```
clear
addpath('functions');
[ data0 junk ]=xlsread('\data\datain.xls');
[ junk names ]=xlsread('\data\names.xls');
%names=names(1,2:end);
index=xlsread('\data\index.xls');
dindex=index(:,1); %dindex=1 for series that are log differenced dindex=3 differencing without logs
index=index(:,2);  %index=1 for 'fast moving' series
%first difference the data where appropriate
data=[];
for i=1:cols(data0);
    if dindex(i)==1
        dat=log(data0(:,i));
        dat=diff(dat)*100;
    elseif dindex(i)==3
        dat=diff(data0(:,i));
    else
        dat=data0(2:end,i);
    end
    data=[data dat];
end
%standardise the data
data=standardise(data);
%load policy rate and standardize it
z=xlsread('\data\baserate.xls');
z=z(2:end);
z=standardise(z);
KK=3;  %number of factors
L=2;  %number of lags in the VAR
N=KK+1; %number of Variables in var K factors plus the interest rate
NN=cols(data);% size of the panel
T=rows(data)
%step 1 of the algorithm set starting values and priors
%get an intial guess for the factor via principal components
pmat=extract(data,KK);
```

$\beta_{0\backslash 0}$

```
beta0=[pmat(1,:) z(1) zeros(1,N)];  %state vector S[t-1/t-1]
ns=cols(beta0);
```

$p_{0\backslash 0}$

```
P00=eye(ns);  %P[t-1/t-1]
```

One can arbitrarily set $R_{ii} = 1$ and $\Omega$ to an identity matrix to start the algorithm

```
rmat=ones(NN,1); %arbitrary starting value for the variance of the idiosyncratic component
Sigma=eye(N);  %arbitrary starting value for the variance of VAR errors
%flat prior for the factor loadings,variances and VAR
reps=5000;
burn=4000;
mm=1;
for m=1:reps;
%gibbs sampling
%step 2 sample factor loadings
fload=[];
floadr=[];
error=[];
for i=1:NN
    y=data(:,i);
    if index(i)==0
```

```
        x=pmat;
```

$$Z_t = \{F_{1t}, F_{2t}, F_{3t}\}$$

```
    else
        x=[pmat z];
```

$$Z_t = \{F_{1t}, F_{2t}, F_{3t}, FFR_t\}$$

```
    end
    M=inv(x'*x)*(x'*y);
```

$H_i^* = \left(\Sigma_{H_i}^{-1} + \frac{1}{R_{ii}} Z_t' Z_t\right)^{-1} \left(\Sigma_{H_i}^{-1} H_{i0} + \frac{1}{R_{ii}} Z_t' X_{it}\right)$ with a flat prior

$V_i^* = \left(\Sigma_{H_i}^{-1} + \frac{1}{R_{ii}} Z_t' Z_t\right)^{-1}$ with a flat prior

```
    V=rmat(i)*inv(x'*x);
    %draw
    ff=M+(randn(1,cols(x))*cholx(V))';

    %save
    if index(i)==0;
        fload=[fload;ff'];
        floadr=[floadr;0];
    else
            fload=[fload;ff(1:end-1)'];
        floadr=[floadr;ff(end)];
    end
    error=[error y-x*ff];
end
%for identification top K by K block of fload is identity
fload(1:KK,1:KK)=eye(KK);
%for identification top K by 1 block of Floadr is zero
floadr(24:24+KK-1,1)=zeros(KK,1);
%step 3 sample variance of the idiosyncratic components from inverse
%gamma
```

sample the variance of the error terms of the observation equation $R_{ii}$ from the inverse Gamma distribution with scale parameter $(X_{it} - Z_t H_i)'(X_{it} - Z_t H_i) + R_{ii0}$ with degrees of freedom $T + V_{R0}$ where $T$ is the length of the estimation sample.

```
rmat=[];
for i=1:NN
    rmati= IG(0,0,error(:,i));
    rmat=[rmat rmati];
end
%step 4 sample VAR coefficients
Y=[pmat z];
X=[lag0(Y,1) lag0(Y,2) ones(rows(Y),1)];
Y=Y(2:end,:);
X=X(2:end,:);
M=vec(inv(X'*X)*(X'*Y));  %conditional mean
```

$B^* = (\Sigma_B^{-1} + \Omega^{-1} \otimes \bar{X}_t' \bar{X}_t)^{-1} \left(\Sigma_B^{-1} vec(B_0) + \Omega^{-1} \otimes \bar{X}_t' \bar{X}_t vec(\hat{B})\right)$ with a flat prior

```
V=kron(Sigma,inv(X'*X)); %conditional variance
```

$D^* = (\Sigma_B^{-1} + \Omega^{-1} \otimes \bar{X}_t' \bar{X}_t)^{-1}$ with a flat prior

```
chck=-1;                    %make sure VAR is stationary
while chck<0
beta=M+(randn(1,N*(N*L+1))*cholx(V))';  %draw for VAR coefficients
S=stability(beta,N,L);
if S==0
    chck=10;
end
end
beta1=reshape(beta,N*L+1,N);
errorsv=Y-X*beta1;
```

