ENPM 667 Final Project Report

1st Abhishek Avhad
Department of Robotics
University of Maryland
College Park, MD, USA
abhi6@umd.edu

2nd Sachin Jadhav
Department of Robotics
University of Maryland
College Park, MD, USA
sjd3333@umd.edu

3rd Qamar Syed
Department of Robotics
University of Maryland
College Park, MD, USA
qsyed@umd.edu

I. Introduction

This report details the analysis of the physical system that can be seen in Figure 1, in which mass M moves along a one dimensional track as a result of force F, with two attached loads m_1 and m_2. Following the system analysis, both a controller and observer are designed for the system. In order to design the LQR and LQG controllers for the crane we follow the following steps:

- We will develop the equations of motion for the system using Lagrangian method.
- Next we linearize the system around an equilibrium point and write the state space representation of this linearized system.
- We then obtain controllability conditions based out of M1, m1, m2, l1, l2.
- The simulation responses are recorded for two scenarios by applying the LQR controller to the original non linear system and the linearized system. We simulate the responses by adjusting the LQR parameters until we get the suitable response and then we perform Lyapunov analysis of this closed loop system to certify the stability.
- After designing the output vector we check the systems observability for some output vectors.
- We determine the best Luenberger Observer for each of the output vectors of they are observable and then simulate the response to input conditions and unit step input when applied to the linearized and non linearized system.
- The final step is to design an output feedback controller for the smallest output vector using the LQG method and illustrate its performance in form of a simulation.

All code can be found in the appendix section and subsections with question part answers are highlighted.

II. System Analysis

A. Equations of Motion

The analysis of the system begins with deriving the equations of motion for the system components. We find position of mass m_1 and m_2 as as per the the current angles $theta_1$ & $theta_2$ respectively.

$$\mathbf{x}_{m_1} = (x - l_1 \sin(\theta_1))\mathbf{i} - (l_1 \cos(\theta_1))\mathbf{j} \tag{1}$$

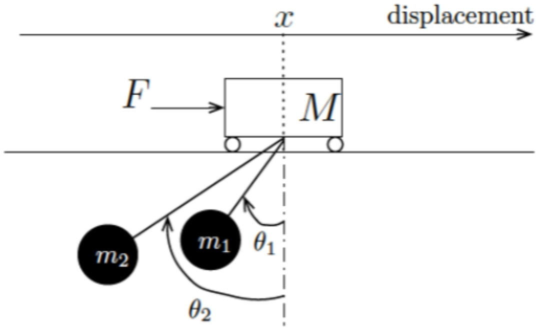


Fig. 1. Physical system subject to analysis.

Similarly,

$$\mathbf{x}_{m_2} = (x - l_2 \sin(\theta_2))\mathbf{i} - (l_2 \cos(\theta_2))\mathbf{j} \tag{2}$$

Differentiating the above equation w.r.t time gives us the velocity of m_1 and m_2 respectively:

$$\mathbf{v}_{m1} = (\dot{x} - l_1 \cos(\theta_1)\dot{\theta}_1)\mathbf{i} + l_1 \sin(\theta_1)\dot{\theta}_1\mathbf{j} \tag{3}$$

$$\mathbf{v}_{m2} = (\dot{x} - l_2 \cos(\theta_2)\dot{\theta}_2)\mathbf{i} + l_2 \sin(\theta_2)\dot{\theta}_2\mathbf{j} \tag{4}$$

As the next step, we can calculate the Kinetic energy (K.E) of the system as follows:
K.E system = (K.E) of M + (K.E) of m_1 + (K.E) of m_2

$$\begin{aligned}\text{K.E} = &\frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\left(\dot{x} - l_1\dot{\theta}_1\cos(\theta_1)\right)^2 \\ &+ \frac{1}{2}m_1(l_1\dot{\theta}_1\sin(\theta_1))^2 + \frac{1}{2}m_2\left(\dot{x} - \dot{\theta}_2 l_2\cos(\theta_2)\right)^2 \\ &+ \frac{1}{2}m_2(l_2\dot{\theta}_2\sin(\theta_2))^2\end{aligned}$$

Similarly, the Total Potential Energy is the sum of Potential energies of M, m_1 and m_2:
P.E system = (P.E) of M + (P.E) of m_1 + (P.E) of m_2

$$\begin{aligned}\text{P.E} &= -m_1 g l_1\cos(\theta_1) - m_2 g l_2\cos(\theta_2) \\ &= -g(m_1 l_1\cos(\theta_1) + m_2 l_2\cos(\theta_2))\end{aligned} \tag{5}$$

We know write the Langrange's Equation i.e.

$$\begin{aligned} L &= \text{K.E} - \text{P.E} \\ &= \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\left(\dot{x} - l_1\dot{\theta}_1\cos(\theta_1)\right)^2 \\ &\quad + \frac{1}{2}m_1(l_1\dot{\theta}_1\sin(\theta_1))^2 + \frac{1}{2}m_2\left(\dot{x} - \dot{\theta}_2 l_2\cos(\theta_2)\right)^2 \\ &\quad + \frac{1}{2}m_2(l_2\dot{\theta}_2\sin(\theta_2))^2 \\ &\quad - (-g(m_1 l_1\cos(\theta_1) + m_2 l_2\cos(\theta_2))) \end{aligned}$$

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

Simplifying the above equation further we obtain:

$$\begin{aligned} \mathcal{L} =& \frac{1}{2}M\dot{x}^2 + \frac{1}{2}(m_1 + m_2)\dot{x}^2 + \frac{1}{2}m_1 l_1^2\dot{\theta}_1^2 \\ & + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2 - \dot{x}(m_1 l_1\dot{\theta}_1\cos(\theta_1) + m_2 l_2\dot{\theta}_2\cos(\theta_2)) \\ & + g(m_1 l_1\cos(\theta_1) + m_2 l_2\cos(\theta_2)) \end{aligned} \tag{6}$$

Using Euler- Lagrange's method to calculate the equations of the motion for the system:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right) - \frac{\partial\mathcal{L}}{\partial x} = F \tag{7}$$

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right) - \frac{\partial\mathcal{L}}{\partial\theta_1} = 0 \tag{8}$$

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_2}\right) - \frac{\partial\mathcal{L}}{\partial\theta_2} = 0 \tag{9}$$

Considering equation 7:
First, calculating the partial differential of L w.r.t $\dot{x}$, we get:

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

Differentiating this equation w.r.t time, we obtain:

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

In equation 7 the differential of L w.r.t x is zero:

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

Substituting these values in 7 we obtain :

$$\begin{aligned} (M + m_1 + m_2)\ddot{x} - m_1 l_1\ddot{\theta}_1\cos(\theta_1) \\ + m_1 l_1\dot{\theta}_1^2\sin(\theta_1) - m_2 l_2\ddot{\theta}_2\cos(\theta_2) \\ + m_2 l_2\dot{\theta}_2^2\sin(\theta_2) = F \end{aligned} \tag{10}$$

Now, obtaining equations for 8:

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

Calculating the partial differential of L w.r.t $\dot{\theta}_1$, we get:

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

Differentiating this equation w.r.t time, we obtain:

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

In equation 8 the differential of L w.r.t θ_1:

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

Inserting back in 8, we obtain the equations as:

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

Solving further we obtain:

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

i.e

$$\boldsymbol{m_1 l_1^2\ddot{\theta}_1 - m_1\ddot{x}_1\cos(\theta_1) + m_1 l_1 g\sin(\theta_1) = 0} \tag{11}$$

Now for equation 9:

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

Calculating the partial differential of L w.r.t $\dot{\theta}_2$, we get:

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

Differentiating this equation w.r.t time, we obtain:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right) = m_2 l_2\ddot{\theta}_2 - [m_2 l_2\ddot{x}\cos(\theta_1) - m_2\dot{x}l_2\dot{\theta}_2\sin(\theta_2)]$$

In equation 8 the differential of L w.r.t θ_2:

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

Solving further we obtain:

$$\begin{aligned} m_2 l_2^2\ddot{\theta}_2 - m_2\ddot{x}l_2\cos(\theta_2) + m_2\dot{\theta}_2\dot{x}l_2\sin(\theta_2) \\ - m_2\dot{\theta}_2\dot{x}l_2\sin(\theta_2) + m_2 l_2 g\sin(\theta_2) = 0 \end{aligned}$$

$$\boldsymbol{m_2 l_2^2\ddot{\theta}_2 - m_2\ddot{x}_2\cos(\theta_2) + m_2 l_2 g\sin(\theta_2) = 0} \tag{12}$$

