ENPM667: Control of Robotic Systems

PROJECT-2

FINAL REPORT

LQR & LQG Design

Koushik Alapati (120239489)

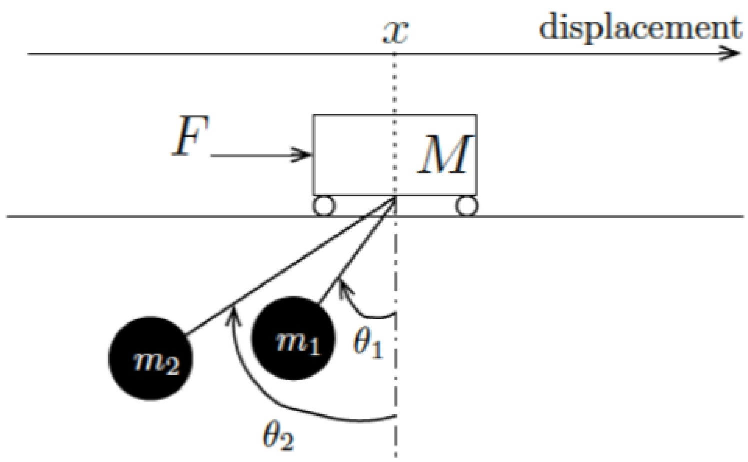
Sai Dinesh Gelam (120167140)

Raghu dharahas Kotla (120378935)

Contents

1.Introduction:

The objective of this project is to analyze and control a crane system that moves along a one-dimensional track. The system consists of a frictionless cart with mass M, actuated by an external force F, and two loads suspended from cables with masses m1 and m2, and lengths l1 and l2. The project is divided into two main components: the first component focuses on obtaining the equations of motion, linearizing the system, checking controllability, and designing an LQR controller. The second component delves into observability analysis, Luenberger observer design, LQG controller design, and reconfiguration for constant reference tracking and disturbance rejection.



The first step is to derive the equations of motion for the crane system. This involves considering the dynamics of the cart and loads, incorporating forces and lengths of cables. The resulting nonlinear equations will be used to formulate the state-space representation. Hence the linearization process involves finding the equilibrium point and linearizing the system around it. Later Conditions for controllability will be determined based on system parameters (M, m1, m2, l1, l2). The focus is on ensuring that the linearized system is controllable, which is essential for effective control design. For given parameter values (M, m1, m2, l1, l2), the controllability of the system is checked, and an LQR controller is designed. The stability of the closed-loop system is analyzed using Lyapunov's indirect method. Next step will be considered to determine observability of the linearized system. In this the Luenberger observer will be designed for output vectors that render the system observability.

2.Equations of motion

Let's define the position (x) of mass m1 as a function of θ_1

$$x_{m_1} = (x - l_1 \sin(\theta_1))\hat{\imath} + (-l_1 \cos(\theta_1)\hat{\jmath} \qquad (1)$$

Velocity can be obtained by differentiating x_{m_1} with respective time we get

$$v_{m_1} = \left(\dot{x} - l_1 \cos(\theta_1)\dot{\theta_1}\right)\hat{\imath} + ((-l_1 \sin(\theta_1)\dot{\theta_1})\hat{\jmath} \qquad (2)$$

Similarly, for m2 we get

$$x_{m_2} = (x - l_2 \sin(\theta_2))\hat{\imath} + (-l_2 \cos(\theta_2)\hat{\jmath} \qquad (3)$$

$$v_{m_2} = \left(\dot{x} - l_2 \cos(\theta_2)\dot{\theta_2}\right)\hat{\imath} + ((-l_2 \sin(\theta_2)\dot{\theta_2})\hat{\jmath}$$

Since we have velocities of both masses, we can now compute kinetic energies of both the masses

$$\mathrm{K} = \frac{1}{2}\mathrm{M}\dot{\mathrm{x}}^2 + \frac{1}{2}\mathrm{m}_1\left(\dot{\mathrm{x}} - \mathrm{l}_1\dot{\theta}_1\cos(\theta_1)\right)^2 + \frac{1}{2}\mathrm{m}_1\left(\mathrm{l}_1\dot{\theta}_1\sin(\theta_1)\right)^2 + \frac{1}{2}\mathrm{m}_2\left(\dot{\mathrm{x}} - \mathrm{l}_2\dot{\theta}_2\cos(\theta_2)\right)^2 + \frac{1}{2}\mathrm{m}_2\left(\mathrm{l}_2\dot{\theta}_2\sin(\theta_2)\right)^2 \qquad (4)$$

And potential energy of the system is given by

$$P = -\mathrm{m}_1\mathrm{g}\mathrm{l}_1 cos(\theta_1) - \mathrm{m}_2\mathrm{g}\mathrm{l}_2 cos(\theta_2)$$
$$= -\mathrm{g}\,(\mathrm{m}_1\mathrm{l}_1\cos(\theta_1) + \mathrm{m}_2\mathrm{g}\mathrm{l}_2\cos(\theta_2)) \qquad (5)$$

Now we have both K and P values to compute langrage Equation as follows.

$$L(q,\dot{q}) = K(q,\dot{q}) - U(q) \qquad (6)$$

Where, $K(q,\dot{q})$ is Kinetic Energy and $U(q)$ is Potential Energy

So we get,

$$\mathrm{L} = \frac{1}{2}\mathrm{M}\dot{\mathrm{x}}^2 + \frac{1}{2}\mathrm{m}_1\dot{\mathrm{x}}^2 + \frac{1}{2}m_1 l_1\dot{\theta}_1^2\cos^2(\theta_1) - m_1 l_1\dot{\theta}_1\dot{x}cos(\theta_1) + \frac{1}{2}m_1 l_1\dot{\theta}_1^2\sin^2(\theta_1) + \frac{1}{2}\mathrm{m}_2\dot{\mathrm{x}}^2 + \frac{1}{2}m_2 l_2\dot{\theta}_2^2\cos^2(\theta_2) - m_2 l_2\dot{\theta}_2\dot{x}cos(\theta_2) + \frac{1}{2}m_2 l_2\dot{\theta}_2^2\sin^2(\theta_2) + g[m_1 l_1\cos(\theta_1) + m_2 l_2\cos(\theta_2)] \qquad (8)$$

On further simplification we get

$$L = \frac{1}{2}\,\mathrm{M}\dot{\mathrm{x}}^2 + \frac{1}{2}\,(\mathrm{m}_1 + m_2)\dot{\mathrm{x}}^2 + + \frac{1}{2} m_1 l_1 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2 \dot{\theta}_2^2 - \dot{x}\big(m_1 l_1 \dot{\theta}_1 cos(\theta_1) + m_2 l_2 \dot{\theta}_2 cos(\theta_2)\,\big) + g[m_1 l_1 \cos(\theta_1) + m_2 l_2 \cos(\theta_2)] \quad (9)$$

Now find Euler- langrange equations as follows.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \quad (10)$$

Using the above equation, for our system,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F \quad (11)$$

$$\frac{\partial L}{\partial \dot{x}} = M\dot{x} + (m_1 + m_2)\dot{x} - m_1 l_1 \dot{\theta}_1 \, cos(\theta_1) - m_2 l_2 \, \dot{\theta}_2 \, cos(\theta_2) \quad (12)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = M\ddot{x} + (m_1 + m_2)\ddot{x} - \left[m_1 l_1 \ddot{\theta}_1 cos(\theta_1) - m_1 l_1 \dot{\theta}_1^2 \, sin(\theta_1)\right] - [m_2 l_2 \, \ddot{\theta}_2 \, cos(\theta_2) - m_2 l_2 \dot{\theta}_2^2 \, sin(\theta_2) \quad (13)$$

Also, we have

$$\frac{\partial L}{\partial x} = 0 \quad (14)$$

Using this the equation can be written as

$$[\mathrm{M} + \mathrm{m1} + \mathrm{m2}]\ddot{\mathrm{x}} - \mathrm{m}_1 \mathrm{l}_1 \ddot{\theta}_1 \cos(\theta_1) + \mathrm{m}_1 \mathrm{l}_1 \dot{\theta}_1^2 \sin(\theta_1) - \mathrm{m}_2 \mathrm{l}_2 \, \dot{\theta}_2 \, \cos(\theta_2) + \mathrm{m}_2 \mathrm{l}_2 \ddot{\theta}_2^2 \sin(\theta_2) = \mathrm{F} \quad (15)$$

And also we know that

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = 0 \quad (16)$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = m_1 l_1^2 \dot{\theta}_1 - m_1 \dot{x} l_1 co\, s(\theta_1) \quad (17)$$

On differentiating with respective time,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) = m_1 l_1^2 \ddot{\theta}_1 - \left[m_1 l_1 \ddot{x_1} cos(\theta_1) - m_1 l_1 \dot{x}\dot{\theta}_1 sin(\theta_1)\right] \tag{18}$$

And

$$\left(\frac{\partial L}{\partial \theta_1}\right) = m_1 l_1 \dot{x}\dot{\theta}_1 \sin(\theta_1) - g m_1 l_1 \sin(\theta_1) \tag{19}$$

Combining the above equations, we get

$$m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \ddot{x_1}\, cos(\theta_1) + m_1 l_1 \dot{x}\dot{\theta}_1\, sin(\theta_1) - m_1 l_1 \dot{x}\dot{\theta}_1 sin(\theta_1) + g m_1 l_1\, sin(\theta_1) = 0 \tag{20}$$

simplification we get second Lagrange equations

$$m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \ddot{x_1}\, cos(\theta_1) + m_1 \dot{x}\dot{\theta}_2 l_1\, sin(\theta_1) = 0 \tag{21}$$

Similarly, the third lagrangian equation can computed as follows

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) - \frac{\partial L}{\partial \theta_2} = 0 \tag{22}$$

$$\frac{\partial L}{\partial \dot{\theta_2}} = m_2 l_2^2 \dot{\theta}_2 - m_2 \dot{x} l_2 co\,s(\theta_2) \tag{23}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) = m_2 l_2^2 \ddot{\theta}_2 - \left[m_2 l_2 \ddot{x_2}\, cos(\theta_2) - m_2 l_2 \dot{x}\dot{\theta}_2\, sin(\theta_2)\right] \tag{24}$$

$$\left(\frac{\partial L}{\partial \theta_2}\right) = m_2 l_2 \dot{x}\dot{\theta}_2 sin(\theta_2) - g m_2 l_2\, sin(\theta_2) \tag{25}$$

, We get

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \ddot{x_2}\, cos(\theta_2) + m_2 l_2 \dot{x}\dot{\theta}_2\, sin(\theta_2) - m_2 l_2 \dot{x}\dot{\theta}_2 sin(\theta_2) + g m_2 l_2\, sin(\theta_2) = 0 \tag{26}$$

On simplification we get

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \ddot{x_2}\, cos(\theta_2) + g m_2 l_2\, sin(\theta_2) = 0 \tag{27}$$

formatting these equations in state space form, we get the following state equations

$$\ddot{x} = \frac{1}{M + m1 + m2}\left[m_1 l_1 \ddot{\theta_1}\, cos(\theta_1) + m_2 l_2\, \ddot{\theta_2}\, cos(\theta_2) - m_1 l_1 \dot{\theta}_1^2\, sin(\theta_1) - m_2 l_2 \dot{\theta}_2^2\, sin(\theta_2) + F\right] \tag{28}$$

$$\ddot{\theta_1} = \frac{\ddot{x} cos\theta_1}{l_1} - \frac{g sin\theta_1}{l_1} \tag{29}$$

$$\ddot{\theta_2} = \frac{\ddot{x} cos\theta_2}{l_2} - \frac{g sin\theta_2}{l_2} \tag{30}$$

So when written in matrix we get the following state space represemtation

$$
\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{-m_1 g \sin\theta_1 \cos\theta_2 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F}{M + m_1 + m_2 - m_1 \cos^2\theta_1 - m_2 \cos^2\theta_2} \\ \dot{\theta}_1 \\ \frac{-m_1 g \sin\theta_1 \cos\theta_2 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F}{(M + m_1 + m_2 - m_1 \cos^2\theta_1 - m_2 \cos^2\theta_2) l_1} - \frac{g \sin\theta_1}{l_1} \\ \dot{\theta}_2 \\ \frac{-m_1 g \sin\theta_1 \cos\theta_2 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F}{(M + m_1 + m_2 - m_1 \cos^2\theta_1 - m_2 \cos^2\theta_2) l_2} - \frac{g \sin\theta_2}{l_2} \end{bmatrix}
$$

3.Linearization

Linearization involves creating a linear approximation of a function at a specific point or a trajectory. In the domain of dynamical systems, this method is used to evaluate the local stability of an equilibrium point in systems characterized by nonlinear differential equations or discrete dynamics. By linearizing, we gain the advantage of using tools designed for linear systems to analyze the behavior of a nonlinear function in the vicinity of a given point.

In our case, the derived equations of motion for the cart system with two pendulums, presented in its state-space form, exhibit nonlinearity due to the presence of sine and cosine components. Solving nonlinear equations can be very challenging. To address this challenge, we opt to linearize the system around the equilibrium point defined by $x = 0, \theta 1 = 0, and\ \theta 2 = 0$ as specified in the problem statement.

The conditions set at equilibrium serve as limiting criteria. By establishing the system at rest $(x = 0, \theta 1 = 0, and\ \theta 2 = 0)$, we create a simplified linearized version, making the analysis more tractable. This strategic linearization allows us to apply well-established techniques for linear system analysis, enabling a focused examination of the system's local behavior around the equilibrium point.

