LAB REPORT

Discrete Control Systems

Submitted By

Yogachand Pasupuleti-41588
Chandrasekhar Moravaneni -41626

Under the Guidance of:
Prof. Dr.-Ing. Karsten Peter

Date of submission
13/09/2024

MASTER OF SCIENCE IN
EMBEDDED SYSTEM DESIGN

HOCHSCHULE BREMERHAVEN

Declaration

I hereby declare that my written submission fully represents the concept in my own words. Furthermore, I affirm that I have adhered to all principles of academic honesty and integrity. I certify that no part of my work has been fabricated, falsified, or misrepresented concerning any concept, data, fact, or source. I acknowledge that any violation of these principles will result in disciplinary measures by the Institute and may also result in legal action if sources are not properly cited or permissions are not obtained when necessary.

Mr. Yogachand Pasupuleti
Mr. Chandrasekhar Moravaneni

Index

Introduction

Discrete control systems, where signals are processed at distinct time intervals. Unlike continuous systems that operate over a continuum of time, discrete systems are analysed and controlled at specific, separate points in time. Deals with how systems can be controlled when signals are sampled and actions are taken at discrete time intervals. Discrete vibrational principles come into play, which typically involve optimizing or controlling the system behaviour over a sequence of discrete time steps. To understand and develop the skills for sophisticated control, an example of Seesaw system is studied in depth in this report. The study of this Lab is divided in separate tasks for better understanding.

Overview:

- A seesaw system consists of a beam and a car. On the Seesaw system, a car is placed at a distance "x" from the centre as shown in Fig.1. The mass of the car is Mf. The torque "τ" is applied at the other end of the Seesaw system. Keeping in mind these variables we build a mathematical model for the given system.

  The Seesaw system to be modelled is represented as below:

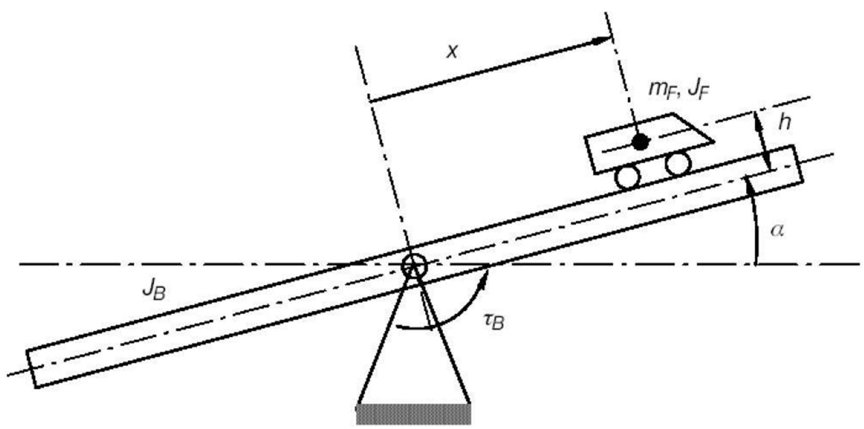


Figure 1:- Seesaw System

Where,

τB: Drive torque of the beam tilt motor

xF: Position of vehicle on the beam

α: Tilt angle of the beam

JF, JB: Momentum of inertia of the vehicle and the beam

mF: Mass of the car

h: Height of the vehicle above the beam middle

Task-1

Make a nonlinear model in MATLAB Simulink for the seesaw system?

Euler-Lagrange formalism and D'Alembert's principle are powerful approaches to derive the equations of motion for dynamic systems like the seesaw, which can exhibit nonlinear behaviour due to its geometry, forces, and constraints.

Lagrange function: First, it is necessary, the kinetic and potential energy of the entire system to describe. Only then can the Lagrangian can be formed.

Where,

$$\boxed{\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial qi}\right) = Qi}$$

Since Seesaw system is a two-dimensional system, it has two degrees of freedom. The polar coordinates 'x' and 'α' describes the dynamic motions of the system.
The coordinate column vector 'q' is given as:

$$\boxed{q = \begin{bmatrix} x_f \\ \alpha \end{bmatrix}}$$

The column vector of the non-conservative forces Q is given as:

$$\boxed{Q = \begin{bmatrix} 0 \\ \tau_B \end{bmatrix}}$$

Where the Drive torque of the beam tilt motor (τB) is applied to the System from the outside.
Lagrange function L is:

$$L = E_{kin-}\,E_{pot}$$
$$= \frac{1}{2} m_F\, x_F^{\ 2} + \frac{1}{2} \cdot\ m_F x_F^{\ 2} a^2 + \frac{1}{2} j_{const} \dot{a}^2$$
$$- m_F g x_F \sin a - m_F g h \cos a$$

Solving for $q_1 = x_F$ and $q_2 = \alpha$ we get:

$$\boxed{\ddot{x}_F = x_F \cdot \dot{\alpha}^2 - g \cdot \sin(\alpha)}$$

$$\ddot{\alpha} = \frac{1}{J_{\text{const}} + m_F \cdot {x_F}^2} \cdot \tau_B - \frac{m_F \cdot g \cdot x_F}{J_{\text{const}} + m_F \cdot {x_F}^2} \cdot \cos(\alpha) + \frac{m_F \cdot g \cdot h}{J_{\text{const}} + m_F \cdot {x_F}^2} \cdot \sin(\alpha) - \frac{2 \cdot m_F \cdot x_F}{J_{\text{const}} + m_F \cdot x_F \Delta^2} \cdot \dot{x}_F \cdot \dot{\alpha}$$

The dynamic seesaw system is described by the above non-linear differential equations.

Using the above-mentioned two equations the nonlinear Seesaw system can be modeled in MATLAB Simulink.

MATLAB CODE:

```
jb=0.5              %Jb=0.5kg*m^2
mf=0.1              %mf=0.1kg
h=0.1               %h=10cm
g=9.81              %g=9.81m*s^-2
Jconst=mf*h^2+jb
```

SIMULINK MODEL:

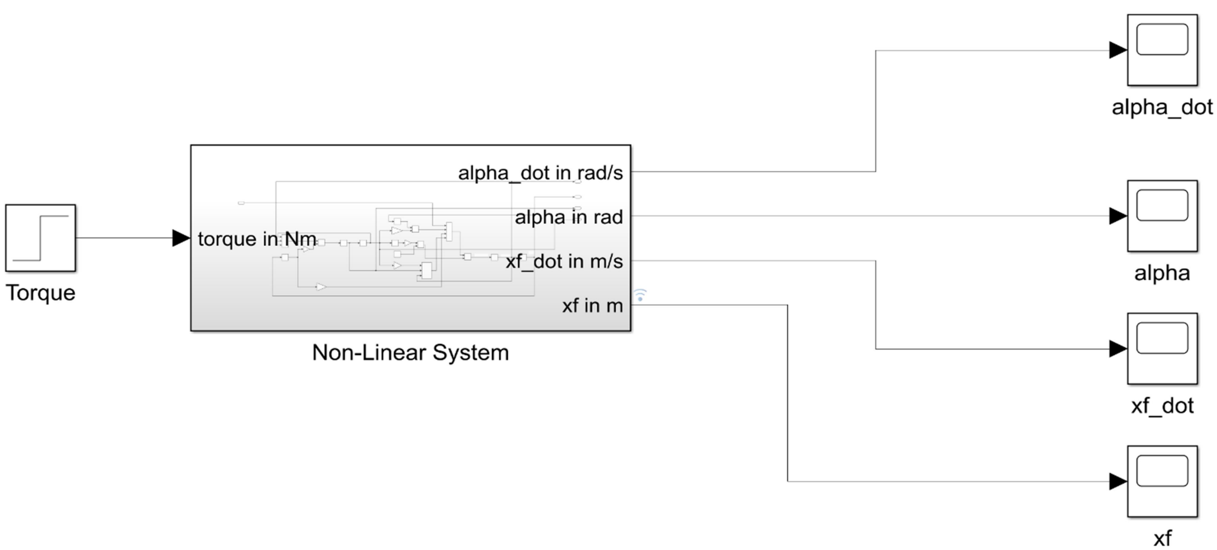


Figure 2: Simulink model for nonlinear Seesaw system

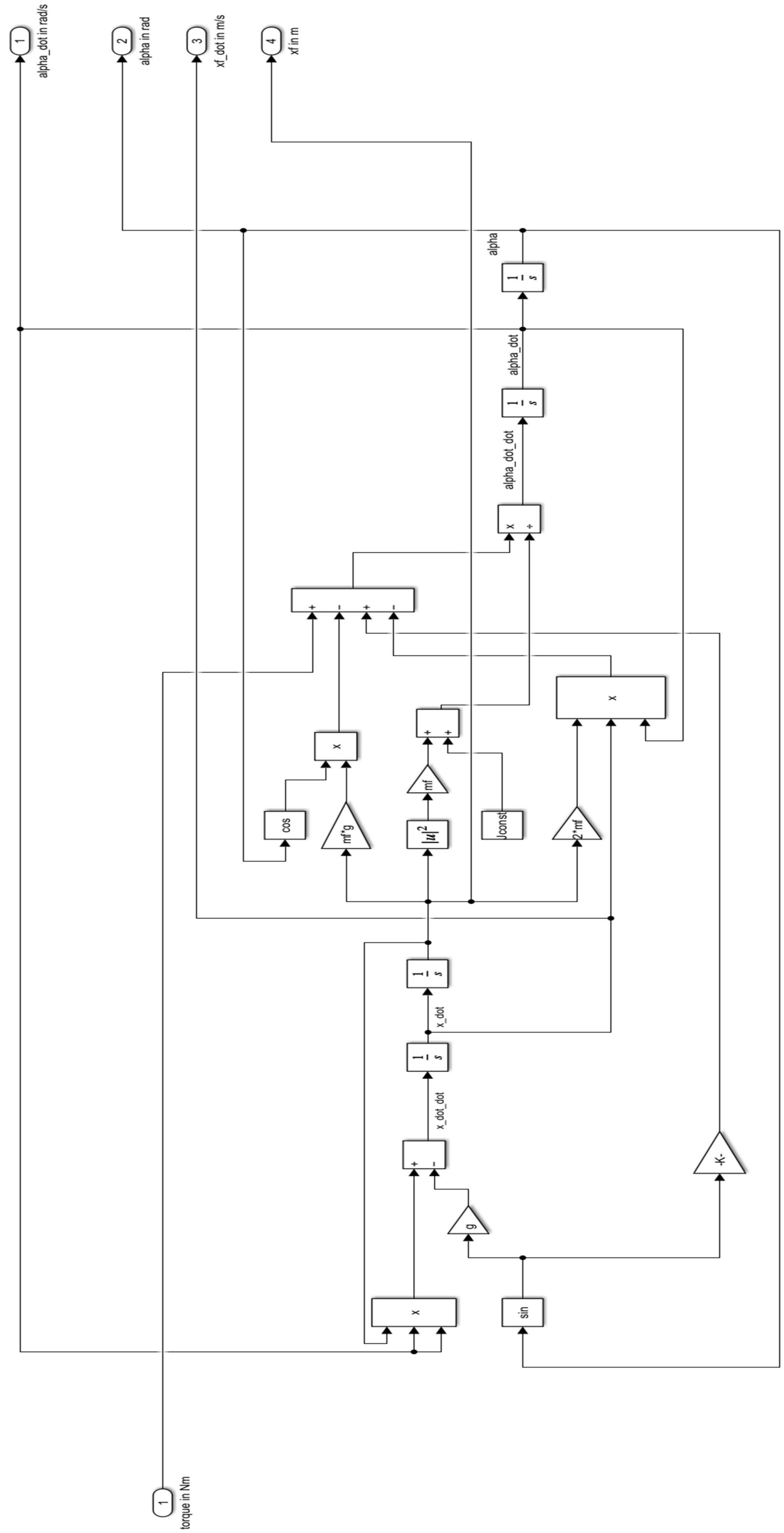


Figure 3: Nonlinear subsystem

Task-2

Linearize the seesaw system and make a linear state-space model for the working point zero in MATLAB Simulink?

Linearization:-

Linearization is a technique used to evaluate the local stability of an equilibrium point in a system of nonlinear differential equations or discrete dynamical systems.

To linearize the Seesaw system around working point zero, the following formula has to be used:

$$f(\dot{\alpha}, \alpha, \dot{x}F, xF, \tau B) \approx f(\dot{\alpha}0, \alpha 0, \dot{x}F0, xF0, \tau B0) + \frac{\partial f(\dot{\alpha}, \alpha, \dot{x}_F, x_F, \tau_B)}{\partial \alpha}\bigg|_{\dot{\alpha}_0, \alpha_0, \dot{x}_{F0}, x_{F0}, \tau_{B0}} (\alpha - \alpha_0) + \cdots$$

$$\dots + \frac{\partial f(\dot{\alpha}, \alpha, \dot{x}_F, x_F, \tau_B)}{\partial \tau B}\bigg|_{\dot{\alpha}_0, \alpha_0, \dot{x}_{F0}, x_{F0}, \tau_{B0}} (\tau_B - \tau_{B0})$$

Solving for both the equations of the Seesaw system's nonlinear movement, we get:

$$\boxed{\ddot{x}_F = -g \cdot \alpha}$$

$$\boxed{\ddot{\alpha} = \frac{1}{J_{\text{Const}}} \cdot \tau_B - \frac{m_F g}{J_{\text{Const}}} \cdot x_F + \frac{m_F g h}{J_{\text{Const}}} \cdot \alpha}$$

The above equation completely describes the linearized form of the Seesaw system. These two equations can represent the state space form of the Seesaw system.

State-Space Representation:-

A linearized, dynamic system corresponds to an ordinary linear first-order differential equations system. It is symbolized the input variable by u and the output by y. The inner size of the system or internal system states are determined by the time varying states of x, where the time-dependence is usually not written. There may be multiple input variables,

For a LTI system, the state space representation is given by:

$$\boxed{\begin{aligned} \underline{\dot{x}} &= \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{u} \\ \underline{y} &= \underline{C} \cdot \underline{x} + \underline{D} \cdot \underline{u} \end{aligned}}$$

The seesaw system consists of two differential equations of second-order.

By converting this two-differential equation into four first-order differential equations, the system can be represented in matrix form as follows:

$$\begin{bmatrix} \ddot{\alpha}(t) \\ \dot{x}(t) \\ \ddot{x}_F(t) \\ \dot{x}_F(t) \end{bmatrix} = \begin{bmatrix} 0 & \frac{m_F g h}{J_{const}} & 0 & -\frac{m_F g}{J_{const}} \\ 1 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\alpha}(t) \\ \alpha(t) \\ \dot{x}_F(t) \\ x_F(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{const}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \underline{u}$$

$$\underline{y} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}}_{\underline{C}} \cdot \underline{x} + 0 \cdot$$

$$\text{Where, } \underline{u} = \tau_B(t)$$

The Matrix described above is used to develop a linear state space model in MATLAB Simulink.

MATLAB CODE:

```
jb=0.5                          %Jb=0.5kg*m^2
mf=0.1                          %mf=0.1kg
h=0.1                           %h=10cm
g=9.81                          %g=9.81m*s^-2
Jconst = mf*h^2+jb


% state space system
% state vector x = [alpha_dot; alpha; xf_dot; xf]

A = [0, mf*g*h/Jconst, 0, -mf*g/Jconst; % System Matrix A
    1, 0, 0, 0;
    0,-g, 0, 0;
    0, 0, 1, 0;]
B = [1/Jconst; % Input Matrix B
    0;
    0;
    0;]
C = eye(4); % Output Matrix C
D = [0;] % Feedthrough Matrix D
```

SIMULINK MODEL:

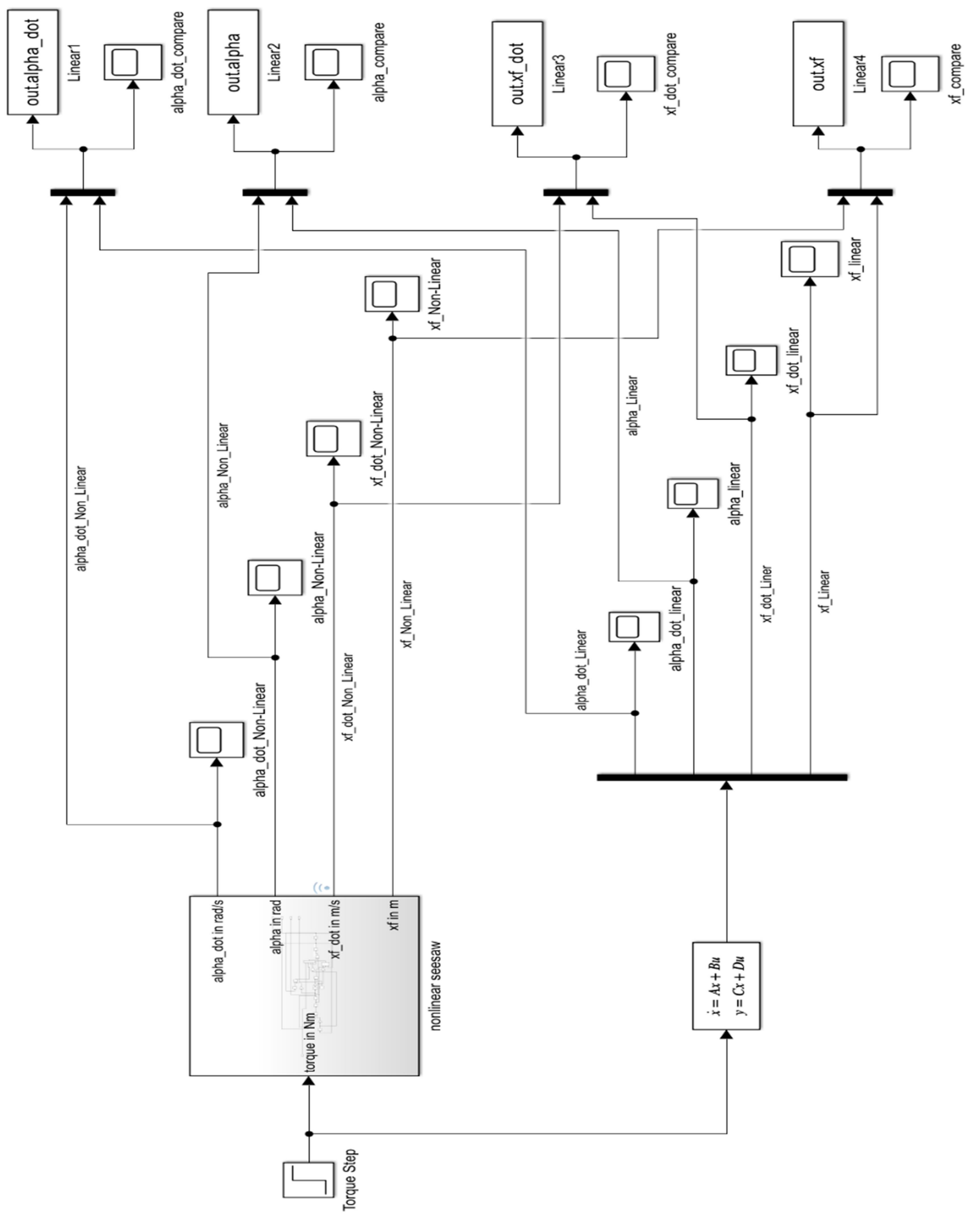


Figure 4: Simulink model for state space linear and non-linear system for seesaw

Task-3

Compare the linear model and the nonlinear model. When does the linear model become imprecise? How far can the working point be away from the initial working point (in state space), upon which the linear system was derived from?

To compare the linear and nonlinear models, the output of both models would be observed with scope.

Comparison of Displacement:-

The graphs for displacement VS time for linear and non-linear models that the downward curve starts a little later for the linear model but steeper than the non-linear model.

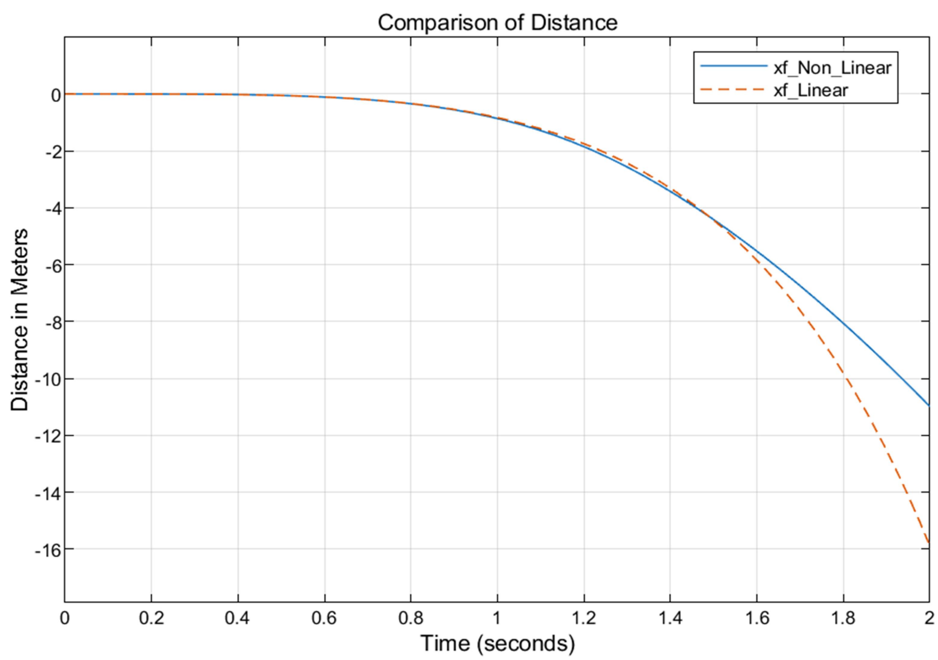


Figure 5: Comparison of distance

Comparison of Velocity-

It is evident from the graphs for velocity VS time for linear and nonlinear models that the downward curve starts a little later for the linear model but is steeper than the nonlinear model.

