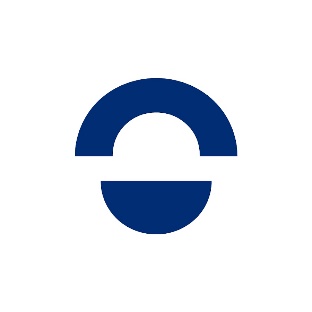
**Discrete Control Systems**

**Lab Report**



**A**

**LAB REPORT**

Submitted as a part of the requirement for

**Lab Task in Discrete Control Systems**

in partial fulfilment for the award of the degree of

**Masters in Embedded System Design**

By

**Krish M. Shah (Mr.no: 41564)**

**Kishan H. Kalariya (Mr.no: 41641)**

Under the guidance of

**Prof. Dr. -Ing. Karsten Peter**

**Table of Contents**

[Table of Contents 2](#_Toc168936206)

[INTRODUCTION 3](#_Toc168936207)

[Task 1 4](#_Toc168936208)

[Task 2 7](#_Toc168936209)

[Task 3 11](#_Toc168936210)

[Task 4 13](#_Toc168936211)

[Task 5 17](#_Toc168936212)

# **INTRODUCTION**

Discrete control system refers to the control theory of discrete – time systems which depends on discrete variational principles. In this report, an example of Seesaw system is taken into account for understanding and developing sophisticated and complex control system. This system is divided into 8 different tasks for better understanding in this report.

OBJECTIVE- The objective of the Lab is to design and model a non-linear dynamic system, a Seesaw system in MATLAB Simulink. Linearize and make linear state space model for a working point. Develop a PI-state-feedback control based on linear model. Convert continuous system to discrete. Develop full state observer and calculate the error feedback vector. Compare and observe the change in initial conditions, reference value steps, disturbance torque in linear and nonlinear systems. The systems to be modelled is as below:

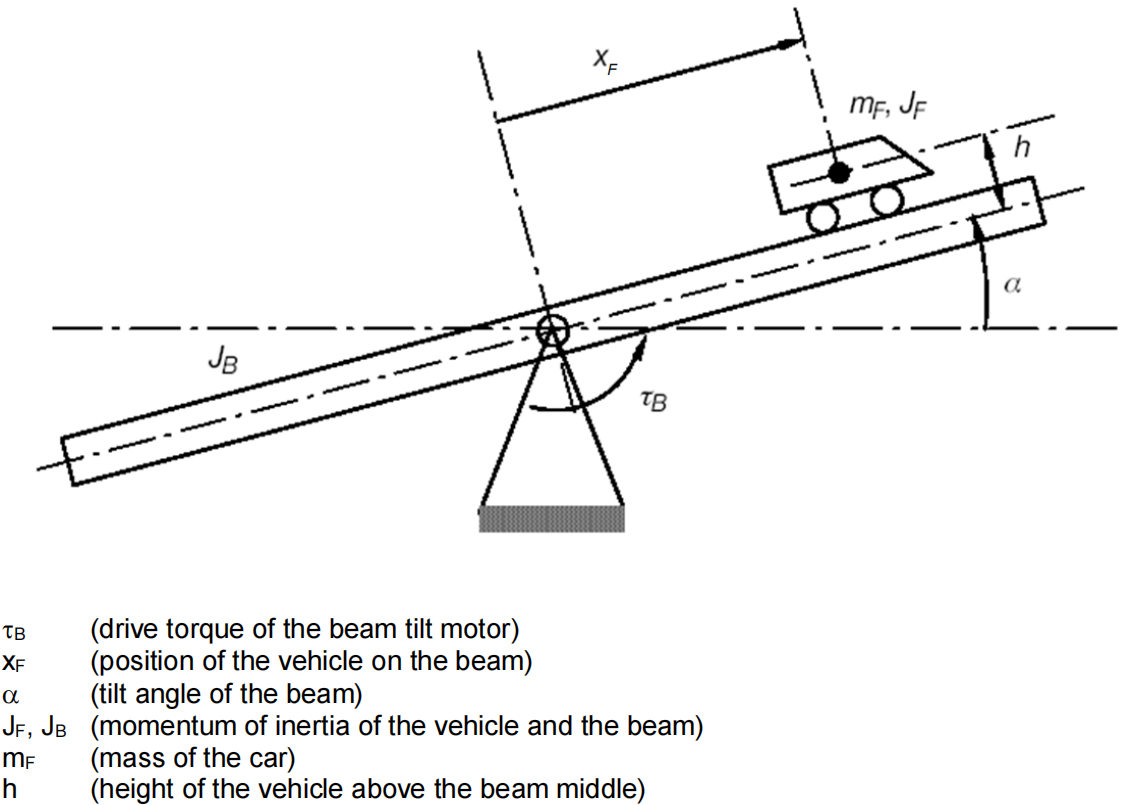


Figure 1: Seesaw System

Where,

: Drive torque of the beam tilt motor

: Position of vehicle on the beam

: Tilt angle of the beam

: Momentum of inertia of the vehicle and the beam

: Mass of the car

: Height of the vehicle above the beam middle

# **Task 1**

* Make a nonlinear model in MATLAB Simulink for the Seesaw System

The equation for non-linear dynamic Seesaw system movement can be modelled by using D’Alembert’s Principle or commonly known as Euler-Lagrange Formulism.

The Lagrange formula that describes the dynamic behaviour of system can be described by:

Where,

and

Lagrange function L is:

Solving for and , we get:

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

Using these two equations the nonlinear Seesaw system can be modelled 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;

%code for getting values from worksapce

alpha=out.alpha;

alpha\_dot=out.alpha\_dot;

xf=out.xf;

xf\_dot=out.xf\_dot;

%code for ploting and placement of graphs

figure=nexttile;

plot (alpha)

title('alpha')

xlabel('time in seconds')

ylabel('angle in radian')

legend('alpha')

figure=nexttile;

plot (alpha\_dot)

title('alpha dot')

xlabel('time in seconds')

ylabel('angle in radian')

legend('alpha dot')

figure=nexttile;

plot(xf\_dot)

title('xf dot')

xlabel('time in seconds')

ylabel('distance in meters')

legend('xf dot')

figure=nexttile;

plot(xf)

title('xf')

xlabel('time in seconds')

ylabel('distance in meters')

legend('xf')

hold off

**Simulink Model:**



Fig 1.1 Non-linear Simulink model

# **Task 2**

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

The process of linearization is used to evaluate the local stability of an equilibrium point in discrete dynamical systems or systems of nonlinear differential equations.

The Seesaw system must be linearized around working point zero using the following formula:

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

The Seesaw system's linearized form is fully described by the equation above.   
The Seesaw system's state space can be represented by these two equations.

