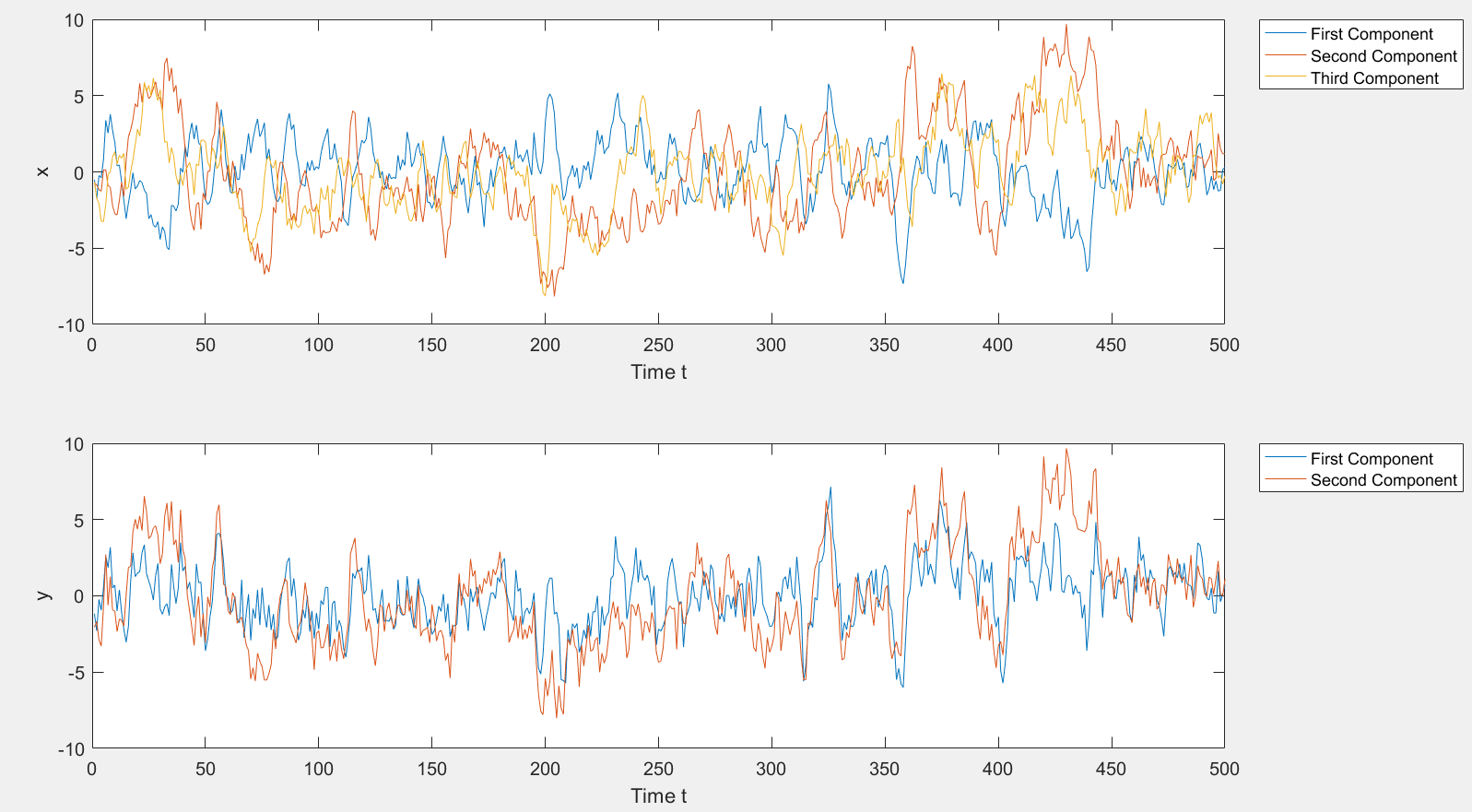
SHAO TANG

Problem 1

(a)



(b)

