UNIVERSITY OF
MARYLAND

ENPM667 - Control of Robotic Systems Final Project on Controllers

AUTHOR:
Sumedh Reddy Koppula, UID:117386066

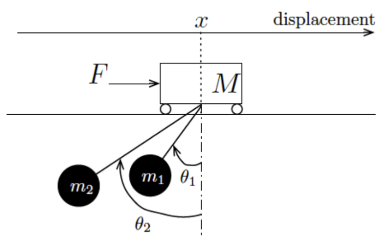
University of Maryland- College Park, 2021
Supervisor: Dr. Waseem A. Malik, PhD, Maryland Robotics Center
INSTITUTE FOR SYSTEMS RESEARCH

Contents

Chapter 1

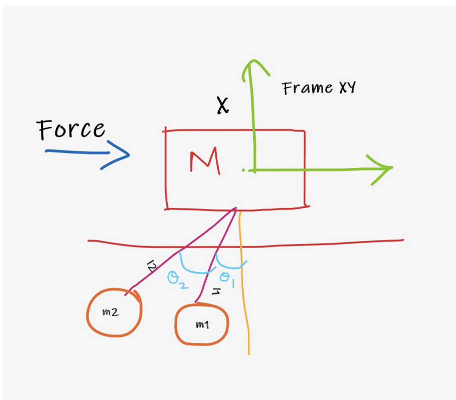
Equations of motion for the system

Given a crane that moves along an one-dimensional track. Which behaves as a frictionless cart with mass M actuated by an external force F that constitutes the input of the system. It is also given that 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.



Crane with two loads m1 and m2

In order to obtain states of the system, Initially I will obtain dynamic equations of the system by calculating Euler Lagrange equations. The probable states of the system are linear velocity of the crane, linear acceleration of the crane with respect to the masses m1, m2 of the loads attached.



Crane with two loads m1 and m2

From the above figure, For the frame XY of the system, The dynamic equations can be written using Euler Lagrange equations.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{X}}\right) - \frac{\partial L}{\partial X} \tag{1.1}$$

Here X is the state variables

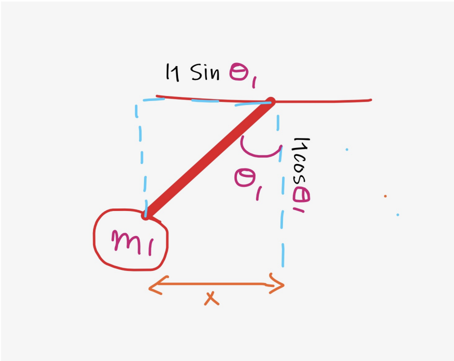
$$\textbf{L = total kinetic energy the system - total potential energy of the system} \tag{1.2}$$

Now, I will calculate the total kinetic energy of the system. We know that, kinetic energy of the system is given by

$$K = \frac{1}{2}M * (\frac{dx}{dt})^2 \tag{1.3}$$

$$K_{total} = K_1 + K_2 + K_{crane} \tag{1.4}$$

In order to find total Kinetic Energy of the system, Initially, Let me calculate the KE_1 of the pendulum with mass m1. Let the position for the pendulum with mass m_1 w.r.t to the frame XY be x_1



position of m1 wrt the crane

$$x_1 = x - l_1 Sin(\theta_1) + l_1 Cos(\theta_1) \tag{1.5}$$

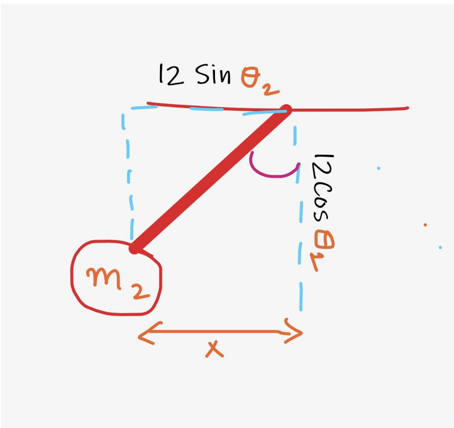
$$K_1 = \frac{1}{2}M * (\frac{dx_1}{dt})^2 \tag{1.6}$$

$$K_1 = \frac{1}{2}M * (\frac{d}{dt}(x - l_1 Sin(\theta_1) + l_1 Cos(\theta_1)))^2 \tag{1.7}$$

$$K_1 = \frac{1}{2}M * (\dot{x} - \cos\theta_1\dot{\theta}_1 - l_1 \sin\theta_1\dot{\theta}_1)^2 \tag{1.8}$$

$$K_1 = \frac{1}{2}M * (\dot{x}^2 + l_1^2\dot{\theta}_1^2 - 2 * \dot{x}l_1\dot{\theta}_1\cos\theta_1) \tag{1.9}$$

Similarly, In order to calculate KE_2 of the pendulum with mass m2. Let the position for the pendulum with mass m_1 w.r.t to the frame XY be x_2



position of m1 wrt the crane

$$x_2 = x - l_2 Sin(\theta_1) + l_2 Cos(\theta_2) \tag{1.10}$$

$$K_2 = \frac{1}{2}M * (\frac{dx_2}{dt})^2 \tag{1.11}$$

$$K_2 = \frac{1}{2}M * (\frac{d}{dt}(x - l_2 Sin(\theta_2) + l_2 Cos(\theta_2)))^2 \tag{1.12}$$

$$K_2 = \frac{1}{2}M * (\dot{x} - \cos\theta_2\dot{\theta}_2 - l_2\sin\theta_2\dot{\theta}_2)^2 \tag{1.13}$$

$$K_2 = \frac{1}{2}M * (\dot{x}^2 + l_2^2\dot{\theta}_2^2 - 2 * \dot{x}l_2\dot{\theta}_2\cos\theta_2) \tag{1.14}$$

Similarly, Let me calculate the Kinetic energy for the crane

$$K_{crane} = \frac{1}{2}M * (\dot{x})^2 \tag{1.15}$$

Total Kinetic Energy is equal to (from equation (1.3))

$$= [\frac{1}{2}M * (\dot{x}^2 + l_1^2\dot{\theta}_1^2 - 2 * \dot{x}l_1\dot{\theta}_1\cos\theta_1] + [\frac{1}{2}M * (\dot{x}^2 + l_2^2\dot{\theta}_2^2 - 2 * \dot{x}l_2\dot{\theta}_2\cos\theta_2)] + [\frac{1}{2}M * (\dot{x})^2]$$

