Discrete Control Systems

Laboratory Report

Submitted in accomplishment of

Lab Tasks

Elemental in the degree of

M.Sc in Embedded Systems Design

By

Girish S. Kulkarni (Mr. Nr.: 38000)

Person in Charge

Prof. Dr.-Ing. Karsten Peter

Subject

DCS

Date of Submission: 29.10.2020

8°35' E

Hochschule Bremerhaven 53°32' N

An der Karlstadt 8
27568 Bremerhaven

Declaration

I declare that this written submission represents the idea in my own words. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can evoke penal action from the sources from whom proper permission has not been taken when needed.

(Mr. Girish S. Kulkarni)

Contents

Introduction

Discrete control systems, refer to the control theory of discrete-time systems which are based on discrete variational principles. 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.

Objective-

The objective of this Lab task is to model a nonlinear dynamic system such as a Seesaw system in Matlab Simulink. Linearize it and make linear state space model for a working point. Make a feedback control based on linear model and calculate feedback vector and preamplifier. Develop a PI-state-feedback control based on the linear model. Convert the continuous system to discrete. Develop a full state observer and calculate the error feedback vector. Compare and understand the effect of change in initial conditions, reference value steps, disturbance torque in linear and nonlinear systems.
The Seesaw system to be modelled is represented as below:

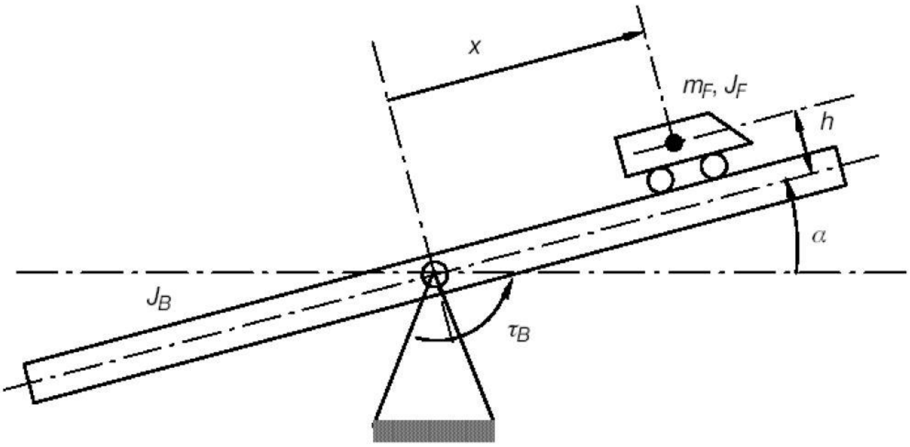


Figure 1: Seesaw System

Where,
τ_B : Drive torque of the beam tilt motor
x_F : Position of vehicle on the beam
α : Tilt angle of the beam
J_F, J_B : Momentum of inertia of the vehicle and the beam
m_F : 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.

The equation for nonlinear movement of the dynamic Seesaw system can be modelled using Euler-Lagrange formulism or D'Alembert's principle.

The Lagrange equation that describes the dynamic behavior of the system completely is given by:

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = Q_i$$

Where,

$q = \begin{bmatrix} x_F \\ \alpha \end{bmatrix}$ and Q = $\begin{bmatrix} 0 \\ \tau_B \end{bmatrix}$

Lagrange function L is:

$$L = E_{kin} - E_{pot}$$

$$= \frac{1}{2} m_F \dot{x}_F{}^2 + \frac{1}{2} \cdot m_F x_F{}^2 \dot{\alpha}^2 + \frac{1}{2} J_{Const} \dot{\alpha}^2 - m_F g x_F \sin\alpha - m_F g h \cos\alpha$$

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

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

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

The above equation completely describes the Seesaw system's nonlinear movement.

Using these two equations the nonlinear Seesaw system can be modeled in Matlab Simulink.

Matlab Code:

Seesaw_nonlinear_ini.m

```
g= 9.81;            %gravitationl pull = 9.81m/s^2
Jb= 0.5;            %Jb =  0.5kg*m^2
mf= 0.1;            %mass = 0.1kg
h= 0.1;             %height = 0.1m
Jconst= Jb+mf*h^2;
```

Figure 2: Matlab code for nonlinear Seesaw system

Simulink Model:

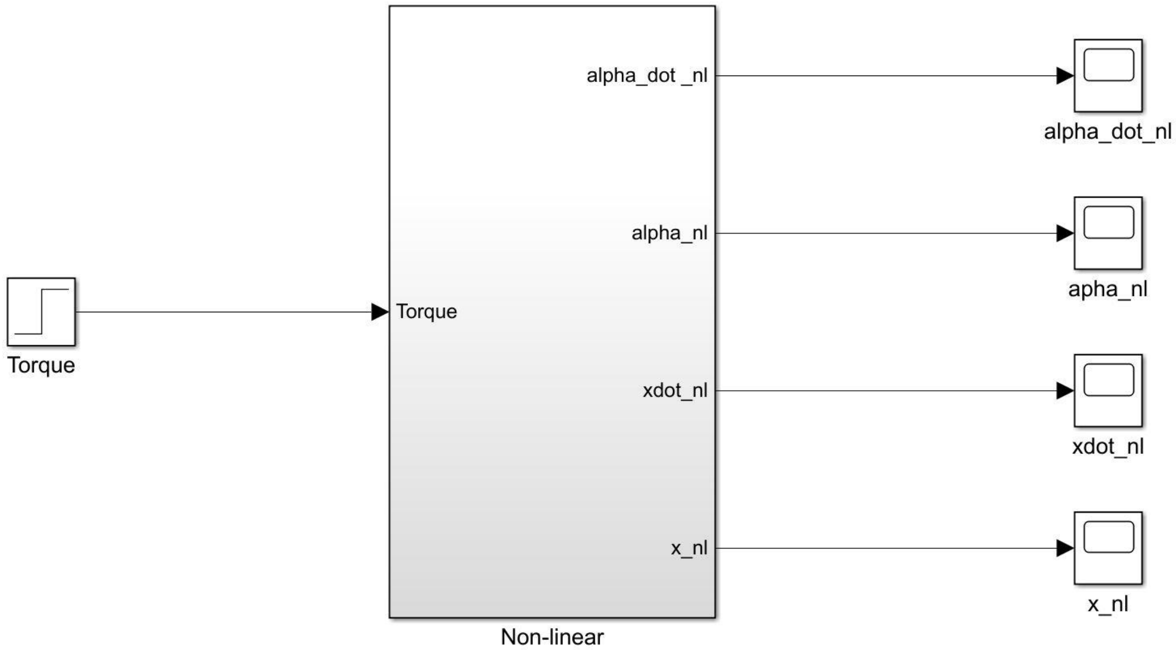


Figure 3: Simulink model for nonlinear Seesaw system

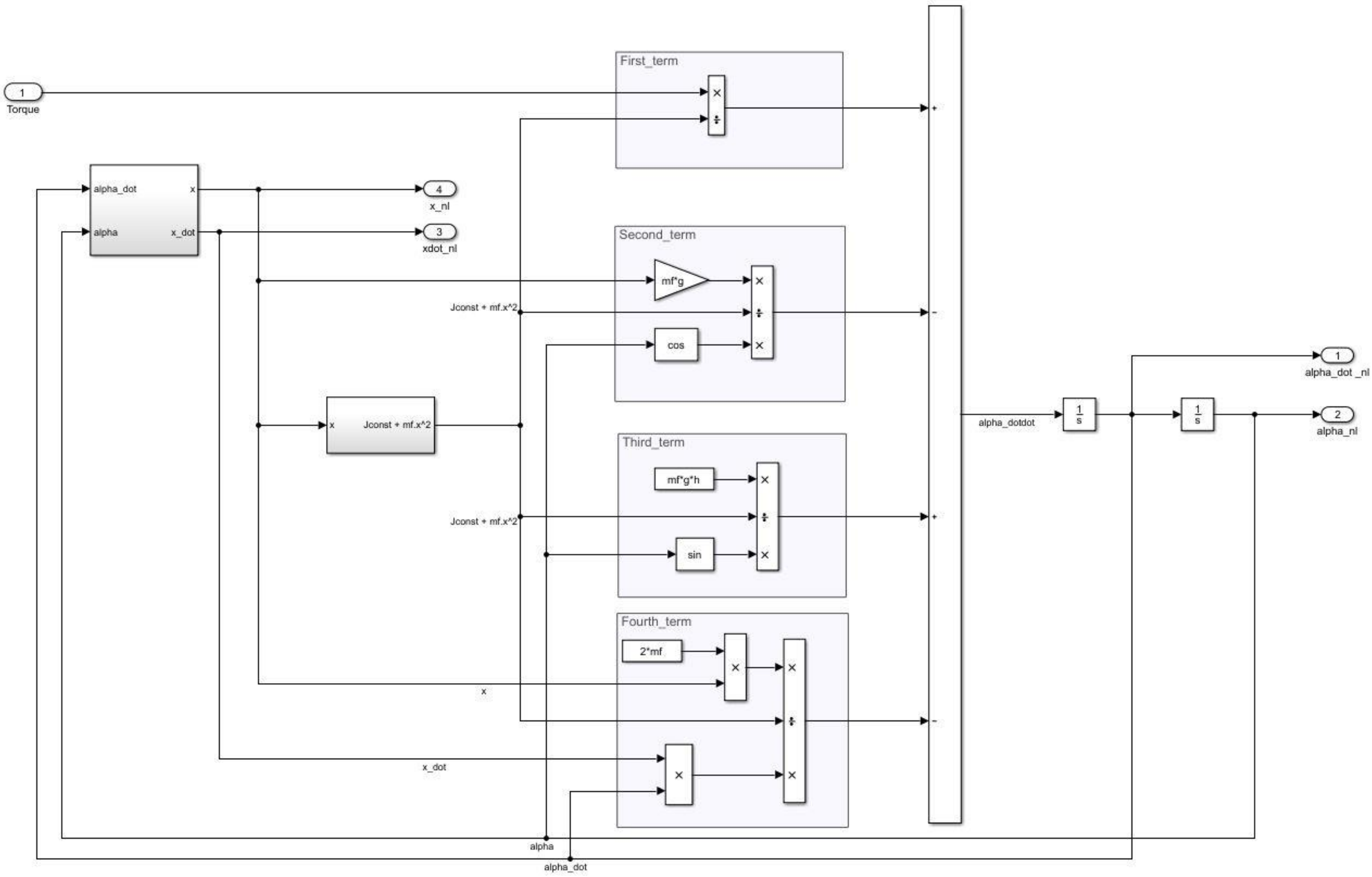


Figure 4: Nonlinear subsystem

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 method for assessing the local stability of an equilibrium point of a system of nonlinear differential equations or discrete dynamical systems.

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

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

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

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

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

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

The above equation completely describes the linearized form of Seesaw system.

These two equations can be used to represent the state space form of Seesaw system.

State-Space Rrepresentation-

A state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations or difference equations. State variables are variables whose values evolve over time in a way that depends on the values they have at any given time and on the externally imposed values of input variables. Output variables’ values depend on the values of the state variables

The LTI state space representation is given as follows:

$$\underline{\dot{x}} = \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{u}$$

$$\underline{y} = \underline{C} \cdot \underline{x} + \underline{D} \cdot \underline{u}$$

The state space form is first order differential equation. The system description of the Seesaw system consists of two second order differential equation, by making it four first order equations, it can be represented in the form of matrix as follows:

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

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

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

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

Matlab code:

```
A= [0,mf*g*h/(Jb+mf*h^2),0,-mf*g/(Jb+mf*h^2);1,0,0,0,;0,-g,0,0;0,0,1,0];
B= [1/(Jb+mf*h^2);0;0;0];
c=[0,0,0,1];
C= eye(4);
D= [0;0;0;0];
```

Figure 5: Matlab code for linear state space model of Seesaw system

Simulink model:

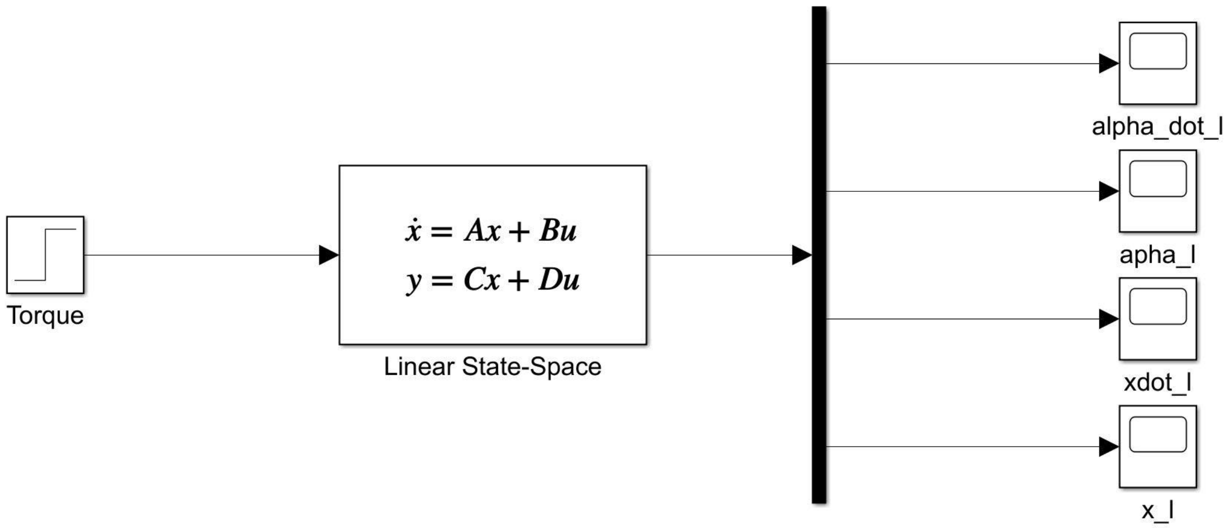


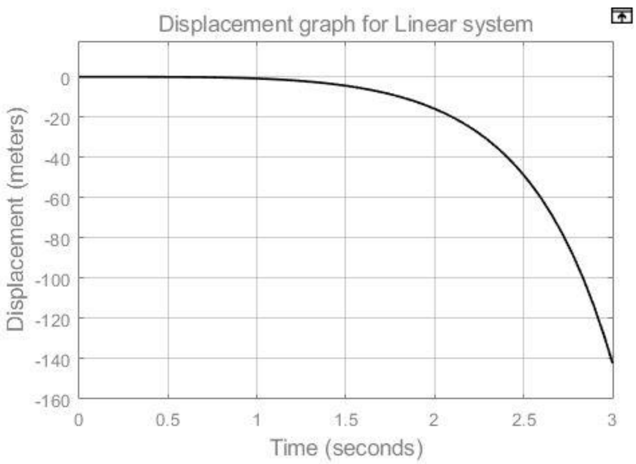
Figure 6: Simulink model for linear state space Seesaw system

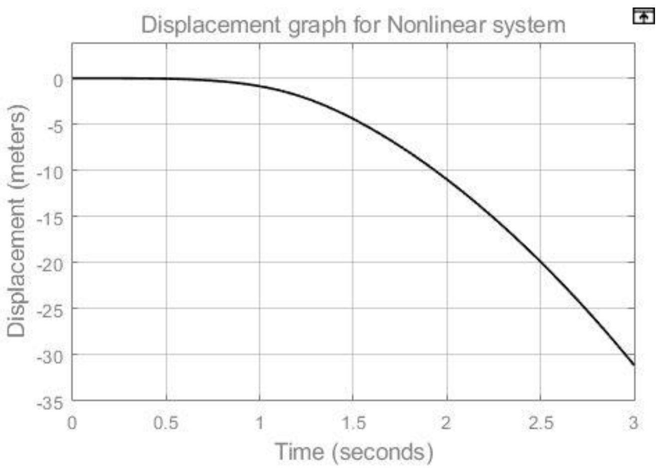
Task 3

- 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 model and nonlinear model, both the models will be arranged in parallel in Simulink and given same input torque. The output of both the models would be observed with scope.

