Econometrics - Problem Set 3

Gualtiero Azzalini

Part (a)

We want to write the system in the following form:

$$A(\theta)X_t = B(\theta)\mathbb{E}_t X_{t+1} + C(\theta)X_{t-1} + D(\theta)\eta_t \tag{1}$$

Where X_t is a $nx1$ state vector, η_t is a $nx1$ vector of innovations. $A(\theta)$, $B(\theta)$, $C(\theta)$, $D(\theta)$ are nxn matrices depending on parameters. In our case $n = 5$, hence the following vectors and matrices enable us to write the system in the required form:

$$X_t = \begin{bmatrix} x_t \\ \pi_t \\ i_t \\ g_t \\ u_t \end{bmatrix} \quad \eta_t = \begin{bmatrix} 0 \\ 0 \\ \varepsilon_{it} \\ \varepsilon_{gt} \\ \varepsilon_{ut} \end{bmatrix}$$

$$A(\theta) = \begin{bmatrix} 1 & 0 & \sigma & -1 & 0 \\ -k & 1 & 0 & 0 & -1 \\ -\phi_x(1-\rho_i) & -\phi_\pi(1-\rho_i) & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B(\theta) = \begin{bmatrix} 1 & \sigma & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho_i & 0 & 0 \\ 0 & 0 & 0 & \rho_g & 0 \\ 0 & 0 & 0 & 0 & \rho_u \end{bmatrix} \quad D(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_i & 0 & 0 \\ 0 & 0 & 0 & \sigma_g & 0 \\ 0 & 0 & 0 & 0 & \sigma_u \end{bmatrix}$$

The code that performs the above is reported here:

```
%% Preliminaries

sigma   = 1;  kappa   = .2;  beta    = .99;  phi_pi  = 1.25;  phi_x   = .25;
rho_i   = .5;  sigma_i = 2;  rho_g   = .75;  rho_u   = .75;  sigma_g = 1;  sigma_u = 1;
```

```
sigma_m = .1; N       = 1000;

%% Write the model as a system of expectational equations
A   = [1                  0                  sigma     -1     0;
       -kappa             1                  0          0    -1;
       -phi_x*(1-rho_i)   -phi_pi*(1-rho_i)  1          0     0;
       0                  0                  0          1     0;
       0                  0                  0          0     1];

B   = [1    sigma   0   0   0;
       0    beta    0   0   0;
       0    0       0   0   0;
       0    0       0   0   0;
       0    0       0   0   0];

C   = [0    0   0       0       0;
       0    0   0       0       0;
       0    0   rho_i   0       0;
       0    0   0       rho_g   0;
       0    0   0       0       rho_u];

D   = [0    0   0         0         0;
       0    0   0         0         0;
       0    0   sigma_i   0         0;
       0    0   0         sigma_g   0;
       0    0   0         0         sigma_u];
```

Part (b)

The original system is:

$$AX_t = B\mathbb{E}_t X_{t+1} + CX_{t-1} + D\eta_t$$

Conjecture the solution is:

$$X_t = FX_{t-1} + G\eta_t$$

so that the one-step ahead forecast is:

$$\mathbb{E}_t X_{t+1} = FX_t$$

and the system becomes:

$$(A-BF)X_t = CX_{t-1} + D\eta_t \Rightarrow X_t = (A-BF)^{-1}CX_{t-1} + (A-BF)^{-1}D\eta_t \tag{2}$$

so that:

$$F = (A-BF)^{-1}C \qquad G = (A-BF)^{-1}D \tag{3}$$

We first solve for F:

$$F = (A-BF)^{-1}C \Rightarrow BF^2 - AF + C = 0$$

Stack (1) in first-order form:

$$\begin{bmatrix} 0_{nxn} & I_n \\ -C & A \end{bmatrix}\begin{bmatrix} X_{t-1} \\ X_t \end{bmatrix} = \begin{bmatrix} I_n & 0_{nxn} \\ 0_{nxn} & B \end{bmatrix}\mathbb{E}_t\begin{bmatrix} X_t \\ X_{t+1} \end{bmatrix} + \begin{bmatrix} 0_{nxn} & 0_{nxn} \\ 0_{nxn} & D \end{bmatrix}\begin{bmatrix} 0_{nx1} \\ \eta_t \end{bmatrix}$$

Define:

$$K = \begin{bmatrix} 0_{nxn} & I_n \\ -C & A \end{bmatrix} \qquad L = \begin{bmatrix} I_n & 0_{nxn} \\ 0_{nxn} & B \end{bmatrix}$$

