OPTIMAL FILTERING AND ESTIMATION

ASSIGNMENT 5

4) CODE

N = 100000;

x = [-5,-4,-3,-2,-1,0,1,2,3,4,5];

f\_x =[0.01,0.02,0.05,0.1,0.02,0.04,0.3,0.1,0.3,0.05,0.01];

len = length(x);

value=zeros(1,N);

for i=1:N

val= rand(1,1);

sum = 0;

for j=1:len

sum = sum+f\_x(j);

if(val <= sum)

value(1,i) = x(j);

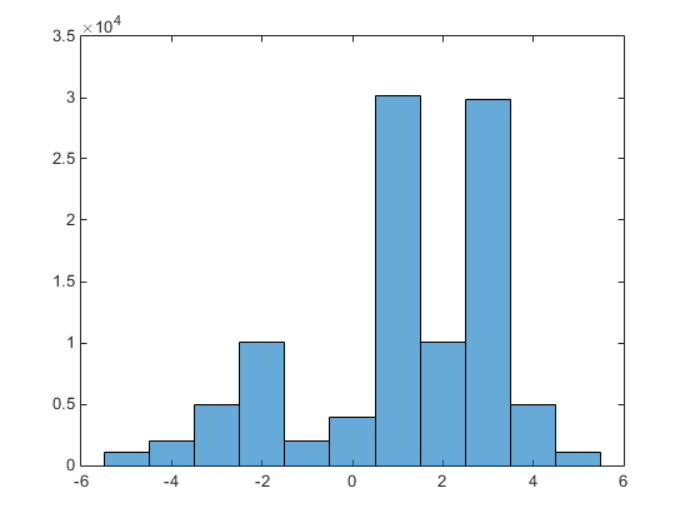
break;

end

end

end

histogram(value(1,:));

GRAPH

The graph is same as the distribution because we can see that at 1 and 3 the probability distribution is about 0.3 and we can also see that the probability distribution at 2 and -2 is 0.1.

Question 3:

4. Analytical method is very hard to solve .It gives a very hard integral to solve since the value varies from positive to negative infinity and there are many poles associated.

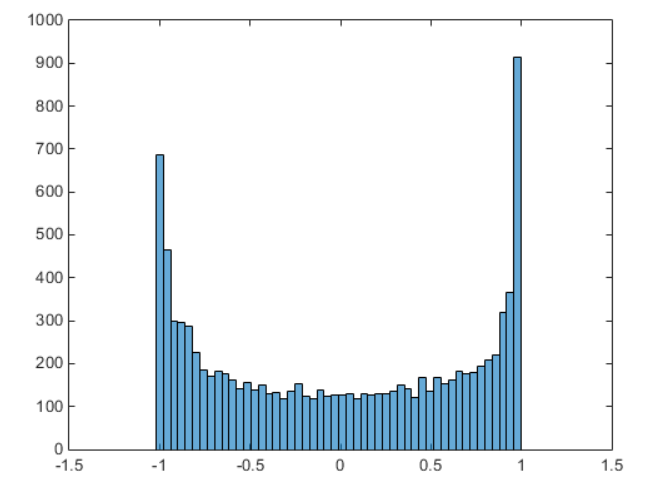
mean\_value =

-0.0096

var\_value =

0.5031

FOR UNIFORM GAUSSIAN DISTRIBUTION:



Part 1:

For Uniform distribution:

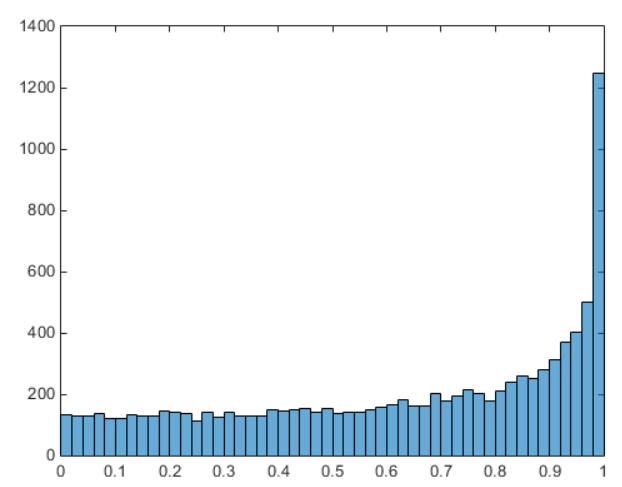
mean\_value =

0.6345

var\_value =

0.0945

GRAPH :