Inserting the values of $\ddot{\theta}_1$ and $\ddot{\theta}_2$ in the equation for $\ddot{x}$:

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

Fig. 2. $\ddot{x}$

$$\ddot{\theta}_1 = l_1\left[\ddot{x}\cos(\theta_1) - g\sin(\theta_1)\right] \tag{13}$$

$$\ddot{\theta}_2 = l_1\left[\ddot{x}\cos(\theta_2) - g\sin(\theta_2)\right] \tag{14}$$

Compiling the values of $\ddot{x}$, $\ddot{\theta}_1$, $\ddot{\theta}_2$ we write the state space model obtained after inserting the value of $\ddot{x}$ in $\ddot{\theta}_1$ & $\ddot{\theta}_2$

Inserting the values of $\ddot{\theta}_1$ and $\ddot{\theta}_2$ in the equation for $\ddot{x}$:

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

Fig. 3. State space model equation

B. Nonlinear State Space Representation

From these equations of motion, a nonlinear state space can be derived.

III. System Linearization

We use Jacobian linearization around an equilibrium point to linearize the system. Equilibrium point is given by x = 0 and $\theta_1 = \theta_2 = 0$. The Jacobians are given by:

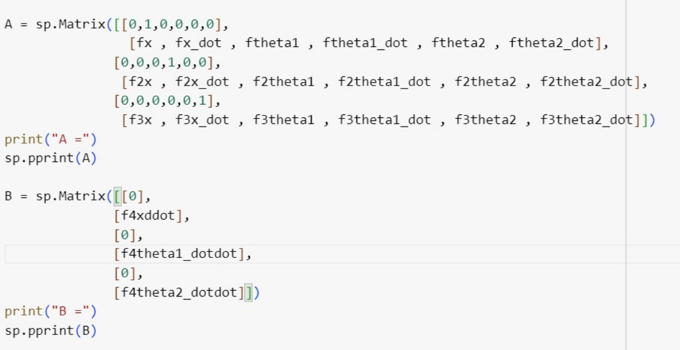
$$A = \begin{bmatrix} \frac{\delta f_1}{\delta X_1} & \frac{\delta f_1}{\delta X_2} & \frac{\delta f_1}{\delta X_3} & \frac{\delta f_1}{\delta X_4} & \frac{\delta f_1}{\delta X_5} & \frac{\delta f_1}{\delta X_6} \\ \frac{\delta f_2}{\delta X_1} & \frac{\delta f_2}{\delta X_2} & \frac{\delta f_2}{\delta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

Fig. 4. Standard Jacobian A

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

Fig. 5. Standard Jacobian B

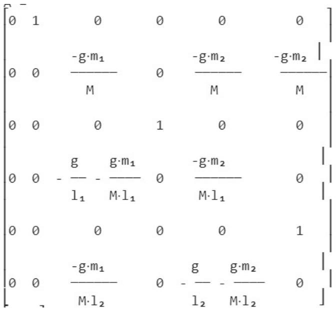
Now using Python Sympy library to get the Jacobian we obtained code for linearization and it is shown in Figure 6.



```
A = sp.Matrix([[0,1,0,0,0,0],
                [fx , fx_dot , ftheta1 , ftheta1_dot , ftheta2 , ftheta2_dot],
               [0,0,0,1,0,0],
               [f2x , f2x_dot , f2theta1 , f2theta1_dot , f2theta2 , f2theta2_dot],
               [0,0,0,0,0,1],
               [f3x , f3x_dot , f3theta1 , f3theta1_dot , f3theta2 , f3theta2_dot]])
print("A =")
sp.pprint(A)

B = sp.Matrix([[0],
               [f4xddot],
               [0],
               [f4theta1_dotdot],
               [0],
               [f4theta2_dotdot]])
print("B =")
sp.pprint(B)
```

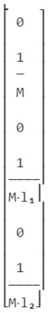
Fig. 6. Code for Linearisation

We obtain matrices A and B in Figures 7 and 8 as following after running the python code:



```
⎡0  1      0        0      0        0   ⎤
⎢                                       ⎥
⎢        -g⋅m₁            -g⋅m₂    -g⋅m₂ ⎥
⎢0  0    ──────     0     ──────   ──────⎥
⎢          M                M        M  ⎥
⎢                                       ⎥
⎢0  0      0        1      0        0   ⎥
⎢                                       ⎥
⎢        g    g⋅m₁        -g⋅m₂         ⎥
⎢0  0  - ── - ────  0     ──────    0   ⎥
⎢        l₁   M⋅l₁         M⋅l₁         ⎥
⎢                                       ⎥
⎢0  0      0        0      0        1   ⎥
⎢                                       ⎥
⎢        -g⋅m₁           g    g⋅m₂      ⎥
⎢0  0    ──────     0  - ── - ────  0   ⎥
⎣         M⋅l₂           l₂   M⋅l₂      ⎦
```

Fig. 7. A matrix



$$\begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M \cdot l_1} \\ 0 \\ \frac{1}{M \cdot l_2} \end{bmatrix}$$

Fig. 8. B matrix

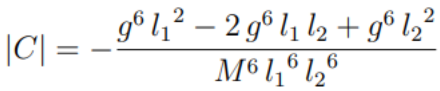
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = A \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + Bu \qquad (15)$$

IV. Controllability Analysis

A system is deemed controllable if its grammian can be inverted. To assess invertibility, one can analyze the Controllability Matrix's rank. If the rank matches the number of state space variables, it signifies the invertibility of the grammian, establishing the system as controllable. It is represented by:

$$C = rank[B, AB, A^2B, A^3B, A^4B] = n \qquad (16)$$

In the context of the linear state space representation of the system, matrices A and B are derived. The number of state space variables, denoted as n, is 6 in this scenario. Upon evaluating the determinant of the parameterized controllability matrix C. Resulting determinant is as expressed below which can be referred in the code.



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

Fig. 9. Paratmetrized controllability matrix C determinant

For the system to achieve controllability, the Controllability Matrix (C) must have full rank (rank = 6). Consequently, the determinant mentioned above should not be zero. Thus, it can be concluded from the given equation that the system is controllable when these conditions are simultaneously met,

$$M, m_1, m_2 > 0 \quad \text{and} \quad l_1 \neq l_2 \qquad (17)$$

V. Controller Design

Given $M = 1000\,\text{kg}$, $m_1 = m_2 = 100\,\text{kg}$, $l_1 = 20\,\text{m}$, and $l_2 = 10\,\text{m}$. In this case, we can see that l_1 is not equal to l_2. Hence, from the condition we obtained in section C, we can say that the system is controllable. Since the rank of the controllability matrix is 6 given that l_1 is not equal to l_2. The system is Controllable.

The gain matrix of the LQR controller is given by:

$$K = R^{-1}B^TP$$

. After extracting the data for Q, R and K we get the state space model as : The state-space model with a feedback matrix K is given by:

$$\dot{x} = (A - BK)x + Bu$$
$$y = Cx + Du$$

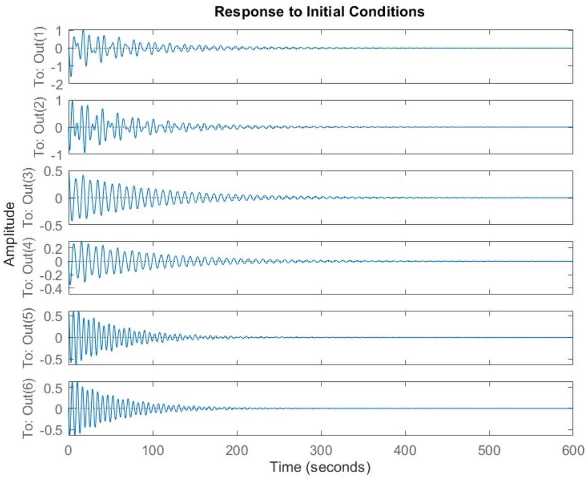


Fig. 10.