(1.16)

Now, Let me calculate Potential energy of the system, The total potential energy of the system is given by

$$P_{total} = P_1 + P_2 + P_{crane} \tag{1.17}$$

Potential energy P_1 of the pendulum with mass m1 is:

$$P_1 = (m_1)(g)(l_1)(1 - (\cos\theta_1)) \tag{1.18}$$

Similarly, Potential energy P_2 of the pendulum with mass m_2 is:

$$P_2 = (m_2)(g)(l_2)(1 - (\cos\theta_2)) \tag{1.19}$$

Potential energy of the crane P_{crane} is zero as the $h_{reference} = 0$

$$P_{total} = (m_1)(g)(l_1)(1 - (\cos\theta_1)) + (m_2)(g)(l_2)(1 - (\cos\theta_2)) \tag{1.20}$$

Substituting equation 1.15 and 1.19 in 1.1 to get Lagrange equation
$L = [\frac{1}{2}M * (\dot{x}^2 + l_1^2\dot{\theta}_1^2 - 2 * \dot{x}l_1\dot{\theta}_1\cos\theta_1] + [\frac{1}{2}M * (\dot{x}^2 + l_2^2\dot{\theta}_2^2 - 2 * \dot{x}l_2\dot{\theta}_2\cos\theta_2)] + [\frac{1}{2}M * (\dot{x})^2]$
$- [(m_1)(g)(l_1)(1 - (\cos\theta_1)) + (m_2)(g)(l_2)(1 - (\cos\theta_2))]$
Now, Let me Calculate

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{X}}) - \frac{\partial L}{\partial X} \tag{1.21}$$

For this system, The state variables X are x, θ_1 and θ_2
Since the crane is acted upon Force F The Euler Lagrange equation is given as

$$F = x[\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}})] - \frac{\partial L}{\partial x} \tag{1.22}$$

Lets find, $(\frac{\partial L}{\partial \dot{x}})$

$$(\frac{\partial L}{\partial \dot{x}}) = \tag{1.23}$$

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

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}}) = \quad (1.24)$$

$((\ddot{x}))(M + m_1 + m_2) - [(m_1)(l_1)((\ddot{\theta_1}))\cos(\theta_2) + (m_2)(l_2)(\ddot{\theta_1}^2\cos\theta_2) + (m_1)(l1)(\dot{\theta}^2\sin\theta_1) + (m_2)(l_2)(\dot{\theta_2}^2\sin\theta_2)$

similarly, for other states θ_1 and θ_2 The Lagrange equation is given by

$$0 = x[\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_1}})] - \frac{\partial L}{\partial \theta_1} \quad (1.25)$$

$$0 = x[\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_2}})] - \frac{\partial L}{\partial \theta_2} \quad (1.26)$$

$$\frac{\partial L}{\partial \theta_1} = m_1(\dot{x})l_1\dot{\theta}_1\sin(\theta_1) - m_1 g l_1 \sin\theta_1 \quad (1.27)$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_1}}) = -m_1(\ddot{x}l_1\cos\theta_1) + m_1 l_1^2\ddot{\theta}_1 + m_1\dot{x}l_1\dot{\theta}_1\sin\theta_1 \quad (1.28)$$

$$\frac{\partial L}{\partial \theta_2} = m_2(\dot{x})l_2\dot{\theta}_2\sin(\theta_2) - m_2 g l_2 \sin\theta_2 \quad (1.29)$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_1}}) = -m_2(\ddot{x}l_2\cos\theta_2) + m_2 l_2^2\ddot{\theta}_2 + m_2\dot{x}l_2\dot{\theta}_2\sin\theta_2 \quad (1.30)$$

substituting equations 1.27 and 1.28 in 1.25. similarly, substituting equations 1.29 and 1.30 in 1.26
on solving equation 1.25 and 1.26 for $\ddot{\theta_1}$ and $\ddot{\theta_2}$

$$\ddot{\theta_1} = \frac{(m_1)(\ddot{x})(l_1)(\cos(\theta_1)) - (m_1)(g)(l_1)(\sin(\theta_1))}{m_1 l_1^2} \quad (1.31)$$

$$\ddot{\theta_2} = \frac{(m_2)(\ddot{x})(l_2)(\cos(\theta_2)) - (m_2)(g)(l_2)(\sin(\theta_2))}{m_2 l_2^2} \quad (1.32)$$

substituting $\ddot{\theta_1}$ and $\ddot{\theta_2}$ in equation 1.24 in order to find $\ddot{x}$

$$\ddot{x} = \frac{F - (m_1)(g)(\sin\theta_1)(\cos\theta_1) - (m_2)(g)(\sin\theta_2)(\cos\theta_2) - (m_1)(l_1)(\sin\theta_1)(\dot{\theta}_1^2) - (m_2)(l_2)(\sin\theta_2)(\dot{\theta}_2^2)}{M + m_1 + m_2 - (m_1\cos{\theta_1}^2) - (m_2\cos{\theta_2}^2)} \quad (1.33)$$

substituting $\ddot{x}$ in $\ddot{\theta_1}$ and $\ddot{\theta_2}$

$$\ddot{\theta_1} = \frac{F\cos\theta_2 - (m_1)(g)(\sin\theta_1)(cos\theta_1)(\cos\theta_2) - (m_2)(g)(\sin\theta_2)(cos\theta_2)(\cos\theta_2)}{l_1(M + m_1\sin{\theta_1}^2) + m_2\sin{\theta_2}^2}$$

(1.34)

$$\frac{(m_1)(l_1)(\sin\theta_1)(cos\theta_1)(\cos\theta_1)\dot{\theta}_1^2 - (m_2)(l_2)(\sin\theta_2)(cos\theta_2)(\cos\theta_2)\dot{\theta}_2^2}{} - \frac{g\sin\theta_2}{l_2} \quad (1.35)$$

$$\ddot{\theta_2} = \frac{F\cos\theta_1 - (m_2)(g)(\sin\theta_2)(cos\theta_1)(\cos\theta_2) - (m_1)(g)(\sin\theta_1)(cos\theta_1)(\cos\theta_1)}{l_2(M + m_1\sin{\theta_1}^2) + m_2\sin{\theta_2}^2}$$

