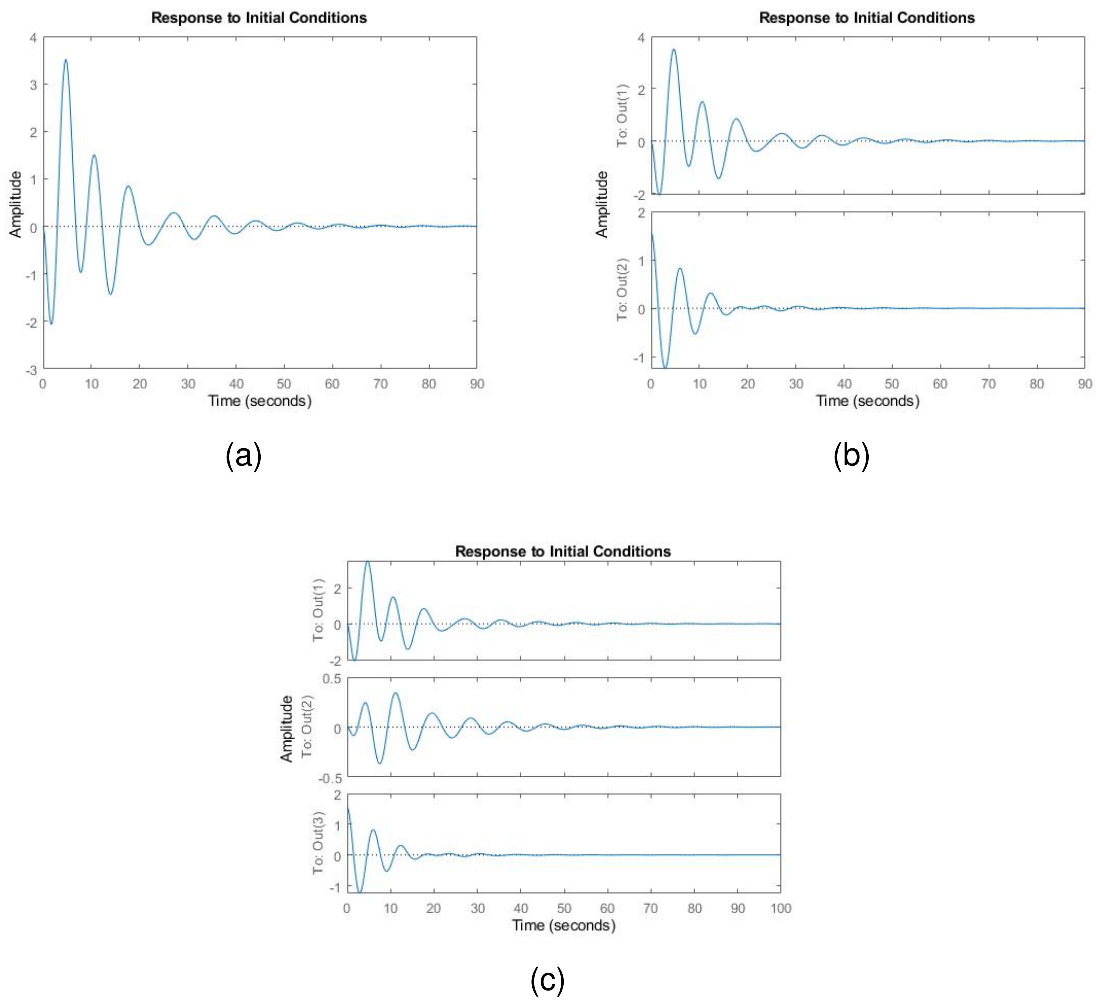
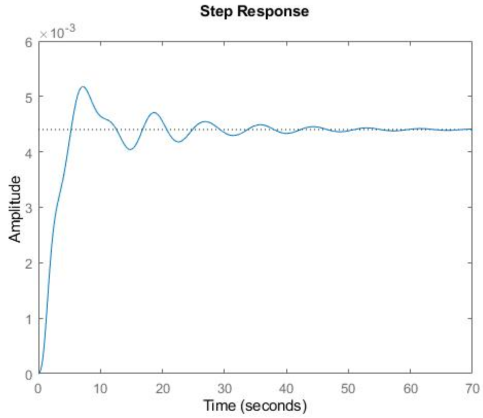


