ECONOMETRIC METHODS II
TA session 4
Estimating State-Space models through Maximum Likelihood

Fernando Pérez Forero*

May 31st, 2012

1 Introduction

In this session we will cover the estimation of a simple State Space Model with classical methods. In order to accomplish this mission it will be necessary to write down the Likelihood Function of the model with the aid of the Kalman Filter (see Lecture Notes).

Most of the material covered in this session can be found in chapter 13 Hamilton (1994) (H), in chapter 3 of Kim and Nelson (1999) (KN) and lecture notes and slides from prof. Kristoffer Nimark (N)[1]. In order to provide a better understanding of each step, I will indicate the number of equation of each reference as follows: (Book, equation).

2 The model

The model described below can be found in section 3.3. of Kim and Nelson (1999) (Application 1). It is based on the work of Peter Clark: "The Cyclical Component of U. S. Economic Activity" *The Quarterly Journal of Economics*, Vol. 102, No. 4 (Nov., 1987), pp. 797-814.

*PhD student. Department of Economics, Universitat Pompeu Fabra, Ramon Trias Fargas 25–27, 08005 Barcelona, Spain (email: fernandojose.perez@upf.edu)

[1]http://www.kris-nimark.net/TS_UPF_2012.html

2.1 Real GDP and unobserved components

Let y_t be the log of the real GDP of the U.S. It is assumed that it can be decomposed in two unobservable components, one that is nonstationary (n_t) and one that is stationary (x_t). The complete model is as follows:

$$y_t = n_t + x_t \tag{2.1}$$

$$n_t = g_{t-1} + n_{t-1} + \upsilon_t \tag{2.2}$$

$$g_t = g_{t-1} + w_t \tag{2.3}$$

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + e_t \tag{2.4}$$

where υ_t, w_t, e_t are *i.i.d.* Gaussian processes with variances σ_υ^2, σ_w^2 and σ_e^2, respectively and $t = 1, \ldots, T$. Besides, the terms n_t and x_t represent a stochastic trend and the cyclical component, respectively. Finally, term g_t is an auxiliary variable that represents a permanent drift in the stochastic trend, therefore it follows a random walk. The purpose of this exercise is to determine the paths of terms n_t and x_t by observing y_t in each period t. The latter can be implemented via the Kalman Filter given the parameter values $\theta = (\phi_1, \phi_2, \sigma_\upsilon, \sigma_w, \sigma_e)'$. However, since θ is unknown it turns out that we must focus our attention in how to estimate it in first place. Here we use classical methods to tackle this issue.

2.2 State Space form and the Likelihood Function

2.2.1 State-space form

Following the cited references the model (2.1) − (2.2) − (2.3) − (2.4) has a state-space form:

$$y_t = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} n_t \\ x_t \\ x_{t-1} \\ g_t \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} n_t \\ x_t \\ x_{t-1} \\ g_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \phi_1 & \phi_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_{t-1} \\ x_{t-1} \\ x_{t-2} \\ g_{t-1} \end{bmatrix} + \begin{bmatrix} \upsilon_t \\ e_t \\ 0 \\ w_t \end{bmatrix} \tag{2.6}$$

If we go back to the lecture notes of the course, the state space system was

$$Z_t = DX_t + \mathbf{v}_t \tag{2.7}$$

$$X_t = AX_{t-1} + C\mathbf{u}_t \tag{2.8}$$

where $Z_t \equiv y_t$, $X_t \equiv [n_t, x_t, x_{t-1}, g_t]'$, $\mathbf{v}_t = 0$ (constant) and therefore $\Sigma_{vv} = 0$, $\mathbf{u}_t = [u_{jt}]$ is a (4×1) vector with $u_{jt} \sim N(0,1)$ for $j = 1, \ldots, 4$ and

$$\begin{aligned} D &\equiv \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}, \; A \equiv \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \phi_1 & \phi_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ C &\equiv \begin{bmatrix} \sigma_v & 0 & 0 & 0 \\ 0 & \sigma_e & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_w \end{bmatrix} \end{aligned} \tag{2.9}$$

Notice also that since the parameter vector θ is constant over time, this is a version of the time-invariant filter described in section 4 of lecture notes. We will cover this issue in the next subsection.