State-Space Representation: A state-space representation represents a physical system mathematically as a collection of

Difference equations or first-order differential equations link the input, output, and state variables. State variables are those whose values fluctuate over time in a manner that is dependent upon both their current values and the values of input variables that are imposed from the outside. Values of state variables influence the values of output variables.

The following is the representation of the LTI state space:

It is a first-order differential equation in state space form. By combining two second order differential equations to create four first order equations, the system description of the Seesaw system can be expressed as a matrix as follows:

Here:

is c

0 is D

Where

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;

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=[0,0,0,1];

D=0;

alpha\_compare=out.alpha\_compare;

alpha\_dot\_compare=out.alpha\_dot\_compare;

xf\_compare=out.xf\_compare;

xf\_dot\_compare=out.xf\_dot\_compare;

%code for ploting and placement of graphs

figure=nexttile;

plot (alpha\_dot\_compare)

title('Angular Velocity Comparision')

xlabel('Time in Seconds')

ylabel('Angular velocity')

legend('alpha dot non-linear','alpha dot linear')

grid on

grid minor

figure=nexttile;

plot (alpha\_compare)

title('Angle Comparision')

xlabel('Time in Seconds')

ylabel('Angle in Radian')

legend('alpha non-linear','alpha linear')

grid on

grid minor

figure=nexttile;

plot(xf\_compare)

plot(xf\_dot\_compare)

title('Displacement Comparision')

xlabel('Time in seconds')

ylabel('Displacement in Meters')

legend('xf','xf linear')

grid on

grid minor

**MATLAB Code Continue:**

figure=nexttile;

plot(xf\_dot\_compare)

title('Velocity Comparision')

xlabel('Time in seconds')

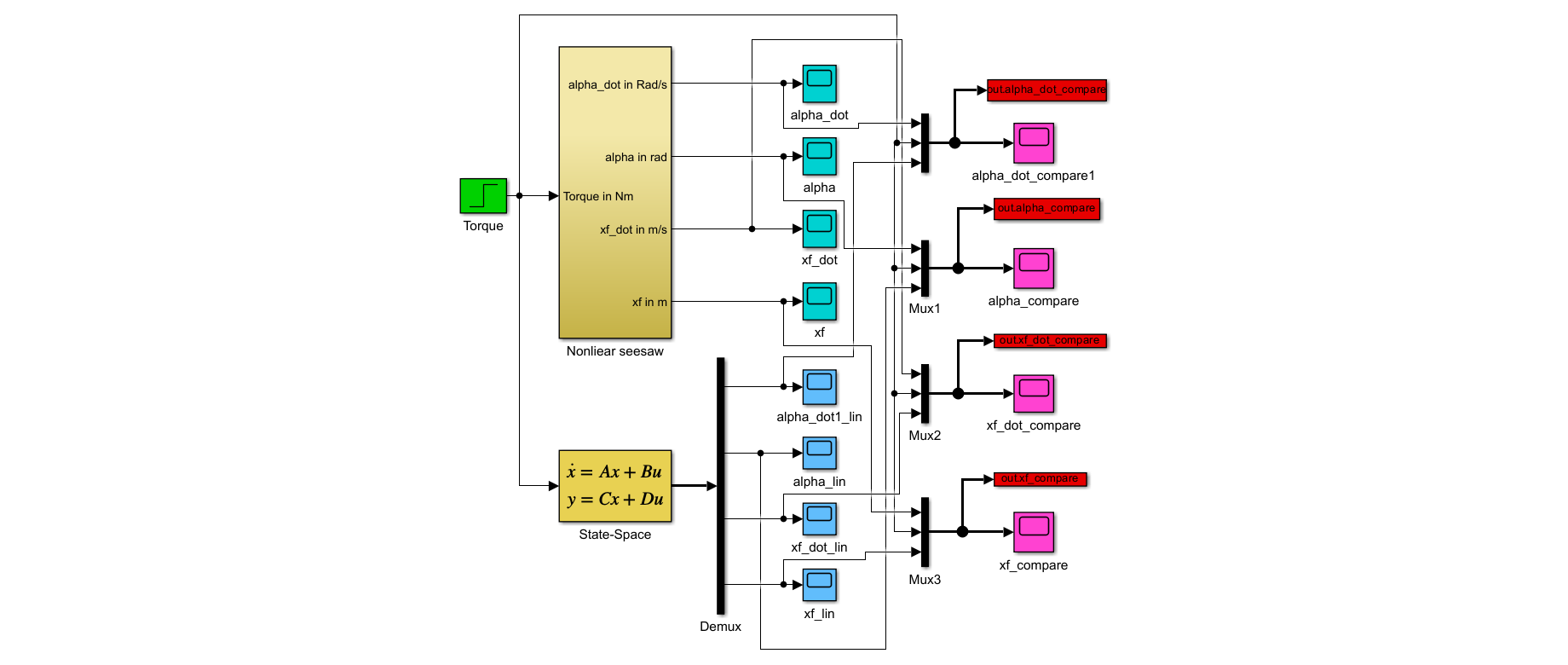
ylabel('Velocity in m/s')

legend('xf dot','xf dot linear')

