University of Maryland, College Park

ENPM667 Final Project

Design of LQR and LQG controller for Two Pendulum Crane

Submitted By:
Mayank Deshpande (120387333)
Tanmay Pancholi (120116711)
Shreyas Acharya (120426643)

Contents

1 Equations of Motion

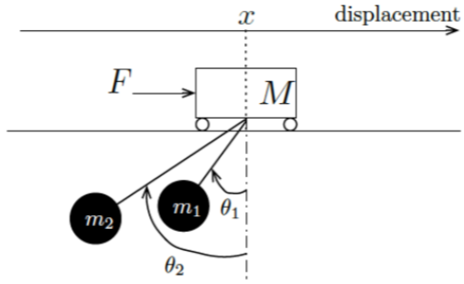


Figure 1: Crane with two suspended loads

We have to define the the motion dynamics of the two pendulum crane. In this project, Euler-Lagrange equation is used for this purpose. The Euler-Lagrange equations factors in the Kinetic Energy and Potential Energy of the system for formulating the equations of motion of the system.

Let us define the position and velocity equations for masses $\mathbf{m_1}$ and $\mathbf{m_2}$. The generalized coordinates as shown in the Figure 1 would be x, $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$.The equations can be defined as:

$$X_1 = (x - l_1 \sin(\theta_1))\,\hat{\imath} + (-l_1 \cos(\theta_1))\,\hat{\jmath}$$

$$X_2 = (x - l_2 \sin(\theta_2))\,\hat{\imath} + (-l_2 \cos(\theta_2))\,\hat{\jmath}$$

Corresponding equations of velocity can be obtained by differentiating the above equations with respect to time:

$$\dot{X}_1 = \left(\dot{x} - l_1 \cos(\theta_1)\,\dot{\theta}_1\right)\hat{\imath} + \left(l_1 \sin(\theta_1)\,\dot{\theta}_1\right)\hat{\jmath}$$

$$\dot{X}_2 = \left(\dot{x} - l_2 \cos(\theta_2)\,\dot{\theta}_2\right)\hat{\imath} + \left(l_2 \sin(\theta_2)\,\dot{\theta}_2\right)\hat{\jmath}$$

The Kinetic energy of cart can be formulated as:

$$K.E. = \frac{1}{2}\dot{x}^2(M) + \frac{1}{2}\left(m_1\dot{X_1}^2\right) + \frac{1}{2}\left(m_2\dot{X_2}^2\right)$$

$$= \frac{1}{2}\dot{x}^2(M) + \frac{1}{2}\left(m_1((\dot{x} - l_1\cos(\theta_1)\dot{\theta}_1)\hat{\imath} + (l_1\sin(\theta_1)\,\dot{\theta}_1)\hat{\jmath})^2\right)$$
$$+\frac{1}{2}\left(m_2((\dot{x} - l_2\cos(\theta_2)\dot{\theta}_2)\hat{\imath} + (l_2\sin(\theta_2)\dot{\theta}_2)\hat{\jmath})^2\right)$$

The Potential Energy of the system can be formulated as:

$$P.E. = -m_1 g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2$$

Therefore, the Lagrange $\mathcal{L}$ can be calculated as:

$$\mathcal{L} = K.E. - P.E.$$

$$\mathcal{L} = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\dot{x}^2 + \frac{1}{2}m_1 l_1^2\dot{\theta}_1^2\cos^2(\theta_1) - m_1 l_1\dot{\theta}_1\dot{x}\cos(\theta_1) + \frac{1}{2}m_1 l_1^2\dot{\theta}_1^2\sin^2(\theta_1)$$
$$+\frac{1}{2}m_2\dot{x}^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2\cos^2(\theta_2) - m_2 l_2\dot{\theta}_2\dot{x}\cos(\theta_2) + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2\sin^2(\theta_1) + m_1 g l_1\cos(\theta_1) + .$$
$$m_2 g l_2\cos(\theta_2)$$

Now we take the above result and solve the following Lagrange equations :

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right) - \left(\frac{\partial\mathcal{L}}{\partial x}\right) = F \tag{1}$$
$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right) - \left(\frac{\partial\mathcal{L}}{\partial\theta_1}\right) = 0 \tag{2}$$
$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_2}\right) - \left(\frac{\partial\mathcal{L}}{\partial\theta_2}\right) = 0 \tag{3}$$

Solving equation 1 we have:

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

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

Therefore,

$$\frac{d}{dt}\left[\frac{\partial\mathcal{L}}{\partial\dot{x}}\right] = \ddot{x}(m_1 + m_2 + M) - m_1 l_1\left(\ddot{\theta}_1 C_1 - S_1\dot{\theta}_1^2\right) - m_2 l_2\left(\ddot{\theta}_2 C_2 - S_2\dot{\theta}_2^2\right) = F \tag{4}$$

Now, Solving equation 2 we have:

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

Also,

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

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

Putting this in equation 2 we get:

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

$$=> m_1 l_1^2 \ddot{\theta}_1 - m_1 l_1 \ddot{x} \cos(\theta_1) + m_1 l_1 g \sin(\theta_1) = 0 \tag{5}$$

Now, solving equation 3 we get:

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

Also we have,

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

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

Therefore, we get

$$m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \ddot{x} \cos(\theta_2) + m_2 l_2 g \sin(\theta_2) = 0 \tag{6}$$

From 5 and 6 we get,

$$\ddot{\theta}_1 = \frac{\cos(\theta_1)\ddot{x} - g\sin(\theta_1)}{l_1} \tag{7}$$

$$\ddot{\theta}_2 = \frac{\cos(\theta_2)\ddot{x} - g\sin(\theta_1)}{l_2} \tag{8}$$

Putting this in equation 4 we have,

$$\ddot{x} = \frac{1}{M + m_1 + m_2}\left[F + m_1 l_1 \ddot{\theta}_1 \cos\theta_1 - m_1 l_1 \dot{\theta}_1^2 \sin\theta_1 + m_2 l_2 \ddot{\theta}_2 \cos\theta_2 - m_2 l_2 \dot{\theta}_2^2 \sin\theta_2\right] \tag{9}$$

1.1 Non-Linear state space representation:

Now, we can represent the above system in non-linear state space form as follows:

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

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

2 Linearizing the system around the equillibrium point

As we can see, the state space representation that we obtained above is non-linear in nature and it is evident that solving such complex equation is not feasible. In order to make the system less complex we linearize the system around an equillibrium point. The equillibrium point given in the problem statement is $x = 0, \theta_1 = 0$ and $\theta = 0$. Considering the following Jacobians equations for Linearizing:

$$A_F = \nabla_x F(x, u)$$
$$B_F = \nabla_u F(x, u)$$

The linearized A_l and B_l matrices are given by:

$$A_l = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} & \frac{\partial f_1}{\partial x_5} & \frac{\partial f_1}{\partial x_6} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} & \frac{\partial f_2}{\partial x_5} & \frac{\partial f_2}{\partial x_6} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} & \frac{\partial f_3}{\partial x_5} & \frac{\partial f_3}{\partial x_6} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} & \frac{\partial f_4}{\partial x_5} & \frac{\partial f_4}{\partial x_6} \\ \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial x_2} & \frac{\partial f_5}{\partial x_3} & \frac{\partial f_5}{\partial x_4} & \frac{\partial f_5}{\partial x_5} & \frac{\partial f_5}{\partial x_6} \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \frac{\partial f_6}{\partial x_3} & \frac{\partial f_6}{\partial x_4} & \frac{\partial f_6}{\partial x_5} & \frac{\partial f_6}{\partial x_6} \end{bmatrix}$$

and

$$B_l = \begin{bmatrix} \frac{\partial f1}{\partial u} \\ \frac{\partial f2}{\partial u} \\ \frac{\partial f3}{\partial u} \\ \frac{\partial f4}{\partial u} \\ \frac{\partial f5}{\partial u} \\ \frac{\partial f6}{\partial u} \end{bmatrix}$$

Substituting values in above equations we get:

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

and

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

Therefore the linearized state space representation is given as:

$$\begin{aligned} \dot{X} &= A_l X + B_l X \\ Y &= CX \end{aligned}$$

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

In the above equation $U = F$ and matrix C is given by:

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

3 Conditions for Controllability

Here, we obtain the controllability conditions of the linearized system for which the system is controllable. The State matrix A and input matrix B obtained by linearization of the system are considered for calculating the rank of this system. This is a LTI system. A LTI system is controllable if the Controllability matrix C(A,B) obtained is a fully ranked matrix.

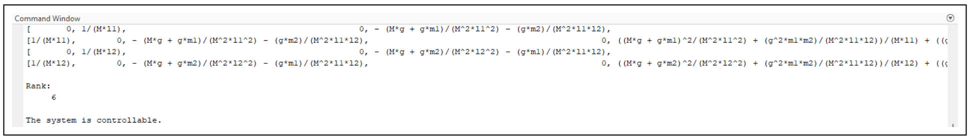
$$\text{rank}(C) = \text{rank}\begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix} = n$$

Here, we assume that the lengths of the load hanging from the crane are different ($l1 \neq l2$) and that the lengths are not equal to 0 ($l1 \neq l2 \neq 0$) , and the controllablity cannot be computed.

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

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

From the code above, the controllability matrix is obtained and we display the rank of the obtained controllability matrix.



```
Command Window
[       0, 1/(M*l1),                              0, - (M*g + g*m1)/(M^2*l1^2) - (g*m2)/(M^2*l1*l2),
[1/(M*l1),        0, - (M*g + g*m1)/(M^2*l1^2) - (g*m2)/(M^2*l1*l2),                              0, ((M*g + g*m1)^2/(M^2*l1^2) + (g^2*m1*m2)/(M^2*l1*l2))/(M*l1) + ((g
[       0, 1/(M*l2),                              0, - (M*g + g*m2)/(M^2*l2^2) - (g*m1)/(M^2*l1*l2),
[1/(M*l2),        0, - (M*g + g*m2)/(M^2*l2^2) - (g*m1)/(M^2*l1*l2),                              0, ((M*g + g*m2)^2/(M^2*l2^2) + (g^2*m1*m2)/(M^2*l1*l2))/(M*l2) + ((g

Rank:
     6

The system is controllable.
```

Figure 2: Rank of the Controllability Matrix

Acccording to the result obtained above, we can confirm that the linearized system is controllable.

4 LQR Controller Design

- To drive the output to desired state, we can design a LQR controller. A LQR controller gives the optimal state-feedback law for minimizing a particular quadratic objective function.
- A LQR cxontroller can be used for stabilizing the system.
- For a LQR controller, if the pair (A,B_k) is stablizable then we can look for k that can minimizes the following cost function:

$$J(k, \vec{X}(0)) = \int_0^\infty \vec{X}^T(t) Q \vec{X}(t) + \vec{U}_k^T(t) R \vec{U}_k(t) dt$$

  where Q is a symmetric positive, semi-definite matrix which penalizes the state variables (i.e bad performance) and R is a positive definite, matrix which penalizes the magnitude of the control input.
- Values of P and Q are taken to modify the system to give optimal output.
- According to the observation matde, lower the value of R, faster is the stabilizaition of the system.

4.1 Check for Controllability

For the given values M = 1000Kg, m1 = m2 = 100Kg, l1 = 20m and l2 = 10m, we check the controllability of the system.

```
% LQR Controller
clear variables;
clc;

% Given values of constraints of the system
M = 1000; % Mass of the Crane in Kgs
m1 = 100; % mass of load 1 in Kgs
m2 = 100; % mass of load 2 in Kgs
l1 = 20; % length of cable 1 in meters
l2 = 10; % length of cable 2 in meters
g = 9.81;

% Values of matrix A and B from the state-space representation of the linearized system
A = [0 1 0 0 0 0;
     0 0 -(g*m1)/M 0 -(g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 (-g*M -m1*g)/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 (-g*M -g*m2)/(M*l2) 0];

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

% Controllability Matrix for checking whether the system is controllable or not.
Contmatrix = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];

% checking the rank of the controllability matrix, matrix should be full rank i.e 6
rank_cont = rank(Contmatrix);
disp("Rank:");
disp(rank_cont);

if (rank(Contmatrix)==6)
    disp("Rank of Controllability matrix matches the order of A, so the System is controllable")
else
    disp("Rank of Controllability matrix does not match the order of A, so the System is uncontrollable")
end
```

Figure 3: Checking controllability of the system for given values

The output for the above code, clearly states that the matrix is controllable for the given values.

```
Command Window
Rank:
     6

Rank of Controllability matrix matches the order of A, so the System is controllable
```

Figure 4: System is controllable

4.2 Simulating the responses of the initial Conditions

4.2.1 Response to the initial conditions when applied to linearized system

We consider the initial conditions as follows:

Here, $x_{initial}$ is the initial state of the system, at t=0. Here $x_{initial}$ is a vector of 6 elements, which corresponds to the different initial states of the system.

```
%Respnse to initial conditions when applied to linearized system.

% The initial conditions are as follows.
X_initial = [0; 0; 6; 0; 9; 0];

% Set the duration to 60 seconds
duration = 60;

% Assumptions: C matrix is represented as the output matrix, which will
% make D=0

C = eye(6); D = 0;
sys1 = ss(A, B, C, D);

% inbuilt function in MATLAB to check the initial response of the system.
figure
initial(sys1, X_initial, duration)
grid on
```

