Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

STATE SPACE MODELS FOR THE ANALYSIS AND FORECASTING OF CLIMATIC TIME SERIES

Tesi di Laurea in Modelli Stocastici

Relatore:
Chiar.mo Prof.
GIACOMO BORMETTI

Co-relatore:
Chiar.mo Prof.
ANDREA PASCUCCI

Presentata da:
FLAVIA PASQUALI

VI Sessione
Anno Accademico 2019-2020

Contents

Introduzione

L'analisi di serie storiche climatiche è un argomento di ricerca che ha acquisito notevole rilevanza negli ultimi anni, soprattutto a causa dei forti cambiamenti climatici in atto. In questa tesi studieremo serie storiche climatiche utilizzando modelli di tipo *state space*. Questo tipo di modelli è caratterizzato dalla scomposizione della serie storica osservata in componenti latenti, raccolte nel vettore degli stati: ogni componente descrive un aspetto importante della serie osservata, come il *trend*, una componente stagionale e una ciclica, la perturbazione dovuta ad un errore di osservazione autoregressivo.

Per comprendere i cambiamenti climatici è necessario sviluppare un modello di questo tipo per le serie storiche climatiche, in quanto esso permette una profonda comprensione del fenomeno grazie ad una precisa descrizione dello stesso e successivamente una sua previsione. A questo proposito, il nostro obiettivo non è una previsione puntuale della serie storica ma la costruzione di scenari probabilistici, che riportino le diverse possibilità e le relative probabilità; questo approccio ha potenziali applicazioni in ambito assicurativo, proponendosi come strumento per la valutazione dei rischi connessi alla variabilità del clima.

In questa tesi miriamo a descrivere il comportamento congiunto di serie storiche climatiche con elevata profondità storica ed elevata dimensione trasversale. Questo comporta delle complicazioni modellistiche e computazionali che proponiamo di risolvere con una strategia a più passi. Innanzitutto utilizziamo un modello strutturale univariato (di facile stima), quindi analizziamo il comportamento congiunto per mezzo dei fattori dinamici (mutuati dalla macro-economia) e gestiamo l'aumento di dimensionalità con la tecnica del *collapsing*; infine simuliamo scenari con il *simulation smoothing* e calcoliamo la distribuzione di previsione.

Nel Capitolo 1 viene introdotto il modello *state space* Gaussiano lineare. Iniziando con la spiegazione delle componenti latenti e della loro forma matriciale, la trattazione continua con la derivazione delle ricorsioni del filtro di Kalman, che permettono di ricostruire gli stati latenti condizionatamente alle osservazioni mediante la massimizzazione della funzione di log-verosimiglianza ottenuta dalla scomposizione degli errori di predizione. Successivamente vengono presentati gli stati *smooth* impiegati poi nel *simulation smoothing*, che consente di simulare serie

storiche condizionali alla serie osservata. Una Sezione riguarda i test di specificazione del modello, essenziali per valutare l'adeguatezza della stessa. Inoltre, viene introdotta la tecnica di inizializzazione esatta con lo scopo di inizializzare, appunto, la stima del modello quando non sono disponibili informazioni sui valori iniziali del filtro.

Il Capitolo 2 riporta le esplorazioni fatte sul modello *state space*: prima con simulazioni Monte-Carlo di serie storiche, poi con serie di dati reali. Le serie storiche simulate vengono generate con diversi modelli, dal semplice *local level* a quello completo di tutte le componenti menzionate. Anche i dati reali vengono analizzati con modelli differenti: il primo è quello proposto da Mills [12], per il quale abbiamo scaricato i dati dell'*Hadley Centre for Climate Prediction and Research* [2]; questo lavoro ci permette di confrontare direttamente la specificazione del modello e i risultati ottenuti. L'altro modello per dati reali è utilizzato per analizzare la serie storica delle temperature massime registrate nella città di Bologna, scaricate dal sito dell'*European Climate Assessment and Dataset* [9]; l'obiettivo è quello di replicare il primo passo della procedura di Li e colleghi [10] e di prendere confidenza con le tecniche introdotte nel primo Capitolo.

