Problem 1:

1. The state and the observation stateare displayed in the first two subplots of Figure 1.

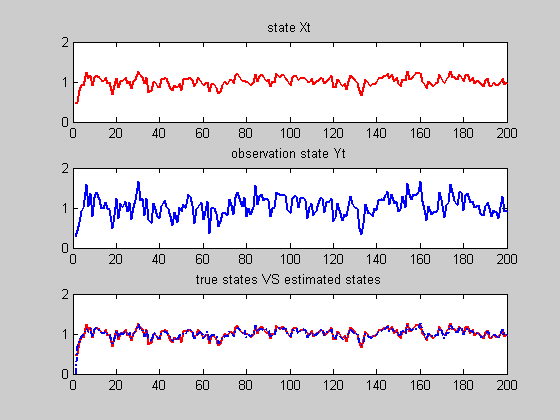


Figure 1: Display Xt, Yt and using SMCM simulate Xt

1. The third subplot of Figure 1 is the true and estimated states using Monte Carlo posterior mean. You can see the true states and the estimated states are of good match, Monte Carlo estimation gives very good simulation.

Then, we get the -error= 0.8479, which shows the estimation accuracy is very good.

1. The Kolmogorov-Smirnov test indicates that all the four posteriors are normal distributed. With all the acceptance of normality and p-values very high.

p =

0.2451 0.2169 0.7194 0.4918

h =

0 0 0 0

The histograms of the samples from the posterior are:

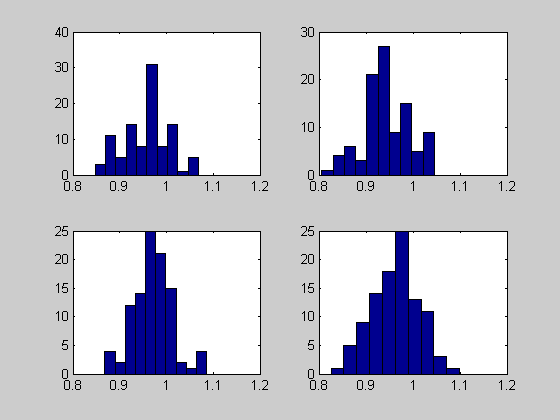


Figure 2 : the histograms of the samples from the posterior at t=50, 100, 150, 200:

Problem 2:

1. The state and the observation stateare displayed in the first two subplots of Figure 3.

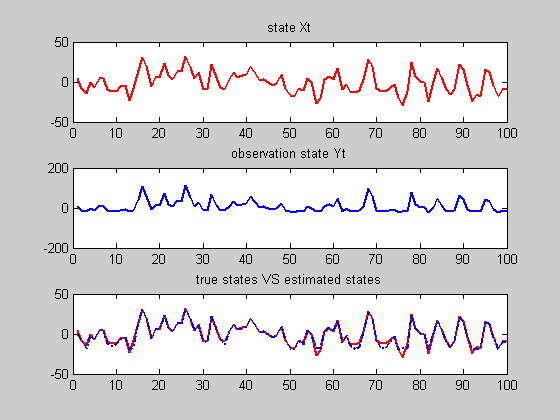


Figure 3: Display Xt, Yt and using SMCM simulate Xt



The third subplot of Figure 3 is the true and estimated states using Monte Carlo posterior mean. You can see the true states and the estimated states are of good match, Monte Carlo estimation gives very good simulation.

Then, we get the -error= 0.9676, which shows the estimation accuracy is very good.



The Kolmogorov-Smirnov test indicates that all the four posteriors are not normal distributed, with all the rejection of normality and p-values very small.

p =

1.0e-004 \*

0.6191 0.0000 0.0042 0.0000

h =

1 1 1 1

The histograms of the samples from the posterior are:

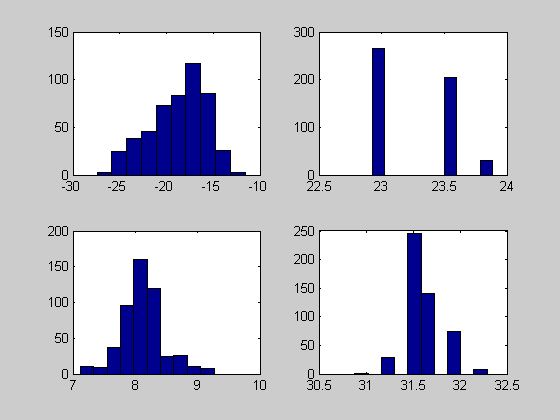


Figure 4: the histograms of the samples from the posterior at t=25, 50, 75, 100

Matlab code:

clear all; close all;

N=200;

u = 0.1\*randn(1);

v = 0.1\*randn(1);

x(1) = rand(1);

y(1) = x(1)^2 + v;

% simulate state Xt and observation Yt

for i=2:N

u = 0.1\*randn(1);

v = 0.1\*randn(1);

x(i)=sqrt(abs(x(i-1))) + u;

y(i)=x(i)^2+v;

end

% display Xt and Yt

figure (1);

subplot(311);

plot(x,'r','LineWidth',2);

title('state Xt');

subplot(312);

plot(y,'LineWidth',2);

title('observation state Yt');

% Sequential Monte Carlo method

% generate 100 samples

x\_s(1,:) = rand(1,100);

for i=2:200

% generate the prediction set

u = 0.1\*randn(1,100);

x\_pred(i,:)=sqrt(abs(x\_s(i-1))) + u;

% compute the weights and normalize

v = 0.1\*randn(1,100);

w(i,:) = normpdf(y(i),x\_pred(i,:).^2,0.1);

w\_n(i,:) = w(i,:)/sum(w(i,:));

W\_n(i,:) = cumsum(w\_n(i,:));

theta\_hat(i,:)= sum(w\_n(i,:).\*x\_pred(i,:));

% resample

for t=1:100

U = rand;

num = find(U-W\_n(i,:)<0);

x\_s(i,t) = x\_pred(i,num(1));

end

end

subplot(3,1,3);

plot(1:200,x,'r-','LineWidth',2);

title('true states VS estimated states');

hold on;

plot(1:200,theta\_hat,'-.', 'LineWidth',2);

%compute R\_square

X\_est = theta\_hat(11:200)';

R\_square= 1-sum((x(11:200)-X\_est).^2)/sum((x(11:200)-mean(X\_est)).^2);

% plot the histogram of the samples from posterior at t=50,100,150,200

figure (2);

for k=1:4

subplot(2,2,k);

data=x\_s(50\*k,:);

hist(data);

[h(k),p(k)]=kstest((data-mean(data))/std(data));

end

p

h