(1.36)

$$\frac{(m_1)(l_1)(\sin\theta_1)(cos\theta_1)(\cos\theta_1)\dot{\theta}_1^2 - (m_2)(l_2)(\sin\theta_2)(cos\theta_2)(\cos\theta_2)\dot{\theta}_2^2}{} - \frac{g\sin\theta_1}{l_1} \quad (1.37)$$

Chapter 2

State-Space representation of the linearized system

It is given that system should be linearized around the equilibrium point specified by x = 0 and $\theta_1 = \theta_2 = 0$, We can linearize by neglecting higher powers and orders of differentiation of θ_1 and θ_2 i.e $\sin\theta_1 = \theta_1$ and $\sin\theta_2 = \theta_2$. similarly $\sin\theta_1 = \sin\theta_2 = 1$
Hence our second order differential equation [1.33] further simplifies as follows

$$\ddot{x} = \frac{F - (m_1)(g)(\theta_1) - (m_2)(g)(\theta_2)}{M} \tag{2.1}$$

Further simplifying other states θ_1 and θ_2

$$\ddot{\theta_1} = \frac{F - (m_1)(g)(\theta_1) - (m_2)(g)(\theta_2) - (M)(g)(\theta_2)}{M(l_1)} \tag{2.2}$$

$$\ddot{\theta_2} = \frac{F - (m_1)(g)(\theta_1) - (m_2)(g)(\theta_2) - (M)(g)(\theta_2)}{M(l_2)} \tag{2.3}$$

From the standard state space equation

$$\dot{X(t)} = AX(t) + BU(t) \tag{2.4}$$

$$Y(t) = CX(t) + DU(t) \tag{2.5}$$

Input to the system is Force F, Now lets choose our **state variables X(t)** as x, $\dot{x}$, θ_1, $\dot{\theta_1}$, θ_2, $\dot{\theta_2}$ such that it satisfies our requirements.

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta_1} \\ \ddot{\theta_1} \\ \dot{\theta_2} \\ \ddot{\theta_2} \end{bmatrix} = A \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta_1} \\ \theta_2 \\ \dot{\theta_2} \end{bmatrix} + BF$$

From above calculated equations we can obtain $\dot{X}$ as

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \frac{F-(m_1)(g)(\theta_1)}{M} \\ \dot{\theta_1} \\ \frac{F-(m_1)(g)(\theta_1)-(m_2)(g)(\theta_2)-(M)(g)(\theta_2)}{M(l_1)} \\ \dot{\theta_2} \\ \frac{F-(m_1)(g)(\theta_1)-(m_2)(g)(\theta_2)-(M)(g)(\theta_2)}{M(l_2)} \end{bmatrix}$$

This will be sufficient to form our state space equation with A and B

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

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + DF$$

Chapter 3

Conditions on M,m1,m2,l1,l2 for which the linearized system is controllable

As we know the condition for system to be controllable is the matrix $C_{control}$ must be a full matrix.

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

```
#Code to calculate determinant of Matrix C_control
from sympy import *
import math

m1, m2, g, M, l1, l2 = symbols("m1 m2 g M l1 l2")

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

CControl = Matrix([[B], [A*B], [(A**2)*B], [(A**2)*B], [(A**3)*B], ...
    [(A**4)*B], [(A**5)*B]])
Cdet= CControl.det()
pprint(CControl)
pprint(Cdet)
```

The determinant of $C_{control} \neq 0$ for a full rank matrix and the determinant of $C_{control}$ is

$$\frac{g^6(l_1 - 1_2)^2}{(M^6(l_1^6)(l_2^6))} \neq 0 \tag{3.1}$$

From this equation we arrive to a conclusion that, there exists a condition for complete controllable is $l_1 \neq l_2$. In other words, $l_1 = l_2$ is the condition when the system is not controllable.

Chapter 4

Design of LQR Controller: Controllability and Stability Analysis

4.0.1 Controllabilty check for Linearized system

It is given in the question that, we can choose M = 1000Kg, m1 = m2 = 100Kg, l1 = 20m and l2 = 10m. Let us substitute the values into our state space equation. The linear state equation is

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} = A \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{11}{20} & 0 & -\frac{1}{20} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{10} & 0 & -\frac{11}{10} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{1000} \\ 0 \\ \frac{1}{20000} \\ 0 \\ \frac{1}{10000} \end{bmatrix} F$$

```
# import sympy
# Copyright Sumedh Koppula
from sympy import *

# Use sympy.eigenvals() method
eigen_matrix = Matrix([[0, 1, 0, 0, 0, 0], [0, 0, -1, 0, -1, 0], [0, 0, 0, ...
    1, 0, 0], [0, 1, -(11/20), 0, -(1/20), 0], [0, 0, 0, 0, 0, 1], [0, 0, ...
    -(1/10), 0, -(11/10), 0]])
result = eigen_matrix.eigenvals()

pprint(result)
```

$$eigenMatrix = \begin{bmatrix} 0.000 + i(0.000) \\ -0.7970 + i(4.9429) \\ 0.02317 - i(1.00875) \\ 0.02317 + i(1.0087) \\ 0.3753 - i(1.04472) \\ 0.3753 + i(1.0447) \end{bmatrix}$$

After analysing the eigen values, seems like every egien value has zero real part, which indicates that the system is locally stable about equilibrium point.
Lets examine the controllablity of the system by calculating the rank of $C_{control}$ Matrix

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

```
# Executed code in matlab
A = [0 1 0 0 0 0;
     0 0 -1 0 -1 0;
     0 0 0 1 0 0;
     0 0 -0.55 0 -0.05 0;
     0 0 0 0 0 1;
     0 0 -0.1 0 -1.1 0 ];
B = [0; 0.001; 0; 0.00005; 0; 1/0.0001];

% Check controllability matrix
cControl = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
disp(cControl)
```

$$C_{control} = 1.0e+04 \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 1.15 \\ 0 & 0 & -1 & 0 & 1.15 & 0 \\ 0 & 0 & 0 & -0.05 & 0 & 0.0825 \\ 0 & 0 & -0.05 & 0 & 0.0825 & 0 \\ 0 & 1 & 0 & -1.1 & 0 & 1.215 \\ 1 & 0 & -1.1 & 0 & 1.215 & 0 \end{bmatrix}$$

The rank of the $C_{control}$ matrix is full rank. However, Lets try to verify the controllability using PBH test for the robustness in the statement.

$$[(\lambda I - A)|B] = \begin{bmatrix} \lambda & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 1 & 0 & 1 & 0 & 0.001 \\ 0 & 1 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.55 & \lambda & 0.05 & 0 & 0.00005 \\ 0 & 0 & 0 & 0 & \lambda & 1 & 0 \\ 0 & 0 & 0.1 & 0 & 1.1 & \lambda & 0.0001 \end{bmatrix}$$

The above matrix is full rank $\forall \lambda \in C_{control}$. Hence, we can state that the system is controllable.