At the premises of equilibrium, we can consider the following.

$$\sin\theta_1 \approx \theta_1 \tag{31}$$

$$\sin\theta_2 \approx \theta_2 \tag{32}$$

$$\cos\theta_1 \approx 1 \tag{33}$$

$$\cos\theta_2 \approx 1 \tag{34}$$

$$\dot{\theta}_1^2 = \dot{\theta}_2^2 \approx 0 \tag{35}$$

Using the above approximations, the we get simplified linearized system, and the linearized state equation as:

$$\ddot{x} = \frac{1}{M + m1 + m2}\left[\mathrm{m_1 l_1 \ddot{\theta}_1 . 1 +\ m_2 l_2}\ \ddot{\theta_2}\ \ -\ 0 -\ 0 + F\right] \tag{36}$$

$$\ddot{\theta}_1 = \frac{\ddot{x}.1}{l_1} - \frac{g\theta_1}{l_1} \tag{37}$$

$$\ddot{\theta}_2 = \frac{\ddot{x}.1}{l_2} - \frac{g\theta_2}{l_2} \tag{38}$$

And state matrix is given by

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta_1} \\ \ddot{\theta_1} \\ \dot{\theta_2} \\ \ddot{\theta_2} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dfrac{-m_1 g\theta_1 - m_2 g\theta_2 + m_1\ddot{x} + m_2\ddot{x} + F}{M + m_1 + m_2} \\ \dot{\theta_1} \\ \dfrac{-m_1 g\theta_1 - m_2 g\theta_2 + F}{Ml_1} - \dfrac{g\theta_1}{l_1} \\ \dot{\theta_2} \\ \dfrac{-m_1 g\theta_1 - m_2 g\theta_2 + F}{Ml_2} - \dfrac{g\theta_2}{l_2} \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta_1} \\ \ddot{\theta_1} \\ \dot{\theta_2} \\ \ddot{\theta_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\dfrac{gm_1}{M} & 0 & -\dfrac{gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{g(M+m1)}{Ml_1} & 0 & -\dfrac{gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\dfrac{gm_1}{Ml_2} & 0 & -\dfrac{g(M+m2)}{Ml_2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta_1} \\ \theta_1 \\ \dot{\theta_2} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{M} \\ 0 \\ \dfrac{1}{Ml_1} \\ 0 \\ \dfrac{1}{Ml_2} \end{bmatrix} F \tag{39}$$

4.Controllability

After linearizing the system, we can observe that the system is time invariant, so the controllability condition of the system can be easily found out by finding the rank of the controllability matrix (G). If the matrix G has full rank of the i.e. number of state variables must be equal to rank of G, then the system is said to be controllable.

$$rank(G) = [B\ AB\ A^2\ B\ A^3B\ A^4B\ A^5B\ ] \tag{40}$$

```
% define xdot = Ax+Bu
% Defining symbols
syms M m1 m2 l1 l2 g
% g = 9.81;
A = [0 1 0 0 0 0;
```

```
     0 0 (-g*m1)/M 0 (-g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 -(g*(M+m1))/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

disp(A);
```

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{g\,m_1}{M} & 0 & -\frac{g\,m_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g\,(M+m_1)}{M\,l_1} & 0 & -\frac{g\,m_2}{M\,l_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{g\,m_1}{M\,l_2} & 0 & -\frac{g\,(M+m_2)}{M\,l_2} & 0 \end{pmatrix} \tag{41}$$

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

$$\begin{pmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M\,l_1} \\ 0 \\ \frac{1}{M\,l_2} \end{pmatrix} \tag{42}$$

Controllability Matrix

```
G = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
disp(G);
```

$$\begin{pmatrix} 0 & \frac{1}{M} & 0 & \sigma_2 & 0 & \sigma_1 \\ \frac{1}{M} & 0 & \sigma_2 & 0 & \sigma_1 & 0 \\ 0 & \frac{1}{M\,l_1} & 0 & \sigma_6 & 0 & \sigma_4 \\ \frac{1}{M\,l_1} & 0 & \sigma_6 & 0 & \sigma_4 & 0 \\ 0 & \frac{1}{M\,l_2} & 0 & \sigma_5 & 0 & \sigma_3 \\ \frac{1}{M\,l_2} & 0 & \sigma_5 & 0 & \sigma_3 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \frac{\frac{g^2\,m_1\,(M+m_1)}{M^2\,l_1} + \frac{g^2\,m_1\,m_2}{M^2\,l_2}}{M\,l_1} + \frac{\frac{g^2\,m_2\,(M+m_2)}{M^2\,l_2} + \frac{g^2\,m_1\,m_2}{M^2\,l_1}}{M\,l_2}$$

$$\sigma_2 = -\frac{g\,m_1}{M^2\,l_1} - \frac{g\,m_2}{M^2\,l_2}$$

$$\sigma_3 = \frac{\frac{g^2\,m_1\,(M+m_2)}{M^2\,{l_2}^2} + \frac{g^2\,m_1\,(M+m_1)}{\sigma_7}}{M\,l_1} + \frac{\frac{g^2\,(M+m_2)^2}{M^2\,{l_2}^2} + \frac{g^2\,m_1\,m_2}{\sigma_7}}{M\,l_2}$$

$$\sigma_4 = \frac{\frac{g^2\,m_2\,(M+m_1)}{M^2\,{l_1}^2} + \frac{g^2\,m_2\,(M+m_2)}{\sigma_7}}{M\,l_2} + \frac{\frac{g^2\,(M+m_1)^2}{M^2\,{l_1}^2} + \frac{g^2\,m_1\,m_2}{\sigma_7}}{M\,l_1}$$

$$\sigma_5 = -\frac{g\,(M+m_2)}{M^2\,{l_2}^2} - \frac{g\,m_1}{\sigma_7}$$

$$\sigma_6 = -\frac{g\,(M+m_1)}{M^2\,{l_1}^2} - \frac{g\,m_2}{\sigma_7}$$

$$\sigma_7 = M^2\,l_1\,l_2$$

The rank of the matrix (G) will be full only when its determinant is not equal to zero. So, using this we come across the following cases

```
disp(simplify(det(G)));
```

$$-\frac{g^6\,(l_1-l_2)^2}{M^6\,{l_1}^6\,{l_2}^6}$$

```
disp(rank(G));
```

Case 1($l1 = l2$): In this case the determinate becomes zero and condition of having full rank (less than 6) doesn't satisfy the rank condition hence the leads to system controllability.

Case 2($l1 \neq l2$): In this case the determinate will not be equal to zero and results in full rank of the matrix(C) hence the controllability of the system.

Hence from the above equation we get that the system will be controllable when the following conditions satisfy.

$$l1 \neq l2, \quad l_1 \neq 0, \quad l_2 \neq 0, \quad M > 0$$

5.LQR Controller

If the system is controllable, to put the system in the stable state we need a controller that can be used to drive the system to get desired value (outputs). One such controller is LQR which is a powerful and widely used technique in control theory for optimizing the performance of linear dynamic systems. The LQR controller aims to minimize a cost function that represents a trade-off between the system's performance and the control effort. Hence gives most optimized gains values for the feedback.

LQR will be designed in order to minimize the following cost function.

$$J\left(k, \vec{X}(0)\right) = \int_0^\infty \vec{X}^T(t) Q\, \vec{X}(t) + \vec{U}_k^T(t) Q \vec{U}(t) dt \tag{43}$$

Where Q is a positive definite symmetric matrix which represents the performance of the system

R is also a positive definite symmetric matrix which represents input efforts

So, choosing right combination of Q and R based on our application takes the system in a the most optimal way.

Considering the initial conditions 1. $x = 10, \dot{x} = 0, \theta_1 = 10°, \dot{\theta}_1 = 0, \theta_2 = 20°, \dot{\theta}_2 = 0$

5.1 Controllability Check

```
M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;
A = double(subs(A, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val, 
l2_val, g_val]));
B = double(subs(B, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val, 
l2_val, g_val]));
C_= ctrb(A,B);
disp(C_);
```

```
   1.0e-03 *

        0    1.0000         0   -0.1472         0    0.1419
   1.0000         0   -0.1472         0    0.1419         0
        0    0.0500         0   -0.0319         0    0.0227
   0.0500         0   -0.0319         0    0.0227         0
        0    0.1000         0   -0.1128         0    0.1249
   0.1000         0   -0.1128         0    0.1249         0
```

Controllability test

```
if(rank(C_) == size(A))
    disp('The system is controllable');
else
    disp('The system is uncontrollable');
end
```

```
The system is controllable.
```

5.2 System Response based on Initial Conditions

```
C = eye(6);
D = 0;
X_0 = [0;0;0.5;0;0.6;0];
sys = ss(A,B,C,D);
figure
initial(sys,X_0)
grid on
```

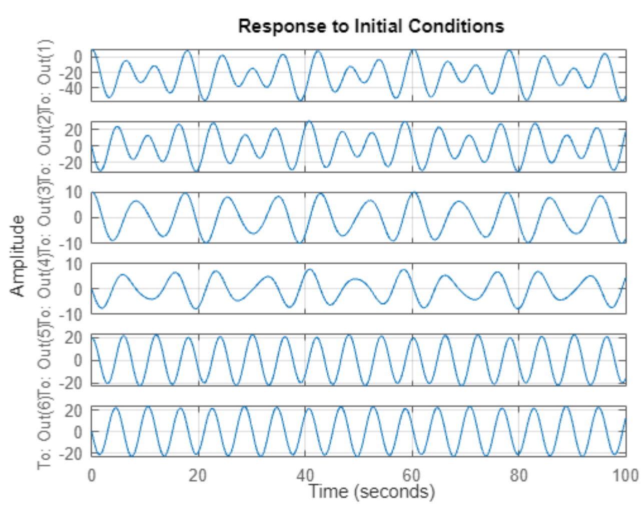


Figure 1:Before Applying LQR

5.3 System Response after LQR is applied to Linearized System:

```
Q = [1000 0 0 0 0 0;  %x penalize
     0 1000 0 0 0 0;  %x_dot penalize
     0 0 1 0 0 0;   %theta1 penalize
     0 0 0 1 0 0;   %theta1_dot penalize
     0 0 0 0 1 0;   %theta2 penalize
     0 0 0 0 0 1];  %theta2_dot penalize
% Q =eye(6)*100;
R = 0.001
```

```
R = 1.0000e-03
```

find the value k

```
[k, P, ~] = lqr(A, B, Q, R)
```

```
k = 1×6

10³ ×

    1.0000    1.8191   -0.2725   -0.9704   -0.1740   -0.5649
P = 6×6

10⁴ ×

    0.1819    0.1154   -0.0970   -0.1444   -0.0565   -0.0823
    0.1154    0.2257   -0.0321   -0.4606   -0.0205   -0.2072
   -0.0970   -0.0321    3.4210    0.0470   -0.0017    0.0248
   -0.1444   -0.4606    0.0470    7.1148    0.0300    0.0781
   -0.0565   -0.0205   -0.0017    0.0300    1.3949    0.0159
   -0.0823   -0.2072    0.0248    0.0781    0.0159    1.4684
```

```
disp(k)
```

```
  1.0e+03 *


    1.0000    1.8191   -0.2725   -0.9704   -0.1740   -0.5649
```

```
system_2 = ss(A-(B*k),B,C,D); %Using the K matrix to define ss
figure
initial(system_2,X_0)
grid on
```

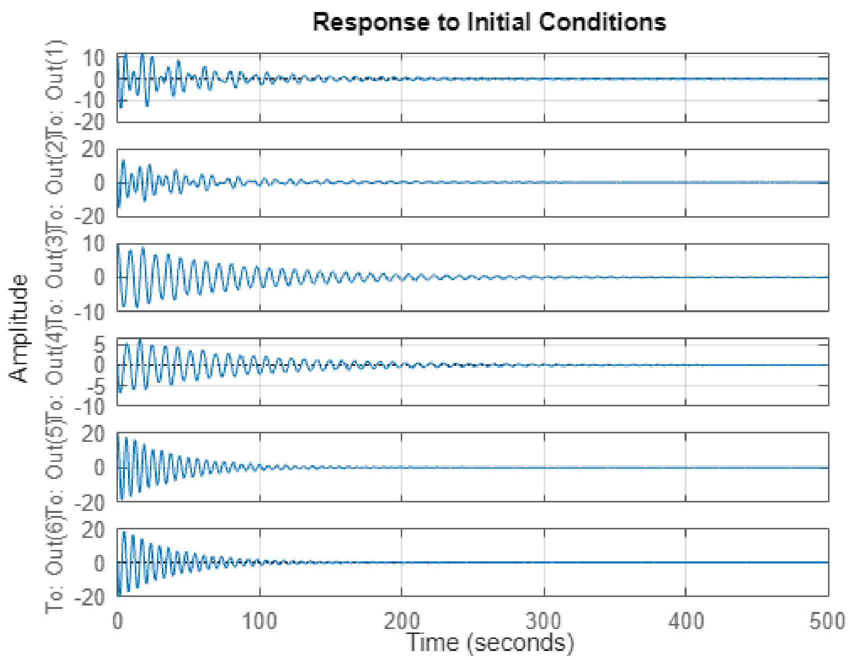


Figure 2: Response for Linearized Output after applying LQR

5.4 System Response after LQR is applied to Non-Linearized System:

We use same gain matrix from LQR (when applied to linearized system) and try to plot the response of the non-linearized system, we get the following system response

Initial Conditions

```
x_0 = [10; 0; 10; 0; 20; 0]
```

```
x_0 = 6×1

    10
     0
    10
     0
    20
     0
```

```
%defining the timespan
    tspan = 0:0.01:10000;
%usindg ode45 function for definining a diff eqn
    [t,x] = ode45(@nonlinear,tspan,x_0);
%plotting the function output on a 2D graph
    subplot(3, 2, 1);
    plot(t,x(:,1),'r')
    xlabel('Time in seconds')
```

```
    ylabel('Output x(t)')
    subplot(3, 2, 2);
    plot(t,x(:,2),'g')
    xlabel('Time in seconds')
    ylabel('Output x_dot(t)')
    subplot(3, 2, 3);
    plot(t,x(:,3),'b')
    xlabel('Time in seconds')
    ylabel('Output theta1(t)')
    subplot(3, 2, 4);
    plot(t,x(:,4),'c')
    xlabel('Time in seconds')
    ylabel('Output theta1_dot(t)')
    subplot(3, 2, 5);
    plot(t,x(:,5),'m')
    xlabel('Time in seconds')
    ylabel('Output theta2(t)')
    subplot(3, 2, 6);
    plot(t,x(:,6),'k')
    xlabel('Time in seconds')
    ylabel('Output theta2_dot(t)')
```

