ELEN 236

LINEAR CONTROL SYSTEMS
WINTER 2018

PROJECT REPORT

-Saravana Kumar Phillip Nagarajan

TABLE OF CONTENT:

1. Abstract

This document provides the system model, simulation and analysis of an electro-mechanical system. The various factors such as: State space, Transfer function, poles, different canonical forms, Controllability, observability, pole placement, designing a controller and an observer for the given system are briefly discussed in this document. It also shows the different state outputs for different conditions with respect to time.

Key Words: System model, Pole placement, Controller, Observer.

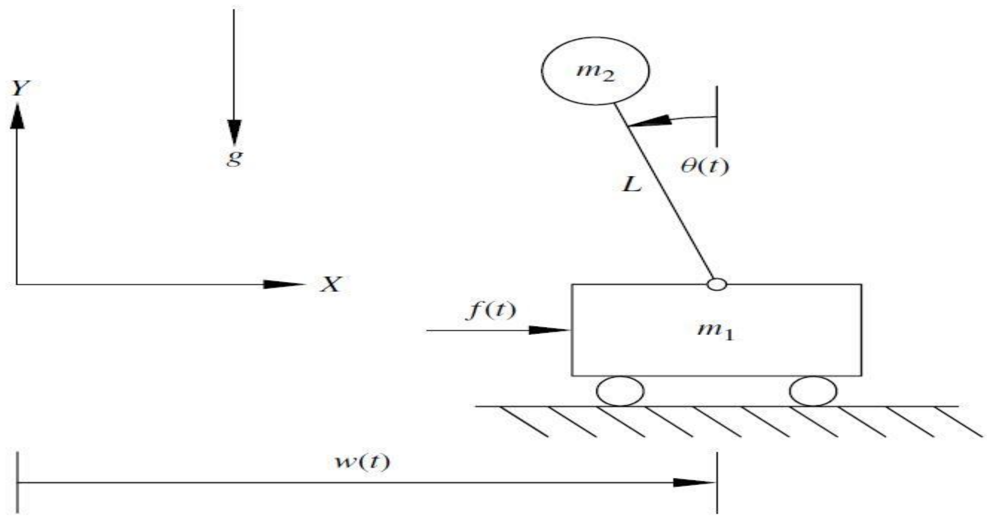
2. Introduction:

The nonlinear, inherently unstable inverted pendulum is given in the figure 2.1.

m1 → Cart mass
m2 → Pendulum point mass
θ(t) → Pendulum angle
w(t) →Cart displacement
f (t) →Input Force
τ(t) →Input torque

Assumption: The pendulum rod is massless.

Here, our aim is to maintain the pendulum angle θ(t) = 0 by using a feedback controller with a sensor for θ(t) and an actuator to produce an input force f (t). There will be two outputs, the pendulum angle θ(t) and the cart displacement w(t). The classical inverted pendulum has only one input, the force f (t). By considering a second case, using a motor to provide a second input τ(t) (not shown) at the rotary joint of figure 1.1. First nonlinear model for this system s derived, i.e., draw the free-body diagrams and writing the correct number of independent ordinary differential equations.



3. Goals and Specification

The goal is to maintain the pendulum angle θ(t) = 0 by using a feedback controller with a sensor (encoder or potentiometer) for θ(t) and an actuator to produce an input force f (t).

For this,

- The state dynamic matrix(A), input matrix(B), output matrix(C), and constant matrix(D) are to be found.
- To check if the system is diagonalizable or not, if it is diagonalizable then the Coordinate Transformations and Diagonal Canonical form are found.

- The system should be checked whether it is controllable or not. If it is controllable then the controller canonical form for the system are found.
- Then, the system is checked for observability or not, and observer canonical form are found.
- The observer canonical form and the controller canonical forms are checked for duality.
- Lyapunov Stability Analysis for the system is checked, i.e. to check whether the system is stable or unstable or marginally stable or asymptotically stable.
- To design a controller and observer for the system by finding the feedback gain and observer gain and to plot all the states and outputs.

4. System model:

Basic mathematical model for a linear time-invariant system consists of the state differential equation and the algebraic output equation:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

Where,

A→ System dynamic matrix

B → Input matrix

C → Output matrix

D → Constant matrix

x(t) → State vector

u(t)→ Input Vector

y(t) → Output vector

- **Case 1: (SISO – Single input – Single Output)**

Single-input: input f (t)

Single-output: output θ(t)

Therefore A, B, C matrix are as follows.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m2g}{m1} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(m1+m2)g}{m1L} & 0 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0 \\ \frac{1}{m1} \\ 0 \\ \frac{1}{m1L} \end{bmatrix} \quad ; \quad C = [1 \quad 0 \quad 0 \quad 0]$$

- **Case 2: (SIMO – Single input – Multiple Output)**

Single-input: input f (t)

Multiple-output: two outputs w(t) and θ(t)

Therefore A, B, C matrix are as follows.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m2g}{m1} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(m1+m2)g}{m1L} & 0 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0 \\ \frac{1}{m1} \\ 0 \\ \frac{1}{m1L} \end{bmatrix} \quad ; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- **Case 3: (MIMO – Multiple input – Multiple Output)**

Multiple-input: Input force f (t) and Input torque τ(t)

Multiple-output: two outputs w(t) and θ(t)

Therefore A, B, C matrix are as follows.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m2g}{m1} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(m1+m2)g}{m1L} & 0 \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 \\ \frac{1}{m1} & \frac{1}{m1L} \\ 0 & 0 \\ \frac{1}{m1L} & \frac{(m1+m2)}{m1m2L} \end{bmatrix} ; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Table (1): Numerical Parameters for the system

Parameter	**Value**	**Units**
m1	2	kg
m2	1	kg
L	0.75	m
g	9.81	m/s^2

Throughout this project, the following numerical parameters are used.

#Matlab code:

```
m1 = 2;
m2 = 1;
L = 0.75;
g = 9.81;
A = [0 1 0 0;0 0 (m2*g)/(m1) 0;0 0 0 1;0 0 ((m1+m2)*g)/(L*m1) 0];
B = [0; 1/m1; 0; 1/(L*m1)];
C = [1 0 0 0];
D = 0;
Statespace=ss(A,B,C,D);
Transferfunction=tf(Statespace);
[num,den]=tfdata(Statespace,'v');
```

#State space Equations for all three cases:

- **Case 1: (SISO – Single input – Single Output)**

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 4.905 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 19.62 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.500 \\ 0 \\ 0.667 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(t)$$

- **Case 2: (SIMO – Single input – Multiple Output)**

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 4.905 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 19.62 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.500 \\ 0 \\ 0.667 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t)$$