2.2.2 The Kalman Filter

In order to infer the value of vector X_t, we do the following: At the beginning of period t we have the estimated value of previous period $\left(X_{t-1|t-1}\right)$ with some covariance matrix $\left(P_{t-1|t-1}\right)$ and we have equation (2.8) as prior information, so that we can forecast a value conditional on information at period $t-1$:

$$X_{t|t-1} = AX_{t-1|t-1} \tag{2.10}$$

Define the variance as

$$P_{t|t-1} \equiv E\left[\left(X_t - X_{t|t-1}\right)\left(X_t - X_{t|t-1}\right)'\right]$$

$$P_{t|t-1} = E\left[\left(AX_{t-1} - AX_{t-1|t-1} - C\mathbf{u}_t\right)\left(AX_{t-1} - AX_{t-1|t-1} - C\mathbf{u}_t\right)'\right]$$

$$P_{t|t-1} = E\left[\left(A\left(X_{t-1} - X_{t-1|t-1}\right) - C\mathbf{u}_t\right)\left(A\left(X_{t-1} - X_{t-1|t-1}\right) - C\mathbf{u}_t\right)'\right]$$

Since $\mathbf{u}_t$ is *i.i.d.*, we have that:

$$P_{t|t-1} = AE\left[\left(X_{t-1} - X_{t-1|t-1}\right)\left(X_{t-1} - X_{t-1|t-1}\right)'\right]A' + CE\left[\mathbf{u}_t\mathbf{u}_t'\right]C'$$

which is equal to

$$P_{t|t-1} = AP_{t-1|t-1}A' + CC' \tag{2.11}$$

New information related with X_t arrives in period t in the form of Z_t according to equation (2.7). As a result, we update our estimate of X_t combining these two sources of information as follows:

$$X_{t|t} = X_{t|t-1} + K_t \left(Z_t - DX_{t|t-1} \right) \tag{2.12}$$

where the term $Z_t - X_{t|t-1} \equiv \widetilde{Z}_t$ is the innovation and K_t is the Kalman gain. The latter can be derived using the projection theorem, that is

$$\left\langle X_t - K_t \widetilde{Z}_t, \widetilde{Z}_t \right\rangle = 0$$

$$E\left[\left(X_t - K_t \widetilde{Z}_t \right) \widetilde{Z}_t' \right] = 0$$

$$E\left[X_t \widetilde{Z}_t' \right] - K_t E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] = 0$$

and as a result we have (see lecture notes):

$$K_t = E\left[X_t \widetilde{Z}_t' \right] \left(E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] \right)^{-1}$$

where

$$E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] = E\left[\left(Z_t - DX_{t|t-1} \right) \left(Z_t - DX_{t|t-1} \right)' \right]$$

$$E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] = E\left[\left(DX_t + \mathbf{v}_t - DX_{t|t-1} \right) \left(DX_t + \mathbf{v}_t - DX_{t|t-1} \right)' \right]$$

$$E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] = E\left[\left(D\left(X_t - X_{t|t-1} \right) + \mathbf{v}_t \right) \left(D\left(X_t - X_{t|t-1} \right) + \mathbf{v}_t \right)' \right]$$

$$E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] = DE\left[\left(X_t - X_{t|t-1} \right) \left(X_t - X_{t|t-1} \right)' \right] D' + \Sigma_{vv} \tag{2.13}$$

so that:

$$K_t = P_{t|t-1} D' \left(DP_{t|t-1} D' + \Sigma_{vv} \right)^{-1} \tag{2.14}$$

Recall (2.12):

$$X_{t|t} = X_{t|t-1} + K_t \left(Z_t - DX_{t|t-1} \right)$$

$$X_{t|t} - X_t = X_{t|t-1} - X_t + K_t \left(Z_t - DX_{t|t-1} \right)$$

$$X_t - X_{t|t-1} = X_t - X_{t|t} + K_t \left(Z_t - DX_{t|t-1} \right)$$

Taking variances

$$P_{t|t-1} = P_{t|t} + K_t E\left[\widetilde{Z}_t \widetilde{Z}_t' \right] K_t'$$

and using (2.13):

$$P_{t|t-1} = P_{t|t} + K_t \left(DP_{t|t-1} D' + \Sigma_{vv} \right) K_t'$$

$$P_{t|t} = P_{t|t-1} - K_t \left(DP_{t|t-1} D' + \Sigma_{vv} \right) K_t'$$

which combined with (2.14) results in

$$P_{t|t} = P_{t|t-1} - P_{t|t-1}D' \left(DP_{t|t-1}D' + \Sigma_{vv}\right)^{-1} DP'_{t|t-1} \tag{2.15}$$

Now, we plug (2.15) in (2.11) expressed in $t+1$:

$$P_{t+1|t} = A\left(P_{t|t-1} - P_{t|t-1}D' \left(DP_{t|t-1}D' + \Sigma_{vv}\right)^{-1} DP'_{t|t-1}\right) A' + CC' \tag{2.16}$$

The time-invariant filter says that K_t and $P_{t|t-1}$ are constant if parameters θ are constant and at the same time the initial covariance matrix of the filter $\left(P_{1|0}\right)$ is the solution of equation (2.16).