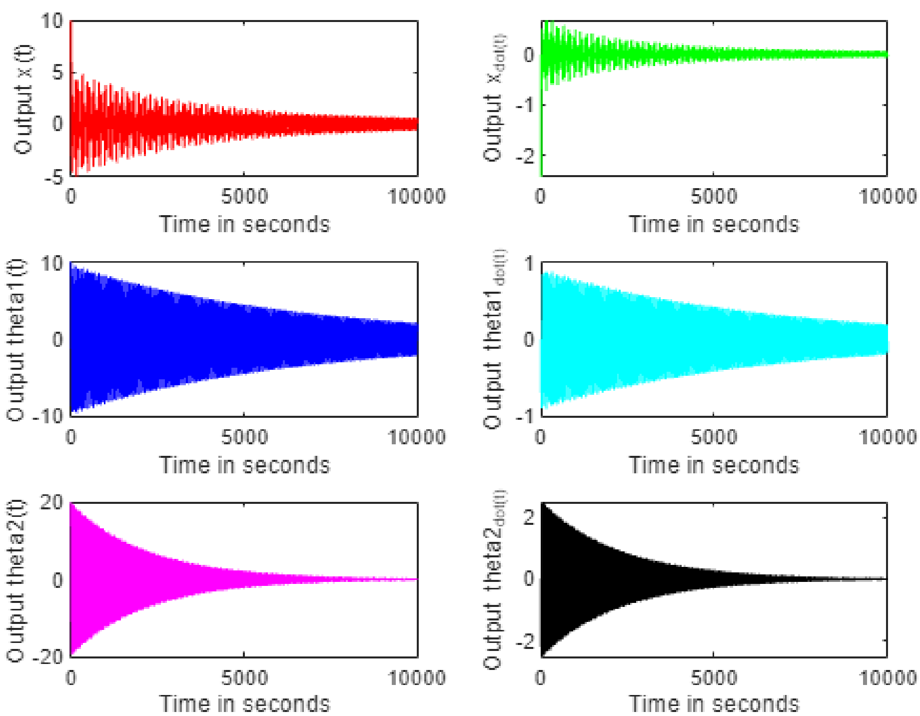


Figure3: plots when LQR applied to Non-linear System

```
function x_dot = nonlinear(t,x)


M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;

%Linearized equations
A = [0 1 0 0 0 0;
    0 0 (-g_val*m1_val)/M_val 0 (-g_val*m2_val)/M_val 0;
    0 0 0 1 0 0;
    0 0 -(g_val*(M_val+m1_val))/(M_val*l1_val) 0 -(g_val*m2_val)/(M_val*l1_val)
0;
    0 0 0 0 0 1;
    0 0 -(g_val*m1_val)/(M_val*l2_val) 0 -(g_val*(M_val+m2_val))/(M_val*l2_val)
0];
B = [0;
    1/M_val;
    0;
    1/(M_val*l1_val);
    0;
    1/(M_val*l2_val)];

% Defining Q and R for LQR definition
Q = [1000 0 0 0 0 0;  %x penalize
    0 100 0 0 0 0;   %x_dot penalize
    0 0 100 0 0 0;   %theta1 penalize
    0 0 0 100 0 0;   %theta1_dot penalize
    0 0 0 0 100 0;   %theta2 penalize
    0 0 0 0 0 100];  %theta2_dot penalize

R = 0.001;

% K from LQR
[K,~,~] = lqr(A, B, Q, R);

% Input eqaution
F = -K*x;

% Initial x_dot condition
x_dot = zeros(6,1);
```

```
% Defining the x_dot matrix
% x_dot(1) = X_dot
x_dot(1) = x(2);

%x_dot(2) = X_dotdot
x_dot(2) = (-(m1_val*g_val*sind(x(3))*cosd(x(5))) -
(m2_val*g_val*sind(x(5))*cosd(x(5))) -(m1_val*l1_val*(x(4))*(x(4))*sind(x(3))) -
(m2_val*l2_val*(x(6))*(x(6))*sind(x(5))) +F)/(M_val +m1_val*(1-cosd(x(3))^2)
+m2_val*(1-cosd(x(5))^2));

%x_dot(3) = theta1_dot
x_dot(3) = x(4);

%x_dot(4) = theta1_dotdot
x_dot(4) = (x_dot(2)*cosd(x(3))-g_val*(sind(x(3))))/l1_val';

%x_dot(5) = theta2_dot
x_dot(5) = x(6);

%x_dot(6) = theta2_dotdot
x_dot(6) = (x_dot(2)*cosd(x(5))-g_val*(sind(x(5))))/l2_val';

end
```

6.Lyapunov stability Test:

Lyapunov stability test is direct stability test criteria in which we check real part of eigen Values of closed loop matrix (A-B_K). So the following are real parts of eigen values of closed loop matrix

```
[v,d] = eig(A-B*k);
disp('eigen values');
```

```
eigen values
```

```
disp(diag(d))
```

```
 -0.8198 + 0.5293i
 -0.8198 - 0.5293i
 -0.0255 + 1.0065i
 -0.0255 - 1.0065i
 -0.0117 + 0.7059i
```

```
  -0.0117 - 0.7059i
```

```
disp(real(diag(d)));
```

```
  -0.8198
  -0.8198
  -0.0255
  -0.0255
  -0.0117
  -0.0117
```

Since all the eigen values are having negative real parts(poles in left half plane) the system is said to be stable

7.Observability of the system

Similar to controllability test we have observability test to check whether the system is observable or not. To test this we follow the following criteria

For this we check the rank of observability matrix which is given by

$$O = \begin{bmatrix} C \\ CA \\ . \\ . \\ CA^{n-1} \end{bmatrix}$$

And the system is said to be observable when the rank of observable matrix is a full rank

So, the observability test is followed as below

7.1 Observability Test

```
syms M m1 m2 l1 l2 g
% g = 9.81;
A = [0 1 0 0 0 0;
     0 0 (-g*m1)/M 0 (-g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 -(g*(M+m1))/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

B = [0;
    1/M;
    0;
    1/(M*l1);
    0;
```

```
    1/(M*l2)];

C_01 = [1 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0];

C_02 = [0 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];

C_03 = [1 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 1 0;
    0 0 0 0 0 0];

C_04 = [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 = 0;
M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;
A = double(subs(A, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
B = double(subs(B, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
```

Defining the System

```
sys1 = ss(A,B,C_01,D);
sys2 = ss(A,B,C_02,D);
```

```
sys3 = ss(A,B,C_03,D);
sys4 = ss(A,B,C_04,D);
```

Observability

```
O1 = obsv(sys1);
O2 = obsv(sys2);
O3 = obsv(sys3);
O4 = obsv(sys4);
```

Rank of Observability Matrix

```
disp(rank(O1))
```

```
    6
```

```
disp(rank(O2))
```

```
    4
```

```
disp(rank(O3))
```

```
    6
```

```
disp(rank(O4))
```

```
    6
```

```
rankarray = [rank(O1), rank(O2), rank(O3), rank(O4)];
```

Print which cases are Observable.

```
if(rank(O1) == 6)
    disp('System is Observable for Output vector x(t) is only passed')
else
    disp('System is not Observable for Output vector x(t) is only passed')
end
```

```
System is Observable for Output vector x(t) is only passed
```

```
if(rank(O2) == 6)
    disp('System is Observable for Output vector theta1(t) and theta2(t) is only passed')
else
    disp('System is not Observable for Output vector theta1(t) and theta2(t) is only passed')
end
```

```
System is not Observable for Output vector theta1(t) and theta2(t) is only passed
```

```
if(rank(O3) == 6)
    disp('System is Observable for Output vector x(t) and theta2(t) is only passed')
else
    disp('System is not Observable for Output vector x(t) and theta2(t) is only passed')
end
```

```
System is Observable for Output vector x(t) and theta2(t) is only passed
```

```
if(rank(O4) == 6)
    disp('System is Observable for Output vector x(t), theta1(t) and theta2(t) is only passed')
else
    disp('System is not Observable for Output vector x(t), theta1(t) and theta2(t) is only passed')
end
```

System is Observable for Output vector x(t), theta1(t) and theta2(t) is only passed

8.Luenberger Observability

The Luenberger Observer, also known as the Luenberger state observer, is a widely utilized tool in control theory for estimating the unmeasurable or difficult-to-measure states of a dynamic system. This observer provides a means to infer the internal state variables of a system by using the available input and output measurements. The primary motivation behind employing Luenberger Observers is to enhance the observability of a system, enabling the estimation of states that are not directly measurable.

8.1 Linear System

$$\hat{\mathrm{X}}(\mathrm{t}) = \mathrm{A}\hat{\mathrm{X}}(\mathrm{t}) + \mathrm{Bu}(\mathrm{t}) + L(Y(t) - C\hat{X}(t))$$

$$\mathrm{Y}(\mathrm{t}) = \mathrm{CX}(\mathrm{t})$$

Here $\hat{\mathrm{X}}(\mathrm{t})$ is the desired state

L is observer gain matrix

$$\dot{\mathrm{X}}_{\mathrm{e}}(\mathrm{t}) = (\mathrm{A} - \mathrm{LC})\mathrm{X}_{\mathrm{e}}(\mathrm{t}) + \mathrm{B}_{\mathrm{D}}\mathrm{u}_{\mathrm{D}}(\mathrm{t})$$

Where $\dot{\mathrm{X}}_{\mathrm{e}}(\mathrm{t})$ is the state estimation error which is given by X(t)- $\hat{\mathrm{X}}(\mathrm{t})$

$$A = \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix}$$

$$B = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$C = [C\ 0]$$

```
syms M m1 m2 l1 l2 g
% g = 9.81;
A = [0 1 0 0 0 0;
     0 0 (-g*m1)/M 0 (-g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 -(g*(M+m1))/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

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

C_01 = [1 0 0 0 0 0];

C_02 = [0 0 1 0 0 0;
    0 0 0 0 1 0];

C_03 = [1 0 0 0 0 0;
    0 0 0 0 1 0];

C_04 = [1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];

D = 0;
M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;
A = double(subs(A, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
B = double(subs(B, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
```

Defining K using LQR

```
Q = [1000 0 0 0 0 0;  %x penalize
     0 1000 0 0 0 0;  %x_dot penalize
```

```
     0 0 1 0 0 0;   %theta1 penalize
     0 0 0 1 0 0;   %theta1_dot penalize
     0 0 0 0 1 0;   %theta2 penalize
     0 0 0 0 0 1];  %theta2_dot penalize
% Q =eye(6)*100;
R = 0.001;
[K,~,~] = lqr(A, B, Q, R);
disp(K);
```

```
  1.0e+03 *


    1.0000    1.8191   -0.2725   -0.9704   -0.1740   -0.5649
```

Defining the system before using the observer

```
sys1 = ss(A,B,C_01,D);
sys2 = ss(A,B,C_02,D);
sys3 = ss(A,B,C_03,D);
sys4 = ss(A,B,C_04,D);
```

Finding L

```
p = [-1; -2; -3; -4; -5; -6];
L1 = place(A',C_01',p)';
disp(L1);
```

```
  1.0e+03 *


    0.0210
    0.1734
   -2.9262
    0.0805
    2.2116
   -1.4493
```

```
L3 = place(A',C_03',p)';
disp(L3);
```

```
   13.0744   -0.8244
   56.2562   -8.4805
  -89.0764   19.7693
  -20.0115   10.9419
```

```
    0.3520    7.9256
    3.4793   13.2122
```

```
L4 = place(A',C_04',p)';
disp(L4);
```

```
    8.5631   -0.8851    0.0000
   17.5219   -4.9484   -0.9810
   -0.9140    9.4369   -0.0000
   -4.1173   20.9385   -0.0491
    0.0000   -0.0000    3.0000
    0.0000   -0.0981    0.9209
```

Define The A_Cap, B_Cap, C_Cap

```
A_Cap_1 = [(A-B*K) B*K;
    zeros(size(A)) (A-L1*C_01)];
C_Cap_1 = [C_01 zeros(size(C_01))];

A_Cap_2 = [(A-B*K) B*K;
    zeros(size(A)) (A-L3*C_03)];
C_Cap_2 = [C_03 zeros(size(C_03))];

A_Cap_3 = [(A-B*K) B*K;
    zeros(size(A)) (A-L4*C_04)];
C_Cap_3 = [C_04 zeros(size(C_04))];

B_Cap = [B;zeros(size(B))];
```

System with Luenberger

```
sys_1 = ss(A_Cap_1, B_Cap, C_Cap_1, D);
sys_2 = ss(A_Cap_2, B_Cap, C_Cap_2, D);
sys_3 = ss(A_Cap_3, B_Cap, C_Cap_3, D);
```

Graph Print for all the N

```
x_0 = [10;0;10;0;20;0;0;0;0;0;0;0];
% Display Plot of x(t)
disp('');
disp('Plot of X(t) when only x(t) is expected and step Graph')
```

```
Plot of X(t) when only x(t) is expected and step Graph
```

```
figure
initial(sys_1,x_0)
```

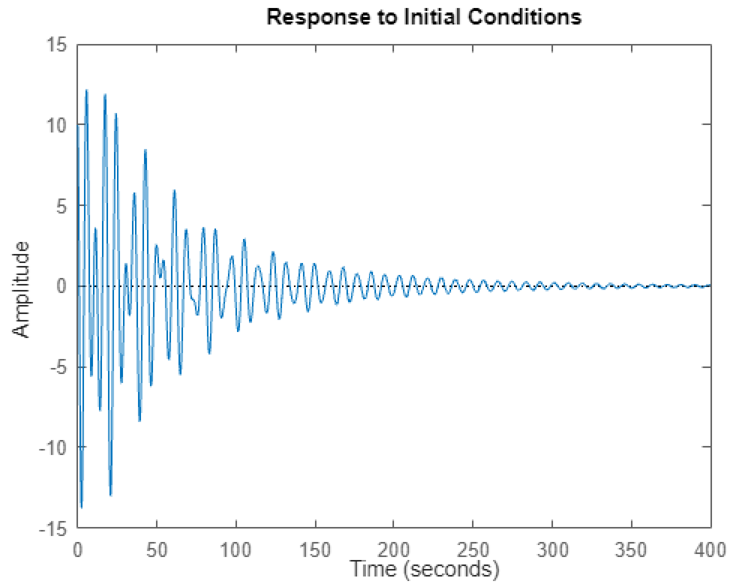


Figure 4: Response Graph of x(t)

```
figure
step(sys_1)
```

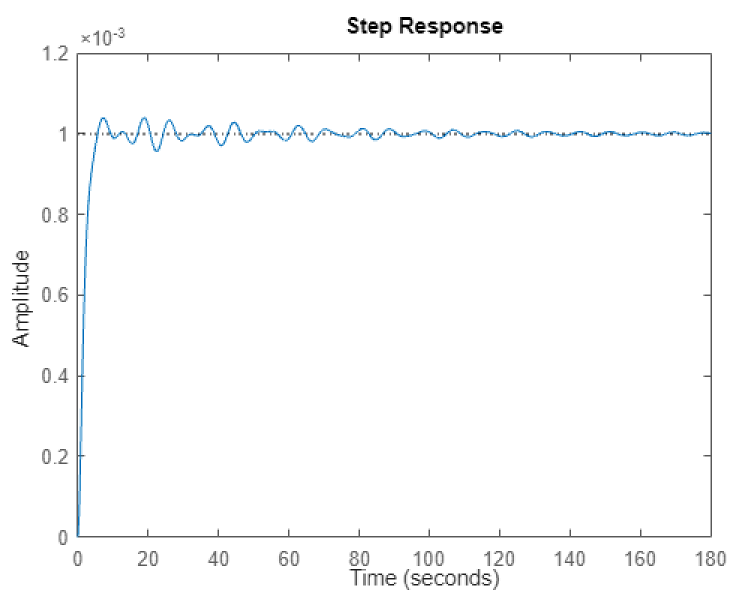


Figure 5: Step of the response x(t)

```
disp('')
```

```
disp('Plot of X(t) and theta2(t) when only x(t) and theta(2) is expected and step Graph')
```

