Hochschule Bremerhaven

Seesaw System

Lab report

ALI HASAN TALHA	33433
MUDDASSIR NAWAZ	32447
WAQAR AKRAM	33419

TABLE OF CONTENTS

TABLE OF FIGURES

ACKNOWLEDGEMENT

We would like to thank Prof. Dr- Ing Karsten Peter for his continuous support and guidance for this lab work.

TASK

The main purpose of this lab was to model a seesaw system and stabilize the system via different control techniques using MATLAB® and SIMULINK®. Two mathematical models, linear and non-linear, are made for simulation purposes and control techniques are applied to both the models and then compared. The control methods used are:

1. Open loop state feedback gain controller.
2. Closed loop PI controller.
3. Closed loop Discrete PI controller.
4. Full state Observer and state reconstruction.
5. Full state Discrete Observer.

MODELLING

A seesaw system with a wagon on top is the intended system to be controlled as shown in the figure below.

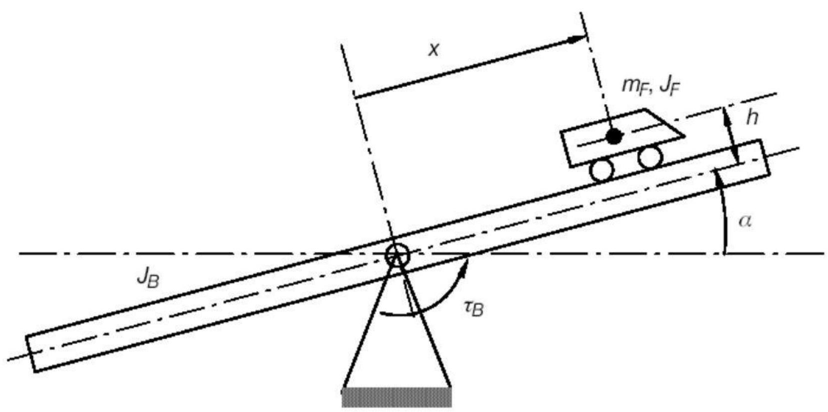


Figure 1 Model of a seesaw system with an automobile on top

Following are the parameters used in figure above:

m_F	Mass of the wagon.	α	Angle seesaw makes with reference to pivot
J_F	Moment of Inertia of the wagon	J_B	Moment of inertia of seesaw
h	Height of the wagon from center of the seesaw	T_B	Load torque on the seesaw (input).
x	Position of the wagon on the seesaw		

MATHEMATICAL MODELS

A mathematical model is required to apply control techniques and stabilize the system. First a non-linear model is made using Euler LaGrange method

NON-LINEAR MODEL

Two matrices q and Q are defined, which correspond to the control variables and input of the seesaw system respectively, as:

$q = [x \quad \alpha]^T$

$Q = [0 \quad T]^T$

First the Kinetic and Potential Energies of the system are calculated.

$E_{\mathbf{kin}} = ½\, m\dot{x}^2 + ½\, J_{\mathbf{total}}\, \acute{\alpha}^2$ (where $J_{\mathbf{total}} = J_{\mathbf{B}} + J_{\mathbf{f}}$) (1)

$J_{\mathbf{F}} = m_{\mathbf{F}}\,(x^2+h^2)$ (2)

$J_{\mathbf{Total}} = \underline{J_{\mathbf{B}} + m_{\mathbf{F}}.h^2} + m_{\mathbf{F}}.x^2$ (3)

In the equation above underlined part is constant whereas $m_F.x^2$ is the varying part.

So $J_B + m_F.h^2 = J_{const}$ (4)

Substituting values from (3) and (4) in (1) we get

$E_{kin} = ½\, m\dot{x}^2 + ½ J_{const}\, \acute{\alpha}^2 + ½\, m_{\mathbf{F}}.x^2\, \acute{\alpha}^2$ (5)

Potential Energy of the system is given as

$E_{pot} = m_{\mathbf{F}}.g.h.\cos\alpha + m_{\mathbf{F}}.g.x.\sin\alpha$

Partially differentiating (E_{kin} - E_{pot}) and simplifying the equation we get the following equations

$$\ddot{\alpha} = \frac{1}{J_{\text{konst}} + m_{\text{F}} \cdot x^2}\tau_{\text{B}} - \frac{m_{\text{F}} \cdot g \cdot x}{J_{\text{konst}} + m_{\text{F}} \cdot x^2}\cos(\alpha) + \frac{m_{\text{F}} \cdot g \cdot h}{J_{\text{konst}} + m_{\text{F}} \cdot x^2}\sin(\alpha) - \frac{2 \cdot m_{\text{F}} \cdot x}{J_{\text{konst}} + m_{\text{F}} \cdot x^2}\dot{x}\dot{\alpha}$$

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

LINEAR MODEL

The non-linear equations are linearized by defining a working boundary at

$$\dot{\alpha} = 0, \alpha = 0, \dot{x} = 0, x = 0, \tau_{\text{B}} = 0$$

We get the following linear equations:

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

$$\ddot{\alpha} = \frac{1}{J_{\text{konst}}} \tau_B - \frac{m_F \cdot g}{J_{\text{konst}}} x + \frac{m_F \cdot g \cdot h}{J_{\text{konst}}} \alpha$$

SIMULINK MODELS

Simulink models which correspond to the mathematical models are made using gain blocks for multiplication and trigonometric functions for cosine and sine

NON-LINEAR MODEL

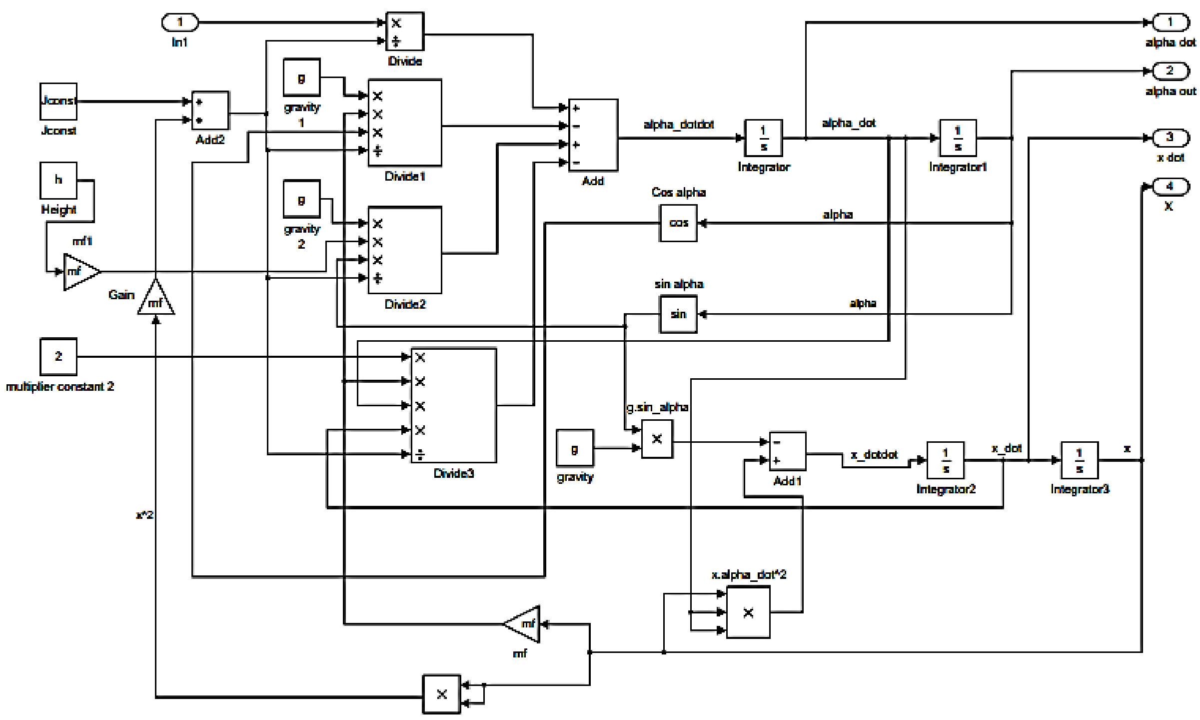


Figure 2 Non- linear model of the seesaw system

LINEAR MODEL

The linear model is made using state space block in SIMULINK®. The state matrices are given as:

$$\underbrace{\begin{bmatrix} 0 & \frac{m_F \cdot g \cdot h}{J_{\text{konst}}} & 0 & -\frac{m_F \cdot g}{J_{\text{konst}}} \\ 1 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \frac{1}{J_{\text{konst}}} \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}}$$

C is an identity matrix of 4x4 and D is a null vector.

