Control of Robotics Systems (ENPM667) Final-Project Report

Authors: Karan Sutradhar (117037672)

Ajinkya Parwekar (117030389)

Date: 12/19/2020

Abstract:

The focus of this project is to understand and implement the core concepts of controls, including State Space Representation, Nonlinear System Design, Linear Quadratic Regulator (LQR) Controller, Linear Quadratic Tracker (LQT) Controller and Luenberger Observer for Double Pendulum on a cart. We used concepts like Controllability, Observability, for systems to develop a robust controller. For the scope, simulations, and validation of this project we will use Mat-Lab and Simulink to model our system.

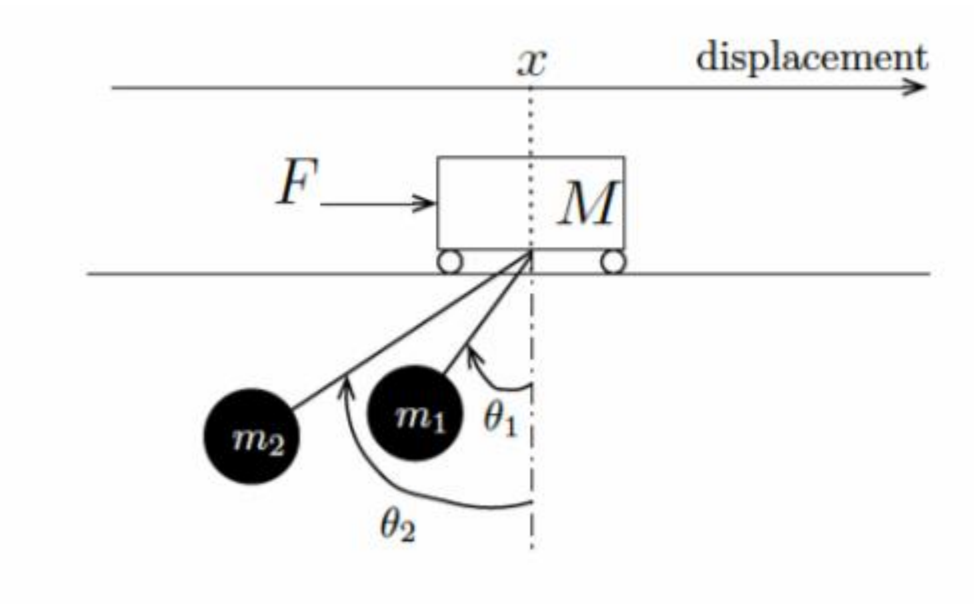
Project Goals:

- Understand and implement the core concepts of controls, including State Space Representation, Nonlinear System Design, Linear Quadratic Regulator (LQR) Controller, Linear Quadratic Tracker (LQT) Controller and Luenberger Observer for Double Pendulum on a cart.
- Usage of concepts like Controllability, Observability, for systems to develop a robust controller.
- To simulate and validate the scope of this project we will use Mat-Lab and Simulink to model our system.

Table of Contents

Problem Statement:

For this project we have considered a friction-less crane of 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 m1 and m2, and the lengths of the cables are l1 and l2, respectively. The following figure depicts the crane and associated variables used throughout this project.



Part A: Equations of motion for Nonlinear State Representation:

To obtain the dynamics of the system given in the problem statement, we have to find out the linear velocity, linear acceleration of the crane along with, angular velocity and angular acceleration of the masses of the pendulum m1 and m2 which are the states of the system.

From the fig 2.0, we are considering (X,Y) as the origin of the reference frame in the system, and then we model the system with the same consideration.

We will use Eurler-Lagrange equation to formulate the motions equations and use it to fabricate the non-linear state space representation. To compute the Eurler-Lagrange equation we need to calculate the kinetic and potential energy of the system.

First we find out the kinetic energy of individual masses, then we add them.

Position of m1 w.r.t the reference frame is given by

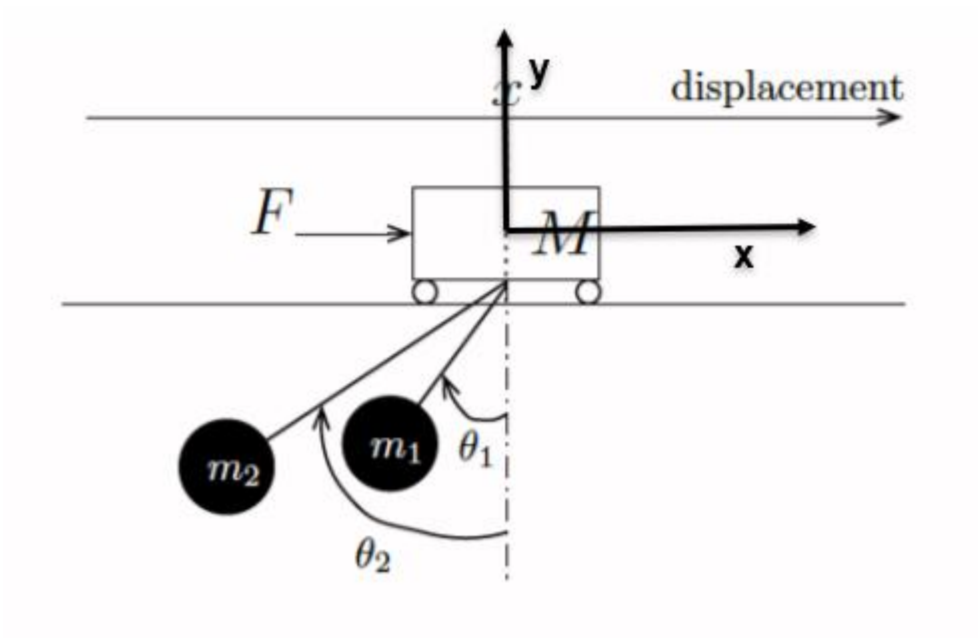
$$x_1 = (x - (l_1)\sin(\theta_1))X + (l_1)\cos(\theta_1)\,Y \quad (1)$$

The position of m2 w.r.t the reference frame is given by

$$x_2 = (x - (l_2)\sin(\theta_2))X + (l_2)\cos(\theta_2)\,Y \tag{2}$$

Differentiating, we get

$$X_1\dot{(t)} = \dot{x} - l_1 C_1 \dot{\theta}_1 + l_1 S_1 \dot{\theta}_1$$
$$\dot{X}_2(t) = \dot{x} - l_2 C_2 \dot{\theta}_2 + \ l_2 S_2 \dot{\theta}_2$$



The total energy of the system is given from the Kinetic energy equation

$$K = 1/2[M\left(\frac{dx}{dt}\right)^2] \tag{3}$$

From equation (3), we have the equation of the Kinetic energy for the individual masses of the system.
Let us first find the kinetic energy of mass1

$$K_1 = 1/2[M\left(m_1\dot{x} - m_1 l_1 \dot{\theta}_1 \cos(\theta_1)^2 + \left(-m_1 l_1 \dot{\theta}_1 \sin(\theta_1)^2\right)\right)]$$

$$K_1 = 1/2[M\left(m_1 \dot{x^2} + m_1 {l_1}^2 \dot{\theta}_1^2 cos^2(\theta_1) - 2m_1 \dot{x} l_1 \dot{\theta}_1 cos\theta_1 + m_1 {l_1}^2 \dot{\theta}_1^2 sin^2(\theta_1)\right)]$$

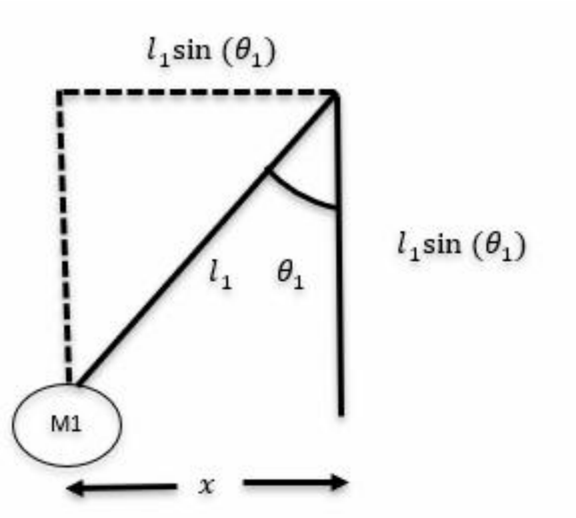
Solving the equation, we get

$$K_1 = 1/2[M\left(m_1 \dot{x^2} + m_1 {l_1}^2 \dot{\theta}_1^2 - 2m_1 \dot{x} l_1 \dot{\theta}_1 cos\theta_1\right)] \tag{4}$$

Let us find the kinetic energy of mass2 using the equation (4),
We will get the kinetic energy of mass2 as similarly as we got the kinetic energy for mass1

$$K_2 = 1/2[M\left(m_2\dot{x}^2 + m_2{l_2}^2\dot{\theta}_2^2 - 2m_2\dot{x}l_2\dot{\theta}_2 cos\theta_2\right)] \tag{5}$$