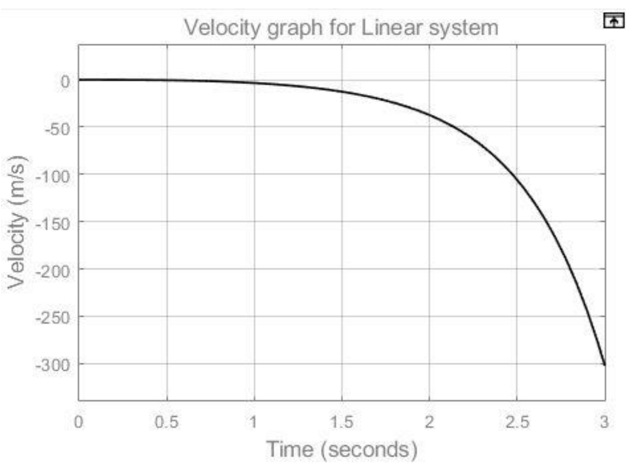
Comparison of Displacement-

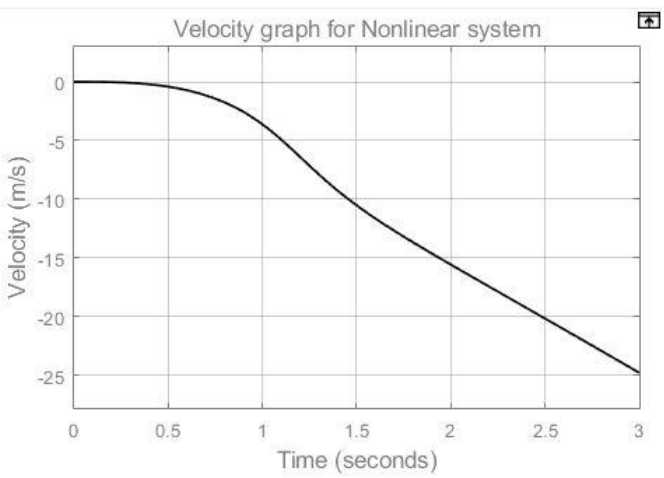




It is evident from the graphs for displacement vs time for linear and nonlinear models that for linear model, the downward curve starts a little later in time, but is steeper as compared to nonlinear model.

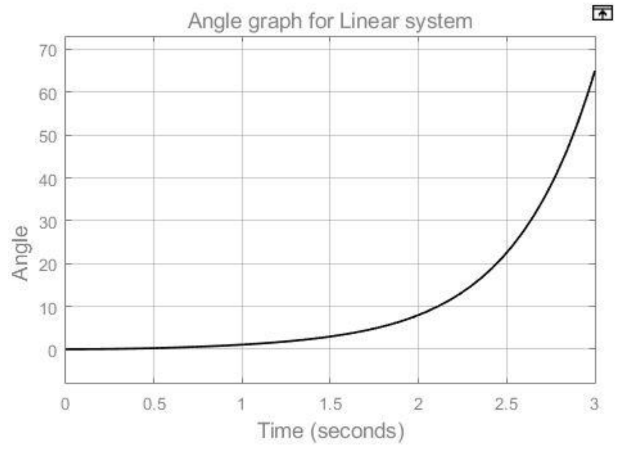
Comparison of Velocity-

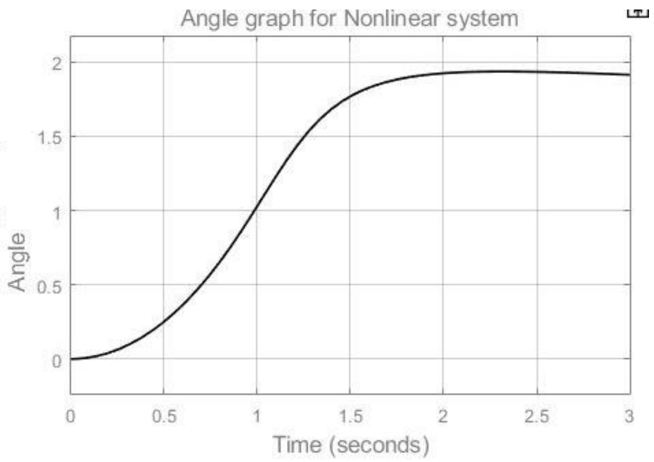




It is evident from the graphs for velocity vs time for linear and nonlinear models that for linear model, the downward curve starts a little later in time, but is steeper as compared to nonlinear model.

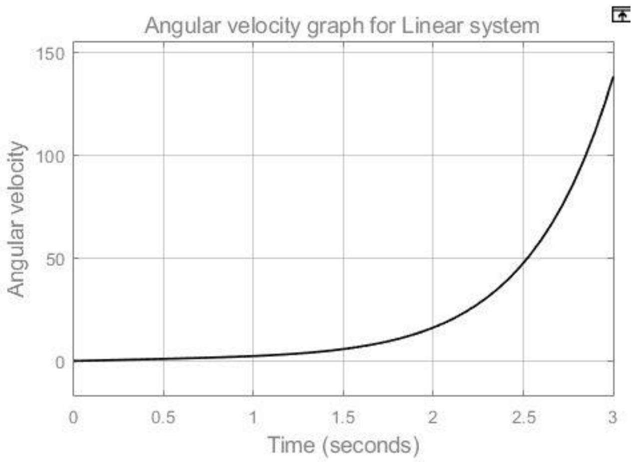
Comparison of Angle of Rotation-

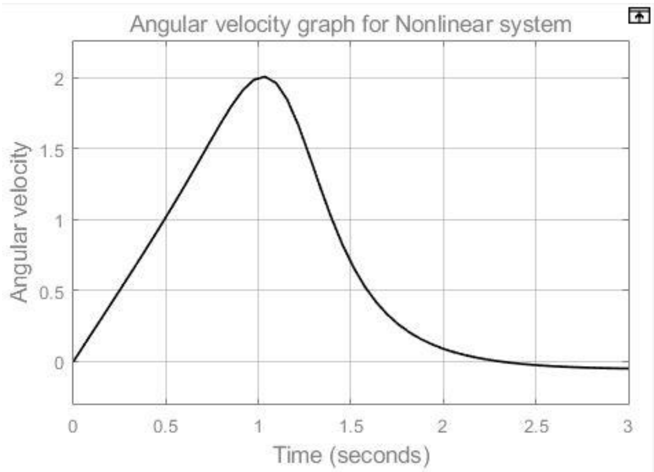




It is evident from the graphs for angular rotation vs time for linear and nonlinear models that for linear model, the curve heads towards while for nonlinear model it doesn't get beyond 90degrees.

Comparison of Angular Velocity-





It is evident from the graphs for angular velocity vs time for linear and nonlinear models that for linear model, the curve heads towards while for nonlinear model it doesn't get beyond 90 degrees and falls down again.

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

State feedback is a method employed in feedback control system theory to place the closed-loop poles of a plant in pre-determined locations in the s-plane. Placing poles is desirable because the location of the poles corresponds directly to the eigenvalues of the system, which control the characteristics of the response of the system. The system must be considered controllable in order to implement this method. It is seen from the previous task that both linear and nonlinear models are not stable. The location of the open loop poles are at-

```
>> open_loop_poles = eig(A)

open_loop_poles =

   2.1170 + 0.0000i
  -2.1170 + 0.0000i
   0.0000 + 2.0703i
   0.0000 - 2.0703i
```

The poles are not in the left half of the s-plane, that means the system is not stable. 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 also involves calculation of state feedback vector k and preamplifier p.

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

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

And formula to calculate pre amplifier to achieve steady state accuracy for state feedback control is given by :

$$p = \frac{1}{\underline{C}^T \cdot (\underline{B} * \underline{k}^T - \underline{A})^{-1} \underline{B}}$$

Matlab Code:

```
S = [B A*B A^2*B A^3*B];       %Contrability matrix
S_inv = inv(S);                %Inverse of Controlability matrix
qT=S_inv(4,:);                 %Last row of Inverse Controlability matrix
p1= -1;                        %Poles location
p2= -1;
p3= -1;
p4= -1;
alpha = poly([p1 p2 p3 p4]);   %Convert roots to polynomial
                               %Applying Ackerman Formula
kT = qT*[alpha(5)*eye(4)+ alpha(4)*A+alpha(3)*A^2 + alpha(2)*A^3 + A^4];
Ct = C(4,:);
Pre_amplifier = 1/(Ct*inv(B*kT-A)*B);
```

Figure 6: Matlab code for state feedback vector and preamplifier

Simulink model:

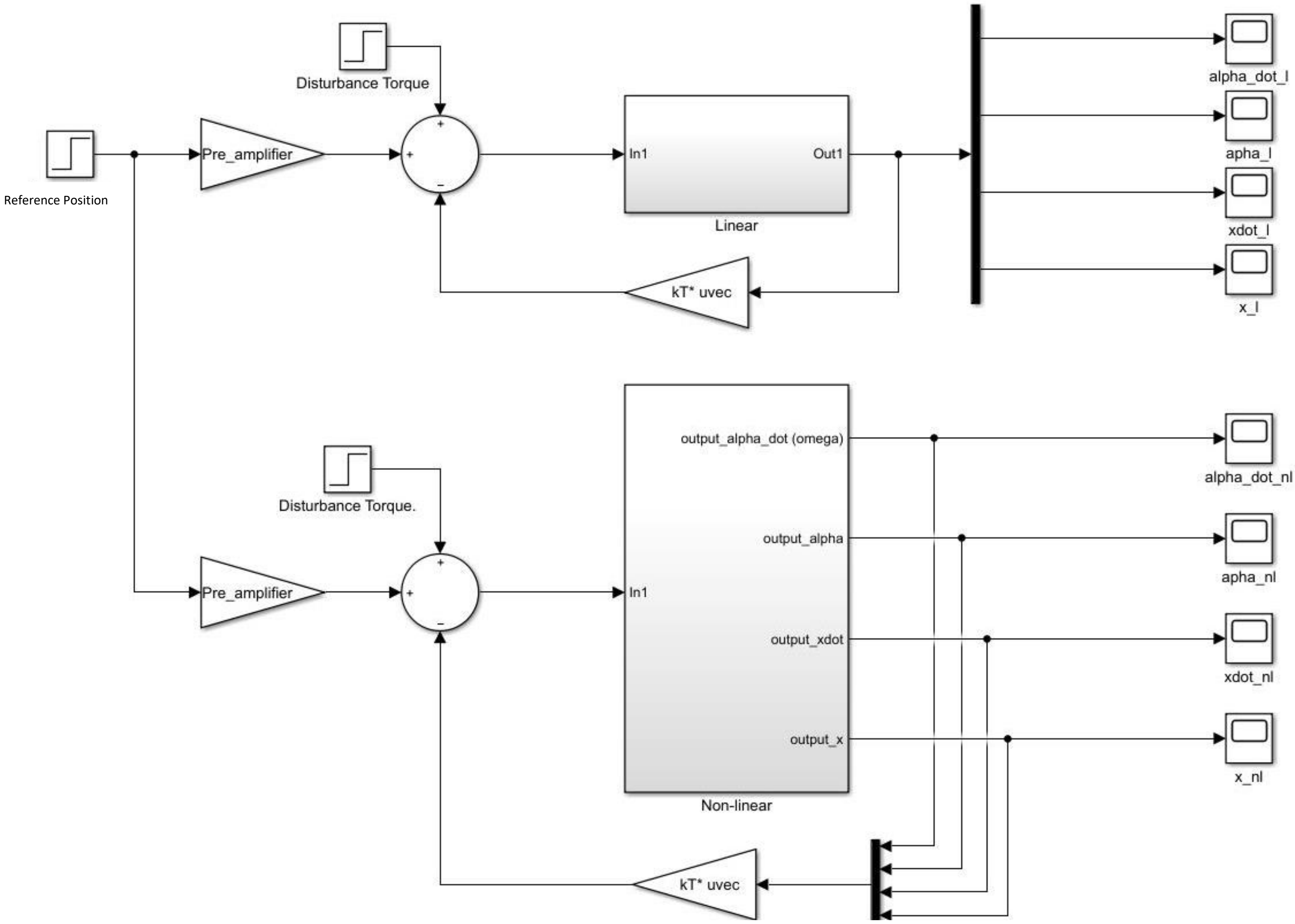
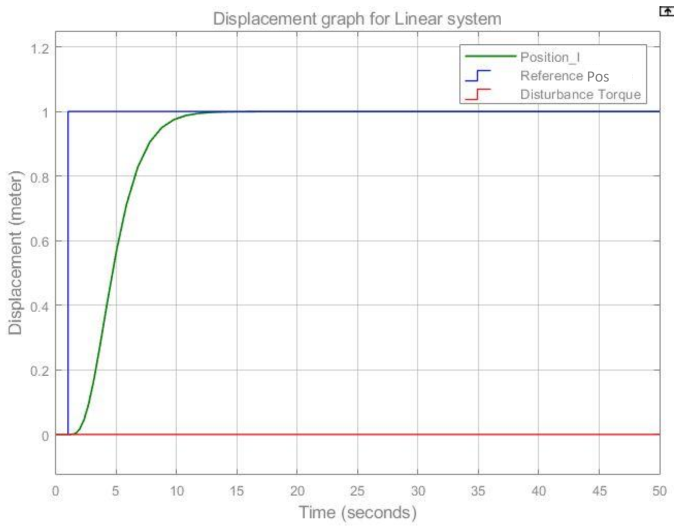
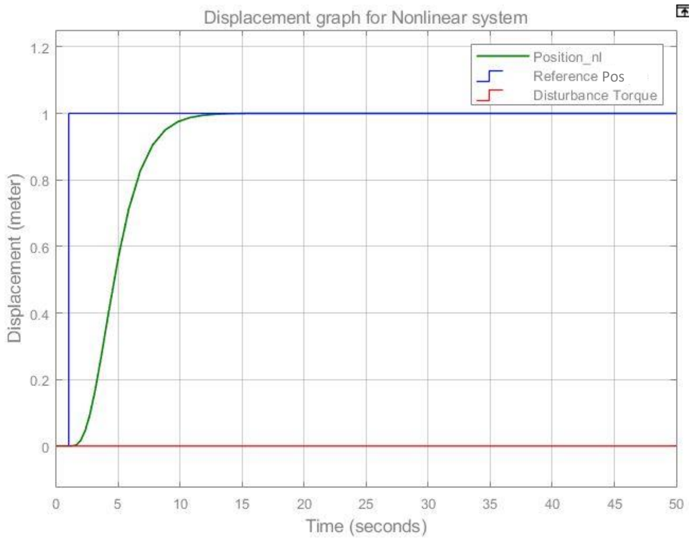