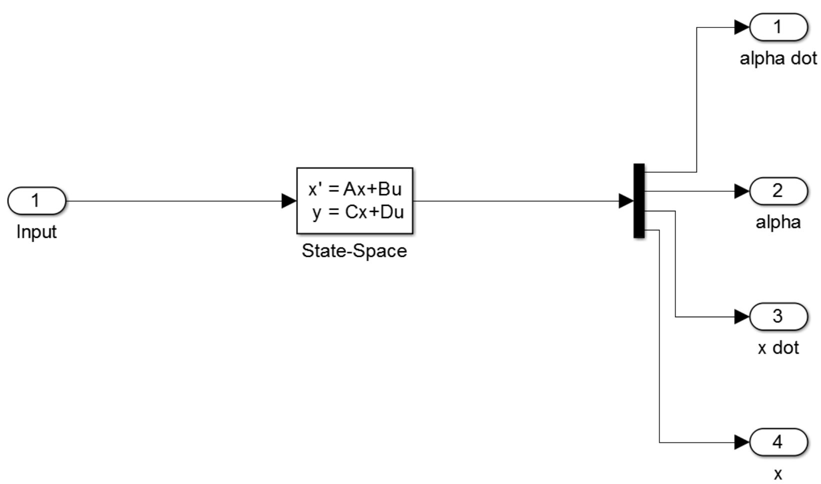


Figure 3: A linear model of the seesaw system

Step input is applied to both systems as shown

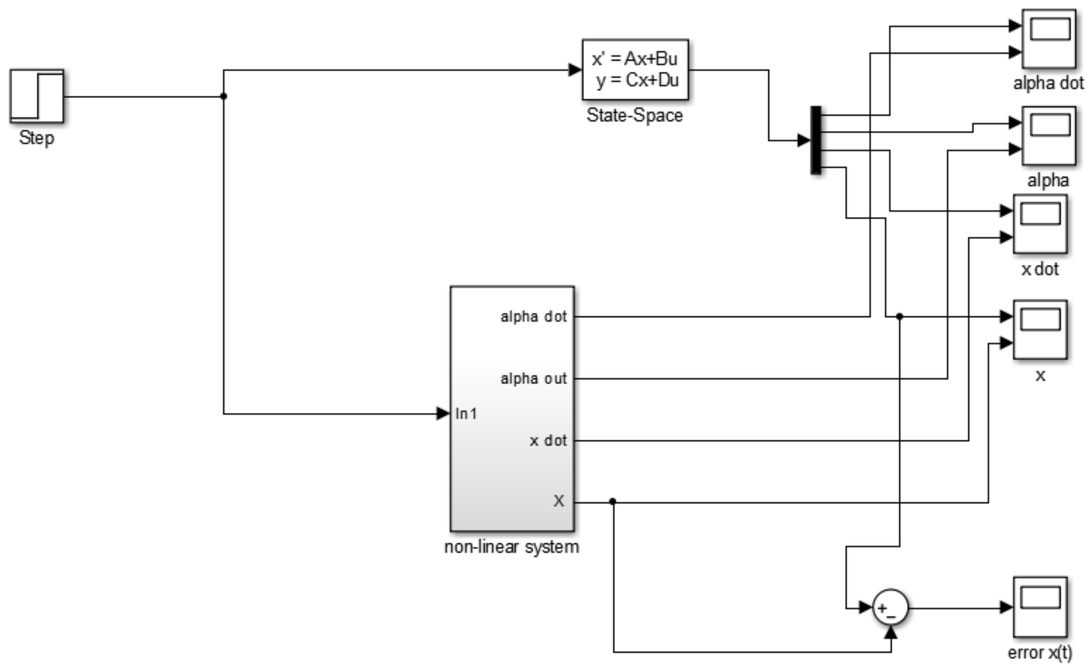


Figure 4 Step input applied to linear and non-linear models

Step response of the linear and non-linear model is shown

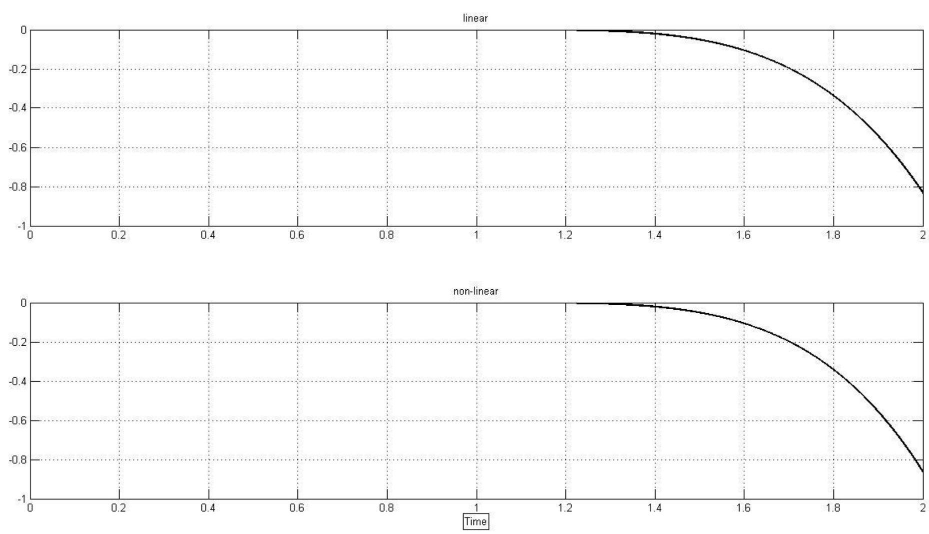


Figure 5 Linear (top) vs non-linear (below) state output x (linear displacement)

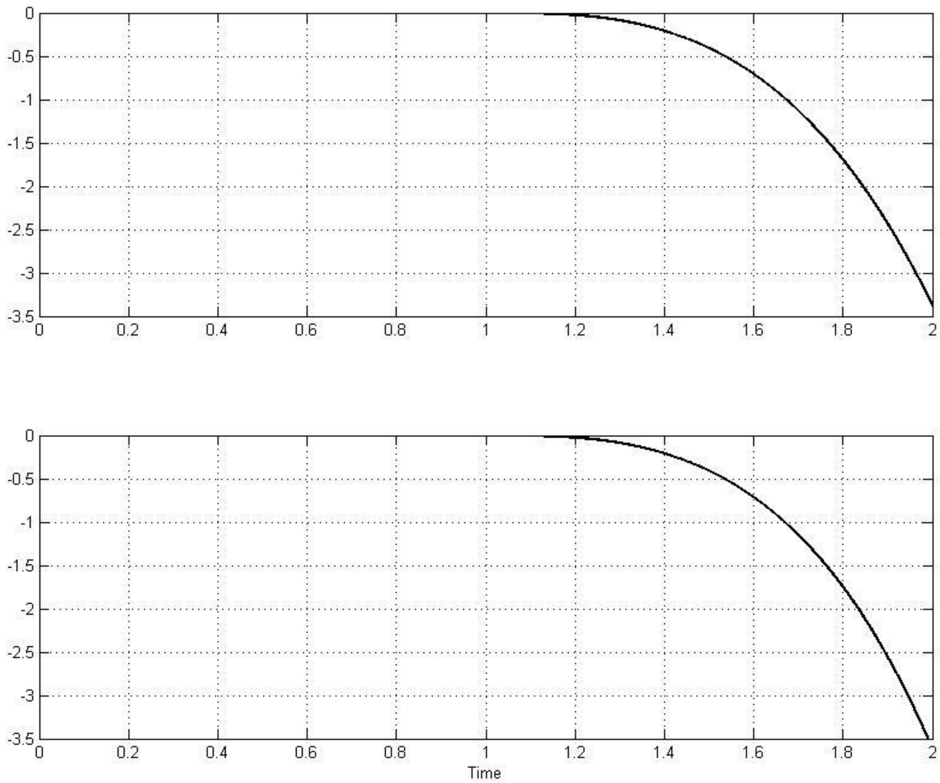


Figure 6 Step response x dot (linear acceleration)

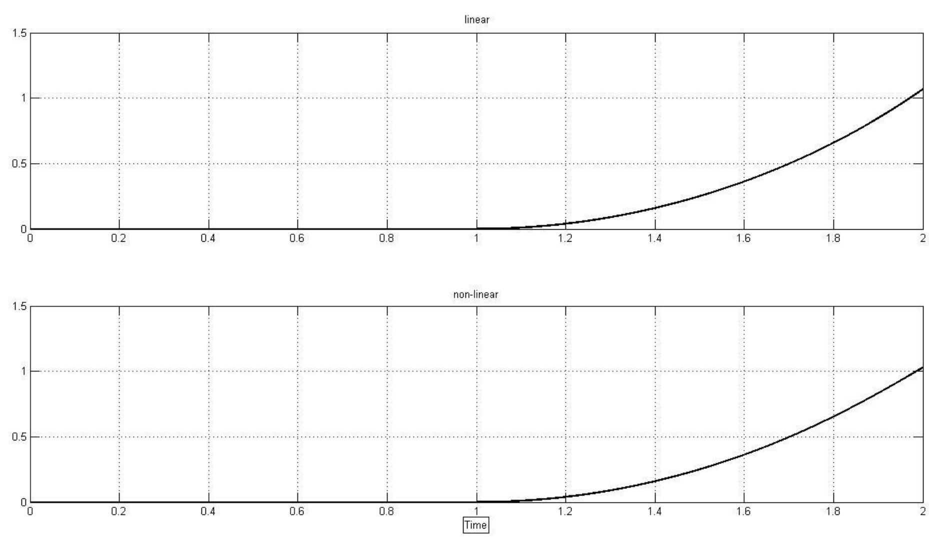


Figure 7 State output alpha (angular displacement)

