THE UNIVERSITY OF TEXAS AT ARLINGTON, TEXAS
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 5327 - 001

SYSTEM IDENTIFICATION & ESTIMATION

Project #1
EXAM

by

SOUTRIK MAITI

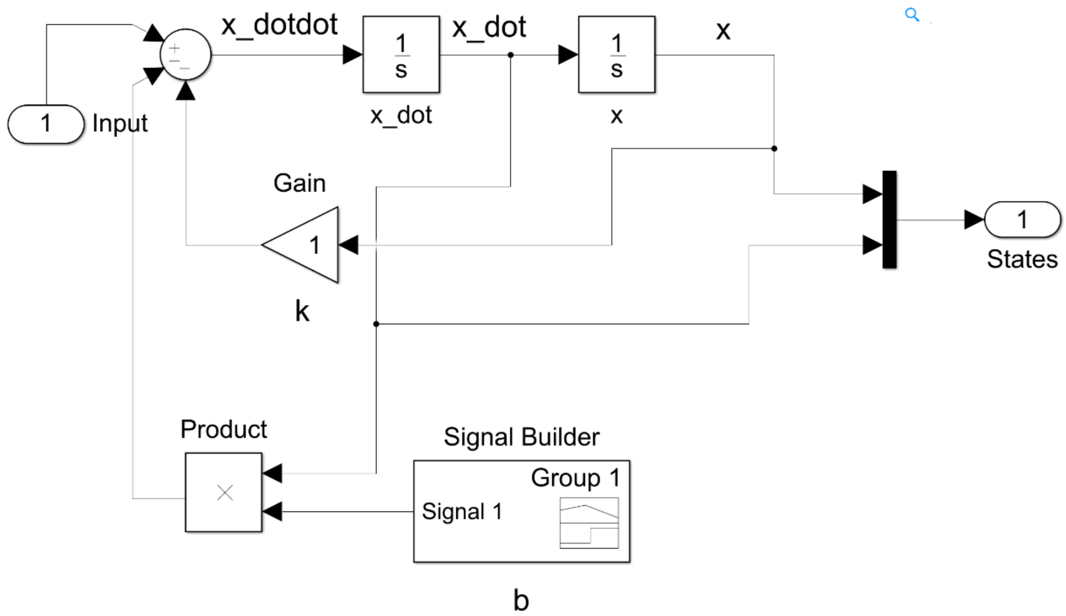
1001569883

Presented to

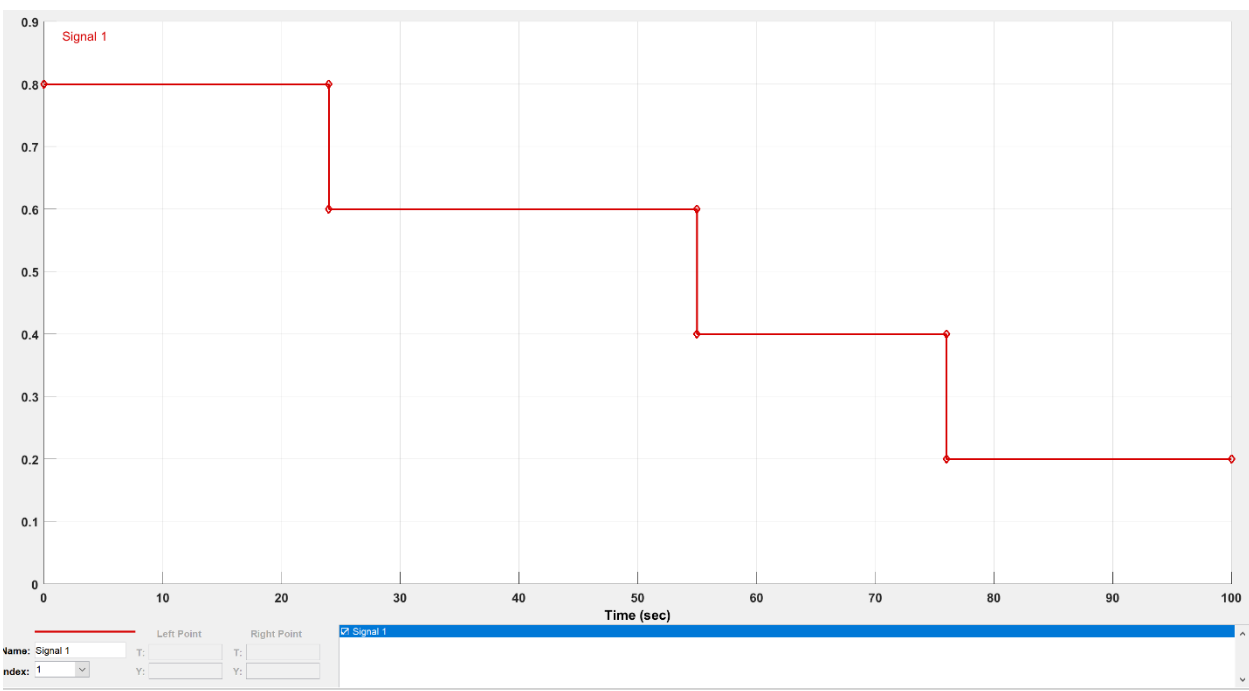
Prof. Michael Niestroy

Oct 25th, 2017

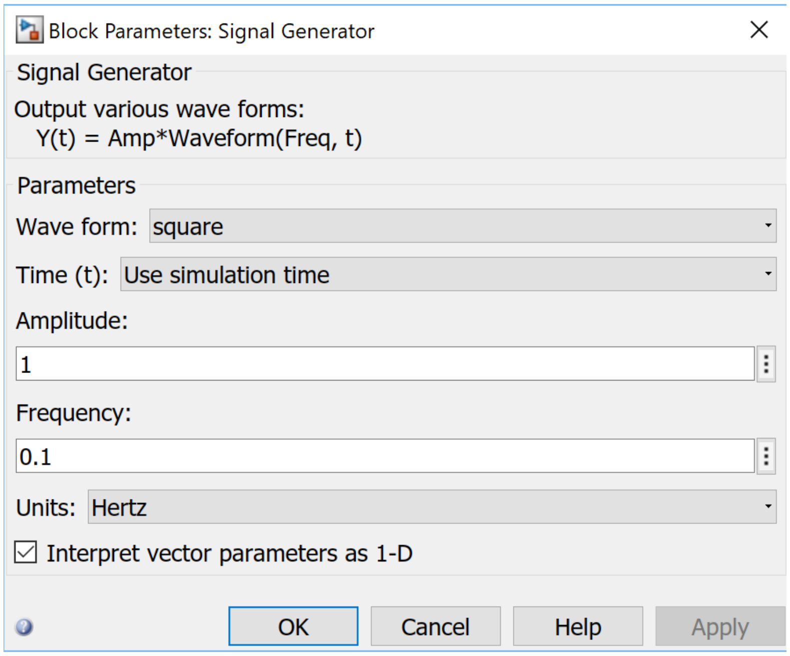
1) Implementing the 2nd order differential equation