Il Capitolo 3 tratta i modelli a fattori dinamici, e riporta perlopiù definizioni e metodi per la stima dei parametri. Ci concentriamo sul metodo a due passi presentato da Doz, Giannone e Reichlin [5], il quale richiede che il modello a fattori sia messo nella forma *state space*. Il Capitolo 4 riporta alcune interessanti esplorazioni con simulazioni di tipo Monte-Carlo.

Il punto centrale di questa tesi è il Capitolo 5. Dopo una breve introduzione sul metodo di Li e colleghi [10], questo viene modificato e applicato a serie storiche di dati reali procedendo con i seguenti passi:

1. download delle serie storiche delle temperature massime registrate in alcune stazioni dell'Emilia-Romagna;

2. analisi di ciascuna di esse separatamente con il modello *state space* e salvataggio in memoria di tutti gli errori di predizione;

3. utilizzo del modello a fattori dinamici sulle serie degli errori di predizione al fine di estrarre informazione residua;

4. simulazione di serie di errori di predizione grazie al *simulation smoothing* del modello a fattori dinamici;

5. calcolo, per ogni stazione, di serie storiche di osservazioni simulate con i parametri stimati al passo 2;

6. ri-stima dei parametri del modello *state space* per le serie simulate e calcolo delle previsioni, per ogni stazione separatamente;

7. media delle previsioni per ottenere la previsione finale di ogni stazione.

Fondamentalmente, iniziamo con il modello *state space* separatamente per ogni serie storica osservata, poi analizziamo tutte insieme le serie degli errori di predizione con il modello a fattori dinamici, dopodichè simuliamo gli errori di predizione e riprendiamo i parametri di ogni singola stazione per calcolare le simulazioni delle osservazioni; quindi procediamo per ogni stazione separatamente fino alle previsioni. Di fatto, in questa tesi, per ragioni di tempo di calcolo, gli ultimi passaggi vengono applicati solo ad una stazione, nello specifico quella di Bologna.

I risultati preliminari ottenuti dalle esplorazioni fatte in questa tesi sono molto promettenti. Abbiamo considerato serie storiche climatiche con limitata profondità storica e ridotta dimensione trasversale, ma solo per ragioni computazionali; d'altro canto, ciò ci ha permesso di verificare e testare rapidamente la specificazione e l'implementazione di ogni passo del metodo. Precedenti test effettuati su base settimanale con serie storiche significativamente più lunghe (fino a 150 anni) e di dimensione trasversale notevolmente maggiore (fino a 10.000) confermano la validità dell'intero approccio.

Introduction

Climatic time series analysis is a research topic which has gained great relevance in recent years, above all because of the remarkable climatic changes taking place. In this thesis we will explore climatic time series using state space models. This kind of models features the decomposition of the observed series in latent components, stored together in the state vector: each component portrays a relevant characteristic of the observed series, like the trend, a seasonal and a cyclical component, the contamination by an autoregressive observation error.

Developing such a model for climatic time series is necessary in order to understand climatic changes: it allows a deep comprehension of the phenomenon thanks to an accurate description and afterwards to a forecast of it. About this, our purpose is not the pointwise prediction of a time series but the construction of probabilistic scenarios which report different possibilities and relative probabilities; this approach has potential applications in insurance area, serving as a tool for evaluating risks concerning climatic variability.

In this thesis we aim to describe the joint behaviour of climatic time series featured by high temporal depth and high cross-sectional dimension. As a consequence, modeling and computational complications emerge, which we propose to deal with a multi-step strategy. First of all we fit a structural univariate model (which is easly estimated) to time series, then we analyse the joint residual information by means of dynamic factors (as usually done in macro-economic studies) and the increase in dimensionality is managed with the collapsing technique; finally we simulate scenarios with the simulation smoothing and compute the forecasting distribution.

Chapter 1 introduces the linear Gaussian state space model. Beginning with explanation of latent components and their matricial form, the discussion continues with the calculus of the Kalman filter recursions, that allow to reconstruct the latent states conditionally to the observations via the maximisation of the log-likelihood function in the prediction error decomposition form. Then smoothed states are presented and employed in the simulation smoothing, which permits to simulate time series conditionally to the observed one. A Section concerns the miss-specification tests of the model: they are essential to evaluate the

adequateness of the model specification. Also, the exact initial technique is introduced in order to initialise the model estimation when no hints of the initial filter values are available.