This was calculated in the LQR code in order to design the controllers. The outputs can be seen in Figures 9 and 10. Response to the initial conditions for the controller applied to the linearized system is shown in Figure 9. Response to initial conditions when the controller is applied to the non linearized system can be seen in Figure 10.

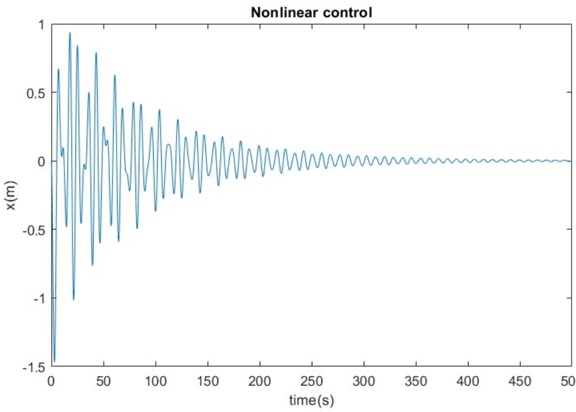


Fig. 11. Response for Non-linearised control

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

VI. Observability Analysis

Observability for the system was analyzed for four different sets of outputs: $x(t)$, $(\theta_1(t), \theta_2(t))$, $(x(t), \theta_2(t))$, and $(x(t), \theta_1(t), \theta_2(t))$. These outputs, in the order written, are obtained using the following C matrices, when using the state vector:

$$C_1 = [1\ 0\ 0\ 0\ 0\ 0]$$

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

$$C_3 = \begin{bmatrix} 1\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 1\ 0 \end{bmatrix} \tag{18}$$

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

A system is determine to be observable if the following equation holds:

$$rank(R_{nxnm}) = n \tag{19}$$

in which the controllability matrix R, defined in the equation below and constructed from the nxn state matrix A and mxn output matrix R, is required to be full rank.

$$R = [C^T\ A^TC^T\ {A^2}^TC^T\ ...\ {A^{n-1}}^TC^T] \tag{20}$$

Since A has already been defined as a $6x6$ matrix, the following equation derived from the two above must be true to guarantee observability for any selected C:

$$rank([C^T\ A^TC^T\ {A^2}^TC^T\ {A^3}^TC^T\ {A^4}^TC^T\ {A^5}^TC^T]) = 6 \tag{21}$$

This condition was checked for each C_n in a MATLAB script and it was determined that of the given output sets and correlating C_n matrices, the system is observable for matrices C_1, C_3, C_4 as their controllability matrix is full rank. The system is not observable for output matrix C_2 as the rank for its controllability matrix was found to be 4.

VII. Observer Design

From the previously determined observable systems, a Luenberger Observer can be designed. This has the following state-space representation:

$$\dot{\hat{\vec{X}}}(t) = A\hat{\vec{X}}(t) + B_K\vec{U_K}(t) + L(\vec{Y}(t) - C\hat{\vec{X}}(t)) \tag{22}$$
$$\hat{\vec{X}}(0) = 0$$

In which $\hat{\vec{X}}$ represents the state estimate from the observer and L is the observer gain matrix. L is a gain applied to the $\vec{Y}(t) - C\hat{\vec{X}}(t)$ term, which is the correction term in the system, and compares the actual output to the output using the estimated state.

The behavior of the error of the state estimate, $\vec{X}_e$, is determined by the term $A - LC$ and the process noise $B_D\vec{U_d}$ in the following equation:

$$\dot{\hat{\vec{X}}}_e = (A - LC)\hat{\vec{X}}_e + B_D\vec{U_d} \tag{23}$$

As A and C are already determined, the L matrix is designed to ensure observer performance. As the observer should be significantly faster at detecting the system than the actual system dynamics, L is selected so that the magnitude of the poles of the observer are significantly larger than that of the system. In this case, this is done via a MATLAB script.

After solving for L, the closed-loop system can be set up using the following equations:

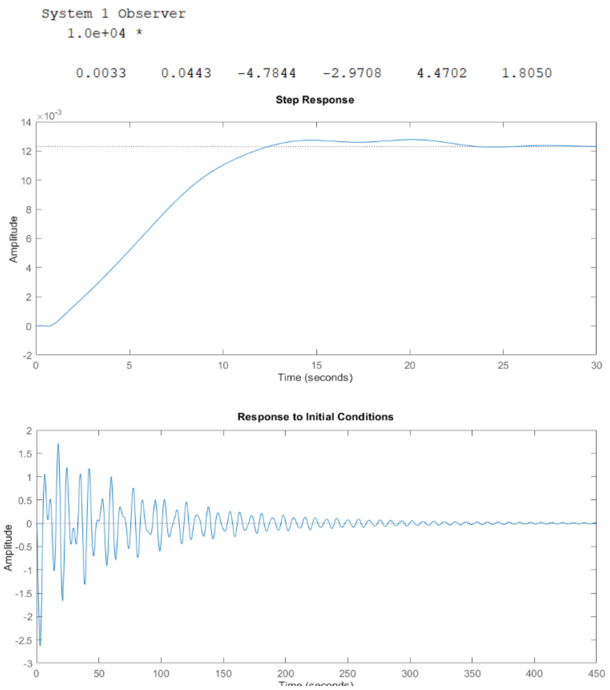
$$[\dot{\vec{X}}(t)\ \dot{\vec{X}}_e(t)]^T = A_c[\vec{X}(t)\ \vec{X}_e(t)]^T + [B_d\ B_d]^T\vec{U_d}(t) \tag{24}$$

in which A_c is defined as:

$$A_c = \begin{bmatrix} A + B_KK & -B_KK \\ 0 & A - LC \end{bmatrix} \tag{25}$$

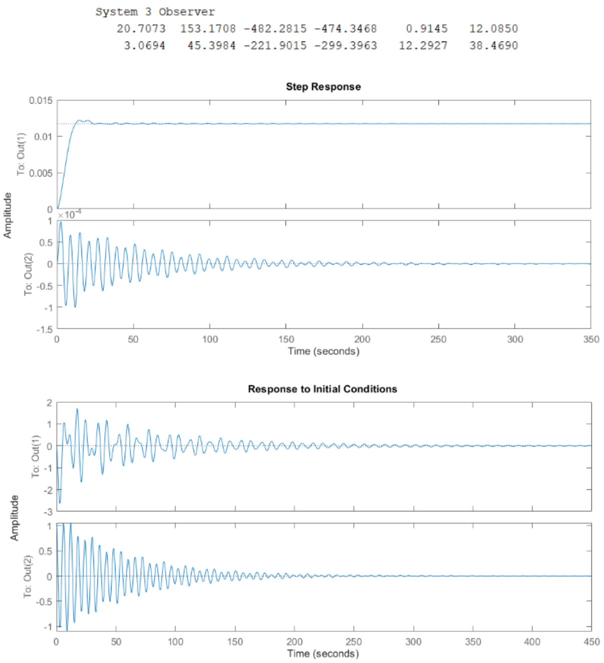
A. Linearized System Observer Results

For each of the system output sets determined to be observable from the Observability Analysis section, a Luenberger observer was designed using the process described above in a MATLAB script. The following figures show the L matrix, step response and initial state response for the observed system.



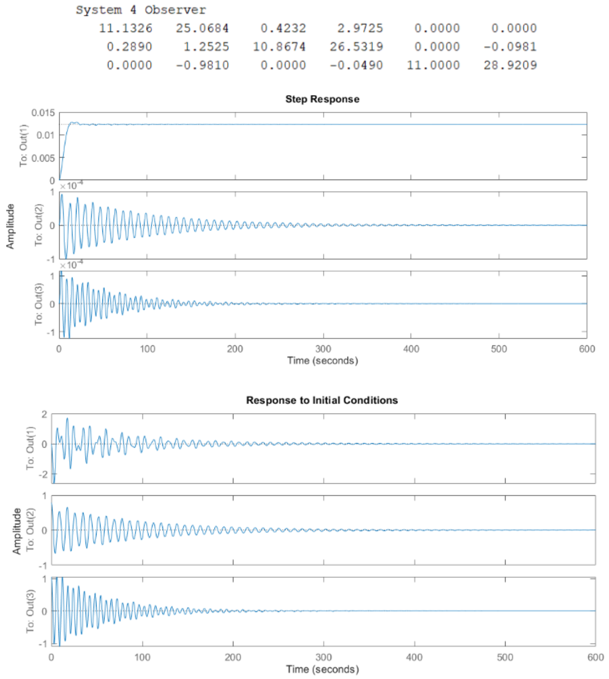
```
System 1 Observer
   1.0e+04 *

    0.0033    0.0443   -4.7844   -2.9708    4.4702    1.8050
```

Fig. 12. System 1 linearized observer



```
System 3 Observer
   20.7073  153.1708 -482.2815 -474.3468    0.9145   12.0850
    3.0694   45.3984 -221.9015 -299.3963   12.2927   38.4690
```

Fig. 13. System 3 linearized observer



```
System 4 Observer
   11.1326   25.0684    0.4232    2.9725    0.0000    0.0000
    0.2890    1.2525   10.8674   26.5319    0.0000   -0.0981
    0.0000   -0.9810    0.0000   -0.0490   11.0000   28.9209
```

Fig. 14. System 4 linearized observer

B. *Nonlinear System Observer Results*

The observer system was also set up using the original nonlinear system equations and MATLAB's nonlinear equation solver. The following figures show the initial state response for each nonlinear observed system.