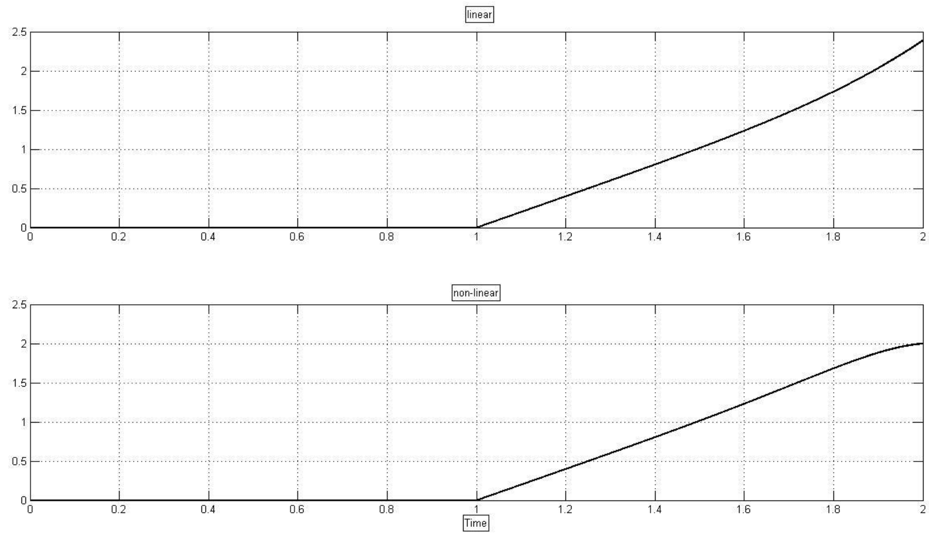


Figure 8 State output alpha dot (angular acceleration)

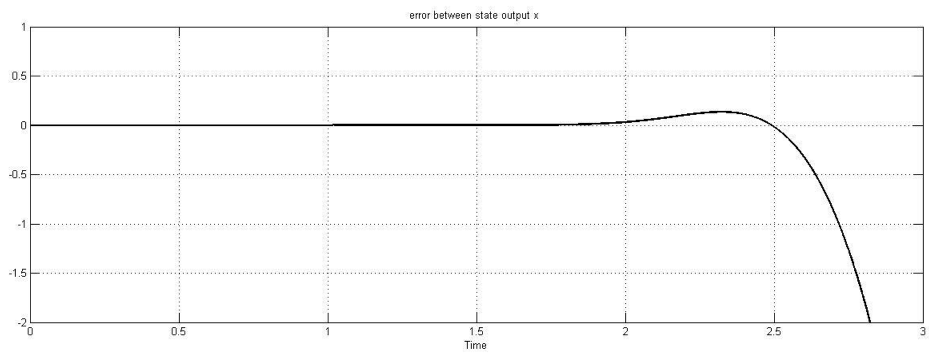


Figure 9 Error between linear and non-linear state output x

STATE FEEDBACK

A proportional controller with state feedback and pre amplifier is used to stabilize both the linear and non-linear system as shown in the figure below:

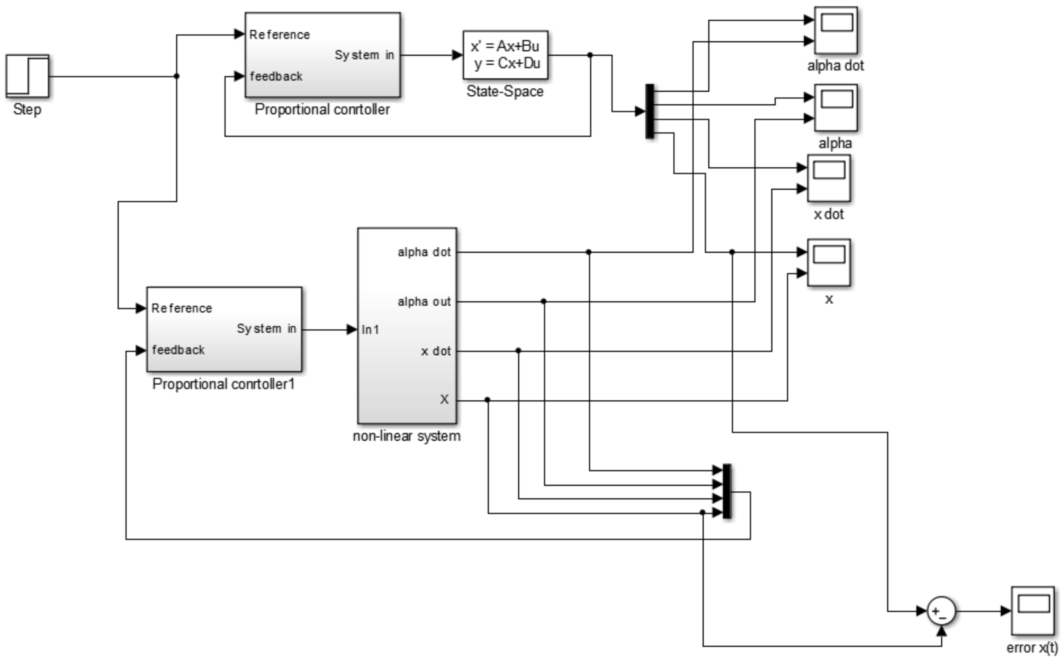


Figure 10 Proportional controller with state feedback

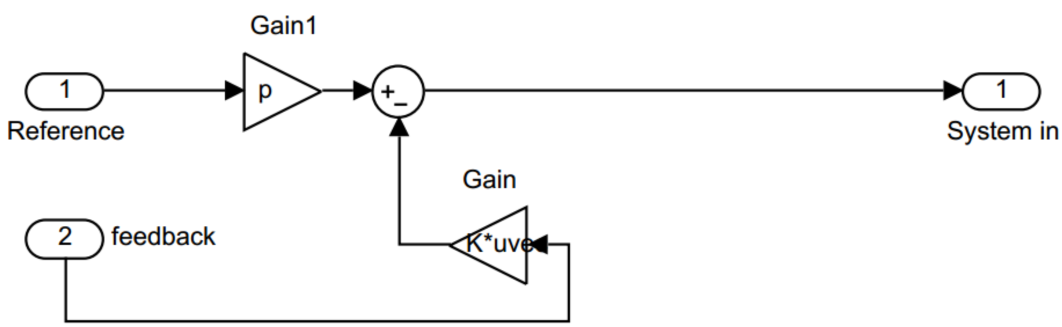


Figure 11 Proportional controller subsystem

The proportional state feedback gain (K) is computed using Ackermann formula and Pre-amplifier (p) is calculated by the formula $P = 1/(C\,(B*K - A)^{-1}*B)$ where K is proportional state feedback gain.

Step responses of linear and non-linear system re shown below:

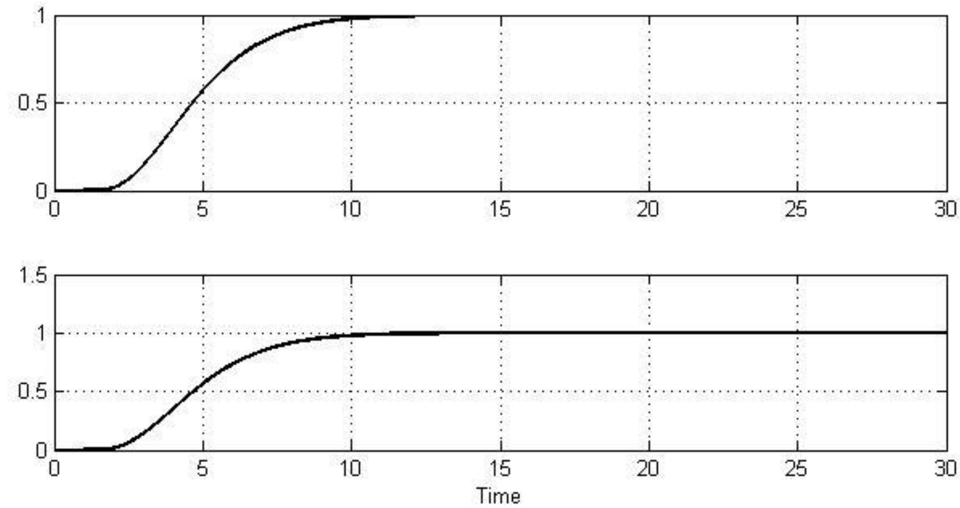


Figure 12 linear (top) vs non-linear (bottom) state output x (linear displacement)

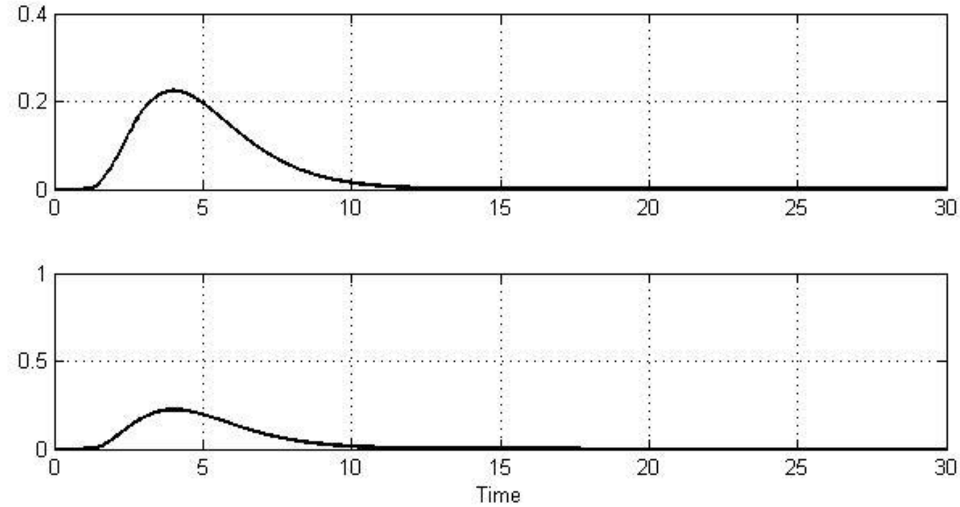


Figure 13 State output x dot (linear acceleration)

