Table of Contents

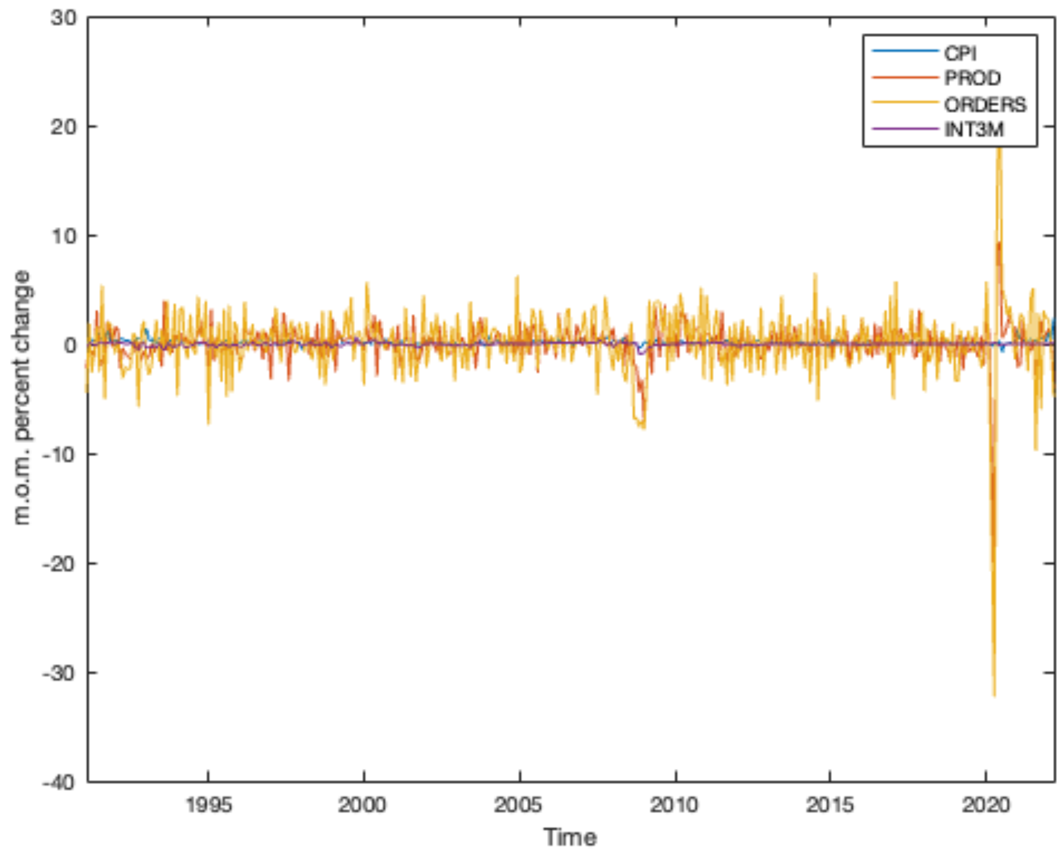
Multivariate Time Series Analysis and Forecasting

Name: Thi Lan Anh Nguyen

```
% This program estimates VAR(p) models using Matlab's built-in functions to
% investigate the inflation dynamics in Germany and conduct an out-of-sample
% forecasting exercise.
```

Part 1

```
clear
clc

% ===|Load the data set and transform the variables into stationary time
 series.

load DE_macroeconomy.mat
% Transform to stationary TS
CPI = diff(log(CPI))*100;
PROD = diff(log(PROD))*100;
ORDERS = diff(log(ORDERS))*100;
INT3M = diff(INT3M);

data = [CPI PROD ORDERS INT3M];
dates = dates(2:end);

% ===| Plot the transformed (stationary) data:
figure
plot(dates,data,'LineWidth',1)
xlabel('Time')
ylabel('m.o.m. percent change')
legend(VARnames{:})
```



Part 2

Estimate VAR(p) models with lags ranging from p = 1 to p = 14.

```
% Set VAR parameters:
T    = size(data, 1);          % Number of observations
K    = size(data, 2);          % Number of variables
pmax = 14;          % Maximum lag
lags = 1:pmax;          % Lags of the VAR models

% Initialization of VAR objects:
EstMdl(pmax) = varm(K, 0);
logL(pmax)   = nan();

% Fit VAR models and compute information criteria:
for p = 1:pmax
    Mdl = varm(K, p);                            % Specify VAR(p) structure
    Mdl.SeriesNames = VARnames;                  % Define variable names
    [EstMdl(p),~,logL(p)] = estimate(Mdl, data);    % Estimate VAR(p) models
 and store results
    NumParams = summarize(EstMdl(p)).NumEstimatedParameters;        % Recover
 number of parameters for each VAR(p)
```

