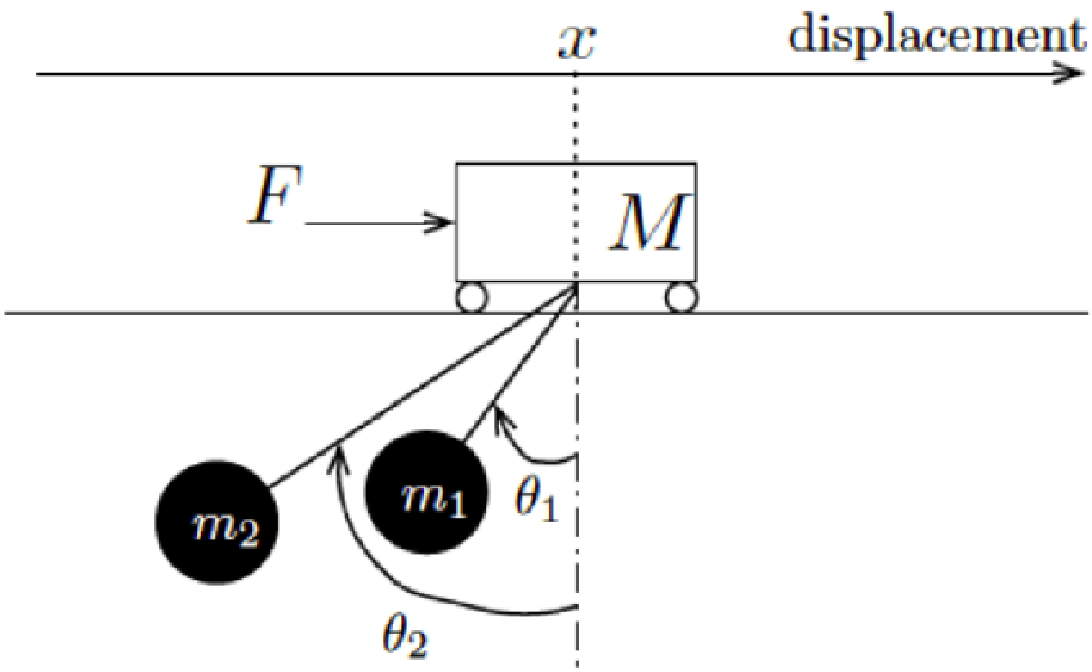
CONTROL OF ROBOT SYSTEMS - ENPM667

PROJECT-2

DESIGN OF A LQR AND LQG CONTROLLER FOR A SYSTEM WITH TWO PENDULUMS SUSPENDED FROM A CRANE

PROJECT BY:

1. **BADRINARAYANAN RAGHUNATHAN SRIKUMAR (119215418)**
2. **VYSHNAV ACHUTHAN POONDI VENKATARAMANI (119304815)**



The notations used in the system above are as follows:

- M: Mass of the cart
- F: Force applied to the cart
- m1: mass of the first pendulum
- m2: mass of the other pendulum
- l1 = length of cable used for pendulum 1
- l2 = length of cable used for pendulum 2

1. EQUATIONS OF MOTION:

The equations of motion for the crane system, will be modeled using Euler-Lagrangian method. We need to find the Kinetic and Potential energy of the system.

1.1 KINETIC ENERGY CALCULATION:

The system's Kinetic energy can be divided into 3 parts namely,

1. The kinetic energy of the cart of mass M.
2. The kinetic energy of pendulum 1.
3. The kinetic energy of pendulum 2.

1.1.1 KINETIC ENERGY OF CART:

The kinetic energy of the cart can be written as

$$\frac{1}{2}MV^2$$

Where V can be written as the sum of squares of components in X and Y directions.

$$V_x = \dot{x}^2$$

$$V_y = 0$$

Hence the Kinetic energy of the cart can be written as:

$$\frac{1}{2}M\dot{x}^2$$

1.1.2 KINETIC ENERGY OF PENDULUM 1:

The kinetic energy of pendulum 1 can be written as

$$\frac{1}{2}m_1 v_1^{\ 2}$$

Where v1 can be written a sum of squares of components in X and Y directions.

$$v_{1x} = \dot{x} - l_1 \cos\theta_1 \dot{\theta}_1$$

$$v_{1y} = -l_1 \sin\theta_1 \dot{\theta}_1$$

Hence the kinetic energy of pendulum 1 can be written as:

$$\frac{1\, m_1 \left(\dot{x}^2 + l_1^{\,2}\cos^2\theta_1\dot{\theta}_1^{\,2} - 2l_1\cos\theta_1\dot{\theta}_1\dot{x}\right)}{2} + \frac{1m_1}{2}\left(l_1^{\,2}\sin^2\theta_1\dot{\theta}_1^{\,2}\right)$$

1.1.3 KINETIC ENERGY OF PENDULUM 2:

The kinetic energy of pendulum 1 can be written as:

$$\frac{1}{2}m_2v_2^{\,2}$$

Where v2 can be written a sum of squares of components in X and Y directions.

$$v_{2x} = \dot{x} - l_2\cos\theta_2\dot{\theta}_2$$

$$v_{2y} = -l_2\sin\theta_2\dot{\theta}_2$$

Hence the kinetic energy of pendulum 1 can be written as:

$$\frac{1\, m_2 \left(\dot{x}^2 + l_2^{\,2}\cos^2\theta_2\dot{\theta}_2^{\,2} - 2l_2\cos\theta_2\dot{\theta}_2\dot{x}\right)}{2} + \frac{1m_2}{2}\left(l_2^{\,2}\sin^2\theta_2\dot{\theta}_2^{\,2}\right)$$

Finally, the Kinetic energy of the whole system can be written as:

$$\frac{1\, m_1 \left(\dot{x}^2 + l_1^{\,2}\cos^2\theta_1\dot{\theta}_1^{\,2} - 2l_1\cos\theta_1\dot{\theta}_1\dot{x}\right)}{2} + \frac{1m_1}{2}\left(l_1^{\,2}\sin^2\theta_1\dot{\theta}_1^{\,2}\right) + \frac{1}{2}M\dot{x}^2$$

$$+ \frac{1\, m_2 \left(\dot{x}^2 + l_2^{\,2}\cos^2\theta_2\dot{\theta}_2^{\,2} - 2l_2\cos\theta_2\dot{\theta}_2\dot{x}\right)}{2} + \frac{1m_2}{2}\left(l_2^{\,2}\sin^2\theta_2\dot{\theta}_2^{\,2}\right)$$

Simplifying it,

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

1.2 POTENTIAL ENERGY CALCULATION:

Considering the ground to be the reference for potential energy, the potential energy for the cart would be zero. The potential energy will only be contributed by the pendulums. Hence the potential energy of the system would be:

$$P = -m_1 g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2$$

1.3 LAGRANGIAN CALCULATION:

Lagrangian of any system can be calculated as:

$$L = K - P$$

Thus, the Lagrangian can be written as:

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

Now, we calculate the derivative of Lagrangian with respect to $\dot{x}, \dot{\theta}_1, \dot{\theta}_2$

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

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

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} = 0$$

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

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

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

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

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

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

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = m_1 l_1^2 \ddot{\theta}_1 - m_1 \ddot{x} l_1 \cos\theta_1 + m_1 l_1 \dot{x}\dot{\theta}_1 \sin\theta_1 - m_1 l_1 \dot{x}\dot{\theta}_1 \sin\theta_1 + m_1 l_1 g\sin\theta_1 = 0$$

$$\ddot{\theta}_1 = \frac{\ddot{x}\cos\theta_1}{l_1} - \frac{g\sin\theta_1}{l_1}$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \dot{\theta}_2 - m_2 l_2 \dot{x}\cos\theta_2$$

$$\frac{\partial L}{\partial \theta_2} = m_2 l_2 \dot{\theta}_2 \dot{x}\sin\theta_2 - m_2 l_2 g\sin\theta_2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} = m_2 l_2^2 \ddot{\theta}_2 - m_2 \ddot{x} l_2 \cos\theta_2 + m_2 l_2 \dot{x}\dot{\theta}_2 \sin\theta_2 - m_2 l_2 \dot{x}\dot{\theta}_2 \sin\theta_2 + m_2 l_2 g\sin\theta_2 = 0$$

$$\ddot{\theta}_2 = \frac{\ddot{x}\cos\theta_2}{l_2} - \frac{g\sin\theta_2}{l_2}$$

Thus we arrive at $\ddot{x}, \ddot{\theta}_1, \ddot{\theta}_2$ as:

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

$$\ddot{\theta}_1 = \frac{\ddot{x}\cos\theta_1}{l_1} - \frac{g\sin\theta_1}{l_1}$$

$$\ddot{\theta}_2 = \frac{\ddot{x}\cos\theta_2}{l_2} - \frac{g\sin\theta_2}{l_2}$$

The state variables for the system can be written as:

$$\begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix}$$

We can write it in a non-linear state space form as follows:

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

B) **LINEARIZATION**

The approximation of a nonlinear system by a linear system is the source of linear system models. These approximations are intended for investigating a system's local behavior when nonlinear effects are considered to be minor. Linearization can be done in two methods namely:

- Jacobian Linearization around Equilibrium Point
- Feedback Linearization

For the inverted pendulum problem, we are taking the approach of changing the nonlinear system to a linear system using the Jacobian Linearization.