Figure 7: Simulink model for state feedback vector and preamplifier

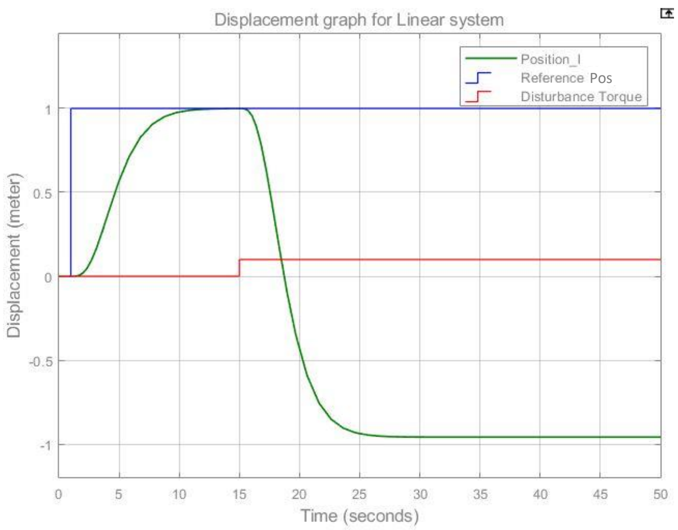
Comparison of displacement (without disturbance torque)-

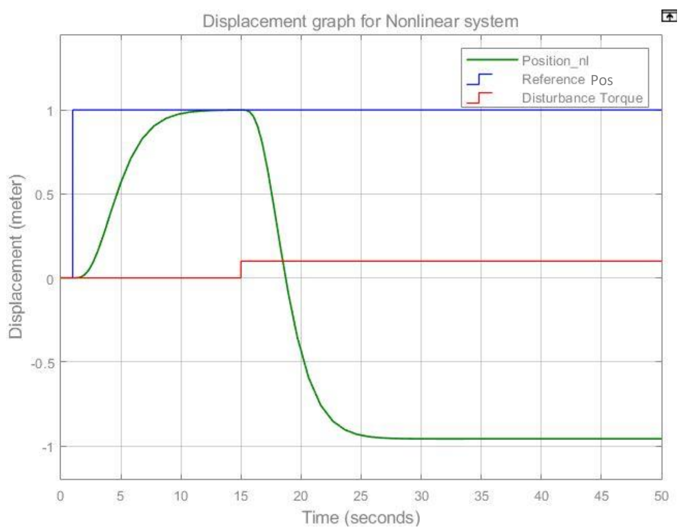




When no disturbance torque is applied, the system stability is achieved and also it follows the reference torque i.e it reaches steady state accuracy with the calculated state feedback vector and preamplifier gain. The system response is same for both linear and nonlinear system.

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 the distortion torque. Even though the system is stable, it does not follow the reference torque for both linear and nonlinear systems.

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.

Using state feedback controller and preamplifier gain makes the system state and the response reaches the steady state accuracy for very small manipulation range that is when the reference torque is less and also there is no disturbance torque at the input.

With PI controller an additional integrator is added along with the Proportional controller. Because of integrator, an additional state is introduced in the system. Thus the order of the augmented system is increased by 1.

The P is interpreted as an element of the state-feedback vector since it feeds back the state z. This leads to the following system description of the augmented system in state-space:

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

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

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

Matlab Code:

```
Api = [A [0;0;0;0]; -Ct 0];
Bpi = [B; 0];
Cpi = [0 0 0 1 0];
Dpi = [0; 0; 0; 0; 0];
Spi = [Bpi Api*Bpi Api^2*Bpi Api^3*Bpi Api^4*Bpi]; %Controlability matrix
Spi_inv = inv(Spi);                 %Inverse of Controlability matrix
qT_pi=Spi_inv(5,1:5);               %Last rot if Inverse Controlability matrix
p1= -1;                             %Poles location
p2= -1;
p3= -1;
p4= -1;
p5= -1;
alpha_pi = poly([p1 p2 p3 p4 p5]); %Convert roots to polynomial
                                    %Applying Ackerman Formula
kpi = qT_pi*[alpha_pi(6)*eye(5) + alpha_pi(5)*Api + alpha_pi(4)*Api^2 +
    alpha_pi(3)*Api^3 + alpha_pi(2)*Api^4 + Api^5];
PI = -kpi(5);                       %Integral gain
kT_pi = kpi(1:4)-Pre_amplifier*Ct; %State feedback value
```

Figure 8: Matlab code for PI controller

Simulink Model:

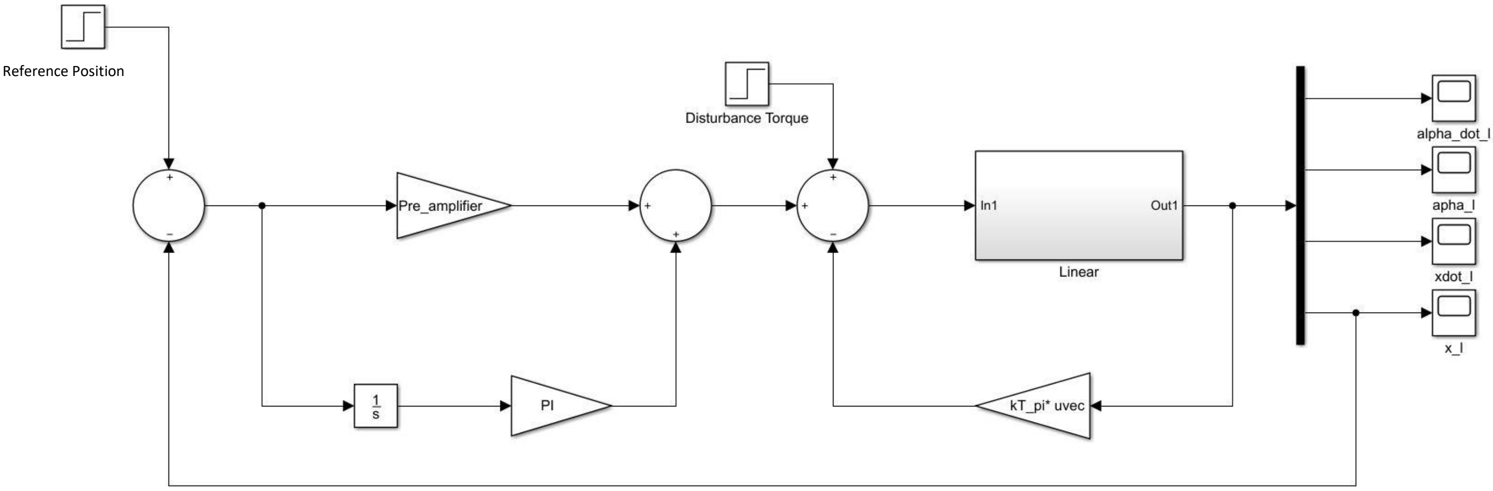
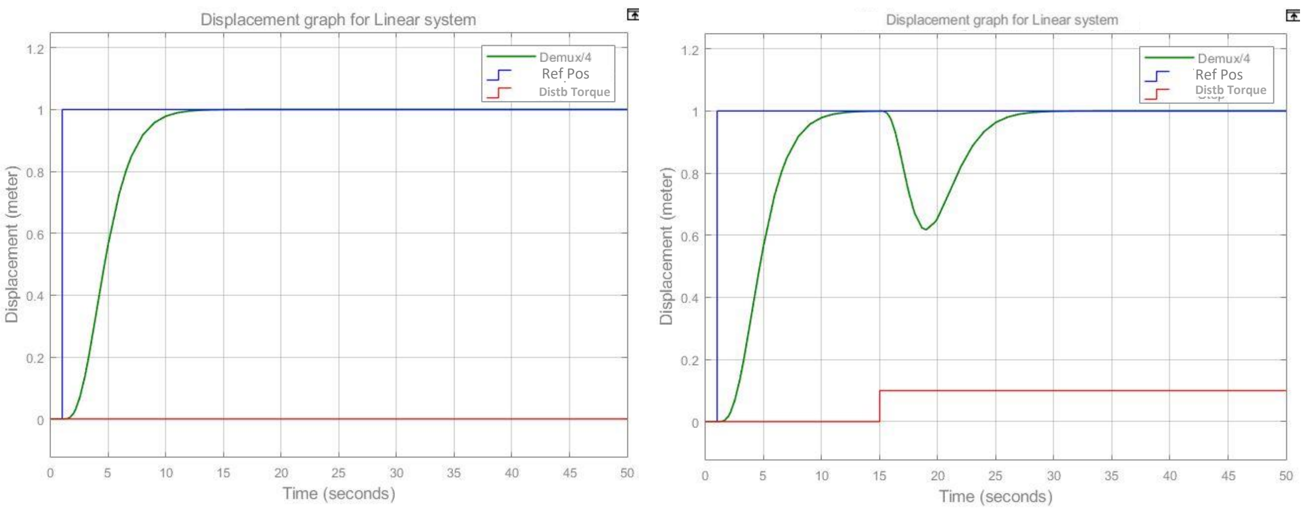
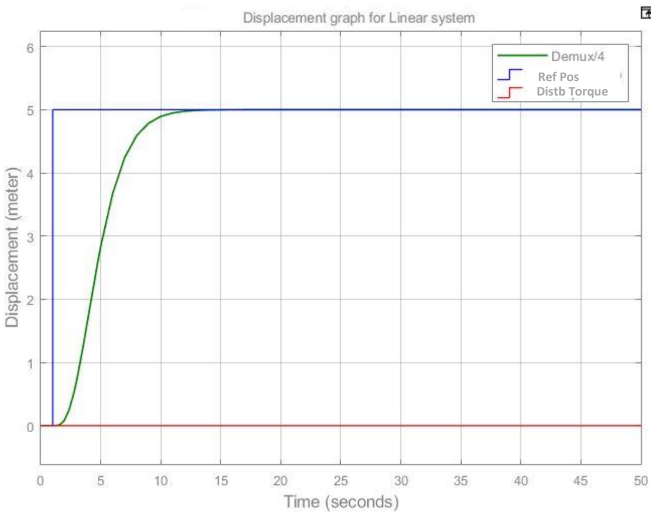


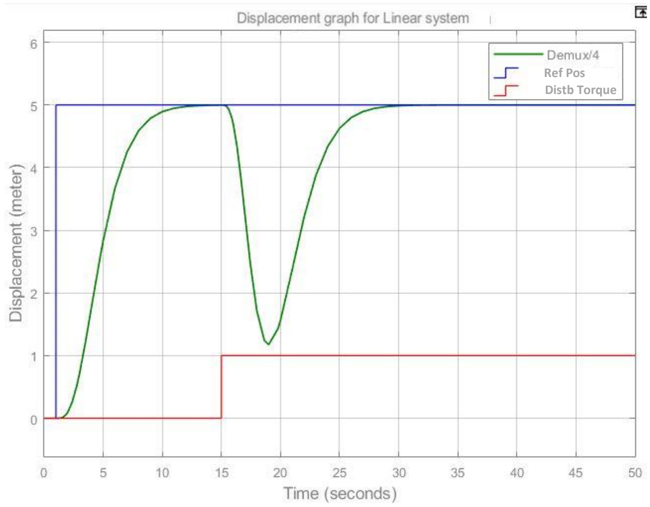
Figure 9: Simulink model of PI controller for Linear system

Observing displacement graphs-



It is evident from the above graphs that, when there is no disturbance, the response of the system follows the reference torque and we get 'S' curve. When disturbance of 0,1 is introduced at 15 sec, then the response of the system also gets disturbed, and follows the reference torque again.





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 15 sec, then the response of the system also gets disturbed, and follows the reference torque again.

Thus, with PI controller, the augmented system can handle different disturbances and torque steps.

Task 6

- Make the PI-part time discrete with a sample rate 8 kHz and test the linear PI-state-feedback on the nonlinear seesaw system.

Discrete time control system is control system in which one or more variable can change only at discrete instants of time. The time interval between two discrete instants is taken to be sufficiently short that the data for the time between them can be approximate.

To discretize the PI-part with the sampling rate of 8 kHz, the desired step response poles considered to be in the left half s-plane needs to be mapped to points within unit circle in z-plane in order for the time-discrete system to be stable. The system needs to be transferred to time discrete state-space. Suitable sampling time and default discretization method of ZOH is selected. The integrator in the time continuous system should be replaced by unit delay in the discrete time system.