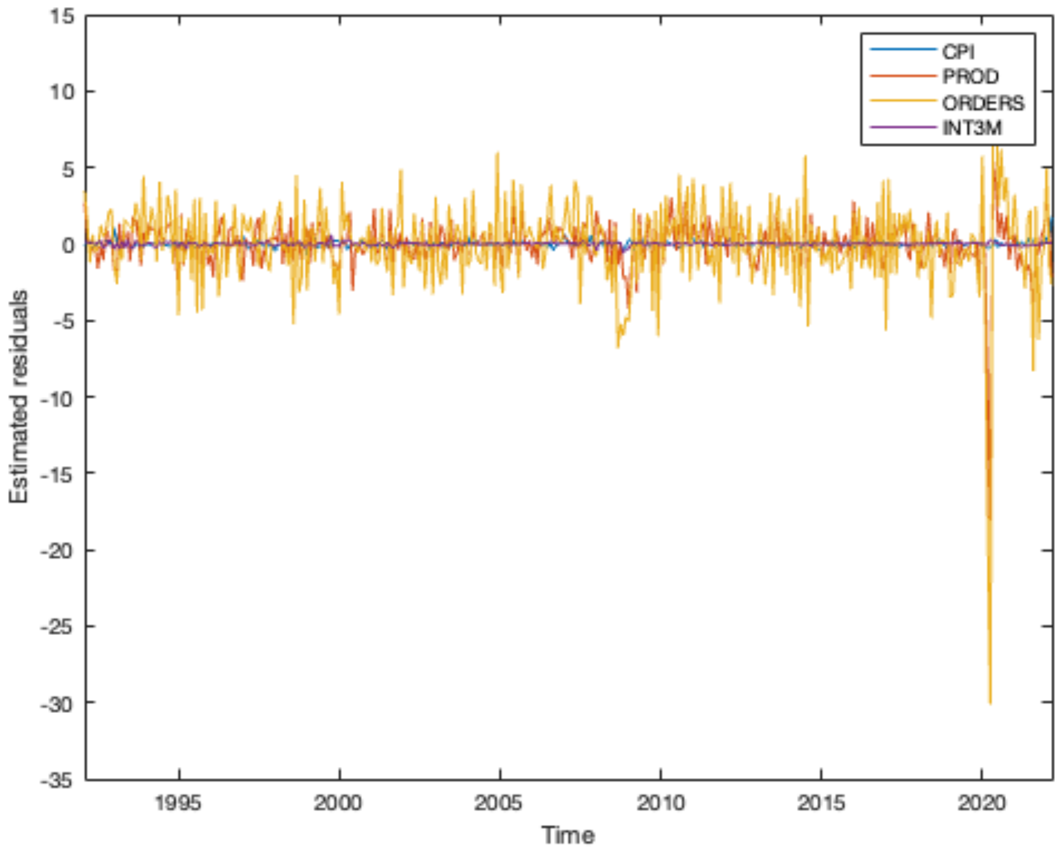
```
    [AIC(p), BIC(p)] = aicbic(logL(p), NumParams, T-p);             % Store
 information criteria
end

[~,p_hata] = min(AIC);        % Best model to AIC
[~,p_hatb] = min(BIC);        % Best model to BIC
display(p_hata)
display(p_hatb)

% According to the Akaike (AIC) model with 12 lags is the best, while for
% Bayesian (BIC) 1-lagged model has the best fit.
```