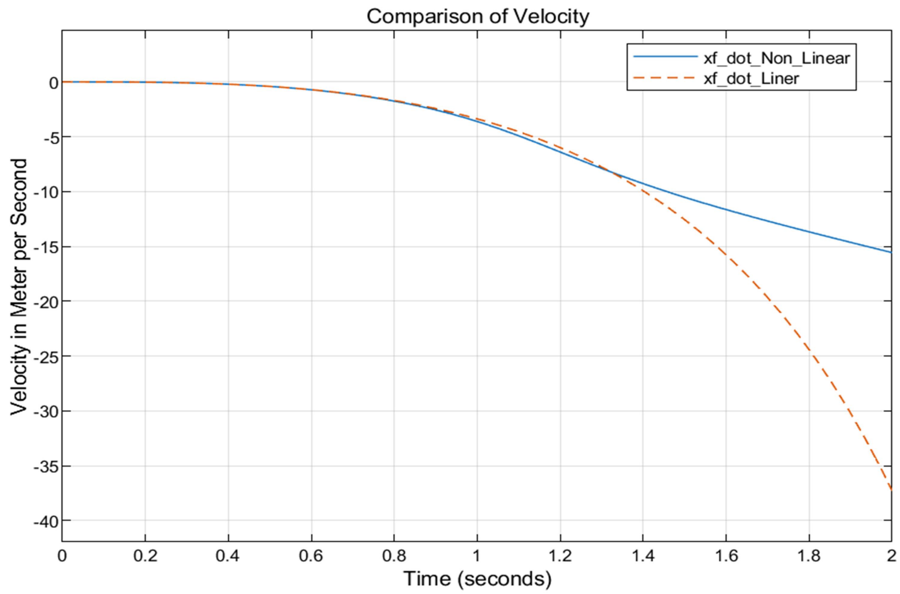


Figure 6: Comparison of Velocity

Comparison of Angle of Rotation-

It is evident from the graphs for angular rotation VS time for linear and nonlinear models that for linear models, the curve heads forward i.e. its keep on increasing as time increases, whereas for nonlinear models it heads forward but doesn’t get beyond 90 degrees i.e. curve increases for some time period and later becomes saturate.

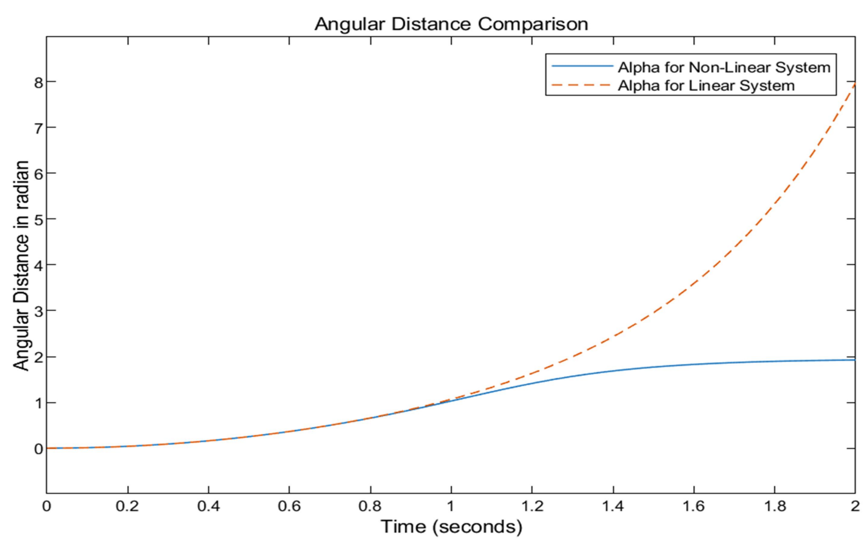


Figure 7: Comparison of Angular Distance

Comparison of Angular Velocity-

It is evident from the graphs for angular velocity VS time for linear and nonlinear models that for linear models, the curve heads forward i.e. its keep on increasing as time increases, whereas for nonlinear models it heads forward but doesn't get beyond 90 degrees and deviate to down fall, i.e. curve increases for some time period and later fall out as time increase and attain saturate.

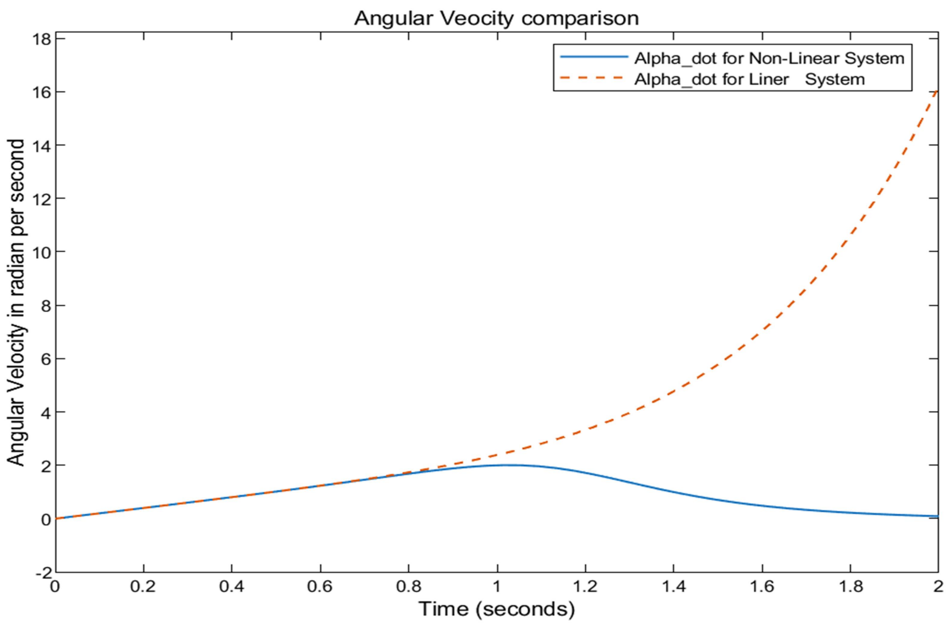


Figure 8: Comparison of Angular Velocity

Task-4

Make a state-feedback control based on the linear seesaw system. Place all poles at -1. Calculate state feedback vector k and preamplifier p. Verify the control on the linear seesaw system. Test the linear state feedback vector k and the preamplifier p on the nonlinear seesaw system. Test it for reference value steps and disturbance torque steps. Try different pole placements? How fast can you make your control?

In feedback control system theory, state feedback is a method for setting a system's closed-loop poles at specific, defined points in the s-plane. Because the pole locations are directly related to the system's eigenvalues, which define the system's response characteristics, this pole arrangement is desirable. The system has to be considered controllable in order to use this technique. It is evident in the previous task that neither linear nor nonlinear models are stable. Due to the open loop poles not being on the left side of the s-plane,

To make the system stable, the poles should be in the left half of the plane. To place all the poles at -1, Ackerman's formula would be used. It involves the calculation of state feedback vector k and preamplifier p.

Ackerman's formula to calculate state feedback vector kT is given by:

$$k^T = q^T \cdot (\alpha_0 \cdot I + \alpha_1 A + \cdots + \alpha_{n-1} \cdot A^{n^{-1}} + A^n)$$

The equation facilitating the determination of the preamplifier necessary for achieving steady-state precision in state feedback control is as follows:

$$p = \frac{1}{C^T . (B * K\_t - A)^{-1} * B}$$

Stability Check:

A preliminary system-stability check can be performed for an open loop system by observing the position of the poles of the system without feedback. A stable system has all poles in the open left half of the S-plane. In case of Linear Systems, the poles of the system are also the eigen-values of the System Matrix $[A]$.

The open loop poles of the system under consideration are calculated using the MATLAB command ‘eig()’. Applying command to the System Matrix of the Linear Seesaw System gives the following values:

eig(A) = -2.1170 + 0.0000i
2.1170 + 0.0000i
0.0000 + 2.0703i
0.0000 - 2.0703i

If a system has multiple poles on the imaginary axis or poles with a positive real part, then it is an unstable system. The open-loop poles calculated above suggest that the system is unstable.

MATLAB CODE:

```
jb=0.5          %Jb=0.5kg*m^2
mf=0.1                  %mf=0.1kg
h=0.1           %h=10cm
g=9.81                  %g=9.81m*s^-2
Jconst=mf*h^2+jb
% state space system
% state vector x = [alpha_dot; alpha; xf_dot; xf]

A = [0, mf*g*h/Jconst, 0, -mf*g/Jconst;
    1, 0, 0, 0;
    0,-g, 0, 0;
    0, 0, 1, 0;]
B = [1/Jconst;
    0;
    0;
    0;]
C = eye(4);
D = [0;]

%building controllability matrix
s = [ B, A*B, A*A*B, A*A*A*B;]

%Check for full rank
rank(s)

%Inverse the controllability matrix
s_inv = inv(s)

% Las row of inverse controllability matrix
q_t = s_inv(4,:)

%Calculating KT

alpha = poly([-1,-1,-1,-1]) % describing polynomial

%assigning values according to the equation
alpha_0 = alpha(5)
alpha_1 = alpha(4)
alpha_2 = alpha(3)
alpha_3 = alpha(2)
alpha_4 = alpha(1)
```

```
K_t = q_t*((alpha_0*eye(4)) + (alpha_1*A) + (alpha_2*A*A) + (alpha_3*A*A*A) +   (alpha_4*A*A*A*A))
Ct = C(4,:);
%preamplifier
p = 1/(Ct*inv((B*K_t)-A)*B);
```

SIMULINK MODEL:

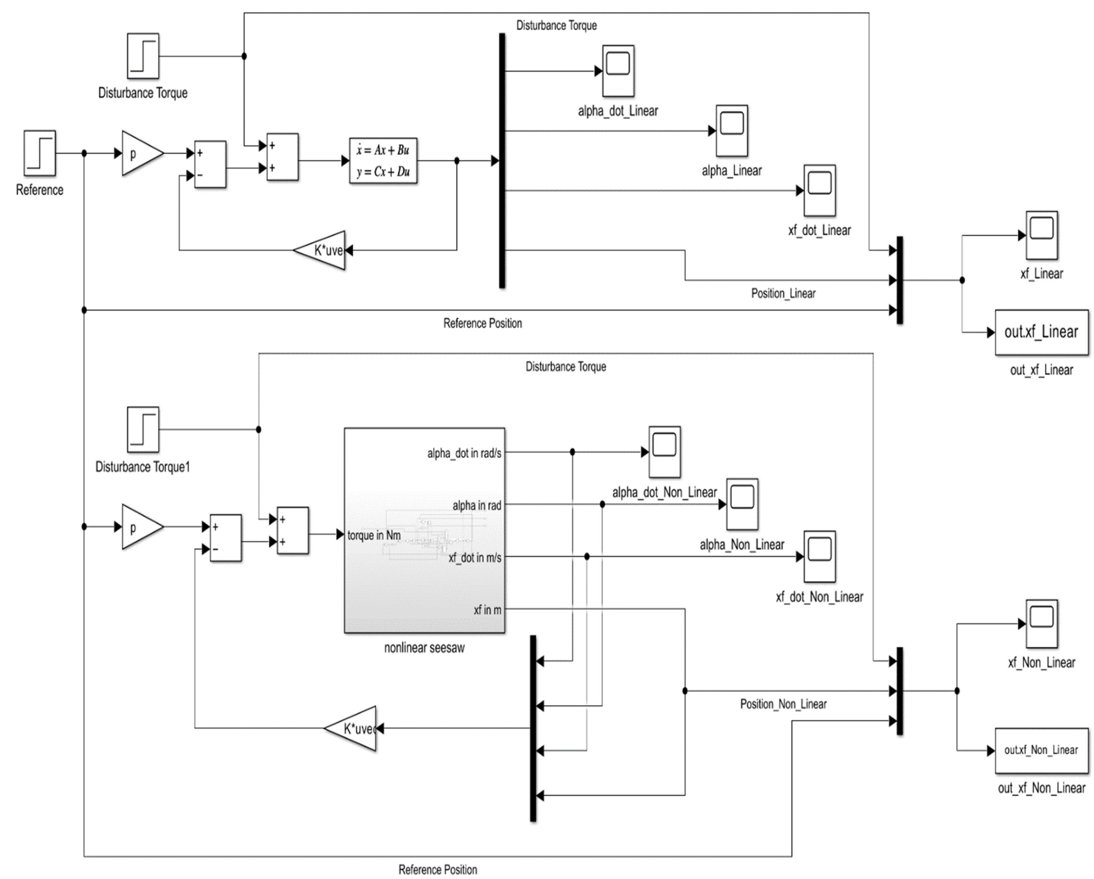


Figure 9: Simulink model for state feedback vector and preamplifier

Comparison of displacement (without disturbance torque):-

When no disturbance torque is applied, the system stability is achieved and follows the reference torque i.e. it reaches steady state accuracy with the calculated state feedback vector and preamplifier gain. The system response is the same for both linear and nonlinear systems.

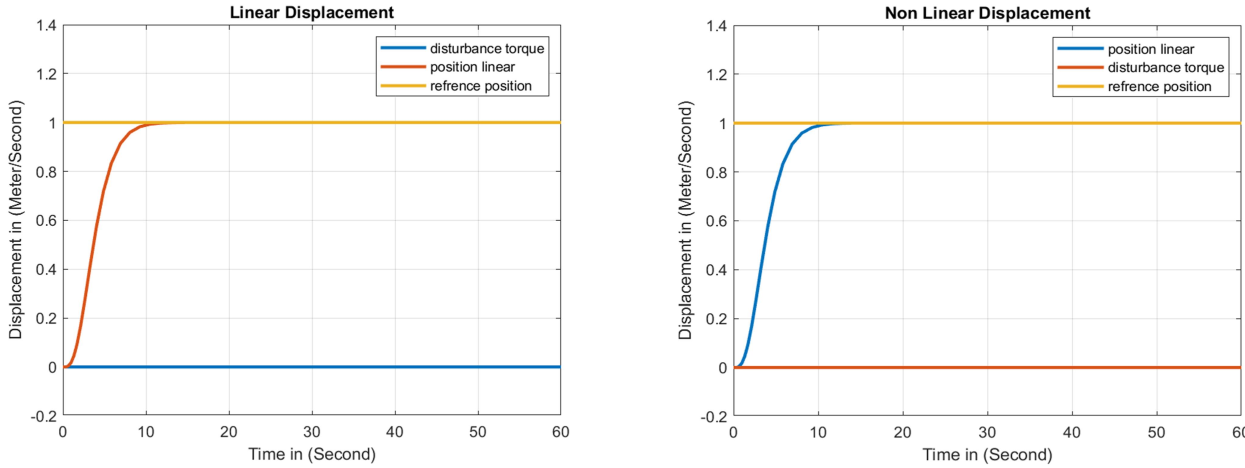


Figure 10: comparison of linear and nonlinear Displacement with Torque=0

Comparison of displacement (with disturbance torque = 0.1 at 15sec):-

When disturbance torque is applied (0.1 Nm at 15 sec), the controller is not able to compensate for the distortion torque. Even though the system is stable, it does not follow the reference torque for both linear and nonlinear systems.

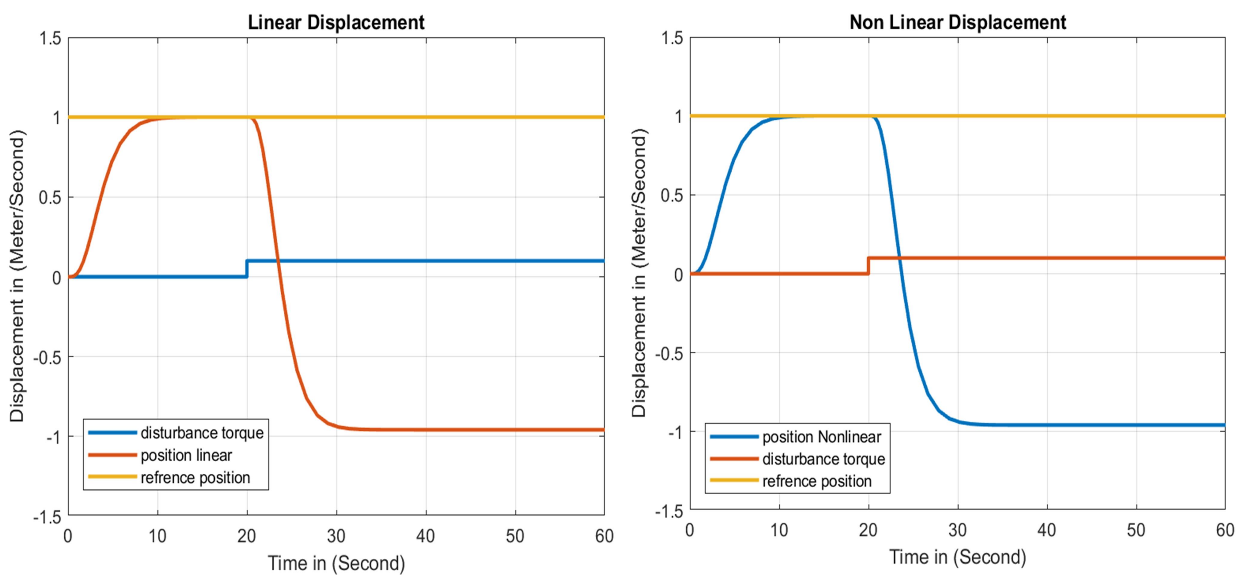
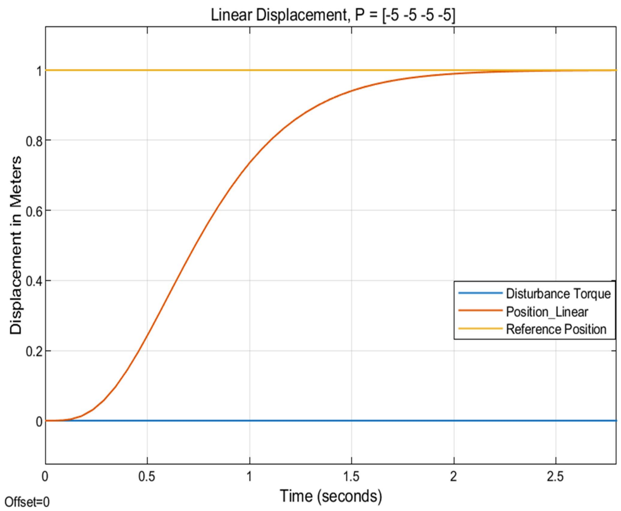
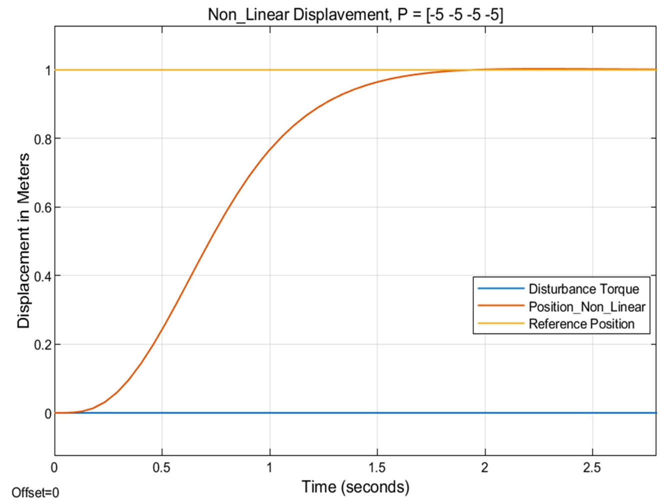
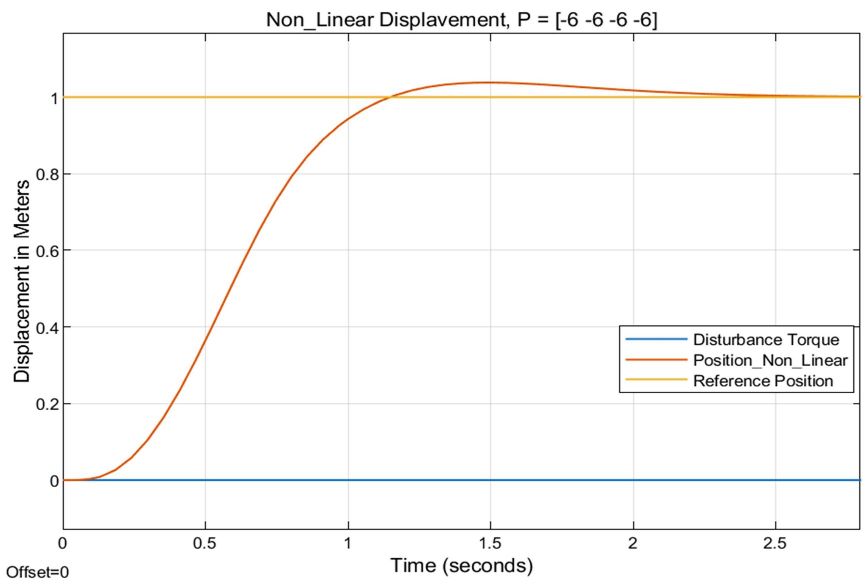


Figure 11: Comparison of linear and non-linear Displacement with Torque=0.1







Above figures depicts the response of linear and non-linear system for pole of different magnitude. As we can see, when pole value is changed from [-1 -1 -1 -1] to [-5 -5 -5 -5], system achieve the stability faster. When pole value is changed to [-6-6-6-6], Non-Linear system overshoot the reference torque before achieving the stability.

We can control the system response for faster time of 2.4s and 1.8s for linear and Non-linear System respectively for the pole magnitude of [-5-5-5-5]

Task-5

Make a PI-state-feedback control based on the linear seesaw system. Place all poles at -1. Calculate the state feedback vector k based on the augmented system (5th order). Verify the control on the linear seesaw system. Test the linear PI-state-feedback control for reference value steps and disturbance torque steps. Discuss different pole placements?

