```
clear
%generate data for a state space model
%Y=Beta[t]*X+e1
%Beta[t]=mu+F*Beta[t-1]+e2
%var(e1)=R
%var(e2)=Q
t=500;
Q=0.001;
R=0.01;
F=1;  %these are fixed
mu=0;  %these are fixed
e1=randn(t,1)*sqrt(R);
e2=randn(t,1)*sqrt(Q);
Beta=zeros(t,1);
Y=zeros(t,1);
X=randn(t,1);
for j=2:t
    Beta(j,:)=Beta(j-1,:)+e2(j,:);
    Y(j)=X(j,:)*Beta(j,:)'+e1(j);
end
%%Step 1 Set up matrices for the Kalman Filter
beta0=zeros(1,1);   %state variable  b[0/0]
p00=1;          %variance of state variable p[0/0]
beta_tt=[];          %will hold the filtered state variable
ptt=zeros(t,1,1);    % will hold its variance
%initialise the state variable
beta11=beta0;
p11=p00;
for i=1:t
    x=X(i);
    %Prediction
beta10=mu+beta11*F';
p10=F*p11*F'+Q;
yhat=(x*(beta10)')';
eta=Y(i,:)-yhat;
feta=(x*p10*x')+R;
%updating
K=(p10*x')*inv(feta);
beta11=(beta10'+K*eta')';
p11=p10-K*(x*p10);
ptt(i,:,:)=p11;
beta_tt=[beta_tt;beta11];
end
%%%%%%%%%%end of Kalman Filter%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Carter and Kohn Backward recursion to calculate the mean and variance of the distribution of the state
%vector
beta2 = zeros(t,1);   %this will hold the draw of the state variable
wa=randn(t,1);
i=t;  %period T
p00=squeeze(ptt(i,:,:));
```

$p_{T\backslash T}$

$\beta_{T\backslash T}$

```
beta2(i,:)=beta_tt(i:i,:)+(wa(i:i,:)*chol(p00));
```

$\beta_T \sim N(\beta_{T\backslash T}, p_{T\backslash T})$

```
%periods T-1..to 1
for i=t-1:-1:1
pt=squeeze(ptt(i,:,:));
```

```
bm=beta_tt(i:i,:)+(pt*F'*inv(F*pt*F'+Q)*(beta2(i+1:i+1,:)-mu-beta_tt(i,:)*F')')';
```

$$\beta_{t\backslash t,B_{t+1}} = \beta_{t\backslash t} + p_{t\backslash t}F'(Fp_{t\backslash t}F' + Q)^{-1}(\beta_{t+1} - \mu - F\beta_{t\backslash t})$$

```
pm=pt-pt*F'*inv(F*pt*F'+Q)*F*pt;
```

$$p_{t\backslash t,B_{t+1}} = p_{t\backslash t} - p_{t\backslash t}F'(Fp_{t\backslash t}F' + Q)^{-1}Fp_{t\backslash t}$$

```
beta2(i:i,:)=bm+(wa(i:i,:)*chol(pm));
```

$$\beta_t \backslash \beta_{t+1} \sim N(\beta_{t\backslash t,B_{t+1}}, p_{t\backslash t,B_{t+1}})$$

```
end
plot([beta_tt beta2 Beta])
axis tight
legend('Kalman filter estimated \beta_{t}','Draw from H(\beta_{t})','true \beta_{t}');
```