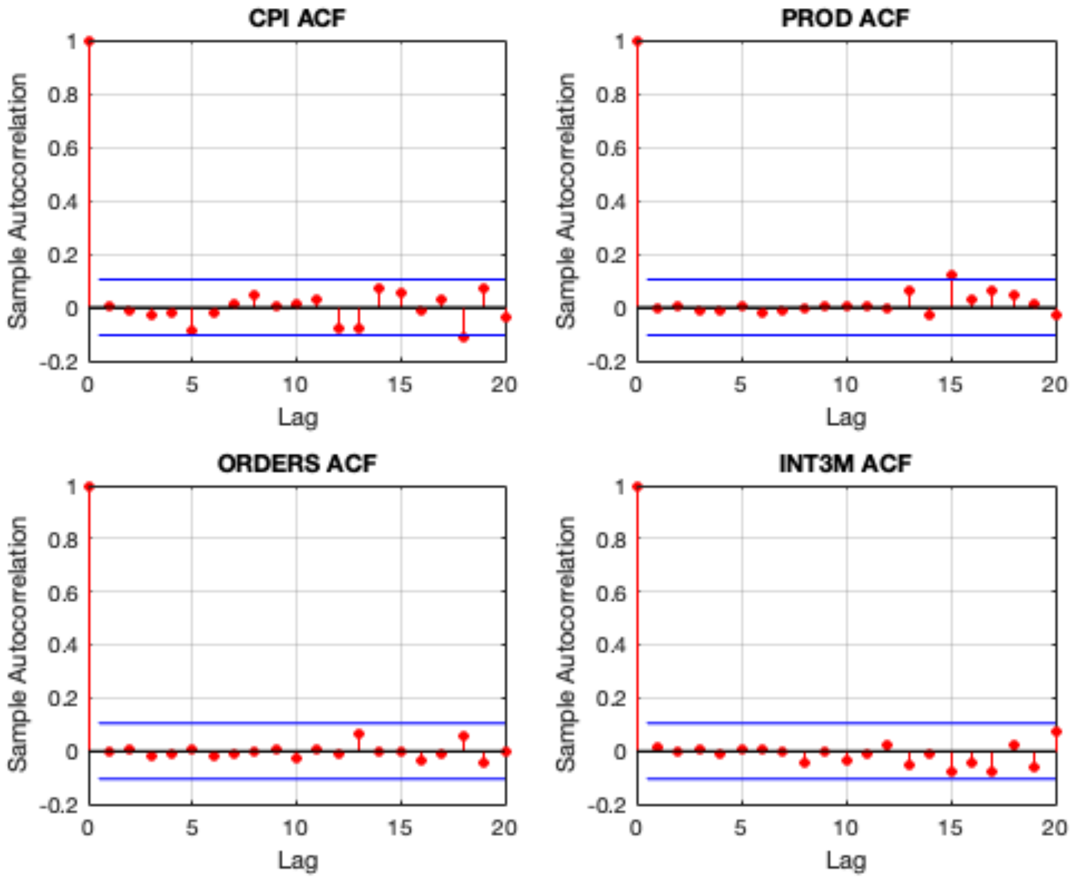
```
p_hata =

    12


p_hatb =

     1
```

Part 3

```
%Which model would you select for impulse response analysis?

% ===|Store your selected model results as BestMdl.
p = p_hata;                                         % Try AIC model*
[BestMdl,~,~,E] = estimate(varm(K, p), data);   % Re-estimate and store
 selected model
BestMdl.SeriesNames = VARnames;    % Define variable names

% ===|Plot the estimated residuals for each equation of the BestMdl VAR(p)
 model in one single figure.
figure
plot(dates(p+1:end),E,'LineWidth',1)
xlabel('Time')
ylabel('Estimated residuals')
legend(VARnames{:})

% ===|Discuss whether serial correlation patterns emerge and decide on model
% rejec-tion/acceptance of BestMdl.
figure
for i = 1:K
    subplot(2,2,i)
    autocorr(E(:, i));
    title([VARnames{i} ' ACF']);
end
% It seems like there is almost no autocorrelation in all 4 equations
% (except 15th lag in PROD)
% -> the model with 12 lags mostly capture the relationship between
% variables => accept model with 12 lags
```

```
% If change p = p_hatb = 1 -> the autocorrelation is still very much
% significant in all 4 variables => reject model with 1 lag.
```

