Control of a Double Pendulum Crane

A project report

submitted by

Shyamsundar Prabhakar Indra

(UID - 119360842)

Sai Surya Sriramoju

(UID - 119224113)

under the supervision of

Dr.Waseem Malik

in partial fulfillment of
the requirements for ENPM667

to the

University of Maryland, College Park

17th December 2022

ACKNOWLEDGEMENT

As a matter of first importance, we offer our genuine thanks to our professor Dr. Waseem Malik, who did a great job at explaining concepts clearly inside the classroom and offering life lessons outside the classroom. We also extend our appreciation to the TAs for their support and help during the project work.

Shyamsundar Prabhakar Indra

(119360842)

Sai Surya Sriramoju

(119224113)

Contents

Chapter 1

Problem Statement

Consider a crane that moves along an one-dimensional track. It behaves as a frictionless cart with mass M actuated by an external force F that constitutes the input of the system. There are two loads suspended from cables attached to the crane. The loads have mass m_1 and m_2, and the lengths of the cables are l_1 and l_2, respectively. The Figure 1 depicts the crane and associated variables used throughout this project.

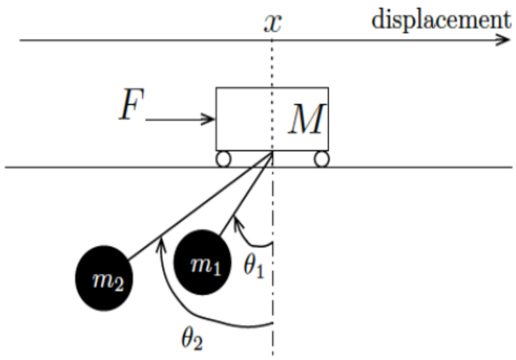

Figure 1.1: Given system diagram

Given Information:

- M = Mass of the cart
- F = Force actuated on the cart
- m_1 = mass of load 1
- m_1 = mass of load 2
- l_1 = length of cable 1 to which mass 1 is suspended

- l_2 = length of cable 2 to which mass 2 is suspended

Assumptions:

- The crane is constrained to move along the x axis and hence has 1 DOF.
- The loads are considered to be point masses with their Center of Mass located at the end of each cable.
- The loads are free to move in the x-y plane and hence have 2 DOF.
- The two cables connecting the loads to the crane are massless and won't contribute to the potential energy or kinetic energy of the system.
- Friction is non-existent in the system.
- There is no rotation about the COM for any of the masses in the system.

NOTE: The codes associated with each part of the solution in pdf format have been linked at the start of every section for easy reference. However the codes as a whole in normal script format have been attached in the appendix for a sense of completeness. When there are plots/important results involved we have included them with the report itself so as to avoid extra pain for the reader.

Chapter 2

First Component - System Formulation and LQR

2.1 A. Equations of Motion and Corresponding Non-Linear State Space Representation.

Codes and outputs for A,B and C can be found here: Link

Problem and Frame of Reference definition:

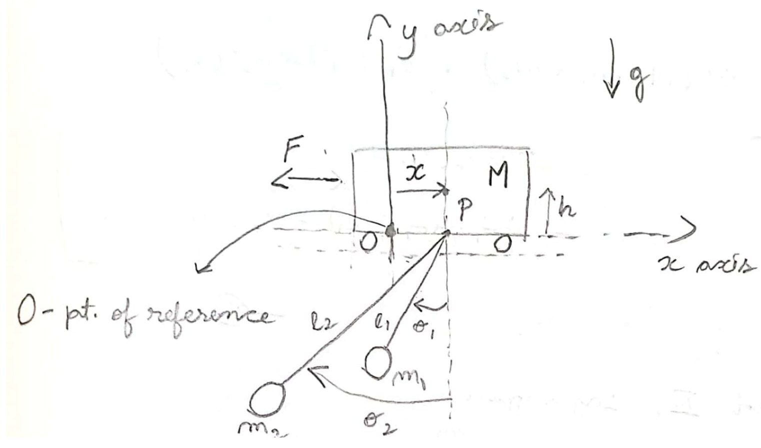

Figure 2.1: Problem definition and assignment of frames

- P is the point of connection of the cables to the cart.
- Point O is the inertial frame of reference and it is located at the same vertical height as point P.

- Horizontal axis is x-axis and Vertical axis is y-axis
- Gravity g acts in the negative y direction
- x is the displacement of the car in positive x direction at any given time instant.
- F is the force acting on the car when 'x' is positive in the negative x axis direction trying to bring the crane back to O.
- All other symbols mean the same as they've been defined in the problem statement.

The above problem can be solved using Newton-Euler or Euler-Lagrange method. For this project, the Euler-Lagrange method has been used for modelling the system. The Euler-Lagrange method requires Kinetic and Potential energies for computing the Lagrangian. The Euler-Lagrange equation is given by:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{q}}\right] - \frac{\partial \mathcal{L}}{\partial q} = F \tag{2.1}$$

where L is the Lagrangian given by the difference of Kinetic Energy (K.E) and Potential Energy (P.E), q is the parameter of consideration and F is the external force associated with a specific parameter. For convenience, it is taken that $sin(\theta_1) = S1$, $cos(\theta_1) = C1$, $sin(\theta_2) = S2$, and $cos(\theta_2) = C2$.

Kinetic Energy computation:

Considering the point of connection of the loads to the cart as the point of reference 'O', the x and y position of each of the masses m_1 and m_2, and that of the crane of mass M are defined as follows:

$$\begin{aligned} x_{m1} &= x - l_1 S_1 \\ y_{m1} &= -l_1 C_1 \end{aligned} \tag{2.2}$$

$$\begin{aligned} x_{m2} &= x - l_2 S_2 \\ y_{m2} &= -l_2 C_2 \end{aligned} \tag{2.3}$$

$$
\begin{aligned}
x_M &= x \\
y_M &= 0
\end{aligned} \tag{2.4}
$$

To compute the velocities associated with each load and the crane, the above equations need to be differentiated with respect to time and differentiated equations are gives as follows:

$$
\begin{aligned}
\dot{x_{m1}} &= \dot{x} - l_1 C_1 \dot{\theta}_1 \\
\dot{y_{m1}} &= l_1 S_1 \dot{\theta}_1
\end{aligned} \tag{2.5}
$$

$$
\begin{aligned}
\dot{x_{m2}} &= \dot{x} - l_2 C_2 \dot{\theta}_2 \\
\dot{y_{m2}} &= l_2 S_2 \dot{\theta}_2
\end{aligned} \tag{2.6}
$$

$$
\begin{aligned}
\dot{x_M} &= \dot{x} \\
\dot{y_M} &= 0
\end{aligned} \tag{2.7}
$$

The final velocities v_1 and v_2 of the loads can be computed as a square of the x and y velocities using Pythagoras theorem as follows:

$$
\begin{aligned}
v_{m_1}^2 &= \dot{x_{m1}}^2 + \dot{y_{m1}}^2 \\
v_{m_1}^2 &= \left(\dot{x} - l_1 C_1 \dot{\theta}_1\right)^2 + \left(l_1 S_1 \dot{\theta}_1\right)^2 \\
v_{m_1}^2 &= \dot{x}^2 + l_1 \dot{\theta}_1^2 - 2 l_1 C_1 \dot{x} \dot{\theta}_1
\end{aligned} \tag{2.8}
$$

$$
\begin{aligned}
v_{m_2}^2 &= \dot{x_{m2}}^2 + \dot{y_{m2}}^2 \\
v_{m_2}^2 &= \left(\dot{x} - l_2 C_2 \dot{\theta}_2\right)^2 + \left(l_2 S_2 \dot{\theta}_2\right)^2 \\
v_{m_2}^2 &= \dot{x}^2 + l_1 \dot{\theta}_2^2 - 2 l_2 C_2 \dot{x} \dot{\theta}_2
\end{aligned} \tag{2.9}
$$

$$
\begin{aligned}
v_M^2 &= \dot{x_M}^2 + \dot{y_M}^2 \\
v_M^2 &= \dot{x_M}^2
\end{aligned} \tag{2.10}
$$

Using the above equations which have given us the velocities of each of the masses associated with the system, since there is no rotation about the COM for any of the masses, the total Kinetic Energy of the system is given by:

$$K.E. = \frac{1}{2}(Mv_M^2) + \frac{1}{2}\left(m_1 v_1^2\right) + \frac{1}{2}\left(m_2 v_2^2\right) \tag{2.11}$$

Substituting the previously computed velocity values, we get:

$$K.E. = \frac{1}{2}(M)\dot{x}^2 + \left(\frac{1}{2}\left(m_1\left(\dot{x} - l_1 C_1 \dot{\theta}_1\right)^2 + \left(l_1 S_1 \dot{\theta}_1\right)^2\right)\right) + \frac{1}{2}\left(m_2\left(\dot{x} - l_2 C_2 \dot{\theta}_2\right)^2 + \left(l_2 S_2 \dot{\theta}_2\right)^2\right) \tag{2.12}$$

Upon simplifying, Kinetic Energy reduces to:

$$K.E. = \frac{1}{2}\dot{x}^2\left(m_1 + m_2 + M\right) + \frac{1}{2}\left(m_1 l_1 \dot{\theta}_1^2\right) + \frac{1}{2}\left(m_2 l_2 \dot{\theta}_2^2\right) - m_1 l_1 \cos_1 \dot{x}\dot{\theta}_1 - m_2 l_2 C_2 \dot{x}\dot{\theta}_2 \tag{2.13}$$

Potential Energy computation

Since O is the point of reference, the crane has a positive potential energy and the loads have negative potential energies. The representation of potential energy of the system is as follows:

$$P.E. = -m_1 g l_1 C_1 - m_2 g l_2 C_2 + mgh \tag{2.14}$$

Lagrangian formulation

The Lagrangian of the system as stated in the above paragraphs is given by KE - PE. Therefore Langrangian L of the system can be represented as:

$$\begin{gathered}\boldsymbol{\mathcal{L}} = K.\mathrm{E.} \; - P.\mathrm{E.} \\ \boldsymbol{\mathcal{L}} = \frac{1}{2}\dot{x}^2\left(m_1 + m_2 + M\right) + \frac{1}{2}\left(m_1 l_1 \dot{\theta}_1^2\right) + \frac{1}{2}\left(m_2 l_2 \dot{\theta}_2^2\right) - m_1 l_1 \cos_1 \dot{x}\dot{\theta}_1 - m_2 l_2 C_2 \dot{x}\dot{\theta}_2 \\ + m_1 g l_1 C_1 + m_2 g l_2 C_2 - mgh\end{gathered} \tag{2.15}$$

Finding the equations of motion

To find the equations of motion, we use the previously obtained Lagrangian in the Euler-Lagranger equation for each of the states x, θ_1 and θ_2.

First we start with computing Euler Lagrange equation for the variable x. The RHS of the equation equals -F since there is a force F acting on the crane in the negative x direction and is associated with the state x. The Euler-Lagrange equation for the state x can be found below:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{x}}\right] - \frac{\partial \mathcal{L}}{\partial x} = -\mathrm{F} \tag{2.16}$$

$$\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \dot{x}} = \dot{x}\left(m_1 + m_2 + M\right) - m_1 l_1 C_1\left(\dot{\theta}_1\right) - m_2 l_2 C_2\left(\dot{\theta}_2\right)$$

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

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{x}}\right] - \frac{\partial \mathcal{L}}{\partial x} = \ddot{x}\left(m_1 + m_2 + M\right) - m_1 l_1\left(\ddot{\theta}_1 C_1 - S_1\dot{\theta}_1^2\right) - m_2 l_2\left(\ddot{\theta}_2 C_2 - S_2\dot{\theta}_2^2\right) = -\mathrm{F} \tag{2.17}$$

There is no torque associated with θ_1 or θ_2, hence the RHS of the Euler-Lagrange equations associated with these two states is zero. The Euler-Lagrange equation of the these two states can be found below:

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right] - \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \theta_1} = 0 \tag{2.18}$$

$$\frac{d}{dt}\left[\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \dot{\theta}_2}\right] - \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \theta_2} = 0 \tag{2.19}$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} &= m_1 l_1^2 \dot{\theta}_1 - m_1 l_1 C_1 \dot{x} \\
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right) &= m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \left(C_1 \ddot{x} - S_1 \dot{x} \dot{\theta}_1\right) \\
\frac{\partial \mathcal{L}}{\partial \theta_1} &= S_1 m_1 l_1 m_1 \dot{x} \dot{\theta}_1 - m_1 g l_1 S_1 \\
\frac{\partial \boldsymbol{L}}{\partial \dot{\theta}_2} &= m_2 l_2^2 \dot{\theta}_2 - m_2 l_2 C_2 \dot{x} \\
\frac{\partial \mathcal{L}}{\partial \theta_2} &= S_2 m_2 l_2 m_2 \dot{x} \dot{\theta}_2 - m_2 g l_2 S_2 \\
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2}\right) &= m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \left(C_2 \ddot{x} - S_2 \dot{x} \dot{\theta}_2\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \dot{\theta}_1}\right) - \frac{\partial \boldsymbol{L}}{\partial \dot{\theta}_1} &= m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \left(C_1 \ddot{x} - S_1 \dot{x} \dot{\theta}_1\right) - S_1 m_1 l_1 m_1 \dot{x} \dot{\theta}_1 + m_1 g l_1 S_1 = 0 \\
\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1}\right) - \frac{\partial \boldsymbol{L}}{\partial \dot{\theta}_1} &= m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 C_1 \ddot{x} + m_1 g l_1 S_1 = 0
\end{aligned} \tag{2.20}
$$

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \boldsymbol{\mathcal{L}}}{\partial \dot{\theta}_2}\right) - \frac{\partial \boldsymbol{\mathcal{L}}}{\partial \dot{\theta}_2} &= m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \left(C_2 \ddot{x} - S_2 \dot{x} \dot{\theta}_2\right) - S_2 m_2 l_2 m_2 \dot{x} \dot{\theta}_2 + m_2 g l_2 S_2 = 0 \\
\frac{d}{dt}\left(\frac{\partial \boldsymbol{L}}{\partial \dot{\theta}_2}\right) - \frac{\partial \boldsymbol{L}}{\partial \dot{\theta}_2} &= m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 C_2 \ddot{x} + m_2 g l_2 S_2 = 0
\end{aligned} \tag{2.21}
$$

From the above equations, the equations of motion can be written as follows:

$$
\ddot{x} = \left(m_1 l_1 \left(\theta_1 \ddot{C}_1 - S_1 \dot{\theta}_1^2\right) + m_2 l_2 \left(\theta_2 C_2 - S_2 \dot{\theta}_2^2\right) + F\right) / \left(m_1 + m_2 + M\right) \tag{2.22}
$$

$$
\ddot{\theta}_1 = \frac{C_1 \ddot{x} - g S_1}{l_1} \tag{2.23}
$$

$$
\ddot{\theta}_2 = \frac{C_2 \ddot{x} - g S_2}{l_2} \tag{2.24}
$$

Non-Linear State Space Representation

From the equations 2.22, 2.23 and 2.24 obtained above, substituting 2.23 and 2.24, in 2.22 we obtain:

$$\ddot{x} = \frac{F - m_1 \left(gS_1C_1 + l_1 s_1 \dot{\theta}_1^2 \right) - m_2 \left(gS_2C_2 + l_2S_2\dot{\theta}_2^2 \right)}{\left(M + m_1 \left(S_1^2 \right) + m_2 \left(S_2^2 \right) \right)} \tag{2.25}$$

Substituting 2.25 back in 2.23 and 2.24, we obtain $\ddot{\theta_1}$ and $\ddot{\theta_2}$ as follows:

$$\ddot{\theta_1} = \frac{C_1}{l_1} \left[\frac{F - m_1 \left(gS_1C_1 + l_1S_1\dot{\theta_1}^2 \right) - m_2 \left(gS_2C_2 + l_2S_2\dot{\theta_2}^2 \right)}{\left(M + m_1 \left(S_1^2 \right) + m_2 \left(S_2^2 \right) \right)} \right] - g\frac{S_1}{l_1} \tag{2.26}$$

$$\ddot{\theta_2} = \frac{C_2}{l_2} \left[\frac{F - m_1 \left(gS_1C_1 + l_1S_1\dot{\theta_1}^2 \right) - m_2 \left(gS_2C_2 + l_2S_2\dot{\theta_2}^2 \right)}{\left(M + m_1 \left(S_1^2 \right) + m_2 \left(S_2^2 \right) \right)} \right] - g\frac{S_2}{l_2} \tag{2.27}$$

Representing 2.25, 2.26 and 2.27 in the matrix form, we get the non-linear state space representation of the system as follows:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{F - m_1 \left(gS_1C_1 + l_1s_1\dot{\theta_1}^2 \right) - m_2 \left(gs_2C_2 + l_2s_2\dot{\theta_2}^2 \right)}{\left(M + m_1 \left(s_1^2 \right) + m_2 \left(s_2^2 \right) \right)} \\ \dot{\theta}_1 \\ \frac{C_1}{l_1} \left[\frac{F - m_1 \left(gS_1C_1 + l_1s_1\dot{\theta}_1^2 \right) - m_2 \left(gS_2C_2 + l_2S_2\dot{\theta}_2^2 \right)}{\left(M + m_1 \left(S_1^2 \right) + m_2 \left(S_2^2 \right) \right)} \right] - g\frac{S_1}{l_1} \\ \dot{\theta}_2 \\ \frac{C_2}{l_2} \left[\frac{F - m_1 \left(gS_1C_1 + l_1S_1\dot{\theta_1}^2 \right) - m_2 \left(gS_2C_2 + l_2S_2\dot{\theta}_2^2 \right)}{\left(M + m_1 \left(S_1^2 \right) + m_2 \left(S_2^2 \right) \right)} \right] - g\frac{S_2}{l_2} \end{bmatrix} \tag{2.28}$$

2.2 B. Linearization of the System

The system of non-linear equations derived above can be linearized with one of the two techniques stated below:

- Linearization by small angle approximation
- Linearization using Jacobian

The second method has been used to linearize the system for the case of this project and only that method will be discussed below.

The system can be linearized by computing the respective Jacobian for A matrix and B matrix as shown below, where F is the Force applied to the crane:

$$\mathrm{A} = \begin{bmatrix} \frac{\delta f_1}{\delta X_1} & \frac{\delta f_1}{\delta X_2} & \frac{\delta f_1}{\delta X_3} & \frac{\delta f_1}{\delta X_4} & \frac{\delta f_1}{\delta X_5} & \frac{\delta f_1}{\delta X_6} \\ \frac{\delta f_2}{\delta X_1} & \frac{\delta f_2}{\delta X_2} & \frac{\delta f_2}{\delta X_3} & \frac{\delta f_2}{\delta X_4} & \frac{\delta f_2}{\delta X_5} & \frac{\delta f_2}{\delta X_6} \\ \frac{\delta f_3}{\delta X_1} & \frac{\delta f_3}{\delta X_2} & \frac{\delta f_3}{\delta X_3} & \frac{\delta f_3}{\delta X_4} & \frac{\delta f_3}{\delta X_5} & \frac{\delta f_3}{\delta X_6} \\ \frac{\delta f_4}{\delta X_1} & \frac{\delta f_4}{\delta X_2} & \frac{\delta f_4}{\delta X_3} & \frac{\delta f_4}{\delta X_4} & \frac{\delta f_4}{\delta X_5} & \frac{\delta f_4}{\delta X_6} \\ \frac{\delta f_5}{\delta X_1} & \frac{\delta f_5}{\delta X_2} & \frac{\delta f_5}{\delta X_3} & \frac{\delta f_5}{\delta X_4} & \frac{\delta f_5}{\delta X_5} & \frac{\delta f_5}{\delta X_6} \\ \frac{\delta f_6}{\delta X_1} & \frac{\delta f_6}{\delta X_2} & \frac{\delta f_6}{\delta X_3} & \frac{\delta f_6}{\delta X_4} & \frac{\delta f_6}{\delta X_5} & \frac{\delta f_6}{\delta X_6} \end{bmatrix} \tag{2.29}$$

$$\mathrm{B} = \begin{bmatrix} \frac{\delta f_1}{\delta F} \\ \frac{\delta f_2}{\delta F} \\ \frac{\delta f_3}{\delta F} \\ \frac{\delta f_4}{\delta F} \\ \frac{\delta f_5}{\delta F} \\ \frac{\delta f_6}{\delta F} \end{bmatrix} \tag{2.30}$$

$$[f_1, f_2, f_3, f_4, f_5, f_6] = [\dot{x}, \ddot{x}, \dot{\theta}_1, \ddot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_2]$$

$$[X_1, X_2, X_3, X_4, X_5, X_6] = [x, \dot{x}, \theta_1, \dot{\theta}1, \theta_2, \dot{\theta}2]$$

Upon using the above method to linearize the system, we get:

$$
\mathrm{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-gm_1}{M} & 0 & \frac{-gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-g(M+m_1)}{Ml_1} & 0 & \frac{-gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-gm_1}{Ml_2} & 0 & \frac{-g(M+m_2)}{Ml_2} & 0 \end{bmatrix} \tag{2.31}
$$

$$
\mathrm{B} = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} \tag{2.32}
$$

Therefore the final linear state space representation of the system is as follows:

$$
\begin{bmatrix} \dot{\boldsymbol{x}} \\ \ddot{x} \\ \dot{\boldsymbol{\theta}}_1 \\ \ddot{\boldsymbol{\theta}}_1 \\ \dot{\boldsymbol{\theta}}_2 \\ \ddot{\boldsymbol{\theta}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-gm_1}{M} & 0 & \frac{-gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-g(M+m_1)}{Ml_1} & 0 & \frac{-gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-gm_1}{Ml_2} & 0 & \frac{-g(M+m_2)}{Ml_2} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \\ \boldsymbol{\theta_1} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{1}} \\ \boldsymbol{\theta_2} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{2}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{bmatrix} F
$$

2.3 C. Conditions for Controllability

A system is said to be controllable if the grammian is invertible. To check if the grammian is invertible, we can check the rank of the Controllability Matrix. If the rank of the Controllability Matrix is same as the number of state space variables, then grammian is invertible and therefore the system is invertible.
Condition for the system to be controllable:

$$\text{rank}(C) = \text{rank}\left[B\ AB\ A^2B\ A^3B\ A^4B\ A^5B\right] = n$$

Where A and B are the obtained from linear state space representation of the system, n is the number of state space variables, which is 6 in this case.
Upon computing the determinant of the paratmetrized controllability matrix C, we get the determinant to be:

$$|C| = -\frac{g^6\, {l_1}^2 - 2\, g^6\, l_1\, l_2 + g^6\, {l_2}^2}{M^6\, {l_1}^6\, {l_2}^6}$$

For the system to be controllable, C matrix should be full rank (rank = 6). Therefore the above determinant should not be equal to zero. Therefore it can be inferred from the above equation that the system is controllable if the following conditions are satisfied simultaneously:

$$M, m_1, m_2 > 0\ and\ l_1 \neq l_2$$

2.4 D. Verifying Controllability & Obtaining LQR Controller

The codes and outputs for part D can be found here: Link

Verifying Controllability

Plugging in the given values of all the variables, $M = 1000Kg, m_1 = m_2 = 100Kg, l_1 = 20m$ and $l_2 = 10m$ in A and B, we get:

$$\text{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.981 & 0 & -0.981 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -0.53955 & 0 & -0.04905 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -0.0981 & 0 & -1.0791 & 0 \end{pmatrix}$$

$$\text{B} = \begin{pmatrix} 0 \\ 0.001 \\ 0 \\ 0.00005 \\ 0 \\ 0.0001 \end{pmatrix}$$

The system with above given parameters is controllable since $M, m_1, m_2 > 0 \; and \; l_1 \neq l_2$. However to cross check, the resultant controllability matrix and its determinant after subsitution are as shown below:

$$\text{C} = \begin{pmatrix} 0 & 0.001 & 0 & -0.00014715 & 0 & 0.00014195 \\ 0.001 & 0 & -0.00014715 & 0 & 0.00014195 & 0 \\ 0 & 0.00005 & 0 & -0.000031882 & 0 & 0.000022736 \\ 0.00005 & 0 & -0.000031882 & 0 & 0.000022736 & 0 \\ 0 & 0.0001 & 0 & -0.00011281 & 0 & 0.00012487 \\ 0.0001 & 0 & -0.00011281 & 0 & 0.00012487 & 0 \end{pmatrix}$$

$$|C| = -1.3926e - 24 \neq 0$$

Designing LQR Controller

In order to design an LQR Controller for a non-linear system, the following steps are followed:

- We first linearize the given non-linear system system.
- We then need to optimize the below cost function J.

$$J = \int_0^\infty \left(\vec{X}(t)^T Q \vec{X}(t) + \left(\vec{U}(t)^T R \vec{U}(t)\right) dt\right.$$

- This is done by obtaining P by solving the Ricatti equation which takes A,B,Q and R as the input parameters (Q and R are the design parameters).

$$Ricatti - Equation : A^T P + PA - PBR^{-1}B^T P + Q = 0$$

- This P is in turn used to minimize the cost function by finding the optimal gain K using:

$$K = R^{-1}B^T P$$

- Finally we use this optimal gain to modify the state-space equation as:

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

$$where\ U = -KX$$

$$\therefore \dot{X} = (A - BK)X$$

As mentioned above Q and R are design parameters which need to be determined by plotting the response graph for different choices of Q and R. Upon doing this process we've chosen the following as our Q and R:

$$
Q = \begin{pmatrix}
\frac{1}{12} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{5}{6} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{25}{8} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{2500}{3} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{25}{8} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{2500}{3}
\end{pmatrix}
$$

$$
R = 0.00001
$$

Using the above values of Q and R, the computed K is as follows:

$$
K = \begin{pmatrix} 91.287093 & 652.9372 & 68.833445 & 9147.9749 & 6588.7588 & 5120.7253 \end{pmatrix}
$$

Upon using the above value of K, the modified state matrix $A - BK$ is shown below:

$$
(A - BK) = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
-0.091287 & -0.65294 & -1.0498 & -9.148 & -7.5698 & -5.1207 \\
0 & 0 & 0 & 1 & 0 & 0 \\
-0.0045644 & -0.032647 & -0.54299 & -0.4574 & -0.37849 & -0.25604 \\
0 & 0 & 0 & 0 & 0 & 1 \\
-0.0091287 & -0.065294 & -0.10498 & -0.9148 & -1.738 & -0.51207
\end{pmatrix}
$$

Poles of the system before the application of the LQR controller:

$$
\begin{pmatrix}
0.0000 + 0.0000i \\
0.0000 + 0.0000i \\
0.0000 + 0.72854i \\
0.0000 - 0.72854i \\
0.0000 + 1.043i \\
0.0000 - 1.043i
\end{pmatrix}
$$

Poles of the system after application of the LQR controller:

$$\begin{pmatrix} -0.45824 + 0.86851\mathrm{i} \\ -0.45824 - 0.86851\mathrm{i} \\ -0.23141 + 0.16309\mathrm{i} \\ -0.23141 - 0.16309\mathrm{i} \\ -0.12155 + 0.74402\mathrm{i} \\ -0.12155 - 0.74402\mathrm{i} \end{pmatrix}$$

According to Lyapunov's indirect method to certify stability, the real parts of all the poles of the system after application of the LQR controller are negative and the poles lie on the left half of the plane, hence the closed-loop system is stable.

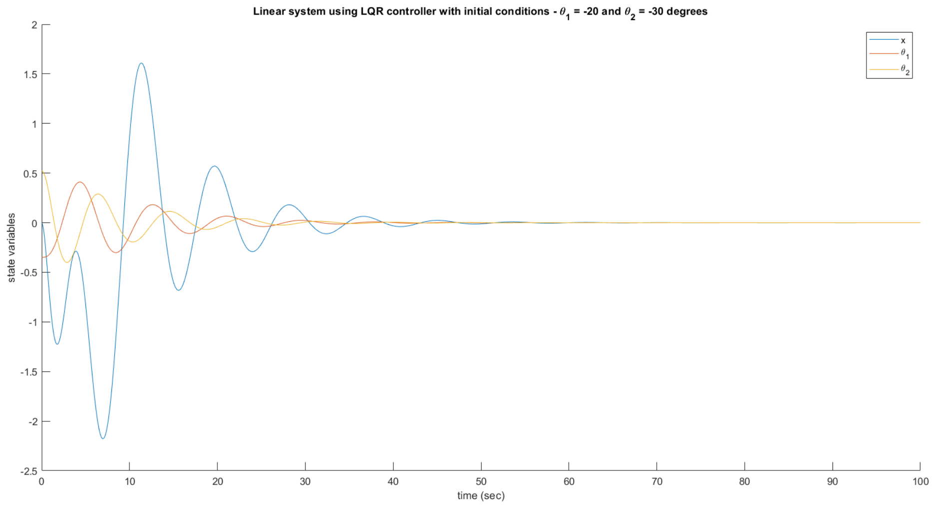

Figure 2.2: Linear system using LQR controller with initial conditions - θ_1 = -20 and θ_2 = -30 degrees

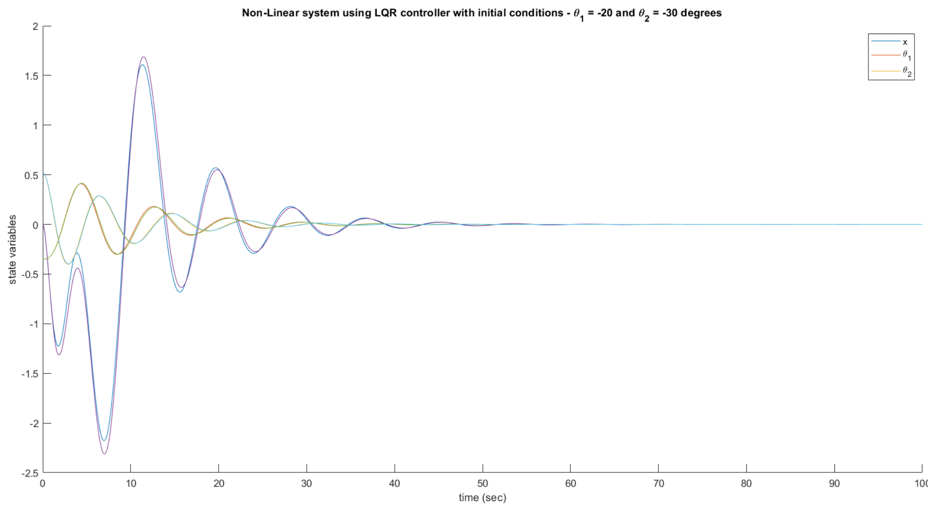

Figure 2.3: Non-Linear system using LQR controller with initial conditions - $\theta_1 =$ -20 and $\theta_2 =$ -30 degrees

Chapter 3

Second Component - Observability, Luenberger Observer and LQG

3.1 E. Observability of the Linearized System

Codes and outputs for E can be found here: Link

We need to check the observability of the linear system for 4 choice of output vectors as follows:

- Output vector = (x), corresponding output vector C1:

$$\text{C1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Output vector = (θ_1, θ_2), corresponding output vector C2:

$$\text{C2} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- Output vector = (x, θ_2), corresponding output vector C3:

$$\text{C3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- Output vector $= (x, \theta_1, \theta_2)$, corresponding output vector C4:

$$\text{C4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Given the ouptut matrix C, and the state matrix A, the given set of output vectors are observable if the observability matrix given below has a full rank, which is 6 here.

$$Obs = [C' \; A'C' \; A'^2C' \; A'^3C' \; A'^4C' \; A'^5C]$$

Upon computing the rank of each of these matrices, the second output vector (θ_1, θ_2) gives an observability matrix which only has a rank $4 \neq 6$. Therefore the linear system is not observable for the output vector (θ_1, θ_2). For question (F), the Luenberger observer won't be computed for this output vector. The ranks of the observability matrix for each output vector has been shown below:

```
fprintf('Rank of observability matrix for Output vector 1 (x) = %d' , Obs1_rank)
Rank of observability matrix for Output vector 1 (x) = 6
fprintf('Rank of observability matrix for Output vector 2 (theta1, theta2) = %d' , Obs2_rank)
Rank of observability matrix for Output vector 2 (theta1, theta2) = 4
fprintf('Rank of observability matrix for Output vector 3 (x, theta2) = %d' , Obs3_rank)
Rank of observability matrix for Output vector 3 (x, theta2) = 6
fprintf('Rank of observability matrix for Output vector 4 (x, theta1, theta2) = %d' , Obs4_rank)
Rank of observability matrix for Output vector 4 (x, theta1, theta2) = 6
```

Figure 3.1: Rank of observability matrix for the four given output vectors

3.2 F. Luenberger Observer

Codes and outputs for F can be found here: Link

The Luenberger Observer is written in state-space representation as:

$$\dot{\hat{X}}(t) = A\hat{X}(t) + B_k U_k(t) + L(Y(t) - C\hat{X}(t));$$

Here, $\hat{X}(t)$ is state estimator, L is observer gain matrix, $Y(t) - C\hat{X}(t)$ is the correction term and $\hat{X}(0) = 0$ (Since there is no discrepancy between observed and estimated state at t=0). The estimation error $X_e(t) = X(t) - \hat{X}(t)$ has the following state space representation:

$$\dot{X}_e(t) = \dot{X}(t) - \dot{\hat{X}}(t)$$
$$\dot{X}_e(t) = AX_e(t) - L(Y(t) - C\hat{X}(t)) + B_d U_d(t)$$