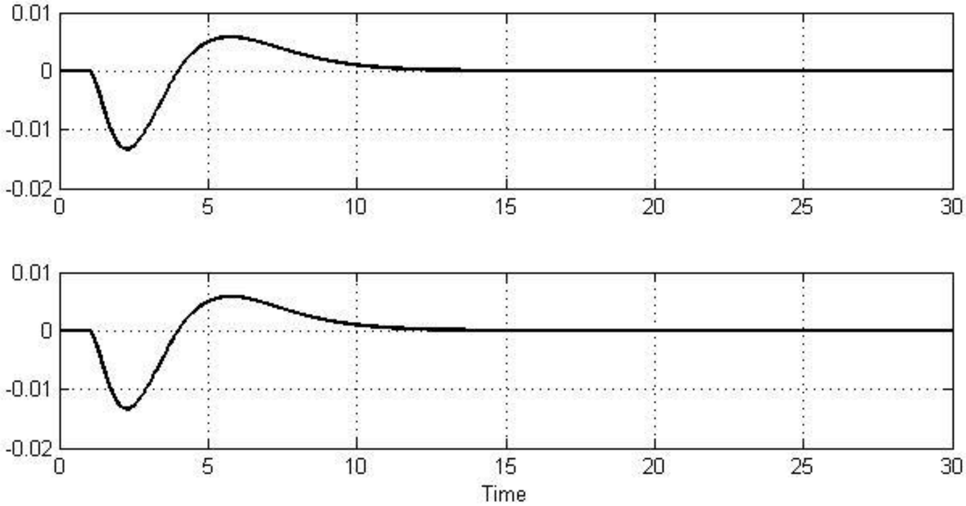


Figure 14 State output alpha (angular displacement)

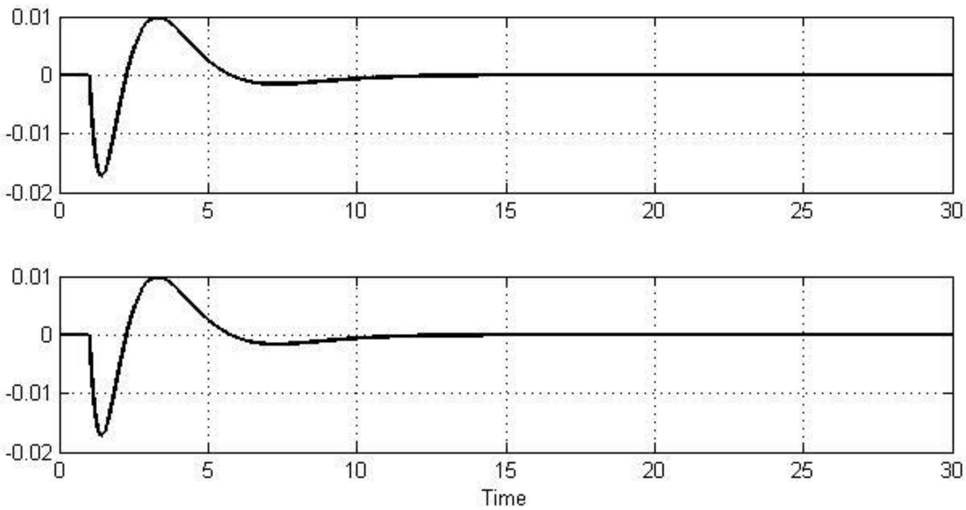
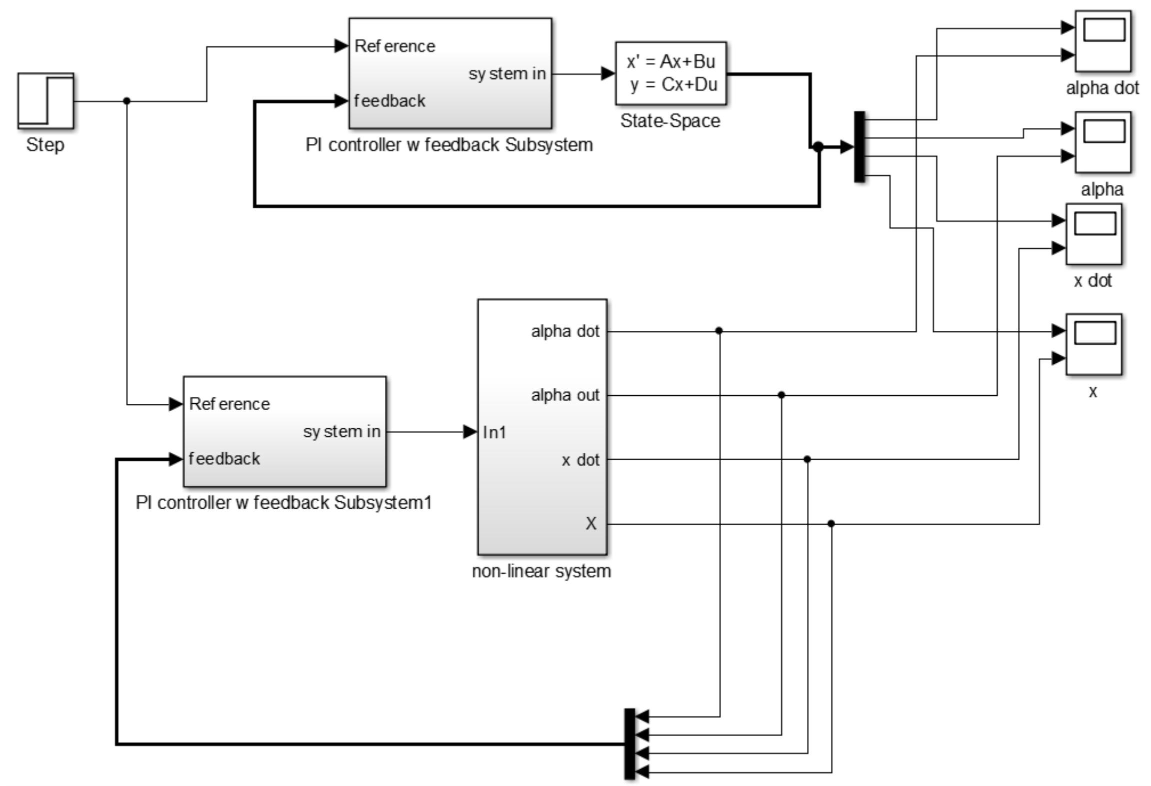


Figure 15 State output alpha dot (angular acceleration)

PI CONTROL

Matrices A and B are augmented resulting in increase in order by 1. Ackermann formula is used again to calculate the new state feedback gain so that the first 4 elements of ackerman's resultant are values of K whereas the last value is -PI. PI controller is used with system and output state feedback to contol the system as shown below.



TIME CONTINUOUS

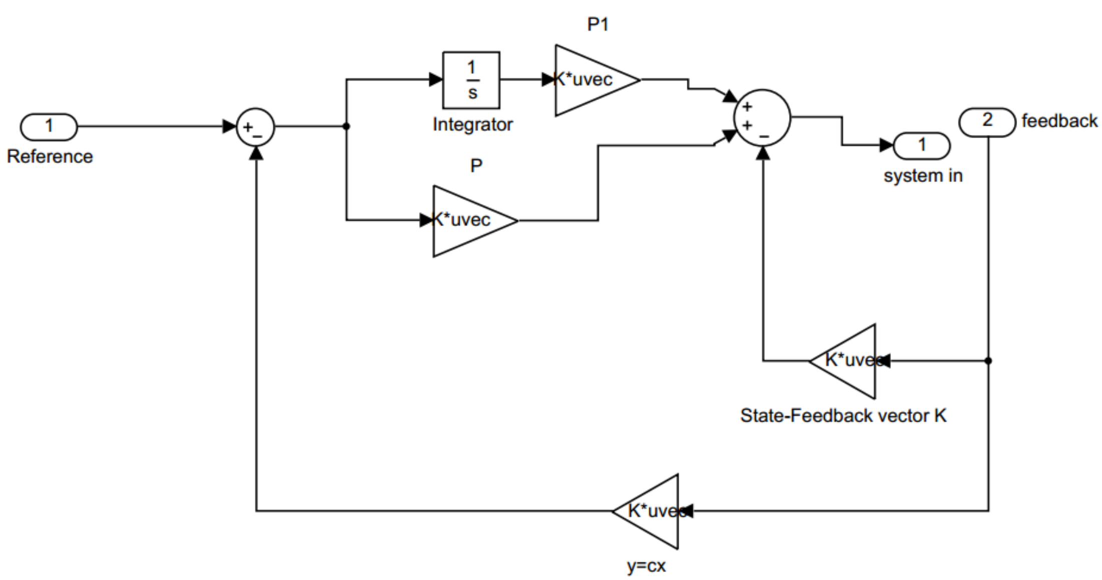


Figure 17 Time Continuous PI controller subsystem.