Ahat =

0.8869 0.0907 -0.1872

-0.3274 0.7981 0.1035

0.1847 0.1000 0.8774

What =

0.9066 0.0058 0.0682

0.0058 1.0029 0.0010

0.0682 0.0010 1.0702

Hhat =

0.9787 0.4960 0.1977

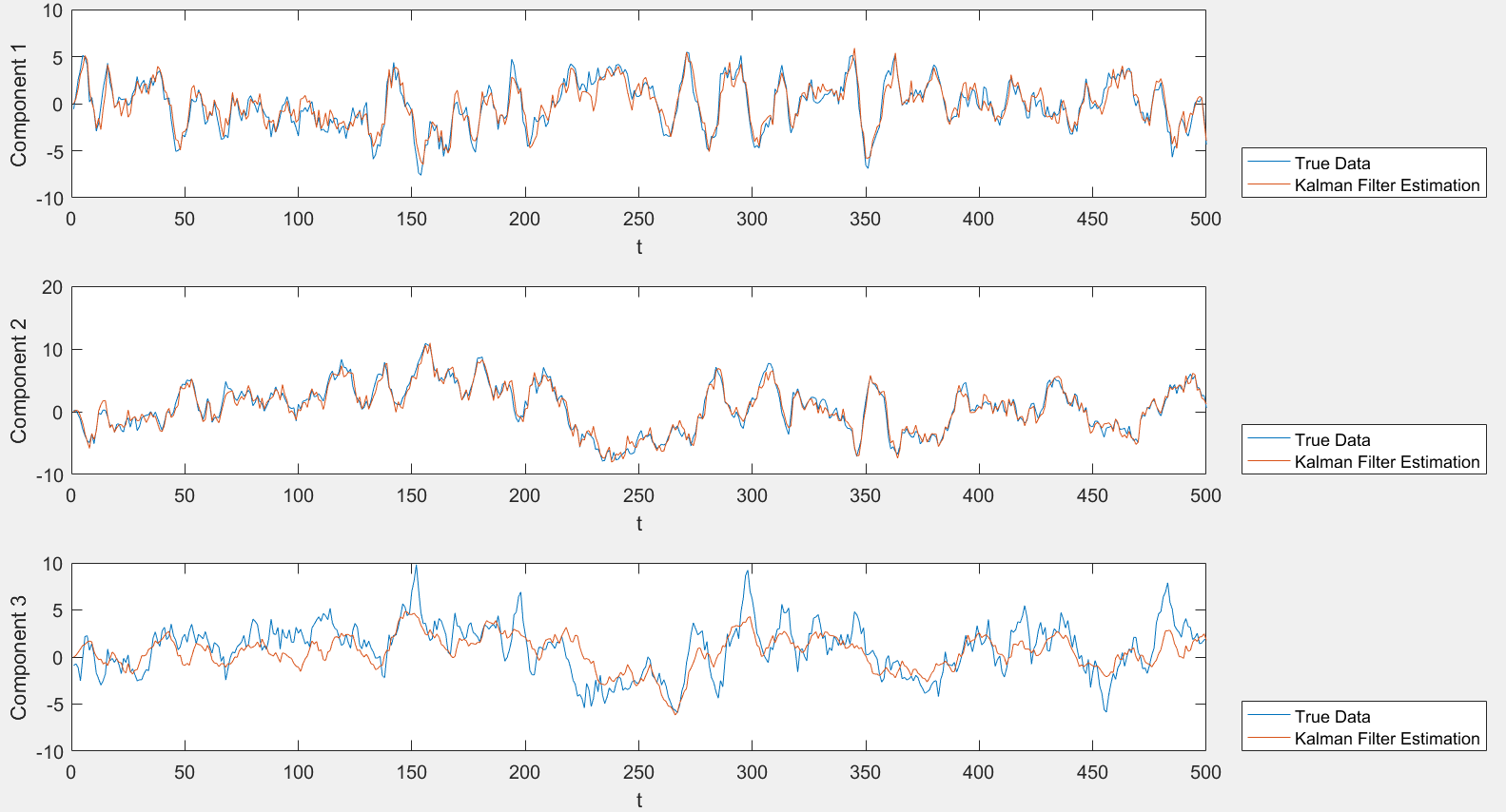
0.4810 1.0066 0.1195

Qhat =

0.4602 -0.0093

-0.0093 0.4139

(C)



For the three components, they do not have the same accuracy. The estimations of the first two components are much better than the third one. X is a three-dimensional vector and we are using the information from Y which is a two dimensional vector. Therefore, one component cannot be estimated well.