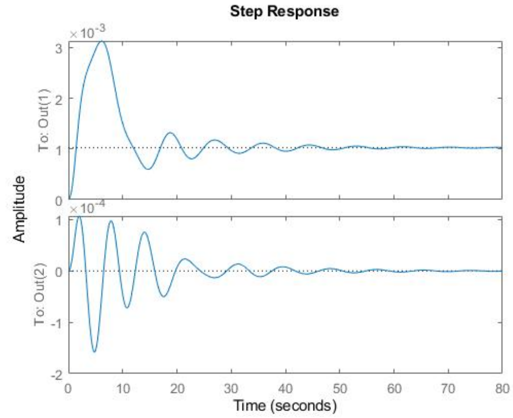
Fig. 7. Initial Response in (a) Case 1, (b) Case 2 and (c) Case 3 of Linearized System

Code for Linearized System:

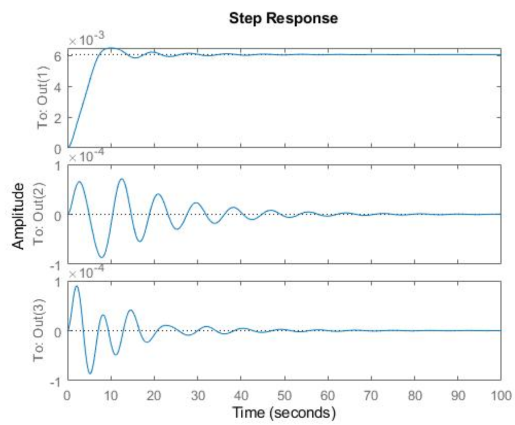
```
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];
C1 = [1 0 0 0 0 0]; %for x as output
```



(a)



(b)



(c)

Fig. 8. Step Response in (a) Case 1, (b) Case 2 and (c) Case 3 of Linearized System

```
C2 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %for x and theta2 as output
C3 = [1 0 0 0 0 0 ;0 0 1 0 0 0; 0 0 0 0 1 0]; % for x, theta1 and
    theta2 as output
D = zeros(1,1);

Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
     0 0 10000 0 0 0;
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
```

```
     0 0 0 0 0 100000];
R = 0.01*ones(1,1);


[K,P,e] = lqr(A,B,Q,R);
X0 = [0; 0; 0; 0; 3.142/2; 0; 0; 0; 0; 0; 0; 0];
p = [-1 -2 -3 -4 -5 -6]; % Poles matrix
L1 = place(A', C1', p)'
L2 = place(A', C2', p)';
L3 = place(A', C3', p)';

Ac1 = [(A-B*K) B*K; zeros(size(A)) (A-L1*C1)];
Bc1 = [B;B]; % Taking Bd = Bk = B
Cc1 = [C1 zeros(size(C1))];

Ac2 = [(A-B*K) B*K; zeros(size(A)) (A-L2*C2)];
Bc2 = [B;B];
Cc2 = [C2 zeros(size(C2))];

Ac3 = [(A-B*K) B*K; zeros(size(A)) (A-L3*C3)];
Bc3 = [B;B];
Cc3 = [C3 zeros(size(C3))];
sys1 = ss(Ac1, Bc1, Cc1, D);
figure
initial(sys1,X0)
figure
step(sys1)

sys2 = ss(Ac2, Bc2, Cc2, D);
figure
initial(sys2,X0)
```

```
figure
step(sys2)

sys3 = ss(Ac3, Bc3, Cc3, D);
figure
initial(sys3,X0)
figure
step(sys3)
```

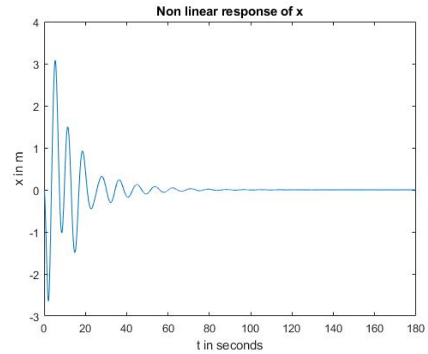


Fig. 9. Initial Response of Nonlinear System in Case 1

2) For Nonlinear System: We use ode45 to solve the ordinary differential equations here.