Here, we assume D $= 0, Y(t) = Cx(t)$. Therefore, the equation can be written as:

$$\dot{X}_e(t) = (A - LC)X_e(t) + B_d U_d(t);$$

The Luenberger observers have been forumlated for the 3 observable output vectors for both the linear and non-linear system. The plots associated with the observed output in case of initial state and step response for each of these observers are attached below:

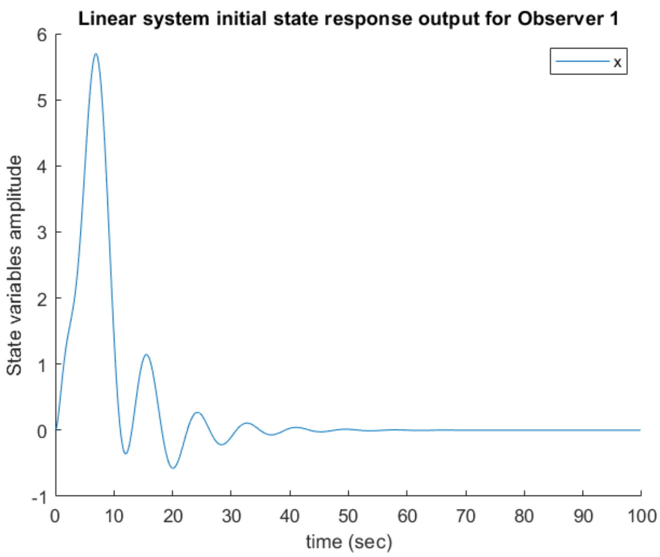

Figure 3.2: Linear initial state response output for Observer 1

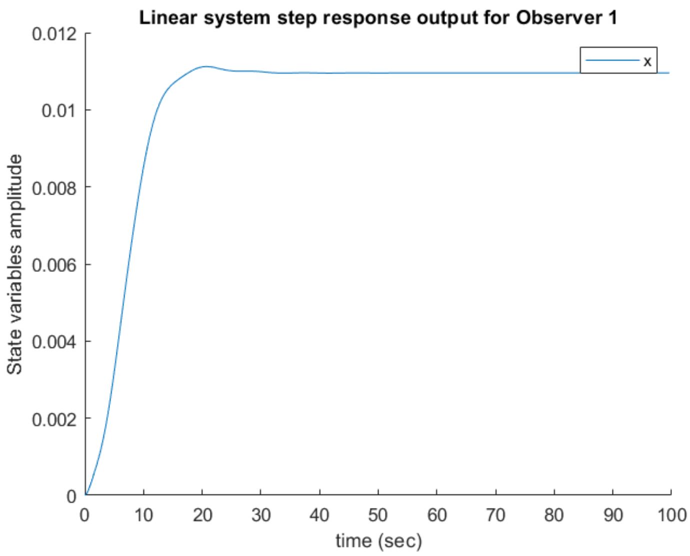

Figure 3.3: Linear system step response output for Observer 1

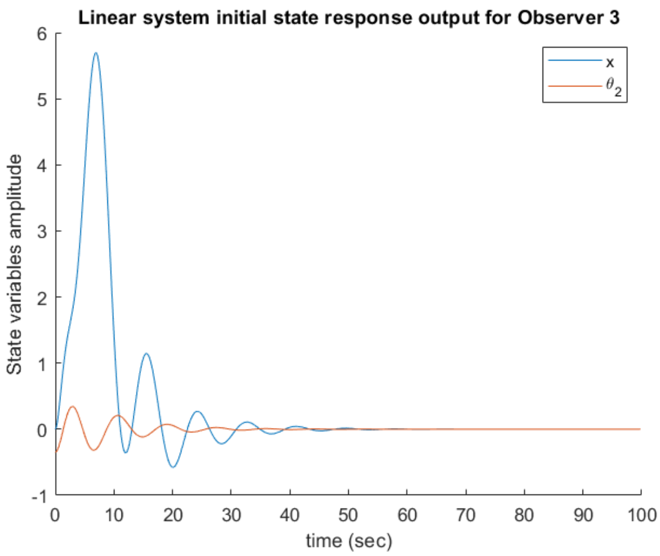

Figure 3.4: Linear system initial state response output for Observer 3

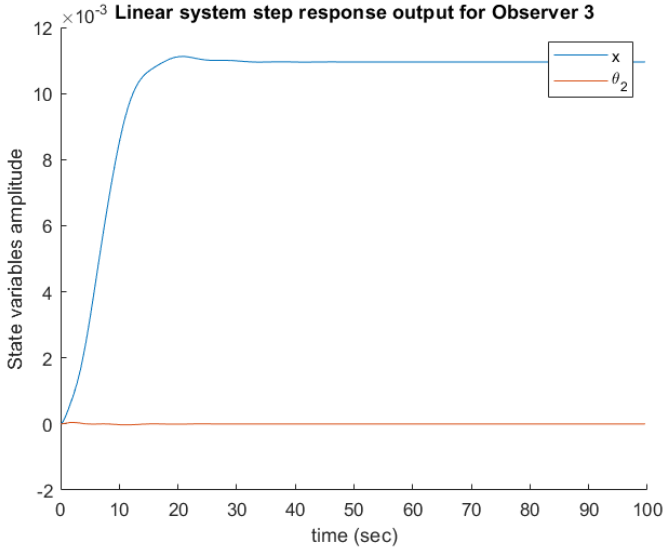

Figure 3.5: Linear system step response output for Observer 3

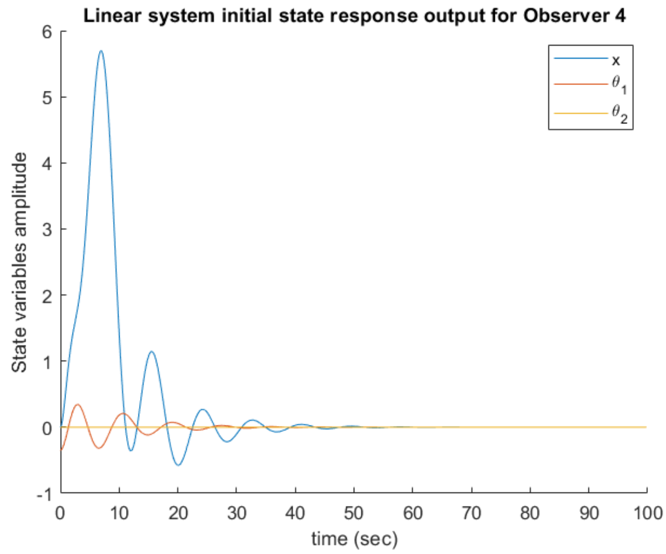

Figure 3.6: Linear system initial state response output for Observer 4

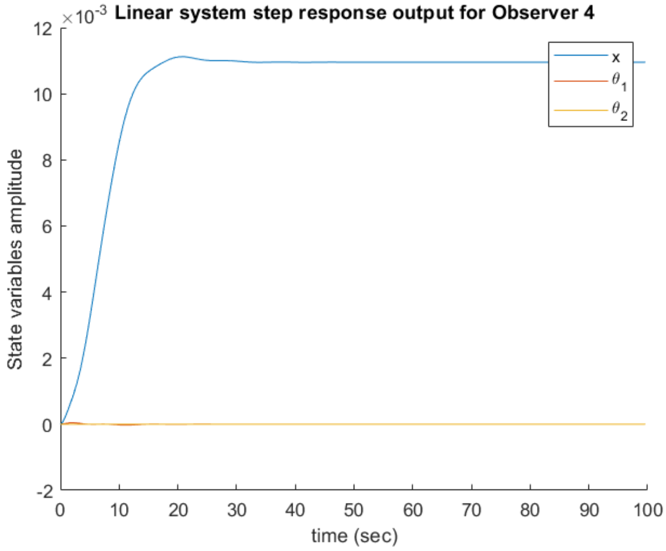

Figure 3.7: Linear system step response output for Observer 4

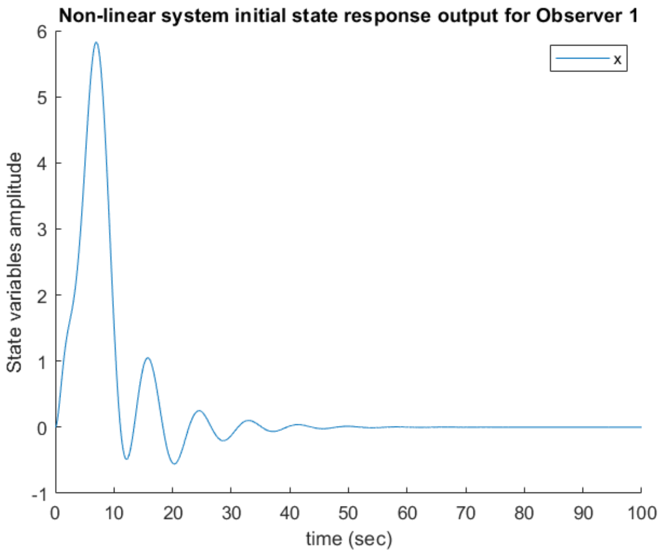

Figure 3.8: Non-linear initial state response output for Observer 1

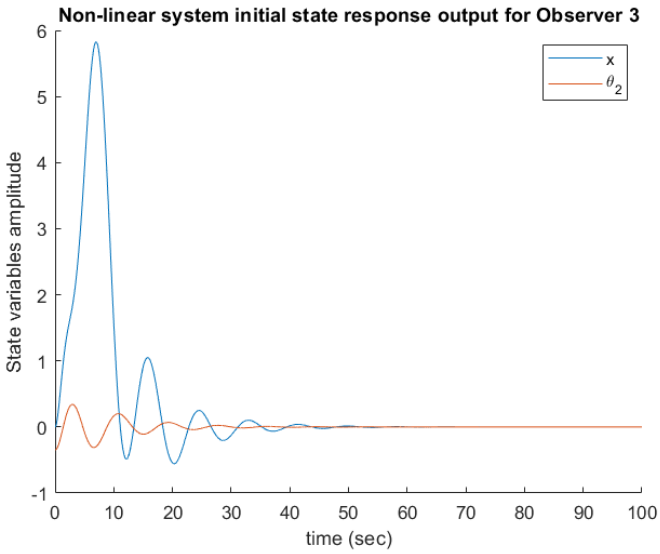

Figure 3.9: Non-linear initial state response output for Observer 3

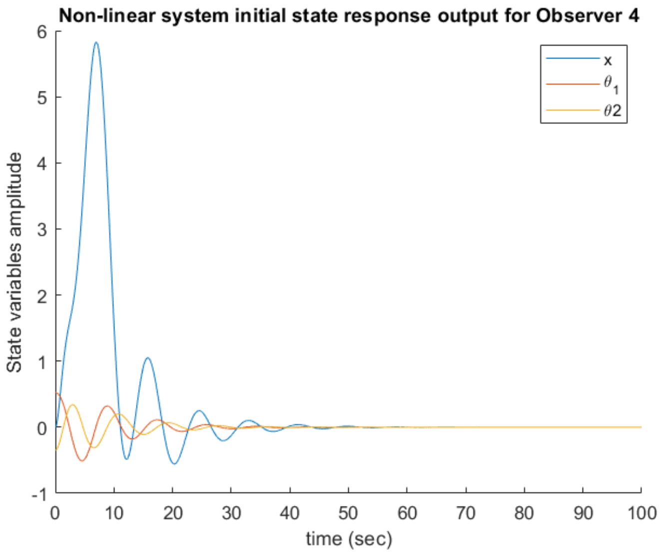

Figure 3.10: Non-linear initial state response output for Observer 4

Chapter 4

G. LQG Controller

Codes and outputs for G can be found here: Link

LQG controller works as the combination of an LQR controller with a Kalman Filter. We need an observer for observing the output and using it as feedback, for which case the output vector x(t) has been chosen, since it's the smallest observable output vector. We tried modelling the non-linear system with a dynamic Gaussian noise for process error and measurement error. However for some reason the solver kept crashing if I dynamically provided noise. Hence static process and measurement noises (which get dynamically declared before solving) were added for the system controlled by the LQG controller.

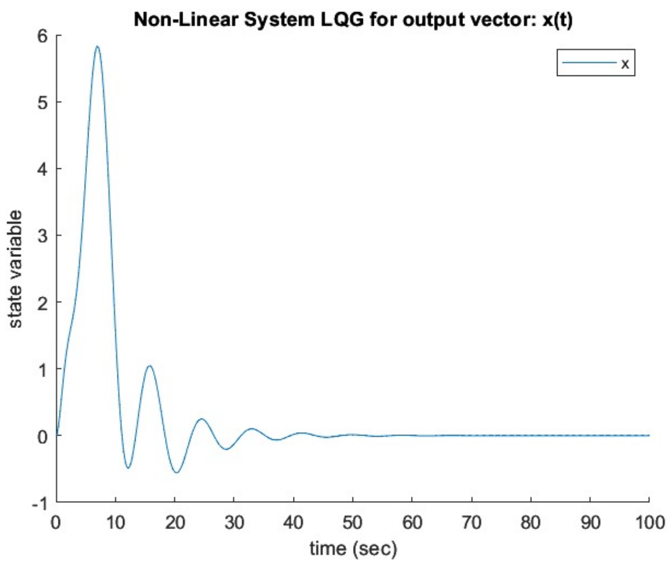

Figure 4.1: Non-Linear System LQG for output vector: x(t)

In order to asymptotically track a constant reference on x, we change the cost function of the LQR part of the LQG controller, as follows:

$$\int_0^{\infty} (\vec{X}(t) - \vec{X}_d)^T \mathbf{Q} (\vec{X}(t) - \vec{X}_d) + (\vec{U}_K(t) - \vec{U}_{\infty})^T \mathbf{R} (\vec{U}_K(t) - \vec{U}_{\infty}) dt$$

Our design of the LQG controller will be able to reject disturbances and bring back the output to the desired reference (or 0) given that the disturbances can be modelled as Gaussian white noise processes.

