Project 2

University of Maryland

December 2021

Instructors:
Waseem Ansar Malik

Students:
Bharadwaj Chukkala (118341705)

Sri Sai Charan Velisetti (117509755)

Semester:
Fall 2021

Course code:
ENPM667

Contents

1 Introduction

- The project focuses on creating LQR and LQG controllers for a crane which has two loads suspended to it via cables attached to the bottom of the crane. We know that these loads have different masses m1, m2 and the cables they are suspended from have different lengths l1 and l2.
- In this project we will start with finding the equations of motion for the system using Lagrangian method. The continuation step here would be to get the non linear state space representation of the system. The next step would be to linearize the system around an equilibrium point and write the state space representation of this linearized system.
- We then move onto obtaining the controllability conditions based out of M, m_1, m_2, l_1, l_2.
- After the controllability conditions are obtained we will be equipped with all that is needed to design an LQR controller for the crane and load system. Before we start with designing the LQR controller, the system needs to be checked whether it is controllable or not, in this case the system is controllable, then the LQR controller is designed. The simulation responses are recorded for two scenarios by applying the LQR controller to the original non linear system and the linearized system. We simulate the responses by adjusting the LQR parameters until we get the suitable response and then we perform Lyapunov analysis of this closed loop system to certify the stability.
- Now after designing an LQR controller for the system, we consider the parameters that we used to obtain controllability conditions and determine if the system observability for some output vectors.
- The problem statement for the project continues to ask us to determine the best Luenberger Observer for each of the output vectors of they are observable and then simulate the response to input conditions and unit step input when applied to the linearized and non linearized system.
- The final step is to design an output feedback controller for the smallest output vector using the LQG method and illustrate its performance in form of a simulation.

2 Equations of motion

- Defining the position for mass m_1 as a function of θ_1

$$x_{m_1} = (x - l_1 \sin(\theta_1))\,\hat{i} + (-l_1 \cos(\theta_1))\,\hat{j} \tag{1}$$

- The velocity equation is a result of differentiating the above equation with respect to time:

$$v_{m_1} = \left(\dot{x} - l_1 \cos(\theta_1)\,\dot{\theta_1}\right)\hat{i} + \left(l_1 \sin(\theta_1)\,\dot{\theta_1}\right)\hat{j} \tag{2}$$

- This allows us to define the position of mass m_2 as a function of θ_2 :

$$x_{m_2} = (x - l_2 \sin(\theta_2))\,\hat{i} + (-l_2 \cos(\theta_2))\,\hat{j} \tag{3}$$

- The second velocity equation is a result of differentiating the above equation with respect to time:

$$v_{m_2} = \left(\dot{x} - l_2 \cos(\theta_2)\,\dot{\theta_2}\right)\hat{i} + \left(l_2 \sin(\theta_2)\,\dot{\theta_2}\right)\hat{j} \tag{4}$$

- We can now formulate the kinetic energy of the system with the two derived velocity equations:

$$\begin{split}\mathrm{K} = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\left(\dot{x} - l_1\dot{\theta}_1\cos(\theta_1)\right)^2 + \frac{1}{2}m_1\left(l_1\dot{\theta}_1\left(\sin(\theta_1)\right)\right)^2 + \frac{1}{2}m_2(\dot{x} - \\ \dot{\theta}_2 l_2\cos(\theta_2))^2 + \frac{1}{2}m_2\left(l_2\dot{\theta}_2\left(\sin(\theta_2)\right)\right)^2\end{split} \tag{5}$$

- Potential energy of the system is given by:

$$P = -m_1 g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_2) = -g\left[m_1 l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_2)\right. \tag{6}$$

- We know that the Lagrange equation is the difference between Kinetic and Potential energies:

$$L = K - P \tag{7}$$

$$\begin{gathered} L = K \cdot E - P \cdot E \\ \mathrm{L} = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\dot{x}^2 + \frac{1}{2}m_1 l_1^2\dot{\theta_1}^2\cos^2(\theta_1) - m_1 l_1\dot{\theta_1}\dot{x}\cos(\theta_1) + \frac{1}{2}m_1 l_1^2\dot{\theta_1}^2\sin^2(\theta_1) \\ +\frac{1}{2}m_2\dot{x}^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2\cos^2(\theta_2) - m_2 l_2\dot{\theta}_2\dot{x}\cos(\theta_2) + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2\sin^2(\theta_1) + g\left[m_1 l_1\cos(\theta_1) + \right. \\ \left. m_2 l_2 \cos(\theta_2)\right] \end{gathered} \tag{8}$$

- Simplifying:

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

- Lyapunov Equations related to the state variables considered for the system are defined as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = F \tag{10}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \left(\frac{\partial L}{\partial \theta_1}\right) = 0 \tag{11}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \left(\frac{\partial L}{\partial \theta_2}\right) = 0 \tag{12}$$

- The computation of the above stated equations results in these following relations -
  Relation 1:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = F \tag{13}$$

$$\frac{\partial L}{\partial \dot{x}} = M\dot{x} + (m_1 + m_2)\dot{x} - m_1 l_1 \dot{\theta}_1 \cos(\theta_1) - m_2 l_2 \dot{\theta}_2 \cos(\theta_2) \tag{14}$$

$$\begin{gathered}\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}}) = M\ddot{x} + (m_1 + m_2)\ddot{x} - [m_1 l_1 \ddot{\theta}_1 \cos(\theta_1) \\ -m_1 l_1 \dot{\theta}_1^2 \sin(\theta_1)] - [m_2 l_2 \ddot{\theta}_2 \cos(\theta_2) - m_2 l_2 \dot{\theta}_2^2 \sin(\theta_2)]\end{gathered} \tag{15}$$

- Also here:

$$\frac{\partial L}{\partial x} = 0 \tag{16}$$

- The first equation can be written as:

$$[M + m_1 + m_2]\,\ddot{x} - m_1 l_1 \ddot{\theta}_1 \cos(\theta_1) + m_1 l_1 \dot{\theta_1}^2 \sin(\theta 1) - m_2 l_2 \ddot{\theta}_2 \cos(\theta_2) + m_2 l_2 \dot{\theta_2}^2 \sin(\theta_2) = F \tag{17}$$

- Now, we know that:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \left(\frac{\partial L}{\partial \theta_1}\right) = 0 \tag{18}$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = m_1 l_1^2 \dot{\theta}_1 - m_1 \dot{x} l_1 \cos(\theta_1) \tag{19}$$

- Differentiating this with respect to time, we get:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = m_1 l_1^2 \ddot{\theta}_1 - \left[m_1 l_1 \ddot{x} \cos(\theta_1) - m_1 \dot{x} l_1 \dot{\theta}_1 \sin(\theta_1)\right] \tag{20}$$

- Also here:

$$\frac{\partial L}{\partial \theta_1} = m_1 l_1 \dot{\theta}_1 \dot{x} \sin(\theta_1) - m_1 l_1 g \sin(\theta_1) \tag{21}$$

- Now, combining these two equations:

$$m_1 l_1^2 \ddot{\theta}_1 - m_1 \ddot{x} l_1 \cos(\theta_1) + m_1 \dot{\theta}_1 \dot{x} l_1 \sin(\theta_1) - m_1 \dot{\theta}_1 \dot{x} l_1 \sin(\theta_1) + m_1 l_1 g \sin(\theta_1) = 0 \tag{22}$$

- By cancelling out the equivalent terms, we get the following equation, Which is the second Lagrange Equation:

$$m_1 l_1^2 \ddot{\theta}_1 - m_1 \ddot{x} l_1 \cos(\theta_1) + m_1 l_1 g \sin(\theta_1) = 0 \tag{23}$$

- Now, to find the third equation from the Lagrange equation, we perform the following calculations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \left(\frac{\partial L}{\partial \theta_2}\right) = 0 \tag{24}$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \dot{\theta}_2 - m_2 \dot{x} l_2 \cos(\theta_2) \tag{25}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = m_2 l_2^2 \ddot{\theta}_2 - \left[m_2 \ddot{x} l_2 \cos(\theta_2) - m_2 \dot{\theta}_2 \dot{x} l_2 \sin(\theta_2)\right] \tag{26}$$

$$\left(\frac{\partial L}{\partial \theta_2}\right) = m_2 \dot{x} l_2 \dot{\theta}_2 \sin(\theta_2) - m_2 l_2 g \sin(\theta_2) \tag{27}$$

- Hence we write :

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \ddot{x} \cos(\theta_2) + m_2 \dot{\theta}_2 \dot{x} l_2 \sin(\theta_2) - m_2 \dot{\theta}_2 \dot{x} l_2 \sin(\theta_2) + m_2 g l_2 \sin(\theta_2) \tag{28}$$

- This implies that after cancellation of terms, we get:

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \ddot{x} \cos(\theta_2) + m_2 g l_2 \sin(\theta_2) = 0 \tag{29}$$

- Before linearizing about the given equilibrium points, we write the equations for the double differentiation components of some of our state variables, as deduced from the equations above:

$$\ddot{x} = \tfrac{1}{M+m_1+m_2}\left[m_1 l_1 \ddot{\theta}_1 \cos\theta_1 + m_2 l_2 \ddot{\theta}_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F\right] \tag{30}$$

$$\ddot{\theta}_1 = \frac{\ddot{x}\cos\theta_1}{l_1} - \frac{g\sin\theta_1}{l_1} \tag{31}$$

$$\ddot{\theta}_2 = \frac{\ddot{x}\cos\theta_2}{l_2} - \frac{g\sin\theta_2}{l_2} \tag{32}$$

2.1 Non linear state space representation

- Considering the system states, we can write the state space form of the non linear system as follows

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{-m_1 g \sin\theta_1 \cos\theta_2 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F}{M+m_1+m_2-m_1\cos^2\theta_1 - m_2\cos^2\theta_2} \\ \dot{\theta}_1 \\ \frac{-m_1 g \sin\theta_1 \cos\theta_2 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F}{(M+m_1+m_2-m_1\cos^2\theta_1 - m_2\cos^2\theta_2) l_1} - \frac{g\sin\theta_1}{l_1} \\ \dot{\theta}_2 \\ \frac{-m_1 g \sin\theta_1 \cos\theta_2 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F}{(M+m_1+m_2-m_1\cos^2\theta_1 - m_2\cos^2\theta_2) l_2} - \frac{g\sin\theta_2}{l_2} \end{bmatrix} \tag{33}$$

3 Linearizing the system around an equilibrium point

- Linearization is finding the linear approximation to a function at a given point. The linear approximation of a function is the first order Taylor expansion around the point of interest. In the study of dynamical systems, linearization is a method for assessing the local stability of an equilibrium point of a system of nonlinear differential equations or discrete dynamical systems. Linearization makes it possible to use tools for studying linear systems to analyse the behaviour of a nonlinear function near a given point.

- As discussed above, the derived equation of motion of the cart system with two pendulum and its represented state space form. We observe that, due to the sine and cosine components, the equations are non linear and it is arduous to solve non linear equations.

