University Of Maryland

College Park

Project-2

ENPM - 667 Control of Robotic Systems

Controller Design for a cart with Two pendulums

Srikumar Muralidharan
116950572

Govind Ajith Kumar
116699488

Controller Design for a cart with Two Pendulums

Srikumar. M
Govind A.K

December 16, 2019

1 Introduction

The project consisted of designing a controller for a cart moving along an axis with two pendulums hanging from it. The project was an exhaustive study on the concepts of Jacobian, Euler Lagrange Equations, Controllability, Observability, LQR, LQG, Lyapunov Stability and all other auxiliary topics taught throughout the ENPM667 Course during the Fall of 2019, by Dr. Wasim Malik at the University of Maryland, College Park. The situation is represented in the image below.

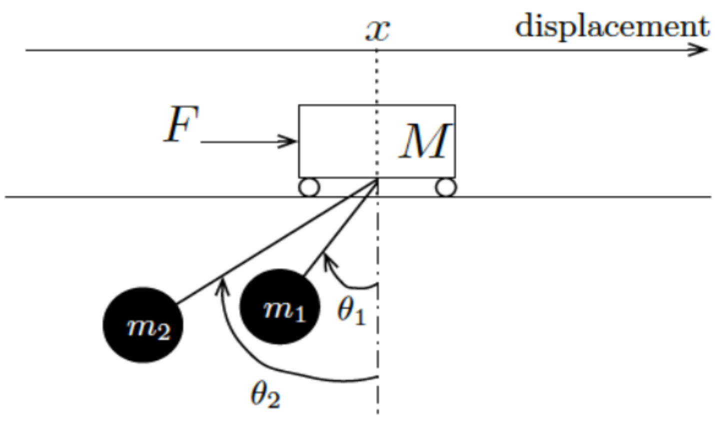


Figure: Problem Statement

2 Answers and Simulations

The answers to the questions are neatly represented in order. Since, the calculations are computationally intensive and very time-consuming to be done manually, the authors of this report have used the help of MATLAB to solve and visualize the solutions, so as to develop a satisfactory, yet exhaustive analysis of the problem.

2.1 Part A - Equation of motion for the system

Let us define the position of mass m1 as a function of θ_1 :

$$x_{m1} = (x - l_1 sin(\theta_1))\hat{i} + (-l_1 cos(\theta_1))\hat{j} \tag{1}$$

Differentiating this equation with respect to time gives us the velocity equation:

$$v_{m1} = (\dot{x} - l_1 cos(\theta_1)\dot{\theta_1})\hat{i} + l_1 sin(\theta_1)\dot{\theta_1}\hat{j} \tag{2}$$

Now, Let us define the position of mass m2 as a function of θ_2 :

$$x_{m2} = (x - l_2 sin(\theta_2))\hat{i} + (-l_2 cos(\theta_2))\hat{j} \tag{3}$$

Differentiating this equation with respect to time gives us the velocity equation:

$$v_{m2} = (\dot{x} - l_2 cos(\theta_2)\dot{\theta_2})\hat{i} + l_2 sin(\theta_2)\dot{\theta_2}\hat{j} \tag{4}$$

Given the 2 velocity equations, we can formulate the Kinetic energy of the system as follows:

$$\text{K.E} = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1(\dot{x} - l_1\dot{\theta}_1 cos(\theta_1))^2 + \frac{1}{2}m_1(l_1\dot{\theta}_1(sin(\theta_1)))^2 + \frac{1}{2}m_2(\dot{x} - \dot{\theta}_2 l_2 cos(\theta_2))^2 + \frac{1}{2}m_2(l_2\dot{\theta}_2(sin(\theta_2)))^2 \tag{5}$$

Also, Potential Energy of the system is:

$$P.E = -m_1 g l_1 cos(\theta_1) - m_2 g l_2 cos(\theta_2) = -g[m_1 l_1 cos(\theta_1) + m_2 g l_2 cos(\theta_2) \tag{6}$$

Creating the Lagrange equation as the sum of Kinetic and Potential energies:

$$L = K.E - P.E \tag{7}$$

$$\text{L}= \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m_1\dot{x}^2 + \frac{1}{2}m_1 l_1^2\dot{\theta_1}^2 cos^2(\theta_1) - m_1 l_1\dot{\theta_1}\dot{x}cos(\theta_1) + \frac{1}{2}m_1 l_1^2\dot{\theta_1}^2 sin^2(\theta_1) + \frac{1}{2}m_2\dot{x}^2 + \frac{1}{2}m_2 l_2^2\dot{\theta_2}^2 cos^2(\theta_2) - m_2 l_2\dot{\theta_2}\dot{x}cos(\theta_2) + \frac{1}{2}m_2 l_2^2\dot{\theta_2}^2 sin^2(\theta_1) + g[m_1 l_1 cos(\theta_1) + m_2 l_2 cos(\theta_2)] \tag{8}$$

Simplifying :

$$L= \frac{1}{2}M\dot{x}^2 + \frac{1}{2}(m_1+m_2)\dot{x}^2 + \frac{1}{2}m_1 l_1^2 \dot{\theta_1}^2 + \frac{1}{2}m_2 l_2^2 \dot{\theta_2}^2 -$$

$$\dot{x}(m_1 l_1 \dot{\theta}_1 cos(\theta_1) + m_2 l_2 \dot{\theta}_2 cos(\theta_2)) + g[m_1 l_1 cos(\theta_1) + m_2 l_2 cos(\theta_2)] \quad (9)$$

The Lyapunov Equations pertaining to the state variables considered for our system are defined as follows:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}}) - (\frac{\partial L}{\partial x}) = F \quad (10)$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_1}}) - (\frac{\partial L}{\partial \theta_1}) = 0 \quad (11)$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_2}}) - (\frac{\partial L}{\partial \theta_2}) = 0 \quad (12)$$

Now, let us compute these relations:
Relation 1:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}}) - (\frac{\partial L}{\partial x}) = F \quad (13)$$

$$\frac{\partial L}{\partial \dot{x}} = M\dot{x} + (m_1+m_2)\dot{x} - m_1 l_1 \dot{\theta}_1 cos(\theta_1) - m_2 l_2 \dot{\theta}_2 cos(\theta_2) \quad (14)$$

Differentiating this with respect to time, we get:

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

Also here:

$$\frac{\partial L}{\partial x} = 0 \quad (16)$$

Hence we can write the first equation as:

$$[M+m_1+m_2]\ddot{x} - m_1 l_1 \ddot{\theta}_1 cos(\theta_1) + m_1 l_1 \dot{\theta_1}^2 sin(\theta 1) - m_2 l_2 \ddot{\theta}_2 cos(\theta_2) + m_2 l_2 \dot{\theta_2}^2 sin(\theta_2) = F (17)$$

Now, we know that:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_1}}) - (\frac{\partial L}{\partial \theta_1}) = 0 \quad (18)$$

$$\frac{\partial L}{\partial \dot{\theta_1}} = m_1 l_1^2 \dot{\theta_1} - m_1 \dot{x} l_1 cos(\theta_1) \tag{19}$$

Differentiating this with respect to time, we get:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_1}}) = m_1 l_1^2 \ddot{\theta_1} - [m_1 l_1 \ddot{x} cos(\theta_1) - m_1 \dot{x} l_1 \dot{\theta_1} sin(\theta_1)] \tag{20}$$

Also here:

$$\frac{\partial L}{\partial \theta_1} = m_1 l_1 \dot{\theta_1} \dot{x} sin(\theta_1) - m_1 l_1 g sin(\theta_1) \tag{21}$$

Now, combining these two equations:

$$m_1 l_1^2 \ddot{\theta_1} - m_1 \ddot{x} l_1 cos(\theta_1) + m_1 \dot{\theta_1} \dot{x} l_1 sin(\theta_1) - m_1 \dot{\theta_1} \dot{x} l_1 sin(\theta_1) + m_1 l_1 g sin(\theta_1) = 0 \tag{22}$$

Cancelling out the equivalent terms, we get the following equation:

$$m_1 l_1^2 \ddot{\theta_1} - m_1 \ddot{x} l_1 cos(\theta_1) + m_1 l_1 g sin(\theta_1) = 0 \tag{23}$$

This is the equation we get from the second Lagrange Equation.

Now, to find the third equation from the Lagrange equation, we perform the following calculations:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_2}}) - (\frac{\partial L}{\partial \theta_2}) = 0 \tag{24}$$

$$\frac{\partial L}{\partial \dot{\theta_2}} = m_2 l_2^2 \dot{\theta_2} - m_2 \dot{x} l_2 cos(\theta_2) \tag{25}$$

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta_2}}) = m_2 l_2^2 \ddot{\theta_2} - [m_2 \ddot{x} l_2 cos(\theta_2) - m_2 \dot{\theta_2} \dot{x} l_2 sin(\theta_2)] \tag{26}$$

$$(\frac{\partial L}{\partial \theta_2}) = m_2 \dot{x} l_2 \dot{\theta_2} sin(\theta_2) - m_2 l_2 g sin(\theta_2) \tag{27}$$

Hence we write :

$$m_2 l_2^2 \ddot{\theta_2} - m_2 l_2 \ddot{x} cos(\theta_2) + m_2 \dot{\theta_2} \dot{x} l_2 sin(\theta_2) - m_2 \dot{\theta_2} \dot{x} l_2 sin(\theta_2) + m_2 g l_2 sin(\theta_2) \tag{28}$$

This implies that after cancellation of terms, we get:

$$m_2 l_2^2 \ddot{\theta_2} - m_2 l_2 \ddot{x} cos(\theta_2) + m_2 g l_2 sin(\theta_2) = 0 \tag{29}$$

