Table of Contents

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Macro HW 9
% Tianli Xia November 29th
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
```

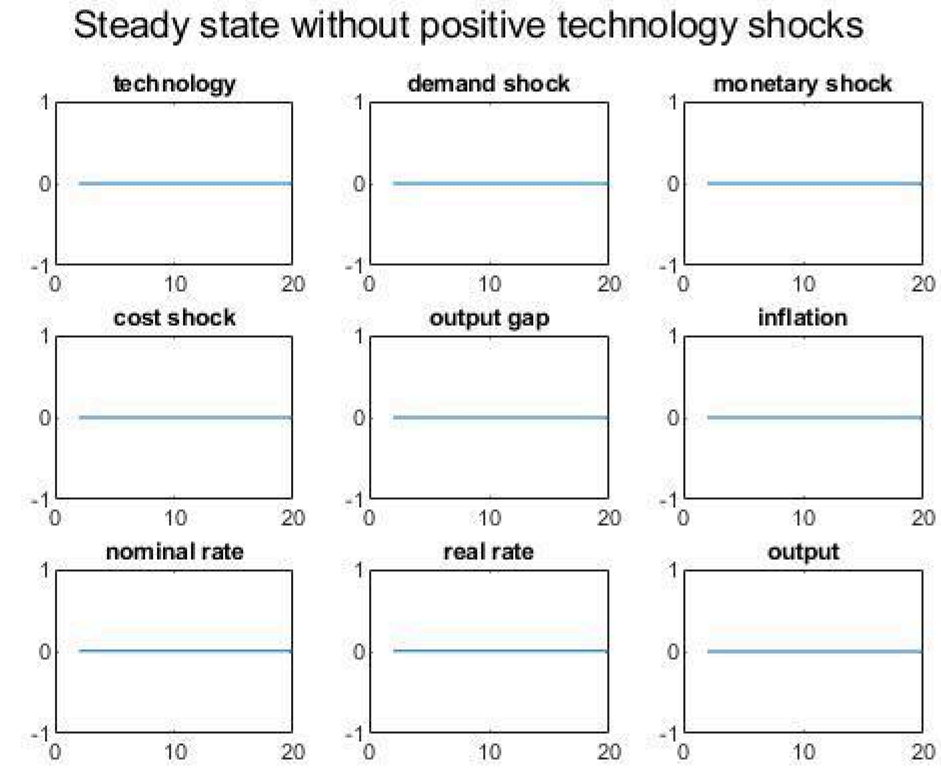
Gnerate the model

```
Model= DSGE;
Model= Model.init; % Find parameters
Model= Model.solve;
```

Impulse Response

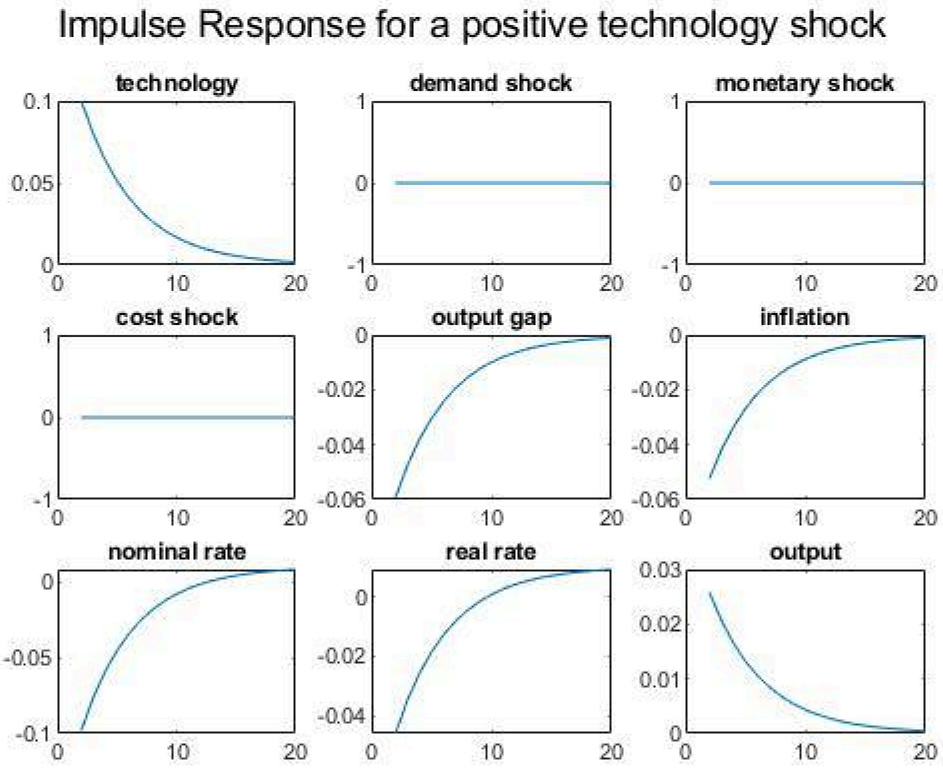
A positive tachnology shock