The system achieves a state of equilibrium and a precise response within a limited manipulation range when the reference torque is reduced and there is no disturbance torque at the input. This is achieved through the utilization of a state feedback controller and preamplifier gain.

Along with the proportional controller, a second integrator is included when using a PI controller. An extra state enters the system as a result of the integrator. As a result, the enhanced system's order is raised by 1.

The proportional component (p) is construed as a constituent of the state-feedback vector, given that it feeds back the system's state (represented as z). This leads to the following system description of the augmented system in state- space:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 \\ \hline -C^T & 0 \end{bmatrix}}_{Api} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{Bpi} + \begin{bmatrix} e & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}$$

$$y(t) = \underbrace{\begin{bmatrix} C^T & 0 \end{bmatrix}}_{Cpi} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}$$

$$\underline{k}^{*T} = \underline{k}^T - p * \underline{C}^T$$

MATLAB CODE:

```
jb=0.5              %Jb=0.5kg*m^2
mf=0.1                      %mf=0.1kg
h=0.1               %h=10cm
g=9.81                      %g=9.81m*s^-2
Jconst=mf*h^2+jb


% state space system
% state vector x =  [alpha_dot; alpha; xf_dot; xf]

A = [0, mf*g*h/Jconst, 0, -mf*g/Jconst;
    1, 0, 0, 0;
    0,-g, 0, 0;
    0, 0, 1, 0;]
B = [1/Jconst;
    0;
    0;
    0;]
C = eye(4);
```

```
D = [0;]

AT = [0, mf*g*h/Jconst, 0, -mf*g/Jconst, 0;
    1, 0, 0, 0, 0;
    0,-g, 0, 0, 0;
    0, 0, 1, 0, 0;
    0, 0, 0, -1, 0;]
BT = [1/Jconst;
    0;
    0;
    0;
    0;]
CT = [0,0,0,1,0;]
DT = [0;]


%building controlability matrix
s = [ BT, AT*BT, AT*AT*BT, AT*AT*AT*BT, AT*AT*AT*AT*BT;]

%Check for full rank
rank(s)

%Inverse the controlability matrix
s_inv = inv(s)

% Las row of inverse controlability matrix
q_t = s_inv(5,:)

%Calculating KT

alpha = poly([-1,-1,-1,-1,-1]) % describing polynomial

%assiging values according to the equation
alpha_0 = alpha(6)
alpha_1 = alpha(5)
alpha_2 = alpha(4)
alpha_3 = alpha(3)
alpha_4 = alpha(2)
alpha_5 = alpha(1)

K = q_t*((alpha_0*eye(5)) + (alpha_1*AT) + (alpha_2*AT*AT) + (alpha_3*AT*AT*AT) + (alpha_4*AT*AT*AT*AT) + (alpha_5*AT*AT*AT*AT*AT))

PI = -K (5);

K_T = K (1:4) - p*Ct;
```

SIMULINK MODEL:-

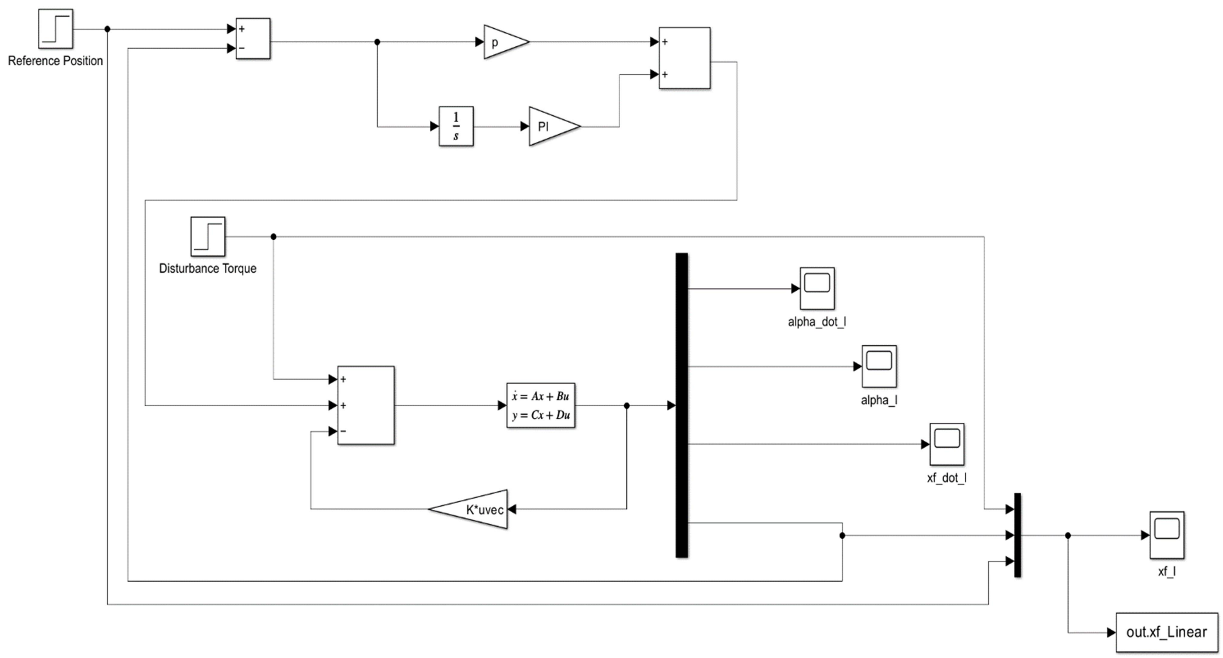


Figure 12: Simulink model of PI Controller for Linear System

Graphs:

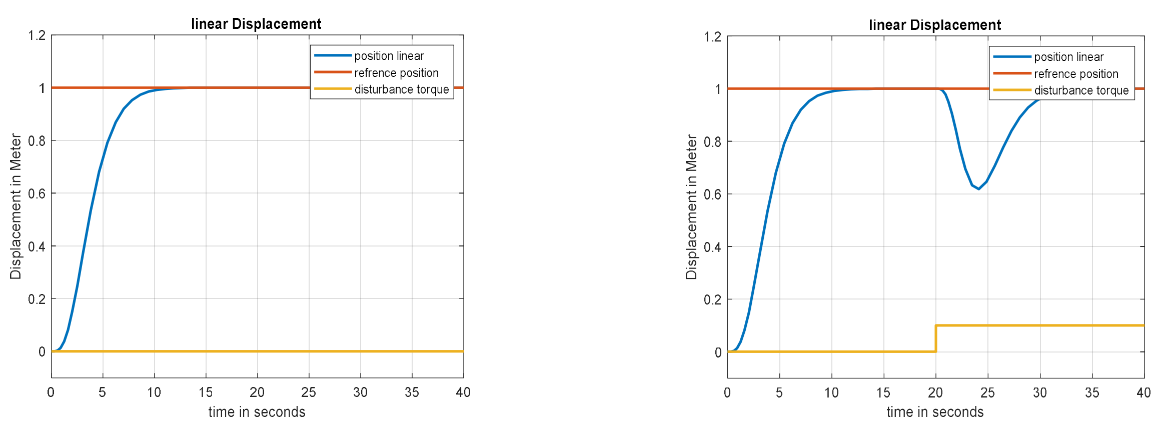


Figure 13: Linear Displacement with no-disturbance and low-disturbance

It is evident from the above graphs that, when there is no disturbance, the response of the system follows the reference torque and attains the stability at reference position. When disturbance of 0.1 is introduced at 20 sec, then the response of the system gets disturbed, and deviates downward, after sometime, system again follows the reference torque and attains stability

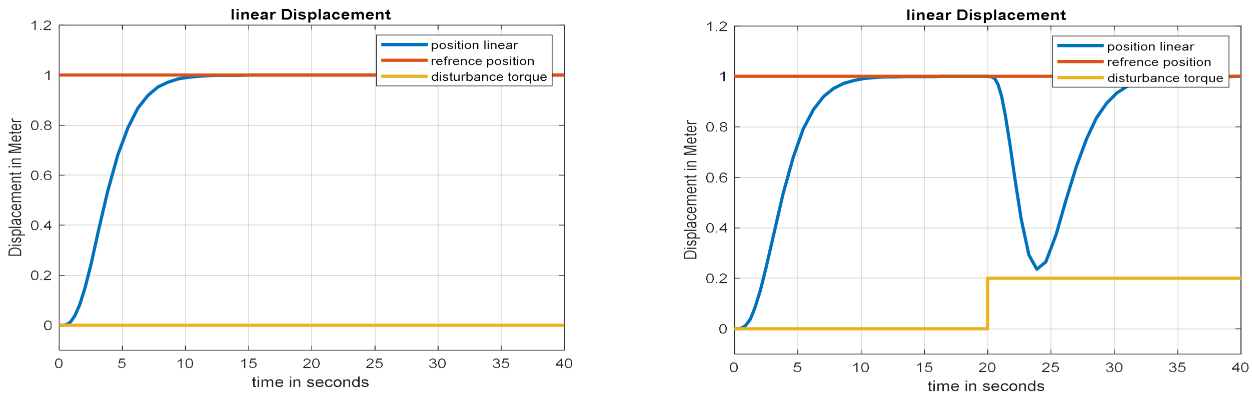
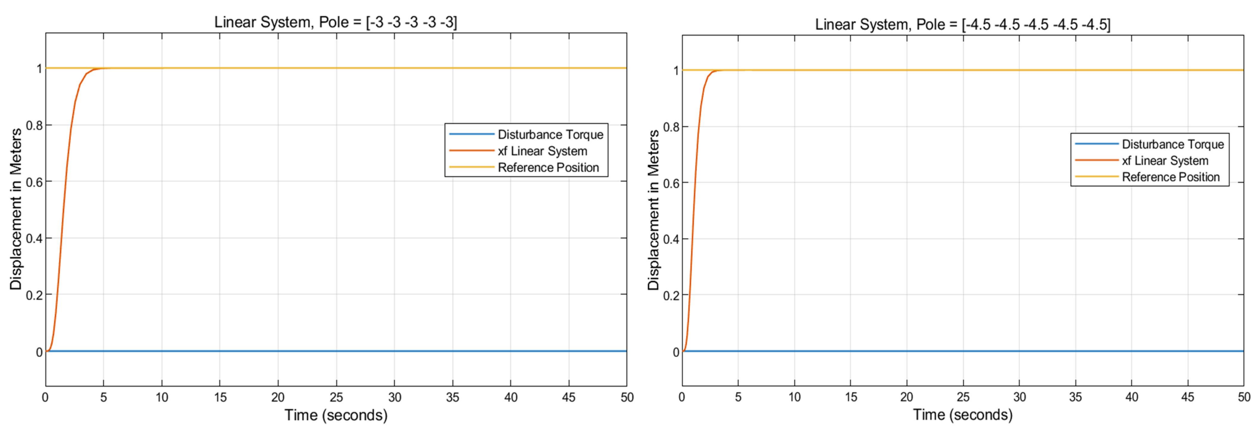


Figure 14: Linear Displacement with no-disturbance and high-disturbance

It is evident from the above graphs that, when the magnitude of reference torque is increased and there is no disturbance, the response of the system follows the reference torque and we get 'S' curve. When disturbance of 1 is introduced at 20 sec, then the response of the system also gets disturbed, and follows the reference torque again.



Above figures represents, the response of PI Controller for Linear System for pole of different magnitude. As we can see, when pole value is changed from [-1 -1 -1 -1 -1] to [-3 -3 -3 -3 -3], systems achieve the stability faster. When pole value is changed to [-4.5 -4.5 -4.5 -4.5], the system response achieves the stability even faster than the previous two case of pole value`s.

Task-6

Make the PI-part time discrete with a sample rate 800 Hz and test the linear PI-state-feedback on the nonlinear seesaw system. Change the sample time. Try different pole placements and different sample times? Discuss the results?

A discrete-time control system is one in which one or more variables change only at specific, discrete moments in time. The time interval between these moments is chosen to be sufficiently short so that the data between them can be approximated.

To discretize the PI component with a sampling rate of 800 Hz, the desired step response poles, which are initially located in the left half of the s-plane, need to be mapped within the unit circle in the z-plane to ensure stability in the discrete-time system. The system should be converted to a discrete-time state-space representation, using an appropriate sampling interval and the default Zero-Order Hold (ZOH) discretization method. Additionally, the integrator in the continuous-time system should be replaced by a unit delay in the discrete-time system.

MATLAB CODE:-

```
jb=0.5              %Jb=0.5kg*m^2
mf=0.1                       %mf=0.1kg
h=0.1               %h=10cm
g=9.81                       %g=9.81m*s^-2
Jconst=mf*h^2+jb
fs = 800                     % Frequancy In Hz
Ts = 1/fs                    % Sample Time in Seconds

% state space system
% state vector x = [Alpha_d_dot; Alpha_d; xf_dot; xf]

A = [0, mf*g*h/Jconst, 0, -mf*g/Jconst;
    1, 0, 0, 0;
    0, -g, 0, 0;
    0, 0, 1, 0;]
B = [1/Jconst;
    0;
    0;
    0;]
C = eye(4);
D = [0;]

AT = [0, mf*g*h/Jconst, 0, -mf*g/Jconst, 0;
    1, 0, 0, 0, 0;
    0, -g, 0, 0, 0;
    0, 0, 1, 0, 0;
    0, 0, 0, -1, 0;]
```

```
BT = [1/Jconst;
    0;
    0;
    0;
    0;]
CT = [0,0,0,1,0;]
DT = [0;]

sys = ss(AT, BT, CT, 0)
sys_d = c2d(sys, Ts)      %Converting the Controlability Matrix in Discrete State Space

[Ad, Bd, Cd, Dd] = ssdata(sys_d)   %Loading the Discrete Value of the A,B,C,D Matrix

Sd = [Bd, Ad*Bd, (Ad^2)*Bd, (Ad^3)*Bd, (Ad^4)*Bd] % Calculating Controlability Matrix in Discrete State Space

%Check for full rank
rank(Sd)

% Inverse the Discrete Controlability Matrix
Sd_inv = inv(Sd)

% Loading the last row of the Discrete Controlability Matrix
qt_d = Sd_inv(5,:)

sys_po = tf(1, poly([-1,-1,-1,-1,-1]))

%Getting the Poly values at Sample Time
sys_Alpha_d = c2d(sys_po, Ts)

[Beta_d,Alpha_d] = tfdata(sys_Alpha_d,'v')

%assiging values according to the equation
Alpha_d_0 = Alpha_d(6)
Alpha_d_1 = Alpha_d(5)
Alpha_d_2 = Alpha_d(4)
Alpha_d_3 = Alpha_d(3)
Alpha_d_4 = Alpha_d(2)
Alpha_d_5 = Alpha_d(1)

%Calculating KT
Kd = qt_d*((Alpha_d_0*eye(5)) + (Alpha_d_1*Ad) + (Alpha_d_2*Ad*Ad) + (Alpha_d_3*Ad*Ad*Ad) + (Alpha_d_4*Ad*Ad*Ad*Ad) + (Alpha_d_5*Ad*Ad*Ad*Ad*Ad))

Pd = -Kd(5)*Ts  % Getting the Integral gain

Kd_T = Kd(1:4) - p*Ct
```

Simulink Model:-

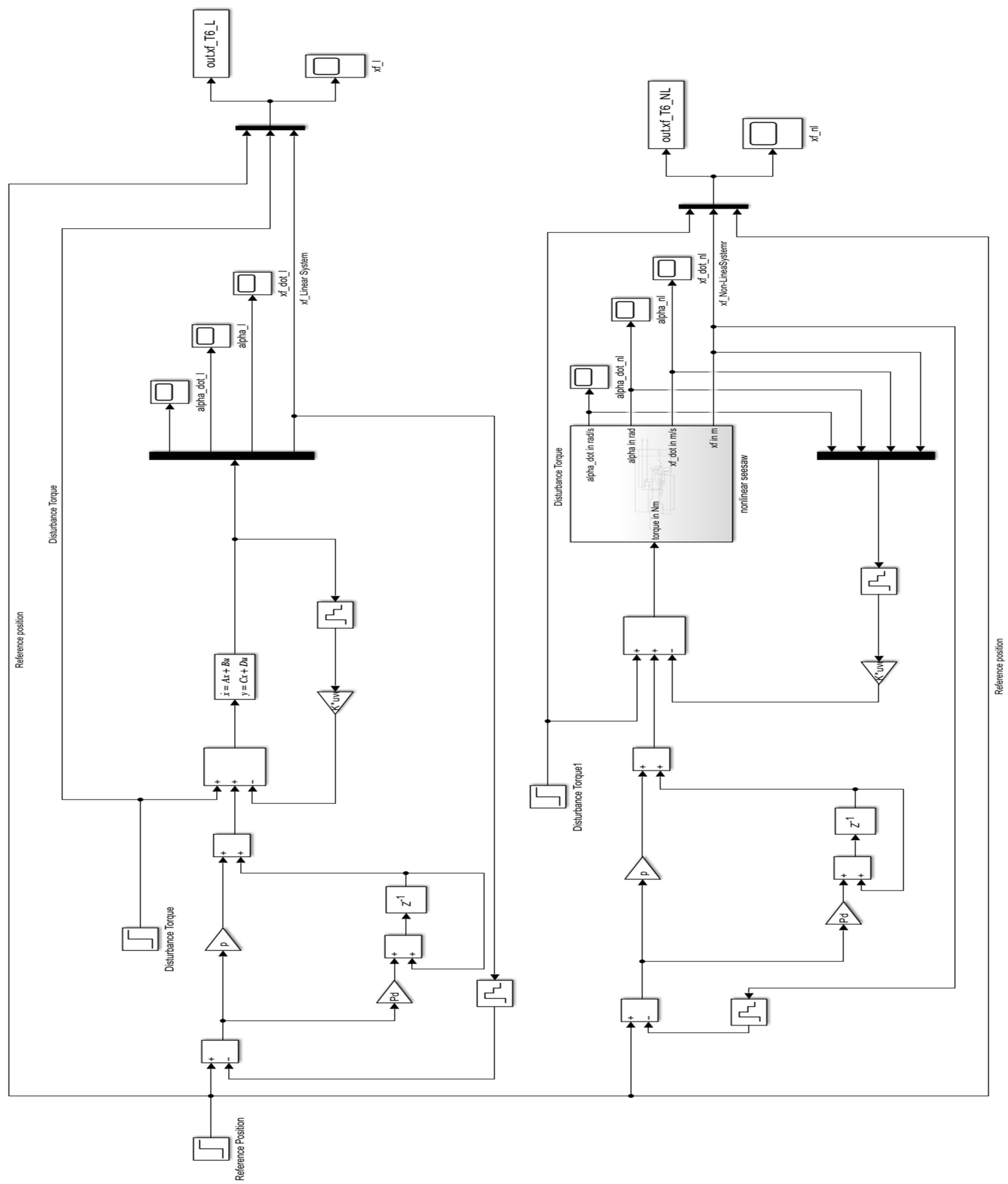
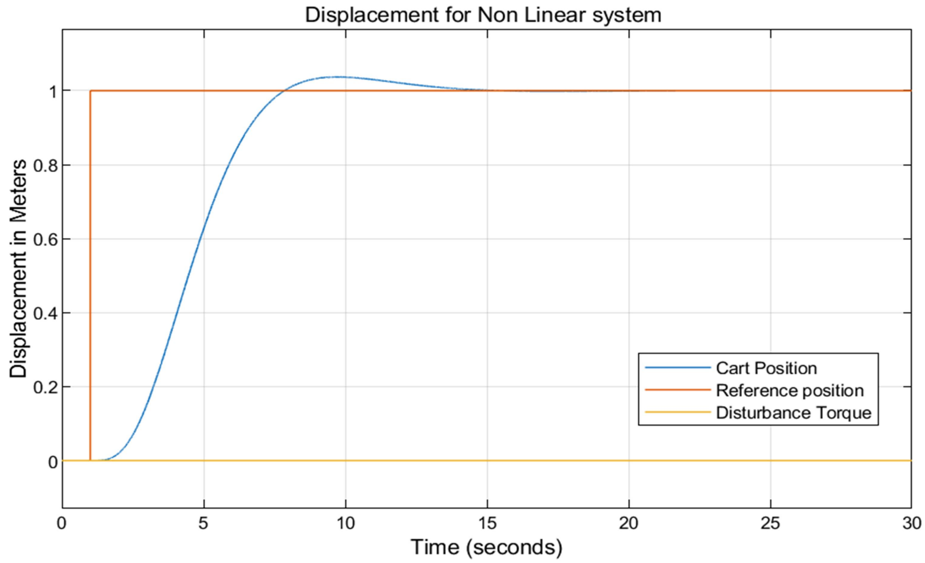


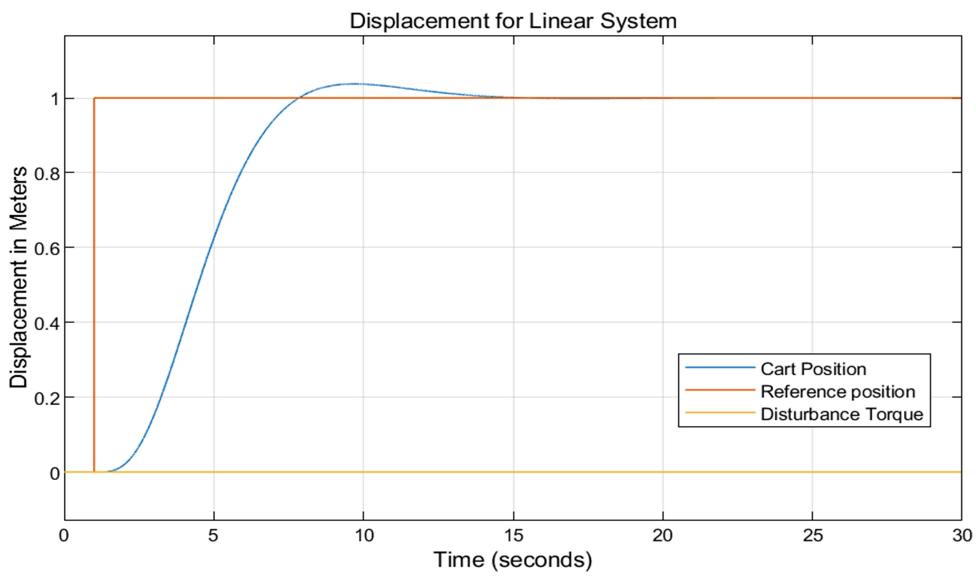
Figure 15: Simulink model of discrete PI Controller