Plot of X(t) and theta2(t) when only x(t) and theta(2) is expected and step Graph:

```
figure
initial(sys_2(1),x0)
```

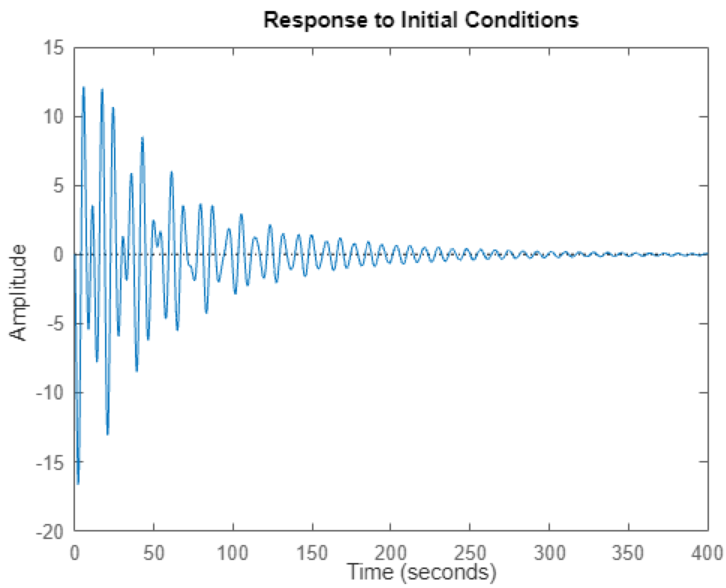


Figure 6: Response Graph of x(t)

```
step(sys_2(1))
```

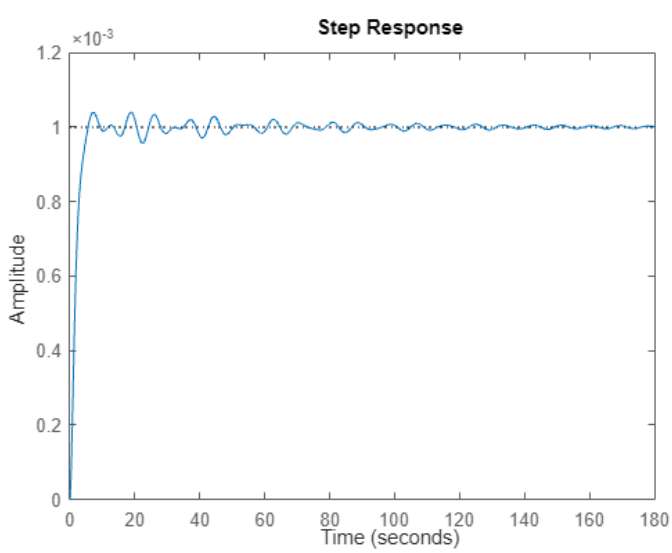


Figure 7: Step Response Graph of x(t)

```
figure
initial(sys_2(2),x0)
```

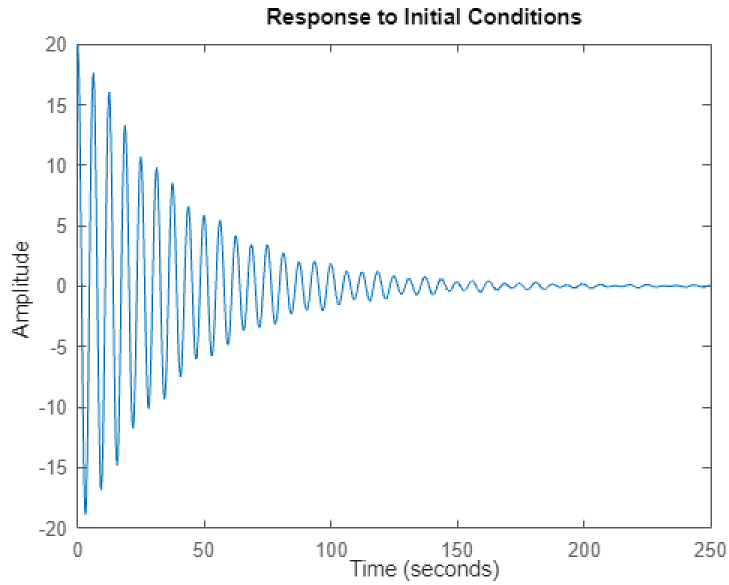


Figure 8: Response Graph of $\theta_2(t)$

```
step(sys_2(2))
```

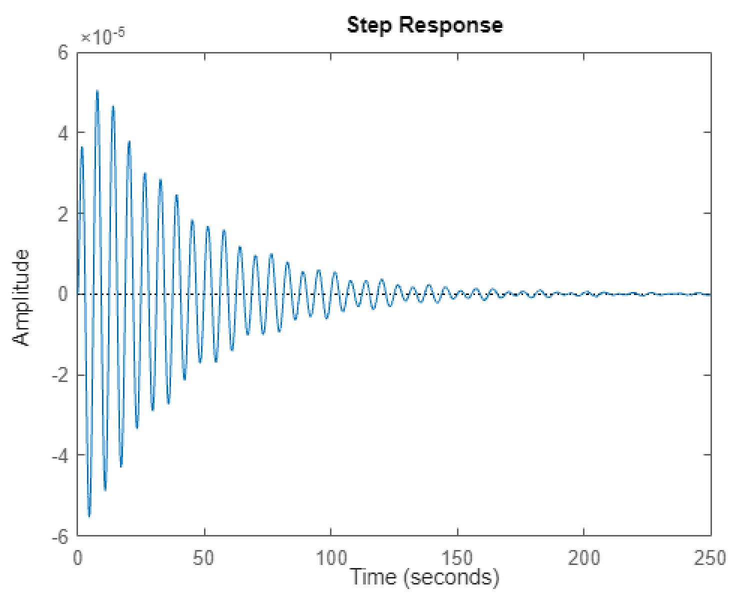


Figure 9: Step Response Graph of $\theta_2(t)$

```
disp('');
disp('Plot of X(t), theta1(t) and theta2(t) when only x(t), theta1(t) and theta2(t) is expected and step Graph')
```

```
Plot of X(t), theta1(t) and theta2(t) when only x(t), theta1(t) and theta2(t) is expected and step Graph
```

```
figure
initial(sys_3(1),x0)
```

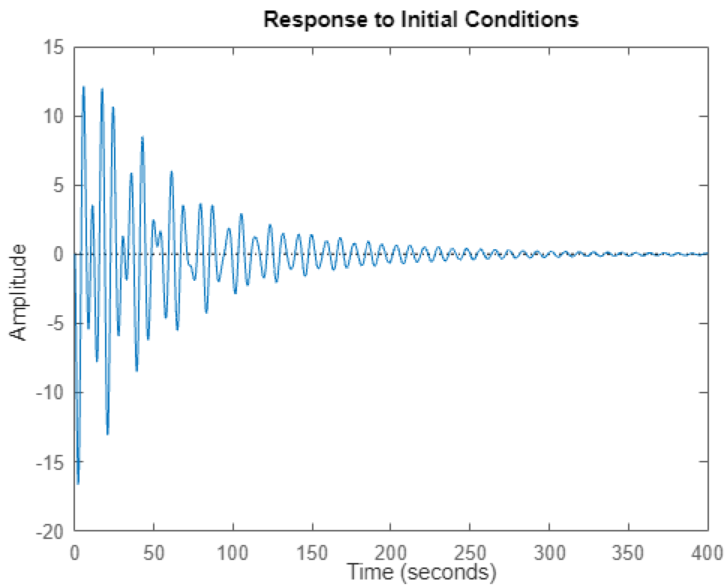


Figure 10: Response Graph of x(t)

```
step(sys_3(1))
```

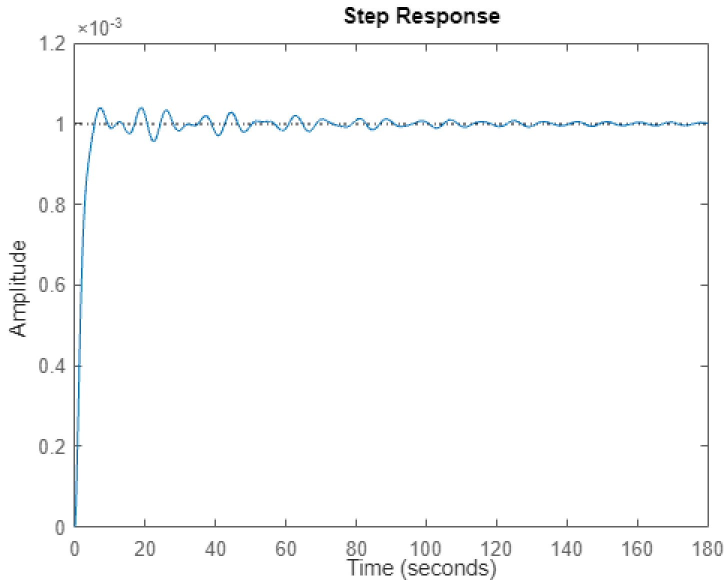


Figure 11: Step Response Graph of x(t)

```
figure
initial(sys_3(2),x0)
```

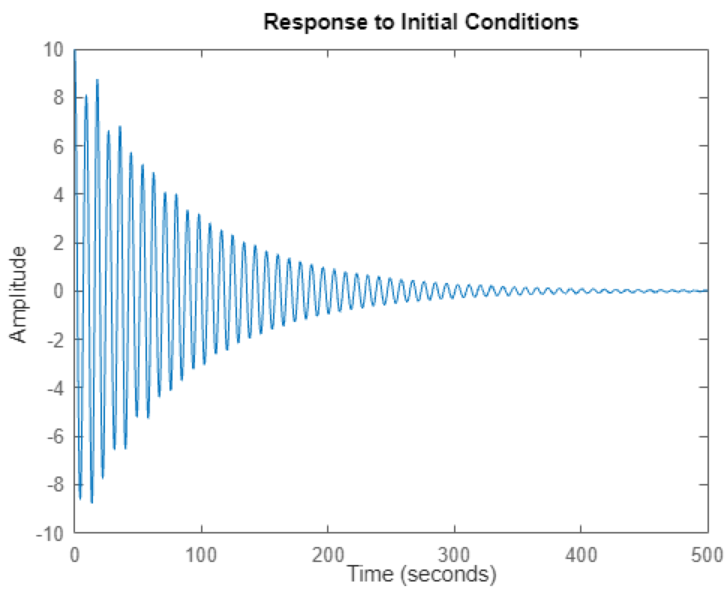


Figure 12: Response Graph of $\theta_1(t)$

```
step(sys_3(2))
```

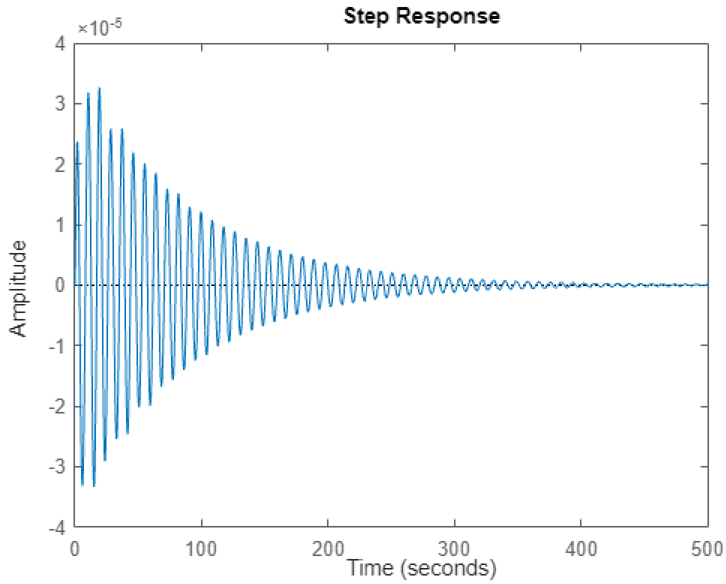


Figure 13: Step Response Graph of $\theta_1(t)$

```
figure
initial(sys_3(3),x0)
```

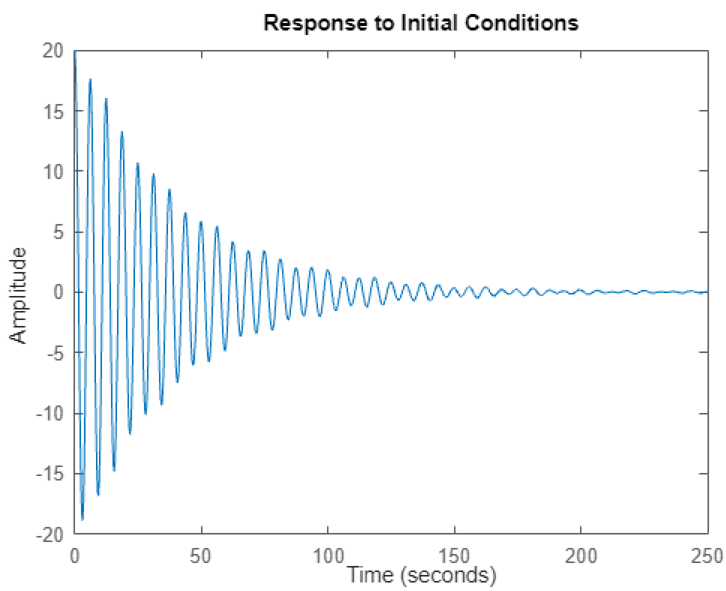


Figure 14: Response Graph of $\theta_2(t)$

```
step(sys_3(3))
```

```
grid on
```