Figure 5: Initial Conditions for obtaining initial Response

x_1 = initial position of the cart
x_2 = initial velocity of the cart
x_3 = initial angle of the first pendulum(θ_1)
x_4 = initial angular velocity of the first pendulum $\left(\dot{\theta}_1\right)$
x_5 = initial angle of the second pendulum(θ_2)
x_6 = initial angular velocity of the second pendulum $\left(\dot{\theta}_2\right)$

When the conditions are applied to the linearized system, we get the initial response as:

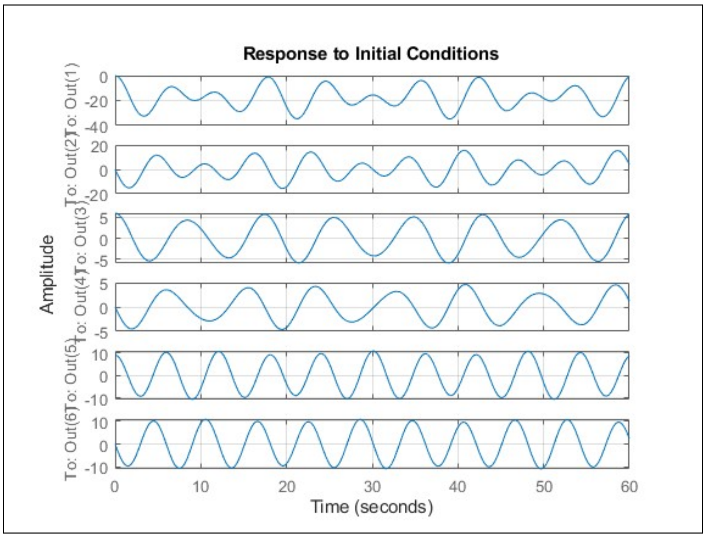


Figure 6: initial Response of the system

Note: We have not considered the cost functions Q and R

```
%Respnse to initial conditions when applied to linearized system.

% The initial conditions are as follows.
X_initial = [0; 0; 6; 0; 9; 0];

% Set the duration to 60 seconds
duration = 60;

% Assumptions: C matrix is represented as the output matrix, which will
% make D=0

C = eye(6); D = 0;
sys1 = ss(A, B, C, D);

% inbuilt function in MATLAB to check the initial response of the system.
figure
initial(sys1, X_initial, duration)
grid on


disp("Response of the linearized system when an LQR controller is obtained:")
% We assume the values of Q and R.
Q = [1500 0 0 0 0 0;
     0 1500 0 0 0 0;
     0 0 1500 0 0 0;
     0 0 0 1500 0 0;
     0 0 0 0 1500 0;
     0 0 0 0 0 1500];
R = 0.001;
```

Figure 7: Cost function Q and R

- Cost functions (Q1 and R1)

In the above code snippet, the initial cost functions (Q and R) are taken to check the initial response of the linearized system.

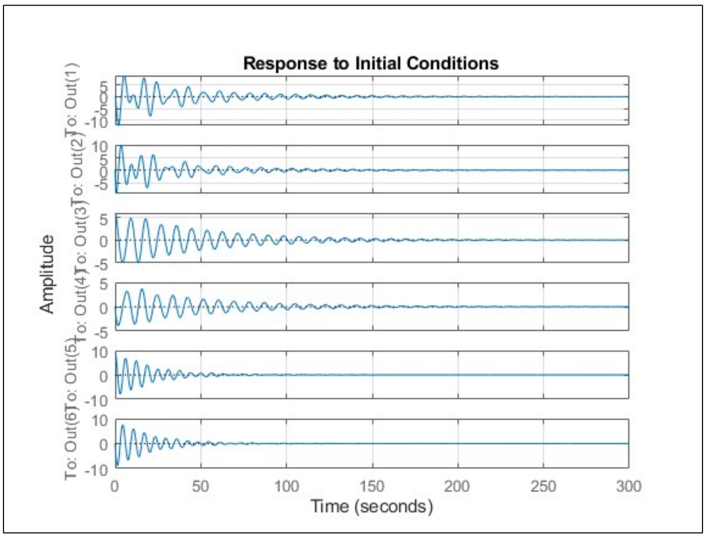


Figure 8: Initial Response

- Cost functions (Q2 and R2)

```
% The initial conditions are as follows.
X_tweaked = [0; 0; 15; 0; 25; 0];
% We assume the values of Q and R.
Q2 = [250 0 0 0 0 0;
      0 250 0 0 0 0;
      0 0 250 0 0 0;
      0 0 0 250 0 0;
      0 0 0 0 250 0;
      0 0 0 0 0 250];
R2 = 0.01;
```

Figure 9: Cost function(Q2, R2)

For the above cost functions, the system response is:

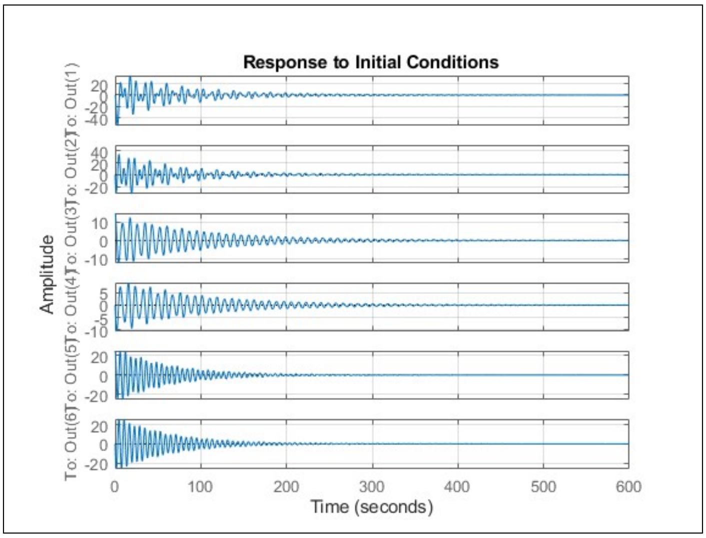


Figure 10: Initial Response for suitable conditions

4.2.2 Response to the initial conditions when applied to non-linear system

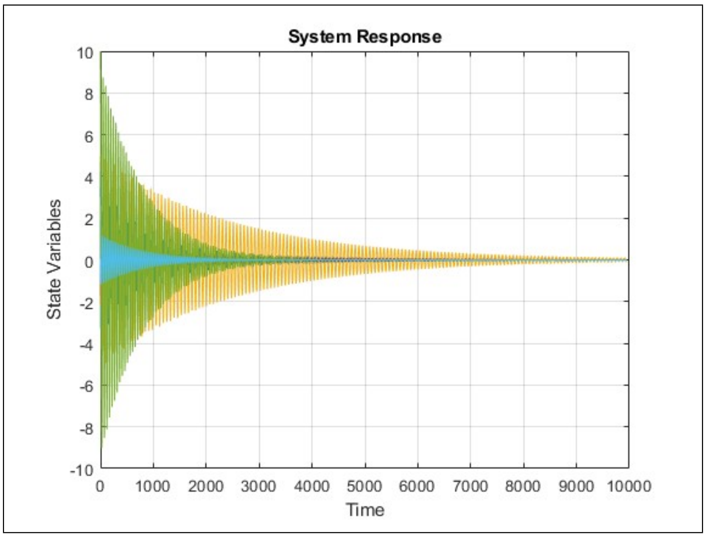


Figure 11: Initial Response for non-linear system

5 Checking the observability

The C matrices corresponding to $x(t)$, (θ_1, θ_2), $(x(t), \theta_2)$ and $(x(t), \theta_1, \theta_2)$ are measured for calculating the whole state

For $x(t)$,

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

For (θ_1, θ_2),

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

For $(x(t), \theta_2)$,

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

For $(x(t), \theta_1, \theta_2)$,

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

The system is considered observable when rank of observability matrix $\mathcal{O}$ is equal to n,

$$\text{O} = \begin{bmatrix} C \\ A_l C \\ A_l^2 C \\ A_l^3 C \\ A_l^4 C \\ A_l^5 C \end{bmatrix}$$

```
syms M m1 m2 l1 l2 g;

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

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

C1 = [1 0 0 0 0 0]; %Formulating with respect to x component
C2 = [0 0 1 0 0 0; 0 0 0 0 1 0]; %Formulating with respect to theta1 and theta2
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %Formulating with respect to x and theta2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %Formulating with respect to x,theta1 and theta2

Obs_check1 = rank([C1' A'*C1' ((A')^2)*C1' ((A')^3)*C1' ((A')^4)*C1' ((A')^5)*C1']);
Obs_check2 = rank([C1' A'*C2' ((A')^2)*C2' ((A')^3)*C2' ((A')^4)*C2' ((A')^5)*C2']);
Obs_check3 = rank([C1' A'*C3' ((A')^2)*C3' ((A')^3)*C3' ((A')^4)*C3' ((A')^5)*C3']);
Obs_check4 = rank([C1' A'*C4' ((A')^2)*C4' ((A')^3)*C4' ((A')^4)*C4' ((A')^5)*C4']);

disp(Obs_check1)
disp(Obs_check2)
disp(Obs_check3)
disp(Obs_check4)
```

Figure 12: Calculating rank of Observability matrix

In the code shown in Figure 14 we get that the rank of all the matrices except the one for (θ_1, θ_2) (which has rank 4) is 6. Therefore all $x(t)$, $(x(t), \theta_2)$ and $(x(t), \theta_1, \theta_2)$ are observable except (θ_1, θ_2).

6 Luenberger Observer

The Luenberger observer is represented as follows:

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

where L is the observer gain matrix and $(Y(t) - C\hat{x}(t))$ is the correction term. We can write the above equation in terms of $X_e(t)$ which is given by:

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

The below plots are resulted from the MATLAB simulation of Luenberger observer:

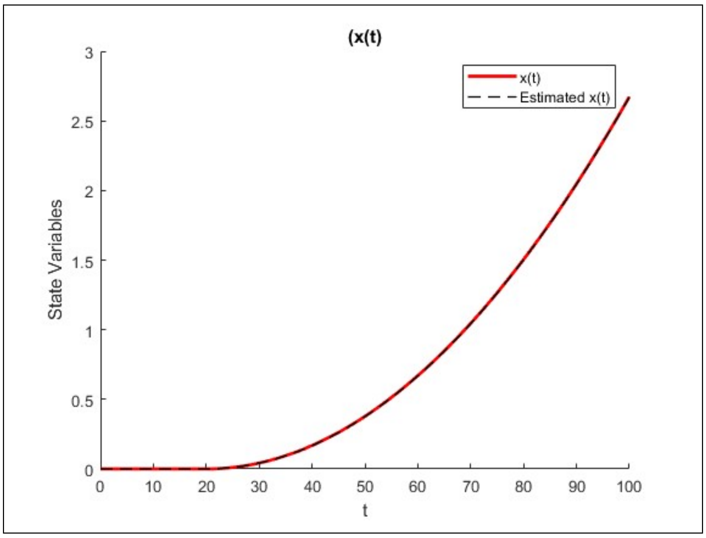


Figure 13: output vector response at step input for x(t)