Apply the generalized Schur decomposition to K and L to get:

$$\bar{Q}'KZ = T \tag{4}$$

$$\bar{Q}'LZ = S \tag{5}$$

where T, S are upper triangular and Q, Z are unitary. Partition them into nxn blocks:

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$
$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \quad S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$$

with the terms associated with the smallest eigenvalues in the upper left blocks. By following the same steps in the slides we get that the solution for F is:

$$F = Z_{21}Z_{11}^{-1} = Q_{11}(T_{11}S_{11}^{-1})Q_{11}^{-1}$$

Replacing this into the formulas above we can get G and X_t.
Therefore we can express the solution in the following state-space form:

$$X_t = F(\theta)X_{t-1} + G(\theta)\eta_t \tag{6}$$

$$Y_t = H(\theta)X_t + J(\theta)v_t \tag{7}$$

Where the elements in (6) were defined above and Y_t is a $mx1$ vector of observables, v_t is $mx1$ vector of measurement errors, $H(\theta)$ is mxn matrix and $J(\theta)$ a mxm matrix depending on parameters. In our case $m=3$ and we get the following:

$$Y_t = \begin{bmatrix} x_t^{obs} \\ \pi_t^{obs} \\ i_t^{obs} \end{bmatrix} v_t = \begin{bmatrix} v_{xt} \\ v_{\pi t} \\ v_{it} \end{bmatrix}$$

$$H(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad J(\theta) = \begin{bmatrix} \sigma_m & 0 & 0 \\ 0 & \sigma_m & 0 \\ 0 & 0 & \sigma_m \end{bmatrix}$$

The code that performs the above is reported here:

```
%% Solve the system and express the solution in state-space form
K = [zeros(5,5)      eye(5,5); -C              A        ];
L = [eye(5,5)        zeros(5,5); zeros(5,5)      B          ];

% QZ decomposition

[S,T,Q,Z] = qz(K,L);
[S,T,Q,Z] = ordqz(S,T,Q,Z,'udi');
Z11 = Z(1:5,1:5); Z21 = Z(6:10,1:5);

% Matrices of the state equation

F = real(Z21*inv(Z11));
G = inv(A-B*F)*D;

% Martices of measurement equation
H   = [1 0 0 0 0;         0 1 0 0 0;         0 0 1 0 0];
J   = [sigma_m 0 0;        0 sigma_m 0;        0 0 sigma_m];
```

Part (c)

First, I have to generate the series of the shocks in the measurement and state equation. The following part of the code computes this:

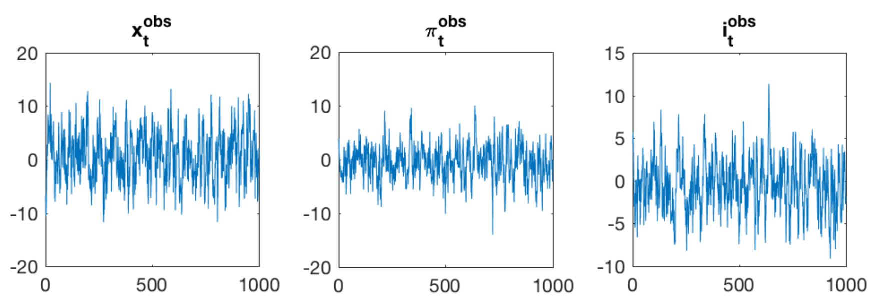
```
%% Use state-space form to simulate time series
% Generate the random components in the state and measurement equations
eta = [zeros(2,N);randn(3,N)];
v   = randn(3,N);
```

The length I have chosen is $N = 1000$. Note that the shocks in the state equation and in the measurement equation are independent. Then using these I can compute X_t and Y_t using (6) and (7).

The code that performs the above is reported here and Figure 1 plots the resulting series.

```
% Generate the series for x, pi, i
Xt_1 = zeros(5,1);
for i=1:N
        Xt(:,i) = F*Xt_1 + G*eta(:,i);
        Xt_1 = Xt(:,i);
        Yt(:,i) = H*Xt(:,i) + J*v(:,i);
end
```

Figure 1: Simulated Series



Part (d)

Using the prediction error decomposition we get:

$$p(Y^T|\theta) = p(Y_0|\theta)\prod_{t=1}^{T} p(Y^t|Y^{t-1},\theta)$$

$$\Rightarrow \log p(Y^T|\theta) = \log p(Y_0|\theta) + \sum_{t=1}^{T}\log p(Y^t|Y^{t-1},\theta)$$