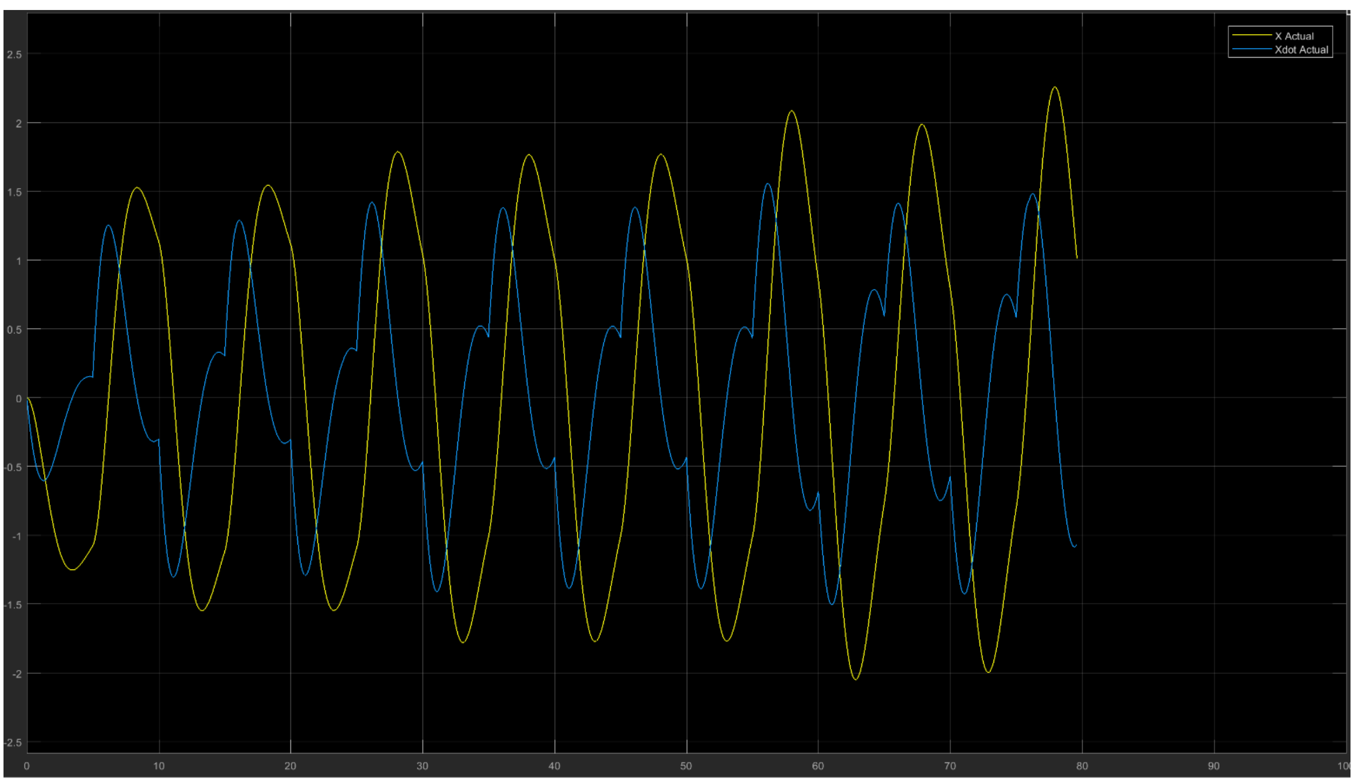
A signal generator is used to give the system, the values of b at different time.



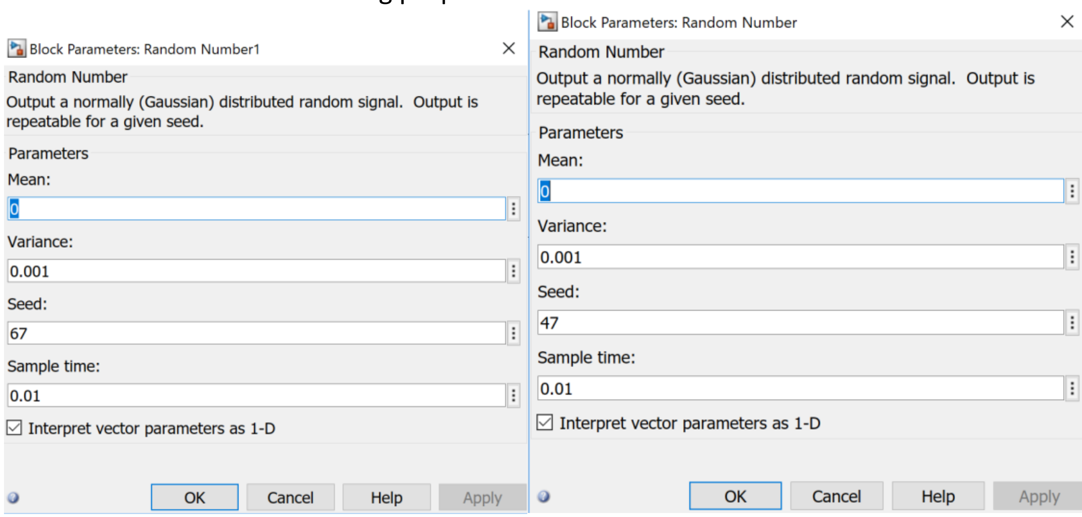
The block parameters of the square wave used as an input to the system.