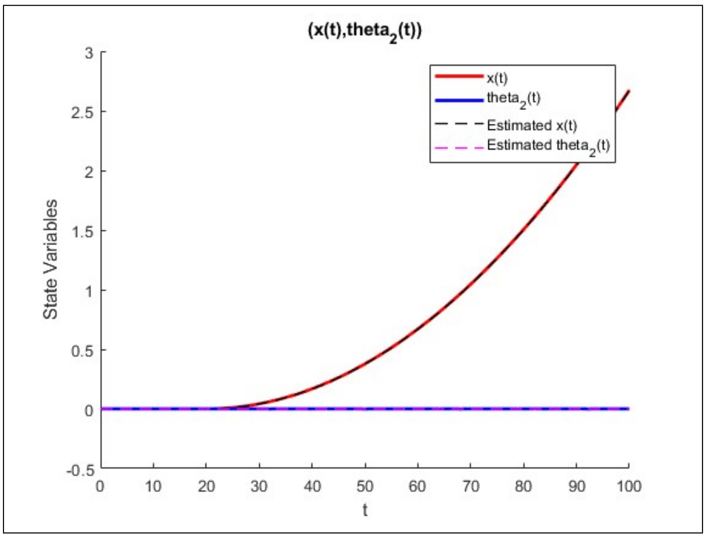


Figure 14: output vector response at step input for x(t), $\theta_2(t)$

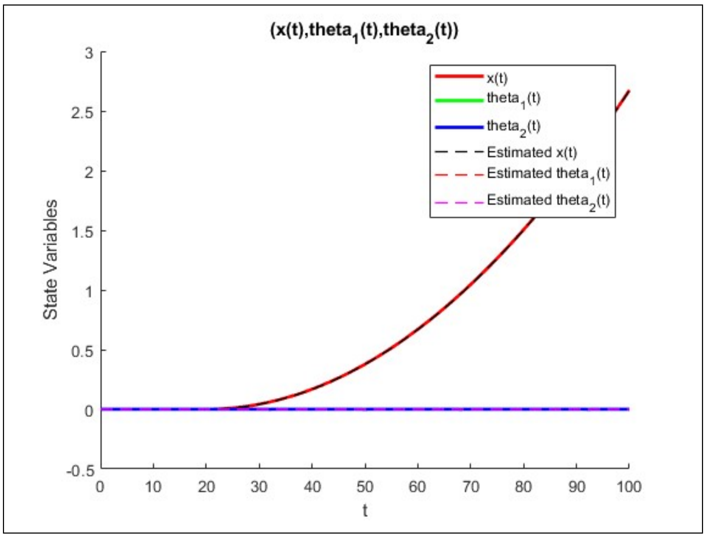


Figure 15: output vector response at step input for x(t), $\theta_2(t), \theta_1(t)$

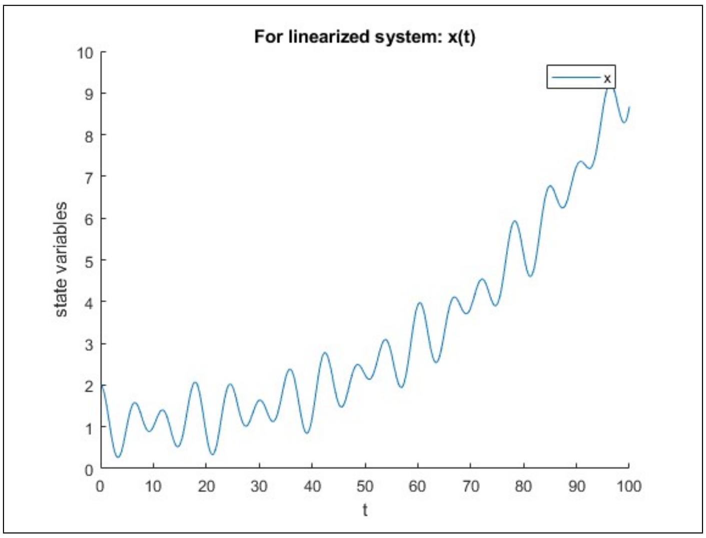


Figure 16: Linear system observer for x(t)

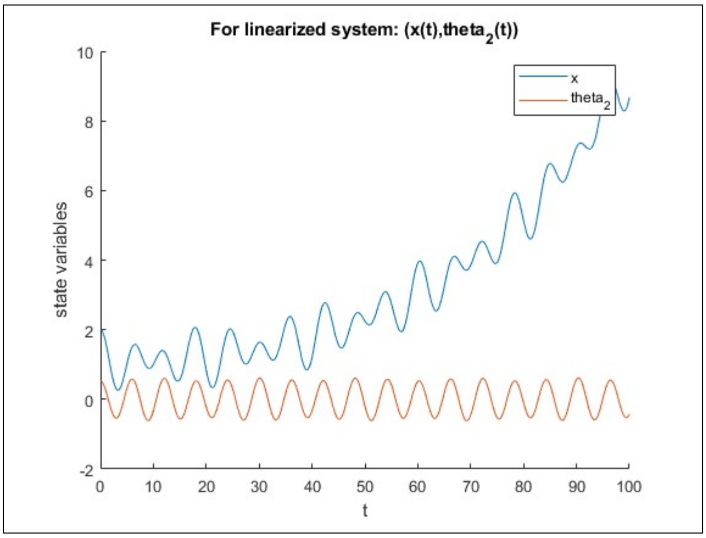


Figure 17: Linear system observer for x(t), $\theta_2(t)$

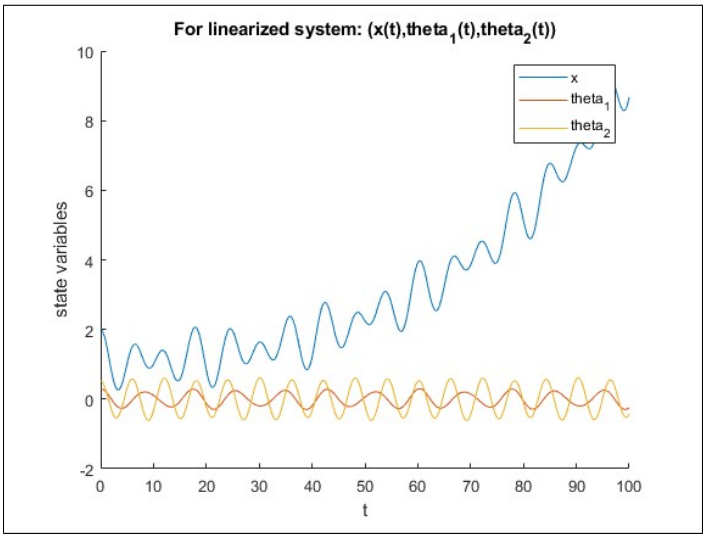


Figure 18: Linear system observer for x(t), $\theta_2(t), \theta_1(t)$

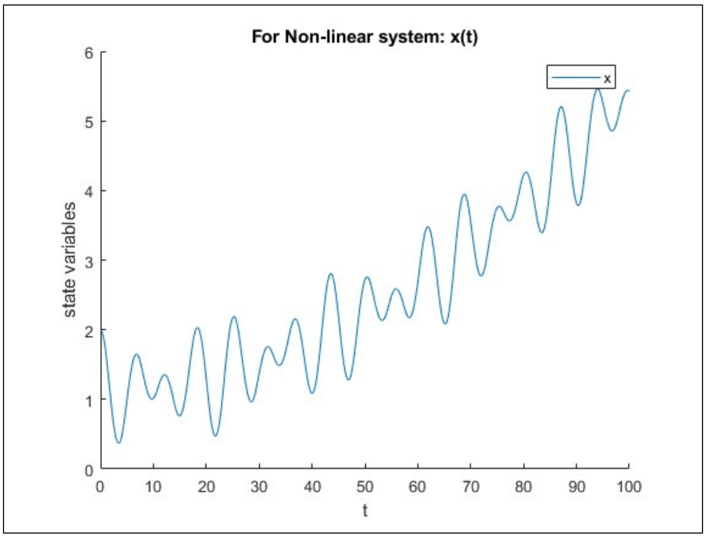


Figure 19: Non-Linear system observer for x(t)

