MODERN CONTROL THEORY COURSE PROJECT

IMPLEMENTING KALMAN FILTER IN A QUADRUPLE-TANK TO ESTIMATE THE LEVEL OF WATER.

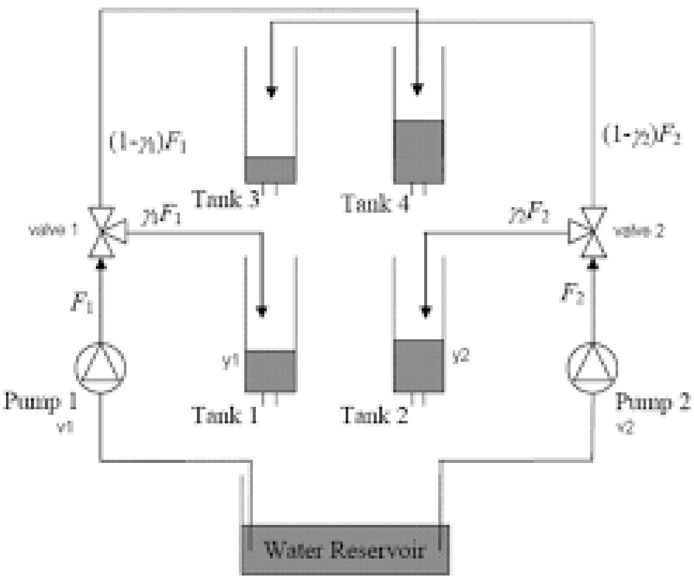
NAME – SREEDARS S J

ROLL NO – ME19B174

What is Kalman Filter?

The Kalman filter produces an estimate of the state of the system as an average of the system's predicted state and of the new measurement using a weighted average. The purpose of the weights is that values with better (i.e., smaller) estimated uncertainty are "trusted" more.

The Quadruple-Tank Process



A schematic diagram of the process is shown above. The target is to control the level in the lower two tanks with two pumps (Tank 1 and Tank 2) . The process inputs are v1 and v2 and (input voltages to the pumps) and the outputs are y1 and y2 (Voltage from level measurement devices).

The Prediction and Update equations

Prediction Equations

Predicted (*a priori*) state estimate $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$

Predicted (*a priori*) estimate covariance $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathsf{T}} + \mathbf{Q}_k$

Update Equations

Updated (*a posteriori*) state estimate $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

Updated (*a posteriori*) estimate covariance $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$

Optimal Kalman gain $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathsf{T}} \mathbf{S}_k^{-1}$

In the above equation the hat operator '^' gives the estimate of the variable.

Values Used for the Quadruple-tank

Area of Each tank

Area of tank 1=28 cm^2 Area of tank 2=32 cm^2

Area of tank 3=28 cm^2 Area of tank 4=32 cm^2

Initial Height of water in each tank

h10=12.4 cm h20 = 12.7 cm;

h30 = 1.8 cm h40=1.4 cm

Area of outlet hole

Tank 1 = 0.071 cm^2 Tank 2 = 0.057 cm^2

Tank 3 = 0.071 cm^2 Tank 4 = 0.057 cm^2

Initial Input Voltages

v1o=3.00 V v2o = 3.00 V

kc = 0.50 V/cm

g = 981 cm/s^2

Parameters to set the valve

γ1 = 0.70 γ2 = 0.6

Using the above Values we Calculate the time-constants as

T1 = 62 seconds T2 = 90 seconds T3 = 23 seconds T4 = 30 seconds

k1 and k2 gives the flow to the tank viz ki*vi

where k1 = 3.33 cm^3/V k2 = 3.35 cm^3/V

State Equations

A = $\begin{bmatrix} -\frac{1}{T1} & 0 & \frac{A3}{A1*T3} & 0 \\ 0 & -\frac{1}{T2} & 0 & \frac{A4}{A2*T4} \\ 0 & 0 & -\frac{1}{T3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T4} \end{bmatrix}$

B = $\begin{bmatrix} \gamma 1 * \frac{k1}{A1} & 0 \\ 0 & \gamma 2 * \frac{k2}{A2} \\ 0 & \frac{(1-\gamma 2)k2}{A3} \\ (1-\gamma 1) * \frac{k1}{A4} & 0 \end{bmatrix}$

C = $\begin{bmatrix} kc & 0 & 0 & 0 \\ 0 & kc & 0 & 0 \end{bmatrix}$

D = 0

Initial Condition we took

x(0)=x posterior initially = $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