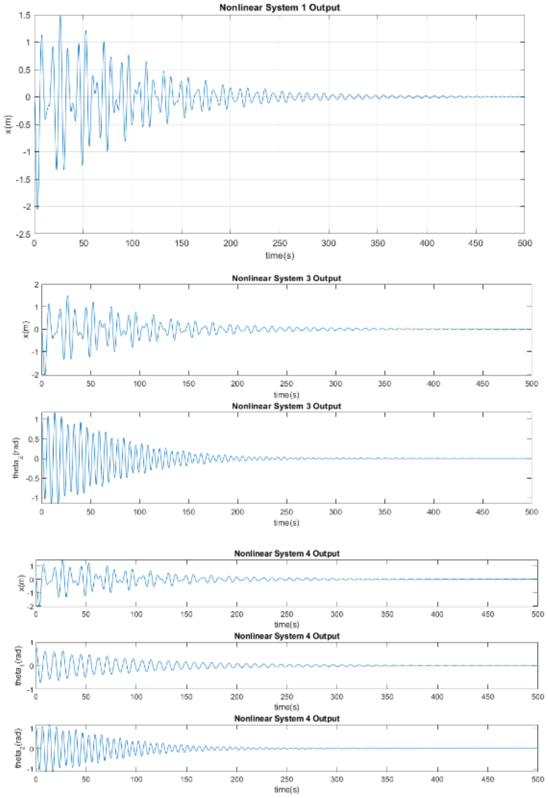


Fig. 15. Nonlinear system responses

VIII. Output Feedback Controller

The output feedback controller is designed using the Linear Quadratic Gaussian Method (LQG), the state-space system of which is represented in the equation below:

$$\begin{aligned}\dot{\vec{X}}(t) &= A\vec{X}(t)+B_K\vec{U_K}(t)+B_D\vec{U_D}(t)\\ \vec{Y}(t) &= C\vec{X}(t)+\vec{V}(t)\end{aligned} \tag{26}$$

Where $\vec{U_D}(t)$ represents the process noise and $\vec{V}(t)$ represents the measurement noise. These are both defined as independent zero mean white Gaussian processes. The system is designed by attempting to minimize the following cost equation:

$$lim_{t\to\infty}E[\vec{X}(t)^TQ\vec{X}(t)+\vec{U}(t)^TR\vec{U}(t)] \tag{27}$$

The way this is done is by separately calculating K and L using LQR and Kalman-Bucy methods, respectively.

A. *Output Feedback Controller Results*

An output feedback controller was designed for $x(t)$ using a MATLAB script shown that made use of LQR and Kalman functions in order to calculate K and L. Noise values were selected arbitrarily and added to the system. The figure below shows the results of the controller when the state is given the same initial state it was in the observer section, which is an approximation of the initial system image. Although the output seems similar to that of the Luenberger observer, this shows the LQG controller is effective, as it has noise added that those systems did not. The system successfully is able to apply similar controls as a Luenberger system with no noise.

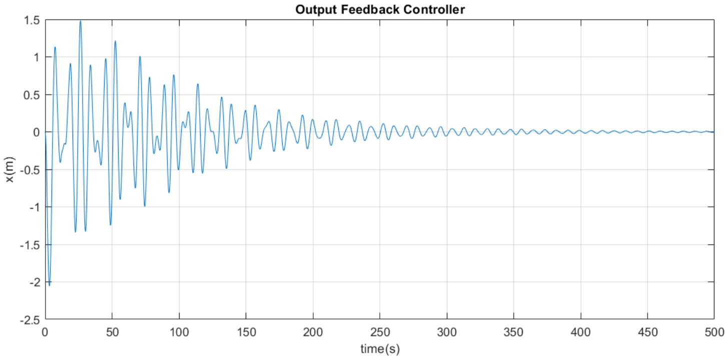


Fig. 16. Output feedback controller simulation

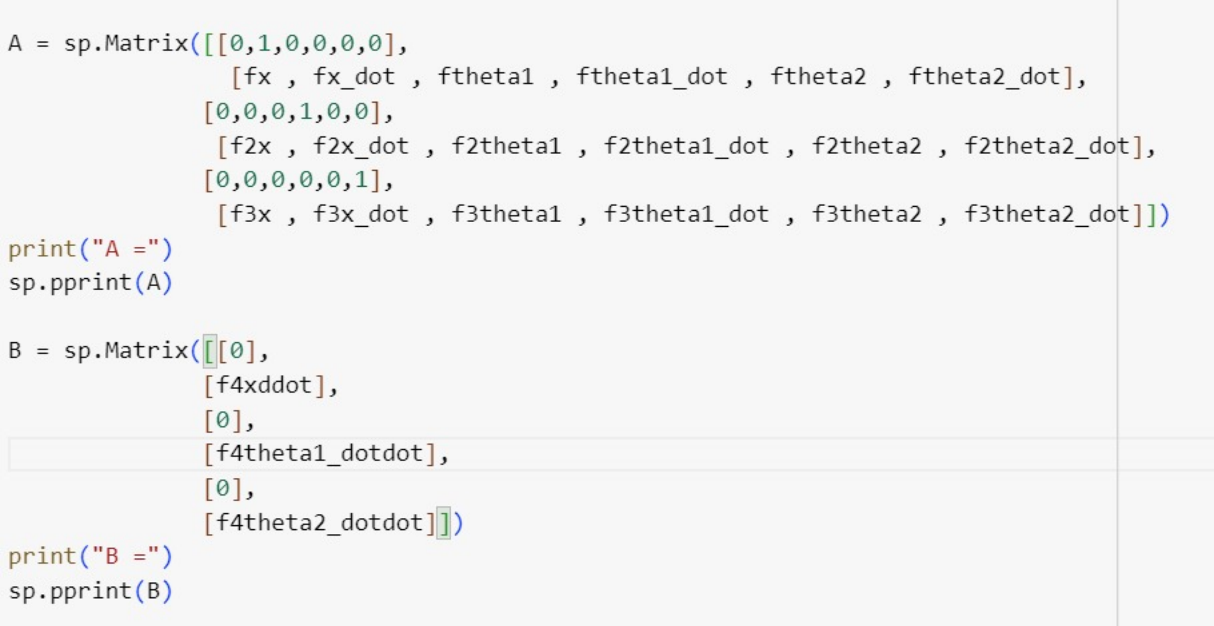
As stated above, and LQG controller contains LQR and Kalman-Bucy methods. In the case of constant reference tracking, the responsibility of this falls onto the LQR controller. The cost function for LQR optimal reference tracking, found in the lecture notes, is as follows:

$$\int_0^\infty (\vec{X}(t) - \vec{X_d})^T Q(\vec{X}(t) - \vec{X_d}) + (\vec{U_K}(t) - \vec{U_\infty})^T R(\vec{U_K}(t) - \vec{U_\infty})\, dt \tag{28}$$

The response of the output feedback controller from constant force disturbances is conditional. As long as they meet the conditions for disturbances that the system is designed to handle (independent zero mean white Gaussian processes), then the system should be able to handle them.