2.2 Part B : Linearizing about equilibrium point

Before we proceed to linearize about the given equilibrium points, we write the equations for the double differentiation components of some of our state variables, as deduced from the equations above:

$$\ddot{x} = \frac{F - ((\frac{g}{2})(m_1 sin(2\theta_1) + m_2 sin(2\theta_2))) - m_1 l_1 \dot{\theta_1}^2 sin(\theta_1) - m_2 l_2 \dot{\theta_2}^2 sin(\theta_2)}{M + m_1 sin(\theta_1)^2 + m_1 sin(\theta_1)^2} \tag{30}$$

$$\ddot{\theta_1} = \frac{1}{l_1}[\ddot{x} cos(\theta_1) - g sin(\theta_1)] \tag{31}$$

$$\ddot{\theta_2} = \frac{1}{l_2}[\ddot{x} cos(\theta_2) - g sin(\theta_2)] \tag{32}$$

For the cart and double pendulum system represented above, let us define the state variables in this order :
State variables (defined in order) :

$$[x; \dot{x}; \theta_1; \dot{\theta_1}; \theta_2; \dot{\theta_2}] \tag{33}$$

From these equations derived above, we create the functions required for creating the Jacobian matrix as follows:
Now, The general state space equations are written as:

$$\mathbf{X} = A\mathbf{X} + B\mathbf{U} \tag{34}$$

Writing the Jacobian Matrix linearized around the equilibrium points given x $= \theta_1 = \theta_2 = 0$:

$$\begin{matrix}
\frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial \dot{x}} & \frac{\partial F_1}{\partial \theta_1} & \frac{\partial F_1}{\partial \dot{\theta_1}} & \frac{\partial F_1}{\partial \theta_2} & \frac{\partial F_1}{\partial \dot{\theta_2}} \\
\frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial \dot{x}} & \frac{\partial F_2}{\partial \theta_1} & \frac{\partial F_2}{\partial \dot{\theta_1}} & \frac{\partial F_2}{\partial \theta_2} & \frac{\partial F_2}{\partial \dot{\theta_2}} \\
\frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial \dot{x}} & \frac{\partial F_3}{\partial \theta_1} & \frac{\partial F_3}{\partial \dot{\theta_1}} & \frac{\partial F_3}{\partial \theta_2} & \frac{\partial F_3}{\partial \dot{\theta_2}} \\
\frac{\partial F_4}{\partial x} & \frac{\partial F_4}{\partial \dot{x}} & \frac{\partial F_4}{\partial \theta_1} & \frac{\partial F_4}{\partial \dot{\theta_1}} & \frac{\partial F_4}{\partial \theta_2} & \frac{\partial F_4}{\partial \dot{\theta_2}} \\
\frac{\partial F_5}{\partial x} & \frac{\partial F_5}{\partial \dot{x}} & \frac{\partial F_5}{\partial \theta_1} & \frac{\partial F_5}{\partial \dot{\theta_1}} & \frac{\partial F_5}{\partial \theta_2} & \frac{\partial F_5}{\partial \dot{\theta_2}} \\
\frac{\partial F_6}{\partial x} & \frac{\partial F_6}{\partial \dot{x}} & \frac{\partial F_6}{\partial \theta_1} & \frac{\partial F_6}{\partial \dot{\theta_1}} & \frac{\partial F_6}{\partial \theta_2} & \frac{\partial F_6}{\partial \dot{\theta_2}}
\end{matrix}$$

The A matrix is shown below:

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m_1 g}{M} & 0 & \frac{-m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{(M+m_1)g}{Ml_1} & 0 & \frac{-m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{Ml_2} & 0 & \frac{(M+m_2)g}{Ml_2} & 0 \end{matrix}$$

The B matrix is shown below:

$$\begin{matrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{matrix}$$

Substitute the values of **A and B** in the state space equation to get the state equation as:

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m_1 g}{M} & 0 & \frac{-m_2 g}{M} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{(M+m_1)g}{Ml_1} & 0 & \frac{-m_2 g}{Ml_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-m_1 g}{Ml_2} & 0 & \frac{(M+m_2)g}{Ml_2} & 0 \end{matrix} \mathbf{X} + \begin{matrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml_1} \\ 0 \\ \frac{1}{Ml_2} \end{matrix} \mathbf{U} \tag{35}$$

calculations have been represented in MATLAB, and have been attached here. Also, please refer the comments in MATLAB code for other details.

Part C - Controllability check

```
%clearing all the previous outputs
clc
clear
%declaring the symbolic variables
syms M m1 m2 l1 l2 g;
% Creating my linearised state space equation using 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];
%declaring the B matrix
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)]
% Writing the controllability matrix as follows
disp("Controllability matrix =");
Ct= [B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B]
disp("Finding the determinant of the controllability matrix=");
%simplifying to check for the condition in which the matrix isn't
%controllable
disp(simplify(det(Ct)));
disp("Displaying the rank of the controllability matrix =");
%calculating the rank of the matrix. The system is controllable only
 if the
%controllability matrix is full rank, that is RANK:6 in this case
rank(Ct)
% From this, we can see that the matrix is invertible for all cases,
% other than the case l1=l2, or for extremely high values of M, m1
% m2, which is not a realistic case. Therefore, we can conclude that
% for l1=l2, the matrix is not controllable. This can be shown as
 follows:
disp("for l1 = l2, Controllability matrix is")
Ct1 = subs(Ct,l1,l2) %using the subs function to make l1 = l2
disp("Displaying rank of the new matrix =")
%Calculating the rank in this case.
rank(Ct1)
%if loop to display the condition of the system in either case
if (rank(Ct1)==rank(Ct))
    disp("System is controllable as new rank and old rank are same")
else
    disp("system is not controllable as new rank and old rank are
 different")
end
```

```
B =

      0
    1/M
      0
```

```
 1/(M*l1)
        0
 1/(M*l2)

Controllability matrix =

Ct =

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

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

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

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

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

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

                                                    0]

Finding the determinant of the controllability matrix=
-(g^6*(l1 - l2)^2)/(M^6*l1^6*l2^6)

Displaying the rank of the controllability matrix =

ans =

     6
```

```
for l1 = l2, Controllability matrix is

Ct1 =

[         0,      1/M,                                                       0,
          - (g*m1)/(M^2*l2) - (g*m2)/(M^2*l2),

                                              0,              ((g^2*m1*(M
 + m1))/(M^2*l2) + (g^2*m1*m2)/(M^2*l2))/(M*l2) + ((g^2*m2*(M + m2))/
(M^2*l2) + (g^2*m1*m2)/(M^2*l2))/(M*l2)]
[      1/M,        0,          - (g*m1)/(M^2*l2) - (g*m2)/(M^2*l2),
                                               0,              ((g^2*m1*(M
 + m1))/(M^2*l2) + (g^2*m1*m2)/(M^2*l2))/(M*l2) + ((g^2*m2*(M + m2))/
(M^2*l2) + (g^2*m1*m2)/(M^2*l2))/(M*l2),

                                0]
[         0, 1/(M*l2),                                                       0, -
 (g*(M + m1))/(M^2*l2^2) - (g*m2)/(M^2*l2^2),

                                      0, ((g^2*m2*(M + m1))/(M^2*l2^2)
 + (g^2*m2*(M + m2))/(M^2*l2^2))/(M*l2) + ((g^2*(M + m1)^2)/(M^2*l2^2)
 + (g^2*m1*m2)/(M^2*l2^2))/(M*l2)]
[ 1/(M*l2),        0, - (g*(M + m1))/(M^2*l2^2) - (g*m2)/(M^2*l2^2),
                                             0, ((g^2*m2*(M + m1))/
(M^2*l2^2) + (g^2*m2*(M + m2))/(M^2*l2^2))/(M*l2) + ((g^2*(M + m1)^2)/
(M^2*l2^2) + (g^2*m1*m2)/(M^2*l2^2))/(M*l2),

                                    0]
[         0, 1/(M*l2),                                                       0, -
 (g*(M + m2))/(M^2*l2^2) - (g*m1)/(M^2*l2^2),

                                      0, ((g^2*m1*(M + m1))/(M^2*l2^2)
 + (g^2*m1*(M + m2))/(M^2*l2^2))/(M*l2) + ((g^2*(M + m2)^2)/(M^2*l2^2)
 + (g^2*m1*m2)/(M^2*l2^2))/(M*l2)]
[ 1/(M*l2),        0, - (g*(M + m2))/(M^2*l2^2) - (g*m1)/(M^2*l2^2),
                                             0, ((g^2*m1*(M + m1))/
(M^2*l2^2) + (g^2*m1*(M + m2))/(M^2*l2^2))/(M*l2) + ((g^2*(M + m2)^2)/
(M^2*l2^2) + (g^2*m1*m2)/(M^2*l2^2))/(M*l2),

                                    0]

Displaying rank of the new matrix =

ans =

     4

system is not controllable as new rank and old rank are different
```