(d)

R2\_filter =

0.8753

0.9557

0.5028

R2\_predict =

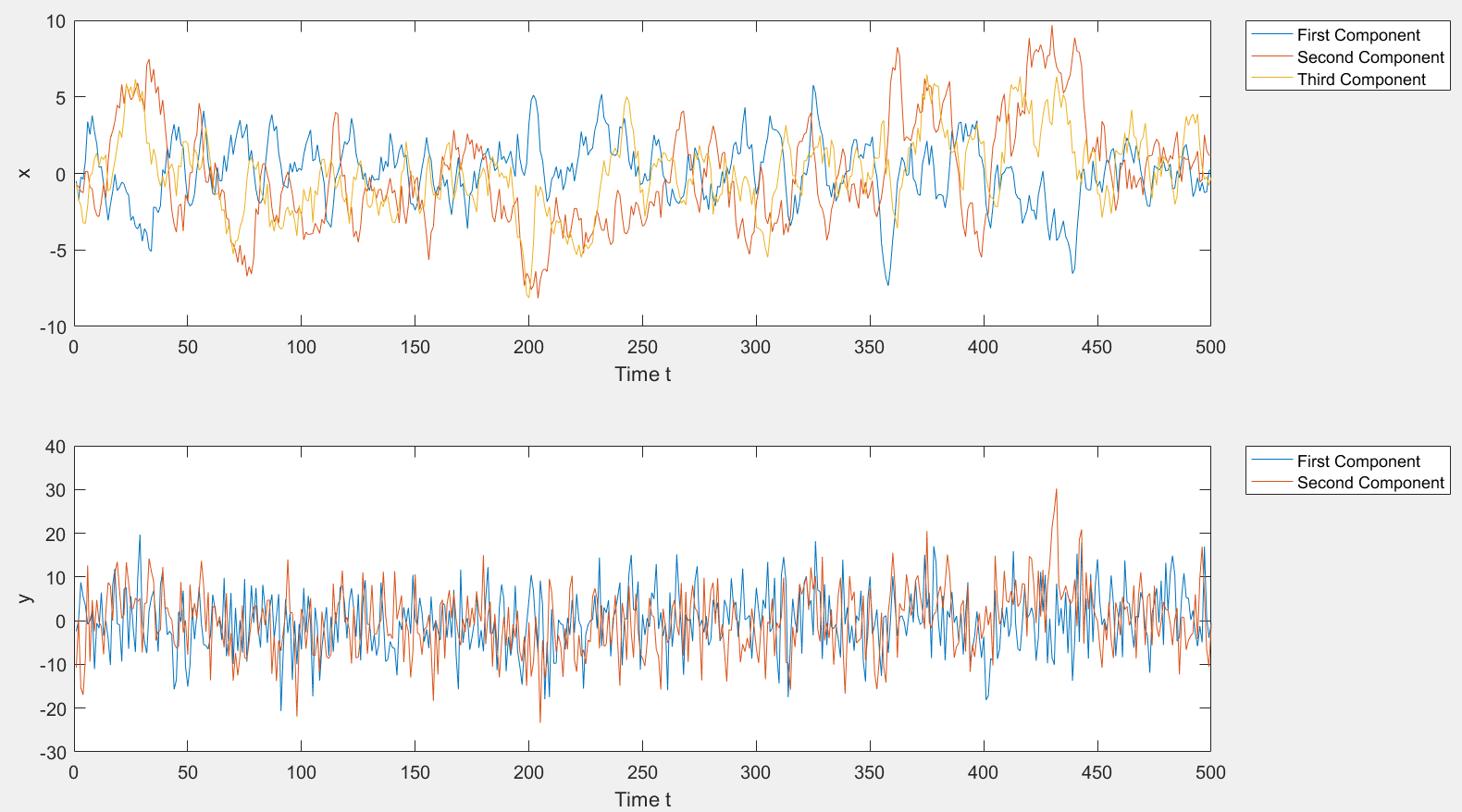
0.6567

0.8671

0.4942

The R2\_filter is much higher for each component and thus the estimation is better.

Problem 2



Ahat =

0.8869 0.0907 -0.1872

-0.3274 0.7981 0.1035

0.1847 0.1000 0.8774

What =

0.9066 0.0058 0.0682

0.0058 1.0029 0.0010

0.0682 0.0010 1.0702

Hhat =

0.7867 0.4603 0.1770

0.3102 1.0663 0.2951

Qhat =

46.0217 -0.9311

-0.9311 41.3867



Now, estimations for all three components are pretty bad. This is due to the fact that we change the noise scale to 50.

R2\_filter =

0.1744

0.4344

0.1964

R2\_predict =

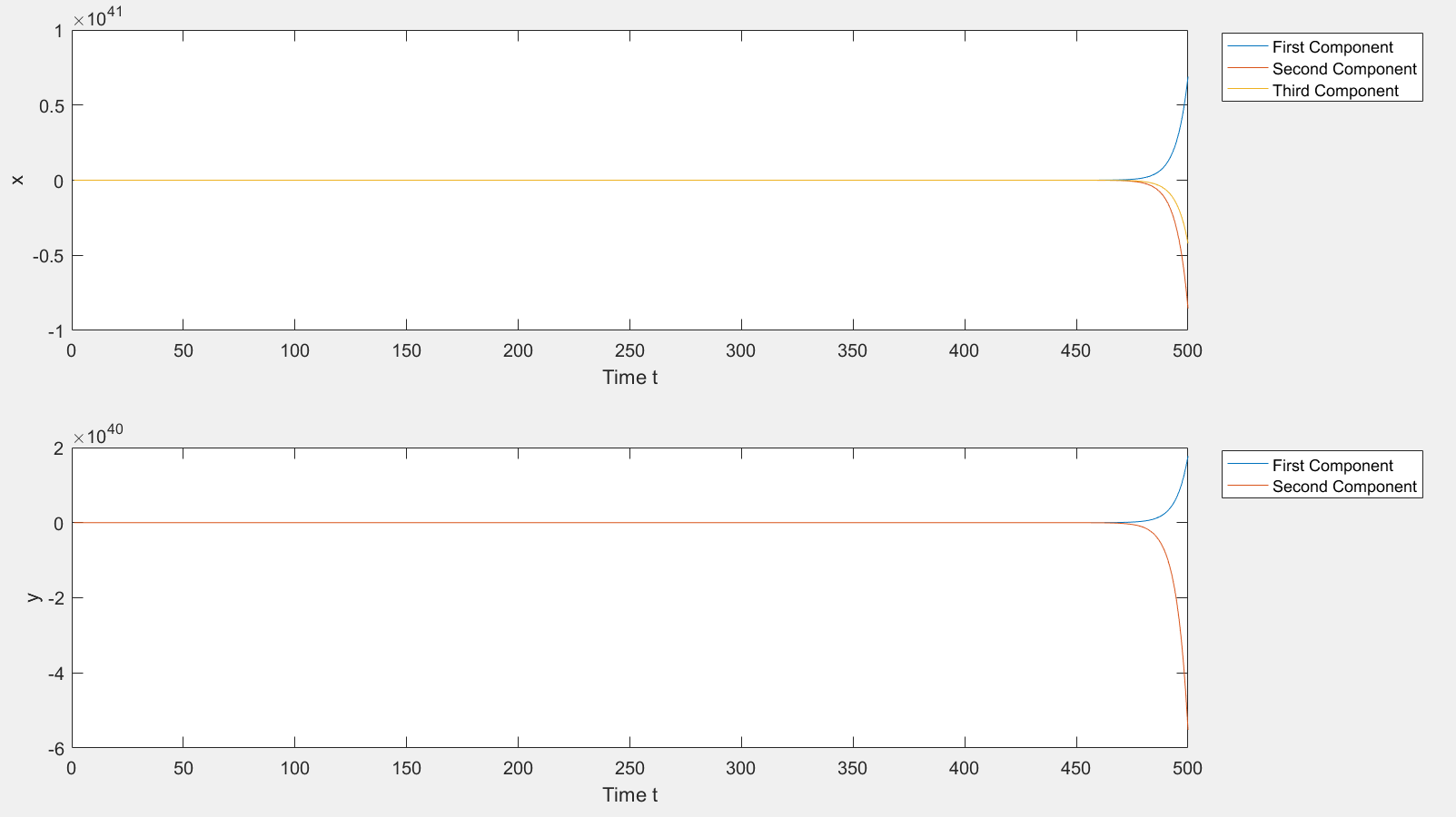
0.1305

0.3677

0.1953

Again, the estimation accuracies are not good for the filter algorithm, since we add too much noise.

Problem 3



Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 3.745753e-18.

> In HW6 (line 39)

Ahat =

-6.6469 -0.2122 -12.2516

4.9796 -0.9249 9.7750

1.5515 -0.5958 1.4990

What =

1.0e+79 \*

0.2742 -0.3399 -0.1674

-4.1184 5.1066 2.5153

-5.2565 6.5178 3.2104

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.

RCOND = 1.336928e-17.

> In HW6 (line 41)

Hhat =

0.0661 0.0411 -0.2749

-0.1748 -0.6717 1.3417

Qhat =

1.0e+79 \*

0.0579 -0.1797

-0.4951 1.5379

R2\_filter =

0.6938

0.9831

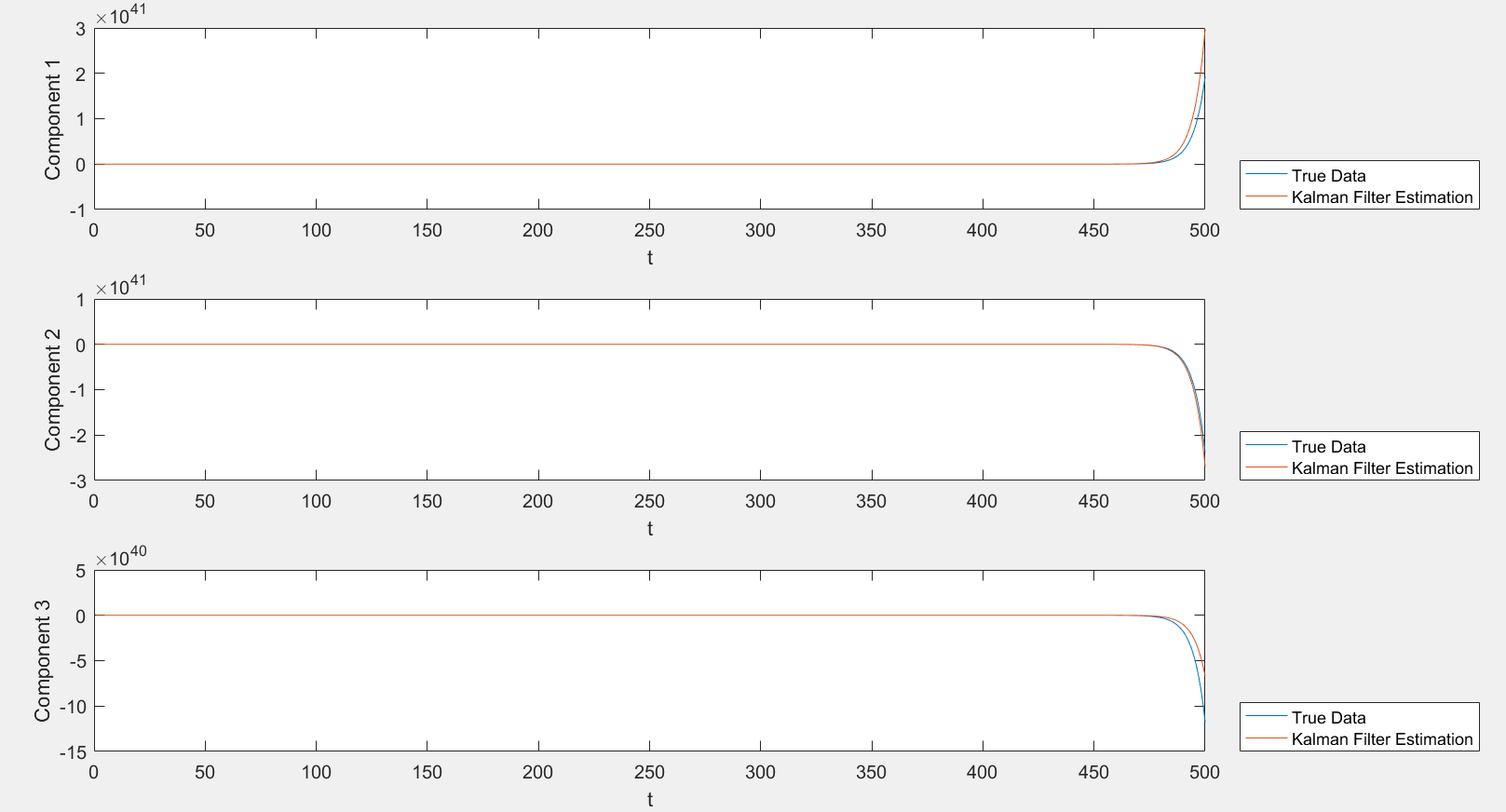
0.8115

R2\_predict =

-32.5593

-21.8737

-21.3095



The process simulated are very unstable (exploded/unbounded). From the warning of matlab, we know that it is related to the eigenvalues of the matrix A.

>> eig(A)

ans =

1.2091

0.7812

0.4097

As the eigenvalue of A is larger than 1, the system is not stable.

Code:

clear all; close all;

rng(1001);

%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 = 0.5\*eye(2);

%Q = 50\*eye(2);

% set initial conditions

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

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

%Problem (a)

% simulate the 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');

leg1 = 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');

leg2 = legend('First Component','Second Component','Location','NorthEastOutside');

%Problem (b)

% 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

%Problem (c)

% 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','southeastoutside')

end

%Problem (d)

% compute R\_squares\_error

R2\_filter=1-sum((x2-xh).^2,2)./sum((x2-mean(x2,2)\*ones(1,M)).^2,2) % filtering estimation

R2\_predict=1-sum((x2-xmh).^2,2)./sum((x2-mean(x2,2)\*ones(1,M)).^2,2) % prediction estimation