To perform linearization, we take around the equilibrium point $x = 0, \theta_1 = 0 \ and \ \theta_2 = 0$. Now we approximate the smaller angles to zero so that the equilibrium is limited. Some of the limiting conditions include $\sin\theta_1 \approx \theta_1$, $\sin\theta_2 \approx \theta_2$, $\cos\theta_1 \approx 1$, $\cos\theta_2 \approx 1$ and $\dot{\theta}_1 = \dot{\theta}_2 \approx 0$. Using these approximations, the following changes to the equations are made

$$\ddot{x} = \frac{1}{M}\left(-m_1 g\theta_1 - m_2 g\theta_2 + F\right) \quad \text{— (1)}$$

$$\ddot{\theta}_1 = \frac{1}{Ml_1}\left(-m_1 g\theta_1 - m_2 g\theta_2 - Mg\theta_1 + F\right) \quad \text{— (2)}$$

$$\ddot{\theta}_2 = \frac{1}{Ml_2}\left(-m_1 g\theta_1 - m_2 g\theta_2 - Mg\theta_2 + F\right) \quad \text{— (3)}$$

And the rest of the variables in the nonlinear equations remain the same. Now according to the Jacobian linearization method. The Jacobian linear equations of a nonlinear system is given as

$$\frac{\mathrm{d}z}{\mathrm{d}t} = Az + Bv \ , \quad w = Cz + Dv$$

Where the jacobian of the solution can be given as:

$$A = \begin{bmatrix} \frac{\partial g_1}{\partial X_1} & \frac{\partial g_1}{\partial X_2} & \frac{\partial g_1}{\partial X_3} & \frac{\partial g_1}{\partial X_4} & \frac{\partial g_1}{\partial X_5} & \frac{\partial g_1}{\partial X_6} \\ \frac{\partial g_2}{\partial X_1} & \frac{\partial g_2}{\partial X_2} & \frac{\partial g_2}{\partial X_3} & \frac{\partial g_2}{\partial X_4} & \frac{\partial g_2}{\partial X_5} & \frac{\partial g_2}{\partial X_6} \\ \frac{\partial g_3}{\partial X_1} & \frac{\partial g_3}{\partial X_2} & \frac{\partial g_3}{\partial X_3} & \frac{\partial g_3}{\partial X_4} & \frac{\partial g_3}{\partial X_5} & \frac{\partial g_3}{\partial X_6} \\ \frac{\partial g_4}{\partial X_1} & \frac{\partial g_4}{\partial X_2} & \frac{\partial g_4}{\partial X_3} & \frac{\partial g_4}{\partial X_4} & \frac{\partial g_4}{\partial X_5} & \frac{\partial g_4}{\partial X_6} \\ \frac{\partial g_5}{\partial X_1} & \frac{\partial g_5}{\partial X_2} & \frac{\partial g_5}{\partial X_3} & \frac{\partial g_5}{\partial X_4} & \frac{\partial g_5}{\partial X_5} & \frac{\partial g_5}{\partial X_6} \\ \frac{\partial g_6}{\partial X_1} & \frac{\partial g_6}{\partial X_2} & \frac{\partial g_6}{\partial X_3} & \frac{\partial g_6}{\partial X_4} & \frac{\partial g_6}{\partial X_5} & \frac{\partial g_6}{\partial X_6} \end{bmatrix}$$

Where $g_1, g_2 \cdots g_6$ are dynamics of the functions of state variables and X are the state variables.

Substituting the values in the A matrix equation, and writing it in the form of $\dot{x} = Ax + BU$, we end up with

$$\begin{bmatrix} \frac{d\dot{x}}{dx} & \frac{d\dot{x}}{d\dot{x}} & \frac{d\dot{x}}{d\theta_1} & \frac{d\dot{x}}{d\dot{\theta}_1} & \frac{d\dot{x}}{d\theta_2} & \frac{d\dot{x}}{d\dot{\theta}_2} \\ \frac{d\ddot{x}}{dx} & \frac{d\ddot{x}}{d\dot{x}} & \frac{d\ddot{x}}{d\theta_1} & \frac{d\ddot{x}}{d\dot{\theta}_1} & \frac{d\ddot{x}}{d\theta_2} & \frac{d\ddot{x}}{d\dot{\theta}_2} \\ \frac{d\ddot{\theta}_1}{dx} & \frac{d\ddot{\theta}_1}{d\dot{x}} & \frac{d\ddot{\theta}_1}{d\theta_1} & \frac{d\ddot{\theta}_1}{d\dot{\theta}_1} & \frac{d\ddot{\theta}_1}{d\theta_2} & \frac{d\ddot{\theta}_1}{d\dot{\theta}_2} \\ \frac{d\dot{\theta}_1}{dx} & \frac{d\dot{\theta}_1}{d\dot{x}} & \frac{d\dot{\theta}_1}{d\theta_1} & \frac{d\dot{\theta}_1}{d\dot{\theta}_1} & \frac{d\dot{\theta}_1}{d\theta_2} & \frac{d\dot{\theta}_1}{d\dot{\theta}_2} \\ \frac{d\ddot{\theta}_2}{dx} & \frac{d\ddot{\theta}_2}{d\dot{x}} & \frac{d\ddot{\theta}_2}{d\theta_1} & \frac{d\ddot{\theta}_2}{d\dot{\theta}_1} & \frac{d\ddot{\theta}_2}{d\theta_2} & \frac{d\ddot{\theta}_2}{d\dot{\theta}_2} \\ \frac{d\dot{\theta}_2}{dx} & \frac{d\dot{\theta}_2}{d\dot{x}} & \frac{d\dot{\theta}_2}{d\theta_1} & \frac{d\dot{\theta}_2}{d\dot{\theta}_1} & \frac{d\dot{\theta}_2}{d\theta_2} & \frac{d\dot{\theta}_2}{d\dot{\theta}_2} \end{bmatrix} \begin{bmatrix} x \\ x \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + B.F$$

Upon substituting the values we got after simplifying the variables, we get

$$\Rightarrow \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{gm_1}{M} & 0 & -\frac{gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-g\left(M+m_1\right)}{Ml_1} & 0 & -\frac{gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{gm_1}{Ml_2} & 0 & \frac{-g\left(M+m_2\right)}{Ml_2} & 0 \end{bmatrix} \begin{bmatrix} x \\ x \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + B.F$$

Now, from equations 1, 2, and 3 we get B as a matrix in terms of F thus, our final linear equation is given as

$$\Rightarrow \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{gm_1}{M} & 0 & -\frac{gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-g\left(M+m_1\right)}{Ml_1} & 0 & -\frac{gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{gm_1}{Ml_2} & 0 & \frac{-g\left(M+m_2\right)}{Ml_2} & 0 \end{bmatrix} \begin{bmatrix} x \\ x \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} F$$

C) CONTROLLABILITY

In state space theory, A system is said to be controllable if the $n \times n$ matrix $w\left(t_0, t_f\right)$ (the grammian of the system) is said to be invertible, ie, the determinant of the controllability matrix is not zero or non-singular. Controllability of the matrix can also be verified, if the rank of matrix C of dimension n x nm is equal to n, then the system is said to be controllable.

$W_c\left(0, t_f\right) = \int_0^t e^{-A\gamma} B_K B_K{}^T e^{-A^T\gamma} \mathrm{d}\gamma$ is invertible if the determinant of the Controllability matrix is not zero.

$$|C| = \left[B \; AB \; A^2B \; A^3B \; A^4B \; A^5B\right] \neq 0$$

The controllability matrix obtained via MATLAB simulation is a 6 x 6 matrix with n =6, thus the rank of matrix C is 6, and hence according to the second theorem, our system is controllable.

Now to verify the first hypothesis, we take the determinant of the matrix and check whether the matrix is singular or non-singular

The Controllability matrix C is given as

```
[        0,      1/M,                                           0,            - (g*m1)/(L1*M^2) - (g*m2)/(L2*M^2),
[     1/M,         0,              - (g*m1)/(L1*M^2) - (g*m2)/(L2*M^2),                                           0,
[        0, 1/(L1*M),                                           0, - (M*g + g*m1)/(L1^2*M^2) - (g*m2)/(L1*L2*M^2),
[1/(L1*M),         0, - (M*g + g*m1)/(L1^2*M^2) - (g*m2)/(L1*L2*M^2),                                           0,
[        0, 1/(L2*M),                                           0, - (M*g + g*m2)/(L2^2*M^2) - (g*m1)/(L1*L2*M^2),
[1/(L2*M),         0, - (M*g + g*m2)/(L2^2*M^2) - (g*m1)/(L1*L2*M^2),                                           0,

>>
>>
```

```
                                                                                                                   0,
           ((g*m1*(M*g + g*m1))/(L1*M^2) + (g^2*m1*m2)/(L2*M^2))/(L1*M) + ((g*m2*(M*g + g*m2))/(L2*M^2) + (g^2*m1*m2)/(L1*M^2))/(L2*M),
                                                                                                                   0,
 ((g*m2*(M*g + g*m1))/(L1^2*M^2) + (g*m2*(M*g + g*m2))/(L1*L2*M^2))/(L2*M) + ((M*g + g*m1)^2/(L1^2*M^2) + (g^2*m1*m2)/(L1*L2*M^2))/(L1*M),
                                                                                                                   0,
 ((g*m1*(M*g + g*m2))/(L2^2*M^2) + (g*m1*(M*g + g*m1))/(L1*L2*M^2))/(L1*M) + ((M*g + g*m2)^2/(L2^2*M^2) + (g^2*m1*m2)/(L1*L2*M^2))/(L2*M),

           ((g*m1*(M*g + g*m1))/(L1*M^2) + (g^2*m1*m2)/(L2*M^2))/(L1*M) + ((g*m2*(M*g + g*m2))/(L2*M^2) + (g^2*m1*m2)/(L1*M^2))/(L2*M)]
                                                                                                                   0]
 ((g*m2*(M*g + g*m1))/(L1^2*M^2) + (g*m2*(M*g + g*m2))/(L1*L2*M^2))/(L2*M) + ((M*g + g*m1)^2/(L1^2*M^2) + (g^2*m1*m2)/(L1*L2*M^2))/(L1*M)]
                                                                                                                   0]
 ((g*m1*(M*g + g*m2))/(L2^2*M^2) + (g*m1*(M*g + g*m1))/(L1*L2*M^2))/(L1*M) + ((M*g + g*m2)^2/(L2^2*M^2) + (g^2*m1*m2)/(L1*L2*M^2))/(L2*M)]
                                                                                                                   0]
```

The matrix representation is tallied as the first 4 columns, the fifth and the sixth column respectively. Now taking the determinant of the matrix we get,

$$|C| = -\frac{\left(L_1{}^2 \cdot g^6 - 2L_1 \cdot L_2 \cdot g^6 + L_2\right)}{M^6 \cdot L_1{}^6 \cdot L_2{}^6}$$

The following is verified in MATLAB

```
val =

-(L1^2*g^6 - 2*L1*L2*g^6 + L2^2*g^6)/(L1^6*L2^6*M^6)
```

From the above determinant value we can see that the system is controllable as the controllability matrix is invertible.

D) LQR CONTROLLER

Given initial conditions to apply for the system are as follows, $M = 1000Kg,\ m_1 = m_2 = 100Kg,\ l_1 = 20m\ and\ l_2 = 10m.$ Having the following initial conditions, we need to find the vector of state feedback controls given by K with the 6 state variables. LQR controller is used to drive the desired state of a system to zero. Before we design an LQR controller, we check for the controllability of the system using the rank condition. Given, A matrix from a linearized form of the state space equation,

$$\Rightarrow A=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{9.81\times 100}{1000} & 0 & -\frac{9.81\times 100}{1000} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-9.81(1000+20)}{1000\times 20} & 0 & -\frac{9.81\times 100}{1000\times 20} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{9.81\times 100}{1000\times 10} & 0 & \frac{-9.81(1000+20)}{1000\times 10} & 0 \end{bmatrix}$$

$$\Rightarrow B=\begin{bmatrix} 0 \\ \frac{1}{1000} \\ 0 \\ \frac{1}{1000\times 20} \\ 0 \\ \frac{1}{1000\times 10} \end{bmatrix}$$

Using these, we find whether our system is controllable by checking whether the rank of our controllability matrix C of dimension n x nm is equal to n.

$$rank[C]=rank\left[B\ AB\ A^2B\ A^3B\ A^4B\ A^5B\right]=n$$

The controllability matrix is given as :

$$C=\begin{bmatrix} 0 & 10^{-3} & 0 & -0.000147 & 0 & 0.00014166 \\ 10^{-3} & 0 & -0.000147 & 0 & 0.00014166 & 0 \\ 0 & 0.00005 & 0 & -0.00003185 & 0 & 0.000022689 \\ 0.00005 & 0 & -0.00003185 & 0 & 0.000022689 & 0 \\ 0 & 10^{-4} & 0 & -0.0001127 & 0 & 0.00012461 \\ 10^{-4} & 0 & -0.0001127 & 0 & 0.00012461 & 0 \end{bmatrix}$$

From the matrix, we can see that rank(C) is 6, which is equal to n, which is also 6. Hence, the system under the given conditions is said to be controllable.

To check whether the system is stable in our given conditions, we check for the Eigenvalues of the system. The Eigenvalues of a given matrix is determined by using the formula

$$=\left|A - I\lambda\right|$$

$$\Rightarrow \left|\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.981 & 0 & -0.981 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -0.5390 & 0 & -0.049 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -0.0981 & 0 & -1.078 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda \end{bmatrix}\right|$$

Upon solving this using the MATLAB code, we get the following Eigenvalues for our given system

$$\Rightarrow \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix} = \begin{bmatrix} 0+0i \\ 0+0i \\ 0+0.728i \\ 0-0.728i \\ 0+1.0425i \\ 0-1.0425i \end{bmatrix}$$

LQR Controller:

In this section, we will design a Linear Quadratic Regulator for our system.Instead of choosing the closed loop eigenvalue positions to achieve a specific goal, the gains for a state feedback controller might be determined by attempting to maximize a cost function in LQR controllers. In a closed system, LQR uses the feedback obtained from the system and generates a feedback gain matrix K.

$\dot{X} = AX + BU \ where, U = -KX$

Because we integrate the cost function from zero to infinity, the LQR controller is also known as an Infinite Horizon problem. We chose Infinity because we want the system to perform precisely throughout all time instants rather than just a single span of time. By modifying the weights of the states in the Q matrix, we hope to minimize the cost function shown below.

$$J = \int_0^{\infty} (\vec{X}(t))^T Q \vec{X}(t) + (\vec{U}(t))^T R \vec{U}(t) \, \mathrm{d}t$$

Where, $Q_x \geq 0$ are symmetric, positive definite matrices, which are of the appropriate dimensions. The dimensions of the matrix are given as follows:

$\vec{X}(t) = n \times 1 \ , \ Q = n \times n \ , \vec{U}(t) = p \times 1, \ R = p \times p$

The solution for the LQR controller is given via a linear law, which is given as

$u = -Q_u^{-1} BT P_x, \ P \in \mathbb{R}$

Using the above linear law, we can derive the following equation which the symmetric matrix satisfies

$A^T P + PA - PBR^{-1} B^T P + Q = 0$

$K = R^{-1} B^T P \ \ and \ U = -KX$

For our system, it is mandatory that the Q matrix and R values are different to control the system in an optimum way and from the graphs we obtain, we can fine-tune our Q and R values which gives us higher system stability. Upon our initial assumption, we are taking Q and R as

```
% LQR weights
Q = [10 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 2500 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 2500 0;
     0 0 0 0 0 0];
R = 0.001;
```

We use MATLAB code to formulate our LQR controller value. The function takes in the following values as input, Q matrix, R rate, A matrix, and B matrix which are system dynamics. The function calculates the optimal gain matrix K for the given system. The function returns K, the solution of the Riccati equation, and the poles for the closed loop. The closed loop poles are given as the EigenValue of $(A) - B \times K$.

As per the code, the updated EigenValues are given as follows

eigen_A_LQR

6x1 complex double

	1
1	-0.2001 + 0.2050i
2	-0.2001 - 0.2050i
3	-0.0807 + 1.0456i
4	-0.0807 - 1.0456i
5	-0.0425 + 0.7281i
6	-0.0425 - 0.7281i
7	

Stability of the system can be identified by using Lyapunov's Indirect Method, which states that, if the Eigenvalues of the system lie on the left half of the s-plane, then the system is said to be controllable. As per the values we obtained above, we can see that the eigenvalues are on the left half of the plane, which denotes that our system is stable.

To check our stability for the linear and non-linear system, we take our initial displacement to be zero, the angle for mass $m_1\ as\ \theta_1 = 10^\circ$ and for mass $m_2\ as\ \theta_2 = 27^\circ$. Upon using these values as initial conditions, the following behavior of systems are visualized graphically :

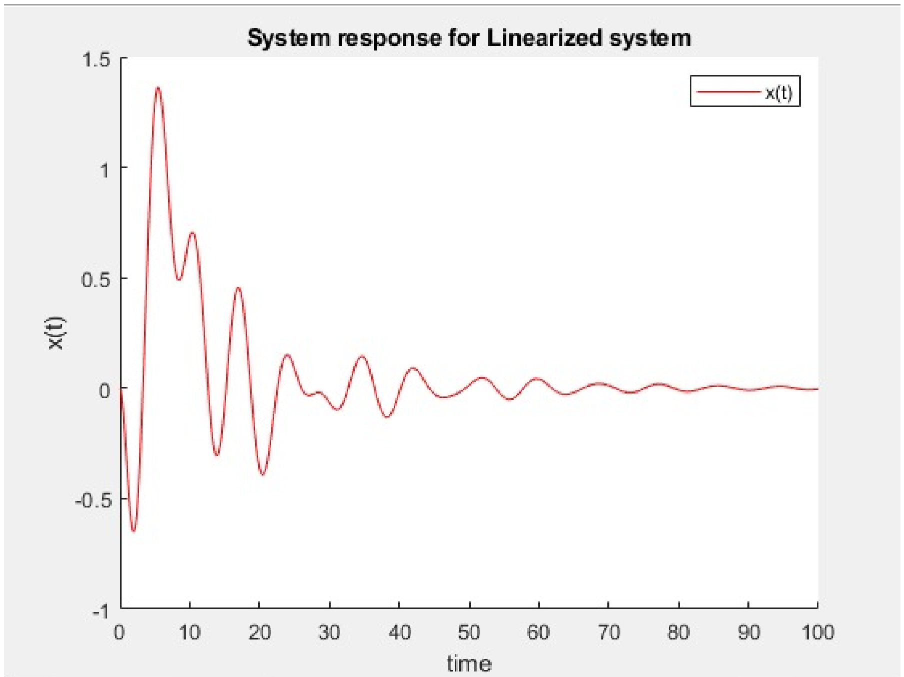


Fig: System Response for Linearized System X(T)

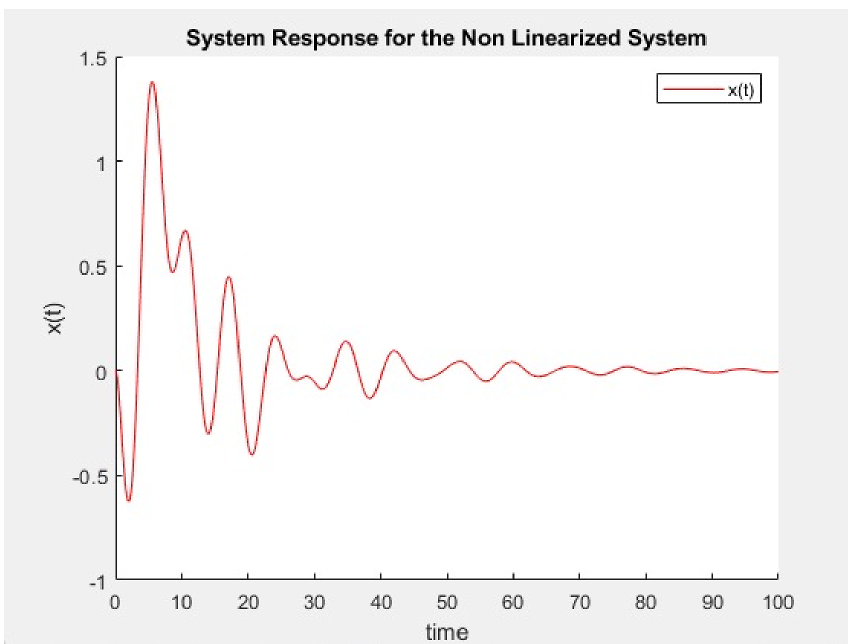


Fig: System Response for Non-Linearized System X(T)

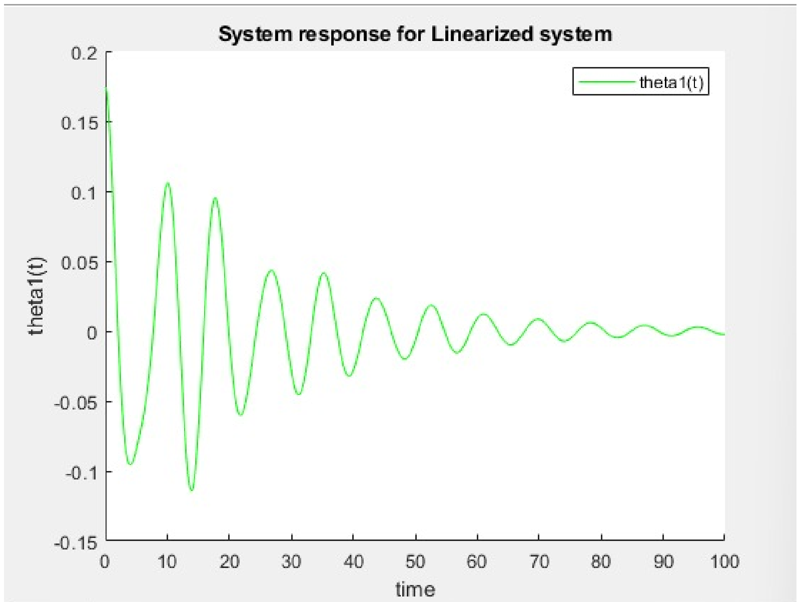


Fig: System Response for Linearized System for the angle of mass 1 After applying LQR

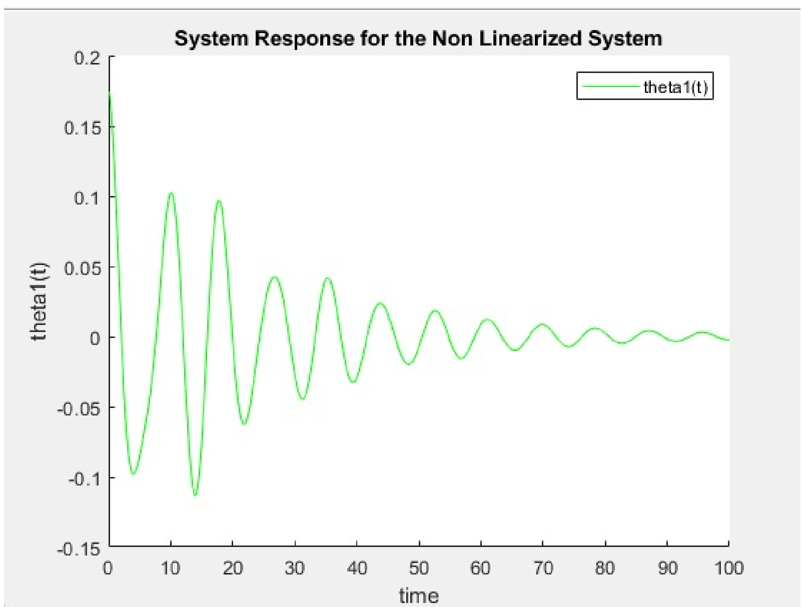


Fig: System Response for Non-Linearized System for the angle of mass 1 after applying LQR

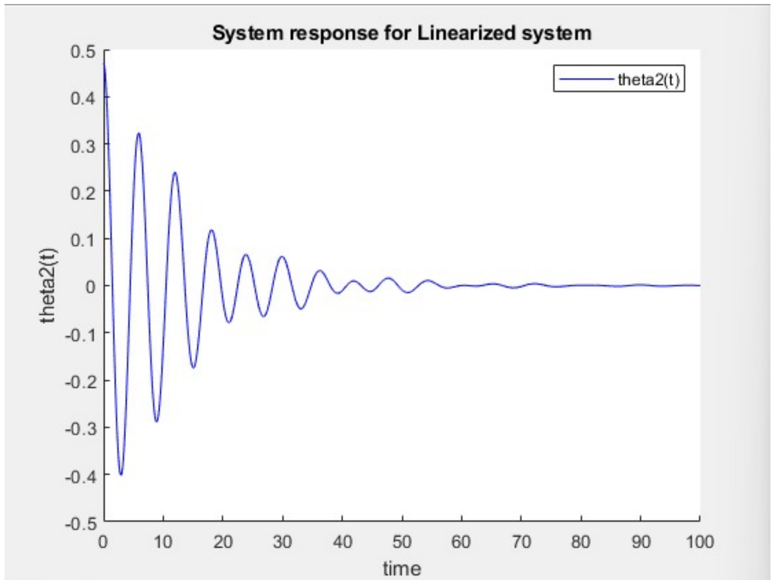


Fig: System Response for Linearized System for the angle of mass 2 after applying LQR

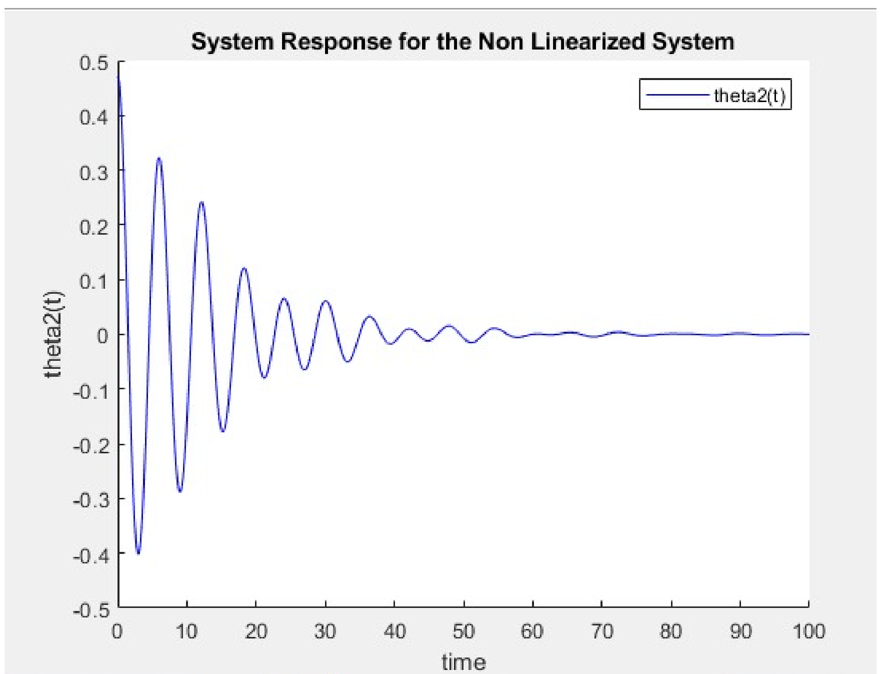


Fig: System Response for Non-Linearized System for the angle of mass 2 after applying LQR

E. OBSERVABILITY

Given, to check the observability of four output vectors, which are listed as below:

a) $x(t)$
b) $\theta_1(t), \theta_2(t)$
c) $x(t), \theta_2(t)$
d) $x(t), \theta_1(t), \theta_2(t)$

In order to check the observability, we will check the rank of the matrix of each vector and for the system to be observable, the rank of the vectors must be 6.

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix} \text{where } C \text{ is } Y = CX(t) + DU(t)$$

Given state variable

$$\vec{X}(t) = \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix}$$

- We now find the observability of the matrix for $x(t)$ to be as

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

For this value of C rank of the observability matrix is 6, hence it is observable

- The observability matrix for the vector $\theta_1(t), \theta_2(t)$ is given as

$$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Rank of C_2 is 4 [From MATLAB]. Hence the vector is not observable

- The observability matrix for the vector $x(t), \theta_2(t)$ is given as

$$C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For this value of C rank of observability matrix is 6, hence it is observable

- The observability matrix for the vector $x(t), \theta_1(t), \theta_2(t)$ is given as

$$C_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For this value of C rank of observability matrix is 6, hence it is observable

F. Luenberger Observer Design

For any given system, we can design an observer system which takes in the input and output of the system, which is assumed to be measurable and generates an estimate of the state of the system. Using the Luenberger Observer design, we can formulate a state observer system, which gives us an estimate of the system's state, using the gain matrix L. For a linear state space system represented as

$$\dot{X}(t) = Ax(t) + Bu(t)$$
$$Y(t) = Cx(t) + Du(t)$$

The Luenberger Observer state space representation is given as:

$$\dot{\widehat{X}}(t) = A\widehat{x}(t) + B_k U_k(t) + L(Y(t) - C\widehat{x}(t))$$

Where L is the gain matrix of the given system, $\widehat{X}(t)$ is the state estimator, and $Y(t) - C\widehat{x}(t)$ is the error correction factor. The error between the input and the observed state can be mathematically defined as

$$e = x - \widehat{x}$$

on differentiating this, we get

$$\dot{e} = \dot{x} - \dot{\widehat{x}}$$

From the first equation, we can get $\dot{x}$ and $\dot{\hat{x}}$

$$e = Ax(t) + B_k U_k(t) - A\hat{x}(t) - B_k U_k(t) - L(Y(t) - C\hat{x}(t))$$

Upon considering D=0 and $Y(t) = Cx(t)$, canceling terms and grouping like terms we get,

$$\dot{e} = (A - LC)\, e\,(t)$$

To get the gain matrix, we use the Kalman-Bucy filter, which takes input as A, C which are dynamic state input and output matrices, measurement noise and process noise, which gives us a gain matrix estimate. Upon using the LQE function in MATLAB function, we get a set of L gain matrices, which we feed to the Luenberger Observer to obtain the following behavior of the system. To generate our graphs, we take initial conditions for $m_1\ as\ \theta_1 = 10°$ and $m_2\ as\ \theta_2 = 27°$.