CODE:

N = 10000;

y = zeros(1,N);

for i=1:N

phi = (180)\*rand(1,1);

%phi = 180\*(randn(1,1)+1);

% To generate a random value gaussian curve

y(1,i) = sind(phi);

end

set(gcf,'color','w');

histogram(y(1,:),50);

mean\_value = mean(y(1,:))

var\_value = var(y(1,:))

QUESTION 6:

PARTICLE FILTER:

RMS ERROR VALUES:

FOR N = 10

rms\_x =

6.2876

FOR N = 100:

rms\_x =

7.2600

FOR N=1000

rms\_x =

8.1049

FOR Q =0.1

rms\_x =

8.2550

FOR Q =1 N =100

rms\_x =

7.2600

FOR Q=10

rms\_x =

7.3937

RMS ERROR FOR EKF: for N = 100

rms\_ekf =

12.7363

RMS ERROR FOR UKF: N = 100

rms\_ukf =

2.9182

In this particular case the performance of UKF is better than Particle Filter but in general Particle Filter performs better than both EKF and UKF.

And UKF performs better than EKF as can be seen from the statistics.

Since the program for Particle Filter is run seperately and not with UKF the comparison may not have that much meaning because the ditribution taken might be different.

CODE:

no\_particles = 10;

N = 100;

x\_act = zeros(N+1,1);

x\_act(1,1) = 0.1;

x = zeros(N+1,no\_particles);

y = zeros(N,no\_particles);

h = zeros(N,no\_particles);

q\_inter = zeros(N,no\_particles);

q = zeros(N,no\_particles);

R = 1;

for p = 2:N+1

w = randn(1,1)+1;

x\_act(p,1) = (x\_act(p-1,1)/2)+(25\*x\_act(p-1,1))/(1+(x\_act(p-1,1)^2))+8\*cos(1.2\*(p-2))+w;

end

for val=1:no\_particles

x(1,val) = 0.01;

end

for i = 2:N+1

for j=1:no\_particles

w = randn(1,1)+1;

x(i,j) = (x(i-1,j)/2)+(25\*x(i-1,j))/(1+(x(i-1,j)^2))+8\*cos(1.2\*(i-2))+w;

end

sum = 0;

for k=1:no\_particles

v = randn(1,1)+10;

y(i-1,k) = (x(i,k)^2)/20 +v;

h(i-1,k) = (x(i,k)^2)/20;

q\_inter(k,i-1) = (1/sqrt(2\*pi)\*R)\*exp(-(y(i-1,k)-h(i-1,k))^2\*R/2);

sum = sum+q\_inter(i-1,k);

end

for l=1:no\_particles

q(i-1,l)= q\_inter(i-1,l)/sum;

end

for no=1:no\_particles

val= rand(1,1);

sum\_new = 0;

for j=1:no\_particles

sum\_new = sum\_new+q(i-1,j);

if(val <= sum\_new)

x(i,no) = x(i,j);

break;

end

end

end

end

mean\_x = mean(x,2);

% to find root mean square error

rms\_x = 0;

error = zeros(N+1,1);

for n=1:N+1

error(n,1) =x\_act(n,1)-mean\_x(n);

rms\_x =rms\_x+(x\_act(n,1)-mean\_x(n))^2;

end

rms\_x= sqrt(rms\_x/(N+1))

figure

set(gcf,'color','w');

plot(error(:,1))

figure

set(gcf,'color','w');

plot(x\_act(:,1));

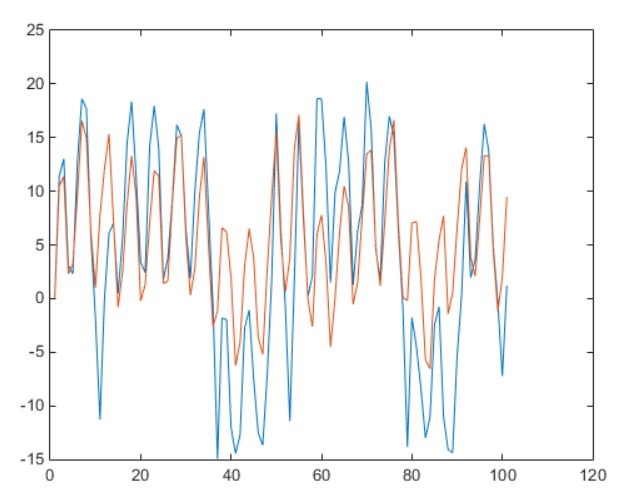
hold on

plot(mean\_x);