Published with MATLAB® R2019b

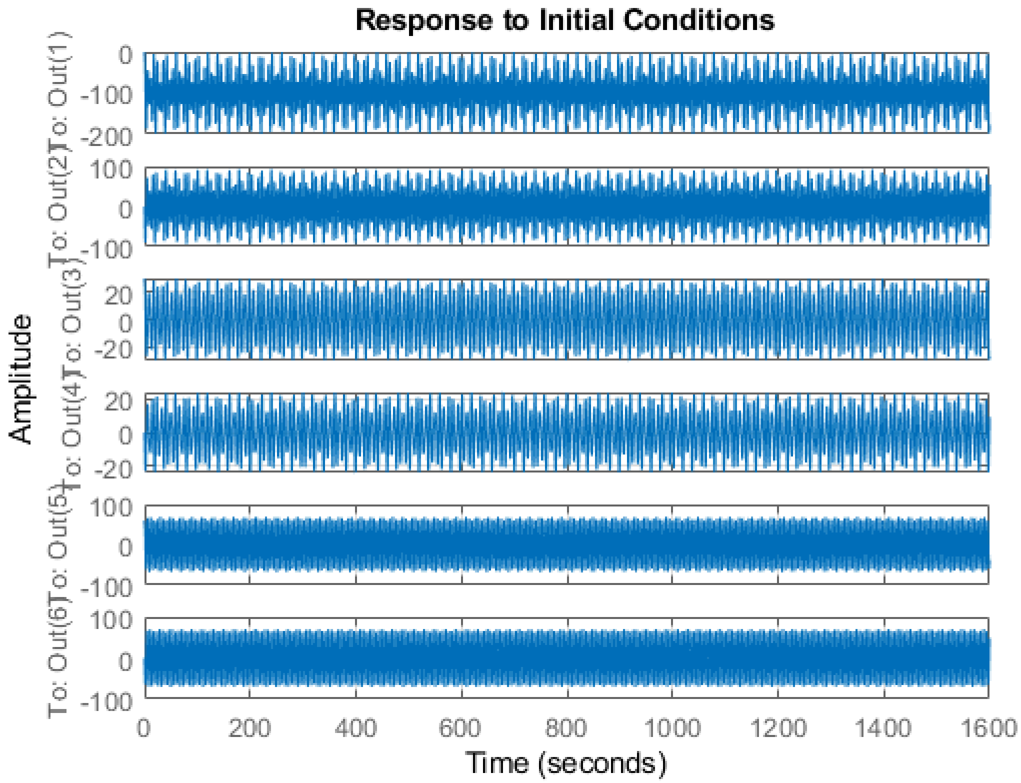
Part D(a) - LQR design for linear system

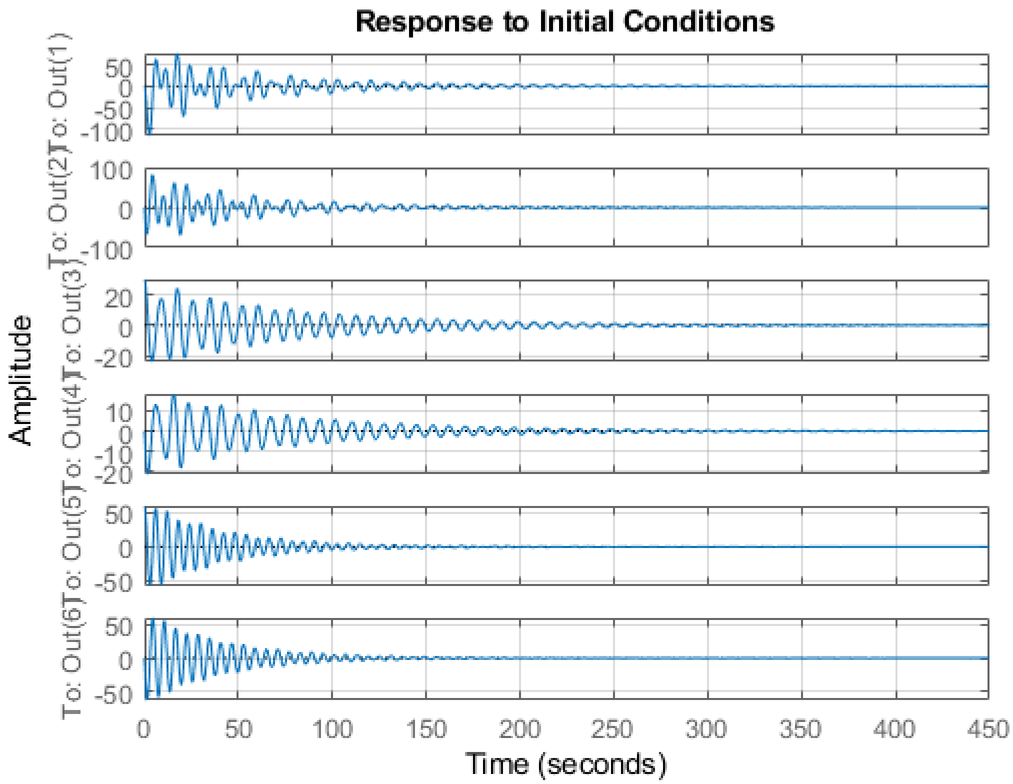
```
%clearing all the previous outputs
clc
clear all

% Given the constraints for the masses and lengths of the strings
M=1000;%Mass of the cart
m1=100;%mass of Pendulum 1
m2=100;%mass of Pendulum 2
l1=20;%length of the string of Pendulum 1
l2=10;%length of the string of Pendulum 2
g=9.81; %declaring the value of the accelertaion due to gravity in m/
s^2
% Porting the A and B matrices here
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)];
% Checking for the controllability of the given system
if (rank(ctrb(A,B))==6)
    disp("Rank of ctrb matches order of A, system is controllable")
else
    disp("Rank of ctrb doesnt matche order of A, system is
 uncontrollable")
end
% Now, we give the values of the initial condition for our state
 variables.
% We are using the state variables in the following format:
% x(t) = [x x_dot theta_1 theta_1_dot theta_2 theta_2_dot]'
% therefore, the initial conditions are as follows.
x_initial = [0;0;30;0;60;0];
%initial disp=0, angles for the two pendulums are as described above.
% We assume the values of Q and R. For Q, we penalize theta's more
 than x.
Q=[100 0 0 0 0 0;
   0 100 0 0 0 0;
   0 0 1000 0 0 0;
   0 0 0 1000 0 0;
   0 0 0 0 1000 0;
   0 0 0 0 0 1000];
R=0.01;
% The above values of Q and R are a trade-off and we use them both to
% develop a system as per our priorities
% Let us assume that C matrix is a direct representation of the output
% matrix, which makes D=0
C = eye(6);% To form a 6 X 6 identity matrix
D = 0; % Initialising the D matrix to be Zero
```

```
% Lets observe the response of the system for the given initial
 conditions
% for the given initial values
disp("For the given initial values, the state response is as
 follows:")
sys1 = ss(A,B,C,D);%MATLAB code for calculating the state space
 representation of the system
figure
initial(sys1,x_initial)%MATLAB inbuilt function to check the initial
 response of the system
grid on   %grid lines visible

disp("Now, seeing the results using an LQR controller")
[K_val, P_mat, Poles] = lqr(A,B,Q,R);%In-built MATLAB code for LQR
 Controllers
K_val %computes the K matrix and displays
P_mat %positive definite matrix calculated for the same
Poles %To see the poles of the given equation
sys2 = ss(A-(B*K_val),B,C,D); %Using the K matrix to define ss
figure
initial(sys2,x_initial)
grid on
% We observe from this, for higher values of Q components, the time it
% takes for the system is die out is much lesser than in other cases.
 Also,
% lower the value of R, the faster the system stabilizes.
```

```
Rank of ctrb matches order of A, system is controllable
For the given initial values, the state response is as follows:
Now, seeing the results using an LQR controller

K_val =

  100.0000  503.1594   60.5544 -232.7664   94.9045  -54.2816


P_mat =

   1.0e+05 *

    0.0050    0.0122   -0.0023   -0.0212   -0.0005   -0.0110
    0.0122    0.0615    0.0095   -0.1116    0.0082   -0.0557
   -0.0023    0.0095    0.8033   -0.0023    0.0137   -0.0335
   -0.0212   -0.1116   -0.0023    1.6872    0.0348    0.0395
   -0.0005    0.0082    0.0137    0.0348    0.4683   -0.0049
   -0.0110   -0.0557   -0.0335    0.0395   -0.0049    0.4833


Poles =

  -0.0120 + 0.7276i
  -0.0120 - 0.7276i
  -0.0245 + 1.0421i
  -0.0245 - 1.0421i
```

```
-0.2066 + 0.2023i
-0.2066 - 0.2023i
```





Published with MATLAB® R2019b

Part D(b) - Non-Linear Calculations

```
%clearing all previous outputs
clear all
clc
%Declaring the new output variables
y0 = [5; 0; 30; 0; 60; 0]
tspan = 0:0.01:5000;%defining the timespan
[t1,y1] = ode45(@doublepend,tspan,y0); %using ode45 function,
 specifically built
%for non-linear systems
plot(t1,y1)%plotting the function output on a 2D graph
grid on% making the grid lines visible
```

```
y0 =

     5
     0
    30
     0
    60
     0
```