Chapter 2 reports explorations with the state space model: firstly on Monte-Carlo simulations of time series, then with real time series. The simulated time series are generated with different models, from the simple local level to the one with all the mentioned components. Also real data go throught different models: the first is Mills' one [12], for which we download the time series of monthly global temperatures from the database of the Hadley Centre for Climate Prediction and Research [2]; this work allows us to directly compare model specification and results. The other model for real data fits the maximum temperature time series recorded in the city of Bologna and downloaded from the site of European Climate Assessment and Dataset [9]; the purpose is to replicate the first step of Li et al.'s [10] procedure and to gain confidence with the techniques introduced in the first Chapter.

Chapter 3 presents a review of the dynamic factor models, mostly reporting definitions and parameters estimation methods. We highlight the two-step method presented by Doz, Giannone and Reichlin [5], which requires the factor model to be cast in the state space form. Chapter 4 reports some interesting numeric explorations on Monte-Carlo simulations.

Is Chapter 5 the highlight of this thesis. After a brief introduction about Li et al.'s [10] method, we modify and apply it to real data time series with the following steps:

1. download of maximum temperatures time series of some stations in Emilia Romagna (Italy);

2. analysis of each of them separately with the state space model and storage of all the prediction errors;

3. fitting the dynamic factor model to the prediction errors series in order to extract residual information;

4. simulating prediction errors series thanks to simulation smoothing in the dynamic factor model;

5. with parameters estimated at step 2 for each observed time series, computing simulated observations time series for each station;

6. re-estimation of state space model parameters according to simulated series and computing forecasts, for each station separately;

7. averaging forecasts to obtain the station final forecast.

Basically, we start with the state space model for each observed time series separately, then we fit the dinamic factor model to the prediction

errors series all together, after that we simulate prediction errors and come back to each station parameters in order to compute the simulated observations; hence we proceed for each station separately till the forecasts. Actually, in this thesis and for computational time reasons, the last steps are applyed to one station only, and specifically to the station of Bologna.

The preliminary results from the explorations performed in this thesis are very promising. We considered climatic time series of limited depth and limited cross-sectional dimension only for computational reasons and to readily verify and test the proper specification and implementation of all the modeling steps. Very preliminary tests performed on a weekly basis for much longer time series (up to 150 years) with a significantly larger cross-sectional dimension (up to 10.000) confirm the viability of the entire approach.

Chapter 1

Linear Gaussian state space model

In this Chapter we introduce a structural model for observed time series with latent components. We will employ filtering and smoothing techniques to estimate model parameters and to perform simulation. Most of the material is taken from the text by Durbin and Koopman [6], to which we refer for further information.

1.1 Introduction

A time series is a set of observations $y_1, \ldots, y_T$ ordered in time. The univariate linear Gaussian state space model for representing a time series can be written in the form

$$\begin{aligned} y_t &= Z_t \alpha_t + \varepsilon_t, & \varepsilon_t &\sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right) \\ \alpha_{t+1} &= T_t \alpha_t + R_t \eta_t, & \eta_t &\sim \mathcal{N}\left(0, Q\right) \\ & & \alpha_1 &\sim \mathcal{N}\left(a_1, P_1\right) \end{aligned} \tag{1.1}$$

where $y_t,\ t = 1, \ldots, T$ are called *observations* and α_t is an unobserved $m \times 1$ vector called *state vector*. The first equation of (1.1) is called *observation equation* and the second is called *state equation*. The matrices Z_t, T_t and R_t are assumed to be known and in this thesis are assumed constant in time (so we could remove subscript t); also the vector a_1 and the matrix P_1 are assumed to be known, while the scalar σ_ε^2 and the matrix Q are to be estimated. The disturbances terms ε_t and η_t are assumed serially independent and independent of each other at all time points. In the table below they are given the dimensions of elements of the state space model at each time point.