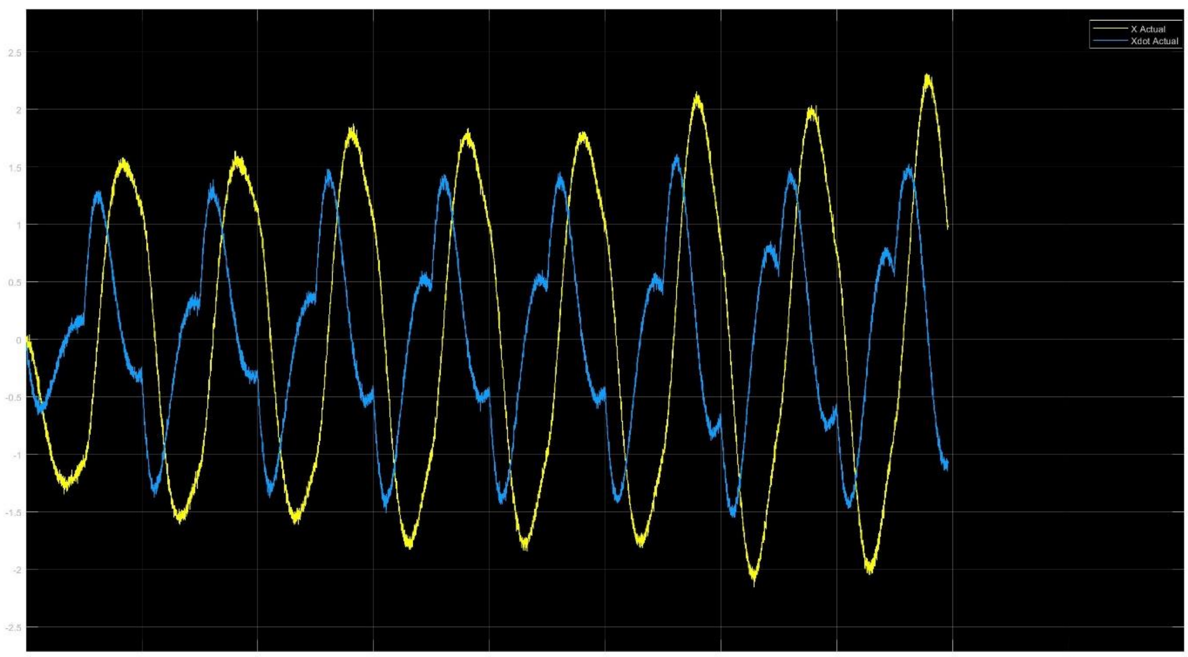


Output of the plant – The actual states of the system

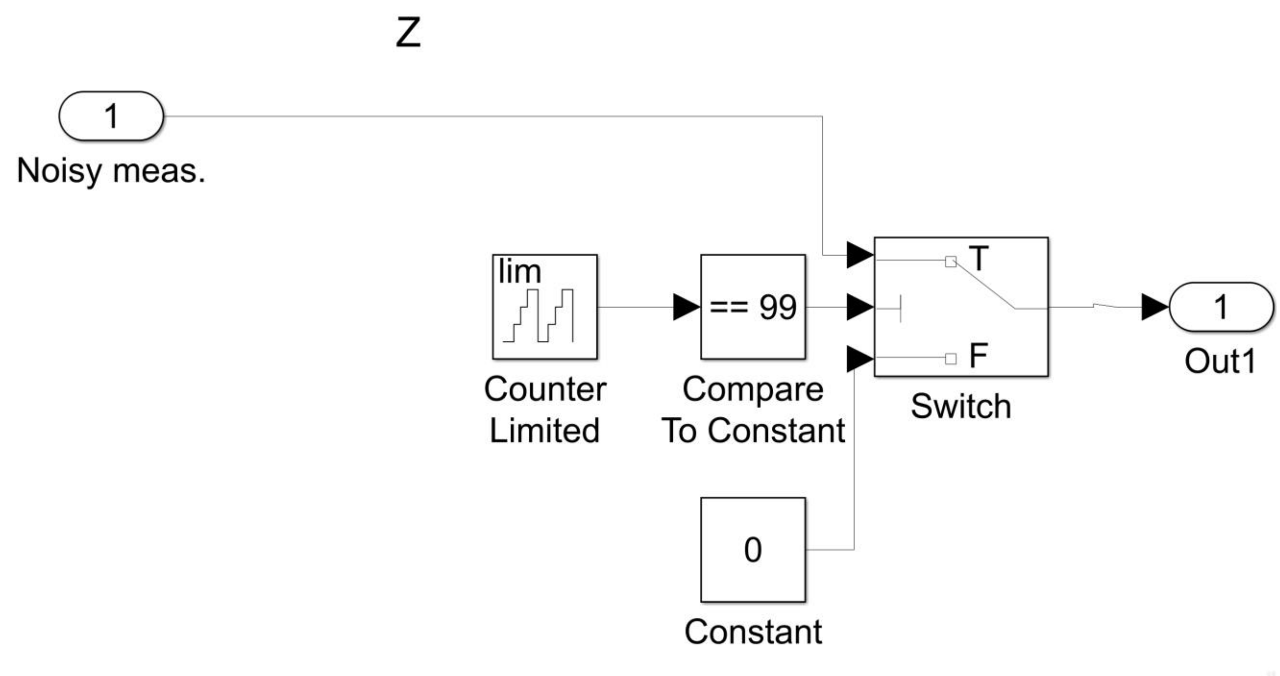


2) Noisy output after adding random noise to the output.
The noise added have the following properties :

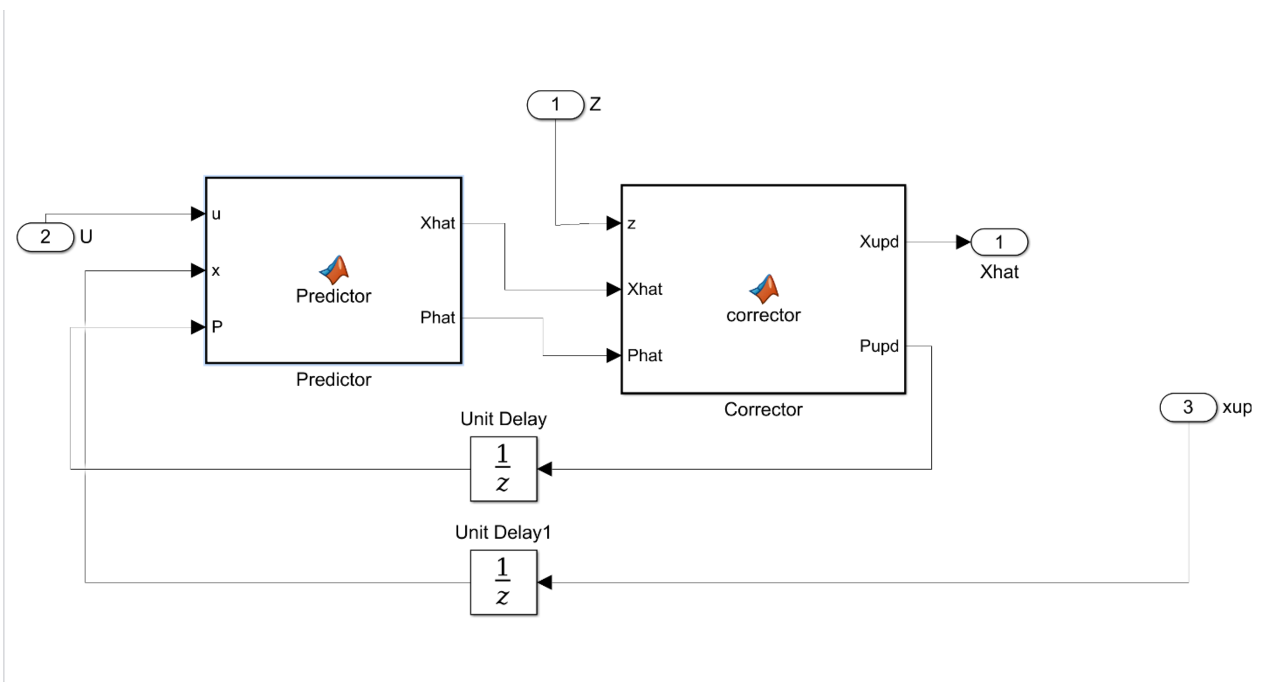




The use of limited counter lets the Kalman filter run at 1Hz.



3) The following shows figure shows a single Kalman filter bank.