Matlab code:

```
fs = 8000;                          %Sampling rate (8kHz)
Ts= 125e-4;                         %Sampling time
sys = ss(Api, Bpi, Cpi, 0);
sys_d = c2d(sys,Ts);                %Continuous to Discrete state space
[Ad, Bd, Cd, Dd]= ssdata(sys_d);
Spi_d = [Bd Ad*Bd Ad^2*Bd Ad^3*Bd Ad^4*Bd]; %Controlability matrix
Spi_inv_d = inv(Spi_d);              %Inverse of Controlability matrix
qT_pi_d=Spi_inv_d(5,1:5);            %Last row of Inverse Controlability
sys_pi =tf(1,poly([-1,-1,-1,-1,-1]));
desys_pi = c2d(sys_pi, Ts);
[n_alpha,d_alpha]=tfdata(desys_pi,'v'); %Geting Transfer function data
                                     %Applying Ackerman's formula
kpi_d = qT_pi_d*(d_alpha(6)*eye(5)+d_alpha(5)*Ad+d_alpha(4)*Ad^2+...
                 d_alpha(3)*Ad^3+d_alpha(2)*Ad^4+Ad^5);
PI_d = -kpi_d(5)*Ts;                 %Integral gain
kT_pi_d = kpi_d(1:4)-Pre_amplifier*Ct;  %State feedback value
```

Figure 10: Matlab code for discrete PI controller

Simulink Model:

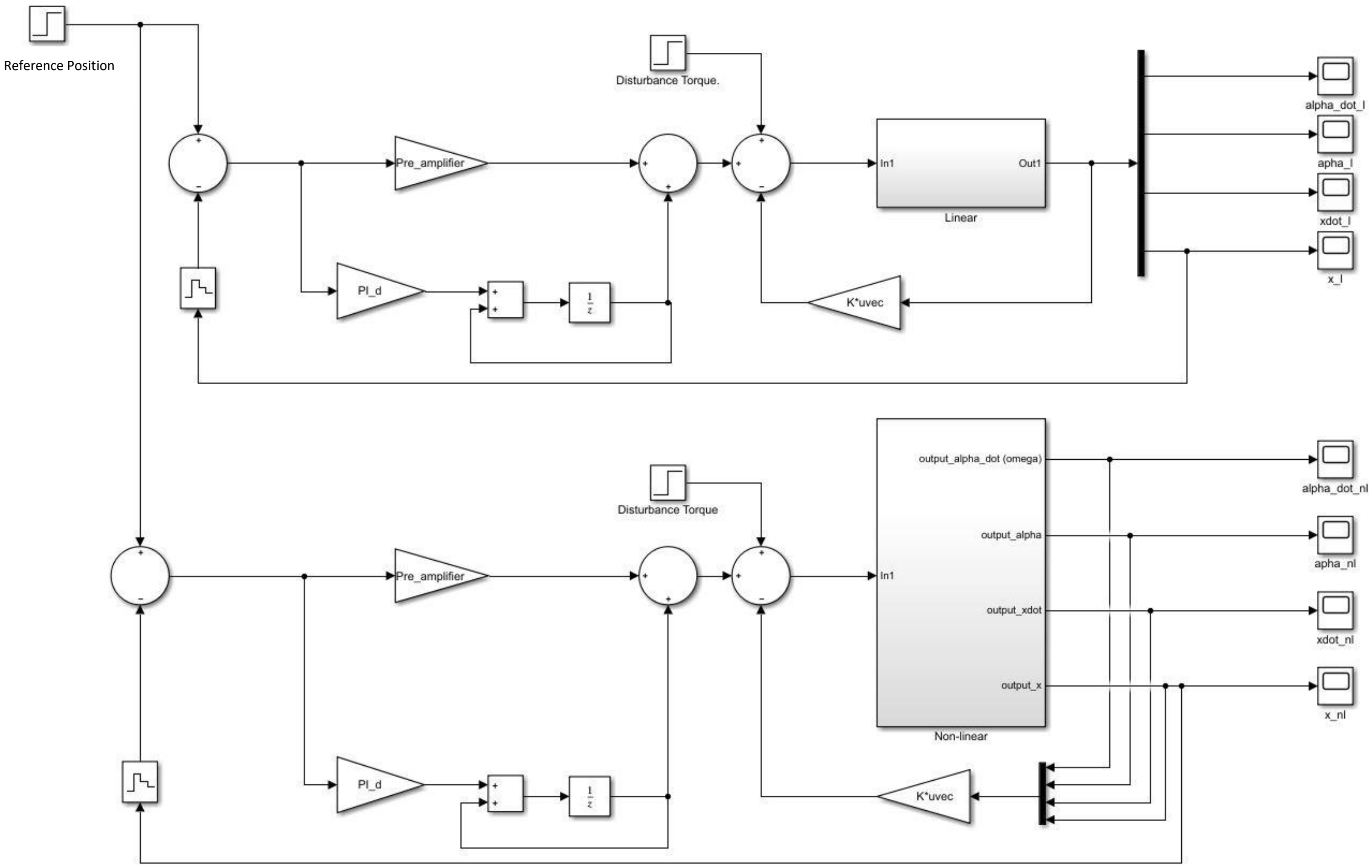
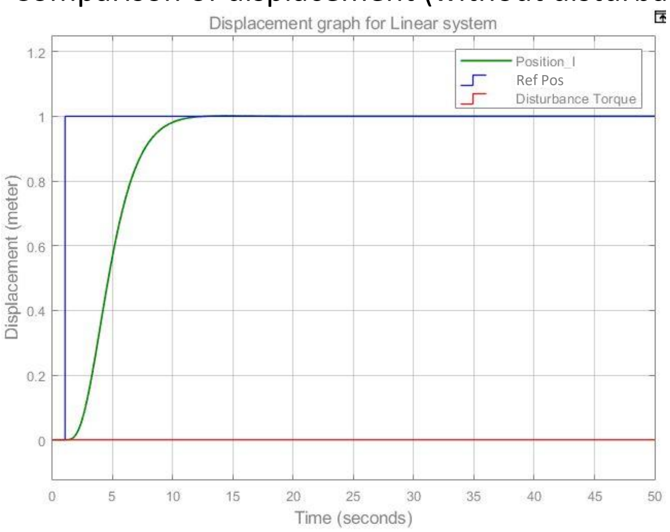
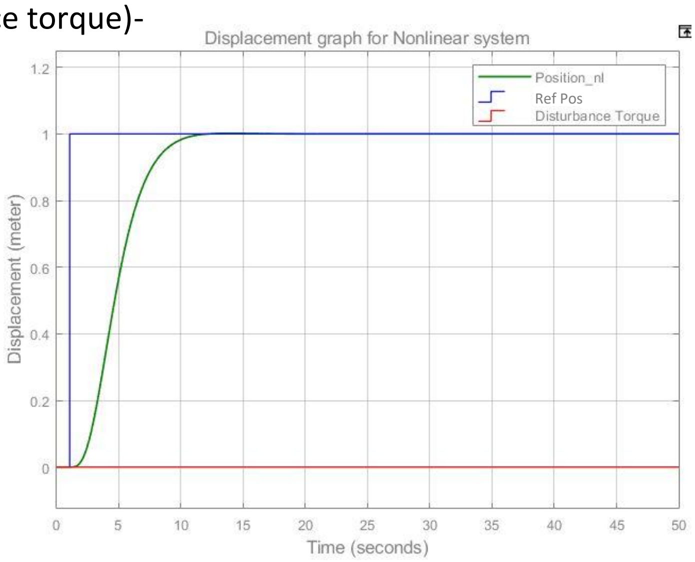


Figure 11: Simulink model for discrete PI controller

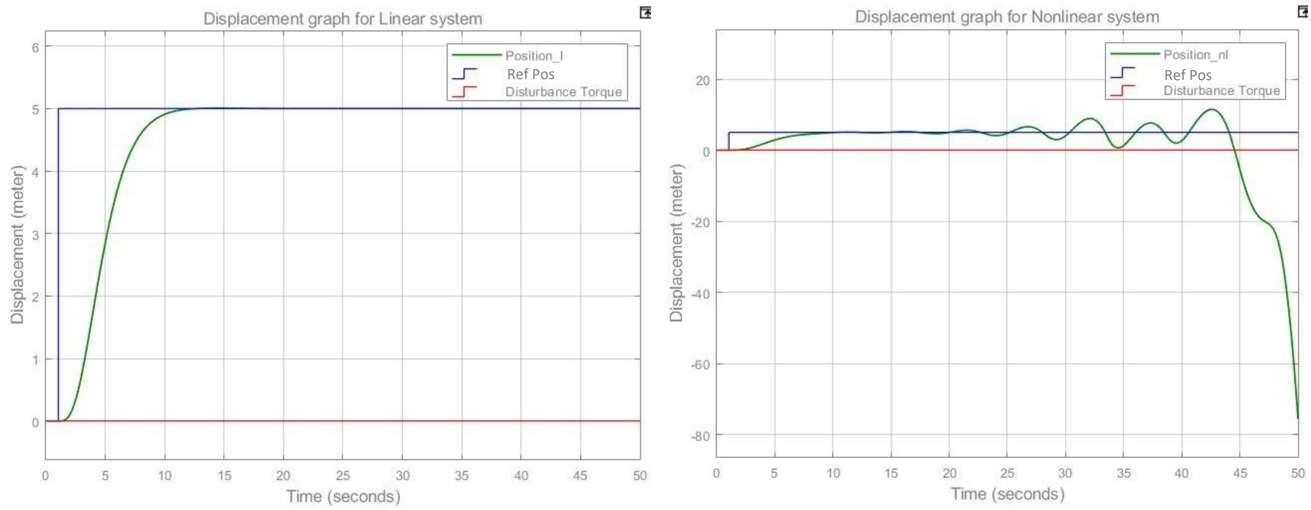
Comparison of displacement (without disturbance torque)-





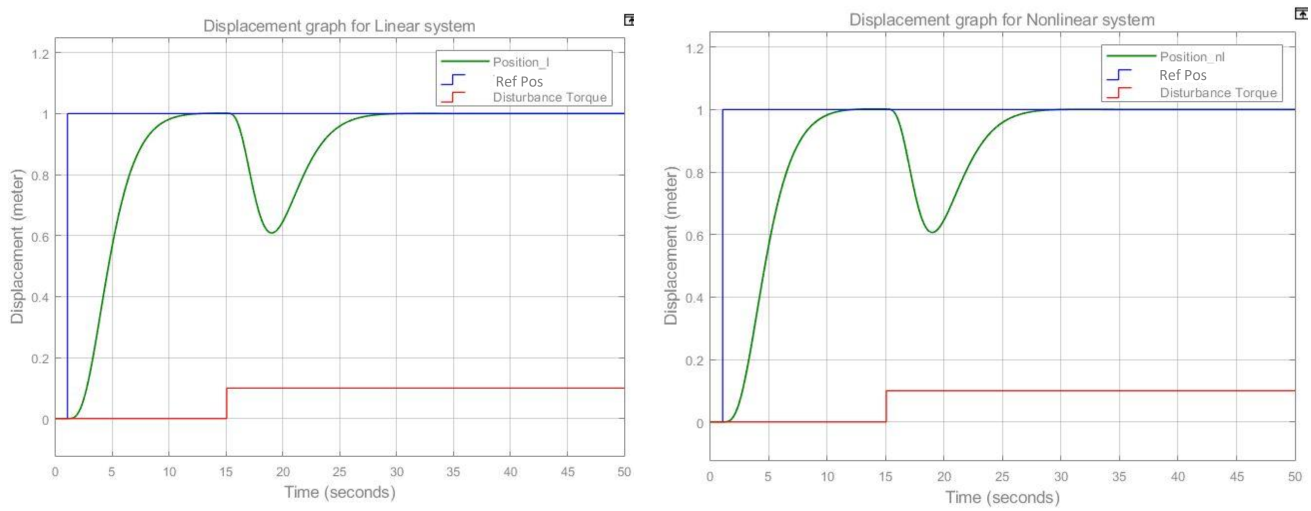
It is evident from the graphs that when there for discrete models when no disturbance torque is applied the response of the system follows the torque and we get ‘S’ curve.

Comparison of displacement (for different reference torque value)-



It is evident from the graph that, for linear system even if the torque reference value is increased, the response is stable and follows the torque. But for nonlinear system, the system becomes unstable with increase in reference torque.

Comparison of displacement (for application of disturbance)-



It is evident from the above graphs that, when disturbance torque of 0.1 at 15 sec is applied, the response of the system follows the reference torque and we get 'S' curve for both linear and nonlinear systems.

Thus, for discrete system with PI controller, the augmented system can handle different disturbances and torque steps but nonlinear systems becomes unstable for higher magnitude of torque steps.

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.

An observer is a dynamic system that is used to estimate the state of a system or some of the states of a system. A full-state observer is used to estimate all the states of the system. The observer can be designed as either a continuous-time system or a discrete-time system. Sometimes a limited measurability exist in technical systems. There may be several reasons: Sensors are sometimes not easy to install or there is not a feasible physical sensor available. Sometimes sensors are susceptible for failures and reduce the process reliability and operations like to reduce its number. Another reason could be that they are too expensive from commercial perspective. Whenever it is not possible or not wanted to measure states by sensor, a state observer may be a feasible solution.

State observers take advantage of a parallel model. Observer uses same state space representation as the original system. However, the output of both the systems is different because of different initial conditions. To reduce the error, error feedback vector is calculated using Ackerman's formula which reduces observation error. To make observer faster than the controller, observer poles are placed at -3.