Step response of the system controlled with PI controller with P- part = 0 are as follows:

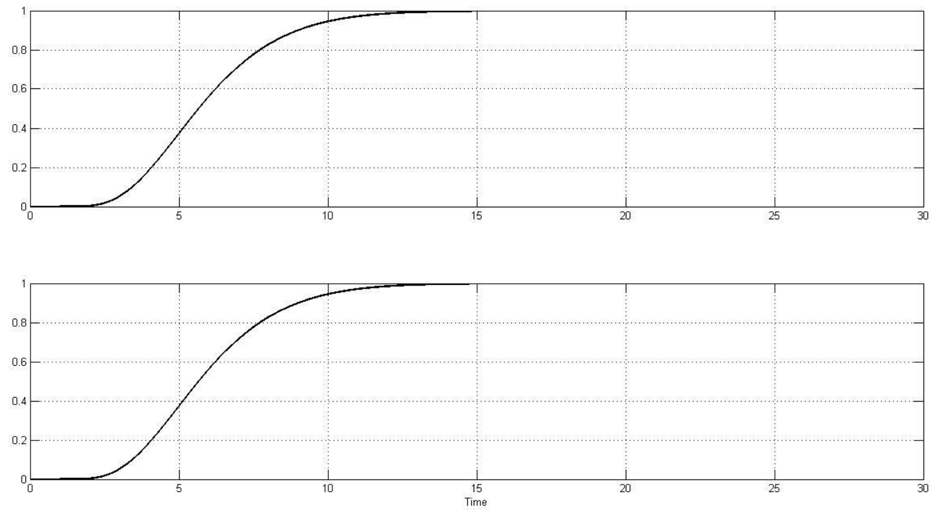


Figure 18 Step response of PI (time continuous) controlled output x (linear displacement)

Other parameters are similar to the proportional controller responses the only difference is that PI controlled system is resilient towards any disturbances where the proportional controller cannot cope with any load or disturbances.

Pre amplifier is the same as for proportional controller as it is not affected by an output feedback as shown on page number 308 (Ludyk, 1995). Proportional state feedback is corrected using the equation K = k-pc also shown on page 309 (Ludyk, 1995)

TIME DISCRETE

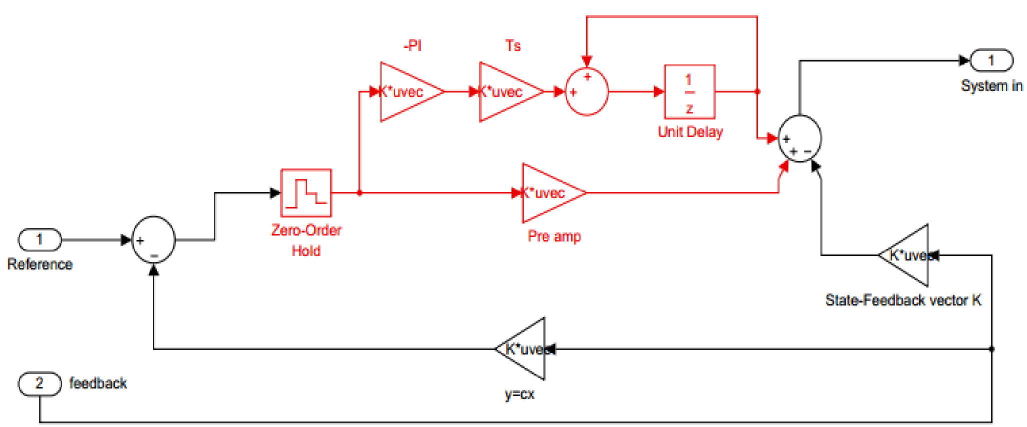


Figure 19 Time discrete PI controller subsystem.

The PI is made part time discrete as shown in the figure above with a sampling frequency of 8 kHz.

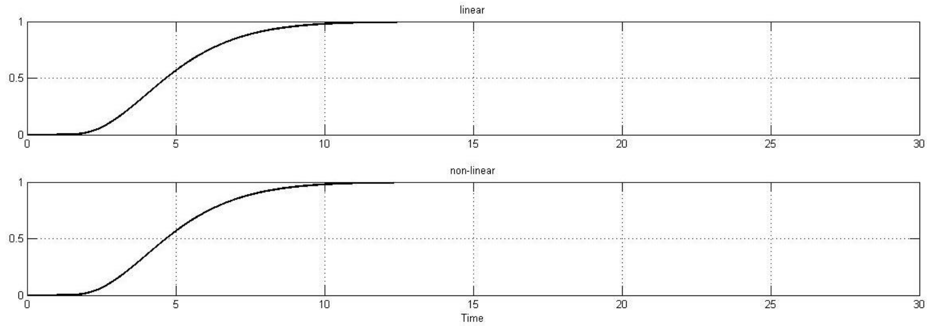


Figure 20 Time discrete PI controller state x (linear displacement)

STATE OBSERVER

Measuring states and making calculations for control is not always possible. Furthermore this approach requires a large number of sensors. The problem with using multiple sensors is their limitations and need for re calibration over time which in turn can be costly. An alternate approach, which is more cost effective, is the use of state observers. State observers require only the system output and have the ability to re-construct states, which can be used to make calculations for proportional, PI feedback and any other feedback controller which requires state feedback. The block diagram below shows how state observers are implemented in Simulink for our seesaw system.

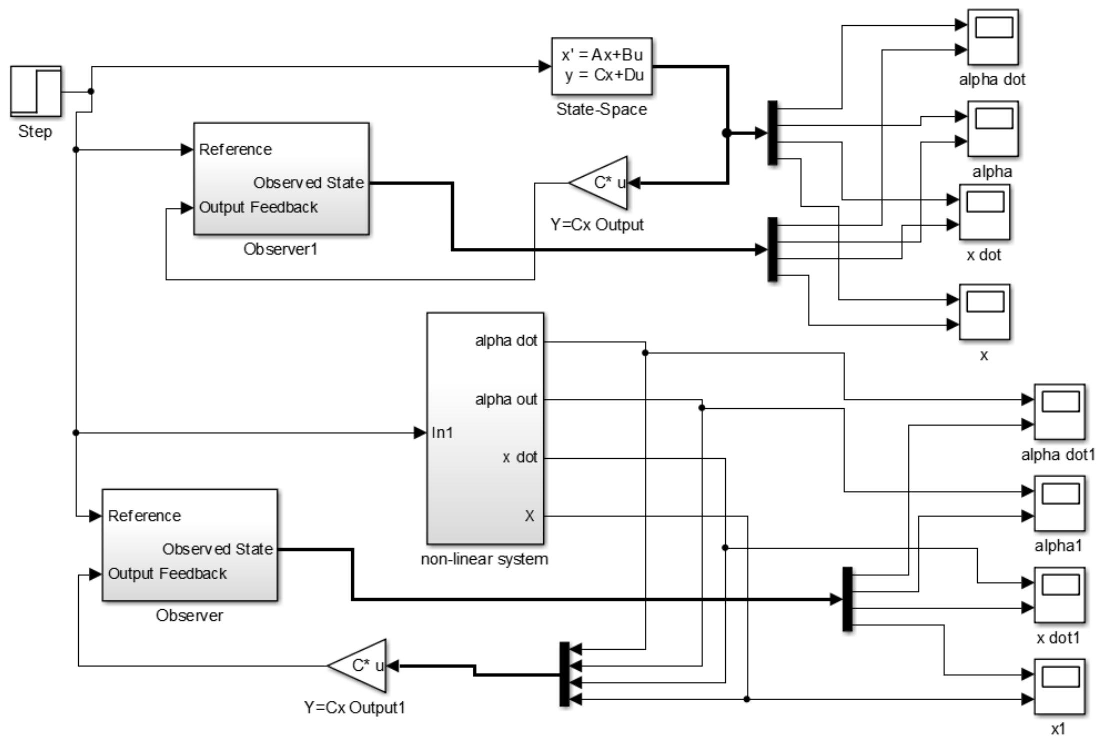


Figure 21 Full State observers are used to reconstruct states and these reconstructed states are compared with actual states.

TIME CONTINUOUS

The diagram below shows the construction of full state observer in time continuous domain.

The calculation of H is similar to the method adopted for calculating proportional Gain . i.e. Ackermann filter, however with a slight modification. The modification is using transpose of matrix C and A in the formula for matlab.

H = acker(A.', C.' , poles)

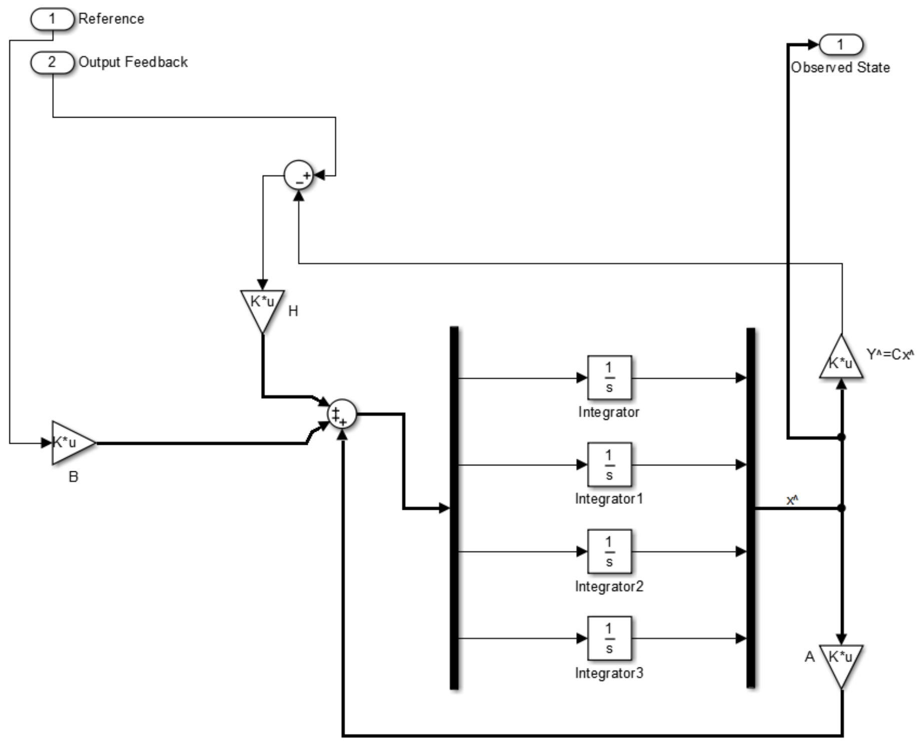


Figure 22: Time continuous full state observer subsystem

Following are the Actual states compared with states observed by the observer:

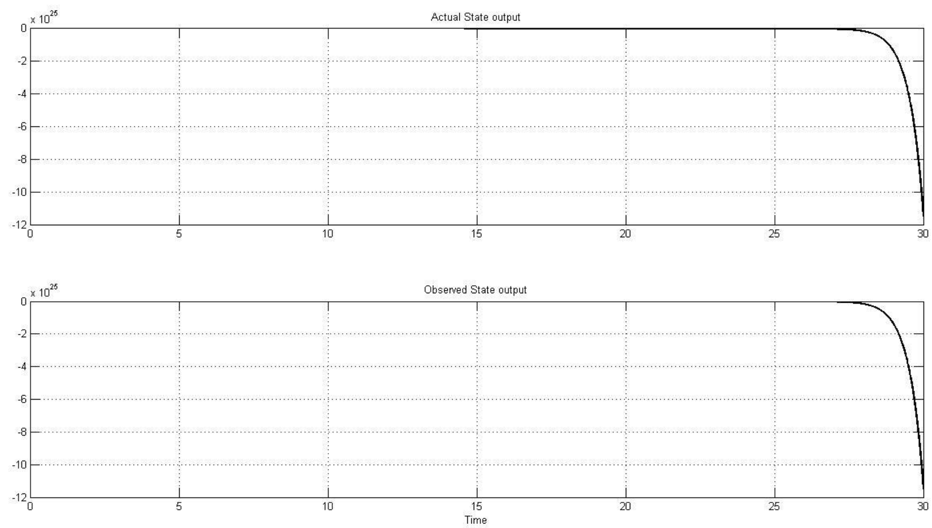


Figure 23 Time continuous observer: Actual vs Observed State x (linear displacement)

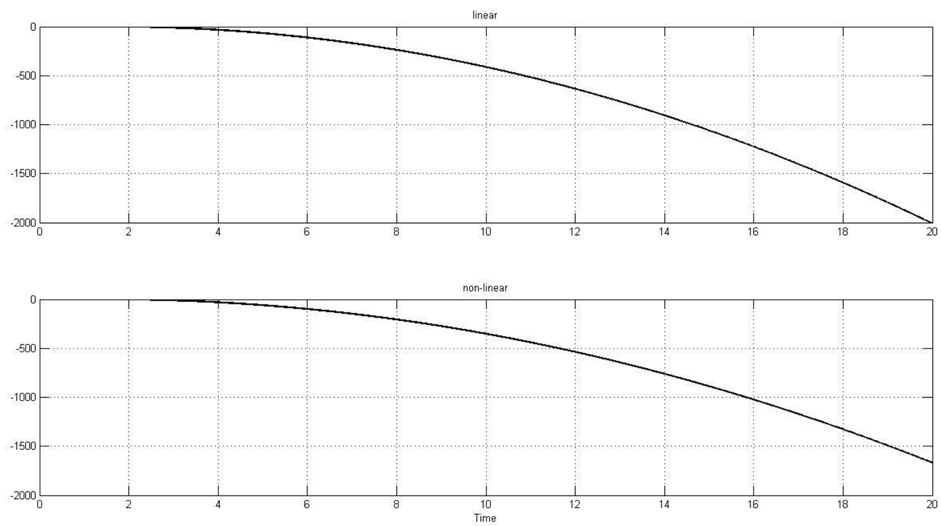


Figure 24 Time continuos observed vs actual state x (linear displacement) for non linear system

TIME DISCRETE

The difference between continuous and discrete time observer is that for discrete observer the poles need to be converted to z-domain.

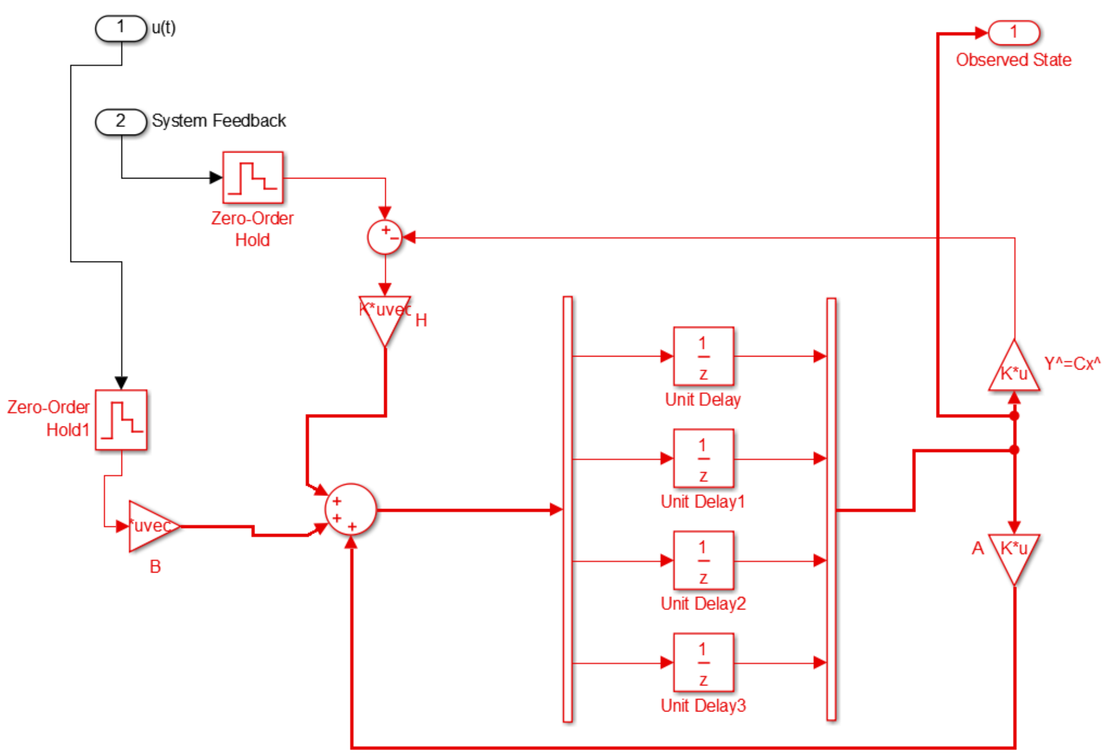


Figure 25 Time discrete full state observer

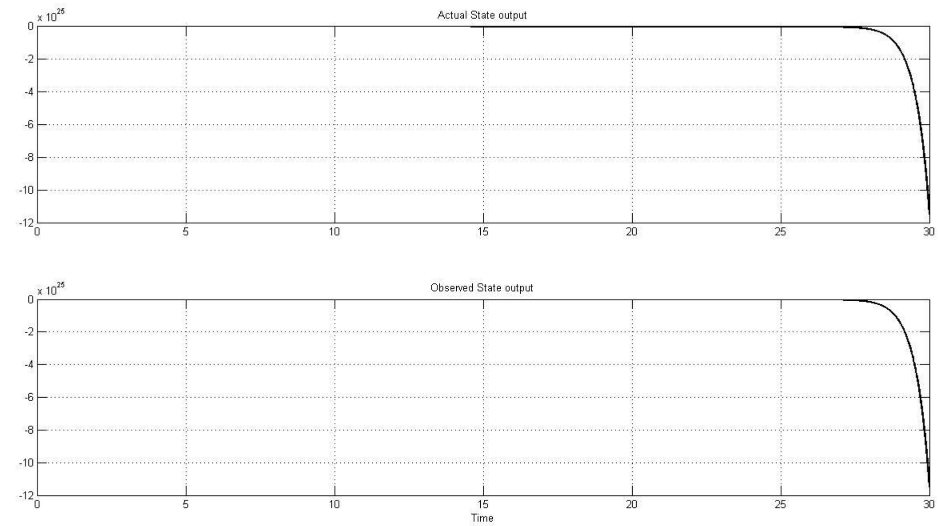


Figure 26 Time discrete full state observer actual vs observed state x (linear displacement) for linear system

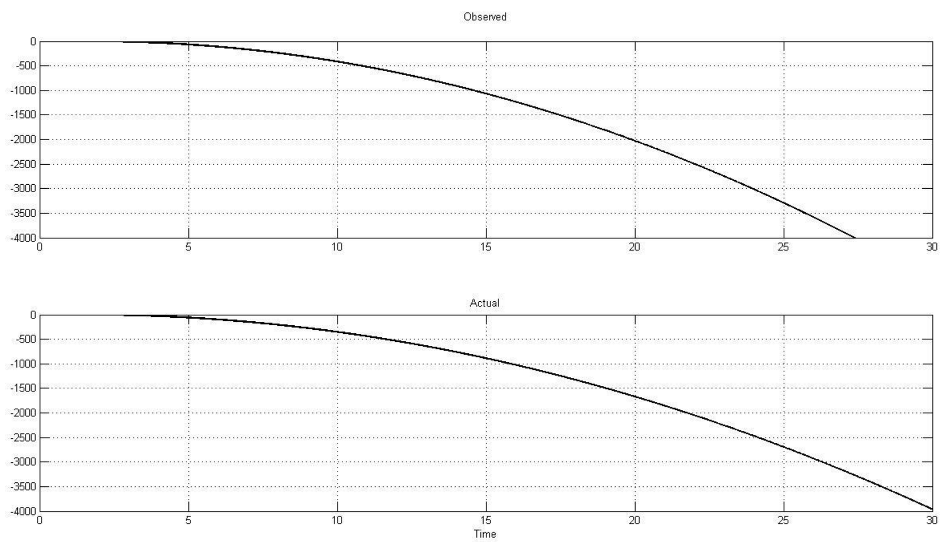


Figure 27 Time discrete full state observer actual vs observed state x for non-linear system

