Control of Robotic Systems

ENPM 667

Final Project – Design of LQR and LQG Controller for an inverted dual-pendulum crane

Submitted on:

12/18/2023

Bhavana B Rao

UID- 120389092

Sri Ram Kothapalli

UID – 120165177

Charith Reddy Gannapu Reddy

UID -119360347

CONTENTS

1. Introduction

Imagine a crane traveling along a one-dimensional track, resembling a frictionless cart with mass M controlled by an external force F, serving as the system input. The crane is connected to two loads via cables, with masses m1 and m2, and cable lengths l1 and l2, respectively. The illustration below represents the crane and the variables referenced in this project.

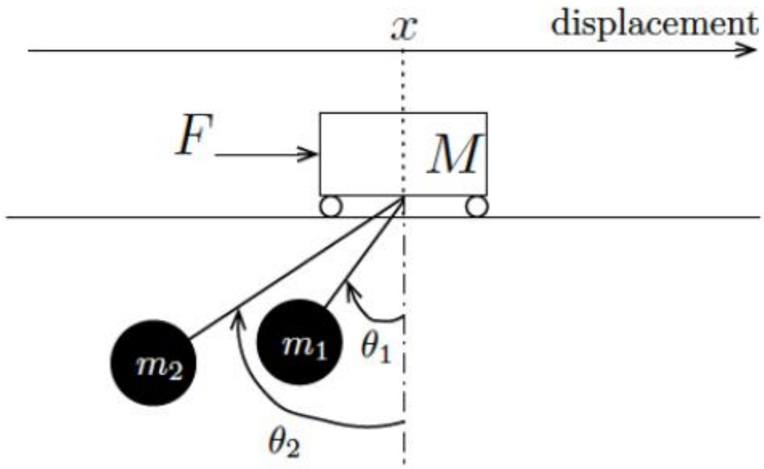

Figure 1. Crane System

Steps for the execution of the project are as follows:

1. The initial phase involves deriving the equations of motion for the system using the Langragian method. Subsequently, we aim to establish the nonlinear state-space representation of the system. The following step entails linearizing the system around an equilibrium point and expressing the state-space representation for this linearized version.

2. Moving forward, we delve into determining the controllability conditions based on the system parameters (M, m1, m2, l1, l2). Once the controllability conditions are established, the project proceeds to design an LQR controller. To ensure controllability, the system undergoes a check, and upon confirmation, the LQR controller is crafted. Simulation responses are then recorded for both the original nonlinear system and the linearized system under the influence of the LQR controller. Adjustments to LQR parameters are made to achieve desired responses, followed by a Lyapunov analysis to verify stability.

3. With an LQR controller in place, the project assesses observability for specific output vectors using the parameters obtained for controllability conditions. The subsequent step involves identifying the optimal Luenberger Observer for observable output vectors and simulating responses to various input conditions for both the linearized and nonlinear systems.

4. The final phase centres on designing an output feedback controller using the LQG method, specifically for the smallest output vector. The controller's performance is then illustrated through simulations.

2. Equations of Motion and non-linear state space representation

The position of mass m1 in x and y direction can be given as follows:

$$x_{m1} = (x - l_1\,sin(\theta_1))\hat{\imath} + (-l_1\,cos(\theta_1))\hat{\jmath} \qquad (1)$$

$$y_{m1} = (-l_1\cos(\theta_1))\hat{\jmath}$$

We can obtain velocity by differentiating Equation (1) with respect to time and we get:

$$v_{m2} = \dot{x} - l_1\cos(\theta_1)\,\dot{\theta_1}\hat{\imath} + l_1\sin(\theta_1)\,\dot{\theta_1}\hat{\jmath} \qquad (2)$$

Similarly, we can compute the position of mass m_2,

$$x_{m2} = (x - l_2\sin(\theta_2))\hat{\imath} + (-l_2\cos(\theta_2))\hat{\jmath} \qquad (3)$$

We can obtain velocity by differentiating Equation (4) with respect to time and we get:

$$v_{m2} = \dot{x} - l_2\cos(\theta_2)\,\dot{\theta_2}\hat{\imath} + l_1\sin(\theta_1)\,\dot{\theta}_1\hat{\jmath} \qquad (4)$$

We can compute the kinetic energy using the velocity equations,

$$K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\left(\dot{x} - l_1\dot{\theta_1}\cos(\theta_1)\right)^2 + \frac{1}{2}m_1\left(l_1\dot{\theta_1}(\sin(\theta_2))\right)^2 + \frac{1}{2}m_2\left(\dot{x} - l_2\dot{\theta_2}\cos(\theta_2)\right)^2 + \frac{1}{2}\left(l_2\dot{\theta_2}(\sin(\theta_2))\right)^2 \qquad (5)$$

Potential energy is given by P = mgh

$$P = -m_1gl_1\cos(\theta_1) - m_2gl_2\cos(\theta_2) = -g[-m_1l_1\cos(\theta_1) - m_2l_2\cos(\theta_2)]$$

The lagrange equation is given by the formula,

L = K.E - P.E

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\dot{x}^2 + \frac{1}{2}m_1l_1^2(\dot{\theta_1})^2\cos^2(\theta_1) - m_1l_1(\dot{\theta_1})\dot{x}\cos(\theta_1) + \frac{1}{2}m_1l_1^2(\dot{\theta_1})^2\sin^2(\theta_1) + \frac{1}{2}m_2\dot{x}^2 + \frac{1}{2}m_2l_2^2(\dot{\theta_2})^2\cos^2(\theta_2) - m_2l_2(\dot{\theta_2})\dot{x}\cos(\theta_2) + \frac{1}{2}m_2l_2^2(\dot{\theta_2})^2\sin^2(\theta_1) + g[m_1l_1\cos(\theta_1) + m_2l_2\cos(\theta_2)] \qquad (6)$$

We can simplify the above equation as follows,

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}(m_1 + m_2)\dot{x}^2 + \frac{1}{2}m_1l_1^2(\dot{\theta_1})^2 + \frac{1}{2}m_2l_2^2(\dot{\theta_2})^2 - \dot{x}\left(m_1l_1(\theta_1)\dot{\cos}(\theta_1) + m_2l_2(\theta_2)\dot{\cos}(\theta_2)\right) + g[m_1l_1\cos(\theta_1) + m_2l_2\cos(\theta_2)] \qquad (7)$$

The Lyapunov Equations pertaining to the state variables considered for our system are defined as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = F \qquad (8)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial \theta_1}\right) = 0 \qquad (9)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \left(\frac{\partial L}{\partial \theta_2}\right) = 0 \qquad (10)$$

Now computing all the above relation by substituting:

First taking

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

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} = M\dot{x}^2 + (m_1 + m_2)\dot{x} - m_1 l_1(\dot{\theta_1})\cos(\theta_1) - m_2 l_2(\dot{\theta_2})\cos(\theta_2)$$

Differentiating the above equation with respect to time:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = M\ddot{x} + (m_1 + m_2)\ddot{x} - \left[m_1 l_1(\dot{\theta_1})\cos(\theta_1) - m_1 l_1(\theta_1)^2\cos(\theta_1)\right] - \left[m_2 l_2(\ddot{\theta_2})\cos(\theta_2) - m_2 l_2(\dot{\theta_2})^2\sin(\theta_2)\right] \qquad (11)$$

Here,

$$\left(\frac{\partial L}{\partial x}\right) = 0$$

Rewriting the equation after combining

$$[M + m_1 + m_2]\ddot{x} - m_1 l_1(\ddot{\theta_1})\cos(\theta_1) + m_1 l_1(\dot{\theta_1})^2\sin(\theta_1) - m_2 l_2(\ddot{\theta_2})\cos(\theta_2) + m_2 l_2(\theta_2)^2\dot{\sin}(\theta_2) = F \qquad (12)$$

From equation (9):

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

Differentiating the above equation with respect to time:

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

Here,

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

Now combining both the above equations:

$$m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \ddot{x}\cos(\theta_1) + m_1 \dot{x} l_1 (\dot{\theta}_1)\sin(\theta_1) - m_1 l_1 (\dot{\theta}_1)\dot{x}\sin(\theta_1) + m_1 l_1 g \sin(\theta_1) = 0 \tag{13}$$

Reducing the equation by cancelling out terms we get:

$$m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \ddot{x}\cos(\theta_1) + m_1 l_1 g \sin(\theta_1) = 0 \tag{14}$$

Equation we obtain from second Lagrange Equation.