grid minor

grid on

**Simulink Model:**

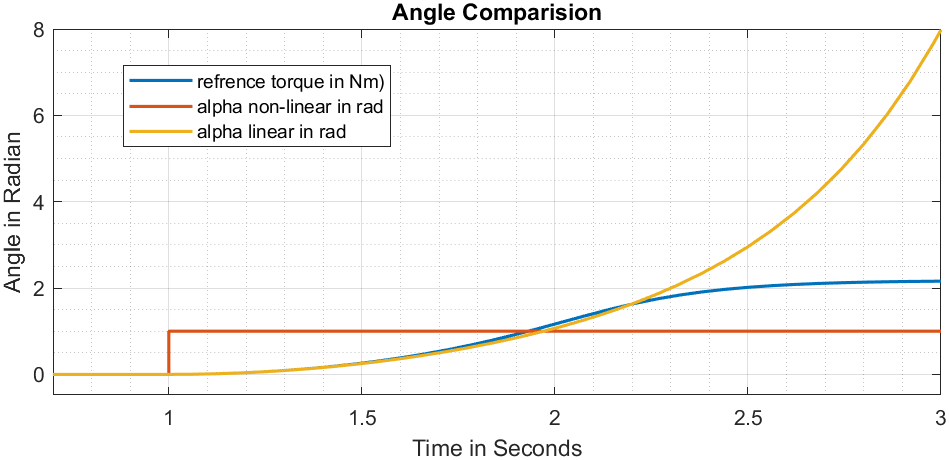


**Fig2.1 Simulink model**

# **Task 3**

**A comparison between the nonlinear and linear models is made.**

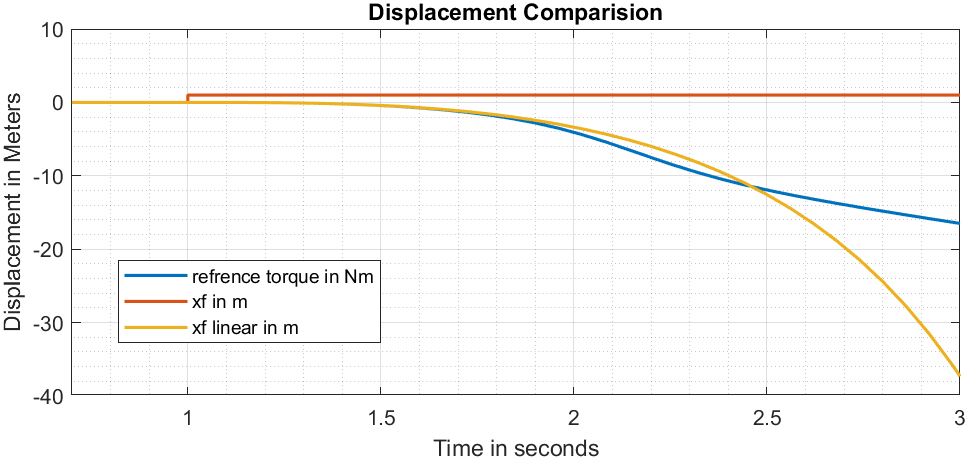
* At what point does the linear model lose its precision? In state-space
* how far can the working point deviate from the original working point from which the linear system was derived?





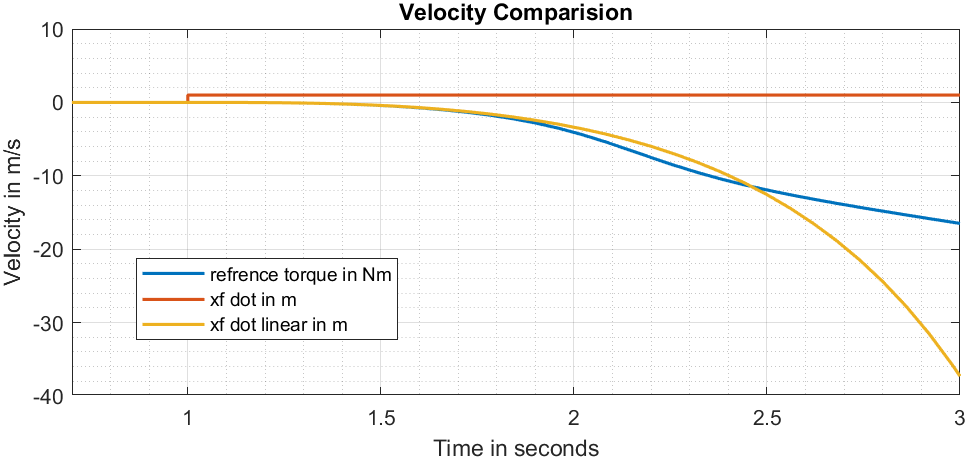
**Fig3.1 Angle Comparison**

The above figure showing angular rotation vs time for both linear and nonlinear models make this clear. Because the curve for a linear model point in that direction, but it doesn't go above 90 degrees for a nonlinear model.



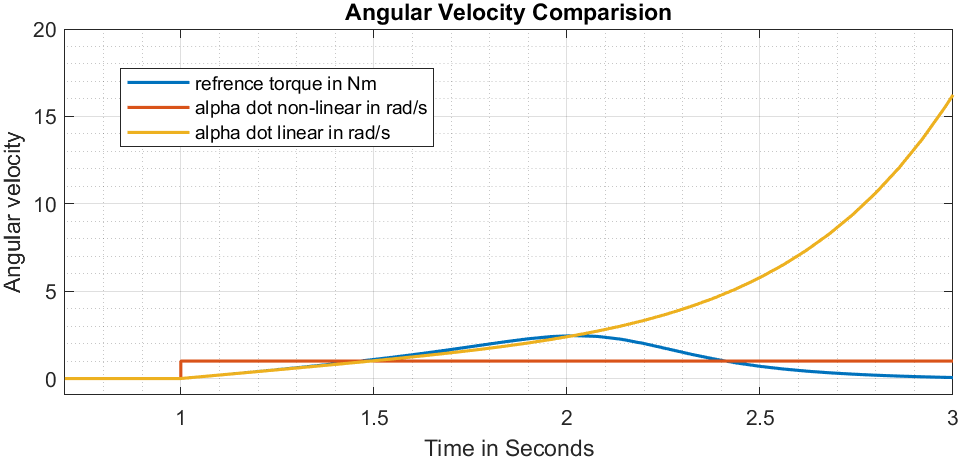
**Fig 3.2 Displacement Comparison**

It is clear from the displacement vs. time graphs for the linear and nonlinear models. Note the downward curve for the linear model is steeper than the nonlinear model, but it starts a little later in time.



**Fig 3.3 Velocity Comparison**

The linear model's downward curve begins a bit later in time than the nonlinear model, but it is steeper overall, as can be seen from the velocity vs time graphs for the two models.



**Fig 3.4 Angular Velocity Comparison**

The above figures showing angular velocity versus time for both linear and nonlinear models make this clear. Thus, the nonlinear model's curve falls back and doesn't rise above 90 degrees, but the linear model's curve heads in that direction.

# **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.

State feedback is a technique used in feedback control system theory to position a plant's closed-loop poles in the s-plane at predefined points. Placing the poles where they directly correlate to the system's eigenvalues, which govern the features of the system's response, makes this arrangement advantageous. This strategy can only be applied if the system is deemed controllable. The preceding task demonstrated the instability of both linear and nonlinear models.

The location of the open loop poles is at:

>> sys\_poles

sys\_poles =

2.1170 + 0.0000i

-2.1170 + 0.0000i

0.0000 + 2.0703i

0.0000 - 2.0703i

Since the poles are not located in the left half of the s-plane, the system is unstable. The poles should be located on the left half of the plane in order to stabilize the system. Ackerman's formula would be applied to set all of the poles to -1. Additionally, the preamplifier P and state feedback vector kT must be calculated.