- To approach this we linearize the system around equilibrium point $x = 0, \theta_1 = 0$ and $\theta_2 = 0$, which is mentioned in the problem statement.
  The limiting condition at equilibrium, we set,

$$\begin{gathered} \sin\theta_1 \approx \theta_1 \\ \sin\theta_2 \approx \theta_2 \\ \cos\theta_1 \approx 1 \\ \cos\theta_2 \approx 1 \\ \dot{\theta}_1^2 = \dot{\theta}_2^2 \approx 0 \end{gathered}$$

- With these approximations, we arrive at these equations:

$$\begin{gathered} \ddot{x} = \frac{1}{M+m_1+m_2}\left[m_1 l_1 \ddot{\theta}_1 \cos\theta_1 + m_2 l_2 \ddot{\theta}_2 \cos\theta_2 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2 + F\right] \\ \ddot{\theta}_1 = \frac{\ddot{x}\cos\theta_1}{l_1} - \frac{g\sin\theta_1}{l_1} \\ \ddot{\theta}_2 = \frac{\ddot{x}\cos\theta_2}{l_2} - \frac{g\sin\theta_2}{l_2} \end{gathered} \tag{34}$$

- The system of equations represented in Eqn 34 can be presented in state space form. Taking $x, \dot{x}, \theta_1, \dot{\theta}_1, \theta_2$ and $\dot{\theta}_2$ as state variables, as shown below,

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{-m_1 g\sin\theta_1\cos\theta_2 - m_2 g\sin\theta_2\cos\theta_2 - m_1 l_1\dot{\theta}_1^2\sin\theta_1 - m_2 l_2\dot{\theta}_2^2\sin\theta_2 + F}{M+m_1+m_2-m_1\cos^2\theta_1 - m_2\cos^2\theta_2} \\ \dot{\theta}_1 \\ \frac{-m_1 g\sin\theta_1\cos\theta_2 - m_2 g\sin\theta_2\cos\theta_2 - m_1 l_1\dot{\theta}_1^2\sin\theta_1 - m_2 l_2\dot{\theta}_2^2\sin\theta_2 + F}{(M+m_1+m_2-m_1\cos^2\theta_1 - m_2\cos^2\theta_2)l_1} - \frac{g\sin\theta_1}{l_1} \\ \frac{-m_1 g\sin\theta_1\cos\theta_2 - m_2 g\sin\theta_2\cos\theta_2 - m_1 l_1\dot{\theta}_1^2\sin\theta_1 - m_2 l_2\dot{\theta}_2^2\sin\theta_2 + F}{(M+m_1+m_2-m_1\cos^2\theta_1 - m_2\cos^2\theta_2)l_2} - \frac{g\sin\theta_2}{l_2} \end{bmatrix} \tag{35}$$

$$y = CX + DU \tag{36}$$

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

- This is similar to $\dot{X} = AX + Bu$. Input force on cart F is represented as u.

4 Obtaining controllability conditions

- The linear time varying system is controllable on (0,T) iff the grammian square matrix of controllability is invertible. The Grammian matrix of controllability C(A, B) are considered to be invertible when the controllability matrix satisfies
- Rank (C(A,B)) = Full Rank
- The Full Rank is given by the order of the matrix (n). The controllability matrix is given by
- When we compute the code and check, we can see that for a condition where $l_1 = l_2$, the determinant will be zero and when $l_1 = 0$ or $l_2 = 0$, the controllability is not computable. Therefore the conditions to obtain controllability are $l_1! = l_2, l_1! = 0 and l_2! = 0$.

$$R = \begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix}$$

```
%Variable symbols are as f
syms M m1 m2 l1 l2 g;
% State space representation of the linearised model
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
disp(A)
```

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

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

B =

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

```
% The Controllability matrix
Cability= [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B]
```

```
Cability =
```

$$\left(\begin{array}{cccccc} 0 & \frac{1}{M} & 0 & \sigma_2 & 0 & \sigma_1 \\ \frac{1}{M} & 0 & \sigma_2 & 0 & \sigma_1 & 0 \\ 0 & \frac{1}{M\,l_1} & 0 & \sigma_6 & 0 & \sigma_4 \\ \frac{1}{M\,l_1} & 0 & \sigma_6 & 0 & \sigma_4 & 0 \\ 0 & \frac{1}{M\,l_2} & 0 & \sigma_5 & 0 & \sigma_3 \\ \frac{1}{M\,l_2} & 0 & \sigma_5 & 0 & \sigma_3 & 0 \end{array}\right)$$

where

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

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

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

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

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

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

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

```
disp("Determinant of the controllability matrix="); disp(simplify(det(Cability)));
```

```
Determinant of the controllability matrix=
```

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

```
%The system is controllable only if the controllability matrix is full rank.
%Rank is 6 in this case, so system is controllable
disp("Rank"); rank(Cability)
```

```
Rank
ans = 6
```

```
% We can see for this case det of matrix will be zero, hence it is
% uncontrollable
disp("for l1 = l2, Controllability matrix is")
```

```
for l1 = l2, Controllability matrix is
```

```
Ct1 = subs(Cability,l1,l2) %using the subs function to make l1 = l2
```

Ct1 =

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

where

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

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

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

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

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

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

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

$$\sigma_8 = M^2\,{l_2}^2$$

```
disp("Displaying rank of the new matrix =")
```

Displaying rank of the new matrix =

```
%Calculating the rank in this case.
rank(Ct1)
```

ans = 4

```
disp("system is not controllable as new rank and old rank
are different")
```

system is not controllable as new rank and old rank are different

```
disp("system is controllable under the condition l1 not equal
to l2")
```

```
system is controllable under the condition l1 not equal to l2
```

5 LQR controller design

- In order to make the system controllable, various controllers can be used to drive the output to the desired value. The type of controller chosen can be depending on the model of system. A PID follows a classical Linear Equation approach, but an LQR focuses on non-linear models. LQR by definition gives the optimal state-feedback law that minimizes certain quadratic objective function. In that sense, LQR is the best controller.
- For the LQR Controller, if A,B_k is stabilizable, then we can look for k that minimizes the following cost:
$$J(k, \vec{X}(0)) = \int_0^\infty \vec{X}^T(t) Q \vec{X}(t) + \vec{U}_k^T(t) R \vec{U}_k(t) dt$$
- Where Q (non-negative definite) Matrix penalizes the bad performance and R (positive definite) Matrix penalizes the actuator effort i.e., the energy consumed by the system.
- Different values of Q and R are taken to control the system in optimum way. Output graphs gives the validation of how good the chosen values are for Q and R matrices.
- Q can be decided on how much approximate time the system should take to stabilize the particular state and what is the error band within which it should work. Lower R can yield faster stability as the energy utilized is less.

5.1 Check for controllability

```
% specifics for the the problem
M= 1000; %Crane mass
m1= 100; % Load 1 mass
m2= 100; % Load 2 mass
l1= 20;  % Cable length of Load 1
l2= 10;  % Cable length of Load 2
g=  9.81;

%The A and B values we got from our state space representation
%Substituting the given values of specifics in A and B
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];

Cont =  ctrb(A,B)
```

```
Cont = 6x6
1.0e+-3 *

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

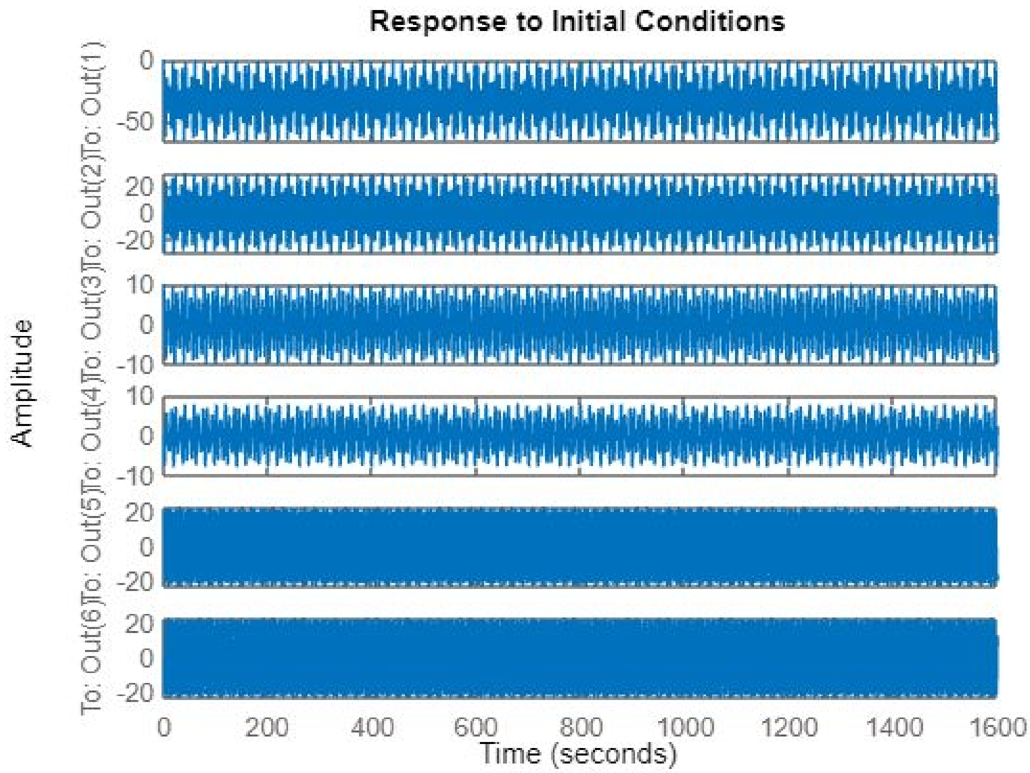
```
if (rank(Cont)==6)
    disp("Rank of ctrb matches order of A, system is controllable")
else
    disp("Rank of ctrb doesnt matche order of A, system is uncontrollable")
end
```

```
Rank of ctrb matches order of A, system is controllable
```

5.2 Simulate responses to initial conditions

5.2.1 When applied to Linearized system

```
% The initial conditions are as follows.
    X_0 = [0;0;10;0;20;0];
% We assume the values of Q and R.
    Q=[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];
    R=0.001;
% Q and R are a part of the cost function of LQR controller
% It is an trade-off between Q and R so we use them both to
% develop a system as per our priorities
% Assumption: C matrix is a direct representation of the output
% matrix, which makes D=0
    C = eye(6); D = 0;
    sys1 = ss(A,B,C,D);
% ss is the MATLAB function  for
% calculating the state space representation of the system
figure
initial(sys1,X_0)
%MATLAB inbuilt function to check the initial response of the system
grid on
```




```
disp("When an LQR controller is taken into consideration of system")
```

```
When an LQR controller is taken into consideration of system
```

```
[K_Gain_mat, Po_def_mat, Poles] = lqr(A,B,Q,R);
%In-built MATLAB code for LQR Controllers
K_Gain_mat
```

```
K_Gain_mat = 1x6
  316.2278   926.8775   -41.7027 -683.4756   44.1102 -334.2650
```

```
Po_def_mat
```

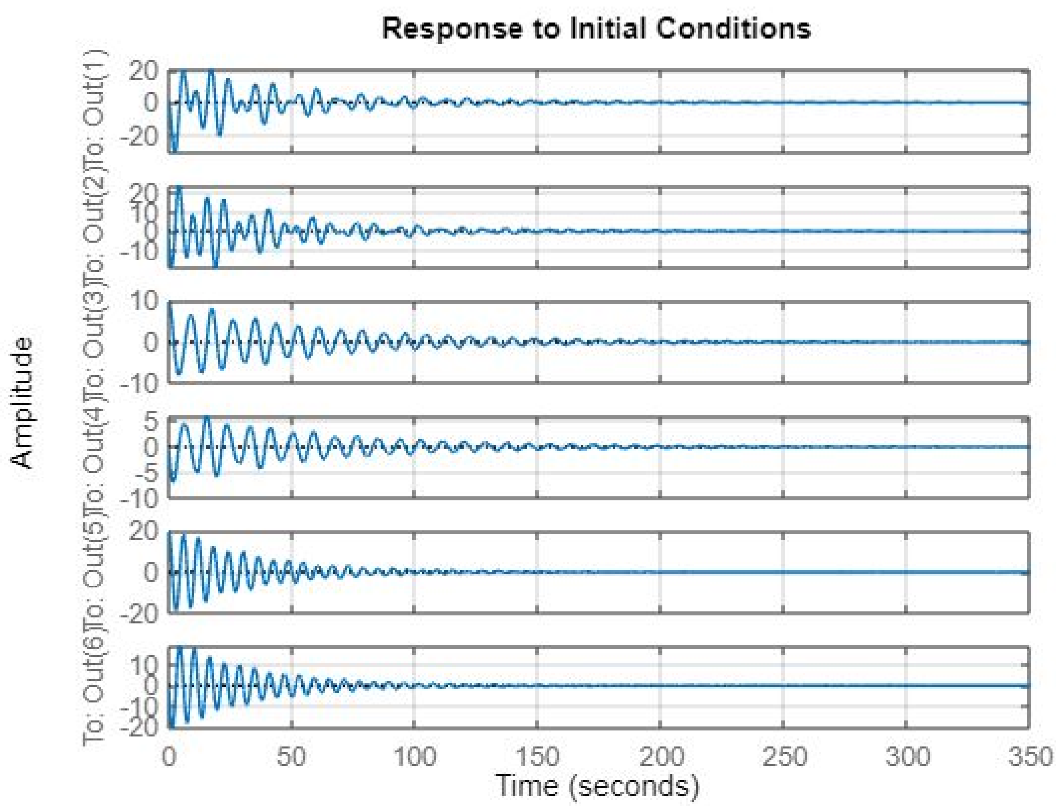
```
Po_def_mat = 6x6
1.0e+04 *