Now, since the innovations are Gaussian and the system is linear the variables are conditionally normal, so the conditional mean an variance are sufficient statistics and hence we can evaluate the log-likelihood via the Kalman filter. The system and matrices to use are those described in part (b), i.e. equations (6) and (7). We only have to specify the covariance matrices. Let Q the nxn covariance matrix of the state and R the mxm matrix of the measurement and F the nxm cross-covariance between state and measurement. In our case, as the innovations ε and v are independent and iid standard normals and I assume that $\varepsilon$and v are independent as it is not differently specified we get:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad F = 0_{5x3}$$

To sum up, the formulas I use for the recursion are:

$$\begin{aligned}
X_{0|0} &= 0_{5x1} \\
P_{0|0} &= I_{5x5} \\
X_{t|t-1} &= FX_{t-1|t-1} \\
P_{t|t-1} &= FP_{t-1|t-1}F' + GG' \\
Y_{t|t-1} &= HX_{t|t-1} \\
V_{t|t-1} &= HP_{t|t-1}H' + JJ' \\
\log p_t &= \log p_{t-1} - \tfrac{1}{2}(\log|V_{t|t-1}| + (Y_t - Y_{t|t-1})' V_{t|t-1}^{-1}(Y_t - Y_{t|t-1})) \\
X_{t|t} &= X_{t|t-1} + P_{t|t-1}H' V_{t|t-1}^{-1}(Y_t - Y_{t|t-1}) \\
P_{t|t} &= P_{t|t-1} - P_{t|t-1}H' V_{t|t-1}^{-1} HP_{t|t-1}
\end{aligned}$$

The initial conditions follow, respectively, from the fact that the system is log-linearized around the steady state (hence in steady state the state vector is zero) and from the fact that the covariance matrix of the state is I_{5x5}.

I then wrote the following function in MATLAB to evaluate the log-likelihood via the Kalman filter in the Gaussian case:

```
%% Kalman Filter
% This program computes the kalman Filter for a linear Gaussian filter
% State:              St = A*St-1 + B*eta_t
% Measurement:        Xt = C*St   + D*v_t
% St is a Nx1 vector of states
% Xt is a Mx1 vector of observables
% eta_t is a Nx1 vector of residuals
% v_t is a Mx1 vector of residuals
% INPUT %
% A is NxN matrix
% B is NxN matrix
% C is MxN matrix
% D is MxM matrix
% OUTPUT %
% loglike is the log-likelihood function
% S(Nx1),P(NxN) are conditional mean and variance of state
% X(Mx1),V(MxM) are conditional mean and variance of state

function loglike = kalman(A,B,C,D,Xt,N);
% Initializing

S     = zeros(N,1);
P     = eye(N,N);
loglike = 0;

for i=1:size(Xt,2)

        % State Equation
        S1    = A*S;
        P1    = A*P*A' + B*B';

        % Measurement equation
        X     = C*S1;
        V     = C*P1*C' + D*D';

        % Likelihood
        loglike = loglike - 0.5*(log(det(V)) + (Xt(:,i)-X)'*(V^(-1))*(Xt(:,i)-X));

        % Updating
        S = S1 + (P1*C')*(V^(-1))*(Xt(:,i)-X);
        P = P1 - (P1*C')*(V^(-1))*(C*P1);
```

```
end
```

Hence putting the elements specific to our case in the above function I can obtain the log-likelihood:

```
%% Given the data evaluate the log-likelihood
loglike = kalman(F,G,H,J,Yt,size(Xt,1));
```

The log-likelihood in this case is $\log p = -3744.94$

Part (e)

Here, I just repeat the same steps from part (a) to (d) for all the values I want to try for each parameter keeping fixed the others and the originally generated series for X_t and Y_t. Figure 2 reports the results. The procedure seems to work well as the likelihood is maximized near the true parameter in all the cases. In practice, to do this, I built up a function in MATLAB that computes what I did above for all the values I have to try. As an example (for the other parameters it's exaclty the same) I report the code I use to try different values for σ. $pset3function$ is the function that does the job just described.

```
step=20;
e=.2;
sigmak = linspace(sigma-e,sigma+e,step);
for i=1:length(sigmak)
        loglike_i(i)=pset3function(sigmak(i),.2,.99,1.25,.25,.5,2,.75,.75,...
                                         1,1,.1,N,eta,v,Xt,Yt);
end
```

Figure 2: Log-likelihood as function of the different parameters