We perform the below calculations to find the third equation:

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

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

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = m_2 l_2^2 \ddot{\theta}_2 - \left[m_2 \ddot{x} l_2 \cos(\theta_2) - m_2 \dot{x}\, \ddot{\theta}_2 l_2 \sin(\theta_2)\right] \tag{15}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \theta_2}\right) = m_2 \dot{x} l_2 \dot{\theta}_2 \sin(\theta_2) - m_2 l_2 g \sin(\theta_2)) \tag{16}$$

We can write by combining the equations:

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

After cancellation of terms, we get:

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 \ddot{x} l_2 \cos(\theta_2) + m_2 \dot{x} \dot{\theta}_2 l_2 \sin(\theta_2) = 0 \tag{17}$$

Combining equations [12],[14] and [17] for writing them in state space form. The non-linear state space form of these equations will be as follows:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dfrac{-m_1 g sin(\theta_1)\, cos(\theta_2) - m_2 g sin(\theta_2) cos(\theta_2) - m_1 l_1 \dot{\theta}_1^{\,2} sin(\theta_1) - m_2 l_2 (\dot{\theta}_2)^2 sin(\theta_2) + F}{M + m_1 + m_2 - m_1 cos^2\theta_1 - m_2 cos^2\theta_2} \\ \dot{\theta}_1 \\ \dfrac{-m_1 g sin(\theta_1)\, cos(\theta_2) - m_2 g sin(\theta_2) cos(\theta_2) - m_1 l_1 \dot{\theta}_1^{\,2} sin(\theta_1) - m_2 l_2 (\dot{\theta}_2)^2 sin(\theta_2) + F}{(M + m_1 + m_2 - m_1 cos^2\theta_1 - m_2 cos^2\theta_2) l_1} - \dfrac{g sin(\theta_1)}{l1} \\ \dot{\theta}_2 \\ \dfrac{-m_1 g sin(\theta_1)\, cos(\theta_2) - m_2 g sin(\theta_1) cos(\theta_2) - m_1 l_1 \dot{\theta}_1^{\,2} sin(\theta_1) - m_2 l_2 (\dot{\theta}_2)^2 sin(\theta_2) + F}{(M + m_1 + m_2 - m_1 cos^2\theta_1 - m_2 cos^2\theta_2) l_2} - \dfrac{g sin(\theta_2)}{l2} \end{bmatrix}$$

3. Linearization of Non-Linear System

Linearization involves finding the linear approximation of a function at a specific point, often achieved through the first-order Taylor expansion around that point. In the context of dynamical systems, linearization serves as a method to assess the local stability of an equilibrium point in a system of nonlinear differential equations or discrete dynamical systems. This technique enables the application of tools designed for the analysis of linear systems to understand the behaviour of a nonlinear function near a given point.

In the case of the derived equation of motion for the cart system with two pendulums and its representation in state-space form, the presence of sine and cosine components renders the equations nonlinear. Solving nonlinear equations can be challenging. To address this, we employ linearization by approximating the system around the equilibrium point where (x = 0), ($\theta_1 = 0$), and ($\theta_2 = 0$), as specified in the problem statement.

The procedure involves setting limiting conditions at equilibrium, allowing for a simplified analysis of the system's dynamics in the vicinity of the chosen equilibrium point.

$$\ddot{x} = \frac{1}{M + m_1 + m_2}\left(m_1 l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 - 0 - 0 + F\right)$$

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

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

Since θ_1 = 0, θ_1 = 0 and x = 0 we can assume the following:

1. $(\dot{\theta}_1)^2 \approx 0$
2. $(\dot{\theta}_2)^2 \approx 0$
3. $sin\,\theta_1 \approx \theta_1$
4. $sin\,\theta_2 \approx \theta_2$
5. $cos\,\theta_1 \approx 1$
6. $cos\,\theta_2 \approx 1$

With the following assumptions we have linearized the terms like sin and cos.

The updated state space equation after linearizing can be written as follows:

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

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-gm_1}{M} & 0 & \frac{-gm_1}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-g(M+m_{1)}}{Ml_1} & 0 & \frac{-gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-gm_1}{Ml_2} & 0 & \frac{-g(M+m_{2)}}{Ml_2} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \dot{\mathrm{x}} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\mathrm{M}} \\ 0 \\ \frac{1}{\mathrm{Ml}_1} \\ 0 \\ \frac{1}{\mathrm{Ml}_2} \end{bmatrix} F$$

4. Controllability

The obtained linearized system has to be checked for its controllability. The time-invariant linear state equation is controllable if and only if the n x nm controllability matrix satisfies

$$\text{rank}\ [Ac] = n, \text{ where } Ac = [B\ AB\ A^2B\ \ldots..\ A^{n-1}B]$$

In this case, n = 6.

The MATLAB code written to determine controllability of the system is as given below:

```
close all;
clear;
```

Defining symbols required to define the system

```
g = sym('g');
m1 = sym('m1');
m2 = sym('m2');
l1 = sym('l1');
l2 = sym('l2');
M = sym('M');
```

Defining system matrices

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


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

Determining Controllability Matrix

```
AB = A*B;
A2B = (A^2)*B;
A3B = (A^3)*B;
A4B = (A^4)*B;
A5B = (A^5)*B;


Ac = [B AB A2B A3B A4B A5B] ;
Ac_dup = [B AB A2B A3B A4B A5B] ;
disp("Controllability Matrix is: ");
disp(Ac);
det_Ac = simplify(det(Ac));
```

```
disp("Determinant of Controllability Matrix: ");
disp(det_Ac);
```

The Controllability Matrix obtained is:

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

where

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

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

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

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

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

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

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

The determinant of the controllability matrix is $-\frac{g^6\,(l_1-l_2)^2}{M^6\,{l_1}^6\,{l_2}^6}$

The system is said to be controllable when its controllability matrix has full rank. The rank of the matrix can be ascertained through its determinant. A non-zero determinant indicates full rank, implying controllability of the system defined by the matrix.

Case1: Determinant of the controllability matrix is non-zero.

We can infer from the determinant that if $l_1 \neq l_2$, the matrix will have full rank and will be controllable.

This becomes zero whenever $l_1 = l_2$.

Case 2: Determinant of the controllability matrix is zero.

The determinant becomes zero whenever $(l_1 = l_2 = 0)$ or $(l_1 = l_2)$, and hence the system becomes uncontrollable.

Hence the conditions for which the system is controllable is: $l_1 \neq l_2, m_1 > 0, m_2 > 0, M > 0$

5. LQR Controller

For linear time invariant systems, LQR computes the state-feedback control $u=-Kx$ that minimizes the quadratic cost function

$$J(K, x(0)) = \int_0^{\infty} x^T(t)Qx(t) + U^T(t)RU(t)\ dt$$

subject to the system dynamics $\dot{x} = Ax + BU$
where Q and R are positive symmetric definite matrices.

The optimal solution is given by the following:

$$K = -R^{-1}B^TP$$

Where P is the solution of the Algebraic Riccati Equation,

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

5.1. LQR Controller for linearized system

The inbuilt MATLAB function [K, S, P] = lqr(A, B, Q, R, N) computes the optimal control gain matrix K, along with the solution S to the corresponding algebraic Riccati equation and the closed-loop poles P. This is performed in the context of continuous-time state-space representations using matrices A and B.

The below code provides an illustrative example of using LQR control to stabilize and optimize the performance of a crane system. It demonstrates how tuning the weighting matrices Q and R influences the system's behaviour. The initial and controlled system responses are visualized to analyse stability and performance improvements.

The initial condition for the system is assumed to be x=0, $\theta_1 = 0.5$ and θ_2=0.6.

```
close all;
clear;

g = 9.8;
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
```

Defining system matrices after substituting the mass and cable lengths

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

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

Defining Controllability Matrix

```
AB = A*B;
A2B = (A^2)*B;
A3B = (A^3)*B;
A4B = (A^4)*B;
A5B = (A^5)*B;


Ac = [B AB A2B A3B A4B A5B] ;
disp("Controllability Matrix is: ");
disp(Ac);
det_Ac = det(Ac);
rank_Ac = rank(Ac);
disp("Determinant of Ac after substituting values is: ");
disp(det_Ac);
disp("Rank of Ac after substituting values is: ");
disp(rank_Ac);
C = eye(6);
D = [0; 0; 0; 0; 0; 0];
```

Assume initial Q and R values

```
Q = eye(6);
R = 0.01;
[K,S,P] = lqr(A,B,Q,R);
K1 = K(1);
At = A - B*K;
%Defining system without a controller
sys1 = ss(A,B,C,D);
%Defining system after adding LQR Controller
sys2 = ss(A-B*K,B,C,D);
e = eig((A-B*K));
disp(e);
x_0 = [0;0;0.5;0;0.6;0];
initial(sys1,x_0);
xlim([0 500]);
title("Without any controller")
initial(sys2,x_0);
xlim([0 500]);
title("With LQR controller, R = 0.01")
%Changing Q and R values
Q = 10*eye(6);
R = 0.001;
[K,S,P] = lqr(A,B,Q,R);


%Defining system after adding LQR Controller
sys3 = ss(A-B*K,B,C,D);
```

```
x_0 = [0;0;0.5;0;0.6;0];
initial(sys3,x_0);
xlim([0 500]);
title("With LQR controller, R = 0.001")
%Changing Q and R values
Q = 100*eye(6);
R = 0.0001;
[K,S,P] = lqr(A,B,Q,R);
K1 = K(1);
At = A - B*K;


%Defining system after adding LQR Controller
sys4 = ss(A-B*K,B,C,D);


e = eig((A-B*K));
x_0 = [0;0;0.5;0;0.6;0];


initial(sys4,x_0);
xlim([0 500]);
title("With LQR controller, R = 0.0001")
```

Given M= 1000 kg, $m_1 = m_2 = 100$kg, $l_1 = 20m, l_2 = 10m$, the system is controllable as the values of l_1 and l_2 are not the same.