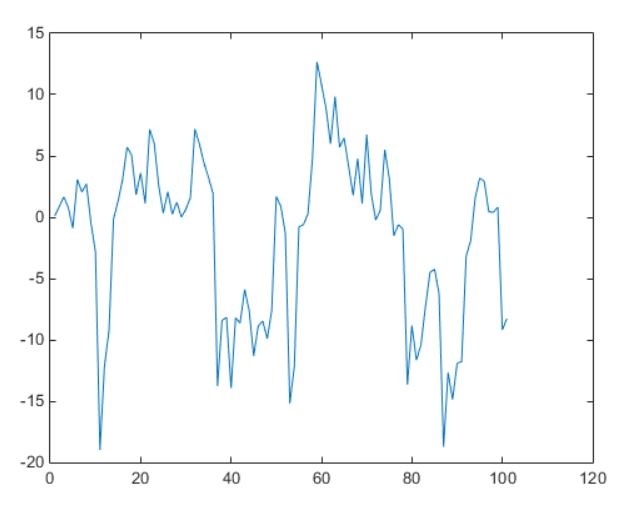
hold off

GRAPH:

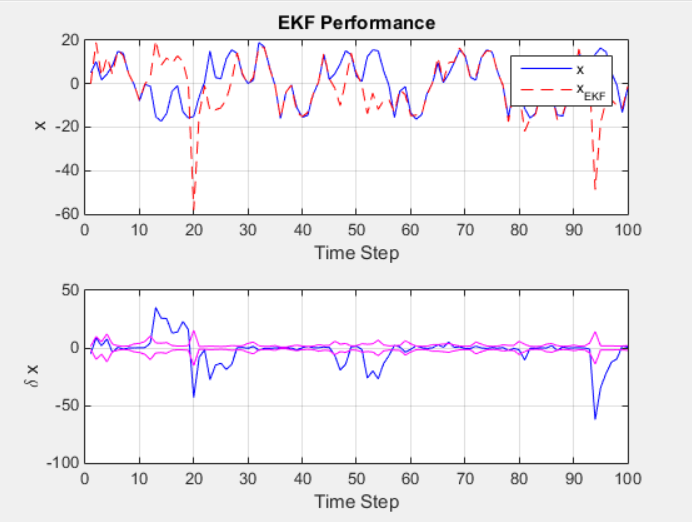
N= 100 and Q = 1

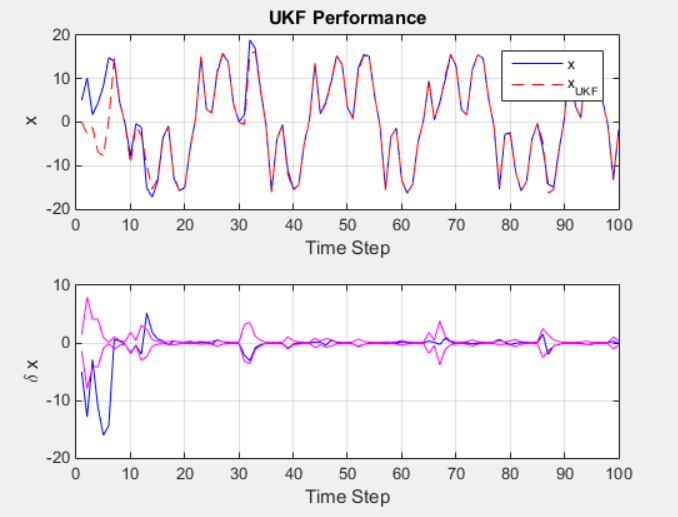


RMS ERROR:



EKF AND UKF :





Question 3:

PDF Distribution

x =[0:0.1:1];

len =length(x);

y = zeros(len,1)

for i=1:len

y(i) = 2/(pi\*(sqrt(1-x(i)^2)));

end

set(gcf,'color','w');

plot(y(:,1),x);

GRAPH:

