ENPM 667- Control of Robotic Systems

PROJECT 2

TECHNICAL REPORT

ON

Designing a LQR and LQG Controller for a cart with two suspended loads

Suhas Nagaraj (UID : 119505373, Directory ID: suhas99)

Swaraj M Rao (UID: 120127007, Directory ID: swarajmr)

Dayanidhi K (UID: 120426711, Directory ID: dotsv)

Table of Contents

PROBLEM DEFINITION

Consider a crane that moves along a one-dimensional track. It behaves as a frictionless cart with mass M actuated by an external force F that constitutes the input of the system. There are two loads suspended from cables attached to the crane. The loads have mass m_1 and m_2, and the lengths of the cables are l_1 and l_2, respectively. The following figure depicts the crane and associated variables used throughout this project.

Part: 1

Obtain the equations of motion for the system and the corresponding nonlinear state-space representation.

Part: 2

Obtain the linearized system around the equilibrium point specified by x = 0 and θ1 = θ2 = 0. Write the state-space representation of the linearized system.

Part: 3

Obtain conditions on M, m_1, m_2, l_1, l_2 for which the linearized system is controllable.

Part: 4

Choose M = 1000Kg, m_1 = m_2 = 100Kg, l_1 = 20m and l_2 = 10m. Check that the system is controllable and obtain an LQR controller. Simulate the resulting response to initial conditions when the controller is applied to the linearized system and to the original nonlinear system. Adjust the parameters of the LQR cost until you obtain a suitable response. Use Lyapunov's indirect method to certify stability (locally or globally) of the closed-loop system.

Part: 5

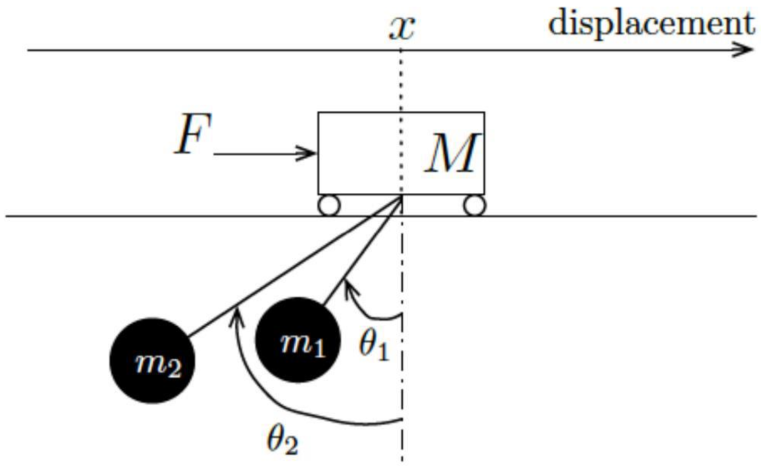
Suppose that you can select the following output vectors: $x(t), (\theta 1(t), \theta 2(t)), (x(t), \theta 2(t))\ or (x(t), \theta 1(t), \theta 2(t))$. Determine for which output vectors the linearized system is observable.

Part: 6

Obtain your "best" Luenberger observer for each one of the output vectors for which the system is observable and simulate its response to initial conditions and unit step input. The simulation should be done for the observer applied to both the linearized system and the original nonlinear system.

Part: 7

Design an output feedback controller for your choice of the "smallest" output vector. Use the LQG method and apply the resulting output feedback controller to the original nonlinear system. Obtain your best design and illustrate its performance in simulation. How would you reconfigure your controller to asymptotically track a constant reference on x? Will your design reject constant force disturbances applied on the cart?



INTRODUCTION

The primary objective of this project is to derive the dynamic equations of a crane with two loads suspended and control its states by implementing LQR and LQG controllers. The nonlinear system will first be linearized around its equilibrium point and then we check for the controllability of the system. Following this, we initialize the given mass and length values to the system, check its stability and design an LQR controller for optimal control. We use the techniques of Lyapunov's indirect method to check for the stability of the closed loop system. The LQR controller designed is implemented for the linearized system and to the original nonlinear system. Then we check for the observability of the output vectors of the closed loop system. For the states which are observable, we designed a Luenberger Observer and tested its response to initial conditions and unit step input. Then an LQG system is implemented to the smallest output vector, and it is simulated for both the linearized system and the original nonlinear system.

EQUATIONS OF MOTION OF THE SYSTEM

The crane system under consideration functions on a linear trajectory, with a frictionless cart (with mass M) driven by an external force (F). There are two loads suspended from this crane, with each load having a mass of $(m1)$ and $(m2)$. The weights are attached to a crane using cables of lengths $(l1)$ and $(l2)$. The suspended masses have the potential to fluctuate as the crane moves along its path, which can impact the overall dynamics of the system. The mobility and load oscillations of the crane are influenced by various elements, including the masses in motion, the cable lengths, and the force exerted on the cart. This generates a dynamic and interconnected mechanical system.

Using Euler Lagrange equations used to derive the given system's motion and dynamics in general coordinates.

The coordinates of the system are x, θ_1, & θ_2

<u>Applying Euler Lagrange equation for each of the coordinate, we get:</u>

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

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

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

<u>Potential Energy of the system:</u>

The potential energy component of the system involves only the 2 masses attached to the cart and it is given by:

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

Kinetic Energy of the System:

For calculating the Kinetic Energy of the cart, we first need to find its velocity components along the x and y axis. For this, we first take the position vector of the mass as a function of variables $\theta_1 and\ \theta_2$. The process is as given below,

We find the position vector by refereeing to the system diagram,

Position Vector of the mass 1 = $r_1(t)$ = $(x - l_1 \sin\theta_1)\ i - \ (l_1 \cos\theta_1)\ j$

Position Vector of the mass 2 = $r_2(t)$ = $(x - l_2 \sin\theta_2)\ i - \ (l_2 \cos\theta_2)\ j$

Velocity of mass 1:

$V_1(t) = \frac{d}{dt} r_1(t)$

$V_1(t) = \left(\dot{x} - l_1\dot{\theta}_1 \cos\theta_1\right) i + \ \left(l_1\dot{\theta}_1 \sin\theta_1\right) j$

Consider $a_1 = \ \dot{x} - \ l_1\dot{\theta}_1 \cos\theta_1$ and $a_2 = \ l_1\dot{\theta}_1 \sin\theta_1$

Then, $V_1 = \ a_1 i + \ a_2 j$

$V_1.V_1 = \ (a_1 i + \ a_2 j)(a_1 i + \ a_2 j)$

$V_1.V_1 = \ a_1^2 i^2 + \ 2\, a_1 a_2\, ij + a_2^2 j^2$

$V_1.V_1 = \ a_1^2 i^2 + a_2^2 j^2$

$V_1^2 = \ a_1^2 + \ a_2^2$

$V_1^2 = \left(\dot{x} - l_1\dot{\theta}_1 \cos\theta_1\right)^2 + \ \left(l_1\dot{\theta}_1 \sin\theta_1\right)^2$

Similarly:

$V_2 = \frac{d}{dt} r_2(t)$

$V_2(t) = \left(\dot{x} - l_2\dot{\theta}_2 \cos\theta_2\right) i + \ \left(l_2\dot{\theta}_2 \sin\theta_2\right) j$

$V_2.V_2 = V_2^2 = \left(\dot{x} - l_2\dot{\theta}_2 \cos\theta_2\right)^2 + \ \left(l_2\dot{\theta}_2 \sin\theta_2\right)^2$

Kinetic Energy of the cart M:

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

Kinetic Energy of mass 1:

$$KE_1 = \frac{1}{2}\ m_1 V_1^2$$

$$KE_1 = \frac{1}{2}\ m_1 \left(\left(\dot{x} - l_1\dot{\theta}_1 \cos\theta_1\right)^2 + \left(l_1\dot{\theta}_1 \sin\theta_1\right)^2 \right)$$

Kinetic Energy of mass 2:

$$KE_2 = \frac{1}{2}\ m_2 V_2^2$$

$$KE_2 = \frac{1}{2}\ m_2 \left(\left(\dot{x} - l_2\dot{\theta}_2 \cos\theta_2\right)^2 + \left(l_2\dot{\theta}_2 \sin\theta_2\right)^2 \right)$$

Total Kinetic energy of the system:

$$KE = KE_M + KE_1 + KE_2$$

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

Substituting the values of Potential energy and Kinetic energy:

$$L = KE - PE$$

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

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

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

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

<u>Applying the Lagrangian equations for x:</u>

$$\frac{d}{dt}\frac{\partial l}{\partial \dot{x}} - \frac{\partial l}{\partial x} = \mathrm{F}$$

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

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

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

Substituting the values:

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

<u>Applying the Lagrangian equations for θ_1:</u>

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

$$\frac{\partial L}{\partial \dot{\theta}_1} = 0 + 2\dot{\theta}_1\left(\frac{1}{2} m_1 l_1^2\right) - m_1\dot{x} l_1 \cos\theta_1$$

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

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

$$\frac{\partial L}{\partial \theta_1} = 0 + 0 + m_1\dot{x}\, l_1\dot{\theta}_1 sin\theta_1 - m_1\, g l_1 \sin\theta_1$$

Substituting in the equation:

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

$$\ddot{\theta}_1 m_1 l_1^2 - m_1 l_1 \ddot{x}\cos\theta_1 + m_1\, g l_1 \sin\theta_1 = 0$$

<u>Applying the Lagrangian equations for θ_2</u>

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

$$\frac{\partial L}{\partial \dot{\theta}_2} = 0 + 0 + 0 + \frac{1}{2} m_2 l_2^2\left(\dot{\theta}_2 x_2\right) - m_2\dot{x}\, l_2 \cos\theta_2$$

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

$$\frac{\partial L}{\partial \theta_2} = 0 + 0 + 0 + 0 + m_2\dot{x}\, l_2\dot{\theta}_2 sin\theta_2 + m_2\, g l_2 \sin\theta_2$$

Substituting in the equation:

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

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \dot{x}\dot{\theta}_2 sin\theta_2 + m_2\, g l_2 \sin\theta_2 = 0$$

From the result obtained after substitution:

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

$m_1 l_1^2\ddot{\theta}_1 - m_1 l_1\ddot{x}\cos\theta_1 + m_1\, g l_1\sin\theta_1 = 0$

$m_2 l_2^2\ddot{\theta}_2 - m_2 l_2\ddot{x}\cos\theta_2 + m_2\, g l_2\sin\theta_2 = 0$

Simplifying:

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

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

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

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

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

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

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

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

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

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

The state space representation of the nonlinear system:

The System States are selected as:

$$X = \left[x, \dot{x}, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2\right]$$

$$\dot{X} = \left[\dot{x}, \ddot{x}, \dot{\theta}_1, \ddot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_2\right]$$

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

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

$$\begin{bmatrix} 0 \\ \frac{1}{(M + m_1 + m_2 - m_1\cos^2\theta_1 - m_2\cos^2\theta_2)} \\ 0 \\ \frac{1}{l_1(M + m_1 + m_2 - m_1\cos^2\theta_1 - m_2\cos^2\theta_2)} \\ 0 \\ \frac{1}{l_2(M + m_1 + m_2 - m_1\cos^2\theta_1 - m_2\cos^2\theta_2)} \end{bmatrix} * F$$

Linearizing the nonlinear system:

For linearizing the system around the equilibrium point of $x = 0,\ \theta_1 = 0\ and\ \theta_2 = 0$, we have considered the small angle assumption technique.

In the small angle assumption method, we consider the angles to be very small, from which we can assume the following things:

$\sin\theta \approx \theta, \cos\theta\ \approx 1\,, \dot{\theta}^2 = 0$

Now, by considering the assumptions made above and by considering the equilibrium point, the system is linearized as follows:

$$\ddot{x} = \left[\frac{F - m_2 g\theta_2 - m_1 g\theta_1}{M}\right]$$

$$\ddot{\theta}_1 = \frac{\left[\frac{F - m_2 g\theta_2 - m_1 g\theta_1}{M}\right]}{l_1} - \frac{g\theta_1}{l_1} = \frac{F - m_2 g\theta_2 - m_1 g\theta_1}{Ml_1} - \frac{g\theta_1}{l_1} = \frac{F - m_2 g\theta_2 - m_1 g\theta_1 - Mg\theta_1}{Ml_1}$$

$$\ddot{\theta}_1 = \frac{\left[\frac{F - m_2 g\theta_2 - m_1 g\theta_1}{M}\right]}{l_2} - \frac{g\theta_2}{l_2} = \frac{F - m_2 g\theta_2 - m_1 g\theta_1}{Ml_2} - \frac{g\theta_2}{l_2} = \frac{F - m_2 g\theta_2 - m_1 g\theta_1 - Mg\theta_2}{Ml_2}$$

The state space representation of the linearized system:

$$X\dot{(t)} = AX(t) + Bu(t)$$

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

Conditions on $M, m_1, m_2,\ l_1,\ l_2$ for which the linearized system is controllable:

The linearized system's state equation is called controllable on the finite time interval $[t_0, t_f]$ if given any initial state $x(t_0) = x_0$, there exists a continuous input signal $u(t)$ such that the corresponding solution satisfies $x(t_f) = 0$ and if the Gramian of the controllability which is given by

$$\omega(t_0, t_f) = \int_{t_0}^{t_f} \Phi(t_0, t_f)\ B(t)\ B^T(t)\Phi(t_0, t_f)dt \quad \text{, is invertible}$$

For the Linear Time invariant state equation, the Gramian of Controllability is invertible if and only if the $n\ x\ nm$ controllability matrix satisfies the following criteria:

$$Rank\ [B_k \quad AB_k \quad A^2B_k \quad \dots\dots A^{n-1}B_k] = n$$

That is the matrix obtained should be a full rank matrix.

The computations have been carried out in MATAB and the output obtained is as follows,

```
% Checking Rank of Grammian of Controllability for the system
clc
clear

%Defining the required symbolic variables to define the system in MATLAB
syms M m1 m2 g l1 l2 F

% A and B Matrix as obtained from the state space equation
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 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];
disp("The A Matrix is : ");
```

```
The A Matrix is :
```

```
disp(A);
```

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

```
B=[0;1/M;0;1/(M*l1);0;1/(M*l2)];
disp("The B Matrix is : ");
```

```
The B Matrix is :
```

```
disp(B);
```

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

```
%Computing the controllability matrix.
ABk=A*B;
A2Bk=A*A*B;
A3Bk=A*A*A*B;
A4Bk=A*A*A*A*B;
A5Bk=A*A*A*A*A*B;

%Defining the controllability matrix
Controllability_Matrix=[B ABk A2Bk A3Bk A4Bk A5Bk];
disp("The Controllability Matrix is : ")
```

```
The Controllability Matrix is :
```

```
disp(Controllability_Matrix);
```

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

Where, s_1 , s_2 , s_3 , s_4 , s_5 , s_6 , s_7, s_8 are,

$$\sigma_1 = -\frac{g\,m_1}{M^2\,l_2} - \frac{g\,m_2}{M^2\,l_2}$$

$$\sigma_2 = \frac{\frac{g^2\,m_2\,(M+m_1)}{\sigma_8} + \frac{g^2\,m_2\,(M+m_2)}{\sigma_8}}{M\,l_2} + \frac{\frac{g^2\,(M+m_1)^2}{\sigma_8} + \frac{g^2\,m_1\,m_2}{\sigma_8}}{M\,l_2}$$

$$\sigma_3 = \frac{\frac{g^2\,m_1\,(M+m_1)}{\sigma_8} + \frac{g^2\,m_1\,(M+m_2)}{\sigma_8}}{M\,l_2} + \frac{\frac{g^2\,(M+m_2)^2}{\sigma_8} + \frac{g^2\,m_1\,m_2}{\sigma_8}}{M\,l_2}$$

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

$$\sigma_5 = -\frac{g (M+m_2)}{\sigma_8} - \frac{g m_1}{\sigma_8}$$

$$\sigma_6 = -\frac{g (M+m_1)}{\sigma_8} - \frac{g m_2}{\sigma_8}$$

$$\sigma_7 = \frac{g^2 m_1 m_2}{M^2 l_2}$$

$$\sigma_8 = M^2 l_2^2$$

Rank test is carried out on the matrix obtained above; The rank test carried out in MATLAB provided the following results

```
%Checking rank of controllability matrix
n=rank(Controllability_Matrix);
disp("The rank of Controllability Matrix is : ")
```

```
The rank of Controllability Matrix is :
```

```
disp(rank(Controllability_Matrix));
```

```
6
```

The matrix of the system is a full rank matrix, and hence it can be said that the Gramian of Controllability is invertible, and that the system is **controllable**.

But we also should find the conditions on which the linearized system is not controllable.

Let's consider the following case, if the cable lengths l_1 *and* l_2 are equal, will the system be controllable?

```
disp(simplify(det(Controllability_Matrix)));
```

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

```
%To obtain conditions on M,m1,m2, l1, l2 for which the linearized system is controllable

% Substituting l2 to l1 to obtain the controllability matrix when the l1=l2
% to check controllability
Controllability_Matrixl1l2=subs(Controllability_Matrix, l1, l2);
disp("The Controllability Matrix when l1 = l2 is : ")
```

```
The Controllability Matrix when l1 = l2 is :
```

```
disp(Controllability_Matrixl1l2);
```

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

```
disp("Rank of Controllability Matrix for l1 = l2 condition is : ")
```

```
Rank of Controllability Matrix for l1 = l2 condition is :
```

```
disp(rank(Controllability_Matrixl1l2));
```

```
    4
```

```
disp("The Determinant of the Controllability Matrix for l1 = l2 condition is : ")
```

```
The Determinant of the Controllability Matrix for l1 = l2 condition is :
```

```
disp(det(Controllability_Matrixl1l2));
```

```
0
```

The Gramian of controllability is not invertible when $l_1 = l_2$. Thus, one of the conditions for the system to be controllable is $l_1 \neq l_2$

Also, the masses M, m_1, m_2 should all be greater than zero, i.e., $M, m_1, m_2 > 0$ for the system to be controllable.

LQR CONTOLLER

If the pair (A, B_k) of the closed loop system is stabilizable, then we look for k that minimizes the following cost and that is,

$$J(K, \vec{x}_0) = \int_0^{\infty} x^T\, Q\, x(t) \;+\; u_k^T(t)\, R\, U_k(t)\, dt \;,$$

where $Q \; and \; R$ are symmetric positive definite matrices.

The optimal solution for the LQR controller is given by

$$K \;=\; -R^{-1} {B_k}^T\, P$$

Where P is the symmetric positive definite solution of the following stationary Riccati Equation,

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

And the optimal control will be $\tilde{u}_k(t) \;=\; K\, \tilde{X}(t)$

The weighing matrices $Q \; and \; R$ are important components of an LQR controller and the compositions of these have a significant influence on the system performance. The number of elements of $Q \; and \; R$ matrices depend on the number of state variables of the system (n) and the number of input variables (m) respectively.

The Q matrix defines the cost associated with the state variables of the system. It is a positive semi-definite matrix. Larger values along the diagonal of the Q matrix indicate a higher cost associated with minimizing the corresponding state variables. By adjusting the values in the Q matrix, the controllers' behavior with respect to the state variables can be made to suit the required controller design.

The R matrix defines the cost associated with the control inputs of the system. It is a positive definite matrix. By adjusting the values in the R matrix, you can influence the controller's behavior with respect to the control inputs.

A common way to tune these matrices is through an iterative process, simulation, or using trial-and-error, considering the dynamics and requirements of the specific system being controlled.

LYAPUNOV'S' INDIRECT METHOD

The indirect method of Lyapunov uses the linearization of a system around the point of interest to determine the local stability of the original system.

For the system, to determine the stability by using Lyapunov Indirect Method, we need to check the eigen values of the state space dynamics $A + B_k$ Matrix,

- If the eigen values of $A + B_k$ have negative real part, then the original system is at least locally stable around the equilibrium point. In this case a Lyapunov function for the linearized system will be valid locally
- If at least one eigen values of $A + B_k$ is positive, then the original system is unstable around the equilibrium point.
- If the eigen values of $A + B_k$ have non positive real but at least one is on the imaginary axis, then the indirect method is inconclusive.

For all the simulations the initial conditions are assumed as:

- $x = 1$
- $\dot{x} = 0$
- $\theta_1 = 10^0$
- $\dot{\theta}_1 = 0$
- $\theta_2 = 20^0$
- $\dot{\theta}_2 = 0$

LQR CONTROLLER APPLIED TO THE LINEAR SYSTEM:

The computations have been carried out in MATAB and the output obtained for the Linear system with an LQR Controller is as follows,

```
%LQR controller for the linearised system
clc;
clear;
% Given system values
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;

% Defining the State Equation matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];
```

```
B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
C=eye(6);
D=0;
```

Checking the controllability of the linearized system

```
%Determining the controllability of the system using inbuilt MATLAB controllability checker
disp(" Rank of Controllability Matrix is :");
```

```
 Rank of Controllability Matrix is :
```

```
disp(rank(ctrb(A,B)));
```

```
     6
```

Here the rank of the system obtained is 6, this is equal to the number of states of the system. The system satisfies the rank test, hence the system is controllable.

Defining the initial conditions and the state weighing matrix ($\boldsymbol{Q}$) and the control input weighing matrix ($\boldsymbol{R}$)

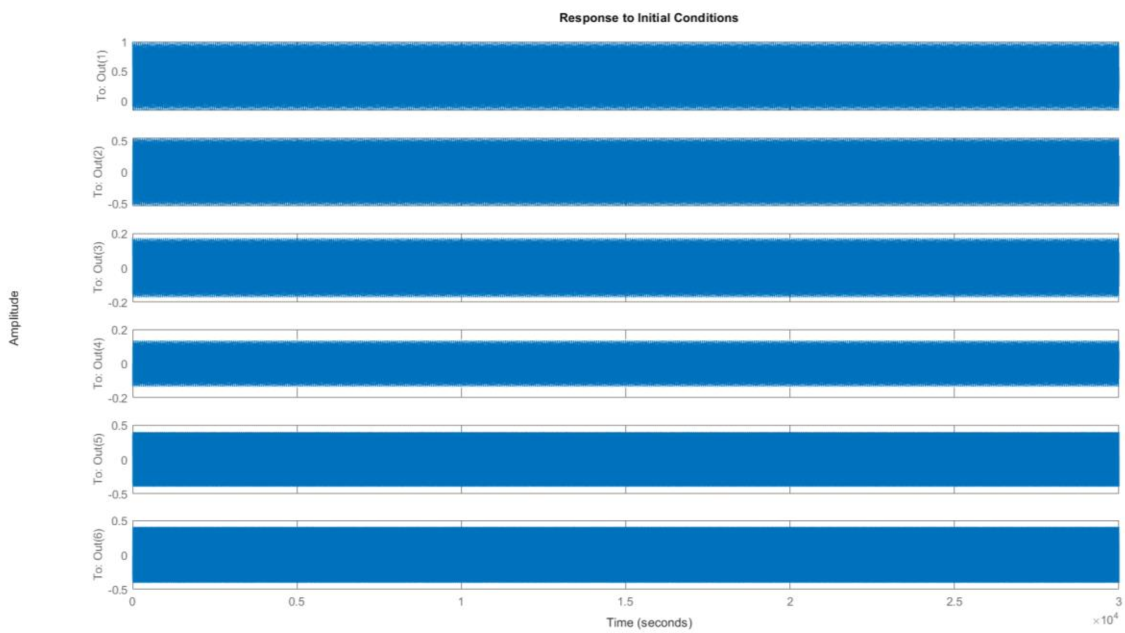
```
% Initial conditions for simulating the response of the system
X_Initial=[1;0;deg2rad(10);0;deg2rad(20);0];
%Assume Values of Q and R such that it stabilizes the system

% Assigning the Q and R cost function values to check the performance of
% the controller system
Q=[10 0 0 0 0 0;
    0 10 0 0 0 0 ;
    0 0 200 0 0 0;
    0 0 0 200 0 0;
    0 0 0 0 200 0;
    0 0 0 0 0 200];

R=0.001;
```

Initial response of the system without LQR controller applied to the system.

```
%Defining the state space representation of the system
system=ss(A,B,C,D);
disp("Inititial response without Controller");
```

```
Inititial response without Controller
```

```
%Plot the output of the initial response of the system
figure
initial(system, X_Initial);
grid on
```



It can be observed that the response of the system without a controller is not stable. We proceed with the implementation of an LQR controller and observe the stability of the system with the assumed conditions.