Graphs:

1. Comparison of Displacement (Without Disturbance Torque)



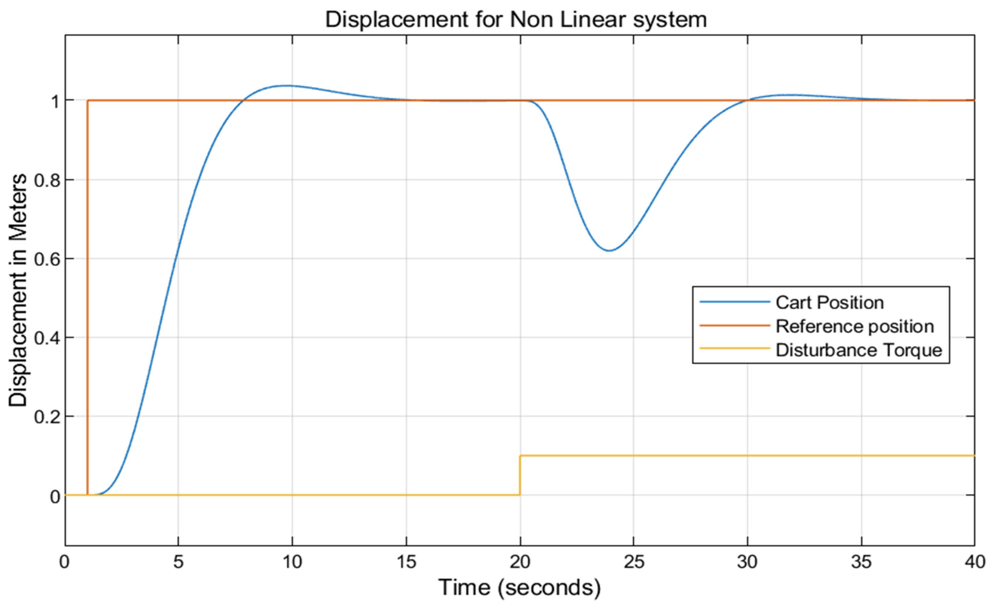
Distance for Non-Linear System



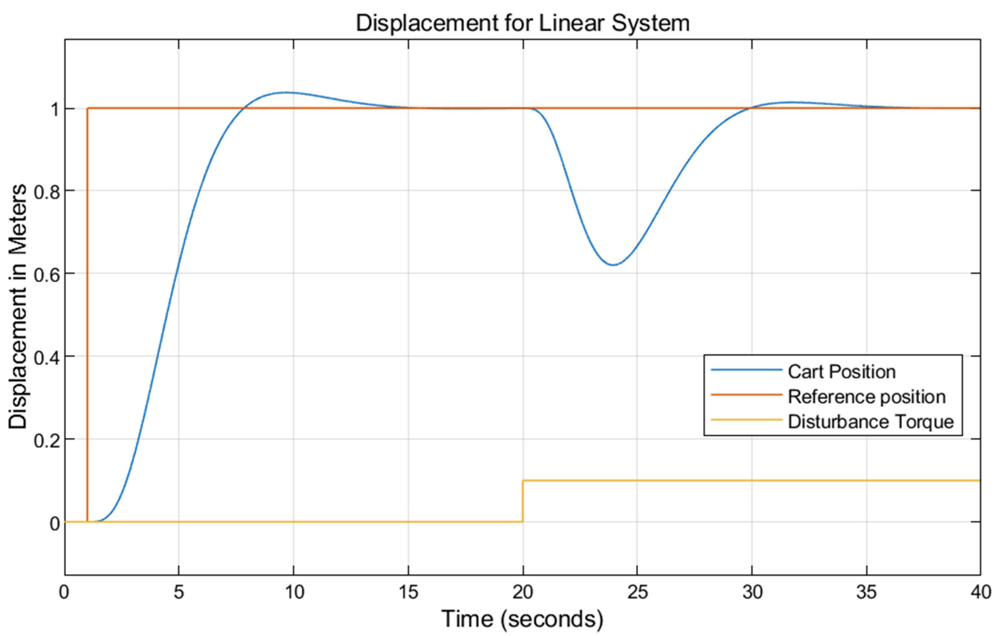
Distance for Linear System

The graph illustrates the response of the systems that manages the displacement of a cart over time without disturbance torque. The response rapidly moves towards the reference position and overshooting slightly before stabilizing around the reference position.

2. Comparison of Displacement (With Disturbance Torque)



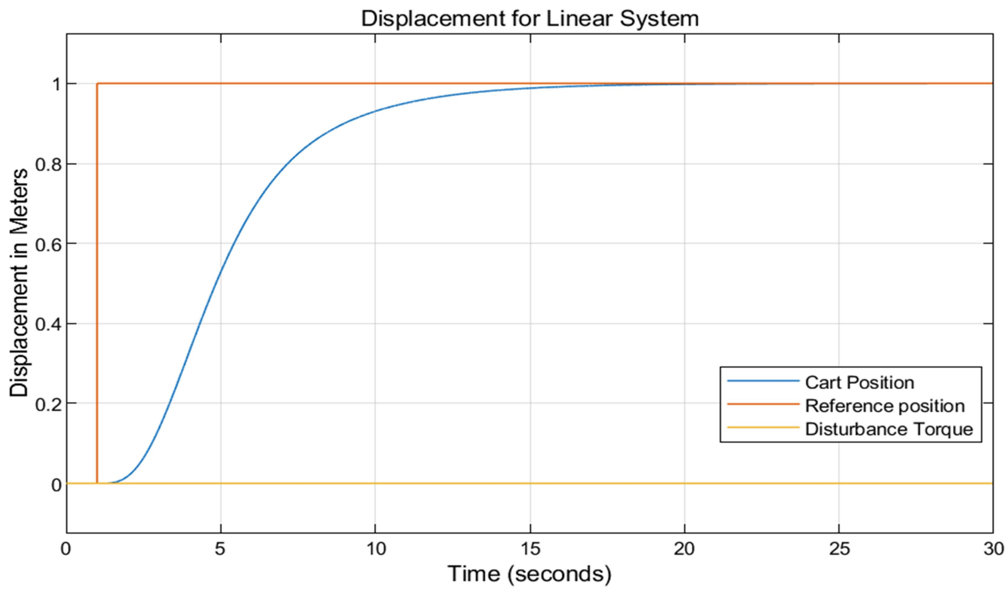
Displacement for Non-Linear System



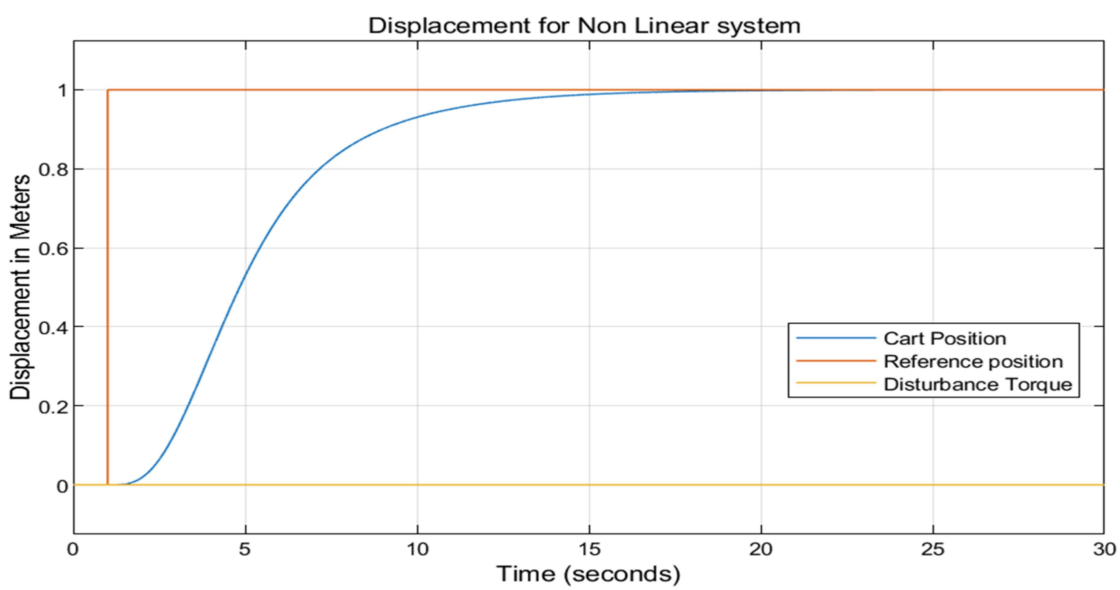
Displacement for Linear System

The graph shows the displacement responses of the systems where the cart position, reference position, and disturbance torque are plotted over time. The response initially accelerates towards the reference position, overshooting slightly before stabilizing around reference position. When a disturbance torque is introduced, causing the response to deviate significantly below the reference position, showing a dip (Steady State Error) in displacement. The systems eventually recover, bringing response back to the reference position and thus attain the stability.

3. Comparison of Displacement (Without Disturbance Torque and sample rate of 1000Hz)

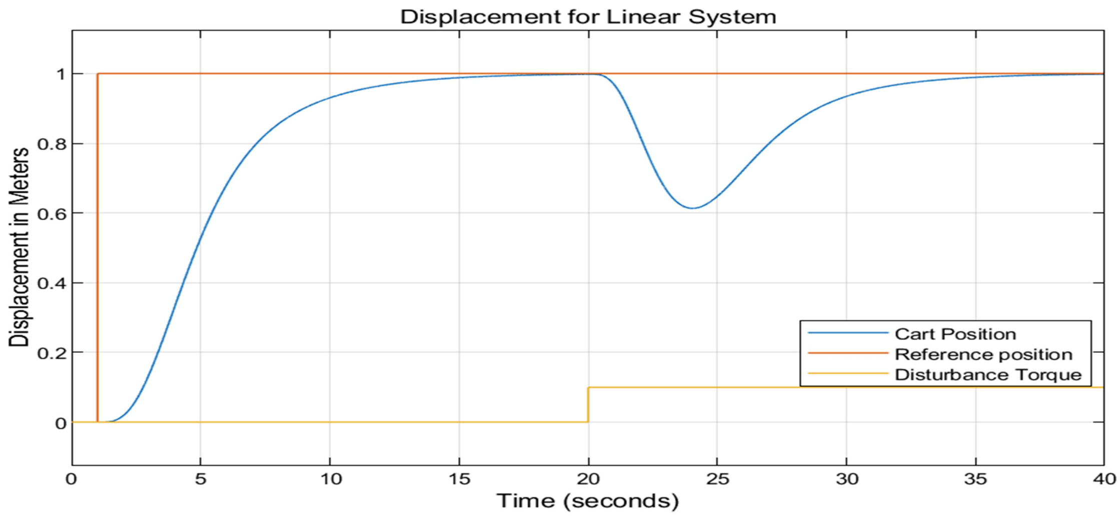


Displacement for Linear System

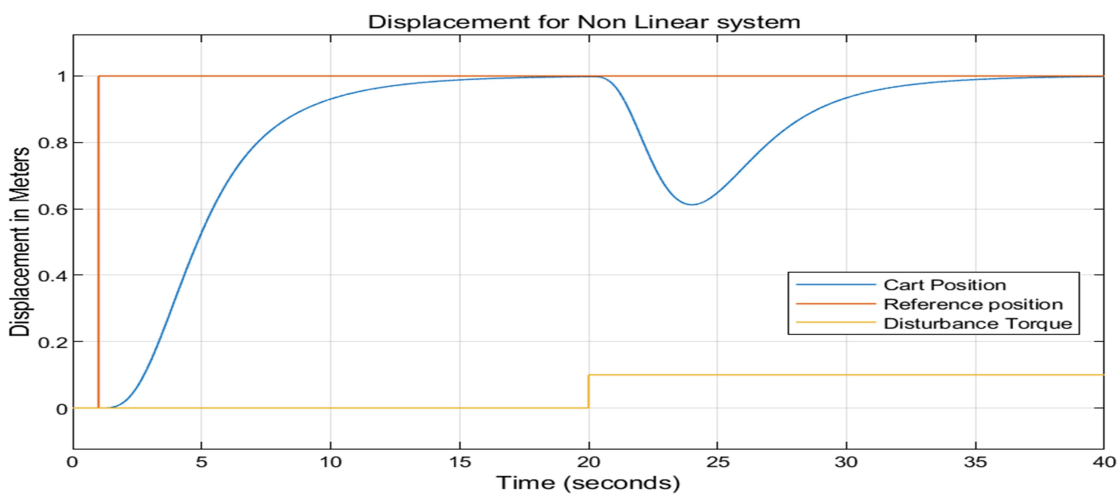


Displacement for Non-Linear System

4. Comparison of Displacement (With Disturbance Torque and sample rate of 1000Hz)



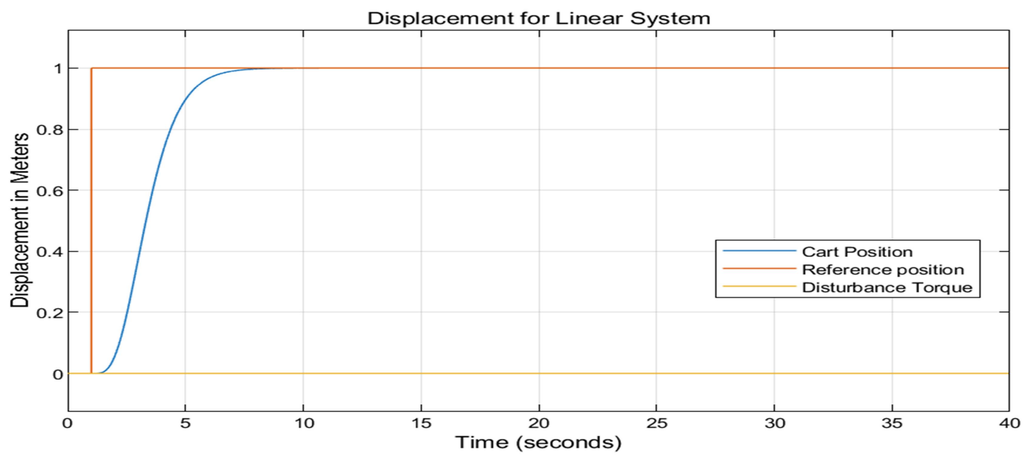
Displacement for Linear System



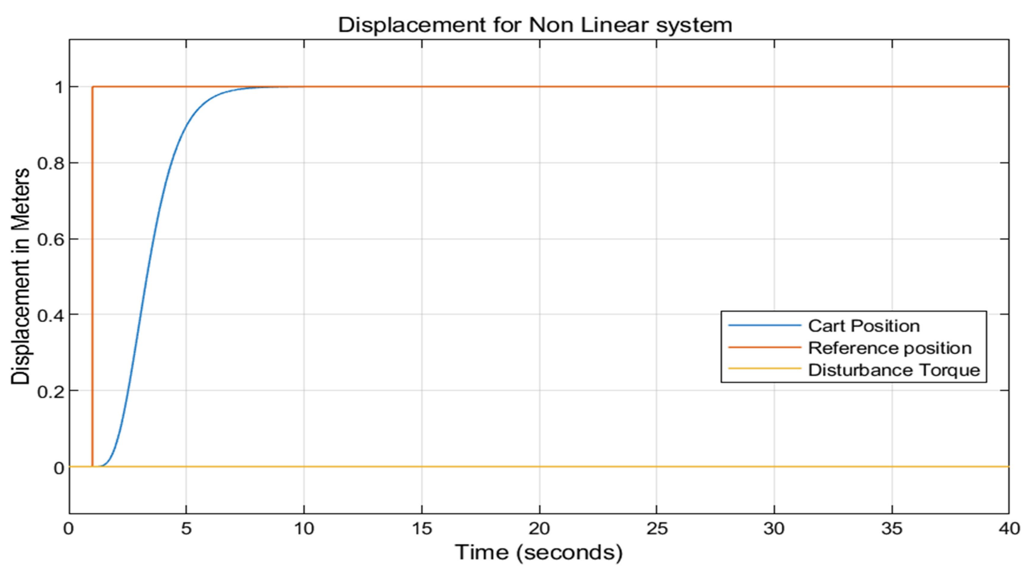
Displacement for Non-Linear System

The graph shows the displacement response of the systems for a Sample rate of 1000Hz and with disturbance torque. When the sample time is changed from 0.08s to 0.001s, the response initially accelerates towards the reference position similar to the graph of the system with 0.08s sample time but without overshooting around reference position. When a disturbance torque is introduced, causing the response to deviate significantly below the reference position, showing a dip (Steady State Error) in displacement. The system eventually recovers, bringing response back to the reference position and thus attains the Stability, similar to the graphs of the system with sample time of 0.08s

5. Comparison of Displacement (Without Disturbance Torque and Pole of -2)



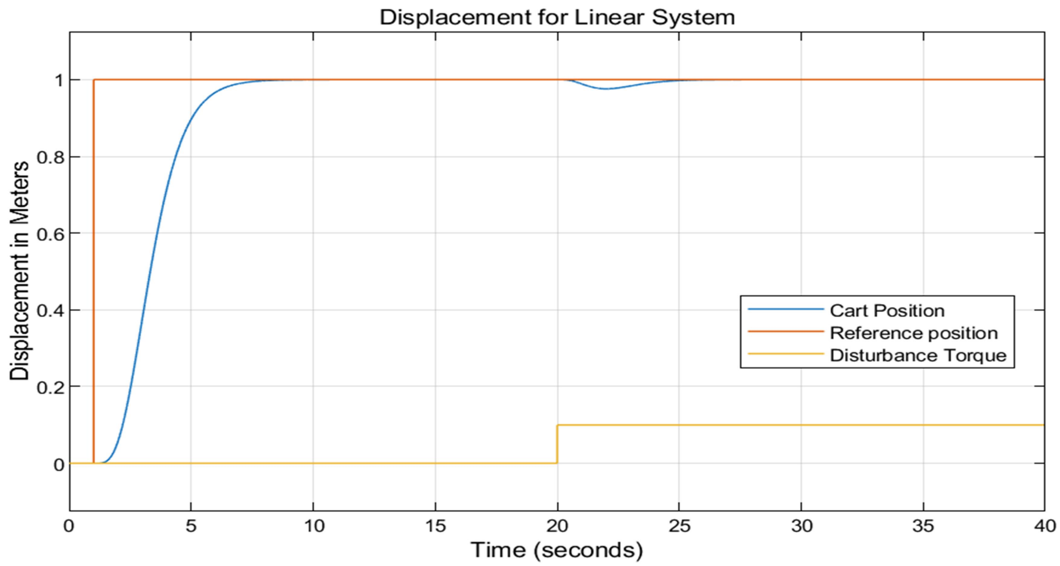
Displacement for Linear System



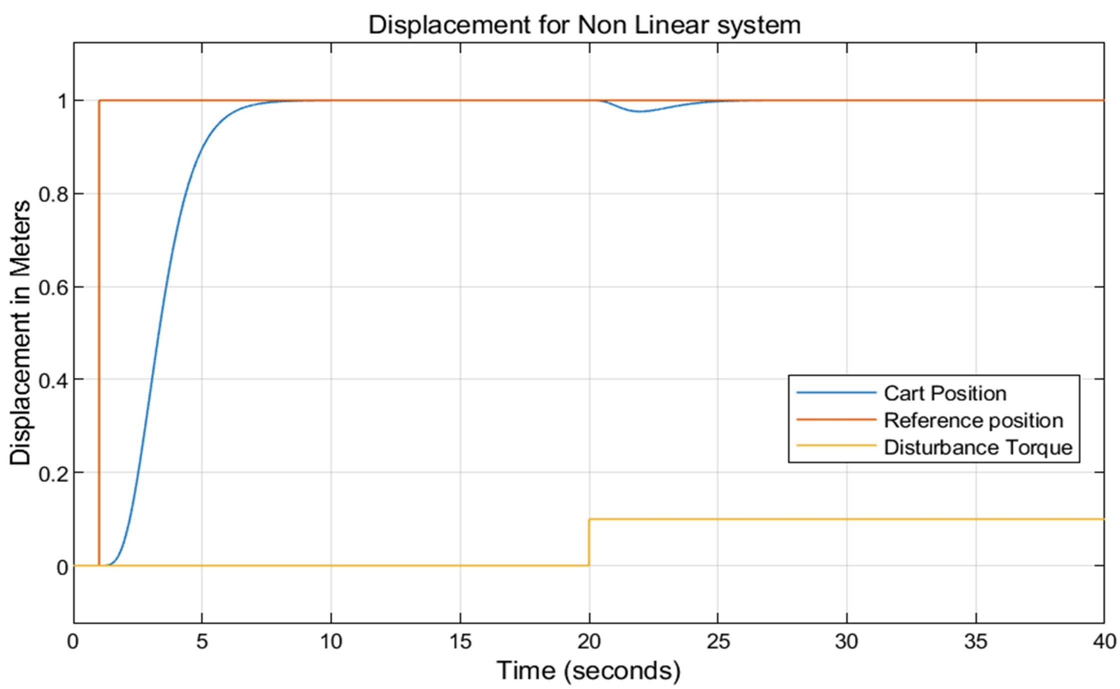
Displacement for Non-Linear System

The graph shows the displacement response of the systems for a Sample rate of 800Hz, pole of -2 and without disturbance torque. When the pole is changed from -1 to -2, similar to the graph of the system with pole of -1, the response initially accelerates towards the reference position but without overshooting around reference position and attaining the stability around the reference point faster than the systems with pole of -1 of sample time 0.08s and 0.001s.