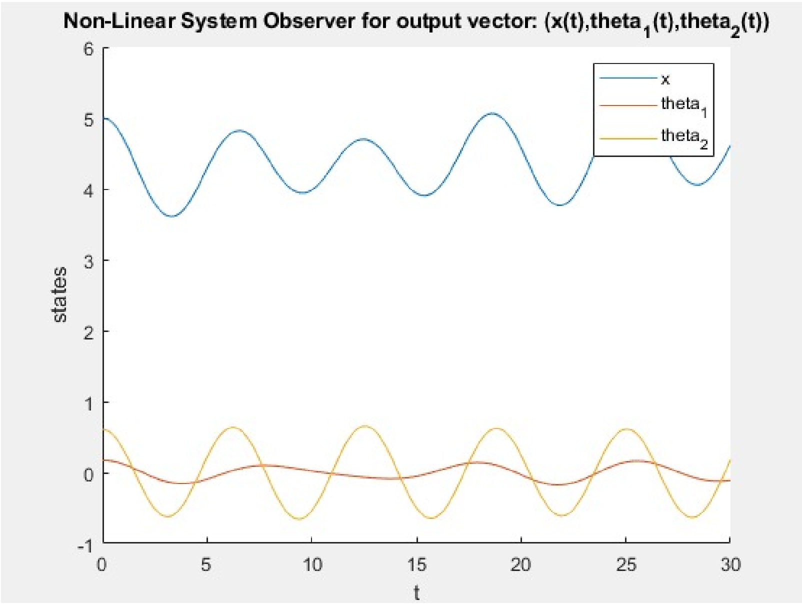


fig: Nonlinear system Observer for vector $x(t), \theta_1(t), \theta_2(t)$

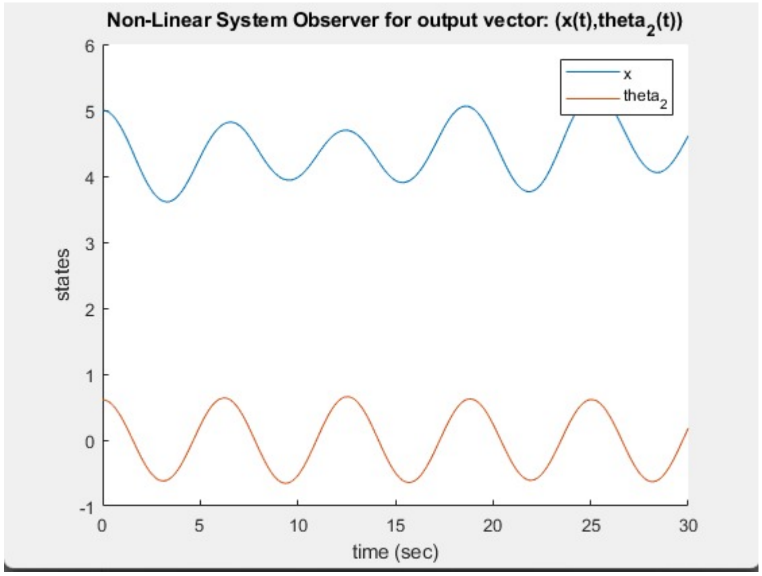


fig: Nonlinear system Observer for vector $x(t), \theta_2(t)$

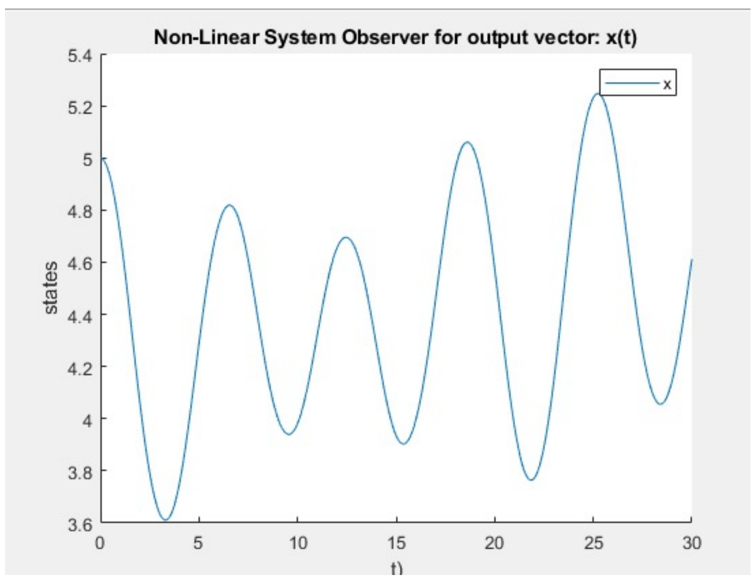


fig: Nonlinear system Observer for vector $x(t)$

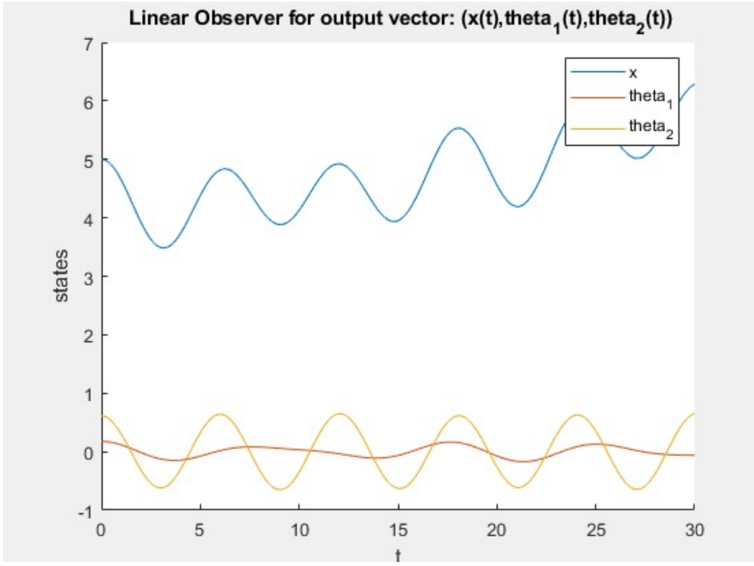


fig: Linear system Observer for vector $x(t), \theta_1(t), \theta_2(t)$

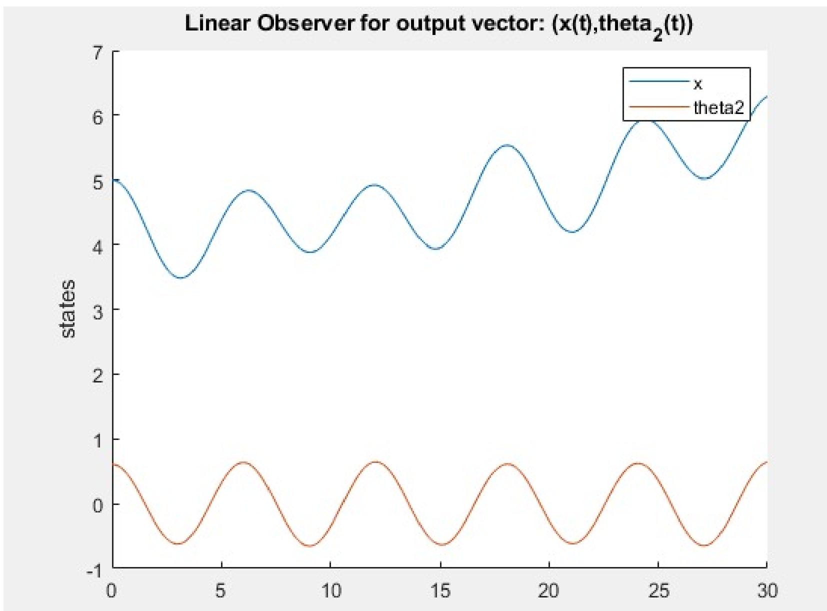


fig: Linear system Observer for vector $x(t), \theta_2(t)$

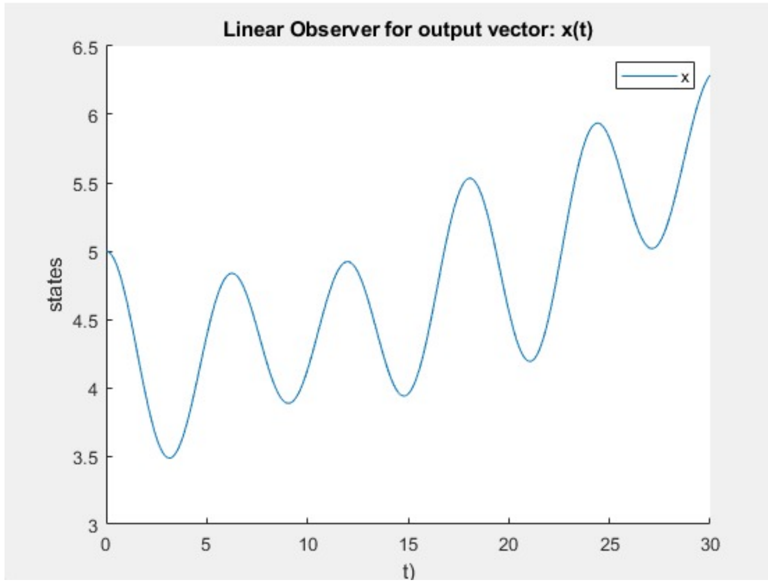


fig : Linear system Observer for vector $x(t)$

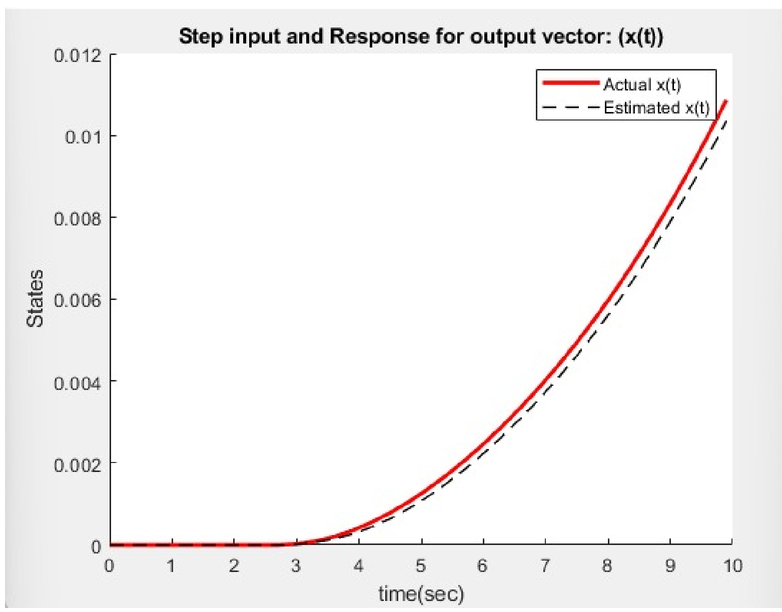


fig : Unit Step Input and Response for vector $x(t)$

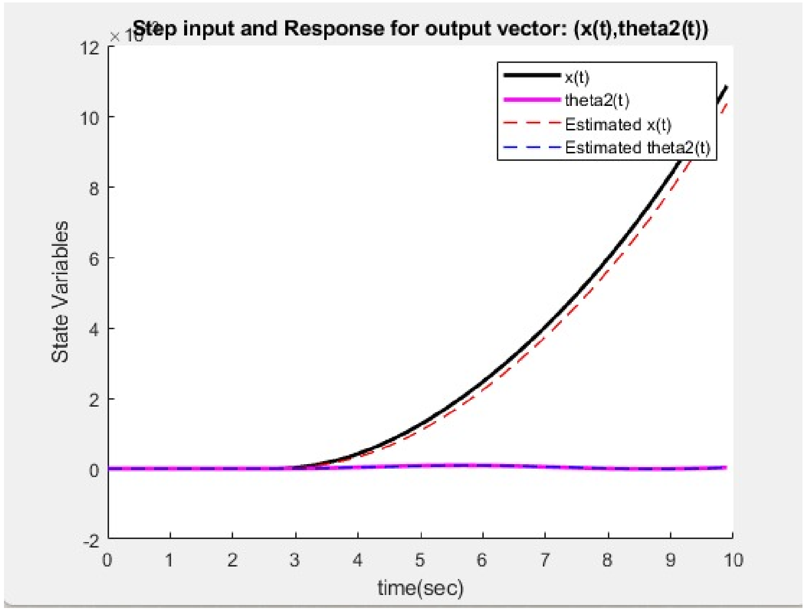


fig : Step Input and Response for vector $x(t), \theta_2(t)$

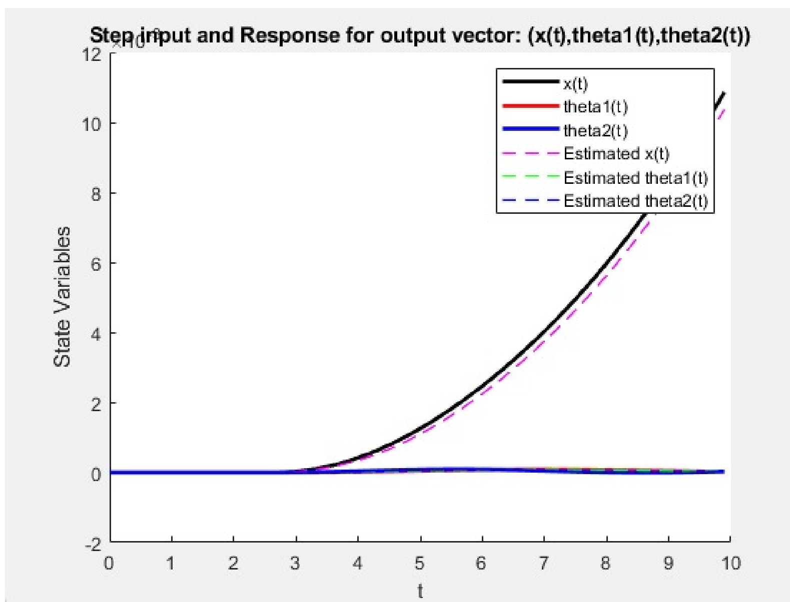


fig: Step Input and Response for vector $x(t), \theta_1(t), \theta_2(t)$

G DESIGN OF LQG CONTROLLER

The LQG or the Linear Gaussian Controller is one of the most used controlling techniques in optimal control and is also extended in solving model predictive control problems. The Gaussian controller is an optimized LQR control and are driven by additive white Gaussian noise. The LQG controller is a combination of LQR and a Kalman filter. The controller is governed by the Kalman-Bucy Filter which is an LQE algorithm that uses a consecutive series of measurements that are observed over an interval of time. The Kalman-Bucy filter produces estimates of unknown variables which can be more accurate compared to the individual measurements.

$$\lim_{t \to \infty} E\left[X_e^{T}(t)\, X_e(t)\right]$$

The cost of the LQG controller is given as

$$J = E\left\{ \lim_{t \to \infty} \left(\frac{1}{\tau} \cdot \int_0^{\tau} \left[x^T\ u^T\right] Q_{u_d u_v} \begin{bmatrix} x \\ u \end{bmatrix} \mathrm{d}t \right) \right\}$$

Where the noise elements are added to the state element, the following are the state equations

$$\dot{X} = AX + Bu + u_d$$

$$Y = C\dot{X} + u_v$$

Where u_d *and* u_v are process noise and measurement noise. The gain matrix is given as

$$Q_{u_d u_v} = E\left(\begin{bmatrix} u_d \\ u_v \end{bmatrix} \left[u_d\ \ u_v \right] \right)$$

Based on the identity matrix for the gain matrix Q, the feedback system obtained is:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3000 \end{bmatrix}$$

$$R = 0.001$$

Now, using the linearized values along with the input matrices A and B, we observe the output from the LQG system and the system from the original system.