```
v=zeros(4,20);
Model.shock(v, 'q1_1', 'Steady state without positive technology
 shocks');
v(1,2)=0.1; % Set a technology shock
Model.shock(v, 'q1_2', 'Impulse Response for a positive technology
 shock');
v=zeros(4,20);
v(2,2)=0.1; % Set a demand shock
Model.shock(v, 'q1_3', 'Impulse Response for a positive demand
 shock');
v=zeros(4,20);
v(3,2)=0.1; % Set a monerary policy shock
Model.shock(v, 'q1_4', 'Impulse Response for a positive monerary
 shock');
v=zeros(4,20);
v(4,2)=0.1; % Set a cost-push shock
Model.shock(v, 'q1_5', 'Impulse Response for a positive cost-push
 shock');
```



Steady state without positive technology shocks

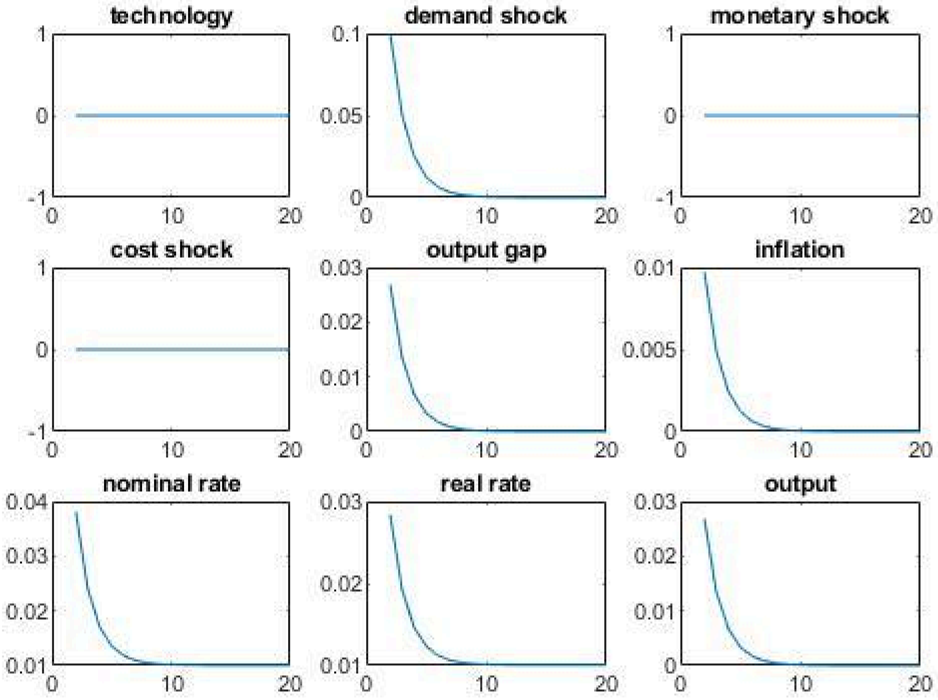
technology; demand shock; monetary shock; cost shock; output gap; inflation; nominal rate; real rate; output



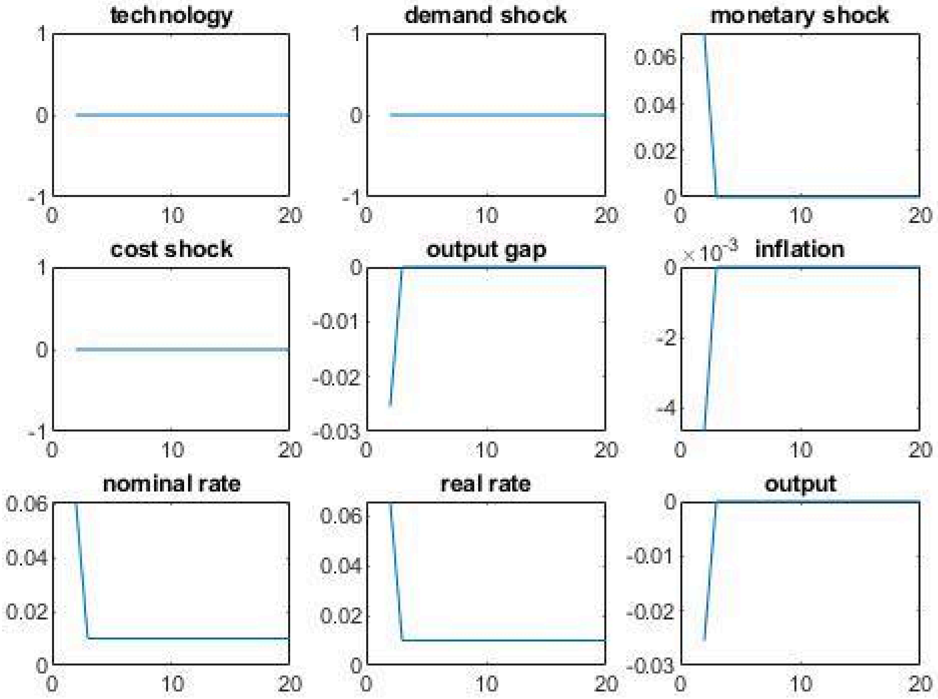
Impulse Response for a positive technology shock

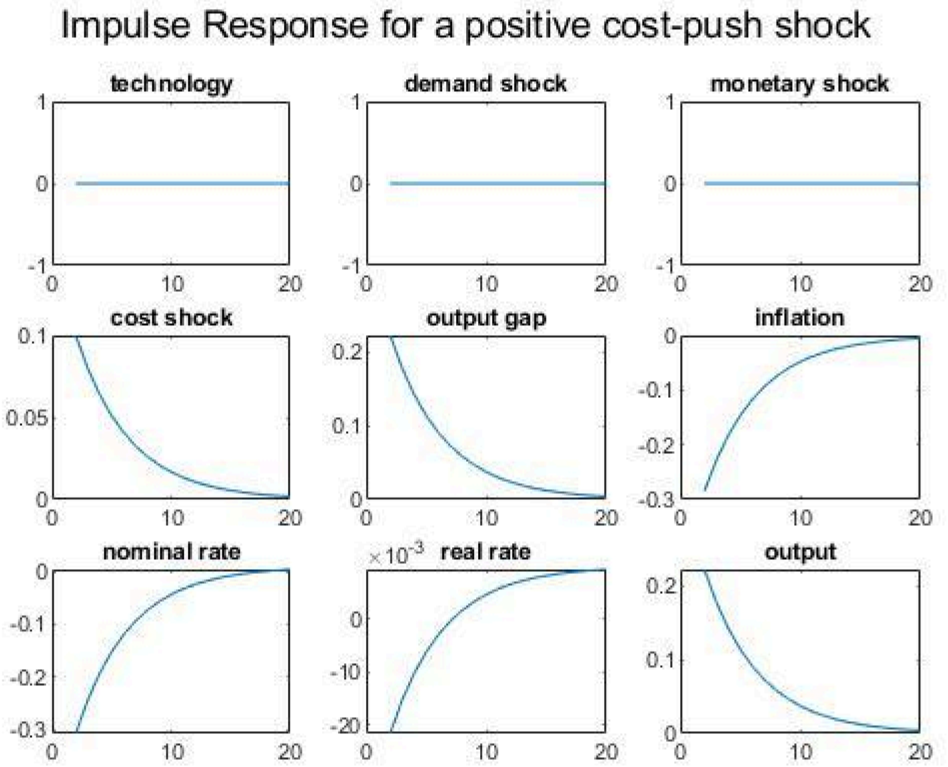
technology; demand shock; monetary shock; cost shock; output gap; inflation; nominal rate; real rate; output

Impulse Response for a positive demand shock



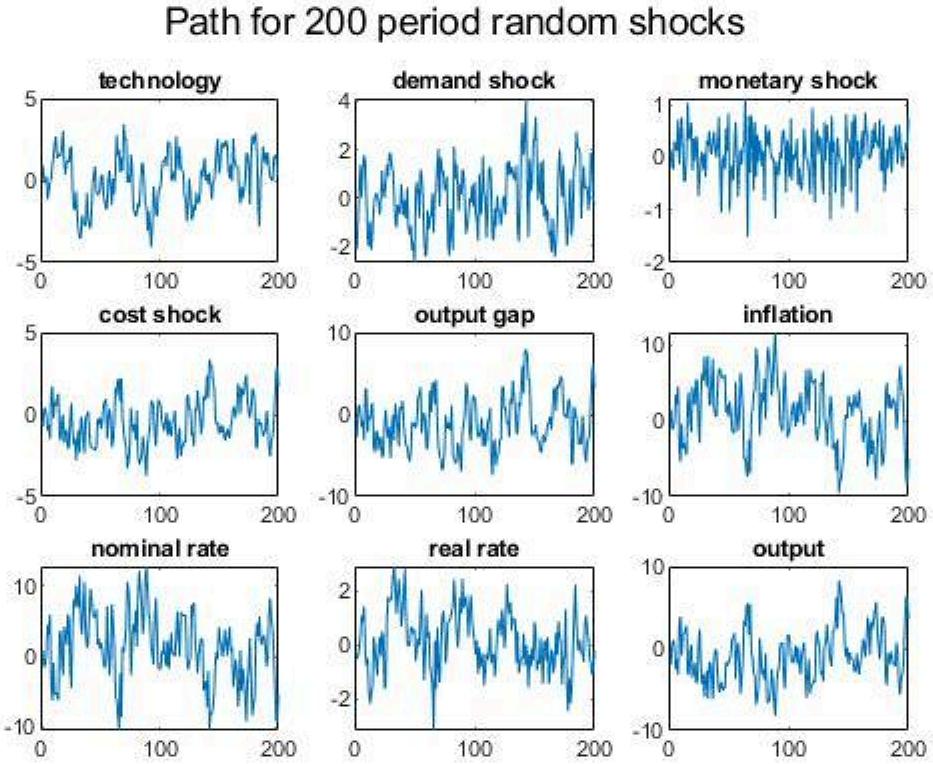
Impulse Response for a positive monerary shock





Simulating: introduce a 200 period random shock:

```
v= Model.C*randn(4,200); % tech shock
[X1, X2, ir, y, r]= Model.shock(v, 'q2', 'Path for 200 period random
 shocks');
```



Path for 200 period random shocks

technology

demand shock

monetary shock

cost shock

output gap

inflation

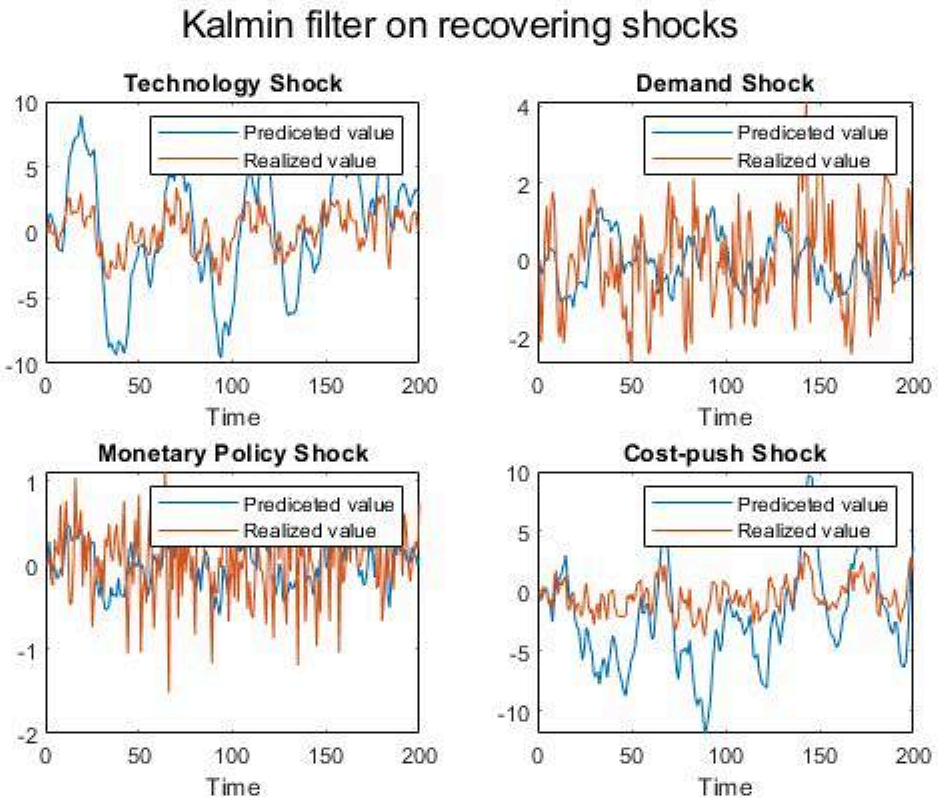
nominal rate

real rate

output

Kalmin Filter Estimation