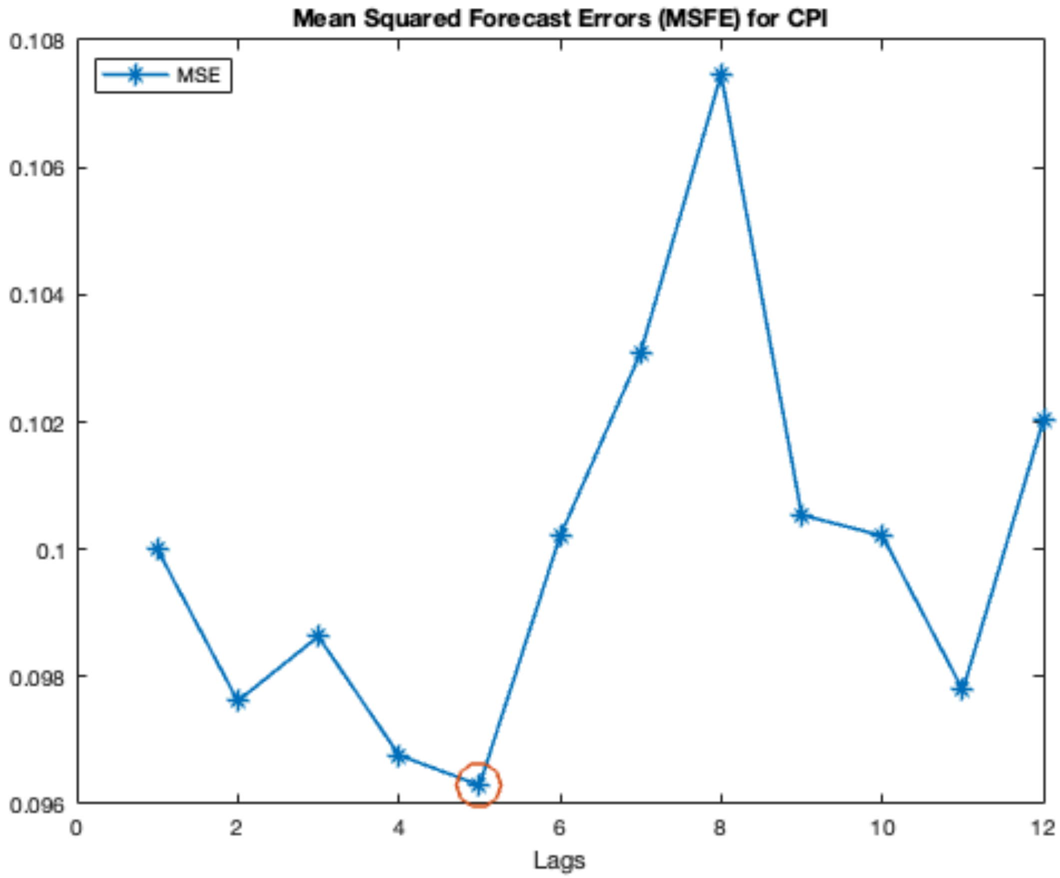


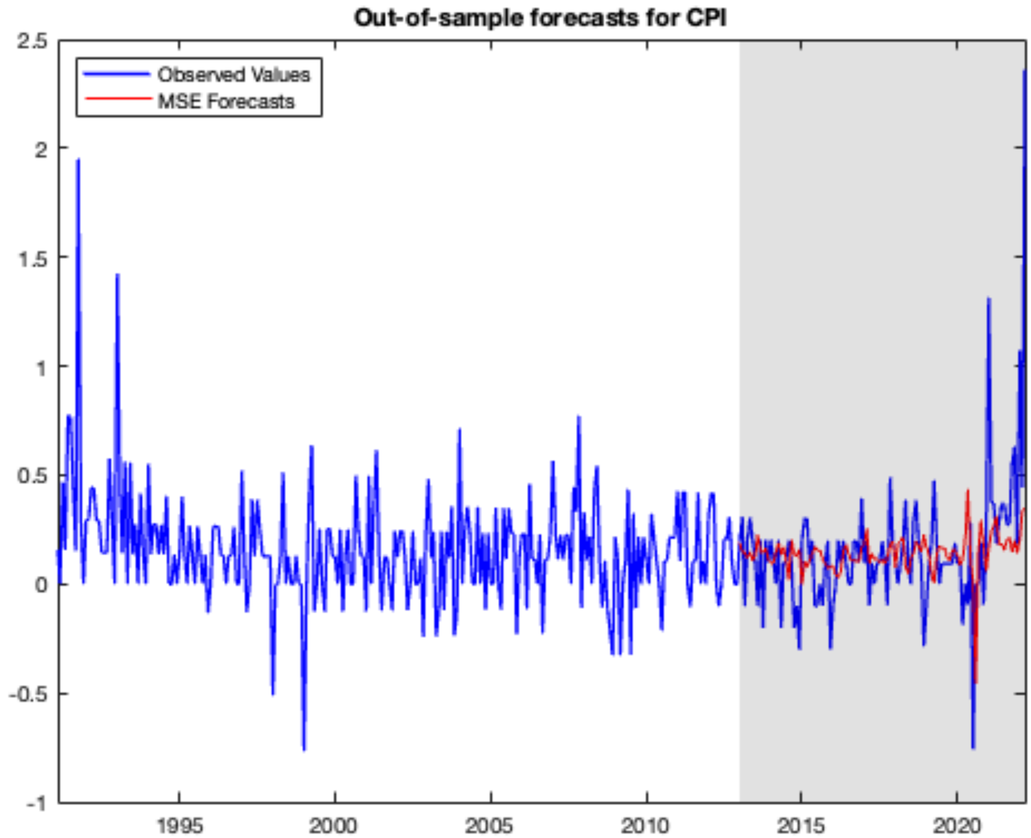


Part 4

```
%Split the data set into the in-sample period from 1991M2 to 2012M12 with
% T = 263 observations and the out-of-sample period from 2013M1 to 2022M3 with
% N = 111 observations. Generate one-step ahead out-of-sample forecasts for
% VAR(p) models with lags ranging from p = 1 to p = 12.
% Use an expanding window analysis


% Set parameters for the out-of-sample forecasting exercise
K = size(data, 2);       % Number of variables
h = 1;                   % One-step ahead forecast horizon
T = 263;                 % Number of observations for the 1st expanding window
N = 111;                 % Number of out-of-sample windows
pmax = 12;               % Maximum lag order

% Pre-allocation of forecast matrices:
y_hat   = cell(N, pmax);  % Cell structure to store conditional mean forecasts
y_MSE   = cell(N, pmax);  % Cell structure to store MSE forecast error
 matrices
e_MSE   = cell(N, pmax);  % Cell structure to store forecast errors under MSE
 loss
```

```
% Compute out-of-sample forecasts for each VAR(p) model and rolling window:
for p = 1:pmax
    for j = 1:N
        EstMdl       = estimate(varm(K, p), data(1:T+j-1, :));               %
 Estimate the VAR(p) with K var and lag p
       [y_hat{j,p}, y_MSE{j,p}] = forecast(EstMdl, h, data(T+j-p: T
+j-1, :)); % Conditional mean forecasts and MSE matrix
        e_MSE{j,p}    = (data(T+j, :) - y_hat{j, p}).^2;                     %
