ENPM – 667

CONTROL OF ROBOTIC SYSTEMS

FINAL PROJECT REPORT

Author: Kumara Ritvik Oruganti (117368963)

Teammate**:** Venkata Sairam Polina **(**118436579**)**

Date: 12/20/2021

Table of Contents

List of Figures

CHAPTER – 1

PROBLEM STATEMENT

1.1 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 in the next chapter depicts the crane and associated variables used throughout this project.

1.2 Problem 1:

Obtain the equations of motion for the system and the corresponding nonlinear state-space representation. This is addressed in the chapter 2 of this report.

1.3 Problem 2:

Obtain the linearized system around the equilibrium point specified by x=0 and $\theta_1 = \theta_2 = 0$. Write the state-space representation of the linearized system. This problem is solved in the Chapter 3.

1.4 Problem 3:

Obtain conditions on M, m_1, m_2, l_1 and l_2 which the linearized system is controllable. The condition is obtained in the Chapter 4 of this project report.

1.5 Problem 4:

Choose $M = 1000$ Kg, $m_1 = m_2 = 100$Kg, $l_1 = 20m$, $l_2 = 10m$. Check that the system is controllable and obtain an LQR controller. Simulate the resulting response to initial conditions when the controller is applied to the linearized system and also to the original nonlinear system. Adjust the parameters of the LQR cost until you obtain a suitable response. Use Lyapunov's indirect method to certify stability (locally or globally) of the closed-loop system. Refer to the Chapter 5 of this report for code and the outputs.

1.6 Problem 5:

Suppose that you can select the following output vectors: $x(t), (\theta_1(t), \theta_2(t)), (x(t), \theta_2(t))$ or $(x(t), \theta_1(t), \theta_2(t))$. Determine for which output vectors the linearized system is observable. The observability is checked using MATLAB and the code is attached in the Chapter 6.

1.7 Problem 6:

Obtain your "best" Luenberger observer for each one of the output vectors for which the system is observable and simulate its response to initial conditions and unit step input. The simulation should be done for the observer applied to both the linearized system and the original nonlinear system. This section is addressed in the Chapter 7.

1.8 Problem 7:

Design an output feedback controller for your choice of the "smallest" output vector. Use the LQG method and apply the resulting output feedback controller to the original nonlinear system. Obtain your best design and illustrate its performance in simulation. How would you reconfigure your controller to asymptotically track a constant reference on *x*? Will you design reject-constant force disturbances applied on the cart? The LQG controller is designed, and the state response is plotted using MATLAB. Refer to the Chapter 8 for simulation results are shown in the Chapter 8.

CHAPTER 2

INTRODUCTION

This project deals with formulation of dynamic equations for crane which suspends two loads, followed by linearizing a non-linear system and the design of LQR controller, Luenberger observer and LQG controller. Controllability, observability for this system is verified along with the simulation of suitable Luenberger observer. The dynamic equations are setup numerically and non-linear state space equation is formulated which is then linearized around an equilibrium point using MATLAB. The remaining problems have been solved using code in MATLAB. The given Crane and suspended loads are shown in figure below.

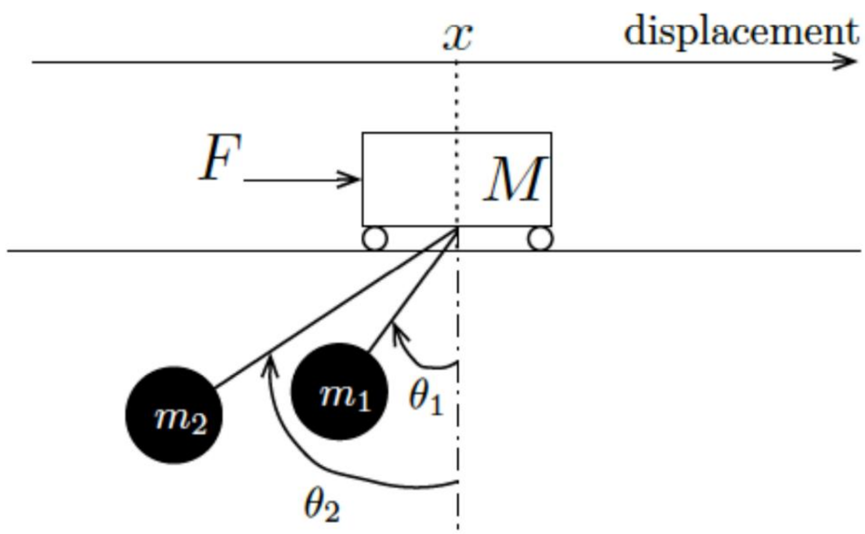

Figure 1 Inverted Dual Pendulum

CHAPTER - 3

EQUATIONS FOR MOTION AND NON-LINEAR STATE SPACE REPRESENTATION

Let us consider, that the crane is aligned along x-axis and two loads are suspended below the crane having masses m_1 and m_2 and their cable lengths are l_1 and l_2 and force (F) acts on the crane. Parameters used for deriving dynamic equations are defined below:

m_1 = mass of load1

m_2 = mass of load2

M= mass of crane

g= acceleration due to gravity

l_1= length of cable 1

l_2= length of cable 2

$\theta_{1=}$ angle made by first load with vertical

$\theta_{2=}$ angle made by second load with vertical

h_1= Effective height of load1 from base of the crane

h_2= Effective height of load2 from base of the crane

F= external force

Position of mass (m_1) in parametric form is given by:

$$r_1 = (x - l_1 sin\theta_1, - l_1 cos\theta_1) \tag{1}$$

$$r_1 = (x - l_1 sin\theta)\ \hat{\imath} - l_1 cos\theta_1\ \hat{\jmath} \tag{2}$$

differentiating r_1 with respect to time we get,

$$\frac{d}{dt}(r_1) = (\dot{x} - l_1 cos\theta_1\ \dot{\theta_1})\ \hat{\imath} + (l_1 sin\theta_1 \dot{\theta_1})\ \hat{\jmath} \tag{3}$$

Let,

$$\frac{d}{dt}(r_1) = V_{1x} + V_{1y} \tag{4}$$

[where $V_{1x} = \dot{x} - l_1 cos\theta_1 \, \dot{\theta_1}$ and $V_{1y} = l_1 sin\theta_1 \dot{\theta_1}$]

Position of mass (m_2) in parametric form is given by:

$$r_2 = (x - l_2 sin\theta_2, - l_1 cos\theta_2) \tag{5}$$

$$r_2 = (x - l_2 sin\theta_2)\, \hat{\imath} - (l_1 cos\theta_2)\, \hat{\jmath} \tag{6}$$

Differentiating r_2 with respect to time we get

$$\frac{d}{dt}(r_2) = (\dot{x} - l_2 cos\theta_2 \, \dot{\theta_2})\, \hat{\imath} + (l_2 sin\theta_2 \dot{\theta_2})\hat{\jmath} \tag{7}$$

$$\frac{d}{dt}(r_2) = V_{2x} + V_{2y} \tag{8}$$

[where $V_{2x} = \dot{x} - l_2 cos\theta_2 \, \dot{\theta}_2$ and $V_{2y} = l_2 sin\theta_2 \dot{\theta_2}$]

2.1 Kinetic Energy of System(K) is calculated as follows:

Where M= mass of cart

m_1= mass of load 1

m_2= mass of load 2

$$K = \frac{1}{2} M\dot{x}^2 + \frac{1}{2} m_1\left(\sqrt{V_{1x} + V_{1y}}\right)^2 + \frac{1}{2} m_2\left(\sqrt{V_{2x} + V_{2y}}\right)^2 \tag{9}$$

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

2.2 Potential Energy of the system(P) is calculated as follows:

For mass m_1, P_1= $m_1 g h_1$

where m_1= mass of load 1

g= acceleration due to gravity

h_1= Effective height of load1 from base of the cart

From figure [1] - $h_1 = -l_1 cos\theta_1$

$$P_1 = -m_1 g l_1 cos\theta_1 \tag{11}$$

similarly, for second load Potential energy is

$$P_2 = -m_2 g l_2 cos\theta_2 \tag{12}$$

Total Potential Energy of system $P = P_1 + P_2$

$$P = -m_1 g l_1 cos\theta_1 \pm m_2 g l_2 cos\theta_2 \tag{13}$$

Lagrangian L is computed as = Kinetic Energy – Potential Energy

= K – P

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

Rearranging terms we get,

$$\begin{aligned} L &= \frac{1}{2}M\dot{x}^2 + \frac{1}{2}\ m_1\left(\dot{x} - l_1 cos\theta_1\ \dot{\theta_1}\right)^2 + \frac{1}{2}\ m_1\left(l_1 sin\theta_1\dot{\theta_1}\ \right)^2 + \frac{1}{2}\ m_2\left(\dot{x} - l_2 cos\theta_2\ \dot{\theta_2}\right)^2 \\ &+ \frac{1}{2}\ m_2\left(l_2 sin\theta_2\dot{\theta_2}\ \right)^2 \qquad + m_1 g l_1 cos\theta_1 \\ &+ m_2 g l_2 cos\theta_2) \end{aligned} \tag{15}$$

From Lagrangian mechanics we can write for the system of crane:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F \tag{16}$$

$$\frac{\partial L}{\partial x} = 0 \quad [since\ no\ x\ terms] \tag{17}$$

$$\frac{\partial L}{\partial \dot{x}} = M\dot{x} + m_1\left(\dot{x} - l_1 cos\theta_1\ \dot{\theta_1}\right) + m_2\left(\dot{x} - l_2 cos\theta_2\ \dot{\theta_2}\right) \tag{18}$$

Differentiating $\frac{\partial L}{\partial \dot{x}}$ with respect to time we get,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = M\ddot{x} + m_1\left[\ddot{x} - l_1\left(cos\theta_1\,\ddot{\theta_1} - \dot{\theta}_1^2 sin\theta_1\right)\right] + m_2\left[\ddot{x} - l_2\left(cos\theta_2\,\ddot{\theta_2} - \dot{\theta}_2^2\,sin\theta_2\right)\right] \quad (19)$$

Substituting equations (19) and (17) in equation (16) we get,

$$F = M\ddot{x} + m_1\ddot{x} + m_2\ddot{x} - m_1 l_1 cos\theta_1\,\ddot{\theta_1} + m_1 l_1 \dot{\theta}_1^2 sin\theta_1 - m_2 l_2 cos\theta_2\,\ddot{\theta_2} + m_2 l_2 \dot{\theta}_2^2\,sin\theta_2 \qquad (20)$$

$$(M + m_1 + m_2)\ddot{x} = F + m_1 l_1 cos\theta_1\,\ddot{\theta_1} - m_1 l_1 \dot{\theta}_1^2 sin\theta_1 + m_2 l_2 cos\theta_2\,\ddot{\theta_2} - m_2 l_2 \dot{\theta}_2^2\,sin\theta_2 \qquad (21)$$

$$\ddot{x} = \frac{F + m_1 l_1 cos\theta_1\,\ddot{\theta_1} - m_1 l_1 \dot{\theta}_1^2 sin\theta_1 + m_2 l_2 cos\theta_2\,\ddot{\theta_2} - m_2 l_2 \dot{\theta}_2^2\,sin\theta_2}{(M + m_1 + m_2)} \qquad (22)$$