```
output = Model.kalmin(X1, X2);
```

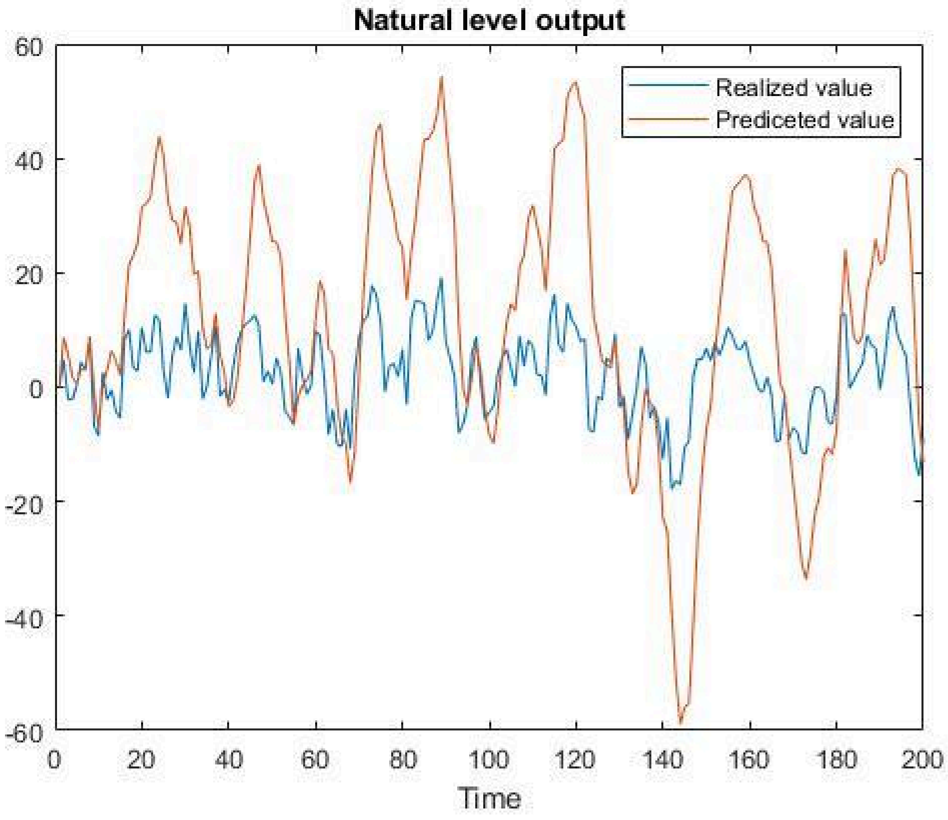


Question 2

Now suppose that we could observe X1=[pi, y, ir], then the filter could be transformed into:

```
pi=X2(1,:);
y= X2(2,:)+Model.ye;
a= X1(1,:);
u= X1(4,:);
ytn= Model.psiya*a + Model.ye - u/Model.kappa;
ztt= [pi; y-Model.ye; ir-Model.psipi*pi-Model.psiy*(y-Model.ye)-
Model.rho];
% Above is some transformation to make filter feasible
C= Model.C;
M= Model.M;
G= [Model.G; 0 0 1 0]+ [0 0 0 0; Model.psiya 0 0 0; 0 0 0 0];
n1=4;

W=zeros(3);

% Step1. recover latent variables
ptt1(:,:,1)= inv(eye(n1)-M)*C*C';
xtt1(:,1)= zeros(4,1);
i=1;
kt(:,:,i)=ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W');
ptt(:,:,i)=ptt1(:,:,i)-
ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W')*G*ptt1(:,:,i);
```

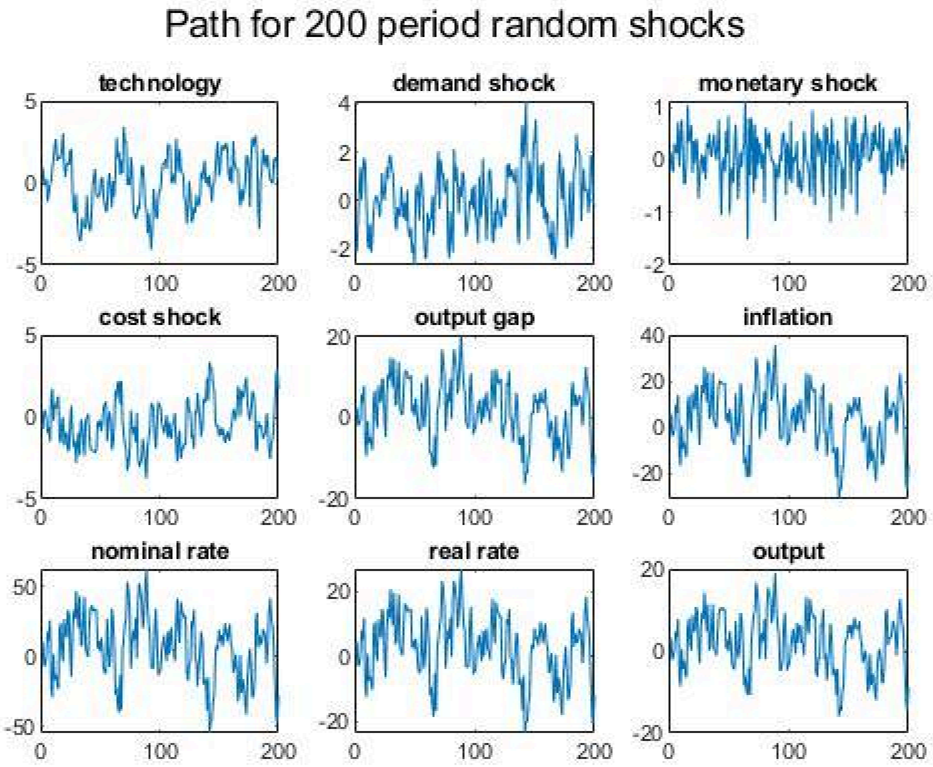
```
ptt1(:,:,i+1)=M*ptt(:,:,i)*M'+C*C';
xtt1(:,i+1)= M*xtt1(:,i);
xtt(:,i)= M*zeros(4,1) +kt(:,:,i)*(ztt(:,i)-G*xtt1(:,i));

Model.period=200;
for i=2:Model.period
    kt(:,:,i)=ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W');
    ptt(:,:,i)=ptt1(:,:,i)-
ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W')*G*ptt1(:,:,i);
    ptt1(:,:,i+1)=M*ptt(:,:,i)*M'+C*C';
    xtt1(:,i+1)= M*xtt1(:,i);
    xtt(:,i)= M*xtt(:,i-1)+kt(:,:,i)*(ztt(:,i)-G*xtt1(:,i));
end

% Step2. use latent variable to find ytn
 ytnhat= Model.psiya*xtt(1,:) + Model.ye - xtt(4,:)/Model.kappa;