Case 1: Output is x

The Output of code below is as shown in Fig. 9.

Code for Luenberger Observer in Nonlinear System for Case 1:

```
y0 = [0; 0; 0; 0; 3.14/2; 0];
tspan = 0:0.01:180;%defining the timespan
[t1,y1] = ode45(@non_linear_observer,tspan,y0);
figure
plot(t1,y1(:,1))
title('Non linear response of x')
xlabel('t in seconds')
ylabel('x in m')
figure
plot(t1,y1(:,2))
title('Non linear response of x_dot')
xlabel('t in seconds')
ylabel('x_dot in m/sec')
figure
plot(t1,y1(:,3))
title('Non linear response of theta1')
```

```
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,4))
title('Non linear response of theta1_dot')
xlabel('t in seconds')
ylabel('theta1_dot in radians/sec')
figure
plot(t1,y1(:,5))
title('Non linear response of theta2')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,6))
title('Non linear response of theta2_dot')
xlabel('t in seconds')
ylabel('theta2_dot in radians/sec')

function dydt = non_linear_observer(t,y)
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];
C = [1 0 0 0 0 0];

Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
```

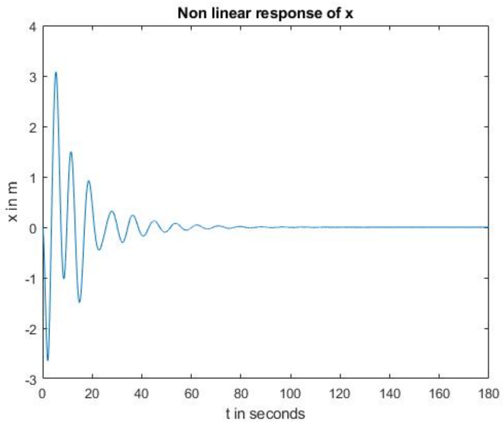
```
      0 0 10000 0 0 0;
      0 0 0 100000 0 0;
      0 0 0 0 10000 0;
      0 0 0 0 0 100000];
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
y1 = [y(1)];
F = -K*y;
p = [-1 -2 -3 -4 -5 -6]; % Poles matrix
L = place(A', C', p)';
correction_term = L*(y1-C*y);
dydt=zeros(6,1);
dydt(1)  = y(2)+correction_term(1); %x_dot
dydt(2)=(F-1000*sin(y(3))*cos(y(3))-1000*sin(y(5))*cos(y(5))
   -(2000*(y(4)^2)*sin(y(3)))-(1000*(y(6)^2)*sin(y(5))))
   /(1000+100*((sind(y(3)))^2)+100*((sind(y(5)))^2))+
   correction_term(2);%x_ddot
dydt(3)= y(4)+correction_term(3); %theta 1D
dydt(4)= (dydt(2)*cos(y(3))-10*(sin(y(3))))/20 +correction_term
   (4); %theta 1 ddot;
dydt(5)= y(6)+correction_term(5); %theta 2D
dydt(6)= (dydt(2)*cos(y(5))-10*(sin(y(5))))/10+correction_term(6)
   ; %theta 2 ddot;
end
```