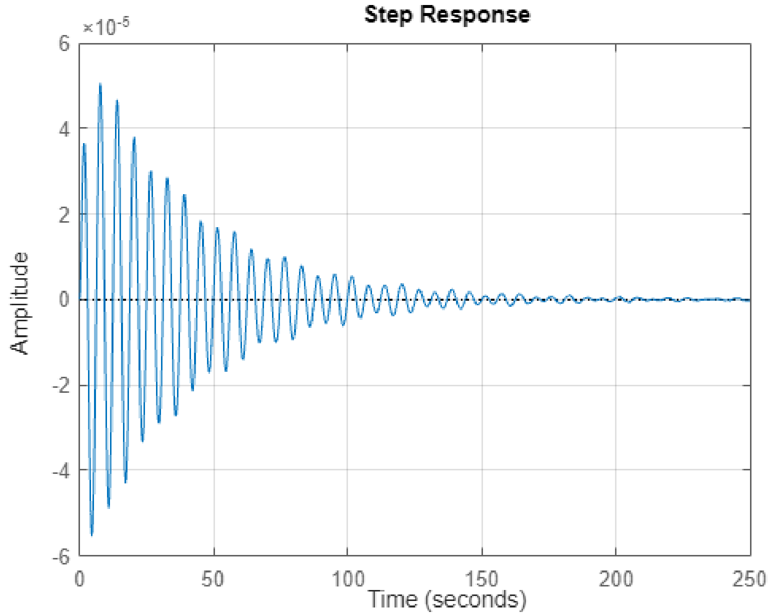


Figure 15: Step Response Graph of $\theta_2(t)$

8.2 Non-Linear Equation

```
%clearing all the previous outputs
clc
clear
% M = 1000Kg, m1 = m2 = 100Kg, l1 = 20m and l2 = 10m substitute in A&B and then
calculate controllability matrix
M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;

%Linearized equations
A = [0 1 0 0 0 0;
    0 0 (-g_val*m1_val)/M_val 0 (-g_val*m2_val)/M_val 0;
    0 0 0 1 0 0;
    0 0 -(g_val*(M_val+m1_val))/(M_val*l1_val) 0 -(g_val*m2_val)/(M_val*l1_val)
0;
    0 0 0 0 0 1;
    0 0 -(g_val*m1_val)/(M_val*l2_val) 0 -(g_val*(M_val+m2_val))/(M_val*l2_val)
0];
B = [0;
```

```
    1/M_val;
    0;
    1/(M_val*l1_val);
    0;
    1/(M_val*l2_val)];

C_01 = [1 0 0 0 0 0];

C_02 = [0 0 1 0 0 0;
    0 0 0 0 1 0];

C_03 = [1 0 0 0 0 0;
    0 0 0 0 1 0];

C_04 = [1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];
D = 0;


Q = [1000 0 0 0 0 0;  %x penalize
     0 1000 0 0 0 0;  %x_dot penalize
     0 0 1 0 0 0;   %theta1 penalize
     0 0 0 1 0 0;   %theta1_dot penalize
     0 0 0 0 1 0;   %theta2 penalize
     0 0 0 0 0 1];  %theta2_dot penalize
R=0.001;
```

Defining the system before using the observer

```
sys1 = ss(A,B,C_01,D);
sys2 = ss(A,B,C_02,D);
sys3 = ss(A,B,C_03,D);
sys4 = ss(A,B,C_04,D);
```

Finding L

```
p = [-1; -2; -3; -4; -5; -6];
L1 = place(A',C_01',p)';
disp(L1);
```

```
  1.0e+03 *


    0.0210

    0.1734
```

```
  -2.9262

   0.0805

   2.2116

  -1.4493
```

```
L3 = place(A',C_03',p)';
disp(L3);
```

```
  13.0744   -0.8244

  56.2562   -8.4805

 -89.0764   19.7693

 -20.0115   10.9419

   0.3520    7.9256

   3.4793   13.2122
```

```
L4 = place(A',C_04',p)';
disp(L4);
```

```
   8.5631   -0.8851    0.0000

  17.5219   -4.9484   -0.9810

  -0.9140    9.4369   -0.0000

  -4.1173   20.9385   -0.0491

   0.0000   -0.0000    3.0000

   0.0000   -0.0981    0.9209
```

```
[K,~,~] = lqr(A, B, Q, R);
disp(K);
```

```
  1.0e+03 *


   1.0000    1.8191   -0.2725   -0.9704   -0.1740   -0.5649
```

System with Luenberger Non -Linear

8.2.1 if x(t) is the Input

When the system has an input of x(t), the l matrix is given by L1 and the resultant graphs are Figure16.

```
L_poles = [-4; -4.5; -5; -5.5; -6; -6.5];
% Specifying the initial condition where estimated states
% are initialized to zero
```

```
x0 = [0;0;10;0;20;0;0;0;0;0;0;0];


% The Luenberger observer obtained when x(t) is observed
% The initial condition
x0_obs = [10;0;10;0;20;0];

t_span = 0:0.01:10000;
% Plotting x(t) for non-linear system when x(t) is observed
[ts,x_dots_L1] = ode45(@(t,x)non_lin_sys(t,x,-K*x,L1,C_01),t_span,x0_obs);
figure
plot(ts,x_dots_L1(:,1))
xlabel('Time in seconds')
ylabel('Output x(t)')
title('Behaviour of system when x(t) is observed')
```

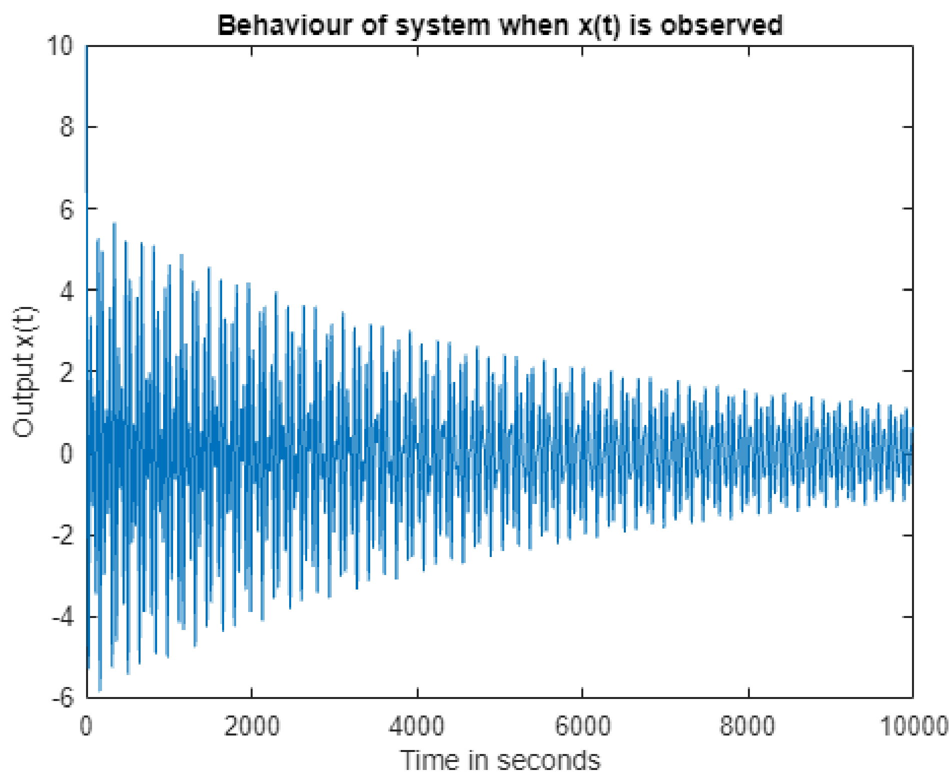


Figure 16: x(t) graph for L1 observer

8.2.2 if x(t) and $\theta_2(t)$is the Input

When the system has an input of x(t) and $\theta_2(t)$, the l matrix is given by L2 and the resultant graphs are

```
% Checking the response of system when x(t) and theta2(t) are observed
[ts,x_dots_L3] = ode45(@(t,x)non_lin_sys3(t,x,-K*x,L3,C_03),t_span,x0_obs);
figure
plot(ts,x_dots_L3(:,1))
xlabel('Time in seconds')
ylabel('Output x(t)')
title('Behaviour of system when x(t)')
```

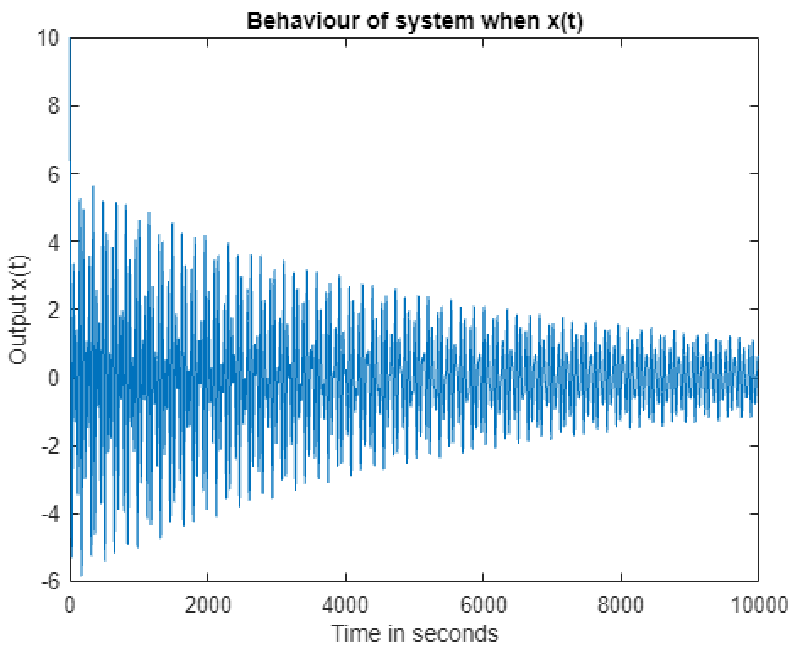


Figure 17: x(t) graph for L2 observer

```
% plot(ts,x_dots_L3(:,5))

figure

plot(ts,x_dots_L3(:,5))
xlabel('Time in seconds')
ylabel('Output theta2(t)')
title('Behaviour of system when $\theta_2 (t)')
```

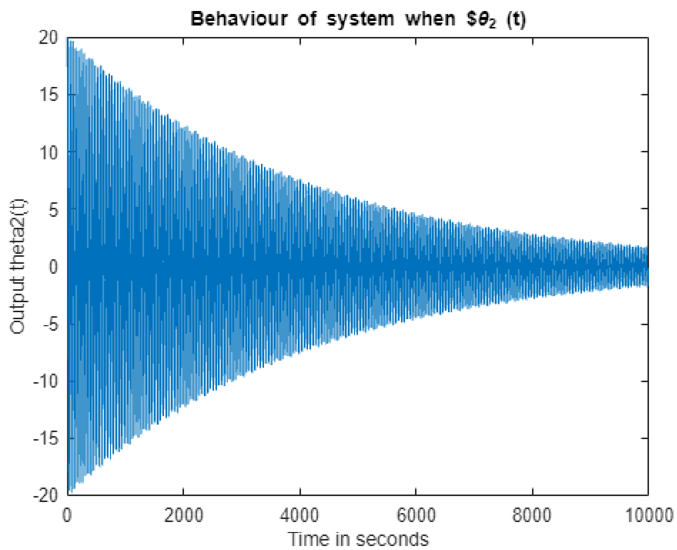


Figure 18: $\theta_2(t)$ graph for L2 observer

8.2.3 if x(t), $\theta_1(t)$ and $\theta_2(t)$is the Input

When the system has an input of x(t) and $\theta_2(t)$, the l matrix is given by L2 and the resultant graphs are

```
% Checking the response of system when x(t), theta1(t) and
% theta2(t) are observed
figure

[ts,x_dots_L4] = ode45(@(t,x)non_lin_sys4(t,x,-K*x,L4,C_04),t_span,x0_obs);
figure
plot(ts,x_dots_L3(:,1))
xlabel('Time in seconds')
ylabel('Output x(t)')
title('Behaviour of system when x(t)')
```

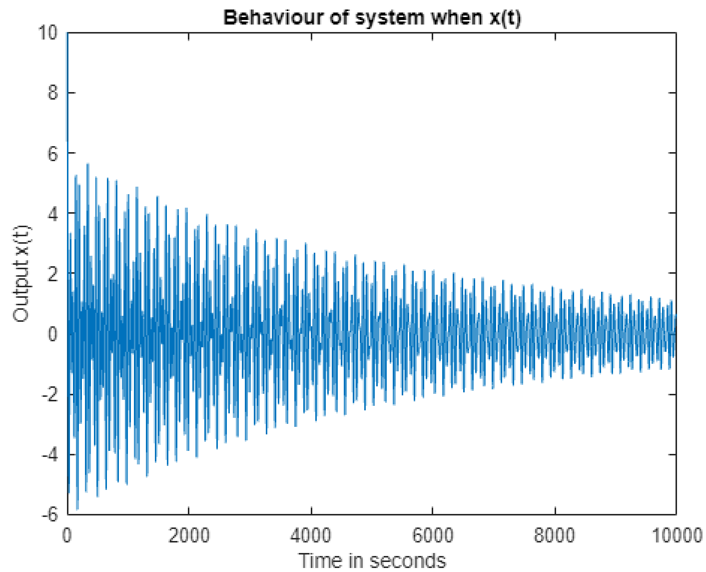


Figure 19: x(t) graph for L3 observer

```
% plot(ts,x_dots_L3(:,5))

figure
plot(ts,x_dots_L3(:,3))
xlabel('Time in seconds')
ylabel('Output $\theta_1 (t)')
title('Behaviour of system when $\theta_1 (t)')
```

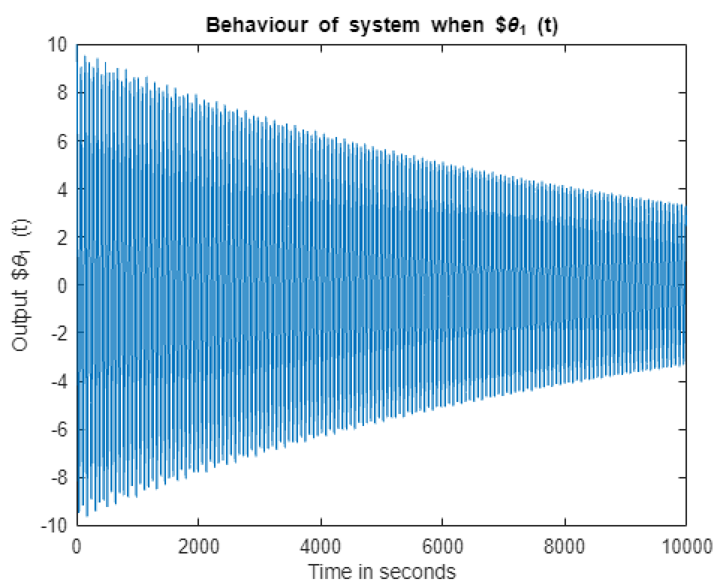


Figure 20: $\theta_1(t)$ graph for L3 observer

```
figure
```

```
plot(ts,x_dots_L3(:,5))
xlabel('Time in seconds')
ylabel('Output $\theta_2 (t)')
title('Behaviour of system when $\theta_2 (t)')
```

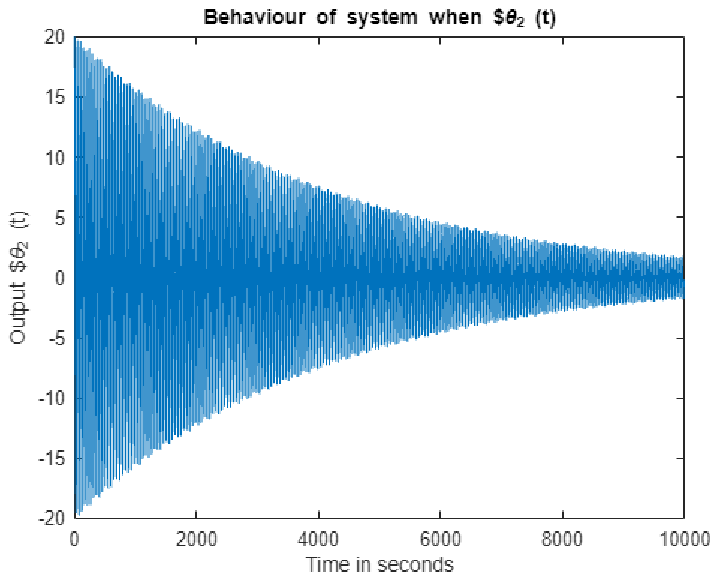


Figure 21: $\theta_2(t)$ graph for L3 observer

```
function x_dot = non_lin_sys(t,X,F,L,C)
x_dot = zeros(6,1);
% Declaring the values of system variables
M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;
x = X(1);
x_d = X(2);
theta1 = X(3);
theta1_d = X(4);
theta2 = X(5);
theta2_d = X(6);
obs = L*(x-C*X);
x_dot(1) = x_d + obs(1);
x_dot(2) = (F-
((m1_val*sind(theta1)*cosd(theta1))+(m2_val*sind(theta2)*cosd(theta2)))*g_val -
(l1_val*m1_val*(x_dot(3)^2)*sind(theta1)) -
(l2_val*m2_val*(x_dot(5)^2)*sind(theta2)))/(m1_val+m2_val+M_val-
(m1_val*(cosd(theta1)^2))-(m2_val*(cosd(theta2)^2)))+obs(2);
```

```
 x_dot(3) = theta1_d+obs(3);
 x_dot(4) = ((cosd(theta1)*x_dot(2)-g_val*sind(theta1))/l1_val) + obs(4);
 x_dot(5) = theta2_d + obs(5);
 x_dot(6) = (cosd(theta2)*x_dot(2)-g_val*sind(theta2))/l2_val + obs(6);
 end


 function x_dot = non_lin_sys3(t,X,F,L,C)
 x_dot = zeros(6,1);
 % Declaring the values of system variables
 M_val = 1000;
 m1_val = 100;
 m2_val = 100;
 l1_val = 20;
 l2_val = 10;
 g_val = 9.81;

 x = X(1);
 dx = X(2);
 theta1 = X(3);
 theta1_d = X(4);
 theta2 = X(5);
 theta2_d = X(6);
 y3 = [x; theta2];
 sum = L*(y3-C*X);
 x_dot(1) = dx + sum(1);
 x_dot(2) = (F-
((m1_val*sind(theta1)*cosd(theta1))+(m2_val*sind(theta2)*cosd(theta2)))*g_val -
(l1_val*m1_val*(x_dot(3)^2)*sind(theta1)) -
(l2_val*m2_val*(x_dot(5)^2)*sind(theta2)))/(m1_val+m2_val+M_val-
(m1_val*(cosd(theta1)^2))-(m2_val*(cosd(theta2)^2)))+sum(2);
 x_dot(3) = theta1_d+sum(3);
 x_dot(4) = ((cosd(theta1)*x_dot(2)-g_val*sind(theta1))/l1_val) + sum(4);
 x_dot(5) = theta2_d + sum(5);
 x_dot(6) = (cosd(theta2)*x_dot(2)-g_val*sind(theta2))/l2_val + sum(6);
 end


 function x_dot = non_lin_sys4(t,X,F,L,C)
 x_dot = zeros(6,1);
 % Declaring the values of system variables
 M_val = 1000;
 m1_val = 100;
 m2_val = 100;
 l1_val = 20;
 l2_val = 10;
```

```
g_val = 9.81;

x = X(1);
dx = X(2);
theta1 = X(3);
theta1_d = X(4);
theta2 = X(5);
theta2_d = X(6);
y4 = [x; theta1; theta2];
sum = L*(y4-C*X);
x_dot(1) = dx + sum(1);
x_dot(2) = (F-
((m1_val*sind(theta1)*cosd(theta1))+(m2_val*sind(theta2)*cosd(theta2)))*g_val -
(l1_val*m1_val*(x_dot(3)^2)*sind(theta1)) -
(l2_val*m2_val*(x_dot(5)^2)*sind(theta2)))/(m1_val+m2_val+M_val-
(m1_val*(cosd(theta1)^2))-(m2_val*(cosd(theta2)^2)))+sum(2);
x_dot(3) = theta1_d+sum(3);
x_dot(4) = ((cosd(theta1)*x_dot(2)-g_val*sind(theta1))/l1_val) + sum(4);
x_dot(5) = theta2_d + sum(5);
x_dot(6) = (cosd(theta2)*x_dot(2)-g_val*sind(theta2))/l2_val + sum(6);
end
```

9.LQG Controller:

The LQG (Linear Quadratic Gaussian) controller is a control strategy for linear time-invariant systems that integrates optimal control (LQR) and state estimation through the Kalman filter. Mathematically, the LQG formulation is defined by a set of differential and algebraic equations describing the closed-loop system dynamics and the associated matrices involved.

We just assume that process noise covariance as identity matrix (*I)*

and measurement noise covariance as 1

from LQG we get the optimal observer gain matrix as

$$L = PC^T\Sigma_V^{-1}$$

Where *P* can be found by following Ricatti Equation

$$AP + PA^T + B_D\Sigma_D B_D^T - PC^T\Sigma_V^{-1}CP = 0$$

So when LQG is applied to our linear system the following response is observed

9.1 LQG For Linear System:

Code snippet:

```
syms M m1 m2 l1 l2 g
% g = 9.81;
A = [0 1 0 0 0 0;
     0 0 (-g*m1)/M 0 (-g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 -(g*(M+m1))/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

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

C_01 = [1 0 0 0 0 0];

C_02 = [0 0 1 0 0 0;
    0 0 0 0 1 0];

C_03 = [1 0 0 0 0 0;
    0 0 0 0 1 0];

C_04 = [1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];

D = 0;
M_val = 1000;
m1_val = 100;
m2_val = 100;
l1_val = 20;
l2_val = 10;
g_val = 9.81;
A = double(subs(A, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
B = double(subs(B, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
```

Defining K using LQR

```
Q = [1000 0 0 0 0 0;  %x penalize
     0 1000 0 0 0 0;  %x_dot penalize
     0 0 1 0 0 0;   %theta1 penalize
     0 0 0 1 0 0;   %theta1_dot penalize
     0 0 0 0 1 0;   %theta2 penalize
     0 0 0 0 0 1];  %theta2_dot penalize
% Q =eye(6)*100;
R = 0.001;
[K,~,~] = lqr(A, B, Q, R);
disp(K);
```

```
  1.0e+03 *


    1.0000    1.8191   -0.2725   -0.9704   -0.1740   -0.5649
```

Initialization

```
x_0 = [10;0;10;0;20;0;0;0;0;0;0;0];
vd = 0.5*eye(6); %Process Noise
vn = 1; %measurement Noise
```

Finding L

```
p = [-1; -2; -3; -4; -5; -6];
L1 = place(A',C_01',p)';
disp(L1);
```

```
  1.0e+03 *


    0.0210

    0.1734

   -2.9262

    0.0805

    2.2116

   -1.4493
```

```
L3 = place(A',C_03',p)';
disp(L3);
```

```
  13.0744   -0.8244
```

```
  56.2562   -8.4805
 -89.0764   19.7693
 -20.0115   10.9419
   0.3520    7.9256
   3.4793   13.2122
```

```
L4 = place(A',C_04',p)';
disp(L4);
```

```
   8.5631   -0.8851    0.0000
  17.5219   -4.9484   -0.9810
  -0.9140    9.4369   -0.0000
  -4.1173   20.9385   -0.0491
   0.0000   -0.0000    3.0000
   0.0000   -0.0981    0.9209
```

k for each system

```
k1 = lqr(A',C_01',vd,vn);
k2 = lqr(A',C_03',vd,vn);
k3 = lqr(A',C_04',vd,vn);
```

System 1 with x(t)

```
A_c1 = [(A-B*K) B*K ; zeros(size(A)) (A-(k1'*C_01))];
B_c1 = [B;B];
C_c1 = [C_01 zeros(size(C_01))];

sys_1 = ss(A_c1,B_c1,C_c1,D);
figure;
initial(sys_1,x_0);
```

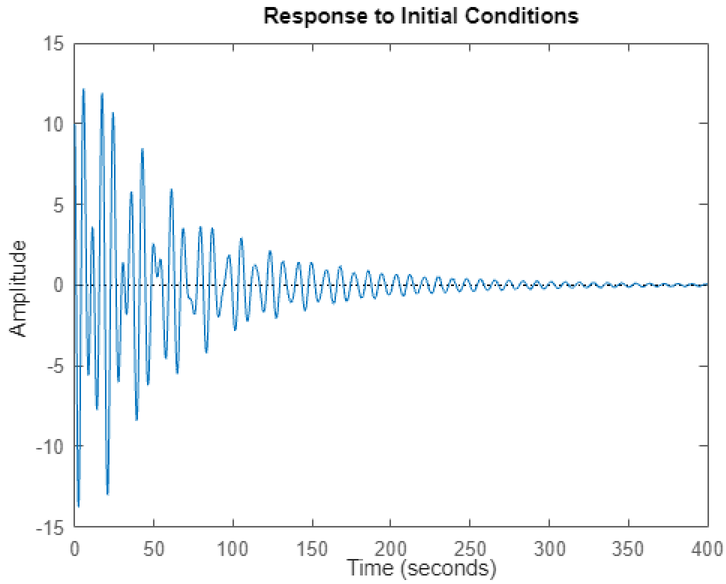


Figure 22: x(t) graph for L1 observer

```
figure;
step(sys_1);
```

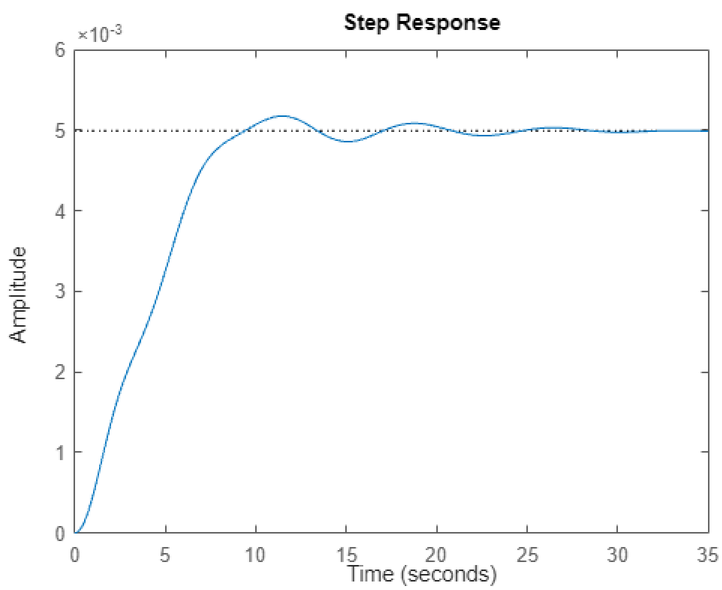


Figure 23: Step Response of x(t) graph for L1 observer

```
disp('')
```

```
disp('Plot of X(t) and theta2(t) when only x(t) and theta(2) is expected and step Graph')
```

```
Plot of X(t) and theta2(t) when only x(t) and theta(2) is expected and step Graph
```

```
A_c3 = [(A-B*K) B*K ; zeros(size(A)) (A-(k2'*C_03))];
B_c3 = [B;B];
C_c3 = [C_03 zeros(size(C_03))];

sys_2 = ss(A_c3,B_c3,C_c3,D);
figure
initial(sys_2(1),x_0)
```