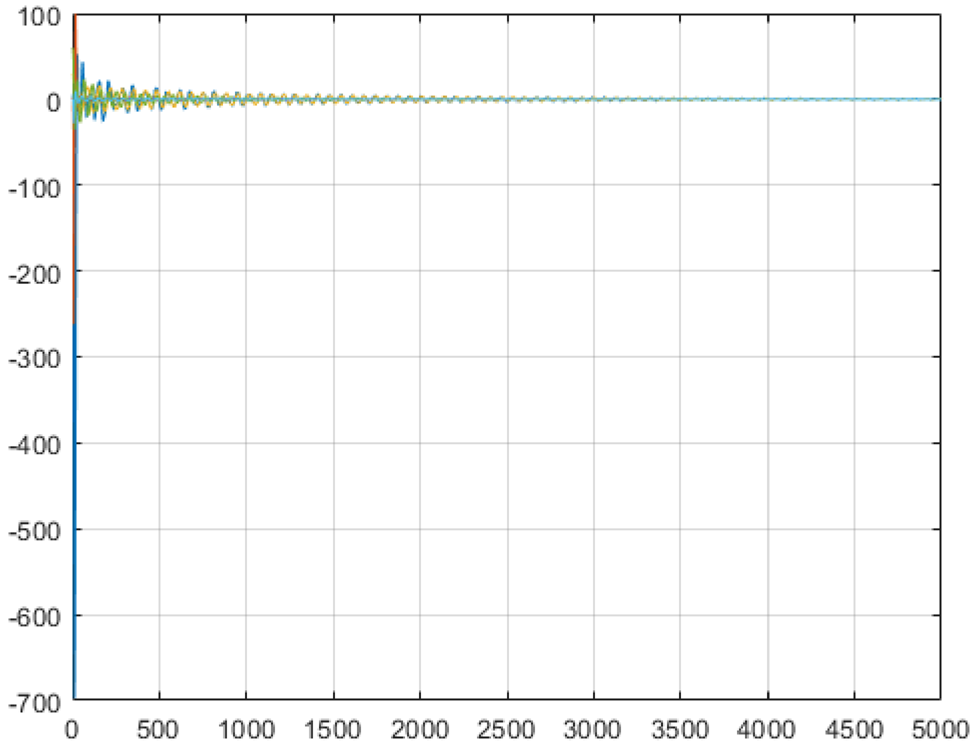
Defining doublepend function

```
function dydt = doublepend(t,y)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
Q=[1000 0 0 0 0 0;
   0 1000 0 0 0 0;
   0 0 100 0 0 0;
   0 0 0 100 0 0;
   0 0 0 0 100 0;
   0 0 0 0 0 100];
R=1;
[K_val, P_mat, Poles] = lqr(A,B,Q,R);
F=-K_val*y;
dydt=zeros(6,1);
```

```
% y(1)=x; y(2)=xdot; y(3)=theta1;   y(4)=theta1dot;  y(5)=theta2;
 y(6)=theta2dot;
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
```

```
dydt =

     0
     0
     0
     0
     0
     0


dydt =

   1.0e-04 *

    0.4019
         0
         0
         0
         0
         0


dydt =

   1.0e-04 *

    0.6028
         0
         0
         0
         0
         0


dydt =

   1.0e-03 *

    0.1608
         0
         0
```

Due to the simulation being run for 5000 tiem steps, all the outputs can't be shown here. However, a PDF containing them all, have been attached in the submission.
Hence, the final output graph is shown below.



Published with MATLAB® R2019b

Part E - Observability

```
%clearing all previous outputs and variables
clc
clear all
%declaring the symbolid variables
syms M m1 m2 l1 l2 g;
% Creating my linearised state space equation using 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)];%Initializing the B matrix
C1 = [1 0 0 0 0 0];  %Corresponding to x component
C2 = [0 0 1 0 0 0; 0 0 0 0 1 0]; %corresponding to theta1 and theta2
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %cooresponding to x and theat2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %cooresponding to x,
 theta1 and theat2
%Matrix to check teh Observability Condition
Ob1 = [C1' A'*C1' A'*A'*C1' A'*A'*A'*C1' A'*A'*A'*A'*C1'
 A'*A'*A'*A'*A'*C1'];
if rank(Ob1)==6
    disp('System is observable, when only x(t) is requested!')
else
    disp('System is not observable, when only x(t) is requested!')
end
%Matrix to check teh Observability Condition
Ob2 = [C2' A'*C2' A'*A'*C2' A'*A'*A'*C2' A'*A'*A'*A'*C2'
 A'*A'*A'*A'*A'*C2'];
if rank(Ob2)==6 %condition for system observability i.e when rank = 6
    disp('System is observable, when only theta1(t) and theta2(t) is
 requested!')
else
    disp('System is not observable, when only theta1(t) and theta2(t)
 is requested!')
end
%Matrix to check teh Observability Condition
Ob3 = [C3' A'*C3' A'*A'*C3' A'*A'*A'*C3' A'*A'*A'*A'*C3'
 A'*A'*A'*A'*A'*C3'];
if rank(Ob3)==6%condition for system observability i.e when rank = 6
    disp('System is observable, when only x(t) and theta2(t) is
 requested!')
else
    disp('System is not observable, when only x(t) and theta2(t) is
 requested!')
end
%Matrix to check teh Observability Condition
Ob4 = [C4' A'*C4' A'*A'*C4' A'*A'*A'*C4' A'*A'*A'*A'*C4'
 A'*A'*A'*A'*A'*C4'];
if rank(Ob4)==6%condition for system observability i.e when rank = 6
```

```
    disp('System is observable, when x(t), theta1(t) and theta2(t) is
 requested!')
else
    disp('System is not observable, when x(t), theta1(t) and theta2(t)
 is requested!')
end
```

```
System is observable, when only x(t) is requested!
System is not observable, when only theta1(t) and theta2(t) is
 requested!
System is observable, when only x(t) and theta2(t) is requested!
System is observable, when x(t), theta1(t) and theta2(t) is requested!
```

Published with MATLAB® R2019b

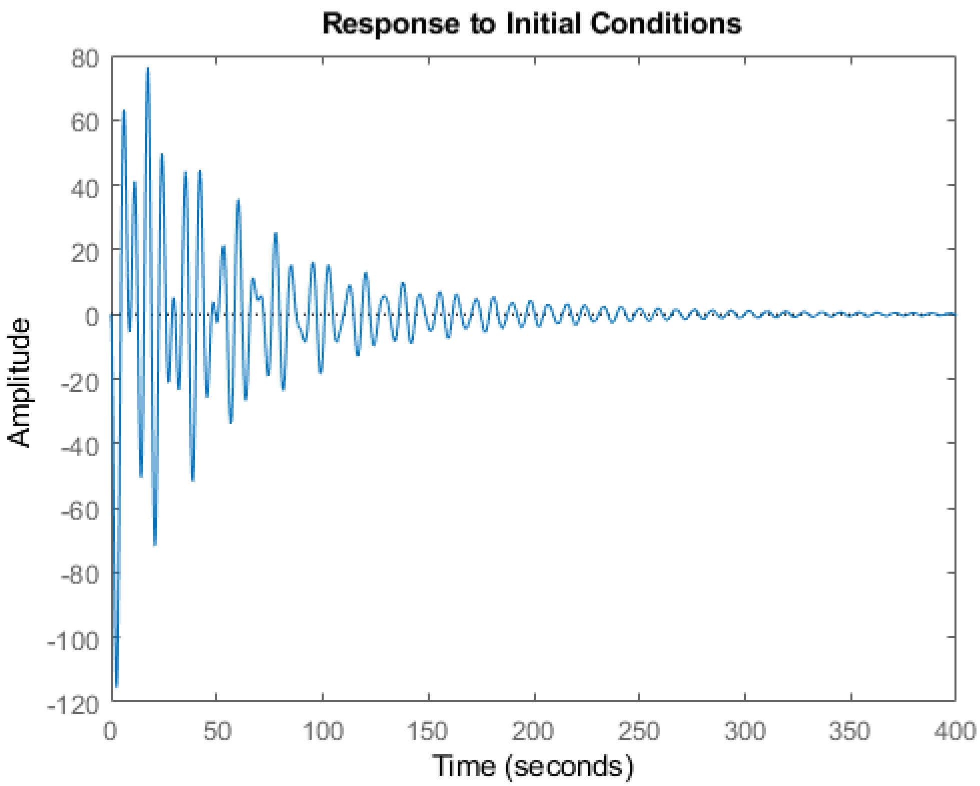
Part F - Luenberger observer for linear systems

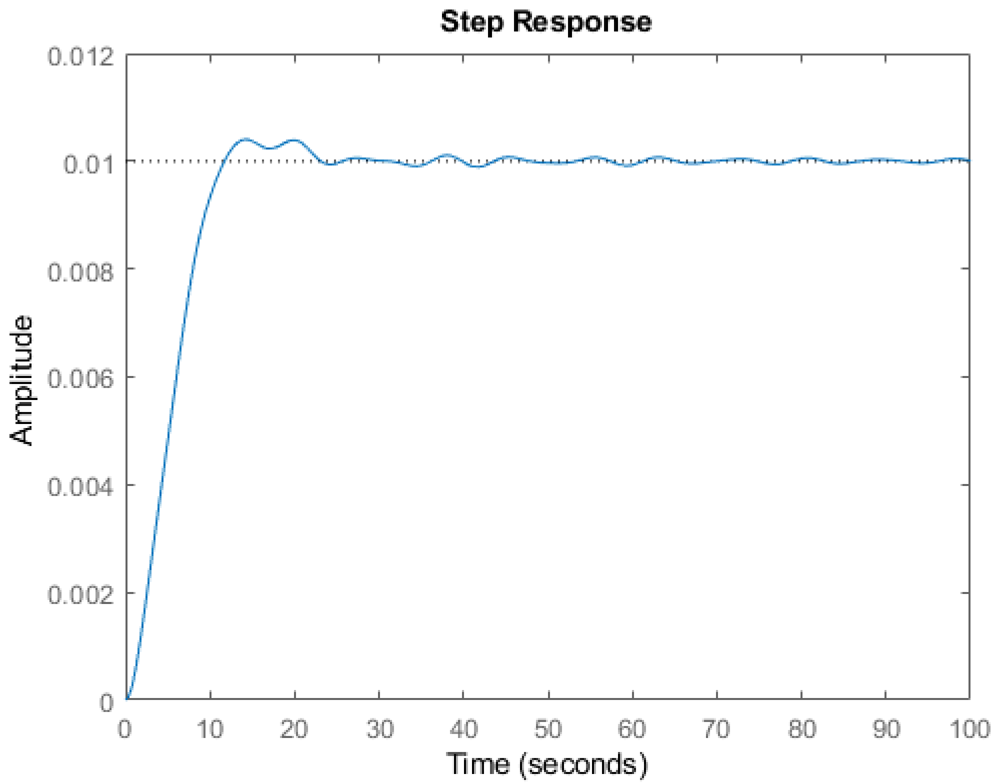
```
%clearing all the previous outputs
clc
clear all
%Substituting values for our M, m1, m2, l1 and l2
M=1000;%Mass of the cart
m1=100;%mass of Pendulum 1
m2=100;%mass of Pendulum 2
l1=20;%length of the string of Pendulum 1
l2=10;%length of the string of Pendulum 2
g=9.81; %declaring the value of the accelertaion due to gravity in m/
s^2
%Defining our matrices as follows
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)];
% From previous case, we have determined that only C1, C3 and C4 were
% observable. Hence, we are going to consider only those 3 cases.
C1 = [1 0 0 0 0 0];  %Corresponding to x component
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %cooresponding to x and theat2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %cooresponding to x,
 theta1 and theat2
D = 0;%declarring the D matrix to be zero
% Cosidering the same Q and R matrices chosen before in our code
Q=[100 0 0 0 0 0;
   0 100 0 0 0 0;
   0 0 1000 0 0 0;
   0 0 0 1000 0 0;
   0 0 0 0 1000 0;
   0 0 0 0 0 1000];
R=0.01; %%these are the cost variables from LQR
% Initial Conditions for Leunberger observer - 12 state variables,
% 6 actual + 6 estimates
x0=[0,0,30,0,60,0,0,0,0,0,0,0];
% state variables order = [x,dx,theta_1,dtheta_1,theta_2,dtheta_2,
% estimates taken in the same order]
% For pole placement, lets choose eigen values with negative real part
poles=[-1;-2;-3;-4;-5;-6];
% Calling LQR function to obtain K matrix
K=lqr(A,B,Q,R);
% Framing L for all three cases where output is observable
% Using the pole placement funciton built into MATLAB
L1 = place(A',C1',poles)' %L1 should be a 6x1 matrix
L3 = place(A',C3',poles)' %L3 should be a 6x2 matrix
L4 = place(A',C4',poles)' %L4 should be a 6x3 matrix

Ac1 = [(A-B*K) B*K; % Luenberger A matrix
```