- **Case 3: (MIMO – Multiple input – Multiple Output)**

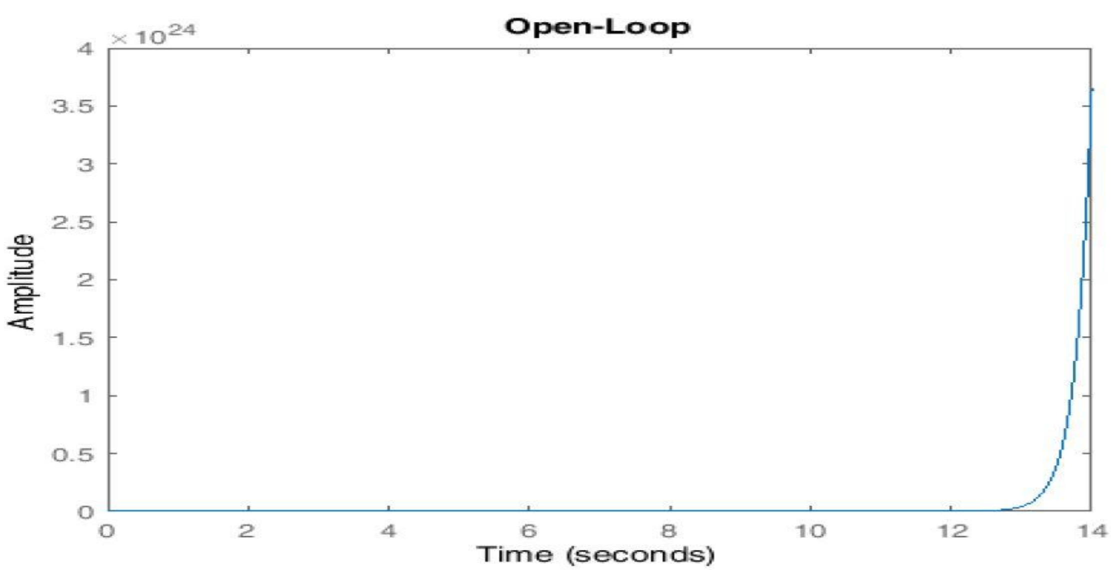
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 4.905 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 19.62 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0.500 & 0.667 \\ 0 & 0 \\ 0.667 & 2.667 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t)$$

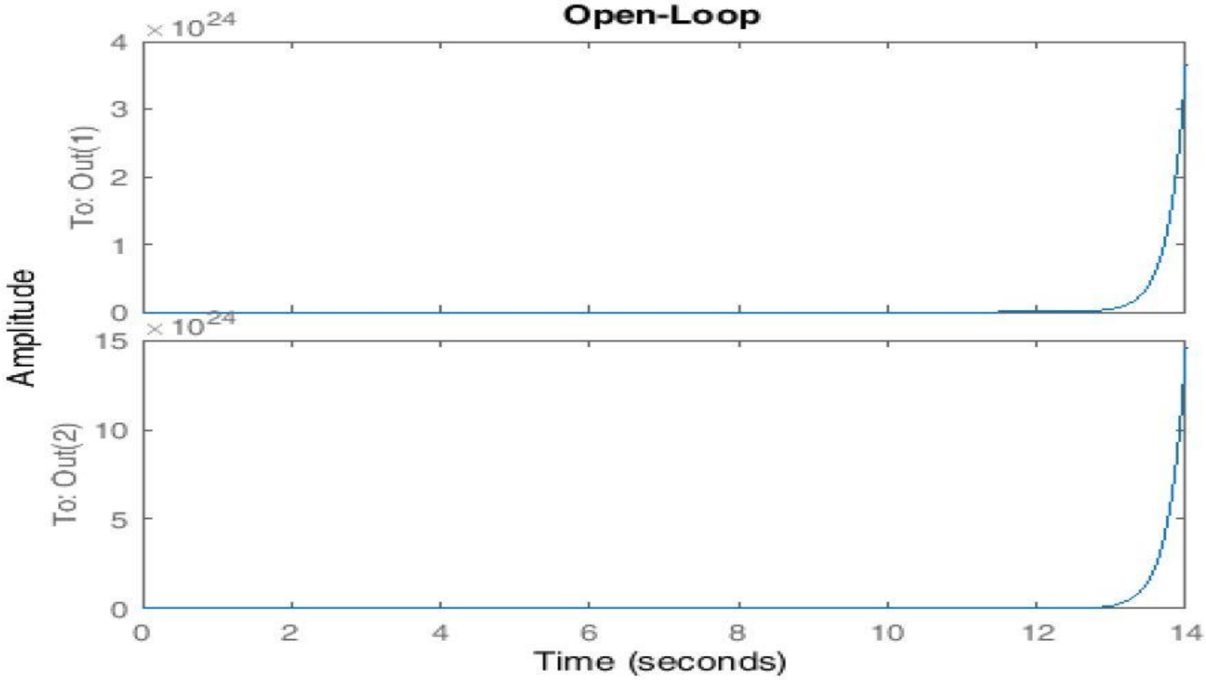
5. Diagonal Canonical Form and Open loop graphs:

A) #Plotting the open-loop state variable,

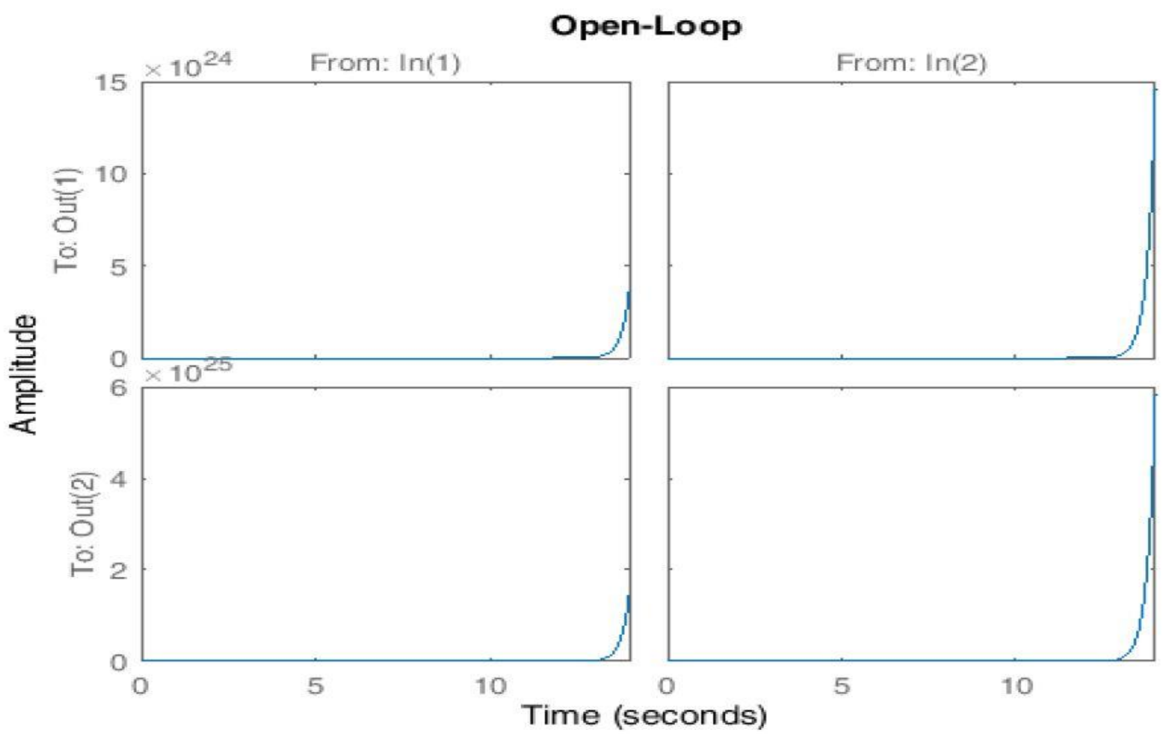
- **For Case 1: (SISO – Single input – Single Output)**



- **For case 2: (SIMO – Single input – Multiple Output)**



- **For case 3: (MIMO – Multiple input – Multiple Output)**



#Eigen Values of the system:

```
Eigs0 =

         0
         0
    4.4294
   -4.4294
```

B) Diagonal Canonical Form for case 1:

```
t=[0:.01:4];
U=[zeros(size(t))];
x0=[0;0;0;0];
CharPoly = poly(A);
Poles = roots(CharPoly);
Eigs0=eig(A);
[V,d] = eig(A);
[Yo,t,Xo]=lsim(Statespace,U,t,x0);
figure;
subplot(211), plot(t,Xo(:,1)); grid;
axis([0 4 -0.2 0.5]);
set(gca,'FontSize',18);
ylabel('{\itx}?1 (\itrad)')
subplot(212), plot(t,Xo(:,2)); grid; axis([0 4 -2 1]);
set(gca,'FontSize',18);
xlabel('\ittime (sec)'); ylabel('{\itx}?2 (\itrad/s)');
[Tdcf,E] = eig(A);
Adcf = inv(Tdcf)*A*Tdcf;
```

- Here, Diagonal canonical form **cannot be found**.
- Because A has **repeated eigenvalues** and the **eigenvectors** are **not independent**.
- This means that **A** is **not diagonalizable** and is, therefore, defective.

5. Controllability check and Controller Canonical Form:

(A) Controllability Check

- **Case 1: Single input – Single output**

```
 P = ctrb(Statespace);
 if (rank(P) == size(A,1))
disp('System is controllable');
else
disp('System is NOT controllable');
 end
```

The system is **controllable**.

- **Mathematical Explanation:**

```
>> P = [B A*B (A^2)*B (A^3)*B]

P =

         0    0.5000         0    3.2700
    0.5000         0    3.2700         0
         0    0.6667         0   13.0800
    0.6667         0   13.0800         0

>> det(P)

ans =

   19.0096
```

Since the determinant(P) is **not equal to zero**, the system is **controllable**.

- **Case 2: Single input – Multiple output**

```
P = ctrb(Statespace);
if (rank(P) == size(A,1))
disp('System is controllable');
else
disp('System is NOT controllable');
end
```

The system is **controllable**.

- **Mathematical Explanation:**

```
>> P = [B A*B (A^2)*B (A^3)*B]

P =

         0    0.5000         0    3.2700
    0.5000         0    3.2700         0
         0    0.6667         0   13.0800
    0.6667         0   13.0800         0

>> det(P)

ans =

   19.0096
```

- Since the determinant(P) is **not equal to zero**, the system is **controllable**.

- **Case 3: Multiple input – Multiple output**

```
P = ctrb(Statespace);
if (rank(P) == size(A,1))
disp('System is controllable');
else
disp('System is NOT controllable');
end
```

The system is **controllable.**

B) Controller canonical form for case 1:

```
CharPoly = poly(A);
a1 = CharPoly(2);
a2 = CharPoly(3);
a3 = CharPoly(4);
Pccfi = [a1 a2 a3 1;a2 a3 1 0;a3 1 0 0; 1 0 0 0];
Tccf = P*Pccfi;
Accf = inv(Tccf)*A*Tccf;
Bccf = inv(Tccf)*B;
Cccf = C*Tccf;
Dccf = D;
```

- Here, Accf, Bccf, Cccf, and Dccf can be written as,

```
Accf =

     0    1.0000         0         0
     0         0    1.0000         0
     0         0         0    1.0000
     0         0   19.6200         0
```

```
Bccf =

         0
   -0.0000
         0
    1.0000
```

```
Cccf =

   -6.5400         0    0.5000         0
```

- Therefore, Controller Canonical Form can be represented as,

$$\dot{x}_{CCF}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 19.62 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y_{CCF}(t) = [-6.54 \quad 0 \quad 0.50 \quad 0]\, x(t)$$

- Here, the controllable canonical form arranges the coefficients of the transfer function denominator across one row of the A matrix and numerator across the C matrix.
- The controllable canonical from is useful for the pole placement controller design technique.

6. Observability check and Observable Canonical Form:

A) Observability check:

- **Case 1: Single input – Single output**

```
Q = obsv(Statespace);
if (rank(Q) == size(A,1))
disp('System is observable');
else
disp('System is NOT observable');
end
```

The 'system is **observable.'** Since determinant of Observability matrix (Q) is not equal to zero.

- **Case 2: Single input – Multiple output**

```
Q = obsv(Statespace);
if (rank(Q) == size(A,1))
disp('System is observable');
else
disp('System is NOT observable');
end
```

The 'system is **observable.'** Since determinant of Observability matrix (Q) is not equal to zero.

- **Case 3: Multiple input – Multiple output**

```
Q = obsv(Statespace);
if (rank(Q) == size(A,1))
disp('System is observable');
else
disp('System is NOT observable');
end
```

The 'system is **observable.'** Since determinant of Observability matrix (Q) is not equal to zero.

B) Observer canonical form for case 1:

```
Qocf = inv(Pccfi);
Tocf = inv(Q)*Qocf;
Aocf = inv(Tocf)*A*Tocf;
Bocf = inv(Tocf)*B;
Cocf = C*Tocf;
Docf = D;
```

- Here, Aocf, Bocf, Cocf and Docf can be written as,

```
Aocf =

         0         0         0         0
    1.0000         0         0         0
         0    1.0000         0   19.6200
         0         0    1.0000         0
```

```
Bocf =

   -6.5400
         0
    0.5000
         0
```

```
Cocf =

     0     0     0     1
```

- Therefore, Observer Canonical Form can be represented as,

$$\dot{x}_{OCF}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 19.62 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} -6.54 \\ 0 \\ 0.50 \\ 0 \end{bmatrix} u(t)$$

$$y_{OCF}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x(t)$$

#Duality relation between CCF and OCF:

Aocf matrix is the transpose of Accf matrix, Bocf matrix is the transpose of Cccf matrix and Cocf matrix is the transpose of Bccf matrix. Therefore it is written as,

$$\mathrm{A}_{OCF} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 19.62 \\ 0 & 0 & 1 & 0 \end{bmatrix} = (\mathrm{A}_{CCF})^T$$

$$\mathrm{B}_{OCF} = \begin{bmatrix} -6.54 \\ 0 \\ 0.50 \\ 0 \end{bmatrix} = (\mathrm{C}_{CCF})^T$$

$$\mathrm{C}_{OCF} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} = (\mathrm{B}_{CCF})^T$$

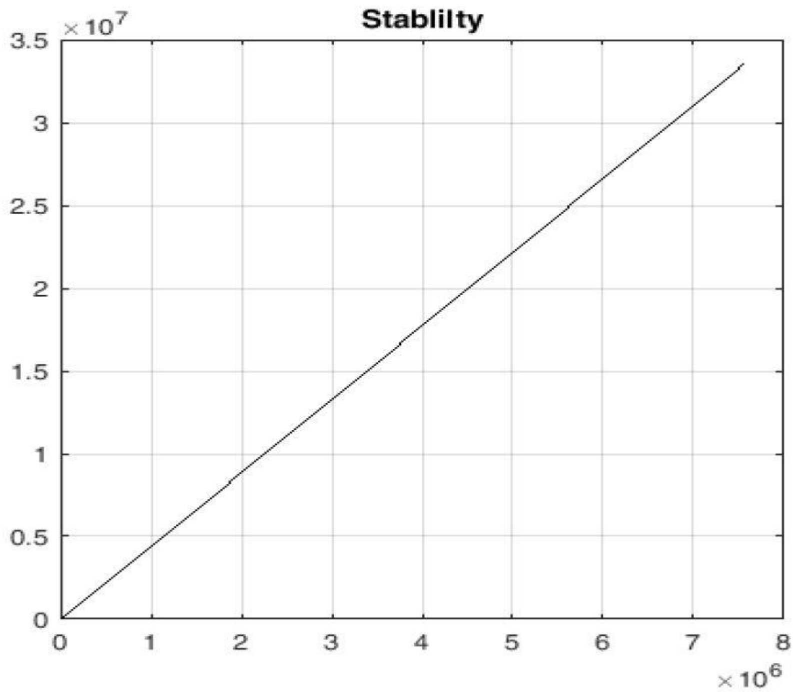
7. Lyapunov Stability Analysis:

A) Checking for Lyapunov Stability Analysis

#For all cases:

```
if (real(poles(1))>=0 | real(poles(2))>=0 | real(poles(3))>=0 | real(poles(4))>=0);
if (real(poles(1)) > 0 | real(poles(2)) > 0 | real(poles(3))>0 | real(poles(4))>0);
disp('System is unstable.');
else
disp('System is marginally stable.');
end
else
if (real(poles(1)) <0 | real(poles(2))<0 | real(poles(3))<0 | real(poles(4))<0);
disp('System is asymptotically stable.');
end
end
```

- The 'system is **Unstable.'**
- Because, there is one positive value in the eigen values that we have calculated for A matrix. So, stability is unstable for all three cases.



#Lyapunov stability analysis will not succeed,

- Because Lyapunov Stability is only a necessary condition and not a sufficient condition to prove stability. There are infinite number of candidate functions to choose from, for Lyapunov analysis.
- So, if the analysis show that the system is stable, then the candidate function chosen is right. Rather, if it shows that the system is unstable, the candidate function chosen might be either wrong or right i.e. the system could be either stable or unstable respectively.
- So, the Lyapunov stability analysis is reliable only when it says that the system is stable.

8. Design of linear control feedback law (K):

A) Desired Eigen values

The four desired eigen values are as follows,

Desired eigen values (1,2) = -1.27 + 3.79j, -1.27 - 3.79j

Desired eigen values (3,4) = -1.88 + 1.24j, -1.88 - 1.24j

B) Designing a State Feedback Control Law (Calculating K)

- **For Case1 and 2:**

```
DesEig = [-1.27 + 3.79j;-1.27 - 3.79j;-1.88 + 1.24j;-1.88 - 1.24j]
K = place(A,B,DesEig)
Ac = A-B*K;
Bc = B;
Cc = C;
Dc = D;
StateSpacec = ss(Ac,Bc,Cc,Dc);
TransferC = tf(StateSpacec);
figure;
step(TransferC);
figure;
step(Transfer,TransferC,10);
legend('Open-loop','Closed-loop');
grid;
axis([0 5 -0.2 0.5]);
```

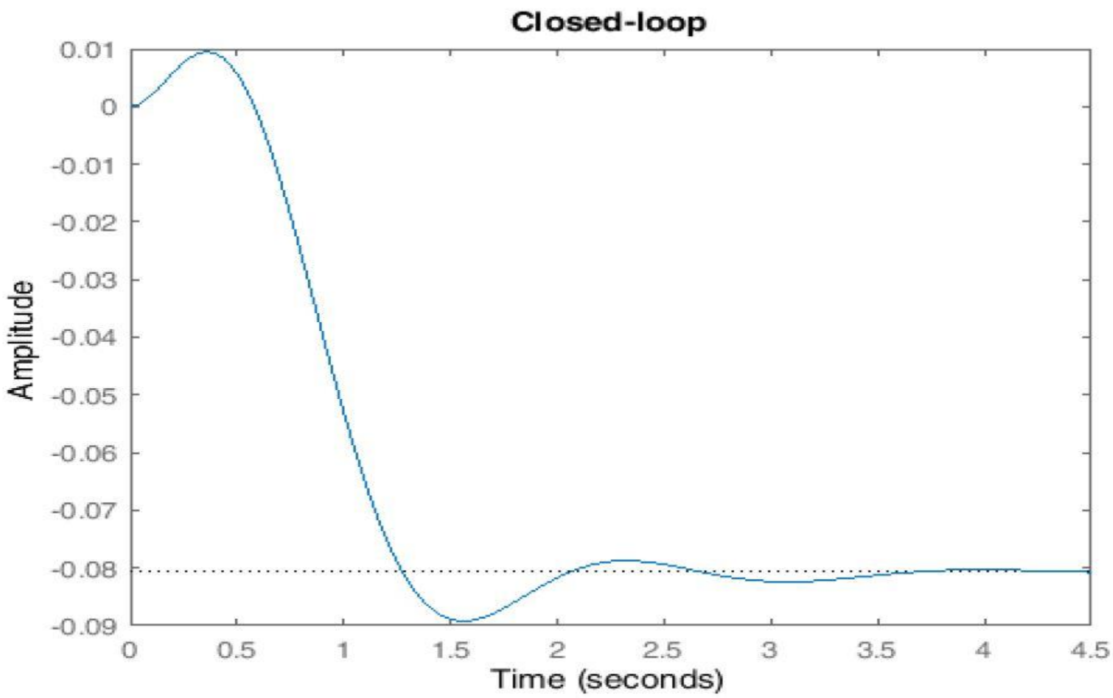
```
K =

  -12.3907  -11.1554   84.6221   17.8166
```

- **For case 3:**

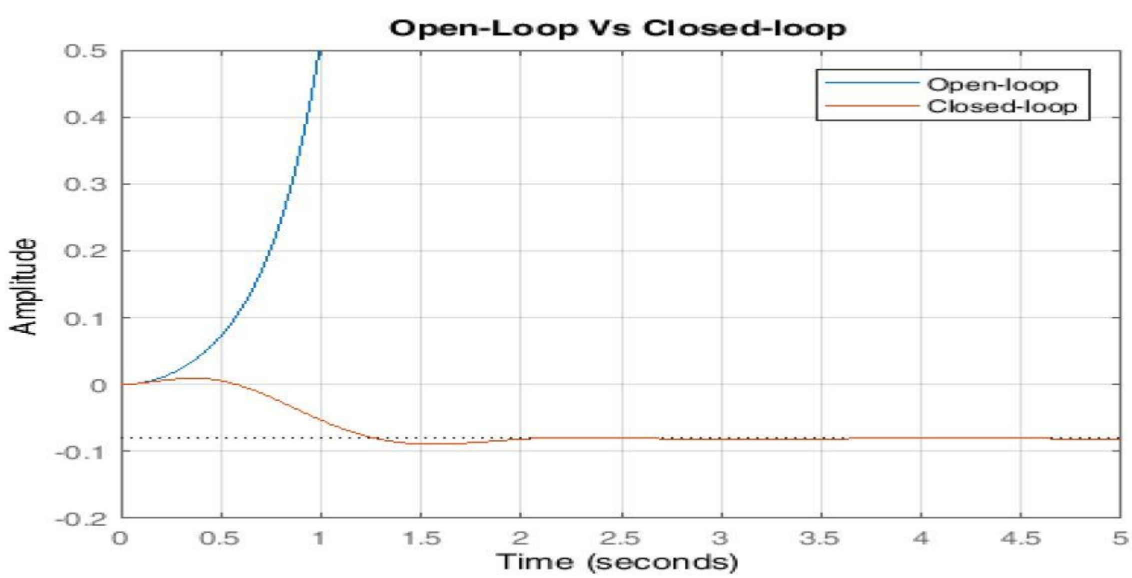
```
K =

19.3473     8.6386    16.4966     5.1908
-6.9171    -3.1153     5.8938    -0.1168
```

- **Case 1: (Single input – Single output)**

#Closed-loop graph:

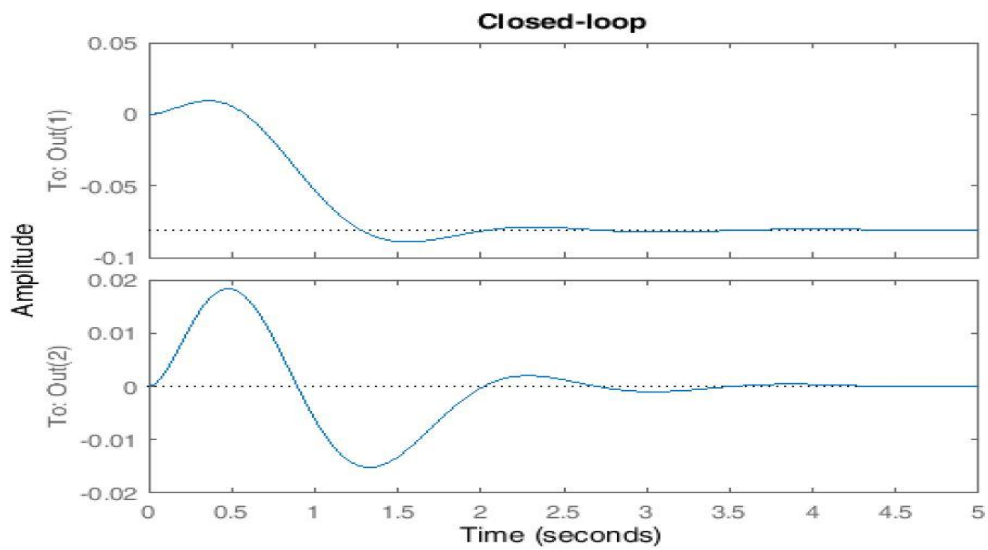


#Open-loop vs Closed-loop graph:

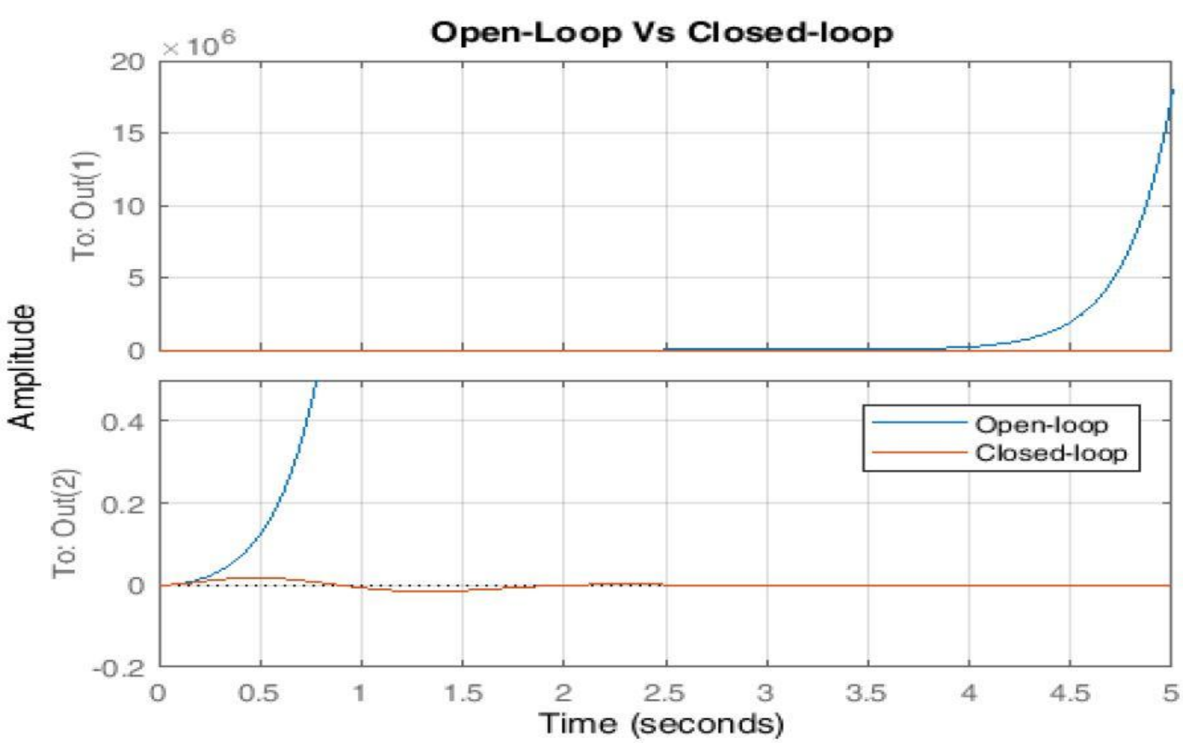


- **Case 2: (Single input – Multi output)**

#Closed loop graph:

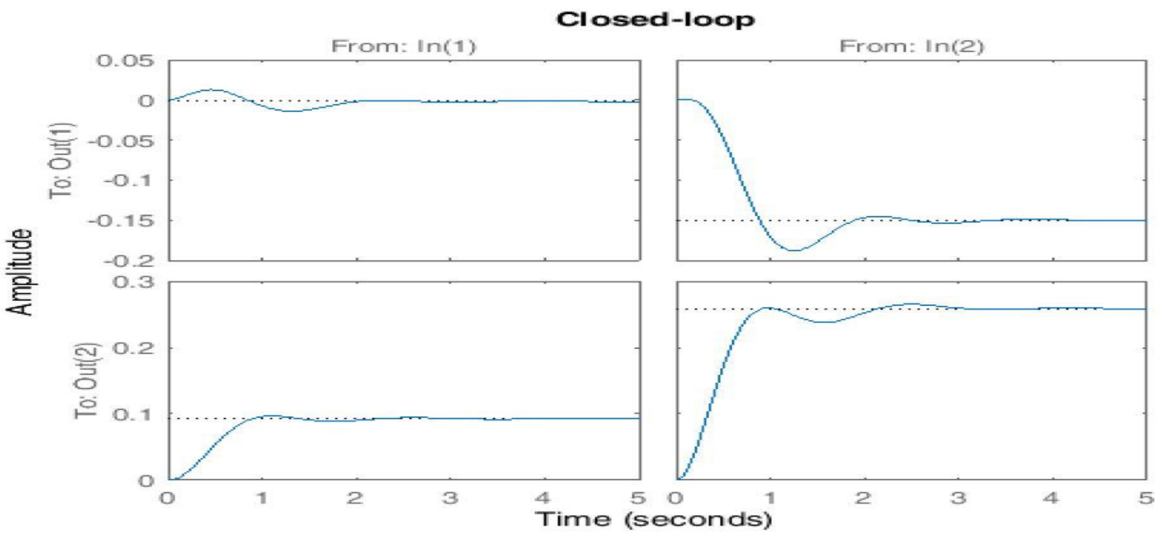


#Open loop vs Closed loop graph:

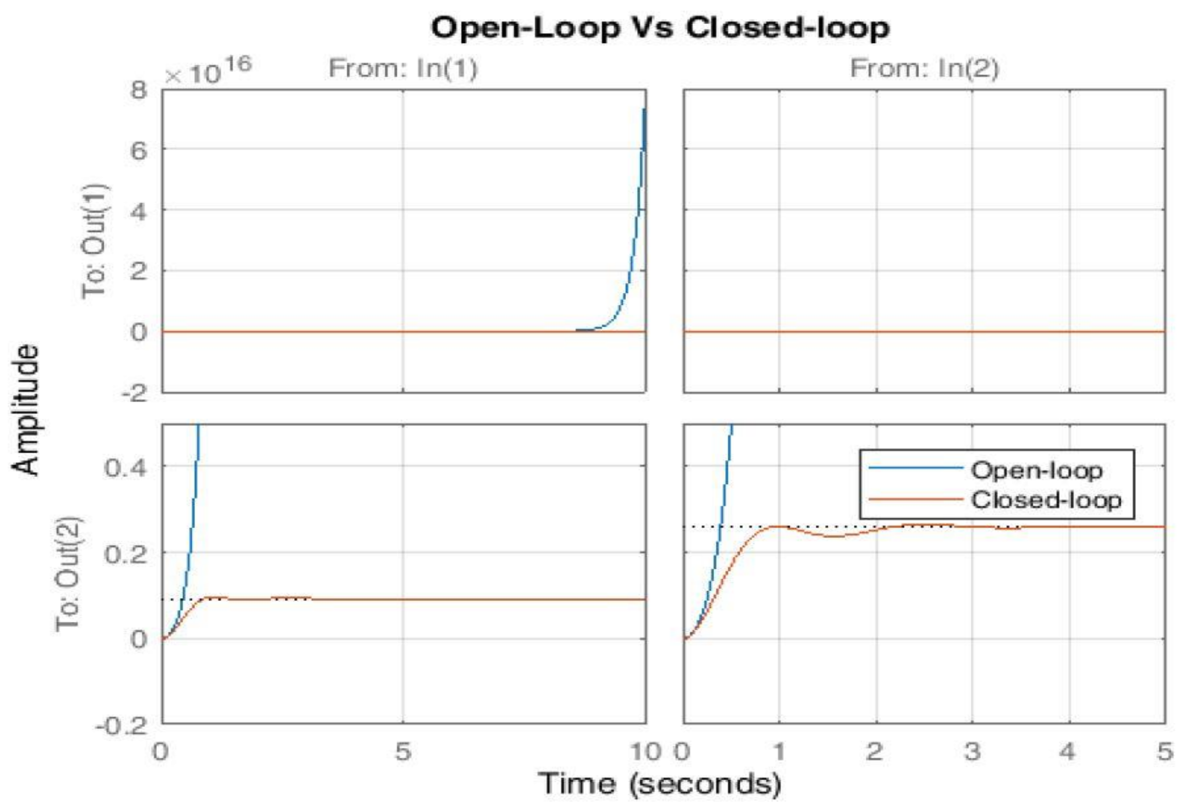


- **Case 3: (Multiple input – Multiple Output)**

#Closed-loop graph:



#Open loop vs Closed loop graph:

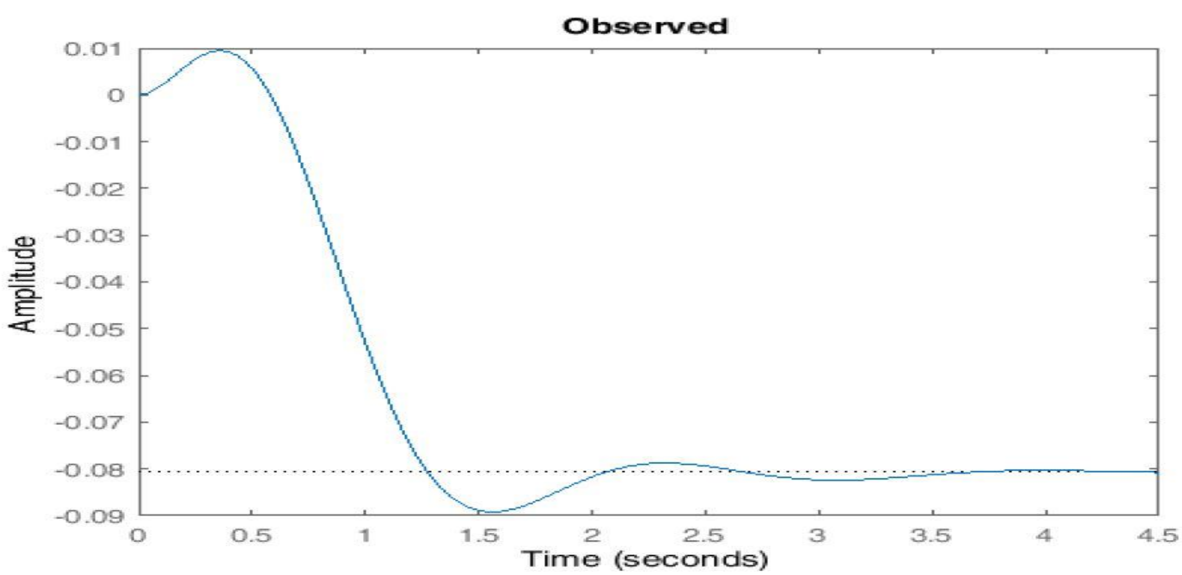


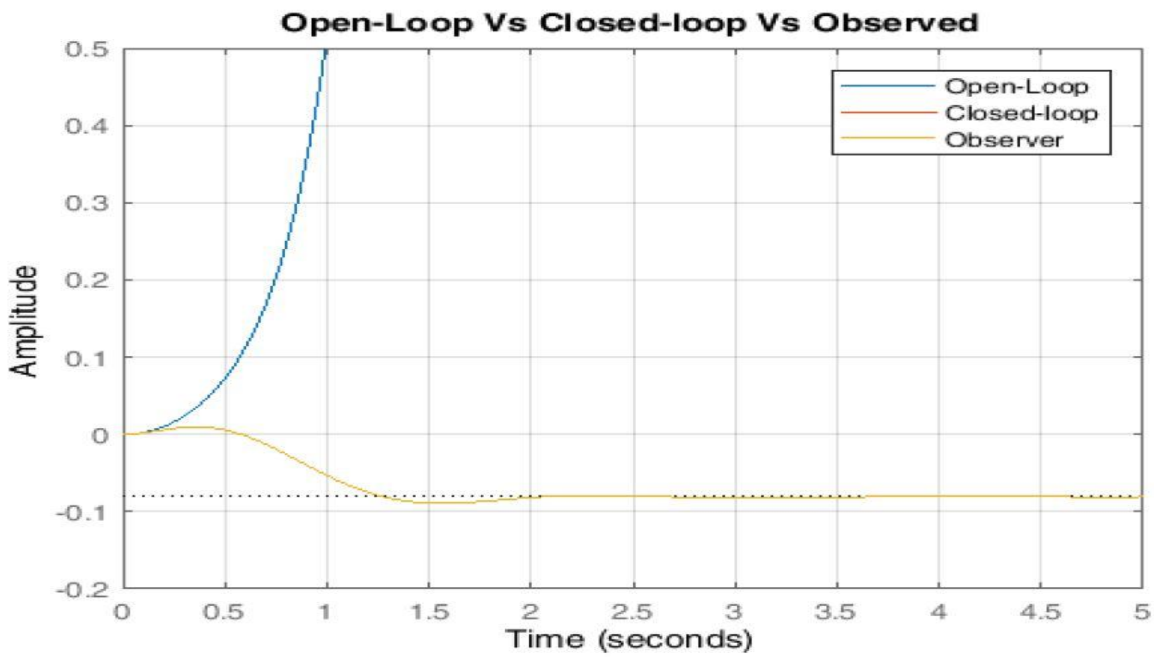
9. Design of linear Observers for state feedback:

#Matlab code for all three cases:

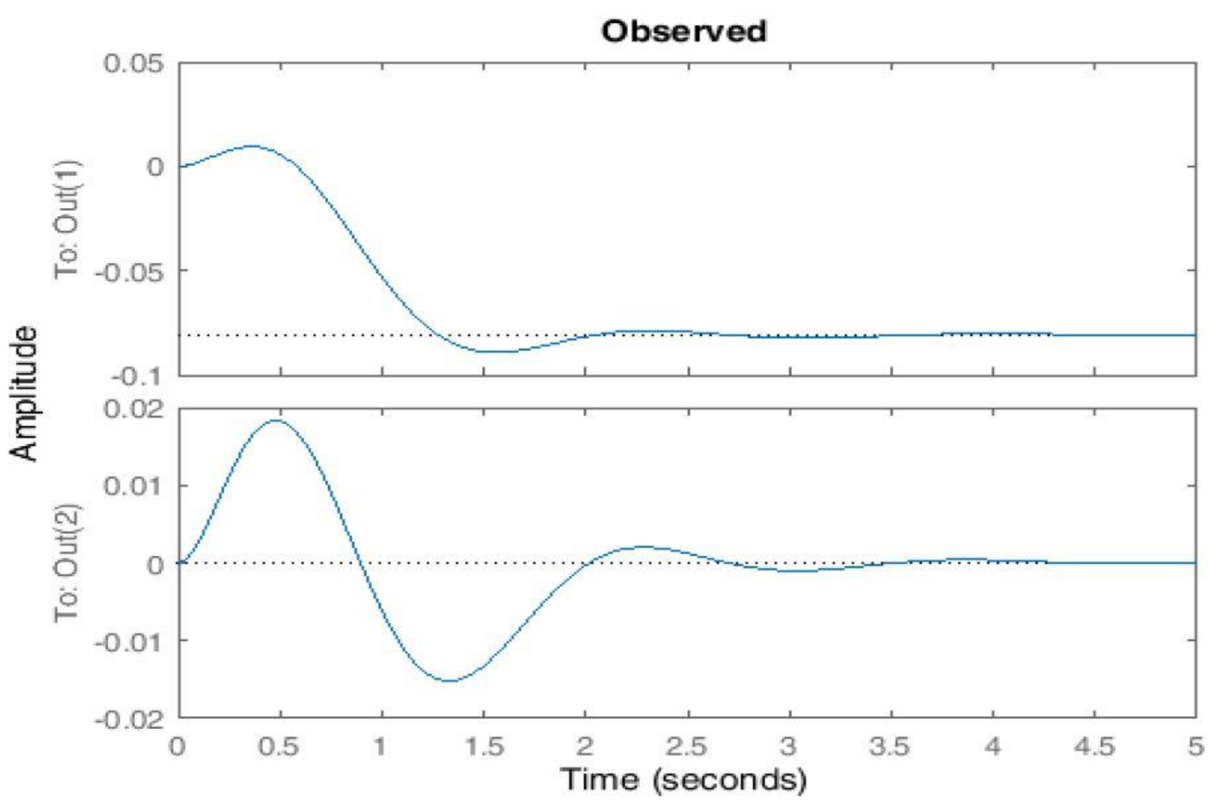
```
ObsEig = 10*DesEig
L = place(A',C', ObsEig)';
Ahat = A-L*C;
eig(Ahat);
Ar = [(A-B*K) B*K;zeros(size(A)) (A-L*C)];
Br = [B;zeros(size(B))];
Cr = [C zeros(size(C))];
Dr = D;
StateSpacer = ss(Ar,Br,Cr,Dr);
Transferr = tf(StateSpacer);
figure;
step(Transferr);
figure;
step(Transfer,TransferC,Transferr);
legend('Open-Loop','Closed-loop','Observer');
grid;
axis([0 5 -0.2 0.5]);
```

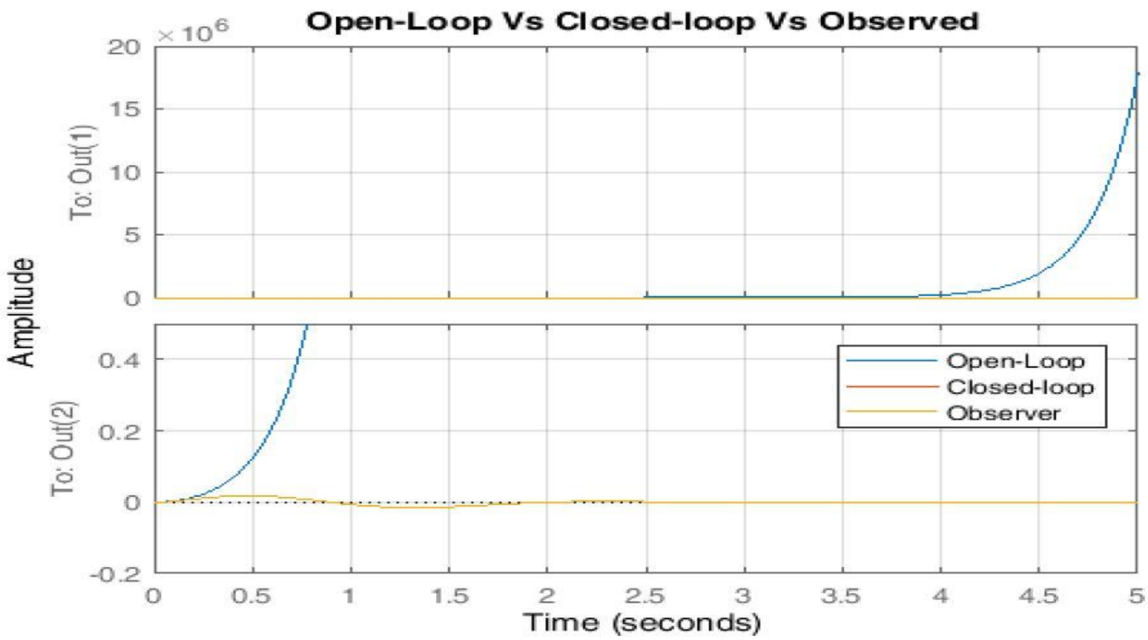
- **For case 1:**



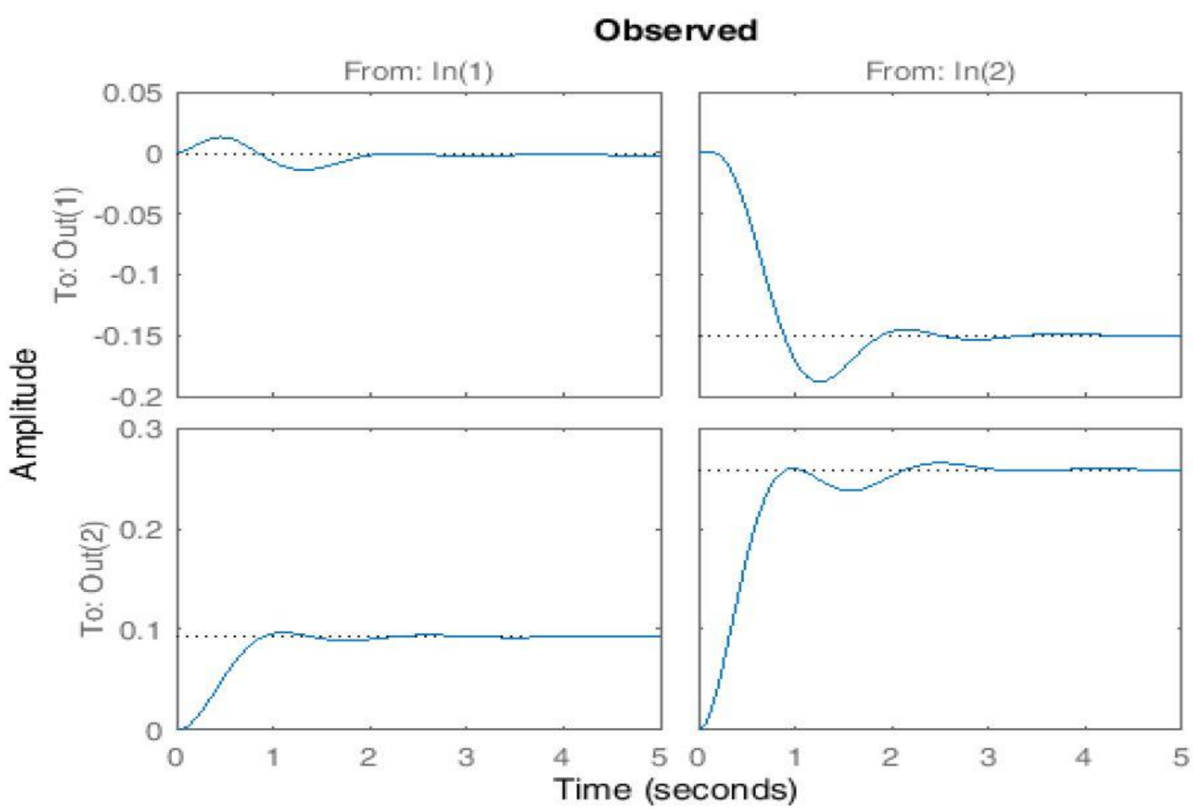

Open-Loop Vs Closed-loop Vs Observed
Open-Loop
Closed-loop
Observer
Amplitude
Time (seconds)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

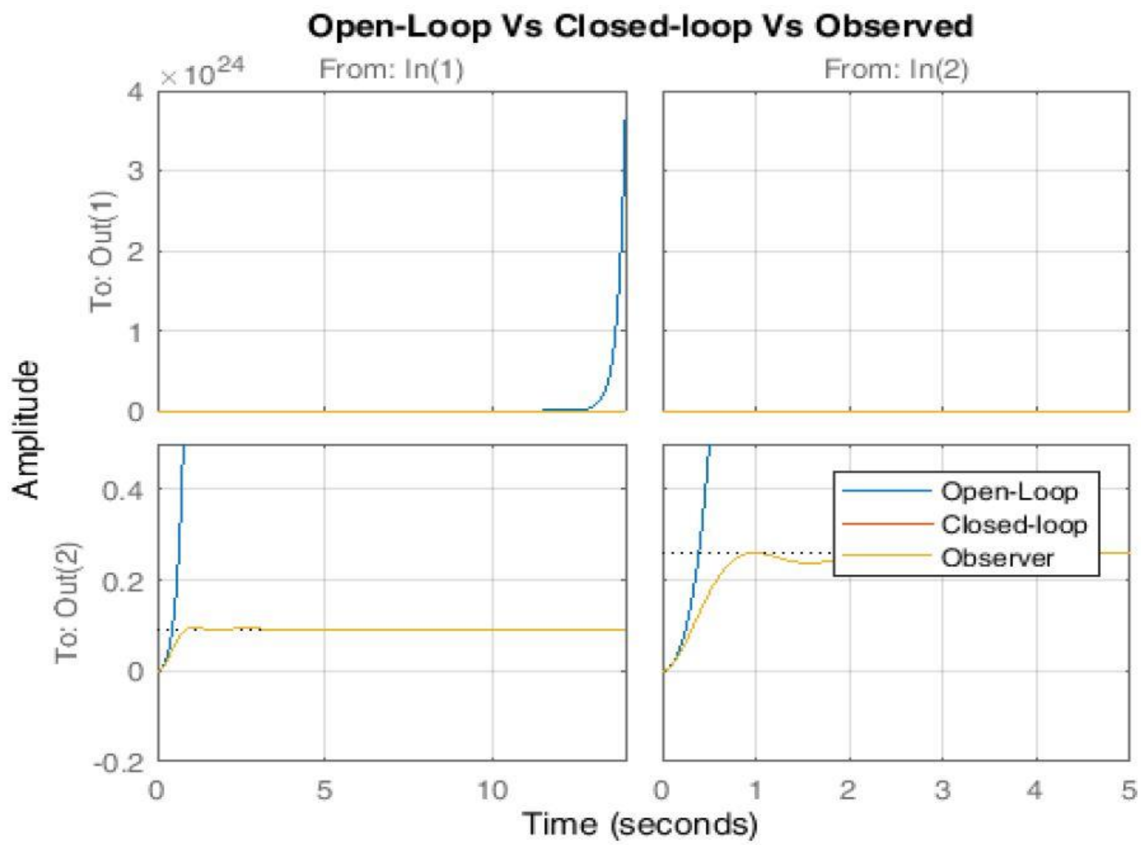
- **For case 2:**


Observed
To: Out(1)
To: Out(2)
Amplitude
Time (seconds)
0.05
0
-0.05
-0.1
0.02
0.01
0
-0.01
-0.02
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Open-Loop Vs Closed-loop Vs Observed
× 10^6
20
15
10
5
0
To: Out(1)
Amplitude
0.4
0.2
0
-0.2
To: Out(2)
Open-Loop
Closed-loop
Observer
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time (seconds)

- **For case 3:**


Observed
From: In(1)
From: In(2)
0.05
0
-0.05
-0.1
-0.15
-0.2
To: Out(1)
Amplitude
0.3
0.2
0.1
0
To: Out(2)
0
1
2
3
4
5
0
1
2
3
4
5
Time (seconds)



Closed-loop controls advantages:

- To reduce errors by automatically adjusting the systems input.
- To improve stability of an unstable system.
- To increase or reduce the systems sensitivity.
- To enhance robustness against external disturbances to the process.
- To produce a reliable and repeatable performance.

10. Conclusion:

This report has discussed the development of a control system right from assessing the controllability, observability and stability using different methods and the system was also manipulated by manual eigenvalue placement via state feedback. To implement this state feedback control law, we need all the state to be fed back as input. But every state in the real world is not measurable. So, we designed observers for an observable state equation to estimate the unmeasurable states.

All the parameters were selected arbitrarily, and we do not know what the "best" values are. For example, what is the best pole configuration for a given specification and available resources? All these "optimal" values can be found by using optimal control.