Linear time continuous model-
Matlab code:

```
x1 = [1,1,1,1];                       %Initial condition
S_obs= [Ct; Ct*A; Ct*A^2; Ct*A^3];    %Controlability matrix
S_obs_inv = inv(S_obs);               %Inverse of Controlability matrix
qT_obs = S_obs_inv(1:4,4);            %Last row of Inverse Controlability matr
p1= -3;                               %Poles location
p2= -3;
p3= -3;
p4= -3;
alpha_obs = poly([p1,p2,p3,p4]);      %Convert roots to polynomial
                                      %Applying Ackerman Formula
k_obs = (alpha_obs(5)*eye(4) + alpha_obs(4)*A + alpha_obs(3)*A^2 ...
            + alpha_obs(2)*A^3 + alpha_obs(1)*A^4)* qT_obs;
h_obs = k_obs;                        %Output error feedback gain
h_trans = transpose(h_obs);
```

Figure 12: Matlab code for linear time continuous observer

Simulink model:

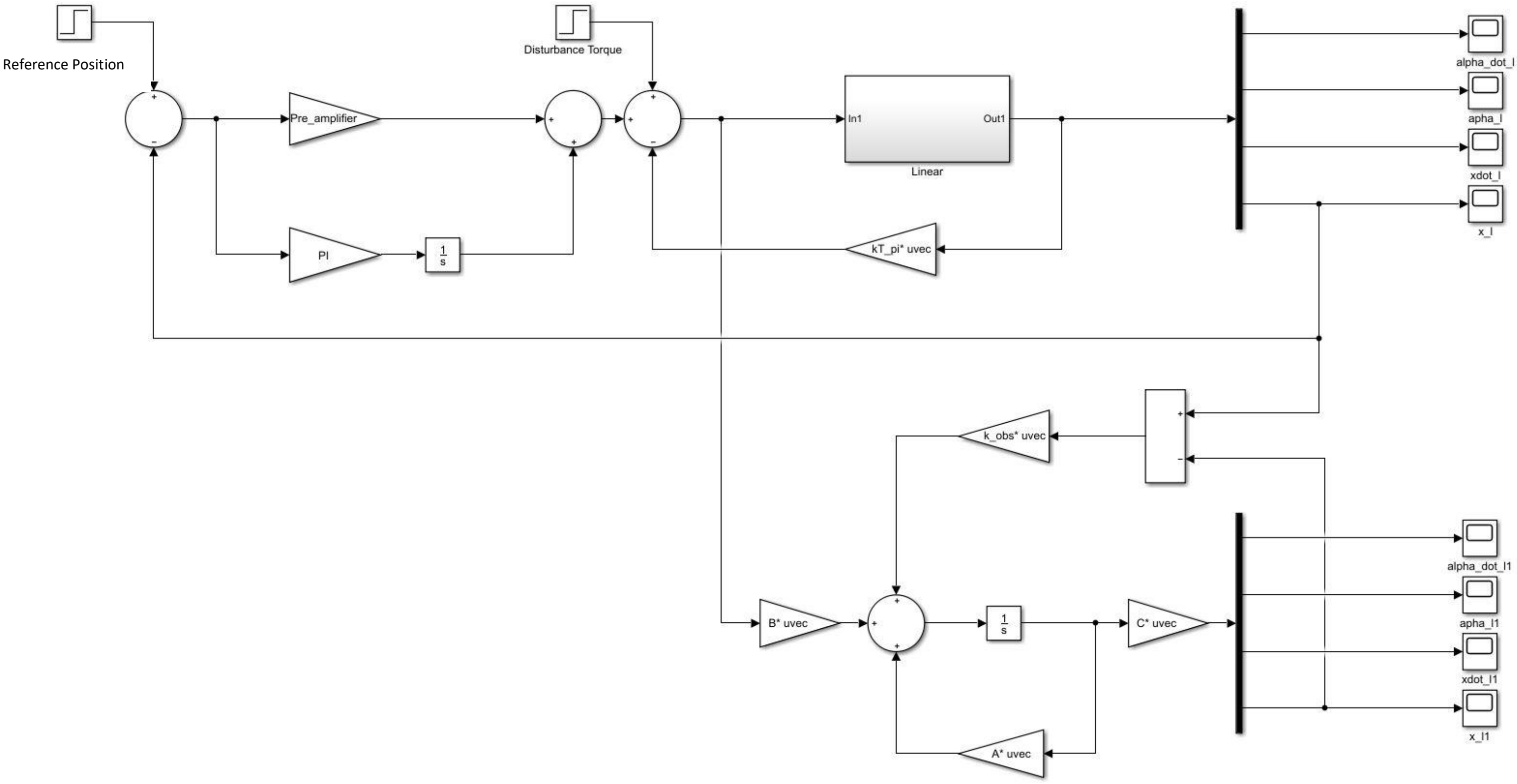
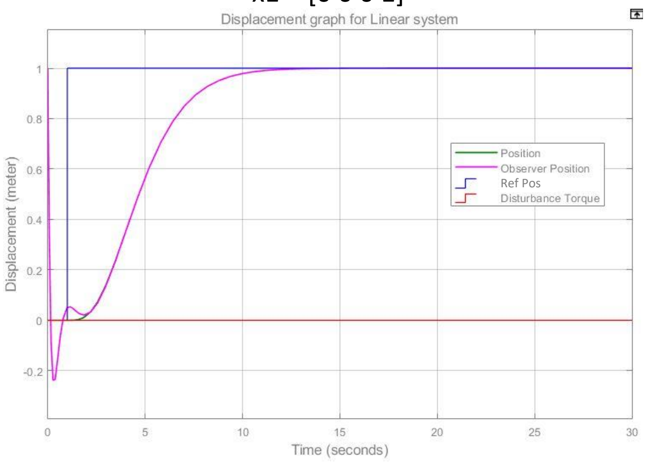


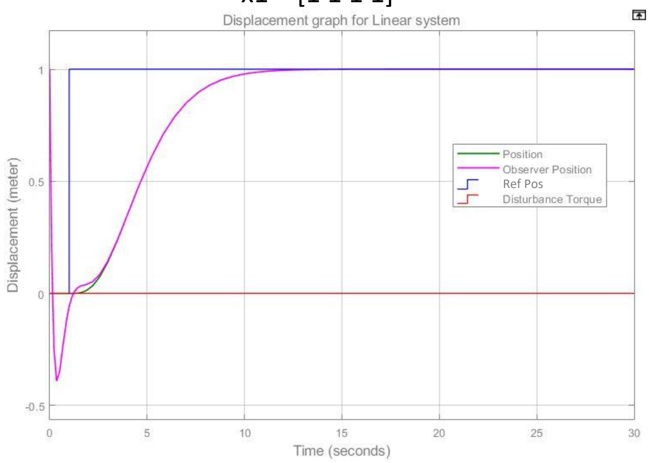
Figure 13: Simulink model for linear time continuous observer

Comparison of displacement (for different initial conditions)

x1 = [0 0 0 1]



x1 = [1 1 1 1]



It is evident from the above graphs that, the response of the system follows the reference torque and we get ‘S’ curve for both, observer and original system. Initially, due to different initial conditions for observer and original system, a spike is observed at observer position.

Linear time discrete model-
Matlab code:

```
fs = 8000;                        %Sampling rate (8kHz)
Ts = 125e-4;                      %Sampling time
sys_obs = ss(A, B, C, D);         %Observer statespace
sys_obs_d = c2d(sys_obs,Ts);      %Continuous to Discrete state space
[A_obs_d, B_obs_d, C_obs_d, D_obs_d]=ssdata(sys_obs_d);
cT_obs_d = C_obs_d(4,1:4);
p1= -3;                           %Poles location
p2= -3;
p3= -3;
p4= -3;
desys_obs = tf(1,poly([p1,p2,p3,p4]));
desys_obs_d = c2d(desys_obs,Ts);    %Continuous to Discrete state space
[num_obs,den_obs]=tfdata(desys_obs_d,'v');
pole_place = roots(den_obs);      %Polynomial for desired poles
k_obs_d = acker(A_obs_d.', cT_obs_d.',pole_place.');%Ackerman Formula
h_obs_d = k_obs_d.';              %Output error feedback gain
kT_obs_d = h_obs_d - Pre_amplifier*C_obs_d(4,4:1);
```

Figure 14: Matlab code for linear time discrete observer

Simulink model:

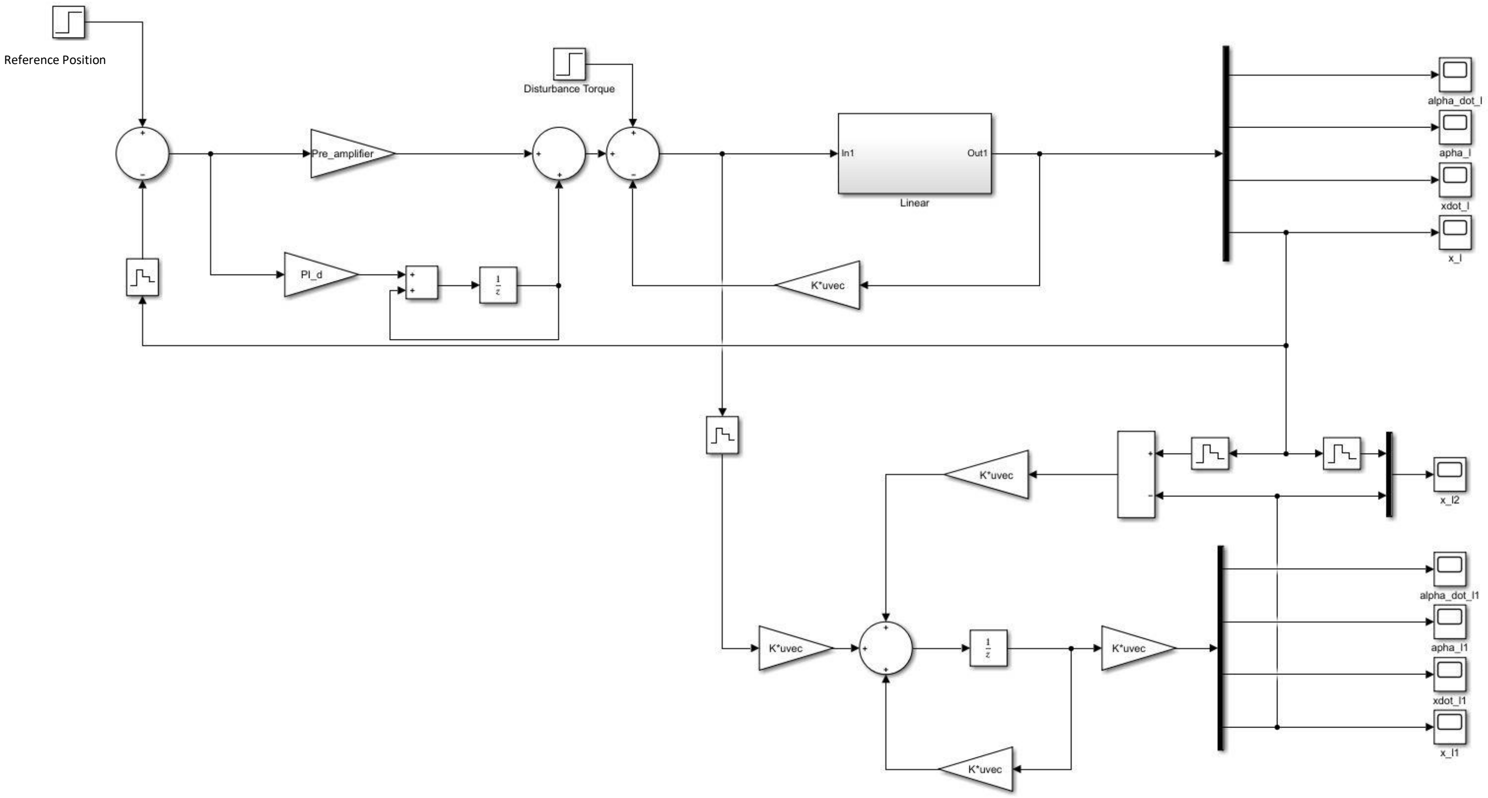
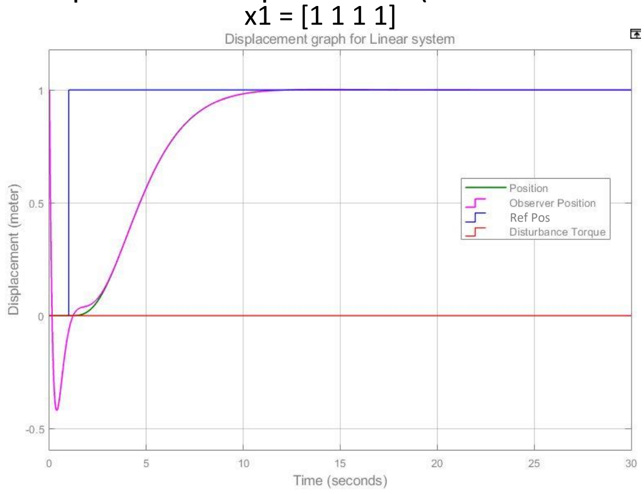
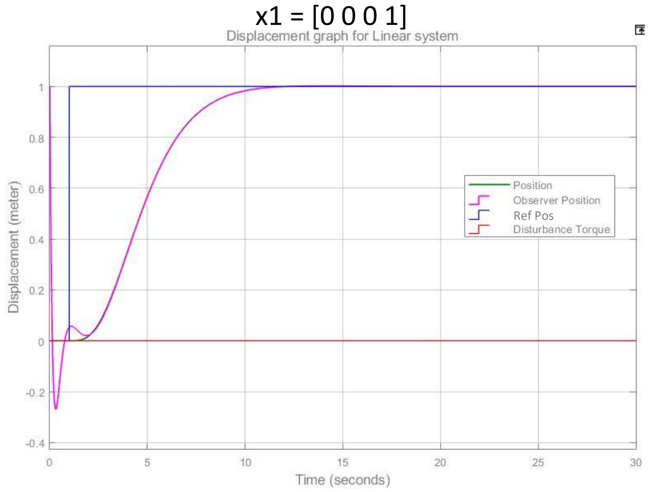


Figure 15: Simulink model for linear time discrete observer

Comparison of displacement (for different initial conditions)





It is evident from the above graphs that, the response of the system follows the reference torque and we get 'S' curve for both, observer and original system. Initially, due to different initial conditions for observer and original system, a spike is observed at observer position.