Now, we calculate the potential energy for the system:



$$P_1 = m_1gl_1 - m_1gl_1cos\theta_1$$

$$P_1 = m_1gl_1(1 - cos\theta_1) \tag{6}$$

Similarly, for m2,

$$P_2 = m_2gl_2(1 - cos\theta_2) \tag{7}$$

Therefore,

$$P = P_1 + P_2 = -m_1gl_1C_1 - m_2gl_2C_2 \tag{8}$$

As we mentioned above, we need to use the Euler-Lagrange Equation,

$$L = K - P \tag{9}$$

For this system

$$K = K_1 + K_2 + K_{Crane} \tag{10}$$

Similarly,

$$P = \ P_1 + \ P_2 + \ P_{Crane} \tag{11}$$

From equations (3), (4), (5)

$$K_{crane} = 1/2[M\left(\frac{dx}{dt}\right)^2] \tag{12}$$

$$K_1 = 1/2[M\left(m_1\dot{x}^2 + m_1{l_1}^2\dot{\theta}_1^2 - 2m_1\dot{x}l_1\dot{\theta}_1 cos\theta_1\right)] \tag{13}$$

$$K_2 = 1/2[M\left(m_2\dot{x}^2 + m_2{l_2}^2\dot{\theta}_2^2 - 2m_2\dot{x}l_2\dot{\theta}_2 cos\theta_2\right)] \tag{14}$$

We know that, $P_{crane} = 0$, as the reference height is zero, h = 0

From equations (6), (7)

$$P_1 = m_1 g l_1(1 - cos\theta_1)$$

$$P_2 = m_2 g l_2(1 - cos\theta_2)$$

Substituting all the values in equation (9) we have,

$$\begin{aligned} L = \frac{1}{2}\dot{x}^2\,(M + m_1 + m_2) - \ m_1\dot{x}l_1\dot{\theta}_1\,cos\theta_1 - \ m_2\dot{x}l_2\dot{\theta}_2\,cos\theta_2 + \frac{1}{2}m_1 l_1\dot{\theta}_1^2 + \frac{1}{2}m_2 l_2\dot{\theta}_2^2 \\ - m_1 g l_1(1 - cos\theta_1) - \ m_2 g l_2(1 - cos\theta_2) \end{aligned} \tag{15}$$

From Euler- Lagrange Equation we know that,

$$F = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} \tag{16}$$

$$L = K - P$$

$$\left[\frac{\partial K}{\partial \dot{x}}\right] - \frac{\partial P}{\partial \dot{x}} = \ \frac{\partial \mathcal{L}}{\partial \dot{x}} \tag{17}$$

$$\left[\frac{\partial K}{\partial \dot{x}}\right] = (M + m_1 + \ \ m_2)\dot{x} - \ (m_1 l_1 C_1\dot{\theta}_1 + \ m_2 l_2 C_2\dot{\theta}_2) \tag{18}$$

$$\frac{\partial P}{\partial \dot{x}} = 0 \tag{19}$$

Substituting these values, we get:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{x}}\right] = (M + m_1 + m_2)\ddot{x} - [(m_1 l_1)(C_1\ddot{\theta}_1 + \dot{\theta}_1{}^2(-S_1) + (m_2 l_2)(C_2\ddot{\theta}_2\dot{\theta}_2{}^2(-S_2)] \quad (20)$$

$$\frac{\partial L}{\partial x} = \frac{\partial K}{\partial x} - \frac{\partial P}{\partial x} \quad (21)$$

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{1}{2}[0] - 0$$

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \quad (22)$$

Thus we get the value of $\ddot{x}$ as:

$$\ddot{x} = \frac{F + m_1 l_1\left(C_1\ddot{\theta}_1 - S_1\dot{\theta}_1{}^2\right) + m_2 l_2\left(C_2\ddot{\theta}_2 - S_2\dot{\theta}_2{}^2\right)}{(m_1 + m_2 + M)} \quad (23)$$

Now, we write the Lagrange equation for θ_1 to find $\ddot{\theta}_1$:

$$\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right] = \frac{\partial K}{\partial \dot{\theta}_1} - \frac{\partial P}{\partial \dot{\theta}_1} \quad (24)$$

$$\frac{\partial P}{\partial \dot{\theta}_1} = 0 \quad (25)$$

$$\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right] = \frac{\partial K}{\partial \dot{\theta}_1} - 0 \quad (26)$$

$$\frac{\partial K}{\partial \dot{\theta}_1} = m_1[-l_1 C_1\dot{x} + l_1{}^2\dot{\theta}_1] \quad (27)$$

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right] = m_1[-\ddot{x}l_1 C_1 + l_1 S_1\dot{x}\dot{\theta}_1 + l_1{}^2\ddot{\theta}_1] \quad (28)$$

$$\left[\frac{\partial \mathcal{L}}{\partial \theta_1}\right] = \frac{\partial K}{\partial \theta_1} - \frac{\partial P}{\partial \theta_1} \quad (29)$$

$$\frac{\partial K}{\partial \theta_1} = m_1 l_1 S_1\dot{x}\dot{\theta}_1 \quad (30)$$

$$\frac{\partial P}{\partial \theta_1} = m_1 g l_1 S_1 \quad (31)$$

The Lagrange equation is written as:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right] - \frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} = 0 \tag{32}$$

Now we substitute the values and after rearranging we get:

$$-m_1 \ddot{x} l_1 C_1 + m_1 l_1 S_1 \dot{x}\dot{\theta}_1 + m_1 {l_1}^2 \ddot{\theta}_1 - m_1 l_1 S_1 \dot{x}\dot{\theta}_1 - m_1 g l_1 S_1 = 0 \tag{33}$$

$$\ddot{\theta}_1 = -\frac{g}{l_1} S_1 + \frac{\ddot{x}}{l_1} C_1 \tag{34}$$

The Lagrange equation to find for θ_2 to find $\ddot{\theta}_2$ is given as:

$$\ddot{\theta}_2 = -\frac{g}{l_2} S_2 + \frac{\ddot{x}}{l_2} C_2 \tag{35}$$

Now substituting value of $\ddot{\theta}_1, \ddot{\theta}_2$ into $\ddot{x}$ and also by substituting the new derived value of $\ddot{x}$ in the equations of $\ddot{\theta}_1$, $\ddot{\theta}_2$ we get all the Non Linear Sate Space variables.

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = f(t, x(t), \dot{x}(t), \theta_1(t), \dot{\theta}_1(t), \theta_2(t), \dot{\theta}_2(t)) \tag{36}$$

The Non-Linear State Space Representation is given by:

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} = \begin{bmatrix} \dfrac{F - m_1\left(gS_1C_1 + l_1S_1{\dot{\theta}_1}^2\right) - m_2\left(gS_2C_2 + l_2S_2{\dot{\theta}_2}^2\right)}{\left(M + m_1({S_1}^2) + m_2({S_2}^2)\right)} \\ \dfrac{C_1}{l_1}\left[\dfrac{F - m_1\left(gS_1C_1 + l_1S_1{\dot{\theta}_1}^2\right) - m_2\left(gS_2C_2 + l_2S_2{\dot{\theta}_2}^2\right)}{\left(M + m_1({S_1}^2) + m_2({S_2}^2)\right)}\right] - g\dfrac{S_1}{l_1} \\ \dfrac{C_2}{l_2}\left[\dfrac{F - m_1\left(gS_1C_1 + l_1S_1{\dot{\theta}_1}^2\right) - m_2\left(gS_2C_2 + l_2S_2{\dot{\theta}_2}^2\right)}{\left(M + m_1({S_1}^2) + m_2({S_2}^2)\right)}\right] - g\dfrac{S_2}{l_2} \end{bmatrix} \tag{37}$$

Part B: Linearization of the state space representation of the system

Our system can be linearized by considering an important condition: consider θ as really small. This assumption leads to following formulations:

$$\sin\theta \approx \theta$$
$$sin^2\theta \approx 0$$
$$\cos\theta \approx 1$$

$$cos^2\theta \approx 1$$

And at the equilibrium point, x = 0, θ_1 = 0, θ_2 = 0

$$M^* = m_1 + m_2 + M$$

Substituting these values in equation of $\ddot{x}, \ddot{\theta}_1 and\ \ddot{\theta}_2$we get:

$$\ddot{x} = \frac{F + m_1 l_1(\ddot{\theta_1}) + m_2 l_2(\ddot{\theta_2})}{(M^*)} \tag{38}$$

$$\ddot{\theta_1} = -\frac{g}{l_1} S_1 + \frac{\ddot{x}}{l_1} C_1 \tag{39}$$

$$\ddot{\theta_2} = -\frac{g}{l_2} S_2 + \frac{\ddot{x}}{l_2} C_2 \tag{40}$$

The above three equations are linearized equations.

Now, considering $\ddot{x}, \ddot{\theta_1}$ and $\ddot{\theta_2}$ are independent of each other

$$\ddot{x} = \frac{F + m_1 l_1 \left(-\frac{g}{l_1} \theta_1 + \frac{\ddot{x}}{l_1}\right) + m_2 l_2 \left(-\frac{g}{l_2} \theta_2 + \frac{\ddot{x}}{l_2}\right)}{(M *)} \tag{41}$$

$$M * \ddot{x} - m_1\ddot{x} - m_2 \ddot{x} = F - m_1 g\theta_1 - m_2 g\theta_2 \tag{42}$$

$$\ddot{x} = \frac{F - m_1 g\theta_1 - m_2 g\theta_2}{(M)} \tag{43}$$

$$\ddot{\theta_1} = \frac{F}{Ml_1} - \frac{\theta_1}{l_1} g\left(\frac{M + m_1}{M}\right) - \frac{\theta_2}{M} m_2 g \tag{44}$$

$$\ddot{\theta_2} = \frac{F}{Ml_2} - \frac{\theta_2}{l_2} g\left(\frac{M + m_2}{M}\right) - \frac{\theta_1}{M} m_1 g \tag{45}$$

The State Space Representation is given by:

$$\dot{X} = AX + BU \tag{46}$$

Where $u = F$.