 Forecast squared errors (MSE loss)
    end
end

% ===| Mean of cummulative forecast errors under MSE
L_MSE = reshape(mean(cell2mat(e_MSE)), [K pmax]);                   % Sample
 mean of e_MSE for each variable k and VAR(p) model
[MSEmin, pMSE] = min(L_MSE(1, :));                                   % Find
 best prediction model for CPI -> 1st row

% ===| Plot mean forecast errors under MSE and MAE loss:
i  = 1;   % Define the variable k to investigate mean forecast errors
figure
h1 = plot(1:pmax,L_MSE(i,:),'Marker','*','LineWidth',1.5);
hold on
plot(pMSE,MSEmin,'Marker','o','MarkerSize',20,'LineWidth',1.5);
xlabel('Lags')
legend('MSE','Location','northwest')
title(['Mean Squared Forecast Errors (MSFE) for ' VARnames{i}]);

% ===| Plot out-of-sample optimal forecasts:
y_hat   = reshape(cell2mat(y_hat),[N,K,pmax]);   % Reshape conditional mean
 forecasts

figure
for i = 1:1
    subplot(1,1,i)
    h1 = plot(dates,data(:, i),'Color','blue','LineWidth',1.3);
    hold on
    h2 = plot(dates(T+1:T+N),y_hat(:,i,pMSE),'Color','red','LineWidth',1);
    title(['Out-of-sample forecasts for ' VARnames{i}]);
    h = gca;
    fill([dates(T+1) h.XLim([2 2]) dates(T+1)],h.YLim([1 1 2 2]),'k',...
    'FaceAlpha',0.1,'EdgeColor','none');
    legend([h1 h2],'Observed Values','MSE Forecasts','Location','northwest')
    hold off
end
```


Mean Squared Forecast Errors (MSFE) for CPI
MSE
0.108
0.106
0.104
0.102
0.1
0.098
0.096
0
2
4
6
8
10
12
Lags



Published with MATLAB® R2021b