Nonlinear time continuous model-

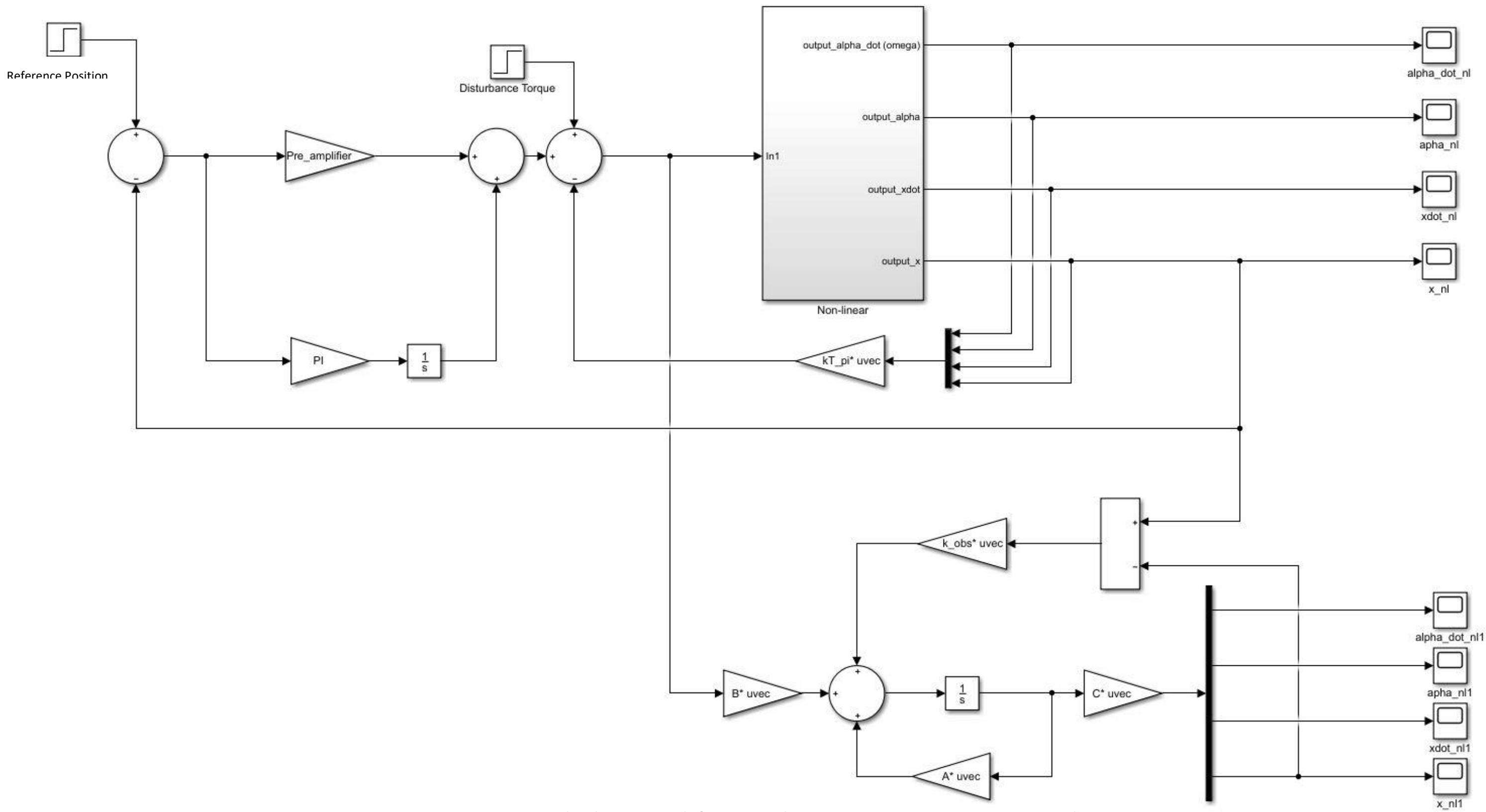
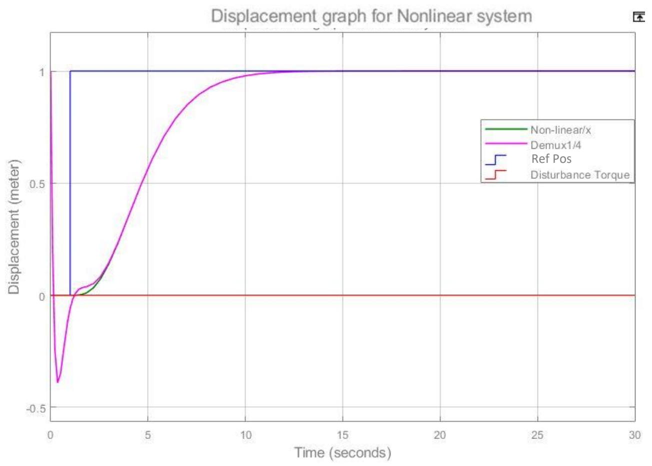


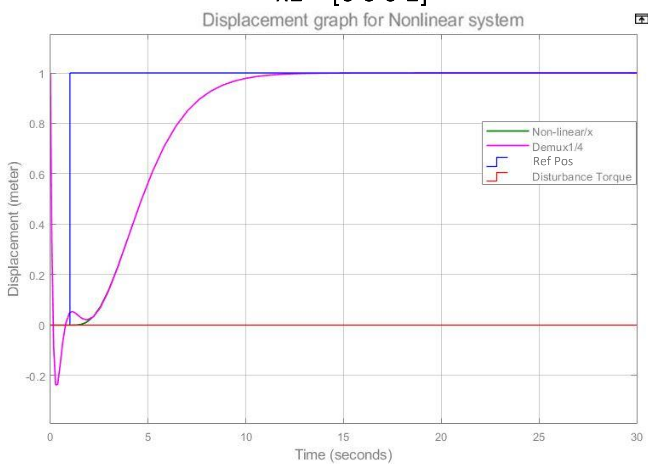
Figure 16: Simulink model for nonlinear time continuous observer

Comparison of displacement (for different initial conditions)

x1 = [1 1 1 1]



x1 = [0 0 0 1]



Similar to time continuous linear model, the response of time continuous nonlinear model also follows the reference torque with initial spike at observer side.

Nonlinear time discrete model-
Simulink model:

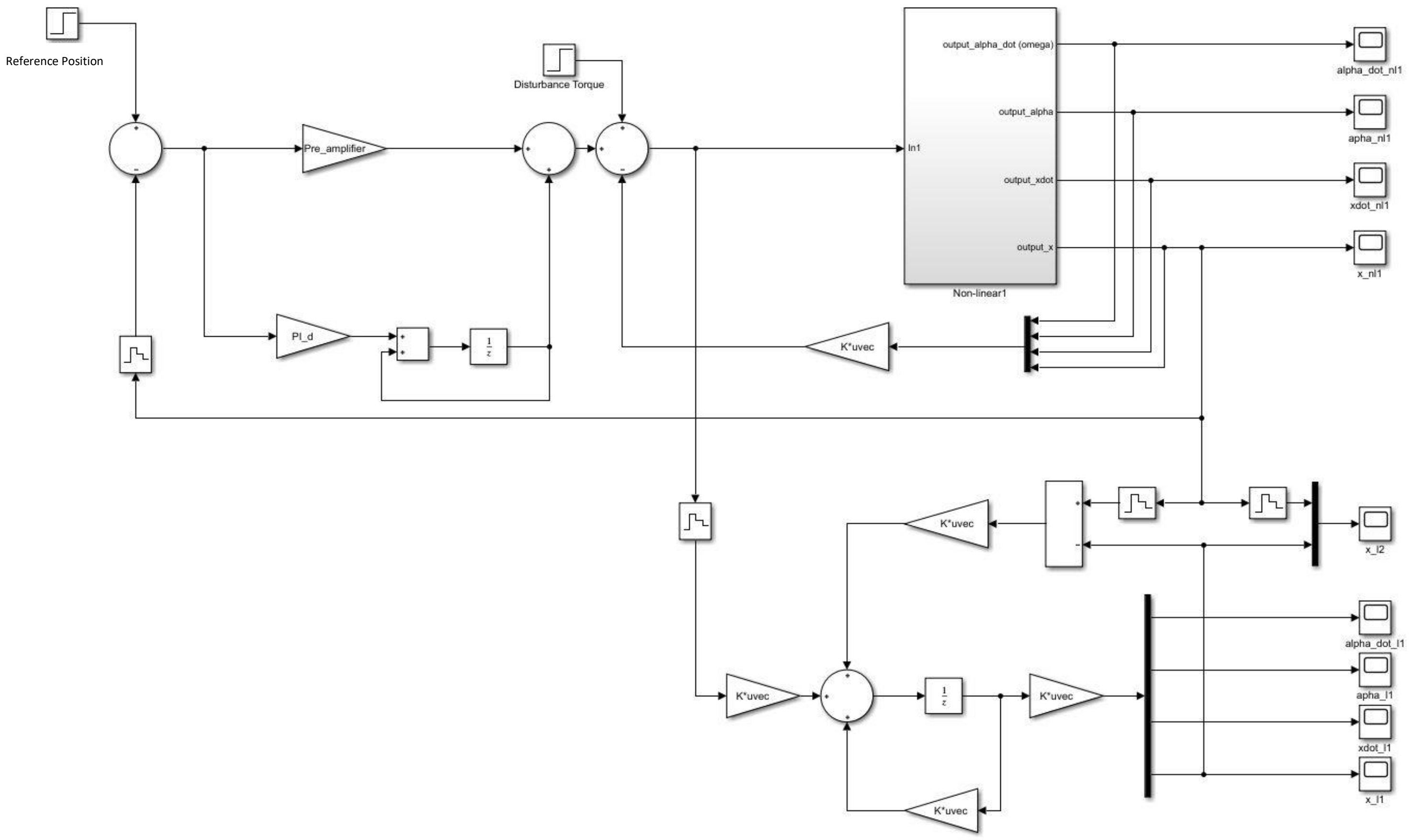
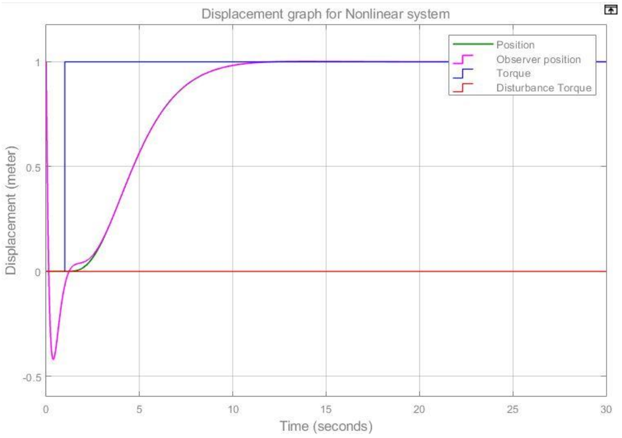


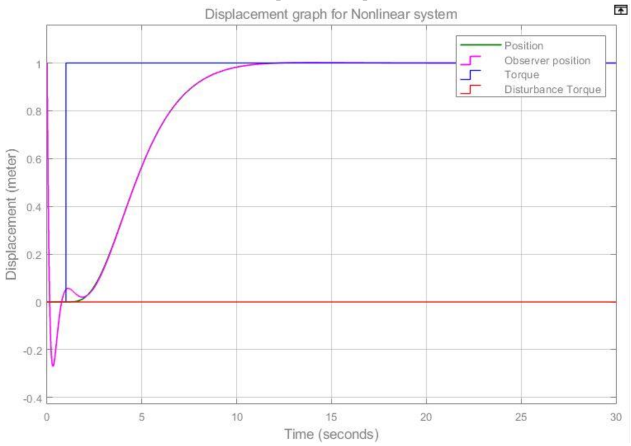
Figure 16: Simulink model for nonlinear time discrete observer

Comparison of displacement (for different initial conditions)

x1 = [1 1 1 1]



x1 = [0 0 0 1]



Similar to time discrete linear model, the response of time nonlinear model also follows the reference torque with initial spike at observer side.

Task 8

- Use the observed states from 7. for the controls 4. – 6. and test them on the linear and non-linear system for different initial conditions and reference value steps and disturbance torque steps.

Linear State feedback with preamplifier along with observer (Task 4)-

Simulink model:

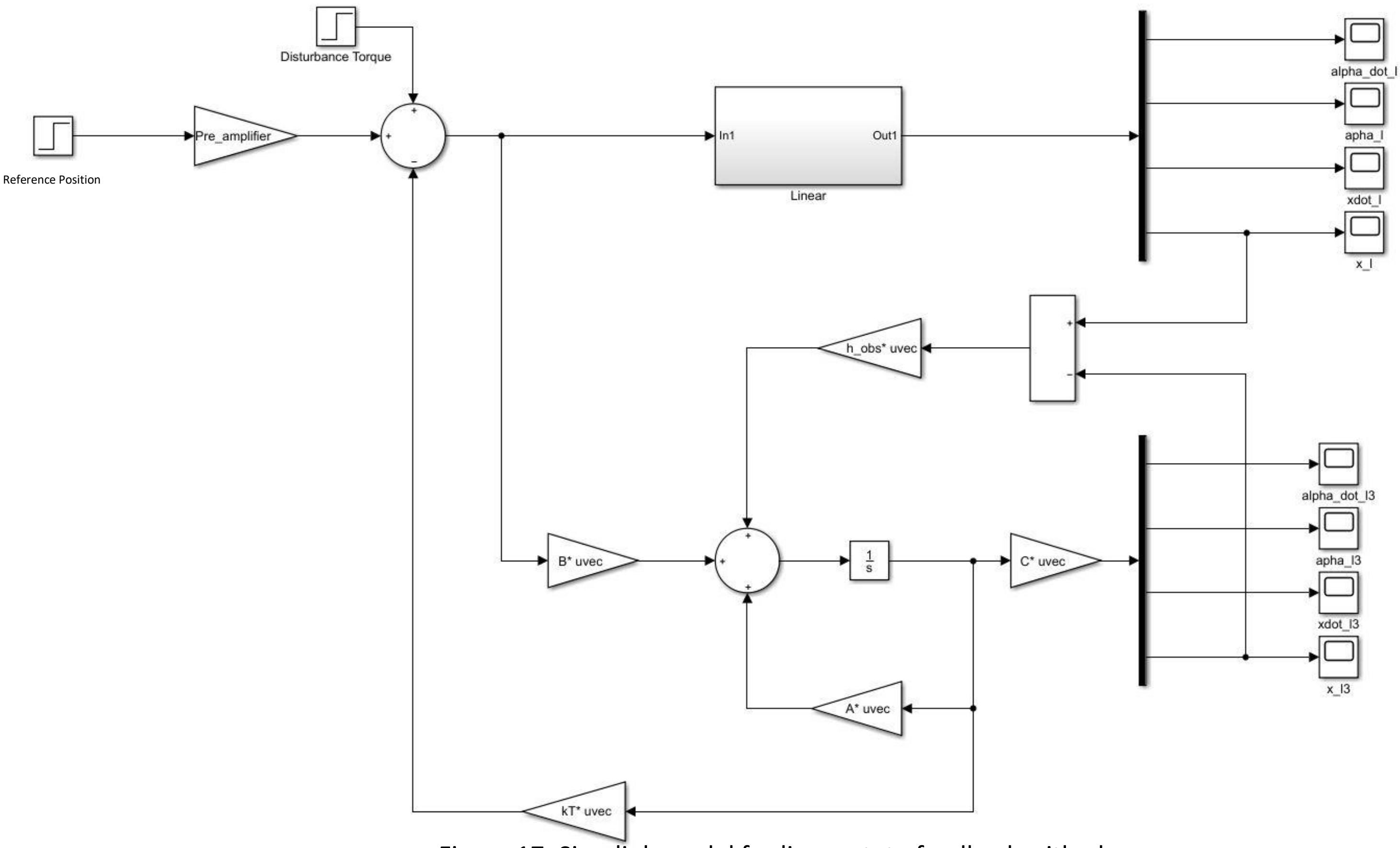
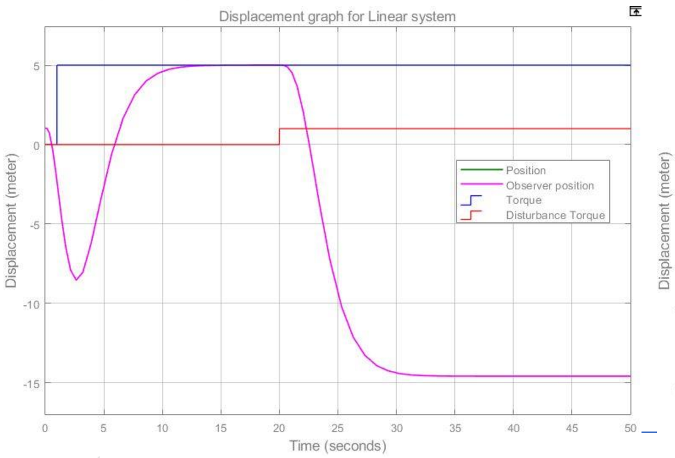