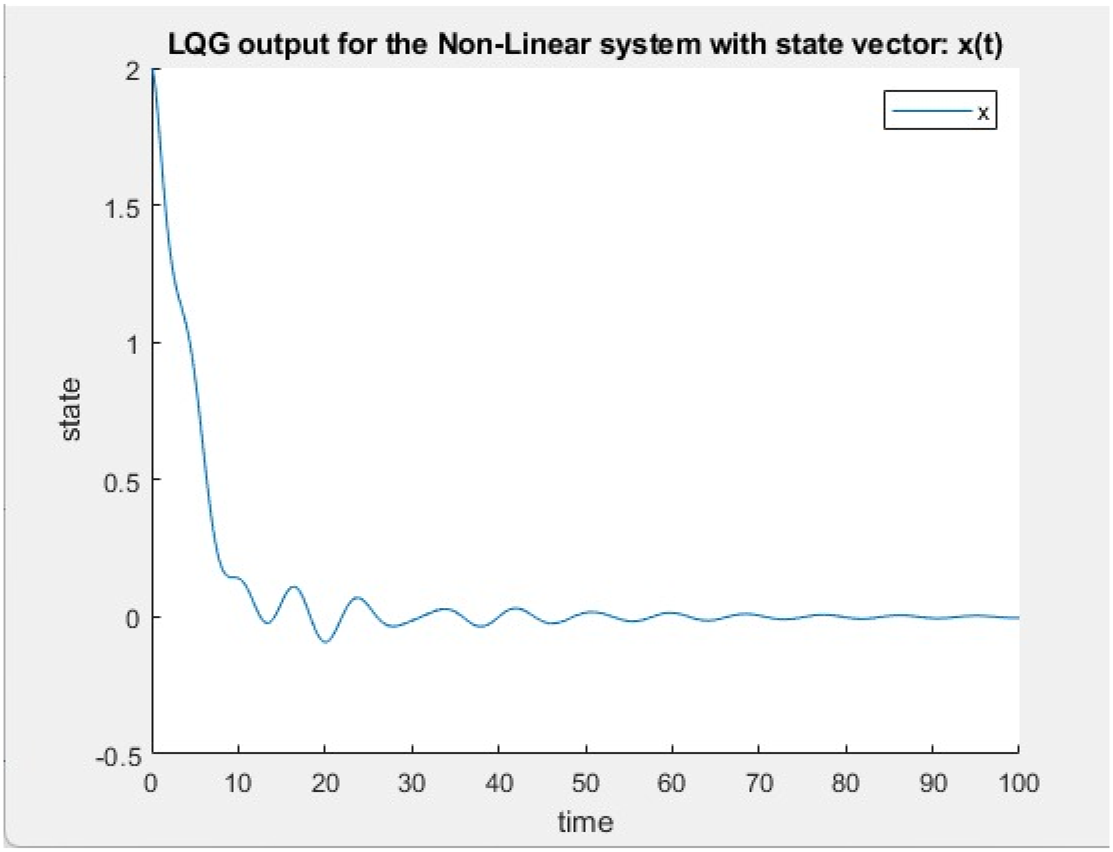


Fig: The LQG Controller Output for Non-linear System with state vector

To asymptotically track a constant reference, we tune the controller such that the variables, x and u reach the referred variable at time $t \to \infty$. During reference tracking, their system generates a cost, in reference to the current constraints to the referred constraints. The main aim is to reduce the cost which is given by

$$\int_0^\infty \left(\vec{X}(t) - \vec{X}_d\right)^T Q\left(\vec{X}(t) - \vec{X}_d\right) + \left(\vec{U}_k - \vec{U}_\infty\right)^T R\left(\vec{U}_k - \vec{U}_\infty\right)\ \mathrm{d}t$$

We can minimize the cost, if there exists a $\vec{U}_\infty$ such that:

$$A\vec{X}_d + B_k\vec{U}_\infty = 0$$

For which the optimal solution is given as

$$U(t) = K\left(\vec{X}(t) - \vec{X}_d\right) + U_\infty$$

Where $K = -R^{-1}B_k{}^T P$ *and* P is the positive definite matrix solution of the Riccati equation. Using these constraints, we can reach the desired output of the system when there's a constant tracking reference x satisfying the equation:

$$A^T P + PA - PBR^{-1}B^T P = -Q$$

Our controller is designed to provide robust support to the mechanical system and rejects any external force applied to the system.

APPENDIX:

1. Controllability_Check:

```
syms L1 L2 m1 m2 g  M

L1 = 20;     % m
L2 = 10;     % m
m1 = 100;    % kg
m2 = 100;    % kg
g = 9.8;     % m/s^2
M = 1000;    % kg

A = [0 1 0 0 0 0; 0 0 -m1*g/M 0 -m2*g/M 0; 0 0 0 1 0 0;
    0 0 -((M*g)+(m1*g))/(M*L1)
    0 -g*m2/(M*L1) 0; 0 0 0 0 0 1;
    0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C = [B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B];
disp(C)
Rank = rank([B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B]);
Det = det([B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B]);
```

2. LQR Controller design:

```
clear variables;
clc;

syms L1 L2 m1 m2 g  M

L1 = 20;     % m
L2 = 10;     % m
m1 = 100;    % kg
m2 = 100;    % kg
g = 9.8;     % m/s^2
M = 1000;    % kg

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

B = transpose([0 1/M 0 1/(l1*M) 0 1/(l2*M)]);
% Output is described by variables x(t),theta1(t),theta2(t).
C = [1 0 0 0 0 0;
     0 0 1 0 0 0;
     0 0 0 0 1 0];
% Force is the only input parameter
D = transpose([1 0 0]);
```

```matlab
% LQR weights
Q = [10 0 0 0 0 0;
     0 0 0 0 0 0;
     0 0 2500 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 2500 0;
     0 0 0 0 0 0];
R = 0.001;

% Get LQR parameters
[K, S, P] = lqr(A, B, Q, R);

% Calculating eigen values after applying LQR
% and getting K values, to check stability using
% Lyapunov's indirect method.
eigen_A = eig(A);
A_LQR = A-B*K;
eigen_A_LQR = eig(A_LQR);

% Define linear system model for LQR controller
system = ss(A-B*K, B, C, D);
tspan = 0:0.1:100;
initial = [0 0 deg2rad(10) 0 deg2rad(27) 0];

% Get system response for linear model
[t,q1] = ode45(@(t,Q) linear_model(t,Q,-K*Q),tspan,initial);
figure(1);
hold on
plot(t,q1(:,1),'r')
plot(t,q1(:,3),'g')
plot(t,q1(:,5),'b')
ylabel('x(t),theta1(t),theta2(t)')
xlabel('time')
title('System response for Linearized system')
legend('x(t)','theta1(t)','theta2(t)')

% Get system response for non-linear model
[t,q2] = ode45(@(t,Q)nonLinear_model(t,Q,-K*Q),tspan,initial);
figure(2);
hold on
plot(t,q2(:,1),'r')
plot(t,q2(:,3),'g')
plot(t,q2(:,5),'b')
ylabel('x(t),theta1(t),theta2(t)')
xlabel('time')
title('System Response for the Non Linearized System')
legend('x(t)','theta1(t)','theta2(t)')
```

3. Observability check:

```
%% Defining variables
syms L1 L2 m1 m2 g  M

% Defining system dynamics
A = [0 1 0 0 0 0; 0 0 -m1*g/M 0 -m2*g/M 0; 0 0 0 1 0 0;
    0 0 -((M*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0; 0 0 0 0 0 1;
    0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C_matrix_1 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0];
C_matrix_2 = [0 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
C_matrix_3 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 1 0];
C_matrix_4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
%% Observability Check
Observability_1 = rank([C_matrix_1' A'*C_matrix_1' ((A')^2)*C_matrix_1' ...
    ((A')^3)*C_matrix_1' ((A')^4)*C_matrix_1' ((A')^5)*C_matrix_1']);
Observability_2 = rank([C_matrix_2' A'*C_matrix_2' ((A')^2)*C_matrix_2' ...
    ((A')^3)*C_matrix_2' ((A')^4)*C_matrix_2' ((A')^5)*C_matrix_2']);
Observability_3 = rank([C_matrix_3' A'*C_matrix_3' ((A')^2)*C_matrix_3' ...
    ((A')^3)*C_matrix_3' ((A')^4)*C_matrix_3' ((A')^5)*C_matrix_3']);
Observability_4 = rank([C_matrix_4' A'*C_matrix_4' ((A')^2)*C_matrix_4' ...
    ((A')^3)*C_matrix_4' ((A')^4)*C_matrix_4' ((A')^5)*C_matrix_4']);
```