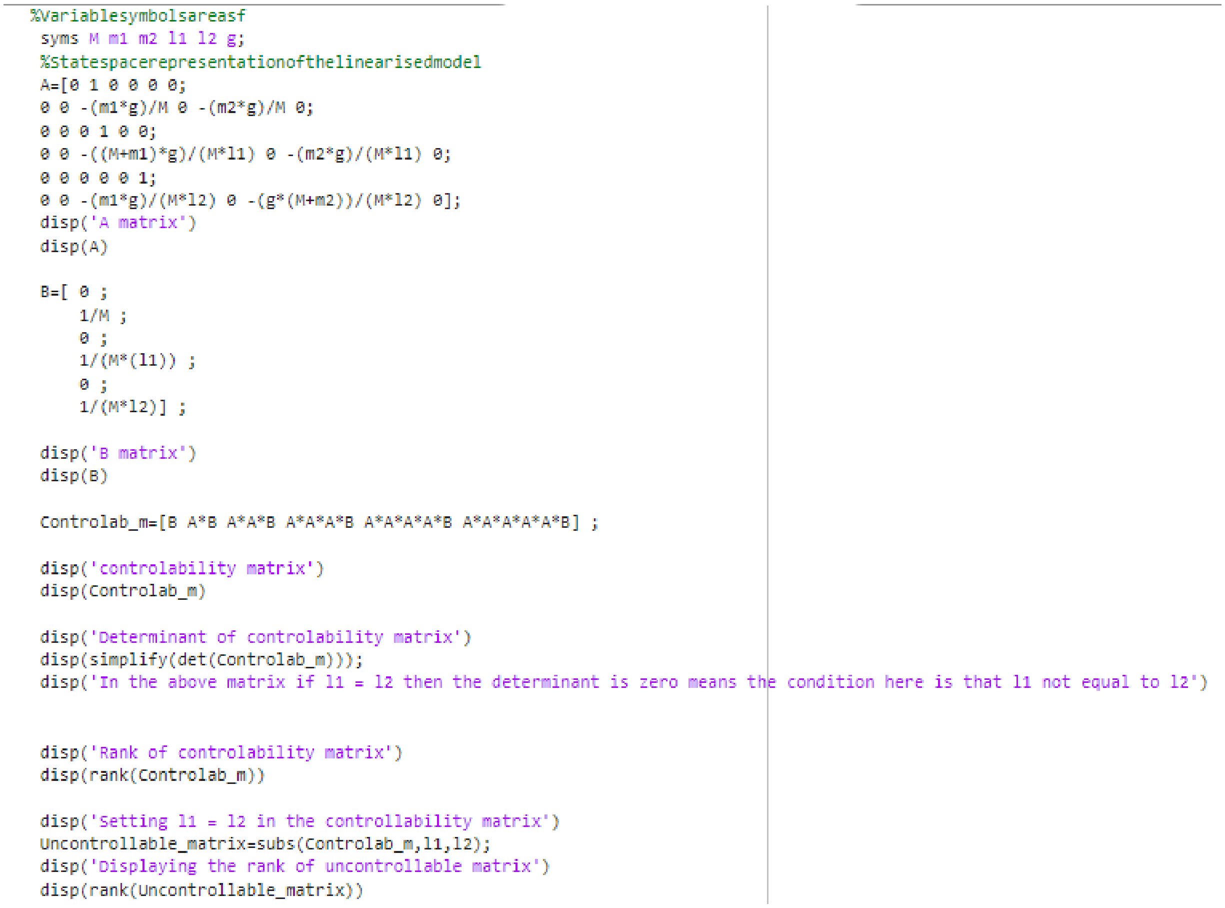
IX. Appendix

A. System Linearization Code



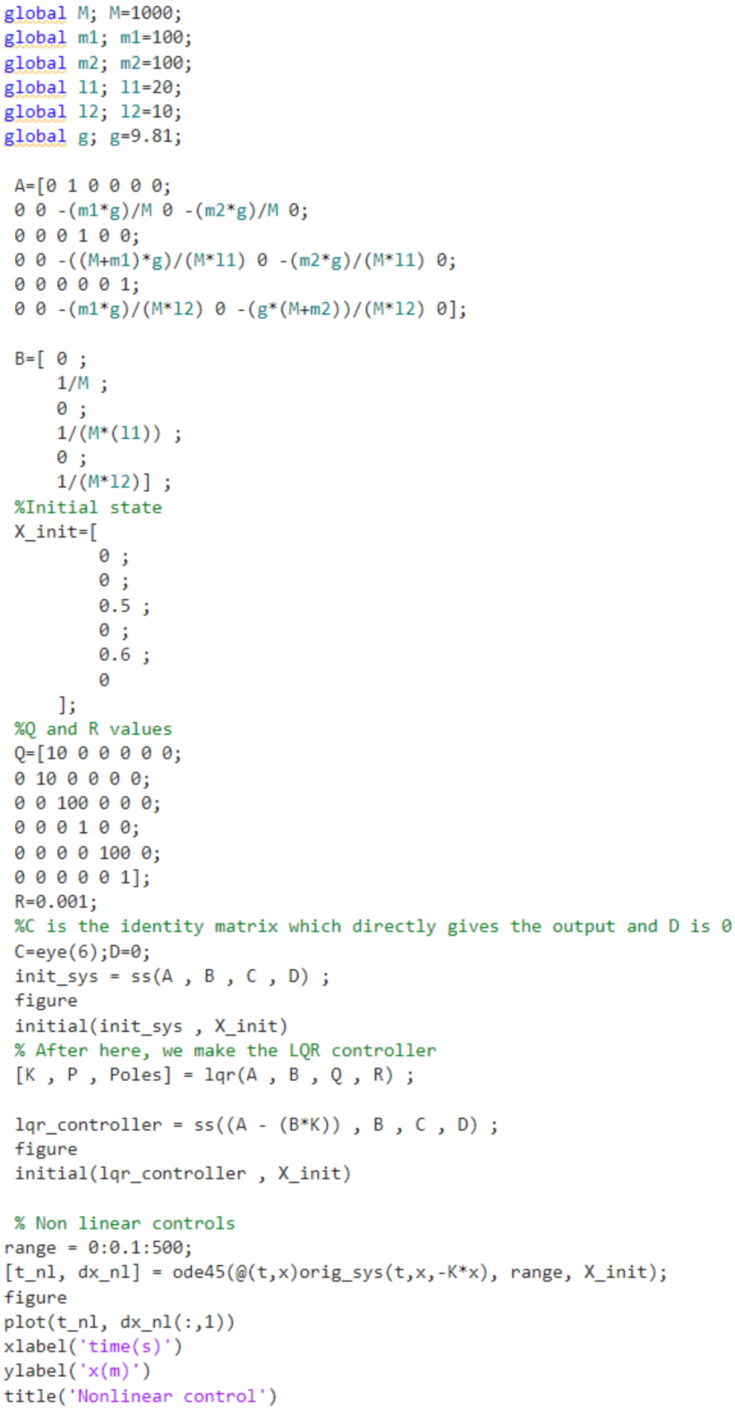
```
A = sp.Matrix([[0,1,0,0,0,0],
                [fx , fx_dot , ftheta1 , ftheta1_dot , ftheta2 , ftheta2_dot],
              [0,0,0,1,0,0],
               [f2x , f2x_dot , f2theta1 , f2theta1_dot , f2theta2 , f2theta2_dot],
              [0,0,0,0,0,1],
               [f3x , f3x_dot , f3theta1 , f3theta1_dot , f3theta2 , f3theta2_dot]])
print("A =")
sp.pprint(A)

B = sp.Matrix([[0],
              [f4xddot],
              [0],
              [f4theta1_dotdot],
              [0],
              [f4theta2_dotdot]])
print("B =")
sp.pprint(B)
```