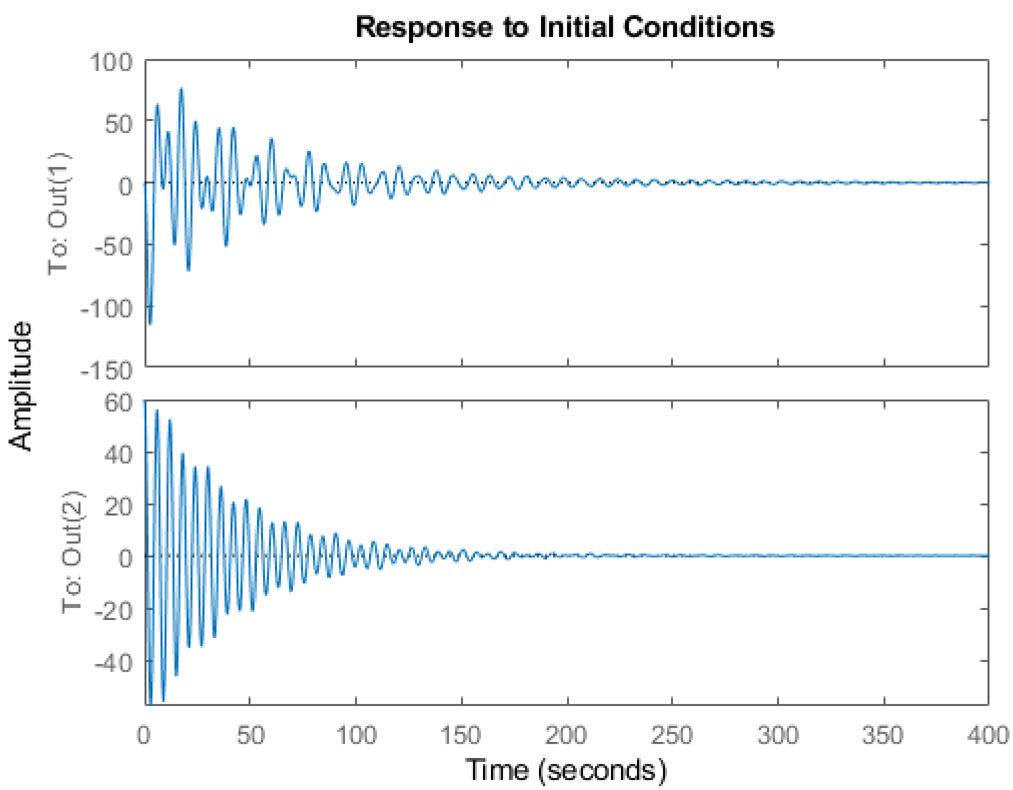
```
        zeros(size(A)) (A-L1*C1)];
Bc = [B;zeros(size(B))];% Luenberger B matrix
Cc1 = [C1 zeros(size(C1))];% Luenberger C matrix

Ac3 = [(A-B*K) B*K;% Luenberger A matrix
        zeros(size(A)) (A-L3*C3)];
Cc3 = [C3 zeros(size(C3))];% Luenberger C matrix

Ac4 = [(A-B*K) B*K;% Luenberger A matrix
        zeros(size(A)) (A-L4*C4)];
Cc4 = [C4 zeros(size(C4))];% Luenberger C matrix

sys1 = ss(Ac1, Bc, Cc1,D);%%MATLAB function to output statespace
 equations
figure % to launch a new figure WINDOW
initial(sys1,x0)%MATLAB inbuilt function to check the initial response
 of the system
figure
step(sys1)%Gives the step response output

sys3 = ss(Ac3, Bc, Cc3,D);%MATLAB function to output statespace
 equations
figure
initial(sys3,x0)%MATLAB inbuilt function to check the initial response
 of the system
figure
step(sys3)%Gives the step response output

sys4 = ss(Ac4, Bc, Cc4, D);%%MATLAB function to output statespace
 equations
figure
initial(sys4,x0)%MATLAB inbuilt function to check the initial response
 of the system
figure
step(sys4)%Gives the step response output

grid on
```

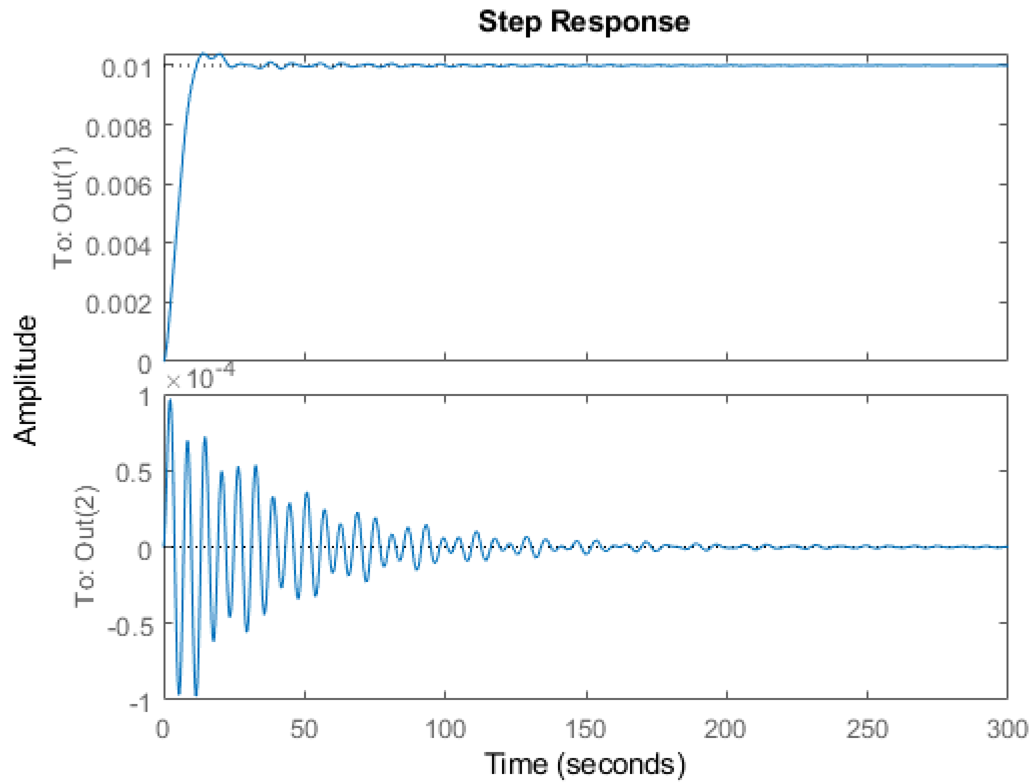
```
L1 =

   1.0e+03 *

    0.0210
    0.1734
   -2.9262
    0.0805
    2.2116
   -1.4493


L3 =

   13.0744   -0.8244
```