Conditional on the factors $F_t$ and the VAR coefficients $B$ the error covariance $\Omega$ has a inverse Wishart posterior with scale matrix $(Y_t - \bar{X}_t B)'(Y_t - \bar{X}_t B) + \Omega_0$ and degrees of freedom $T$+$V_0$.

```
%sample VAR covariance
scale=errorsv'*errorsv;
Sigma=iwpQ(T,inv(scale));
%step 5 prepare matrices for the state space
%Y=H*factors+e
%factors=MU+F*Factors(-1)+v
%e~N(0,R)
%v~N(0,Q)
%matrix of factor loadings
```

$$\underbrace{\begin{pmatrix} b_{11} & . & b_{13} & \gamma_1 & 0 & 0 & 0 & 0 \\ b_{21} & & . & . & 0 & 0 & 0 & 0 \\ . & & . & . & 0 & 0 & 0 & 0 \\ . & & . & . & 0 & 0 & 0 & 0 \\ . & & . & . & 0 & 0 & 0 & 0 \\ . & & . & . & 0 & 0 & 0 & 0 \\ . & & . & . & 0 & 0 & 0 & 0 \\ b_{M1} & & b_{M3} & \gamma_M & 0 & 0 & 0 & 0 \\ & & & 1 & 0 & 0 & 0 & 0 \end{pmatrix}}_{H}$$

```
H=zeros(NN,(KK+1)*L);
H(1:rows(fload),1:KK+1)=[fload floadr];
H(rows(floadr)+1,KK+1)=1;
%matrix R
```

$$VAR(v_t) = R = \begin{pmatrix} R_1 & 0 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & R_M & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
R=diag([rmat 0]);
%vector MU
```

$$\underbrace{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\mu}$$

```
MU=[beta1(end,:)';zeros(N*(L-1),1)]';
%matrix F
```

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} & A_{18} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & A_{27} & A_{28} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} & A_{37} & A_{38} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} & A_{47} & A_{48} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$F$

```
F=[beta1(1:N*L,:)';eye(N*(L-1),N*L)];
%matrix Q
```

$$VAR(e_t) = Q = \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} & 0 & 0 & 0 & 0 \\ Q_{12} & Q_{22} & Q_{23} & Q_{24} & 0 & 0 & 0 & 0 \\ Q_{13} & Q_{23} & Q_{33} & Q_{34} & 0 & 0 & 0 & 0 \\ Q_{14} & Q_{24} & Q_{23} & Q_{44} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
Q=zeros(rows(F),rows(F));
Q(1:N,1:N)=Sigma;
%Carter and Kohn algorithm to draw the factor
beta_tt=[];          %will hold the filtered state variable
ptt=zeros(T,ns,ns);    % will hold its variance
% %%%%%%%%%%%Step 6a run Kalman Filter
i=1;
x=H; This is no longer data but a matrix of coefficients
%Prediction
beta10=MU+beta0*F';
p10=F*P00*F'+Q;
yhat=(x*(beta10)')';
eta=[data(i,:) z(i,:)]-yhat;
```

$$\begin{pmatrix} X_{1t} \\ X_{2t} \\ X_{3t} \\ . \\ . \\ . \\ . \\ X_{Mt} \\ FFR_t \end{pmatrix}$$

$\tilde{X}_{it}$

```
feta=(x*p10*x')+R;
%updating
K=(p10*x')*inv(feta);
beta11=(beta10'+K*eta')';
```

```
p11=p10-K*(x*p10);
beta_tt=[beta_tt;beta11];
ptt(i,:,:)=p11;
for i=2:T
    %Prediction
beta10=MU+beta11*F';
p10=F*p11*F'+Q;
yhat=(x*(beta10)')';
eta=[data(i,:) z(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
% Backward recursion to calculate the mean and variance of the distribution of the state
%vector
beta2 = zeros(T,ns);   %this will hold the draw of the state variable
jv=1:3; %index of state variables to extract 3 factors to extract
wa=randn(T,ns);
f=F(jv,:);
```

$$F^*$$

```
q=Q(jv,jv);
```

$$Q^*$$

```
mu=MU(jv);
```

$$\mu^*$$

```
i=T;  %period t
p00=squeeze(ptt(i,jv,jv));
beta2(i,jv)=beta_tt(i:i,jv)+(wa(i:i,jv)*cholx(p00));   %draw for beta in period t from N(beta_tt,ptt)
%periods t-1..to .1
for i=T-1:-1:1
pt=squeeze(ptt(i,:,:));
```

$$B_{t/t}+P_{t/t}F^{*\prime}(F^*P_{t/t}F^{*\prime}+Q^*)^{-1}[\tilde{B}^*_{t+1}-\mu^*-F^*B_{t\backslash t}]$$

```
bm=beta_tt(i:i,:)+(pt*f'*inv(f*pt*f'+q)*(beta2(i+1:i+1,jv)-mu-beta_tt(i,:)*f')')';
```

$$P_{t/t}-P_{t/t}F^{*\prime}(F^*P_{t/t}F^{*\prime}+Q^*)^{-1}F^*P_{t/t}$$

```
pm=pt-pt*f'*inv(f*pt*f'+q)*f*pt;
beta2(i:i,jv)=bm(jv)+(wa(i:i,jv)*cholx(pm(jv,jv)));
end
pmat=beta2(:,1:3);   %update the factors
if m>burn
    %compute impulse response
    A0=cholx(Sigma);
    yhat=zeros(36,N);
vhat=zeros(36,N);
vhat(3,1:N)=[0 0 0 1];
for i=3:36
 yhat(i,:)=[yhat(i-1,:) yhat(i-2,:) 1]*[beta1(1:N*L,:);zeros(1,N)]+vhat(i,:)*A0;
end
yhat1=yhat*H(:,1:KK+1)';  %impulse response for the panel
irfmat(mm,1:36,1:NN+1)=(yhat1);
mm=mm+1;
```

```
end

end
irf=prctile(irfmat,[50 16 84],1);
figure(1)
j=1
for i=1:size(irf,3)
subplot(4,10,j)
plotx1(squeeze(irf(:,:,i))');
title(strcat('\fontsize{8}', names(i)))
j=j+1
axis tight
end
```

*Published with MATLAB® 7.9*