Implementing the LQR controller:

```
% Implementing the LQR Control for visualising to intial conditions
[K, S, Poles] = lqr(system,Q,R);

%To display the output of the LQR Controller Applied to the system along
%with the cost functions
disp("The gain matrix K is:");
```

```
The gain matrix K is:
```

```
disp(K);
```

```
  100.0000  506.5567  141.5696  -24.6679  182.1430   75.9540
```

```
disp("Solution to the associated Ricattie Equation : ")
```

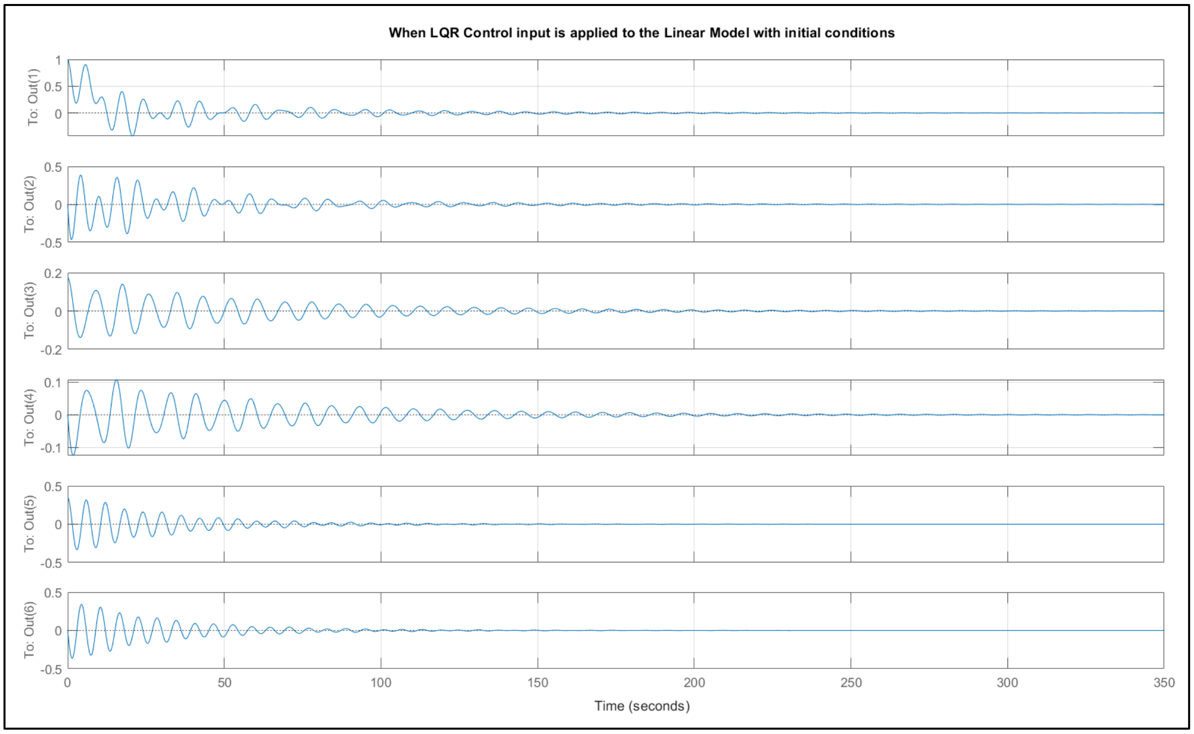
```
Solution to the associated Ricattie Equation :
```

```
disp(S);
```

```
   1.0e+04 *

    0.0051    0.0123   -0.0002   -0.0229    0.0008   -0.0119
    0.0123    0.0624    0.0216   -0.1164    0.0157   -0.0593
   -0.0002    0.0216    1.0925   -0.0100    0.0199   -0.0698
   -0.0229   -0.1164   -0.0100    2.2329    0.0696    0.0232
    0.0008    0.0157    0.0199    0.0696    0.6536   -0.0097
   -0.0119   -0.0593   -0.0698    0.0232   -0.0097    0.6572
```

The output of the Linearized system with an LQR controller is as follows,

```
%To obtain state space representation of the closed loop system
system2=ss(A-(B*K), B,C,D);

%Plotting the output of the closed loop system with the chosen initial
%condition
figure
initial(system2,X_Initial);
title("When LQR Control input is applied to the Linear Model with initial conditions");
grid on
```



It can be observed from the graph that the LQR Controller Stabilizes the system. In the above graph, Out(1) refers to state x, Out(2) refers to state $\dot{x}$, Out(3) refers to state θ_1, Out(4) refers to state $\dot{\theta}_1$, Out(5) refers to state θ_2 and Out(4) refers to state $\dot{\theta}_2$.

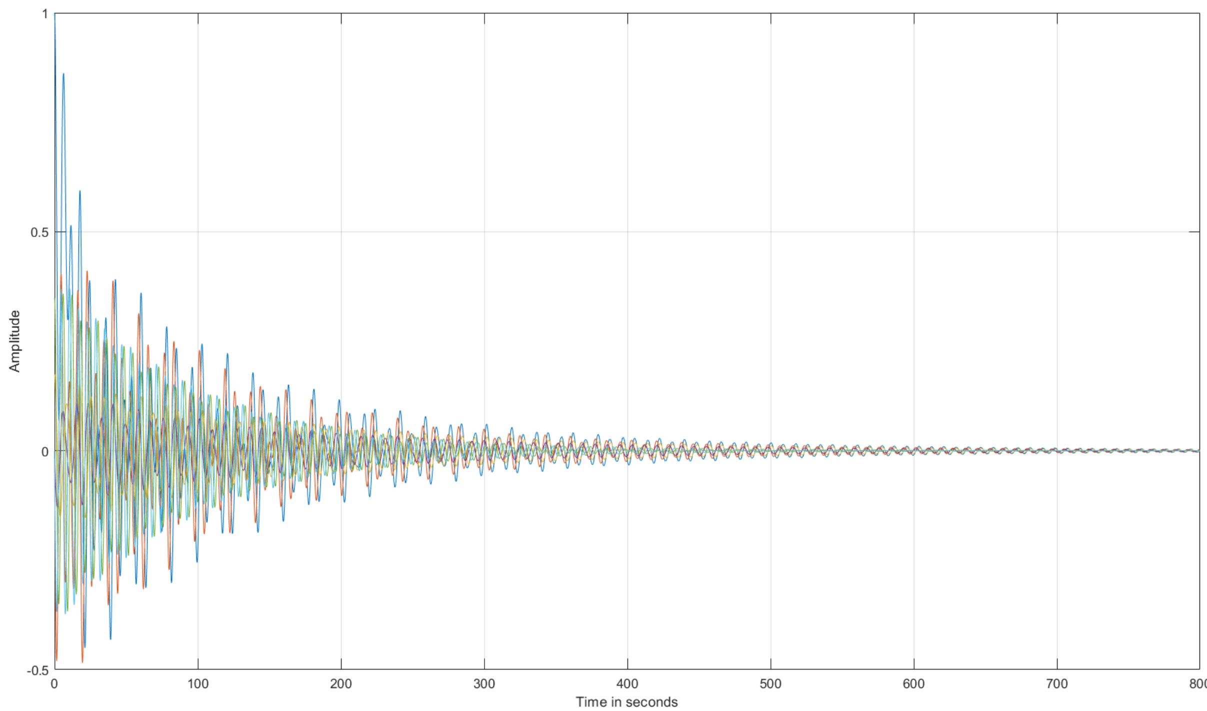
LQR Applied to the Non-Linear System

The computations have been carried out in MATAB and the output obtained for the Non-Linear system with an LQR Controller is as follows,

```
clc;
clear;

%Setting Initial Conditions
y0=[1;0;deg2rad(10);0;deg2rad(20);0];

%Defining time span and sampling rate for the graph generation
timespan=0:0.1:800;

%Utilizing ODE45 for defining the differential Equation
[t1,y1]= ode45(@twosuspendedmass,timespan,y0);


function ydot=twosuspendedmass(t1,y)
    % Given system values
    M=1000;
    m1=100;
    m2=100;
    l1=20;
    l2=10;
    g=9.81;

    % Define system matrices
    A=[0 1 0 0 0 0;
        0 0 (-m1*g/M) 0 (-m2*g/M) 0;
        0 0 0 1 0 0;
        0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
        0 0 0 0 0 1;
        0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];

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

    %Assume Values of Q and R (cost functions)such that it stabilizes the system
    Q=[10 0 0 0 0 0;
        0 10 0 0 0 0 ;
        0 0 200 0 0 0;
        0 0 0 200 0 0;
        0 0 0 0 200 0;
        0 0 0 0 0 200];

    R=0.01;

    C=eye(6);
    D=0;

    %Obtaining the Gain Matrix for simulating the system
    [K, S, P]=lqr(A,B,Q,R);

    %Force
    F=-K*y;


    ydot=zeros(6,1);
    %From state space representation, y(1) = x
    ydot(1)=y(2);

    %From state space representation, y(2) = x_dot
    ydot(2)= (F - (g/2)*(m1*sin(2*y(3)) + m2*sin(2*y(5))) - (m1*l1*(y(4)^2)*sin(y(3))) - (m2*l2*(y(6)^2)*sin(y(5)))) / (M+m1*((sin(y(3)))^

    %From state space representation, y(3) = theta1
    ydot(3)=y(4);
```

```
    %From state space representation, y(4) = theta1_dot
    ydot(4)= (ydot(2)*cos(y(3)) - g*(sin(y(3))))/l1;

    %From state space representation, y(5) = theta2
    ydot(5)=y(6);


    %From state space representation, y(6) = theta2_dot
    ydot(6)=(ydot(2)*cos(y(5)) - g*(sin(y(5))))/l2;

end

%Plotting all graphs from the obtained values
disp("The graph for LQR Applied to the Non Linear System")
```

Response of the Non Linear system with an LQR Controller



The Stability of the system is determined by employing Lyapunov's Indirect method, the eigen values of the state space dynamics $(A + B_k)$ matrix obtained are

$\lambda_1 =$ -0.3420 + 0.3216i

$\lambda_2 =$ -0.3420 - 0.3216i

$\lambda_3 =$ -0.0251 + 1.0373i

$\lambda_4 =$ -0.0251 - 1.0373i

$\lambda_5 =$ -0.0140 + 0.7234i

$\lambda_6 =$ -0.0140 - 0.7234i

The real part of all the eigen values are negative, hence according to the Lyapunov's Indirect method the system in locally stable

OBSERVABILITY

For the Linear Time invariant time system, the state equation is observable if and only if the $np \ x \ n$ observability matrix satisfies the following condition.

$$Rank \begin{bmatrix} C \\ CA \\ CA^2 \\ . \\ . \\ CA^{n-1} \end{bmatrix} = n$$

The observability of for the selected output vectors obtained through MATLAB code is as follows,

```
clc;
clear;

% Given information
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;

% Define system matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];


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

%---------------------------------------------------------------------------------------------%
%Selecting x(t) as the output vector
x_only=[1 0 0 0 0 0];
%Obtaining the observability Matrix
observability_x_only=obsv(A,x_only);
disp("The observability matrix for x(t) as the output vector is: ");
```

```
The observability matrix for x(t) as the output vector is:
```

```
disp(observability_x_only);
```

```
    1.0000         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0   -0.9810         0   -0.9810
         0         0    0.6255         0    1.1067         0
         0         0         0    0.6255         0    1.1067
```

```
%Rank of Observability Matrix
x_o=rank(observability_x_only);
disp('Rank of observability matrix for x(t) as the output vector is : ');
```

```
Rank of observability matrix for x(t) as the output vector is :
```

```
disp(x_o);
```

```
6
```

```
%Checking for full rank
if x_o == 6
    disp('The selected output vector x(t) is full rank hence the system is observable.');
else
    disp('The selected output vector x(t) is not full rank hence the system is not observable.');
end
```

```
The selected output vector x(t) is full rank hence the system is observable.
```

```
%-----------------------------------------------------------------------------------------------%
%Selecting thetha1(t) and theta2(t) as the output vector
theta1_theta2=[0 0 1 0 0 0;
    0 0 0 0 1 0];

%Obtaining the observability Matrix
observabilitytheta1_theta2=obsv(A,theta1_theta2);
disp("The observability matrix thetha 1(t) and theta 2(t) as the output vector is: ");
```

```
The observability matrix thetha 1(t) and theta 2(t) as the output vector is:
```

```
disp(observabilitytheta1_theta2);
```

```
     0         0    1.0000         0         0         0
     0         0         0         0    1.0000         0
     0         0         0    1.0000         0         0
     0         0         0         0         0    1.0000
     0         0   -0.5395         0   -0.0491         0
     0         0   -0.0981         0   -1.0791         0
     0         0         0   -0.5395         0   -0.0491
     0         0         0   -0.0981         0   -1.0791
     0         0    0.2959         0    0.0794         0
     0         0    0.1588         0    1.1693         0
     0         0         0    0.2959         0    0.0794
     0         0         0    0.1588         0    1.1693
```

```
%Rank of Observability Matrix
disp('Rank of observability matrix for thetha 1(t) and theta 2(t) as the output vector is: ');
```

```
Rank of observability matrix for thetha 1(t) and theta 2(t) as the output vector is:
```

```
t1t2=rank(observabilitytheta1_theta2);
disp(t1t2);
```

```
4
```

```
%If the observability mnatrix is full rank then the selected output vector
%is observable
if t1t2 == 6
    disp('The selected output vector theta1(t) and theta2(t) is full rank hence the system is observable.');
else
    disp('The selected output is not full rank vector, so theta1(t) and theta2(t) combination is not observable.');
end
```

```
The selected output is not full rank vector, so theta1(t) and theta2(t) combination is not observable.
```

```
%-----------------------------------------------------------------------------------------------%
%Selecting x(t) and theta2(t) as the output vector
x_theta2=[1 0 0 0 0 0;
          0 0 0 0 1 0];

%Obtaining the observability Matrix
observability_x_theta2=obsv(A,x_theta2);
disp("The observability matrix x(t) and theta 2(t) as the output vector is: ")
```

```
The observability matrix x(t) and theta 2(t) as the output vector is:
```

```
disp(observability_x_theta2);
```

```
    1.0000         0         0         0         0         0
         0         0         0         0    1.0000         0
         0    1.0000         0         0         0         0
         0         0         0         0         0    1.0000
         0         0   -0.9810         0   -0.9810         0
         0         0   -0.0981         0   -1.0791         0
         0         0         0   -0.9810         0   -0.9810
         0         0         0   -0.0981         0   -1.0791
         0         0    0.6255         0    1.1067         0
         0         0    0.1588         0    1.1693         0
         0         0         0    0.6255         0    1.1067
         0         0         0    0.1588         0    1.1693
```

```
%If the observability mnatrix is full rank then the selected output vector
%is observable
disp('Rank of observability matrix for x(t) and theta 2(t) as the output vector is: ');
```

```
Rank of observability matrix for x(t) and theta 2(t) as the output vector is:
```

```
xt2=rank(observability_x_theta2);
disp(xt2);
```

```
     6
```

```
if xt2 == 6
    disp('The selected output vector x(t) and theta2(t) is full rank hence the system is observable.');
else
    disp('The selected output vector x(t) and theta2(t) is not full rank hence the system is not observable.');
end
```

```
The selected output vector x(t) and theta2(t) is full rank hence the system is observable.
```

```
%-----------------------------------------------------------------------------------------------%
%Selecting x(t) and theta1(t) and theta2(t) as the output vector
x_theta1_theta2=[1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];

%Obtaining the observability Matrix
observability_x_theta1_theta2=obsv(A,x_theta1_theta2);
disp("The observability matrix x(t), theta 1(t)  and theta 2(t) as the output vector is: ");
```

```
The observability matrix x(t), theta 1(t)  and theta 2(t) as the output vector is:
```

```
disp(observability_x_theta1_theta2);
```

```
disp(observability_x_theta1_theta2);
```

```
    1.0000         0         0         0         0         0
         0         0    1.0000         0         0         0
         0         0         0         0    1.0000         0
         0    1.0000         0         0         0         0
         0         0         0    1.0000         0         0
         0         0         0         0         0    1.0000
         0         0   -0.9810         0   -0.9810         0
         0         0   -0.5395         0   -0.0491         0
         0         0   -0.0981         0   -1.0791         0
         0         0         0   -0.9810         0   -0.9810
         0         0         0   -0.5395         0   -0.0491
         0         0         0   -0.0981         0   -1.0791
         0         0    0.6255         0    1.1067         0
         0         0    0.2959         0    0.0794         0
         0         0    0.1588         0    1.1693         0
         0         0         0    0.6255         0    1.1067
```

```
x_t1_t2=rank(observability_x_theta1_theta2);
disp('Rank of observability matrix for x(t), thetha 1(t) and theta 2(t) as the output vector is: ');
```

```
Rank of observability matrix for x(t), thetha 1(t) and theta 2(t) as the output vector is:
```

```
disp(x_t1_t2);
```

```
     6
```

```
%If the observability mnatrix is full rank then the selected output vector
%is observable
if x_t1_t2 == 6
    disp('The selected output vector x(t),theta1(t) and theta2(t) is full rank hence the system is observable.')
else
    disp('The selected output vector x(t),theta1(t) and theta2(t) is not full rank hence the system is not observable.')
end
```

```
The selected output vector x(t),theta1(t) and theta2(t) is full rank hence the system is observable.
```

In conclusion, among the selected output vectors

- Only x is chosen: The controllability matrix is a full rank matrix; thus, x is observable.
- When $(\theta_1,\ \theta_2)$ is chosen: The controllability matrix is not a full rank matrix and hence $(\theta_1,\ \theta_2)$ is not observable.
- When $(x,\ \theta_2)$ is chosen: The controllability matrix is a full rank matrix and hence $(x,\ \theta_2)$ is observable.
- When $(x, \theta_1,\ \theta_2)$ is chosen: The controllability matrix is a full rank matrix and hence $(x, \theta_1,\ \theta_2)$ is observable.

LUENBERGER OBSERVER:

The Luenberger Observer, also known as the Luenberger State Observer or just State Observer, is a state estimation technique used in control systems. It is employed when not all states of a system are directly measurable, but the goal is to estimate the entire state vector based on available measurements. The basic idea behind the Luenberger Observer is to create a mathematical model (observer) that mimics the dynamics of the actual system. This observer utilizes the available measurements to estimate the unmeasured or unobservable states of the system.
The dynamics of the Luenberger Observer is given by the following state space representation.

$$\hat{\dot{x}}(t) = A\hat{x}(t) + \mathrm{B}_k\vec{U}_k + L(\,\mathrm{y(t)} - \mathrm{C}\hat{x}(\mathrm{t}))\ \ ;\ \ \hat{x}(0) = 0$$

Where L is the observer gain matrix and
$(\,\mathrm{y(t)} - \mathrm{C}\hat{x}(\mathrm{t}))$ is the correction term.

The following is the state space representation of the closed loop system,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_e(\mathrm{t}) \end{bmatrix} = \begin{bmatrix} A + B_k k & -B_k k \\ 0 & A - LC \end{bmatrix}\begin{bmatrix} x(t) \\ x_e(t) \end{bmatrix} + \begin{bmatrix} B_D \\ B_D \end{bmatrix}\mathrm{U}_D(\mathrm{t})$$

The observer gain matrix, L, is designed to optimize the trade-off between estimation accuracy and stability. The Luenberger Observer is particularly useful in situations where not all states of a system are directly measurable, but it is crucial to have an estimate of the entire state vector for control purposes. It is commonly used in conjunction with state feedback controllers to create a full-state feedback control system.
The Luenberger Observer is not applied to the (θ_1, θ_2) output vector as the combination of output vectors (θ_1, θ_2) is not observable. The Luenberger observer is applied to the rest of the output vectors considered and its response to the initial conditions and unit step input is simulated for both the linear and the nonlinear system.

Luenberger observer for Linearized system:

A Luenberger observer is implemented to the Linearized system which is controlled by an LQR Controller.

```
clc;
clear;
close all;

% Given information
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;

%Defining system matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];

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

C=eye(6);
D=0;

%Defining the output vectors
x_only = [1 0 0 0 0 0];
x_theta2 = [1 0 0 0 0 0; 0 0 0 0 1 0];
x_theta1_theta2 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
```

```
% Cosidering the same Q and R matrices chosen before in our code
Q=[10 0 0 0 0 0;
    0 10 0 0 0 0 ;
    0 0 200 0 0 0;
    0 0 0 200 0 0;
    0 0 0 0 200 0;
    0 0 0 0 0 200];

R=0.001;

%Initial State, here we are considering 6 actual and 6 estimates
x0=[1,0,deg2rad(10),0,deg2rad(20),0,0,0,0,0,0,0];

%Define Poles
poles=[-1;-2;-3;-4;-5;-6];
```

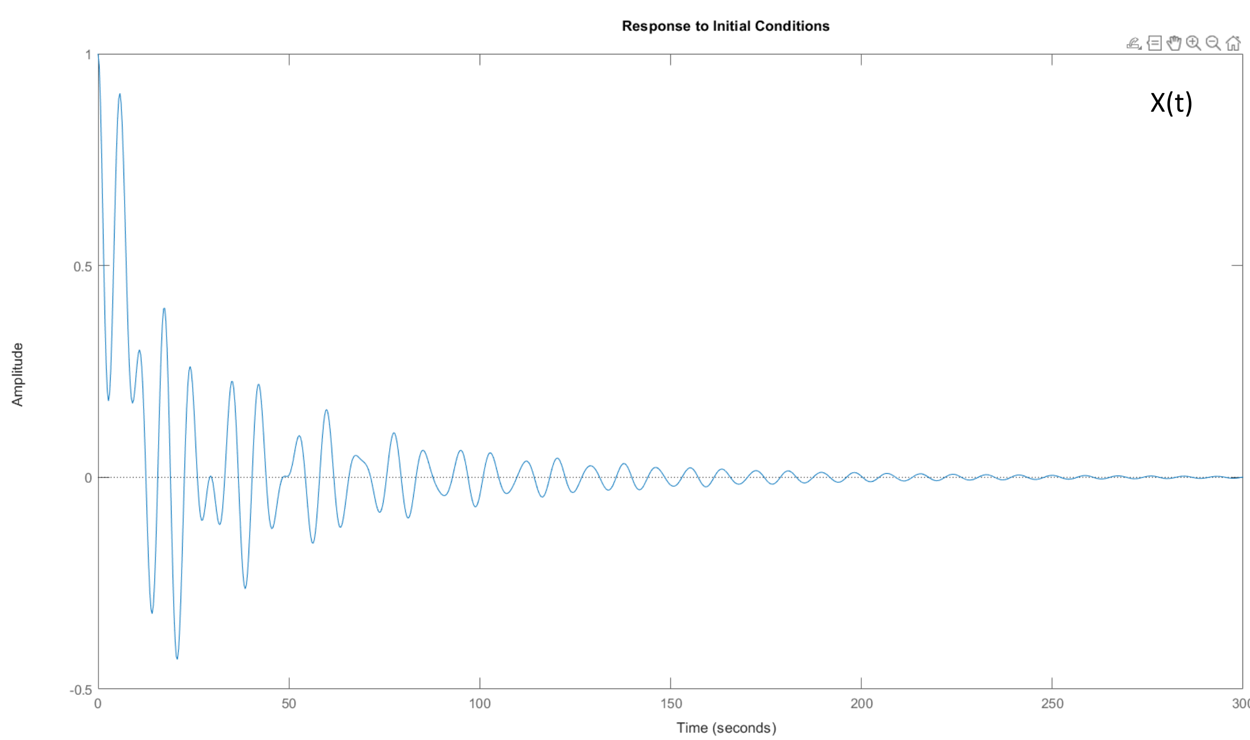
```
% Calling LQR function to obtain K matrix
[K,S,P]=lqr(A,B,Q,R);

%Placing the poles
Luenberger_x_only = place(A',x_only',poles)';

Luenberger_x_theta2 = place(A',x_theta2',poles)';
Luenberger_x_theta1_theta2 = place(A',x_theta1_theta2',poles)';

%
B_d = [B;zeros(size(B))];
```

Luenberger Observer Applied to the system where x(t) is the selected output vector

```
%System one x(t)
A_x_only = [(A-B*K) B*K; % Luenberger A matrix
        zeros(size(A)) (A-Luenberger_x_only*x_only)];

C_x_only = [x_only zeros(size(x_only))];% Luenberger C matrix

system_x_only = ss(A_x_only, B_d, C_x_only,D);
disp('Response for X(t) output vector system')
```

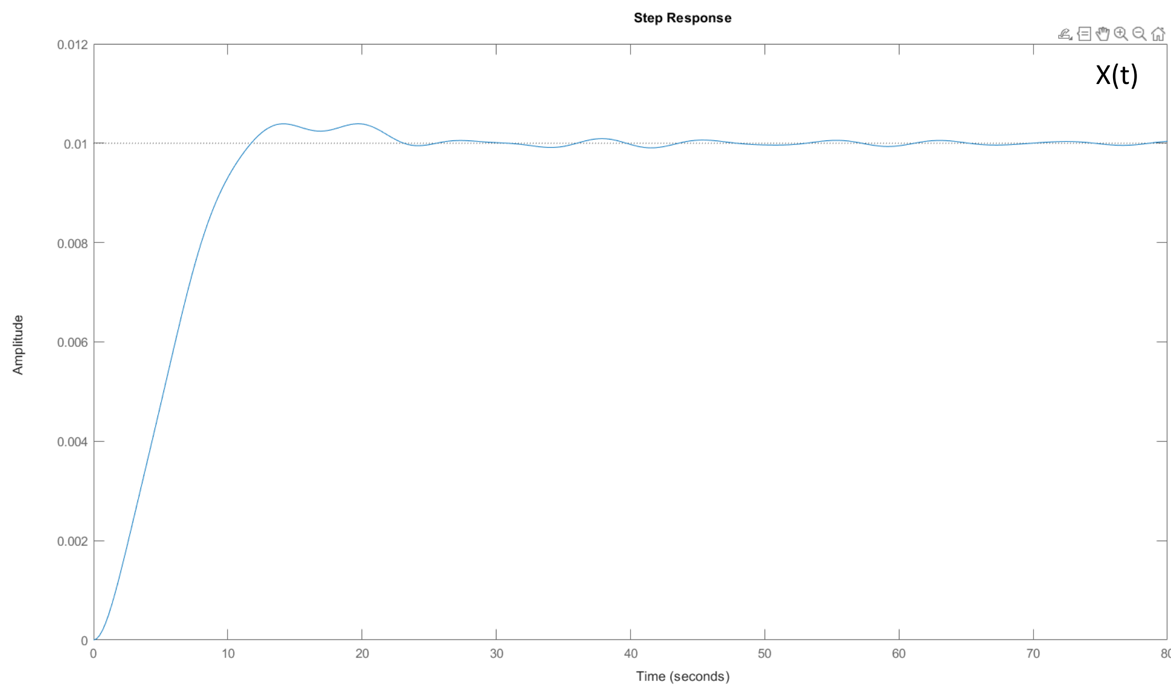
```
Response for X(t) output vector system
```

```
figure
initial(system_x_only,x0)
```



Utilising the step function in MATLAB a step input is applied to the system and the following output is obtained

```
figure
step(system_x_only)
```



Luenberger Observer Applied to the system where x(t) and θ_2(t) is the selected output vector

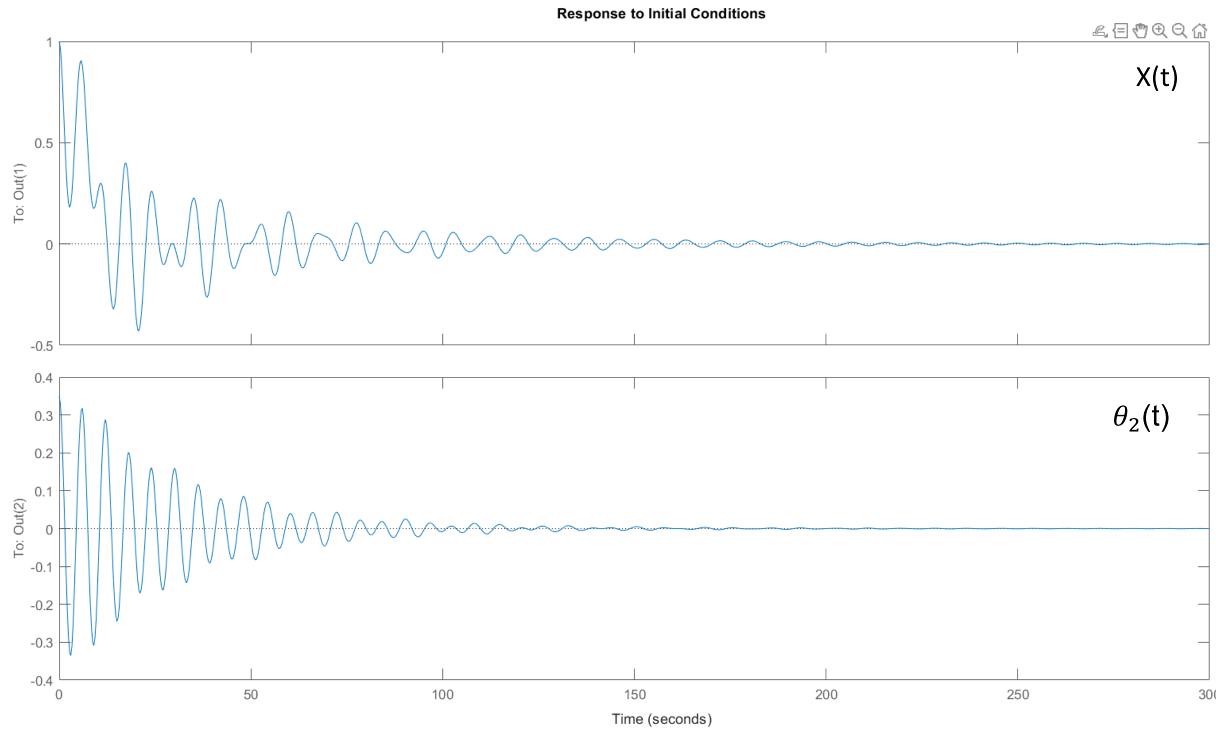
```
%System X(t) and theta2(t)
A_x_theta2=[(A-B*K) B*K;
        zeros(size(A)) (A-Luenberger_x_theta2*x_theta2)];

C_x_theta2 = [x_theta2 zeros(size(x_theta2))];% Luenberger C matrix

system_x_theta2 = ss(A_x_theta2, B_d, C_x_theta2,D);
disp('Response for X(t),theta_1(t) output vector system')
```

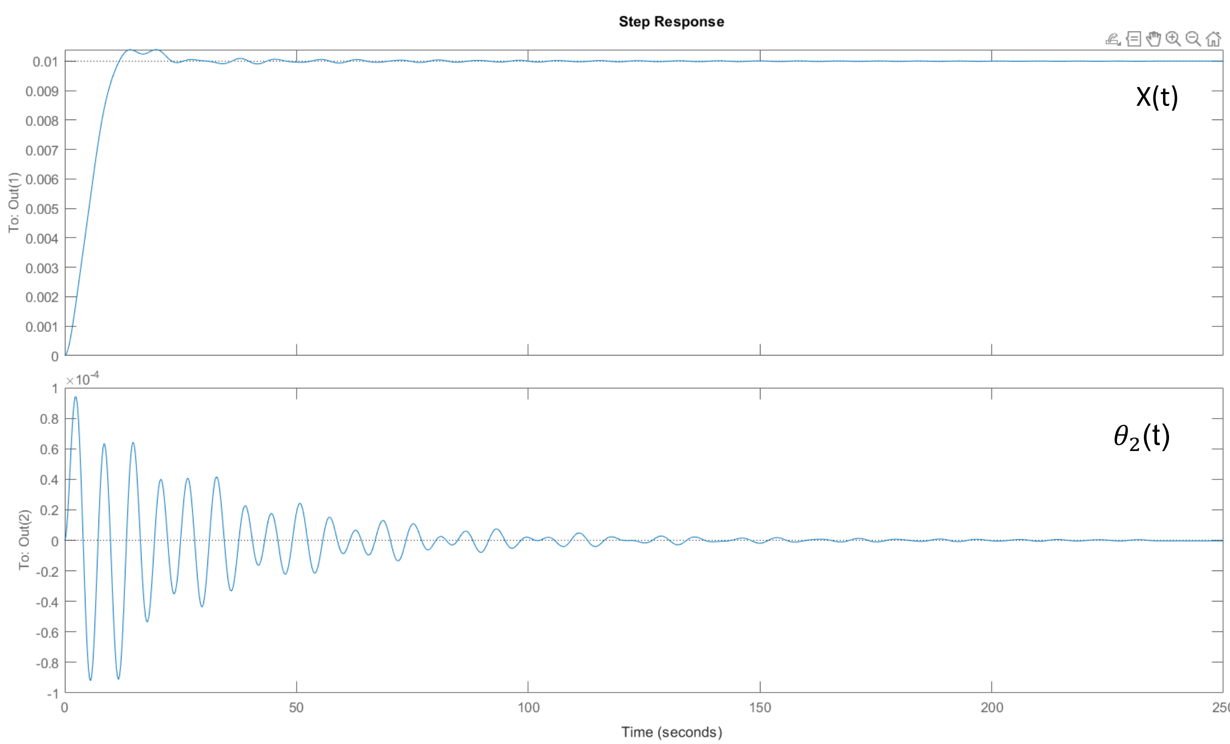
```
Response for X(t),theta_1(t) output vector system
```

```
figure
initial(system_x_theta2,x0)
```

Utilising the step function in MATLAB a step input is applied to the system and the following output is obtained:

```
figure
step(system_x_theta2)
```



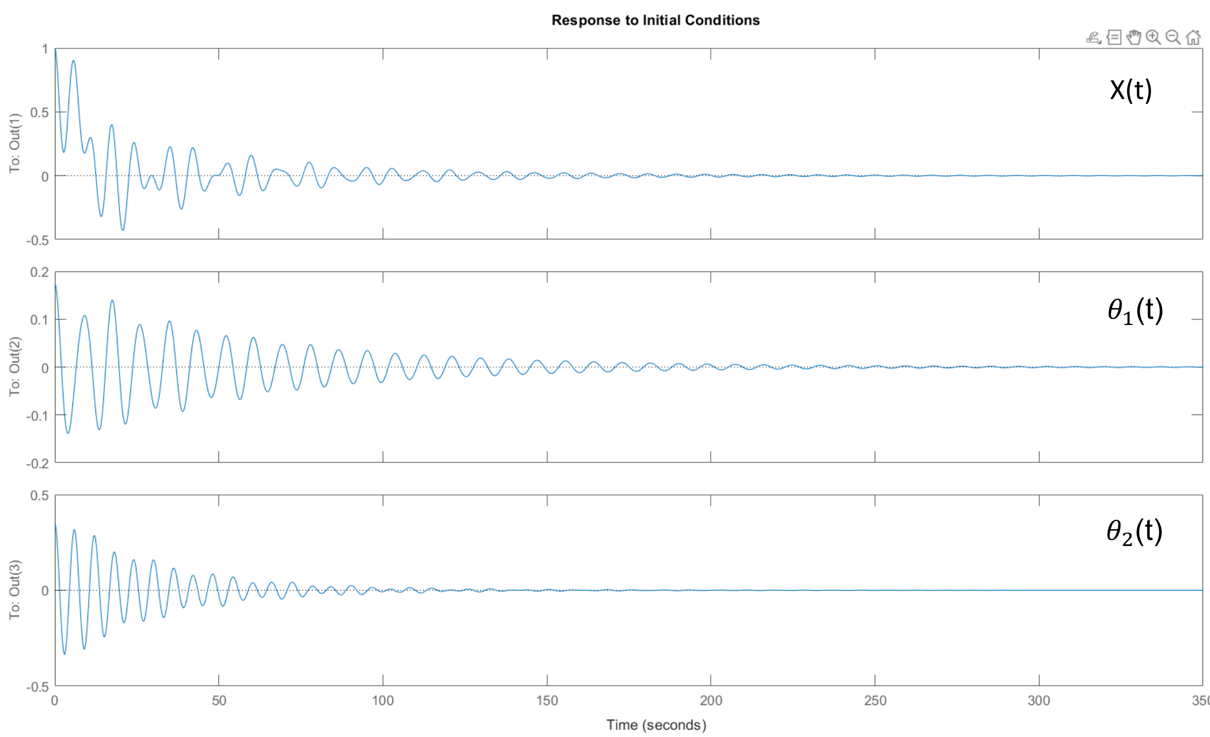


Luenberger Observer Applied to the system where x(t), θ_1(t) and θ_2(t) is the selected output vector

```
%System x(t), theta1(t) and theta2(t)
A_x_theta1_theta2 = [(A-B*K) B*K;
        zeros(size(A)) (A-Luenberger_x_theta1_theta2*x_theta1_theta2)];
B_d = [B;zeros(size(B))];
C_x_theta1_theta2 = [x_theta1_theta2 zeros(size(x_theta1_theta2))];

system_x_theta1_theta2 = ss(A_x_theta1_theta2, B_d, C_x_theta1_theta2,D);
disp('Response for X(t),theta_1(t) and theta_2(t) output vector system')
```

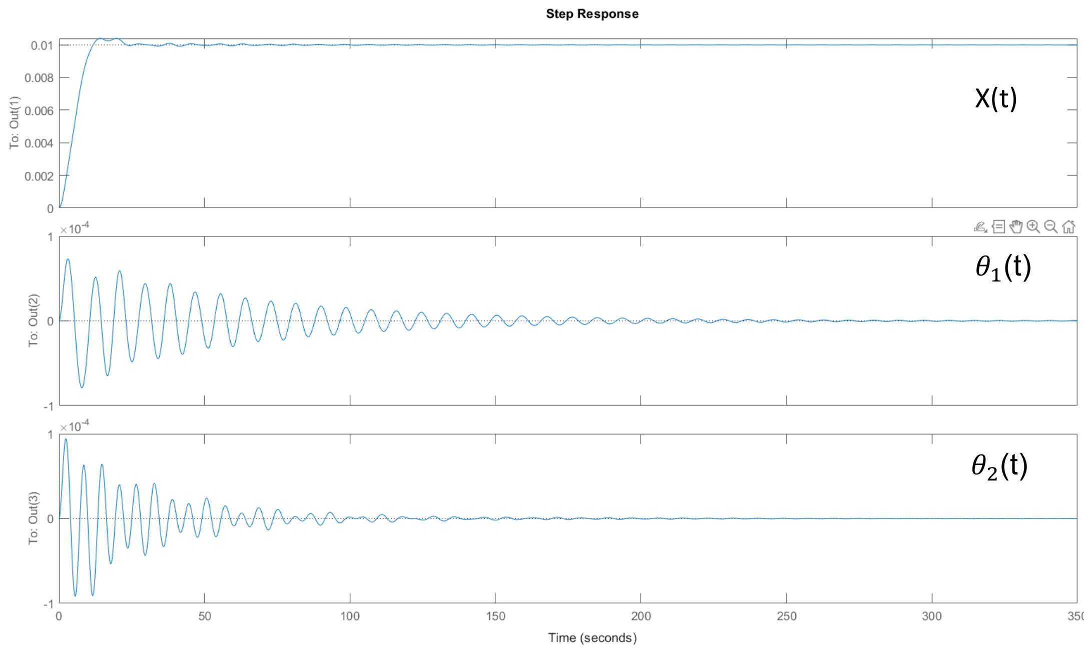
```
Response for X(t),theta_1(t) and theta_2(t) output vector system
```

```
figure
initial(system_x_theta1_theta2,x0)
```



Utilising the step function in MATLAB a step input is applied to the system and the following output is obtained:

```
step(system_x_theta1_theta2)
```


Step Response
0.01
0.008
0.006
0.004
0.002
0
To: Out(1)
X(t)
1
×10^-4
0
-1
To: Out(2)
θ1(t)
1
×10^-4
0
-1
To: Out(3)
θ2(t)
0
50
100
150
200
250
300
350
Time (seconds)

LQG CONTROLLER

Linear-quadratic-Gaussian (LQG) control is a contemporary method in the field of state-space engineering that is used to create optimal dynamic regulators. It allows you to balance regulatory performance and control effort, while considering process disruptions and measurement noise.

Considering the state space representations which is given by,

$$\dot{X}(t) = A\,X(t) + B_k U_k(t) + B_d U_d(t)$$

$$Y(t) = CX(t) + V(t)$$

Where, UD and V(t) are independent joint zero mean white gaussian processes with covariance ED and Ev.

The cost that is to be minimised is given by,

$$\lim_{n \to \infty} E[X^T(t)\; Q\; X(t)\; + U^T(t) R\; U(t)$$

The structure of the optimal solution is given by the standard output feedback configuration with the Luenberger Observer and the optimal K and L are computed separately using the LQR and Kalman-Bucy methods. This is done due to separation principle.

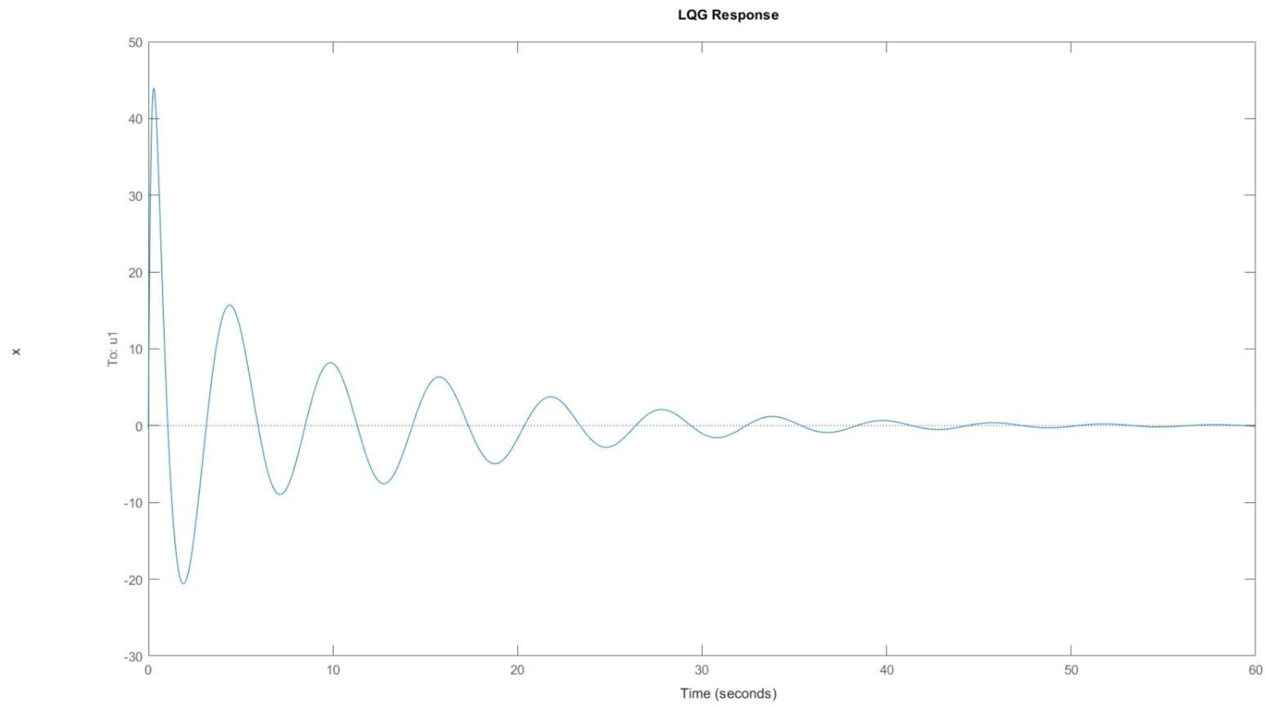
For the smallest output vector(x(t)), the output feedback controller is implemented. Using the LQG controller the resulting output feedback controller is applied to the original nonlinear system. The performance of the system is illustrated in the MATLAB simulation as follows,

LQG Controller Applied to the Linearized System

```
clc;
clear;
close all;

% Given system values
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;


% Defining the State Equation matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];

B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
C=eye(6);
D=0;

% Assigning the Q and R cost function values to check the performance of
% the controller system
Q=[10 0 0 0 0 0;
    0 10 0 0 0 0 ;
    0 0 200 0 0 0;
    0 0 0 200 0 0;
    0 0 0 0 200 0;
    0 0 0 0 0 200];

R=0.001;


%Output Vectors
x_only=[1 0 0 0 0 0];
x_theta2=[1 0 0 0 0 0;
          0 0 0 0 1 0];
x_theta1_theta2=[1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];

%Process Noise
PNC= eye(7);


%Measurement Noise
n1=wgn(6,1,5);
n2=wgn(1,1,5);
MNC=[n1;n2]*[n1' n2'];

%Defining the state state space model of the system
system1=ss(A,B,x_only,D);

%LQG Controller for the system
lqgss1=lqg(system1,PNC,MNC);

%Setting Initial Conditions
X_Initial=[1;0;deg2rad(10);0;deg2rad(20);0];
```

The LQG Controller is applied to the Linearized system and the system is simulated for the initial conditions.

```
%Setting Initial Conditions
X_Initial=[1;0;deg2rad(10);0;deg2rad(20);0];

%Plot the response of the system to the initial conditions
figure
initial(lqgss1,X_Initial)
title('LQG Response')
xlabel('Time')
ylabel('x')
```

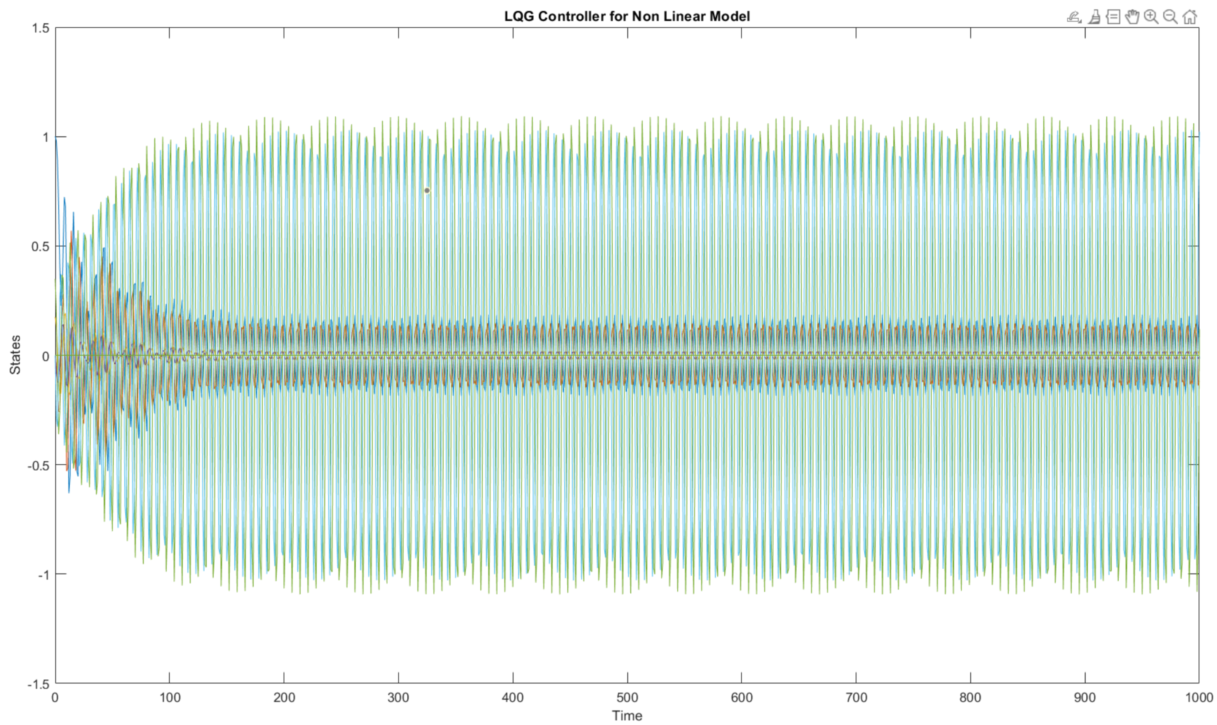


The behaviour of the linearized system under the influence of the LQG Controller is as shown above. Here we can observe that the system is nearly stabilizing at the 50th second.

LQG Controller Applied to the Non-Linear System

```
%Simulating the Non Linear System
X_Initial1=[1,0,deg2rad(10),0,deg2rad(20),0,0,0,0,0,0,0];%Setting Initial Conditions for Non linear system
tspan=0:1:1000;%Time span for simulating the non linear system
[t,y]=ode45(@lqg_system,tspan,X_Initial1);

%Plotting the response of the Non Linear System
figure
plot(t,y)
title('LQG Controller for Non Linear Model')
xlabel('Time')
ylabel('States')

function y = lqg_system(ts,Init)
    % Given system values
    M = 1000;
    m1 = 100;
    m2 = 100;
    l1 = 20;
    l2 = 10;
    g = 9.81;

    theta1=Init(3);
    theta1_dot=Init(4);
    theta2=Init(5);
    theta2_dot=Init(6);

    % Define system matrices
    A=[0 1 0 0 0 0;
        0 0 (-m1*g/M) 0 (-m2*g/M) 0;
        0 0 0 1 0 0;
        0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;


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

    B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
    %Assume Values of Q and R (cost functions)such that it stabilizes the system
    Q=[10 0 0 0 0 0;
        0 10 0 0 0 0 ;
        0 0 200 0 0 0;
        0 0 0 200 0 0;
        0 0 0 0 200 0;
        0 0 0 0 0 200];

    R=0.001;

    %Considering x(t) as the output vector
    x_only=[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];

    %Obtaining the Gain Matrix for simulating the system
    [K,~,~]=lqr(A,B,Q,R);
    F=-K*Init(1:6);

    %States of the system
    y=zeros(12,1);
    x_ddot=(F-(m1*l1*(theta1_dot)*(theta1_dot)*sin(theta1))+(m2*g*cos(theta2)*sin(theta2))-(m1*g*cos(theta1)*sin(theta1))-(m2*l2*thet
    theta1_ddot = (x_ddot*cos(theta1)-g*sin(theta1))/l1;
    theta2_ddot = (x_ddot*cos(theta2)-g*sin(theta2))/l2;

    %Defining Process noise
    PN=eye(6);

    %Gain Matrix of the system obtained through a LQR Controller
    K_gain= lqr(A',x_only',PN,1);
```

```
%State Estimates
hat = (A-K_gain*x_only)*Init(7:12);

% State Space Representation
y(1) = Init(2);
y(2) = x_ddot;
y(3) = Init(4);
y(4) = theta1_ddot;
y(5) = Init(6);
y(6) = theta2_ddot;
y(7) = hat(1);
y(8) = hat(2);
y(9) = hat(3);
y(10) = hat(4);
y(11) = hat(5);
y(12) = hat(6);

end
```



<u>How would you reconfigure your controller to asymptotically track a constant reference on x?</u>

To make our controller asymptotically track a constant reference on x, we must use the concept of Optimal reference tracking. The cost function that we want to minimize changes to:

$$\int_0^\infty (x(t)-x_d)^T Q(x(t)-x_d) + (U_k(t)-U_\infty)^T R(U_k(t)-U_\infty) dt$$

If there is U_∞ such that $Ax_d + B_k U_\infty = 0$, then the optimal solution is:

$U(t) = K(X(t) - X_d) + U_\infty$

Where $K = -R^{-1}B^T P$ and P is the positive definite solution for the following Ricatti equation: $A^T P + PA - PBR^{-1}B^T P = -Q$

We can achieve this by rewriting our state equations by considering the following:

$\tilde{x}(t) = x(t) - x_d$ and $\widetilde{U_K}(t) = U_K(t) - U_\infty$ to get:

$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B_K\widetilde{U_K}(t) + B_D U_D(t)$

We need to apply a standard LQR to the representation above in order to track a constant reference on x

Will your design reject constant force disturbances applied on the cart?

Our design will not reject the constant force disturbances applied on the card. To achieve this, we need to modify our design by augmenting an integral term to the state. The is done as follows:

$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B_k\widetilde{U_K}(t) + B_D U_D(t)$ changes to:

$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \overline{B_k}\widetilde{U_K}(t) + \overline{B_D}U_D(t)$

Where,

$$\bar{x}(t) = \begin{bmatrix} \tilde{x}(t) \\ x_I(t) \end{bmatrix},\ x_I(t) = \int_0^t \tilde{x}(\tau)\, d\tau\ ,\ \bar{A} = \begin{bmatrix} A & 0 \\ I & 0 \end{bmatrix},\ \overline{B_k} = \begin{bmatrix} B_k \\ 0 \end{bmatrix},\ \overline{B_D} = \begin{bmatrix} B_D \\ 0 \end{bmatrix}$$

Conclusions

The project's main goal of designing a LQR and a LQG controller for the nonlinear system was achieved. The system was initially linearized to apply techniques like stability tests, controllability tests on the system. The designed controllers were employed both to the linearized system and the original nonlinear system.

Simulation Videos

PART C – https://drive.google.com/file/d/1hsZARUMFBvbZF0ECsAzE5C62pVbJv5B4/view?usp=sharing

PART D – https://drive.google.com/file/d/1Ld1NhmEfJ4bYEMrhK2h-WmO02KJ98LqD/view?usp=sharing

PART E – https://drive.google.com/file/d/1dfjUvvII00JIvkkNQOukoPlH4RHtr3Jm/view?usp=sharing\

PART F – https://drive.google.com/file/d/1s3PCH092P8_-ILEjECFpj1bUgyRrxu_1/view?usp=sharing

PART G – https://drive.google.com/file/d/1JvdHsWAPGkET78fUWSOakoAelBPbyxr1/view?usp=sharing

ABSTRACT

Codes used for Simulations on MATLAB

C_Controllability.mlx file

```
% Checking Rank of Grammian of Controllability for the system
clc
clear


%Defining the required symbolic variables to define the system in MATLAB
syms M m1 m2 g l1 l2 F


% A and B Matrix as obtained from the state space equation
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 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];
disp("The A Matrix is : ");
disp(A);
B=[0;1/M;0;1/(M*l1);0;1/(M*l2)];
disp("The B Matrix is : ");
disp(B);


%Computing the controllability matrix.
ABk=A*B;
A2Bk=A*A*B;
A3Bk=A*A*A*B;
A4Bk=A*A*A*A*B;
A5Bk=A*A*A*A*A*B;


%Defining the controllability matrix
Controllability_Matrix=[B ABk A2Bk A3Bk A4Bk A5Bk];
disp("The Controllability Matrix is : ")
disp(Controllability_Matrix);


%Checking rank of controllability matrix
n=rank(Controllability_Matrix);
disp("The rank of Controllability Matrix is : ")
```

```
disp(rank(Controllability_Matrix));


%Computing the determinant of the controllability matrix
disp("The Determinant of the controllability matrix is : ")
disp(simplify(det(Controllability_Matrix)));
```

```
%To obtain conditions on M,m1,m2, l1, l2 for which the linearized system is
controllable


% Substituting l2 to l1 to obtain the controllability matrix when the l1=l2
% to check controllability
Controllability_Matrixl1l2=subs(Controllability_Matrix, l1, l2);
disp("The Controllability Matrix when l1 = l2 is : ")
disp(Controllability_Matrixl1l2);
disp("Rank of Controllability Matrix for l1 = l2 condition is : ")
disp(rank(Controllability_Matrixl1l2));
disp("The Determinant of the Controllability Matrix for l1 = l2 condition is : ")
disp(det(Controllability_Matrixl1l2));


if (det(Controllability_Matrixl1l2))==0
    disp("The controllability matrix is not full rank and also the determinant is
equal to zero ");
    disp("Thus for the system to be controllable the lengths of strings
suspending the two masses must to be unequal with respect to each other");
else
    disp("The controllability matrix is full rank, so the system is controllable
");
    disp("The system is controllable when l1 =l2");
end
```

LQR_Linear.mlx file

```
%LQR controller for the linearised system
clc;
clear;
close all;
% Given system values
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
```

```
l2 = 10;
g = 9.81;


% Defining the State Equation matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];


B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
C=eye(6);
D=0;


%Determining the controllability of the system using inbuilt MATLAB
controllability checker
disp(" Rank of Controllability Matrix is :");
```

```
 Rank of Controllability Matrix is :
```

```
disp(rank(ctrb(A,B)));
```

```
     6
```

```
if rank(ctrb(A,B)) == 6
    disp("The system is controllable")
else
    disp("The system is not Controllable")
end
```

```
The system is controllable
```

```
% Initial conditions for simulating the response of the system
X_Initial=[1;0;deg2rad(10);0;deg2rad(20);0];
%Assume Values of Q and R such that it stabilizes the system


% Assigning the Q and R cost function values to check the performance of
% the controller system
Q=[10 0 0 0 0 0;
    0 10 0 0 0 0 ;
    0 0 200 0 0 0;
    0 0 0 200 0 0;
    0 0 0 0 200 0;
    0 0 0 0 0 200];


R=0.001;
```

```
%Defining the state space representation of the system
system=ss(A,B,C,D);
disp("Inititial response without Controller");
```

```
Inititial response without Controller
```

```
%Plot the output of the initial response of the system
figure
initial(system, X_Initial);
grid on
```

LQR_NonLinear.mlx file

```
clc;
clear;
close all;

%Setting Initial Conditions
y0=[1;0;deg2rad(10);0;deg2rad(20);0];

%Defining time span and sampling rate for the graph generation
timespan=0:0.1:800;

%Utilizing ODE45 for defining the differential Equation
[t1,y1]= ode45(@twosuspendedmass,timespan,y0);

%Plotting all graphs from the obtained values
disp("The graph for LQR Applied to the Non Linear System")
```

```
The graph for LQR Applied to the Non Linear System
```

```
plot(t1,y1)
xlabel('Time in seconds');
ylabel('Amplitude')
grid on
```

E_Observability.mlx file

```
clc;
clear;


% Given information
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;


% Define system matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];




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


%--------------------------------------------------------------------------------
-----------------%
%Selecting x(t) as the output vector
x_only=[1 0 0 0 0 0];
%Obtaining the observability Matrix
observability_x_only=obsv(A,x_only);
disp("The observability matrix for x(t) as the output vector is: ");
The observability matrix for x(t) as the output vector is:
disp(observability_x_only);
    1.0000         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0   -0.9810         0   -0.9810         0
         0         0         0   -0.9810         0   -0.9810
         0         0    0.6255         0    1.1067         0
         0         0         0    0.6255         0    1.1067
%Rank of Observability Matrix
x_o=rank(observability_x_only);
disp('Rank of observability matrix for x(t) as the output vector is : ');
```

```
Rank of observability matrix for x(t) as the output vector is :
disp(x_o);
     6
%Checking for full rank
if x_o == 6
    disp('The selected output vector x(t) is full rank hence the system is observable.');
else
    disp('The selected output vector x(t) is not full rank hence the system is not observable.');
end
The selected output vector x(t) is full rank hence the system is observable.


%------------------------------------------------------------------------------------------%
%Selecting thetha1(t) and theta2(t) as the output vector
theta1_theta2=[0 0 1 0 0 0;
    0 0 0 0 1 0];


%Obtaining the observability Matrix
observabilitytheta1_theta2=obsv(A,theta1_theta2);
disp("The observability matrix thetha 1(t) and theta 2(t) as the output vector is: ");
The observability matrix thetha 1(t) and theta 2(t) as the output vector is:
disp(observabilitytheta1_theta2);
         0         0    1.0000         0         0         0
         0         0         0         0    1.0000         0
         0         0         0    1.0000         0         0
         0         0         0         0         0    1.0000
         0         0   -0.5395         0   -0.0491         0
         0         0   -0.0981         0   -1.0791         0
         0         0         0   -0.5395         0   -0.0491
         0         0         0   -0.0981         0   -1.0791
         0         0    0.2959         0    0.0794         0
         0         0    0.1588         0    1.1693         0
         0         0         0    0.2959         0    0.0794
         0         0         0    0.1588         0    1.1693


%Rank of Observability Matrix
disp('Rank of observability matrix for thetha 1(t) and theta 2(t) as the output vector is: ');
Rank of observability matrix for thetha 1(t) and theta 2(t) as the output vector is:
t1t2=rank(observabilitytheta1_theta2);
disp(t1t2);
     4
%If the observability mnatrix is full rank then the selected output vector
```

```
%is observable
if t1t2 == 6
    disp('The selected output vector theta1(t) and theta2(t) is full rank hence
the system is observable.');
else
    disp('The selected output is not full rank vector, so theta1(t) and theta2(t)
combination is not observable.');
end
```

```
The selected output is not full rank vector, so theta1(t) and theta2(t) combination is not
observable.
```

```
%-----------------------------------------------------------------------------------
-----------------%
%Selecting x(t) and theta2(t) as the output vector
x_theta2=[1 0 0 0 0 0;
          0 0 0 0 1 0];


%Obtaining the observability Matrix
observability_x_theta2=obsv(A,x_theta2);
disp("The observability matrix x(t) and theta 2(t) as the output vector is: ")
```

```
The observability matrix x(t) and theta 2(t) as the output vector is:
```

```
disp(observability_x_theta2);
```

```
    1.0000         0         0         0         0         0
         0         0         0         0    1.0000         0
         0    1.0000         0         0         0         0
         0         0         0         0         0    1.0000
         0         0   -0.9810         0   -0.9810         0
         0         0   -0.0981         0   -1.0791         0
         0         0         0   -0.9810         0   -0.9810
         0         0         0   -0.0981         0   -1.0791
         0         0    0.6255         0    1.1067         0
         0         0    0.1588         0    1.1693         0
         0         0         0    0.6255         0    1.1067
         0         0         0    0.1588         0    1.1693
```

```
%If the observability mnatrix is full rank then the selected output vector
%is observable
disp('Rank of observability matrix for x(t) and theta 2(t) as the output vector
is: ');
```

```
Rank of observability matrix for x(t) and theta 2(t) as the output vector is:
```

```
xt2=rank(observability_x_theta2);
disp(xt2);
```

```
     6
```

```
if xt2 == 6
    disp('The selected output vector x(t) and theta2(t) is full rank hence the
system is observable.');
```

```
else
    disp('The selected output vector x(t) and theta2(t) is not full rank hence
the system is not observable.');
end
```

```
The selected output vector x(t) and theta2(t) is full rank hence the system is observable.
```

```
%-----------------------------------------------------------------------------------
-----------------%
%Selecting x(t) and theta1(t) and theta2(t) as the output vector
x_theta1_theta2=[1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];


%Obtaining the observability Matrix
observability_x_theta1_theta2=obsv(A,x_theta1_theta2);
disp("The observability matrix x(t), theta 1(t)  and theta 2(t) as the output
vector is: ");
```

```
The observability matrix x(t), theta 1(t)  and theta 2(t) as the output vector is:
```

```
disp(observability_x_theta1_theta2);
```

```
    1.0000         0         0         0         0         0
         0         0    1.0000         0         0         0
         0         0         0         0    1.0000         0
         0    1.0000         0         0         0         0
         0         0         0    1.0000         0         0
         0         0         0         0         0    1.0000
         0         0   -0.9810         0   -0.9810         0
         0         0   -0.5395         0   -0.0491         0
         0         0   -0.0981         0   -1.0791         0
         0         0         0   -0.9810         0   -0.9810
         0         0         0   -0.5395         0   -0.0491
         0         0         0   -0.0981         0   -1.0791
         0         0    0.6255         0    1.1067         0
         0         0    0.2959         0    0.0794         0
         0         0    0.1588         0    1.1693         0
         0         0         0    0.6255         0    1.1067
         0         0         0    0.2959         0    0.0794
         0         0         0    0.1588         0    1.1693
```

```
x_t1_t2=rank(observability_x_theta1_theta2);
disp('Rank of observability matrix for x(t), thetha 1(t) and theta 2(t) as the
output vector is: ');
```

```
Rank of observability matrix for x(t), thetha 1(t) and theta 2(t) as the output vector is:
```

```
disp(x_t1_t2);
```

```
     6
```

```
%If the observability mnatrix is full rank then the selected output vector
%is observable
if x_t1_t2 == 6
    disp('The selected output vector x(t),theta1(t) and theta2(t) is full rank 
hence the system is observable.')
else
    disp('The selected output vector x(t),theta1(t) and theta2(t) is not full 
rank hence the system is not observable.')
end
```

```
The selected output vector x(t),theta1(t) and theta2(t) is full rank hence the system is 
observable.
```