2.2.3 The Likelihood Function

According to Hamilton (1994), the forecasts $X_{t|t-1}$ and $Z_{t|t-1} \equiv DX_{t|t-1}$ that were derived using linear projections are optimal among the set of linear forecasts. Furthermore, if the innovations v_t, w_t, e_t and the initial state $X_{0|0}$ are normally distributed, the forecasts $X_{t|t-1}$ and $Z_{t|t-1}$ are optimal among forecasts of any functional form. As a result, the distribution of the forecast is

$$Z_{t|t-1} \sim N\left(DX_{t|t-1}, DP_{t|t-1}D' + \Sigma_{vv}\right)$$

or (H,13.4.1):

$$\begin{aligned} f\left(Z_{t|t-1}\right) &= (2\pi)^{-n/2} \det\left(DP_{t|t-1}D' + \Sigma_{vv}\right)^{-1/2} \times \\ &\quad \exp\left(\widetilde{Z}'_t \left(DP_{t|t-1}D' + \Sigma_{vv}\right)^{-1} \widetilde{Z}_t\right) \end{aligned} \tag{2.17}$$

where $\widetilde{Z}_t \equiv Z_t - DX_{t|t-1}$, $n = \dim\left(X_t\right)$ and $t = 1, \ldots, T$. The log-likelihood function is therefore

$$l\left(\theta\right) = \prod_{t=1}^{T} \ln\left[f\left(Z_{t|t-1}\right)\right] \tag{2.18}$$

which is equal to

$$\begin{aligned} l\left(\theta\right) &= -\frac{Tn}{2}\ln\left[2\pi\right] - \frac{T}{2}\ln\left[\det\left(DP_{t|t-1}D' + \Sigma_{vv}\right)\right] \\ &\quad + \sum_{t=1}^{T}\left(Z_t - DX_{t|t-1}\right)' \left(DP_{t|t-1}D' + \Sigma_{vv}\right)^{-1} \left(Z_t - DX_{t|t-1}\right) \end{aligned} \tag{2.19}$$

Ideally, the function $l\left(\theta\right)$ is sufficiently well-behaved so that it is possible to find a θ^* associated with a global maximum. This task is not necessarily easy to accomplish, therefore in this notes we look for stable routines that produce accurate numerical results.

We will cover this issue in the next section.

2.2.4 The Kalman Smoother

So far we have learnt how to estimate the unobservable components vector X_t conditional on observing actual data Z_t and previous data Z_{t-1}, i.e. $X_{t|t}$ and $X_{t|t-1}$. Here we show that it is possible to infer the value of X_t for each period t given the entire dataset $Z^T = \{Z_1, Z_2, \ldots, Z_T\}$, i.e. $X_{t|T}$. For this procedure we need to focus our attention on the transition equation (2.8). We start the smoothing with the last filtered observation $X_{T|T}$, i.e. we don't need to update it since it is already conditioned on T. Now consider the updating equation

$$X_{t|T} = X_{t|t} + J_t \left(X_{t+1} - X_{t+1|t} \right) \tag{2.20}$$

which means that the smoothed value $X_{t|T}$ is a function of the filtered (real time) value $X_{t|t}$ and the innovation on X in the next period $\widetilde{X}_{t+1} \equiv \left(X_{t+1} - X_{t+1|t} \right)$. We assume that these two terms are orthogonal, therefore we can apply the projection theorem again

$$\left\langle X_t - J_t \widetilde{X}_{t+1}, \widetilde{X}_{t+1} \right\rangle = 0$$

$$E\left[\left(X_t - J_t \widetilde{X}_{t+1} \right) \widetilde{X}'_{t+1} \right] = 0$$

$$E\left[X_t \widetilde{X}'_{t+1} \right] - J_t E\left[\widetilde{X}_{t+1} \widetilde{X}'_{t+1} \right] = 0$$

and as a result we have (see lecture notes):

$$J_t = E\left[X_t \widetilde{X}'_{t+1} \right] \left(E\left[\widetilde{X}_{t+1} \widetilde{X}'_{t+1} \right] \right)^{-1}$$

where

$$E\left[\widetilde{X}_{t+1} \widetilde{X}'_{t+1} \right] = E\left[\left(X_{t+1} - X_{t+1|t} \right) \left(X_{t+1} - X_{t+1|t} \right)' \right]$$

$$E\left[\widetilde{X}_{t+1} \widetilde{X}'_{t+1} \right] = P_{t+1|t} \tag{2.21}$$

On the other hand

$$E\left[X_t \widetilde{X}'_{t+1} \right] = E\left[X_t \left(X_{t+1} - X_{t+1|t} \right)' \right]$$

$$E\left[X_t \widetilde{X}'_{t+1} \right] = E\left[X_t \left(A X_t + C \mathbf{u}_{t+1} - A X_{t|t} \right)' \right]$$

$$E\left[X_t \widetilde{X}'_{t+1} \right] = E\left[X_t \left(A \left(X_t - X_{t|t} \right) + C \mathbf{u}_{t+1} \right)' \right]$$

$$E\left[X_t \widetilde{X}'_{t+1} \right] = E\left[X_t \left(X_t - X_{t|t} \right)' A' \right] + \underbrace{E\left[X_t \mathbf{u}'_{t+1} \right]}_{=0} C'$$

$$E\left[X_t \widetilde{X}'_{t+1} \right] = E\left[\left(X_t - X_{t|t} + X_{t|t} \right) \left(X_t - X_{t|t} \right)' A' \right]$$

$$E\left[X_t\widetilde{X}'_{t+1}\right] = \underbrace{E\left[\left(X_t - X_{t|t}\right)\left(X_t - X_{t|t}\right)'\right]}_{=P_{t|t}}A' + \underbrace{E\left[X_{t|t}\left(X_t - X_{t|t}\right)'\right]}_{=0}A'$$

$$E\left[X_t\widetilde{X}'_{t+1}\right] = P_{t|t}A' \tag{2.22}$$

As a result from (2.21) and (2.22) we get

$$J_t = P_{t|t}A'P^{-1}_{t+1|t} \tag{2.23}$$

That is, the Kalman Smoother can be derived using the output of the Kalman Filter.

3 Estimating the model in Matlab

3.1 Data description

In order to estimate vector θ we use quarterly data of real GDP of the U.S. for the period 1952:I-1995:III, i.e. the same data of Kim and Nelson (1999)[2]. Now, let's take a look to the logarithm of this data in Figure 3.1

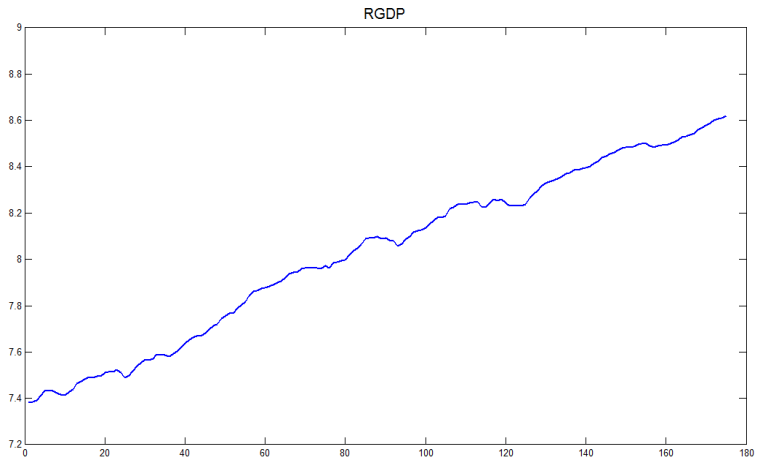

Figure 3.1: Natural logarithm of Real GDP (1952:I-1995:III

3.2 The Log-Likelihood Function

Next step is to write down a program for the log-likelihood function. The structure of the program is the following:

[2]This dataset can be downloaded from http://www.econ.washington.edu/user/cnelson/markov/prgmlist.htm together with GAUSS codes from the same reference. Since we will work on Matlab, I only use the dataset but not the GAUSS programs. For symmetry purposes, I initialize the filter using the same values as these authors, i.e. $X_{0|0} = 0$ and $P_{0|0} = 100 \times I$.

1. Assign values for each entry of vector $\theta = (\phi_1, \phi_2, \sigma_v, \sigma_w, \sigma_e)'$.
2. Given θ, set matrices A, C, D of the state-space form and Σ_{vv} as in (2.5) − (2.6) and (2.9).
3. Set the initial values $X_{0|0} = \mathbf{0}_{n\times 1}$ and $P_{1|0}$ as the solution of equation (2.11).
4. For each $t = 1, \ldots, T$ compute $X_{t|t-1}$ and $P_{t|t-1}$ using (2.10) and (2.11) and compute also $X_{t|t}$ and $P_{t|t}$ using (2.12) and (2.15).
5. At each step t, given $X_{t|t-1}$, $P_{t|t-1}$ and state-space matrices from step 2, evaluate equation (2.17).
6. Compute the final result using (2.18).