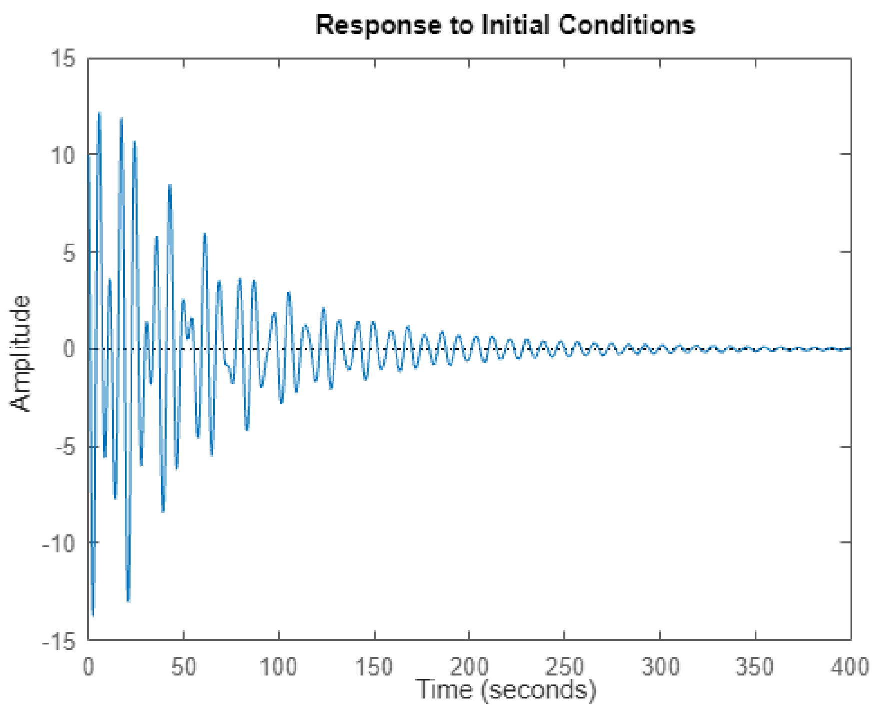


Figure 23: Response of x(t) graph for L2 observer

```
step(sys_2(1))
```

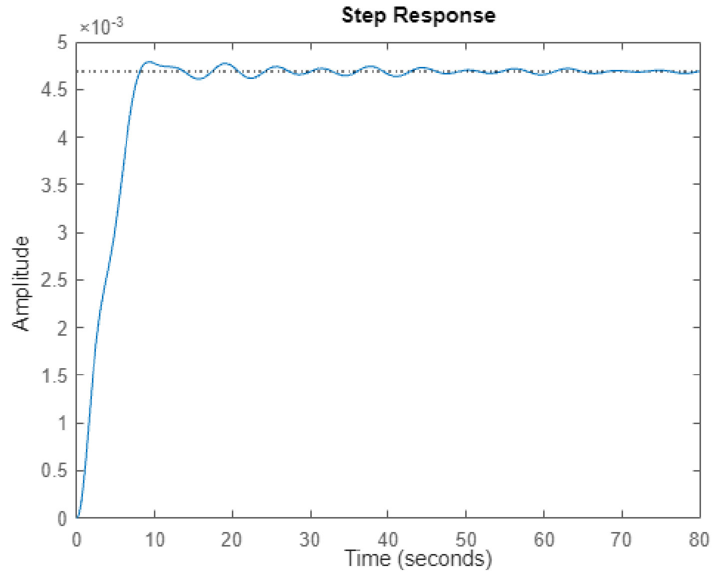


Figure 24: Step Response of $x(t)$ graph for L2 observer

```
figure
initial(sys_2(2),x_0)
```

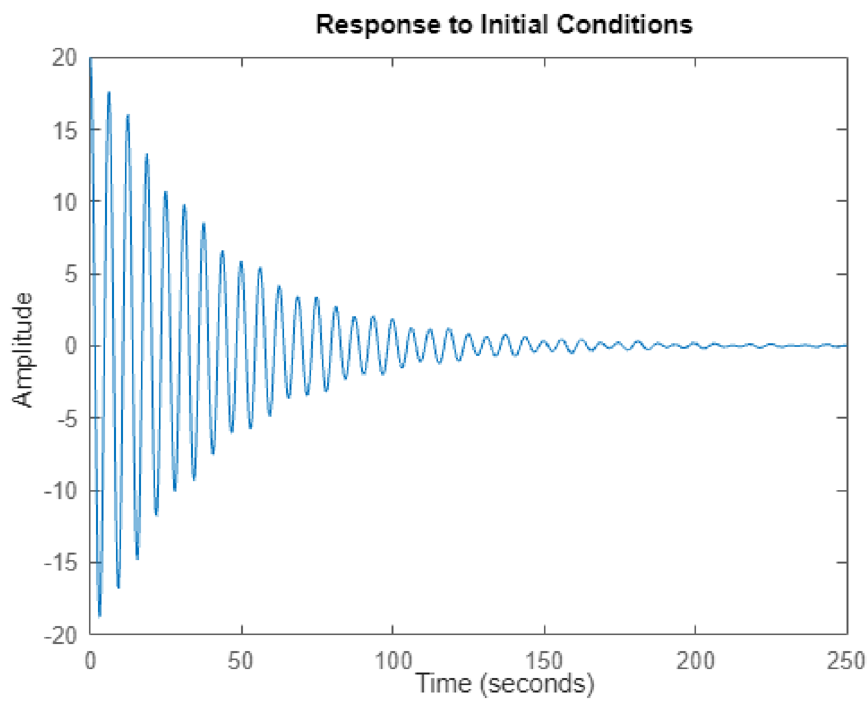


Figure 25: Response of $\theta_2(t)$ graph for L2 observer

```
step(sys_2(2))
```

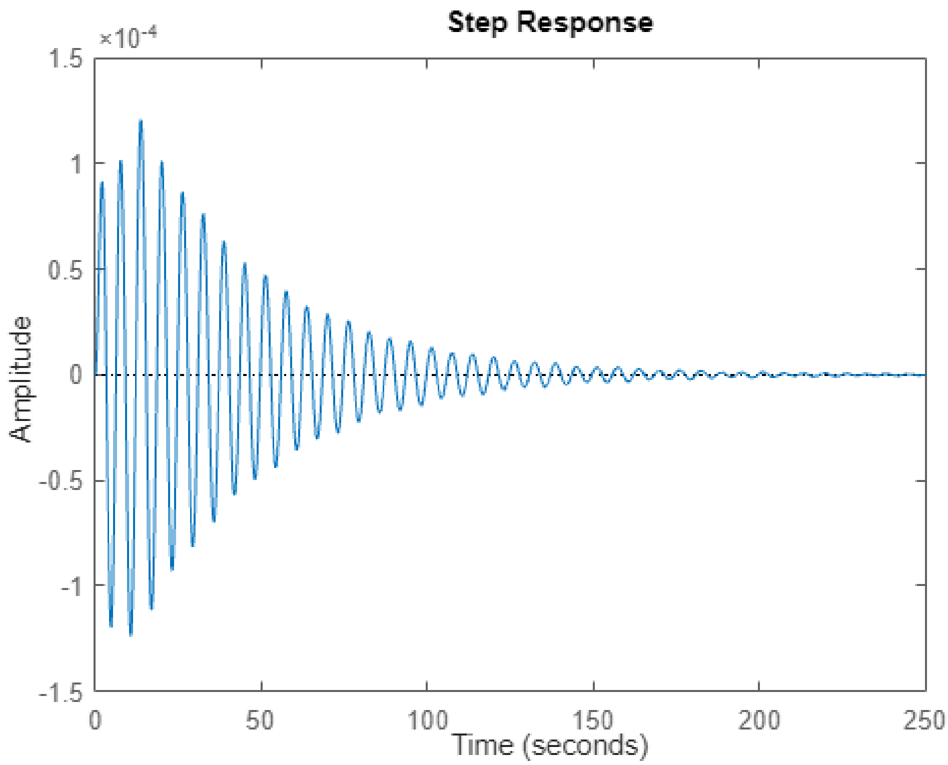


Figure 26: Step Response of $\theta_2(t)$ graph for L2 observer

System 3 with x(t), theta1(t) and theta2(t)

```
disp('')
disp('Plot of X(t) and theta2(t) when only x(t) and theta(2) is expected and
step Graph')
```

```
Plot of X(t) and theta2(t) when only x(t) and theta(2) is expected and step Graph
```

```
A_c4 = [(A-B*K) B*K ; zeros(size(A)) (A-(k3'*C_04))];
B_c4 = [B;B];
C_c4 = [C_04 zeros(size(C_04))];

sys_3 = ss(A_c4,B_c4,C_c4,D);
figure
initial(sys_3(1),x0)
```

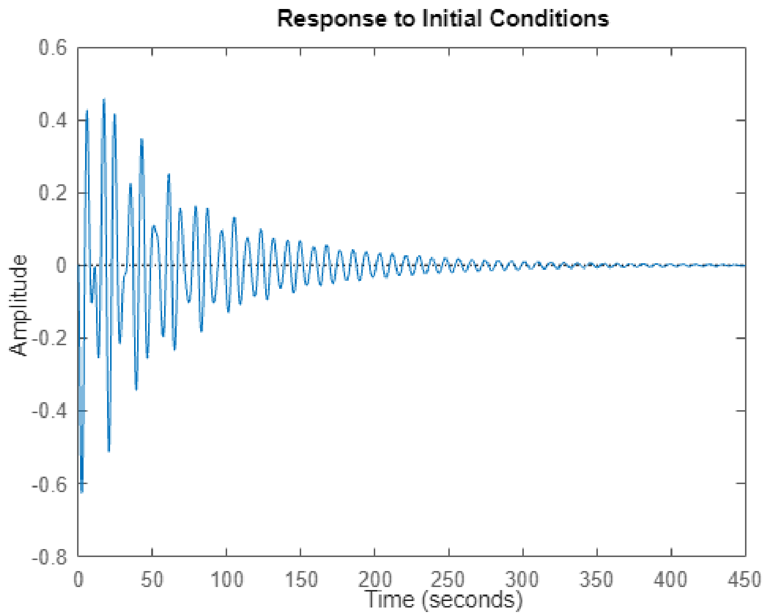


Figure 27: Response of x(t) graph for L3 observer

```
step(sys_3(1))
```

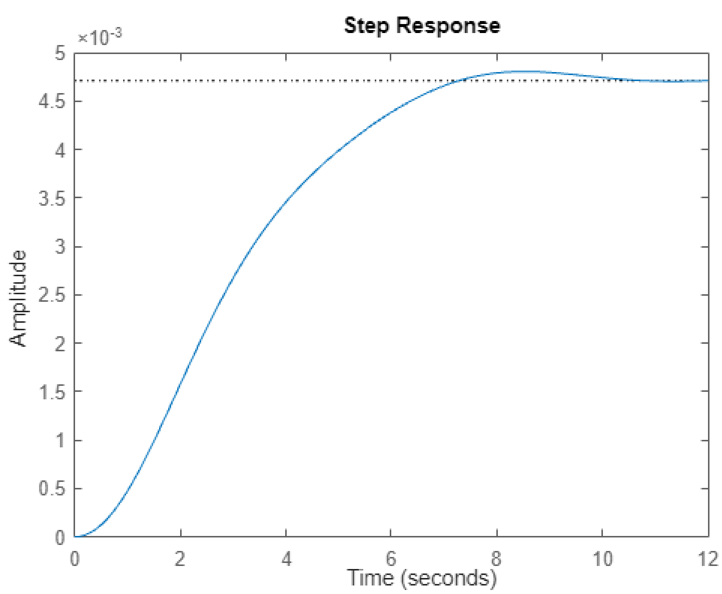


Figure 28: Step Response of x(t) graph for L3 observer

```
figure
initial(sys_3(2),x0)
```

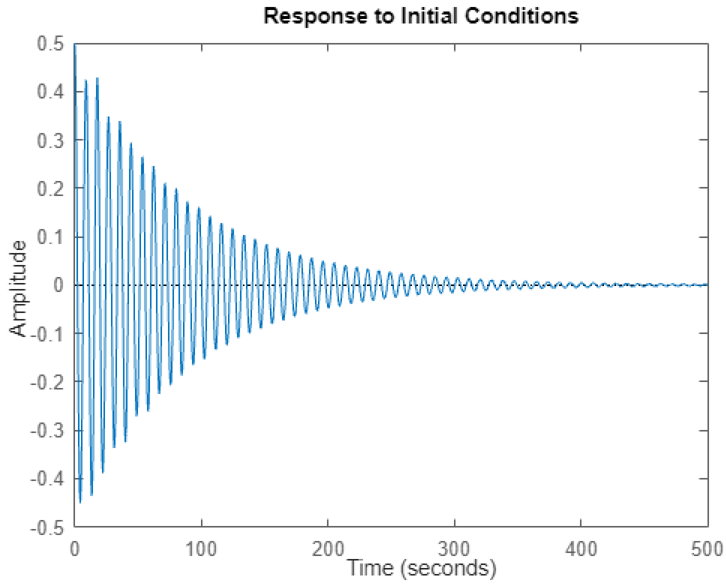


Figure 29: Response of $\theta_1(t)$ graph for L3 observer

```
step(sys_3(2))
```

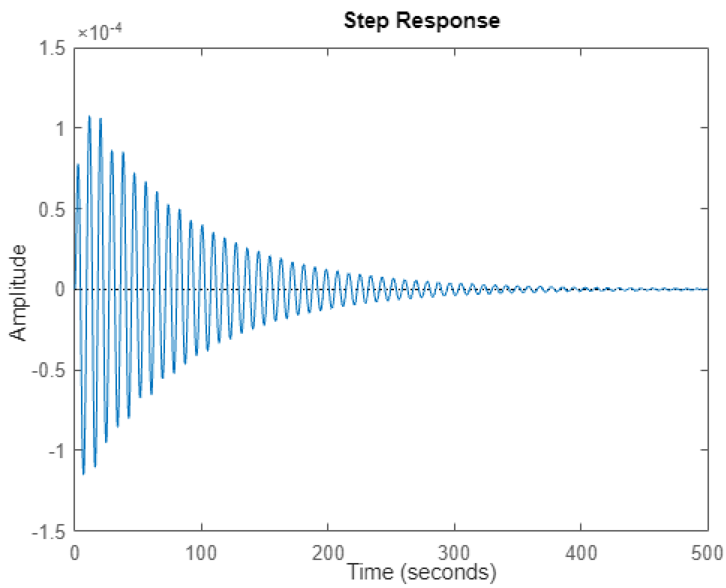


Figure 30: Step Response of $\theta_1(t)$ graph for L3 observer

```
figure
initial(sys_3(3),x0)
```