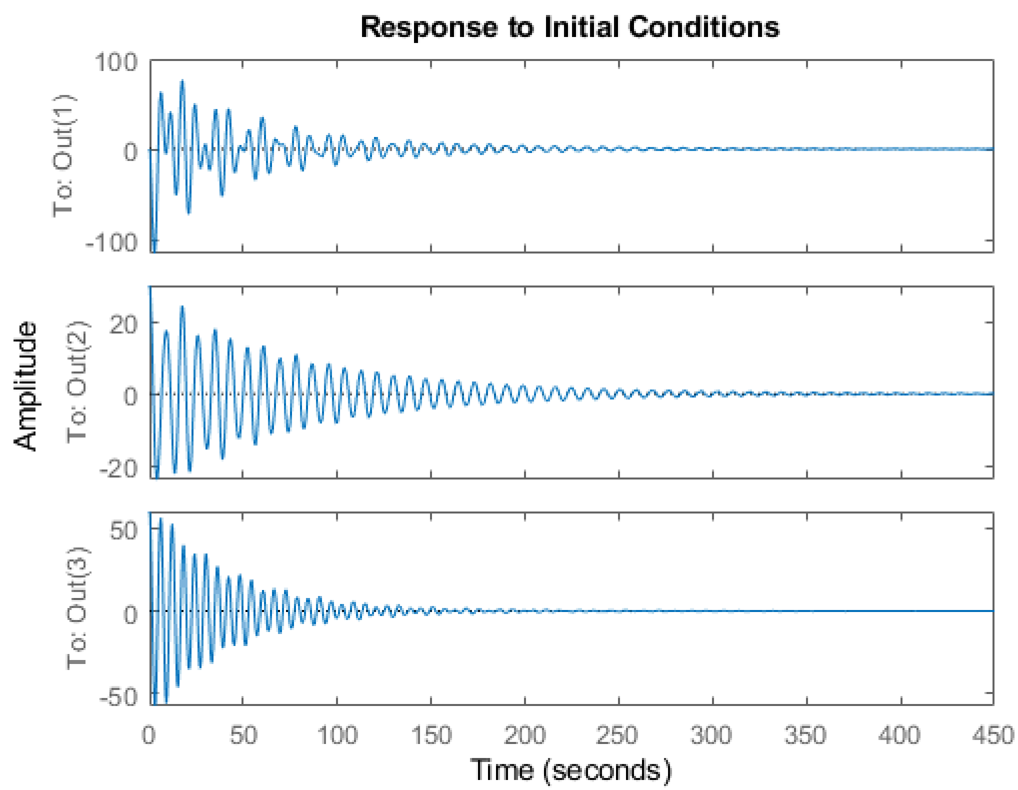
```
   56.2562   -8.4805
  -89.0764   19.7693
  -20.0115   10.9419
    0.3520    7.9256
    3.4793   13.2122


L4 =

    8.5631   -0.8851    0.0000
   17.5219   -4.9484   -0.9810
   -0.9140    9.4369   -0.0000
   -4.1173   20.9385   -0.0491
    0.0000   -0.0000    3.0000
    0.0000   -0.0981    0.9209
```

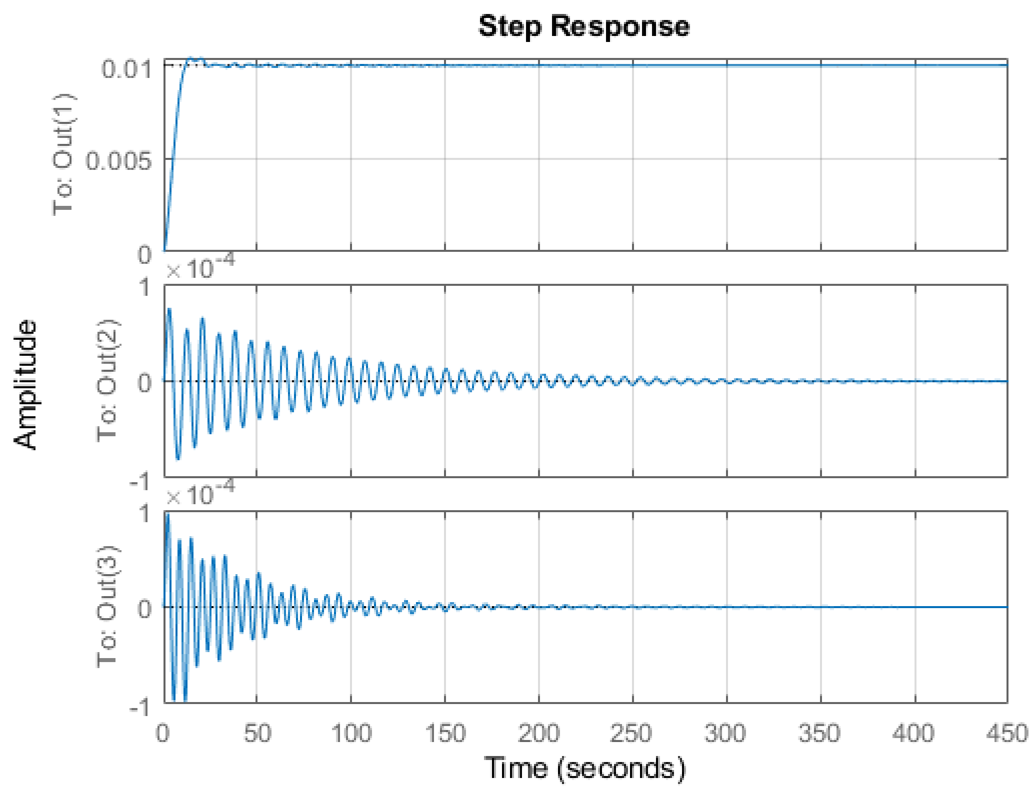



Step Response
Amplitude
Time (seconds)
0.012
0.01
0.008
0.006
0.004
0.002
0
0
10
20
30
40
50
60
70
80
90
100


Response to Initial Conditions
Amplitude
To: Out(1)
To: Out(2)
Time (seconds)
100
50
0
-50
-100
-150
60
40
20
0
-20
-40
0
50
100
150
200
250
300
350
400


Step Response
Amplitude
To: Out(1)
0.01
0.008
0.006
0.004
0.002
0
To: Out(2)
× 10^-4
1
0.5
0
-0.5
-1
0
50
100
150
200
250
300
Time (seconds)


Response to Initial Conditions
Amplitude
To: Out(1)
100
0
-100
To: Out(2)
20
0
-20
To: Out(3)
50
0
-50
0
50
100
150
200
250
300
350
400
450
Time (seconds)