PROPORTIONAL CONTROLLER USING OBSERVED STATES

A proportional controller is designed using states from the observer and then connected with the system to be controlled. Block diagram is shown below:

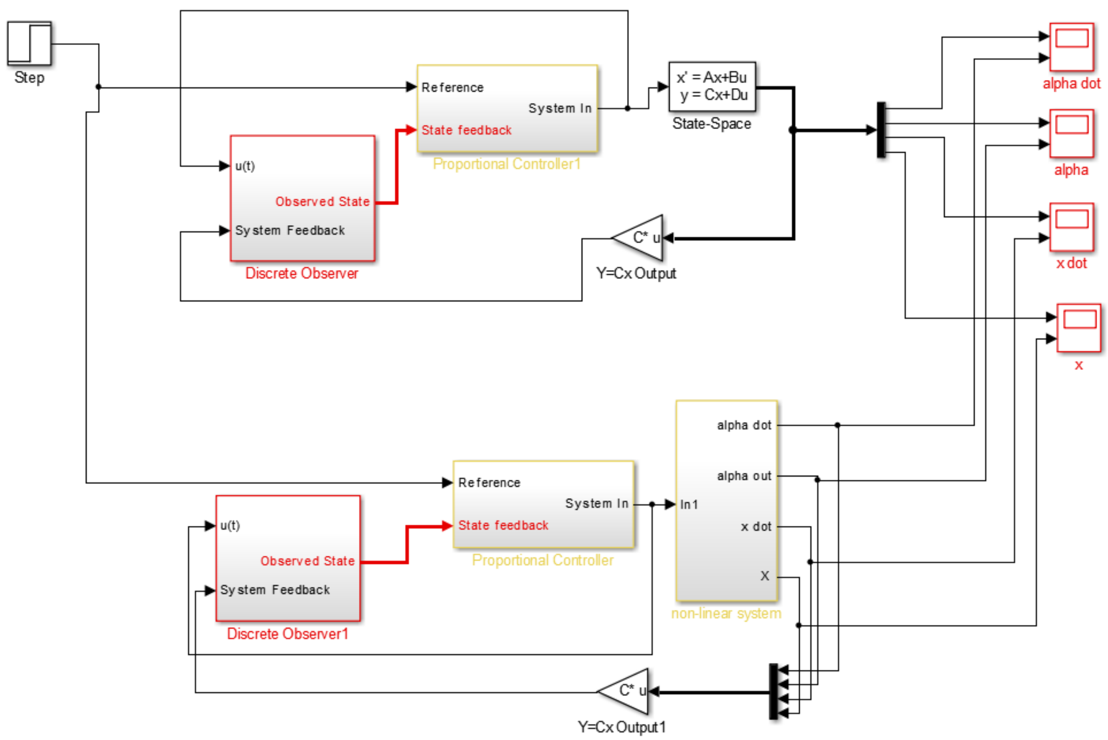


Figure 28 Proportional control on linear and nonlinear system using states from observer

The output of linear and non-linear system are shown below:

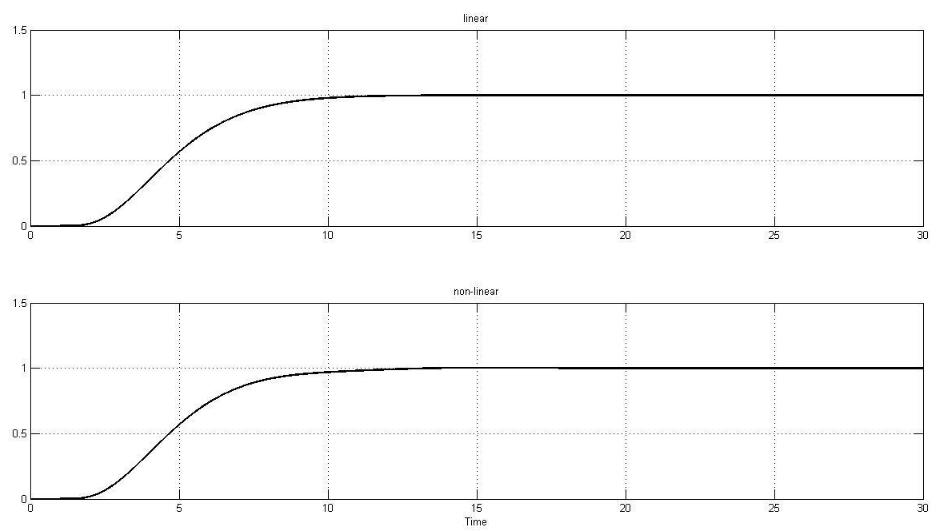


Figure 29 Proportional controller step response using full state discrete observer

PI CONTROLLER USING OBSERVED STATES

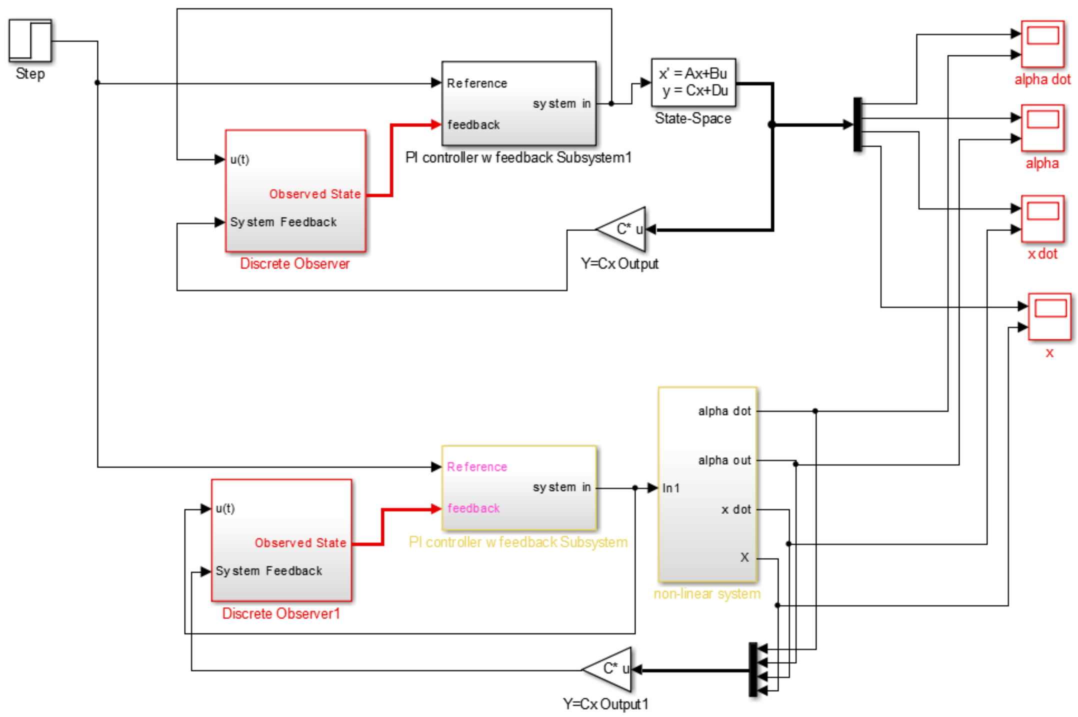


Figure 30 Time continuous PI controller with state feedback from time discrete state observer

Step response of linear and non-linear system is shown below.

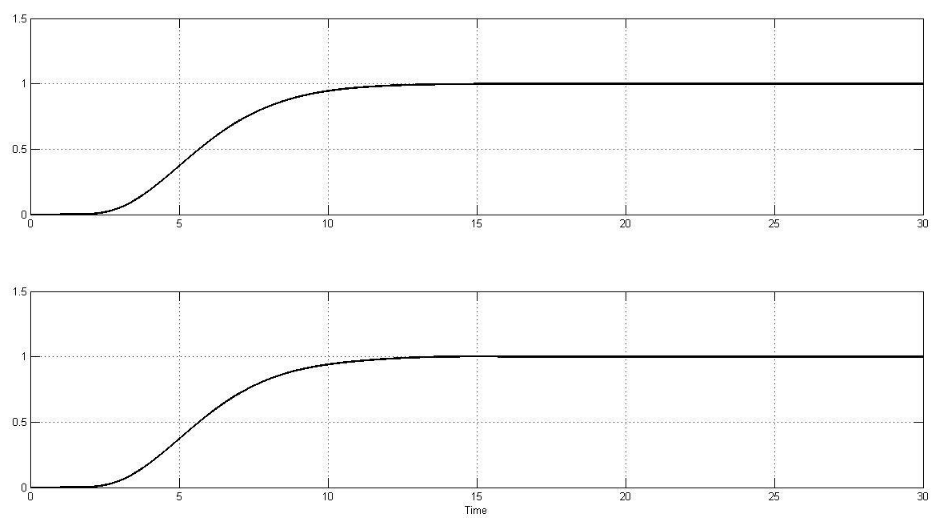


Figure 31 Step response to a time continuous PI controlled system with full state time discrete observer

