clear all;close all;

load('C:\Documents and Settings\goober\Desktop\midterm\_2\_train.mat');

[M, C0] = size(rate);

% centralization

mean\_pos = mean(kin(:,1:2)); % only need to do on pos as mean of vel and acc are already about 0

kin(:,1:2) = kin(:,1:2) - ones(M,1)\*mean\_pos;

mean\_rate = mean(rate);

rate = rate - ones(M,1)\*mean\_rate;

%Given A,H,W,Q simulate {x}{y}using Kalman filter

[R1,N]=size(kin')

[R2,N]=size(rate')

X=kin';

Y=rate';

%A,H,W,Q are unknown, identify them

%using {xt}{yt}in train dataset to estimate

%A=XX1\*XX2

%Ah=(X(:,2:end)\*X(:,1:end-1)')\*(X(:,1:end-1)\*X(:,1:end-1)')^-1;

xx1=0;

for k=2:N

xx1=xx1+X(:,k)\*X(:,k-1)';

end

xx2=0;

for k=2:N

xx2=xx2+X(:,k-1)\*X(:,k-1)';

end

Ah=xx1\*(xx2^-1);

%W

xx3=0;

for k=2:N

xx3=xx3+X(:,k)\*X(:,k)';

end

xx4=0;

for k=2:N

xx4=xx4+X(:,k-1)\*X(:,k)';

end

Wh=(xx3-Ah\*xx4)/(N-1);

%H

yx=0;

for k=1:N

yx=yx+Y(:,k)\*X(:,k)';

end

xx5=0;

for k=1:N

xx5=xx5+X(:,k)\*X(:,k)';

end

Hh=yx\*(xx5^-1);

%Q

yy=0;

for k=1:N

yy=yy+Y(:,k)\*Y(:,k)';

end

xy=0;

for k=1:N

xy=xy+X(:,k)\*Y(:,k)';

end

Qh=(yy-Hh\*xy)/N;

%Q=(Y(:,1:end)\*Y(:,1:end)'-H2\*X(:,1:end)\*Y(:,1:end)')/N

%use Kalman filter to estimate {xt} by {yt}&

%Ah,Hh,Wh,Qh

load('C:\Documents and Settings\goober\Desktop\midterm\_2\_test.mat');

%centralize {yt}-rate

[M, C0] = size(rate);

rate = rate - ones(M,1)\*mean\_rate;

d = size(kin,2);%d=dimension of xt=4 in this case

[M1,C]=size(kin');

[M2,C]=size(rate');

Xt=kin';

Yt=rate';

%initials% define the size of all matrices

N=C;

Xm=zeros(d,N);

P=zeros(d,d,N);

Pm=zeros(d,d,N);

K=zeros(d,M2,N);

Xm(:,1)=zeros(d,1);

P(:,:,1)=zeros(d,d);

Xh=zeros(d,N);

Xh(:,1)=randn(d,1);

I=eye(d);

for k=2:N

Xm(:,k)=Ah\*Xh(:,k-1);

Pm(:,:,k)=Ah\*P(:,:,k-1)\*Ah'+Wh;

K(:,:,k)=Pm(:,:,k)\*Hh'\*(Hh\*Pm(:,:,k)\*Hh'+Qh)^-1;

P(:,:,k)=(I-K(:,:,k)\*Hh)\*Pm(:,:,k);

Xh(:,k)=Xm(:,k)+K(:,:,k)\*(Yt(:,k)-Hh\*Xm(:,k));

end

Xh(1:2,:) = Xh(1:2,:) + mean\_pos'\*ones(1,N);%add mean of kin back to estimated kin (xt)

Xm(1:2,:) = Xm(1:2,:) + mean\_pos'\*ones(1,N);

figure(1)

subplot(211)

xp1=[Xh(1,:);Xt(1,:)];

plot(xp1');

title('x-position');

subplot(212)

xp2=[Xh(2,:);Xt(2,:)];

plot(xp2');

title('y-position');

%compute the R^2 error of each component in

%xh(estimated):Xt vs Xh

%then compare with the R^2 error

%with xm(intemediate):Xt vs Xm

Rs1=zeros(2,1);

for i=1:2;

r1=0;

for k=1:N

r1=r1+(Xt(i,k)-Xh(i,k))^2;

end

r2=0;

for k=1:N

r2=r2+(Xt(i,k)-mean(Xt(i,:)))^2;

end

Rs1(i)=1-(r1/r2);

end

Rs1% R^2 in xt(estimated)

Rm=zeros(2,1);

for i=1:2;

r3=0;

for k=1:N

r3=r3+(Xt(i,k)-Xm(i,k))^2;

end

r4=0;

for k=1:N

r4=r4+(Xt(i,k)-mean(Xt(i,:)))^2;

end

Rm(i)=1-(r3/r4);

end

Rm

% Using Sequential Monte Carlo Methods to infer the hand movement

% clear ('x2','kin','y2','rate','N','xh','xmh','P','Pm','K','k','t');

% clear ('kin','M','rate','t');

load('C:\Documents and Settings\goober\Desktop\midterm\_2\_test.mat');

[M,C]=size(rate);

rate = rate - ones(M,1)\*mean\_rate;

d=size(kin,2);

x3=kin';

y3=rate';

n=20;

X=zeros(d,M);

S=zeros(d,n,M);

xh=ones(d,n,M);

xh(:,:,1)=randn(d,1,n);

S(:,:,1)= randn(d,1,n);

theta=zeros(d,M);

w=ones(n,M)/n;

w2=ones(n,M);

weight=ones(M,1);

for t=2:M

%prediction step

for j=1:n

mu=ones(4,n);

mu(:,j)=Ah\*xh(:,j,t-1);

S(:,j,t)=mvnrnd(mu(:,j),Wh);

%update

w(j,t)=mvnpdf(y3(:,t),Hh\*S(:,j,t),Qh);

end

weight(t)=sum(w(:,t));

w2(:,t)=w(:,t)/weight(t);

theta(:,t)=S(:,:,t)\*w2(:,t);

for j=1:n

v=[1:n];

r=randsample(v,1,true,w2(:,t));

xh(:,j,t)=S(:,r,t);

end

end

X=theta;

X(1:2,:) = X(1:2,:) + mean\_pos'\*ones(1,M);%add mean of kin back to estimated kin (xt)

figure(20)

subplot(211)

xp1=[X(1,:);x3(1,:)];

plot(xp1');

legend('Estimated','True data')

title('x-position');

subplot(212)

xp2=[X(2,:);x3(2,:)];

plot(xp2');

legend('Estimated','True data')

title('y-position');

Rs2=zeros(2,1);

for i=1:2

r1=0;

for k=1:M

r1=r1+(x3(i,k)-X(i,k))^2;

end

r2=0;

for k=1:M

r2=r2+(x3(i,k)-mean(x3(i,:)))^2;

end

Rs2(i)=1-(r1/r2);

end

Rs2