B. Controllability Analysis Code



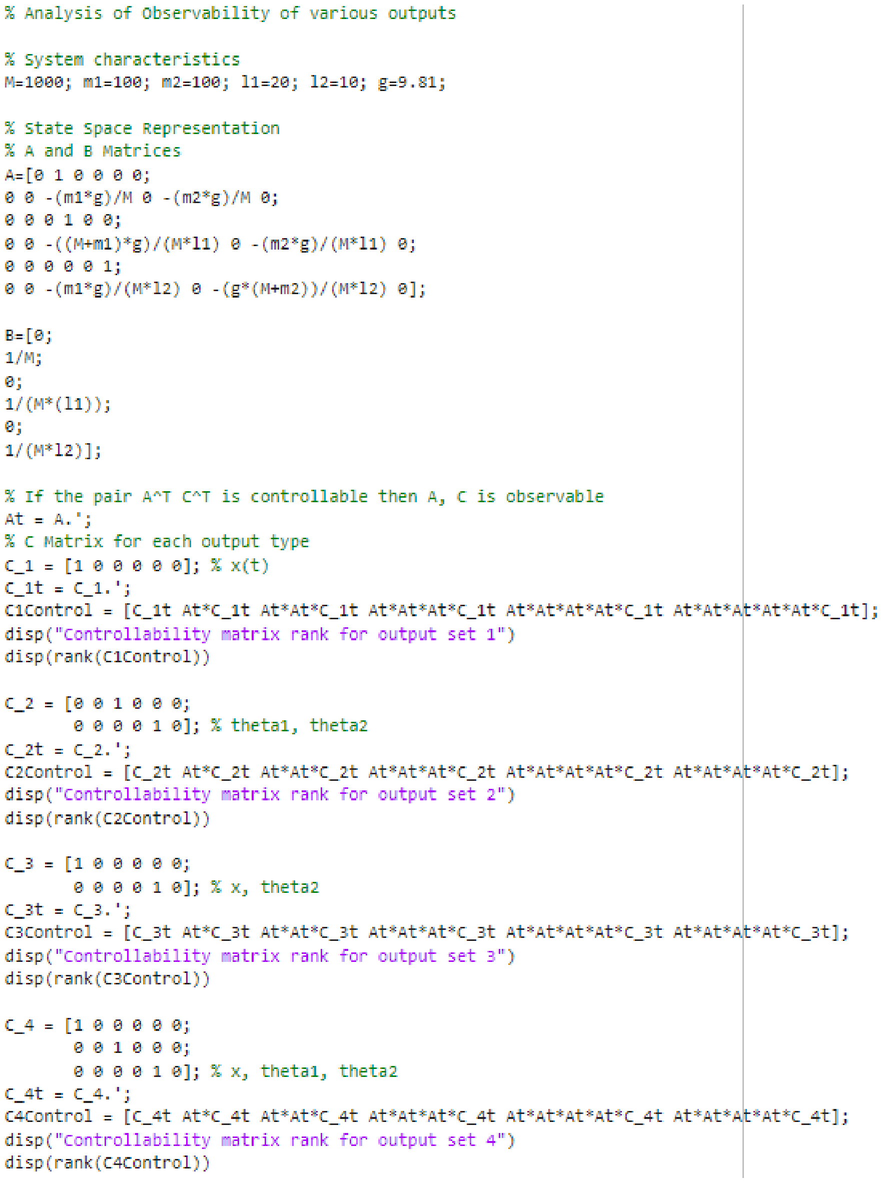
```
%Variablesymbolsareasf
 syms M m1 m2 l1 l2 g;
 %Statespacerepresentationofthelinearisedmodel
 A=[0 1 0 0 0 0;
 0 0 -(m1*g)/M 0 -(m2*g)/M 0;
 0 0 0 1 0 0;
 0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
 0 0 0 0 0 1;
 0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
 disp('A matrix')
 disp(A)

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

 disp('B matrix')
 disp(B)

 Controlab_m=[B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B] ;

 disp('controlability matrix')
 disp(Controlab_m)

 disp('Determinant of controlability matrix')
 disp(simplify(det(Controlab_m)));
 disp('In the above matrix if l1 = l2 then the determinant is zero means the condition here is that l1 not equal to l2')


 disp('Rank of controlability matrix')
 disp(rank(Controlab_m))

 disp('Setting l1 = l2 in the controllability matrix')
 Uncontrollable_matrix=subs(Controlab_m,l1,l2);
 disp('Displaying the rank of uncontrollable matrix')
 disp(rank(Uncontrollable_matrix))
```

C. LQR Controller Code



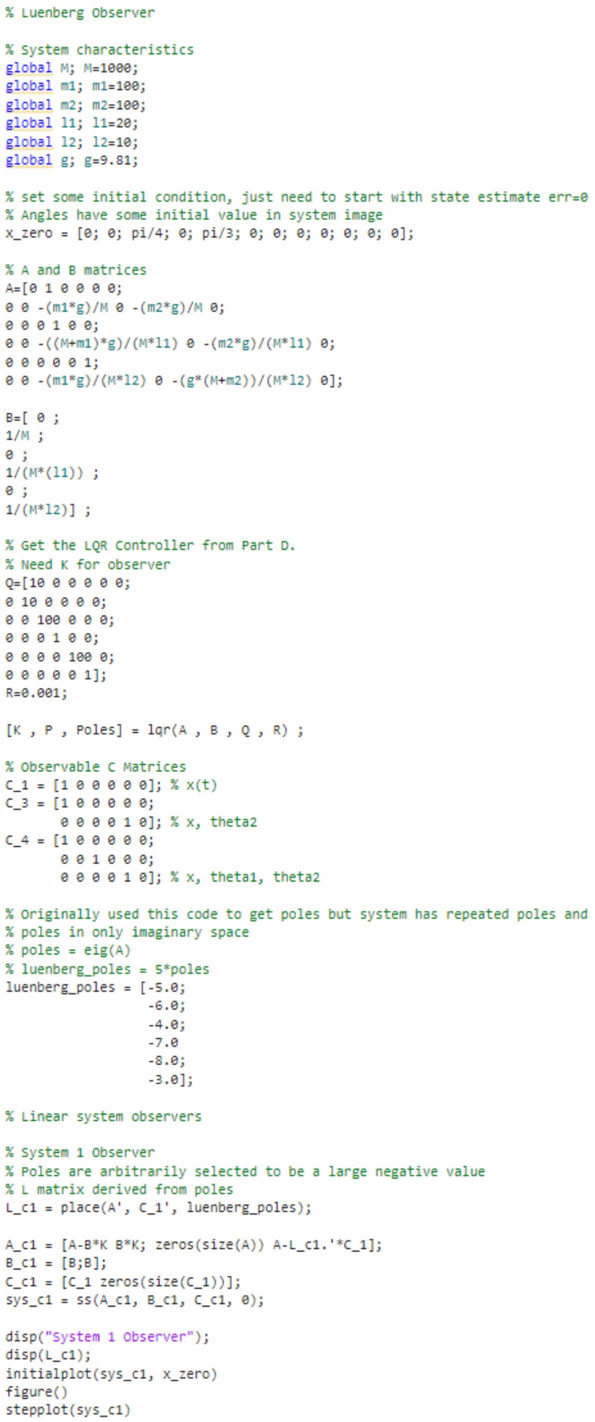
```
global M; M=1000;
global m1; m1=100;
global m2; m2=100;
global l1; l1=20;
global l2; l2=10;
global g; g=9.81;

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

 B=[ 0 ;
     1/M ;
     0 ;
     1/(M*(l1)) ;
     0 ;
     1/(M*l2)] ;
 %Initial state
 X_init=[
         0 ;
         0 ;
         0.5 ;
         0 ;
         0.6 ;
         0
     ];
 %Q and R values
 Q=[10 0 0 0 0 0;
 0 10 0 0 0 0;
 0 0 100 0 0 0;
 0 0 0 1 0 0;
 0 0 0 0 100 0;
 0 0 0 0 0 1];
 R=0.001;
 %C is the identity matrix which directly gives the output and D is 0
 C=eye(6);D=0;
 init_sys = ss(A , B , C , D) ;
 figure
 initial(init_sys , X_init)
 % After here, we make the LQR controller
 [K , P , Poles] = lqr(A , B , Q , R) ;

 lqr_controller = ss((A - (B*K)) , B , C , D) ;
 figure
 initial(lqr_controller , X_init)

 % Non linear controls
range = 0:0.1:500;
[t_nl, dx_nl] = ode45(@(t,x)orig_sys(t,x,-K*x), range, X_init);
figure
plot(t_nl, dx_nl(:,1))
xlabel('time(s)')
ylabel('x(m)')
title('Nonlinear control')
```