Using Lyapunov's Indirect Method,

$$J = \begin{bmatrix} \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \theta_1} & \frac{\partial \ddot{x}}{\partial \dot{\theta}_1} & \frac{\partial \ddot{x}}{\partial \theta_2} & \frac{\partial \ddot{x}}{\partial \dot{\theta}_2} \\ \frac{\partial \ddot{\theta}_1}{\partial x} & \frac{\partial \ddot{\theta}_1}{\partial \dot{x}} & \frac{\partial \ddot{\theta}_1}{\partial \theta_1} & \frac{\partial \ddot{\theta}_1}{\partial \dot{\theta}_1} & \frac{\partial \ddot{\theta}_1}{\partial \theta_2} & \frac{\partial \ddot{\theta}_1}{\partial \dot{\theta}_2} \\ \frac{\partial \ddot{\theta}_2}{\partial x} & \frac{\partial \ddot{\theta}_2}{\partial \dot{x}} & \frac{\partial \ddot{\theta}_2}{\partial \theta_1} & \frac{\partial \ddot{\theta}_2}{\partial \dot{\theta}_1} & \frac{\partial \ddot{\theta}_2}{\partial \theta_2} & \frac{\partial \ddot{\theta}_2}{\partial \dot{\theta}_2} \end{bmatrix} \tag{47}$$

The stability of the system is also checked using Lyapunov's indirect method. For our input A matrix, if all the eigen values have no positive real part, then the system is said to be locally stable.

The linearized system can be represented as:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-gm_1}{M} & 0 & \frac{-gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-g(m_1+M)}{Ml_1} & 0 & \frac{-gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-gm_1}{Ml_2} & 0 & \frac{-g(m_2+M)}{Ml_2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} F \tag{48}$$

Also,

$$C = I_{6\times 6} \text{ and } D = 0$$

The MATLAB simulation code for given initial values and its response is given below:

```
clc
clear all

% Defining given values
M = 1000;
m1 = 100; m2 = 100;
l1 = 20; l2 = 10;
g = 10;

% State Space form of the system
% X = A*x +B*u

A= [0 1 0 0 0 0
    0 0 (-g*m1)/M 0 (-g*m2)/M 0
    0 0 0 1 0 0
    0 0 (-g*(m1+M))/(M*l1) 0 (-g*m2)/(M*l1) 0
    0 0 0 0 0 1
```

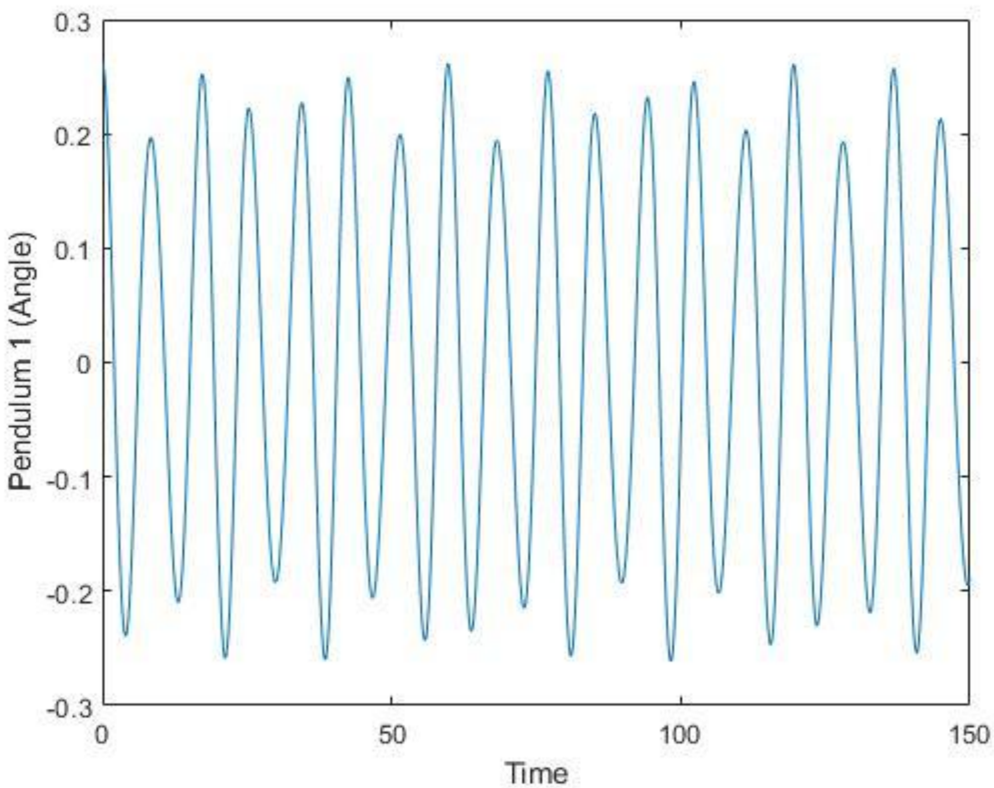
```
    0 0 (-g*m1)/(M*l2) 0 (-g*(m2+M))/(M*l2) 0 ];

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

C = [ 1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

% Initial Value Conditions
% 1. Initial Value Conditions are taken as arbitrary
% 2. Both linear & angular velocities and position are assumed to be zero
% 3. Pendulam initial positions are 15*pi/180 and %20*Pi/180

  X = [0,0,15*pi/180,0,20*pi/180,0];
  t = 0:0.01:150;
  dim_t = size(t);
  F = zeros(dim_t);

% Defining the Parameters of the State for Visualization
  State = {'x' 'xdot' 'theta1' 'theta1dot' 'theta2' 'theta2dot'};
  input = {'F'};
  Outputs = {'x'; 'alpha1'; 'alpha2'};

% State
  sys = ss(A,B,C,0,'statename',State,'inputname',input,'outputname',Outputs);
  [Y, t_T, X_T] = lsim(sys, F, t, X);

% Visualization

  figure,
  plot(t, Y(:,2));
  xlabel('Time'); ylabel('Pendulum 1 (Angle)');

  figure,
  plot(t, Y(:,3));
  xlabel('Time'); ylabel('Pendulum 2 (Angle)');

  figure,
  plot(t, Y(:,1));
  xlabel('Time'); ylabel('Crane Position');
```

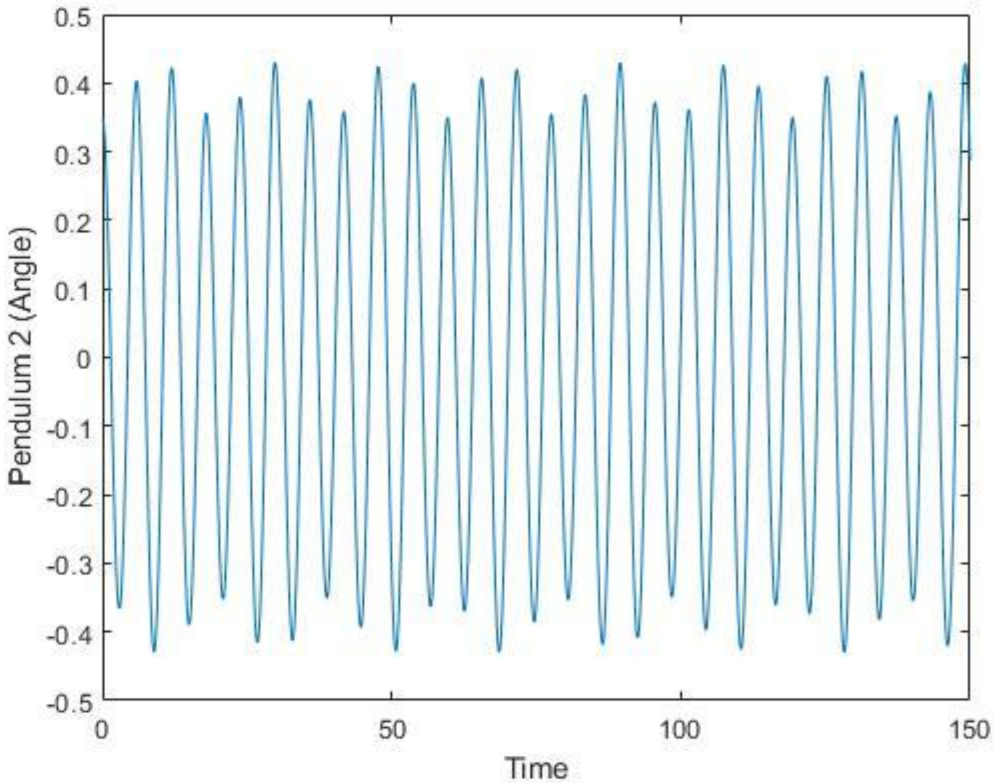
The output obtained was:

```
A =       0    1.0000         0         0         0         0
          0         0   -1.0000         0   -1.0000         0
          0         0         0    1.0000         0         0
          0         0   -0.5500         0   -0.0500         0
```