Besides these 6 steps, some details must be taken into account when constructing a likelihood function:

1. The function must have only one argument, i.e. θ. If your function has additional inputs, use `varargin` instead, otherwise the optimization routine will not work. Another possibility is to use `global` variables.
2. Most of optimization routines are minimizers, therefore you should make sure that your function is actually -1 times the theoretical one, in this case given by (2.19).
3. It is likely that for some regions of θ the value of $l(\theta)$ does not exist, at least in the real space. You should control for that and assign an arbitrary very low value for these cases. The latter will help the optimization routine to not explore these regions more than one time.
4. Computation of some matrices such as the Kalman Gain (K) in (2.14) often demands inverting some other matrices. This might sound trivial but sometimes this is could be a nightmare if the inverted matrix is near-singular. The command `inv` is the most commonly used but it could be very slow and very often unstable. Alternatively you can use left or right division (`mldivide(A,B)` $\Longleftrightarrow$ `A\B` or `mrdivide(B,A)` $\Longleftrightarrow$ `B/A`). However, sometimes this is not enough and for some parameter values the inverted matrix of the Kalman Gain formula will deliver unreasonable values. Another possibility is to use the command `pinv`, which is the Moore-Penrose pseudoinverse of a matrix.

The matlab code for this function can be found in Appendix A.1.

3.3 Numerical Maximization

Once we have the m-file for the function (`logl_rgdpxn.m`), the next step is to create a loop for maximization. Here we show two different optimization routines with the aim to show the sensitivity of results.

3.3.1 csminwel.m

The first optimization routine (`csminwel.m`) belongs to professor Christopher Sims's optimization package[3]. These routines are popular in the field because of their stability across many problems. The description of this program from its website is: *"csminwel: minimization. Uses a quasi-Newton method with BFGS update of the estimated inverse hessian. It is robust against certain pathologies common on likelihood functions. It attempts to be robust against "cliffs", i.e. hyperplane discontinuities, though it is not really clear whether what it does in such cases succeeds reliably."*.

We will use this routine starting from several values in order to find a global maximum. The structure of the loop is as follows:

1. Draw a large set of initial values $\Theta_0 \equiv \left\{\theta_0^1, \theta_0^2, \ldots, \theta_0^j, \ldots, \theta_0^N\right\}$ from an arbitrary distribution.

2. For each $j = 1, \ldots, N$ do the following: Given θ_0^j, execute the routine `csminwel.m` using `logl_rgdpxn.m` as a handling function until it finds an optimum θ_j^*. Store θ_j^* and $l\left(\theta_j^*\right)$ in different vectors.

3. Let $L^* = \left\{l\left(\theta_1^*\right), l\left(\theta_2^*\right), \ldots, l\left(\theta_j^*\right), \ldots, l\left(\theta_N^*\right)\right\}$. Find $\theta^{ML} = \arg\max_{\theta_j^*} L^*$.

4. Compute the inverse of the Information Matrix associated with $l\left(\theta^{ML}\right)$ and take the square root of the main diagonal as the standard deviations.

A crucial point in this algorithm is to set the initial size of the step. This can be modified in the file `csminit.m`. The matlab code for this loop can be found in Appendix A.2.

3.3.2 Simulated Annealing

The algorithm uses elements of grid search with random movements. See details Goffe et al. (1994) and in lecture notes. here we describe the following loop:

1. Set the upper and lower bounds for the parameter space, i.e. θ_{Min} and θ_{Max} and select an initial point θ_0.

[3]The set of programs related with this optimization routine can be donwloaded from http://sims.princeton.edu/yftp/optimize/mfiles/

2. Set initial temperature, reduction parameters, etc.

3. Start the loop according to lecture notes (`simannb2.m`) using `logl_rgdpxn.m` as a handling function until it finds an optimum θ^{ML}[4].

The matlab code for this loop can be found in Appendix A.3.

3.4 Kalman Filtering and Smoothing

Given a set of optimal parameter values θ^{ML}, it is now worth to explore the paths of unobserved components. We need to vectors: i) real time estimates $X_{t|t}$ and ii) smoothed estimates $X_{t|T}$. In otder to get these results we construct the following loop:

1. **State-Space form:** Given θ^{ML}, set matrices A, C, D and Σ_{vv} as in (2.5) − (2.6) and (2.9).

2. **Initialization:** Set the initial values $X_{0|0} = \mathbf{0}_{n\times 1}$ and $P_{1|0}$ as the solution of equation (2.11).