P(0)=posterior state error covariance initially = $\begin{bmatrix} 10^5 & 0 & 0 & 0 \\ 0 & 10^5 & 0 & 0 \\ 0 & 0 & 10^5 & 0 \\ 0 & 0 & 0 & 10^5 \end{bmatrix}$

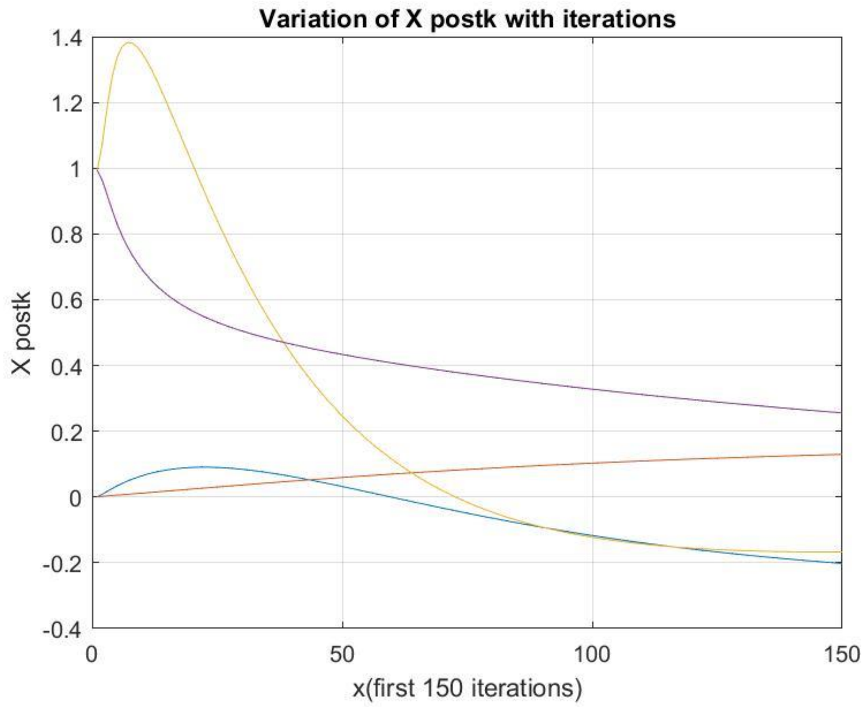
Q=The process noise covariance matrix = 10*$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

R=The Measurement noise covariance matrix = 2*$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Results and Inferences from the Kalman filter implemented in MATLAB.

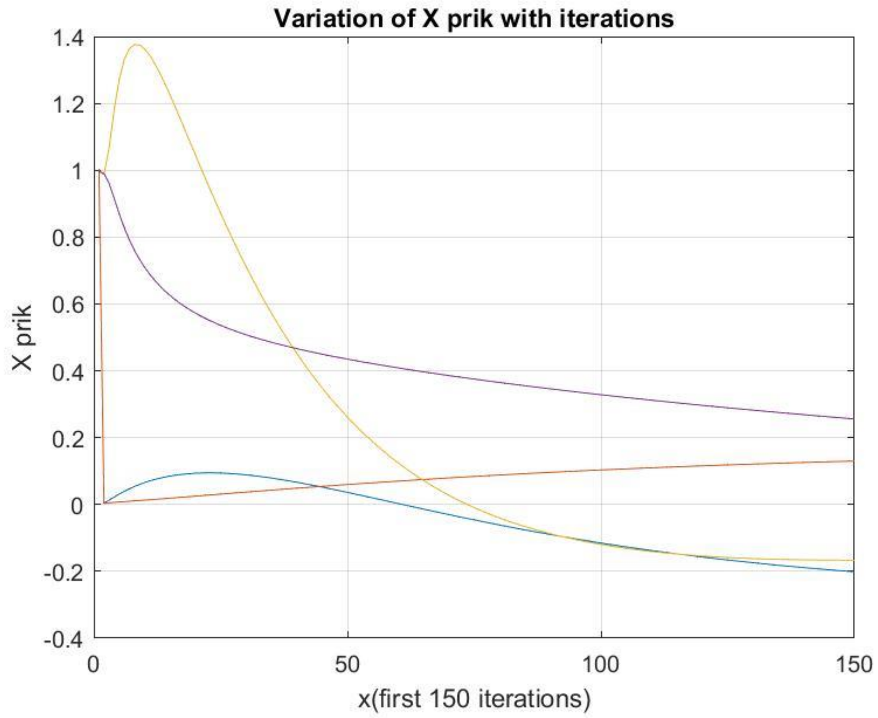
1) **The Updated state estimate**

Plot(X_post_store,k) - You can see that after many iterations the graph converges. The updated state estimate is a 4x4 Matrix with unit as cm.



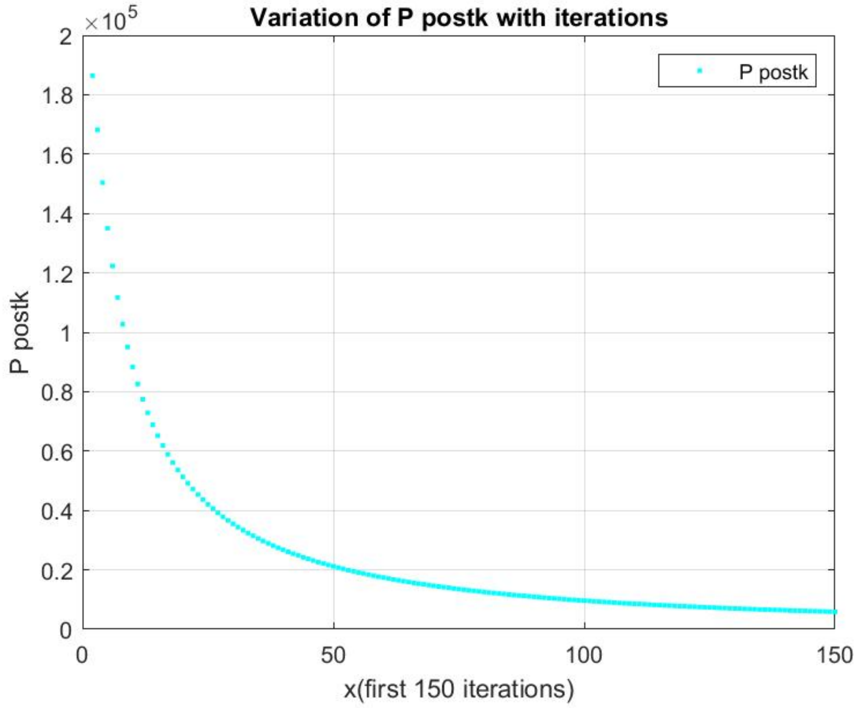
2)The Predicted State estimate

Plot(X_pri_store,k) - You can see that after many iterations the graph converges. The predicted state estimate is a 4x4 Matrix with unit as cm.



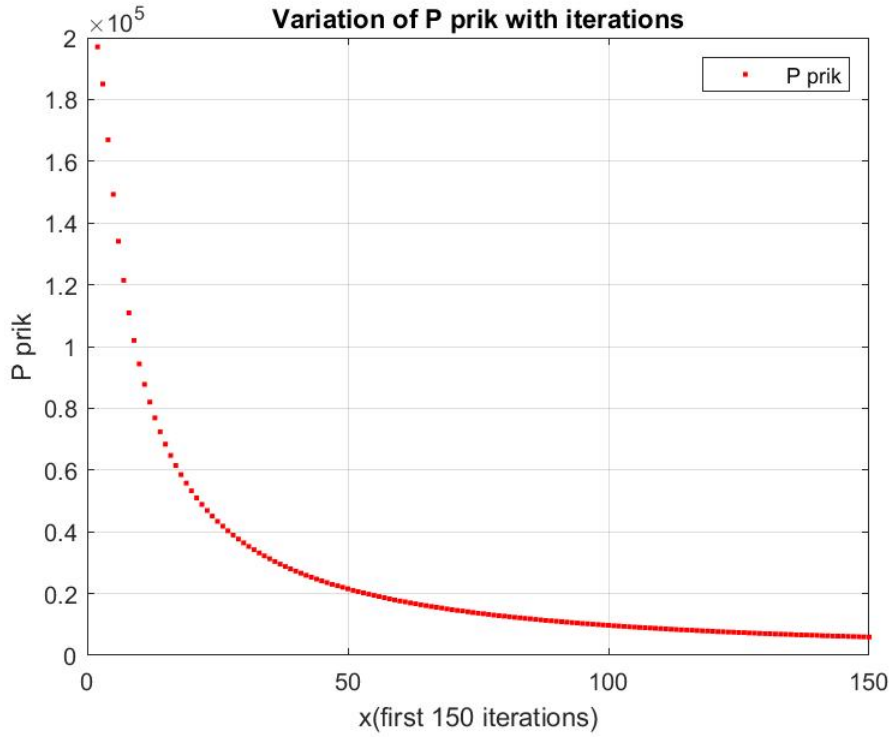
3) **The Updated/Posterior State Error Covariance**

Plot(P_post_store,k) - This is a 4x4 matrix but during plotting we only take the trace of the matrix because the non diagonal elements are close to zero.
You can see from the graph that it converges.



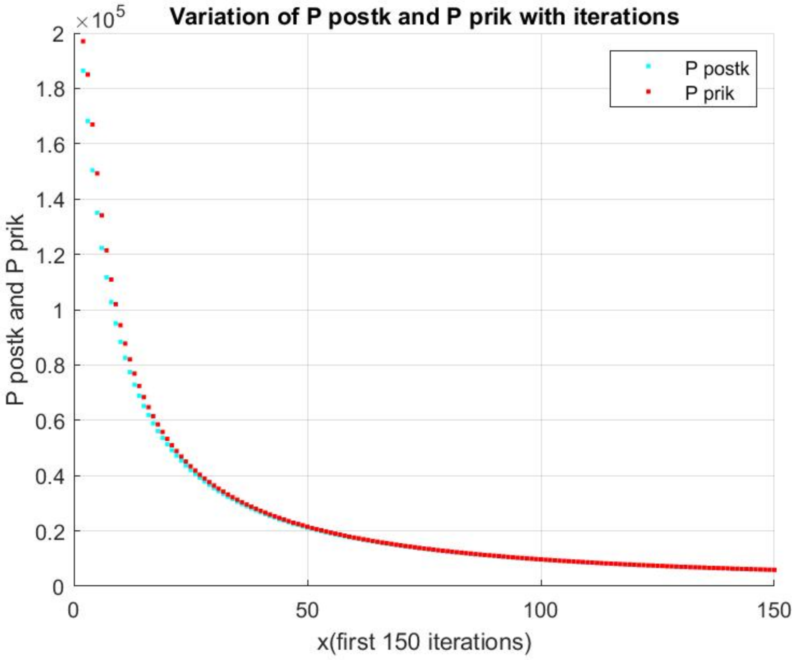
4) **The Predicted/Prior State Error Covariance**

Plot(P_post_store,k) - This is a 4x4 matrix but during plotting we only take the trace of the matrix because the non-diagonal elements are close to zero.
You can see from the graph that it converges.



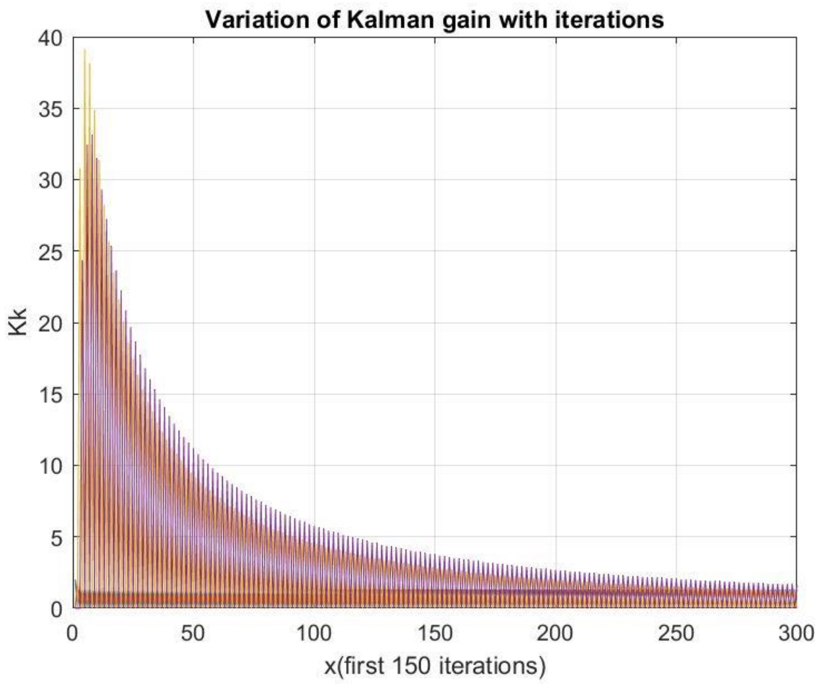
5)Comparing the Prior and posterior state error covariance.

Plot(P_prior_store,P_post_store,k) – Here you can see that the updated error covariance is smaller than the predicted error covariance because the filter is more certain of the state estimate after the measurement is utilised in the update stage.



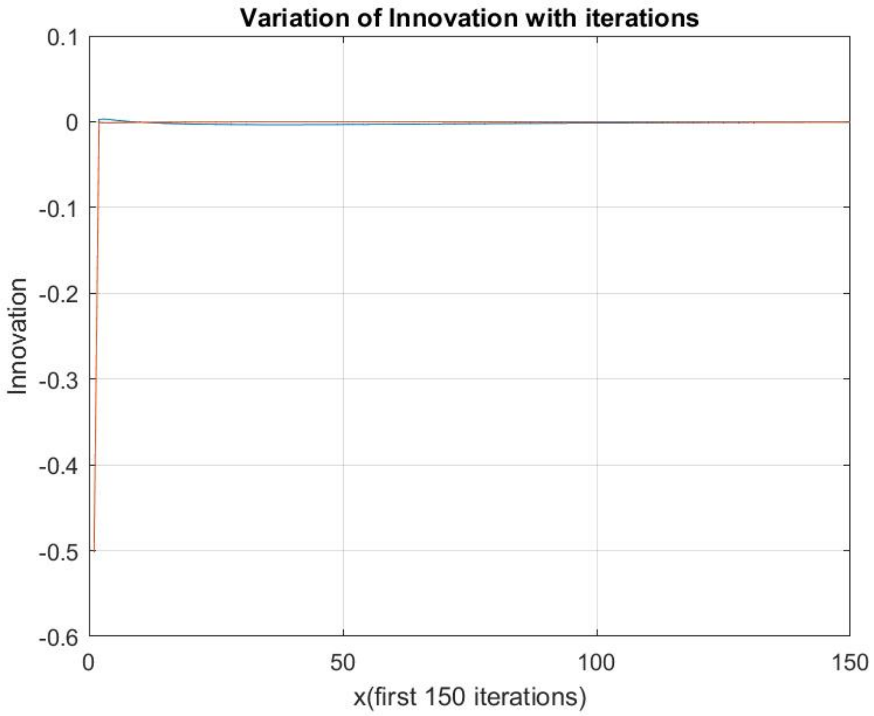
6)The Kalman gain

Plot(K_store,k) - The Kalman-gain is the weight given to the measurements and current-state estimate, which can be "tuned" to achieve a particular performance.
From the graph you can infer that the Kalman gain reaches equilibrium ie. the Kalman gain becomes constant after many iterations. This is a 4x2 matrix



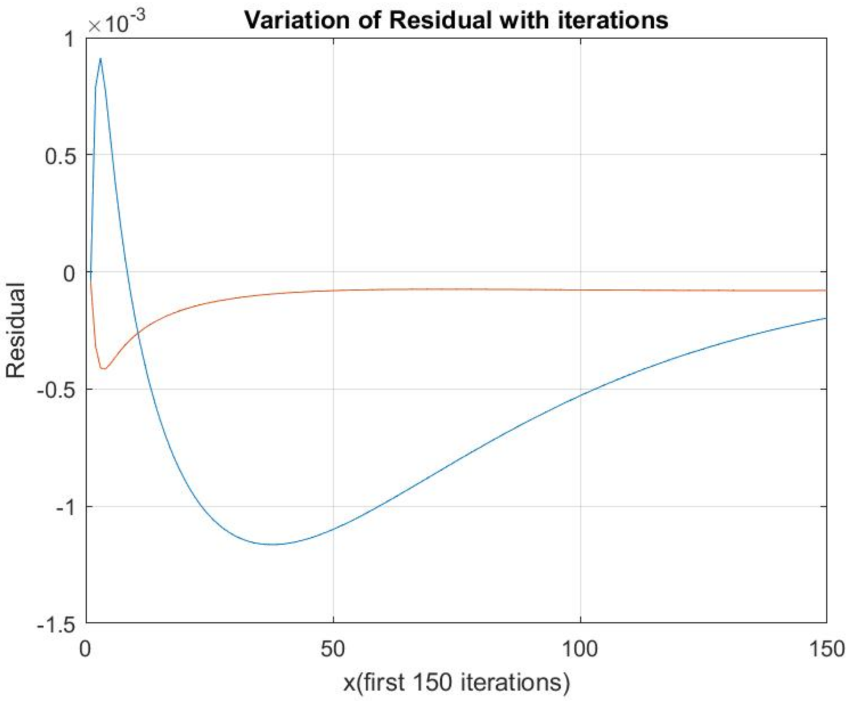
7) Innovation

Plot(Innov_store,k) - The innovation is defined as the difference between the observation (measurement) and its prediction made using the information available at time . This is a 2x1 matrix.