    0.0293    0.0380   -0.0216   -0.0606   -0.0106   -0.0330
    0.0380    0.1145   -0.0028   -0.2216    0.0021   -0.1076
   -0.0216   -0.0028    1.3615    0.0184    0.0073   -0.0230
   -0.0606   -0.2216    0.0184    2.9066    0.0458    0.0788
```

```
   -0.0106    0.0021    0.0073    0.0458    0.6550    0.0006
   -0.0330   -0.1076   -0.0230    0.0788    0.0006    0.7026
```

```
Poles
```

```
Poles = 6x1 complex
  -0.0161 + 0.7213i
  -0.0161 - 0.7213i
  -0.0298 + 1.0347i
  -0.0298 - 1.0347i
  -0.3838 + 0.3543i
  -0.3838 - 0.3543i
```

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



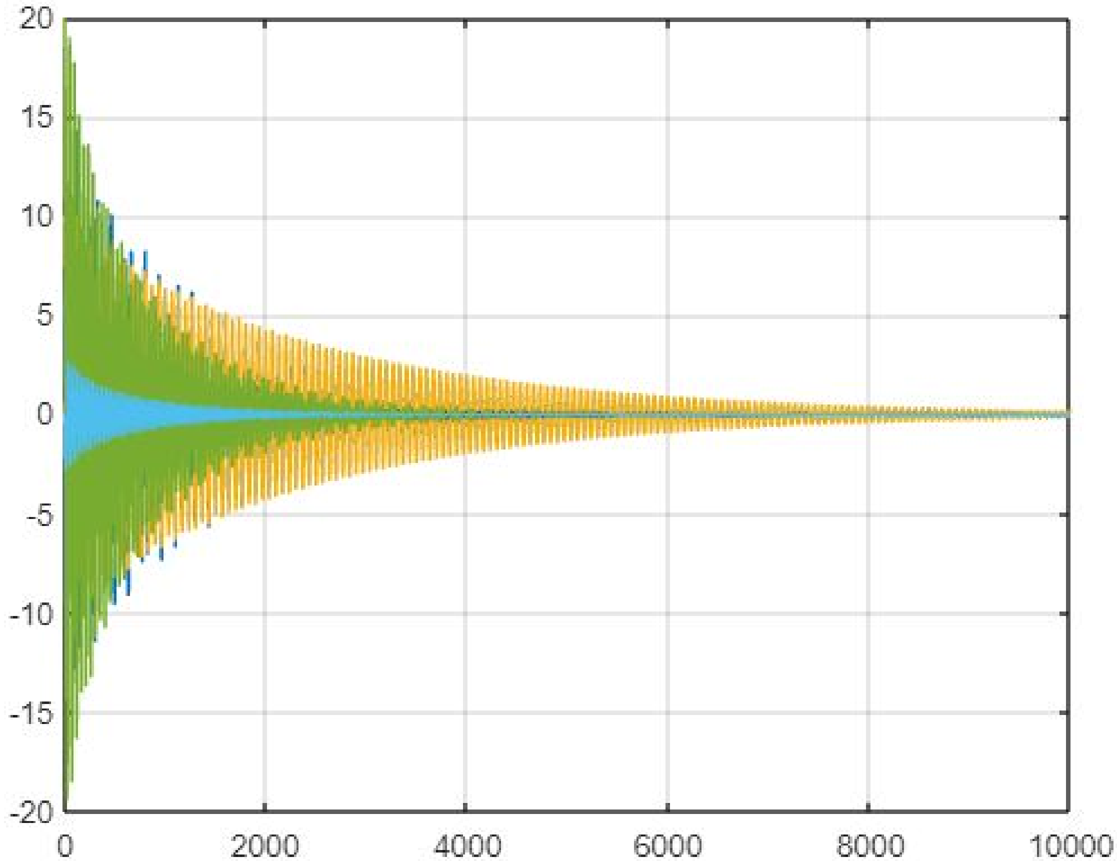
```
%observation: lower the R value, faster the stabilization
%but the energy consumed is also much higher
```

5.2.2 When applied to non linear system

```
% Declaring the new output variables
% An x, theta_1 and theta_2 values are defined
% The system has the single derivatives of the values
% All of them are state variables and contribute to y0
y0 = [10; 0; 10; 0; 20; 0]
```

```
y0 = 6x1
     5
     0
    10
     0
    20
     0
```

```
%defining the timespan
    tspan = 0:0.01:10000;
%using ode45 function for defininining a diff eqn
    [t1,y1] = ode45(@twoload,tspan,y0);
%plotting the function output on a 2D graph
    plot(t1,y1)
    grid on
```




```
function dydt = twoload(t1,y)
%substituting the values of M, m1, m2, l1, l2
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
Q=[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];
R=0.01;
[K_Gain_mat, Po_def_mat, Poles] = lqr(A,B,Q,R);
F=-K_Gain_mat*y;
dydt=zeros(6,1);
% y(1)=x;
```

```
dydt(1) = y(2);
%y(2)=xdot;
dydt(2)=(F-(g/2)*(m1*sind(2*y(3))+m2*sind(2*y(5)))-(m1*l1*(y(4)^2)*sind(y(3)))
-(m2*l2*(y(6)^2)*sind(y(5))))/(M+m1*((sind(y(3)))^2)+m2*((sind(y(5)))^2)); %X_DD
%y(3)=theta1;
dydt(3)= y(4);
%y(4)=theta1dot;
dydt(4)= (dydt(2)*cosd(y(3))-g*(sind(y(3))))/l1'; %theta 1 Ddot;
%y(5)=theta2;
dydt(5)= y(6); %theta 2D
%y(6)=theta2dot;
dydt(6)= (dydt(2)*cosd(y(5))-g*(sind(y(5))))/l2; %theta 2Ddot;
end
```

5.3 Lyapunov stability analysis

- When we see the eigen values of (ABK), it should be in left half plane for the system to be stable. Eigen values of (ABK) are as follows:

  $(\lambda_1 = -0.0161 + 0.7213i)$
  $(\lambda_2 = -0.0161 - 0.7213i)$
  $(\lambda_3 = -0.0298 + 1.0347i)$
  $(\lambda_4 = -0.0298 - 1.0347i)$
  $(\lambda_5 = -0.3838 + 0.3543i)$
  $(\lambda_6 = -0.3838 - 0.3543i)$

- Real parts of all the eigen values are negative, and thus with the help of Lyapunov's indirect stability criterion, we can say that the system is stable.

6 Observability of the system

- Observability of a control system is the ability of the system to determine the internal states of the system by observing the output in a finite time interval when input is provided to the system.
- When we consider a LTI discrete system in the state space form
  $X(k+1) = AX(k) \quad X(0) = X = unknown$
- This will have an output of Y=CX+DU
- Now the Observability matrix for this is $O\,(A, C) = \quad [CCACA2......CAn - 1]T$
- When the $Rank(O(A,C))$ =s full rank, then the system that we consider is observable.

```
syms M m1 m2 l1 l2 g;
% Creating a linearised state space equation using A and B 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 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];         %Initializing the B matrix
C1 = [1 0 0 0 0 0];                           %Corresponding to x component
C2 = [0 0 1 0 0 0; 0 0 0 0 1 0];              %corresponding to theta1 and theta2
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0];              %cooresponding to x and theta2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %cooresponding to x,theta1 and theta2
%Matrix to check th Observability Condition
Observability1 = [C1' A'*C1' A'*A'*C1'
    A'*A'*A'*C1' A'*A'*A'*A'*C1' A'*A'*A'*A'*A'*C1'];
Observability2 = [C2' A'*C2' A'*A'*C2'
    A'*A'*A'*C2' A'*A'*A'*A'*C2' A'*A'*A'*A'*A'*C2'];
Observability3 = [C3' A'*C3' A'*A'*C3'
A'*A'*A'*C3' A'*A'*A'*A'*C3' A'*A'*A'*A'*A'*C3'];
Observability4 = [C4' A'*C4' A'*A'*C4'
A'*A'*A'*C4' A'*A'*A'*A'*C4' A'*A'*A'*A'*A'*C4'];
rankArray = [rank(Observability1),rank(Observability2)
,rank(Observability3),rank(Observability4)];
```

```
% Iterating Over rankArray
for i = 1:4
  switch(i)
      case 1
          if rankArray(i)==6 %Checking if rank is 6
              disp('System is observable, when only x(t) is requested!')
            else
              disp('System is not observable, when only x(t) is requested!')
          end
      case 2
           if rankArray(i)==6 %Checking if rank is 6
              disp('System is observable, when only
              theta1(t) and theta2(t) is requested!')
           else
              disp('System is not observable, when only
              theta1(t) and theta2(t) is requested!')
           end
      case 3
           if rankArray(i)==6 %Checking if rank is 6
                   disp('System is observable, when only
                   x(t) and theta2(t) is requested!')
            else
                disp('System is not observable, when
                x(t), theta1(t) and theta2(t) is requested!')
           end

      case 4
           if rankArray(i)==6 %Checking if rank is 6
                disp('System is observable, when x(t),
                theta1(t) and theta2(t) is requested!')
            else
                disp('System is not observable, when
                x(t), theta1(t) and theta2(t) is requested!')
            end

  end
end
```

```
System is observable, when only x(t) is requested!
System is not observable, when only theta1(t) and theta2(t) is requested!
System is observable, when only x(t) and theta2(t) is requested!
System is observable, when x(t), theta1(t) and theta2(t) is requested!
```

7 Luenberger Observer

- The state space representation of the linear system:

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

We have created a Luenberger observer for the input vectors with which system is observable. The observer system can be represented as shown below,

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

- L is the observer gain and $\hat{X}(t)$ is our estimated state from output $y(t)$. We can find L by placing our poles for $A - LC$ in negative left-half plane. For better convergence we have put it at 3 times the distance from the eigen values of $A - BK$, where K is our feedback gain.
- The code below will simulate the non linear and linear system given with the initial condition and a step input at time $t = 20$ s.

7.1 Simulation of response to input conditions and unit step input

7.1.1 When applied to Linearized system

```
M=1000;%Mass of the Crane
m_1=100;%mass of Load 1
m_2=100;%mass of Load 2
l_1=20;%length of the string of Load 1
l_2=10;%length of the string of Load 2
g=9.81; %declaring the value of the accelertaion due to gravity in m/s^2
%Defining our matrices as follows
A=[0 1 0 0 0 0;
    0 0 -(m_1*g)/M 0 -(m_2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m_1)*g)/(M*l_1) 0 -(m_2*g)/(M*l_1) 0;
    0 0 0 0 0 1;
    0 0 -(m_1*g)/(M*l_2) 0 -(g*(M+m_2))/(M*l_2) 0];
B=[0; 1/M; 0; 1/(M*l_1); 0; 1/(M*l_2)];
% From previous case, we have determined that only C_1, C_3 and C_4 were
% observable. Hence, we are going to consider only those 3 cases.
C_1 = [1 0 0 0 0 0];  %Corresponding to x component
C_3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %corresponding to x and theta2
C_4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
%corresponding to x, theta1 and theta2
D = 0;               %declaring the D matrix to be zero
\\
```

```
% Considering the same Q and R matrices chosen before in our code
Q=[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];
R=0.01;
%The cost variables from LQR
% Initial Conditions for Leunberger observer - 12 state variables,
% 6 actual + 6 estimates
x0=[0,0,30,0,60,0,0,0,0,0,0,0];
% state variables order = [x,dx,theta_1,dtheta_1,theta_2,dtheta_2,
% estimates taken in the same order]
% For pole placement, lets choose eigen values with negative real part
poles=[-1;-2;-3;-4;-5;-6];
% Calling LQR function to obtain K matrix
K=lqr(A,B,Q,R);
% Framing L for all three cases where output is observable
% Using the pole placement funciton built into MATLAB
L_1 = place(A',C_1',poles)' %L1 should be a 6x1 matrix
```