LQR_Luenberger_Linear.mlx

```
clc;
clear;
close all;


% Given information
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;


%Defining system matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];


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


C=eye(6);
D=0;
```

```
%Defining the output vectors
x_only = [1 0 0 0 0 0];
x_theta2 = [1 0 0 0 0 0; 0 0 0 0 1 0];
x_theta1_theta2 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];


% Cosidering the same Q and R matrices chosen before in our code
Q=[10 0 0 0 0 0;
    0 10 0 0 0 0 ;
    0 0 200 0 0 0;
    0 0 0 200 0 0;
    0 0 0 0 200 0;
    0 0 0 0 0 200];


R=0.001;


%Initial State, here we are considering 6 actual and 6 estimates
x0=[1,0,deg2rad(10),0,deg2rad(20),0,0,0,0,0,0,0];


%Define Poles
poles=[-1;-2;-3;-4;-5;-6];


% Calling LQR function to obtain K matrix
[K,S,P]=lqr(A,B,Q,R);


%Placing the poles
Luenberger_x_only = place(A',x_only',poles)';
Luenberger_x_theta2 = place(A',x_theta2',poles)';
Luenberger_x_theta1_theta2 = place(A',x_theta1_theta2',poles)';


%
B_d = [B;zeros(size(B))];


%System one x(t)
A_x_only = [(A-B*K) B*K; zeros(size(A-B*K)) (A-Luenberger_x_only*x_only)];


C_x_only = [x_only zeros(size(x_only))];% Luenberger C matrix


system_x_only = ss(A_x_only, B_d, C_x_only,D);
disp('Response for X(t) output vector system')
```

Response for X(t) output vector system

```
figure
initial(system_x_only,x0)
```

```
figure
step(system_x_only)
```

```
%System X(t) and theta2(t)
A_x_theta2=[(A-B*K) B*K;
        zeros(size(A)) (A-Luenberger_x_theta2*x_theta2)];


C_x_theta2 = [x_theta2 zeros(size(x_theta2))];% Luenberger C matrix


system_x_theta2 = ss(A_x_theta2, B_d, C_x_theta2,D);
disp('Response for X(t),theta_1(t) output vector system')
```

Response for X(t),theta_1(t) output vector system

```
figure
initial(system_x_theta2,x0)
```

```
figure
step(system_x_theta2)
```

```
%System x(t), theta1(t) and theta2(t)yyyyyyyyyy
A_x_theta1_theta2 = [(A-B*K) B*K;
        zeros(size(A)) (A-Luenberger_x_theta1_theta2*x_theta1_theta2)];
B_d = [B;zeros(size(B))];
C_x_theta1_theta2 = [x_theta1_theta2 zeros(size(x_theta1_theta2))];


system_x_theta1_theta2 = ss(A_x_theta1_theta2, B_d, C_x_theta1_theta2,D);
disp('Response for X(t),theta_1(t) and theta_2(t) output vector system')
```

Response for X(t),theta_1(t) and theta_2(t) output vector system

```
figure
initial(system_x_theta1_theta2,x0)
```

```
step(system_x_theta1_theta2)
```

```
%Definiing the time span for simulating the system with initial conditions
t_span = 0:0.01:600;
y0 = [1; 0; deg2rad(10); 0; deg2rad(20); 0];


%Function to simulate the Non Linear System
[ts, x_dot] = ode45(@non_lin_sys, t_span, y0);


%Plotting the Results of the system
figure
subplot(3, 2, 1)
plot(ts, x_dot(:, 1))
title('x')
xlabel('Time in seconds')
ylabel('x')
grid on


subplot(3, 2, 2)
plot(ts, x_dot(:, 2))
title('x\_dot')
xlabel('Time in seconds')
ylabel('x\_dot')
grid on


subplot(3, 2, 3)
plot(ts, x_dot(:, 3))
title('theta1')
xlabel('Time in seconds')
ylabel('theta1')
grid on


subplot(3, 2, 4)
plot(ts, x_dot(:, 4))
title('theta1\_dot')
xlabel('Time in seconds')
ylabel('theta1\_dot')
grid on


subplot(3, 2, 5)
plot(ts, x_dot(:, 5))
title('theta2')
xlabel('Time in seconds')
```

```
ylabel('theta2')
grid on


subplot(3, 2, 6)
plot(ts, x_dot(:, 6))
title('(theta2)\_dot')
xlabel('Time in seconds')
ylabel('(theta2)\_dot')
grid on
```

```
function ydot = non_lin_sys(~, x)



    % Declaring the values of system variables
    M = 1000;
    m1 = 100;
    m2 = 100;
    l1 = 20;
    l2 = 10;
    g = 9.81;

    % Declaring the A matrix of the system
    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 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
        0 0 0 0 0 1;
        0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];

    % Declaring the B matrix of the system
    B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];

    C=[1 0 0 0 0 0 ];
    % Defining the Q and R matrices for LQR controller
    Q=[10 0 0 0 0 0;
        0 10 0 0 0 0 ;
        0 0 200 0 0 0;
        0 0 0 200 0 0;
        0 0 0 0 200 0;
        0 0 0 0 0 200];

    R=0.001;
```

```
    K = lqr(A,B,Q,R);
    f = -K*x;

    poles=[-1;-2;-3;-4;-5;-6];
    %Luenberger Observer Gains


    %Observer State Estimates
    L=place(A',C', poles)';


    ydot = zeros(6,1);
    % State x_dot_1 is x_2
    ydot(1) = x(2);
    % State x_dot_2 is
    ydot(2) = (f - (g/2)*(m1*sin(2*x(3)) + m2*sin(2*x(5))) -
(m1*l1*(x(4)^2)*sin(x(3))) - (m2*l2*(x(6)^2)*sin(x(5)))) /
(M+m1*((sin(x(3)))^2)+m2*((sin(x(5)))^2));
    % State x_dot_3 is theta 1
    ydot(3) = x(4);
    % State x_dot_4 is theta1 dot
    ydot(4) = (ydot(2)*cos(x(3)) - g*(sin(x(3))))/l1;
    % State x_dot_5 is theta2
    ydot(5) = x(6);
    % State x_dot_6 is theta2 dot
    ydot(6) = (ydot(2)*cos(x(5)) - g*(sin(x(5))))/l2;


end
```

LQG_Linear_Non_Linear.mlx file

```
clc;
clear;
close all;


% Given system values
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;
```

```
% Defining the State Equation matrices
A=[0 1 0 0 0 0;
    0 0 (-m1*g/M) 0 (-m2*g/M) 0;
    0 0 0 1 0 0;
    0 0 ((-g*(M+m1))/(l1*M)) 0 ((-g*m2)/(l1*M)) 0;
    0 0 0 0 0 1;
    0 0 ((-g*m1)/(l2*M)) 0 ((-g*(M+m2))/(l2*M)) 0];


B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
C=eye(6);
D=0;


% Assigning the Q and R cost function values to check the performance of
% the controller system
Q=[10 0 0 0 0 0;
    0 10 0 0 0 0 ;
    0 0 200 0 0 0;
    0 0 0 200 0 0;
    0 0 0 0 200 0;
    0 0 0 0 0 200];


R=0.001;


%Output Vectors
x_only=[1 0 0 0 0 0];
x_theta2=[1 0 0 0 0 0;
          0 0 0 0 1 0];
x_theta1_theta2=[1 0 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 0 1 0];


%Process Noise
PNC= eye(7);


%Measurement Noise
n1=wgn(6,1,5);
n2=wgn(1,1,5);
MNC=[n1;n2]*[n1' n2'];


%Defining the state state space model of the system
```

```
system1=ss(A,B,x_only,D);


%LQG Controller for the system
lqgss1=lqg(system1,PNC,MNC);


%Setting Initial Conditions
X_Initial=[1;0;deg2rad(10);0;deg2rad(20);0];


%Plot the response of the system to the initial conditions
figure
initial(lqgss1,X_Initial)
title('LQG Response')
xlabel('Time')
ylabel('x')
```

```


%Simulating the Non Linear System
X_Initial1=[1,0,deg2rad(10),0,deg2rad(20),0,0,0,0,0,0,0];%Setting Initial
Conditions for Non linear system
tspan=0:1:1000;%Time span for simulating the non linear system
[t,y]=ode45(@lqg_system,tspan,X_Initial1);


%Plotting the response of the Non Linear System
figure
plot(t,y)
title('LQG Controller for Non Linear Model')
xlabel('Time')
ylabel('States')
```

```
function y = lqg_system(ts,Init)
    % Given system values
    M = 1000;
    m1 = 100;
    m2 = 100;
    l1 = 20;
    l2 = 10;
    g = 9.81;


    theta1=Init(3);
    theta1_dot=Init(4);
    theta2=Init(5);
    theta2_dot=Init(6);
```

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

    B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
    %Assume Values of Q and R (cost functions)such that it stabilizes the system
    Q=[10 0 0 0 0 0;
        0 10 0 0 0 0 ;
        0 0 200 0 0 0;
        0 0 0 200 0 0;
        0 0 0 0 200 0;
        0 0 0 0 0 200];

    R=0.001;

    %Considering x(t) as the output vector
    x_only=[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];

    %Obtaining the Gain Matrix for simulating the system
    [K,~,~]=lqr(A,B,Q,R);
    F=-K*Init(1:6);

    %States of the system
    y=zeros(12,1);
    x_ddot=(F-
(m1*l1*(theta1_dot)*(theta1_dot)*sin(theta1))+(m2*g*cos(theta2)*sin(theta2))-
(m1*g*cos(theta1)*sin(theta1))-
(m2*l2*theta2_dot*theta2_dot*sin(theta2)))/(M+m1+m2-(m1*cos(theta1)*cos(theta1))-
(m1*cos(theta1)*cos(theta1)));
    theta1_ddot = (x_ddot*cos(theta1)-g*sin(theta1))/l1;
    theta2_ddot = (x_ddot*cos(theta2)-g*sin(theta2))/l2;

    %Defining Process noise
    PN=eye(6);

    %Gain Matrix of the system obtained through a LQR Controller
    K_gain= lqr(A',x_only',PN,1);


    %State Estimates
    hat = (A-K_gain*x_only)*Init(7:12);
```

```
    % State Space Representation
    y(1) = Init(2);
    y(2) = x_ddot;
    y(3) = Init(4);
    y(4) = theta1_ddot;
    y(5) = Init(6);
    y(6) = theta2_ddot;
    y(7) = hat(1);
    y(8) = hat(2);
    y(9) = hat(3);
    y(10) = hat(4);
    y(11) = hat(5);
    y(12) = hat(6);


end
```