% Step3. Plot them
time=1:1:Model.period;
plot(time, ytn)
hold on
plot(time, ytnhat)
legend("Realized value","Prediceted value")
xlabel("Time")
title("Natural level output")
print -djpeg -r600 q4.jpg
```

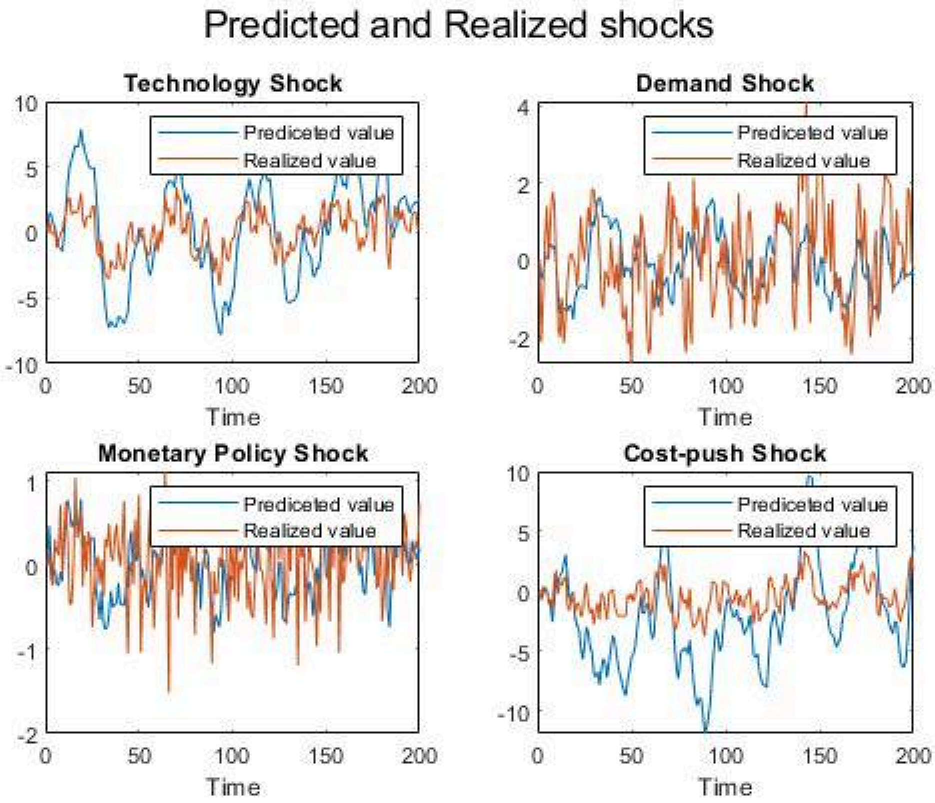


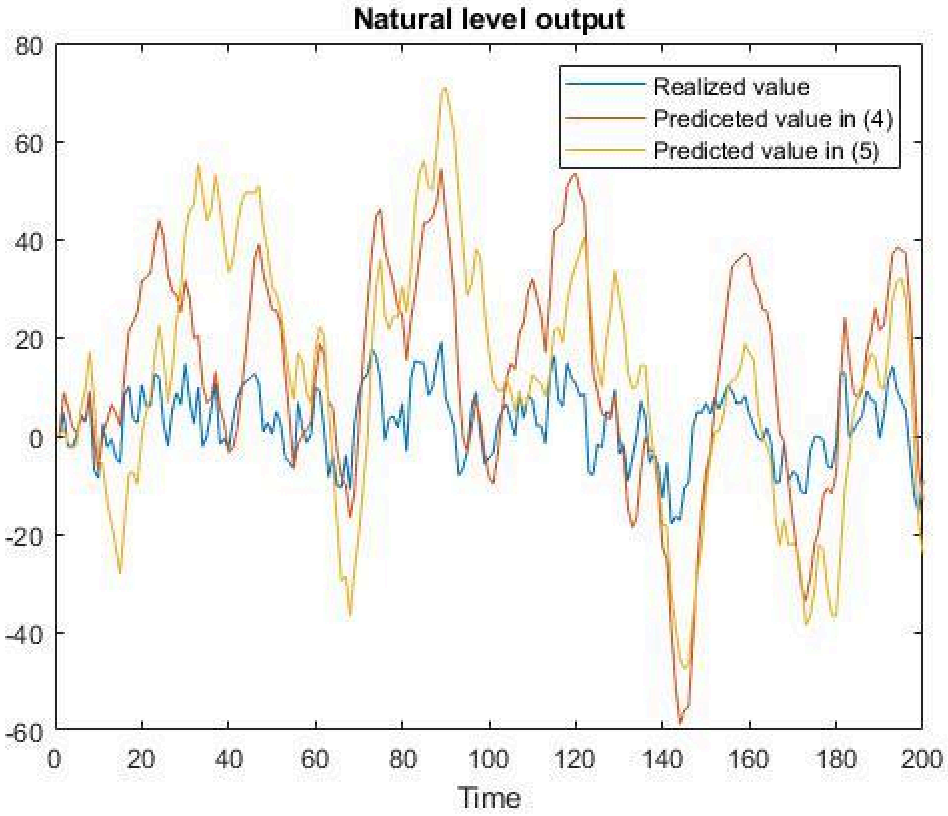
Question 5: Solve the model