```
L1 = 6x1
1.0e+03 *

    0.0210
    0.1734
   -2.9262
    0.0805
    2.2116
   -1.4493
```

```
L_3 = place(A',C_3',poles)' %L3 should be a 6x2 matrix
```

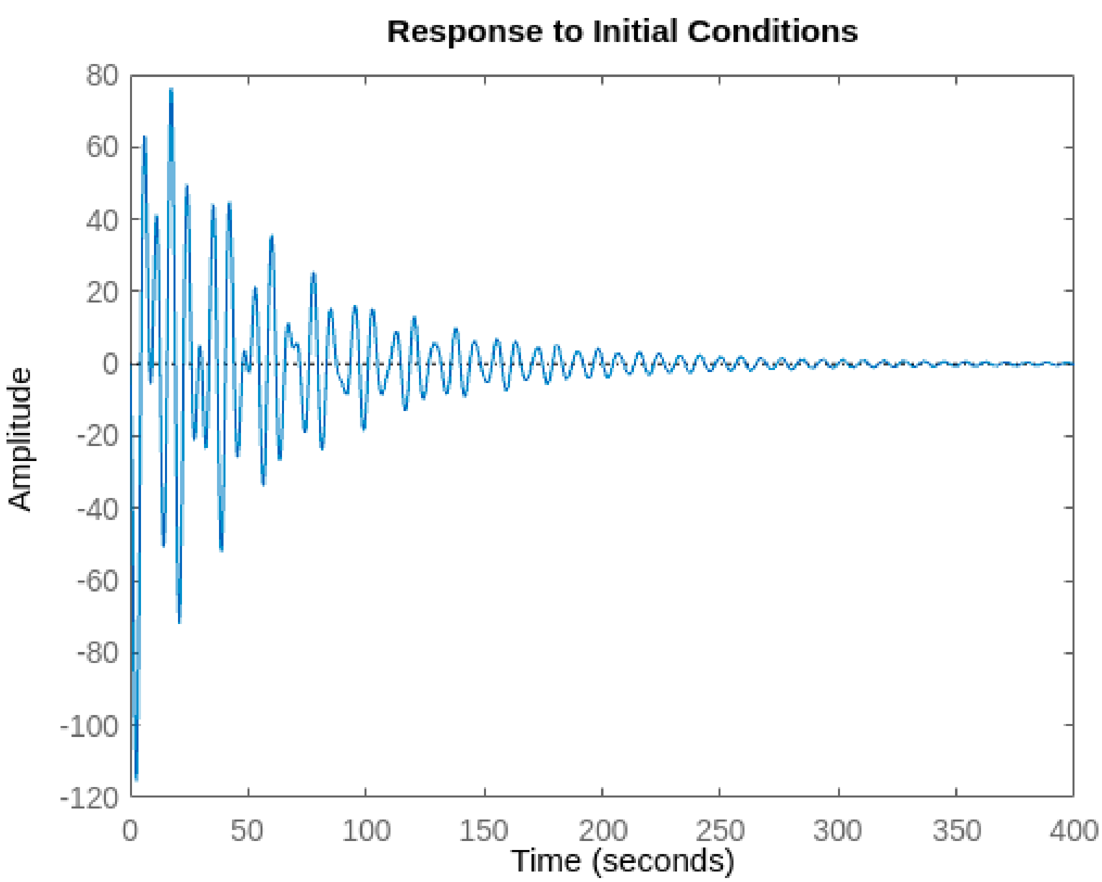
```
L3 = 6x2
   13.0744   -0.8244
   56.2562   -8.4805
  -89.0764   19.7693
  -20.0115   10.9419
    0.3520    7.9256
    3.4793   13.2122
```

```
L_4 = place(A',C_4',poles)' %L4 should be a 6x3 matrix
```

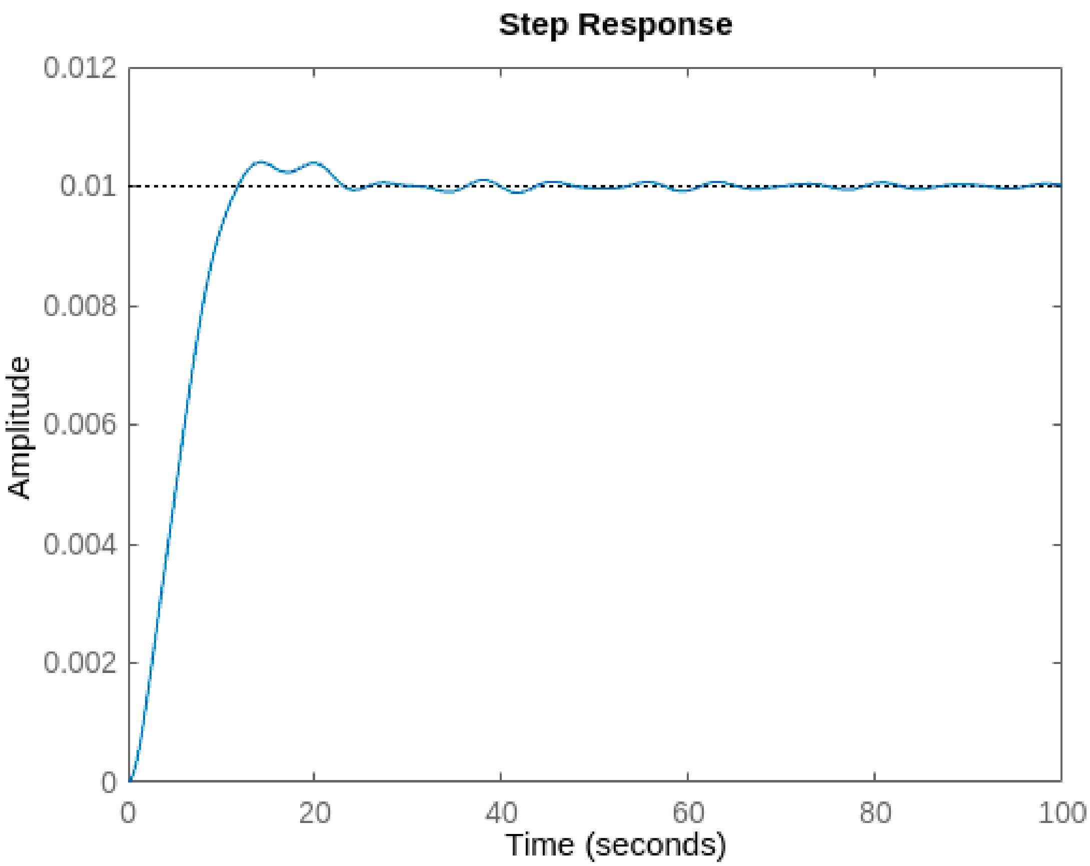
```
L4 = 6x3
    8.5631   -0.8851    0.0000
   17.5219   -4.9484   -0.9810
   -0.9140    9.4369   -0.0000
   -4.1173   20.9385   -0.0491
    0.0000   -0.0000    3.0000
    0.0000   -0.0981    0.9209
```

```
A_c1 = [(A-B*K) B*K; % Luenberger A matrix
        zeros(size(A)) (A-L_1*C_1)];
B_c = [B;zeros(size(B))];% Luenberger B matrix
C_c1 = [C_1 zeros(size(C_1))];% Luenberger C matrix

A_c3 = [(A-B*K) B*K;% Luenberger A matrix
        zeros(size(A)) (A-L_3*C_3)];
C_c3 = [C_3 zeros(size(C_3))];% Luenberger C matrix