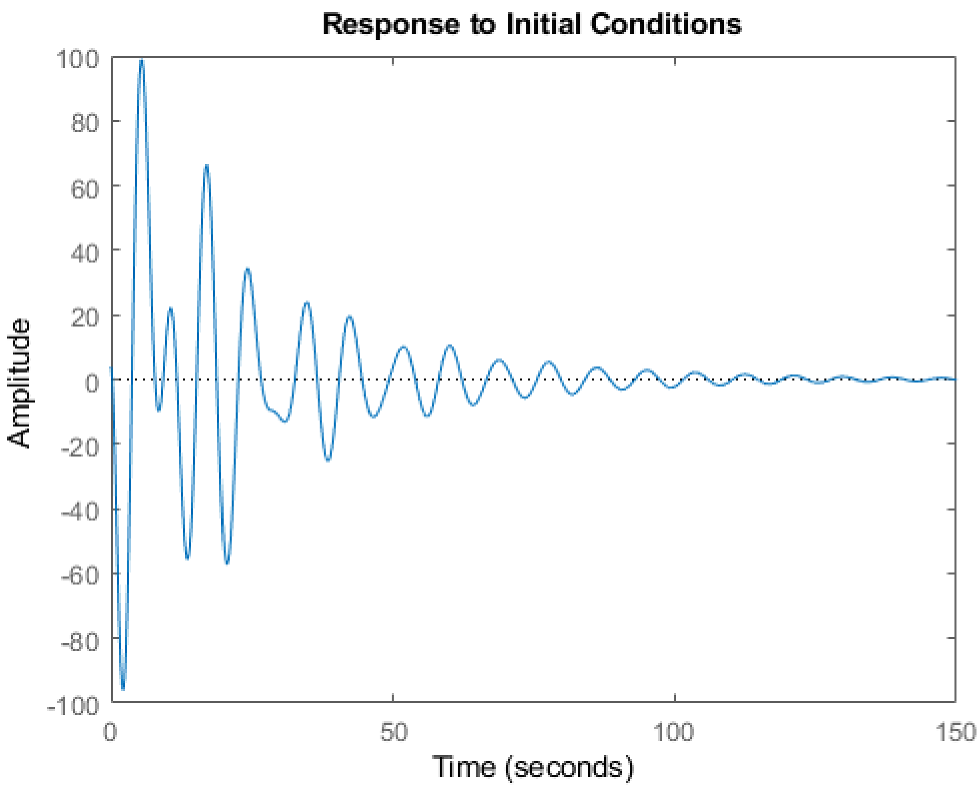


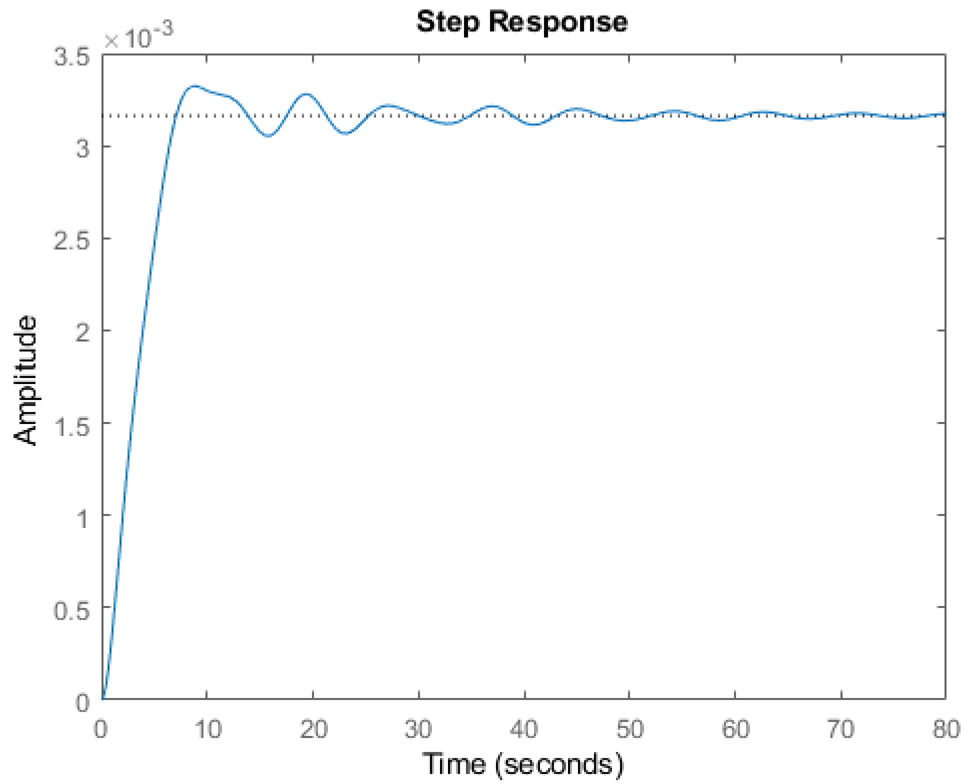
Published with MATLAB® R2019b

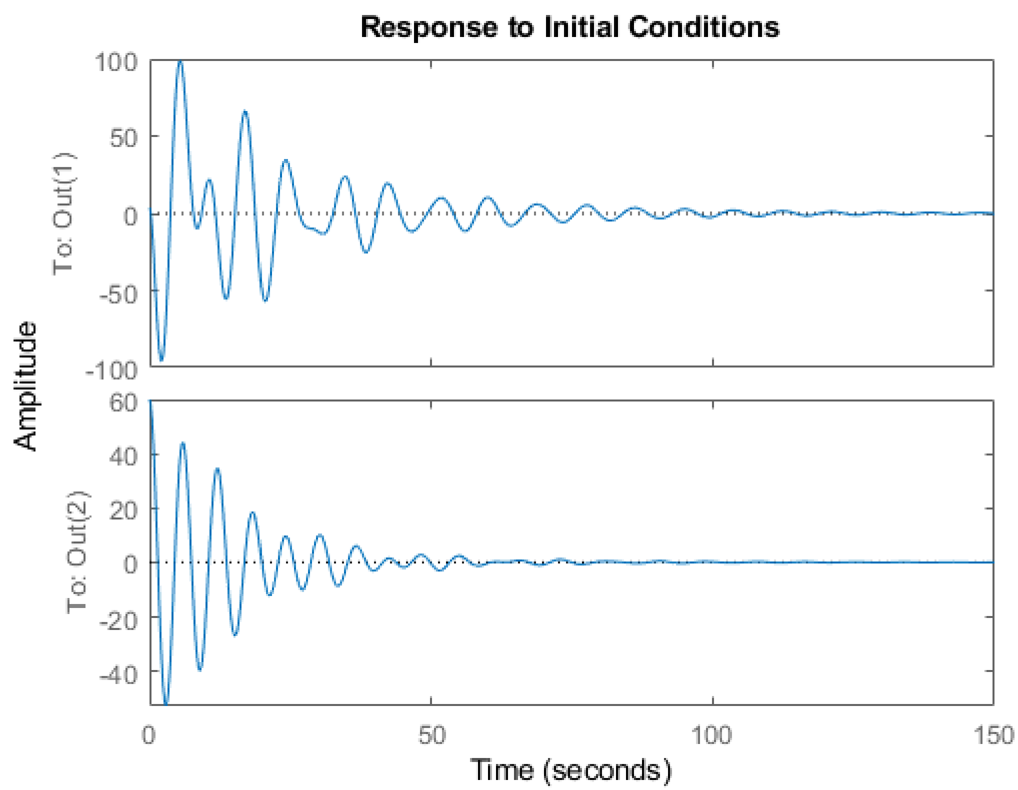
Part G(a) - LQG linear component

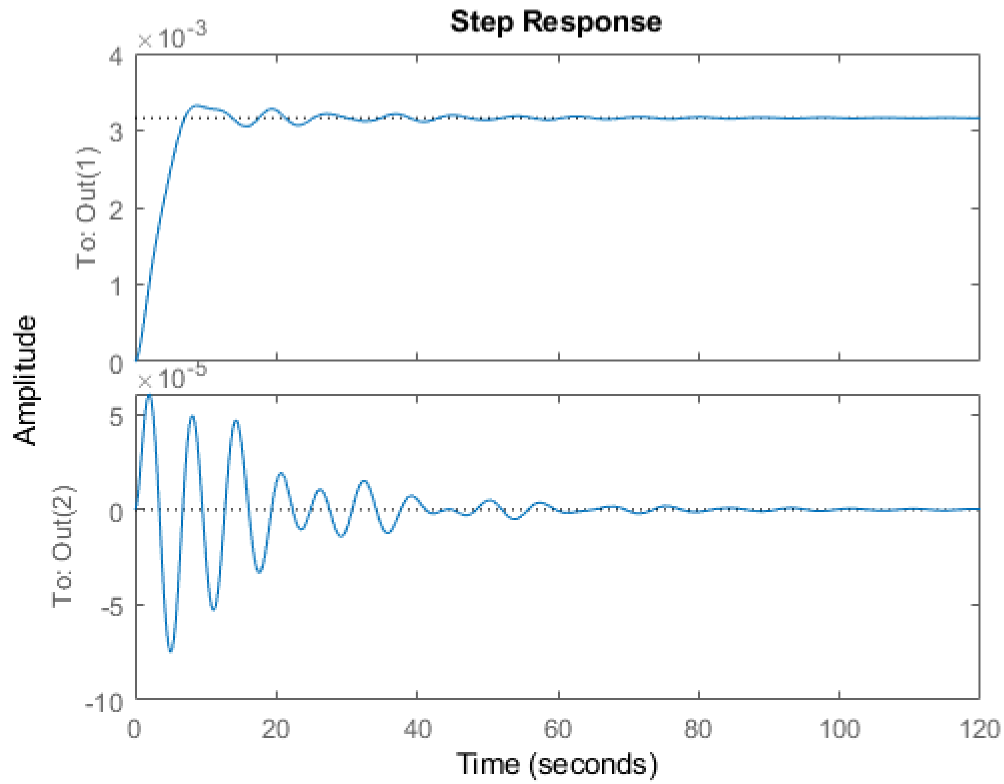
```
%clearing all the previous outputs
clc
clear all
%Substituting values for our M, m1, m2, l1 and l2
m1 = 100; %Mass of the pendulum1
m2 = 100; %Mass of the pendulum2
M = 1000; %Mass of the cart
l1 = 20; %length of the string of Pendulum 1
l2 = 10; %length of the string of Pendulum 2
g = 9.8; %declaring the value of the accelertaion due to gravity in m/
s^2

%Defining our matrices as follows
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)];
% Cosidering the same Q and R matrices chosen before in our code
Q=[100 0 0 0 0 0;
   0 100 0 0 0 0;
   0 0 1000 0 0 0;
   0 0 0 1000 0 0;
   0 0 0 0 1000 0;
   0 0 0 0 0 1000];
R=0.001; %these are the cost variables from LGR
% From previous case, we have determined that only C1, C3 and C4 were
% observable. Hence, we are going to consider only those 3 cases.
C1 = [1 0 0 0 0 0];  %Corresponding to x component
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0]; %corresponding to x and theat2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0]; %corresponding to x,
 theta1 and theat2
D = 0;
% Initial Conditions for Leunberger observer - 12 state variables,
% 6 actual + 6 estimates
x0 = [4;0;30;0;60;0;0;0;0;0;0;0];
% Calling LQR function to obtain K matrix
K =lqr(A,B,Q,R);
vd=0.3*eye(6); %process noise
vn=1; %measurement noise

K_pop1=lqr(A',C1',vd,vn)'; %gain matrix of kalman filter for C1
K_pop3=lqr(A',C3',vd,vn)'; %gain matrix of kalman filter for C3
K_pop4=lqr(A',C4',vd,vn)'; %gain matrix of kalman filter for C4

% Observing state space corresponding C1 observable system
sys1 = ss([(A-B*K) B*K; zeros(size(A)) (A-K_pop1*C1)],
 [B;zeros(size(B))],[C1 zeros(size(C1))], D);
```

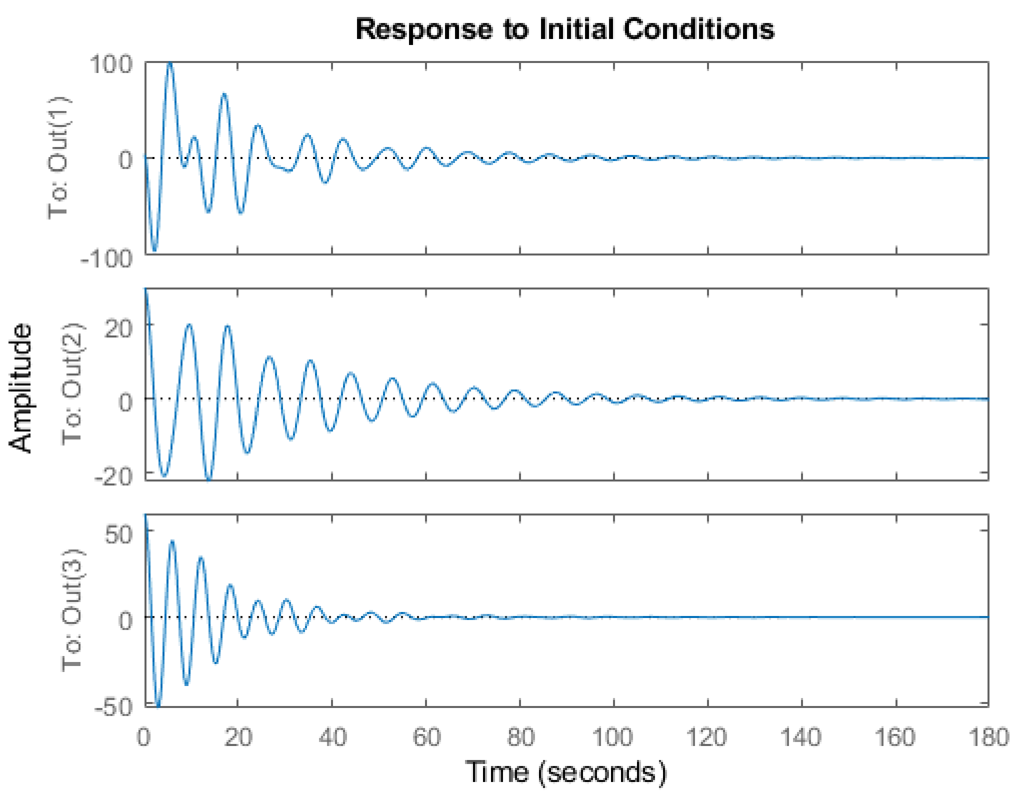
```
figure
initial(sys1,x0)
figure
step(sys1)

% Observing state space corresponding C3 observable system
sys3 = ss([(A-B*K) B*K; zeros(size(A)) (A-K_pop3*C3)],
 [B;zeros(size(B))],[C3 zeros(size(C3))], D);
figure
initial(sys3,x0)
figure
step(sys3)

% Observing state space corresponding C4 observable system
sys4 = ss([(A-B*K) B*K; zeros(size(A)) (A-K_pop4*C4)],
 [B;zeros(size(B))],[C4 zeros(size(C4))], D);
figure
initial(sys4,x0)
figure
step(sys4)

grid on
```