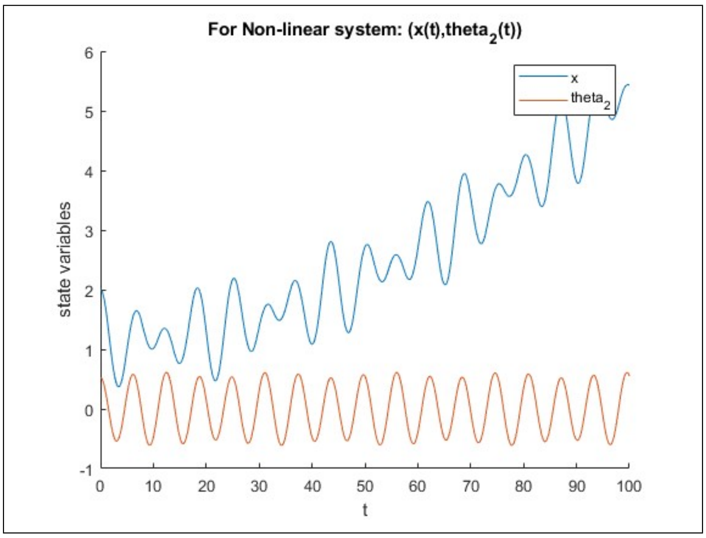


Figure 20: Non-Linear system observer for x(t), $\theta_2(t)$

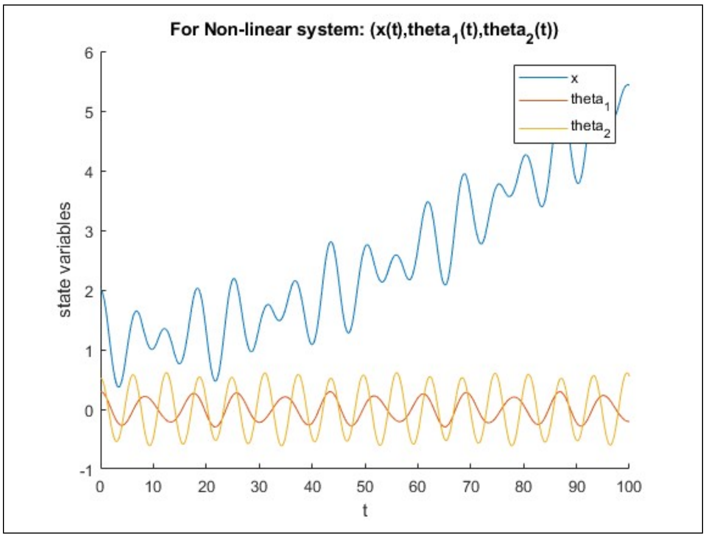


Figure 21: Non-Linear system observer for x(t), $\theta_2(t), \theta_1(t)$

7 LQG

The output vector prompts a reconfiguration of the controller for asymptotic tracking of a constant reference on x. This involves adjusting the Q penalty matrix, emphasizing the penalty on x(t) by setting it to 1, and potentially zeroing out penalties for other states.

With the configured Q penalty matrix, the controller exhibits robustness against constant force disturbances applied to the cart, effectively rejecting most disturbances on the cart's position x. The design's strength is evident in the minimal noise observed in the state, as depicted in the figure on the next page. This robustness extends to disturbances modeled in the controller design itself.

The graph illustrating these conditions incorporates White Noise (Vn) at 0.001, System Noise (Bd) at 0.01, and a 6x6 magnitude matrix. The Q matrix has its (1, 1) element set to 1000, while the rest of the diagonal elements are set to zero. The system starts from an initial state of x0 = 1 and converges to a final state of x = 20.

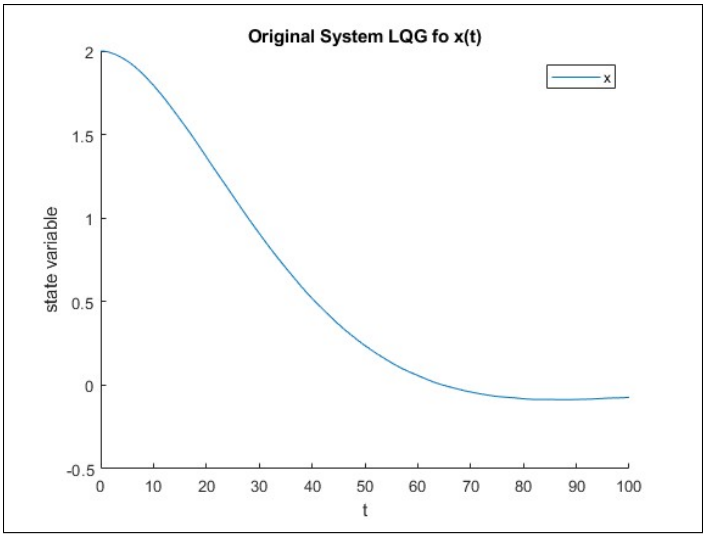


Figure 22: Non-Linear LQG

8 APPENDIX

8.1 Controllability Test.m

```
clear variables;
clc;
% Defining the variables
syms M m1 m2 l1 l2 g;

% state-space representation of the linearized system
A = [0 1 0 0 0 0;
    0 0 -(g*m1)/M 0 -(g*m2)/M 0;
    0 0 0 1 0 0;
    0 0 (-g*M -m1*g)/(M*l1) 0 -(g*m2)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(g*m1)/(M*l2) 0 (-g*M -g*m2)/(M*l2) 0];
disp("A")
disp(A)

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

disp("B")
disp(B)

% Controllability Matrix for checking whether the system is controllable
% not.

Contmatrix = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
disp(Contmatrix)

% checking the rank of the controllability of the matrix, matrix should b

rank_cont = rank(Contmatrix);
disp("Rank:");
disp(rank_cont);

% Condition of controllability (to check the determinant of the matrix

% Check if the system is controllable
isControllable = rank(Contmatrix) == 6;
```

```
% Display the result
if isControllable
    disp('The system is controllable.');
else
    disp('The system is not controllable.');
end
```

8.2 LQR.m

```
% LQR Controller
clear variables;
clc;

% Given values of constraints of the system
M = 1000; % Mass of the Crane in Kgs
m1 = 100; % mass of load 1 in Kgs
m2 = 100; % mass of load 2 in Kgs
l1 = 20; % length of cable 1 in meters
l2 = 10; % length of cable 2 in meters
g = 9.81;

% Values of matrix A and B from the state-space representation of the lin
A = [0 1 0 0 0 0;
     0 0 -(g*m1)/M 0 -(g*m2)/M 0;
     0 0 0 1 0 0;
     0 0 (-g*M -m1*g)/(M*l1) 0 -(g*m2)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(g*m1)/(M*l2) 0 (-g*M -g*m2)/(M*l2) 0];

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

% Controllability Matrix for checking whether the system is controllable
Contmatrix = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];

% checking the rank of the controllability matrix, matrix should be full
rank_cont = rank(Contmatrix);
disp("Rank:");
disp(rank_cont);
```

```
if (rank(Contmatrix)==6)
    disp("Rank of Controllability matrix matches the order of A, so the S
else
    disp("Rank of Controllability matrix does not match the order of A, s
end

%Respnse to initial conditions when applied to linearized system.

% The initial conditions are as follows.
X_initial = [0; 0; 6; 0; 9; 0];

% Set the duration to 60 seconds
duration = 60;

% Assumptions: C matrix is represented as the output matrix, which will
% make D=0

C = eye(6); D = 0;
sys1 = ss(A, B, C, D);

% inbuilt function in MATLAB to check the initial response of the system.
figure
initial(sys1, X_initial, duration)
grid on


disp("Response of the linearized system when an LQR controller is obtaine
% We assume the values of Q and R.
Q = [1500 0 0 0 0 0;
     0 1500 0 0 0 0;
     0 0 1500 0 0 0;
     0 0 0 1500 0 0;
     0 0 0 0 1500 0;
     0 0 0 0 0 1500];
R = 0.001;

% Q and R are the initial cost function defined in the LQR controller
% We use both the cost functions to get over the tradeoff
% inbuilt MATLAB function for designing an LQR controller
```

```
[K_Gain_matrix, Po_def_Ric, Poles] = lqr(A,B,Q,R);


sys_2 = ss(A-(B*K_Gain_matrix),B,C,D);
% Using the K matrix to define ss
figure
initial(sys_2,X_initial)
grid on


% The tweaked conditions are as follows.
X_tweaked = [0; 0; 15; 0; 25; 0];
% We assume the values of Q and R.
Q2 = [250 0 0 0 0 0;
     0 250 0 0 0 0;
     0 0 250 0 0 0;
     0 0 0 250 0 0;
     0 0 0 0 250 0;
     0 0 0 0 0 250];
R2 = 0.01;

% Q and R are the tweaked cost functions defined in the LQR controller
% We use both cost functions to get over the tradeoff
% inbuilt MATLAB function for designing an LQR controller
[K_Gain_matrix2, Po_def_Ric2, Poles2] = lqr(A,B,Q2,R2);


sys_3 = ss(A-(B*K_Gain_matrix2),B,C,D);
% Using the K matrix to define ss
figure
initial(sys_3,X_tweaked)
grid on
```

8.3 Linear Quadratic Regulator Non-linear.m

```
clear variables;
clc;

% Simulation of the original conditions obtained by the LQR controller to
% the original non-linear system
```

```
% The system will have single derivative of the of the values and all sta
% variables contribute to y_0

% Given values of constraints of the system
M = 1000; % Mass of the Crane in Kgs
m1 = 100; % mass of load 1 in Kgs
m2 = 100; % mass of load 2 in Kgs
l1 = 20; % length of cable 1 in meters
l2 = 10; % length of cable 2 in meters
g = 9.81;

% Initial conditions
y_0 = [5; 0; 5; 0; 10; 0];

% Defining the duration of the simulation
tspan = 0:0.01:9950;

% System parameters
params.M = M;
params.m1 = m1;
params.m2 = m2;
params.l1 = l1;
params.l2 = l2;
params.g = g;

%using inbuilt MATLAB function(ode45) to define the diff eqn
[t, y] = ode45(@(t, y) twoload(t, y, params), tspan, y_0);

% Plotting the function output on a 2D graph
figure
plot(t, y)
grid on
title('System Response')
xlabel('Time')
ylabel('State Variables')



function dydt = twoload(~, y, params)
```

```
M = params.M;
m1 = params.m1;
m2 = params.m2;
l1 = params.l1;
l2 = params.l2;
g = params.g;

% Define system 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 -(M*g+m2*g)/(M*l2) 0];

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

% Defining Q and R matrices for LQR
Q = [1500 0 0 0 0 0;
  0 1500 0 0 0 0;
  0 0 1500 0 0 0;
  0 0 0 1500 0 0;
  0 0 0 0 1500 0;
  0 0 0 0 0 1500];
R=0.01;

% Computing the LQR gain matrix
K_Gainmat = lqr(A,B,Q,R);

% control force based on the gain matrix and the current state
F=-K_Gainmat*y;


% Define system dynamics
dydt = zeros(6, 1);
dydt(1) = y(2);
dydt(2)=(F-(g/2)*(m1*sind(2*y(3))+m2*sind(2*y(5)))-(m1*l1*(y(4)^2)*si
dydt(3) = y(4);
dydt(4) = (dydt(2) * cosd(y(3)) - g * sind(y(3))) / l1;
dydt(5) = y(6);
```

```
    dydt(6) = (dydt(2) * cosd(y(5)) - g * sind(y(5))) / l2;
end
```

8.4 Observability.m

```
clear variables;
clc;

syms M m1 m2 l1 l2 g;

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

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

C1 = [1 0 0 0 0 0]; %Formulating with respect to x component
C2 = [0 0 1 0 0 0; 0 0 0 0 1 0]; %Formulating with respect to theta1 and
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %Formulating with respect to x and theta
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %Formulating with respect t

Obs_check1 = rank([C1' A'*C1' ((A')^2)*C1' ((A')^3)*C1' ((A')^4)*C1' ((A'
Obs_check2 = rank([C1' A'*C2' ((A')^2)*C2' ((A')^3)*C2' ((A')^4)*C2' ((A'
Obs_check3 = rank([C1' A'*C3' ((A')^2)*C3' ((A')^3)*C3' ((A')^4)*C3' ((A'
Obs_check4 = rank([C1' A'*C4' ((A')^2)*C4' ((A')^3)*C4' ((A')^4)*C4' ((A'

disp(Obs_check1)
disp(Obs_check2)
disp(Obs_check3)
disp(Obs_check4)
```