Case 2: Output is $[x, \theta_2]^T$

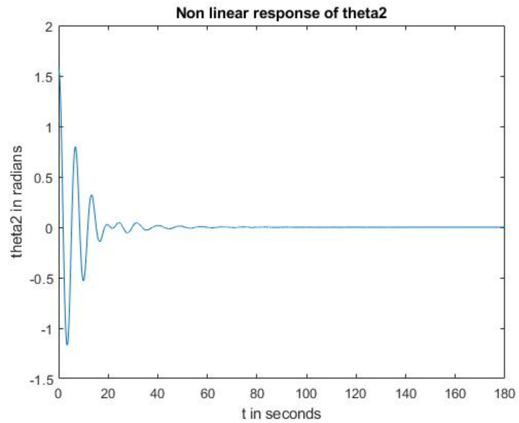
The output for Case 2 is shown in Fig. 10

Code for Luenberger Observer in Nonlinear System for Case 2:

```
y0 = [0; 0; 0; 0; 3.14/2; 0];
tspan = 0:0.01:180;%defining the timespan
[t1,y1] = ode45(@non_linear_observer,tspan,y0);
```



(a)



(b)

Fig. 10. Initial Response of Nonlinear System in Case 2

```
figure
plot(t1,y1(:,1))
title('Non linear response of x')
xlabel('t in seconds')
ylabel('x in m')
figure
plot(t1,y1(:,2))
title('Non linear response of x_dot')
xlabel('t in seconds')
ylabel('x_dot in m/sec')
figure
plot(t1,y1(:,3))
title('Non linear response of theta1')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,4))
title('Non linear response of theta1_dot')
xlabel('t in seconds')
```

```
ylabel('theta1_dot in radians/sec')
figure
plot(t1,y1(:,5))
title('Non linear response of theta2')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,6))
title('Non linear response of theta2_dot')
xlabel('t in seconds')
ylabel('theta2_dot in radians/sec')

function dydt = non_linear_observer(t,y)
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];
C = [1 0 0 0 0 0; 0 0 0 0 1 0];

Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
     0 0 10000 0 0 0;
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
     0 0 0 0 0 100000];
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
```

```
y1 = [y(1);y(5)];
F = -K*y;
p = [-1 -2 -3 -4 -5 -6]; % Poles matrix
L = place(A', C', p)';
correction_term = L*(y1-C*y);
dydt=zeros(6,1);
dydt(1) = y(2)+correction_term(1); %x_dot
dydt(2)=(F-1000*sin(y(3))*cos(y(3))-1000*sin(y(5))*cos(y(5))
   -(2000*(y(4)^2)*sin(y(3)))-(1000*(y(6)^2)*sin(y(5))))
   /(1000+100*((sind(y(3)))^2)+100*((sind(y(5)))^2))+
   correction_term(2);%x_ddot
dydt(3)= y(4)+correction_term(3); %theta 1D
dydt(4)= (dydt(2)*cos(y(3))-10*(sin(y(3))))/20 +correction_term
   (4); %theta 1 ddot;
dydt(5)= y(6)+correction_term(5); %theta 2D
dydt(6)= (dydt(2)*cos(y(5))-10*(sin(y(5))))/10+correction_term(6)
   ; %theta 2 ddot;
end
```

Case 3: Output is $[x, \theta_1, \theta_2]^T$

The output for Case 3 is shown in Fig. 11

Code for Luenberger Observer in Nonlinear System for Case 3:

```
y0 = [0; 0; 0; 0; 3.14/2; 0];
tspan = 0:0.01:180;%defining the timespan
[t1,y1] = ode45(@non_linear_observer,tspan,y0);
figure
plot(t1,y1(:,1))
title('Non linear response of x')
xlabel('t in seconds')
ylabel('x in m')
figure
```