6. Comparison of Displacement (With Disturbance Torque and Pole of -2)



Displacement of Linear System



Displacement for Non-Linear System

The Graph shows the displacement response of the system for a sample rate of 800Hz, pole of -2 and with disturbance torque. When the pole is changed from -1 to -2 the response initially accelerates towards the reference position similar to the graph of the system with pole with -1 but without overshooting around reference position attaining the stability around the reference point faster than the systems with pole of -1 of sample time 0.08s and 0.001s when a disturbance torque is introduced, the response of the system experiences only a slight deviation(Steady state Error) before stabilizing back to the reference position.

Task-7

Make a linear full-state observer based on the linear seesaw system. Place all poles at -3. Calculate the error feedback vector h. Validate the observer on the linear seesaw system for different initial conditions. Make the parallel model time discrete (zero-order-hold) with a sample rate of 8 kHz. Test the linear full-state observer on the nonlinear seesaw system with different initial conditions, different pole placements and different sample times?

A dynamic system that is used to estimate the state of another dynamic system is called an observer. Only the latter system's measured input and output are used to arrive at this estimate. An observer is said to be a full-order state observer when its order coincides with the order of the system being observed. The "residual or the difference between the observed output and the matching value produced by a model synthesized within the observer is calculated by full order observer, this is how the full order observer works. The observer's model then uses this residual as an input after it has been multiplied by a gain vector. The observer becomes an asymptotically stable dynamic system by choosing the gain appropriately, guaranteeing that the estimation error gradually approaches zero.

State observers utilize a model that runs parallel. The observer uses the same representation of state space as the original system. However, the output of both systems is different because of separate starting circumstances. Ackerman's formula is used to construct the error feedback vector, which decreases observation error. The observer poles are set to -3 to make the observer easier than the controller.

The observer can be designed as either a continuous-time system or a discrete-time system. A high sampling rate of 800 Hz is chosen for designing a discrete full-state observer in the second part of this task.

In this particular task, a model of the plant is introduced, which shares the same system description as the Linear Seesaw System. This alignment is achieved by setting the system matrix A, the input vector B, and the output vector Ct of the parallel model equal to the linear system. By employing the Ackermann formula, the manipulation output-error feedback vector h is calculated, which minimizes the state observer feedback error.

The output-error feedback vector 'h' employing Ackermann's formula is given as:

$$h = (\beta_0 I + \beta_0 A + \beta_{n-1} A^{n-1} + A^n)\, t_1$$

The following values are obtained using the MATLAB:

h_obs = -11.2967 -11.2487 54.1958 12.0000

Linear Time Continuous Observer System:

MATLAB CODE:

```
jb=0.5             %Jb=0.5kg*m^2
mf=0.1                      %mf=0.1kg
h=0.1              %h=10cm
g=9.81                      %g=9.81m*s^-2
Jconst=mf*h^2+jb


% state space system
% state vector x = [alpha_dot_obs; alpha_obs; xf_dot; xf]

A = [0, mf*g*h/Jconst, 0, -mf*g/Jconst;
    1, 0, 0, 0;
    0,-g, 0, 0;
    0, 0, 1, 0;]
B = [1/Jconst;
    0;
    0;
    0;]
C = eye(4);
D = [0;]
x1 = [0,0,0,1]

%building Observer controllability matrix
S_obs = [Ct; Ct*A; Ct*A*A; Ct*A*A*A;]

%Inv the Observer Controllability Matrix
S_obs_inv = inv(S_obs)

%Loading the last Row of Matrix
qt_obs = S_obs_inv(:,4)

%Getting the Polynomial
alpha_obs = poly([-3,-3,-3,-3])

%assigning values according to the equation
alpha_obs0 = alpha_obs(5)
alpha_obs1 = alpha_obs(4)
alpha_obs2 = alpha_obs(3)
alpha_obs3 = alpha_obs(2)
alpha_obs4 = alpha_obs(1)

K_obs = ((alpha_obs0*eye(4)) + (alpha_obs1*A) + (alpha_obs2*A*A) + (alpha_obs3*A*A*A) +
(alpha_obs4*A*A*A*A))*qt_obs

h_obs = K_obs              %Output error Feedback

h_trans = transpose(h_obs)
```

SIMULINK MODEL:-

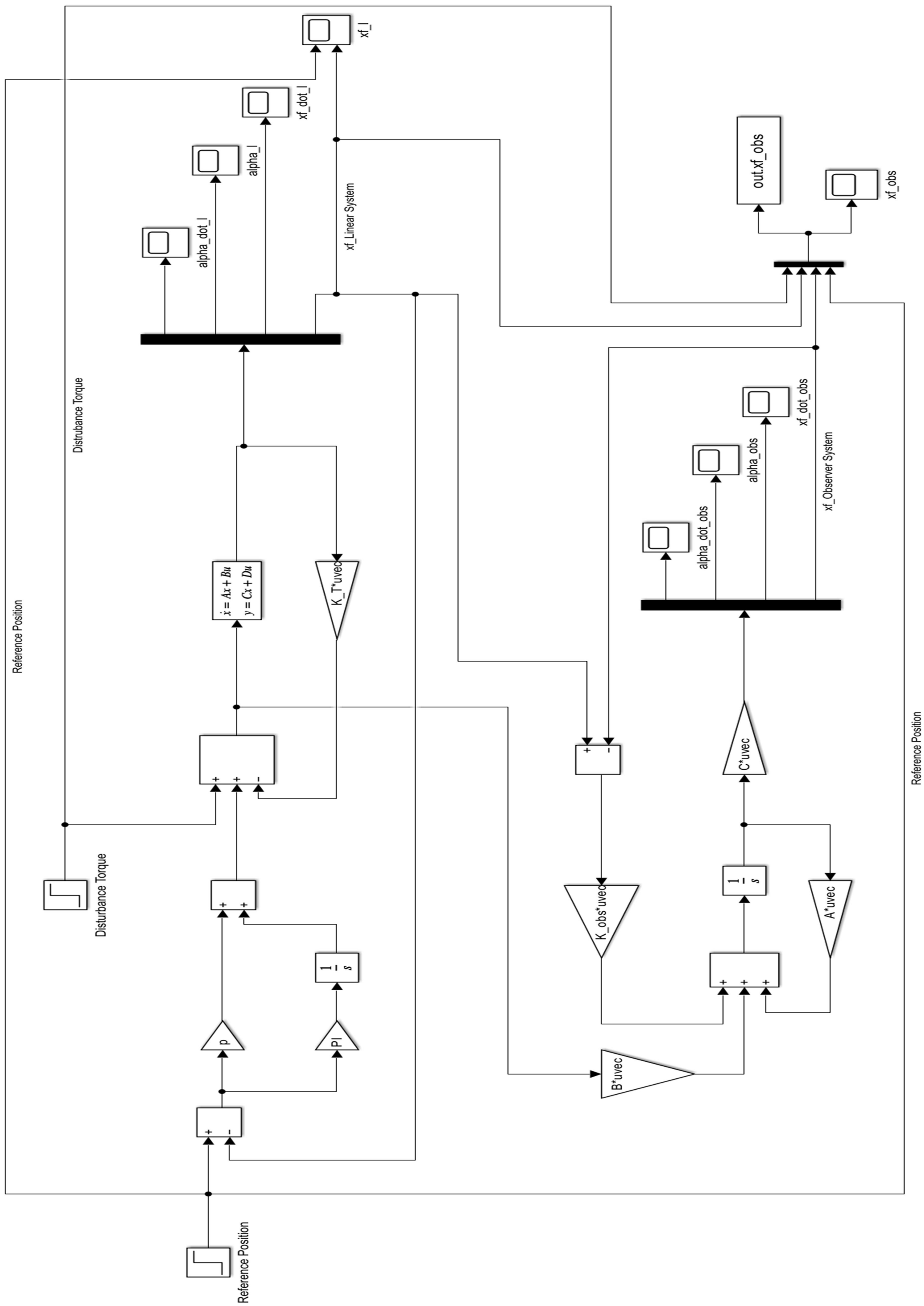


Figure 16: Simulink Model of Linear Time Continuous Observer

Output:

1. Comparison of Displacement of Observer (For Different Initial condition)

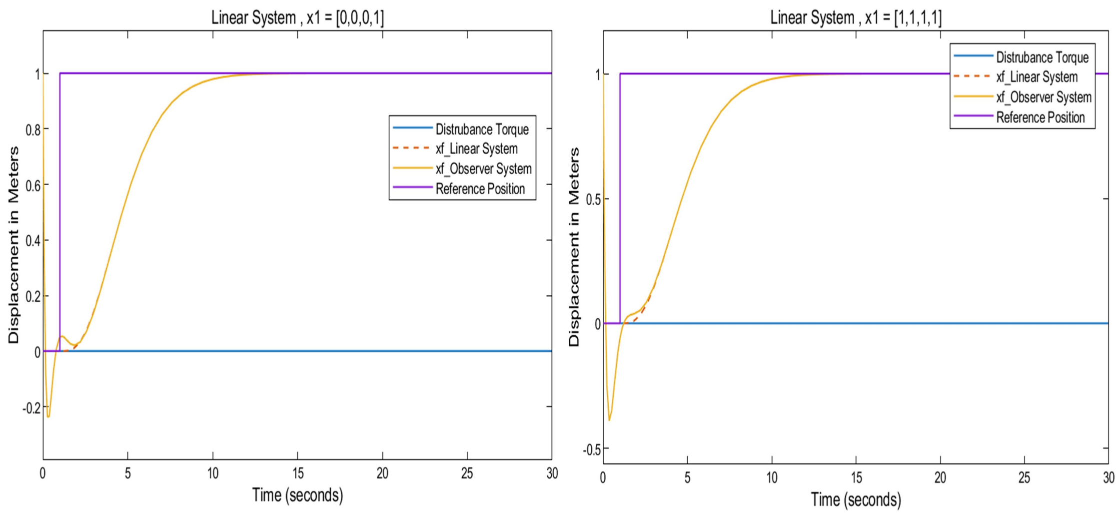


Figure 17: Linear Time Continuous System Observer Outputs

Figures 17 denote the output state of the Linear System along with the observer output.

When the input conditions were set as x1= [0 0 0 1], it is observed that the observer output goes in the negative direction in the beginning, a spike is observed at observer position. Thus, there is an error in the observer output. Soon, however the error is compensated, the error term then approaches to 0 and the observer follows the output of the Linear System.

When the initial conditions are set to [1 1 1 1]. Once again, the observer output overshoots in the opposite direction of the reference but negative spike is increased compared to previous initial condition and similar to the previous initial condition the error soon reaches 0 and the observer follows the output of the Linear System.

Linear Time Discrete Observer System:

MATLAB Code:-

```
jb=0.5              %Jb=0.5kg*m^2
mf=0.1                      %mf=0.1kg
h=0.1               %h=10cm
g=9.81                      %g=9.81m*s^-2
Jconst=mf*h^2+jb

% state space system
% state vector x = [alpha_dot_obs; alpha_obs; xf_dot; xf]

A = [0, mf*g*h/Jconst, 0, -mf*g/Jconst;
    1, 0, 0, 0;
    0,-g, 0, 0;
    0, 0, 1, 0;]
B = [1/Jconst;
    0;
    0;
    0;]
C = eye(4);
D = [0;]
x1 = [0,0,0,1];
fs = 800
Ts = 1/fs                   %Sampling time
sys_obs = ss(A,B,C,D);
sys_obs_d = c2d(sys_obs,Ts);
[Ad_obs, Bd_obs, Cd_obs, Dd_obs] = ssdata(sys_obs_d);
cT_obs_d = Cd_obs(4,1:4);
desys_obs = tf(1,poly([-3,-3,-3,-3]));
desys_obs_d = c2d(desys_obs,Ts);
[num_obs,den_obs] = tfdata(desys_obs_d,'V');
pole_place = roots(den_obs);
k_obs_d = acker(Ad_obs.',cT_obs_d.',pole_place.');
h_obs_d = k_obs_d.';
```

Simulink Model:-

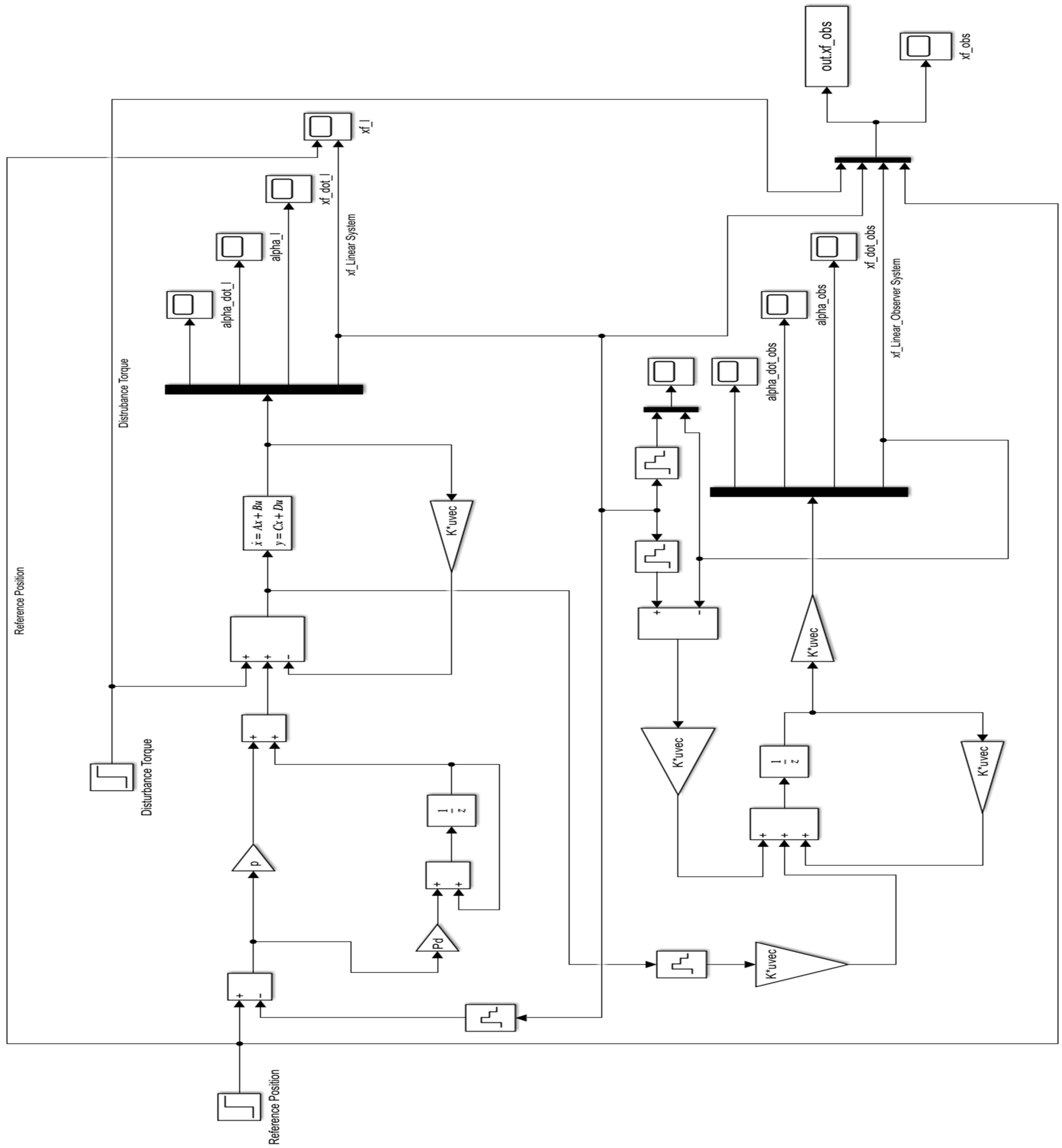
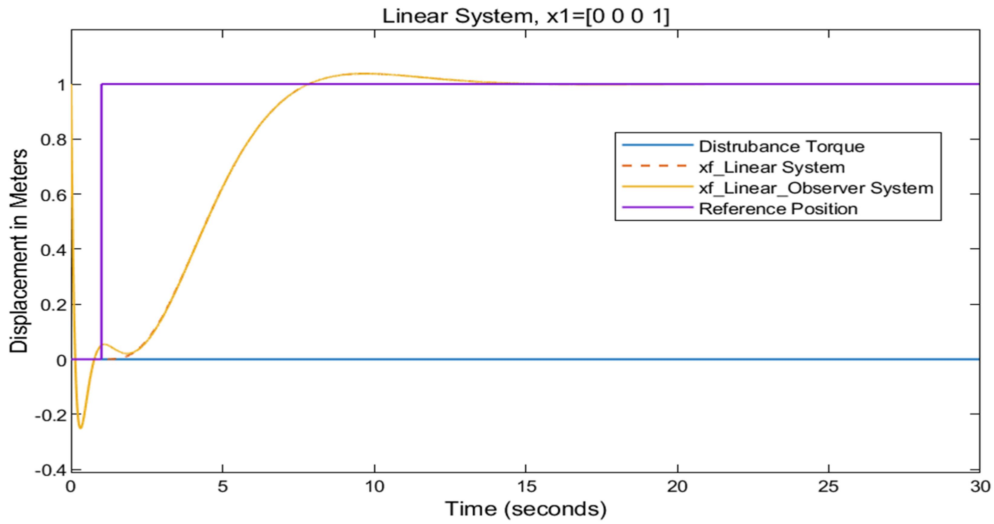


Figure 18: Simulink Model of Linear Time Discrete Observer

Output:

1. Comparison of Displacement of Observer (For Different Initial condition)



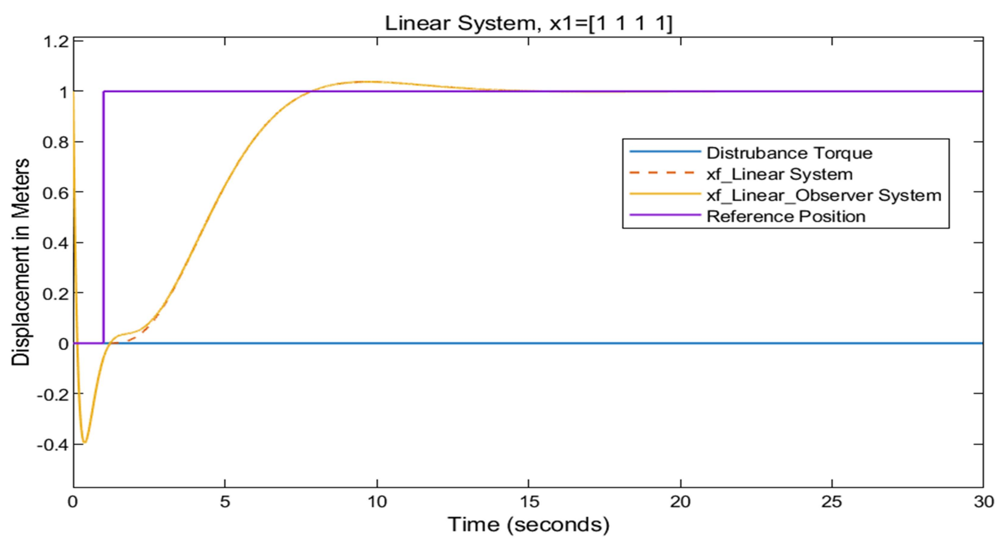


Figure 19: Linear Time Discrete System Observer Outputs

Figures 19 represent the output of the discrete state observer. It is observed that the output behaviour does not change with this transformation from time continuous and time discrete state linear system but the output error feedback vector h completely changes, i.e from MATLAB code (h_obs_d = -0.0141 -0.0140 0.0674 0.0150).