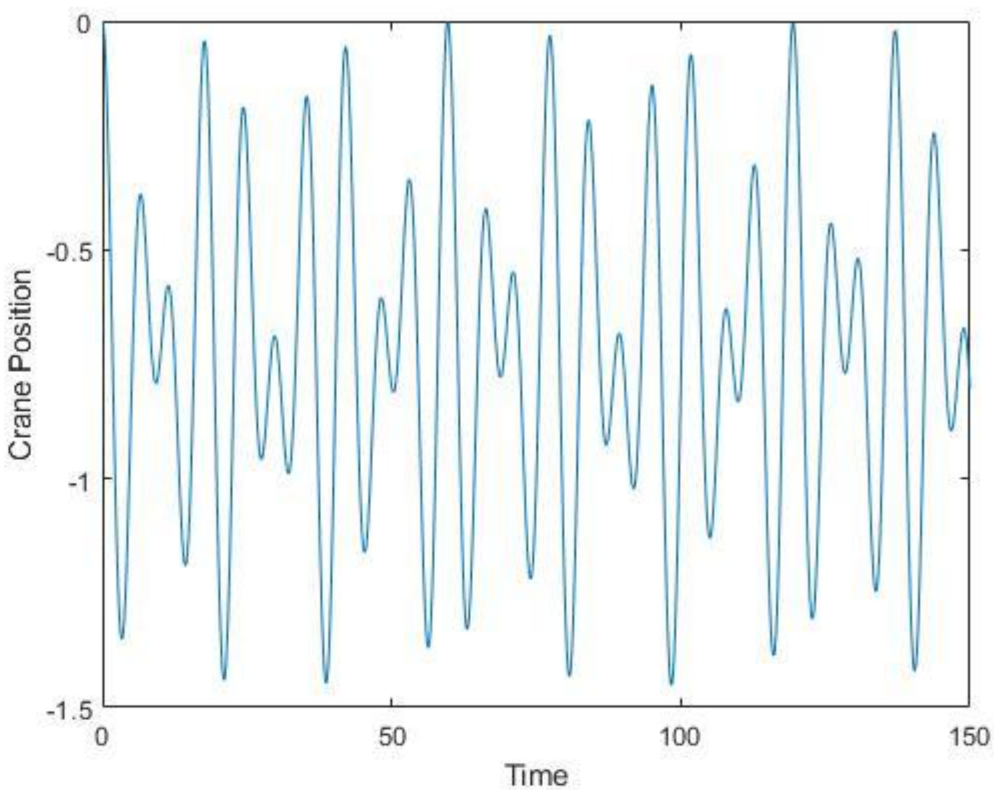
```
         0           0           0           0           0      1.0000
         0           0     -0.1000           0     -1.1000           0

B = 1.0e-03 *
          0
     1.0000
          0
     0.0500
          0
     0.1000

C =  1     0     0     0     0     0
     0     0     1     0     0     0
     0     0     0     0     1     0
```

The output plots are given below:





0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
Pendulum 2 (Angle)
0
50
100
150
Time



0
-0.5
-1
-1.5
Crane Position
0
50
100
150
Time

Part C: Conditions for the linear system to be Controllable

For checking the controllability of the system, we calculate the rank of the following matrix C:

$$C = [B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B]$$

Using MATLAB script, the determinant of the above matrix C can be given as:

$$|C| = \frac{-(g^6 l_1^2 - 2l_1 l_2 g^6 + g^6 l_2^2)}{M^6 l_1^6 l_2^6} \tag{49}$$

Using the rank technique, it can be found out that whether the system is controllable or not.
To find the condition for system to be not controllable, we put $|C| = 0$
Therefore,

$$\frac{-(g^6 l_1^2 - 2l_1 l_2 g^6 + g^6 l_2^2)}{M^6 l_1^6 l_2^6} = 0$$

$$-(g^6 l_1^2 - 2l_1 l_2 g^6 + g^6 l_2^2) = 0$$

$$-g^6(l_1^2 - 2l_1 l_2 + l_2^2) = 0$$

$$\text{i.e. } (l_1 - l_2)^2 = 0$$

$$\text{i.e. } l_1 = l_2 \tag{50}$$

Therefore, we can say that the system is not controllable when $l_1 = l_2$

The MATLAB code to find the determinant expression of C matrix is given below:

```
clc
clear all

syms g m1 m2 l1 l2 M
% State Space form of the system
% X = A*x +B*u

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

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

```
    0
    1/(M*l2)];

C = [ 1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

mat = [B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B];
rank_mat = rank([B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B])
det(mat)
```

The output obtained was:

```
rank_mat =

     6

ans =

-(g^6*l1^2 - 2*g^6*l1*l2 + g^6*l2^2)/(M^6*l1^6*l2^6)
```

Part D: LQR Controller Design

For designing the LQR Controller, following values were considered for the variables:
Mass of Trolley, M = 1000kg
Mass on pendulum 1, m1 = 100kg
Length of pendulum 1, l1 = 20m
Mass on pendulum 2, m2 = 100kg
Length of pendulum 2, l2 = 10m

The LQR Controller uses a cost function, which is given by:

$$J = \int_0^{\infty} (x^T Q x \; + \; u^T R u) dt \tag{51}$$

Upon minimization, this function gives rise to Riccati equation, given by:

$$A^T + PA - PBR^{-1}B^T P + Q = 0 \tag{52}$$

Using this equation, we find the value of P. Then we find the controller gain K in $u = -Kx$ using

$$-K = R^{-1}BP \tag{53}$$

The value of K is calculated using LQR function in MATLAB. The initial conditions are assumed to be: $X = [0 \quad 0 \quad 15 \quad 0 \quad 20 \quad 0]$. The elements of the Q matrix were taken as:

$$Q(1,1) = 100$$

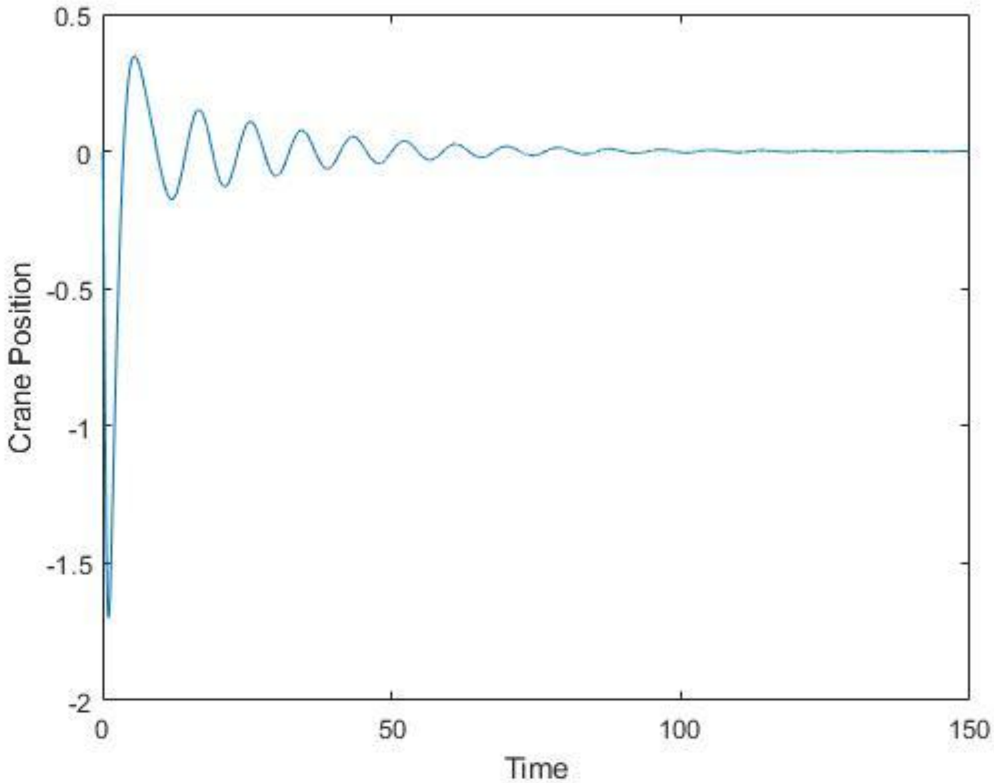
$$Q(2,2) = 2000$$
$$Q(3,3) = 50000$$
$$Q(4,4) = 120000$$
$$Q(5,5) = 450000$$
$$Q(6,6) = 800000$$

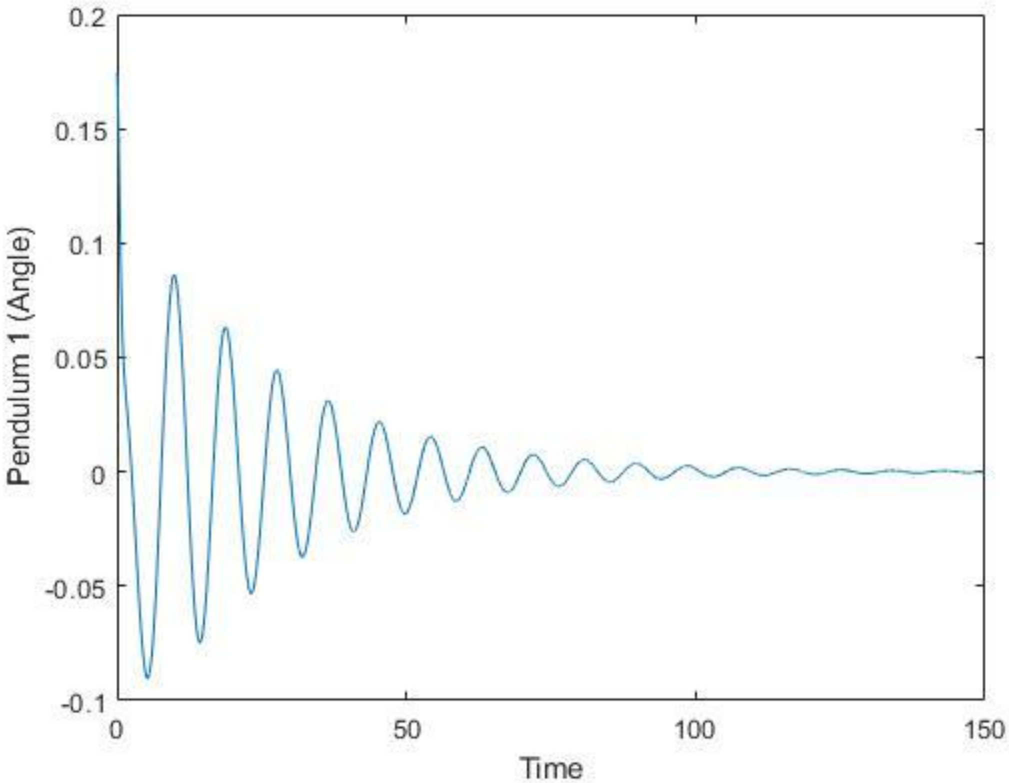
The MATLAB simulation code for the designed LQR controller is given below:

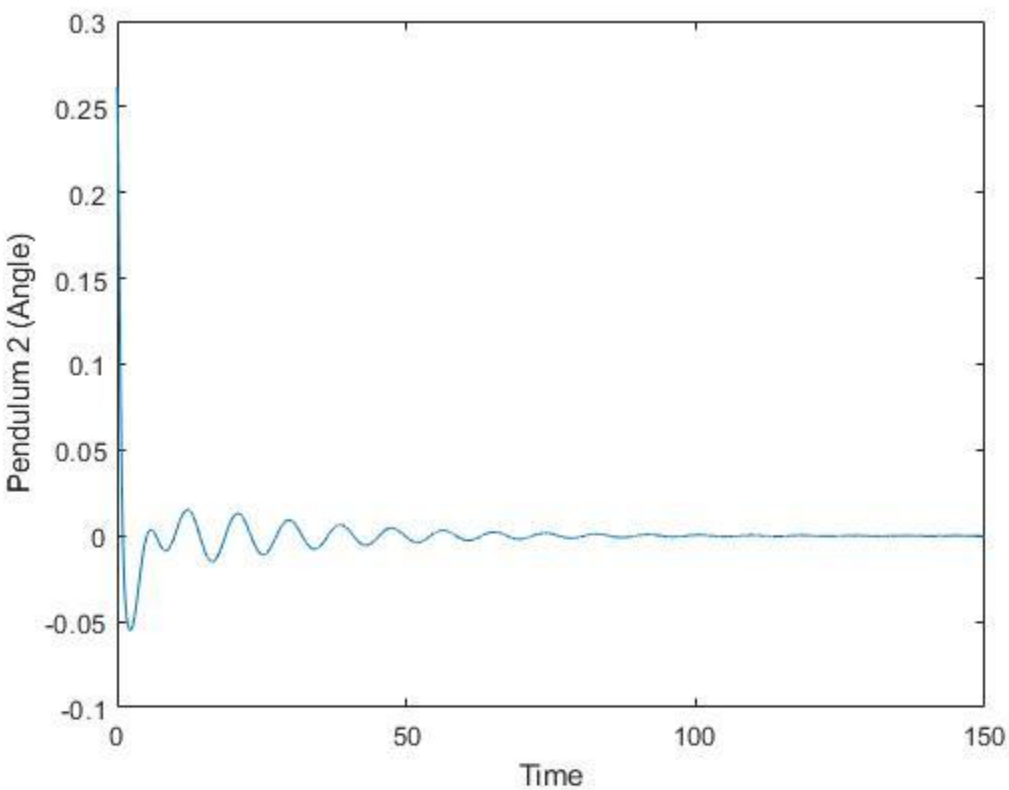
```
clc
clear all

% Defining given values
M = 1000;
m1 = 100; m2 = 100;
l1 = 20; l2 = 10;
g = 10;

% State Space form of the system
% X = A*x +B*u

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

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

C = [ 1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

D = 0;

% Putting the values of Q & R

Q = (C')*(C);

% Assigning the Values in Q using Trial & Error Method

Q(1,1) = 50000000;
Q(3,3) = 600000000;
Q(5,5) = 7000000000;

% Selecting the Ideal Value of R
```

```
R = 1;

% Designing the LQR

% Calculating the Optimal Gain Matrix K

K = lqr(A, B, Q, R);

% Calculating the New Value of A using K

A_New = (A - (B*K));

% Creating the Observability Matrix

States = {'x' 'x_dot' 'theta1' 'theta1_dot' 'theta2' 'theta2_dot'};
Inputs = {'r'};
Outputs = {'x'; 'phi1'; 'phi2'};

% Creating the State Space Model

ClosSS = ss(A_New, B, C, D,'statename', States, 'inputname', Inputs,
'outputname', Outputs);

% Initializing Conditions

X0 = [0;
      0;
      10*pi/180;
      0;
      15*pi/180;
      0];

t = 0:0.01:150;

Temp = size(t);
F = zeros(Temp);

% Simulating the Time Response of Dynamic System to Arbitrary Inputs

[Y, tTemp, XTemp] = lsim(ClosSS, F, t, X0);


% Visualization

figure,
plot(t, Y(:,1));
xlabel('Time'); ylabel('Crane Position');

figure,
plot(t, Y(:,2));
xlabel('Time'); ylabel('Pendulum 1 (Angle)');
```

```
figure,
plot(t, Y(:,3));
xlabel('Time'); ylabel('Pendulum 2 (Angle)');
```




Pendulum 1 (Angle)
Time
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
50
100
150


Pendulum 2 (Angle)
Time
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
50
100
150

The MATLAB simulation code of the LQR Controller for the controllable system is given below:

```
clc
clear all

% Defining given values
M = 1000;
m1 = 100; m2 = 100;
l1 = 20; l2 = 10;
g = 10;

% State Space form of the system
% X = A*x +B*u
A= [0 1 0 0 0 0
    0 0 (-g*m1)/M 0 (-g*m2)/M 0
    0 0 0 1 0 0
    0 0 (-g*(m1+M))/(M*l1) 0 (-g*m2)/(M*l1) 0
    0 0 0 0 0 1
    0 0 (-g*m1)/(M*l2) 0 (-g*(m2+M))/(M*l2) 0 ];

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

C = [ 1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

% Controllability Check
disp('Controllability Check')

rank_mat = rank([B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B]);

if rank_mat == 6
    disp('System is Controllable')
else
    disp('System is Not Controllable')
end

%% DESIGN FOR LQR CONTROLLER
% Since we Observed that the system is Controllable,
% We determine the state Feedback (K)

% Adjusting Cost  'Q' until the system is controllable
Q = diag([100 2000 50000 120000 450000 800000]);
R = 0.01;

% State FeedBack 'K'
K = lqr(A,B,Q,R);
```

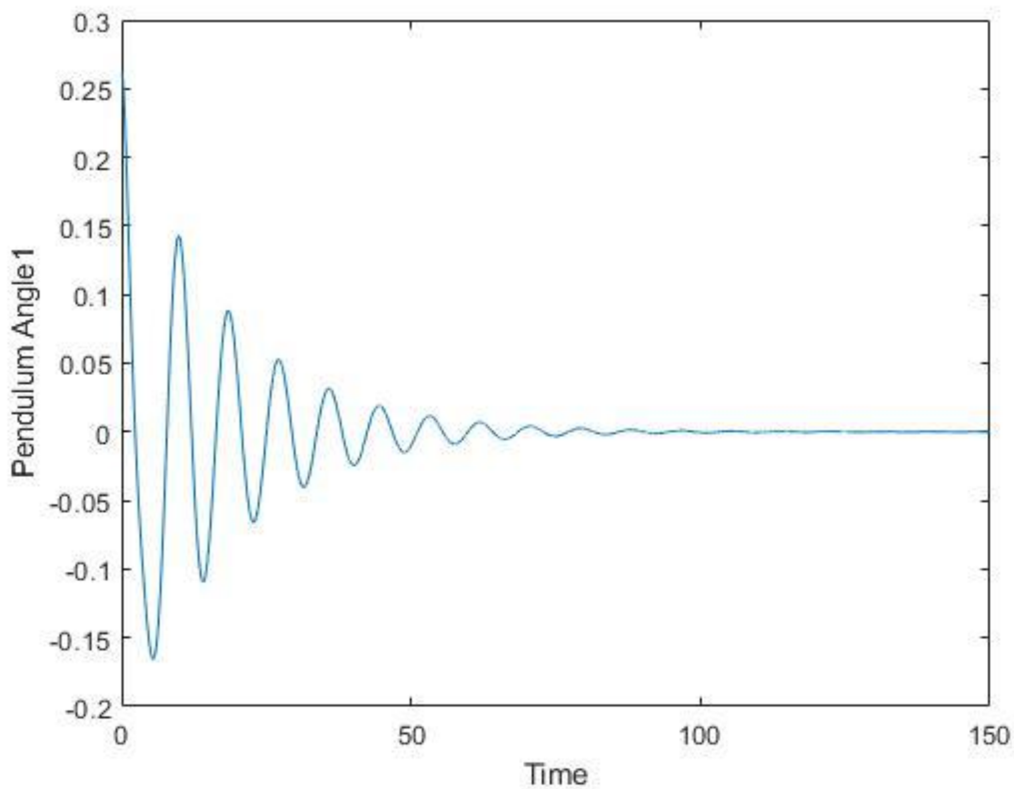
```
disp(K)

% New State Space System equation will be 'X= (A-B*K)x + B*U'

A_N=[A-B*K];
States={'x' 'x_dot' 'theta1' 'theta1_dot' 'theta2' 'theta2_dot'};
input={'F'};
Outputs = {'x'; 'alpha1'; 'alpha2'};

% Converting to a State Space Model
sys=ss(A_N,B,C,0,'statename',States,'inputname',input,'outputname',Outputs);

% Initial Value Conditions
% 1. Initial Value Conditions are taken as arbitrary
% 2. Both linear & angular velocities and position are assumed to be zero
% 3. Pendulam initial positions are 15*pi/180 and %20*Pi/180

X=[0;0;15*pi/180;0;20*pi/180;0];
t = 0:0.01:150;
dim_t= size(t);
F = zeros(dim_t);
[Y, t_T, X_T] = lsim(sys, F, t, X);
size(Y)
%% Visualization
figure,
plot(t, Y(:,1));
xlabel('Time'); ylabel('Crane Position');

figure,
plot(t, Y(:,2));
xlabel('Time'); ylabel('Pendulum Angle1');

figure,
plot(t, Y(:,3));
xlabel('Time'); ylabel('Pendulum Angle2');
% Using the Lynapunov indirect Method to Obtain Stability

%System has been Linearized
Ly_St= eig(A_N)
if real(Ly_St)<1
    disp('System is Stable')
else
    disp('System is not Stable')
end
```

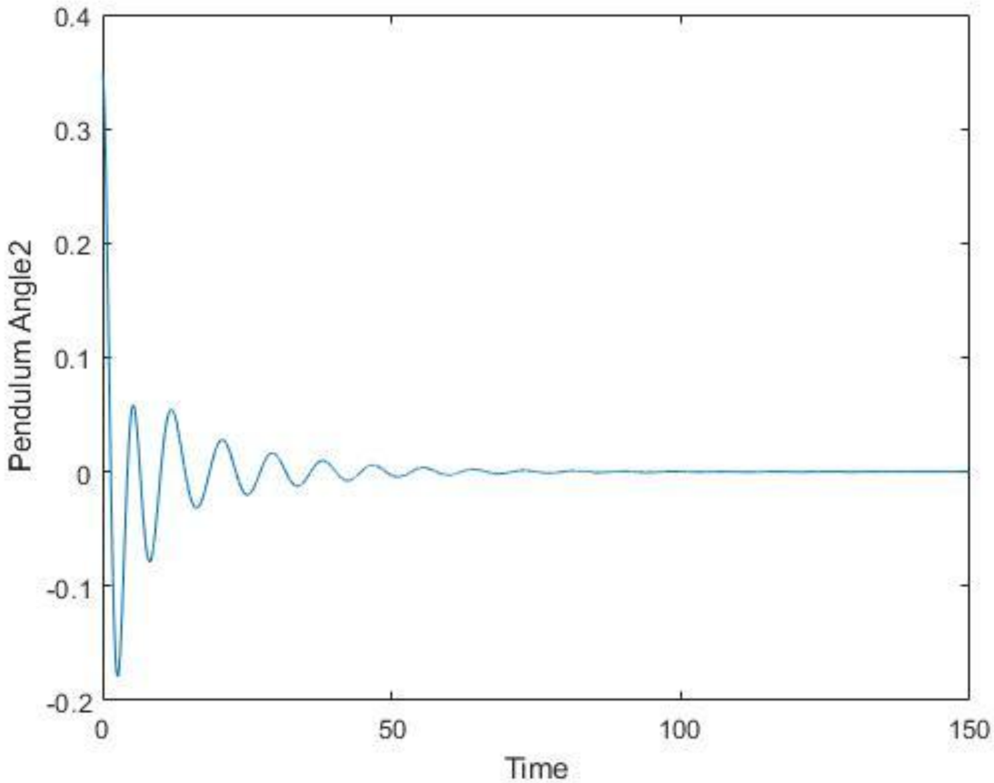
The output obtained was:

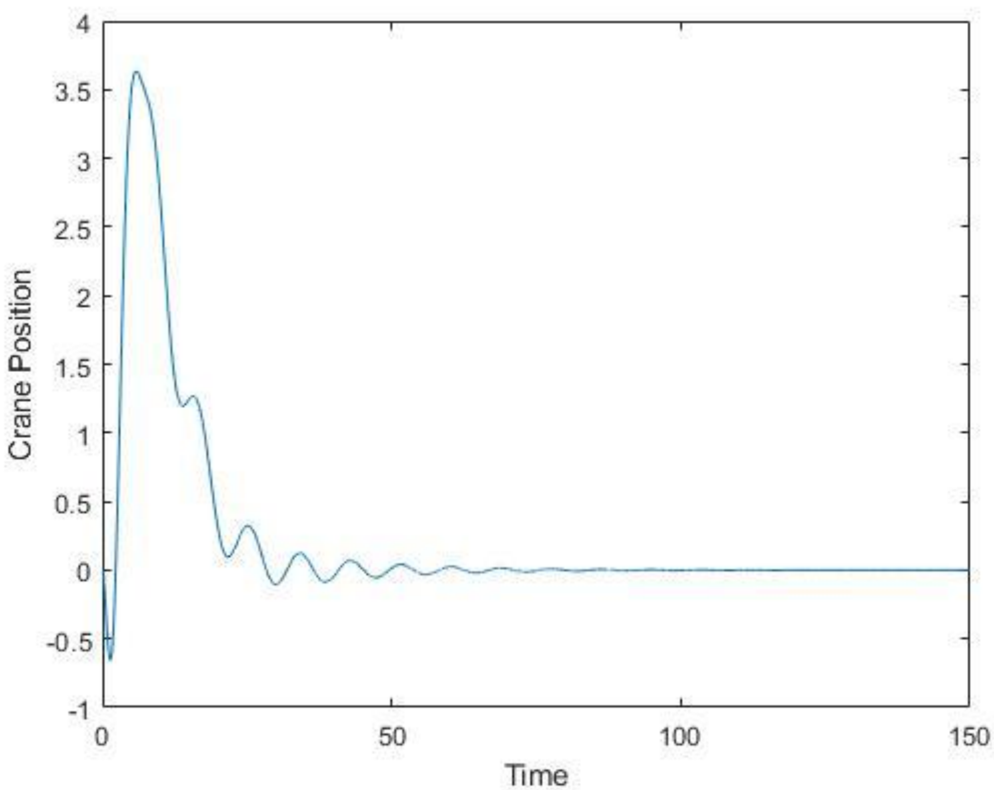
```
Controllability Check
System is Controllable
   1.0e+03 *

    0.1000    0.7719   -0.2418    4.7802    8.0344    6.6619
```

```
ans =
       15001           3
Ly_St =

  -0.5304 + 0.9775i
  -0.5304 - 0.9775i
  -0.2583 + 0.1035i
  -0.2583 - 0.1035i
  -0.0498 + 0.7207i
  -0.0498 - 0.7207i

System is Stable
```

The visual output plot is given below:




0.4
0.3
0.2
0.1
0
-0.1
-0.2
Pendulum Angle2
0
50
100
150
Time


4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
Crane Position
0
50
100
150
Time

Part E: Observability

We checked the Observability of the system for 4 different cases as: x(t), (t1, t2), (x, t2) and (x, t1, t2). We found that the system is observable for all the cases except for (t1, t2).

We used the matrix C given below and found out its rank for each case and compared with the rank of our input matrix A to find out the observability of the system.

$$C = [B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B]$$

The MATLAB simulation code for checking the observability of the system for different cases is given below:

```
clc
clear all

% Defining given values
M = 1000;
m1 = 100; m2 = 100;
l1 = 20; l2 = 10;
g = 10;

% State Space form of the system
% X = A*x +B*u

A= [0 1 0 0 0 0
    0 0 (-g*m1)/M 0 (-g*m2)/M 0
    0 0 0 1 0 0
    0 0 (-g*(m1+M))/(M*l1) 0 (-g*m2)/(M*l1) 0
    0 0 0 0 0 1
    0 0 (-g*m1)/(M*l2) 0 (-g*(m2+M))/(M*l2) 0 ]

A_dimension = 6;

% Checks for Different Cases

% Case 1: for x(t)

disp('Observability check for Case-x(t)')

C1=[1 0 0 0 0 0];
Rank1 = rank([C1;C1*A;C1*(A^2);C1*(A^3);C1*(A^4);C1*(A^5)])

disp('Observability check for Case-x(t)')

if Rank1 == A_dimension
    disp('System is Observable for x(t)')
else
    disp('System is not Observable')
end
```

```
% Case 2: for (t1,t2)

C2=[0 0 1 0 0 0
    0 0 0 0 1 0];

Rank2 = rank([C2;C2*A;C2*(A^2);C2*(A^3);C2*(A^4);C2*(A^5)])

disp('Observability check for Case-(t1,t2)')

if Rank2 == A_dimension
    disp('System is Observable for (t1,t2)')
else
    disp('System is not Observable')
end

%Case 3 for (x,t2)

C3=[1 0 0 0 0 0
    0 0 0 0 1 0];

Rank3 = rank([C3;C3*A;C3*(A^2);C3*(A^3);C3*(A^4);C3*(A^5)])

disp('Observability check for Case-(x,t2)')

if Rank3 == A_dimension
    disp('System is Observable for (x,t2)')
else
    disp('System is not Observable')
end


%Case 4 for (x,t1,t2)

C4=[1 0 0 0 0 0
    0 0 1 0 0 0
    0 0 0 0 1 0];

Rank4 = rank([C4;C4*A;C4*(A^2);C4*(A^3);C4*(A^4);C4*(A^5)])

disp('Observability check for Case-(x,t1,t2)')

if Rank4 == A_dimension
    disp('System is Observable for (x,t1,t2)')
else
    disp('System is not Observable')
end
```

The output for the above code is:

```
Observability check for Case-x(t)
```

```
Rank1 =

     6

Observability check for Case-x(t)
System is Observable for x(t)

Rank2 =

     4

Observability check for Case-(t1,t2)
System is not Observable

Rank3 =

     6

Observability check for Case-(x,t2)
System is Observable for (x,t2)

Rank4 =

     6

Observability check for Case-(x,t1,t2)
System is Observable for (x,t1,t2)
```

Part F: Luenberger Observer

The Luenberger Observer is given by the equation:

$$\dot{X} = AX + BU + LC(X - \hat{X}) \tag{54}$$

Putting $X_e = X - \hat{X}$, we get,

$$\dot{X_e} = (A - LC)\dot{X_e} \tag{55}$$

Now, the 'best' possible Luenberger Observer can be developed by by simulating it using both the original nonlinear system as well as the linearized version.
The conditions for the same are:

For $A^T and\ C^T$ to be stabilizable, they should be detectable and observable.
Also, for $(A - LC)$ to be stable, the matrix $= (A - LC)^T$ must be stable.

The MATLAB simulation code for the Luenberger Observer is given below:

```
clc
clear all

% Defining given values
M = 1000;
m1 = 100; m2 = 100;
l1 = 20; l2 = 10;
g = 10;

% State Space form of the system
% X = A*x +B*u

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

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

C = [ 1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

  % The error Dynamics in the State Space Observer is (A-L*C)*e

  % We know from the Obersvability Checks that
  % The pair A,C is observable only for following 3 cases:
  % x(t), (x,t2) and (x,t1,t2)

Q = diag([100 2000 50000 120000 450000 800000]);
R = 0.01;

disp(' State FeedBack')
% State FeedBack 'K'
K = lqr(A,B,Q,R);
disp(K)

disp('Eigen Values of (A-B*K)')
% The Pole Placement should be faster than the eigen values of (A-B*K)
Eig_Vals = eig(A-B*K);
disp(Eig_Vals)
```

```
% Arbitrary Pole Placement
Poles = [-2 -4 -5 -6 -7 -8];

% State Outputs for the system that are Observable

C1 = [1 0 0 0 0 0];

C3 = [1 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];

L1 = place(A', C1',Poles)';
L2 = place(A', C3',Poles)';
L3 = place(A', C4',Poles)';

% Defining System Matrices for each state of the Observer

% For Case1: x(t)
A_L1 = [(A-B*K) (B*K)
        zeros(6,6) (A-L1*C1)];
B_L1 = [B
        zeros(size(B))];
C_L1 = [C1 zeros(size(C1))];

% For Case2: (x(t), t2)
A_L2 = [(A-B*K) (B*K)
        zeros(6,6) (A-L2*C3)];
B_L2 = [B
        zeros(size(B))];
C_L2 = [C3 zeros(size(C3))];

% For Case3: (x(t), t2, t1)
A_L3 = [(A-B*K) (B*K)
        zeros(6,6) (A-L3*C4)];
B_L3 = [B
       zeros(size(B))];
C_L3 = [C4 zeros(size(C4))];

% State Space representation of the Linear system
SS1 = ss(A_L1,B_L1,C_L1,0)
SS2 = ss(A_L2,B_L2,C_L2,0);
SS3 = ss(A_L3,B_L3,C_L3,0);

% Visualizing Positions From STEP function

figure(1)
step(SS1)
title('observability for Case1: x(t)')
xlabel('Time')
ylabel('Position')
```

```
figure(2)
step(SS2)
title('observability for Case2: (x(t), t2)')
xlabel('Time')
ylabel('Position')

figure(3)
step(SS3)
title('observability for Case3: (x(t), t2, t1)')
xlabel('Time')
ylabel('Position')
```

The output obtained was:

```
Eigen Values of (A-B*K)
  -0.5304 + 0.9775i
  -0.5304 - 0.9775i
  -0.2583 + 0.1035i
  -0.2583 - 0.1035i
  -0.0498 + 0.7207i
  -0.0498 - 0.7207i
```

Here, since all the eigen values have no positive real part, we can say that the system is locally stable.

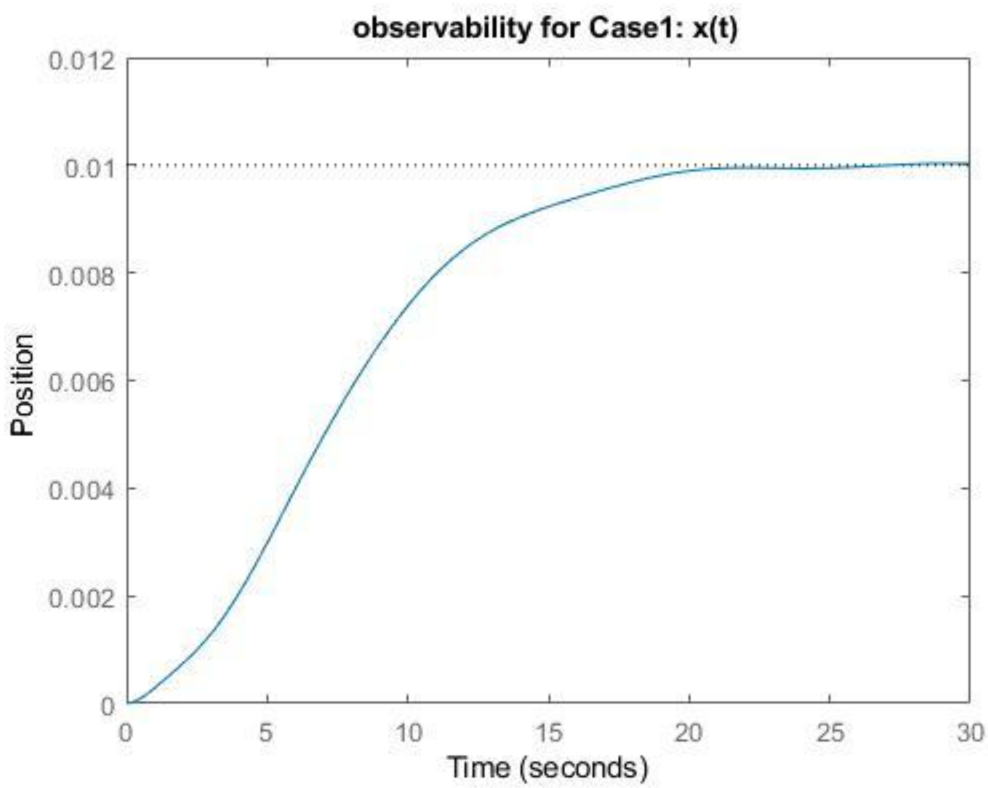
The state space model matrices obtained for x(t) are given below:

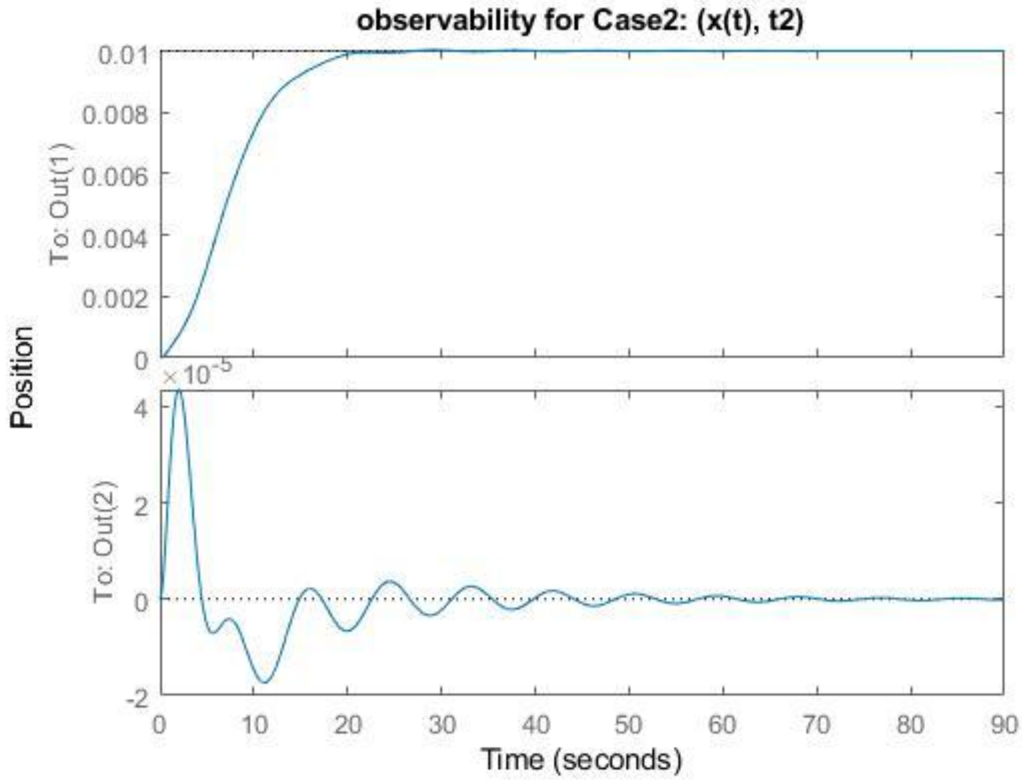
```
SS1 =

  A =
                x1          x2          x3          x4
x5          x6          x7
   x1            0           1           0           0
0           0           0
   x2         -0.1     -0.7719     -0.7582       -4.78      -
9.034      -6.662         0.1
   x3            0           0           0           1
0           0           0
   x4       -0.005     -0.0386     -0.5379      -0.239      -
0.4517     -0.3331       0.005
   x5            0           0           0           0
0           1           0
   x6        -0.01    -0.07719    -0.07582      -0.478      -
1.903     -0.6662        0.01
   x7            0           0           0           0
0           0         -32
```

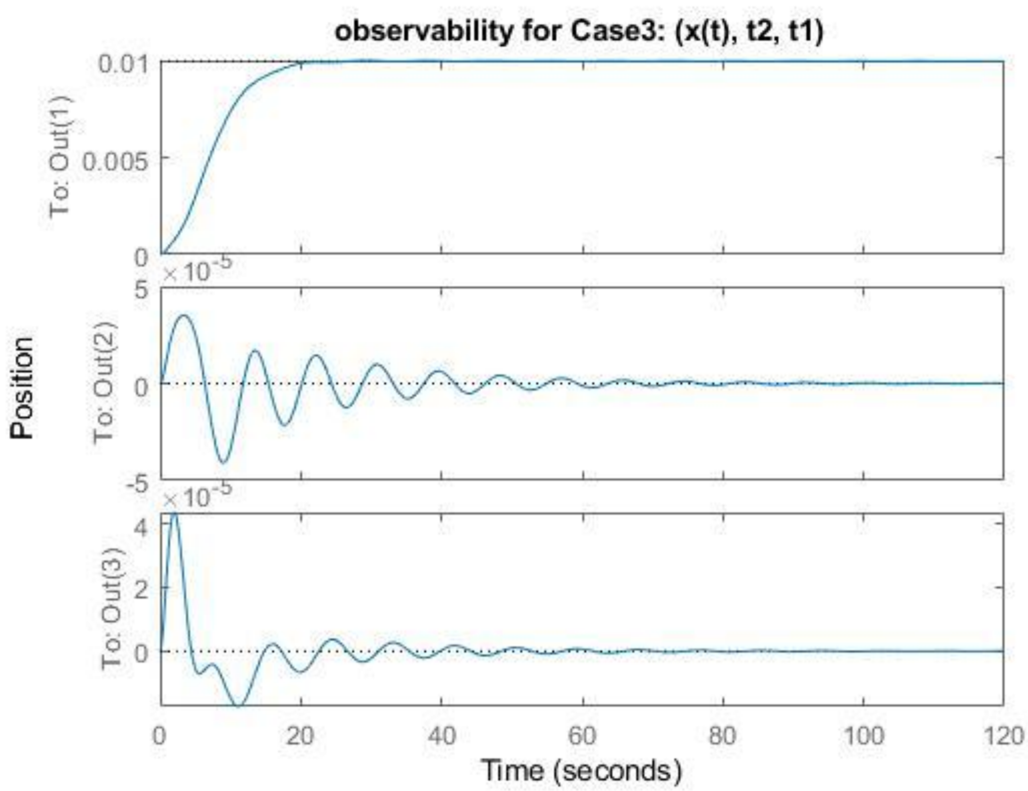
```
   x8             0            0            0            0
0            0      -413.4
   x9             0            0            0            0
0            0   3.445e+04
   x10            0            0            0            0
0            0   1.698e+04
   x11            0            0            0            0
0            0  -3.172e+04
   x12            0            0            0            0
0            0       -7581

                 x8           x9          x10          x11
x12
   x1             0            0            0            0
0
   x2        0.7719      -0.2418         4.78        8.034
6.662
   x3             0            0            0            0
0
   x4        0.0386     -0.01209        0.239       0.4017
0.3331
   x5             0            0            0            0
0
   x6       0.07719     -0.02418        0.478       0.8034
0.6662
   x7             1            0            0            0
0
   x8             0           -1            0           -1
0
   x9             0            0            1            0
0
   x10            0        -0.55            0        -0.05
0
   x11            0            0            0            0
1
   x12            0         -0.1            0         -1.1
0

  B =
             u1
   x1         0
   x2     0.001
   x3         0
   x4     5e-05
   x5         0
   x6    0.0001
   x7         0
```

```
   x8         0
   x9         0
   x10        0
   x11        0
   x12        0

  C =
        x1   x2   x3   x4   x5   x6   x7   x8   x9  x10  x11
x12
   y1    1    0    0    0    0    0    0    0    0    0    0
0

  D =
       u1
   y1   0
```

The visualization output is given below:




observability for Case2: (x(t), t2)
Position
To: Out(1)
To: Out(2)
0.01
0.008
0.006
0.004
0.002
0
× 10^-5
4
2
0
-2
0
10
20
30
40
50
60
70
80
90
Time (seconds)


observability for Case3: (x(t), t2, t1)
Position
To: Out(1)
To: Out(2)
To: Out(3)
0.01
0.005
0
× 10^-5
5
0
-5
× 10^-5
4
2
0
0
20
40
60
80
100
120
Time (seconds)

Part G: LQG Controller

The LQG controller is a combination of LQR and Kalman Filter.
The state space representation for the LQG controller is given as:

$$\dot{X} = AX + BU + B\omega \tag{56}$$

$$Y = CX + v \tag{57}$$

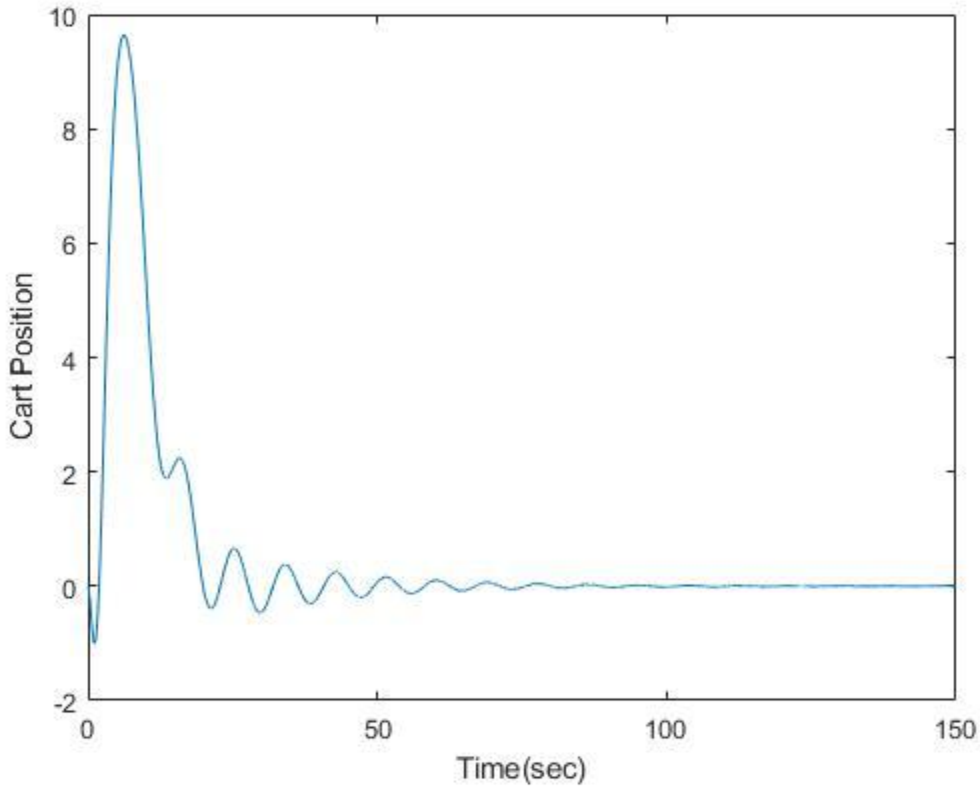
Where, ω is the process noise and v is the measurement noise.

The MATLAB simulation code for the LQG controller is given below:

```
clc
clear all

% Defining given values
M = 1000;
m1 = 100; m2 = 100;
l1 = 20; l2 = 10;
g = 10;

% State Space form of the system
% X = A*x +B*u

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

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

C = [ 1 0 0 0 0 0
      0 0 1 0 0 0
      0 0 0 0 1 0];

% Considering FeedBack Control for the Case x(t)
C1 = [1 0 0 0 0 0];
Q = diag([100 2000 50000 120000 450000 800000]);
R = 0.01;
disp('State FeedBack')
K = lqr(A,B,Q,R);
```

```
disp(K)


Poles = [-2 -4 -5 -6 -7 -8];
L1 = place(A',C1',Poles)';

% Kalman Estimator
stat_space = ss(A,[B B],C,0);
R1 = 0.01; Q1  =0.05;
sensors = [1];
W = [1];
[~,L,~] = kalman(stat_space,Q1,R1,[],sensors,W);

% Defining The Parameters of the State for Visualization
States = {'x' 'x_dot' 'theta1' 'theta1_dot' 'theta2'
'theta2_dot','e_1','e_2','e_3','e_4','e_5','e_6'};
input = {'F'};
Outputs = {'x'};

A_L1 = [(A-B*K) (B*K)
        zeros(6,6) (A-L1*C1)];
B_L1 = [B
        zeros(size(B))];
C_L1 = [C1 zeros(size(C1))];

stat_space2 =
ss(A_L1,B_L1,C_L1,0,'statename',States,'inputname',input,'outputname',Outputs
);


X = [0;0;45*pi/180;0;35*pi/180;0;0;0;0;0;0;0];
t = 0:0.01:150;
dim_t= size(t);
F = zeros(dim_t);
[Y, t_T, X_T] = lsim(stat_space2, F, t, X);

figure(1)
plot(t,Y(:,1))
ylabel('Cart Position')
xlabel('Time(sec)')
```

The visualization output for the LQG system, for a step input given below:



References:

1. F. Borrelli and T. Keviczky, "Distributed LQR Design for Identical Dynamically Decoupled Systems," in IEEE Transactions on Automatic Control, vol. 53, no. 8, pp. 1901-1912, Sept. 2008, doi: 10.1109/TAC.2008.925826.
2. N. Sun, Y. Wu, X. Liang and Y. Fang, "Nonlinear Stable Transportation Control for Double-Pendulum Shipboard Cranes With Ship-Motion-Induced Disturbances," in IEEE Transactions on Industrial Electronics, vol. 66, no. 12, pp. 9467-9479, Dec. 2019, doi: 10.1109/TIE.2019.2893855.