3. **Filtering:** For each $t = 1, \ldots, T$ compute $X_{t|t-1}$ using (2.10) and $X_{t|t}$ using (2.12). For this step, store also $P_{t|t-1}$ and $P_{t|t}$.

4. **Smoothing:** Given the sequence $\left\{X_{t|t-1}, X_{t|t}, P_{t|t-1}, P_{t|t}\right\}_{t=1}^{T}$, for each $t = T - 1, \ldots, 1$ compute $X_{t|T}$ using (2.20).

The matlab code for this loop can be found in Appendix A.4

4 Results

The maximum likelihood estimates of the model are reported in Table 4.1. Results are slightly different across methods, therefore we pick the one that has the maximum likelihood (Simulated Annealing).

[4]The function `simannb2.m` is a matlab translation of Goffe et al. (1994)'s original Fortran code. It has been made by Knoek van Soest <a_j_van_soest@fbw.vu.nl> and there is also a bug fix in 2000 by Anne Su <sua@bme.ri.ccf.org>. Furthermore, I've modified the code for using `varargin` instead of `global`.

Point estimates		
θ	Simulated Annealing	csminwel
ϕ_1	1.2825	1.2861
ϕ_2	−0.2925	−0.2961
σ_v	0.0001	0.0005
σ_w	0.0087	0.0087
σ_e	0.0001	0.0001
$l\left(\theta^{ML}\right)$	557.2278	557.2139

Table 4.1: Maximum Likelihood estimates for 1952-1995

Furthermore, given the optimal parameter values, we compute the trajectories of unobserved components n_t and x_t using the Kalman Filter and given the parameter values θ^{ML}. We first plot the trend component against the true data in Figure 4.3 and then the stationary cyclical component in Figure 4.2

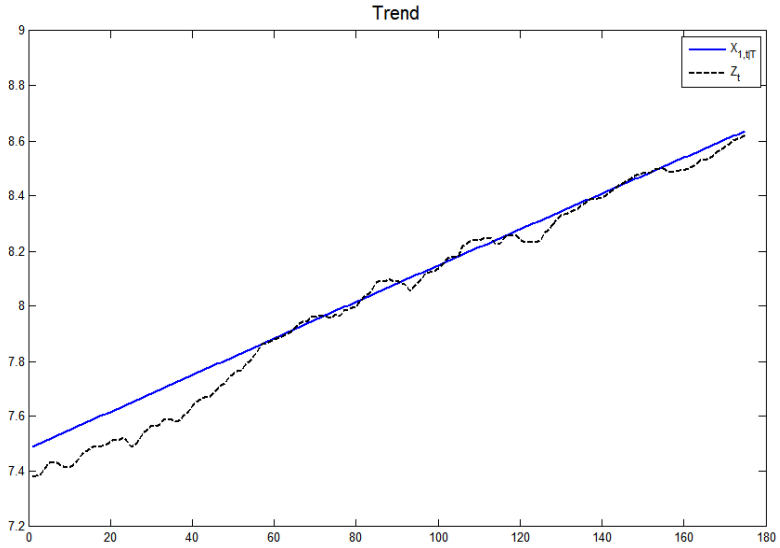

Figure 4.1: Smoothed trend component and the data

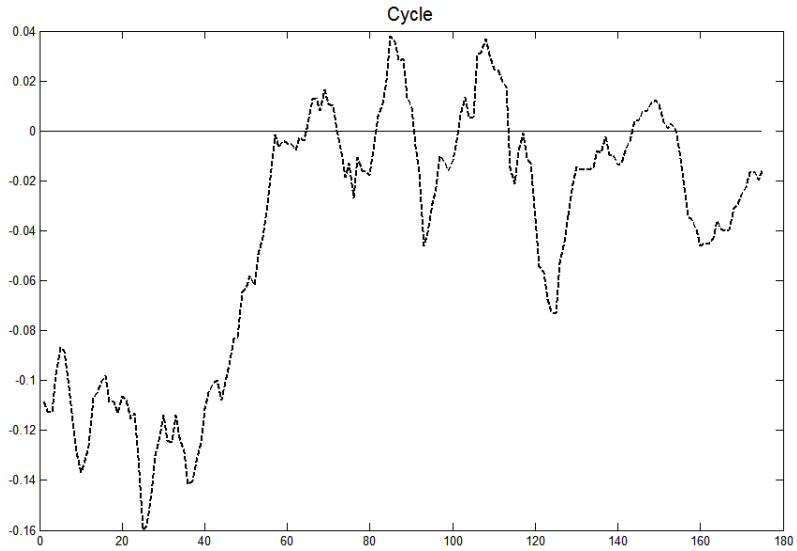

Figure 4.2: Smoothed cyclical component

However, it is worth to explore the paths of filtered (predicted and updated components) against the smoothed values. Here we can see that the difference is significant, but course this difference is specific to this experiment.