D. Observability Analysis Code

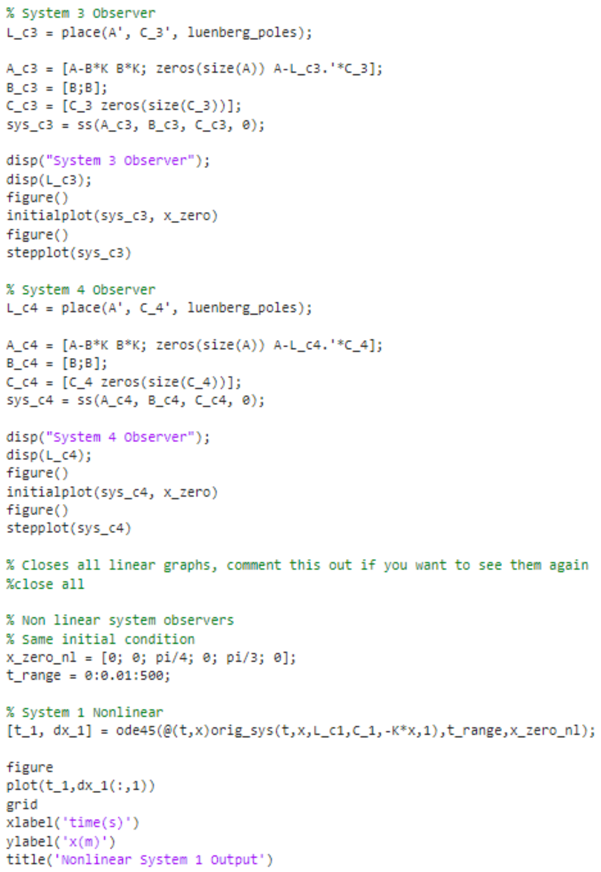


```
% Analysis of Observability of various outputs

% System characteristics
M=1000; m1=100; m2=100; l1=20; l2=10; g=9.81;

% State Space Representation
% A and B 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)];

% If the pair A^T C^T is controllable then A, C is observable
At = A.';
% C Matrix for each output type
C_1 = [1 0 0 0 0 0]; % x(t)
C_1t = C_1.';
C1Control = [C_1t At*C_1t At*At*C_1t At*At*At*C_1t At*At*At*At*C_1t At*At*At*At*At*C_1t];
disp("Controllability matrix rank for output set 1")
disp(rank(C1Control))

C_2 = [0 0 1 0 0 0;
       0 0 0 0 1 0]; % theta1, theta2
C_2t = C_2.';
C2Control = [C_2t At*C_2t At*At*C_2t At*At*At*C_2t At*At*At*At*C_2t At*At*At*At*C_2t];
disp("Controllability matrix rank for output set 2")
disp(rank(C2Control))

C_3 = [1 0 0 0 0 0;
       0 0 0 0 1 0]; % x, theta2
C_3t = C_3.';
C3Control = [C_3t At*C_3t At*At*C_3t At*At*At*C_3t At*At*At*At*C_3t At*At*At*At*C_3t];
disp("Controllability matrix rank for output set 3")
disp(rank(C3Control))

C_4 = [1 0 0 0 0 0;
       0 0 1 0 0 0;
       0 0 0 0 1 0]; % x, theta1, theta2
C_4t = C_4.';
C4Control = [C_4t At*C_4t At*At*C_4t At*At*At*C_4t At*At*At*At*C_4t At*At*At*At*C_4t];
disp("Controllability matrix rank for output set 4")
disp(rank(C4Control))
```

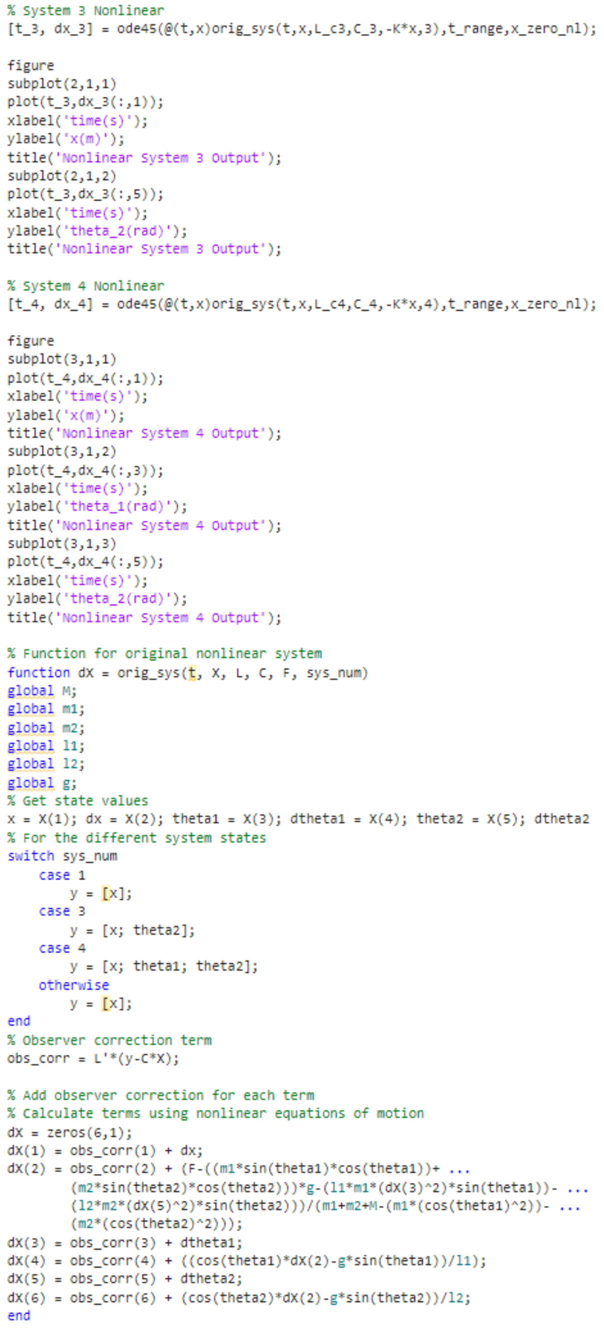
E. Luenberger Observer Code



```
% Luenberg Observer

% System characteristics
global M; M=1000;
global m1; m1=100;
global m2; m2=100;
global l1; l1=20;
global l2; l2=10;
global g; g=9.81;

% set some initial condition, just need to start with state estimate err=0
% Angles have some initial value in system image
x_zero = [0; 0; pi/4; 0; pi/3; 0; 0; 0; 0; 0; 0; 0];

% A and B 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)] ;

% Get the LQR Controller from Part D.
% Need K for observer
Q=[10 0 0 0 0 0;
0 10 0 0 0 0;
0 0 100 0 0 0;
0 0 0 1 0 0;
0 0 0 0 100 0;
0 0 0 0 0 1];
R=0.001;

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

% Observable C Matrices
C_1 = [1 0 0 0 0 0]; % x(t)
C_3 = [1 0 0 0 0 0;
       0 0 0 0 1 0]; % x, theta2
C_4 = [1 0 0 0 0 0;
       0 0 1 0 0 0;
       0 0 0 0 1 0]; % x, theta1, theta2

% Originally used this code to get poles but system has repeated poles and
% poles in only imaginary space
% poles = eig(A)
% luenberg_poles = 5*poles
luenberg_poles = [-5.0;
                  -6.0;
                  -4.0;
                  -7.0
                  -8.0;
                  -3.0];

% Linear system observers

% System 1 Observer
% Poles are arbitrarily selected to be a large negative value
% L matrix derived from poles
L_c1 = place(A', C_1', luenberg_poles);

A_c1 = [A-B*K B*K; zeros(size(A)) A-L_c1.'*C_1];
B_c1 = [B;B];
C_c1 = [C_1 zeros(size(C_1))];
sys_c1 = ss(A_c1, B_c1, C_c1, 0);

disp("System 1 Observer");
disp(L_c1);
initialplot(sys_c1, x_zero)
figure()
stepplot(sys_c1)
```