4. Luenberger Observer Design:

```
clear all

%% Defining variables
syms m1 g m2 M L1 L2
m1 = 100;
m2 = 100;
M = 1000;
L1 = 20;
L2 = 10;
g = 9.81;
initial_states = [5 0 deg2rad(10) 0 deg2rad(35) 0];
tspan = 0:0.3:10;
```

```
%% Observability Check
A = [0 1 0 0 0 0; 0 0 -m1*g/M 0 -m2*g/M 0; 0 0 0 1 0 0;
    0 0 -((M*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0; 0 0 0 0 0 1;
    0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C1 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0];
C2 = [0 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
C3 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 1 0];
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
D = [0; 0; 0];
Observability1 = rank([C1' A'*C1' ((A')^2)*C1' ((A')^3)*C1' ((A')^4)*C1' ...
    ((A')^5)*C1']);
Observability2 = rank([C2' A'*C2' ((A')^2)*C2' ((A')^3)*C2' ((A')^4)*C2' ...
    ((A')^5)*C2']);
Observability3 = rank([C3' A'*C3' ((A')^2)*C3' ((A')^3)*C3' ((A')^4)*C3' ...
    ((A')^5)*C3']);
Observability4 = rank([C4' A'*C4' ((A')^2)*C4' ((A')^3)*C4' ((A')^4)*C4' ...
    ((A')^5)*C4']);
% We note that, for state vector ((theta1(t),theta2(t)), the system is not
% observable. Hence, we wouldn't design an observer for that.


Actual_sys1 = ss(A,B,C1,D);
Actual_sys3 = ss(A,B,C3,D);
Actual_sys4 = ss(A,B,C4,D);


%% Luenberger Observer Design
Process_noise = 0.5*eye(6);                %Process Noise
Measurement_noise = 0.1*eye(3);          %Measurement Noise
[L_Matrix1,~,~] = lqe(A,Process_noise,C1,Process_noise,Measurement_noise);
[L_Matrix3,~,~] = lqe(A,Process_noise,C3,Process_noise,Measurement_noise);
[L_Matrix4,P,E] = lqe(A,Process_noise,C4,Process_noise,Measurement_noise);

A_Closedloop1 = A-(L_Matrix1*C1);
A_CLosedloop3 = A-(L_Matrix3*C3);
A_CLosedloop4 = A-(L_Matrix4*C4);

Estimated_sys1 = ss(A_Closedloop1,[B L_Matrix1],C1,0);
Estimated_sys3 = ss(A_CLosedloop3,[B L_Matrix3],C3,0);
Estimated_sys4 = ss(A_CLosedloop4,[B L_Matrix4],C4,0);
```

```
%% Estimated vs actual state vectors for step input
unitStep(10:length(tspan)) = 0.5;

[y1,~] = lsim(Actual_sys1,unitStep,tspan);
[x1,~] = lsim(Estimated_sys1,[unitStep;y1'],tspan);

[y3,~] = lsim(Actual_sys3,unitStep,tspan);
[x3,~] = lsim(Estimated_sys3,[unitStep;y3'],tspan);

[y4,~] = lsim(Actual_sys4,unitStep,tspan);
[x4,t] = lsim(Estimated_sys4,[unitStep;y4'],tspan);

figure();
hold on
plot(t,y1(:,1),'r','Linewidth',2)
plot(t,x1(:,1),'k--','Linewidth',1)
ylabel('States')
xlabel('time(sec)')
legend('Actual x(t)','Estimated x(t)')
title('Step input and Response for output vector: (x(t))')
hold off

figure();
hold on
plot(t,y3(:,1),'k','Linewidth',2)
plot(t,y3(:,3),'m','Linewidth',2)
plot(t,x3(:,1),'r--','Linewidth',1)
plot(t,x3(:,3),'b--','Linewidth',1)
ylabel('State Variables')
xlabel('time(sec)')
legend('x(t)','theta2(t)','Estimated x(t)','Estimated theta2(t)')
title('Step input and Response for output vector: (x(t),theta2(t))')
hold off

figure();
hold on
plot(t,y4(:,1),'k','Linewidth',2)
plot(t,y4(:,2),'r','Linewidth',2)
plot(t,y4(:,3),'b','Linewidth',2)
plot(t,x4(:,1),'m--','Linewidth',1)
plot(t,x4(:,2),'g--','Linewidth',1)
plot(t,x4(:,3),'b--','Linewidth',1)
ylabel('State Variables')
xlabel('t')
legend('x(t)','theta1(t)','theta2(t)','Estimated x(t)', ...
    'Estimated theta1(t)','Estimated theta2(t)')
title(['Step input and Response for output vector:' ...
    ' (x(t),theta1(t),theta2(t))'])
hold off
```

```matlab
%% Observer design response for Linearized system
[t,q1] = ode45(@(t,q)linear_model_Observer1(t,q,L_Matrix1), ...
    tspan,initial_states);
figure();
hold on
plot(t,q1(:,1))
ylabel('states')
xlabel('t)')
title('Linear Observer for output vector: x(t)')
legend('x')
hold off

[t,q3] = ode45(@(t,q)linear_model_Observer3(t,q,L_Matrix3),tspan, ...
    initial_states);
figure();
hold on
plot(t,q3(:,1))
plot(t,q3(:,5))
ylabel('states')
xlabel('t')
title('Linear Observer for output vector: (x(t),theta_2(t))')
legend('x','theta2')
hold off

[t,q4] = ode45(@(t,q)linear_model_Observer4(t,q,L_Matrix4),tspan, ...
    initial_states);
figure();
hold on
plot(t,q4(:,1))
plot(t,q4(:,3))
plot(t,q4(:,5))
ylabel('states')
xlabel('t')
title(' Linear Observer for output vector: (x(t),theta_1(t),theta_2(t))')
legend('x','theta_1','theta_2')
hold off

%%
%% Observer design response for Non Linear system
[t,q1] = ode45(@(t,q)nonLinear_model_Observer1(t,q,1,L_Matrix1),tspan, ...
    initial_states);
figure();
hold on
plot(t,q1(:,1))
ylabel('states')
xlabel('t)')
title('Non-Linear System Observer for output vector: x(t)')
legend('x')
hold off
```

```
[t,q3] = ode45(@(t,q)nonLinear_model_Observer3(t,q,1,L_Matrix3),tspan, ...
    initial_states);
figure();
hold on
plot(t,q3(:,1))
plot(t,q3(:,5))
ylabel('states')
xlabel('time (sec)')
title('Non-Linear System Observer for output vector: (x(t),theta_2(t))')
legend('x','theta_2')
hold off

[t,q4] = ode45(@(t,q)nonLinear_model_Observer4(t,q,1,L_Matrix4),tspan, ...
    initial_states);
figure();
hold on
plot(t,q4(:,1))
plot(t,q4(:,3))
plot(t,q4(:,5))
ylabel('states')
xlabel('t')
title(['Non-Linear System Observer for output vector: (x(t),theta_1(t),' ..
    'theta_2(t))'])
legend('x','theta_1','theta_2')
hold off
```

5. LQG Controller Design:

```
clear all
```

```
%% Defining variables
syms m1 g m2 M L1 L2 x dx
m1 = 100;
m2 = 100;
M = 1000;
L1 = 20;
L2 = 10;
g = 9.81;
tspan = 0:0.1:100;
% q = [x dx t1 dt1 t2 dt2];
%Enter initial conditions
initial_states = [2 0 deg2rad(2) 0 deg2rad(5) 0];
```

```
%% Linearized Model
A = [0 1 0 0 0 0; 0 0 -m1*g/M 0 -m2*g/M 0; 0 0 0 1 0 0;
    0 0 -((M*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0; 0 0 0 0 0 1;
    0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C1 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0];
D = [1;0;0];
Actual_sys1 = ss(A,B,C1,D);
```

```
%% LQR Controller
Q = [100 0 0 0 0 0;
     0 1000 0 0 0 0;
     0 0 3000 0 0 0;
     0 0 0 0 0 0;
     0 0 0 0 3000 0;
     0 0 0 0 0 3000];
R = 0.001;
[K,S,~] = lqr(A,B,Q,R);
LQR_sys = ss(A-B*K,B,C1,D);
% step(sys,200);
```

```
%% Kalman Estimator Design
Process_noise = 0.01*eye(6);                %Process Noise
Measurement_noise = 0.001;                      %Measurement Noise
[L_Matrix1,P,E] = lqe(A,Process_noise,C1,Process_noise ...
    ,Measurement_noise*eye(3)); %Considering vector output: x(t)
Ac1 = A-(L_Matrix1*C1);
Estimated_sys1 = ss(Ac1,[B L_Matrix1],C1,0);


%% Non-linear Model LQG Response
[t,q1] = ode45(@(t,q)nonLinear_model_Observer1(t,q,-K*q,L_Matrix1), ...
    tspan,initial_states);
figure();
hold on
plot(t,q1(:,1))
ylabel('state')
xlabel('time')
title('LQG output for the Non-Linear system with state vector: x(t)')
legend('x')
hold off
```