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

And the following formula can be used to determine the pre amplifier needed to ensure steady state accuracy for state feedback control:

**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;

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=[0,0,0,1];

D=0;

sys\_poles = eig(A);

S = [B A\*B A^2\*B A^3\*B]; %controllability matrix

rank(S) % check for full rank of controllability

S\_inv = inv(S); %inverse of controllability matrix

qT = S\_inv (4,:); %taking last row of controllablity

p1=-1; %setting all poles to -1

p2=-1;

p3=-1;

p4=-1;

alpha = poly([p1 p2 p3 p4]); %converting roots to polynomial

%applying akermanns formula

kT = qT\*(alpha(5)\*eye(4)+alpha(4)\*A+alpha(3)\*A^2+alpha(2)\*A^3+A^4);

% Ct = C(4,:);

P=1/(C\*inv(B\*kT-A)\*B); % pre amplifier

linear\_displacement=out.linear;

non\_linear\_displacement=out.non\_linear;

% %code for ploting and placement of graphs

figure=nexttile;

plot (linear\_displacement)

title('Linear Displacement')

xlabel('Time in Seconds')

ylabel('Displacement in meters')

legend('position linear','refrence position','disturbance torque')

grid on

grid minor

figure=nexttile;

plot (non\_linear\_displacement)

title('Non Liear Displacement')

xlabel('Time in Seconds')

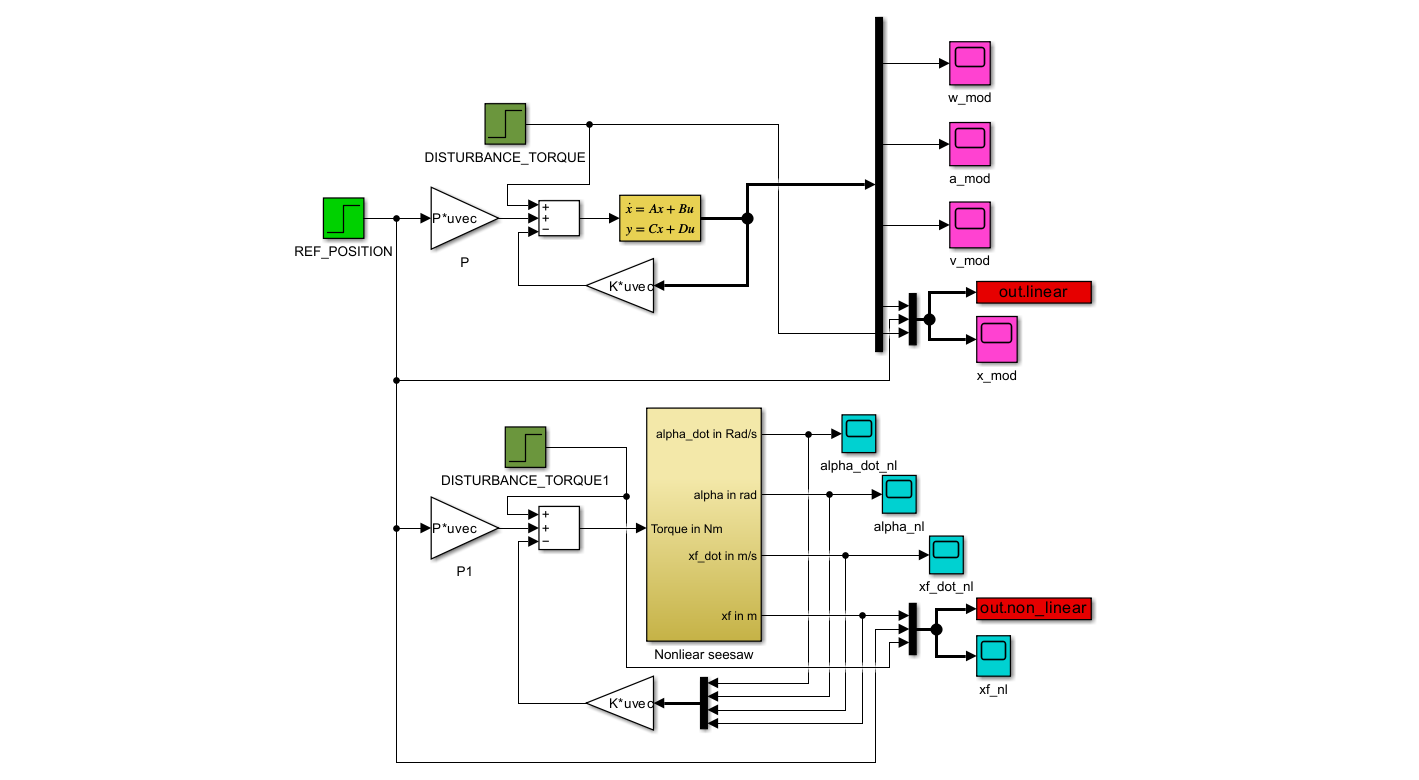
ylabel('Displacement in meters')

legend('position linear','refrence position','disturbance torque')

grid on

grid minor

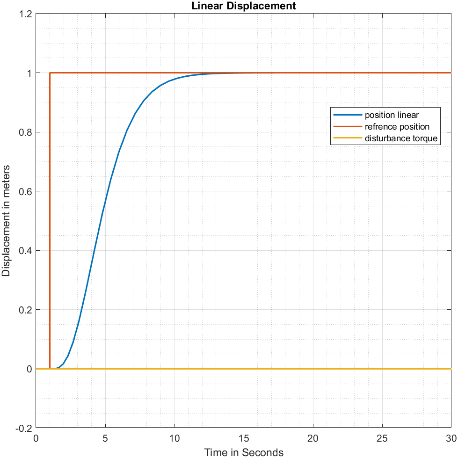
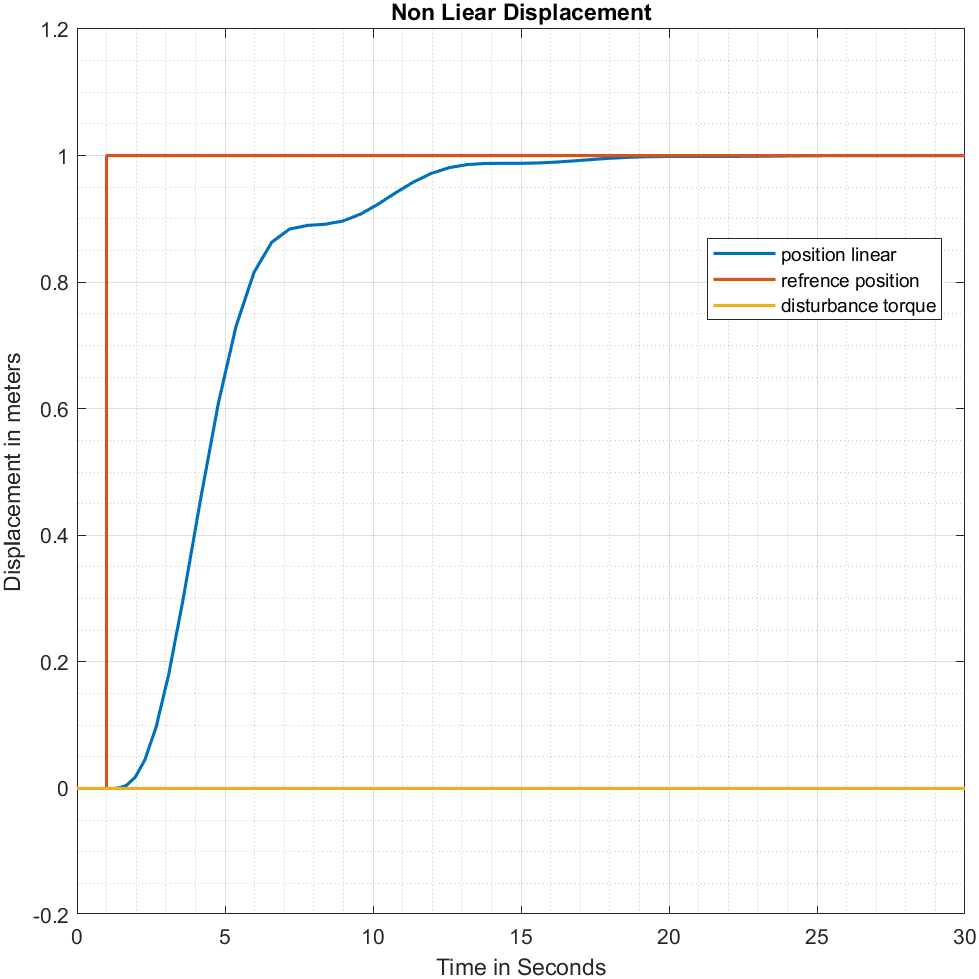
**Simulink Model**:





**Fig 4.1 Simulink model**

Comparison of displacement (**without disturbance torque**):

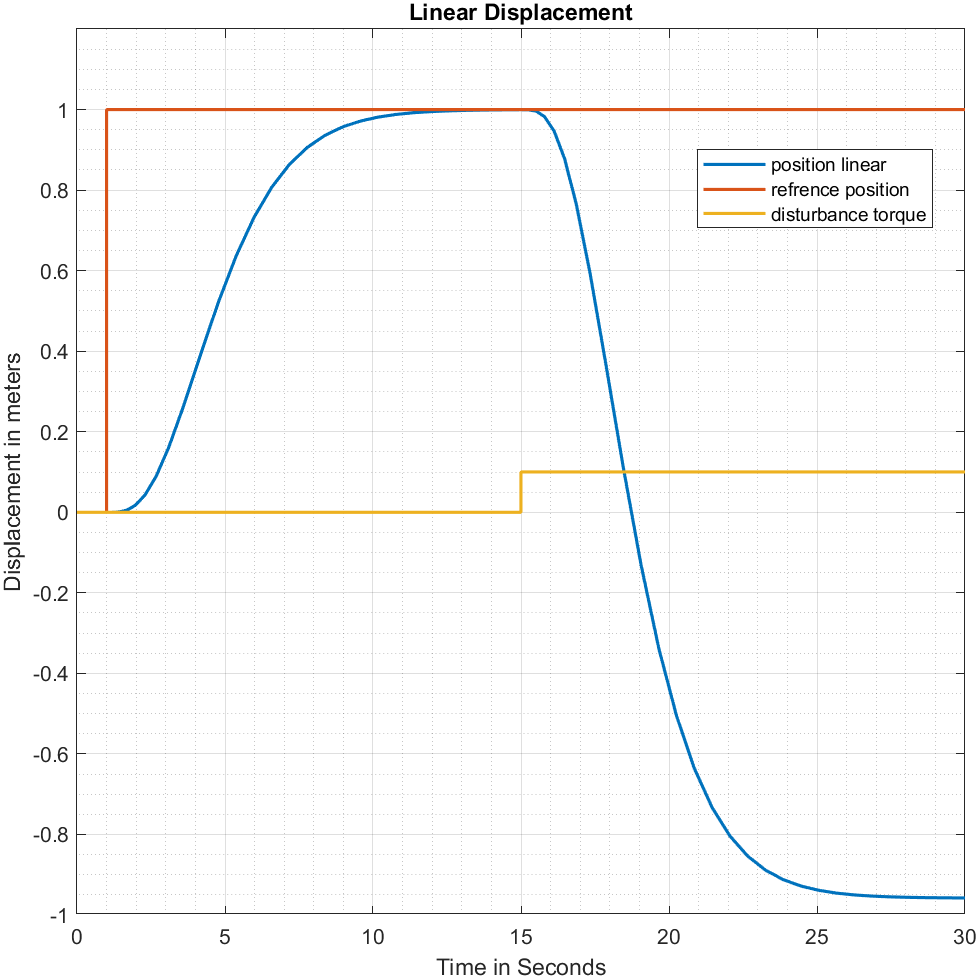
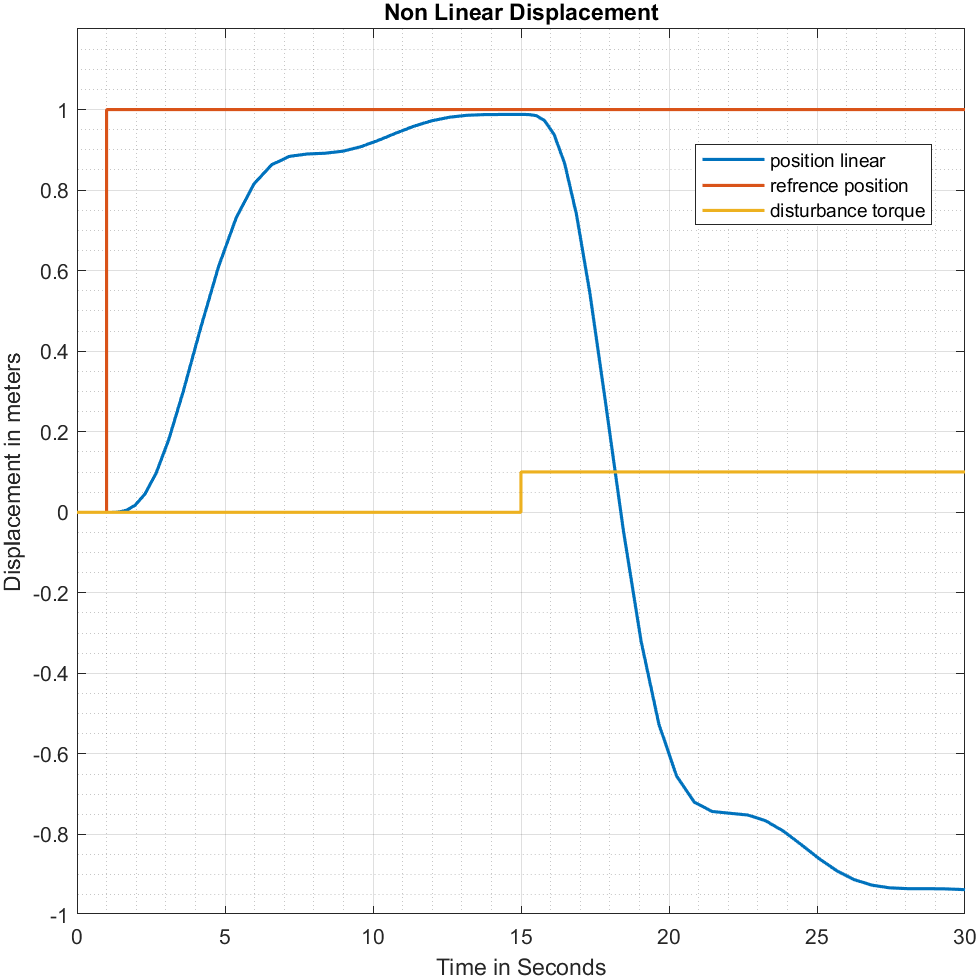




