# Output gap estimation through Kalman filter

We are now going to estimate a simple trend/cycle model of GDP augmented with a simple Philips curve equation linking GDP and inflation. The model is defined as follows:

*Observation equations*

*State equations*

where is log GDP, the output gap (cycle), potential GDP (the trend) and inflation. Errors are assumed to be normally distributed, i.i.d. and uncorrelated.

Note that this model implies that the trend (potential) follows a random walk with drift. Actually, the definition of the trend/cycle decomposition gives . We also have:

So that we have .

To implement the model in EViews, use the provided program to load the data and transform the input data. Help on Eviews command can be accessed by hitting F1.

## Step 1

Setup appropriate coefficient vectors for the model parameters , , and the error variances using the *coef* command.

## Step 2

Establish estimated starting values using an HP filter for the trend and cycle. Denote the trend of log GDP by *lgdpt* and the cycle by *lgdpc*.

## Step 3

Estimate initial values for the coefficients and error variance by simple OLS. OLS estimation in EViews can be done using the following code (for a regression on an intercept):

*equation eqname.ls y = c(1)*

## Step 4

Store initial values for coefficients and variances. Standard errors can be accessed using the *eqname.@se* command. Temporary values can be stored using the following code:

*!temp = x*

## Step 5

Declare a state space model named kalman using the *sspace* command.

## Step 6

Set initial values for the parameters and variances in the model using the following code:

*kalman.append @param [coefname1] [val1] [coefname2] [val 2] …*

## Step 7

Declare the two observation equations using the following command:

*kalman.append @signal [equation]*

## Step 8

Declare the state equation using the following command:

*kalman.append @state [equation]*

As EViews does not allow state equations to feature lags of the state variables beyond the first lag (see help), you will have to do what is called *state augmentation* and declare a second state equations for the lagged variable.

*kalman.append @state ygap1 = ygap(-1)*

## Step 9

Estimate the model by MLE.

## Step 10

Compute smoothed unobserved variables (trend and cycle) using the *makestates* command.

## Step 11

Plot and compare with an HP filter the estimated trend and cycle for log GDP