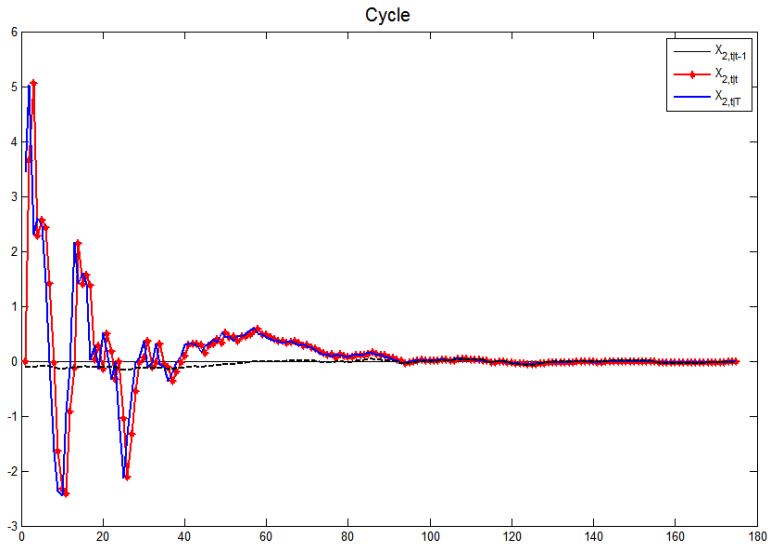

Figure 4.3: All cyclical components: predicted, filtered and smoothed

A Matlab codes

A.1 Log-Likelihood Function

```
function L=logl_rgdpxn(theta,varargin)
    phi1=theta(1);
    phi2=theta(2);
    sigma_v=exp(theta(3));
    sigma_e=exp(theta(4));
    sigma_w=exp(theta(5));

    % Stationarity restrictions

    temp=[(phi1+phi2<0.99),(-phi1+phi2<0.99),(abs(phi2)<0.99)...
   ,sigma_v>=1e-4,sigma_e>=1e-4,sigma_w>=1e-4];

    if sum(temp)==length(temp)
    Z=varargin{1};
    T=size(Z,1);
    % 1.  State-Space form

    A=[1 0 0 1;
    0 phi1 phi2 0;
    0 1 0 0;
    0 0 0 1];

    C=zeros(4,4);
    C(1,1)=sigma_v;
    C(2,2)=sigma_e;
    C(4,4)=sigma_w;

    D=[ones(1,2),zeros(1,2)];

    Sigma_vv=0;

    % 2.  Initialization

    n=size(A,1);
```

```
X=zeros(n,T);
P_tt=100*eye(n);

% Likelihood evaluation

L=0;

for t=1:T
% Prediction
if t==1
X_t1=zeros(n,1);
else
X_t1=A*X(:,t-1);
end
P_tt1=A*P_tt*A'+C*C';

% Updating
Omega=D*P_tt1*D'+Sigma_vv;
Omegainv=eye(size(Z,2))/Omega;
Kt=P_tt1*D'/Omega;
Ztilde=Z(t,1)-D*X_t1;
X(:,t)=X_t1+Kt*Ztilde;
P_tt=P_tt1-Kt*Omega*Kt';

% Log-Likelihood
L=L-0.5*(log(2*pi)+log(det(Omega)))-0.5*Ztilde'*Omegainv*Ztilde;
end

if isreal(L)==1
L=-L; % for minimization
else
L=8e30;
end

else
L=8e30; % nonstationary
end
```

```
end
```

A.2 Numerical Maximization via csminwel.m

```
%% 2.  Maximum Likelihood Estimation (csminwel.m)
param=5;
warning off all
Nbar=100;
LB_Theta=[-2,-1,log(1e-4),log(1e-4),log(1e-4)];
UB_Theta=[2, 1, log(std_y), log(std_y), log(std_y)];
theta_0T=zeros(Nbar,param);
theta_finT=zeros(Nbar,param);
LoglT=zeros(Nbar,1);
tic
for k=1:Nbar
% Settings for csminwel usage
theta0=unifrnd(LB_Theta,UB_Theta);
 tol_crit=1e-5;
 max_iter=1e100;

 H0=eye(param);
 [LogL,theta_fin,grad_theta,h_grad,itct,fcount,retcodeh] = csminwel(...
@logl_rgdpxn,theta0,H0,[],tol_crit,max_iter,Z);

 theta_0T(k,:)=theta0;
 theta_finT(k,:)=theta_fin;
 LoglT(k,1)=-LogL;
end
toc
[Logl_fin,ind_max]=max(real((LoglT)));
theta_fin=theta_finT(ind_max,:);
theta0=theta_0T(ind_max,:);
savefile='rgdpxn_csminwel.mat';
save(savefile,'theta_fin','Logl_fin','h_grad')
```