MATLAB Code for the Kalman filter (where b=0.8) :-

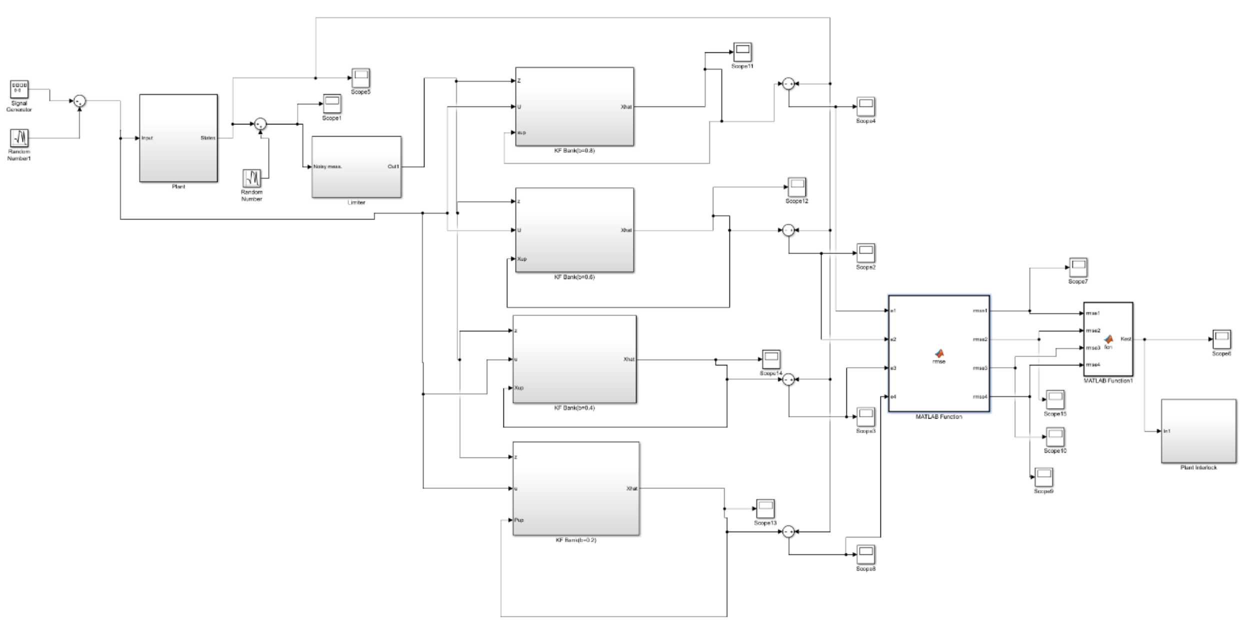
Predictor block

```
function [Xhat,Phat] = Predictor(u,x,P)
Q=10*eye(2);b =[0; 1];
a=[0 1;-1 -.8]; %[0 1;-k -b]
ad=expm(a*0.01);
bd=inv(a)*(ad-eye(2))*b;
f=[-a Q;
   zeros(2,2) a'];
g=expm(f*0.01);
Qd=g(3:4,3:4)'*g(1:2,3:4);
Xhat=ad*x+bd*u;                   %State prediction
Phat=ad*P+ad'*Qd;                 %Covariance prediction
End
```

Corrector block

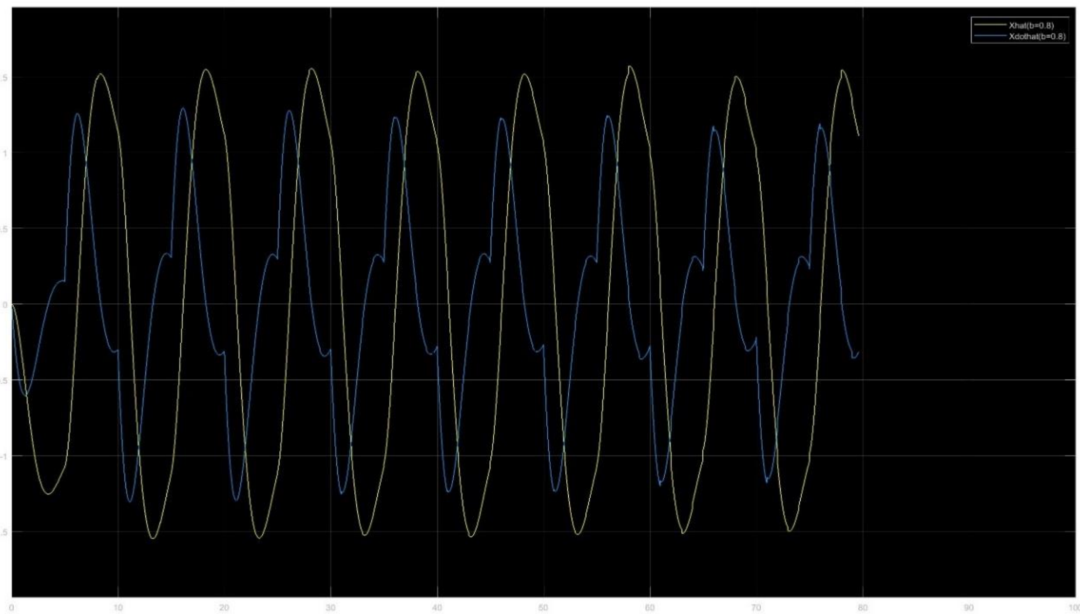
```
function [Xupd,Pupd] = corrector(z,Xhat,Phat)
h=eye(2);
r=eye(2);
if z(1) == 0 && z(2) == 0
    Xupd = Xhat;
    Pupd = Phat;
else
    rd= 1 /0.01 * r;
    K=Phat*h'*inv(h*Phat*h'+rd);
    Pupd=(eye(2)-K*h)*Phat*(eye(2)-K*h)'+K*rd*K';
    Xupd=Xhat+K*(z-h*Xhat);
end
end
```

Block diagram of the entire plant

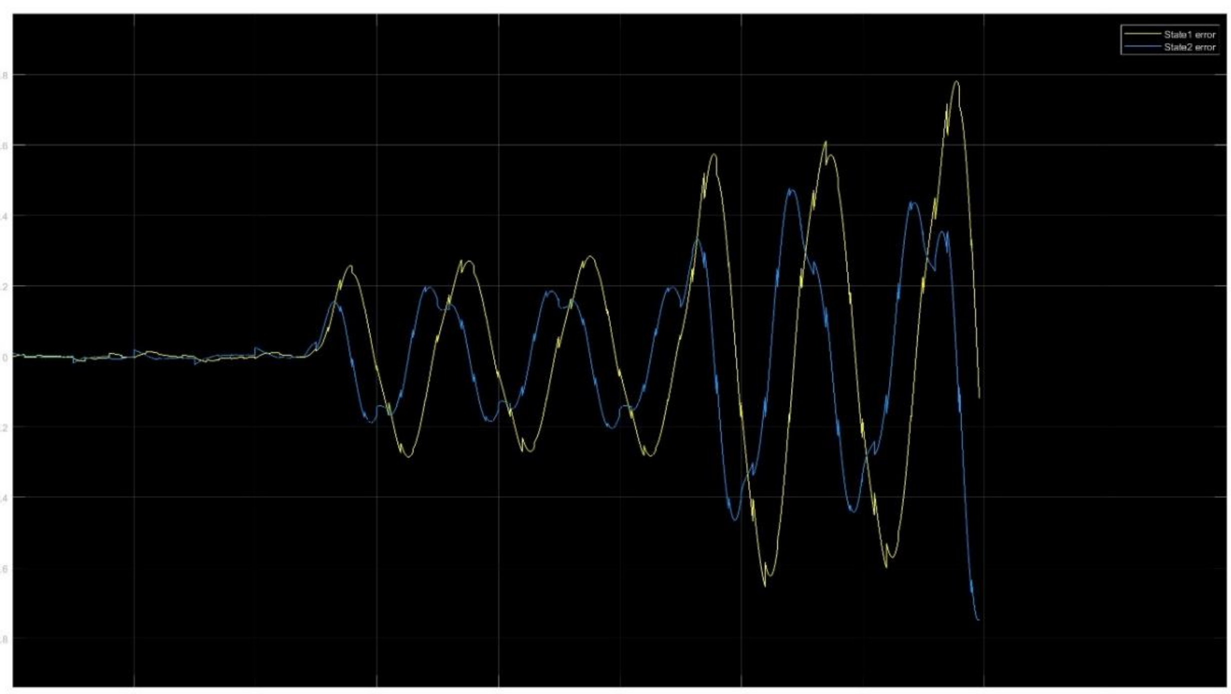


The values of Q and R selected are: Q=10*eye(2), R= eye(2).

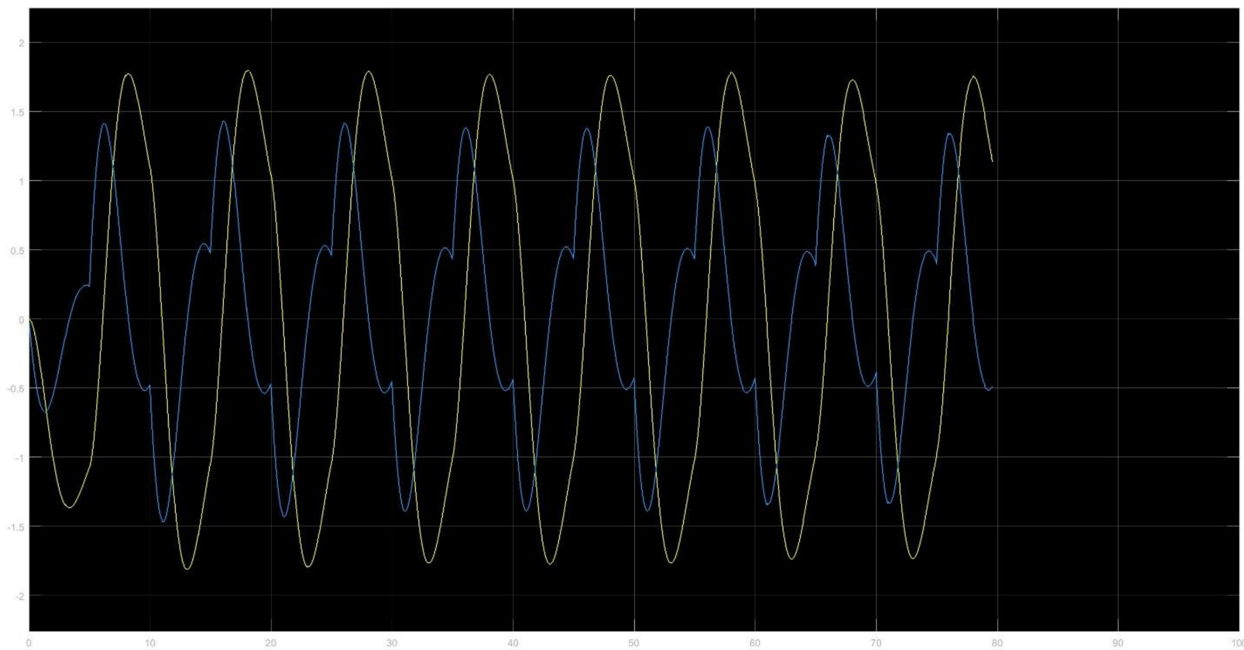
b=0.8



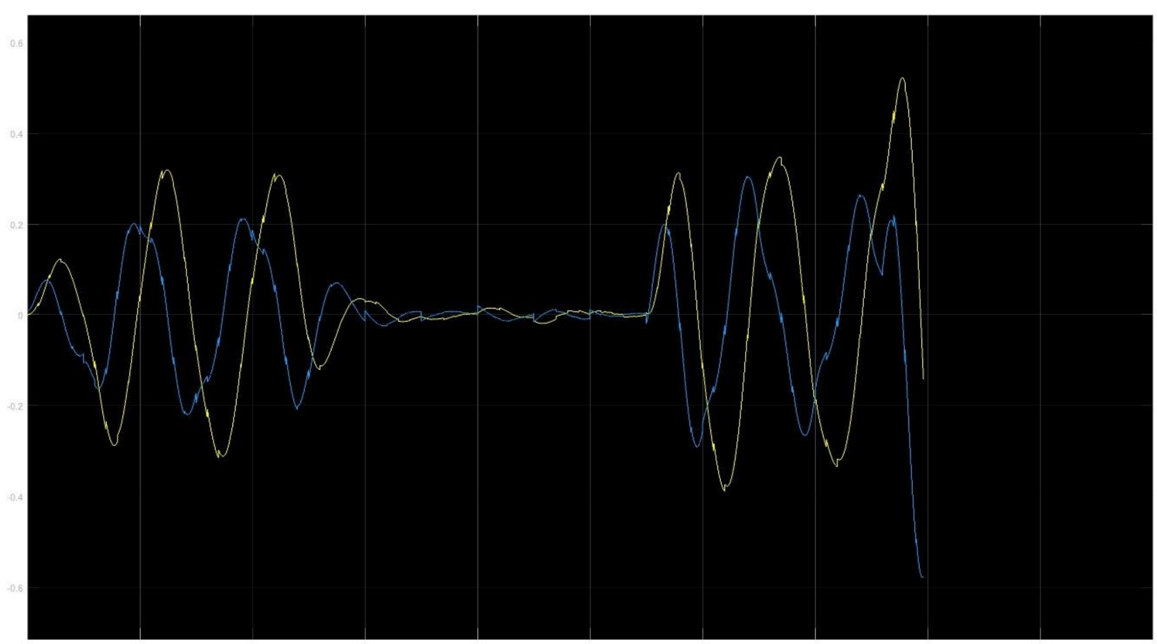
Error (b=0.8)



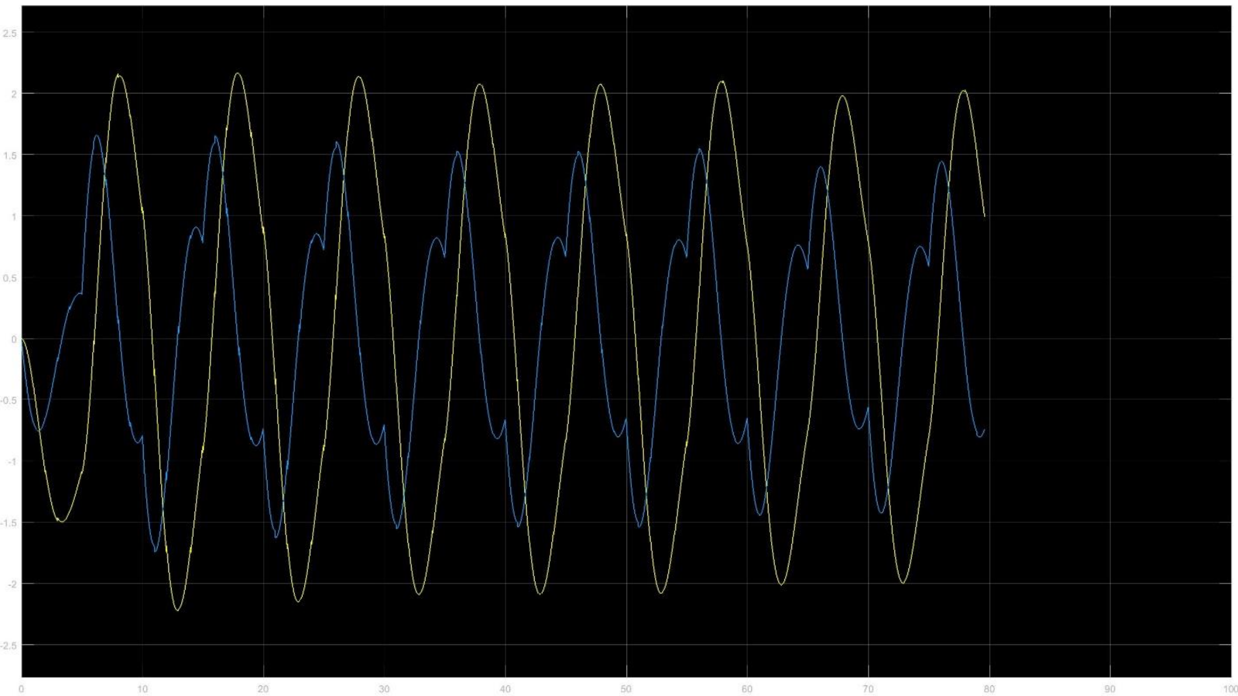
b=0.6



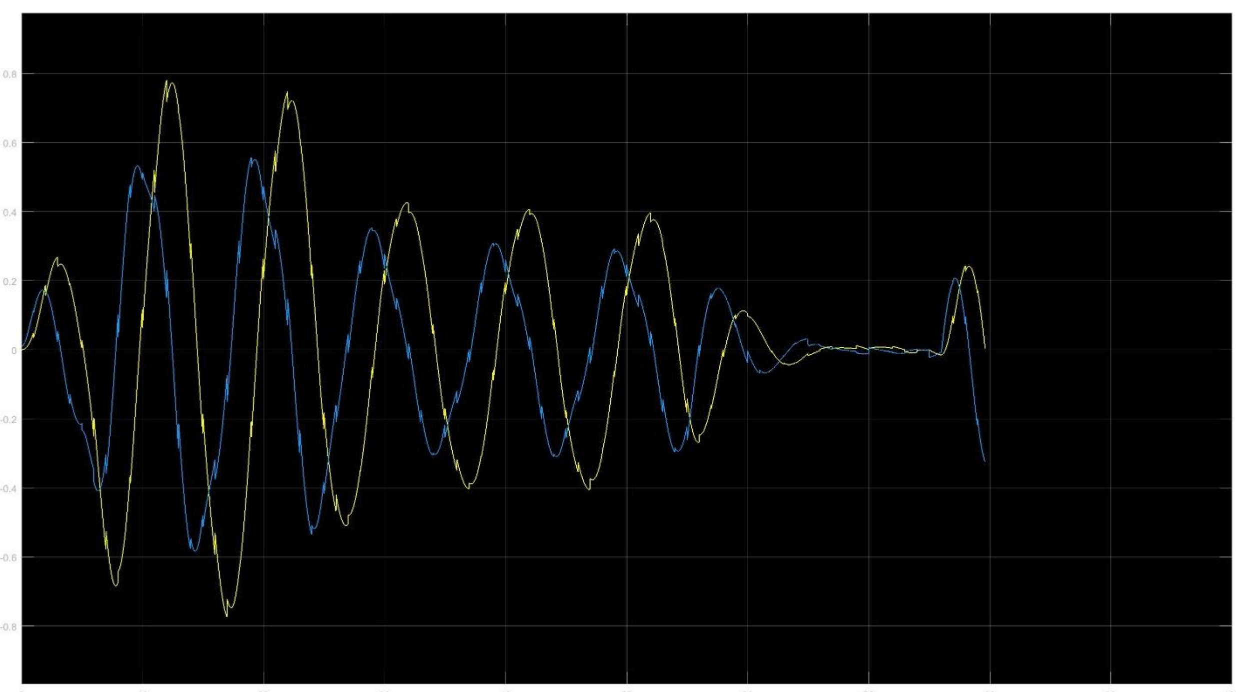
Error (b=0.6)



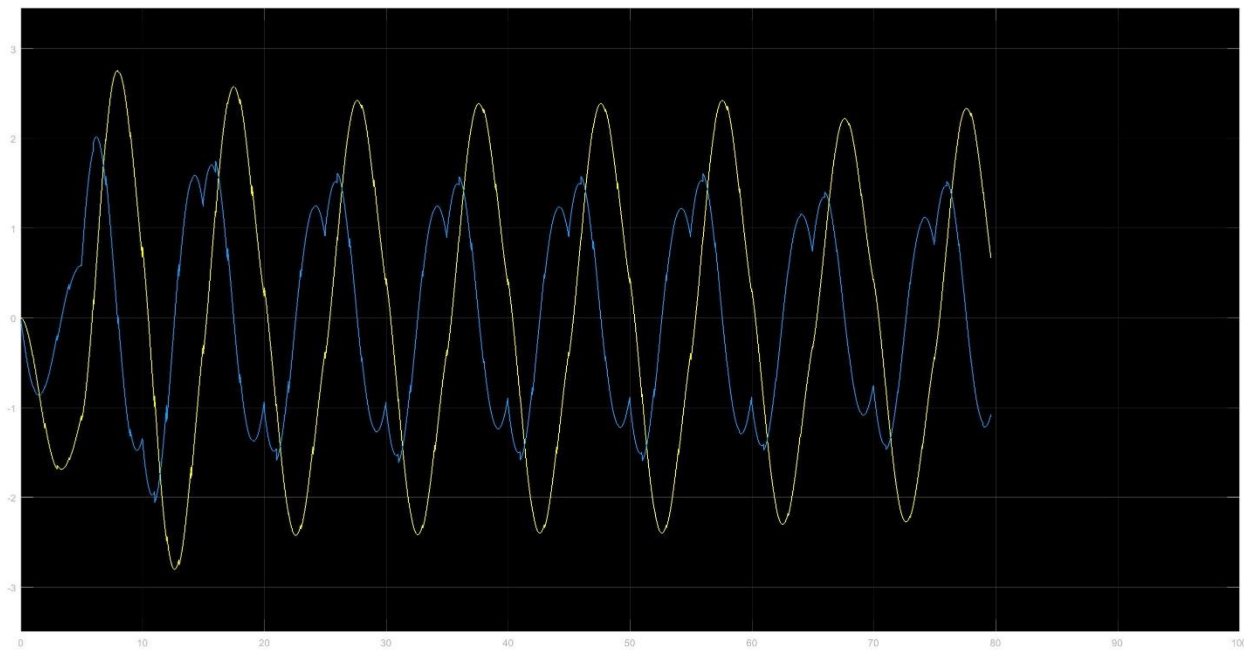
b = 0.4



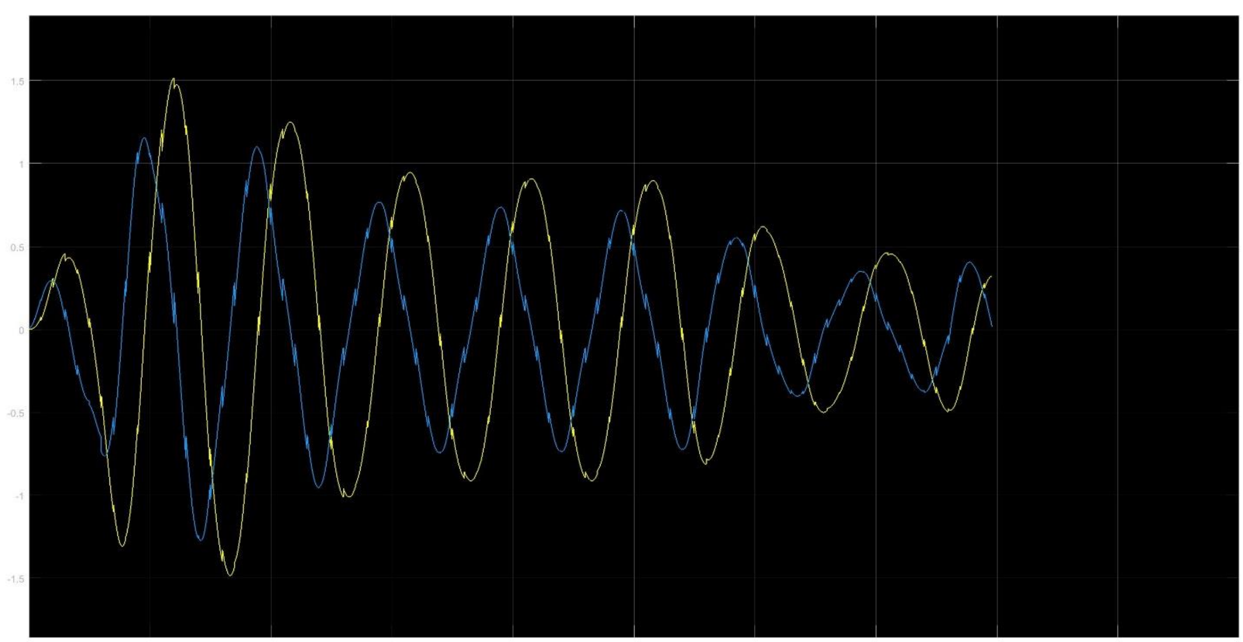
Error (b=0.4)



b = 0.2



Error (b=0.2)



4) To normalize the error, we take the RMSE of the errors and observe the different plots for different values of b.

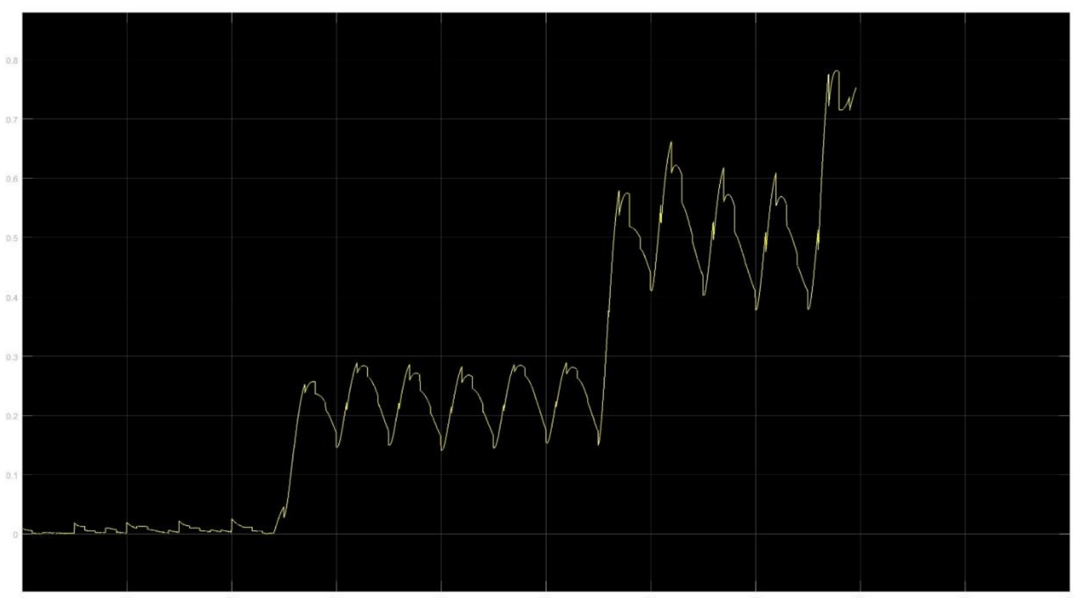
   The MATLAB code for RMSE calculation is as follows :

```
function [rmse1,rmse2,rmse3,rmse4] = rmse(e1,e2,e3,e4)

rmse1=sqrt(e1(1)^2+e1(2)^2);
rmse2=sqrt(e2(1)^2+e2(2)^2);
rmse3=sqrt(e3(1)^2+e3(2)^2);
rmse4=sqrt(e4(1)^2+e4(2)^2);
end
```

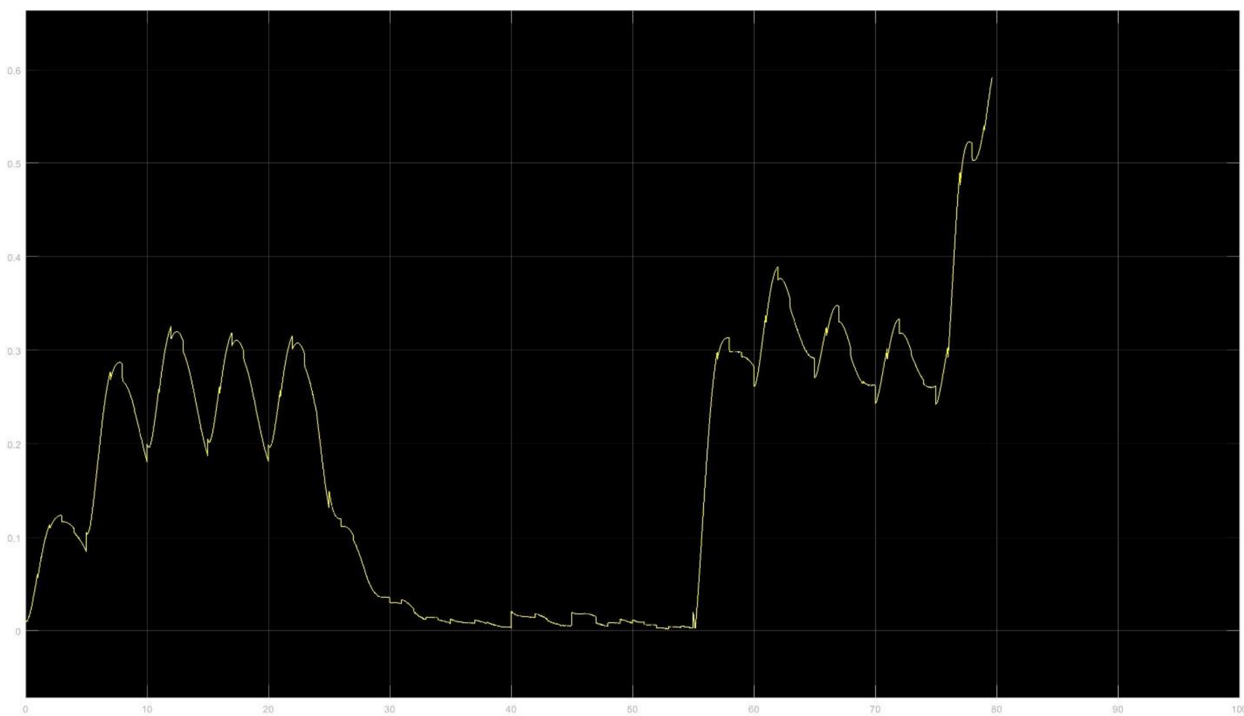
The MATLAB code for K estimates is as follows :

```
function Kest = fcn(rmse1,rmse2,rmse3,rmse4)

[j,k]=min([rmse1,rmse2,rmse3,rmse4]);
Kest=.8;
switch k
    case 1
        Kest=.8;
    case 2
        Kest=0.6;
    case 3
        Kest=0.4;
    case 4
        Kest=0.2;
end
end
```

For b = 0.8

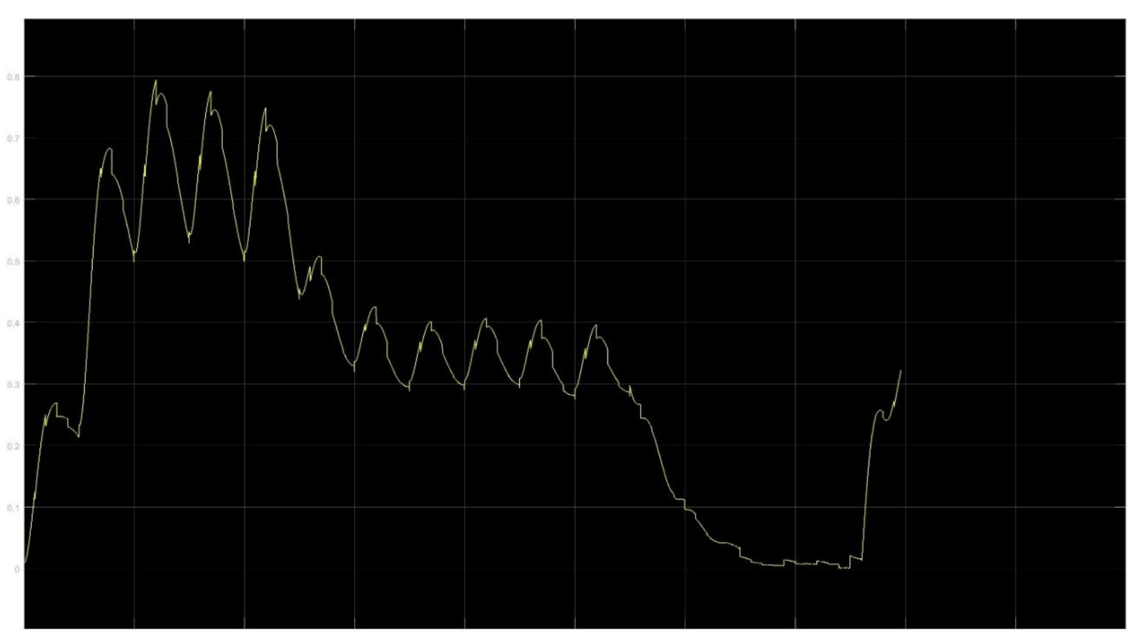


b = 0.6



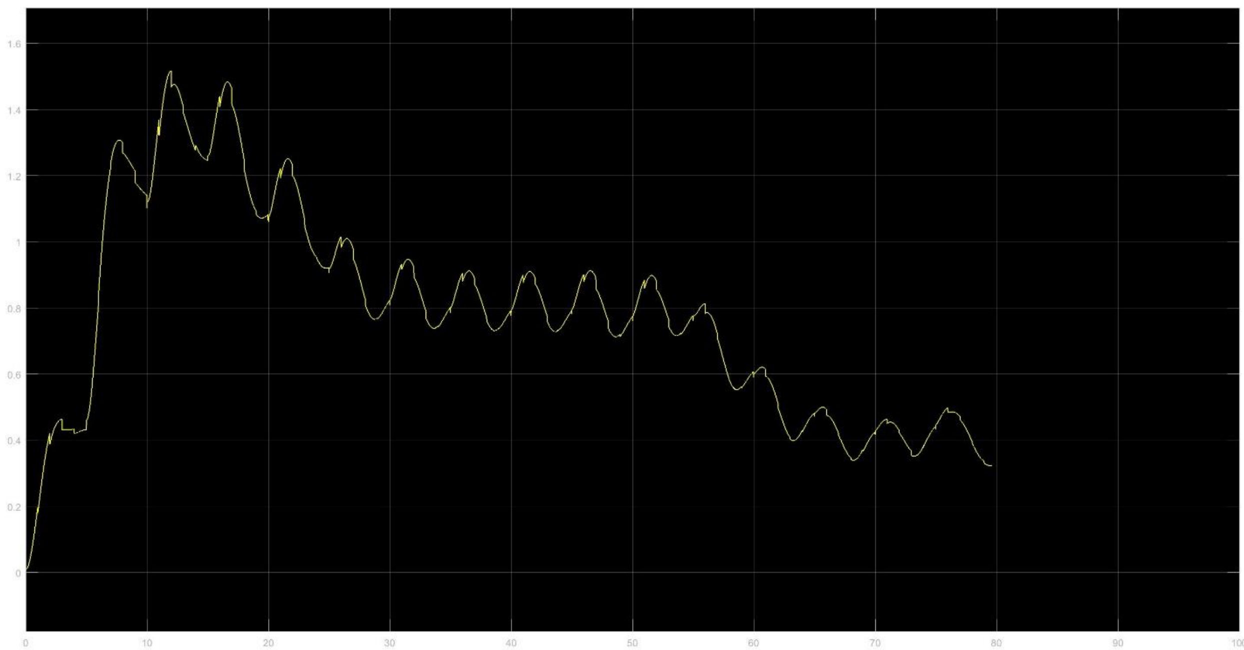
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100

b = 0.4



0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100

b = 0.2



From the above plots, at b = 0.6, the error is minimum for a longer time. This means that the filter performs optimally when b=0.6. Therefore select the Kalman filter with b=0.6.

The following figure shows that the system continues to work till 79s although the third failure occurred at around 76s. It took 3s for the system to stop.