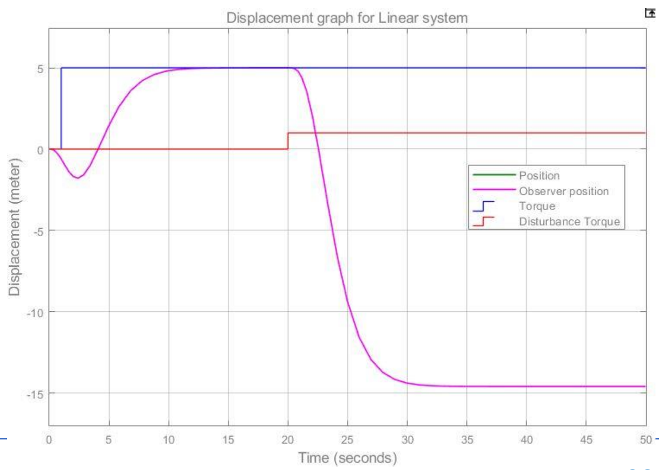
Figure 17: Simulink model for linear state feedback with observer

Comparison of displacement (for different initial conditions)

x1 = [1 1 1 1], Reference T = 5, Distb T = 1 at 20 s



x1 = [1 0 0 0], Reference T = 5, Distb T = 1 at 20 s



For linear state feedback with preamplifier and observer, the system response becomes inaccurate after application of disturbance torque.

Nonlinear State feedback with preamplifier along with observer (Task 4)-

Simulink model:

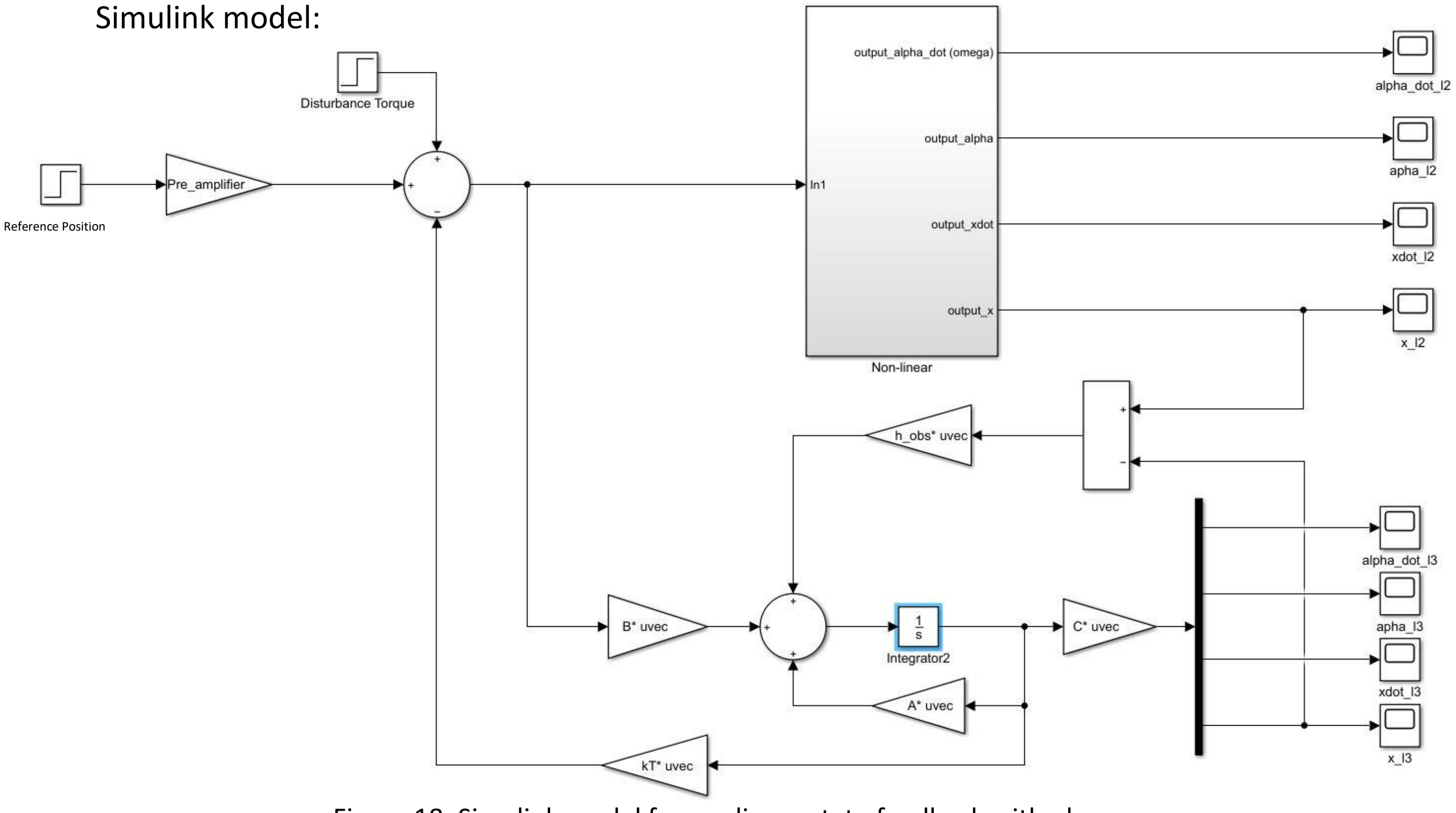
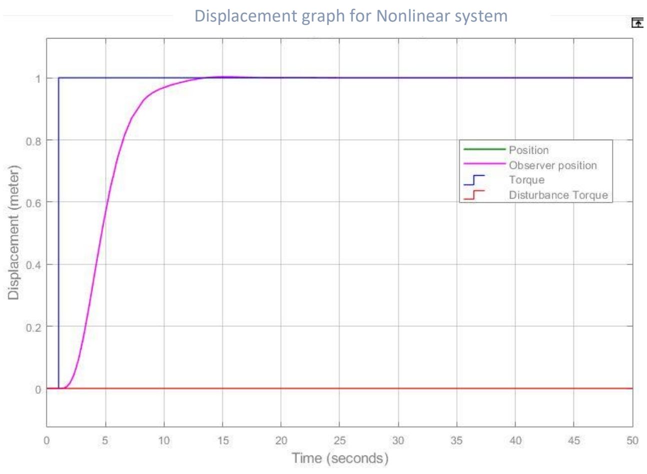


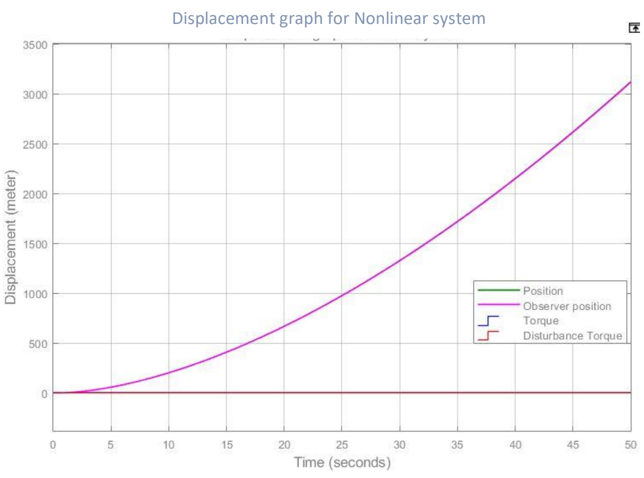
Figure 18: Simulink model for nonlinear state feedback with observer

Comparison of displacement (for different conditions)

x1 = [0 0 0 0], Reference T = 1, Distb T = 1



x1 = [1 0 0 0], Reference T = 2, Distb T = 1 at 20 s



For nonlinear state feedback with, system response becomes unstable if the magnitude of the reference torque is increased or initial condition is changed or disturbance torque is introduced.

Linear time continuous PI-state-feedback along with observer (Task 5)
Simulink model:

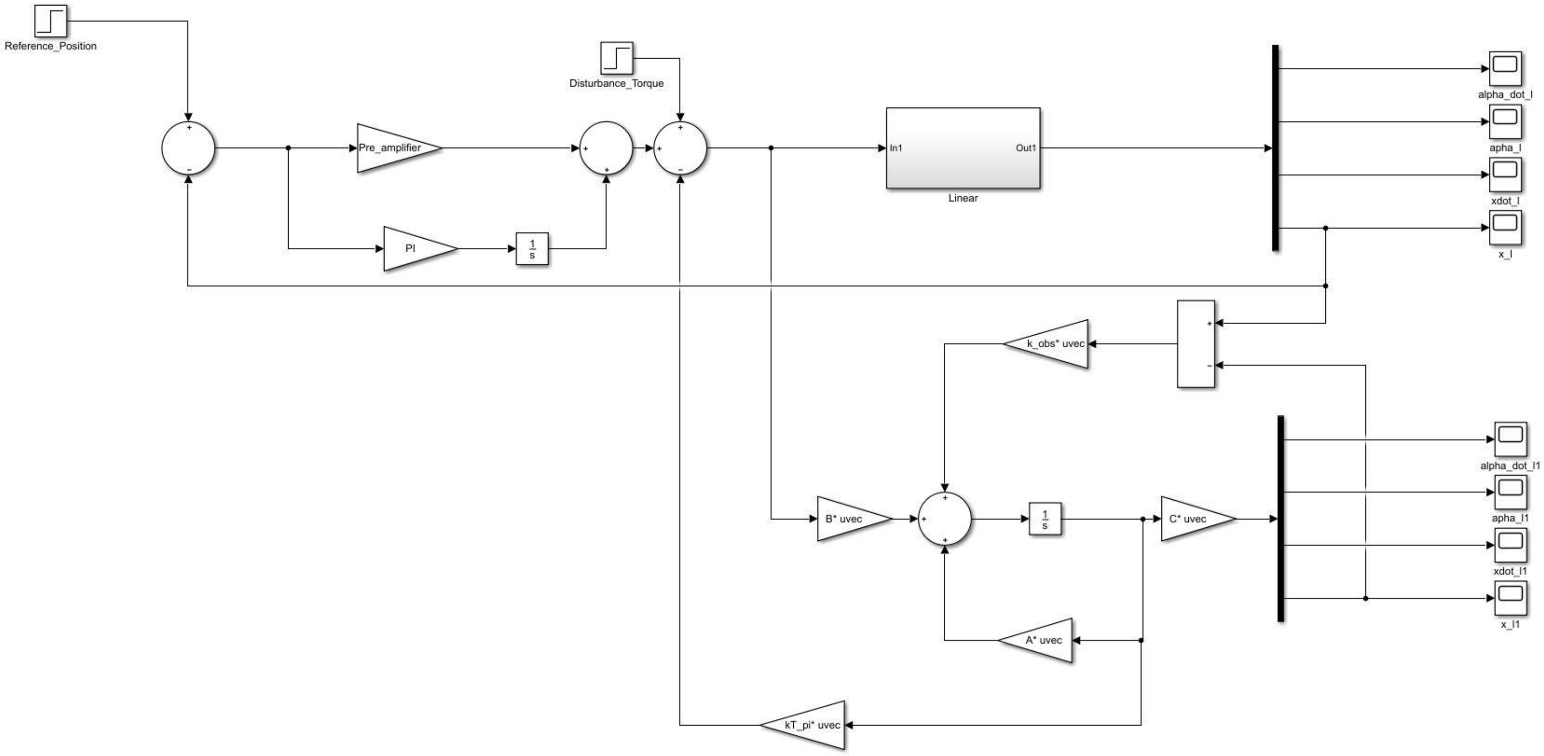
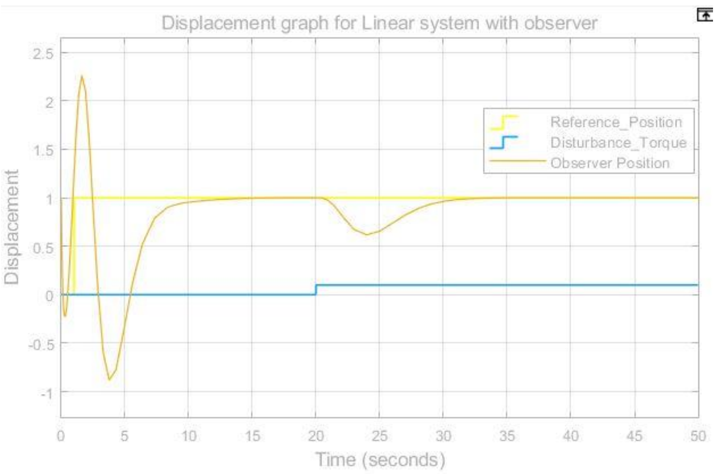


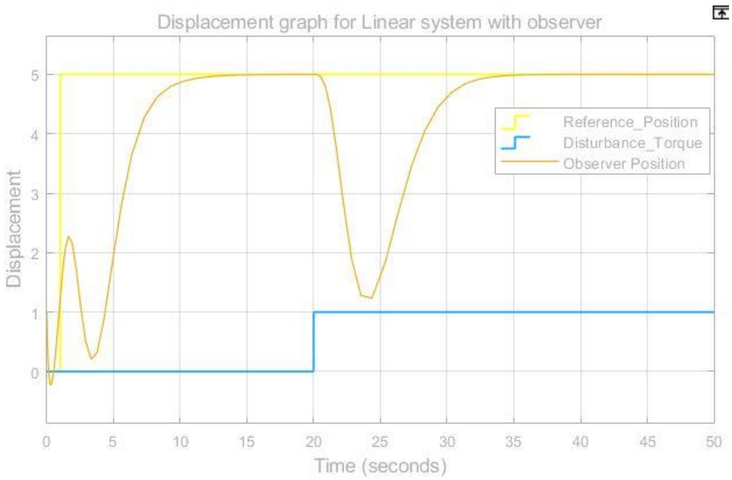
Figure 19: Simulink model for PI state feedback with observer for linear model

Comparison of displacement (for different conditions)

X1 = [0 0 0 1], Reference T = 1, Distb T = 0.1



X1 = [0 0 0 1], Reference T = 5, Distb T = 1 at 20 s



For linear PI-state-feedback with observer, the response of the system is accurate even if there is change in magnitude of reference torque or change of disturbance torque.