A_c4 = [(A-B*K) B*K;% Luenberger A matrix
        zeros(size(A)) (A-L_4*C_4)];
C_c4 = [C_4 zeros(size(C_4))];% Luenberger C matrix
```

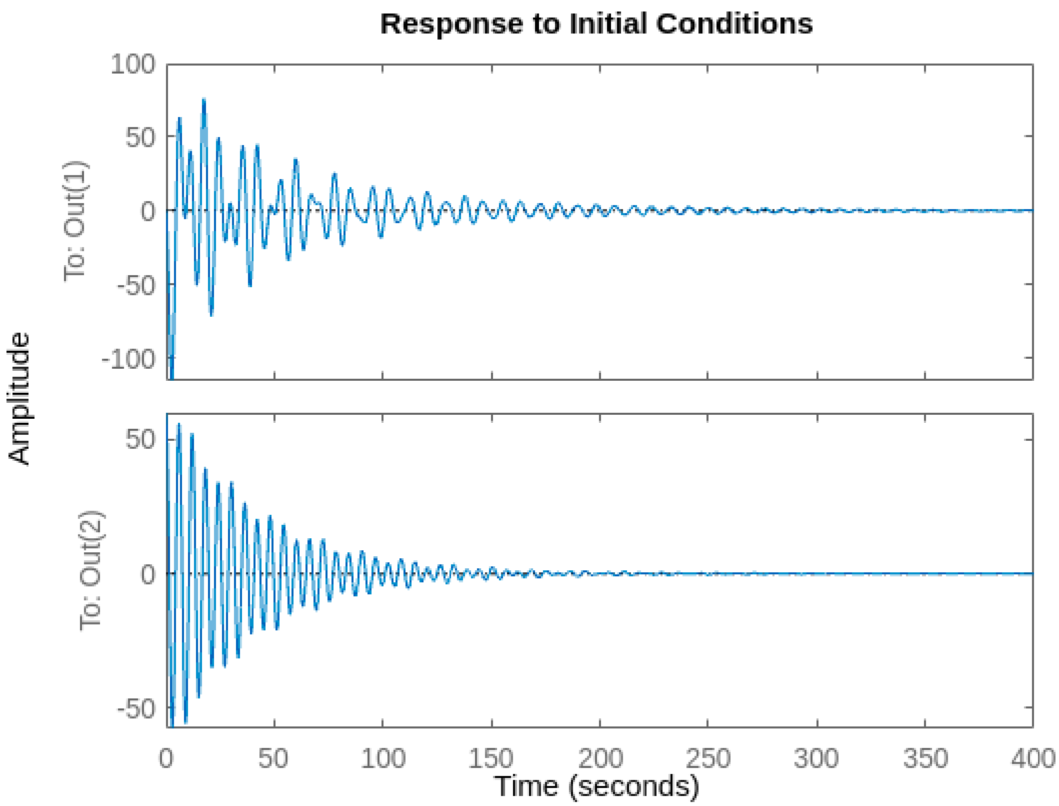
```
%%MATLAB function to output state space equations
sys_1 = ss(A_c1, B_c, C_c1,D);
figure % to launch a new figure WINDOW
initial(sys_1,x0)
%MATLAB function to check the initial response of the system
```



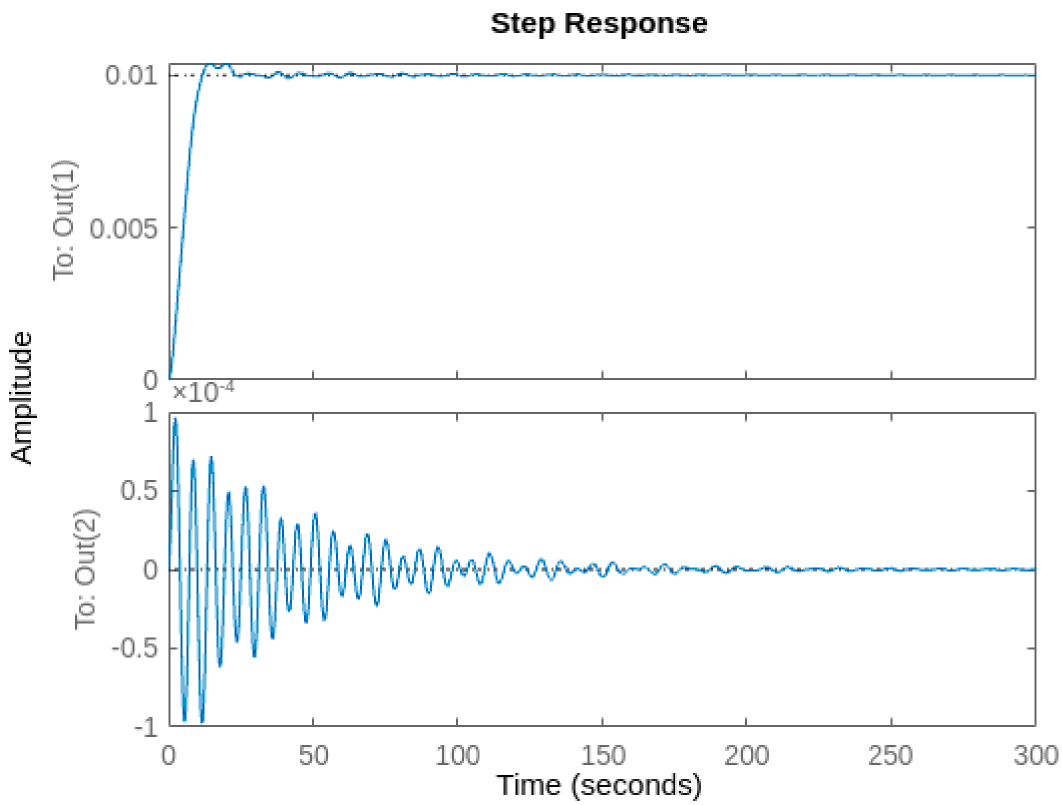
```
figure
step(sys_1)%Gives the step response output
```



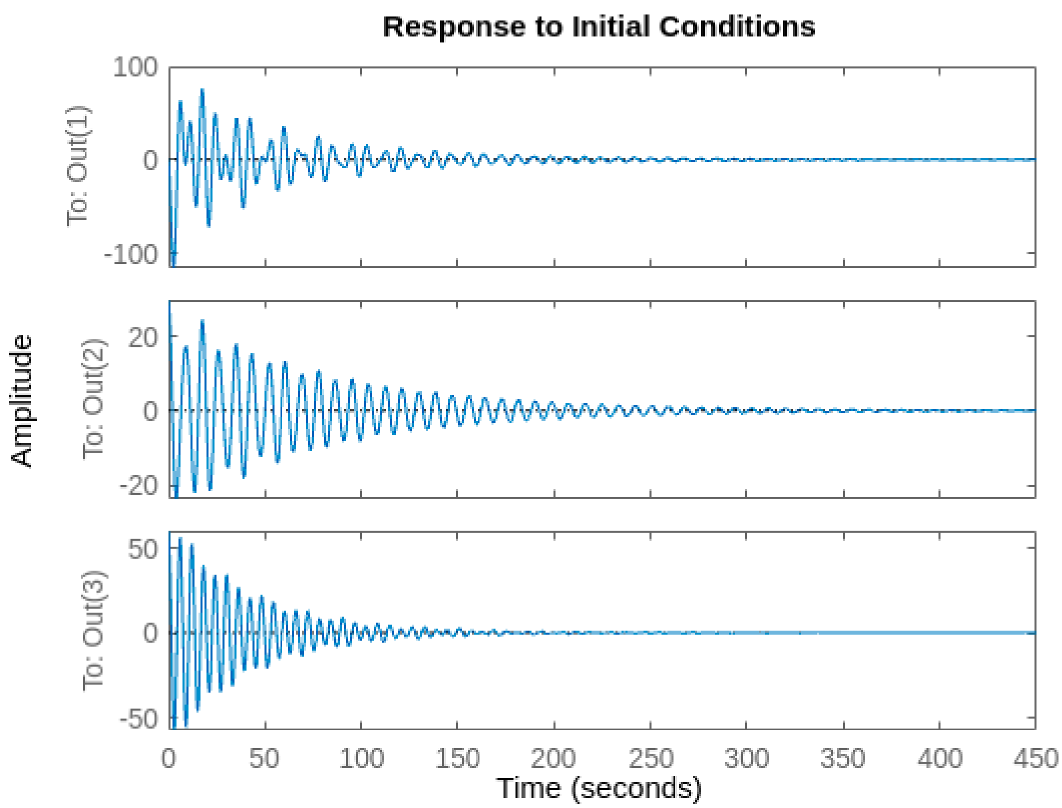
```
sys_3 = ss(A_c3, B_c, C_c3,D);%MATLAB function to output statespace equations
figure
initial(sys_3,x0)
%MATLAB inbuilt function to check the initial response of the system
```



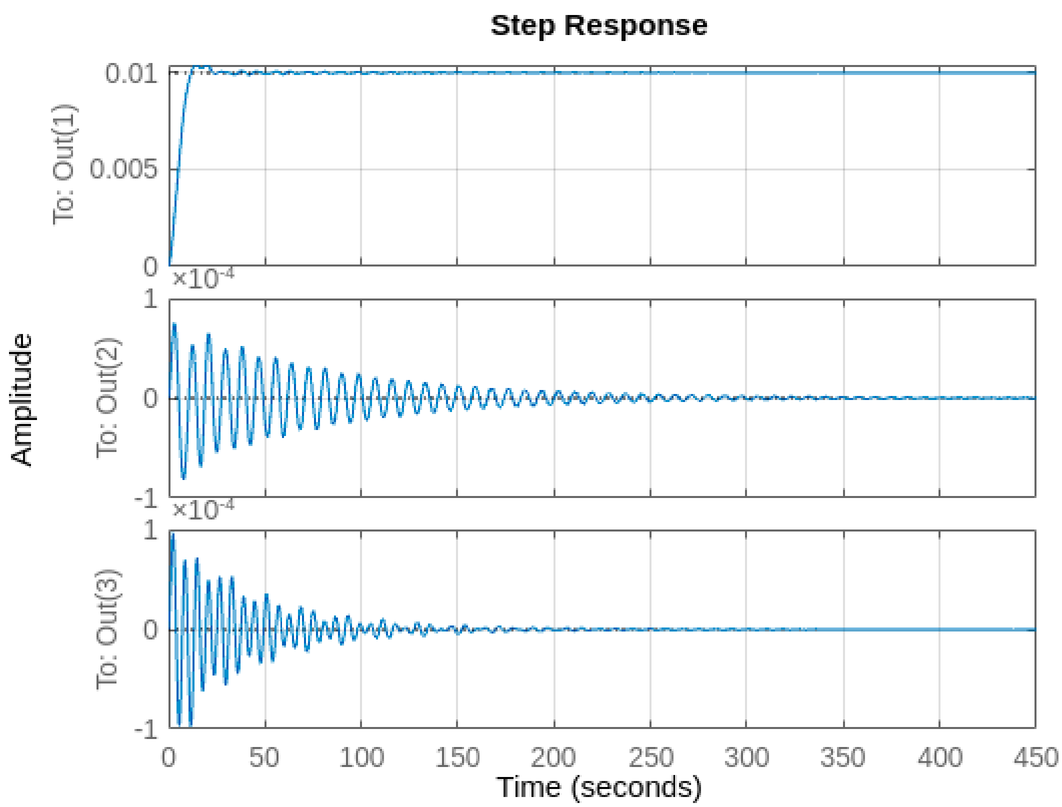
```
figure
step(sys_3)%Gives the step response output
```



```
%%MATLAB function to output statespace equations
sys_4 = ss(A_c4, B_c, C_c4, D);
figure
initial(sys_4,x0)
%MATLAB inbuilt function to check the initial response of the system
```



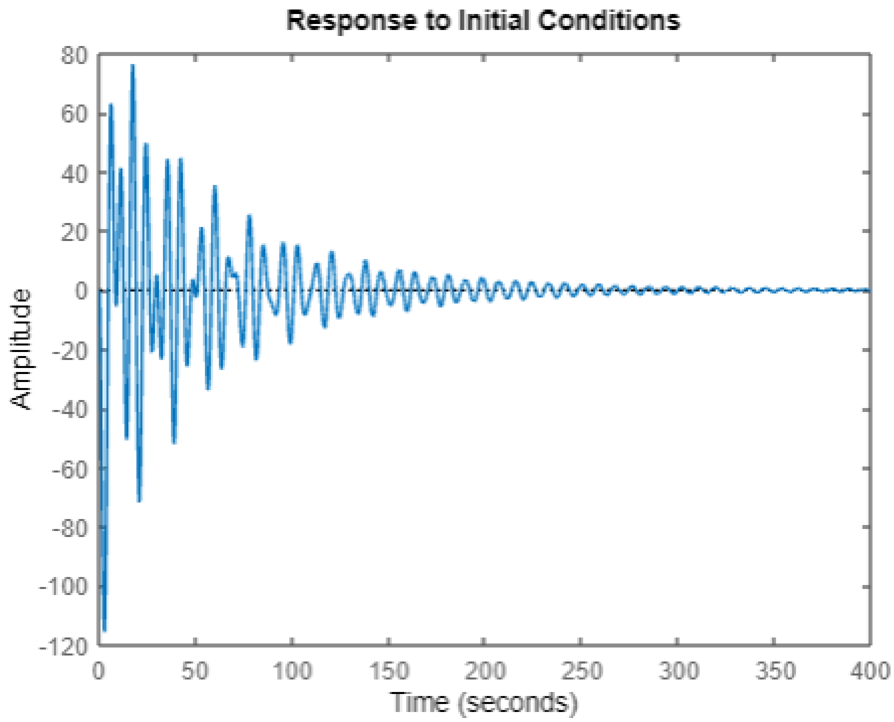
```
figure
step(sys_4)%Gives the step response output

grid on
```



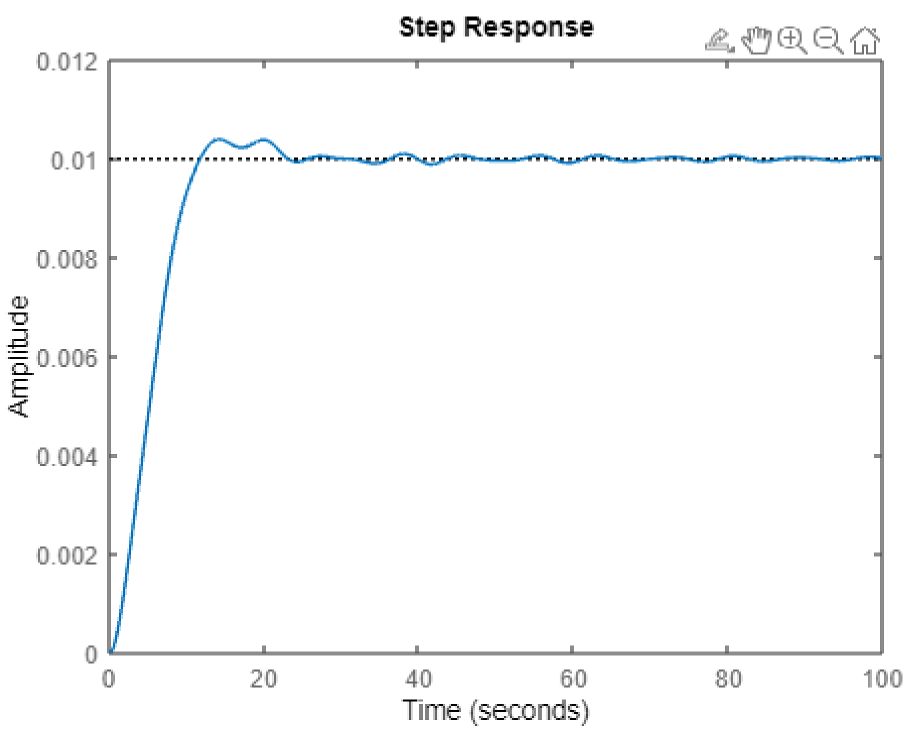
7.1.2 When applied to original non linear system

```
%clearing all the previous outputs
clc
clear
%Substituting values for our M, m1, m2, l1 and l2
M=1000;%Mass of the cart
m_1=100;%mass of Pendulum 1
m_2=100;%mass of Pendulum 2
l_1=20;%length of the string of Pendulum 1
l_2=10;%length of the string of Pendulum 2
g=9.81; %declaring the value of the accelertaion due to gravity in m/s^2
%declaring the state space matrices
A=[0 1 0 0 0 0;
 0 0 -(m_1*g)/M 0 -(m_2*g)/M 0;
 0 0 0 1 0 0;
 0 0 -((M+m_1)*g)/(M*l_1) 0 -(m_2*g)/(M*l_1) 0;
 0 0 0 0 0 1;
 0 0 -(m_1*g)/(M*l_2) 0 -(g*(M+m_2))/(M*l_2) 0];
B=[0; 1/M; 0; 1/(M*l_1); 0; 1/(M*l_2)];
% We have previously found out only C1, C3 and C4 are observable.
% Hence we will only use C1, C3 and C4.

