**Problem 1:**

1. The plot of {} and {} for t=1,2,…500:



Figure 1: The plot of {} and {}

1. Assuming A, H, W, Q are unknown, their estimates are:

>> Ahat

Ahat =

0.8675 0.0969 -0.1815

-0.2924 0.7546 0.1330

0.2492 0.0759 0.9160

>> Hhat

Hhat =

0.9789 0.4917 0.2002

0.5103 1.0111 0.1031

>> What

What =

0.9253 0.0029 0.0266

0.0029 0.9320 0.0278

0.0266 0.0278 0.9563

>> Qhat

Qhat =

0.5459 -0.0094

-0.0094 0.5333

(c)



Figure 2: Plot of the true and estimate states using Kalman Filter for each component

Comment:

For the three components, they do not have the same accuracy. The first two components have very good accuracy while the third one is not as good as the first two.

The reasons are: First, x is three dimensional data while y is only two dimensional data, if we use y to estimate x, there must be one dimension (component) in x that is not good enough. Second, if we write out the measurement equation, we have:



We can see  are dominated by and  are dominated by , but do not have much dominance in  and(just like noise term), so we cannot expect good accuracy using to estimate .

(d ) The  for the filtering estimation and prediction estimation are:

>> R2m

R2m =

0.8185

0.9345

0.4069

>> R2p

R2p =

0.5671

0.8043

0.3707

Comment:

For each component, the for the filtering estimation is much higher than it for the prediction estimation, which means filtering estimation has better estimation accuracy.

**Problem 2:**

If we change, we do not have good estimation for all of the three components (as shown in Figure 3):



Figure 3: Plot of the true and estimate states using Kalman Filter ()

Now, the of each component for the filtering estimation and prediction estimation are:

>> R2m

R2m =

0.1226

0.3242

0.1591

>> R2p

R2p =

0.0843

0.2553

0.1755

We can see the estimation accuracy is very bad both for the filtering estimator and prediction estimator. It is due to the reason that we add too large noise and it hides the two main effect processes, so it does not perform as good as before.

**Problem 3:**

****

Figure 4: The plot of {} and {}



Figure 5: Plot of the true and estimate states using Kalman Filter for each component

Comment:

We can see the process is very unstable. The reason is because there is an eigenvalue of Matrix A which is larger than 1.

>> eig(A)

ans =

1.2091

0.7812

0.4097

Code:

clear all; close all;

% A = [0.9 0.1 -0.2; -0.3 0.8 0.1; 0.2 0.1 0.9];

A = [0.9 -0.2 -0.1; -0.2 0.9 0.3; 0.0 0.3 0.6];

H = [1 0.5 0.2; 0.5 1 0.1];

W = eye(3);

Q = 50\*eye(2);

% set initial condition

x(:,1) = randn(3,1);

y(:,1) = H\*x(:,1)+ mvnrnd(zeros(1,2),Q)';

% simulate training data

M=500;

for k=2:M

x(:,k)=A\*x(:,k-1)+mvnrnd(zeros(1,3),W)';

y(:,k)=H\*x(:,k) + mvnrnd(zeros(1,2),Q)';

end

% plots of Xt and Yt

figure(1);

subplot(2,1,1);

plot(x');

hleg1 = legend('First Component','Second Component','Third Component',...

'Location','NorthEastOutside')

xlabel('time t');

ylabel('x');

subplot(2,1,2);

plot(y');

xlabel('time t');

ylabel('y');

hleg2 = legend('First Component','Second Component','Location','NorthEastOutside')

% estimates of A,W,H,Q (assume they are unknown)

Ahat = x(:,2:M)\*x(:,1:M-1)'\*inv(x(:,1:M-1)\*x(:,1:M-1)')

What = (x(:,2:M)\*x(:,2:M)'-Ahat\*x(:,1:M-1)\*x(:,2:M)')/(M-1);

Hhat = y\*x'\*inv(x\*x');

Qhat = (y\*y'-Hhat\*x\*y')/M;

% simulate another set of data(using true value of A, H, W, Q)

x2(:,1) = randn(3,1);

y2(:,1) = H\*x2(:,1) + mvnrnd(zeros(1,2),Q)';

for k=2:M

x2(:,k) = A\*x2(:,k-1)+mvnrnd(zeros(1,3),W)';

y2(:,k) = H\*x2(:,k) + mvnrnd(zeros(1,2),Q)';

end

% estimate using Kalman Filter Algorithm (using estimated value of A, H, W, Q)

xh=zeros(3,1);

xmh=xh;

P=zeros(3);

Pm=P;

K=zeros(3,2);

for k=2:M

xmh(:,k) = Ahat\*xh(:,k-1);

Pm(:,:,k) = Ahat\*P(:,:,k-1)\*Ahat'+What;

K(:,:,k) = Pm(:,:,k)\*Hhat'\*inv(Hhat\*Pm(:,:,k)\*Hhat'+Qhat);

P(:,:,k) = (eye(3)-K(:,:,k)\*Hhat)\*Pm(:,:,k);

xh(:,k) = xmh(:,k)+K(:,:,k)\*(y2(:,k)-Hhat\*xmh(:,k));

end

figure(2)

for i=1:3

subplot(3,1,i)

plot(1:M,x2(i,:), 1:M,xh(i,:));

xlabel('t')

ylabel(['Component ' num2str(i)])

legend('True Data','Kalman Filter Estimation','location','northwest')

end

% compute R\_squares\_error

R2m=1-sum((x2-xh).^2,2)./sum((x2-mean(x2,2)\*ones(1,M)).^2,2); % filtering estimation

R2p=1-sum((x2-xmh).^2,2)./sum((x2-mean(x2,2)\*ones(1,M)).^2,2); % prediction estimation