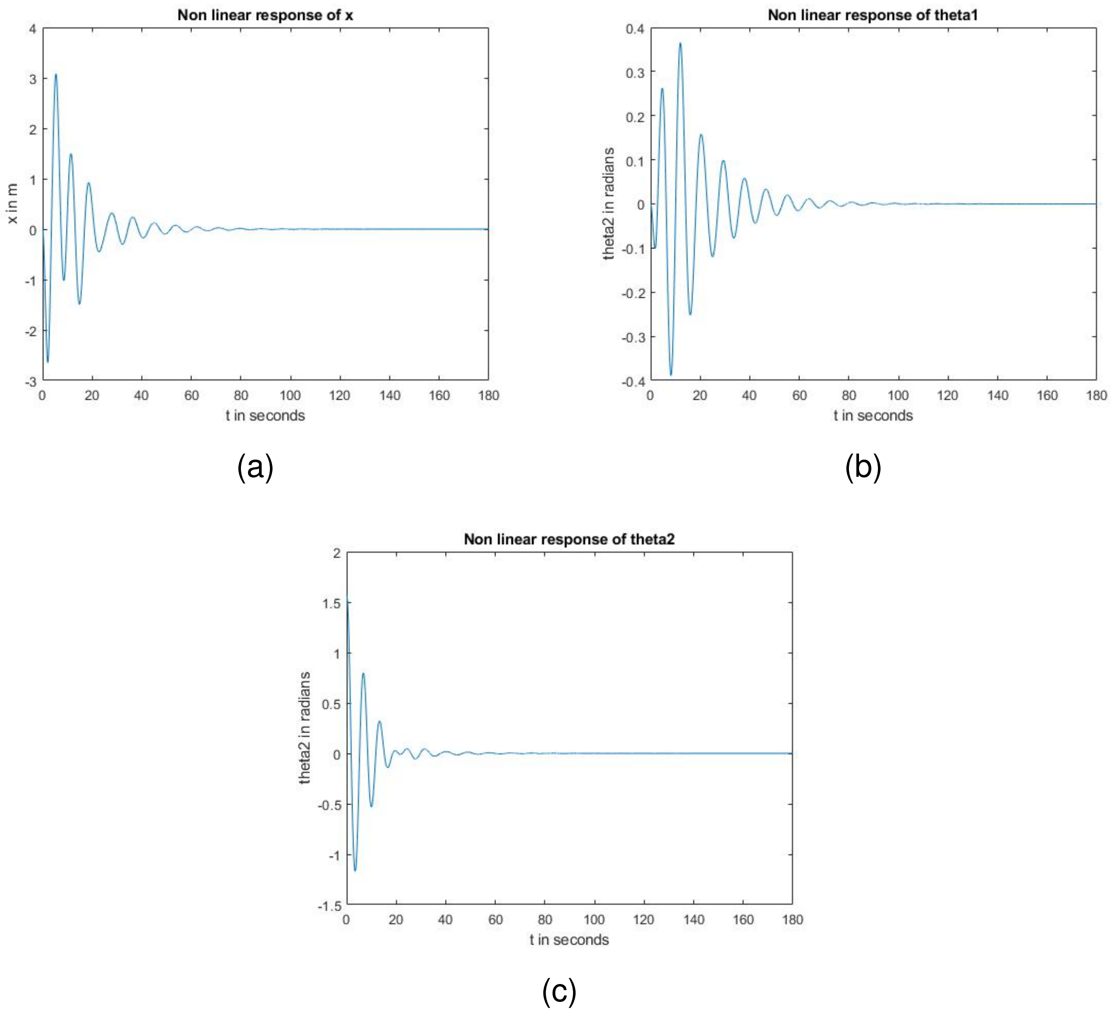
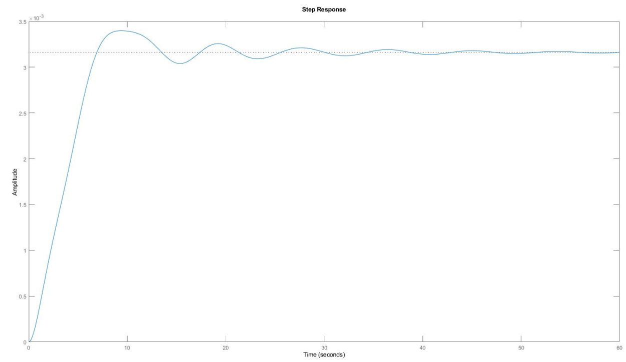