C_1 = [1 0 0 0 0 0]; %the output measurment for x component
C_3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %the output measurment for x and theat2
C_4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
%the output measurment for x, theta1 and theat2
D = 0;%declarring the D matrix to be zero


% declaring the same Q and R matrices
Q=[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];
R=0.01; %%these are the cost variables from LQR

% Initial Conditions for Leunberger observer - 12 state variables,
% considering 6 actual + 6 estimates x0=[0,0,30,0,60,0,0,0,0,0,0,0];
% state variables order = [x,dx,theta_1,dtheta_1,theta_2,dtheta_2,
%estimates taken in the same order]
% For pole placement, lets choose eigen values with negative real part
poles=[-1;-2;-3;-4;-5;-6];
% Calling LQR function to obtain K matrix
K=lqr(A,B,Q,R);
```

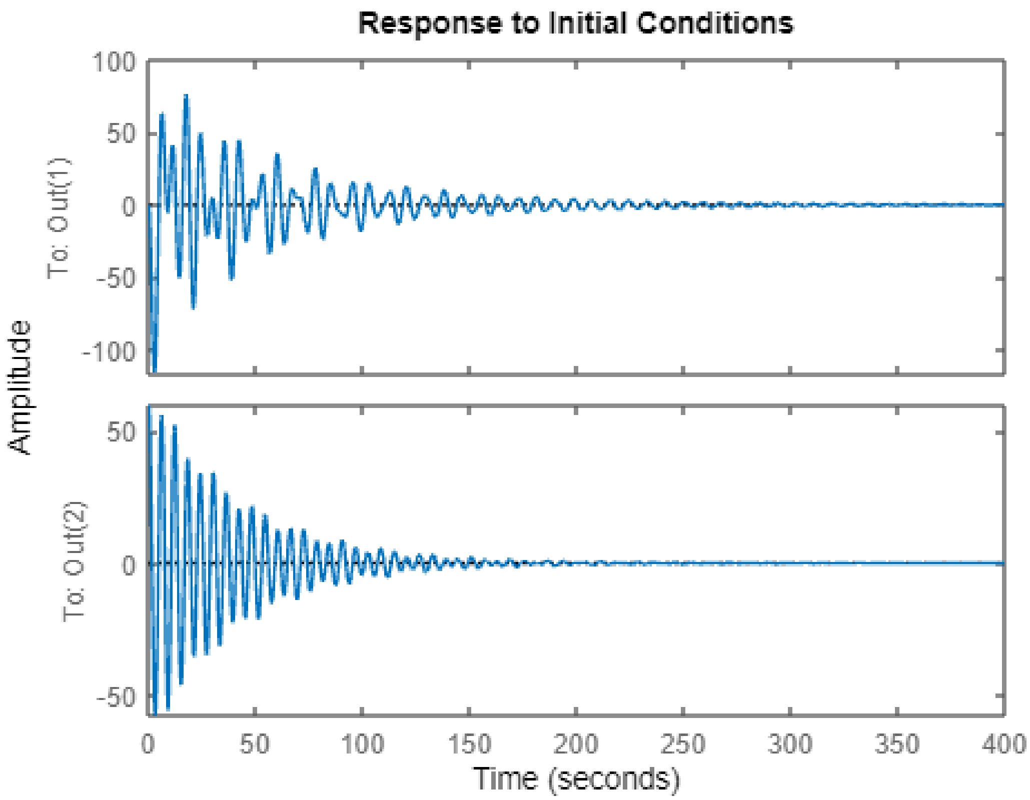
```
% Framing L for all three cases where output is observable
% Using the pole placement funciton built into MATLAB
L_1 = place(A',C_1',poles)'; %L1 should be a 6x1 matrix
L_3 = place(A',C_3',poles)'; %L3 should be a 6x2 matrix
L_4 = place(A',C_4',poles)'; %L4 should be a 6x3 matrix
A_c1 = [(A-B*K) B*K; zeros(size(A)) (A-L_1*C_1)];% Luenberger A matrix
B_c = [B;zeros(size(B))];% Luenberger B matrix
C_c1 = [C_1 zeros(size(C_1))];% Luenberger C matrix
A_c3 = [(A-B*K) B*K; zeros(size(A)) (A-L_3*C_3)];% Luenberger A matrix

C_c3 = [C_3 zeros(size(C_3))];% Luenberger C matrix
A_c4 = [(A-B*K) B*K; zeros(size(A)) (A-L_4*C_4)];% Luenberger A matrix
C_c4 = [C_4 zeros(size(C_4))];% Luenberger C matrix

sys_1 = ss(A_c1, B_c, C_c1,D);%%MATLAB function to output statespace equations
figure % to launch a new figure WINDOW
%MATLAB inbuilt function to check the initial response  of the system
initial(sys_1,x0)
```



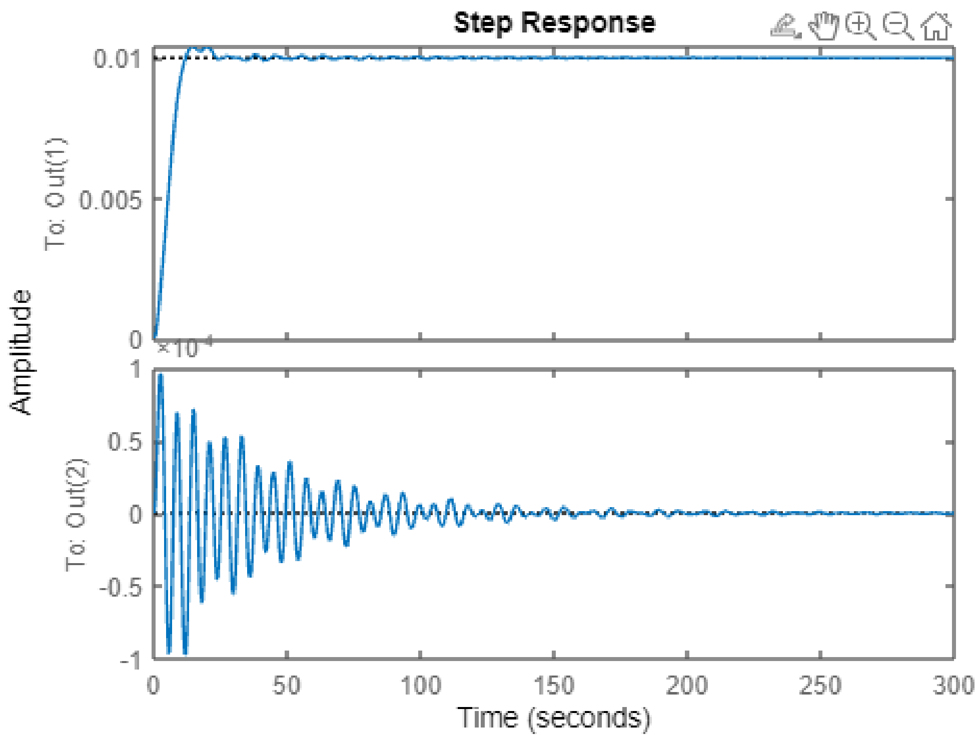

```
figure
step(sys_1)%Gives the step response output
```



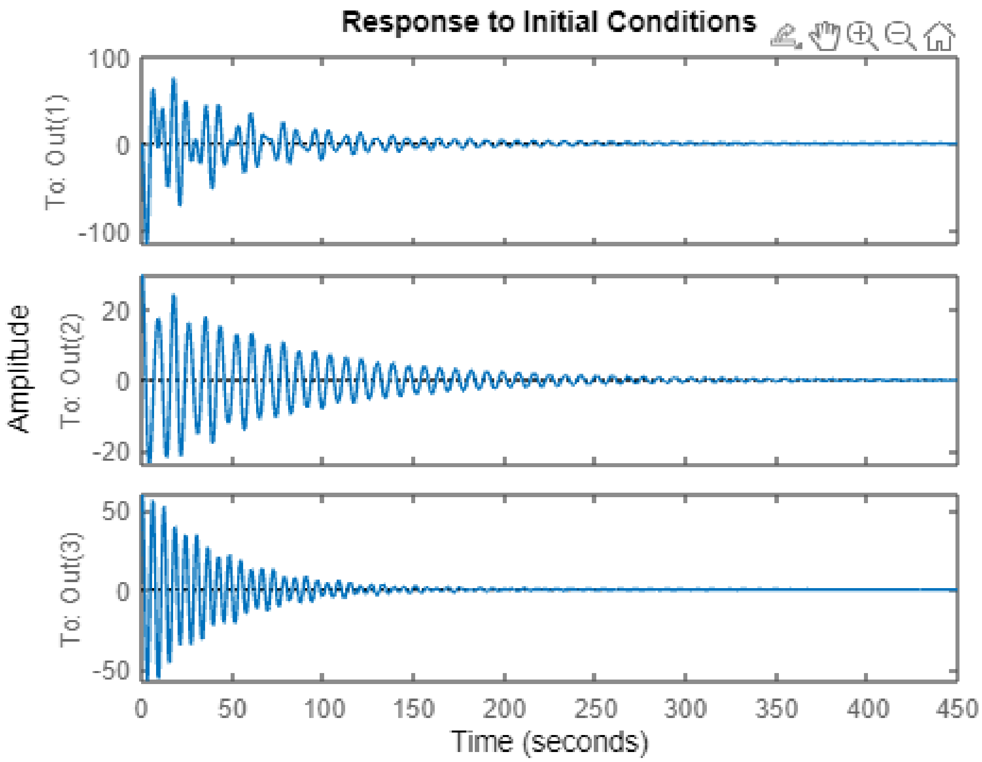
```
sys_3 = ss(A_c3, B_c, C_c3,D);%MATLAB function to output statespace equations
figure
initial(sys_3,x0)
%MATLAB inbuilt function to check the initial response of the system
```



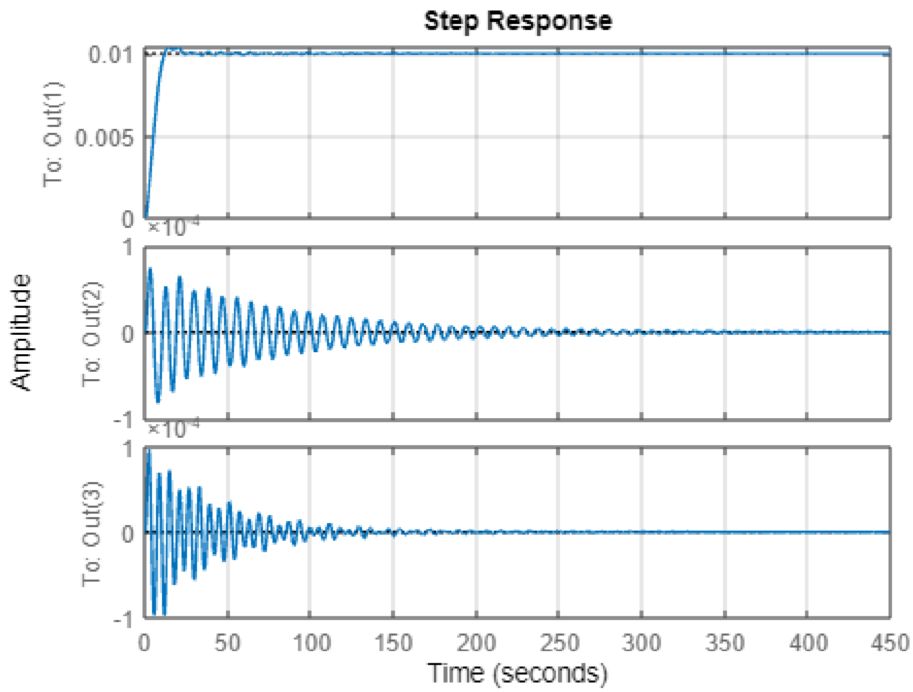
```
figure
step(sys_3)%Gives the step response output
```



```
sys_4 = ss(A_c4, B_c, C_c4, D);%%MATLAB function to output statespace equations
figure
initial(sys_4,x0)
%MATLAB inbuilt function to check the initial response of the system
```