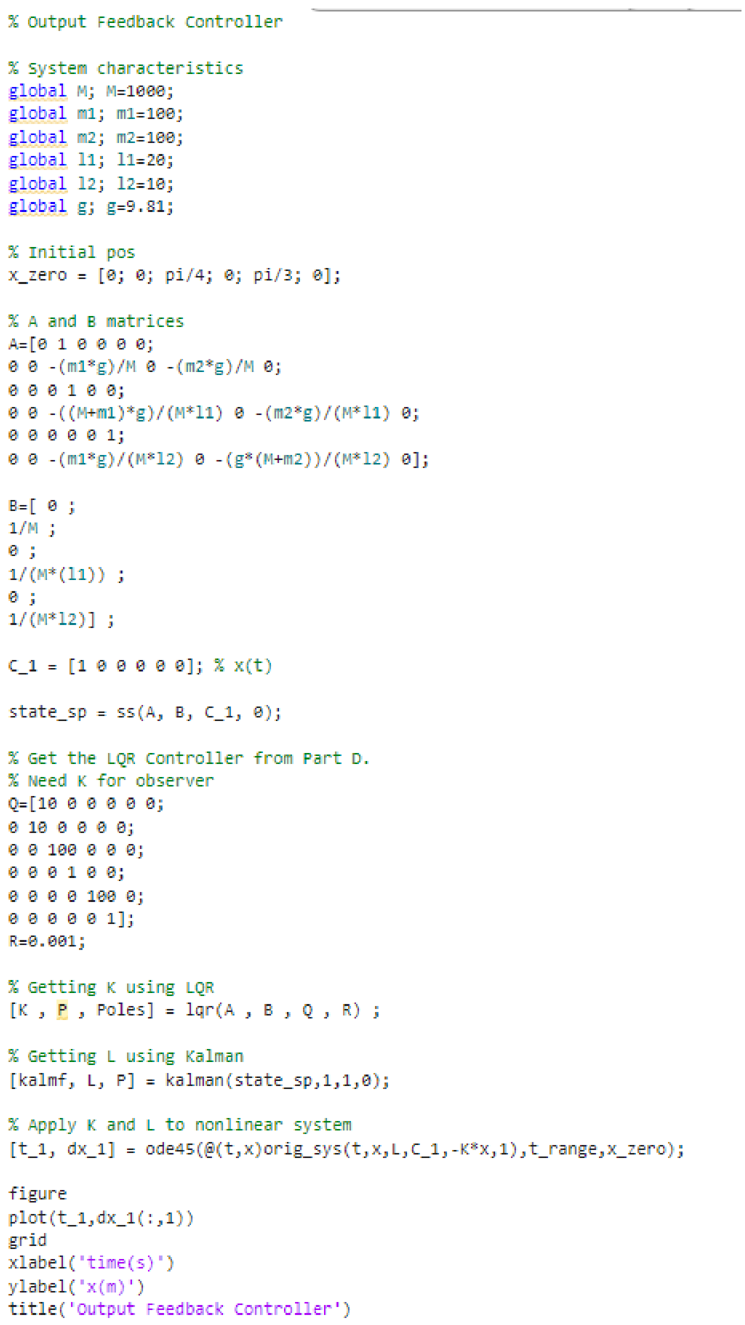


```
% System 3 Observer
L_c3 = place(A', C_3', luenberg_poles);

A_c3 = [A-B*K B*K; zeros(size(A)) A-L_c3.'*C_3];
B_c3 = [B;B];
C_c3 = [C_3 zeros(size(C_3))];
sys_c3 = ss(A_c3, B_c3, C_c3, 0);

disp("System 3 Observer");
disp(L_c3);
figure()
initialplot(sys_c3, x_zero)
figure()
stepplot(sys_c3)

% System 4 Observer
L_c4 = place(A', C_4', luenberg_poles);

A_c4 = [A-B*K B*K; zeros(size(A)) A-L_c4.'*C_4];
B_c4 = [B;B];
C_c4 = [C_4 zeros(size(C_4))];
sys_c4 = ss(A_c4, B_c4, C_c4, 0);

disp("System 4 Observer");
disp(L_c4);
figure()
initialplot(sys_c4, x_zero)
figure()
stepplot(sys_c4)

% Closes all linear graphs, comment this out if you want to see them again
%close all

% Non linear system observers
% Same initial condition
x_zero_nl = [0; 0; pi/4; 0; pi/3; 0];
t_range = 0:0.01:500;

% System 1 Nonlinear
[t_1, dx_1] = ode45(@(t,x)orig_sys(t,x,L_c1,C_1,-K*x,1),t_range,x_zero_nl);

figure
plot(t_1,dx_1(:,1))
grid
xlabel('time(s)')
ylabel('x(m)')
title('Nonlinear System 1 Output')
```



```
% System 3 Nonlinear
[t_3, dx_3] = ode45(@(t,x)orig_sys(t,x,L_c3,C_3,-K*x,3),t_range,x_zero_nl);

figure
subplot(2,1,1)
plot(t_3,dx_3(:,1));
xlabel('time(s)');
ylabel('x(m)');
title('Nonlinear System 3 Output');
subplot(2,1,2)
plot(t_3,dx_3(:,5));
xlabel('time(s)');
ylabel('theta_2(rad)');
title('Nonlinear System 3 Output');

% System 4 Nonlinear
[t_4, dx_4] = ode45(@(t,x)orig_sys(t,x,L_c4,C_4,-K*x,4),t_range,x_zero_nl);

figure
subplot(3,1,1)
plot(t_4,dx_4(:,1));
xlabel('time(s)');
ylabel('x(m)');
title('Nonlinear System 4 Output');
subplot(3,1,2)
plot(t_4,dx_4(:,3));
xlabel('time(s)');
ylabel('theta_1(rad)');
title('Nonlinear System 4 Output');
subplot(3,1,3)
plot(t_4,dx_4(:,5));
xlabel('time(s)');
ylabel('theta_2(rad)');
title('Nonlinear System 4 Output');

% Function for original nonlinear system
function dX = orig_sys(t, X, L, C, F, sys_num)
global M;
global m1;
global m2;
global l1;
global l2;
global g;
% Get state values
x = X(1); dx = X(2); theta1 = X(3); dtheta1 = X(4); theta2 = X(5); dtheta2
% For the different system states
switch sys_num
    case 1
        y = [x];
    case 3
        y = [x; theta2];
    case 4
        y = [x; theta1; theta2];
    otherwise
        y = [x];
end
% Observer correction term
obs_corr = L'*(y-C*X);

% Add observer correction for each term
% Calculate terms using nonlinear equations of motion
dX = zeros(6,1);
dX(1) = obs_corr(1) + dx;
dX(2) = obs_corr(2) + (F-((m1*sin(theta1)*cos(theta1))+ ...
        (m2*sin(theta2)*cos(theta2)))*g-(l1*m1*(dX(3)^2)*sin(theta1))- ...
        (l2*m2*(dX(5)^2)*sin(theta2)))/(m1+m2+M-(m1*(cos(theta1)^2))- ...
        (m2*(cos(theta2)^2)));
dX(3) = obs_corr(3) + dtheta1;
dX(4) = obs_corr(4) + ((cos(theta1)*dX(2)-g*sin(theta1))/l1);
dX(5) = obs_corr(5) + dtheta2;
dX(6) = obs_corr(6) + (cos(theta2)*dX(2)-g*sin(theta2))/l2;
end
```

F. LQG Controller Code



```
% Output Feedback Controller

% System characteristics
global M; M=1000;
global m1; m1=100;
global m2; m2=100;
global l1; l1=20;
global l2; l2=10;
global g; g=9.81;

% Initial pos
x_zero = [0; 0; pi/4; 0; pi/3; 0];

% A and B 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)] ;

C_1 = [1 0 0 0 0 0]; % x(t)

state_sp = ss(A, B, C_1, 0);

% Get the LQR Controller from Part D.
% Need K for observer
Q=[10 0 0 0 0 0;
0 10 0 0 0 0;
0 0 100 0 0 0;
0 0 0 1 0 0;
0 0 0 0 100 0;
0 0 0 0 0 1];
R=0.001;

% Getting K using LQR
[K , P , Poles] = lqr(A , B , Q , R) ;

% Getting L using Kalman
[kalmf, L, P] = kalman(state_sp,1,1,0);

% Apply K and L to nonlinear system
[t_1, dx_1] = ode45(@(t,x)orig_sys(t,x,L,C_1,-K*x,1),t_range,x_zero);

figure
plot(t_1,dx_1(:,1))
grid
xlabel('time(s)')
ylabel('x(m)')
title('Output Feedback Controller')
```