```
Model= Model.solve2; % Numerically solve the model
[X1, X2, ir, y, r]= Model.shock(v, 'q5', 'Path for 200 period random
 shocks');
output2 = Model.kalmin(X1, X2, 'q5_1', "Predicted and Realized
 shocks");

% Question 5: Plot
pi=X2(1,:);
y= X2(2,:)+Model.ye;
a= X1(1,:);
u= X1(4,:);
ytn= Model.psiya*a - Model.ye - u/Model.kappa;
ztt= [pi; y-Model.ye; ir-Model.psipi*pi-Model.psiy*(y-Model.ye)-
Model.rho];
% Above is some transformation to make filter feasible
C= Model.C;
M= Model.M;
G= [Model.G; 0 0 1 0]+ [0 0 0 0; Model.psiya 0 0 0; 0 0 0 0];
n1=4;

W=zeros(3);

% Step1. recover latent variables
ptt1(:,:,1)= inv(eye(n1)-M)*C*C';
xtt1(:,1)= zeros(4,1);
i=1;
kt(:,:,i)=ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W');
ptt(:,:,i)=ptt1(:,:,i)-
ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W')*G*ptt1(:,:,i);
ptt1(:,:,i+1)=M*ptt(:,:,i)*M'+C*C';
xtt1(:,i+1)= M*xtt1(:,i);
xtt(:,i)= M*zeros(4,1) +kt(:,:,i)*(ztt(:,i)-G*xtt1(:,i));

Model.period=200;
for i=2:Model.period
    kt(:,:,i)=ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W');
    ptt(:,:,i)=ptt1(:,:,i)-
ptt1(:,:,i)*G'*inv(G*ptt1(:,:,i)*G'+W*W')*G*ptt1(:,:,i);
    ptt1(:,:,i+1)=M*ptt(:,:,i)*M'+C*C';
    xtt1(:,i+1)= M*xtt1(:,i);
    xtt(:,i)= M*xtt(:,i-1)+kt(:,:,i)*(ztt(:,i)-G*xtt1(:,i));
end

% Step2. use latent variable to find ytn
 ytnhat2= Model.psiya*xtt(1,:) + Model.ye - xtt(4,:)/Model.kappa;

% Step3. Plot them
time=1:1:Model.period;
plot(time, ytn)
hold on
plot(time, ytnhat)
```

```
hold on
plot(time, ytnhat2)
legend("Realized value","Prediceted value in (4)","Predicted value in
 (5)")
xlabel("Time")
title("Natural level output")
print -djpeg -r600 q5_2.jpg
```




Predicted and Realized shocks
Technology Shock
Prediceted value
Realized value
10
5
0
-5
-10
0
50
100
150
200
Time
Demand Shock
Prediceted value
Realized value
4
2
0
-2
0
50
100
150
200
Time
Monetary Policy Shock
Prediceted value
Realized value
1
0
-1
-2
0
50
100
150
200
Time
Cost-push Shock
Prediceted value
Realized value
10
5
0
-5
-10
0
50
100
150
200
Time


Natural level output
Realized value
Prediceted value in (4)
Predicted value in (5)
80
60
40
20
0
-20
-40
-60
0
20
40
60
80
100
120
140
160
180
200
Time

Published with MATLAB® R2018a