A.3 Numerical Maximization via Simulated Annealing

```
%% 3.  Simulated annealing
   param=5;
   warning off all
   LB_Theta=[-2,-1,log(1e-4),log(1e-4),log(1e-4)];
   UB_Theta=[2, 1, log(std_y), log(std_y), log(std_y)];
   theta_draw=LB_Theta+0.5*(UB_Theta-LB_Theta);
   sa_t= 5; %starting temperature
   sa_rt=.85;
   sa_nt=5;
   sa_ns=20;
   warning off all;
   [theta_fin]=simannb2('logl_rgdpxn', theta_draw, LB_Theta', UB_Theta', sa_t,
sa_rt, sa_nt, sa_ns, 1,Z);
   % theta, starting values
   % LB, lower bound on optimization parameters
   % UB, upper bound on optimization parameters
   % sa_t, initial temperature, try starting with 5
   % sa_rt, temperature reduction factor, conservative choice is .85
   % sa_nt, number of times through ns loop before temperature reduction (recommended
value:  5)
   % sa_ns, number of times through function before stepsize adjustment
   % (recommended value:  20)
   Logl_fin=-logl_rgdpxn(theta_fin,Z);
   savefile='rgdpxn_SA.mat';
   save(savefile, 'theta_fin')
```

A.4 The Kalman Filtering and Smoothing

```
%% 4.  Kalman Filtering and Smoothing
   load rgdpxn_SA.mat
   %load rgdpxn_csminwel.mat
   phi1=theta_fin(1);
   phi2=theta_fin(2);
   sigma_v=exp(theta_fin(3));
   sigma_e=exp(theta_fin(4));
```

```
sigma_w=exp(theta_fin(5));
T=size(Z,1);
% 4.1.  State-Space form

A=[1 0 0 1;
 0 phi1 phi2 0;
 0 1 0 0;
 0 0 0 1];

C=zeros(4,4);
C(1,1)=sigma_v;
C(2,2)=sigma_e;
C(4,4)=sigma_w;

D=[ones(1,2),zeros(1,2)];

Sigma_vv=0;

% 4.2.  Initialization

n=size(A,1);
Xtt1=zeros(n,T);
Xtt=zeros(n,T);
XtT=zeros(n,T);
P_tt1=zeros(n,n,T);
P_tt=zeros(n,n,T);
P_tT=zeros(n,n,T);
X_00=zeros(n,1);
P_00=100*eye(n);
% 4.3.  Kalman Filter

for t=1:T
 % Prediction
 if t==1
 Xtt1(:,t)=A*X_00;
 P_tt1(:,:,t)=A*P_00*A'+C*C';
 else
 Xtt1(:,t)=A*Xtt(:,t-1);
```

```
 P_tt1(:,:,t)=A*P_tt(:,:,t-1)*A'+C*C';
 end

 % Updating
 Omega=D*P_tt1(:,:,t)*D'+Sigma_vv;
 Kt=P_tt1(:,:,t)*D'/Omega;
 Ztilde=Z(t,1)-D*Xtt1(:,t);
 Xtt(:,t)=Xtt1(:,t)+Kt*Ztilde;
 P_tt(:,:,t)=P_tt1(:,:,t)-Kt*Omega*Kt';
end
% 4.4.  Kalman Smoother
XtT(:,T)=Xtt(:,T);
P_tT(:,:,T)=P_tt(:,:,T);
for t=T-1:-1:1
 Jt=P_tt(:,:,t)*A'/P_tt1(:,:,t+1);
 XtT(:,t)=Xtt(:,t)+Jt*(XtT(:,t+1)-Xtt1(:,t+1));
 P_tT(:,:,t)=P_tt(:,:,t)+Jt*(P_tT(:,:,t+1)-P_tt1(:,:,t+1))*Jt';
end
```

References

Goffe, W. L., G. D. Ferrier, and J. Rogers (1994). Global optimization of statistical functions with simulated annealing. *Journal of Econometrics 60*, 65–99.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.

Kim, C.-J. and C. R. Nelson (1999). *State-Space Models with Regime-Switching: Classical and Gibbs-Sampling Approaches with Applications*. MIT Press.