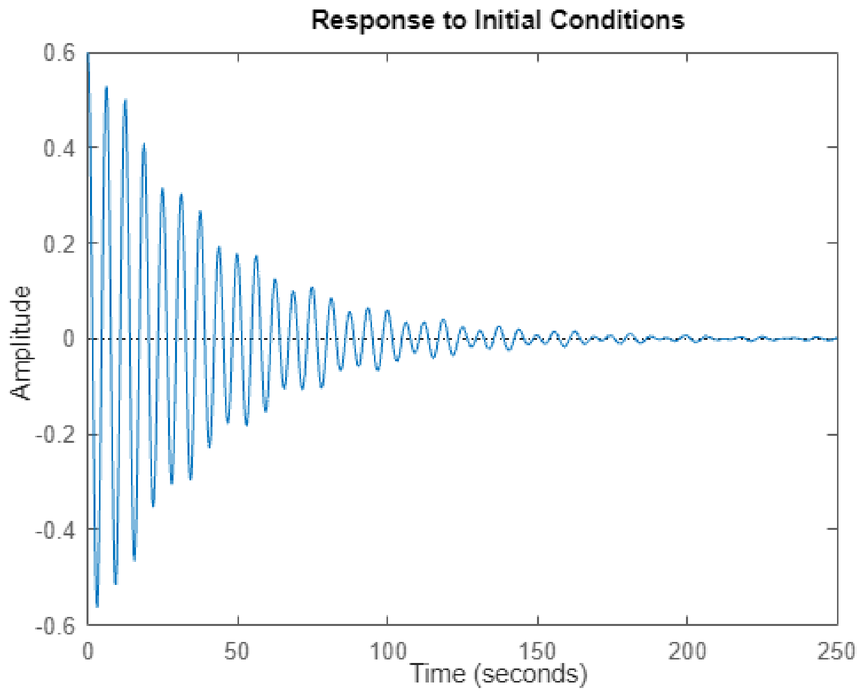


Figure 31: Response of $\theta_2(t)$ graph for L3 observer

```
step(sys_3(3))
```

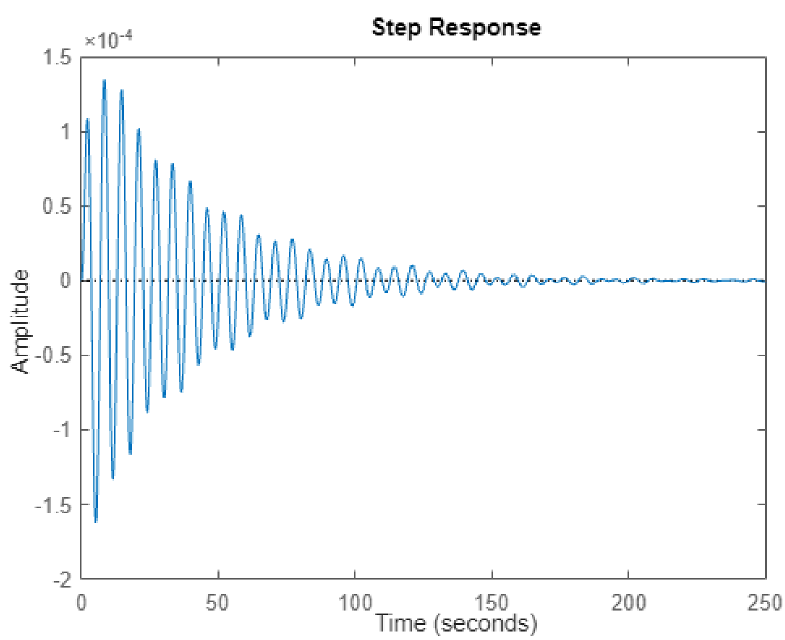


Figure 32: Step Response of $\boldsymbol{\theta_2(t)}$ graph for L3 observer

From the results we have seen that for the given inputs the LQG is better compared to Luenberger Observer(LQE) is reaching to 0 quicker and the system is filtered quickly.

9.2 LQG For Non-Linear System

Based on the results from the Linearized system we used the smallest output system that is sending x(t), to the LQG controller.

```
% Clear the workspace and close all figures
clc; clear; close all;
t_span = 0:0.01:100;
% Solve the ODE and obtain the state vector
[t, x] = ode45(@nonlinear_lqg, t_span, [10; 0; 10; 0; 20; 0;0;0;0;0;0;0]);


figure;
plot(t, x(:, 1));
hold on;
plot(t, x(:, 2));
plot(t, x(:, 3));
plot(t, x(:, 4));
plot(t, x(:, 5));
plot(t, x(:, 6));
xlabel('Time');
ylabel('States');
title('State change vs Time');
grid on;
hold off;
```

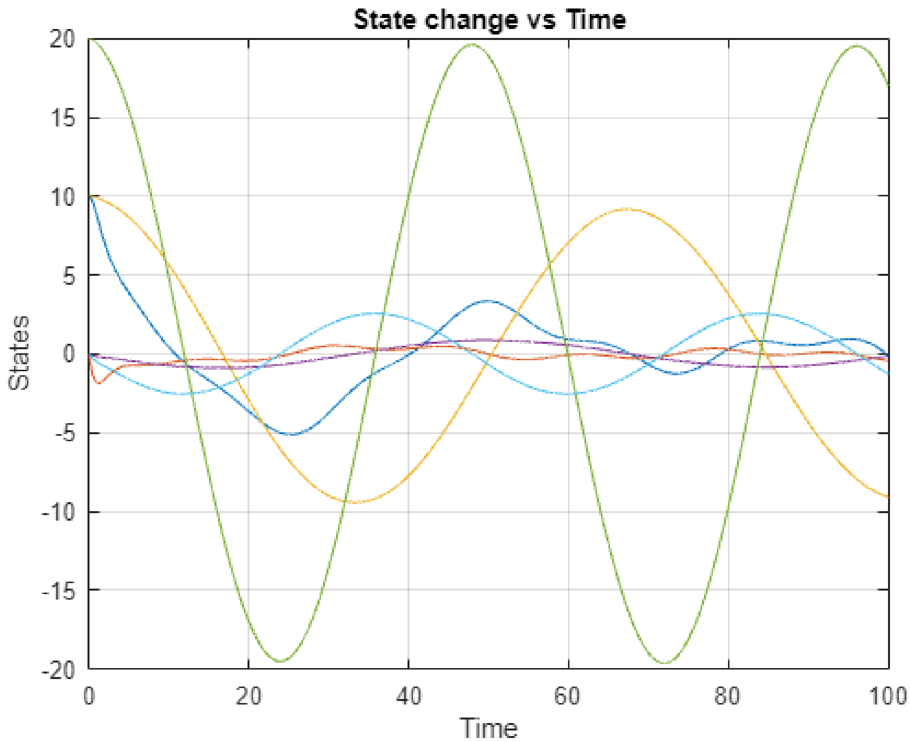


Figure 33: State Response vs time

The Graph represents the values after passing through the LQR. The estimated results plot is as shown in Figure 33.

```
figure;
plot(t, x(:, 7));
hold on;
plot(t, x(:, 8));
plot(t, x(:, 9));
plot(t, x(:, 10));
plot(t, x(:, 11));
plot(t, x(:, 12));
xlabel('Time');
ylabel('States');
title('Estimated State change vs Time');
grid on;
hold off;
```

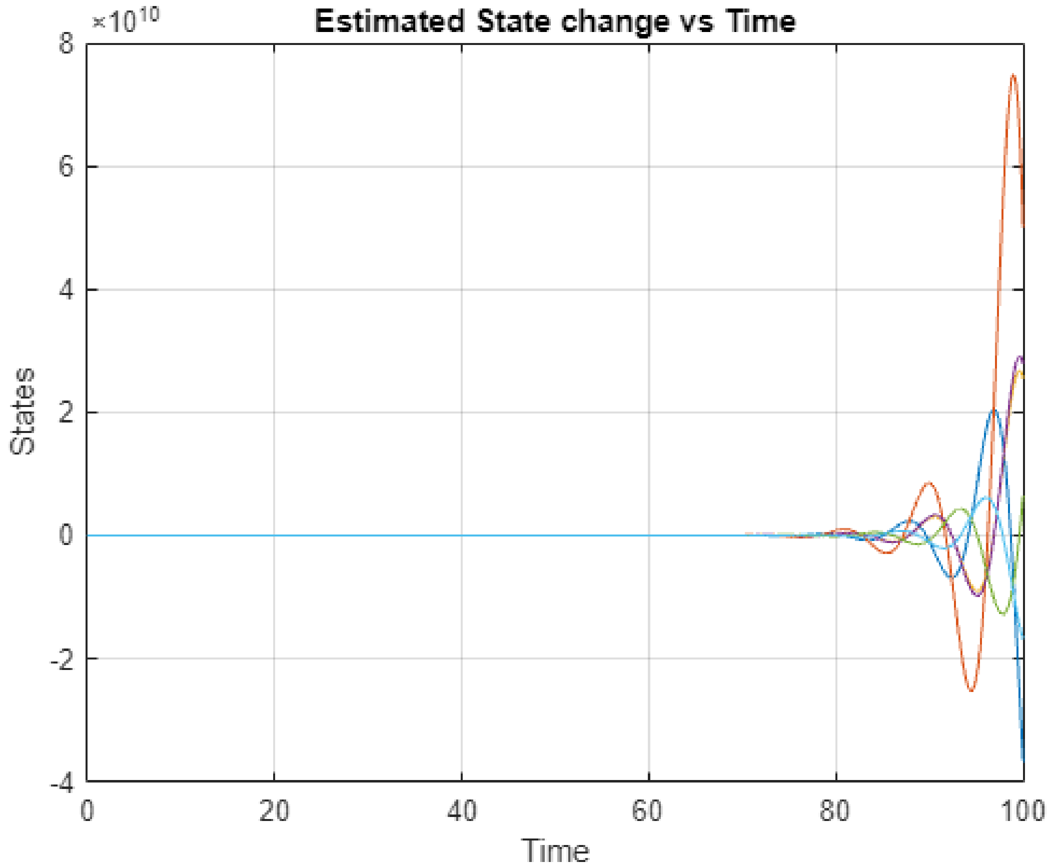


Figure 34: Estimated State Response vs time

The Graph represents the values after passing through the Kalman filter the estimated results plot is as shown in Figure 34.

```
function x_dot = nonlinear_lqg(t, x)
    x_dot = zeros(6,1);
    syms M m1 m2 l1 l2 g
% g = 9.81;
A = [0 1 0 0 0 0;
     0 0 (-g*m1)/M 0 (-g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 -(g*(M+m1))/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

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

```
    % C Matrix
    C_01 = [1 0 0 0 0 0];

    D = 0;

    %Defining the Mass and lengths
    M_val = 1000;
    m1_val = 100;
    m2_val = 100;
    l1_val = 20;
    l2_val = 10;
    g_val = 9.81;
    A = double(subs(A, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));
    B = double(subs(B, [M, m1, m2, l1, l2, g], [M_val, m1_val, m2_val, l1_val,
l2_val, g_val]));


    Q = [1000 0 0 0 0 0;  %x penalize
     0 1000 0 0 0 0;  %x_dot penalize
     0 0 1 0 0 0;   %theta1 penalize
     0 0 0 1 0 0;   %theta1_dot penalize
     0 0 0 0 1 0;   %theta2 penalize
     0 0 0 0 0 1];  %theta2_dot penalize
    % Q =eye(6)*100;
    R = 0.001;
    [K,~,~] = lqr(A, B, Q, R);
    F = -K*x(1:6);
    vd=0.2*eye(6); %process noise
    vn=1; %measurement noise

    k1 = lqr(A',C_01',vd,vn);
    sd = (A-k1'*C_01)*x(7:12);

    x_dot(1) = x(2);
    x_dot(2) = (-(m1_val*g_val*sind(x(3))*cosd(x(5))) -
(m2_val*g_val*sind(x(5))*cosd(x(5))) -(m1_val*l1_val*(x(4))*(x(4))*sind(x(3))) -
(m2_val*l2_val*(x(6))*(x(6))*sind(x(5))) +F)/(M_val +m1_val*(1-cosd(x(3))^2)
+m2_val*(1-cosd(x(5))^2));
    x_dot(3) = x(4);
    x_dot(4) = (x_dot(2)*cosd(x(3))-g_val*(sind(x(3))))/l1_val';
    x_dot(5) = x(6);
    x_dot(6) = (x_dot(2)*cosd(x(5))-g_val*(sind(x(5))))/l2_val';
    x_dot(7) = x(2) - x(10);
    x_dot(8) = x_dot(2) - sd(2);
```

```
    x_dot(9) = x(4) - x(11);
    x_dot(10) = x_dot(4) - sd(4);
    x_dot(11) = x(6) - x(12);
    x_dot(12) = x_dot(6) - sd(6);
end
```

Tracking a constant reference on x

To achieve asymptotic tracking of a constant reference on X the controller can be reconfigured to minimize a cost function that penalizes the deviation of the state X from the desired reference state X_d So the modified cost function is given by.

$$J = \int_0^\infty \left((x - x_d)^T . Q . (x - x_d) + (u - u_\infty)^T . R . (u - u_\infty)\right) dt$$

Where:

Q and R are positive definite weighting matrices

u is steady state control input

The modification involves adjusting the values of Q to prioritize the tracking of X and R to minimize control effort. This optimization aims to asymptotically drive $X \text{ to } X_d$

Rejecting constant force disturbances

Regarding constant force disturbances applied on the cart, the LQG controller, when designed to minimize the mentioned cost function, is inherently robust to disturbances. If the force disturbances are assumed to be Gaussian in nature, the Kalman filter component of the LQG controller will effectively estimate and compensate for these disturbances. This capability ensures that the controller can maintain stability and achieve accurate tracking even in the presence of constant force disturbances applied to the cart. The robustness of the LQG controller is a valuable attribute for handling uncertainties and disturbances in practical control system applications.

SIMULATION RESULTS VIDEO LINK: https://youtu.be/hGSncaZw_rQ