As we know,

$$\mathrm{L} = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}\,m_1\left(\dot{x} - l_1 cos\theta_1\,\dot{\theta_1}\right)^2 + \frac{1}{2}\,m_1\left(l_1 sin\theta_1 \dot{\theta_1}\right)^2 + \frac{1}{2}\,m_2\left(\dot{x} - l_2 cos\theta_2\,\dot{\theta_2}\right)^2 + \frac{1}{2}\,m_2\left(l_2 sin\theta_2 \dot{\theta_2}\right)^2 + m_1 g l_1 cos\theta_1 + m_2 g l_2 cos\theta_2) \qquad (23)$$

Writing Lagrangian equation for load 1,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = 0 \qquad (24)$$

$$\frac{\partial L}{\partial \theta_1} = m_1(\dot{x} - l_1 cos\theta_1\,\dot{\theta_1})(l_1 sin\theta_1 \dot{\theta_1}) + m_1(l_1 sin\theta_1 \dot{\theta_1})\,l_1 cos\theta_1\,\dot{\theta_1} - m_1 g l_1 sin\theta_1 \qquad (25)$$

$$\frac{\partial L}{\partial \theta_1} = m_1\dot{x}\,l_1 sin\theta_1 \dot{\theta_1} - m_1 g l_1 sin\theta_1 \qquad (26)$$

$$\frac{\partial L}{\partial \theta_1} = m_1\, l_1 sin\theta_1\left[\dot{x}\dot{\theta_1}\ - g\right] \tag{27}$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = m_1(\dot{x} - l_1 cos\theta_1\ \dot{\theta_1})(-l_1 cos\theta_1) + m_1(l_1 sin\theta_1\dot{\theta_1}\,)(l_1 sin\theta_1) \tag{28}$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = -m_1\dot{x}l_1 cos\theta_1 + m_1 l_1^2\dot{\theta_1} \tag{29}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = -m_1\ddot{x}l_1 cos\theta_1 + m_1\dot{x}l_1 sin\theta_1\ \dot{\theta}_1 + m_1 l_1^2\ddot{\theta_1} \tag{30}$$

Substituting (27) and (30) in equation in (24), we get

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1} = -m_1\ddot{x}l_1 cos\theta_1 + m_1\dot{x}l_1 sin\theta_1\ \dot{\theta}_1 + m_1 l_1^2\ddot{\theta_1} - \left(m_1\ l_1 sin\theta_1\left[\dot{x}\dot{\theta_1}\ - g\right]\right) \tag{31}$$

Up on simplifying we get,

$$\ddot{\theta_1} = \frac{\ddot{x}cos\theta_1 - gsin\theta_1}{l_1} \tag{32}$$

Similarly, we get

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2} = 0 \tag{33}$$

$$\ddot{\theta_2} = \frac{\ddot{x}cos\theta_2 - gsin\theta_2}{l_2} \tag{34}$$

Substituting (32) and (34) in (22), we get

Equation (22) is:

$$\ddot{x} = \frac{F + m_1 l_1 cos\theta_1\ \ddot{\theta_1} - m_1 l_1\dot{\theta}_1^2 sin\theta_1 + m_2 l_2 cos\theta_2\ \ddot{\theta_2} - m_2 l_2\dot{\theta}_2^2\ sin\theta_2}{(M + m_1 + m_2)} \tag{22}$$

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

$$(M + m_1 + m_2)\ddot{x}$$
$$= F + m_1\ddot{x}cos\theta_1^2 - m_1 gcos\theta_1 sin\theta_1 \ + m_2\ddot{x}cos\theta_2^2$$
$$- m_2 gcos\theta_2 sin\theta_2(\ddot{x}cos\theta_2 - gsin\theta_2) - m_1 l_1\dot{\theta}_1^2 sin\theta_1$$
$$- m_2 l_2\dot{\theta}_2^2 \ sin\theta_2 \tag{36}$$

$(M + m_1 + m_2 - m_1 cos\theta_1^2 - m_2 cos\theta_2^2)\ddot{x} = F - m_1 gcos\theta_1 sin\theta_1 \ - m_2 gcos\theta_2 sin\theta_2 -$
$m_1 l_1\dot{\theta}_1^2 sin\theta_1 - m_2 l_2\dot{\theta}_2^2 \ sin\theta_2$ (37)

Now,

$$\ddot{x} = \frac{1}{(M + m_1 + m_2 - m_1 cos\theta_1^2 - m_2 cos\theta_2^2)}\left[F - m_1 sin\theta_1\left[gcos\theta_1 + l_1\dot{\theta}_1^2\right] - m_2 sin\theta_2\left[gcos\theta_2 + l_2\dot{\theta}_2^2\right]\right. \tag{38}$$

Substituting equation (38) in (32)

$$\ddot{\theta}_1 = \frac{1}{l_1}\left[\frac{\left[F - m_1 sin\theta_1\left[gcos\theta_1 + l_1\dot{\theta}_1^2\right] - m_2 sin\theta_2\left[gcos\theta_2 + l_2\dot{\theta}_2^2\right]\right.}{(M + m_1 sin\theta_1^2 + m_2 sin\theta_2^2)}\left[\frac{cos\theta_1 - gsin\theta_1}{1}\right]\right] \tag{39}$$

Substituting equation (38) in (34)

$$\ddot{\theta}_2 = \frac{1}{l_2}\left[\frac{\left[F - m_1 sin\theta_1\left[gcos\theta_1 + l_1\dot{\theta}_1^2\right] - m_2 sin\theta_2\left[gcos\theta_2 + l_2\dot{\theta}_2^2\right]\right.}{(M + m_1 sin\theta_1^2 + m_2 sin\theta_2^2)}\left[\frac{cos\theta_2 - gsin\theta_2}{1}\right]\right] \tag{40}$$

Possible choice of state is:

$$X = \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} \tag{40}$$

Non-linear state space equation is:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dfrac{[F - m_1 sin\theta_1\left[gcos\theta_1 + l_1\dot{\theta}_1^2\right] - m_2 sin\theta_2\left[gcos\theta_2 + l_2\dot{\theta}_2^2\right]}{(M + m_1 sin\theta_1^2 + m_2 sin\theta_2^2)} \\ \dot{\theta}_1 \\ \dfrac{1}{l_1}\left[\dfrac{[F - m_1 sin\theta_1\left[gcos\theta_1 + l_1\dot{\theta}_1^2\right] - m_2 sin\theta_2\left[gcos\theta_2 + l_2\dot{\theta}_2^2\right]}{(M + m_1 sin\theta_1^2 + m_2 sin\theta_2^2)}\left[\dfrac{cos\theta_1 - gsin\theta_1}{1}\right]\right] \\ \dot{\theta}_2 \\ \dfrac{1}{l_2}\left[\dfrac{[F - m_1 sin\theta_1\left[gcos\theta_1 + l_1\dot{\theta}_1^2\right] - m_2 sin\theta_2\left[gcos\theta_2 + l_2\dot{\theta}_2^2\right]}{\left(M + m_1 sin\theta_1^2 + m_2 sin\theta_2^2\right)}\left[\dfrac{cos\theta_2 - gsin\theta_2}{1}\right]\right] \end{bmatrix} \quad (41)$$

CHAPTER - 4

LINEARIZING THE SYSTEM AROUND EQUILIBRIUM POINT AND LINEARIZED STATE SPACE REPRESENTATION

Dynamic equations derived above are non-linear and it needs to be linearized around an equilibrium point such that it can be tested for properties such as controllability and stabilizability. This equilibrium point has the property to remain at same state for the given input. The above equations can be linearized to Linear Time Invariant system.

Linearization around state variables is given by the Jacobian linearization matrix below:

J=

$$\begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial \dot{x}} & \frac{\partial F_1}{\partial \theta_1} & \frac{\partial F_1}{\partial \dot{\theta}_1} & \frac{\partial F_1}{\partial \theta_2} & \frac{\partial F_1}{\partial \dot{\theta}_2} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial \dot{x}} & \frac{\partial F_2}{\partial \theta_1} & \frac{\partial F_2}{\partial \dot{\theta}_1} & \frac{\partial F_2}{\partial \theta_2} & \frac{\partial F_2}{\partial \dot{\theta}_2} \\ \frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial \dot{x}} & \frac{\partial F_3}{\partial \theta_1} & \frac{\partial F_3}{\partial \dot{\theta}_1} & \frac{\partial F_3}{\partial \theta_2} & \frac{\partial F_3}{\partial \dot{\theta}_2} \\ \frac{\partial F_4}{\partial x} & \frac{\partial F_4}{\partial \dot{x}} & \frac{\partial F_4}{\partial \theta_1} & \frac{\partial F_4}{\partial \dot{\theta}_1} & \frac{\partial F_4}{\partial \theta_2} & \frac{\partial F_4}{\partial \dot{\theta}_2} \\ \frac{\partial F_5}{\partial x} & \frac{\partial F_5}{\partial \dot{x}} & \frac{\partial F_5}{\partial \theta_1} & \frac{\partial F_5}{\partial \dot{\theta}_1} & \frac{\partial F_5}{\partial \theta_2} & \frac{\partial F_5}{\partial \dot{\theta}_2} \\ \frac{\partial F_6}{\partial x} & \frac{\partial F_6}{\partial \dot{x}} & \frac{\partial F_6}{\partial \theta_1} & \frac{\partial F_6}{\partial \dot{\theta}_1} & \frac{\partial F_6}{\partial \theta_2} & \frac{\partial F_6}{\partial \dot{\theta}_2} \end{bmatrix} \quad (42)$$

```
% Jacobian Linearization of the Non Linear System of Dual Pendulum
% Suspended on a crane.
clc;
clear all;
close all;
syms x x_dot theta1 theta_dot_1 theta2 theta_dot_2 F M m1 m2 g l1 l2
```

```
  x_ddot = (F-(m1*sin(theta1))*(g*cos(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sin(theta2)*(g*cos(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sin(theta1)*sin(theta1)+
m2*sin(theta2)*sin(theta2));
  theta1_ddot = (x_ddot*cos(theta1)-g*sin(theta1))/l1
```

theta1_ddot =

$$-\frac{g\sin(\theta_1)+\dfrac{\cos(\theta_1)\,\left(m_1\sin(\theta_1)\,\left(l_1\,\dot{\theta}_1{}^2+g\cos(\theta_1)\right)-F+m_2\sin(\theta_2)\,\left(l_2\,\dot{\theta}_2{}^2+g\cos(\theta_2)\right)\right)}{m_1\sin(\theta_1)^2+m_2\sin(\theta_2)^2+M}}{l_1}$$

```
  theta2_ddot = (x_ddot*cos(theta2)-g*sin(theta2))/l2;



  A_intermediate = [diff(x_dot,x) diff(x_dot,x_dot) diff(x_dot,theta1) diff(x_dot,theta_dot_1)
diff(x_dot,theta2) diff(x_dot,theta_dot_2);
      diff(x_ddot,x) diff(x_ddot,x_dot) diff(x_ddot,theta1) diff(x_ddot,theta_dot_1)
diff(x_ddot,theta2) diff(x_ddot,theta_dot_2);
      diff(theta_dot_1,x) diff(theta_dot_1,x_dot) diff(theta_dot_1,theta1)
diff(theta_dot_1,theta_dot_1) diff(theta_dot_1,theta2) diff(theta_dot_1,theta_dot_2);
      diff(theta1_ddot,x) diff(theta1_ddot,x_dot) diff(theta1_ddot,theta1)
diff(theta1_ddot,theta_dot_1) diff(theta1_ddot,theta2) diff(theta1_ddot,theta_dot_2);
      diff(theta_dot_2,x) diff(theta_dot_2,x_dot) diff(theta_dot_2,theta1)
diff(theta_dot_2,theta_dot_1) diff(theta_dot_2,theta2) diff(theta_dot_2,theta_dot_2);
      diff(theta2_ddot,x) diff(theta2_ddot,x_dot) diff(theta2_ddot,theta1)
diff(theta2_ddot,theta_dot_1) diff(theta2_ddot,theta2) diff(theta2_ddot,theta_dot_2);
      ]
```