Fig. 11. Initial Response of Nonlinear System in Case 3

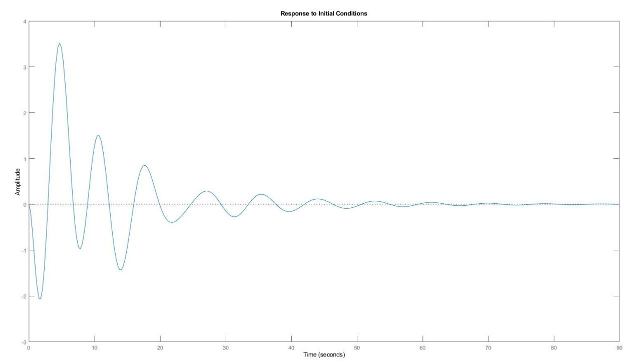
```
plot(t1,y1(:,2))
title('Non linear response of x_dot')
xlabel('t in seconds')
ylabel('x_dot in m/sec')
figure
plot(t1,y1(:,3))
title('Non linear response of theta1')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
```

```
plot(t1,y1(:,4))
title('Non linear response of theta1_dot')
xlabel('t in seconds')
ylabel('theta1_dot in radians/sec')
figure
plot(t1,y1(:,5))
title('Non linear response of theta2')
xlabel('t in seconds')
ylabel('theta2 in radians')
figure
plot(t1,y1(:,6))
title('Non linear response of theta2_dot')
xlabel('t in seconds')
ylabel('theta2_dot in radians/sec')

function dydt = non_linear_observer(t,y)
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];
C = [1 0 0 0 0 0 ;0 0 1 0 0 0; 0 0 0 0 1 0];
Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
     0 0 10000 0 0 0;
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
     0 0 0 0 0 100000];
```

```
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
y1 = [y(1); y(3); y(5)];
F = -K*y;
p = [-1 -2 -3 -4 -5 -6]; % Poles matrix
L = place(A', C', p)';
correction_term = L*(y1-C*y);
dydt=zeros(6,1);
dydt(1) = y(2)+correction_term(1); %x_dot
dydt(2)=(F-1000*sin(y(3))*cos(y(3))-1000*sin(y(5))*cos(y(5))
   -(2000*(y(4)^2)*sin(y(3)))-(1000*(y(6)^2)*sin(y(5))))
   /(1000+100*((sind(y(3)))^2)+100*((sind(y(5)))^2))+
   correction_term(2);%x_ddot
dydt(3)= y(4)+correction_term(3); %theta 1D
dydt(4)= (dydt(2)*cos(y(3))-10*(sin(y(3))))/20 +correction_term
   (4); %theta 1 ddot;
dydt(5)= y(6)+correction_term(5); %theta 2D
dydt(6)= (dydt(2)*cos(y(5))-10*(sin(y(5))))/10+correction_term(6)
   ; %theta 2 ddot;
end
```



(a)



(b)

Fig. 12. (a) Initial Response and (b) Step Response of Linearized System for LQG

G. LQG (Linear Quadratic Gaussian) Controller Design

We solve the Riccatti equation:

$$AP + PA^T - PC^T\Sigma_V^{-1}CP = -B_D\Sigma_D B_D^T \tag{12}$$

The P matrix which satisfies the Riccatti Equation in Eq. 12 gives us the optimal observer gain L as $L = PC^T\Sigma_V^{-1}$. We use the lqr() function to solve the Ricatti equation by giving parameters A^T instead of A, C^T instead of C, Σ_D instead of Q and Σ_V instead of R since original Ricatti equation for the LQR is of the form $A^TP + PA - PB_K^TR^{-1}B_KP = -Q$.

1) LQG Linear: The results of the code below can be seen in Fig. 12. We have taken the smallest output vector $x(t)$.

Code for LQG Linear:

```
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];

C = [1 0 0 0 0 0]; %for output vector x
D = zeros(1,1);

Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
     0 0 10000 0 0 0;
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
     0 0 0 0 0 100000];
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
X0 = [0; 0; 0; 0; 3.142/2; 0; 0; 0; 0; 0; 0; 0];
vd=0.1*eye(6); %process noise
vn=1; %measurement noise

L=lqr(A',C',vd,vn)'
sys = ss([(A-B*K) B*K; zeros(size(A)) (A-L*C)], [B;B],[C zeros(
   size(C))], D);
figure
initial(sys,X0)
```

```
figure
step(sys)
```

Now we also implement LQG for original non-linear system.