Non- Linear Time Continuous Observer System:

Simulink Model:

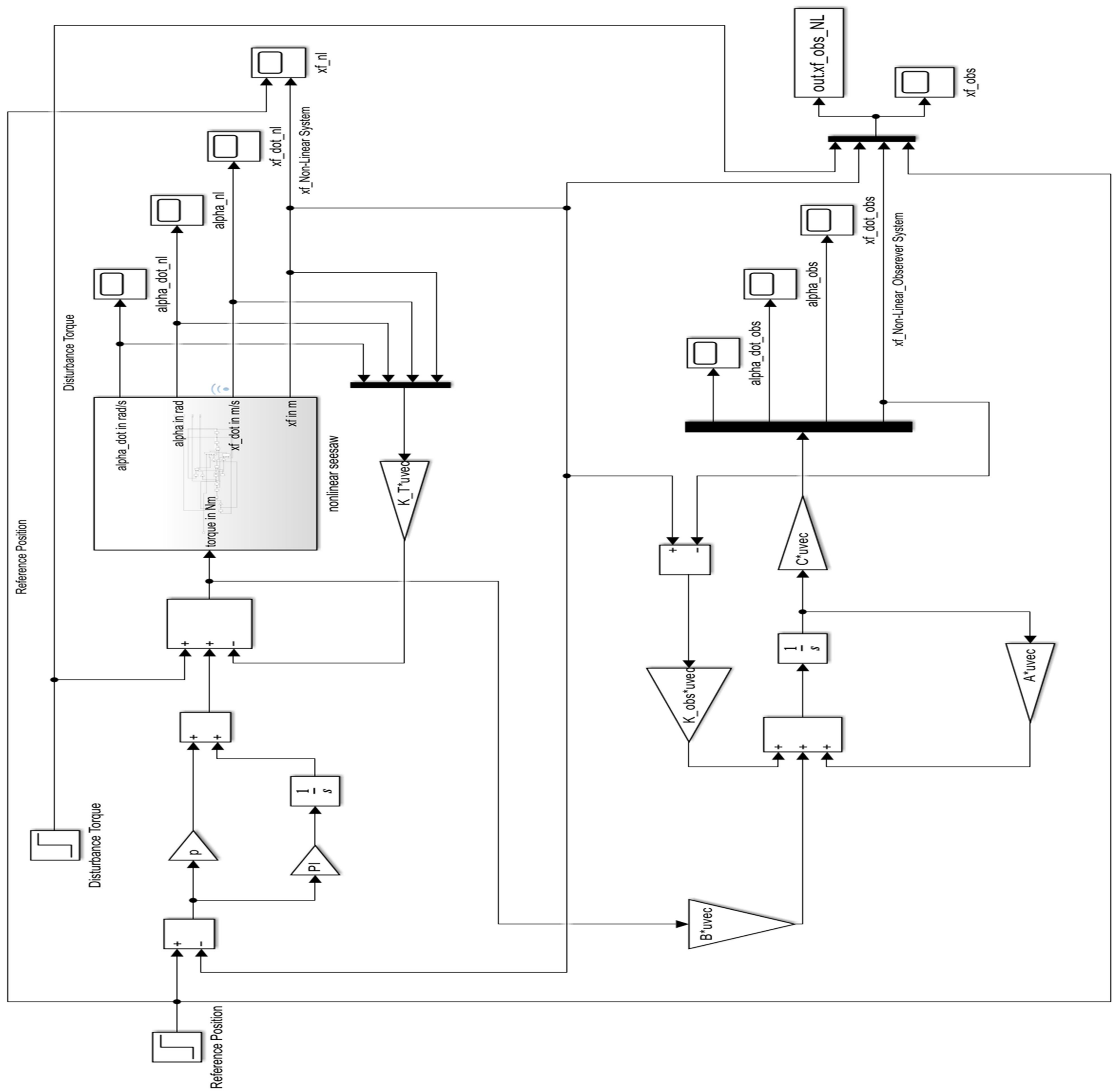
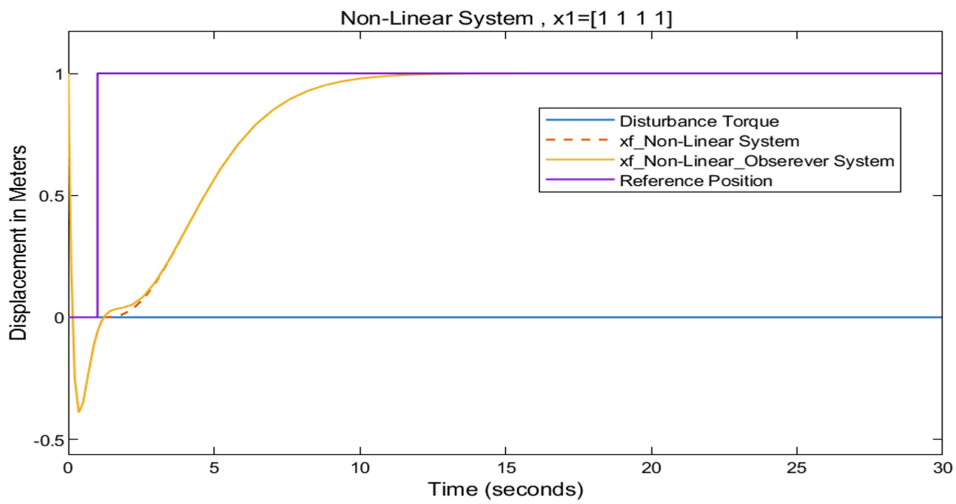


Figure 20: Simulink Model of Non-Linear Time Continuous Observer

<u>Output</u>:

Comparison of Displacement of Observer (For Different Initial condition)



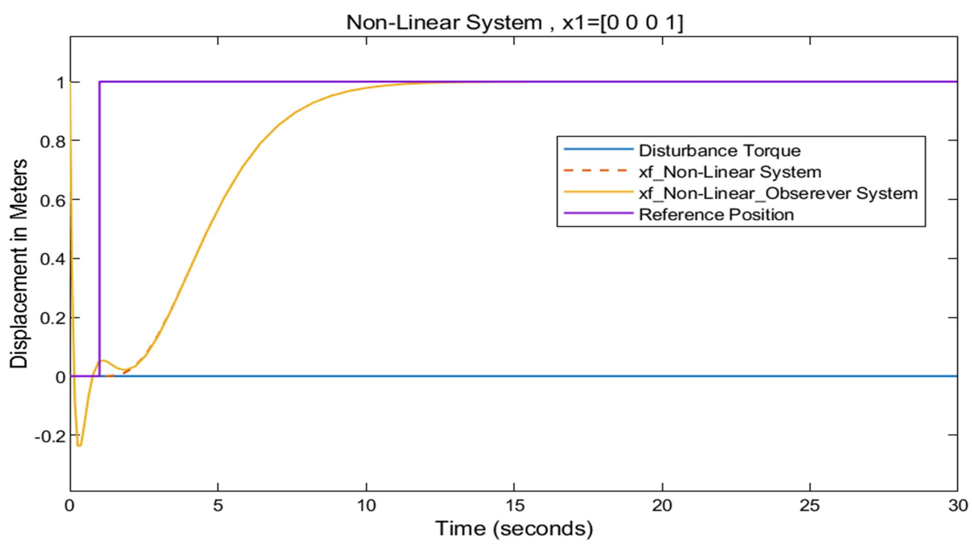


Figure 21: Non-Linear Time Continuous System Observer Outputs

Similar to time continuous linear model, the response of time continuous nonlinear model also follows the reference torque with initial spike at observer side and error soon reaches 0. The observer follows the output of the Non-Linear System.

Non- Linear Time Discrete Observer System:

Simulink Model:

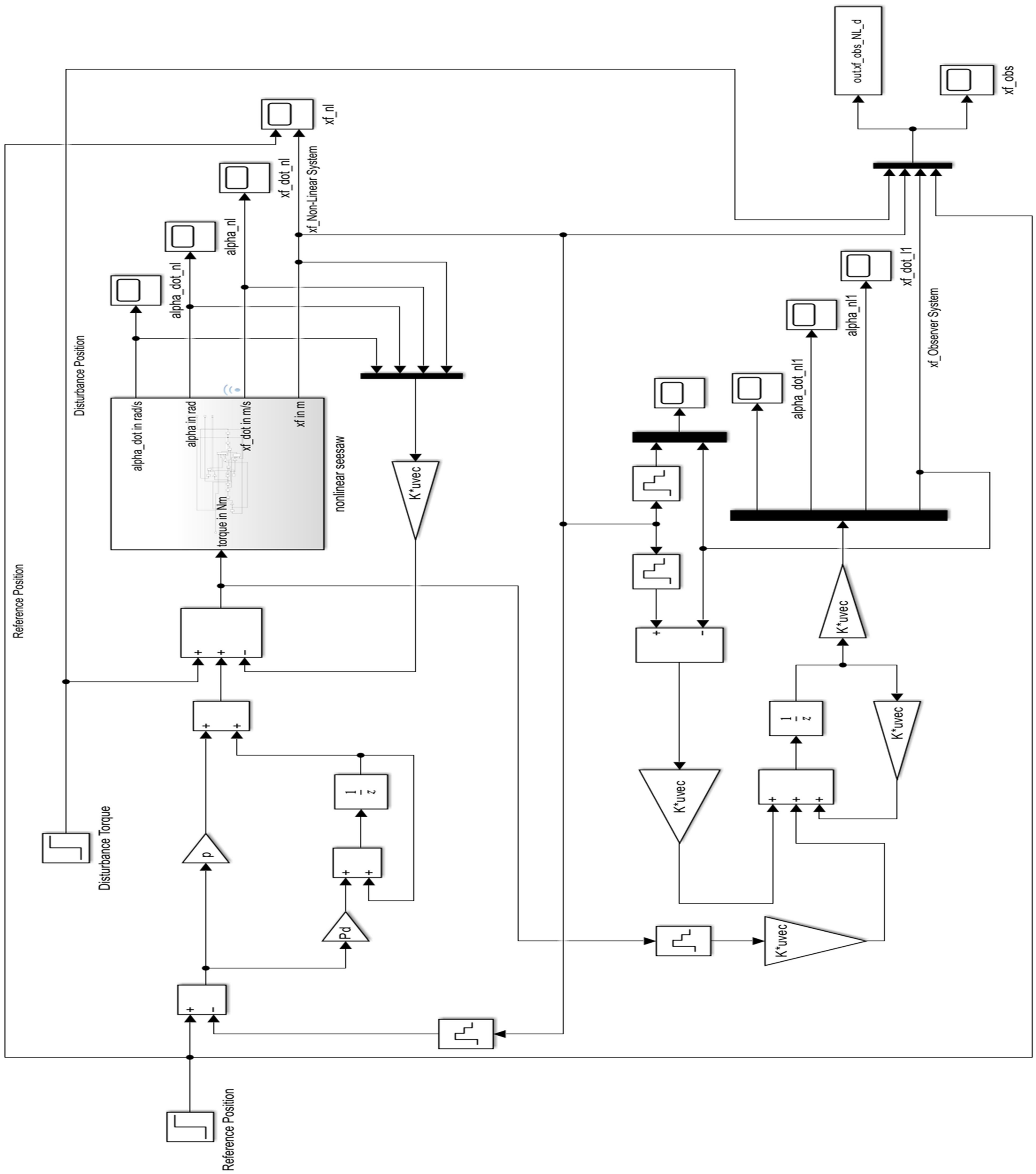


Figure 22: Simulink Model of Non-Linear Time Discrete Observer.

Output:-

Comparison of Displacement of Observer (For Different Initial condition)

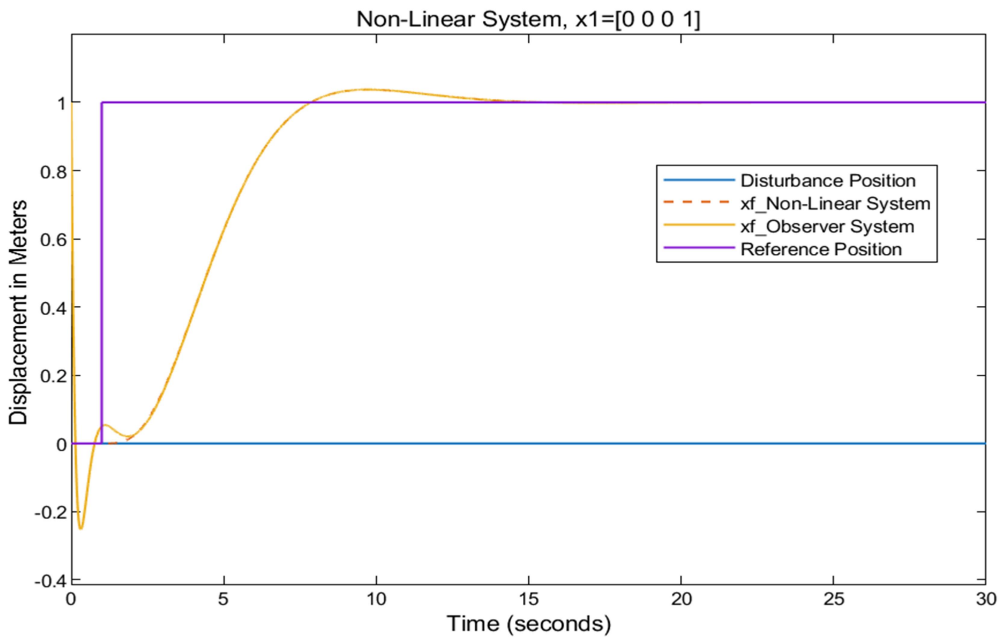
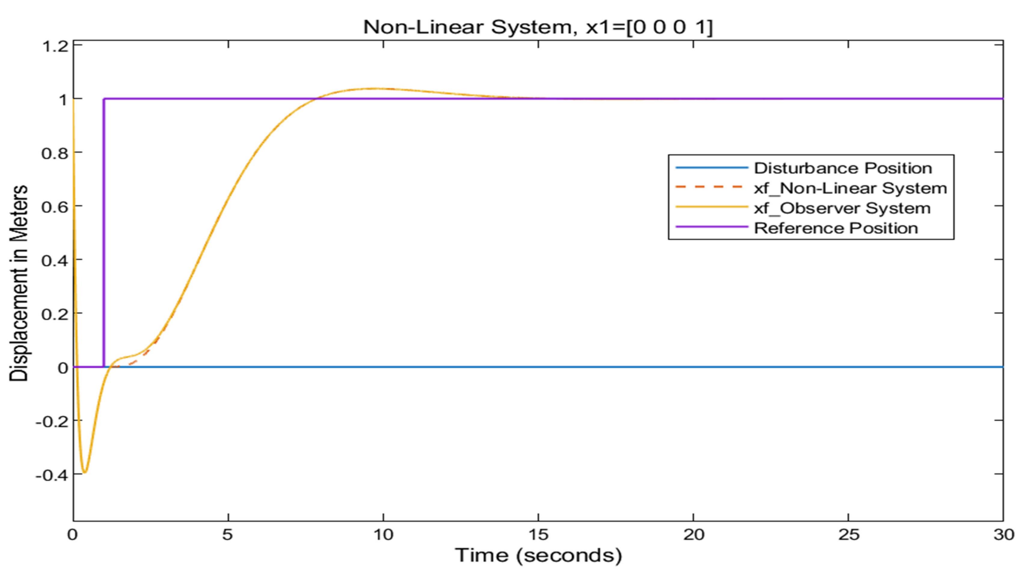


Figure 23: Non-Linear Time Discrete System Observer Outputs



Similar to time discrete linear model, the response of time discrete nonlinear model also follows the reference torque with initial spike at observer side and error soon reaches 0. The observer follows the output of the Non-Linear System.

Comparison of Linear and Non-Linear Observer Systems:

1. Comparison of Time Continuous Observer Systems (For X1 = [1 0 0 1]):

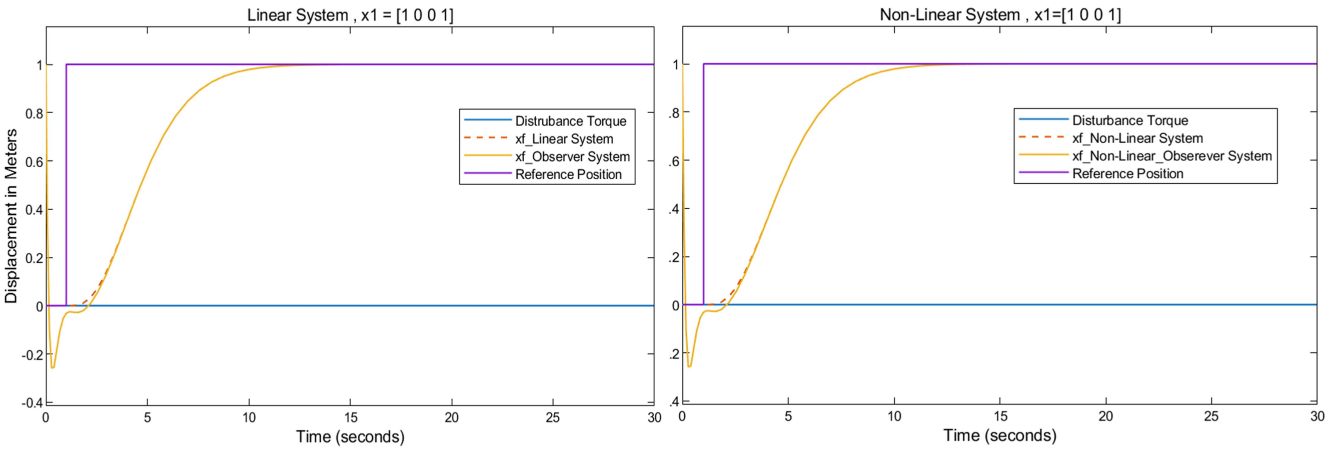


Figure 24: Linear and Non-Linear Time Continuous System Observer Outputs

Figure 24, denotes the Observer State Systems output of Linear and Non-Linear System respectively. Both Systems (Linear and Non-Linear) shows similar responses when initial condition is changed, whereas compared to previous case ([0 0 0 1]) of Linear and Non-Linear Time Continuous Observer Systems, responses is almost similar but the observer systems take longer to counteract the initial disturbance and move towards the reference, whereas for x1 = [0 0 0 1], the observer systems rapidly moves towards the reference without significant fluctuations below it.

2. Comparison of Time Continuous Observer Systems (For Pole = [-5 -5 -5 -5]):

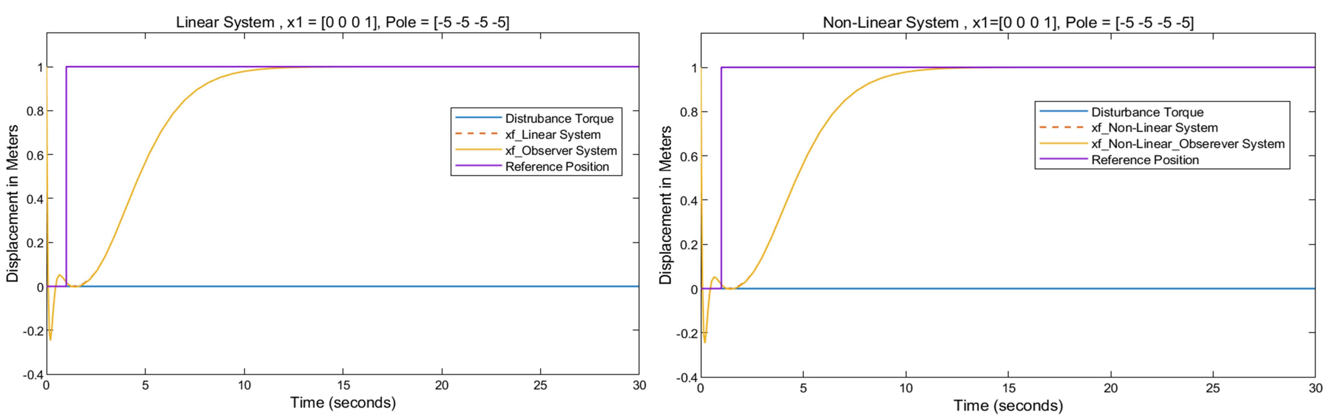


Figure 25: Non-Linear Time Continuous System Observer Outputs

Figure 25, denotes the Observer State Systems output of Linear and Non-Linear System respectively Both Systems (Linear and Non-Linear) shows similar responses when Pole is changed, Whereas compared to previous case ([-3 -3 -3 -3]) of Linear and Non-Linear Time Continuous Observer Systems, responses is almost similar but the observer systems takes less times to counteract the initial disturbance and move towards the reference, whereas for Pole = [-3 -3 -3 -3], it takes slightly more times to counteract the initial disturbance and move towards the reference.

3. Comparison of Time Discrete Observer Systems (For X1 = [1 0 0 1] and f= 1000Hz):

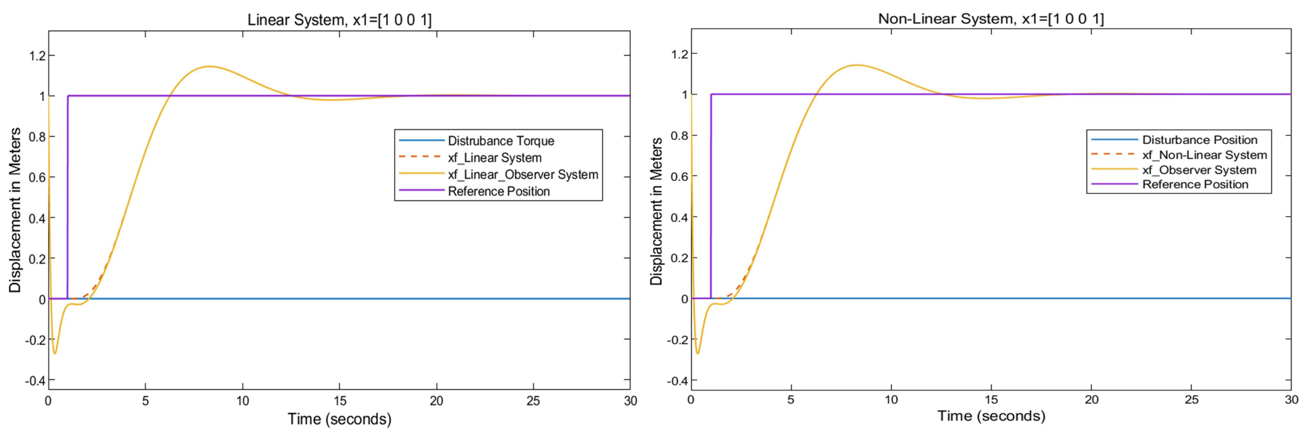
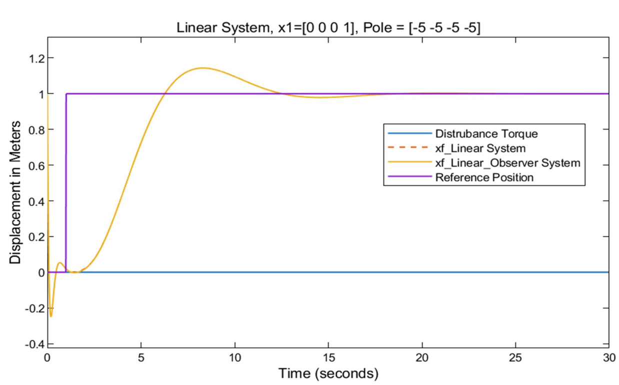


Figure 26: Linear and Non-Linear Time Discrete System Observer Outputs

Figure 26, denotes the Observer State Systems output of Linear and Non-Linear System respectively. Both Systems (Linear and Non-Linear) shows similar responses when initial condition is changed, whereas compared to previous case ([0 0 0 1]) of Linear and Non-Linear Time Discrete Observer Systems, responses is almost similar but the observer systems take longer to counteract the initial disturbance and move towards the reference, whereas for x1 = [0 0 0 1] of Time Discrete Systems, the observer systems rapidly move towards the reference without significant fluctuations below it.