From the LQR function, we get gain matrix K,

K = [10.0000 155.1171 -12.6608 -224.9890 -6.2965 -112.9358]

P =
```
-0.0006 + 0.7282i
-0.0006 - 0.7282i
-0.0010 + 1.0425i
-0.0010 - 1.0425i
-0.0646 + 0.0645i
-0.0646 - 0.0645i
```

In the LQR cost function, Q determines the system's performance, influencing its speed of stabilization, while R governs the actuator energy, indicating the amount of torque or energy expended in achieving desired outcomes.

Higher Q values penalize state vectors, reducing errors and promoting faster stability. Conversely, higher R values minimize energy expenditure.

Figure 2 represents the system's response to initial conditions without any feedback, and it can be observed that the output does not converge.

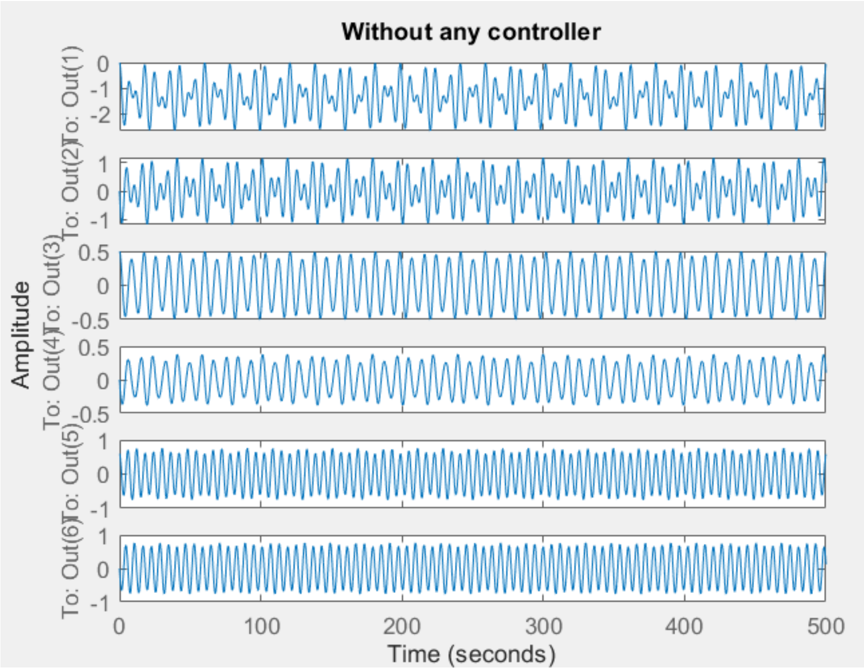

Figure 2. System response to initial conditions

Figure 3 represents the system response after LQR Controller has been applied. Initially R is taken to be 0.01 and Q as identity matrix. It is observed that the system output still does not converge to zero.

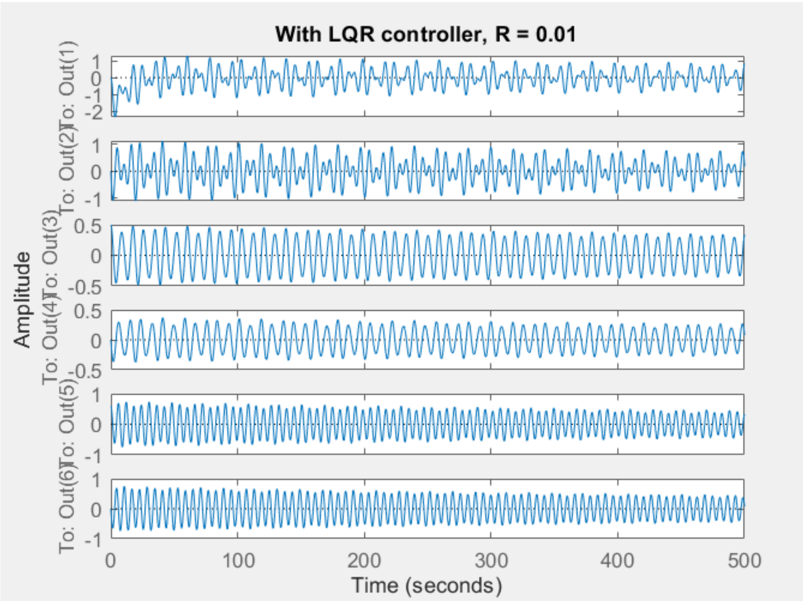

Figure 3. System response to initial condition after LQR Controller has been applied.

We further increase the $Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$, and decrease R = 0.001. In this case the system output reaches zero after about 300s.

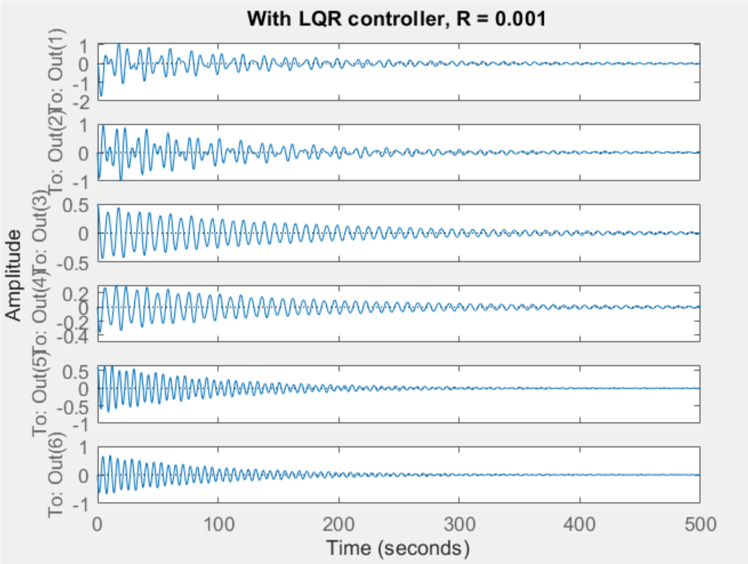

Figure 4. System response to initial condition with R = 0.001

Further increasing Q to $\begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$, and decrease R = 0.0001.

The system reaches zero faster, in about 100 seconds.

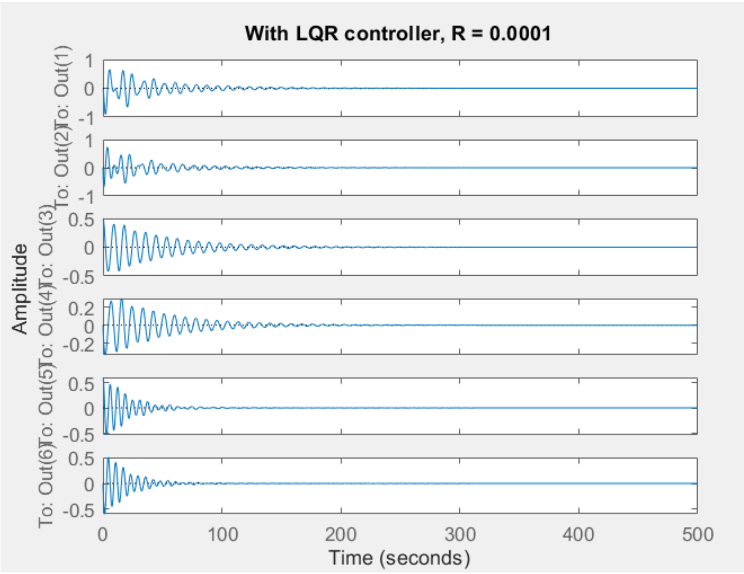

Figure 5. System response to initial condition with R = 0.0001

For the objective of stabilizing the system and disregarding energy considerations, decreasing R, and increasing Q in the problem statement accelerates the convergence of state vectors to zero.

5.2. LQR Controller for original non-linear system

```
% Clear the workspace and close all figures
clc; clear; close all;


t_span = 0:0.01:1000;
% Solve the ODE and obtain the state vector
[t, x] = ode45(@nonlinear, t_span, [0; 0; 0.5; 0; 0.6; 0]);


figure;
plot(t, x(:, 1),  'DisplayName', 'x');
hold on;
plot(t, x(:, 2),  'DisplayName', 'x_d');
plot(t, x(:, 3),  'DisplayName', 'theta1');
plot(t, x(:, 4),  'DisplayName', 'theta1_d');
plot(t, x(:, 5),  'DisplayName', 'theta2');
plot(t, x(:, 6),  'DisplayName', 'theta2_d');


xlim([0, 200]);
xlabel('Time');
ylabel('State vector');
legend('Location', 'best');

%Lyapunov indirect method for stability analysis
g = 9.8;
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
A = [0 1 0 0 0 0;
        0 0 ((-g*m1)/M) 0 ((-g*m2)/M) 0;
        0 0 0 1 0 0;
        0 0 ((-g*(M+m1))/(M*l1)) 0 -((g*m2)/(M*l1)) 0;
        0 0 0 0 0 1;
        0 0 ((-g*m1)/(M*l2)) 0 ((-g*(M+m2))/(M*l2)) 0 ];


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


Q = 100 * eye(6);
R = 0.001;
```

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


e = eig(A-B*K);
disp(e);

function dx = nonlinear(t, x)
    dx = zeros(6,1);
    g = 9.8;
    M = 1000;
    m1 = 100;
    m2 = 100;
    l1 = 20;
    l2 = 10;

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


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


    Q = 100 * eye(6);
    R = 0.001;


    [K,~,~] = lqr(A, B, Q, R);
    F = -K*x;


    dx(1) = x(2);
    dx(2) = (-m1*g*sin(x(3))*cos(x(5)) - m2*g*sin(x(5))*cos(x(5)) -
m1*l1*x(4)^2*sin(x(3))-m2*l2*x(6)^2*sin(x(5))+F)/(M+m1+m2-m1*cos(x(3))^2-
m2*cos(x(5))^2);
    dx(3) = x(4);
    dx(4) = (-m1*g*sin(x(3))*cos(x(5)) - m2*g*sin(x(5))*cos(x(5)) -
m1*l1*x(4)^2*sin(x(3))-m2*l2*x(6)^2*sin(x(5))+F)/(l1*(M+m1+m2-m1*cos(x(3))^2-
m2*cos(x(5))^2)) - (g*sin(x(3)))/l1;
    dx(5) = x(6);
    dx(6) = (-m1*g*sin(x(3))*cos(x(5)) - m2*g*sin(x(5))*cos(x(5)) -
m1*l1*x(4)^2*sin(x(3))-m2*l2*x(6)^2*sin(x(5))+F)/(l2*(M+m1+m2-m1*cos(x(3))^2-
m2*cos(x(5))^2)) - (g*sin(x(5)))/l2;
```

```
end
```

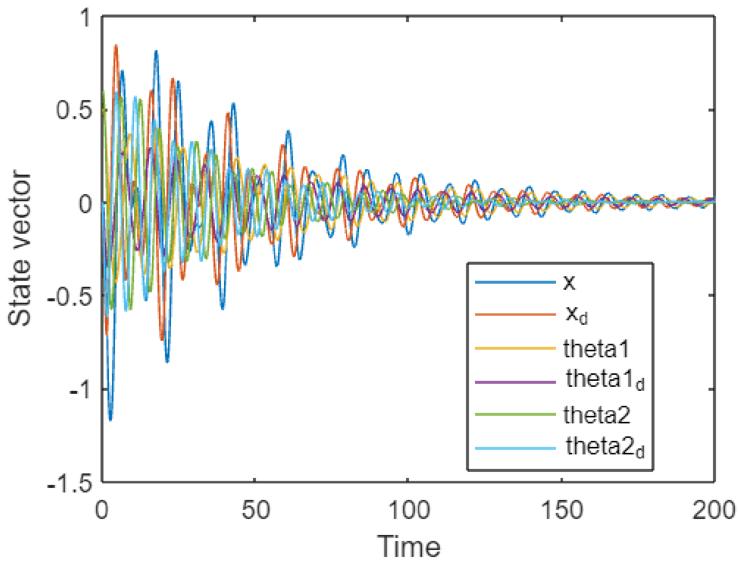

Figure 6. Non-linearized system response

5.3. Lyapunov's indirect method for stability analysis

Instead of seeking a Lyapunov function directly applicable to the nonlinear system, an alternative approach involves linearizing the system around the origin. By utilizing quadratic Lyapunov functions for the linearized system, one can aim to establish "local" stability of the origin. To be more precise, if the eigenvalues of the A matrix of the linearized system lie in the open left-half plane (LHP), it implies that the nonlinear system is locally asymptotically stable (L.A.S.).

The A matrix obtained after applying controller is (A-BK), and the eigen values of this matrix are:

$$\begin{aligned}
\lambda_1 &= -0.3838 + 0.3543i \\
\lambda_2 &= -0.3838 - 0.3543i \\
\lambda_3 &= -0.0298 + 1.0342i \\
\lambda_4 &= -0.0298 - 1.0342i \\
\lambda_5 &= -0.0161 + 0.7209i \\
\lambda_6 &= -0.0161 - 0.7209i
\end{aligned}$$

And as it can be observed, the real values all lie in the negative half plane.

6. Observability

The LTI system,

$$\dot{x(t)} = Ax(t) + Bu(t) \;, x(0) = 0$$

$$y(t) = Cx(t) + Du(t)$$

is called "observable" if the knowledge of u(t) and y(t) over some finite time interval $0 \leq t \leq t_f$ is enough to uniquely determine x0.

If the pair (A^T, C^T) is controllable then we say that (A, C) is observable.

The observability matrix is given as Ao = [C^T A^TC^T ….. $(A^T)^{n-1}C^T$]. And the system is said to be observable when the matrix Ao has full rank.

```
clear; close all;
g = 9.8;
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;


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


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


C = eye(6);
D = 0;


Q = 100*eye(6);
R = 0.001;
```

For output vector x(t), C is

```
disp("Output vector is x(t), C:");
```

```
Output vector is x(t), C:
```

```
C1 = [1 0 0 0 0 0];
disp(C1);
```

$$C1 = [1\,0\,0\,0\,0\,0]$$

For output vector {x(t),$\theta 1(t), \theta 2(t)\}$, C is

```
disp("Output vector is x(t),theta1(t), theta2(t)");
```

```
Output vector is x(t), theta1(t), theta2(t)
```

```
C2 = [1 0 0 0 0 0;
      0 0 1 0 0 0;
      0 0 0 0 1 0];
disp(C2);
```

$$C\_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For output vector {x(t),$\theta 2(t)\}$, C is

```
disp("Output vector is x(t),theta2(t)");
```

```
Output vector is x(t),theta2(t)
```

```
C3 = [1 0 0 0 0 0;
      0 0 0 0 1 0];
disp(C3);
```

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

For output vector $\{\theta 1(t), \theta 2(t)\}$, C is

```
disp("Output vector is theta1(t),theta2(t)");
```

```
Output vector is theta1(t),theta2(t)
```

```
C4 = [0 0 1 0 0 0;
      0 0 0 0 1 0];
disp(C4);
```

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

Define the observability matrices for the above 4 outputs

```
AO1 = [C1.' A.'*C1.' (A.')^2*C1.' (A.')^3*C1.' (A.')^4*C1.' (A.')^5*C1.'];
AO2 = [C2.' A.'*C2.' (A.')^2*C2.' (A.')^3*C2.' (A.')^4*C2.' (A.')^5*C2.'];
AO3 = [C3.' A.'*C3.' (A.')^2*C3.' (A.')^3*C3.' (A.')^4*C3.' (A.')^5*C3.'];
AO4 = [C4.' A.'*C4.' (A.')^2*C4.' (A.')^3*C4.' (A.')^4*C4.' (A.')^5*C4.'];


rank1 = rank(AO1);
rank2 = rank(AO2);
rank3 = rank(AO3);
rank4 = rank(AO4);
```

If the observability matrix is of full rank, then the system is observable.