8) Residual

Plot(Resid_store,k) - The residual is the difference between a measurement and the value predicted by the filter.



```
clc
clear
close all

A1=28; A2=32; A3=28; A4=32;              %Given parametrs from lecture
a1=0.071; a2=0.057; a3=0.071;a4=0.057;
kc=0.50;
g=981;
                                         %and

h10=12.4; h20=12.7; h30=1.8; h40=1.4;   %Given parameters from the Reference paper
v10=3.00; v20=3.00;
k1=3.33; k2=3.35;
y1=0.70; y2=0.60;
T1=62; T2=90; T3=23; T4=30;


A=[-1/T1 0 A3/(T3*A1) 0;0 -1/T2 0 A4/(A2*T4);0 0 -1/T3 0;0 0 0 -1/T4];   %Matrix A B C D
B=[y1*k1/A1 0;0 y2*k2/A2 ;0 (1-y2)*k2/A3 ;(1-y1)*k1/A4 0];
H=[kc 0 0 0 ;0 kc 0 0];
D=0;

sys=ss(A,B,H,D);
sysd=c2d(sys,0.1); %convert from continous time to discrete time


Q=10*eye(4);            %Noise matrix
R=2*eye(2);
P_postk=10^5*eye(4);
Uk=0;

hValues = xlsread('Measurements.xlsx');  %Reading from the Excel Sheet
% hValues(1,:)'-[h10;h20;h30;h40]
X_postk=[1;1;1;1];%hValues(1,:)'-[h10;h20;h30;h40]
% Z_truek=hValues(1,1:2)-[h10;h20];
m= size(hValues);


X_post_store =[];    %Introducing matrix to store the values
X_pri_store=[];
P_post_store=[];
P_pri_store=[];
K_store=[];
Innov_store=[];
Resid_store=[];


  for i = 1:150                  %taking the first 150 iterations
    X_prik=sysd.A*X_postk;       %Prediction equations
     P_prik=sysd.A*P_postk*transpose(sysd.A)+Q;

    Kk=P_prik*transpose(sysd.C)*inv(sysd.C*P_prik*transpose(sysd.C)+R);             %Update equations
    Z_estk=sysd.C*X_prik;
    Z_truek = hValues(i,1:2)'-[h10;h20];
    Ek=Z_truek-Z_estk;
    X_postk=X_prik+Kk*Ek;
    P_postk=P_prik-Kk*sysd.C*P_prik;
    resid=Z_truek-H*X_postk;

    X_post_store = [X_post_store , X_postk];
    X_pri_store=[X_pri_store,X_prik];
    P_post_store=[P_post_store,P_postk];
    P_pri_store=[P_pri_store,P_prik];
```

```
    K_store=[K_store,Kk];
    Innov_store=[Innov_store,Ek];
    Resid_store=[Resid_store,resid];

end

figure                  %Plotting the 8 graphs
plot(1:150,X_post_store)
xlabel('x(first 150 iterations)')
ylabel('X postk')
grid on
title('Variation of X postk with iterations')

figure
plot(1:150,X_pri_store)
 xlabel('x(first 150 iterations)')
ylabel('X prik')
grid on
title('Variation of X prik with iterations')

 figure
plot(1:600,P_post_store)
xlabel('x(first 150 iterations)')
ylabel('P postk')
grid on
title('Variation of P postk with iterations')

figure
plot(1:600,P_pri_store)
xlabel('x(first 150 iterations)')
ylabel('P prik')
grid on
title('Variation of P prik with iterations')

figure
hold on
plot(1:600,P_post_store)
plot(1:600,P_pri_store)
xlabel('x(first 150 iterations)')
ylabel('Ppostk and Pprik')
title('Variation of P postk and P prik with iterations')
grid on

figure
plot(1:300,K_store)
xlabel('x(first 150 iterations)')
ylabel('Kk')
grid on
title('Variation of Kalman gain with iterations')

figure
plot(1:150,Innov_store)
 xlabel('x(first 150 iterations)')
ylabel('Innovation')
grid on
title('Variation of Innovation with iterations')

figure
plot(1:150,Resid_store)
 xlabel('x(first 150 iterations)')
ylabel('Residual')
grid on
title('Variation of Residual with iterations')
```