Chapter 5

Appendix

5.1 Part A, B, and C

```
syms x(t) t1(t) t2(t) x_tt(t) t1_tt(t) t2_tt(t)
syms M m1 m2 l1 l2 g h F

x_t = diff(x,t);

t1_t = diff(t1,t);

t2_t = diff(t2,t);

% Using the Lagrangian computed in the report

L = (x_t^2*(M+m1+m2)/2)
+ (m1*(t1_t^2)*(l1^2))/2
+ (m2*(t2_t^2)*(l2^2)/2) ...
+ (m1*l1*cos(t1)*(g-x_t*t1_t))
+ (m2*l2*cos(t2)*(g-x_t*t2_t)) - M*g*h ;

eqn1 = diff(L,x) - diff(diff(L,x_t),t) + F == 0;
```

```
eqn2 = diff(L,t1) - diff(diff(L,t1_t),t) == 0;

eqn3 = diff(L,t2) - diff(diff(L,t2_t),t) == 0;

sol_t1_tt = isolate(eqn2, diff(t1,t,t));
sol_t2_tt = isolate(eqn3, diff(t2,t,t));

eqn1 = subs(eqn1,
{diff(t1,t,t),
diff(t2,t,t)},
{rhs(sol_t1_tt),
rhs(sol_t2_tt)});

sol_x_tt = isolate(eqn1, diff(x,t,t));

sol_t1_tt = subs(sol_t1_tt, diff(x,t,t), rhs(sol_x_tt));

sol_t2_tt = subs(sol_t2_tt, diff(x,t,t), rhs(sol_x_tt));

A = sym('A%d%d', [6 6]);
B = sym('B%d%d', [6,1]);

A(1,1) = 0;
A(1,2) = 1;
A(1,3) = 0;
A(1,4) = 0;
A(1,5) = 0;
A(1,6) = 0;

A(2,1) = subs( diff(rhs(sol_x_tt), x), {x, t1, t2}, {0, 0, 0} );
A(2,2) = subs( diff(rhs(sol_x_tt), x_t), {x, t1, t2}, {0, 0, 0} );
A(2,3) = subs( diff(rhs(sol_x_tt), t1), {x, t1, t2}, {0, 0, 0} );
```

```
A(2,4) = subs( diff(rhs(sol_x_tt), t1_t), {x, t1, t2}, {0, 0, 0} );
A(2,5) = subs( diff(rhs(sol_x_tt), t2), {x, t1, t2}, {0, 0, 0} );
A(2,6) = subs( diff(rhs(sol_x_tt), t2_t), {x, t1, t2}, {0, 0, 0} );

A(3,1) = 0;
A(3,2) = 0;
A(3,3) = 0;
A(3,4) = 1;
A(3,5) = 0;
A(3,6) = 0;

A(4,1) = subs( diff(rhs(sol_t1_tt), x), {x, t1, t2}, {0, 0, 0} );
A(4,2) = subs( diff(rhs(sol_t1_tt), x_t), {x, t1, t2}, {0, 0, 0} );
A(4,3) = subs( diff(rhs(sol_t1_tt), t1), {x, t1, t2}, {0, 0, 0} );
A(4,4) = subs( diff(rhs(sol_t1_tt), t1_t), {x, t1, t2}, {0, 0, 0} );
A(4,5) = subs( diff(rhs(sol_t1_tt), t2), {x, t1, t2}, {0, 0, 0} );
A(4,6) = subs( diff(rhs(sol_t1_tt), t2_t), {x, t1, t2}, {0, 0, 0} );

A(5,1) = 0;
A(5,2) = 0;
A(5,3) = 0;
A(5,4) = 0;
A(5,5) = 0;
A(5,6) = 1;

A(6,1) = subs( diff(rhs(sol_t2_tt), x), {x, t1, t2}, {0, 0, 0} );
A(6,2) = subs( diff(rhs(sol_t2_tt), x_t), {x, t1, t2}, {0, 0, 0} );
A(6,3) = subs( diff(rhs(sol_t2_tt), t1), {x, t1, t2}, {0, 0, 0} );
A(6,4) = subs( diff(rhs(sol_t2_tt), t1_t), {x, t1, t2}, {0, 0, 0} );
A(6,5) = subs( diff(rhs(sol_t2_tt), t2), {x, t1, t2}, {0, 0, 0} );
A(6,6) = subs( diff(rhs(sol_t2_tt), t2_t), {x, t1, t2}, {0, 0, 0} );
```

```
B(1,1) = 0;
B(2,1) = subs( diff(rhs(sol_x_tt), F), {x, t1, t2}, {0, 0, 0} );
B(3,1) = 0;
B(4,1) = subs( diff(rhs(sol_t1_tt), F), {x, t1, t2}, {0, 0, 0} );
B(5,1) = 0;
B(6,1) = subs( diff(rhs(sol_t2_tt), F), {x, t1, t2}, {0, 0, 0} );

cont_mat = [B A*B A^2*B A^3*B A^4*B A^5*B];
determinant_cont = det(cont_mat);
```

5.2 Part D

```
syms m1 g m2 M L1 L2 x dx
m1 = 100;
m2 = 100;
M = 1000;
L1 = 20;
L2 = 10;
g = 9.81;
tspan = 0:0.1:100;

% Initial conditions
X0 = [0 0 -deg2rad(20) 0 deg2rad(30) 0];

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*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0;
0 0 0 0 0 1;
0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
```

```
D = [1;0;0];
Rank = rank([B A*B (A^2)*B (A^3)*B (A^4)*B (A^5)*B]);

Q = [1/12 0 0 0 0 0;
0 1/(12*0.1) 0 0 0 0;
0 0 1/(8*0.2^2) 0 0 0;
0 0 0 1/(12*0.01^2) 0 0;
0 0 0 0 1/(8*0.2^2) 0;
0 0 0 0 0 1/(12*0.01^2)];
R = 0.00001;
[K,S,P] = lqr(A,B,Q,R);
sys = ss(A-B*K,B,C,D);
% step(sys,200);
eigenvalues_after_LQR = eig(A-B*K);

[t,x_linear] = ode45(@(t,X)linear(t,X,-K*X),tspan,X0);
figure(1);
hold on
plot(t,x_linear(:,1))
plot(t,x_linear(:,3))
plot(t,x_linear(:,5))
ylabel('state variables')
xlabel('time (sec)')
title('Linear system using LQR controller with initial
conditions - \theta_1 = -20 and \theta_2 = -30 degrees')
legend('x','\theta_1','\theta_2')

[t,x_nlinear] = ode45(@(t,X)nonLinear(t,X,-K*X),tspan,X0);
figure(2);
hold on
plot(t,x_nlinear(:,1))
plot(t,x_nlinear(:,3))
```

```
plot(t, x_nlinear(:,5))
ylabel('state variables')
xlabel('time (sec)')
title('Non-Linear system using LQR controller with initial
conditions - \theta_1 = -20 and \theta_2 = -30 degrees')
legend('x','\theta_1','\theta_2')

function dX = linear(t,X,U)
m1 = 100; m2 = 100; M = 1000; 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*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0;
0 0 0 0 0 1;
0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
dX = A*X + B*U;
end

function dX = nonLinear(t,X,F)
m1 = 100; m2 = 100; M = 1000; L1 = 20; L2 = 10; g = 9.81;
x = X(1);
dx = X(2);
t1 = X(3);
dt1 = X(4);
t2 = X(5);
dt2 = X(6);
dX=zeros(6,1);
dX(1) = dx;
dX(2) = (F-((m1*sin(t1)*cos(t1))+
(m2*sin(t2)*cos(t2)))*g -
```

```
(L1*m1*(dX(3)^2)*sin(t1)) -
(L2*m2*(dX(5)^2)*sin(t2)))/(m1+m2+M-(m1*(cos(t1)^2))
-(m2*(cos(t2)^2)));
dX(3) = dt1;
dX(4) = (cos(t1)*dX(2)-g*sin(t1))/L1;
dX(5) = dt2;
dX(6) = (cos(t2)*dX(2)-g*sin(t2))/L2;
end
```

5.3 Part E

```
syms m1 g m2 M L1 L2
m1 = 100;
m2 = 100;
M = 1000;
L1 = 20;
L2 = 10;
g = 9.81;
q0 = [2 0 deg2rad(30) 0 deg2rad(10) 0];
tspan = 0:0.1:100;

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*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0;
0 0 0 0 0 1;
0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C1 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0];
C2 = [0 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
C3 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 1 0];
```

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

D = [0; 0; 0];

Obs1_rank = rank([C1' A'*C1'
((A')^2)*C1'
((A')^3)*C1'
((A')^4)*C1'
((A')^5)*C1']);
Obs2_rank = rank([C2' A'*C2'
((A')^2)*C2'
((A')^3)*C2'
((A')^4)*C2'
((A')^5)*C2']);
Obs3_rank = rank([C3' A'*C3'
((A')^2)*C3'
((A')^3)*C3'
((A')^4)*C3'
((A')^5)*C3']);
Obs4_rank = rank([C4' A'*C4'
((A')^2)*C4'
((A')^3)*C4'
((A')^4)*C4'
((A')^5)*C4']);

sys1 = ss(A,B,C1,D);
sys3 = ss(A,B,C3,D);
sys4 = ss(A,B,C4,D);

fprintf('Rank of observability matrix for
Output vector 1 (x) = %d' , Obs1_rank);
```

```
fprintf('Rank of observability matrix for Output
vector 2 (theta1, theta2) = %d' , Obs2_rank);

fprintf('Rank of observability matrix for Output
vector 3 (x, theta2) = %d' , Obs3_rank);

fprintf('Rank of observability matrix for Output
vector 4 (x, theta1, theta2) = %d' , Obs4_rank);
```

5.4 Part F

```
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)];

% We previously determined that only C1, C3 and C4 are observable.
% Hence we will only use C1, C3 and C4.

% Output measurement vectors
```

```
C1 = [1 0 0 0 0 0];
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0;];
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];

D = 0;

Q = [1/12 0 0 0 0 0;
    0 1/(12*0.1) 0 0 0 0;
    0 0 1/(8*0.2^2) 0 0 0;
    0 0 0 1/(12*0.01^2) 0 0;
    0 0 0 0 1/(8*0.2^2) 0;
    0 0 0 0 0 1/(12*0.01^2)];

R = 0.00001;

% Considering 6 actual states + 6 estimate errors as the new state
X0=[0,0,deg2rad(30),0,deg2rad(-20),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]

% Calling LQR function to obtain K gain matrix
K=lqr(A,B,Q,R);

 poles=[-3;-6;-9;-12;-15;-18];
% Computing the L matrix by placing the negative poles to an even more
% negative value using place() MATLAB command

L1 = place(A',C1',poles)';
L3 = place(A',C3',poles)';
L4 = place(A',C4',poles)';
```

```
A_c1 = [(A-B*K) B*K; zeros(size(A)) (A-L1*C1)];% Luenberger A matrix
C_c1 = [C1 zeros(size(C1)); zeros(size(C1)) C1];% Luenberger C matrix

A_c3 = [(A-B*K) B*K; zeros(size(A)) (A-L3*C3)];% Luenberger A matrix
C_c3 = [C3 zeros(size(C3)); zeros(size(C3)) C3];% Luenberger C matrix

A_c4 = [(A-B*K) B*K; zeros(size(A)) (A-L4*C4)];% Luenberger A matrix
C_c4 = [C4 zeros(size(C4)); zeros(size(C4)) C4];% Luenberger C matrix

B_c = [B;zeros(size(B))];% Luenberger B matrix

tspan_lin_init = 0:100/499:100;
tspan_lin_step = 0:100/252:100;

%%MATLAB function to output statespace equations
sys_1 = ss(A_c1, B_c, C_c1,D);

 %In built function to give the response of the system to initial state
op_lin1_init = initial(sys_1,X0,100);



figure;
hold on
plot(tspan_lin_init(1:499),op_lin1_init(:,1))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Linear system initial state response output for Observer 1')
legend('x')

op_lin1_step = step(sys_1,100);%Gives the step response output
```

```
figure;
hold on
plot(tspan_lin_step(1:252),op_lin1_step(:,1))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Linear system step response output for Observer 1')
legend('x')
%MATLAB function to output statespace equations
sys_3 = ss(A_c3, B_c, C_c3,D);
%MATLAB inbuilt function to check the initial response of the system
op_lin3_init = initial(sys_3,X0,100);

figure;
hold on
plot(tspan_lin_init(1:499),op_lin3_init(:,1))
plot(tspan_lin_init(1:499),op_lin3_init(:,2))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Linear system initial state response output for Observer 3')
legend('x','\theta_2')

op_lin3_step = step(sys_3,100);% Gives the step response output

figure;
hold on
plot(tspan_lin_step(1:252),op_lin3_step(:,1))
plot(tspan_lin_step(1:252),op_lin3_step(:,2))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Linear system step response output for Observer 3')
legend('x','\theta_2')
%%MATLAB function to output statespace equations
```

```
sys_4 = ss(A_c4, B_c, C_c4, D);
%MATLAB inbuilt function to check the initial response of the system
op_lin4_init = initial(sys_4,X0,100);

figure;
hold on
plot(tspan_lin_init(1:499),op_lin3_init(:,1))
plot(tspan_lin_init(1:499),op_lin3_init(:,2))
plot(tspan_lin_init(1:499),op_lin3_init(:,3))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Linear system initial state response output for Observer 4')
legend('x','\theta_1','\theta_2')

op_lin4_step = step(sys_4,100);%Gives the step response output

figure;
hold on
plot(tspan_lin_step(1:252),op_lin3_step(:,1))
plot(tspan_lin_step(1:252),op_lin3_step(:,2))
plot(tspan_lin_step(1:252),op_lin3_step(:,3))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Linear system step response output for Observer 4')
legend('x','\theta_1','\theta_2')

tspan = 0:0.01:100

[t,y_nlinearobs1] = ode45(@(t,X)nlinear_obs(t,X,C1),tspan,X0);
figure;
hold on
plot(tspan,y_nlinearobs1(:,1))
```

```
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Non-linear system initial state response output for Observer 1')
legend('x')

[t,y_nlinearobs3] = ode45(@(t,X)nlinear_obs(t,X,C3),tspan,X0);
figure;
hold on
plot(tspan,y_nlinearobs3(:,1))
plot(tspan,y_nlinearobs3(:,5))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Non-linear system initial state response output for Observer 3')
legend('x','\theta_2')

[t,y_nlinearobs4] = ode45(@(t,X)nlinear_obs(t,X,C4),tspan,X0);
figure;
hold on
plot(tspan,y_nlinearobs4(:,1))
plot(tspan,y_nlinearobs4(:,3))
plot(tspan,y_nlinearobs4(:,5))
ylabel('State variables amplitude')
xlabel('time (sec)')
title('Non-linear system initial state response output for Observer 4')
legend('x','\theta_1','\theta2')

function dX = nlinear_obs(t,X,C)

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 = [1/12 0 0 0 0 0;
   0 1/(12*0.1) 0 0 0 0;
   0 0 1/(8*0.2^2) 0 0 0;
   0 0 0 1/(12*0.01^2) 0 0;
   0 0 0 0 1/(8*0.2^2) 0;
   0 0 0 0 0 1/(12*0.01^2)];

R = 0.00001;

D = 0;

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

F=-K_val*X(1:6);
poles=[-3;-6;-9;-12;-15;-18];

L = place(A',C',poles)';

temp = (A-L*C)*X(7:12);
```

```
dX=zeros(12,1);

dX(1) = X(2);%DX;
dX(2)=(F-(g/2)*(m1*sin(2*X(3))+m2*sin(2*X(5)))
-(m1*l1*(X(4)^2)*sin(X(3)))
-(m2*l2*(X(6)^2)*sin(X(5))))/(M+m1*((sin(X(3)))^2)
+m2*((sin(X(5)))^2));%DDX
dX(3)= X(4); %Dtheta1
dX(4)= (dX(2)*cos(X(3))-g*(sin(X(3))))/l1'; %DDtheta 1;
dX(5)= X(6); %Dtheta2
dX(6)= (dX(2)*cos(X(5))-g*(sin(X(5))))/l2; %DDtheta 2;
dX(7:12) = temp;
end
```

5.5 Part G

```
syms m1 g m2 M L1 L2 x dx
m1 = 100;
m2 = 100;
M = 1000;
L1 = 20;
L2 = 10;
g = 9.81;
tspan = 0:0.1:100;
% q = [x dx t1 dt1 t2 dt2];
%Enter initial conditions
X0 = [0,0,deg2rad(-20),0,deg2rad(30),0,0,0,0,0,0,0];

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*g)+(m1*g))/(M*L1) 0 -g*m2/(M*L1) 0;
0 0 0 0 0 1;
```

```
0 0 -m1*g/(M*L2) 0 -((M*g)+(m2*g))/(M*L2) 0];
B = [0; 1/M; 0; 1/(L1*M); 0; 1/(L2*M)];
C1 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0];
d = [1;0;0];
sys1 = ss(A,B,C1,d);

Q = [1/12 0 0 0 0 0;
    0 1/(12*0.1) 0 0 0 0;
    0 0 1/(8*0.2^2) 0 0 0;
    0 0 0 1/(12*0.01^2) 0 0;
    0 0 0 0 1/(8*0.2^2) 0;
    0 0 0 0 0 1/(12*0.01^2)];

R = 0.00001;

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

sys = ss(A-B*K,B,C1,d);

poles=[-3;-6;-9;-12;-15;-18];

L1 = place(A',C1',poles)';

vd=0.3*rand(1)*eye(6);              %Process Noise
vn=0.1*rand(1);                       %Measurement Noise
Ac1 = A-(L1*C1);
e_sys1 = ss(Ac1,[B L1],C1,0);

[t,X1] = ode45(@(t,X)nLinearLQG(t,X,C1,vd,vn),tspan,X0);
figure();
hold on
plot(t,X1(:,1))
```

```
ylabel('State Variable Amplitude')
xlabel('time (sec)')
title('Non-Linear System LQG for output vector: x(t)')
legend('x')
hold off

function dX = nLinearLQG(t,X,C,vd,vn)

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 = [1/12 0 0 0 0 0;
    0 1/(12*0.1) 0 0 0 0;
    0 0 1/(8*0.2^2) 0 0 0;
    0 0 0 1/(12*0.01^2) 0 0;
    0 0 0 0 1/(8*0.2^2) 0;
    0 0 0 0 0 1/(12*0.01^2)];

R = 0.00001;
```

```
D = 0;

K_val = lqr(A,B,Q,R);
F=-K_val*X(1:6);

% poles=[-3;-6;-9;-12;-15;-18];
%
% L = place(A',C',poles)';
%
% temp = (A-L*C)*X(7:12);

K_del=lqr(A',C',vd,vn)';

sd =(A-K_del*C)*X(7:12);

dX=zeros(12,1);

dX(1) = X(2);
dX(2)=(F-(g/2)*(m1*sin(2*X(3))+m2*sin(2*X(5)))
-(m1*l1*(X(4)^2)*sin(X(3)))
-(m2*l2*(X(6)^2)*sin(X(5))))/(M+m1*((sin(X(3)))^2)
+m2*((sin(X(5)))^2));%xDD
dX(3)= X(4);
dX(4)= (dX(2)*cos(X(3))-g*(sin(X(3))))/l1';
dX(5)= X(6);
dX(6)= (dX(2)*cos(X(5))-g*(sin(X(5))))/l2;
dX(7)= sd(1);
dX(8)= sd(2);
dX(9)= sd(3);
dX(10)= sd(4);
dX(11)= sd(5);
```

```
dX(12)=  sd(6);

end
```