# ENPM – 667 CONTROL OF ROBOTIC SYSTEMS

# PROJECT – 2

# DESIGN AND SIMULATION OF LQR AND LQG CONTROLLERS


**BY**

**Varun Lakshmanan – 120169595**

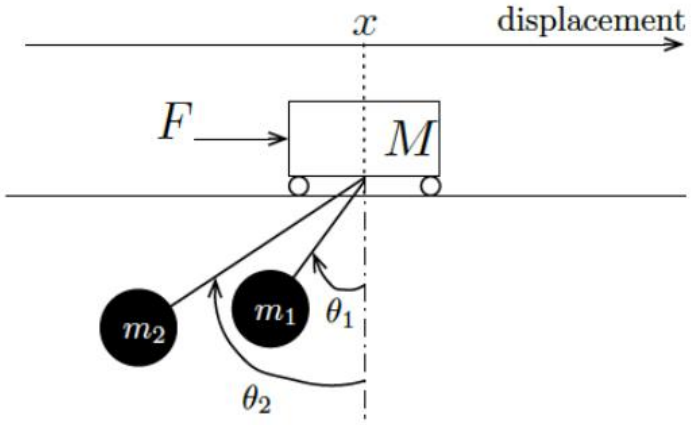
**Sai Jagadeesh Muralikrishnan – 120172243**

**Nitish Ravisankar Raveendran - 120385506**

# TABLE OF CONTENTS

# CHAPTER – 1

# DERIVING EQUATIONS OF MOTION FOR THE CRANE WITH TWO MASSES.




The above diagram shows a crane which behaves as a frictionless cart, moving in one-dimensional track with mass M actuated by an external force F. Loads with masses $m_1$and $m_2$ are suspended from the cables attached to the crane. $l_1$and $l_1$ are lengths of the cables.

First, the position vectors for the masses $m_1$, $m_2$ and $M$ are found.

For Mass 1,

$$m_{1\_}x = x - l_1 sin\theta_1 \quad \text{-(1.1)}$$

$$m_1 y = -l_1 cos\theta_1 \quad \text{-(1.2)}$$

$$\boldsymbol{m_1(t) = (x - l_1 sin\theta_1)i - (l_1 cos\theta_1)j} \quad \text{-(1.3)}$$

For Mass 2,

$$m_2 x = x - l_2 sin\theta_2 \quad \text{-(1.4)}$$

$$m_2 y = -l_2 cos\theta_2 \quad \text{-(1.5)}$$

$$\boldsymbol{m_2(t) = (x - l_2 sin\theta_2)i - (l_2 cos\theta_2)j} \quad \text{-(1.6)}$$

For the mass of the Crane,

$$M_x = x \quad \text{-(1.7)}$$

$$M_y = 0 \quad \text{-(1.8)}$$

$$M(t) = xi \quad \text{-(1.9)}$$

Velocities are calculated by differentiating the position vectors of each mass.

$$v_1(t) = \left[\dot{x} - l_1 cos(\theta_1)\dot{\theta_1}\right]i + \left[l_1 sin\theta_1\dot{\theta_1}\right]j \quad \text{-(1.10)}$$

$$v_2(t) = [\dot{x} - l_2 \cos(\theta_2)\,\dot{\theta_2}]i + \left[l_2 sin\theta_2\dot{\theta_2}\right]j \quad \text{-(1.11)}$$

$$v_M(t) = \dot{x} \quad \text{-(1.12)}$$

Taking squares of velocities,

$$v_1^2(t) = \left[\dot{x(t)} - l_1\dot{\theta_1(t)}\cos\left(\theta_1(t)\right)\right]^2 + \left[l_1\dot{\theta_1(t)}\sin\left(\theta_1(t)\right)\right]^2 \quad \text{-(1.13)}$$

$$v_2^2(t) = \left[\dot{x(t)} - l_2\dot{\theta_2(t)}\cos\left(\theta_2(t)\right)\right]^2 + \left[l_2\dot{\theta_2(t)}\sin\left(\theta_2(t)\right)\right]^2 \quad \text{-(1.14)}$$

$$v_M^2(t) = \left[\dot{x(t)}\right]^2 \quad \text{-(1.15)}$$

The Kinetic Energy of the System is given by the formula:

$$K.E = \frac{1}{2}Mv_M^2(t) + \frac{1}{2}m_1v_1^2(t) + \frac{1}{2}m_2v_2^2(t) \quad \text{-(1.16)}$$

$$K.E = \frac{1}{2}M\left[\dot{x(t)}\right]^2 + \frac{1}{2}m_1\left[\left[\dot{x(t)} - l_1\dot{\theta_1(t)}\,cos\left(\theta_1(t)\right)\right]^2 + \left[l_1\dot{\theta_1(t)}\,sin\left(\theta_1(t)\right)\right]^2\right] + \frac{1}{2}m_2\left[\left[\dot{x(t)} - l_2\dot{\theta_2(t)}\,cos\left(\theta_2(t)\right)\right]^2 + \left[l_2\dot{\theta_2(t)}\,sin\left(\theta_2(t)\right)\right]^2\right] \quad \text{-(1.17)}$$

$$K.E = \frac{1}{2}M\left[\dot{x(t)}\right]^2 + \frac{1}{2}m_1\left[\left[\dot{x(t)} - l_1\dot{\theta_1(t)}\,cos\left(\theta_1(t)\right)\right]^2 + \left[l_1\dot{\theta_1(t)}\,sin\left(\theta_1(t)\right)\right]^2\right] + \frac{1}{2}m_2\left[\left[\dot{x(t)} - l_2\dot{\theta_2(t)}\,cos\left(\theta_2(t)\right)\right]^2 + \left[l_2\dot{\theta_2(t)}\,sin\left(\theta_2(t)\right)\right]^2\right] \quad \text{-(1.18)}$$

$$K.E = \frac{1}{2}M\left[\dot{x(t)}\right]^2 + \frac{1}{2}m_1\dot{x(t)}^2 - m_1l_1\dot{\theta_1(t)}\cos\left(\theta_1(t)\right)\dot{x(t)} + \frac{1}{2}m_1l_1^2\dot{\theta_1(t)}^2\cos^2\left(\theta_1(t)\right) + \frac{1}{2}m_1l_1^2\dot{\theta_1(t)}^2\sin^2\left(\theta_1(t)\right) + \frac{1}{2}m_2\dot{x(t)}^2 - m_2l_2\dot{\theta_2(t)}\cos\left(\theta_2(t)\right)\dot{x(t)} + \frac{1}{2}m_2l_2^2\dot{\theta_2(t)}^2\cos^2\left(\theta_2(t)\right) + \frac{1}{2}m_2l_2^2\dot{\theta_2(t)}^2\sin^2\left(\theta_2(t)\right) \quad \text{-(1.19)}$$

$$\boldsymbol{K.E = \frac{1}{2}\dot{x(t)}^2[M + m_1 + m_2] + \frac{1}{2}\left[m_1l_1^2\dot{\theta_1(t)}^2 + m_2l_2^2\dot{\theta_2(t)}^2\right] - m_1l_1\dot{\theta_1(t)}\cos\left(\theta_1(t)\right)\dot{x(t)} - m_2l_2\dot{\theta_2(t)}\cos\left(\theta_2(t)\right)\dot{x(t)}} \quad \text{-(1.20)}$$

The Potential Energy of each mass is given by:

$$P.E_1 = -m_1gl_1\cos(\theta_1) \quad \text{-(1.21)}$$

$$P.E_2 = -m_2gl_2\cos(\theta_2) \quad \text{-(1.22)}$$

$$P.E_M = Mgh \quad \text{-(1.23)}$$

The Total Potential Energy of the system is Given by:

$$\boldsymbol{P.E = -m_1gl_1\cos(\theta_1) - m_2gl_2\cos(\theta_2) + Mgh} \quad \text{-(1.24)}$$

The equations of motion here are obtained by Euler-Lagrange Equations, where $L$ is the difference between Kinetic Energy and Potential given as:

$$L = K.E - P.E \quad \text{-(1.25)}$$

$$L = \frac{1}{2}\dot{x(t)}^2[M + m_1 + m_2] + \frac{1}{2}\left[m_1 l_1^2 \dot{\theta_1(t)}^2 + m_2 l_2^2 \dot{\theta_2(t)}^2\right] - m_1 l_1 \dot{\theta_1(t)} \cos\left(\theta_1(t)\right) \dot{x(t)} - m_2 l_2 \dot{\theta_2(t)} \cos\left(\theta_2(t)\right) \dot{x(t)} + m_1 g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_2) - Mgh \quad \text{-(1.26)}$$

Since $x, \theta_1$ and $\theta_2$ are the varying parameters, we are solving Lagrange for three cases,

Case-1

Where, we take

$$\boldsymbol{\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x}}\right] - \frac{\partial L}{\partial x} = F} \quad \text{-(1.27)}$$

Solving for Eq-(27)

$$\left[\frac{\partial L}{\partial \dot{x}}\right] = \dot{x(t)}[M + m_1 + m_2] - m_1 l_1 \dot{\theta_1(t)} \cos\left(\theta_1(t)\right) - m_2 l_2 \dot{\theta_2(t)} \cos\left(\theta_2(t)\right) \quad \text{-(1.28)}$$

$$\frac{\partial L}{\partial x} = 0 \quad \text{-(1.29)}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x}}\right] = \ddot{x(t)}[M + m_1 + m_2] - m_1 l_1\left[\ddot{\theta_1(t)} \cos\left(\theta_1(t)\right) - \dot{\theta_1(t)}^2 \sin\left(\theta_1(t)\right)\right] - m_2 l_2\left[\ddot{\theta_2(t)} \cos\left(\theta_2(t)\right) - \dot{\theta_2(t)}^2 \sin\left(\theta_2(t)\right)\right] \quad \text{-(1.30)}$$

The Final Equation for Case-1 is:

$$\boxed{\boldsymbol{\ddot{x(t)}[M + m_1 + m_2] - m_1 l_1\left[\ddot{\theta_1(t)} \cos\left(\theta_1(t)\right) - \dot{\theta_1(t)}^2 \sin\left(\theta_1(t)\right)\right] - m_2 l_2\left[\ddot{\theta_2(t)} \cos\left(\theta_2(t)\right) - \dot{\theta_2(t)}^2 \sin\left(\theta_2(t)\right)\right] = F}} \quad \text{-(1.31)}$$

Case-2

Where, we take.

$$\boldsymbol{\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta_1}}\right] - \frac{\partial L}{\partial \theta_1} = 0} \quad \text{-(1.32)}$$

$$\left[\frac{\partial L}{\partial \dot{\theta_1}}\right] = m_1 l_1^2 \dot{\theta_1(t)} - m_1 l_1 \cos\left(\theta_1(t)\right) \dot{x(t)} \quad \text{-(1.33)}$$

$$\frac{\partial L}{\partial \theta_1} = [m_1 l_1 \dot{\theta_1}(t) \sin(\theta_1(t))\, \dot{x}(t) - m_1 g l_1 \sin(\theta_1(t)) \quad \text{-(1.34)}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta_1}}\right] = m_1 l_1^2 \ddot{\theta_1}(t) - m_1 l_1 [\ddot{x}(t) \cos(\theta_1(t)) - \dot{x}(t) sin(\theta_1(t)) \dot{\theta_1}(t)] \quad \text{-(1.35)}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta_1}}\right] - \frac{\partial L}{\partial \theta_1} = m_1 l_1^2 \ddot{\theta_1}(t) - m_1 l_1 [\ddot{x}(t) \cos(\theta_1(t)) - \dot{x}(t) sin(\theta_1(t)) \dot{\theta_1}(t)] - [m_1 l_1 \dot{\theta_1}(t) \sin(\theta_1(t))\, \dot{x}(t) - m_1 g l_1 \sin(\theta_1(t))] = 0 \quad \text{-(1.36)}$$

$$m_1 l_1^2 \ddot{\theta_1}(t) - m_1 l_1 \ddot{x}(t) \cos(\theta_1(t)) + m_1 l_1 \dot{x}(t) sin(\theta_1(t)) \dot{\theta_1}(t) - m_1 l_1 \dot{\theta_1}(t) \sin(\theta_1(t))\, \dot{x}(t) + m_1 g l_1 \sin(\theta_1(t)) = 0 \quad \text{-(1.37)}$$

The Final equation for Case-2 is:

$$\boxed{\boldsymbol{m_1 l_1^2 \ddot{\theta_1}(t) - m_1 l_1 \ddot{x}(t) \cos(\theta_1(t)) + m_1 g l_1 \sin(\theta_1(t)) = 0}} \quad \text{-(1.38)}$$

CASE-3

Where, we take.

$$\boldsymbol{\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta_2}}\right] - \frac{\partial L}{\partial \theta_2} = 0} \quad \text{-(1.39)}$$

$$\left[\frac{\partial L}{\partial \dot{\theta_2}}\right] = m_2 l_2^2 \dot{\theta_2}(t) - m_2 l_2 \cos(\theta_2(t))\, \dot{x}(t) \quad \text{-(1.40)}$$

$$\frac{\partial L}{\partial \theta_2} = [m_2 l_2 \dot{\theta_2}(t) \sin(\theta_2(t))\, \dot{x}(t) - m_2 g l_2 \sin(\theta_2(t))] \quad \text{-(1.41)}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta_2}}\right] = m_2 l_2^2 \ddot{\theta_2}(t) - m_2 l_2 [\ddot{x}(t) \cos(\theta_2(t)) - \dot{x}(t) sin(\theta_2(t)) \dot{\theta_2}(t)] \quad \text{-(1.42)}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta_2}}\right] - \frac{\partial L}{\partial \theta_2} = m_2 l_2^2 \ddot{\theta_2}(t) - m_2 l_2 [\ddot{x}(t) \cos(\theta_2(t)) - \dot{x}(t) sin(\theta_2(t)) \dot{\theta_2}(t)] - [m_2 l_2 \dot{\theta_2}(t) \sin(\theta_2(t))\, \dot{x}(t) - m_2 g l_2 \sin(\theta_2(t))] = 0 \quad \text{-(1.43)}$$

$$m_2 l_2^2 \ddot{\theta_2}(t) - m_2 l_2 [\ddot{x}(t) \cos(\theta_2(t)) - \dot{x}(t) sin(\theta_2(t)) \dot{\theta_2}(t)] - [m_2 l_2 \dot{\theta_2}(t) \sin(\theta_2(t))\, \dot{x}(t) - m_2 g l_2 \sin(\theta_2(t))] = 0 \quad \text{-(1.44)}$$

The Final Equation for Case-3 is:

$$\boldsymbol{m_2 l_2^2 \ddot{\theta}_2(t) - m_2 l_2 \ddot{x}(t) \cos(\theta_2(t)) + m_2 g l_2 \sin(\theta_2(t)) = 0} \quad \text{-(1.45)}$$

Using equation (31) for obtaining $\ddot{x}(t)$.

$$\boldsymbol{\ddot{x}(t)} = \frac{\left[\boldsymbol{F + m_1 l_1 [\ddot{\theta}_1(t) \cos(\theta_1(t)) - \dot{\theta}_1(t)^2 \sin(\theta_1(t))] + m_2 l_2 [\ddot{\theta}_2(t) \cos(\theta_2(t)) - \dot{\theta}_2(t)^2 \sin(\theta_2(t))]}\right]}{[\boldsymbol{M + m_1 + m_2}]} \quad \text{-(1.46)}$$

Using equation (38) for obtaining $\ddot{\theta}_1(t)$.

$$\ddot{\theta}_1(t) = \frac{[m_1 l_1 \ddot{x}(t) \cos(\theta_1(t)) - m_1 g l_1 \sin(\theta_1(t))]}{m_1 l_1^2} \quad \text{-(1.47)}$$

$$\ddot{\theta}_1(t) = m_1 l_1 [\ddot{x}(t) \cos(\theta_1(t)) - g \sin(\theta_1(t))] / m_1 l_1^2 \quad \text{-(1.48)}$$

$$\boldsymbol{\ddot{\theta}_1(t) = \frac{[\ddot{x}(t) \cos(\theta_1(t)) - g \sin(\theta_1(t))]}{l_1}} \quad \text{-(1.49)}$$

Using equation (45) for obtaining $\ddot{\theta}_2(t)$.

$$\ddot{\theta}_2(t) = \frac{[m_2 l_2 \ddot{x}(t) \cos(\theta_2(t)) - m_2 g l_2 \sin(\theta_2(t))]}{m_2 l_2^2} \quad \text{-(1.50)}$$

$$\ddot{\theta}_2(t) = m_2 l_2 [\ddot{x}(t) \cos(\theta_2(t)) - g \sin(\theta_2(t))] / m_2 l_2^2 \quad \text{-(1.51)}$$

$$\boldsymbol{\ddot{\theta}_2(t) = \frac{[\ddot{x}(t) \cos(\theta_2(t)) - g \sin(\theta_2(t))]}{l_2}} \quad \text{-(1.52)}$$

Substituting $\ddot{\theta}_1(t)$ and $\ddot{\theta}_2(t)$ in $\ddot{x}(t)$ we get,

$$\ddot{x}(t) = \frac{\left[F + m_1 \left[\left[\ddot{x}(t) \cos(\theta_1(t))^2 - g \sin(\theta_1(t)) \cos(\theta_1(t))\right] - l_1 \dot{\theta_1}(t)^2 \sin(\theta_1(t))\right] + m_2 \left[\left[\ddot{x}(t) \cos(\theta_2(t))^2 - g \sin(\theta_2(t)) \cos(\theta_2(t))\right] - l_2 \dot{\theta_2}(t)^2 \sin(\theta_2(t))\right]\right]}{[M + m_1 + m_2]}$$

-(1.53)

Simplifying the above equation step by step,

$$x\ddot{(t)}[M + m_1 + m_2] = \Bigg[F + m_1 \Big[\Big[x\ddot{(t)} \cos\big(\theta_1(t)\big)^2 - g \sin\big(\theta_1(t)\big) \cos\big(\theta_1(t)\big)\Big] - l_1 \dot{\theta_1}(t)^2 \sin\big(\theta_1(t)\big)\Big] + m_2 \Big[\Big[x\ddot{(t)} \cos\big(\theta_2(t)\big)^2 - g \sin\big(\theta_2(t)\big) \cos\big(\theta_2(t)\big)\Big] - l_2 \dot{\theta_2}(t)^2 \sin\big(\theta_2(t)\big)\Big]\Bigg] \quad \text{-(1.54)}$$

$$x\ddot{(t)}[M + m_1 + m_2] - m_1 x\ddot{(t)} \cos\big(\theta_1(t)\big)^2 - m_2 x\ddot{(t)} \cos\big(\theta_2(t)\big)^2 = \big[F - m_1 g \sin\big(\theta_1(t)\big) \cos\big(\theta_1(t)\big) - m_1 l_1 \dot{\theta_1}(t)^2 \sin\big(\theta_1(t)\big) - m_2 g \sin\big(\theta_2(t)\big) \cos\big(\theta_2(t)\big) - m_2 l_2 \dot{\theta_2}(t)^2 \sin\big(\theta_2(t)\big)\big] \quad \text{-(1.55)}$$

$$x\ddot{(t)} \Big[M + m_1 + m_2 - m_1 \cos\big(\theta_1(t)\big)^2 - m_2 \cos\big(\theta_2(t)\big)^2\Big] = \big[F - m_1 g \sin\big(\theta_1(t)\big) \cos\big(\theta_1(t)\big) - m_1 l_1 \dot{\theta_1}(t)^2 \sin\big(\theta_1(t)\big) - m_2 g \sin\big(\theta_2(t)\big) \cos\big(\theta_2(t)\big) - m_2 l_2 \dot{\theta_2}(t)^2 \sin\big(\theta_2(t)\big)\big] \quad \text{-(1.56)}$$

$$x\ddot{(t)} \Big[M + m_1(1 - \cos\big(\theta_1(t)\big)^2) - m_2(1 - \cos\big(\theta_2(t)\big)^2)\Big] = \big[F - m_1 g \sin\big(\theta_1(t)\big) \cos\big(\theta_1(t)\big) - m_1 l_1 \dot{\theta_1}(t)^2 \sin\big(\theta_1(t)\big) - m_2 g \sin\big(\theta_2(t)\big) \cos\big(\theta_2(t)\big) - m_2 l_2 \dot{\theta_2}(t)^2 \sin\big(\theta_2(t)\big)\big] \quad \text{-(1.57)}$$

$$x\ddot{(t)}\big[M + m_1 \sin^2\big(\theta_1(t)\big) + m_2 \sin^2\big(\theta_2(t)\big)\big] = \big[F - m_1 g \sin\big(\theta_1(t)\big) \cos\big(\theta_1(t)\big) - m_1 l_1 \dot{\theta_1}(t)^2 \sin\big(\theta_1(t)\big) - m_2 g \sin\big(\theta_2(t)\big) \cos\big(\theta_2(t)\big) - m_2 l_2 \dot{\theta_2}(t)^2 \sin\big(\theta_2(t)\big)\big] \quad \text{-(1.58)}$$

Now, Euler equation for $x\ddot{(t)}$ is given by:

$$\ddot{x}(t) = \frac{[F - m_1 g \sin(\theta_1(t)) \cos(\theta_1(t)) - m_1 l_1 \dot{\theta_1}(t)^2 \sin(\theta_1(t)) - m_2 g \sin(\theta_2(t)) \cos(\theta_2(t)) - m_2 l_2 \dot{\theta_2}(t)^2 \sin(\theta_2(t))]}{[M + m_1 \sin^2(\theta_1(t)) + m_2 \sin^2(\theta_2(t))]}$$

-(1.59)

Substituting $\ddot{x}(t)$ in $\ddot{\theta_1}(t)$ and $\ddot{\theta_2}(t)$ we get Euler angles as given below,

$$\ddot{\theta_1}(t) = \frac{\cos(\theta_1(t))}{l_1} \frac{[F - m_1 g \sin(\theta_1(t)) \cos(\theta_1(t)) - m_1 l_1 \dot{\theta_1}(t)^2 \sin(\theta_1(t)) - m_2 g \sin(\theta_2(t)) \cos(\theta_2(t)) - m_2 l_2 \dot{\theta_2}(t)^2 \sin(\theta_2(t))]}{[M + m_1 \sin^2(\theta_1(t)) + m_2 \sin^2(\theta_2(t))]} - \frac{g \sin(\theta_1(t))}{l_1}$$

-(1.60)

$$\ddot{\theta_2}(t) = \frac{\cos(\theta_2(t))}{l_2} \frac{[F - m_1 g \sin(\theta_1(t)) \cos(\theta_1(t)) - m_1 l_1 \dot{\theta_1}(t)^2 \sin(\theta_1(t)) - m_2 g \sin(\theta_2(t)) \cos(\theta_2(t)) - m_2 l_2 \dot{\theta_2}(t)^2 \sin(\theta_2(t))]}{[M + m_1 \sin^2(\theta_1(t)) + m_2 \sin^2(\theta_2(t))]} - \frac{g \sin(\theta_2(t))}{l_2}$$

-(1.61)

## Obtaining Non-Linear State Space Representation from the equations of motion

The Non-Linear State Space Representation obtained by using equations (1.59), (1.60) and (1.61) is given below:

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

-(1.62)

# CHAPTER – 2

# LINEARIZATION OF NON- LINEAR SYSTEM

Linearization of a Non-Linear System can be done by two methods:

- By Using Small angle approximations.
- By calculating Jacobian matrices

In this Project we have used small angle approximations in which we approximate the main Trigonometric values to linearize the non-linear system around the equilibrium points $x = 0, \theta_1 = 0, \theta_2 = 0$.

The following are the approximations we took to linearize the system:

$$\theta_1^2 \approx \theta_2^2 \approx 0$$

$$sin\theta_1 \approx \theta_1$$

$$sin\theta_2 \approx \theta_2$$

$$cos\theta_1 \approx 1 - \theta_1^2/2 \approx 1$$

$$cos\theta_2 \approx 1 - \theta_2^2/2 \approx 1$$

Implementing these approximations in the Euler equations $\ddot{x}(t), \ddot{\theta_1}(t)$ and $\ddot{\theta_2}(t)$ we get:

$$\ddot{x}(t) = \frac{[F - m_1 g\theta_1(t) - m_2 g\theta_2(t)]}{M} \quad \text{-(2.1)}$$

$$\ddot{\theta_1}(t) = \frac{[F - m_1 g\theta_1(t) - m_2 g\theta_2(t) - Mg\theta_1(t)]}{Ml_1} \quad \text{-(2.2)}$$

$$\ddot{\theta_2}(t) = \frac{[F - m_1 g\theta_1(t) - m_2 g\theta_2(t) - Mg\theta_2(t)]}{Ml_2} \quad \text{-(2.3)}$$

Using the above equations, we can form a state space representation of the linearized system:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \\ \dot{\theta_1}(t) \\ \ddot{\theta_1}(t) \\ \dot{\theta_2}(t) \\ \ddot{\theta_2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_1 g/M & 0 & -m_2 g/M & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -(M+m_1)g/Ml_1 & 0 & -m_2 g/Ml_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -m_1 g/Ml_2 & 0 & -(M+m_2)g/Ml_2 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta_1}(t) \\ \theta_2(t) \\ \dot{\theta_2}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 1/Ml_1 \\ 0 \\ 1/Ml_2 \end{bmatrix} \quad \text{-(2.4)}$$

The above state space representation is of the form,

$$\dot{x} = Ax + Bu$$

$$y = Cx + D$$

Where,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m_1 g/M & 0 & -m_2 g/M & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -(M+m_1)g/Ml_1 & 0 & -m_2 g/Ml_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -m_1 g/Ml_2 & 0 & -(M+m_2)g/Ml_2 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 1/Ml_1 \\ 0 \\ 1/Ml_2 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \\ \dot{\theta_1}(t) \\ \ddot{\theta_1}(t) \\ \dot{\theta_2}(t) \\ \ddot{\theta_2}(t) \end{bmatrix}, \; x = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta_1}(t) \\ \theta_2(t) \\ \dot{\theta_2}(t) \end{bmatrix} \; C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \; D = 0$$

# CHAPTER 3

# CONTROLLABILITY CONDITONS

The controllability conditions for the system are checked by doing the following steps:

- Checking the order of the state space representation by verifying the order of matrices $A, B$.
- Checking if the below condition satisfies:

$$\text{Rank}(\ [B_k : AB_k : A^2B_k : A^3B_k : A^4B_k : A^5B_k : A^6B_k]) = n$$

- If Rank satisfies the rank of the n matrix, then the system is controllable.
- Rank can be found by checking the number of linearly independent rows.
- The system is controllable for $l_1 \neq l_2$, $m_1, m_2, M, l_1, l_2$, the controllability matrix satisfies the order of matrix n, and the determinant is nonzero.
- When $l_1 = l_2$ the system is uncontrollable, since the determinant of the controllability matrix for this case is zero.

For Controllability Matrix: (Controllable)

```
Determinant: -(g^6*l1^2 + g^6*l2^2 - 2*g^6*l1*l2)/(M^6*l1^6*l2^6)
Rank: 6
System is controllable due to rank = n
```

For $l_1 = l_2$, Controllability Matrix:

```
Determinant: 0
Rank: 4
System is not controllable due to rank != n
```

The Controllability is checked using MATLAB code.

# CHAPTER 4

# LQR CONTROLLER

A linear quadratic controller is a controller in which the system dynamics are represented by a set of linear differential equations and the cost is described by a quadratic function. Since, the system is already linearized.

The LQR controller is designed using following steps:

- Optimizing the Cost Function:

$$\boldsymbol{J\left(K,\vec{X}(0)\right)=\int_0^{\infty}\vec{X}^{T}(t)Q\vec{X}(t)+\overrightarrow{U_k}^{T}(t)R\overrightarrow{U_K}(t)dt} \qquad \text{-(4.1)}$$

  Where $Q$ and $R$ are positive definite matrices.

- Where P is the symmetric positive solution of the following Riccati equation, which is used to minimize the cost function:

$$\boldsymbol{A^TP+PA-PBR^{-1}B^TP=-Q} \qquad \text{- (4.2)}$$

- Finding the optimal gain K:

$$\boldsymbol{K=-R^{-1}B_k^TP} \qquad \text{-(4.3)}$$

- The optimal gain is further used to modify the state-space equation.

$$\boldsymbol{\dot{x}=(A-BK)x} \qquad \text{-(4.4)}$$

For $Q$ and $R$ we have chosen the following values:

$$Q=\begin{bmatrix}20&0&0&0&0&0\\0&200&0&0&0&0\\0&0&1000&0&0&0\\0&0&0&1000&0&0\\0&0&0&0&200&0\\0&0&0&0&0&20\end{bmatrix}$$

$$R=0.005$$

For Linear System 1: (This is before the implementation of optimal gain)

The P Matrix obtained through the simulation is given below:

```
P :
   1.0e+04 *

    0.0096    0.0132    0.0050   -0.0399   -0.0018   -0.0119
    0.0132    0.0632    0.0639   -0.1817    0.0032   -0.0593
    0.0050    0.0639    2.1705   -0.0438   -0.0648   -0.1305
   -0.0399   -0.1817   -0.0438    4.3255    0.1260   -0.0958
   -0.0018    0.0032   -0.0648    0.1260    0.5288   -0.0002
   -0.0119   -0.0593   -0.1305   -0.0958   -0.0002    0.5499
```

The State Feedback gain matrix K is found as:

```
K :
  200.0000  963.0228  973.6805  499.9186  189.1229 -181.7721
```

The Poles are calculated using the eigenvalues of (A)

```
Poles:
  -0.0389 + 1.0262i
  -0.0389 - 1.0262i
  -0.0424 + 0.7181i
  -0.0424 - 0.7181i
  -0.4036 + 0.1159i
  -0.4036 - 0.1159i
```

The Stability is checked by **Lyapunov indirect Method**, where the eigen values of matrix A are checked. Here, all the poles have negative real part, so the poles are placed on left half plane. Hence, the system is stable.

For Linear System 2: (This is after the implementation of optimal gain)

The P Matrix is:

```
P :
   1.0e+04 *

    0.0082    0.0066    0.0030   -0.0352   -0.0012   -0.0074
    0.0066    0.0248    0.0468   -0.1367    0.0026   -0.0300
    0.0030    0.0468    1.4356   -0.0482   -0.0290   -0.1207
   -0.0352   -0.1367   -0.0482    2.8764    0.1191   -0.0384
   -0.0012    0.0026   -0.0290    0.1191    0.2934   -0.0007
   -0.0074   -0.0300   -0.1207   -0.0384   -0.0007    0.3029
```

The K Matrix is:

```
K :
   82.8427   299.1037   646.6854    66.3002   169.9168   -32.7244
```

The Poles from the eigen values of (A-BK):

```
Poles:
  -0.0454 + 1.0165i
  -0.0454 - 1.0165i
  -0.0504 + 0.7121i
  -0.0504 - 0.7121i
  -0.3591 + 0.0000i
  -0.7184 + 0.0000i
```

Using Lyapunov indirect stability test, we can say that the system is stable.

We considered Q and R values in a way that the system is efficiently controllable.

Q is decided depending on the time taken by the system for stabilizing a particular state. We used the lowest possible value for R to obtain stability quickly, where the energy utilization is less.

The State Space Responses of both Linear systems are given below:

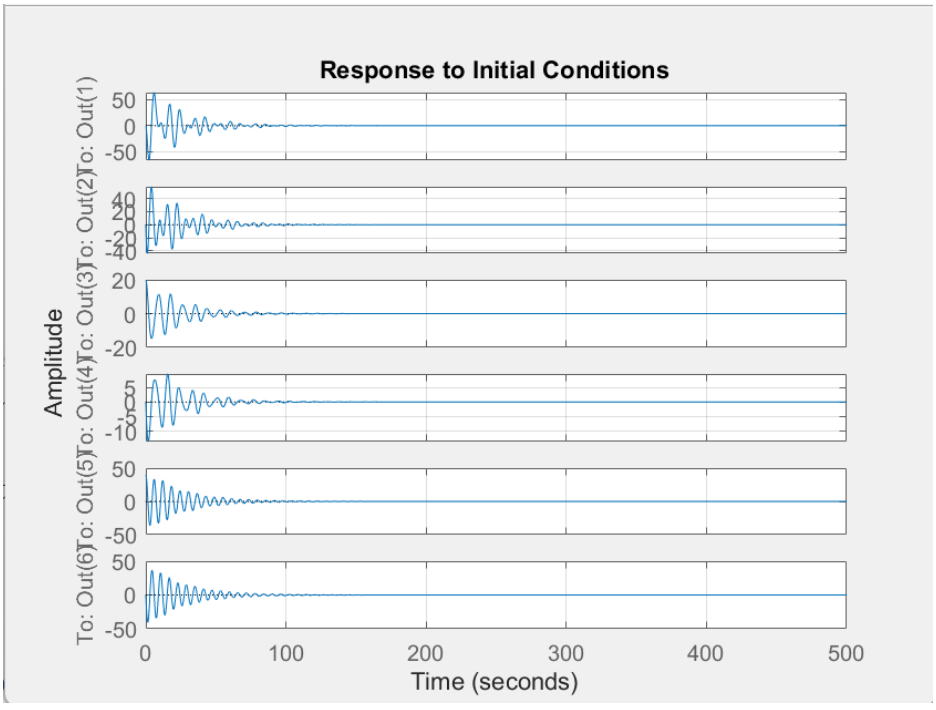



*Figure 1: Simulation of Linear system after optimal gain implementation*

For Non -Linear System:

We used initial conditions as $\theta_1(t) = 20\ degrees,\ \theta_2(t) = 40\ degrees$

The P matrix is:

```
P:
   1.0e+04 *

    0.0096    0.0132    0.0050   -0.0399   -0.0018   -0.0119
    0.0132    0.0632    0.0639   -0.1817    0.0032   -0.0593
    0.0050    0.0639    2.1705   -0.0438   -0.0648   -0.1305
   -0.0399   -0.1817   -0.0438    4.3255    0.1260   -0.0958
   -0.0018    0.0032   -0.0648    0.1260    0.5288   -0.0002
   -0.0119   -0.0593   -0.1305   -0.0958   -0.0002    0.5499
```

The State feedback gain K:

```
K:
  200.0000  963.0228  973.6805  499.9186  189.1229 -181.7721
```

The poles derived from the eigenvalues of matrix A are:

```
Poles:
  -0.0389 + 1.0262i
  -0.0389 - 1.0262i
  -0.0424 + 0.7181i
  -0.0424 - 0.7181i
  -0.4036 + 0.1159i
  -0.4036 - 0.1159i
```

The system is stable, since the real part of every pole is negative.

The Initial State Response of the Non-Linear system is given below:

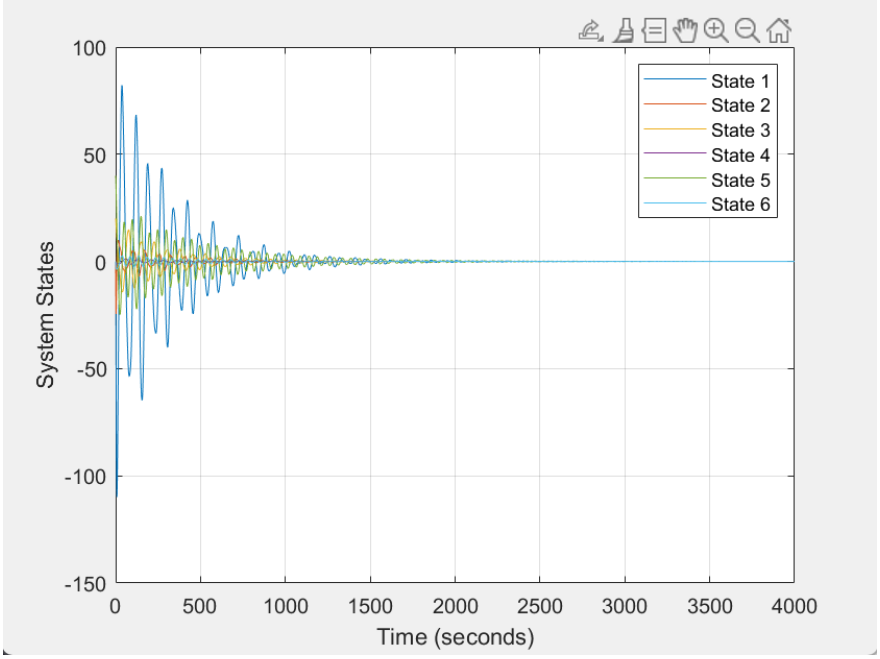



*Figure 2:Simulation of Non-linear system using LQR controller.*

# CHAPTER – 5

# SYSTEM OBSERVABILITY

We are considering output vectors,

$$x(t), \left(\theta_1(t), \theta_2(t)\right), \left(x(t), \theta_2(t)\right), \left(x(t), \theta_1(t), \theta_2(t)\right)$$

Now we will be having four observer matrices $C_1, C_2, C_3, C_4$

Where,

$$C_1 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \text{ for } x(t)$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \text{ for } \left(\theta_1(t), \theta_2(t)\right)$$

$$C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ for } \left(x(t), \theta_2(t)\right)$$

$$C_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ for } \left(x(t), \theta_1(t), \theta_2(t)\right)$$

To check for observability, we are using the formula:

$$\boldsymbol{Rank_{Observability} = rank\left(\left[C^T\ A^T C^T\ A^{T^2} C^T\ A^{T^3} C\ A^{T^4} C\ A^{T^5} C\right]\right)} \qquad \textbf{-(5.1)}$$

Using this formula, the rank of each observer is checked and compared with the $n$ matrix.

After comparing we get the following results:

```
>> Part_E_Obervability
Rank of observability matrix when monitoring x(t) = 6
Rank of observability matrix when monitoring theta1(t), theta2(t) = 4
Rank of observability matrix when monitoring x(t) and theta2(t) = 6
Rank of observability matrix  when monitoring x(t),theta1(t) and theta2(t) = 6
For case 1 x(t) is observable.
For case 2 theta1(t), theta2(t) is not observable.
For case 3 x(t), theta2(t) is observable.
For case 4 x(t), theta1(t), theta2(t) is observable.
```

After analyzing the results, we found that $x(t)$, $\left(x(t), \theta_2(t)\right)$, $\left(x(t), \theta_1(t), \theta_2(t)\right)$ are observable and $\left(\theta_1(t), \theta_2(t)\right)$ is not observable, since it did not satisfy the observability conditions.

# CHAPTER 6

# LUENBERGER OBSERVER

The state-space representation of the Luenberger observer is given as follows.

$$\boldsymbol{\hat{x}(t) = A\hat{x}(t) + B_K U_K(t) + L\left(Y(t) - C\hat{x}(t)\right)} \qquad \textbf{-} (6.1)$$

Where,

$L$ – observer matrix

$C\hat{x}(t)$ – Correction term

$$x_e(t) = x(t) - \hat{x}(t) \qquad \text{-(6.2)}$$

The equation (6.2) is the estimation error.

The Estimation error has the following state-space representation:

$$\dot{x_e}(t) = Ax_e(t) - L\big(Y(t) - C\hat{x}(t)\big) + B_D U_D(t) \quad \text{-(6.3)}$$

Assuming $D = 0$ and $Y(t) = Cx(t)$

$$\dot{x_e}(t) = Ax_e(t) - LC\big(x(t) - \hat{x}(t)\big) + B_D U_D(t) \quad \text{-(6.4)}$$

$$\boldsymbol{\dot{x}_e(t) = (A - LC)x_e(t) + B_D U_D(t)} \quad \text{-(6.5)}$$

Here, we are only considering $C_1, C_3$ and $C_4$ since they are observable matrices.

The MATLAB simulations for Luenberger Observer for each observable matrices in Linear system are shown below for the following conditions.

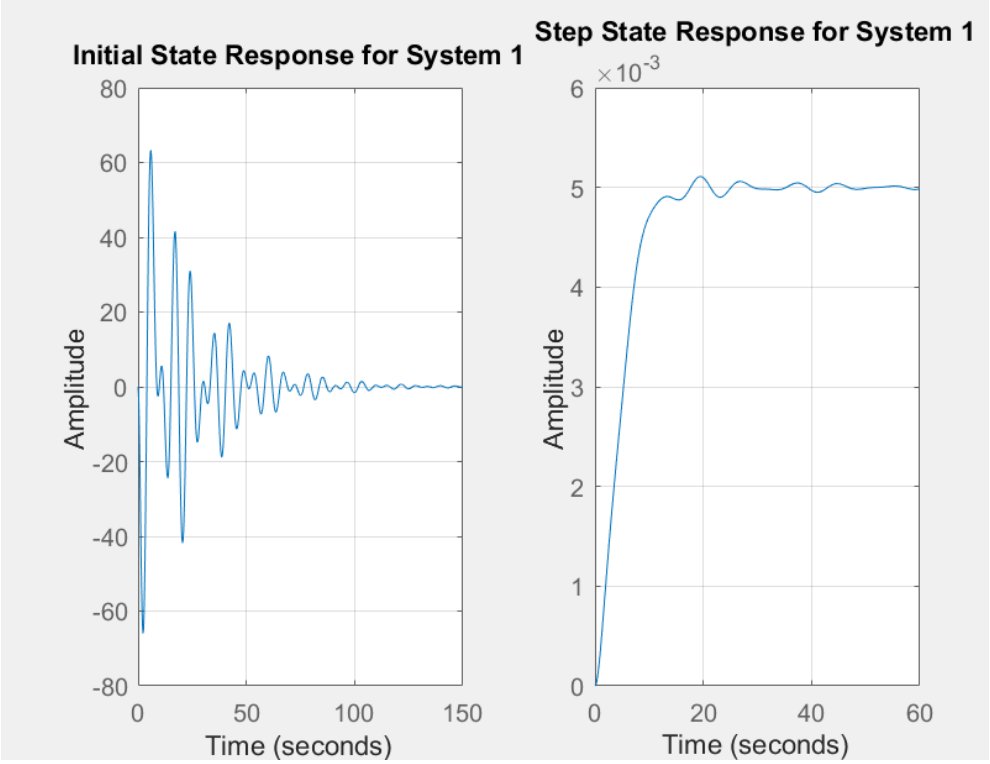



*Figure 3: Simulation for Luenberger observer 1*

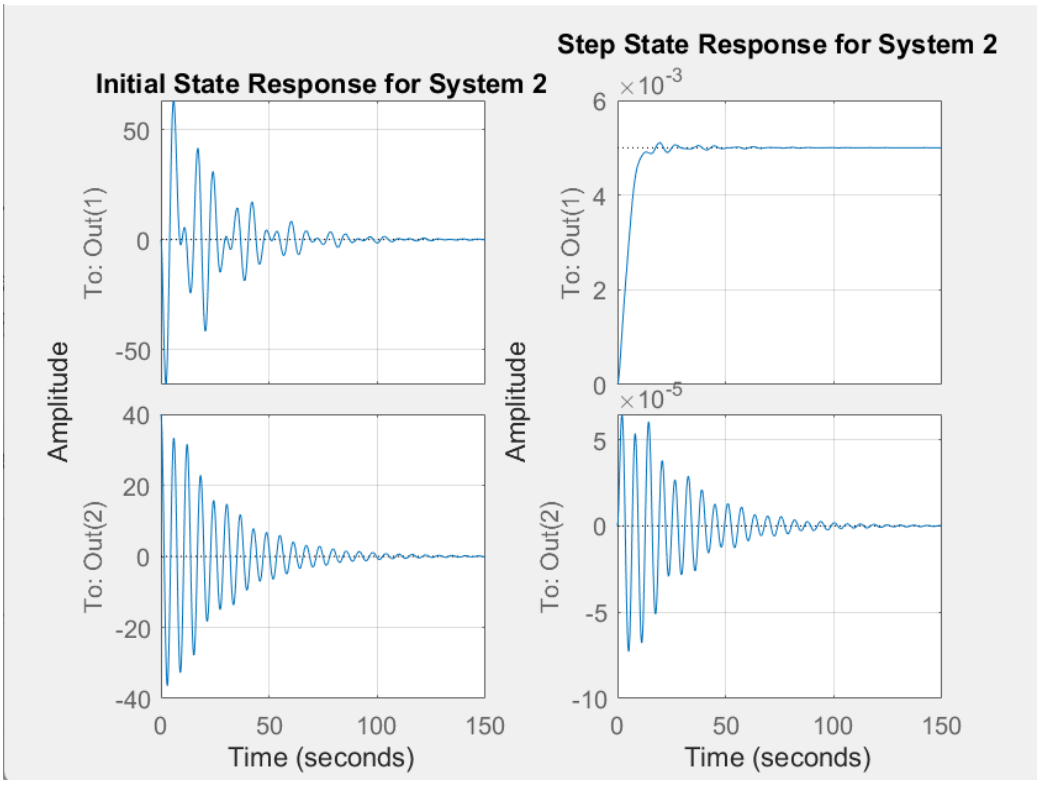



*Figure 4: Simulation for Luenberger observer 2*

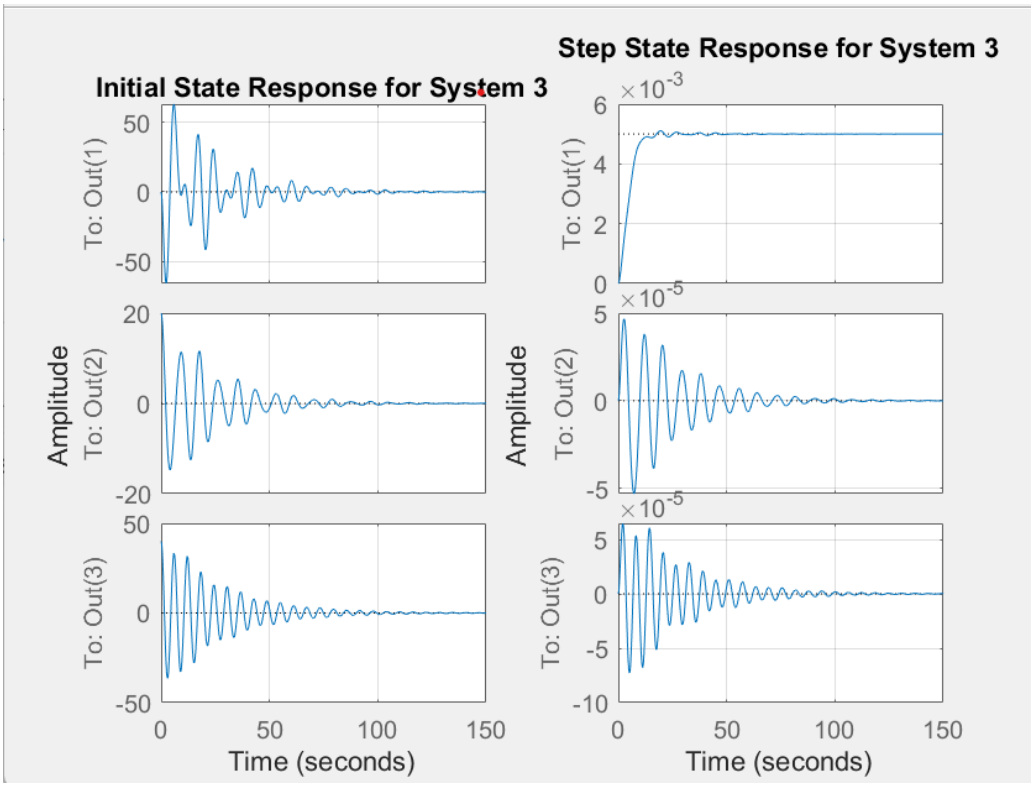
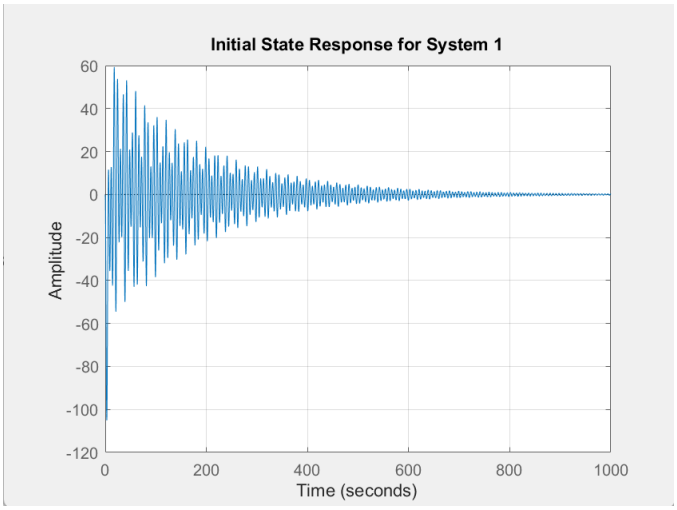



*Figure 5: Simulation for Luenberger observer 3*

The simulations are taken for the output vectors $x(t)$, $\big(x(t), \theta_2(t)\big)$, $\big(x(t), \theta_1(t), \theta_2(t)\big)$ respectively.

The Simulations of Luenberger observer for nonlinear system are given below:



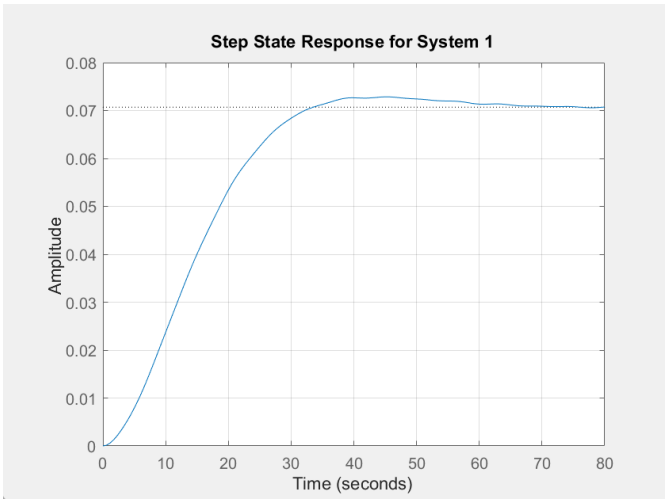
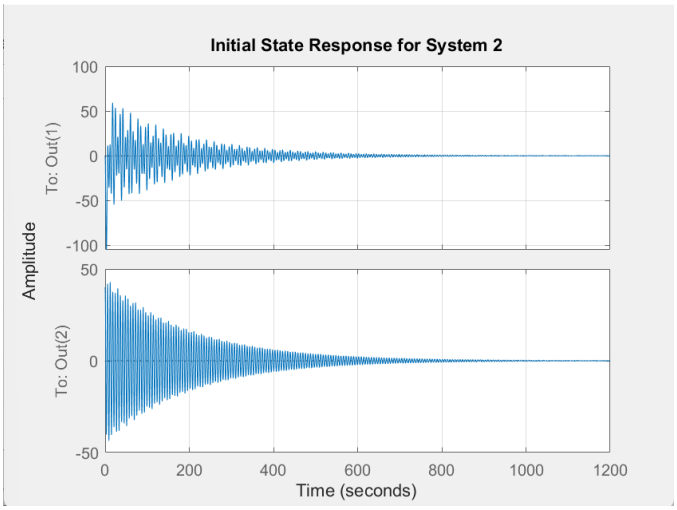




*Figure 6: Luenberger Simulation for System 1(Non-Linear)*



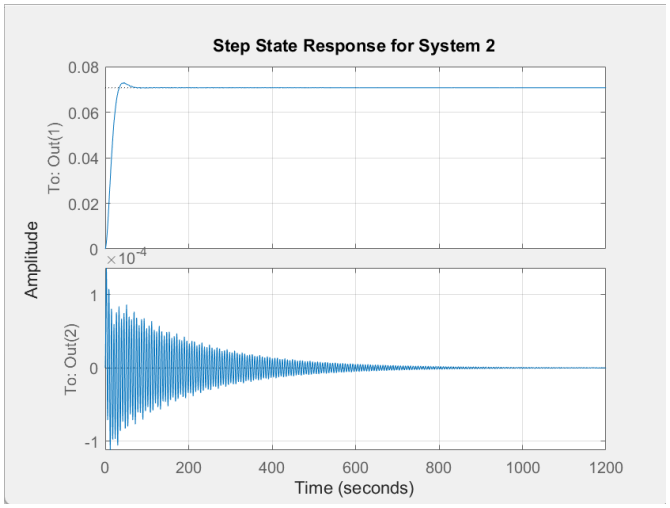
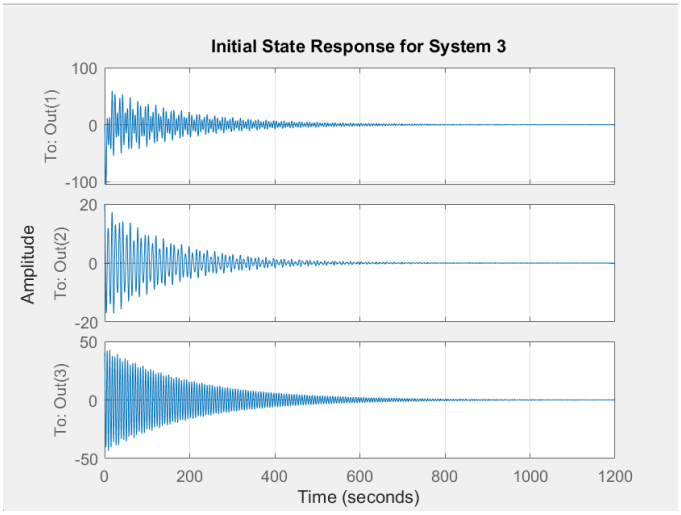




*Figure 7:Luenberger Simulation for system 2(Non-Linear*



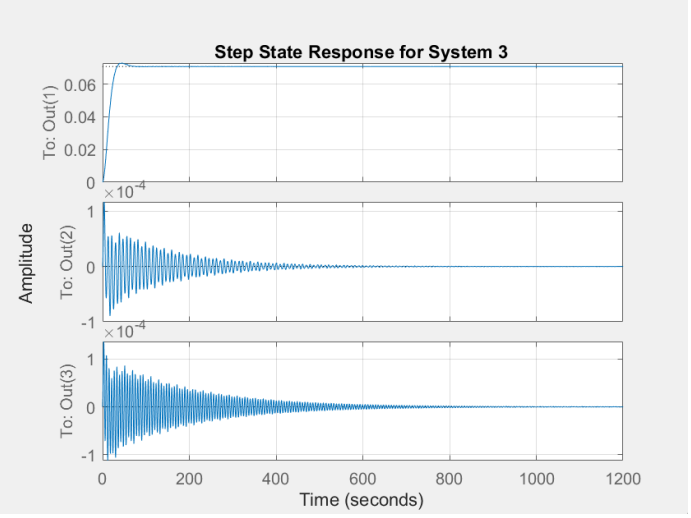




*Figure 8: Luenberger Observer for system 3(Non-Linear)*

# CHAPTER -7

# LQG CONTROLLER

The Linear Quadratic Gaussian controller which is the combination of Kalman-Filter and the Linear Quadratic Regulator. The LQG controller is designed by the following steps:

- Choosing the Smallest possible vector: Comparing the observable matrices $C_4$ was the largest among the other two observable matrices and $C_3$ and $C_1$. Here, we choose the output vector $x(t)$.
- The LQR controller is designed.
- The Kalman-Bucy filter is designed with small values of.
- The LQR controller and Kalman-Bucy filter are combined to design the LQG controller.

**How would you reconfigure your controller to asymptotically track a constant reference on x? Will your design reject constant force disturbances applied on the cart?**

To asymptotically track a constant reference on $x$. We would add a reference term and augment the system to reject constant force disturbances. Our model will not reject constant force disturbances due to error while doing the augmentation.

The Simulation of the LQG controller shows the following states:

$$x(t) - Position$$

$$\dot{x}(t) - Velocity$$

$$\theta_1(t) - Angle\ of\ mass\ 1$$

$$\dot{\theta_1}(t) - Angular\ velocity\ of\ mass\ 1$$

$$\theta_2(t) - Angle\ of\ mass\ 2$$

$$\dot{\theta}_2(t) - Angular\ velocity\ of\ mass\ 2$$

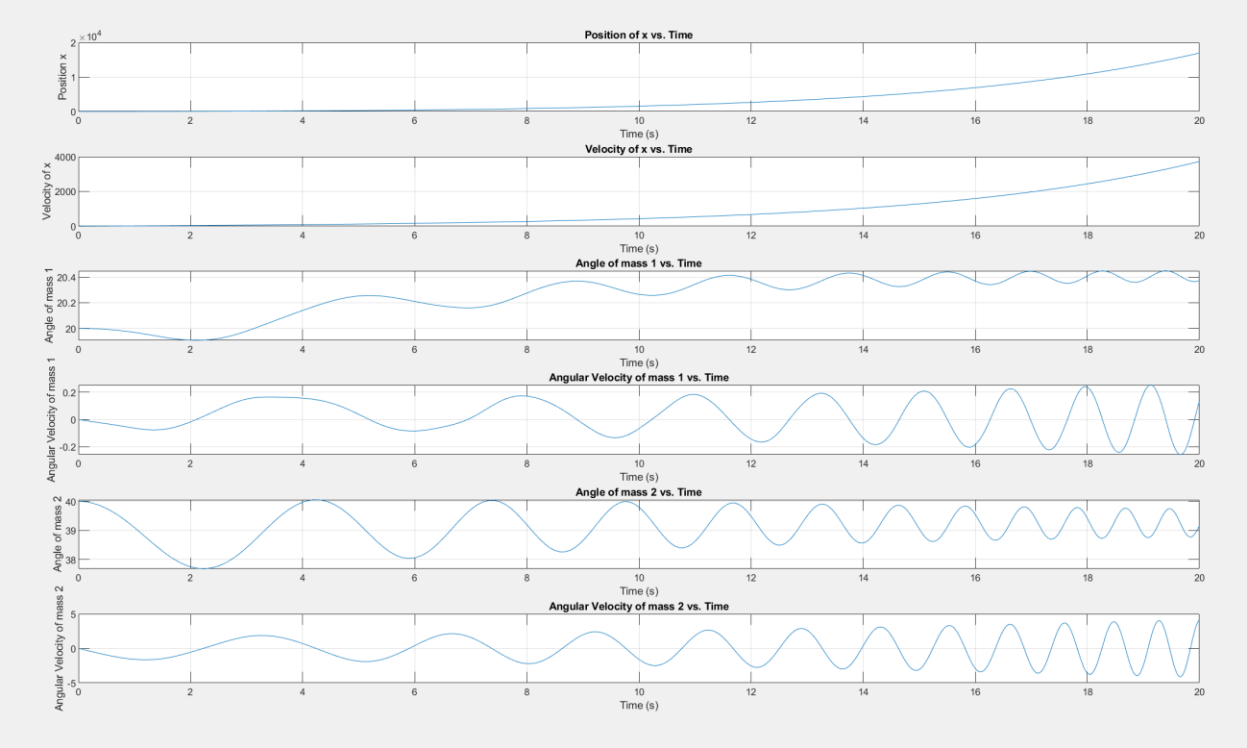



*Figure 9: LQR Simulation*

The goal is to reduce the damper of the system to attain stability.

# APPENDIX

## 1. PART -C CONTROLLABILITY

```
% Part_C_Controllability Conditions for Linearized System

function Part_C_Controllability

     % Initializing the matrix variables for matrix A and B
    syms M m1 m2 l1 l2 g;

    % Call the function to check controllability of the system
    Control_check(M, m1, m2, l1, l2, g);
end

function Control_check(M, m1, m2, l1, l2, g)
    % Constructing A and B matrices
    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 -((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)];

    % the Controllability Matrix
    Ctrl = [B A*B A^2*B A^3*B A^4*B A^5*B];

    % Displaying the properties of the controllability matrix
    disp_func(Ctrl, 'Controllability Matrix');

    % Adjusting for l1 = l2 to check for the condtion of controllability
    Ctrl_new = subs(Ctrl, l1, l2);
    disp_func(Ctrl_new, 'Controllability Matrix for l1 = l2');
end

function disp_func(matrix, title)
    % Displays rank and determinant of a matrix
    fprintf('\n%s:\n', title);
    disp(matrix);
    fprintf('Determinant: %s\n', char(det(matrix)));
    fprintf('Rank: %d\n', rank(matrix));
    % Check controllability
    if rank(matrix) == size(matrix, 1)
        disp('System is controllable due to rank = n');
    else
        disp('System is not controllable due to rank != n');
    end
end


```

## 2. PART -D LQR FOR LINEAR SYSTEM

```
% Part_D_LQR for linear system

% The Parameters given for system
m1=100; % in kg
m2=100; % in kg
M=1000; % in kg
l1=20; % in meters
l2=10; % in meters
g=9.81; % m/s
% implementing Matrices A and B
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 -((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)];
% The Initial Conditions for the system is given.
% x(t) = [x, x_dot, theta_1, theta_1_dot, theta_2, theta_2_dot].
x_initial = [0;0;20;0;40;0];
% The Initial Displacement is zero.
% The Values of Q and R are assumed.
Q=[20 0 0 0 0 0;
   0 200 0 0 0 0;
   0 0 1000 0 0 0;
   0 0 0 1000 0 0;
   0 0 0 0 200 0;
   0 0 0 0 0 20];
R=0.0005;
C = eye(6);% C = I, Which is found by using eye function.

% Considering Matrix D is zero.
D = 0;
% The Response of the system for the given initial conditions is
disp("The State Response for initial conditions:")
lin_system = ss(A,B,C,D);% The state space of the linearized system is
   calculated using inbuilt MATLAB function ss

% Inbuilt Functions in MATLAB are used for LQR controller
disp(" Linear System Output: ")
[K, P, Poles] = lqr(A,B,Q,R);

disp("P :")
disp(P); % Print P matrix
disp("K :")
disp(K); % Print K matrix
disp("Poles:")
disp(Poles); % Print Poles of the system

% The Response of the system when A = (A-BK)x for closed loop system
disp("The State Space Response for closed loop system" )
A1 = A-(B*K);
```

```
55.
56. % Implementing LQR controller for lin_system1
57. disp(" Linear System Output: ")
58. [K, P, Poles] = lqr(A1,B,Q,R);
59.
60. disp("P :  ")
61. disp(P); % Print P matrix for second system
62. disp("K :  ")
63. disp(K); % Print K matrix for second system
64. disp("Poles:  ")
65. disp(Poles); % Print Poles of the second system
66.
67.
68. lin_system1 = ss(A1,B,C,D);
69. figure
70. initial(lin_system1,x_initial)
71. grid on
```

## 3. PART -D LQR FOR NON LINEAR SYSTEM

```
4. % Part_D_LQR for non-linear system
5.
6. %Defining the output intial variables
7. y_intial = [0; 0; 20; 0; 40; 0];
8. tspan = 0:0.01:4000;% tspan is the time span for th system
9. [t_new,y_new] = ode45(@nonlinear,tspan,y_intial); %using ode45 function for
   non-linear systems
10. plot(t_new,y_new)
11. grid on
12.
13. % Defining non-linear function to call during the ode45 function
14. function dydt = nonlinear(t,y)
15.
16. m1=100; % in kg
17. m2=100; % in kg
18. M=1000; % in kg
19. l1=20; % in meters
20. l2=10; % in meters
21. g=9.81; % m/s
22.
23. A=[0 1 0 0 0 0;
24.     0 0 -(m1*g)/M 0 -(m2*g)/M 0;
25.     0 0 0 1 0 0;
26.     0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
27.     0 0 0 0 0 1;
28.     0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
29. B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
30.
31. Q=[20 0 0 0 0 0;
32.    0 200 0 0 0 0;
33.    0 0 1000 0 0 0;
34.    0 0 0 1000 0 0;
35.    0 0 0 0 200 0;
```

```
   0 0 0 0 0 20];
R=0.0005;

[K, P, Poles] = lqr(A,B,Q,R); % using in built function for lqr controlller

disp("P:")
disp(P);

disp("K:")
disp(K);

disp("Poles:")
disp(Poles);

% Using the optimal gain matrix K, to get the force for the equation to
% find state feedback
F=-K*y;

dydt=zeros(6,1);
dydt(1) = y(2); %XD
dydt(2)=(F-(g/2)*(m1*sind(2*y(3))+m2*sind(2*y(5)))-
   (m1*l1*(y(4)^2)*sind(y(3)))-
   (m2*l2*(y(6)^2)*sind(y(5))))/(M+m1*((sind(y(3)))^2)+m2*((sind(y(5)))^2));%xDD
dydt(3)= y(4); %theta 1D
dydt(4)= (dydt(2)*cosd(y(3))-g*(sind(y(3))))/l1'; %theta 1 Ddot;
dydt(5)= y(6); %theta 2D
dydt(6)= (dydt(2)*cosd(y(5))-g*(sind(y(5))))/l2; %theta 2Ddot;
end
```

## 4. PART -E OBSERVABILITY

```
% Part_E_Observability Assessment

function Part_E_Obervability
    % State-Space Matrices

    m1=100; % in kg
    m2=100; % in kg
    M=1000; % in kg
    l1=20; % in meters
    l2=10; % in meters
    g=9.81; % m/s

    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)];
    C1 = [1 0 0 0 0 0];% C1 matrix to monitor x(t)
    C2 = [0 0 1 0 0 0; 0 0 0 0 1 0];% C2 matrix to monitor theta1(t),
  theta2(t)
```

```
    C3 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 1 0];% C3 matrix to monitor x(t), theta2(t)
    C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];% C4 matrix to monitor x(t), theta1(t), theta2(t)

    n = 6 ; % n rows in the A matrix which is 6 in this case

    % Compute Observability Ranks
    Observ_rank1 = ObservabilityRankCalc(A, C1);
    Observ_rank2 = ObservabilityRankCalc(A, C2);
    Observ_rank3 = ObservabilityRankCalc(A, C3);
    Observ_rank4 = ObservabilityRankCalc(A, C4);

    % Display Results
    fprintf('Rank of observability matrix when monitoring x(t) = %d\n', Observ_rank1);
    fprintf('Rank of observability matrix when monitoring theta1(t), theta2(t) = %d\n', Observ_rank2);
    fprintf('Rank of observability matrix when monitoring x(t) and theta2(t) = %d\n', Observ_rank3);
    fprintf('Rank of observability matrix  when monitoring x(t),theta1(t) and theta2(t) = %d\n', Observ_rank4);


    % Check and Display Observability
    Observ_check(Observ_rank1, n, 'For case 1 x(t)');
    Observ_check(Observ_rank2, n, 'For case 2 theta1(t), theta2(t)');
    Observ_check(Observ_rank3, n, 'For case 3 x(t), theta2(t)');
    Observ_check(Observ_rank4, n, 'For case 4 x(t), theta1(t), theta2(t)');
end


function Observ_Rank = ObservabilityRankCalc(A, C)
    % Computes the rank of the observability matrix for a given system
    Observ_Rank = rank([C' A'*C' A'^2*C' A'^3*C' A'^4*C' A'^5*C']);
end

function Observ_check(check_obs, n, result)
    % Checks and prints if a system is observable
    if check_obs == n
        fprintf('%s is observable.\n', result);
    else
        fprintf('%s is not observable.\n', result);
    end
end
```

## 5. PART -F Luenberger observer linear system

```
% Part_F_Luenberger_observer_linear_system


% Define System Parameters
m1=100; % in kg
```

```
m2=100; % in kg
M=1000; % in kg
l1=20; % in meters
l2=10; % in meters
g=9.81; % m/s

% State-Space Matrices
A = [0 1 0 0 0 0;
     0 0 -(m1*g)/M 0 -(m2*g)/M 0;
     0 0 0 1 0 0;
     0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

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

% Output Matrices for Observable Cases
C1 = [1 0 0 0 0 0];% C1 matrix to monitor x(t)
C3 = [1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 1 0];% C3 matrix to monitor x(t),
   theta2(t)
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];% C4 matrix to monitor x(t),
   theta1(t), theta2(t)
% Lqr
Q = diag([20 200 1000 1000 200 20]);%Used diag function to mention the
   diagonal components in the Q matrix.
R = 0.0005;
K = lqr(A,B,Q,R); % using in built function for lqr controlller

% Design of Luenberger Observer
poles = -1:-1:-6;
L1 = place(A', C1', poles)'; % Luenberger matrix L1 by pole placement
L3 = place(A', C3', poles)'; %Luenberger matrix L3
L4 = place(A', C4', poles)'; %Luenberger matrix L4

% Define Systems for Simulation
systems = {
    struct('A', A - B * K, 'B', B, 'C', C1, 'L', L1);
    struct('A', A - B * K, 'B', B, 'C', C3, 'L', L3);
    struct('A', A - B * K, 'B', B, 'C', C4, 'L', L4)
};

% Initial Conditions and 6 estimate values
x_init = [0, 0, 20, 0, 40, 0, 0, 0, 0, 0, 0, 0];

% loop for finding augumented matrix and simulations
for i = 1:length(systems)
    sys = systems{i};
    A_aug = [sys.A sys.B * K; zeros(size(sys.A)) sys.A - sys.L * sys.C];
    B_aug = [sys.B; zeros(size(sys.B))];
    C_aug = [sys.C zeros(size(sys.C))];
    D = 0;

    ss_sys = ss(A_aug, B_aug, C_aug, D);

    figure;
```

```
    subplot(1, 2, 1);
    initial(ss_sys, x_init);
    title(sprintf('Initial State Response for System %d', i));
    grid on;

    subplot(1, 2, 2);
    step(ss_sys);
    title(sprintf('Step State Response for System %d', i));
    grid on;
end

```

## 6. PART -F Luenberger observer for nonlinear system

```
% Part_F_Luenberger_observer_non-linear_system


% Define System Parameters
m1=100; % in kg
m2=100; % in kg
M=1000; % in kg
l1=20; % in meters
l2=10; % in meters
g=9.81; % m/s

% State-Space Matrices
A = [0 1 0 0 0 0;
     0 0 -(m1*g)/M 0 -(m2*g)/M 0;
     0 0 0 1 0 0;
     0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
     0 0 0 0 0 1;
     0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];

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

% Output Matrices for Observable Cases
C1 = [1 0 0 0 0 0];% C1 matrix to monitor x(t)
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0];% C3 matrix to monitor x(t), theta2(t)
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];% C4 matrix to monitor x(t),
   theta1(t), theta2(t)
% Lqr
Q = diag([10 200 2000 1000 200 10]);%Used diag function to mention the
   diagonal components in the Q matrix.
R = 0.05;
K = lqr(A,B,Q,R); % using in built function for lqr controlller

% Design of Luenberger Observer
poles = -1:-1:-6;
L1 = place(A', C1', poles)'; % Luenberger matrix L1 by pole placement
L3 = place(A', C3', poles)'; %Luenberger matrix L3
L4 = place(A', C4', poles)'; %Luenberger matrix L4

% Define Systems for Simulation
```

```
systems = {
    struct('A', A - B * K, 'B', B, 'C', C1, 'L', L1);
    struct('A', A - B * K, 'B', B, 'C', C3, 'L', L3);
    struct('A', A - B * K, 'B', B, 'C', C4, 'L', L4)
};

% Initial Conditions and 6 estimate values
x_init = [0, 0, 20, 0, 40, 0, 0, 0, 0, 0, 0, 0];

% loop for finding augumented matrix and plot simulations
for i = 1:length(systems)
    sys = systems{i};
    A_aug = [sys.A sys.B * K; zeros(size(sys.A)) sys.A - sys.L * sys.C];
    B_aug = [sys.B; zeros(size(sys.B))];
    C_aug = [sys.C zeros(size(sys.C))];
    D = 0;

    ss_sys = ss(A_aug, B_aug, C_aug, D);

    figure;
    initial(ss_sys, x_init);
    title(sprintf('Initial State Response for System %d', i));
    grid on;

    figure;
    step(ss_sys);
    title(sprintf('Step State Response for System %d', i));
    grid on;
end

```

## PART -G LQG CONTROLLER

```
% Part_G_LQG_Non-Linear

% Define System Parameters
M = 1000; % Mass of the base
m1 = 100; % Mass of pendulum 1
m2 = 100; % Mass of pendulum 2
l1 = 20;  % Length of pendulum 1
l2 = 10;  % Length of pendulum 2
g = 9.81; % Acceleration due to gravity

% State-space Matrices for the 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 -((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]; % A matrix

```

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

C = [1 0 0 0 0 0]; % Measurement matrix, measuring only position 'x'

% LQR Controller Design
Q = [20 0 0 0 0 0;
     0 200 0 0 0 0;
     0 0 1000 0 0 0;
     0 0 0 1000 0 0;
     0 0 0 0 200 0;
     0 0 0 0 0 20];
R = 0.005;
[K, ~, ~] = lqr(A, B, Q, R);

% Designing State Estimator using Kalman Filter
% Defining process noise covariance (W) and measurement noise covariance (V)
W = 0.1 * eye(6);
V = 0.1;
[L, ~, ~] = lqe(A, eye(6), C, W, V);

% Initial Conditions for the Nonlinear System
x_init = [0; 0; 20; 0; 40; 0]; % Initial state

% Time Span for Simulation
tspan = [0 20];
step_time = 1; % Time for step input

% Nonlinear System Simulation using LQG Controller
[t, y] = ode45(@(t, x) nonlinear_system_with_lqg(t, x, K, L, A, B, C, M, m1,
   m2, l1, l2, g, step_time, V), tspan, x_init);

% Extracting control input and noise from the output
control_input = y(:, 2);
noise_array = y(:, 6);

% Plot Results

% Plot Results
figure;
% Plotting the position 'x' over time
subplot(6,1,1);
plot(t, y(:, 1));
xlabel('Time (s)');
ylabel('Position x');
title('Position of x vs. Time');
grid on;

% Plotting other state variables
subplot(6,1,2);
plot(t, y(:, 2));
xlabel('Time (s)');
ylabel('Velocity of x');
title('Velocity of x vs. Time');
grid on;

```

```
subplot(6,1,3);
plot(t, y(:, 3));
xlabel('Time (s)');
ylabel('Angle of mass 1');
title('Angle of mass 1 vs. Time');
grid on;

subplot(6,1,4);
plot(t, y(:, 4));
xlabel('Time (s)');
ylabel('Angular Velocity of mass 1');
title('Angular Velocity of mass 1 vs. Time');
grid on;

subplot(6,1,5);
plot(t, y(:, 5));
xlabel('Time (s)');
ylabel('Angle of mass 2');
title('Angle of mass 2 vs. Time');
    grid on;

    subplot(6,1,6);
    plot(t, y(:, 6));
    xlabel('Time (s)');
    ylabel('Angular Velocity of mass 2');
    title('Angular Velocity of mass 2 vs. Time');
    grid on;

    % Function defining nonlinear system dynamics with LQG controller
    function [diff, u, noise] = nonlinear_system_with_lqg(t, x, K, L, A, B,
C, M, m1, m2, l1, l2, g, step_time, V)
        y = C * x;
        x_hat = A * x + B * (-K * x) + L * (y - C * x);

        % Unit step input
        step_input = (t >= step_time);

        % Control input
        u = -K * x_hat + step_input;

        % Simulated noise
        noise = randn * sqrt(V); % Using the measurement noise covariance
'V'

        % System dynamics eqution of non linear system
        diff = zeros(6, 1);
        diff(1) = x(2);
        diff(2) = (u - (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);
        diff(3) = x(4);
        diff(4) = (diff(2)*cos(x(3)) - g*sin(x(3))) / l1;
        diff(5) = x(6);
        diff(6) = (diff(2)*cos(x(5)) - g*sin(x(5))) / l2;
    end
```