```
figure
step(sys_4)%Gives the step response output
grid on
```

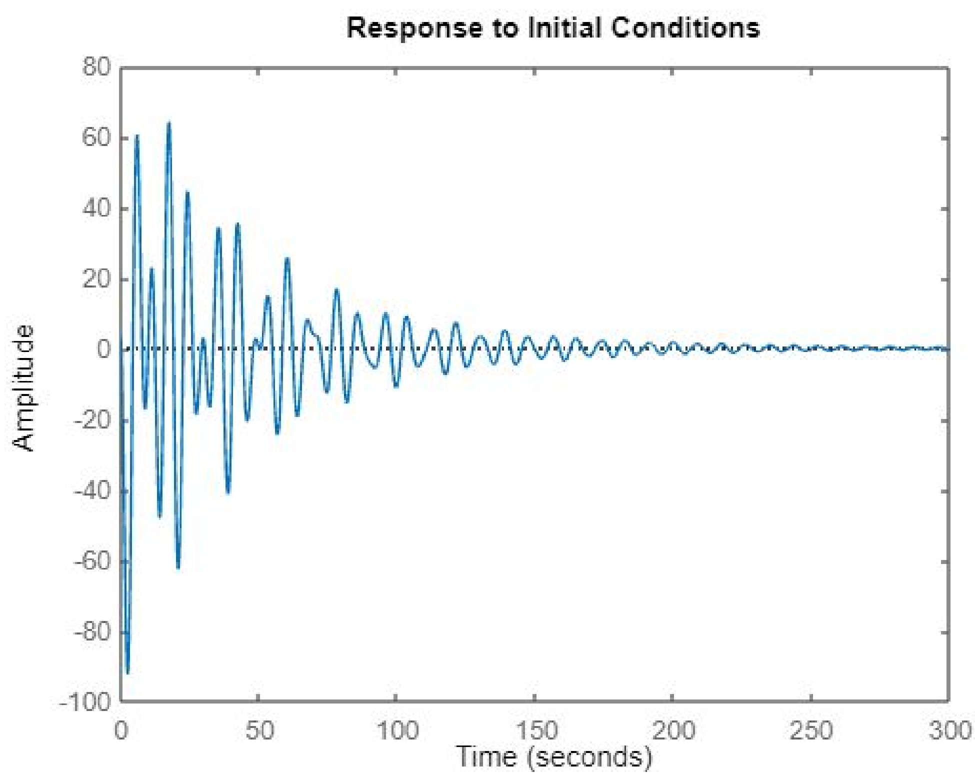


8 LQG Controller Design

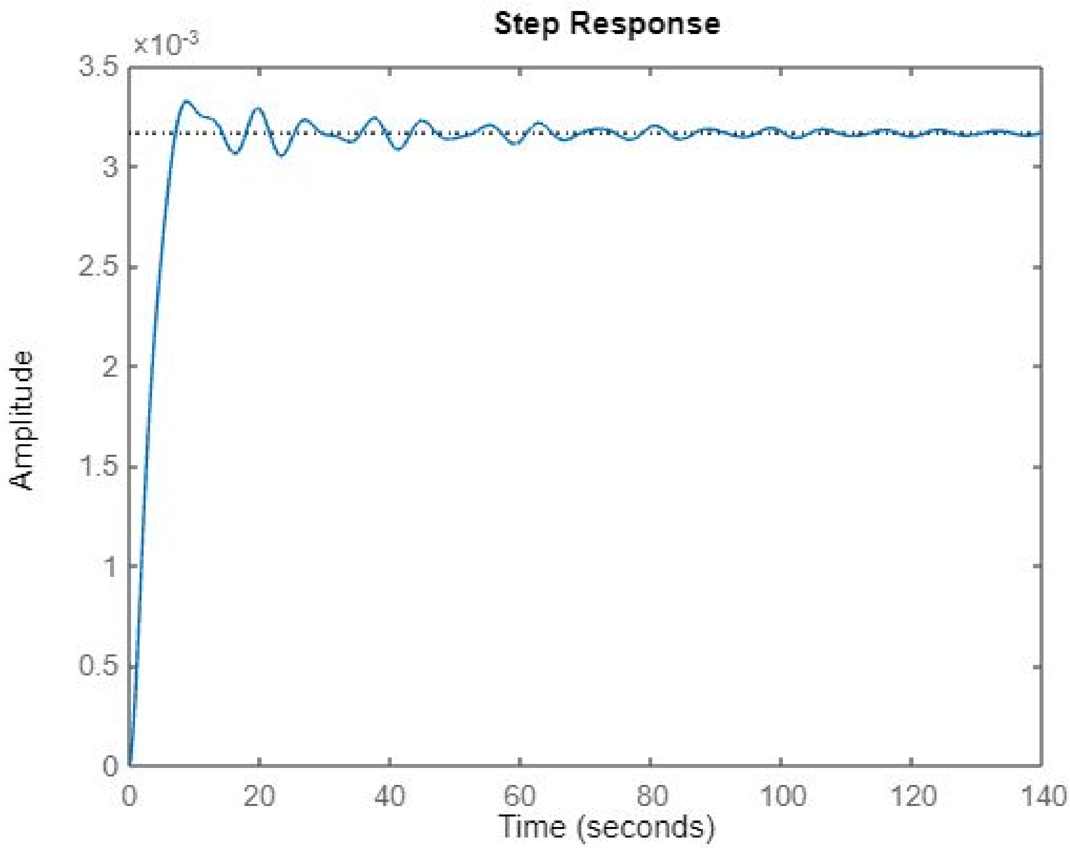
8.1 Linear Part

```
%Defining our matrices as follows
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
% Cosidering the same Q and R matrices chosen before in our code
Q=[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];
R=0.001; %these are the cost variables from LQR
% From previous case, we have determined that only C1, C3 and C4 were
% observable. Hence, we are going to consider only those 3 cases.
C1 = [1 0 0 0 0 0];  %Corresponding to x component
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %corresponding to x and theat2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %corresponding to x, theta1 and theat2
D = 0;
% Initial Conditions for Leunberger observer - 12 state variables,
% 6 actual + 6 estimates
x0 = [4;0;30;0;60;0;0;0;0;0;0;0];
% Calling LQR function to obtain K matrix
K =lqr(A,B,Q,R);
vd=0.3*eye(6); %process noise
vn=1; %measurement noise
```

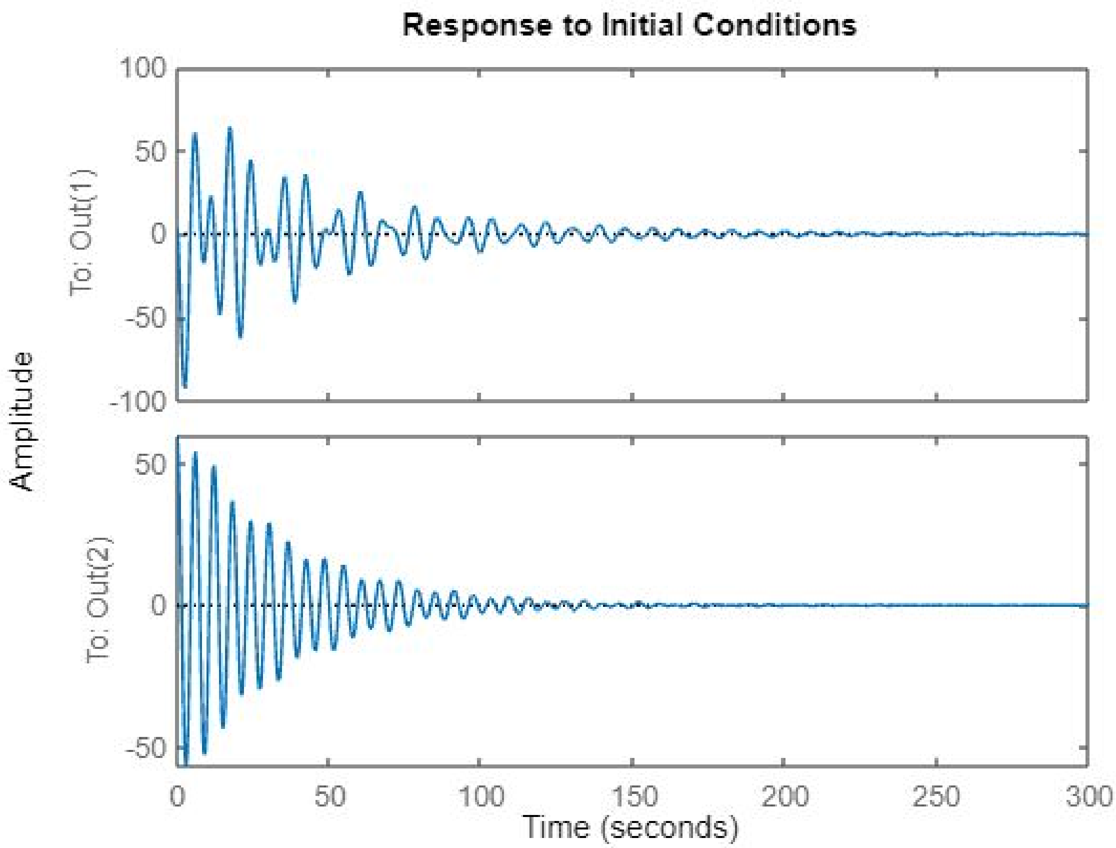
```
K_pop1=lqr(A',C1',vd,vn)'; %gain matrix of kalman filter for C1
K_pop3=lqr(A',C3',vd,vn)'; %gain matrix of kalman filter for C3
K_pop4=lqr(A',C4',vd,vn)'; %gain matrix of kalman filter for C4