2. Comparison of Time Discrete Observer Systems (For Pole = [-5 -5 -5 -5] and f = 1000Hz):



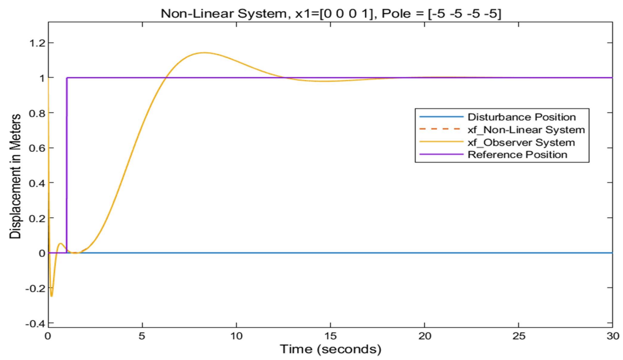


Figure 27: Non-Linear Time Continuous System Observer Outputs

Figure 27, denotes the Observer State Systems output of Linear and Non-Linear System respectively. Both Systems (Linear and Non-Linear) shows similar responses when Pole is changed, whereas compared to previous case ([-3 -3 -3 -3]) of Linear and Non-Linear Time Discrete Observer Systems, responses is almost similar but the observer systems take less times to counteract the initial disturbance and move towards the reference, whereas for Pole = [-3 -3 -3 -3] of Time Discrete Systems, it takes slightly more times to counteract the initial disturbance and move towards the reference.

Task-8

Use the observed states from 7. For the controls 4. – 6. And test them on the linear and non-linear system. Discuss different initial conditions, pole placements, reference value steps and disturbance torque steps?

Simulink Model of State feedback observer for the Linear System (Task 4):

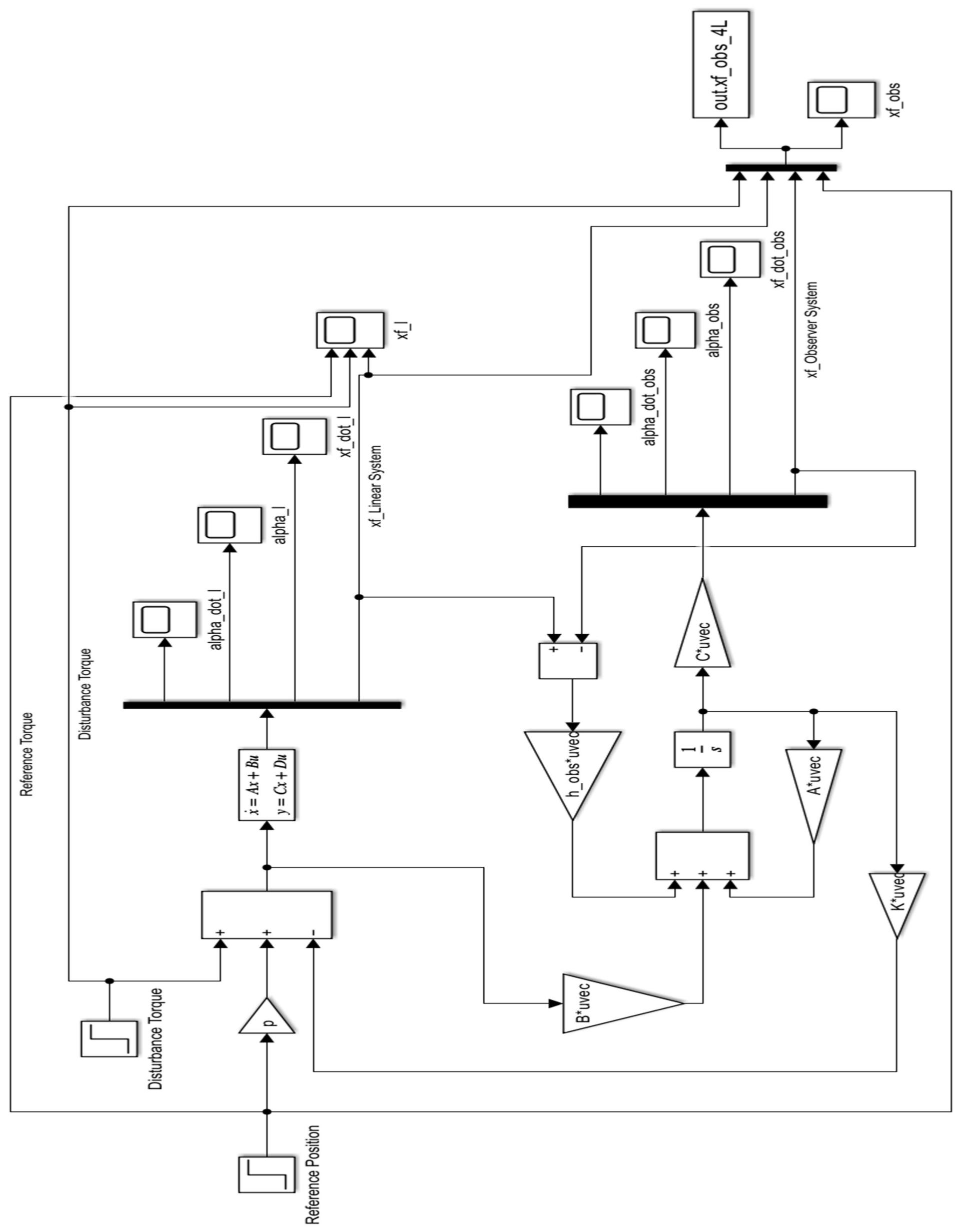


Figure 28: Simulink Model of State feedback observer for the Linear System.

Output:-

Comparison of displacement (For Different Initial Conditions, Pole, Disturbance Torque and Reference Torque.

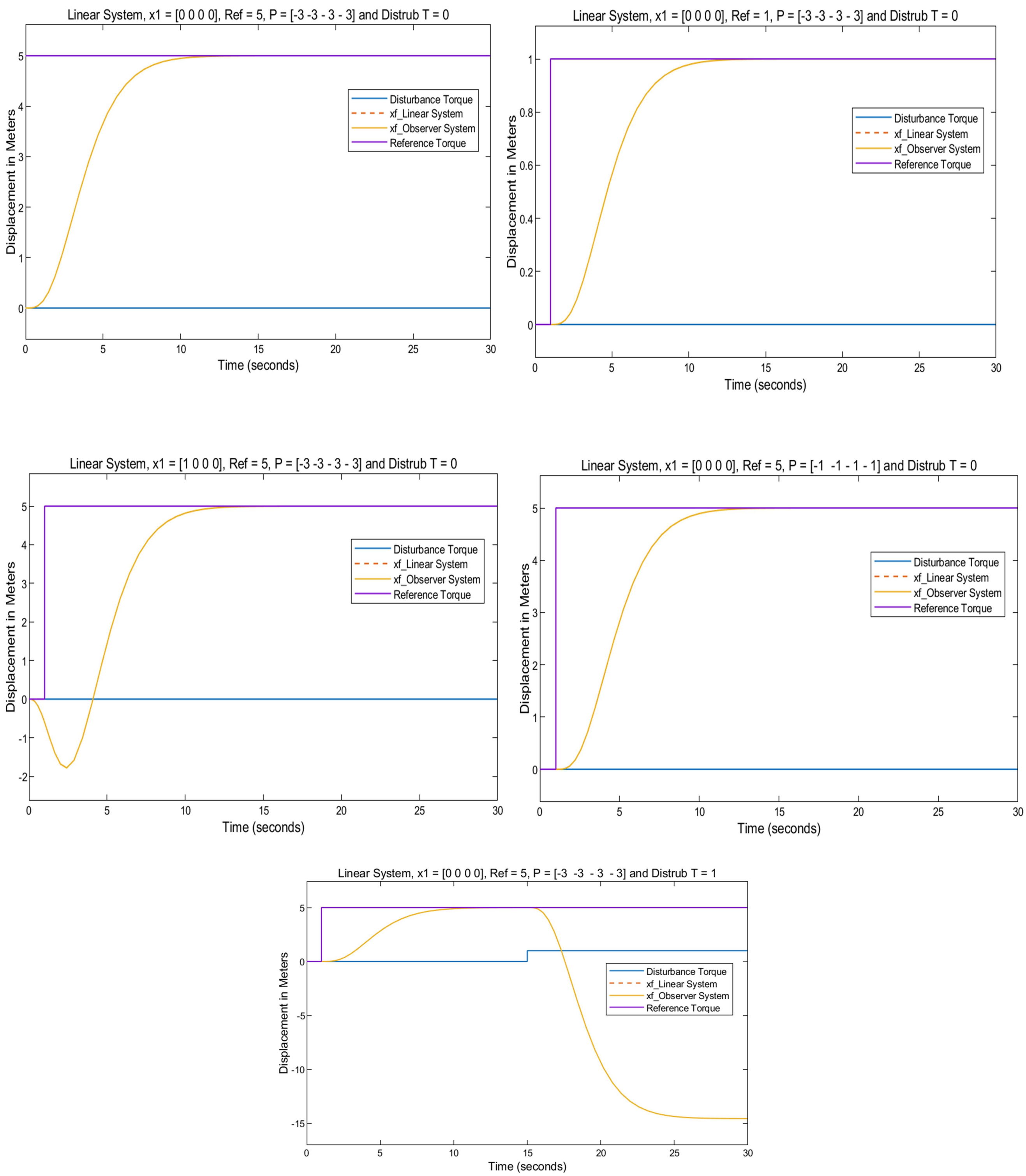


Figure 29: Outputs of State feedback observer for the Linear System.

For Linear System state feedback with observer, the system response follows the reference torque and when disturbance torque is applied, response get deviates from the reference torque towards negative values. When initial condition is changed, response goes in the negative direction in the beginning, a spike is observed. Thus, there is an error in the output, however the error is compensated, the error term then approaches to 0 and the response follows the output of the Linear System.

Simulink Model of State feedback observer for Non-Linear System (Task 4):

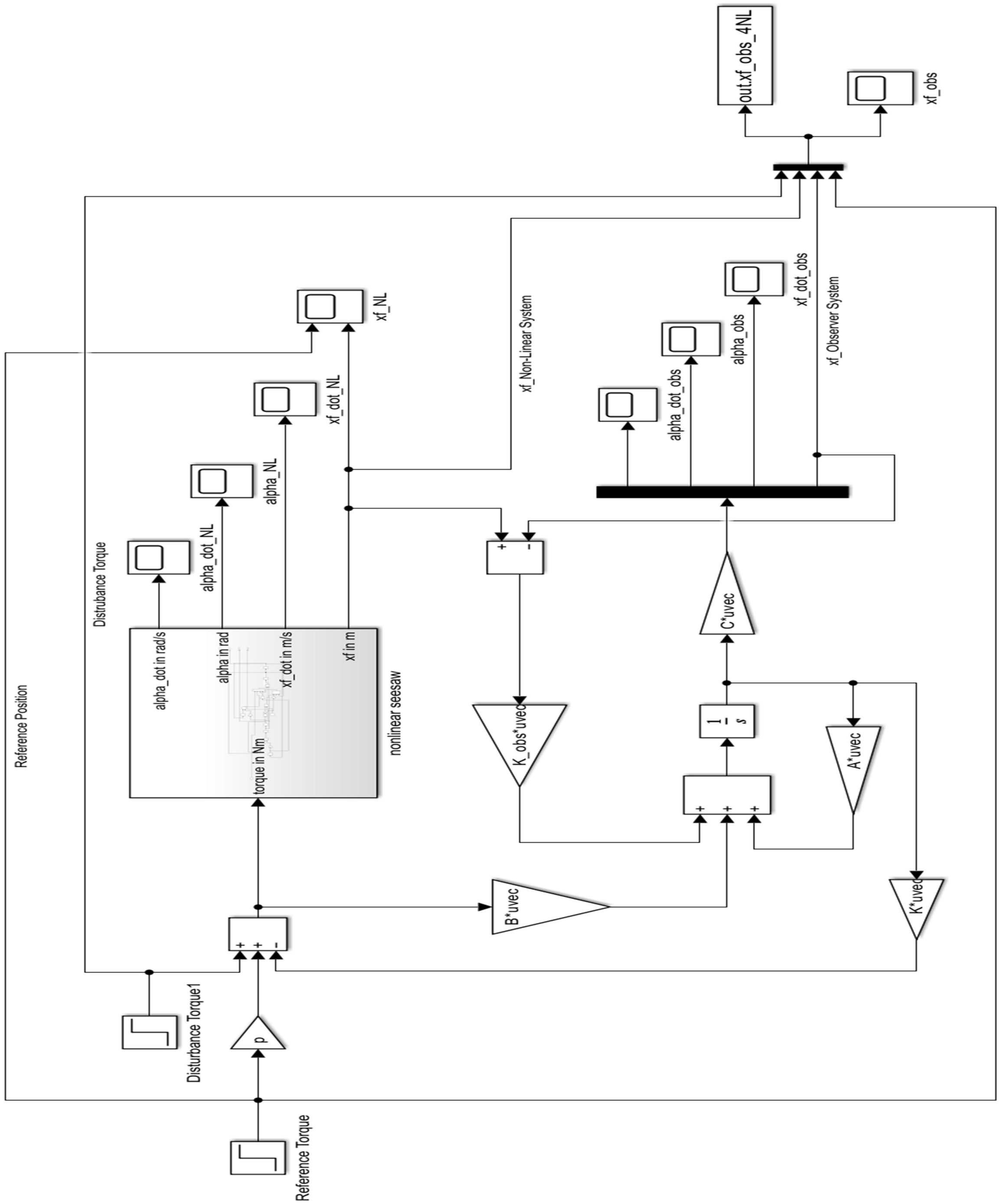


Figure 30: Simulink Model State feedback observer for the Non-Linear System

Output:

Comparison of displacement (For Different Initial Conditions, Pole, Disturbance Torque and Reference Torque):

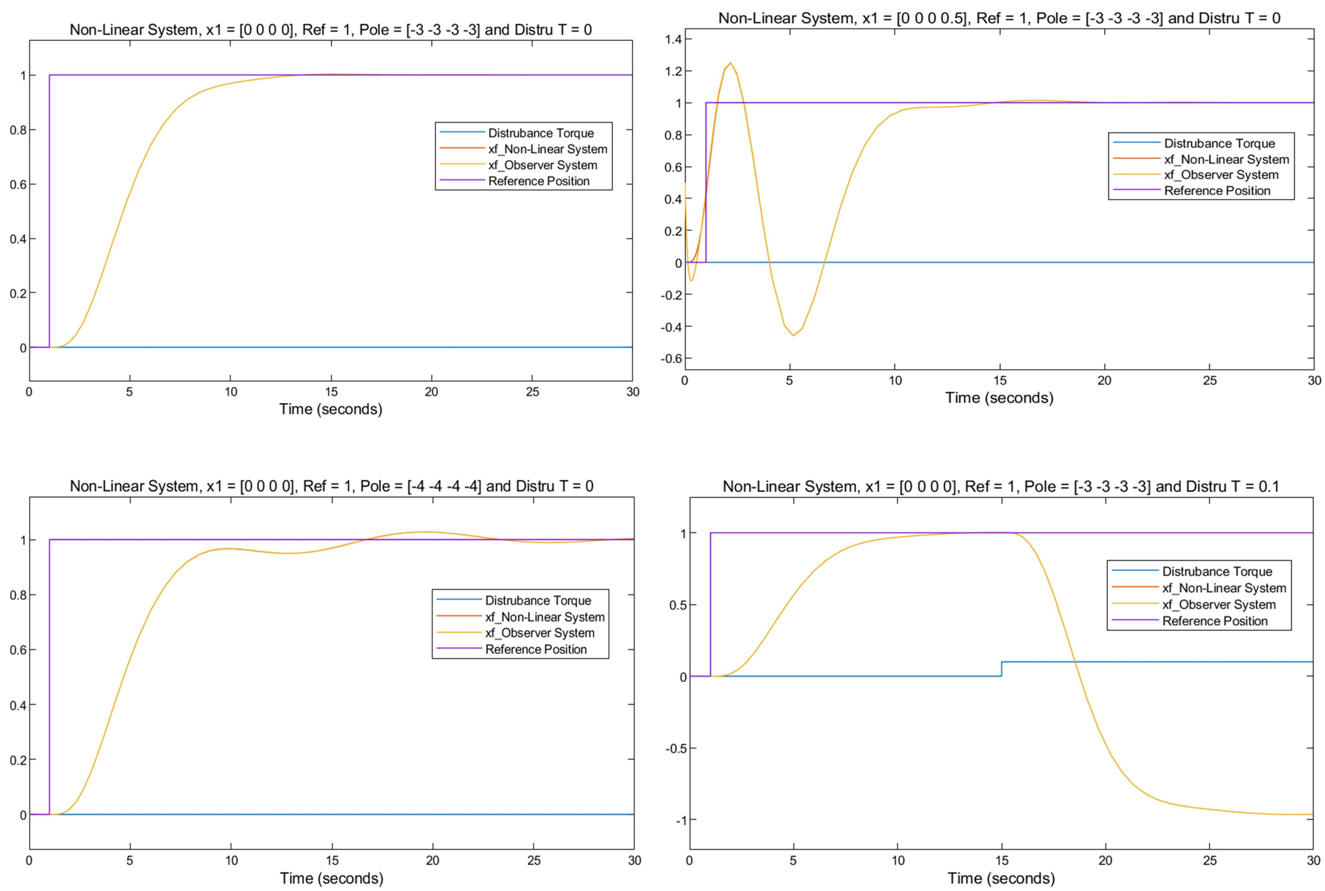


Figure 31: Outputs of State feedback observer for the Non-Linear System

For Non-linear System state feedback with observer, the system response follows the reference torque and when disturbance torque is applied, response get deviates from the reference torque towards negative values. When initial condition is changed, the system will deviate from the working point and become unstable. Thus, there is an error in the output, however the error is compensated, the error term then approaches to 0 and the response follows the output of the Linear System.

Simulink Model of State feedback observer for Linear System with PI controller (task 5)

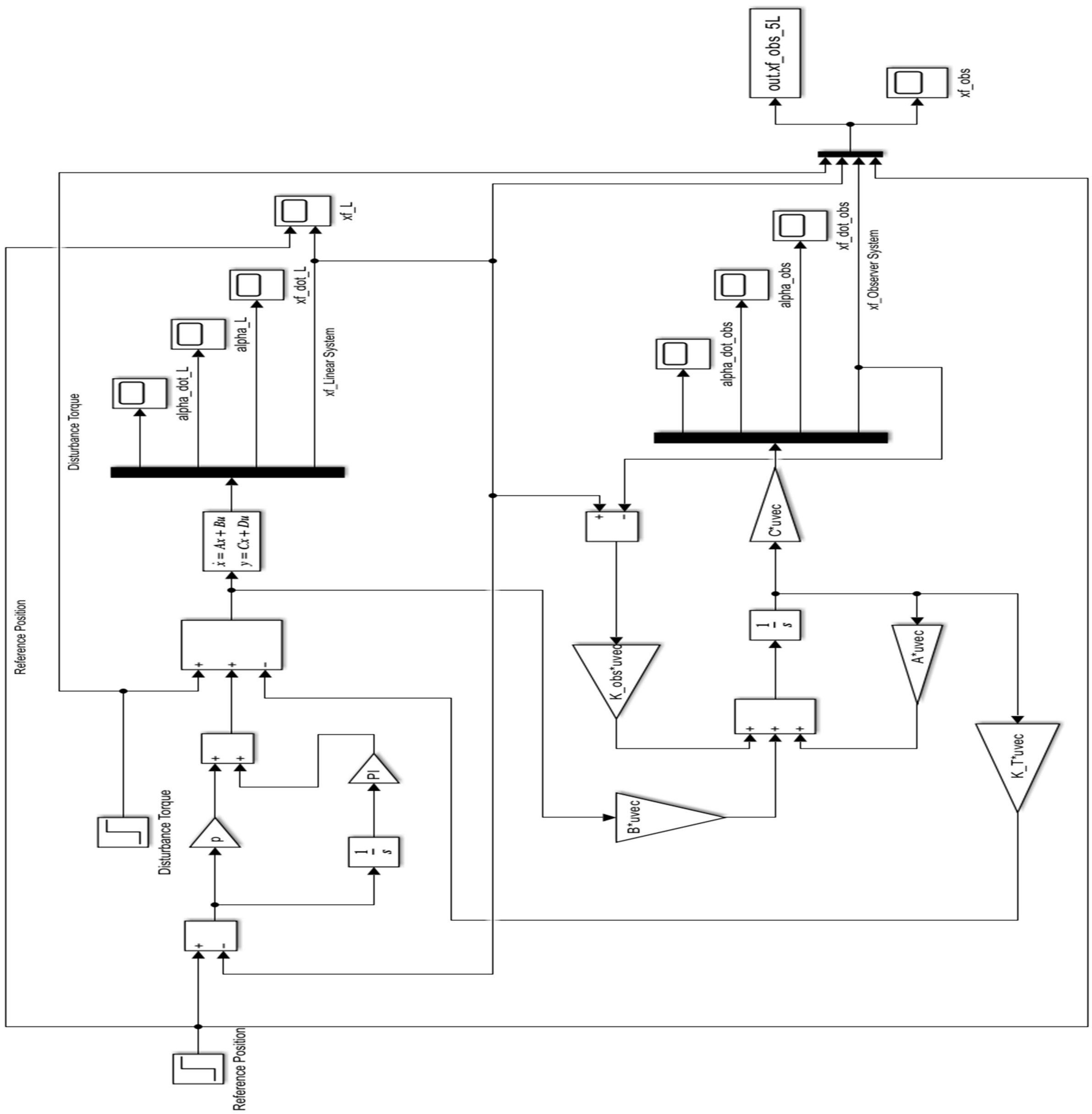


Figure 32: Simulink model for PI state feedback with observer for linear model

Output:

Comparison of displacement (For Different Initial Conditions, Pole, Disturbance Torque and Reference Torque):