8.5 Luenberger.m

```
clear variables;
clc;
% Defining variables
```

```
m1 = 100;
m2 = 100;
M = 1000;
l1 = 20;
l2 = 10;
g = 9.81;
q0 = [2 0 deg2rad(17) 0 deg2rad(30) 0];
t_span = 0:0.1:100;

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

B = [0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
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([c1' A'*c1' ((A')^2)*c1' ((A')^3)*c1' ((A')^4)*c1' ((A')^5)*c
Obs2 = rank([c2' A'*c2' ((A')^2)*c2' ((A')^3)*c2' ((A')^4)*c2' ((A')^5)*c
Obs3 = rank([c3' A'*c3' ((A')^2)*c3' ((A')^3)*c3' ((A')^4)*c3' ((A')^5)*c
Obs4 = rank([c4' A'*c4' ((A')^2)*c4' ((A')^3)*c4' ((A')^4)*c4' ((A')^5)*c

sys1 = ss(A,B,c1,d);
sys3 = ss(A,B,c3,d);
sys4 = ss(A,B,c4,d);

% Kalman Estimator Design
Bd = 0.01*eye(6);                    %Process Noise
Vn = 0.001;                          %Measurement Noise
[L1,P,E] = lqe(A,Bd,c1,Bd,Vn*eye(3));
[L3,P,E] = lqe(A,Bd,c3,Bd,Vn*eye(3));
[L4,P,E] = lqe(A,Bd,c4,Bd,Vn*eye(3));

Ac1 = A-(L1*c1);
```

```
Ac3 = A-(L3*c3);
Ac4 = A-(L4*c4);
e_sys1 = ss(Ac1,[B L1],c1,0);
e_sys3 = ss(Ac3,[B L3],c3,0);
e_sys4 = ss(Ac4,[B L4],c4,0);


% Generating plot for step input
u_Step = 0*t_span;
u_Step(200:length(t_span)) = 1;

[y1,t] = lsim(sys1,u_Step,t_span);
[x1,t] = lsim(e_sys1,[u_Step;y1'],t_span);

[y3,t] = lsim(sys3,u_Step,t_span);
[x3,t] = lsim(e_sys3,[u_Step;y3'],t_span);

[y4,t] = lsim(sys4,u_Step,t_span);
[x4,t] = lsim(e_sys4,[u_Step;y4'],t_span);

figure();
hold on
plot(t,y1(:,1),'r','Linewidth',2)
plot(t,x1(:,1),'k--','Linewidth',1)
ylabel('State Variables')
xlabel('t')
legend('x(t)','Estimated x(t)')
title('(x(t)')
hold off

figure();
hold on
plot(t,y3(:,1),'r','Linewidth',2)
plot(t,y3(:,3),'b','Linewidth',2)
plot(t,x3(:,1),'k--','Linewidth',1)
plot(t,x3(:,3),'m--','Linewidth',1)
ylabel('State Variables')
xlabel('t')
legend('x(t)','theta_2(t)','Estimated x(t)','Estimated theta_2(t)')
title('(x(t),theta_2(t))')
```

```
hold off

figure();
hold on
plot(t,y4(:,1),'r','Linewidth',2)
plot(t,y4(:,2),'g','Linewidth',2)
plot(t,y4(:,3),'b','Linewidth',2)
plot(t,x4(:,1),'k--','Linewidth',1)
plot(t,x4(:,2),'r--','Linewidth',1)
plot(t,x4(:,3),'m--','Linewidth',1)
ylabel('State Variables')
xlabel('t')
legend('x(t)','theta_1(t)','theta_2(t)','Estimated x(t)','Estimated theta
title('(x(t),theta_1(t),theta_2(t))')
hold off

% Leuenberger Observer Response for linearized system
[t,q1] = ode45(@(t,q)linearObs1(t,q,L1),t_span,q0);
figure();
hold on
plot(t,q1(:,1))
ylabel('state variables')
xlabel('t')
title('For linearized system: x(t)')
legend('x')
hold off

[t,q3] = ode45(@(t,q)linearObs3(t,q,L3),t_span,q0);
figure();
hold on
plot(t,q3(:,1))
plot(t,q3(:,5))
ylabel('state variables')
xlabel('t')
title('For linearized system: (x(t),theta_2(t))')
legend('x','theta_2')
hold off

[t,q4] = ode45(@(t,q)linearObs4(t,q,L4),t_span,q0);
figure();
```

```
hold on
plot(t,q4(:,1))
plot(t,q4(:,3))
plot(t,q4(:,5))
ylabel('state variables')
xlabel('t')
title('For linearized system: (x(t),theta_1(t),theta_2(t))')
legend('x','theta_1','theta_2')
hold off

% Leuenberger Observer Response for the original non-linear system
[t,q1] = ode45(@(t,q)nonLinearObs1(t,q,1,L1),t_span,q0);
figure();
hold on
plot(t,q1(:,1))
ylabel('state variables')
xlabel('t')
title('For Non-linear system: x(t)')
legend('x')
hold off

[t,q3] = ode45(@(t,q)nonLinearObs3(t,q,1,L3),t_span,q0);
figure();
hold on
plot(t,q3(:,1))
plot(t,q3(:,5))
ylabel('state variables')
xlabel('t')
title('For Non-linear system: (x(t),theta_2(t))')
legend('x','theta_2')
hold off

[t,q4] = ode45(@(t,q)nonLinearObs4(t,q,1,L4),t_span,q0);
figure();
hold on
plot(t,q4(:,1))
plot(t,q4(:,3))
plot(t,q4(:,5))
ylabel('state variables')
xlabel('t')
```

```
title('For Non-linear system: (x(t),theta_1(t),theta_2(t))')
legend('x','theta_1','theta_2')
hold off
```

8.6 LQG.m

```
clear variables;
clc;

% Defining variables

m1 = 100;
m2 = 100;
M = 1000;
l1 = 20;
l2 = 10;
g = 9.81;
t_span = 0:0.1:100;
% q = [x dx t1 dt1 t2 dt2];
%Enter initial conditions
q0 = [2 0 deg2rad(0) 0 deg2rad(0) 0];

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

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

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);

% LQR Controller
Q = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0; 0 0
R = 0.1;
```

```
[K,S,P] = lqr(A,B,Q,R);
sys = ss(A-B*K,B,c1,d);
% step(sys,200);

% Kalman Estimator Design
Bd = 0.01*eye(6);                    %Process Noise
Vn = 0.001;                          %Measurement Noise
[L1,P,E] = lqe(A,Bd,c1,Bd,Vn*eye(3)); %Considering vector output: x(t)
Ac1 = A-(L1*c1);
e_sys1 = ss(Ac1,[B L1],c1,0);

% Non-linear Model LQG Response
[t,q1] = ode45(@(t,q)nonLinearObs1(t,q,-K*q,L1),t_span,q0);
figure();
hold on
plot(t,q1(:,1))
ylabel('state variable')
xlabel('t')
title('Original System LQG fo x(t)')
legend('x')
hold off
```

9 SIMULATIONS

The link of the Google drive for simulation is https://drive.google.com/file/d/1RbtqaX3azuyVcVZx0h
crdCMGRYsj9L/view?usp=sharing