% Observing state space corresponding to C1 observable of the system
sys1 = ss([(A-B*K) B*K; zeros(size(A)) (A-K_pop1*C1)],
[B;zeros(size(B))],[C1 zeros(size(C1))], D);
figure
initial(sys1,x0)
```



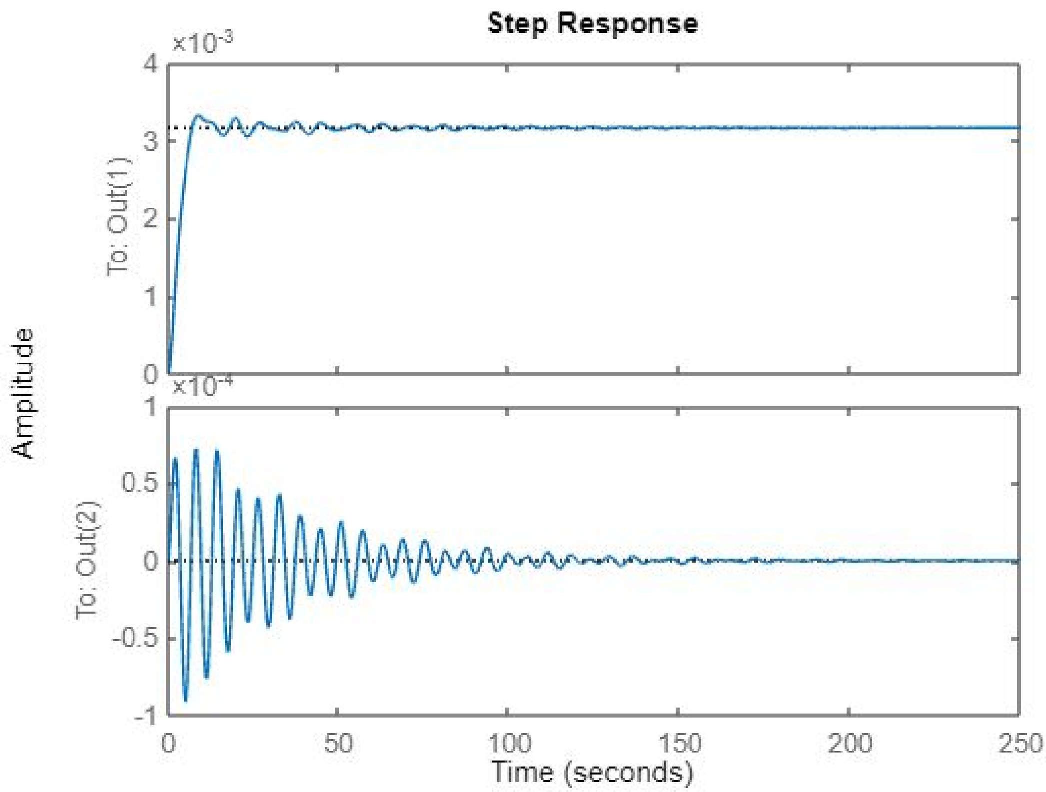
```
figure
step(sys1)
```



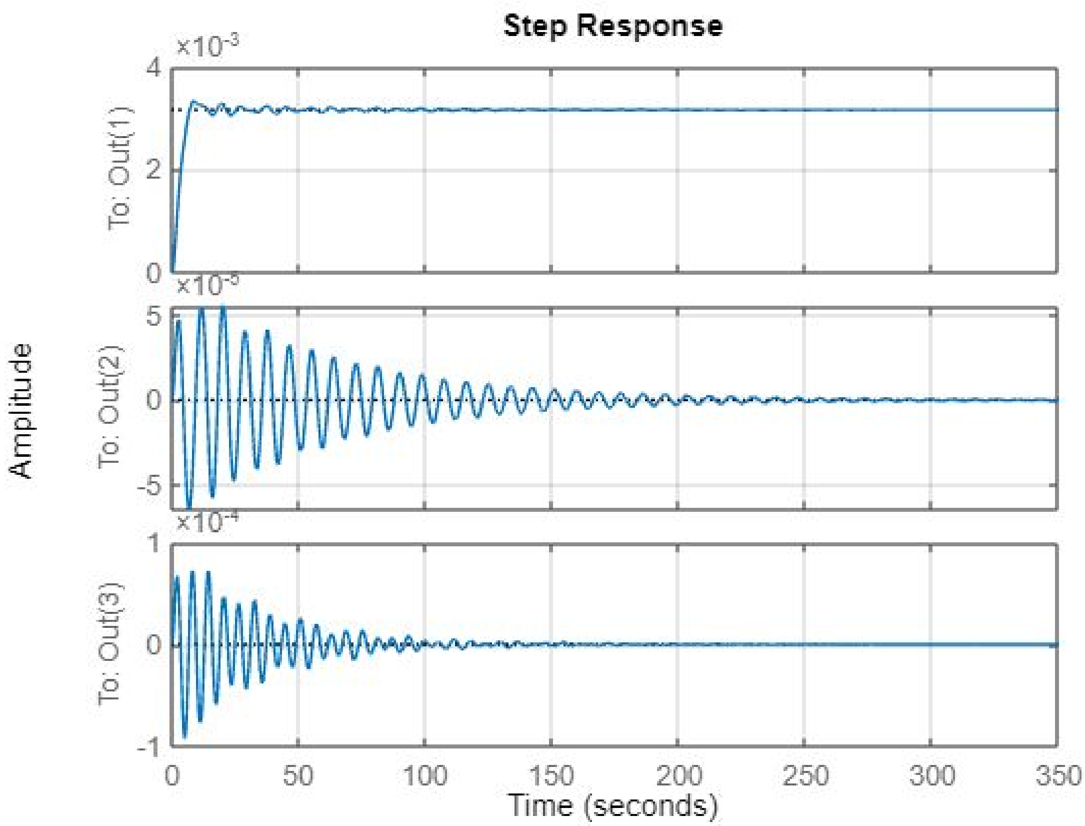
```
% Observing state space corresponding to C3 observable of the system
sys3 = ss([(A-B*K) B*K; zeros(size(A)) (A-K_pop3*C3)],
[B;zeros(size(B))],[C3 zeros(size(C3))], D);
figure
initial(sys3,x0)
```



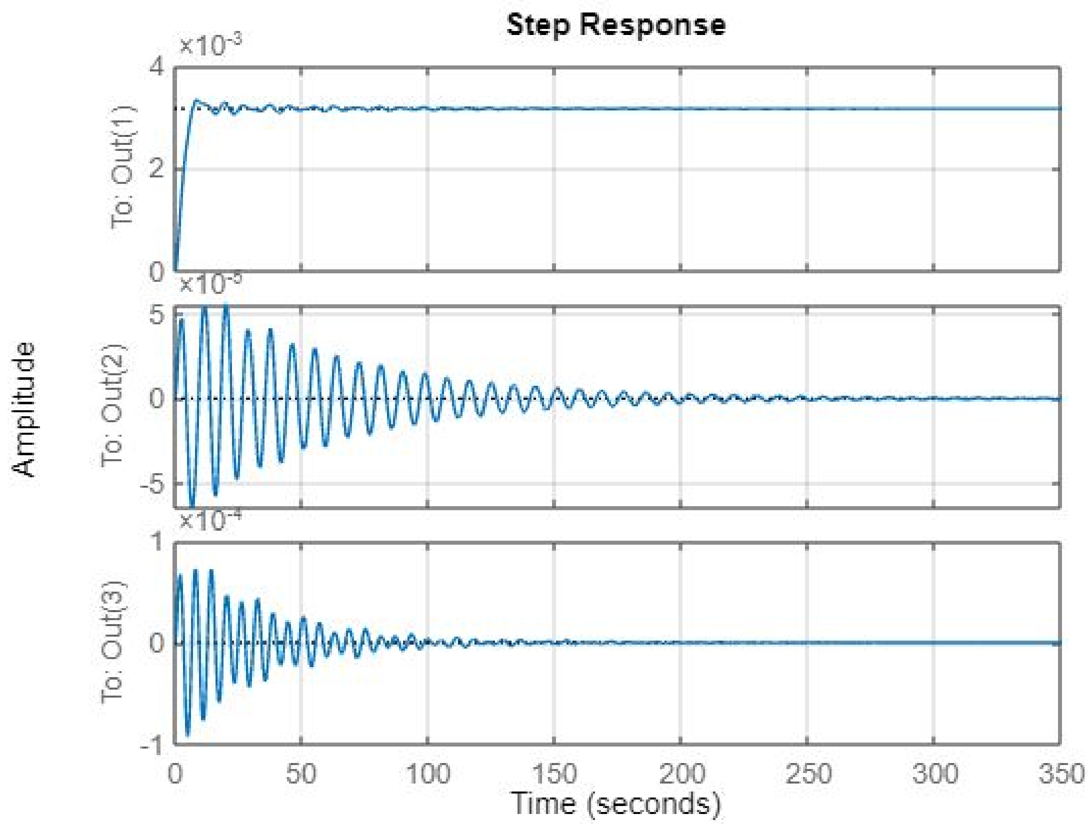
```
figure
step(sys3)
```



```
% Observing state space corresponding to C4 observable of the system
sys4 = ss([(A-B*K) B*K; zeros(size(A)) (A-K_pop4*C4)],
[B;zeros(size(B))],[C4 zeros(size(C4))], D);
figure
initial(sys4,x0)
```



```
figure
step(sys4)

grid on
```

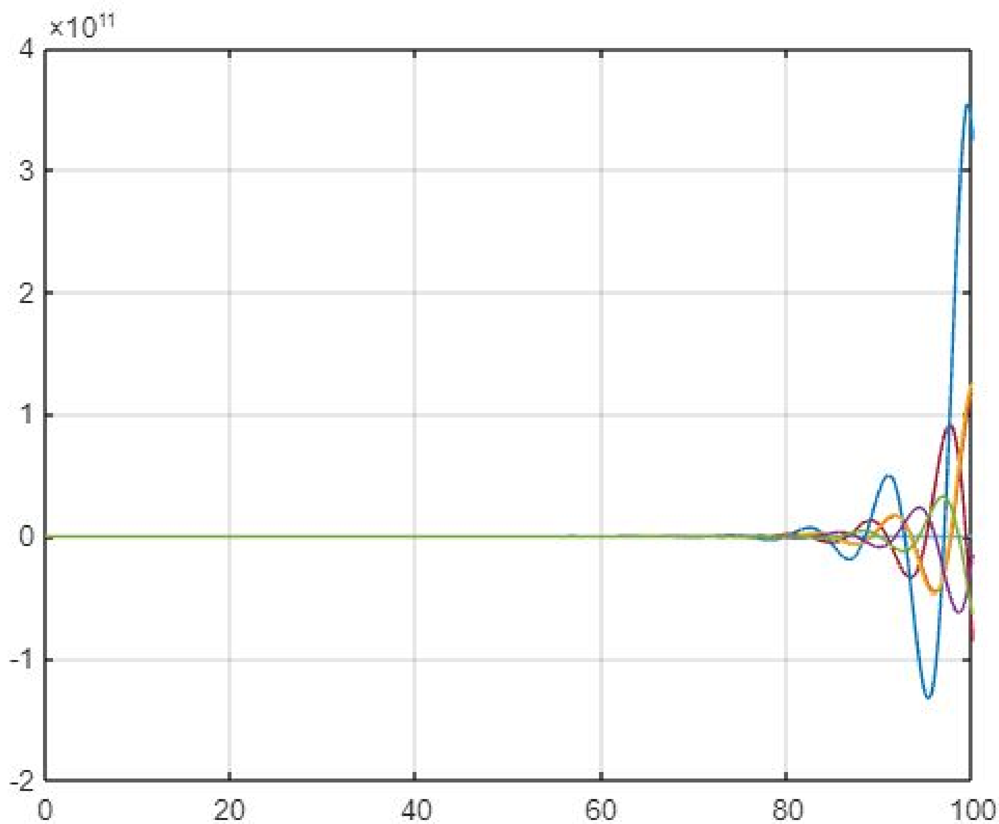


8.2 Non Linear Part

```
x_initial = [0;0;30;0;60;0;0;0;0;0;0;0]
```

```
x_initial = 12x1
     0
     0
    30
     0
    60
     0
     0
     0
     0
     0
```

```
tspan=0:0.1:100;
[t,x] = ode45(@twoload_lqg,tspan,x_initial);
plot(t,x)
grid on
```



```
%% Function doublepend_lqg for non linear LQG control
```

```
function dydt = twoload_lqg(t,y)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
Q=[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];
R=0.01;% From previous case, we have determined that only C1, C3 and C4 were
% observable. Hence, we are going to consider only those 3 cases.
C1 = [1 0 0 0 0 0];  %Corresponding to x component
%C2 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %corresponding to x and theat2
% C3 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %corresponding to x, theta1 and theat2
D = 0;
K_val = lqr(A,B,Q,R);
F=-K_val*y(1:6);

vd=0.3*eye(6);
vn=1;
K_pop=lqr(A',C1',vd,vn)';
%K_pop3 = lqr(A',C2',vd, vn)';

sd =(A-K_pop*C1)*y(7:12);
%sd2 =(A-K_pop3*C1)*y(7:12);

dydt=zeros(12,1);
% y(1)=x; y(2)=xdot; y(3)=theta1;   y(4)=theta1dot;
% y(5)=theta2;    y(6)=theta2dot;
dydt(1) = y(2);%XD;
dydt(2)=(F-(g/2)*(m1*sind(2*y(3))+m2*sind(2*y(5)))-(m1*l1*(y(4)^2)*sind(y(3)))
-(m2*l2*(y(6)^2)*sind(y(5)))))/(M+m1*((sind(y(3)))^2)+m2*((sind(y(5)))^2));%xDD
dydt(3)= y(4);%theta 1D;
dydt(4)= (dydt(2)*cosd(y(3))-g*(sind(y(3))))/l1';%theta 1 Ddot;
dydt(5)= y(6);%theta 2D
dydt(6)= (dydt(2)*cosd(y(5))-g*(sind(y(5))))/l2;%theta 2Ddot;
dydt(7)= y(2)-y(10);
```

```
dydt(8)= dydt(2)-sd(2);
dydt(9)= y(4)-y(11);
dydt(10)= dydt(4)-sd(4);
dydt(11)= y(6)-y(12);
dydt(12)= dydt(6)-sd(6);
end
```

- To asymptotically track a constant reference on x we re-configure our controller: For the most optimal Reference Tracking, our aim is to minimize the following cost function:

  $$\int_0^{\infty} (X(t) - X(d))^T(t)Q(X(t) - X(d)) + (Uk - U\infty)^T R(Uk - U\infty)dt$$

  LQG and LQR sections of the controller are modified to minimize the above mentioned cost function. This gives us the required Optimal Reference Tracking.

- Yes, this design will accommodate constant force disturbances applied on the Cart. Under the assumption that the force disturbances are Gaussian in nature, the controller will account for these disturbances.