Linear discrete PI-state-feedback along with observer (Task 6)
Simulink model:

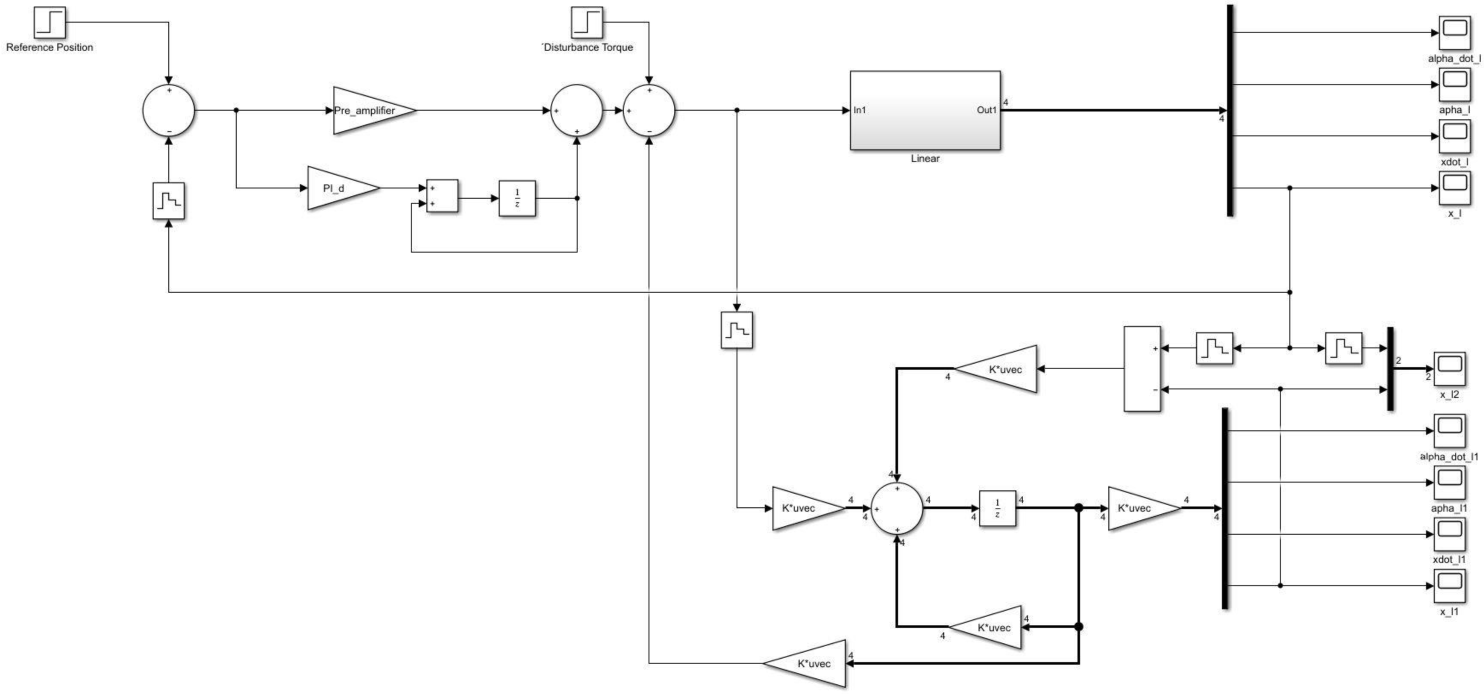
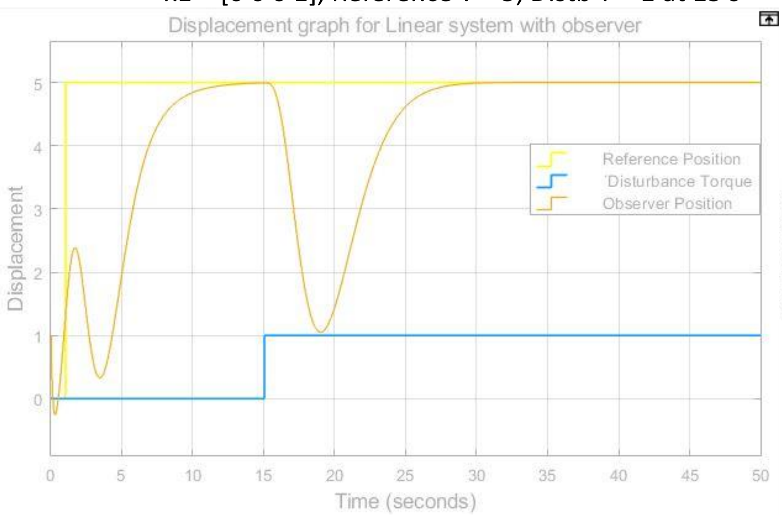


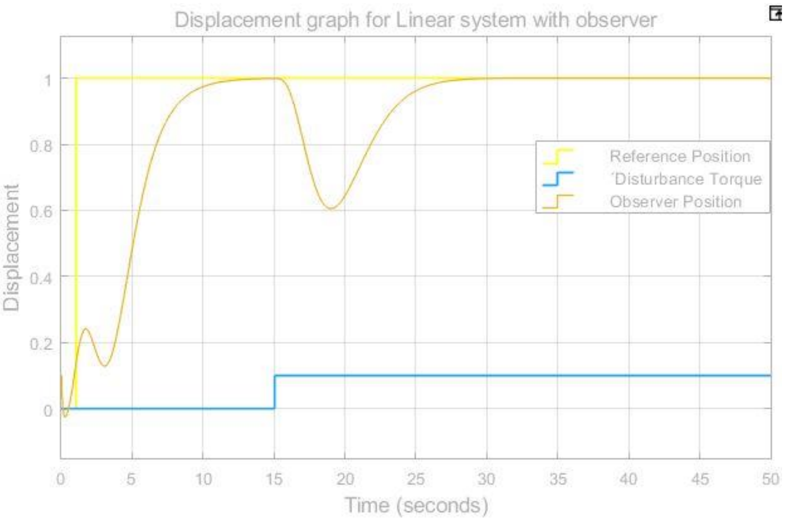
Figure 20: Simulink model for discrete PI state feedback with observer for linear model

Comparison of displacement (for different conditions)

x1 = [0 0 0 1], Reference T = 5, Distb T = 1 at 15 s



x1 = [0 0 0 0.1], Reference T = 1, Distb T = 0.1 at 15 s



For linear discrete PI-state-feedback with observer, the response of the system is accurate even if there is change in initial condition or magnitude of reference torque of introduction of disturbance torque.

Nonlinear discrete PI-state-feedback along with observer (Task 6)
Simulink model:

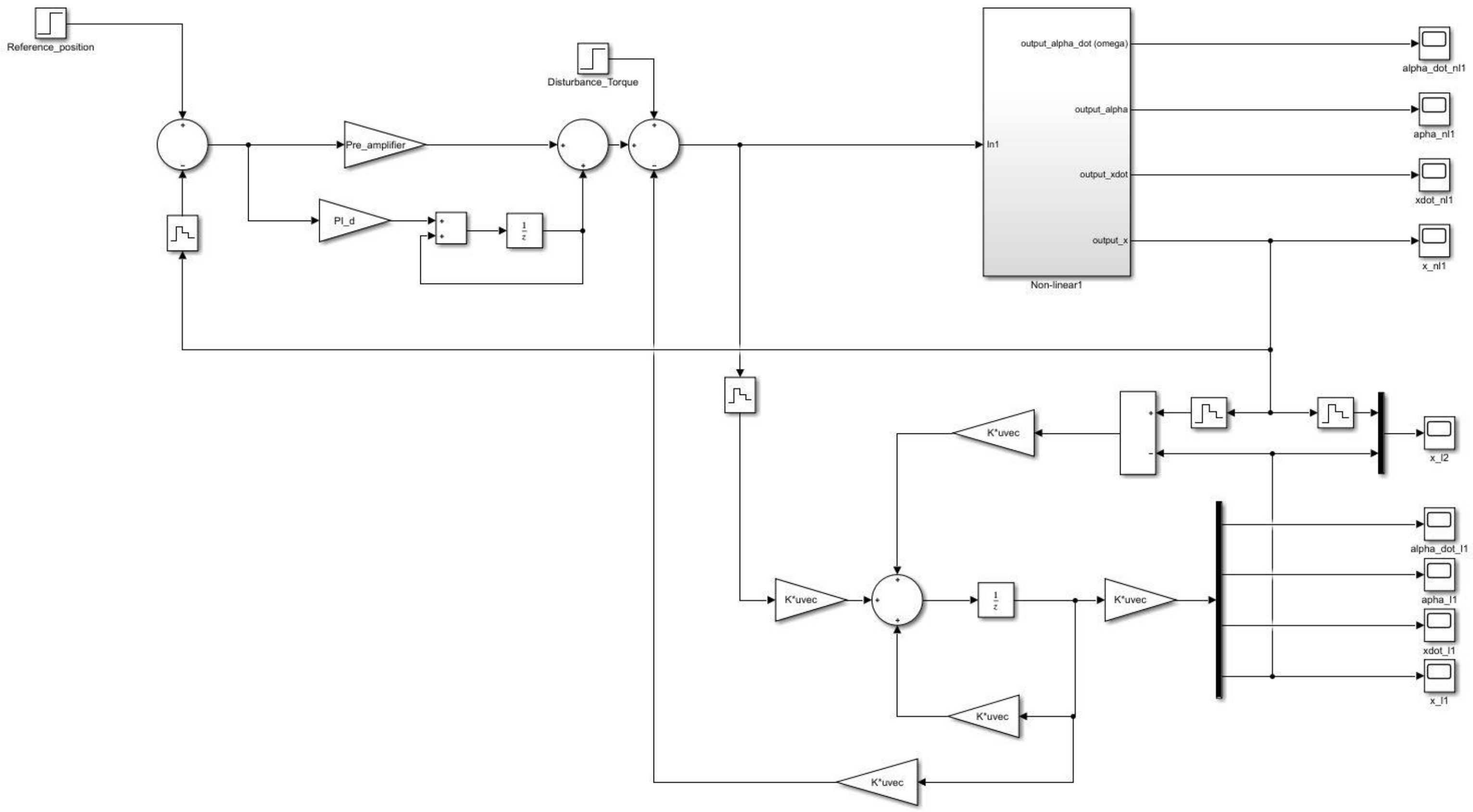
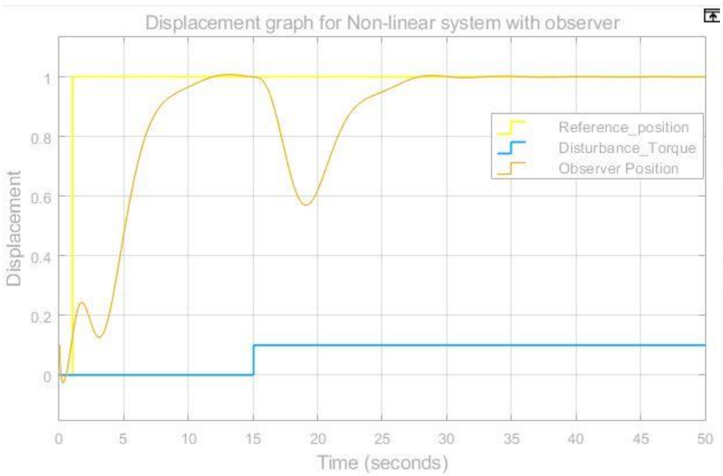


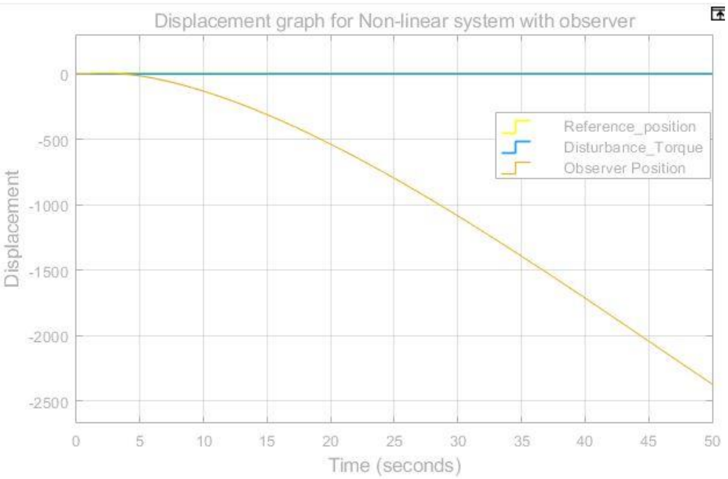
Figure 21: Simulink model for discrete PI state feedback with observer for nonlinear model

Comparison of displacement (for different conditions)

X1 = [0 0 0 0.1], Reference T = 1, Distb T = 0.1 at 15s



X1 = [0 0 0 1], Reference T = 5, Distb T = 1 at 15 s



For nonlinear discrete PI-state-feedback with observer, the response of the system is accurate even if there is small change in initial condition or disturbance torque. But becomes inaccurate if there is large change in reference position, disturbance torque and initial conditions.

Conclusion

The skills for sophisticated development and control of Seesaw system was studied in depth by making nonlinear dynamic system in Matlab Simulink, linearizing it and making linear state space model for a working point, making a feedback control based on linear model and calculating feedback vector and preamplifier, developing a PI-state-feedback control based on the linear model, converting the continuous system to discrete, developing a full state observer and calculate the error feedback vector, comparing and understanding the effect of change in initial conditions, reference value steps, disturbance torque in linear and nonlinear systems.

Thus, in the process of achieving the objective, the knowledge on control systems was deepened and understanding of discrete control systems was increased.