By Linearizing around equilibrium point we get, A_intermediate =

$$
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{2\,m_1 \cos(\theta_1) \sin(\theta_1)\,\sigma_2}{{\sigma_1}^2} - \frac{\sigma_4}{\sigma_1} & -\frac{2\,l_1\,m_1\,\dot{\theta}_1 \sin(\theta_1)}{\sigma_1} & \frac{2\,m_2 \cos(\theta_2) \sin(\theta_2)\,\sigma_2}{{\sigma_1}^2} - \frac{\sigma_3}{\sigma_1} & -\frac{2\,l_2\,m_2\,\dot{\theta}_2 \sin(\theta_2)}{\sigma_1} \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & -\frac{g \cos(\theta_1) - \frac{\sin(\theta_1)\,\sigma_2}{\sigma_1} + \frac{\cos(\theta_1)\,\sigma_4}{\sigma_1} - \frac{2\,m_1 \cos(\theta_1)^2 \sin(\theta_1)\,\sigma_2}{{\sigma_1}^2}}{l_1} & -\frac{2\,m_1\,\dot{\theta}_1 \cos(\theta_1) \sin(\theta_1)}{\sigma_1} & -\frac{\frac{\cos(\theta_1)\,\sigma_3}{\sigma_1} - \frac{2\,m_2 \cos(\theta_1) \cos(\theta_2) \sin(\theta_2)\,\sigma_2}{{\sigma_1}^2}}{l_1} & -\frac{2\,l_2\,m_2\,\dot{\theta}_2 \cos(\theta_1) \sin(\theta_2)}{l_1\,\sigma_1} \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & -\frac{\frac{\cos(\theta_2)\,\sigma_4}{\sigma_1} - \frac{2\,m_1 \cos(\theta_1) \cos(\theta_2) \sin(\theta_1)\,\sigma_2}{{\sigma_1}^2}}{l_2} & -\frac{2\,l_1\,m_1\,\dot{\theta}_1 \cos(\theta_2) \sin(\theta_1)}{l_2\,\sigma_1} & -\frac{g \cos(\theta_2) - \frac{\sin(\theta_2)\,\sigma_2}{\sigma_1} + \frac{\cos(\theta_2)\,\sigma_3}{\sigma_1} - \frac{2\,m_2 \cos(\theta_2)^2 \sin(\theta_2)\,\sigma_2}{{\sigma_1}^2}}{l_2} & -\frac{2\,m_2\,\dot{\theta}_2 \cos(\theta_2) \sin(\theta_2)}{\sigma_1}
\end{pmatrix}
$$

where

$$\sigma_1 = m_1 \sin(\theta_1)^2 + m_2 \sin(\theta_2)^2 + M$$

$$\sigma_2 = m_1 \sin(\theta_1)\,\sigma_6 - F + m_2 \sin(\theta_2)\,\sigma_5$$

$$\sigma_3 = m_2 \cos(\theta_2)\,\sigma_5 - g\,m_2 \sin(\theta_2)^2$$

$$\sigma_4 = m_1 \cos(\theta_1)\,\sigma_6 - g\,m_1 \sin(\theta_1)^2$$

$$\sigma_5 = l_2\,{\dot{\theta}_2}^2 + g \cos(\theta_2)$$

$$\sigma_6 = l_1\,{\dot{\theta}_1}^2 + g \cos(\theta_1)$$

```
A = subs(A_intermediate,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2],[0,0,0,0,0,0])
```

A =

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

```
B = [x_dot;x_ddot;theta_dot_1;theta1_ddot;theta_dot_2;theta2_ddot];
```

```
B = subs(B,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,F],[0,0,0,0,0,0,1])
```

B =

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

The state space representation of linearized system is,

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

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-gm_1}{M} & 0 & \frac{-gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g + \frac{gm_1}{M}}{l_1} & 0 & \frac{-gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-gm_1}{Ml_2} & 0 & -\frac{g + \frac{gm_2}{M}}{l_2} & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_1} \end{bmatrix} U(t) \qquad (43)$$

The matrix obtained by Jacobian linearization is A,

Where A and B are given by,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-gm_1}{M} & 0 & \frac{-gm_2}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g + \frac{gm_1}{M}}{l_1} & 0 & \frac{-gm_2}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-gm_1}{Ml_2} & 0 & -\frac{g + \frac{gm_2}{M}}{l_2} & 0 \end{bmatrix} \tag{44}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_1} \end{bmatrix} \tag{45}$$

CHAPTER - 5

CONTROLLABILITY

4.1 Conditions for linearized system to be controllable:

A linear state equation is called controllable on the finite time interval [t_0, t_f] if a continuous input signal U(t) can drive the system from initial state X(t_0) =X_0 to origin X(t_f)=0. A pair (A, B) is said to be controllable if the grammian of controllability W_c is invertible.

Determinant of controllability matrix:4

$$|C| = \frac{g^6 l_1^2 - 2g^6 l_1 l_2 + g^6 l_2^2}{M^6 l_1^6 l_2^6} \tag{46}$$

$$rank = ([B_k \quad AB_k \quad A^2B_k \quad \cdots \quad \cdots \quad A^{n-1}B_k\ ])_{n*nm} = n \tag{47}$$

$$\Rightarrow \frac{g^6 l_1^2 - 2g^6 l_1 l_2 + g^6 l_2^2}{M^6 l_1^6 l_2^6} \neq 0 \tag{48}$$

$$\Rightarrow g^6 l_1^2 - 2g^6 l_1 l_2 + g^6 l_2^2 \neq 0 \tag{49}$$

$$\Rightarrow (g^3 l_1 - g^3 l_2)^2 \neq 0 \tag{50}$$

$$\Rightarrow (g^3 l_1)^2 \neq g^3 l_2 \tag{51}$$

$$\Rightarrow l_1 \neq l_2 \tag{52}$$

Equation (52) is the condition for the system to be controllable.

```
%Check for Controllability
%Linearized using Jacobian Linearization
clc;
clear all;
close all;
syms x x_dot theta1 theta_dot_1 theta2 theta_dot_2 F M m1 m2 g l1 l2
```

```
  x_ddot = (F-(m1*sin(theta1))*(g*cos(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sin(theta2)*(g*cos(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sin(theta1)*sin(theta1)+
m2*sin(theta2)*sin(theta2));
  theta1_ddot = (x_ddot*cos(theta1)-g*sin(theta1))/l1;
  theta2_ddot = (x_ddot*cos(theta2)-g*sin(theta2))/l2;



  A_intermediate = [diff(x_dot,x) diff(x_dot,x_dot) diff(x_dot,theta1) diff(x_dot,theta_dot_1)
diff(x_dot,theta2) diff(x_dot,theta_dot_2);
      diff(x_ddot,x) diff(x_ddot,x_dot) diff(x_ddot,theta1) diff(x_ddot,theta_dot_1)
diff(x_ddot,theta2) diff(x_ddot,theta_dot_2);
      diff(theta_dot_1,x) diff(theta_dot_1,x_dot) diff(theta_dot_1,theta1)
diff(theta_dot_1,theta_dot_1) diff(theta_dot_1,theta2) diff(theta_dot_1,theta_dot_2);
      diff(theta1_ddot,x) diff(theta1_ddot,x_dot) diff(theta1_ddot,theta1)
diff(theta1_ddot,theta_dot_1) diff(theta1_ddot,theta2) diff(theta1_ddot,theta_dot_2);
      diff(theta_dot_2,x) diff(theta_dot_2,x_dot) diff(theta_dot_2,theta1)
diff(theta_dot_2,theta_dot_1) diff(theta_dot_2,theta2) diff(theta_dot_2,theta_dot_2);
      diff(theta2_ddot,x) diff(theta2_ddot,x_dot) diff(theta2_ddot,theta1)
diff(theta2_ddot,theta_dot_1) diff(theta2_ddot,theta2) diff(theta2_ddot,theta_dot_2);
      ];
  disp("A Matrix")
```

A Matrix

```
  A = subs(A_intermediate,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2],[0,0,0,0,0,0])
```

A =

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

```
B = [x_dot;x_ddot;theta_dot_1;theta1_ddot;theta_dot_2;theta2_ddot];
disp("B Matrix")
```

B Matrix

```
B = subs(B,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,F],[0,0,0,0,0,0,1])
```

B =

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

```
C = [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
disp(['Rank of Controllability Matrix is ',num2str(rank(C))]);
```

Rank of Controllability Matrix is 6

```
%For the system to be controllable, it should be non singular. That means
%the determinant of the controllability matrix should not be equal to zero.
disp("For the system to be controllable, the determinant of the controllability matrix should be non singular")
```

For the system to be controllable, the determinant of the controllability matrix should be non singular

```
det(C) %!=0
```

ans = |C|=

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

CHAPTER - 6

LQR CONTROLLER

5.1 Test for controllability:

Controllability is the existence of a control input(U) such that in the absence of disturbances the systems state changes from initial state to desired state at any time instant. Controllability of a pair (A, B_k) can be verified by finding the rank of controllability matrix which is given as

$$rank = ([B_k \quad AB_k \quad A^2B_k \quad \cdots \quad \cdots \quad A^{n-1}B_k\ ])_{n*nm} = n$$

5.2 Linear Quadratic Regulator (LQR) control

If the pair (A, B_k) is stabilizable, the cost function of linear Quadratic Regulator is given by

$$J(K, X_0) = \int_0^{\propto} X^T(t)QX(t) + U_k(t)RU_K\, dt$$

The cost function is minimized using the control input U and a regulating matrix R and a matrix Q that imposes cost which is applied to linear system.

Where the gain can be found from the equation $-R^{-1}B^TP$, Where P is a solution of Riccati equation $A^TP + PA - PBR^{-1}B^TP = -Q$. Q is positive definite and symmetric.