```
disp("Rank of A01:");
```

```
Rank of A01:
```

```
disp(rank1);
```

```
     6
disp("Rank of A02:");
Rank of A02:
disp(rank2);
     6
disp("Rank of A03:");
Rank of A03:
disp(rank3);
     6
disp("Rank of A04:");
Rank of A04:
disp(rank4);
     4
```

For outputs x(t), {x(t),$\theta 1(t), \theta 2(t)$}, {x(t),$\theta 2(t)$}, the system is observable as the observability matrices have full rank for the above 3 cases.

7. Luenberger Observer

A Luenberger Observer, also known as a state observer or Luenberger observer, is a mathematical tool used in control theory to estimate the internal state variables of a system from the system's output. The Luenberger Observer is designed to provide an estimate of the unmeasured state variables by using the system dynamics and the available output measurements.

The general form of a Luenberger Observer is given by the following state-space representation:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\big(y(t) - C\hat{x}(t)\big), x(0) = 0$$

Where,

- $\hat{x}$ is the estimated state error.
- A, B, and C are matrices defining the system dynamics.
- u is the control input.
- L is the observer gain matrix which is designed to place the observer poles at desired locations for stability and performance.

7.1. Luenberger Observer for Linear system

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


%Assume the poles such that they are in negative half plane
observer_poles = [-1,-2,-3,-4,-5,-6];
% Choose the observer gain matrix L
L1 = place(A', C1', observer_poles);
disp(L1);
```

```
   1.0e+03 *
```

L_1 = [0.0210   0.1734   -2.9329   0.0792   2.2176   -1.4496]

```
L2 = place(A', C2', observer_poles);
disp(L2);
```

$$L_2 = \begin{bmatrix} 8.5631 & 17.5219 & -0.9140 & -4.1173 & 0.0 & 0.0 \\ -0.8851 & -4.9474 & 9.4369 & 20.9390 & 0 & -0.0980 \\ 0 & -0.9800 & 0 & -0.0491 & 3.0 & 0.9220 \end{bmatrix}$$

```
L3 = place(A', C3', observer_poles);
disp(L3);
```

$$L_3 = \begin{bmatrix} 13.0743 & 56.2564 & -89.1734 & -20.0624 & 0.3520 & 3.4792 \\ -0.8243 & -8.4778 & 19.7841 & 10.9530 & 7.9257 & 13.2136 \end{bmatrix}$$

For first observable system:

```
A_c1 = [(A-B*K) B*K ; zeros(size(A)) (A-(L1'*C1))];
B_c1 = [B;B];
C_c1 = [C1 zeros(size(C1))];
```

```
sys1 = ss(A_c1,B_c1,C_c1,D);
x_0 = [0;0;0.5;0;0.6;0;0;0;0;0;0;0];
initial(sys1,x_0);
```

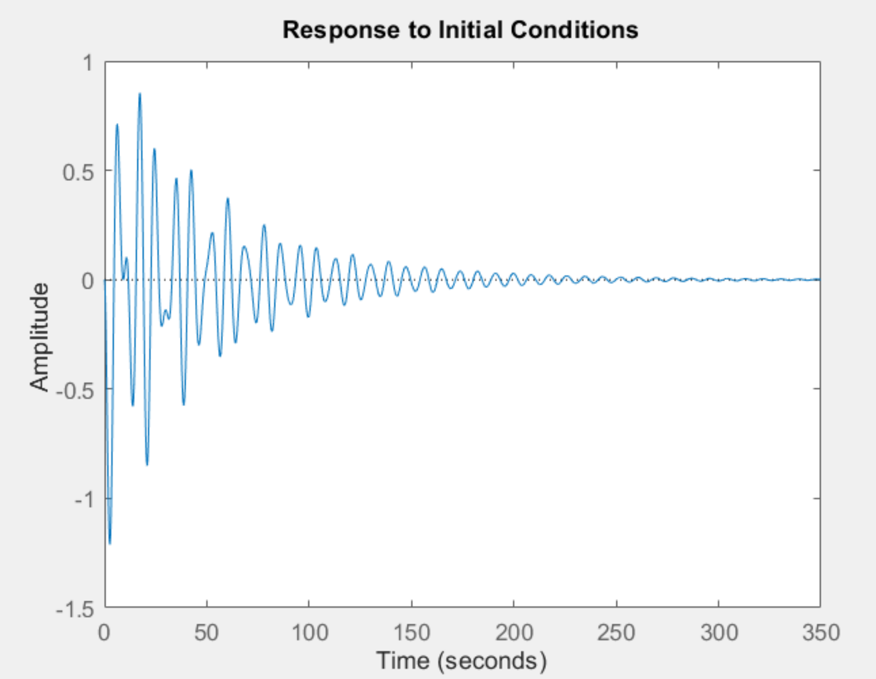

Figure 7. System response to initial condition when output is x(t)

```
step(sys1);
```

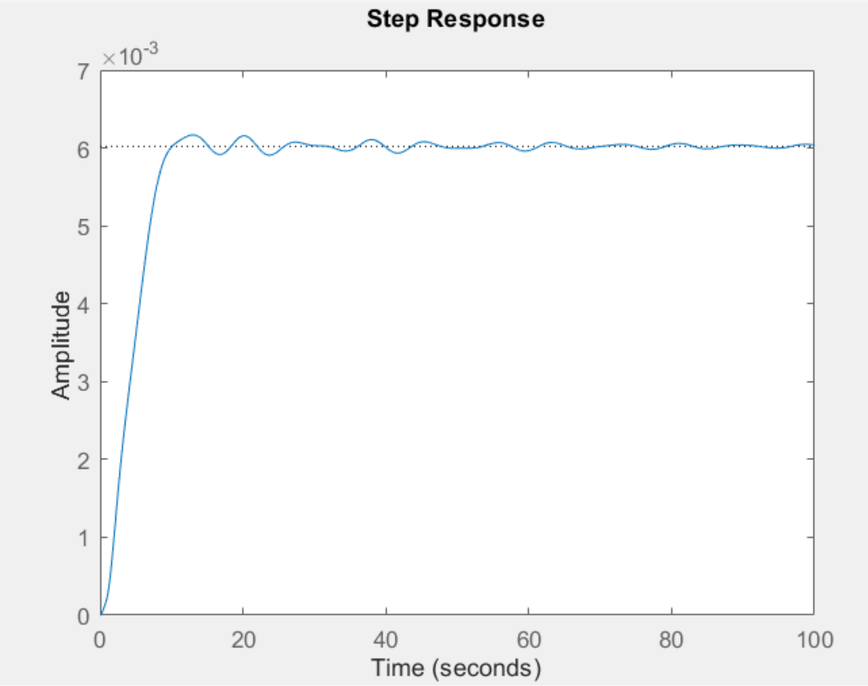

Figure 8. Step response when output is x(t)

For second observable system:

```
A_c2 = [(A-B*K) B*K ; zeros(size(A)) (A-(L2'*C2))];
B_c2 = [B;B];
C_c2 = [C2 zeros(size(C2))];
```

```
sys2 = ss(A_c2,B_c2,C_c2,D);
initial(sys2,x_0);
```

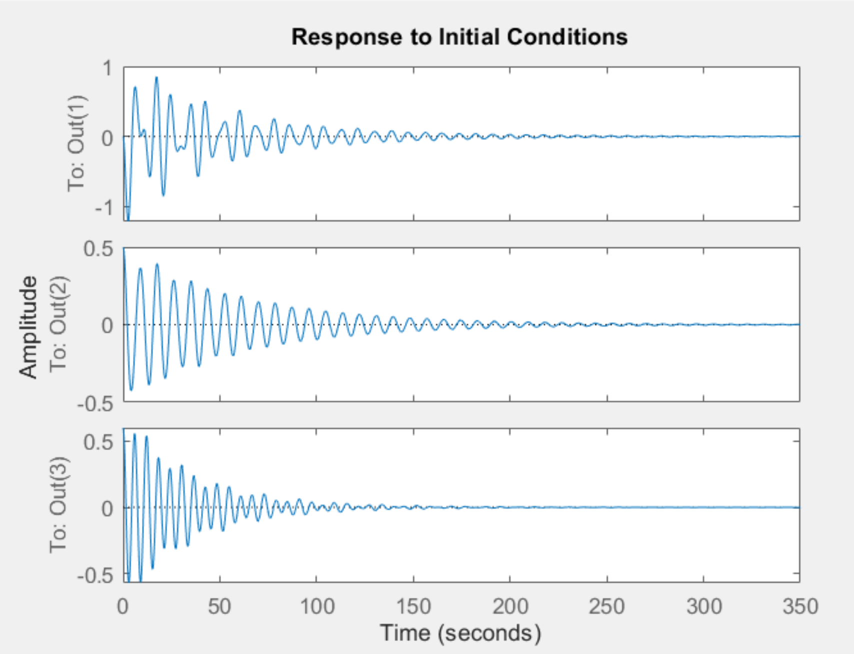

Figure 9. System response to initial condition when output is $\{x(t), \theta 1(t), \theta 2(t)\}$

```
step(sys2);
```