4.0.2 Design of LQR controller

Now, Lets calculate the optimal solution for the LQR controller of the linearized system. For this, Initially, lets calculate cost function which later can be used to minimize the cost function.

$$J = \int_0^\infty (X^T(t) + U^T(t)RU(t))dt \tag{4.1}$$

The optimal solution is given by the following controller:

$$K = -R^{-1}B_K^T P \tag{4.2}$$

The state feedback equation is given by

$$U(t) = -KX(t) = -R^{-1}B_K^T PX(t) \tag{4.3}$$

where P is the symmetric positive solution of the following stationary Riccati equation

$$A^T P + PA - PBR^{-1}B_T P = -Q \tag{4.4}$$

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + DF$$

From the output state equation, We can find the value of **Q**

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \end{bmatrix} + DF$$

$$Q = C^T C \tag{4.5}$$

where C is obtained from above output equation

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

the value of **R** is considered to be 1.5, But I will Adjust the parameters of the LQR until I obtain a suitable response. The below is the initial gain matrix with R = 1.5

$$K = \begin{bmatrix} 0.8165 & 44.2637 & -1.2817 & -71.5884 & -0.6476 & -35.9934 \end{bmatrix}$$

4.0.3 Analysing stablity of the system using Lyapunov's indirect method

The eigen values for the inital gain K for R = 1.5 and Q = C^TC are

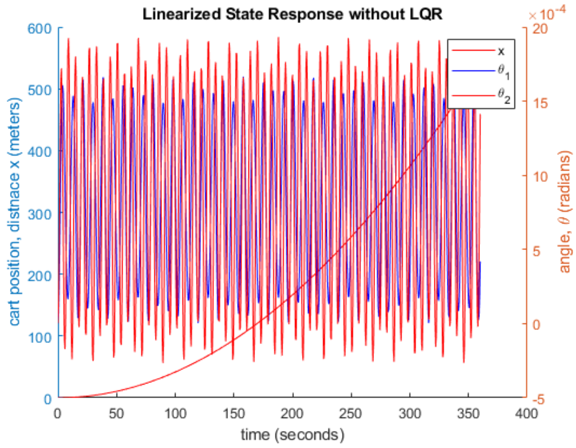
$$A - BK = \begin{bmatrix} -0.0001 + 1.0531i \\ -0.0001 - 1.0531i \\ -0.0184 + 0.0184i \\ -0.0184 - 0.0184i \\ -0.0001 + 0.7356i \\ -0.0001 - 0.7356i \end{bmatrix}$$

Analysis: All the eigen values have negative real part indicating that the system is stable. It can be inferred that since the real part of the eigen values are almost zero, There is huge chance that the system may be marginally stable.

4.0.4 Adjusting parameters of the LQR cost to obtain a suitable response

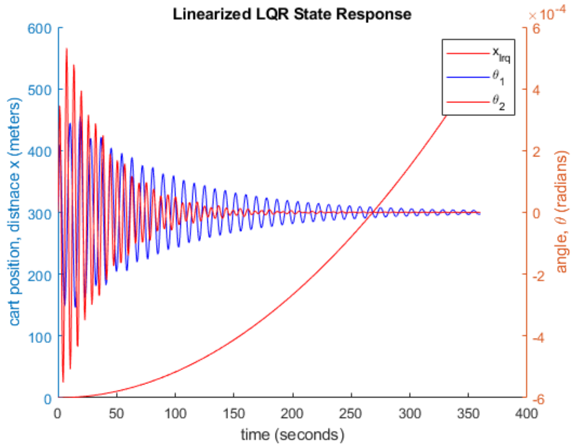
I have simulated the system with and without LQR feedback for 3 minutes by imparting an external force of 10 N, Here are the interesting outcomes I have come up with. The oscillations of the both pendulums have been decreasing in angle after adjusting the parameters of Q and R.

Case 0: I have analysed the system,i.e the cart position and the angles of the loads without LQR feedback,I observed that the system has huge number of oscillations, which is undesired. This can be shown in the below figure.



State Response without LQR

Case 1: Initially, I had given the values of R as 1.5 and Q as [1 0 0 0 0 0 0, 0 0 1 0 0 0, 0 0 0 0 1 0], I observed that the system has few number of undesired oscillations, which is undesired.
Case 2: Later, I had given the values of R = 0.005 i.e decreased R and Q as [9 0 0 0 0 0 0, 0 0 1 0 0 0, 0 0 0 0 1 0] i.e increased Q as to attain desired optimal solution with suitable response , I observed that the system has comparatively less number of undesired oscillations. As shown in the below figure.



State Response with LQR

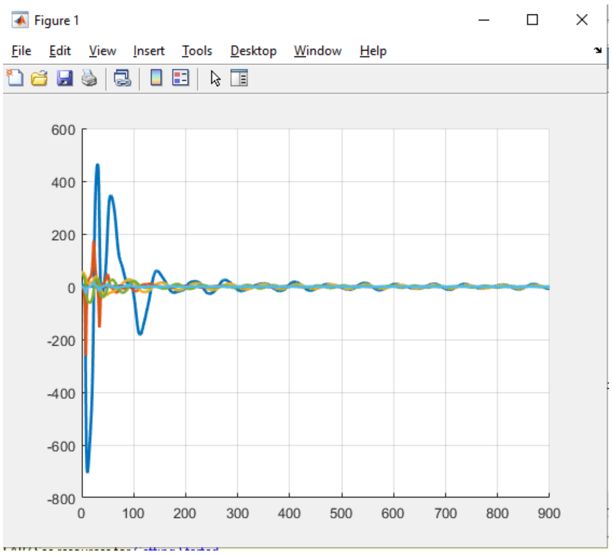
Stability analysis after adjusting the R and Q parameters for feedback system
By adjusting R and Q, It is pretty clear that, the oscillations in the both pendulums have been drastically decreased. Below the optimal solution controller values

$$K = 1.0e + 03 * \begin{bmatrix} 1.3416 & 1.7367 & -0.4337 & -1.1658 & -0.3253 & -0.7382 \end{bmatrix}$$

$$A - BK = \begin{bmatrix} -0.7642 + 0.8316i \\ -0.7642 - 0.8316i \\ -0.0268 + 1.0181i \\ -0.0268 - 1.0181i \\ -0.0113 + 0.7120i \\ -0.0113 - 0.7120i \end{bmatrix}$$

From the obtained eigen values for the above K, It can be inferred that, all the eigen values are in left half of the plane indicating the feedback system is stable.

4.0.5 LQR Resulting Response to the original nonlinear system



State Response of LQR applied to non-linear system

Chapter 5

Observability for Linearized System

In state space the observability matrix is given as

$$O = \begin{bmatrix} C \\ CA \\ CA_2 \\ CA_3 \\ . \\ . \\ CA_{n-1} \end{bmatrix}$$

A system is said to be observable if the O matrix is full rank n. For our linear system, Lets calculate observability for given output vectors

Case 1: state x only, rank = 6

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Case 2: states θ_1 and θ_2, rank = 4

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

Case 3: states x and θ_2, rank = 6

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

Case 4: states x, θ_1 and θ_2, rank = 6

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

```
% observability check
syms mass1 mass2 M l1 l2 F
g=10;
X0 = [0,0,0,0,0,0];

A = [0 1 0 0 0 0;
     0 0 -g*mass1/M 0 -g*mass2/M 0;
     0 0 0 1 0 0;
     0 0 -g/l1*(1+mass1/M) 0 -g/l1*(mass2/M) 0;
     0 0 0 0 0 1;
     0 0 -g/l2*(mass1/M) 0 -g/l2*(1+mass2/M) 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];

% substituting system values
M=1000; mass1=100;mass2=100; l1=20; l2=10; g=10;
A = double(subs(A)); B = double(subs(B));

C_1 = [1 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 0 0];
obs = obsv(A,C_1);
observability_C1 = rank(obs)

C_2 = [0 0 0 0 0 0;
       0 0 1 0 0 0;
       0 0 0 0 1 0];
obs = obsv(A,C_2);
observability_C2 = rank(obs)

C_3 = [1 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 1 0];
obs = obsv(A,C_3);
observability_C3 = rank(obs)

C_4 = [1 0 0 0 0 0;
       0 0 1 0 0 0;
       0 0 0 0 1 0];
obs = obsv(A,C_4);
observability_C4 = rank(obs)

Result:

observability_C1 = 6
observability_C2 = 4
observability_C3 = 6
observability_C4 = 6
```

5.0.1 Luenberger observer for controllable output vectors

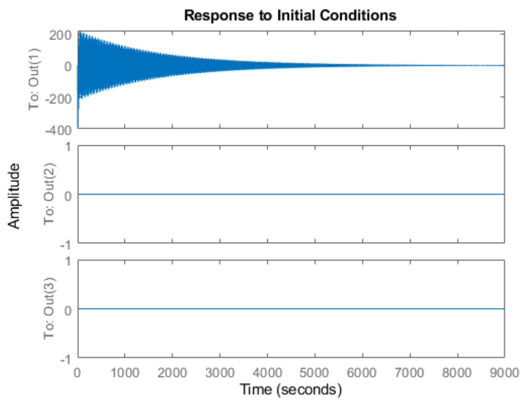
Now, Let me obtain Luenberger observer for each output vectors which are observable. Let me construct Luenberger Observer: We know that closed loop Luenberger Observer is given by the following state-space representation

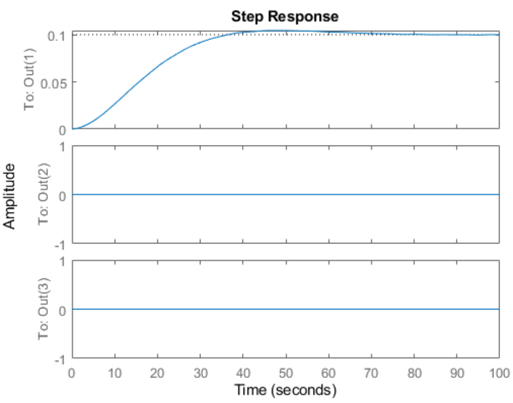
$$\begin{bmatrix} \dot{X} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A + B_k K & -B_k K \\ 0 & A - LC \end{bmatrix} * \begin{bmatrix} X \\ \varepsilon \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U$$

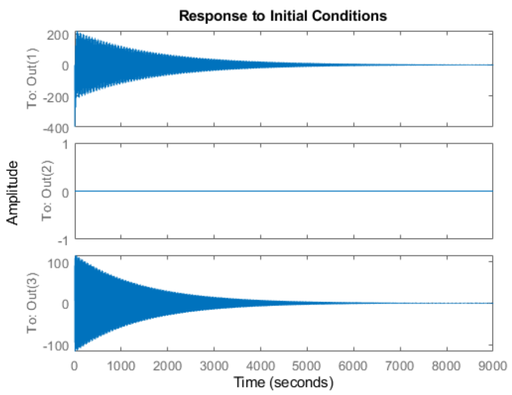
$$\overrightarrow{\dot{\hat{X}}(t)} = A\overrightarrow{U(\hat{t})_k} + B_k\overrightarrow{X(\hat{t})} + L(\overrightarrow{Y(\hat{t})} - C\overrightarrow{X(\hat{t})}) \tag{5.1}$$

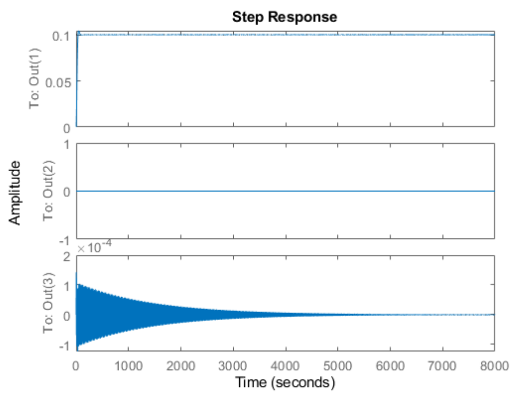
and the error in the observed state is $\dot{\varepsilon}$

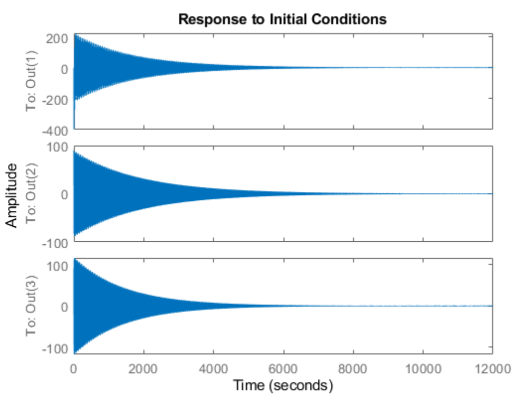
$$\dot{\varepsilon} = \dot{X} - \dot{\hat{X}} = (A - LC)\varepsilon \tag{5.2}$$

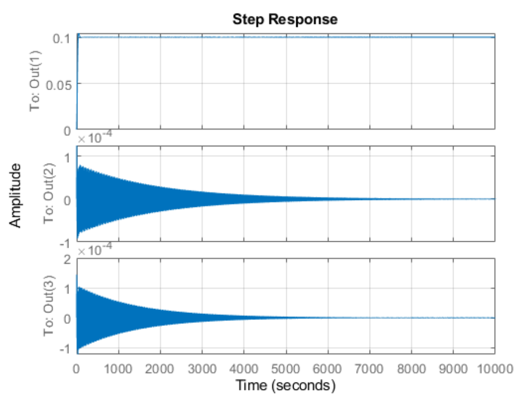


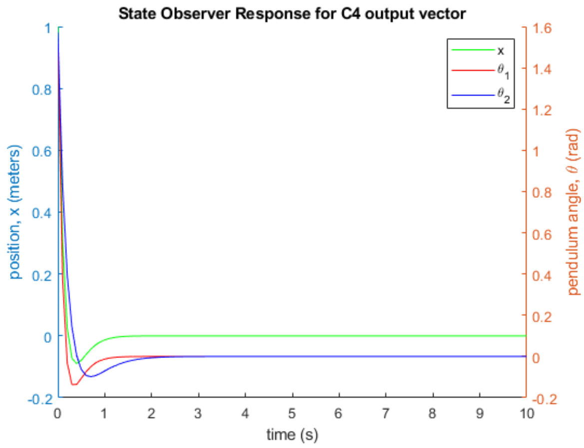












The initial conditions are assumed as $X_{initial} = [0, 0, 90, 0, 90, 0, 0, 0, 0, 0, 0, 0]$. The above figures portrays output for closed loop observer system with initial conditions and with step input for observable output vectors.

Chapter 6

Design of output feedback controller using LQG method

6.0.1 Design of output feedback controller

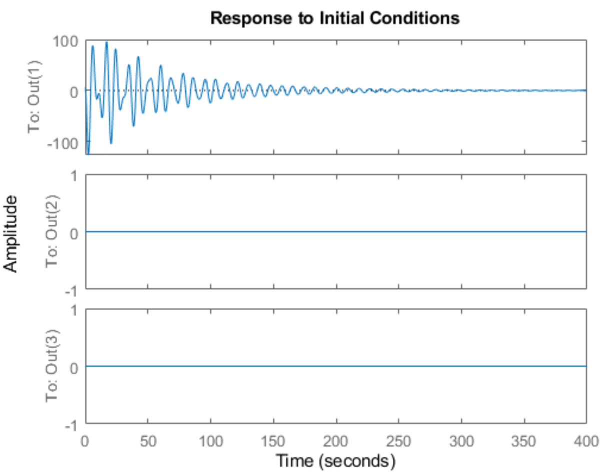
It is given in the question to design a feedback controller using smallest output vector, Hence, I am choosing x output vector with C_1 Observability as discussed in previous sections. The state space equation for LQG is given as

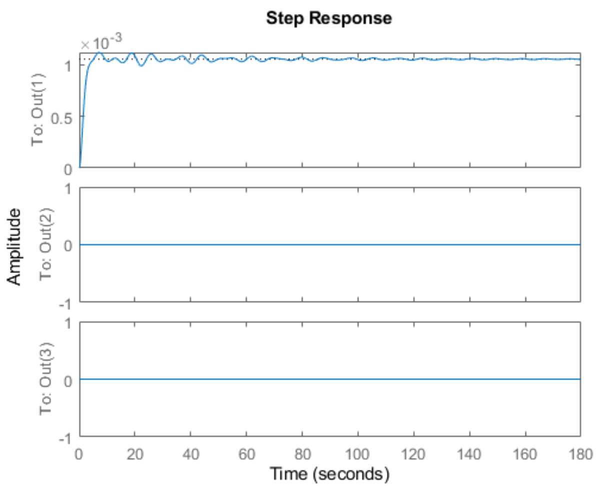
$$\dot{X}(t) = AX(t) + B_k U_k(t) + B_D U_D(t) \tag{6.1}$$

$$Y(t) = CX(t) + V(t) \tag{6.2}$$

$U_D(t)$ is the process noise and V(t) is the Measurement noise and the cost we want to minimize is

$$\lim_{t \to \infty} E[X_T(t)QX(t) + U_T(t)RU(t)] \tag{6.3}$$





The above figures shows the output of LQG controller for step input using separation principle i.e standard output feedback configuration with the Luenberger Observer and the optimal K and L are computed Separately using the LQR and Kalman-Bucy methods. In my code, $gain_k$ is gain matrix of kalman filter.

Chapter 7

Code

7.0.1 Linear LQR

```
% linearized LQR
syms mass1 mass2 M l1 l2 F
g=10;

A = [0 1 0 0 0 0;
     0 0 -g*mass1/M 0 -g*mass2/M 0;
     0 0 0 1 0 0;
     0 0 -g/l1*(1+mass1/M) 0 -g/l1*(mass2/M) 0;
     0 0 0 0 0 1;
     0 0 -g/l2*(mass1/M) 0 -g/l2*(1+mass2/M) 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
control = simplify( [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B] )
control_test = length(A) - rank(control)
var = simplify(det(control))
solve(var==0, M,mass1,mass2,l1,l2, 'ReturnConditions', true);
[ans.M ans.mass1 ans.mass2 ans.l1 ans.l2]
% substituting system values
M=1000; mass1=100;mass2=100; l1=20; l2=10; g=10;
A = double(subs(A)); B = double(subs(B));
% stability check
stability = eig(A)
% controlabilty check after substitution
control = ctrb(A,B)
control_test = det(control)
control_test = length(A) - rank(control)
Q = C'*C;
R=1.5;
[K_init,P_init,e_init] = lqr(A,B,Q,R);
% Stability check using Lyapunov's indirect method
Stablity_Lyp = eig(A-B*K_init)
display(K_init)
R = 0.005;
Q = C'*C ;
display(Q)
% Adjusting the values of Q
Q(1,1) = 9000
[K_load,P_load,e_load] = lqr(A,B,Q,R)
```

```
Stablity_Lyp_load = eigs(A-B*K_load)
display(Stablity_Lyp_load)

% LQR simulation for 3 min
A_feedback = (A-B*K_load);
% state response without LQR
ss_init = ss(A ,B,C,0);
% state response with LQR
ss_lqr = ss(A_feedback,B,C,0);

t = 0:0.5:360;
force = 10*ones(size(t));
[y_init,t,x]=lsim(ss_init,force,t);
[y_lqr,t,x_lqr]=lsim(ss_lqr,force,t);
% Plotting state response without LQR
figure
yyaxis left
hold on
plot(t,y_init(:,1), 'r-');
title('Linearized State Response without LQR')
xlabel('time (seconds)')
ylabel('position, distnace x (meters)')
yyaxis right
plot(t,y_init(:,2), 'b-');
plot(t,y_init(:,3), 'r-');
ylabel('angle, \theta (radians)')
legend('x','\theta_1','\theta_2')
hold off

% Plotting state response with LQR
figure
yyaxis left
hold on
plot(t,y_init(:,1), 'r-');
title('Linearized LQR State Response')
xlabel('time (seconds)')
ylabel('position, distnace x (meters)')
yyaxis right
plot(t,y_lqr(:,2), 'b-');
plot(t,y_lqr(:,3), 'r-');
ylabel('angle, \theta (radians)')
legend('x_{lrq}','\theta_1','\theta_2')
hold off
```

7.0.2 Non Linear LQR

```
% Non Linear LQR
clear all
x_initial = [9; 0; 55; 0; 45; 0]
span = 0:1:900;
[tout,xout] = ode45(@odesolver,span,x_initial);
hold on
plot(tout,xout,'LineWidth',2)
grid on
hold off

function dxdt = odesolver(~,x)
M=1000;
ms1=100;
```

```
ms2=100;
l1=20;
l2=10;
g=10;
A=[0 1 0 0 0 0;
    0 0 -(ms1*g)/M 0 -(ms2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+ms1)*g)/(M*l1) 0 -(ms2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(ms1*g)/(M*l2) 0 -(g*(M+ms2))/(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];
Q = C'*C;

R = 0.01;
[Kgain, ~, ~] = lqr(A,B,Q,R);
F=-Kgain*x;
dxdt=zeros(6,1);

dxdt(1)  = x(2);
dxdt(2)=(F-(g/2)*(ms1*sind(2*x(3))+ms2*sind(2*x(5)))-(ms1*l1*(x(4)^2)*sind(x(3)))-(ms2*l
dxdt(3)= x(4);
dxdt(4)= ((dxdt(2)*cosd(x(3))-g*(sind(x(3)))))/l1');
dxdt(5)= x(6);
dxdt(6)= ((dxdt(2)*cosd(x(5))-g*(sind(x(5)))))/l2);
end
```

7.0.3 Obervability

```
% observability check
syms mass1 mass2 M l1 l2 F
g=10;

A = [0 1 0 0 0 0;
     0 0 -g*mass1/M 0 -g*mass2/M 0;
     0 0 0 1 0 0;
     0 0 -g/l1*(1+mass1/M) 0 -g/l1*(mass2/M) 0;
     0 0 0 0 0 1;
     0 0 -g/l2*(mass1/M) 0 -g/l2*(1+mass2/M) 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];

% substituting system values
M=1000; mass1=100;mass2=100; l1=20; l2=10; g=10;
A = double(subs(A)); B = double(subs(B));



C_1 = [1 0 0 0 0 0;
       0 0 0 0 0 0;
       0 0 0 0 0 0];
obs = obsv(A,C_1);
observability_C1 = rank(obs)
```

```
C_2 = [0  0  0  0  0  0;
       0  0  1  0  0  0;
       0  0  0  0  1  0];
obs = obsv(A,C_2);
observability_C2 = rank(obs)

C_3 = [1  0  0  0  0  0;
       0  0  0  0  0  0;
       0  0  0  0  1  0];
obs = obsv(A,C_3);
observability_C3 = rank(obs)

C_4 = [1  0  0  0  0  0;
       0  0  1  0  0  0;
       0  0  0  0  1  0];
obs = obsv(A,C_4);
observability_C4 = rank(obs)
```

7.0.4 Luenberger observer

```
clc; clear; close all;
% Luenberger observer for Linear system
syms mass1 mass2 M l1 l2 F
g=10;

A = [0 1 0 0 0 0;
     0  0 -g*mass1/M 0 -g*mass2/M 0;
     0  0  0  1  0  0;
     0  0 -g/l1*(1+mass1/M)  0 -g/l1*(mass2/M)  0;
     0  0  0  0  0  1;
     0  0 -g/l2*(mass1/M)  0 -g/l2*(1+mass2/M)  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];
Q = C'*C;
R = 0.01;
C_1 = [1  0  0  0  0  0;
       0  0  0  0  0  0;
       0  0  0  0  0  0];

C_3 = [1  0  0  0  0  0;
       0  0  0  0  0  0;
       0  0  0  0  1  0];

C_4 = [1  0  0  0  0  0;
       0  0  1  0  0  0;
       0  0  0  0  1  0];

% substituting system values
M=1000; mass1=100;mass2=100; l1=20; l2=10; g=10;
A = double(subs(A));
B = double(subs(B));

% Pole placement
p = [-3 -4 -5 -6 -7 -8];
X_inital = [0,0,45,0,55,0,0,0,0,0,0,0];
% LQR for optimum control
```

```
K_t = lqr(A,B,Q,R);
% Constructing state estimator for C_1 output vector
Luen_1 = place(A',C_1',p)'
A_obs_1 = [(A-B*K_t) B*K_t;zeros(size(A)) (A-Luen_1*C_1)];
B_obs_1 = [B ;zeros(size(B))];
C_obs_1 = [C_1 zeros(size(C_1))];
est_c1 = ss(A_obs_1,B_obs_1,C_obs_1,0);
t1 = 0:0.1:10;
unit_step = 1*ones(size(t1));
[yc1,t1,xc1]=lsim(est_c1,unit_step,t1,X_inital);

% Constructing state estimator for C_3 output vector
Luen_3 = place(A',C_3',p)'
A_obs_3 = [(A-B*K_t) B*K_t;zeros(size(A)) (A-Luen_3*C_3)];
B_obs_3 = [B ;zeros(size(B))];
C_obs_3 = [C_3 zeros(size(C_3))];
est_c3 = ss(A_obs_3,B_obs_3,C_obs_3,0);
t3 = 0:0.1:10;
unit_step = 1*ones(size(t3));
[yc3,t3,xc3]=lsim(est_c3,unit_step,t3,X_inital);

% Constructing state estimator for C_4 output vector
Luen_4 = place(A',C_4',p)'
A_obs_4 = [(A-B*K_t) B*K_t;zeros(size(A)) (A-Luen_4*C_1)];
B_obs_4 = [B ;zeros(size(B))];
C_obs_4 = [C_4 zeros(size(C_4))];
est_c4 = ss(A_obs_4,B_obs_4,C_obs_4,0);
t4 = 0:0.05:1;
unit_step = 1*ones(size(t4));
[yc4,t4,xc4]=lsim(est_c4,unit_step,t4,X_inital);

figure
initial(est_c1,X_inital)
figure
step(est_c1)


figure
initial(est_c3,X_inital)
figure
step(est_c3)

figure
initial(est_c4,X_inital)
figure
step(est_c4)
grid on
```

7.0.5 LQG for Linear system

```
clc; clear; close all;
% LQG for Linear system
syms mass1 mass2 M l1 l2 F
g=10;

A = [0 1 0 0 0 0;
     0 0 -g*mass1/M 0 -g*mass2/M 0;
     0 0 0 1 0 0;
     0 0 -g/l1*(1+mass1/M) 0 -g/l1*(mass2/M) 0;
```

```
     0  0  0  0  0  1;
     0  0  -g/l2*(mass1/M)  0  -g/l2*(1+mass2/M)  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];
Q = C'*C;
Q(1,1)  = 9000
Q(2,1)  = 1000
R = 0.01;
C_1 = [1  0  0  0  0  0;
       0  0  0  0  0  0;
       0  0  0  0  0  0];

C_3 = [1  0  0  0  0  0;
       0  0  0  0  0  0;
       0  0  0  0  1  0];

C_4 = [1  0  0  0  0  0;
       0  0  1  0  0  0;
       0  0  0  0  1  0];

% substituting system values
M=1000; mass1=100;mass2=100; l1=20; l2=10; g=10;
A = double(subs(A));
B = double(subs(B));
X_inital = [6,0,90,0,90,0,0,0,0,0,0,0];

% Defining noises
p_noise= 0.1*eye(6)
m_noise= 1;

% LQR for optimum control
K_t = lqr(A,B,Q,R);
% Constructing state estimator for C_1 output vector
gain_k = lqr(A', C_1', p_noise, m_noise)'
A_obs_1 = [(A-B*K_t) B*K_t;zeros(size(A)) (A-gain_k*C_1)];
B_obs_1 = [B ;zeros(size(B))];
C_obs_1 = [C_1 zeros(size(C_1))];
est_c1 = ss(A_obs_1,B_obs_1,C_obs_1,0);

figure
initial(est_c1,X_inital)
figure
step(est_c1)
```

7.0.6 LQG for Non-linear systems

```
% Non Linear LQG
clear all
x_initial = [0;0;45;0;55;0;0;0;0;0;0;0];
span = 0:0.1:50;
[tout,xout] = ode45(@odesolver,span,x_initial);
hold on
plot(tout,xout,'LineWidth',2)
grid on
hold off
```

```
function dxdt = odesolver(~,x)
M=1000;
ms1=100;
ms2=100;
l1=20;
l2=10;
g=10;
A=[0 1 0 0 0 0;
    0 0 -(ms1*g)/M 0 -(ms2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+ms1)*g)/(M*l1) 0 -(ms2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(ms1*g)/(M*l2) 0 -(g*(M+ms2))/(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];
Q = C'*C;
Cm = [10 0 0 0 0 0];
R = 0.01;
[Kgain, ~, ~] = lqr(A,B,Q,R);
F=-Kgain*x;
p_noise=0.1*eye(6);
m_noise=1;
K_lqg=lqr(A',Cm',p_noise,m_noise)';
lm =(A-K_lqg*Cm)*y(7:12);

dxdt=zeros(12,1);
dxdt(1) = x(2);
dxdt(2)=(F-(g/2)*(ms1*sind(2*x(3))+ms2*sind(2*x(5)))-(ms1*l1*(x(4)^2)*sind(x(3)))-(ms2*l
dxdt(3)= x(4);
dxdt(4)= ((dxdt(2)*cosd(x(3))-(g)*(sind(x(3))))/l1');
dxdt(5)= x(6);
dxdt(6)= ((dxdt(2)*cosd(x(5))-(g)*(sind(x(5))))/l2);
dxdt(7)= x(2)-x(10);
dxdt(8)= dxdt(2)-lm(2);
dxdt(9)= x(4)-x(11);
dxdt(10)= dxdt(4)-lm(4);
dxdt(11)= x(6)-x(12);
dxdt(12)= dxdt(6)-lm(6);
end
```

Additionally attaching all the code in github at
https://github.com/sumedhreddy90/Controls-LQR-LQG