If Q >> R, then the states converge to desired state faster than the system with Q << R.

```
%LQR CONTROLLER DESIGN for Linear and Non Linear System
%Linearized Model is from Jacobian Linearization.


clc;
clear all;
close all;
syms x x_dot theta1 theta_dot_1 theta2 theta_dot_2 F M m1 m2 g l1 l2
```

```
x_ddot = (F-(m1*sin(theta1))*(g*cos(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sin(theta2)*(g*cos(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sin(theta1)*sin(theta1)+
m2*sin(theta2)*sin(theta2));
theta1_ddot = (x_ddot*cos(theta1)-g*sin(theta1))/l1;
theta2_ddot = (x_ddot*cos(theta2)-g*sin(theta2))/l2;



A_intermediate = [diff(x_dot,x) diff(x_dot,x_dot) diff(x_dot,theta1) diff(x_dot,theta_dot_1)
diff(x_dot,theta2) diff(x_dot,theta_dot_2);
    diff(x_ddot,x) diff(x_ddot,x_dot) diff(x_ddot,theta1) diff(x_ddot,theta_dot_1)
diff(x_ddot,theta2) diff(x_ddot,theta_dot_2);
    diff(theta_dot_1,x) diff(theta_dot_1,x_dot) diff(theta_dot_1,theta1)
diff(theta_dot_1,theta_dot_1) diff(theta_dot_1,theta2) diff(theta_dot_1,theta_dot_2);
    diff(theta1_ddot,x) diff(theta1_ddot,x_dot) diff(theta1_ddot,theta1)
diff(theta1_ddot,theta_dot_1) diff(theta1_ddot,theta2) diff(theta1_ddot,theta_dot_2);
    diff(theta_dot_2,x) diff(theta_dot_2,x_dot) diff(theta_dot_2,theta1)
diff(theta_dot_2,theta_dot_1) diff(theta_dot_2,theta2) diff(theta_dot_2,theta_dot_2);
    diff(theta2_ddot,x) diff(theta2_ddot,x_dot) diff(theta2_ddot,theta1)
diff(theta2_ddot,theta_dot_1) diff(theta2_ddot,theta2) diff(theta2_ddot,theta_dot_2);];
A = subs(A_intermediate,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2],[0,0,0,0,0,0]);
B = [x_dot;x_ddot;theta_dot_1;theta1_ddot;theta_dot_2;theta2_ddot];
B = subs(B,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,F],[0,0,0,0,0,0,1]);

A1 = subs(A,[M,m1,m2,l1,l2,g],[1000,100,100,20,10,9.8]);
% A = vpa(A)Type equation here.
%Initial Eigen Values of the Linearized System
eigen_values = vpa(eig(A1))
```

eigen_values =

$$\begin{pmatrix} 0 \\ 0 \\ 1.0424809393673408583055086703293\,\mathrm{i} \\ 0.72817133358556940178485707773643\,\mathrm{i} \\ -1.0424809393673408583055086703293\,\mathrm{i} \\ -0.72817133358556940178485707773643\,\mathrm{i} \end{pmatrix}$$

```
%Calculating the initial state response
C = eye(6)
```

C = 6×6

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

```
D = 0;
%Taking step input = F=1;
% vpa(A1);
New_A = [0 1 0 0 0 0; 0 0 -0.98 0 -0.98 0; 0 0 0 1 0 0;0 0 -0.539 0 -0.049 0;0 0 0 0 0 1;0 0 -
0.098 0 -1.078 0];
% B = subs(B,[F,M,l1,l2],[1,1000,20,10]);
vpa(B);
New_B = [0;0.001;0;0.00005;0;0.0001];
initial_state = [3,0.3,20,1,10,2];
% Initial State Response
state_space = ss(New_A,New_B,C,D);
figure
initial(state_space,initial_state);
```

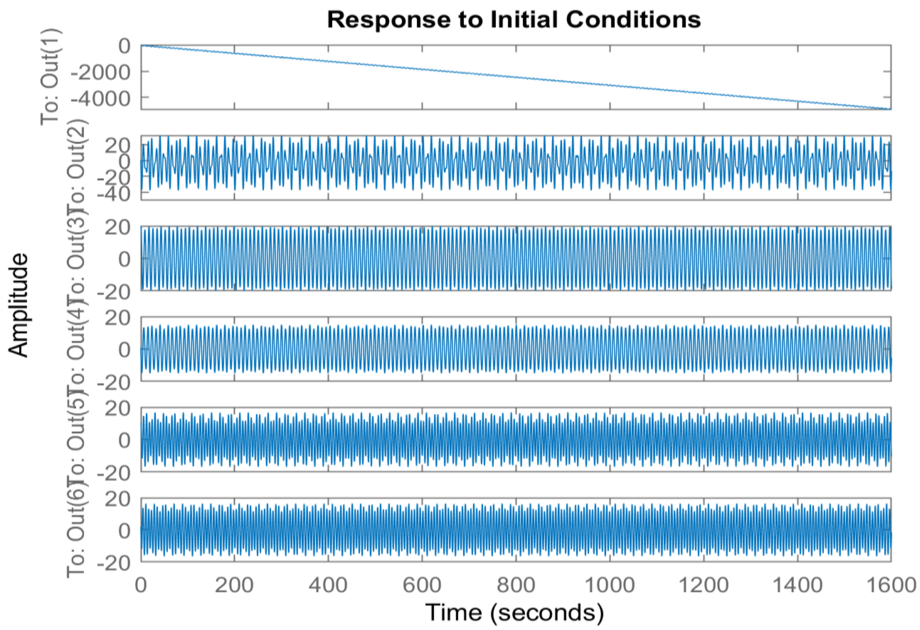

Figure 2 Response to initial conditions

```
Q = [1 0 0 0 0 0;0 1 0 0 0 0;0 0 0.5 0 0 0;0 0 0 0.5 0 0;0 0 0 0 0.2 0;0 0 0 0 0 0.2];
R = 10;
% The below for loop is used for calculating the response of the system for
% different Q values and Different R Values to minimize the cost function

% for mult = 1:100:2000
%     figure
%     [K,P,Poles] = lqr(New_A,New_B,Q*2000,(R/mult)*100);
%     state_space = ss(New_A-(New_B*K),zeros(6,1),C,D)
%     initial(state_space,initial_state)
% end

figure
[K,P,Poles] = lqr(New_A,New_B,Q*20000,R/10000);
eig(New_A-(New_B*K))
```

ans = 6×1 complex

-4.3249 + 0.0000i

-1.0325 + 0.0000i

-0.0232 + 0.9902i

-0.0232 - 0.9902i

-0.0127 + 0.7000i

-0.0127 - 0.7000i

```
state_space = ss(New_A-(New_B*K),zeros(6,1),C,D);
initial(state_space,initial_state);
title('LQR Controller Response to Initial Conditions')
xlabel('Time')
ylabel('Output States')
```

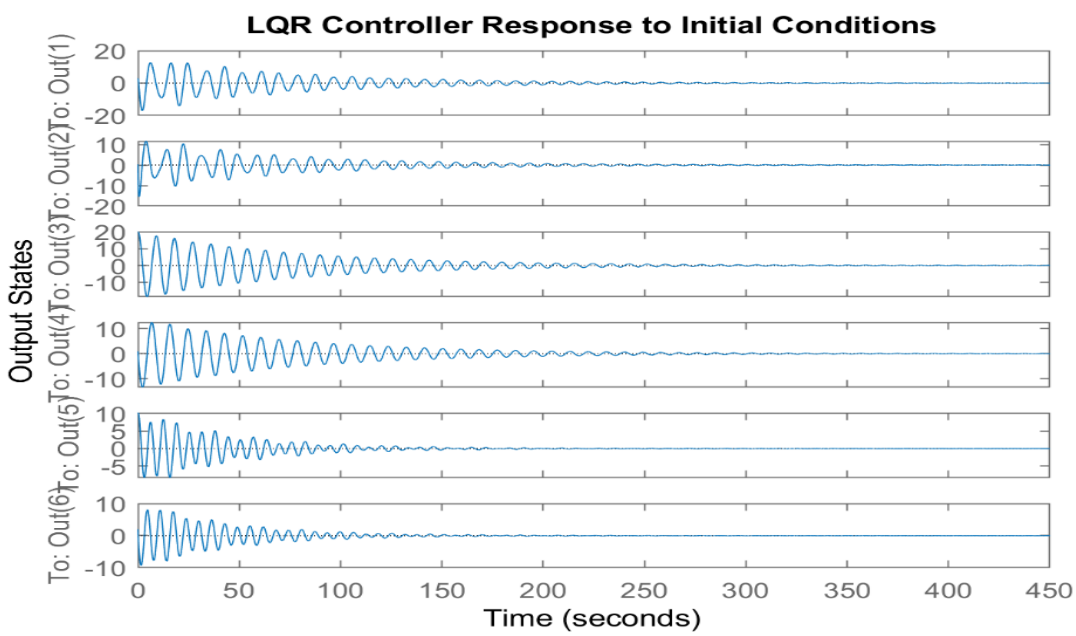

Figure 3 LQR controller response to initial conditions

```
disp("Eigen Values of the system after applying LQR Controller is: ")
```

Eigen Values of the system after applying LQR Controller is:

```
Poles
```

Poles = 6×1 complex

```
  -0.0127 + 0.7000i
  -0.0127 - 0.7000i
  -0.0232 + 0.9902i
  -0.0232 - 0.9902i
  -1.0325 + 0.0000i
  -4.3249 + 0.0000i
```

```
disp("Eigen Values of the P Matrix is: ")
```

Eigen Values of the P Matrix is:

```
eig(P)
```

ans = 6×1

$10^6 \times$

```
0.0043
0.0248
0.1927
0.2037
0.6297
1.3105
```

```
%Non Linear Model:
simulation_time = 0:1:2000;
[time,out] = ode45(@ode45_callback,simulation_time,initial_state);
figure
plot(time,out)
title('LQR response to Non-Linear System')
xlabel('Time')
```

```
ylabel('State outputs')
```

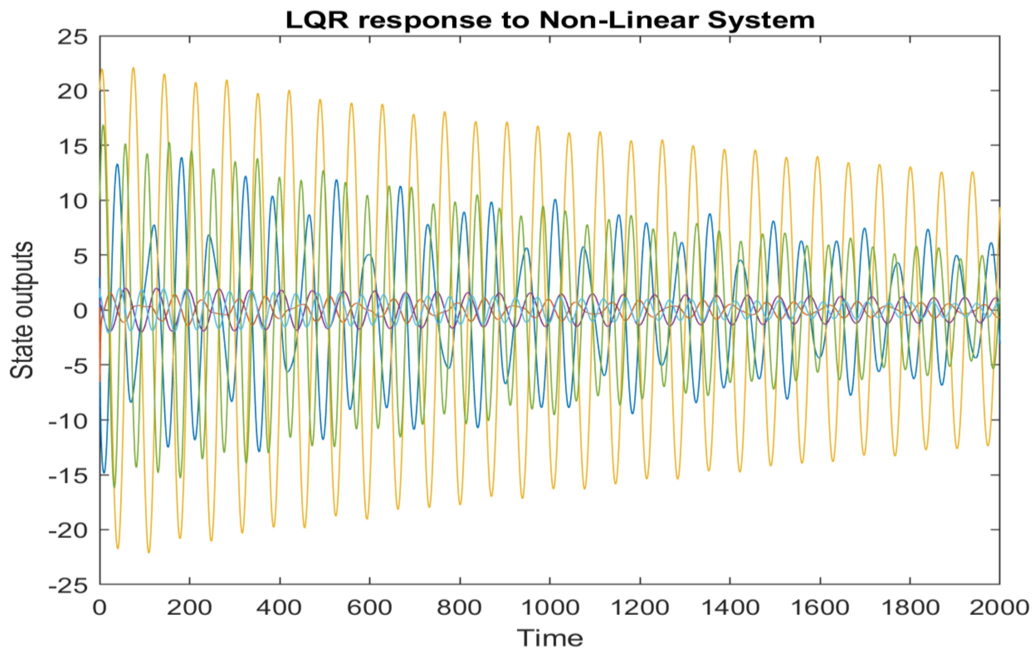

Figure 4 LQR response to non-linear system

5.2 LQR Controller for Non-Linear Model – Callback function

```
function output = ode45_callback(t,x)
[A,B,Q,R] = AB();
[K,~,~] = lqr(A,B,Q,R);
%We are giving feedback U=-KX where X is the initial state Now
F = -K*x;
output = zeros(6,1);
M = 1000;
m1 = 100;
m2 = 100;
theta1 = x(3);
theta2 = x(5);
theta_dot_1 = x(4);
theta_dot_2 = x(6);
```

```
l1 = 20;
l2 = 10;
g = 9.8;

x_ddot = (F-(m1*sind(theta1))*(g*cosd(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sind(theta2)*(g*cosd(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sind(theta1)*sind(thet
a1)+m2*sind(theta2)*sind(theta2));
theta1_ddot = (x_ddot*cosd(theta1)-g*sind(theta1))/l1;
theta2_ddot = (x_ddot*cosd(theta2)-g*sind(theta2))/l2;
%look at the non linear system representation in Chapter X
output(1) = x(2); %because initial state has x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2
output(2) = x_ddot;
output(3) = x(4);
output(4) = theta1_ddot;
output(5) = x(6);
output(6) = theta2_ddot;
end
```

CHAPTER - 7

CHECK FOR OBSERVABILITY FOR GIVEN STATE VARIABLES

The linear state equation is said to be observable on the interval [t_0, t_f] if any initial state

$x(t_0) = x_0$ is uniquely determined by the corresponding response y(t), where t∈ [[t_0, t_f]

```
% Observability
% Jacobian Linearization of the Non Linear System of Dual Pendulum
% Suspended on a crane.
clc;
clear all;
close all;
syms x x_dot theta1 theta_dot_1 theta2 theta_dot_2 F M m1 m2 g l1 l2


x_ddot = (F-(m1*sin(theta1))*(g*cos(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sin(theta2)*(g*cos(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sin(theta1)*sin(theta1)+
m2*sin(theta2)*sin(theta2));
theta1_ddot = (x_ddot*cos(theta1)-g*sin(theta1))/l1;
theta2_ddot = (x_ddot*cos(theta2)-g*sin(theta2))/l2;


A_intermediate = [diff(x_dot,x) diff(x_dot,x_dot) diff(x_dot,theta1) diff(x_dot,theta_dot_1)
diff(x_dot,theta2) diff(x_dot,theta_dot_2);
    diff(x_ddot,x) diff(x_ddot,x_dot) diff(x_ddot,theta1) diff(x_ddot,theta_dot_1)
diff(x_ddot,theta2) diff(x_ddot,theta_dot_2);
    diff(theta_dot_1,x) diff(theta_dot_1,x_dot) diff(theta_dot_1,theta1)
diff(theta_dot_1,theta_dot_1) diff(theta_dot_1,theta2) diff(theta_dot_1,theta_dot_2);
    diff(theta1_ddot,x) diff(theta1_ddot,x_dot) diff(theta1_ddot,theta1)
diff(theta1_ddot,theta_dot_1) diff(theta1_ddot,theta2) diff(theta1_ddot,theta_dot_2);
```

```
    diff(theta_dot_2,x) diff(theta_dot_2,x_dot) diff(theta_dot_2,theta1)
diff(theta_dot_2,theta_dot_1) diff(theta_dot_2,theta2) diff(theta_dot_2,theta_dot_2);
    diff(theta2_ddot,x) diff(theta2_ddot,x_dot) diff(theta2_ddot,theta1)
diff(theta2_ddot,theta_dot_1) diff(theta2_ddot,theta2) diff(theta2_ddot,theta_dot_2);
    ];
```

```
A = subs(A_intermediate,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2],[0,0,0,0,0,0])
```

A =

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

```
B = [x_dot;x_ddot;theta_dot_1;theta1_ddot;theta_dot_2;theta2_ddot];
B = subs(B,[x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,F],[0,0,0,0,0,0,1])
```

B =

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

```
% From the given problem statement, the output vectors are
% x,(theta1,theta2),(x,theta2),(x,theta1,theta2)
```

```
C_1 = [1 0 0 0 0 0]; %because Y = CX + DU. We take C matrix such that it accounts for the state variables
C_2 = [0 0 1 0 0 0;
       0 0 0 0 1 0];
C_3 = [1 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];
% We know that the system is observable when rank[C' A'C'..... A'^nC']=n

O1 = [C_1' A'*C_1' A'*A'*C_1' A'*A'*A'*C_1' A'*A'*A'*A'*C_1' A'*A'*A'*A'*A'*C_1'];
O2 = [C_2' A'*C_2' A'*A'*C_2' A'*A'*A'*C_2' A'*A'*A'*A'*C_2' A'*A'*A'*A'*A'*C_2'];
O3 = [C_3' A'*C_3' A'*A'*C_3' A'*A'*A'*C_3' A'*A'*A'*A'*C_3' A'*A'*A'*A'*A'*C_3'];
O4 = [C_4' A'*C_4' A'*A'*C_4' A'*A'*A'*C_4' A'*A'*A'*A'*C_4' A'*A'*A'*A'*A'*C_4'];
display(['The rank of observability matrix for the choice of state x = ',num2str(rank(O1))])
```

The rank of observability matrix for the choice of state x = 6

```
display(['The rank of observability matrix for the choice of state (theta1,theta2) = ',num2str(rank(O2))])
```

The rank of observability matrix for the choice of state (theta1, theta2) = 4

```
display(['The rank of observability matrix for the choice of state (x,theta2) = ',num2str(rank(O3))])
```

The rank of observability matrix for the choice of state (x, theta2) = 6

```
display(['The rank of observability matrix for the choice of state (x,theta1,theta2) = ',num2str(rank(O4))])
```

The rank of observability matrix for the choice of state (x, theta1, theta2) = 6

CHAPTER – 8

LUENBERGER OBSERVER

Luenberger observer is given by following state space representation

$$\dot{\hat{\vec{X}}}(t) = \mathbf{A}\hat{\vec{X}}(t) + \mathbf{B}_K\vec{U}_K(t) + \mathbf{L}(\vec{Y}(t) - \mathbf{C}\hat{\vec{X}}(t)), \quad \vec{X}(0) = 0$$

Where L=observer gain matrix and $Y(t) - CX(t) = correction\ term$

The Luenberger observer gain matrix can be obtained by placing the poles of A^T, C^T. The poles are generally 3 times larger than the present position of poles. Here, we have taken 10 times larger poles. The gain matrices for the observers will be high in our case.

7.1 Luenberger Observer for Linear Model

```
function [C_1,C_3,C_4] = get_C_matrices()
C_1 = [1 0 0 0 0 0];
C_3 = [1 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];
End
```

```
function [A,B,Q,R]=AB()
A = [0 1 0 0 0 0; 0 0 -0.98 0 -0.98 0; 0 0 0 1 0 0;0 0 -0.539 0 -0.049 0;0 0 0 0 0 1;0 0 -0.098 0 -
1.078 0];
B = [0;0.001;0;0.00005;0;0.0001];
R=0.001;
Q = [1 0 0 0 0 0;0 1 0 0 0 0;0 0 0.5 0 0 0;0 0 0 0.5 0 0;0 0 0 0 0.2 0;0 0 0 0 0 0.2];
Q=Q*20000;
```

```
end
```

```
%Luenberger Observer for Linear System
clc;
clear all;
close all;
[A,B,Q,R] = AB();
%From the observability check, we got that only C_1, C_3, C_4 are
%observable.

[C_1,C_3,C_4] = get_C_matrices();

eig(A)
```

ans = 6×1 complex

```
0.0000 + 0.0000i
0.0000 + 0.0000i
-0.0000 + 0.7282i
-0.0000 - 0.7282i
0.0000 + 1.0425i
0.0000 - 1.0425i
```

```
req_poles1 = [-10;-20;-30;-40;-50;-60];
req_poles2 = [-10;-20;-30;-40;-50;-50];
req_poles3 = [-10;-20;-30;-50;-50;-50];
% We get Luenberger Observer matrix for each C when we try place the poles at the required
position.
% We also know that placing poles for (A,C)' will give same
% poles for (A,C)
rank(C_1)
```

ans = 1

```
rank(C_3)
```

ans = 2

```
rank(C_4)
```

ans = 3

```
% We can place poles only with multiplicity of rank(C) matrices. Otherwise,
% we cannot place the poles at the required positions. Generally, the poles
% of the estimate (observer) should go faster than the system. Hence, we
% choose multiplicity of 10 for the pole placement.
Luenberger1 = place(A',C_1',req_poles1);
L1 = Luenberger1'
```

L1 = 6×1

$10^9 \times$

0.0000
0.0000
-0.3666
-1.4828
0.3658
1.4663

```
Luenberger3 = place(A',C_3',req_poles2);
L3 = Luenberger3'
```

L3 = 6×2

$10^5 \times$

0.0012 -0.0000
0.0491 -0.0004

-0.7979	0.0189
-4.0854	0.2428
0.0000	0.0008
0.0033	0.0149

```
Luenberger4 = place(A',C_4',req_poles3);
L4 = Luenberger4'
```

L4 = 6×3

$10^3 \times$

0.0798	-0.0019	0.0001
1.4908	-0.0959	0.0062
-0.0019	0.0602	-0.0000
-0.0949	0.5086	-0.0006
0.0001	-0.0000	0.0700
0.0072	-0.0007	0.9990

```
[K,P,Poles] = lqr(A,B,Q,R);
%Defining the state-space matrices for Linearized system with observer
%As mentioned in the class notes, The closed loop system has [x;x_hat] =
%[Ak BK;0 A-LC] Only difference is that we took feedback as -KX insted of
%KX
%X_hat has 6 state estimates. Hence each of the matrix element has 6 columns, 6 rows
AL1 = [(A-B*K) B*K;zeros(size(A)) (A-L1*C_1)];
% size(AL1)
BL1 = [B;zeros(size(B))];
%Because in the output we are observing the required state variables.
CL1 = [C_1 zeros(size(C_1))];
DL1 = 0;

AL3 = [(A-B*K) B*K;zeros(size(A)) (A-L3*C_3)];
```

```
% size(AL3)
BL3 = [B;zeros(size(B))];
%Because in the output we are observing the required state variables.
CL3 = [C_3 zeros(size(C_3))];
DL3 = 0;
AL4 = [(A-B*K) B*K;zeros(size(A)) (A-L4*C_4)];
% size(AL4)
BL4 = [B;zeros(size(B))];
%Because in the output we are observing the required state variables.
CL4 = [C_4 zeros(size(C_4))];
DL4 = 0;
initial_state = [3,0.3,20,1,10,2,0,0,0,0,0,0]; %last 6 colums specify the initial state of the
observer estimate which is zero
statespace1 = ss(AL1,BL1,CL1,DL1);
statespace3 = ss(AL3,BL3,CL3,DL3);
statespace4 = ss(AL4,BL4,CL4,DL4);
figure
step(statespace1)
title('Step Response for Observer 1')
xlabel('Time')
ylabel('State Variables output')
```

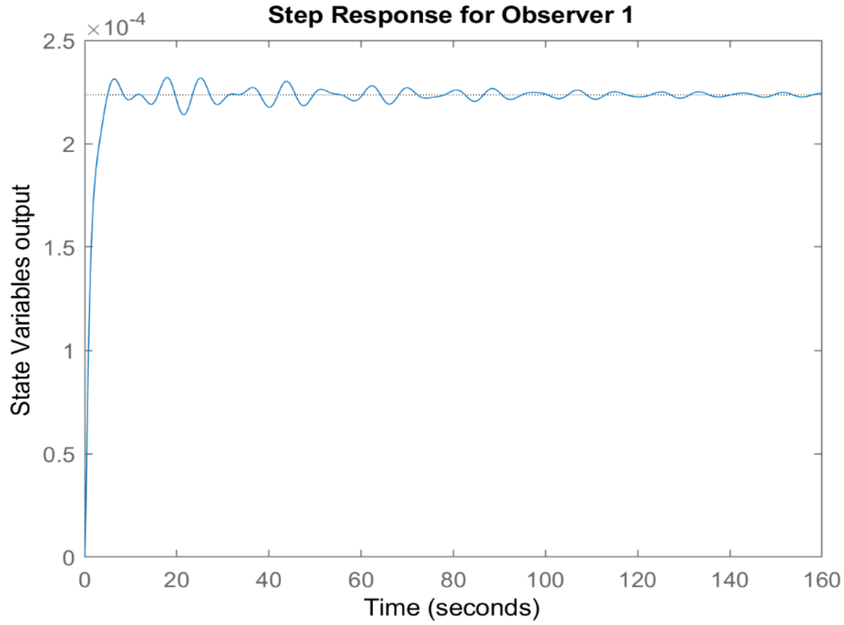

Figure 5 Step response for observer 1

```
figure
initial(statespace1,initial_state)
title('Initial Response for Observer 1')
xlabel('Time')
ylabel('State Variables output')
```

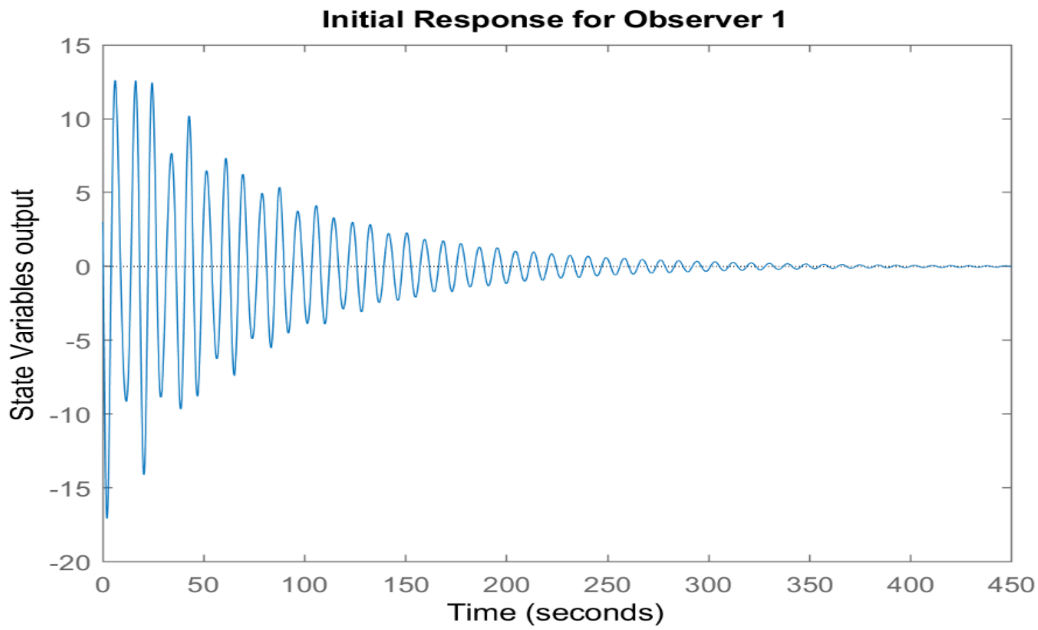

Figure 6 Initial response for observer 1

```
figure
step(statespace3)
title('Step Response for Observer 3')
xlabel('Time')
ylabel('State Variables output')
```

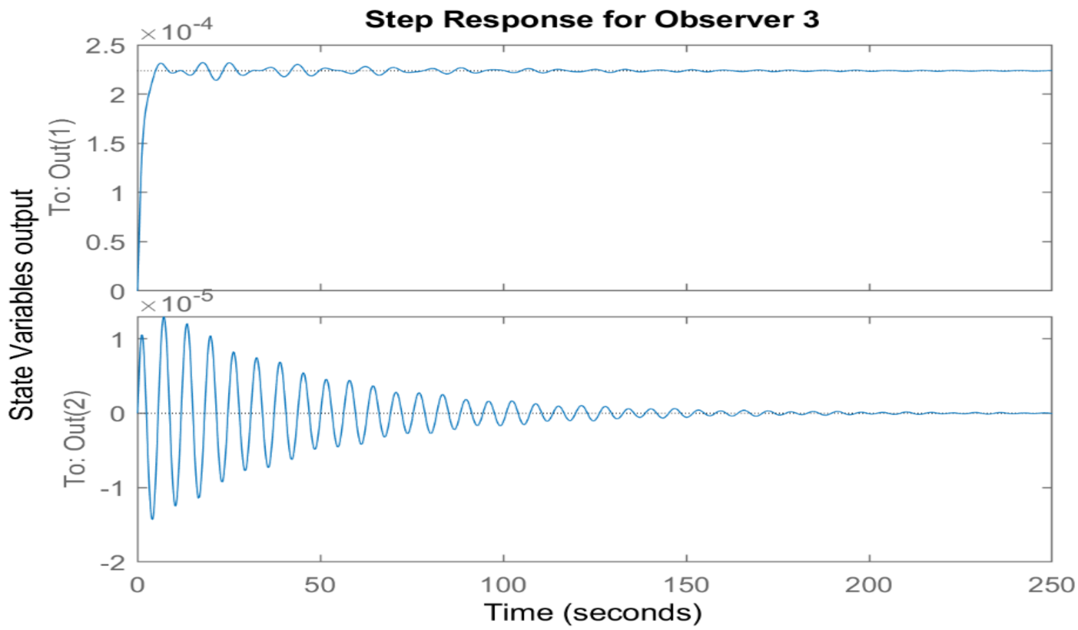

Figure 7 Step response for observer 3

```
figure
initial(statespace3,initial_state)
title('Initial Response for Observer 3')
xlabel('Time')
ylabel('State Variables output')
```

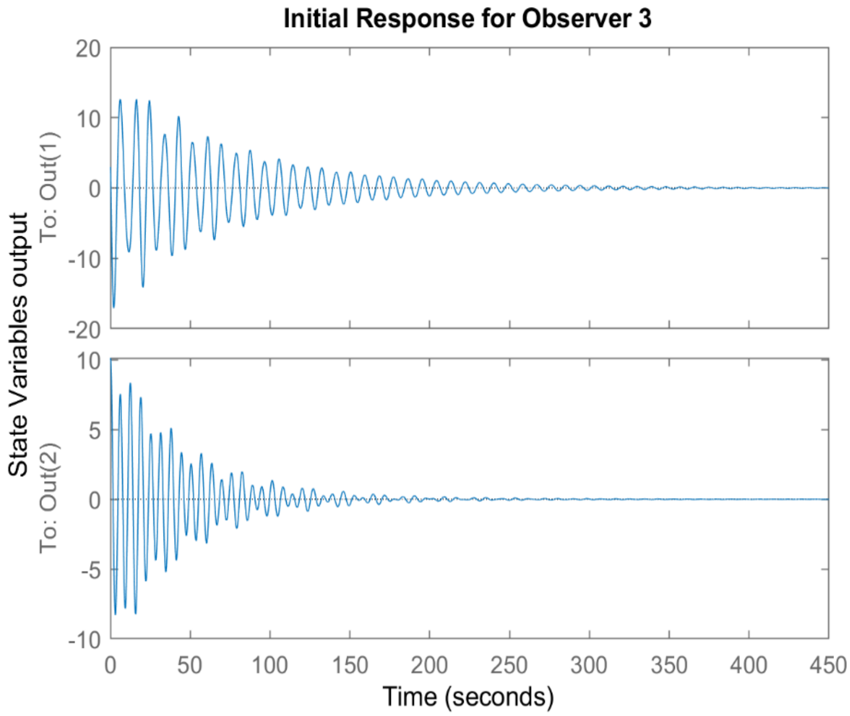

Figure 8 Initial response for observer 3

```
figure
step(statespace4)
title('Step Response for Observer 4')
xlabel('Time')
ylabel('State Variables output')
```

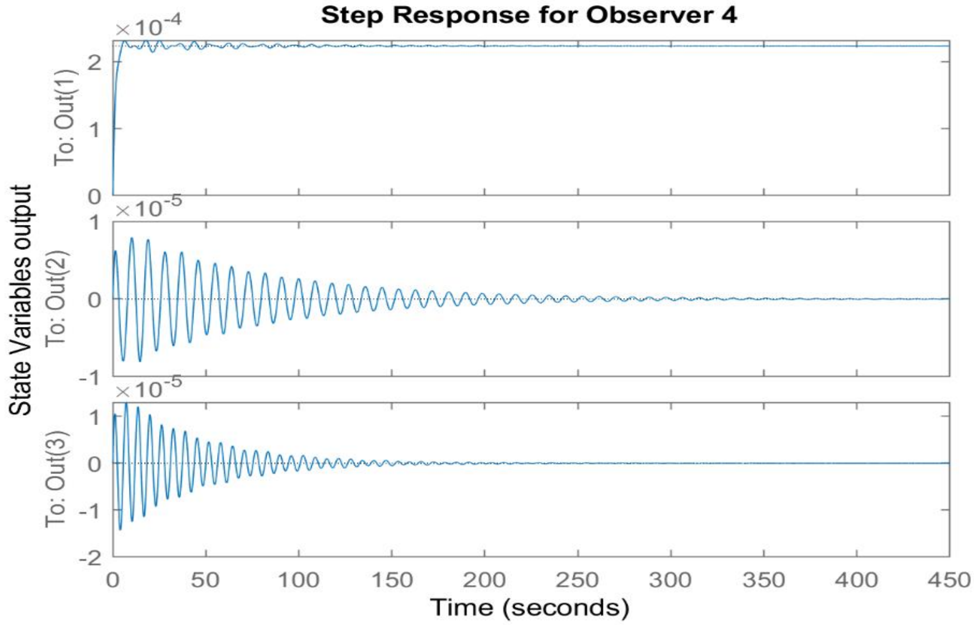

Figure 9 Step response for observer 4

```
figure
initial(statespace4,initial_state)
title('Initial Response for Observer 4')
xlabel('Time')
ylabel('State Variables output')
```

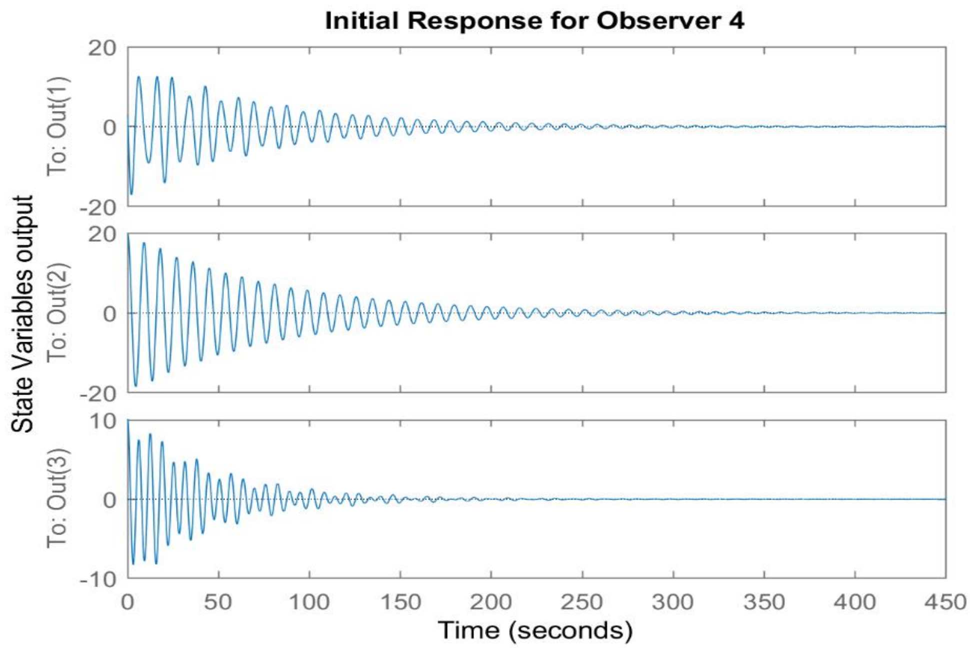

Figure 10 Initial response for observer 4

```
%Luenberger Observer for Non Linear System Model:
%Note that we are plotting the estimates. We are not not plotting the
%error

simulation_time = 0:1:1000;
[time,out] = ode45(@ode45_callback_luenberger_1,simulation_time,initial_state);
figure
plot(time,out)
title('Luenberger Observer1 for Non Linear System')
xlabel('Time')
ylabel('Estimates and State outputs')
```

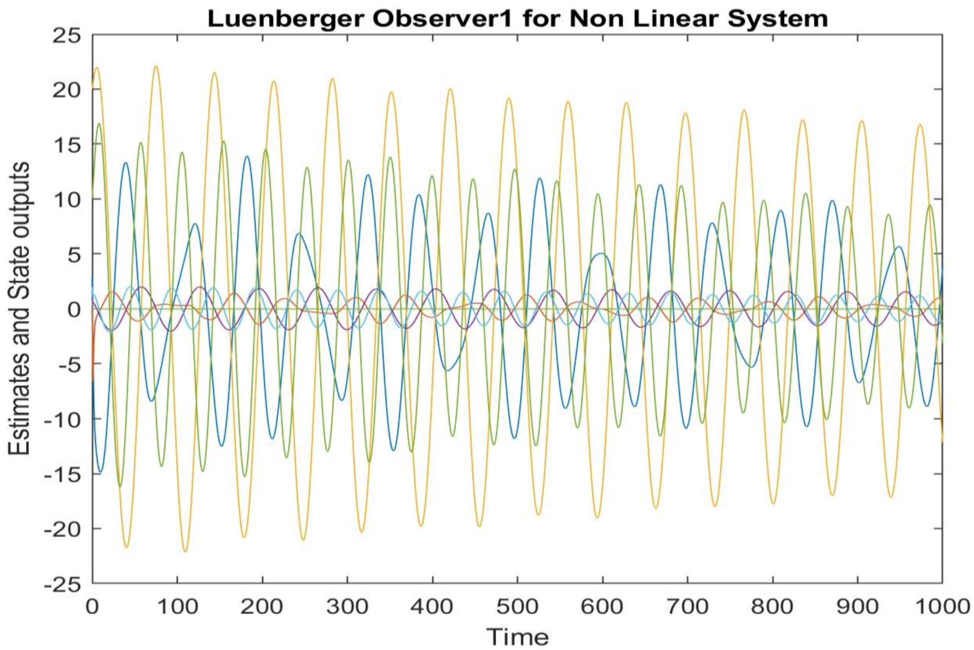

Figure 11 Luenberger observer 1 for non-linear system

```
[time,out] = ode45(@ode45_callback_luenberger_2,simulation_time,initial_state);
figure
plot(time,out)
title('Luenberger Observer3 for Non Linear System')
xlabel('Time')
ylabel('Estimates and State outputs')
```

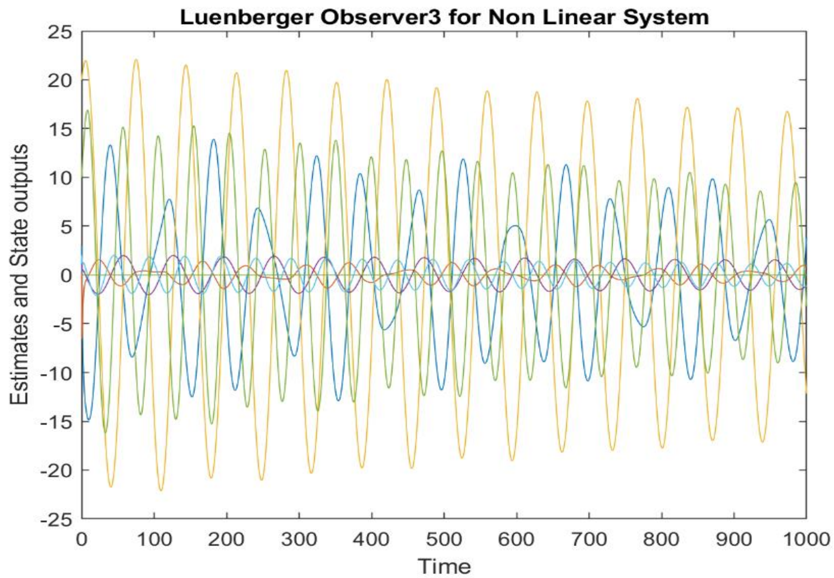

Figure 12 Luenberger observer 3 for nonlinear system

```
figure
[time,out] = ode45(@ode45_callback_luenberger_3,simulation_time,initial_state);
plot(time,out)
title('Luenberger Observer4 for Non Linear System')
xlabel('Time')
ylabel('Estimates and State outputs')
```

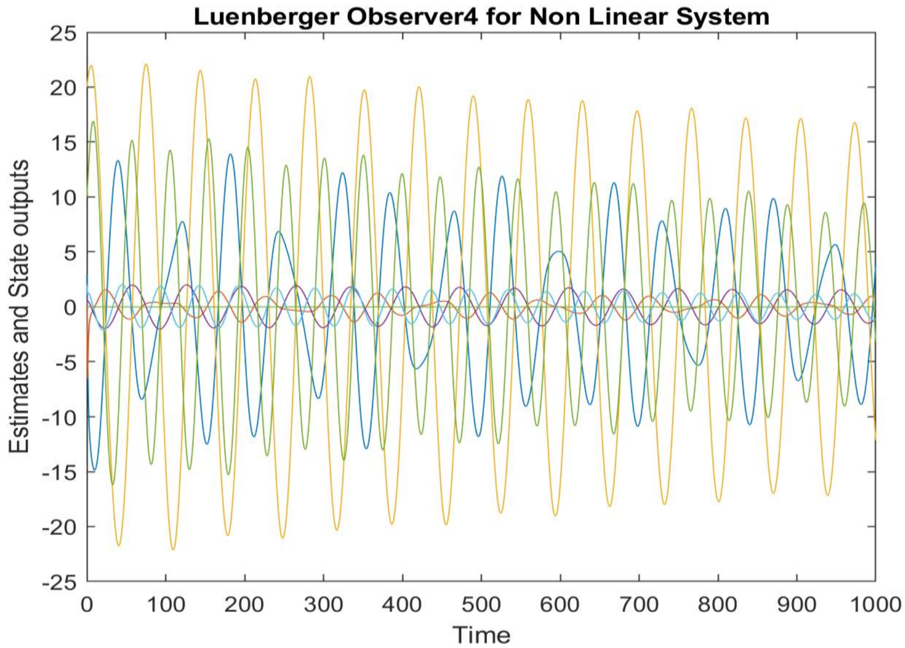

Figure 13 Luenberger observer 4 for non-linear system

7.2 Luenberger Observer 1 for Non-Linear Model

```
function output = ode45_callback_luenberger_1(t,x)
M = 1000;
m1 = 100;
m2 = 100;
theta1 = x(3);
theta2 = x(5);
theta_dot_1 = x(4);
theta_dot_2 = x(6);
l1 = 20;
l2 = 10;
g = 9.8;
[A,B,Q,R] = AB();
[C_1,~,~] = get_C_matrices();
[K,~,~] = lqr(A,B,Q,R);
F= -K*x(1:6);
```

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

  x_ddot = (F-(m1*sind(theta1))*(g*cosd(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sind(theta2)*(g*cosd(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sind(theta1)*sind(thet
a1)+m2*sind(theta2)*sind(theta2));
  theta1_ddot = (x_ddot*cosd(theta1)-g*sind(theta1))/l1;
  theta2_ddot = (x_ddot*cosd(theta2)-g*sind(theta2))/l2;



  req_poles1 = [-10;-20;-30;-40;-50;-60];
  Luenberger1 = place(A',C_1',req_poles1);
  L1 = Luenberger1';

  est = (A-L1*C_1)*x(7:12);

  output(1) = x(2); %because initial state has
x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,Estimation states (6)
  output(2) = x_ddot;
  output(3) = x(4);
  output(4) = theta1_ddot;
  output(5) = x(6);
  output(6) = theta2_ddot;
  output(7) = est(1);
  output(8) = est(2);
  output(9) = est(3);
  output(10) = est(4);
  output(11) = est(5);
  output(12) = est(6);
  end
```

7.3 Luenberger Observer 3 for Non-Linear Model

```
function output = ode45_callback_luenberger_2(t,x)
M = 1000;
m1 = 100;
m2 = 100;
theta1 = x(3);
theta2 = x(5);
theta_dot_1 = x(4);
theta_dot_2 = x(6);
l1 = 20;
l2 = 10;
g = 9.8;
[A,B,Q,R] = AB();
[~,C_3,~] = get_C_matrices();
[K,~,~] = lqr(A,B,Q,R);
F= -K*x(1:6);

output = zeros(12,1);

x_ddot = (F-(m1*sind(theta1))*(g*cosd(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sind(theta2)*(g*cosd(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sind(theta1)*sind(thet
a1)+m2*sind(theta2)*sind(theta2));
theta1_ddot = (x_ddot*cosd(theta1)-g*sind(theta1))/l1;
theta2_ddot = (x_ddot*cosd(theta2)-g*sind(theta2))/l2;



req_poles2 = [-10;-20;-30;-40;-50;-50];
Luenberger3 = place(A',C_3',req_poles2);
L3 = Luenberger3';
```

```
est = (A-L3*C_3)*x(7:12);

output(1) = x(2); %because initial state has
x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,Estimation states (6)
output(2) = x_ddot;
output(3) = x(4);
output(4) = theta1_ddot;
output(5) = x(6);
output(6) = theta2_ddot;
output(7) = est(1);
output(8) = est(2);
output(9) = est(3);
output(10) = est(4);
output(11) = est(5);
output(12) = est(6);
end
```

7.4 Luenberger Observer 4 for Non-Linear Model

```
function output = ode45_callback_luenberger_3(t,x)
M = 1000;
m1 = 100;
m2 = 100;
theta1 = x(3);
theta2 = x(5);
theta_dot_1 = x(4);
theta_dot_2 = x(6);
l1 = 20;
l2 = 10;
g = 9.8;
[A,B,Q,R] = AB();
[~,~,C_4] = get_C_matrices();
```

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

  output = zeros(12,1);

  x_ddot = (F-(m1*sind(theta1))*(g*cosd(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sind(theta2)*(g*cosd(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sind(theta1)*sind(thet
a1)+m2*sind(theta2)*sind(theta2));
  theta1_ddot = (x_ddot*cosd(theta1)-g*sind(theta1))/l1;
  theta2_ddot = (x_ddot*cosd(theta2)-g*sind(theta2))/l2;
  req_poles3 = [-10;-20;-30;-50;-50;-50];
  Luenberger4 = place(A',C_4',req_poles3);
  L4 = Luenberger4';

  est = (A-L4*C_4)*x(7:12);

  output(1) = x(2); %because initial state has
x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,Estimation states (6)
  output(2) = x_ddot;
  output(3) = x(4);
  output(4) = theta1_ddot;
  output(5) = x(6);
  output(6) = theta2_ddot;
  output(7) = est(1);
  output(8) = est(2);
  output(9) = est(3);
  output(10) = est(4);
  output(11) = est(5);
  output(12) = est(6);
  end
```

CHAPTER – 9

LQG CONTROLLER

9.1 LQG Controller for Linear Model

The LQG controller is a combination of a Kalman filter with a linear quadratic regulator (LQR). According to the separation principle the state estimator and state feedback can be designed independently. The structure of the optimal solution is given by standard output feedback configuration with the Luenberger observer with optimal K and L are computed separately using LQR and Kalman-Bucy method. The LQG method in MATLAB takes the weighted process noise and measurement noise as inputs and outputs the required state space form of the system. Note that, every-time we run the LQG controller code, the response of the system is different as the noise terms differ due to the white gaussian noise. The wgn function requires communication toolbox.

```
% LQG For Linear Model
clc;
clear all;
close all;
[A,B,Q,R] = AB();
[C_1,C_3,C_4] = get_C_matrices();
D=0;
QXU = eye(7);
% Generate noise
w = wgn(6,1,5);
v = wgn(1,1,5);
QWV = [w;v]*[w' v'];
%Since we were asked to take only the smallest output vector, we use the
% C_1 which is having the lowest order in the observable states.
%Note that as the noise is white gaussian random noise, the output may
%change everytime we run the code.
LQG_state_space = lqg(ss(A,B,C_1,D),QXU,QWV)
```

LQG_state_space =

A =

	x1_e	x2_e	x3_e	x4_e	x5_e	x6_e
x1_e	-2.549	1	0	0	0	0
x2_e	-2.96	-0.049	-0.9785	0.07827	-0.9792	0.03919
x3_e	0.5448	0	0	1	0	0
x4_e	-0.7294	-0.00245	-0.5389	0.003913	-0.04896	0.001959
x5_e	0.783	0	0	0	0	1
x6_e	-0.5853	-0.0049	-0.09785	0.007827	-1.078	0.003919

B =

	y1
x1_e	2.549
x2_e	2.959
x3_e	-0.5448
x4_e	0.7293
x5_e	-0.783
x6_e	0.5852

C =

	x1_e	x2_e	x3_e	x4_e	x5_e	x6_e
u1	-1	-49	1.532	78.27	0.7671	39.19

D =

y1

u1 0

Input groups:

Name Channels

Measurement 1

Output groups:

Name Channels

Controls 1

Continuous-time state-space model.

```
initial_state = [3,0.3,20,1,10,2];
figure
initial(LQG_state_space,initial_state)
title('LQG Controller for Linear Model')
xlabel('Time')
ylabel('State Output x')
```

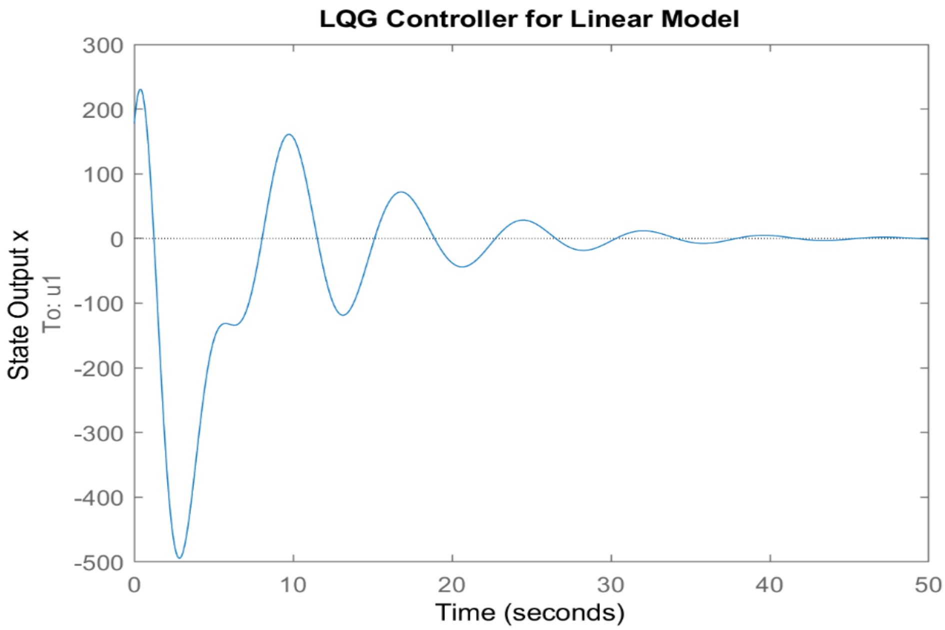

Figure 14 LQG controller for linear model

```
%Uncomment Below lines to get LQG controller for all states
% w = wgn(11,1,5);
% v = wgn(1,1,5);
% QWV = [w;v]*[w' v'];
% LQG_state_space = lqg(ss(A,B,eye(6),D),QXU,QWV)
% initial_state = [3,0.3,20,1,10,2];
% figure
% initial(LQG_state_space,initial_state)
% title('LQG Controller for Linear Model')
% xlabel('Time')
% ylabel('State Outputs')

%LQG (using kalman filter gain) for non-linear system model
initial_state = [3,0.3,20,1,10,2,0,0,0,0,0,0];
simulation_time = 0:1:2000;
[time,out] = ode45(@ode45_callback_lqg,simulation_time,initial_state);
figure
```

```
plot(time,out)
title('LQG Controller for Non Linear Model')
xlabel('Time')
ylabel('State Estimates')
```

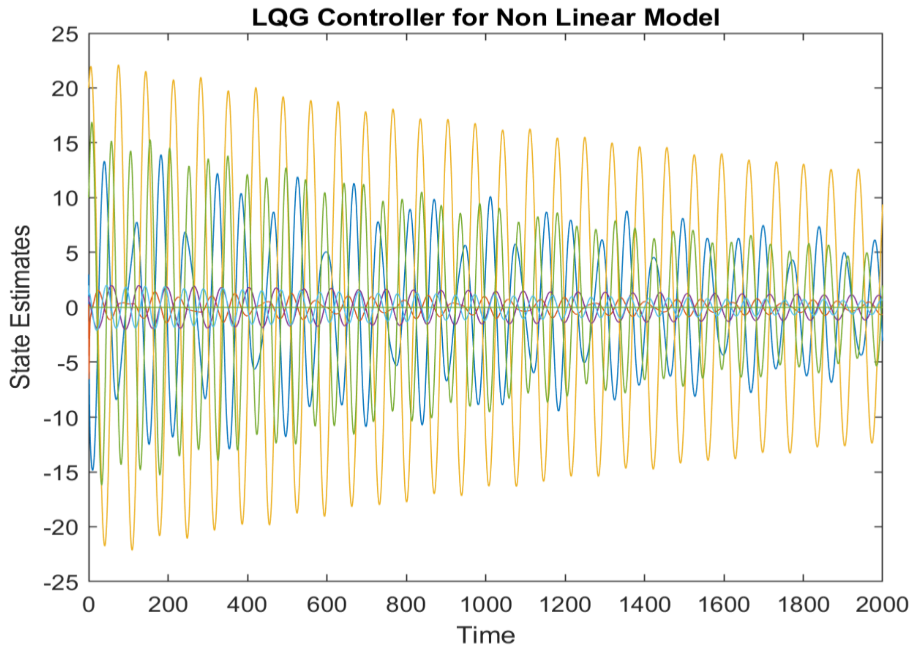

Figure 15 LQG controller for non-linear model

9.2 LQG Controller for Non-Linear Model

```
function output = ode45_callback_lqg(t,x)
M = 1000;
m1 = 100;
m2 = 100;
theta1 = x(3);
theta2 = x(5);
theta_dot_1 = x(4);
theta_dot_2 = x(6);
l1 = 20;
l2 = 10;
g = 9.8;
```

```
  [A,B,Q,R] = AB();
  [C_1,~,~] = get_C_matrices();
  [K,~,~] = lqr(A,B,Q,R);
  F= -K*x(1:6);

  output = zeros(12,1);

  x_ddot = (F-(m1*sind(theta1))*(g*cosd(theta1)+l1*theta_dot_1*theta_dot_1)-
m2*sind(theta2)*(g*cosd(theta2)+l2*theta_dot_2*theta_dot_2))/(M+m1*sind(theta1)*sind(thet
a1)+m2*sind(theta2)*sind(theta2));
  theta1_ddot = (x_ddot*cosd(theta1)-g*sind(theta1))/l1;
  theta2_ddot = (x_ddot*cosd(theta2)-g*sind(theta2))/l2;
  p_noise = eye(6);
  m_noise = 1;
  k_gain = lqr(A',C_1',p_noise,m_noise)';
  est = (A-k_gain*C_1)*x(7:12);

  output(1) = x(2); %because initial state has
x,x_dot,theta1,theta_dot_1,theta2,theta_dot_2,Estimation states (6)
  output(2) = x_ddot;
  output(3) = x(4);
  output(4) = theta1_ddot;
  output(5) = x(6);
  output(6) = theta2_ddot;
  output(7) = est(1);
  output(8) = est(2);
  output(9) = est(3);
  output(10) = est(4);
  output(11) = est(5);
  output(12) = est(6);
  end
```

CHAPTER – 10

ASSUMPTIONS

- The cart has no friction between the ground and the wheels.
- The motion is along one direction.
- The poles for the Luenberger observer are placed far away from the systems' poles.
- The code is provided with comments wherever it is thought as necessary to understand.
- The plots for the non-linear system simulations constitute of the estimations instead of the error in the state variables.
- Since the simulation is taking long time, the time for non-linear system is given with 2000 seconds.
- One can observe that the estimates are also damping, which means that the system will reach equilibrium point if given more time.
- The acceleration due to gravity is assumed to be 9.8 m/s.
- There is no collision between the two loads.
- The length of the strings doesn't change with the time and tension in the string.
- The state variables are taken from the figure 1 and are self-explanatory.
- Since our side goal is to look at the linearized system response for large values of initial conditions, the initial state is given with values that are far from the origin (equilibrium point).

CHAPTER – 11

CONCLUSION

The given set of problems were successfully implemented using MATLAB. The LQG controller takes the noise terms to output a system. Hence, when a constant disturbance is applied on the input, that disturbance can be modelled as an external disturbance and LQG controller design can reject it. To address the tracking of constant reference, the state equation of the system can be modelled with a term and the same system can be used to track the reference. The other way of tracking a reference is to introduce a new state variable, which is an integral term of the error in the LQR controller state equation. The resultant controller is also called as Linear Quadratic Integral (LQI controller). The github link for the project can be found here.

CHAPTER - 12

FUTURE SCOPE

- The Q and R matrices of LQR controller can be tuned more extensively to get optimal response of the system.
- The response of the non-linear system for the given initial states can be studied more.
- Since we are looking at estimates, the error can also be plotted to get the response of the non-linear model.
- We would also like to analyze the Luenberger observer gain matrices for different pole placements.