2) LQG Non-Linear: The results of the code below can be seen in Fig. 13. We have taken the smallest output vector $x(t)$.

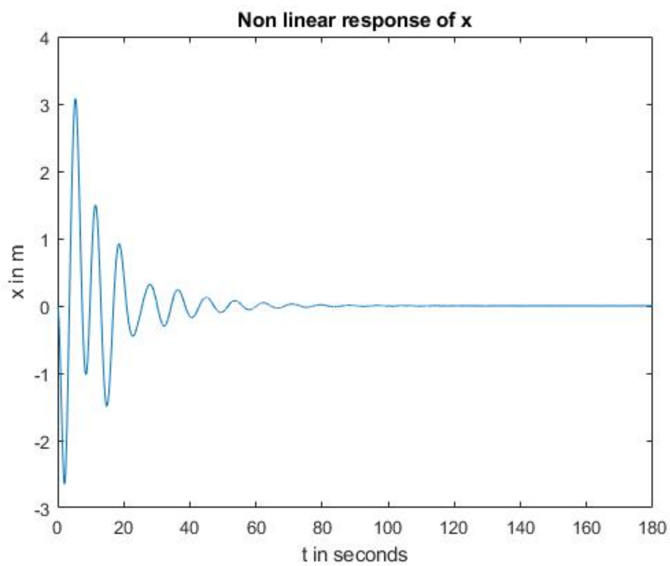


Fig. 13. Initial Response of Original Nonlinear System for LQG

Code for LQG Nonlinear:

```
y0 = [0; 0; 0; 0; 3.14/2; 0];
tspan = 0:0.01:180;%defining the timespan
[t1,y1] = ode45(@non_linear_lqg,tspan,y0);
figure
plot(t1,y1(:,1))
title('Non linear response of x')
xlabel('t in seconds')
ylabel('x in m')
function dydt = non_linear_lqg(t,y)
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -11/20 0 -1/20 0;
     0 0 0 0 0 1;
     0 0 -1/10 0 -11/10 0];

B = [0; 1/1000; 0; 1/20000; 0; 1/10000];
C = [1 0 0 0 0 0];

Q = [1000 0 0 0 0 0;
     0 10 0 0 0 0;
     0 0 10000 0 0 0;
     0 0 0 100000 0 0;
     0 0 0 0 10000 0;
     0 0 0 0 0 100000];
R = 0.01*ones(1,1);
[K,P,e] = lqr(A,B,Q,R);
y1 = [y(1)];
F = -K*y;
```

```
vd=0.1*eye(6); %process noise
vn=1; %measurement noise
L = lqr(A',C',vd,vn)';
correction_term = L*(y1-C*y);
dydt=zeros(6,1);
dydt(1)  = y(2)+correction_term(1); %x_dot
dydt(2)=(F-1000*sin(y(3))*cos(y(3))-1000*sin(y(5))*cos(y(5))
   -(2000*(y(4)^2)*sin(y(3)))-(1000*(y(6)^2)*sin(y(5))))
   /(1000+100*((sind(y(3)))^2)+100*((sind(y(5)))^2))+
   correction_term(2);%x_ddot
dydt(3)= y(4)+correction_term(3); %theta 1D
dydt(4)= (dydt(2)*cos(y(3))-10*(sin(y(3))))/20 +correction_term
   (4); %theta 1 ddot;
dydt(5)= y(6)+correction_term(5); %theta 2D
dydt(6)= (dydt(2)*cos(y(5))-10*(sin(y(5))))/10+correction_term(6)
   ; %theta 2 ddot;
end
```

REFERENCES

[1] W. Malik, ”ENPM667 Final Project Instructions”, Nov. 2021, University of Maryland-College Park.