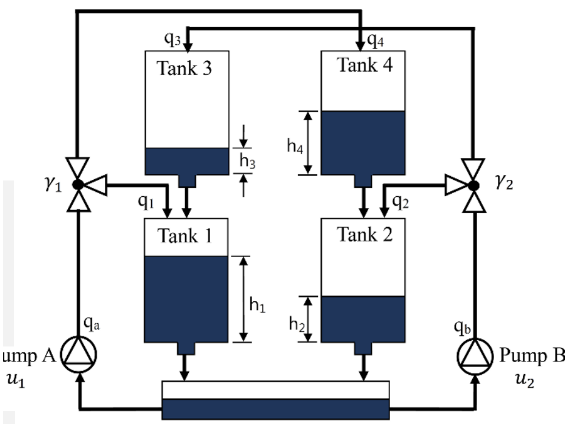
CH5120: Course Project 1

CH20B065 Mihir Singh

Part - 1 : Kalman Filter

Problem Statement

The quadruple tank process consists of four interconnected water tanks. Given below is the schematic diagram. The equations representing the dynamics of the system are given as follows:-



$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1$$
$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2$$
$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2$$
$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1$$

where

A_i cross-section of Tank i;
a_i cross-section of the outlet hole;
h_i water level.

Data Generation:

Solve the set of non-linear equations of the four-tank system, as given below, using forward difference method or by using ODE-45 of *Matlab* (or equivalent in *Python,* etc.), for generating a typical data set of 10,000. This data set is to be used for comparing the performance of the Kalman Filter & Particle Filter.

Kalman Filter Implementation

Formulate a *Kalman Filter* for this four tank problem, for estimating *h-3 & h-4 (not measured)* and obtaining the filtered values for *h-1 & h-2 (measured). Verify the resulting values of h-1, h-2, h-3 & h-4, by comparing the same with the generated data set.*

Obtain all the following plots:

- Prior Estimate of **x(k-)** (state variable), with **k**
- Posterior Estimate of **x(k+)(**state variable), with **k**
- Prior Estimate of Covariance, **P(k-)** with **k**
- Posterior Estimate of Covariance, **P(k+)** with **k**
- Kalman-Gain, **K(k)** with **k**
- Prior Residues/Innovations, **r(k-)** with **k**
- Posterior Residues, **r(k+)** with **k**

Code:

```
clc;

clear;

close all;

%Initial Values

A1 = 28; %(cm^2)

A2 = 32;

A3 = 28;

A4 = 32;

a1 = 0.071; a3 = 0.071; %(cm^2)

a2 = 0.057; a4 = 0.057;

kc = 0.5; %(V/cm)

g = 981; %(cm/s^2)

gamma1 = 0.7; gamma2 = 0.6; %constants that are determined from valve position

k1 = 3.33; k2 = 3.35; %(cm^3/Vs)

v1 = 3;v2 = 3; %(V)

h0 = [12.4; 12.7; 1.8; 1.4];

%Linearized State Representation Matrices

T1 = (A1/a1)*(2*h0(1)/g)^0.5;

T2 = (A2/a2)*(2*h0(2)/g)^0.5;

T3 = (A3/a3)*(2*h0(3)/g)^0.5;

T4 = (A4/a4)*(2*h0(4)/g)^0.5;

Ac = [-1/T1, 0, A3/(A1*T3), 0; 0, -1/T2, 0, A4/(A2*T4); 0, 0, -1/T3, 0; 0, 0,
0, -1/T4];

Bc = [gamma1*k1/A1, 0; 0, gamma2*k2/A2; 0, (1-gamma2)*k2/A3; (1-gamma1)*k1/A4,
0];
```

```
Cc = [kc, 0, 0, 0; 0, kc, 0, 0];

Ad = expm(Ac);

Bd = integral(@(tau) expm(Ac*tau),0,0.04,'ArrayValued',true)*Bc;

Cd = Cc;

u = [v1;v2];

%Ordinary Differential Equation System

tspan = linspace(0,400,10000);

[t,h] = ode45(@ODE,tspan,h0);

y = Cc*h' + 0.001*eye(2)*randn(2,10000);

%Kalman Filter application

x_prior = zeros(4,10001);

x_posterior = zeros(4,10001);

Z_est_prior = zeros(2,10001);

Z_est_posterior = zeros(2,10001);

x0 = h0;

x_posterior(:,1) = x0;

x_prior(:,1) = x0;

P_prior = zeros(4, 4, 10001);

K = zeros(4, 2, 10001);

P0 = 0.01*eye(4); %arbitrary initial state error covariance matrix

P_posterior(:,:,1) = P0;

resid_prior = zeros(4,10001);

resid_posterior = zeros(4,10001);

Q = 0.00001*eye(4); %covariance matrix of noise in state equation

R = 0.00001*eye(2); %covariance matrix of noise in measurement equation

for k=1:10000
```

```
%Prediction
x_prior(:,k+1) = Ad*x_posterior(:,k) + Bd*u;
P_prior(:,:,k+1) = Ad*P_posterior(:,:,k)*Ad'+ Q;
%Update
K(:,:,k+1) = P_prior(:,:,k)*Cd'*inv(Cd*P_prior(:,:,k+1)*Cd' + R);
Z_est_prior(:,k+1) = Cd*x_prior(:,k+1);
resid_prior(:,k+1) = h(k,:)' - x_prior(:,k+1);
x_posterior(:,k+1) = x_prior(:,k) + K(:,:,k+1)*(y(:,k)-Cd*x_prior(:,k));
Z_est_posterior(:,k+1) = Cd*x_posterior(:,k+1);
resid_posterior(:,k+1) = h(k,:)' - x_posterior(:,k+1);
P_posterior(:,:,k+1) = (eye(4) - K(:,:,k+1)*Cd)*P_prior(:,:,k+1);
end
%Visualising the results
k_span = 2:10001;
%Visualisation of simulation
figure;
plot(t,h(:,1),t,h(:,2),t,h(:,3),t,h(:,4))
legend('h1','h2','h3','h4')
title('Simulation results')
%Prior Estimates
figure;
plot(k_span, x_prior(1,2:10001), 'LineWidth', 2);
hold on
plot(k_span, x_prior(2,2:10001), 'LineWidth', 2);
hold on
plot(k_span, x_prior(3,2:10001), 'LineWidth', 2);
```

```
hold on

plot(k_span, x_prior(4,2:10001), 'LineWidth', 2);

hold on

legend('h1','h2','h3','h4')

title('Plots of Prior')

%Posterior Estimates

figure;

plot(k_span, x_posterior(1,2:10001), 'LineWidth', 2);

hold on

plot(k_span, x_posterior(2,2:10001), 'LineWidth', 2);

hold on

plot(k_span, x_posterior(3,2:10001), 'LineWidth', 2);

hold on

plot(k_span, x_posterior(4,2:10001), 'LineWidth', 2);

hold on

legend('h1','h2','h3','h4')

title('Plots of Posterior')

%Posterior Residuals

figure;

plot(k_span, resid_posterior(1,2:10001), 'LineWidth', 2);

hold on

plot(k_span, resid_posterior(2,2:10001), 'LineWidth', 2);

hold on

plot(k_span, resid_posterior(3,2:10001), 'LineWidth', 2);

hold on

plot(k_span, resid_posterior(4,2:10001), 'LineWidth', 2);
```

```
hold on

legend('h1','h2','h3','h4')

title('Plots of posterior residuals')

%Prior Residuals

figure;

plot(k_span, resid_prior(1,2:10001), 'LineWidth', 2);

hold on

plot(k_span, resid_prior(2,2:10001), 'LineWidth', 2);

hold on

plot(k_span, resid_prior(3,2:10001), 'LineWidth', 2);

hold on

plot(k_span, resid_prior(4,2:10001), 'LineWidth', 2);

hold on

legend('h1','h2','h3','h4')

title('Plots of prior residuals')

%Prior Covariance

figure;

subplot(4,4,1);

plot(k_span,squeeze(P_prior(1,1,2:10001)),'LineWidth', 2);

subplot(4,4,2);

plot(k_span,squeeze(P_prior(1,2,2:10001)),'LineWidth', 2);

subplot(4,4,3);

plot(k_span,squeeze(P_prior(1,3,2:10001)),'LineWidth', 2);

subplot(4,4,4);

plot(k_span,squeeze(P_prior(1,4,2:10001)),'LineWidth', 2);

subplot(4,4,5);
```

```
plot(k_span,squeeze(P_prior(2,1,2:10001)),'LineWidth', 2);
subplot(4,4,6);
plot(k_span,squeeze(P_prior(2,2,2:10001)),'LineWidth', 2);
subplot(4,4,7);
plot(k_span,squeeze(P_prior(2,3,2:10001)),'LineWidth', 2);
subplot(4,4,8);
plot(k_span,squeeze(P_prior(2,4,2:10001)),'LineWidth', 2);
subplot(4,4,9);
plot(k_span,squeeze(P_prior(3,1,2:10001)),'LineWidth', 2);
subplot(4,4,10);
plot(k_span,squeeze(P_prior(3,2,2:10001)),'LineWidth', 2);
subplot(4,4,11);
plot(k_span,squeeze(P_prior(3,3,2:10001)),'LineWidth', 2);
subplot(4,4,12);
plot(k_span,squeeze(P_prior(3,4,2:10001)),'LineWidth', 2);
subplot(4,4,13);
plot(k_span,squeeze(P_prior(4,1,2:10001)),'LineWidth', 2);
subplot(4,4,14);
plot(k_span,squeeze(P_prior(4,2,2:10001)),'LineWidth', 2);
subplot(4,4,15);
plot(k_span,squeeze(P_prior(4,3,2:10001)),'LineWidth', 2);
subplot(4,4,16);
plot(k_span,squeeze(P_prior(4,4,2:10001)),'LineWidth', 2);
sgtitle('Prior Covariances')
%Posterior Covariances
figure;
```

```
subplot(4,4,1);
plot(k_span,squeeze(P_posterior(1,1,2:10001)),'LineWidth', 2);
subplot(4,4,2);
plot(k_span,squeeze(P_posterior(1,2,2:10001)),'LineWidth', 2);
subplot(4,4,3);
plot(k_span,squeeze(P_posterior(1,3,2:10001)),'LineWidth', 2);
subplot(4,4,4);
plot(k_span,squeeze(P_posterior(1,4,2:10001)),'LineWidth', 2);
subplot(4,4,5);
plot(k_span,squeeze(P_posterior(2,1,2:10001)),'LineWidth', 2);
subplot(4,4,6);
plot(k_span,squeeze(P_posterior(2,2,2:10001)),'LineWidth', 2);
subplot(4,4,7);
plot(k_span,squeeze(P_posterior(2,3,2:10001)),'LineWidth', 2);
subplot(4,4,8);
plot(k_span,squeeze(P_posterior(2,4,2:10001)),'LineWidth', 2);
subplot(4,4,9);
plot(k_span,squeeze(P_posterior(3,1,2:10001)),'LineWidth', 2);
subplot(4,4,10);
plot(k_span,squeeze(P_posterior(3,2,2:10001)),'LineWidth', 2);
subplot(4,4,11);
plot(k_span,squeeze(P_posterior(3,3,2:10001)),'LineWidth', 2);
subplot(4,4,12);
plot(k_span,squeeze(P_posterior(3,4,2:10001)),'LineWidth', 2);
subplot(4,4,13);
plot(k_span,squeeze(P_posterior(4,1,2:10001)),'LineWidth', 2);
```

```
subplot(4,4,4);
plot(k_span,squeeze(P_posterior(4,2,2:10001)),'LineWidth', 2);
subplot(4,4,15);
plot(k_span,squeeze(P_posterior(4,3,2:10001)),'LineWidth', 2);
subplot(4,4,16);
plot(k_span,squeeze(P_posterior(4,4,2:10001)),'LineWidth', 2);
sgtitle('Posterior Covariances')
%Kalman Filters
figure;
subplot(4,2,1);
plot(k_span,squeeze(K(1,1,2:10001)),'LineWidth', 2);
subplot(4,2,2);
plot(k_span,squeeze(K(2,1,2:10001)),'LineWidth', 2);
subplot(4,2,3)
plot(k_span,squeeze(K(3,1,2:10001)),'LineWidth', 2);
subplot(4,2,4);
plot(k_span,squeeze(K(4,1,2:10001)),'LineWidth', 2);
subplot(4,2,5);
plot(k_span,squeeze(K(1,2,2:10001)),'LineWidth', 2);
subplot(4,2,6);
plot(k_span,squeeze(K(2,2,2:10001)),'LineWidth', 2);
subplot(4,2,7);
plot(k_span,squeeze(K(3,2,2:10001)),'LineWidth', 2);
subplot(4,2,8);
plot(k_span,squeeze(K(4,2,2:10001)),'LineWidth', 2);
sgtitle('Kalman Filters')
```

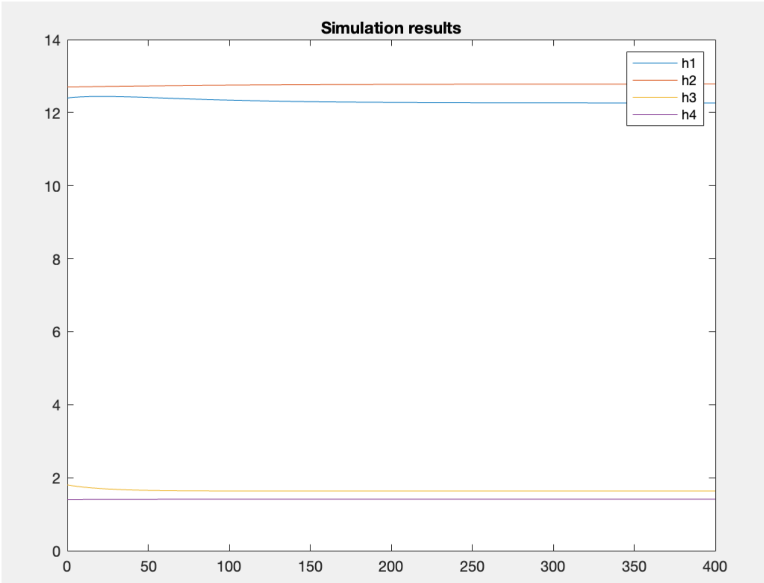
```
%h3 and h4

figure;

subplot(2,1,1)

plot(k_span, x_posterior(3,2:10001), k_span, h(:,3), 'LineWidth', 2);

legend('Estimated h3','Simulated h3')

title('h3 estimates')

hold on

subplot(2,1,2)

plot(k_span, x_posterior(4,2:10001), k_span, h(:,4), 'LineWidth', 2);

legend('Estimated h4','Simulated h4')

title('h4 estimtates')

hold on

function dhdt = ODE(t,h)

A1 = 28; %(cm^2)

A2 = 32;

A3 = 28;

A4 = 32;

a1 = 0.071; a3 = 0.071; %(cm^2)

a2 = 0.057; a4 = 0.057;

kc = 0.5; %(V/cm)

g = 981; %(cm/s^2)

gamma1 = 0.7; gamma2 = 0.6; %constants that are determined from valve postion

k1 = 3.33; k2 = 3.35; %(cm^3/Vs)

v1 = 3;v2 = 3; %(V)

dhdt = zeros(4,1);

dhdt(1) = -a1/A1*(2*g*h(1))^0.5 + a3/A1*(2*g*h(3))^0.5 + gamma1*k1/A1*v1;
```

```
dhdt(2) = -a2/A2*(2*g*h(2))^0.5 + a4/A2*(2*g*h(4))^0.5 + gamma2*k2/A2*v2;

dhdt(3) = -a3/A3*(2*g*h(3))^0.5 + (1-gamma2)*k2*v2/A3;

dhdt(4) = -a4/A4*(2*g*h(4))^0.5 + (1-gamma1)*k1*v1/A4;

end
```