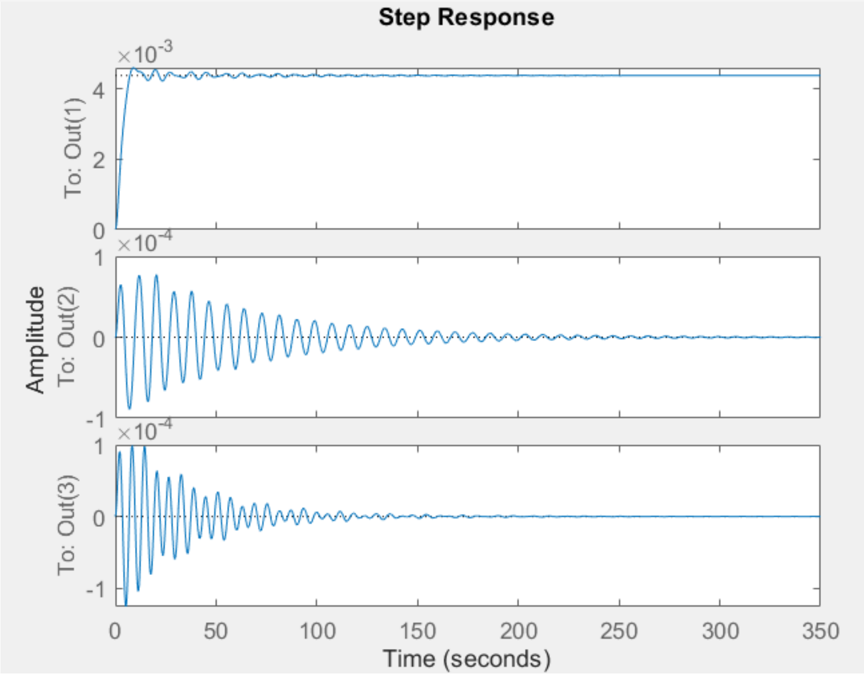

Figure 10. Step response when output is $\{x(t), \theta 1(t), \theta 2(t)\}$

For third observable system:

```
A_c3 = [(A-B*K) B*K ; zeros(size(A)) (A-(L3'*C3))];
B_c3 = [B;B];
C_c3 = [C3 zeros(size(C3))];


sys3 = ss(A_c3,B_c3,C_c3,D);
```

```
initial(sys3,x_0);
```

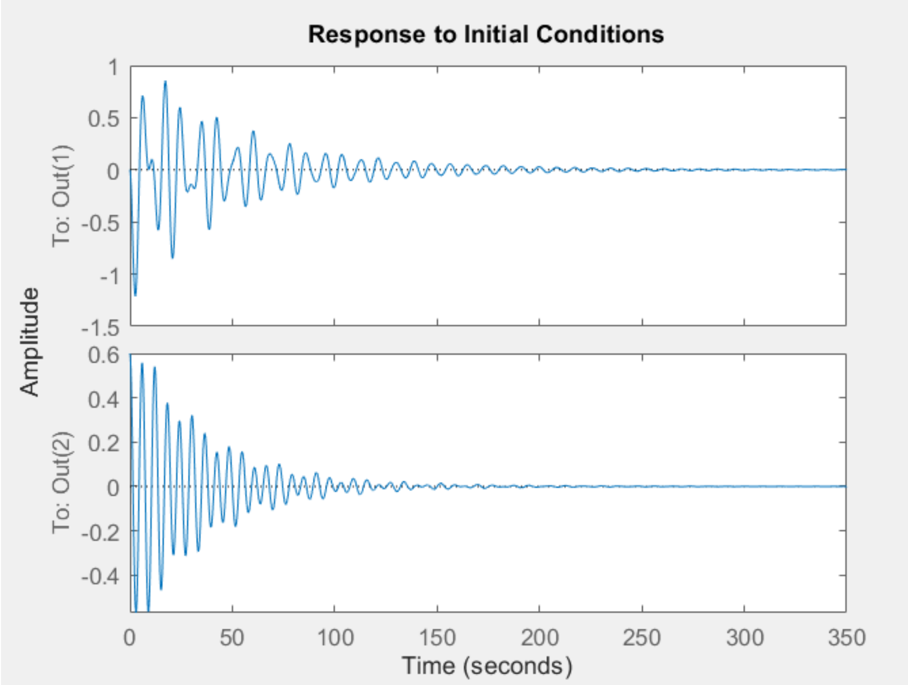

Figure 11. System response to initial condition when output is $\{x(t), \theta 2(t)\}$

```
step(sys3);
```

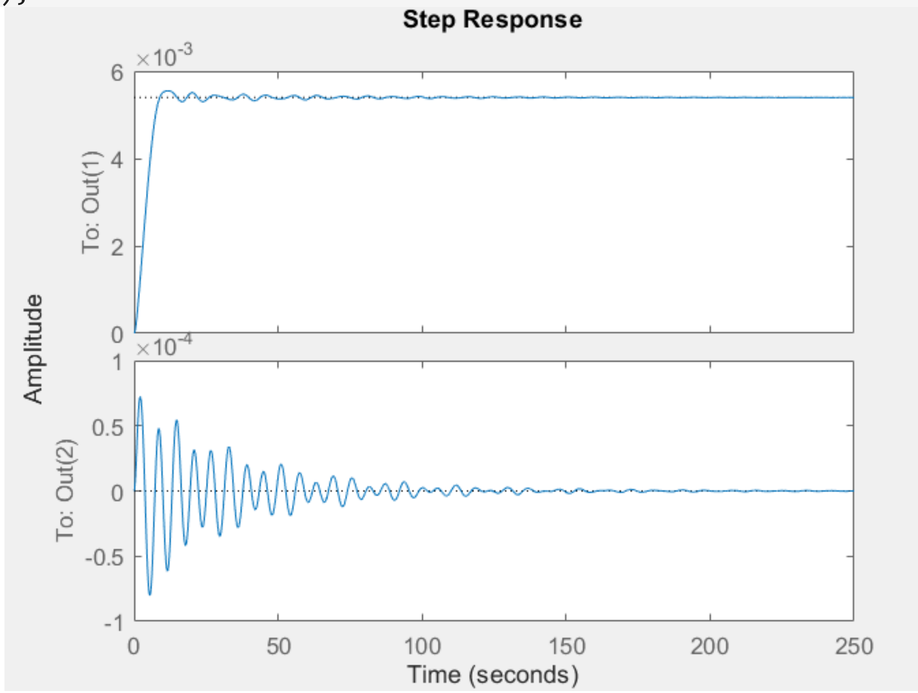

Figure 12. Step response when output is $\{x(t), \theta 2(t)\}$

7.2. Luenberger Observer for Non-linear system

```
% Defining system matrices
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;


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



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


Q = 100*eye(6);


R = 0.001
```

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

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


D = 0;


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

```
    6
```

```
C1 = [1 0 0 0 0 0];


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


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

Assume the desired poles for the luenberger observer, and calculate the observer gain for the three different outputs.

```
L_poles = [-1; -2; -3; -4; -5; -6];


L1 = place(A', C1', L_poles);
L2 = place(A', C2', L_poles);
L3 = place(A', C3', L_poles);


x_0 = [0;0;0.5;0;0.6;0];
```

Luenberger Observer for output x(t)

```
t_span = 0:0.01:500;
[t,x_L1] = ode45(@(t,x)nonlinear_luen(t,x,-K*x,L1,C1),t_span,x_0);
figure
plot(t,x_L1(:,1))
grid
xlabel('Time')
ylabel('State')
title('For Output x(t)')
```

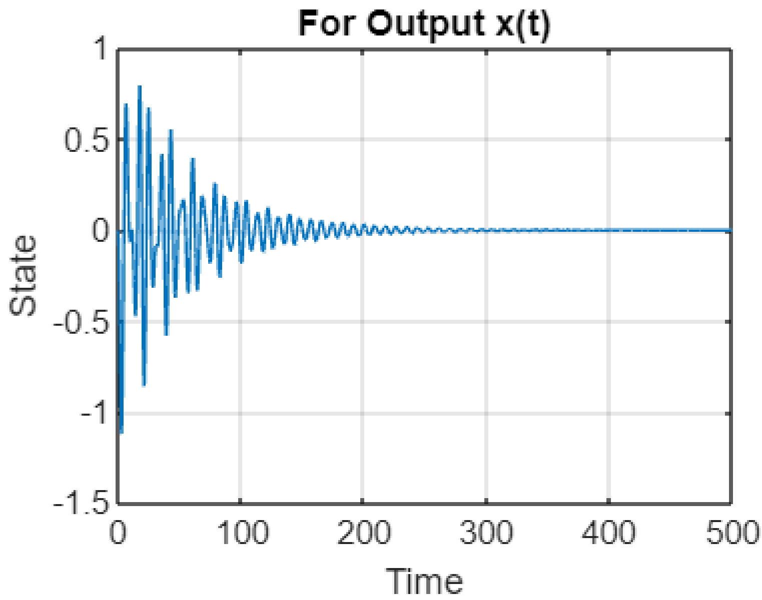

Figure 13. Nonlinear system response for output x(t)

Luenberger Observer for output {x(t), $\theta 2(t)$}