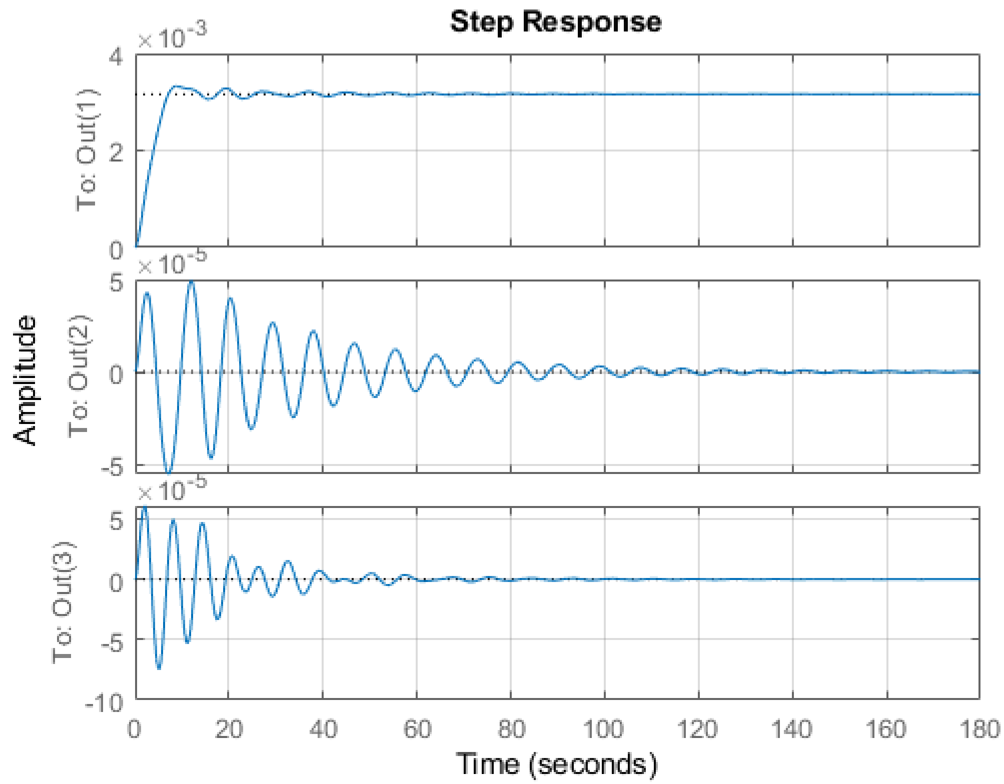



Step Response
×10^-3
3.5
3
2.5
2
1.5
1
0.5
0
Amplitude
0
10
20
30
40
50
60
70
80
Time (seconds)


Response to Initial Conditions
100
50
0
-50
-100
To: Out(1)
60
40
20
0
-20
-40
To: Out(2)
Amplitude
0
50
100
150
Time (seconds)


Step Response
×10^-3
4
3
2
1
0
To: Out(1)
Amplitude
×10^-5
5
0
-5
-10
To: Out(2)
0
20
40
60
80
100
120
Time (seconds)


Response to Initial Conditions
100
0
-100
To: Out(1)
Amplitude
20
0
-20
To: Out(2)
50
0
-50
To: Out(3)
0
20
40
60
80
100
120
140
160
180
Time (seconds)



Published with MATLAB® R2019b

Part G(b) - LQG controller Non-Linear Calculations

```
clc
clear all

x_initial = [0;0;30;0;60;0;0;0;0;0;0;0]
tspan=0:0.1:100;
[t,x] = ode45(@doublepend_lqg,tspan,x_initial);
plot(t,x)
grid on
```

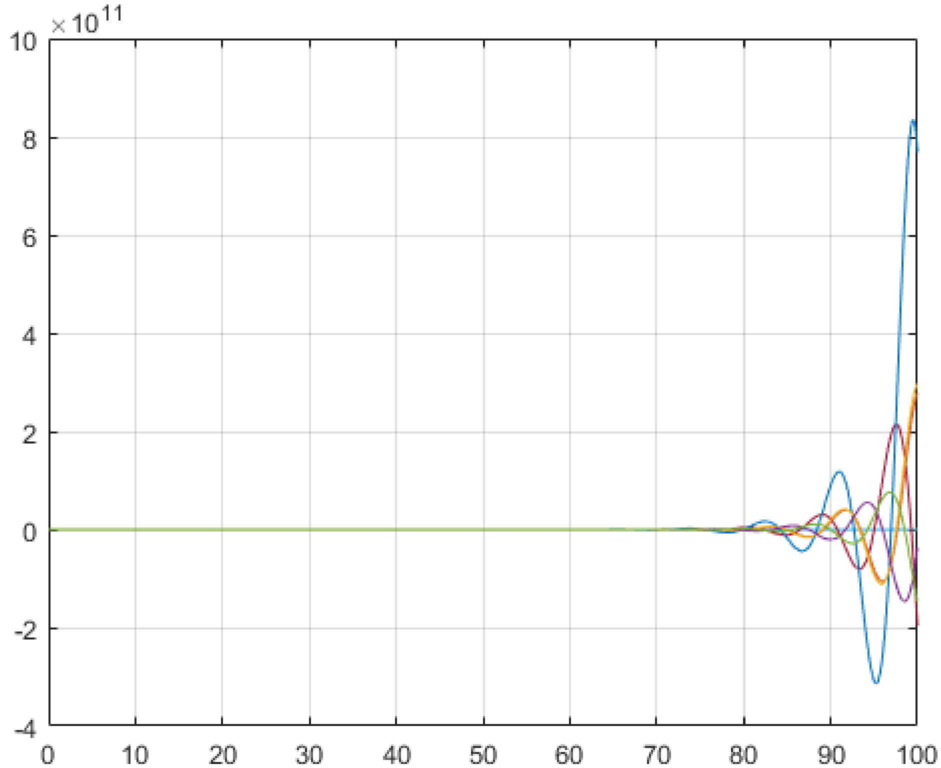
```
x_initial =

     0
     0
    30
     0
    60
     0
     0
     0
     0
     0
     0
     0
```

Function doublepend_lqg for non linear LQG control

```
function dydt = doublepend_lqg(t,y)
M=1000;
m1=100;
m2=100;
l1=20;
l2=10;
g=9.81;
A=[0 1 0 0 0 0;
    0 0 -(m1*g)/M 0 -(m2*g)/M 0;
    0 0 0 1 0 0;
    0 0 -((M+m1)*g)/(M*l1) 0 -(m2*g)/(M*l1) 0;
    0 0 0 0 0 1;
    0 0 -(m1*g)/(M*l2) 0 -(g*(M+m2))/(M*l2) 0];
B=[0; 1/M; 0; 1/(M*l1); 0; 1/(M*l2)];
Q=[1000 0 0 0 0 0;
   0 1000 0 0 0 0;
   0 0 100 0 0 0;
   0 0 0 100 0 0;
```

```
   0  0  0  0  100  0;
   0  0  0  0  0  100];
R=1;% From previous case, we have determined that only C1, C3 and C4
 were
% observable. Hence, we are going to consider only those 3 cases.
C1 = [1  0  0  0  0  0];  %Corresponding to x component
% C1 = [1  0  0  0  0  0;  0  0  0  0  1  0]; %corresponding to x and theat2
% C1 = [1  0  0  0  0  0;  0  0  1  0  0  0;  0  0  0  0  1  0]; %corresponding to x,
 theta1 and theat2
D = 0;
K_val = lqr(A,B,Q,R);
F=-K_val*y(1:6);

vd=0.3*eye(6);
vn=1;
K_pop=lqr(A',C1',vd,vn)';

sd =(A-K_pop*C1)*y(7:12);

dydt=zeros(12,1);
% y(1)=x; y(2)=xdot; y(3)=theta1;   y(4)=theta1dot;  y(5)=theta2;
 y(6)=theta2dot;
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;
dydt(7)= y(2)-y(10)
dydt(8)= dydt(2)-sd(2)
dydt(9)= y(4)-y(11)
dydt(10)= dydt(4)-sd(4)
dydt(11)= y(6)-y(12)
dydt(12)= dydt(6)-sd(6)
end
```

```
dydt =

     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
```



Published with MATLAB® R2019b