**Fig 4.2 Comparison of linear and non-linear Displacement with Torque=0**

The system achieves stability when no disturbance torque is supplied, and it also follows the reference torque, meaning that it accomplishes steady state accuracy using the preamplifier gain and estimated state feedback vector. For both linear and nonlinear systems, the same response is obtained from the system.

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





**Fig 4.3 Comparision of Linear and Non-linear Displacement with Torque=0.1**

The controller is unable to offset the distortion torque when the disturbance torque (0.1 Nm at 15 sec) is applied. The system does not follow the reference torque for either linear or nonlinear systems, even though it is stable.

# **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.

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.

Together 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.

Given that it returns the state z, the P is understood to be a component of the state-feedback vector. The augmented system in state-space can be described as follows as a result of this:

Here:

**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=[0,0,0,1];

D=0;

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=[0,0,0,1];

D=0;

sys\_poles = eig(A);

S = [B A\*B A^2\*B A^3\*B]; %controllability matrix

rank(S) % check for full rank of controllability

S\_inv = inv(S); %inverse of controllability matrix

qT = S\_inv (4,:); %taking last row of controllablity

p1=-1; %setting all poles to -1

p2=-1;

p3=-1;

p4=-1;

alpha = poly([p1 p2 p3 p4]); %converting roots to polynomial

%applying akermanns formula

kT = qT\*(alpha(5)\*eye(4)+alpha(4)\*A+alpha(3)\*A^2+alpha(2)\*A^3+A^4);

P=1/(C\*inv(B\*kT-A)\*B); % pre amplifier

Api= [A [0;0;0;0]; -C 0];

Bpi= [B; 0];

Cpi= [0 0 0 1 0];

Dpi= [0;0;0;0;0];

**Continued Matlab Code**:

%Contobility matrix

Spi= [Bpi Api\*Bpi Api^2\*Bpi Api^3\*Bpi Api^4\*Bpi];

%Inverse of controbility matrix

Spi\_inv = inv(Spi);

%last Row of controliblity matrix

qT\_pi = Spi\_inv(5,1:5);

p1= -1;

p2= -1;

p3= -1;

p4= -1;

p5= -1;

%convert roots to polynomial

alpha\_pi= poly([p1 p2 p3 p4 p5]);

%apply akkermann 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);

%State Feedback Value

kT\_pi = kpi(1:4)-P\*C;

% code for ploting and graph placement

linear\_displacement=out.linear1;

plot (linear\_displacement)

title('Linear Displacement')

xlabel('Time in Seconds')

ylabel('Displacement in meters')

legend('position linear','refrence position','disturbance torque')

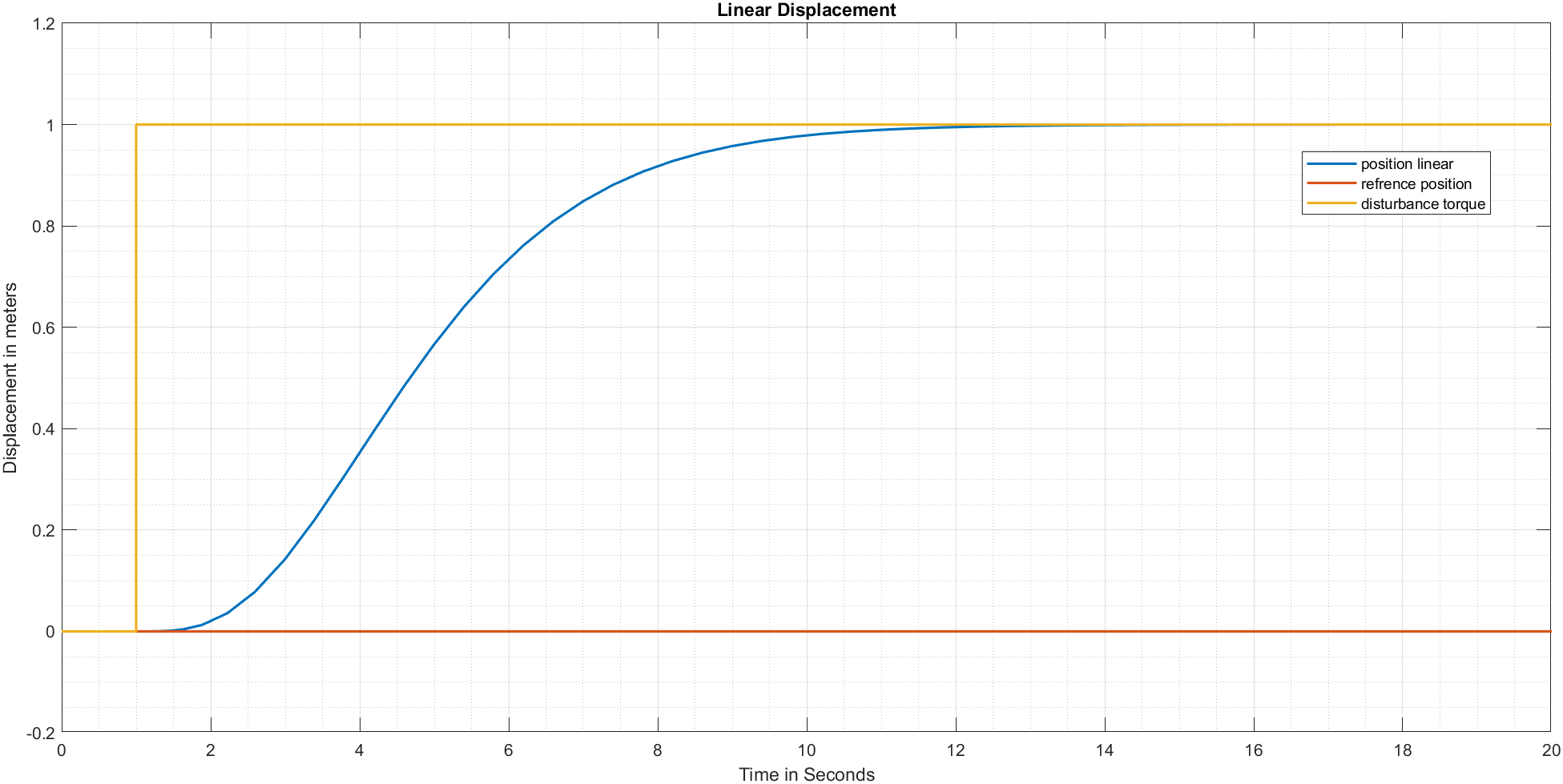
grid on

grid minor

Simulation Model:

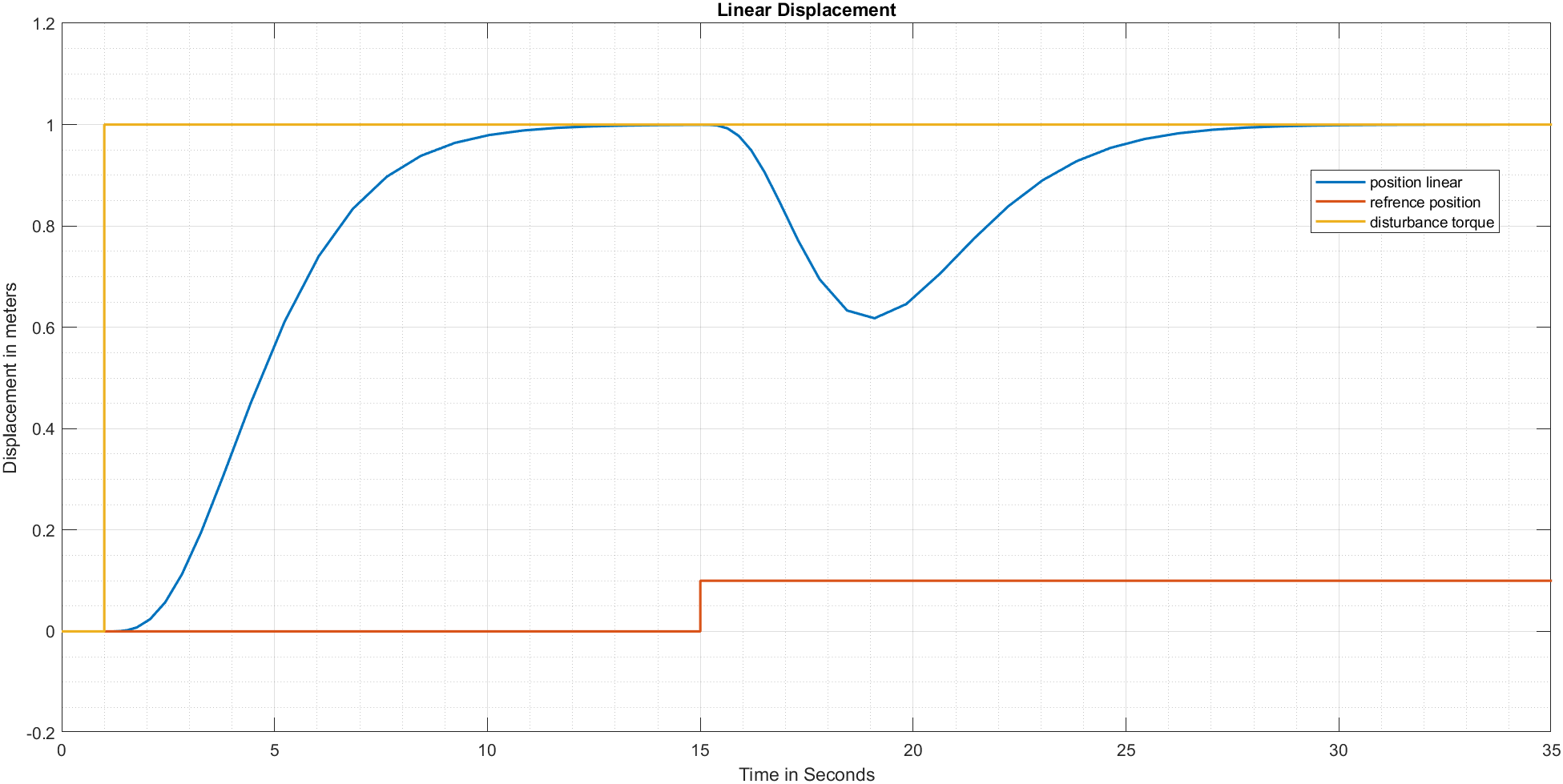


**Fig 5.1 Simulink model for Task 5**





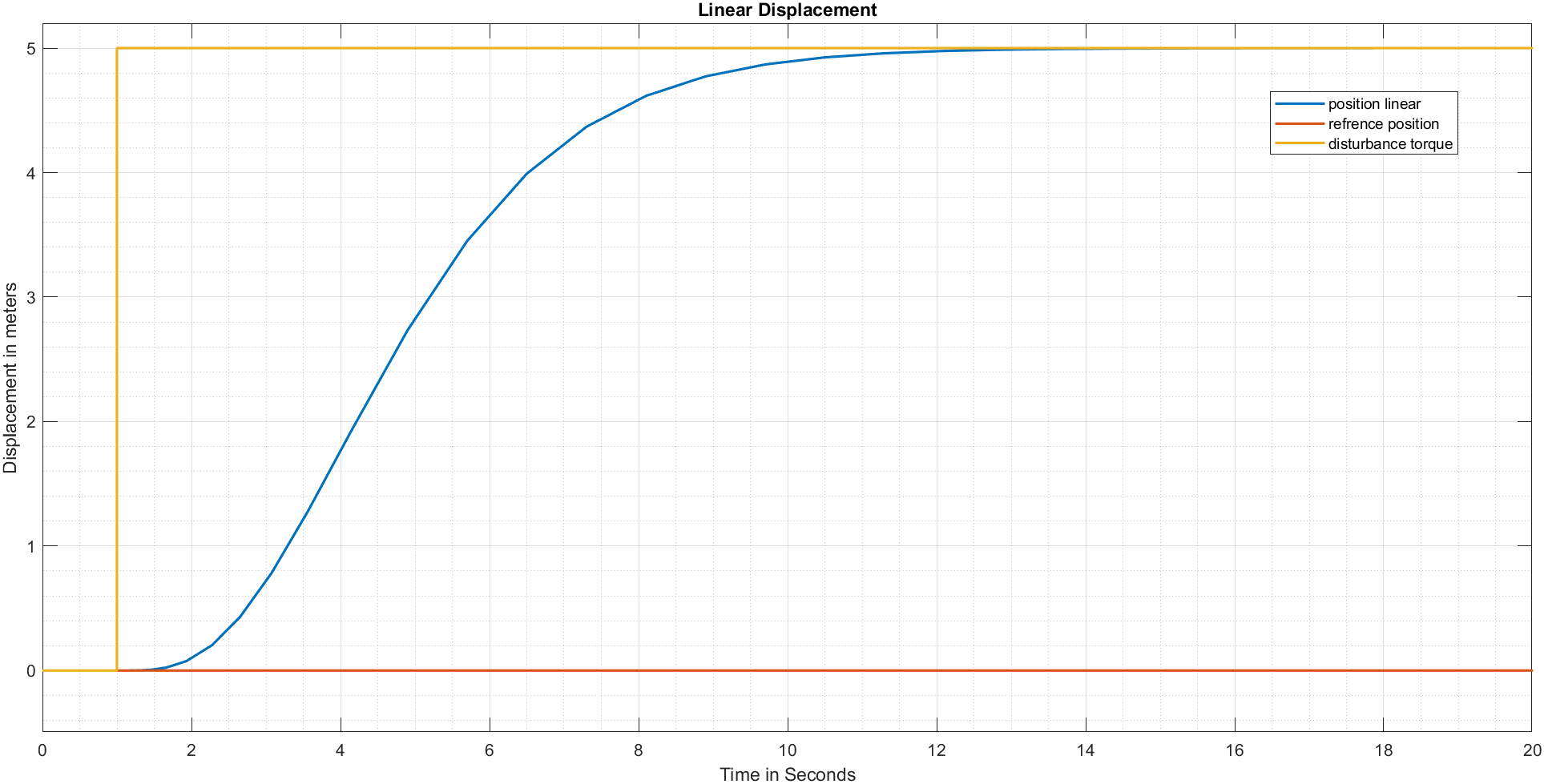
**Fig 5.2** Linear Displacement **with no disturbance**

****



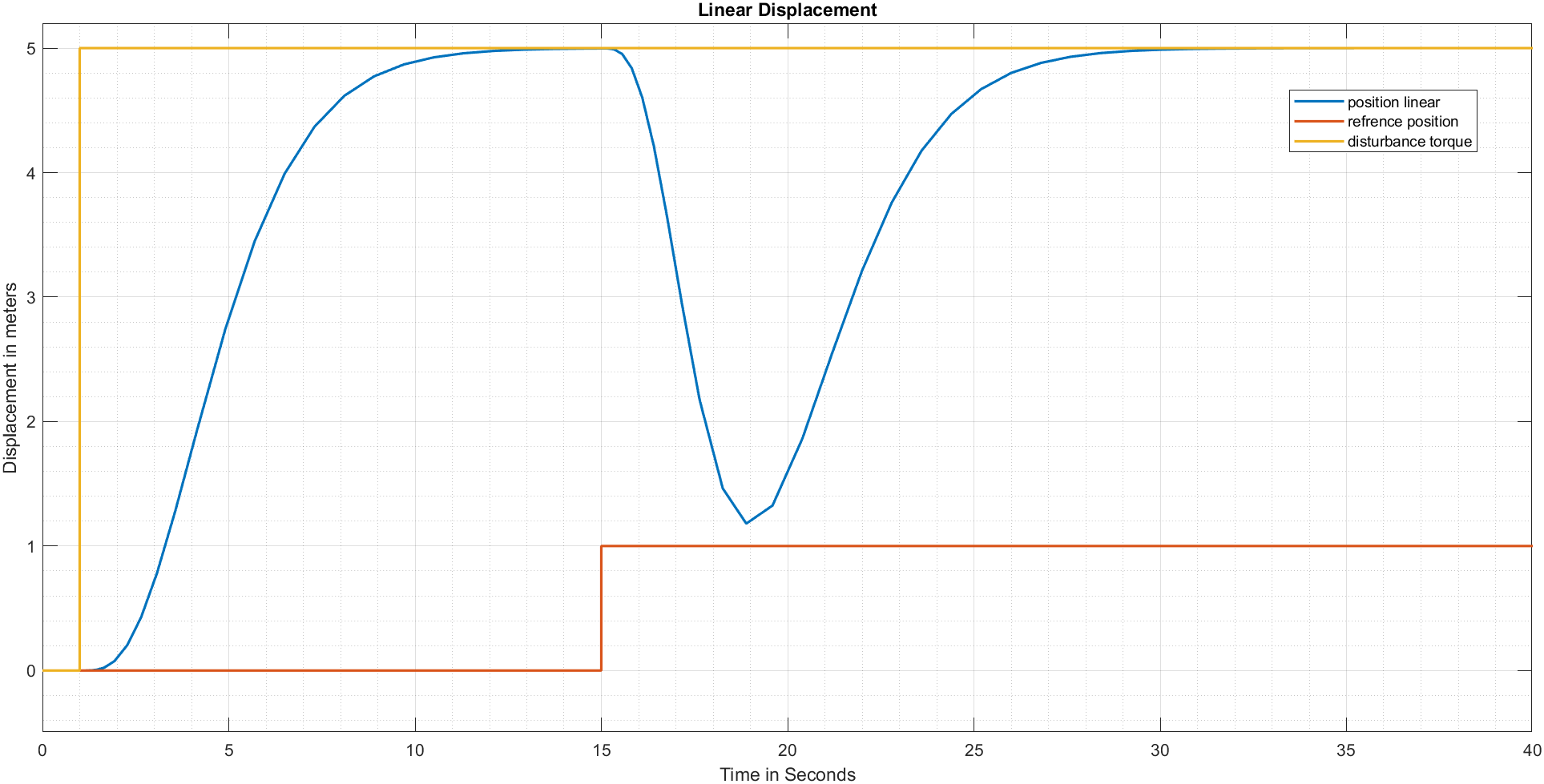
**Fig 5.3** Linear Displacement **with low disturbance**

The figures above make it clear that in the absence of any disturbance, the system's response follows the reference torque, giving rise to the "S" curve. The system's reaction becomes disrupted and resumes following the reference torque after 15 seconds with disturbance torque of 0.1 added.





**Fig 5.4** Linear Displacement **with No disturbance**





**Fig 5.5** Linear Displacement **with High disturbance**

Now it is clear from above figures that in the absence of any disturbance, the system's response follows the reference torque, giving rise to the "S" curve. The system's reaction becomes disrupted and resumes following the disturbance torque of 1 added at 15 seconds.

# **Task 6**