```
figure
hold on
[t,x_L2] = ode45(@(t,x)nonlinear_luen1(t,x,-K*x,L2',C2),t_span,x_0);
plot(t,x_L2(:,1))
plot(t,x_L2(:,5))
grid
xlabel('Time')
ylabel('State')
title('For output x(t),theta2')
legend({'x', 'theta2'},'Location','Best')
hold off
```

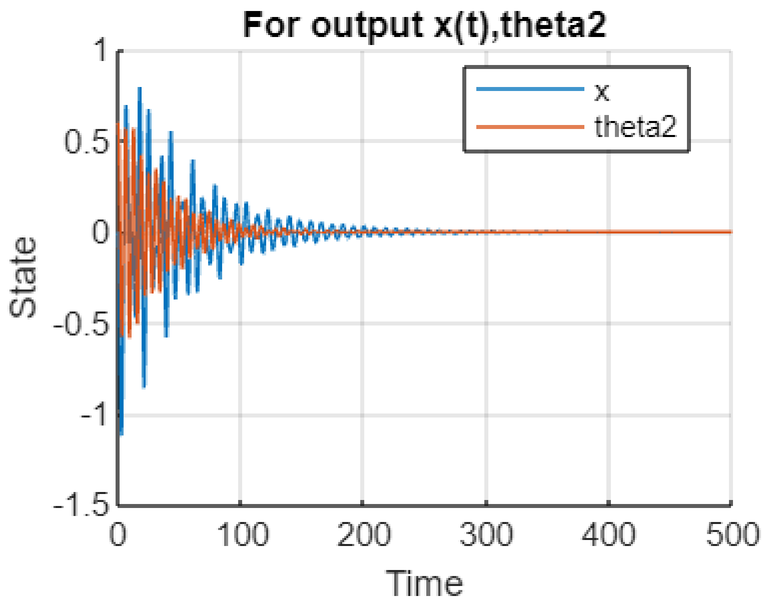

Figure 14. Nonlinear system response for output {x(t),$\theta 2(t)$}

Luenberger Observer for output {x(t),$\theta 1(t)$, $\theta 2(t)$}

```
figure
hold on
[t,x_L3] = ode45(@(t,x)nonlinear_luen2(t,x,-K*x,L3',C3),t_span,x_0);
plot(t,x_L3(:,1))
plot(t,x_L3(:,3))
plot(t,x_L3(:,5))
grid
xlabel('Time')
ylabel('State')
title('For output x(t),theta1,theta2')
legend({'x', 'theta1', 'theta2'},'Location','Best')
hold off
```

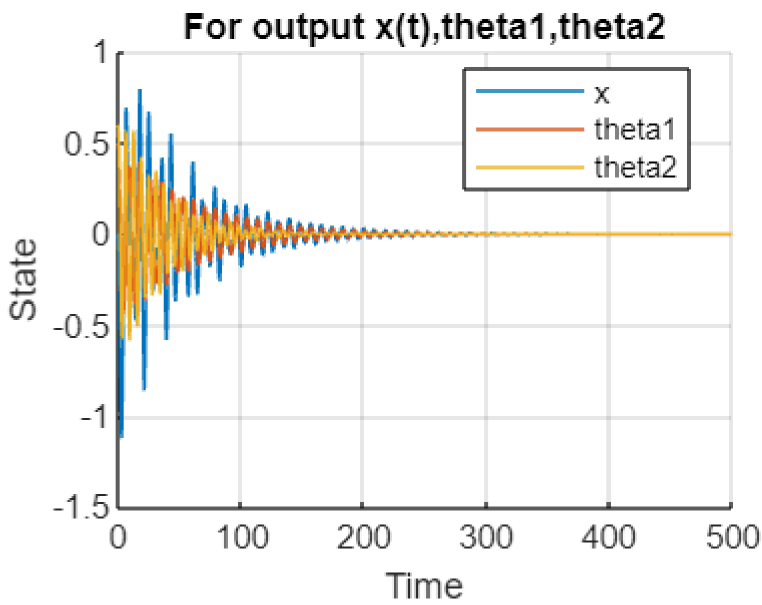

Figure 15. Nonlinear system response for output {x(t),$\theta 1(t)$, $\theta 2(t)$}

```
function dx = nonlinear_luen(t,X,F,L,C)
dx = zeros(6,1);
% Declaring the values of system variables
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;
x = X(1);
x_d = X(2);
theta1 = X(3);
theta1_d = X(4);
theta2 = X(5);
theta2_d = X(6);
obs = L*(x-C*X);
dx(1) = x_d + obs(1);
dx(2) = (F-((m1*sin(theta1)*cos(theta1))+(m2*sin(theta2)*cos(theta2)))*g -
(l1*m1*(dx(3)^2)*sin(theta1)) - (l2*m2*(dx(5)^2)*sin(theta2)))/(m1+m2+M-
(m1*(cos(theta1)^2))-(m2*(cos(theta2)^2)))+obs(2);
dx(3) = theta1_d+obs(3);
dx(4) = ((cos(theta1)*dx(2)-g*sin(theta1))/l1) + obs(4);
dx(5) = theta2_d + obs(5);
dx(6) = (cos(theta2)*dx(2)-g*sin(theta2))/l2 + obs(6);
end


function dx = nonlinear_luen1(t,X,F,L,C)
dx = zeros(6,1);
% Declaring the values of system variables
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;


x = X(1);
x_d = X(2);
theta1 = X(3);
theta1_d = X(4);
theta2 = X(5);
theta2_d = X(6);
y3 = [x; theta2];
obs1 = L*(y3-C*X);
dx(1) = x_d + obs1(1);
dx(2) = (F-((m1*sin(theta1)*cos(theta1))+(m2*sin(theta2)*cos(theta2)))*g -
(l1*m1*(dx(3)^2)*sin(theta1)) - (l2*m2*(dx(5)^2)*sin(theta2)))/(m1+m2+M-
(m1*(cos(theta1)^2))-(m2*(cos(theta2)^2)))+obs1(2);
dx(3) = theta1_d+obs1(3);
dx(4) = ((cos(theta1)*dx(2)-g*sin(theta1))/l1) + obs1(4);
```

```
dx(5) = theta2_d + obs1(5);
dx(6) = (cos(theta2)*dx(2)-g*sin(theta2))/l2 + obs1(6);
end


function dx = nonlinear_luen2(t,X,F,L,C)
dx = zeros(6,1);
% Declaring the values of system variables
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.81;


x = X(1);
x_d = X(2);
theta1 = X(3);
theta1_d = X(4);
theta2 = X(5);
theta2_d = X(6);
y4 = [x; theta1; theta2];
obs2 = L*(y4-C*X);


dx(1) = x_d + obs2(1);
dx(2) = (F-((m1*sin(theta1)*cos(theta1))+(m2*sin(theta2)*cos(theta2)))*g -
(l1*m1*(dx(3)^2)*sin(theta1)) - (l2*m2*(dx(5)^2)*sin(theta2)))/(m1+m2+M-
(m1*(cos(theta1)^2))-(m2*(cos(theta2)^2)))+obs2(2);
dx(3) = theta1_d+obs2(3);
dx(4) = ((cos(theta1)*dx(2)-g*sin(theta1))/l1) + obs2(4);
dx(5) = theta2_d + obs2(5);
dx(6) = (cos(theta2)*dx(2)-g*sin(theta2))/l2 + obs2(6);
end
```

8. LQG Controller

The Linear Quadratic Gaussian Controller is designed for LTI systems with additive white Gaussian noise. LQG is a combination of two main components: LQR for state feedback and Kalman filter for state estimation.

$$\dot{x}(t) = Ax(t) + B_K u_K(t) + B_D u_D(t)$$

$$y(t) \ = \ Cx(t) \ + \ V(t)$$

$u_D(t)$ is the process noise and $V(t)$ is the measure noise. They are both zero mean white gaussian processes with covariance $\sum_D$ *and* $\sum_V$.

The optimal solution is L = $PC^T\sum_V{}^{-1}$, where P is the solution of $A^T P + PA^T + B_D \sum_D^{-1} B_D^T - PC^T \sum_v^{-1} CP = 0$

8.1. LQG for Linear System

The smallest output vector is chosen to be x(t), and the LQG is designed for the same.

```
close all;
clear;


% Given system parameters
g = 9.8;
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;


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


B = [0;
    (1/M);
    0;
    (1/(M*l1));
    0;
    (1/(M*l2))];
Q = 100*eye(6);
R = 0.001;
C = eye(6);
D = 0;
```

```
C1 = [1 0 0 0 0 0];


x0 = [0;0;0.5;0;0.6;0;0;0;0;0;0;0];


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


vd=0.5*eye(6); %process noise
vn=1; %measurement noise
```

Defining K for all the observable systems

```
K1 = lqr(A',C1',vd,vn);
K2 = lqr(A',C2',vd,vn);
K3 = lqr(A',C3',vd,vn);
```

For first observable system:

```
A_c1 = [(A-B*K) B*K ; zeros(size(A)) (A-(K1'*C1))];
B_c1 = [B;B];
C_c1 = [C1 zeros(size(C1))];


sys1 = ss(A_c1,B_c1,C_c1,D);
figure;
initial(sys1,x0);
```

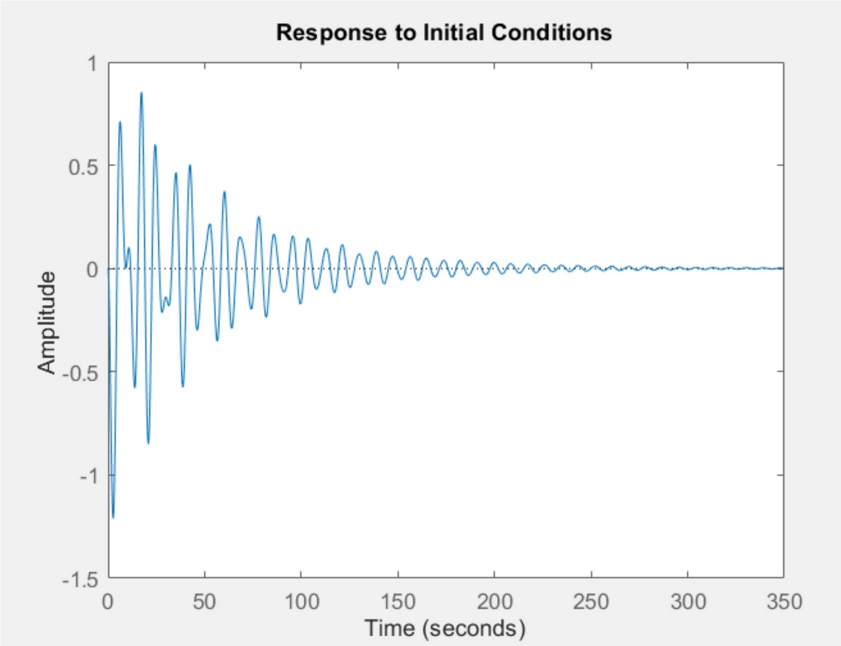

Figure 16. LQG Controller for output x(t)

```
figure;
step(sys1);
```

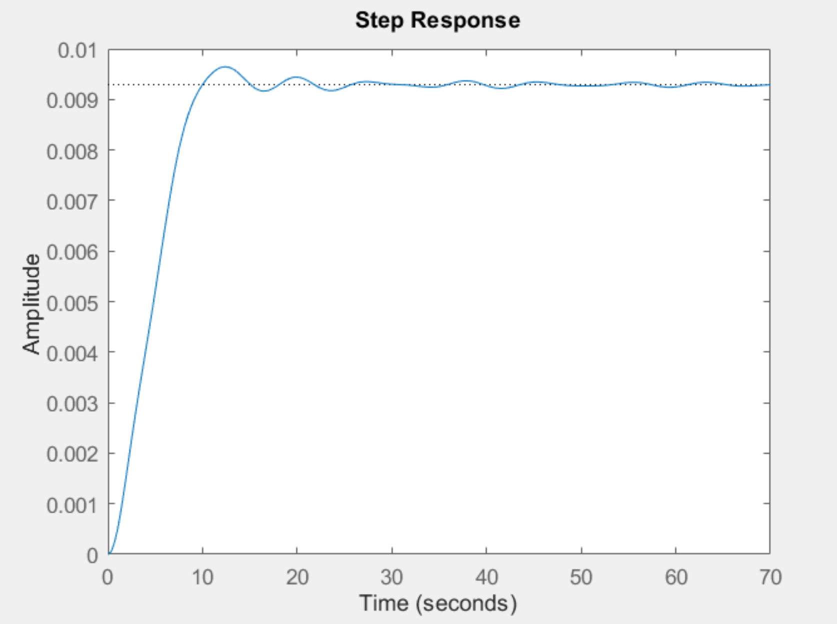

Figure 17. LQG Controller step response for output x(t)

8.2. LQG for Non-linear system:

```
% Clear the workspace and close all figures
clc; clear; close all;
t_span = 0:0.01:100;
% Solve the ODE and obtain the state vector
[t, x] = ode45(@nonlinear_lqg, t_span, [0; 0; 0.5; 0; 0.6; 0;0;0;0;0;0;0]);



figure;
plot(t, x(:, 1),  'DisplayName', 'x_1');
hold on;
plot(t, x(:, 2),  'DisplayName', 'x_2');
plot(t, x(:, 3),  'DisplayName', 'x_3');
plot(t, x(:, 4),  'DisplayName', 'x_4');
plot(t, x(:, 5),  'DisplayName', 'x_5');
plot(t, x(:, 6),  'DisplayName', 'x_6');
plot(t, x(:, 7),  'DisplayName', 'x_7');
plot(t, x(:, 8),  'DisplayName', 'x_8');
plot(t, x(:, 9),  'DisplayName', 'x_9');
plot(t, x(:, 10), 'DisplayName', 'x_{10}');
plot(t, x(:, 11), 'DisplayName', 'x_{11}');
plot(t, x(:, 12), 'DisplayName', 'x_{12}');


xlabel('Time');
ylabel('State Value');
title('State Evolution Over Time');
```

```
legend('Location', 'Best');
grid on;
hold off;
```

```
function dx = nonlinear_lqg(t, x)
    dx = zeros(6,1);
    g = 9.8;
    M = 1000;
    m1 = 100;
    m2 = 100;
    l1 = 20;
    l2 = 10;

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

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



    Q = 10 * eye(6);
    R = 0.001;


    C1 = [1 0 0 0 0 0];


    [K,~,~] = lqr(A, B, Q, R);
    F = -K*x(1:6);


    vd=0.3*eye(6); %process noise
    vn=1; %measurement noise

    K1 = lqr(A',C1',vd,vn);
    sd = (A-K1'*C1)*x(7:12);


    dx(1) = x(2);
```

```
    dx(2) = (-m1*g*sin(x(3))*cos(x(5)) - m2*g*sin(x(5))*cos(x(5)) -
m1*l1*x(4)^2*sin(x(3))-m2*l2*x(6)^2*sin(x(5))+F)/(M+m1+m2-m1*cos(x(3))^2-
m2*cos(x(5))^2);
    dx(3) = x(4);
    dx(4) = (-m1*g*sin(x(3))*cos(x(5)) - m2*g*sin(x(5))*cos(x(5)) -
m1*l1*x(4)^2*sin(x(3))-m2*l2*x(6)^2*sin(x(5))+F)/(l1*(M+m1+m2-m1*cos(x(3))^2-
m2*cos(x(5))^2)) - (g*sin(x(3)))/l1;
    dx(5) = x(6);
    dx(6) = (-m1*g*sin(x(3))*cos(x(5)) - m2*g*sin(x(5))*cos(x(5)) -
m1*l1*x(4)^2*sin(x(3))-m2*l2*x(6)^2*sin(x(5))+F)/(l2*(M+m1+m2-m1*cos(x(3))^2-
m2*cos(x(5))^2)) - (g*sin(x(5)))/l2;
    dx(7) = x(2) - x(10);
    dx(8) = dx(2) - sd(2);
    dx(9) = x(4) - x(11);
    dx(10) = dx(4) - sd(4);
    dx(11) = x(6) - x(12);
    dx(12) = dx(6) - sd(6);
end
```

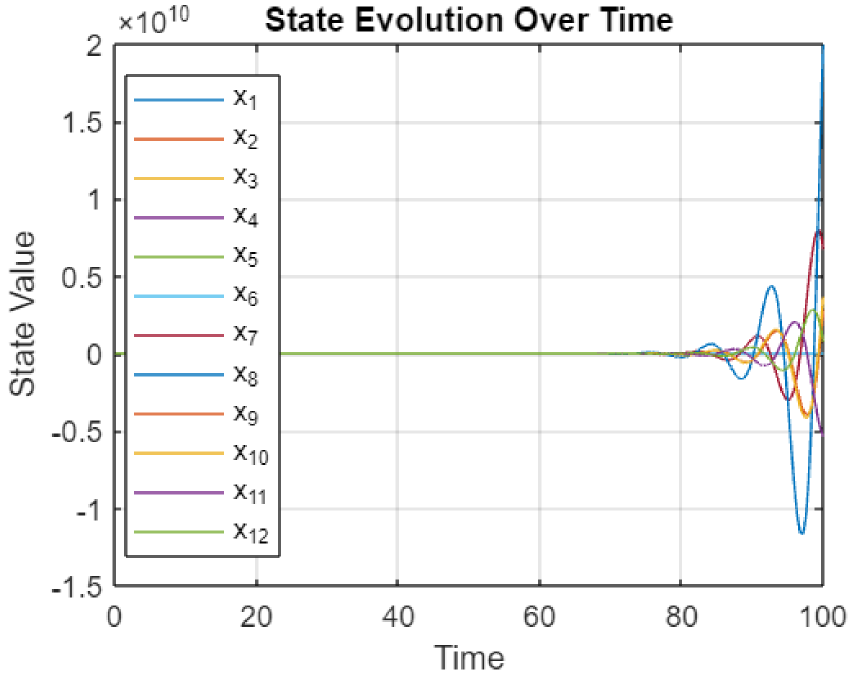

Figure 18. LQG for nonlinear system

To reconfigure the controller to asymptotically track a constant reference x, we have to minimize the following cost:

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

If U_∞ exists such that: $Ax_d + B_k U_\infty \; = \; 0$
The system can reject all Gaussian noise or disturbance applied on the cart.

Note: The MATLAB files have been run and the desired output has been obtained, which is shown in this video: [MATLAB Simulation]