Time discrete PI controller can also be used with discrete state observer. The state diagram of one such system is given as:

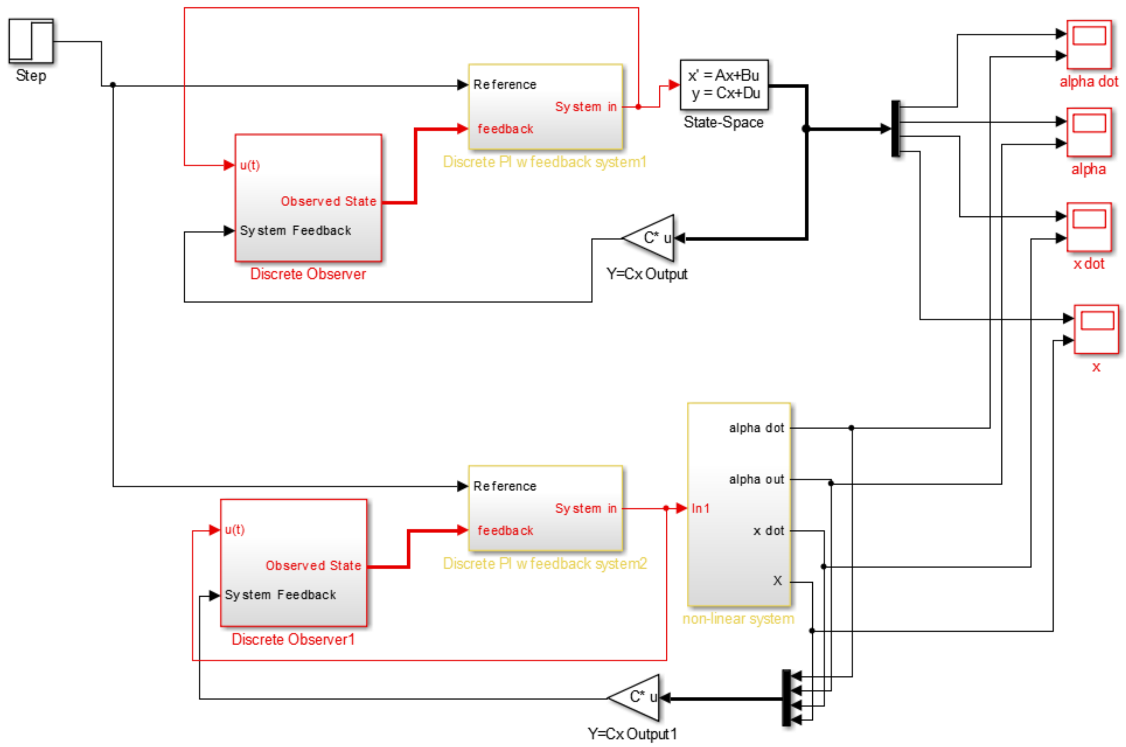


Figure 32 Time discrete PI controller with a time discrete state observer

Step response of a time discrete PI controlled system is similar to time continuous time PI controlled system

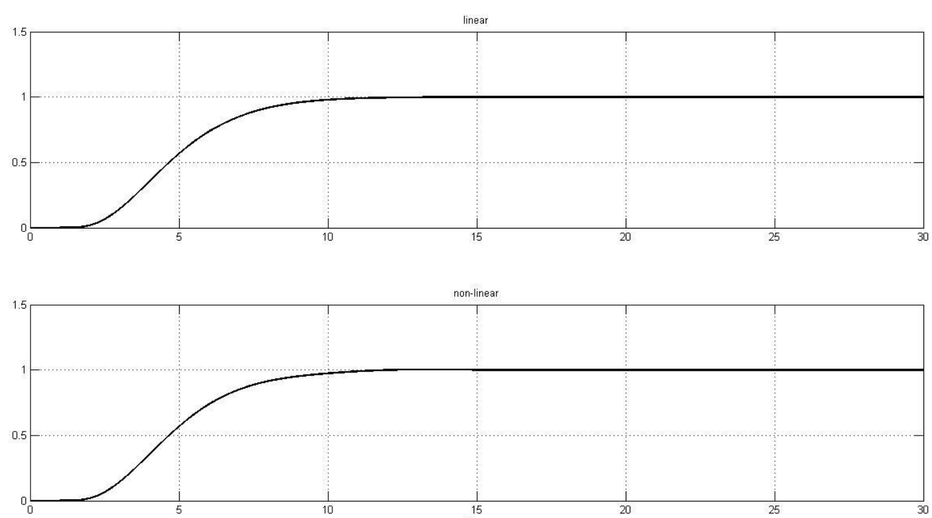


Figure 33 Step response of time discrete PI with time discrete observer

A system with different initial condition as compared to the state observer gives the following step response:

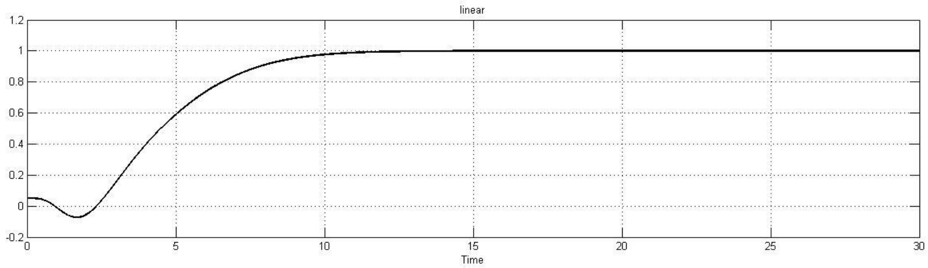


Figure 34 Step response of PI controlled system with different initial conditions as compared to observer

The step response shows that the observer is capable of overcoming any difference in initial condition i.e. it can stabilize an already unstable system or bring an unstable system to stable position. The initial dip can be attributed to observer training.

MATLAB® CODE

```
%---------------------------- Seesaw.ini ----------------------------%
%--------------------------------------------------------------------%
clear all
clc
%--------------------------------------------------------------------%
%---- Initial declarations for both linear and non-linear system ----%
%--------------------------------------------------------------------%

Jb = 0.5;                                   % Moment of Inertia.
mf = 0.1;                                   % Mass of the wagon.
h = 0.1;                                    % Height of the wagon.
g = 9.81;                                   % Gravitational acceleration.
Jconst = Jb + mf*h*h;                       % Jconst

%--------------------------------------------------------------------%
%---- Matrices defined for linear system State Space Model ----------%
%--------------------------------------------------------------------%

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;0;0;0];

%--------------------------------------------------------------------%
%------------------- Pole placement at -1 ---------------------------%
%--------------------------------------------------------------------%

p0 = 1;
pol = [-p0,-p0,-p0,-p0];

%--------------------------------------------------------------------%
%--- Calculation of state feedback vector using ackermann formula ---%
%--------------------------------------------------------------------%

k=acker(A,B,pol);
C = [0 0 0 1];                  % New C matrix for calculating pre-amp.
```

```
p = 1/(C*((B*k-A)^-1)*B);   % Pre-amplifier for simple state feedback.
sts = ss(A,B,c,D);
%------------------------------------------------------------------%
%-------------------- PI calculations ------------------------------%
%------------------------------------------------------------------%

A_hat =vertcat([A zeros(4,1)],[-C 0]); %make A^ = |A  0| and B^= |B|
B_hat = vertcat(B,0);                  %           |-C 0|         |0|
pol1 = [-p0,-p0,-p0,-p0,-p0];          % Pole placement @ -1.
K_hat = acker(A_hat,B_hat,pol1);       % Ackermann for new K and p.
K2 = K_hat(1:4);                       %    K
PI = K_hat(5);                         % Proportional gain for PI.
k2 = K2-p*C;                           % State feedback correction.

%------------------------------------------------------------------%
%------------------------ Zeitdiskrete -----------------------------%
%------------------------------------------------------------------%

Ts = 1/8000;                           % Sampling Time
sysd =c2d(sts,Ts,'zoh');               % Continuous to Discrete.
[Ad,Bd,Cd,Dd] = ssdata(sysd);          % Discrete State Space Matrices.

%------------------------------------------------------------------%
%------------------- Zustandsbeobachter ----------------------------%
%------------------------------------------------------------------%

pp = 3;                                % Pole Placement @ -3
pol2 = [-pp,-pp,-pp,-pp];
ct = C.';
HT = acker(A.',ct,pol2);               % Ackermann for Observer
H = HT.';

%------------------------------------------------------------------%
%------------------- Time Discrete Observer ------------------------%
%------------------------------------------------------------------%

syms s;                                % Symbol 's' for equation.
expr = (s+pp)^4;                       % (s+3)^4.
den =expand (expr);                    % expand expression.
tfden = sym2poly(den);
num = 1;
sysp = tf(num,tfden);                  % Transfer function 1/(s+pp)^4.
sysdp = c2d(sysp,Ts);                  % Continuous to discrete
[z,polt,kk] = zpkdata(sysdp);          % Poles extracted from discrete
                                       % transfer function.
pol = cell2mat(polt);                  % Convert cell to row matrix
Htd = acker(Ad.',ct,pol.');            % Ackermann
Hd = Htd.';

%------------------------------------------------------------------%
%------------------------ END OF CODE ------------------------------%
%------------------------------------------------------------------%
```

REFERENCES

Ludyk, G. (1995). *Theoretische Regelungstechnik 1 Grundlagen, Synthese linearer Regelungssysteme.* Bremen,Newyork,Berlin: Springer.