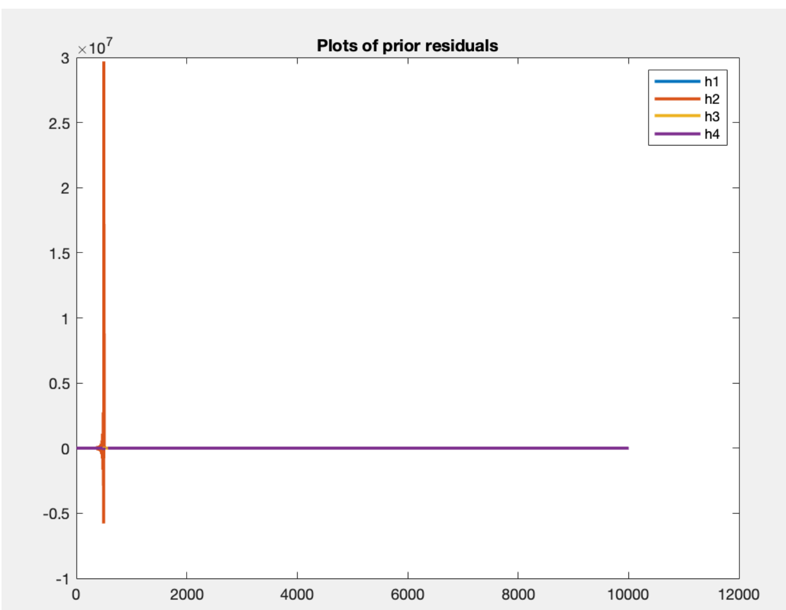
Assumptions:

- **Linear System Model:** The Kalman filter assumes that both the state transition model (system dynamics) and the measurement model are linear.
- **Gaussian Noise**: The filter assumes that the process noise (system noise) and measurement noise are both Gaussian distributed.
- **Stationarity**: The Kalman filter assumes that the statistical properties of the system and measurement noise do not change over time.
- **Independence of Noise Sources**: It is assumed that the process noise and measurement noise are independent of each other.

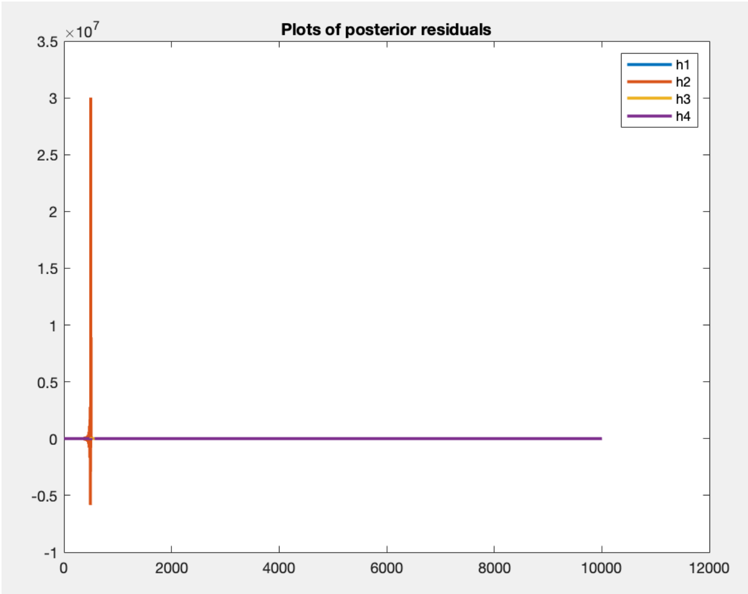
Plots:

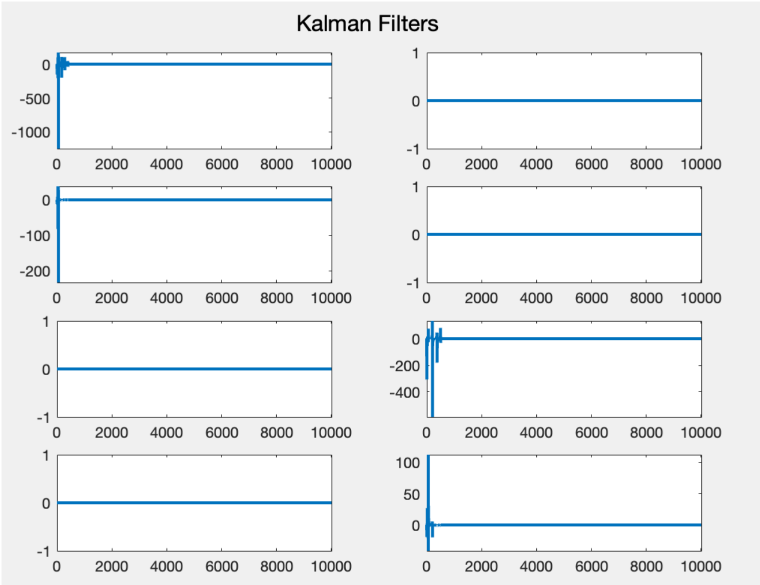


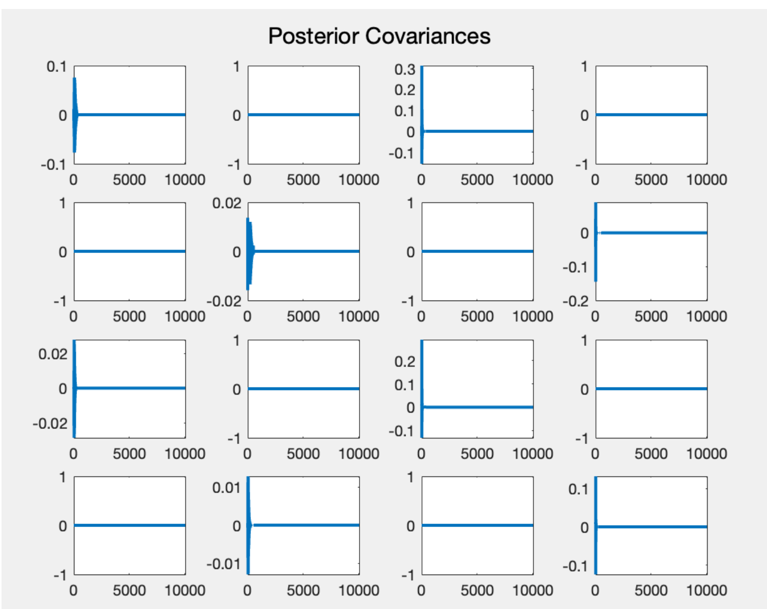
Simulation results
h1
h2
h3
h4
14
12
10
8
6
4
2
0
0
50
100
150
200
250
300
350
400

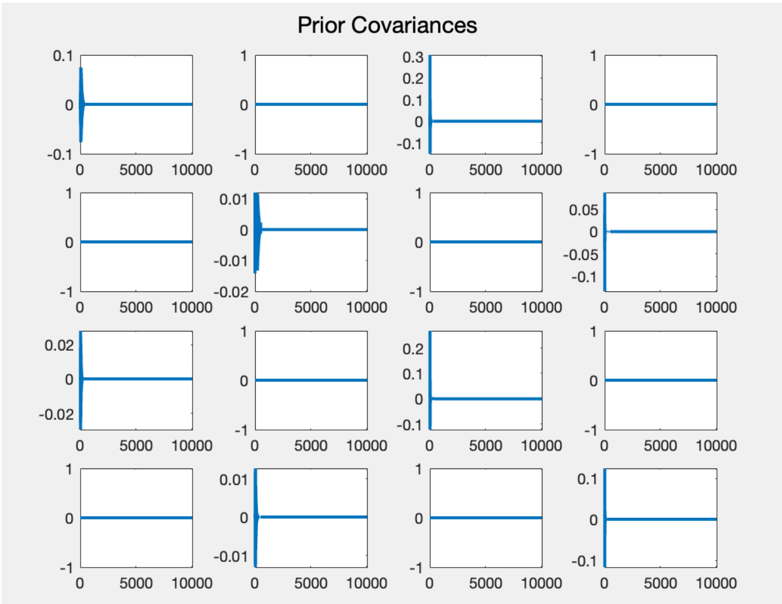


Plots of prior residuals
×10^7
h1
h2
h3
h4
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2000
4000
6000
8000
10000
12000


Plots of posterior residuals
×10^7
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2000
4000
6000
8000
10000
12000
h1
h2
h3
h4


Kalman Filters


Posterior Covariances



Results:

The Kalman filter is designed to provide optimal estimates, minimizing the mean squared error of the estimate. In other words, it strives to provide the most accurate estimate while considering measurement noise and model uncertainty.

From the plots of h1,h2,h3, and h4 we can see that it does a reasonably great job at estimating the states that provide an output that very closely matches the measurement at any instant. The error in predicting the measurements of h1 and h2 converges to zero which means that the Kalman filter has done well in correctly estimating the states.