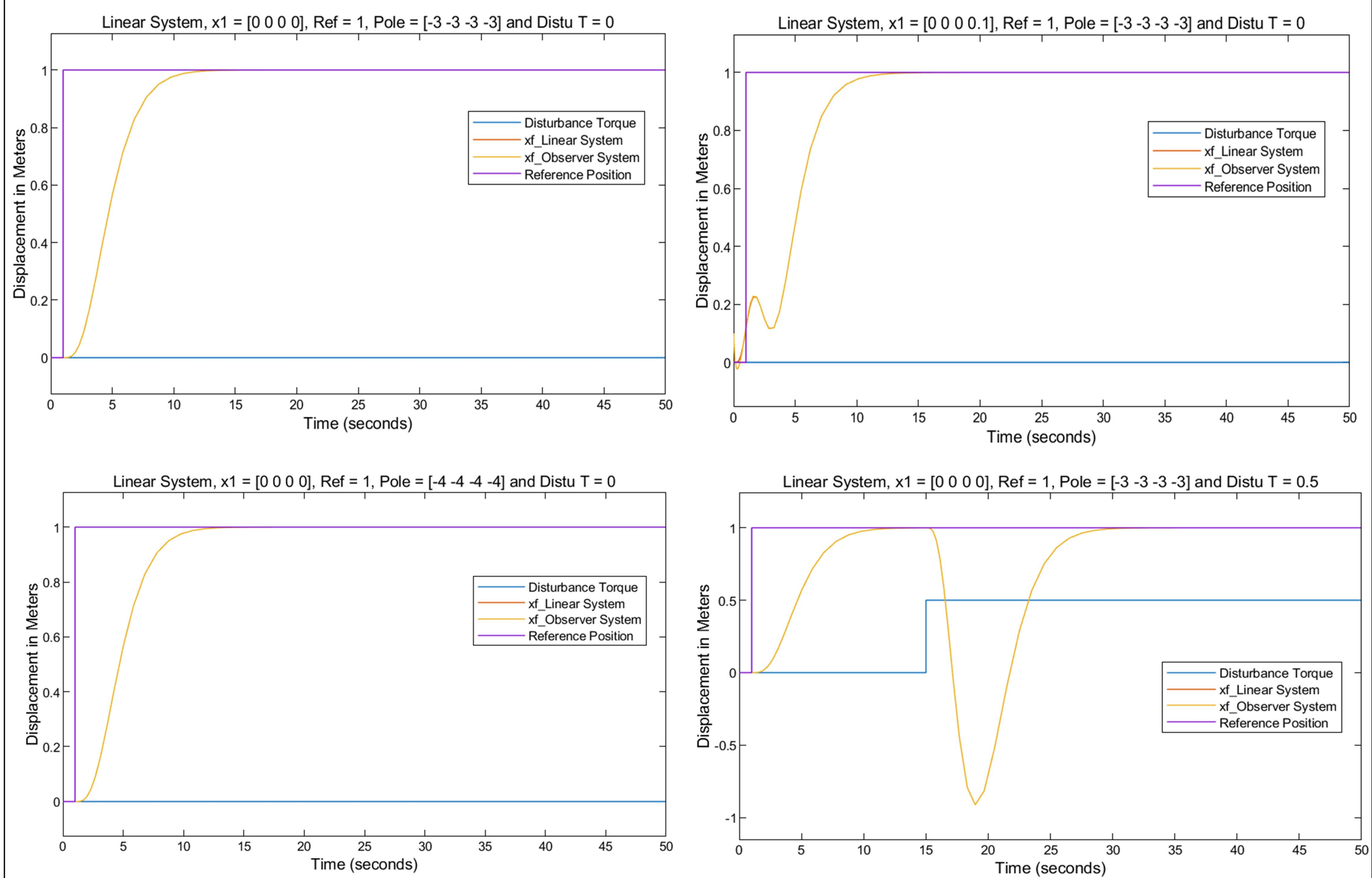


Figure 33: Outputs of PI state feedback with observer for Linear System

For Linear System PI state feedback with observer, the system response follows the reference torque and when disturbance torque is applied, response get deviates from the reference torque towards negative values. When initial condition is changed, the output of the system reaches a steady state after the initial oscillation. The response of the system is accurate even if there is change in magnitude of Pole.

Simulink Model of State feedback Discrete Observer for Linear System with Discrete PI controller (task 6)

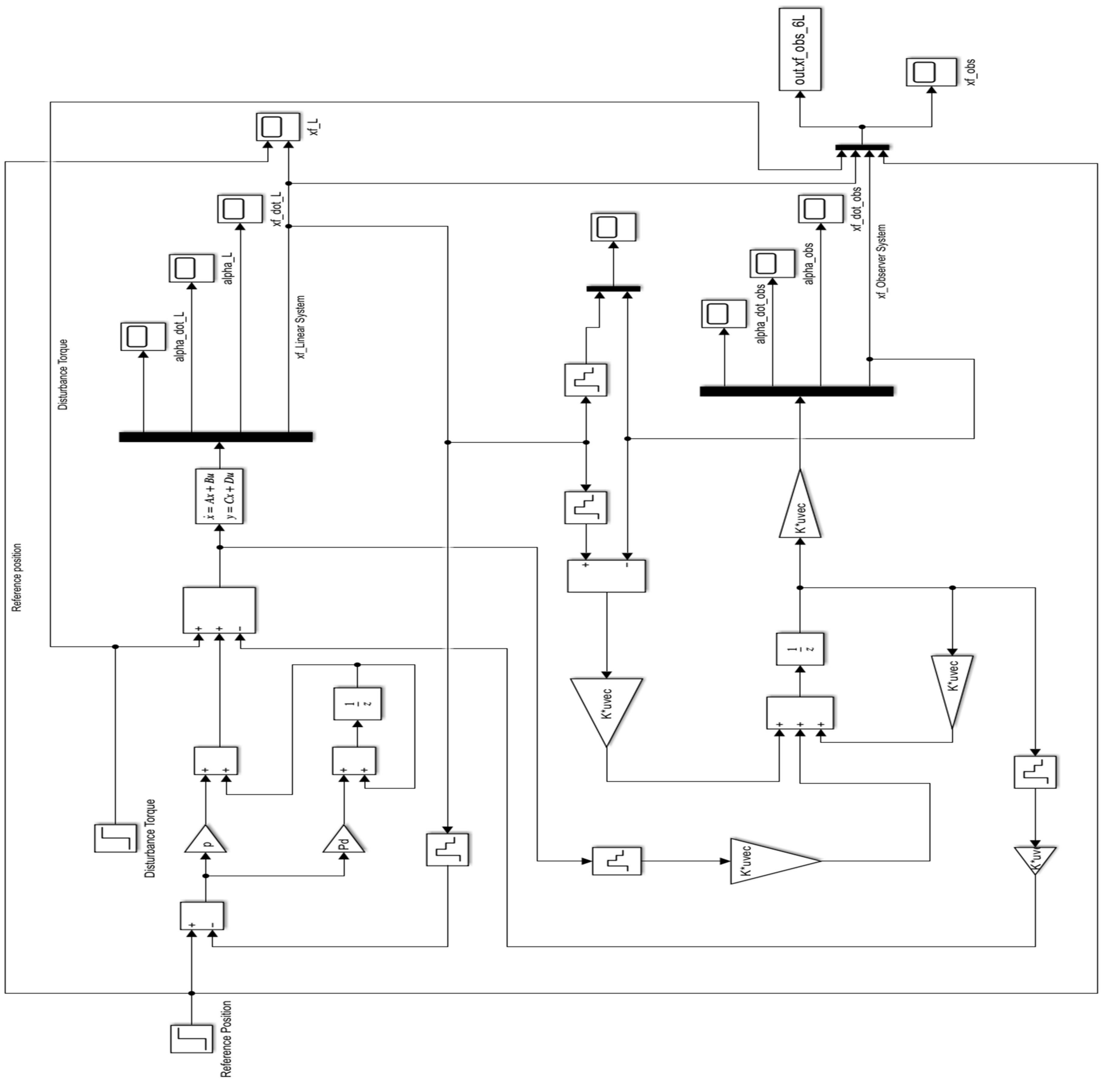


Figure 34: State feedback Discrete Observer for Linear System with Discrete PI controller

❑ Output:

Comparison of displacement (For Different Initial Conditions, Pole, Disturbance Torque and Reference Torque):

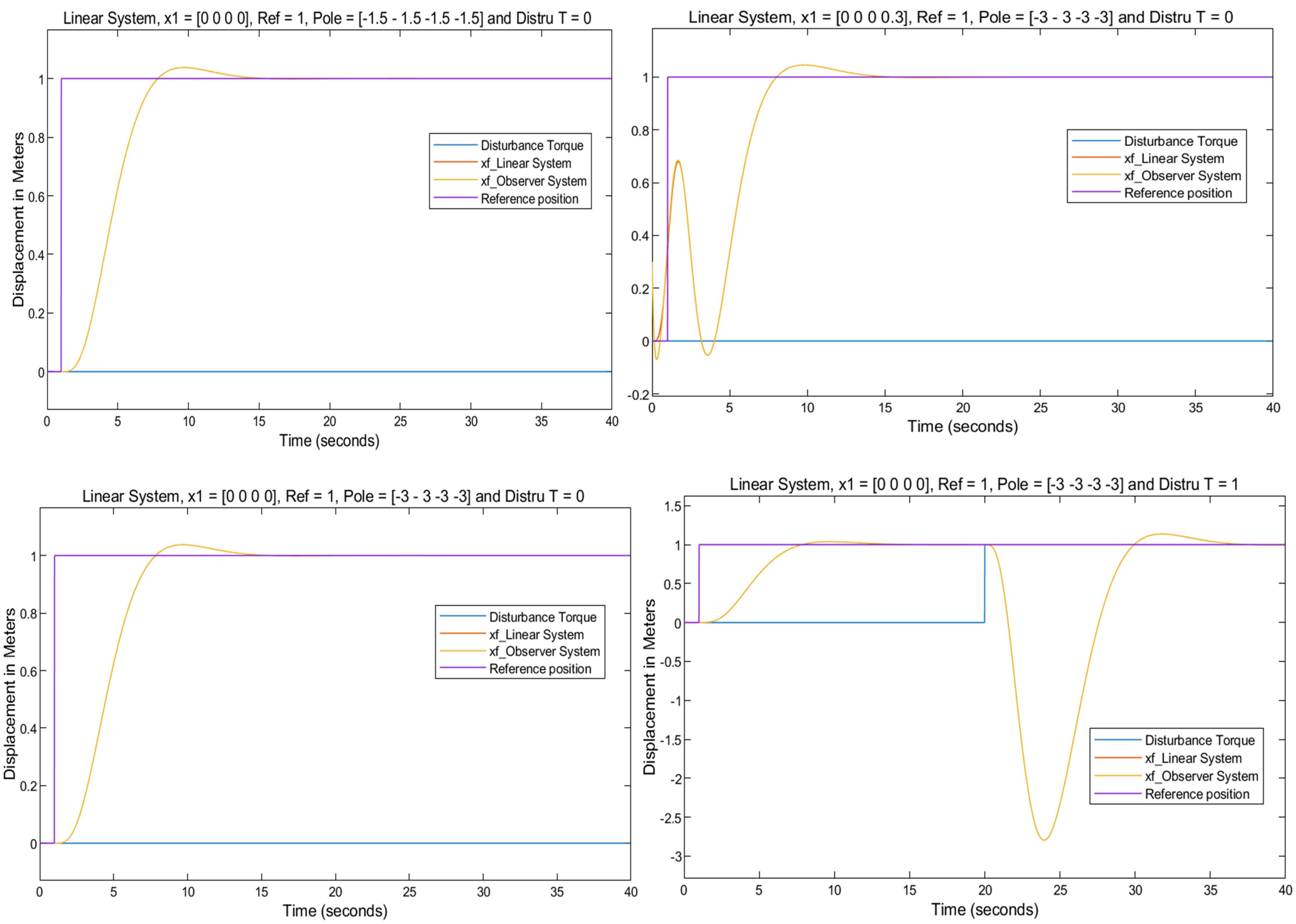


Figure 35: Outputs of State feedback Discrete Observer for Linear System with Discrete PI controller

For Linear System with Discrete PI controller with observer, the system response follows the reference torque and when disturbance torque is applied, response get deviates from the reference torque towards negative values. When initial condition is changed, the output of the system reaches a steady state after the initial oscillation. The response of the system is accurate even if there is change in magnitude of Pole, response of the System is similar to the Time Continuous PI controller.

Simulink Model of State Feedback Discrete Observer for Non-Linear System with Discrete PI controller (task 6)

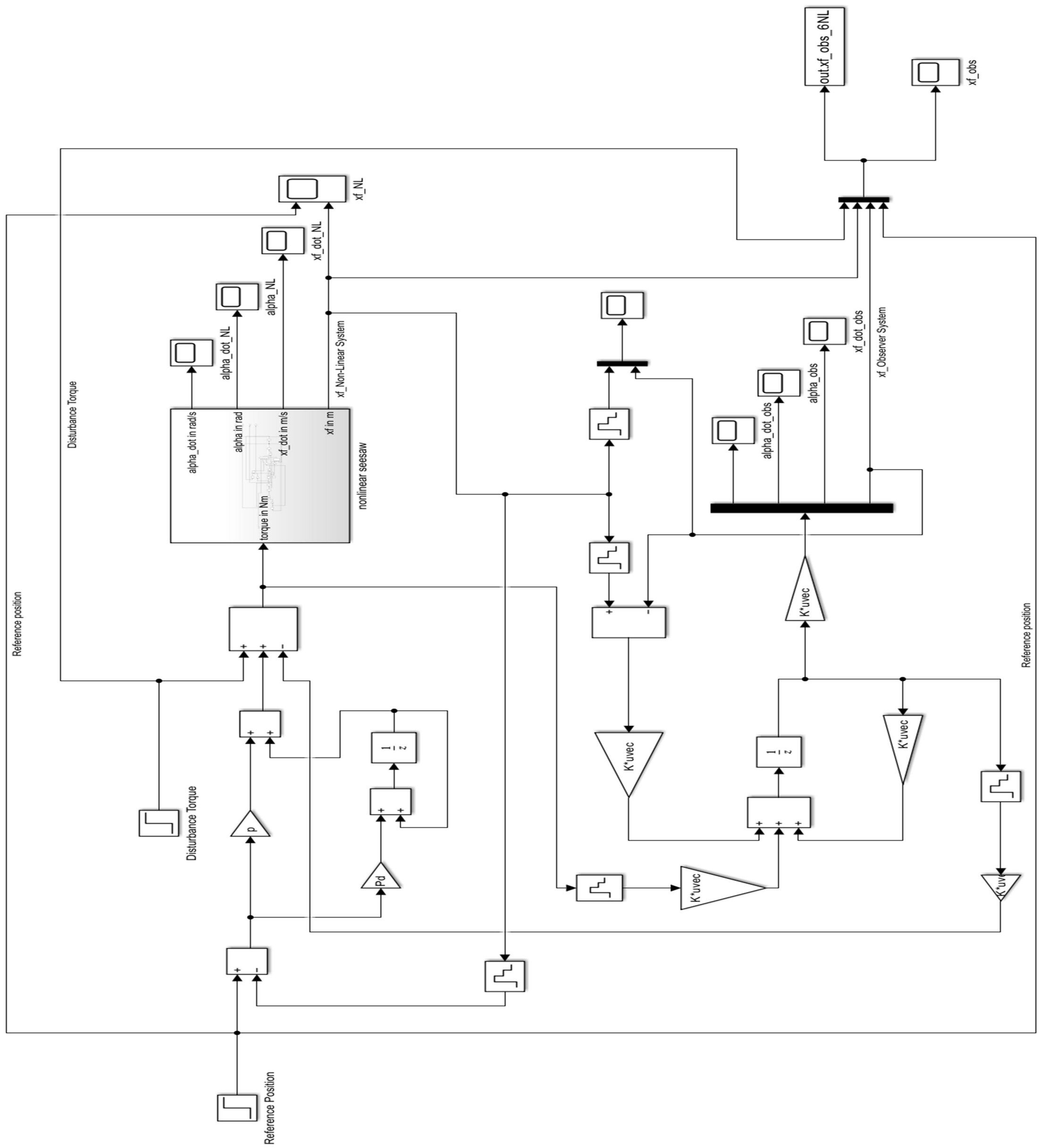


Figure 36: State feedback Discrete Observer for Non-Linear System with Discrete PI controller

Output:

Comparison of displacement (For Different Initial Conditions, Pole, Disturbance Torque and Reference Torque):

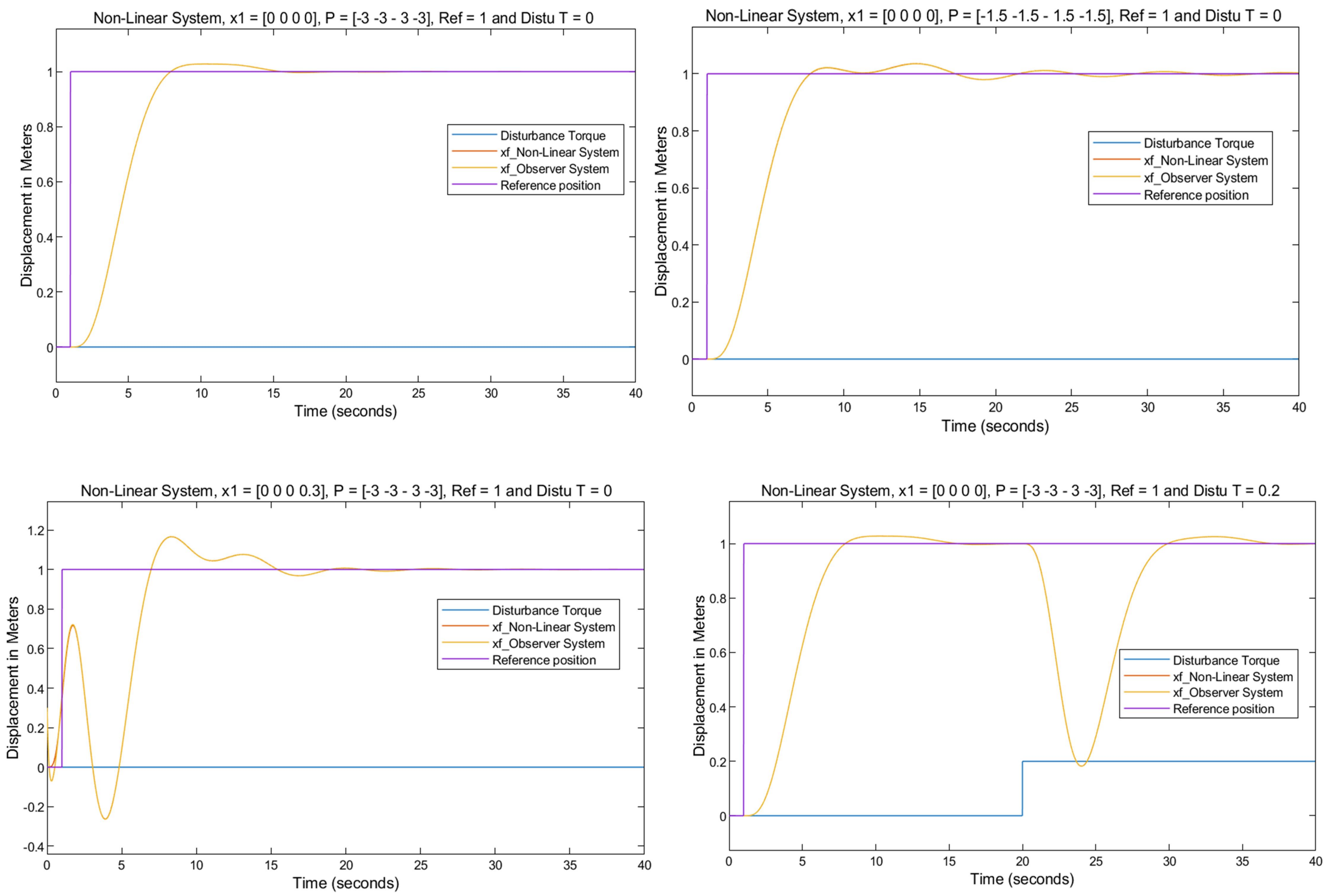


Figure 37: Outputs of State feedback Discrete Observer for Non-Linear System with Discrete PI controller

For Non-Linear System with Discrete PI controller with observer, the system response follows the reference torque but overshooting slightly before stabilizing around the reference position. When disturbance torque is applied, response gets deviates from the reference torque towards negative values. When initial condition is changed, the output of the system reaches a steady state after the initial oscillation but significant oscillation is observed around reference position before stabilizing compared to linear discrete observer system. Whereas, for a change in magnitude of Pole, the response of the system is similar to linear discrete observer system but oscillates around the reference torque.

CONCLUSION

The report provides a comprehensive study on the control of a Seesaw system, focusing on both linear and nonlinear modelling and various control strategies. Initially, the system was modelled using nonlinear dynamics, which were then linearized around a working point to simplify control analysis. The control strategies include state-feedback control, PI controllers, and state observers. The state-feedback control uses pole placement to stabilize the system, by positioning poles values, where the system achieved steady-state accuracy and compensated for disturbances, although more aggressive placements caused overshooting in nonlinear control systems. The PI controller, both in continuous and discrete forms, are used to reduce steady-state errors, which improves performance, especially when disturbances were introduced. Here, Continuous and discrete PI controllers were compared, with higher sampling rates of 800Hz.

The report also explores observer-based control, where a full-state observer was used to estimate the system's internal states. Ensuring the system could still be effectively controlled even with limited state information, by quickly correcting initial estimation errors. Different pole placements for the observer system were tested, demonstrating faster error correction with more aggressive placements. Comparing the linear and nonlinear models, it was noted that linear control strategies generally performed well on the nonlinear system, especially when tuned properly. The use of state feedback with full-state observers proved effective for controlling the system even with partial information, quickly correcting initial estimation errors and ensuring robust performance across different conditions.

Overall, the study demonstrates the effectiveness of linear control strategies applied to nonlinear systems and also the importance of careful tuning and model-based control for achieving stability and optimal performance.