Scalars	**Vectors**		**Matrices**	
y_t	α_t	$m \times 1$	T	$m \times m$
ε_t	Z	$1 \times m$	R	$m \times r$
σ_ε^2	η_t	$r \times 1$	Q	$r \times r$

1.2 Latent components

The observed time series y_t could be decomposed into a set of unobserved components in the additive model

$$y_t = \mu_t + \gamma_t + c_t + \varphi_t + \varepsilon_t \tag{1.2}$$

where μ_t is a slowly varying component called *trend*, γ_t is a periodic component of fixed period called *seasonal*, c_t is a periodic component of fixed period generally greater than γ_t's called *cycle* and φ_t is a stationary autoregressive component of order p, *AR(p)*; each of these components represents a different feature of the evolution of the series. The disturbance ε_t has constant variance σ_ε^2, introduced before.

In the next Sections we describe in detail each component and finally we will assemble the system matrices.

1.2.1 Trend component

Trend is assumed to be generated by a first-order autoregressive model with a slope term generated by a random walk:

$$\begin{aligned} \mu_{t+1} &= \mu_t + \nu_t + \xi_t, & \xi_t &\sim \mathcal{N}\left(0, \sigma_\xi^2\right); \\ \nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim \mathcal{N}\left(0, \sigma_\zeta^2\right). \end{aligned}$$

1.2.2 Seasonal component

We suppose there are s 'months' (or 'weeks' or 'days') per 'year' and express the seasonal component in the trigonometric form of quasi-random walk model

$$\gamma_t = \sum_{j=1}^{\lfloor s/2 \rfloor} \gamma_{j,t} \tag{1.3}$$

where $\lfloor a \rfloor$ is the largest integer $\leq a$ and

$$\begin{aligned} \gamma_{j,t+1} &= \gamma_{j,t} \cos \lambda_j + \gamma_{j,t}^* \sin \lambda_j + \omega_{j,t}, & \omega_{j,t} &\sim \mathcal{N}\left(0, \sigma_\omega^2\right), \quad j = 1, \ldots, \lfloor s/2 \rfloor \\ \gamma_{j,t+1}^* &= -\gamma_{j,t} \sin \lambda_j + \gamma_{j,t}^* \cos \lambda_j + \omega_{j,t}^*, & \omega_{j,t}^* &\sim \mathcal{N}\left(0, \sigma_\omega^2\right) \end{aligned}$$

in which $\omega_{j,t}$ and $\omega_{j,t}^*$ terms are independent $\mathcal{N}\left(0, \sigma_\omega^2\right)$ variables and $\lambda_j = 2\pi j/s$ are fixed frequencies. The $\gamma_{j,t+1}^*$ component simply allows the seasonal to be modelled as a stochastic combination of sine and cosine waves and its interpretation is not particularly important. Instead, it is remarkable that if the stochastic terms are zero then the values of γ_t defined by (1.3) are periodic with period s by taking

$$\begin{aligned} \gamma_{j,t} &= \bar{\gamma}_j \cos \lambda_j t + \bar{\gamma}_j^* \sin \lambda_j t \\ \gamma_{j,t}^* &= -\bar{\gamma}_j \sin \lambda_j t + \bar{\gamma}_j^* \cos \lambda_j t \end{aligned}$$

which satisfy the deterministic part of previous recursions for $\gamma_{j,t+1}$ and $\gamma^*_{j,t+1}$. Moreover, those recursions provide in turn a recursion for

$$\gamma_t = \sum_{j=1}^{s/2} \left(\bar{\gamma}_j \cos \lambda_j t + \bar{\gamma}^*_j \sin \lambda_j t\right)$$

with λ_j defined as before, which is periodic with period s. As a consequence of all, it results that each set of s seasonal components defined as in (1.3) sums to zero when the stochastic terms $\omega_{j,t}$ and $\omega^*_{j,t}$ are zero.

Moreover, we should remark that in a model with a trend component the stochastic terms of the seasonal components have to be setted to zero, in order to allow a good estimation of the model itself.

1.2.3 Cycle component

Also the cycle component is modelled by a trigonometric model:

$$\begin{aligned} c_{t+1} &= \rho\left(c_t \cos \lambda_c + c^*_t \sin \lambda_c\right) + \bar{\omega}_t, & \bar{\omega}_t &\sim \mathcal{N}\left(0, \sigma^2_{\bar{\omega}}\right) \\ c^*_{t+1} &= \rho\left(-c_t \sin \lambda_c + c^*_t \cos \lambda_c\right) + \bar{\omega}^*_t, & \bar{\omega}^*_t &\sim \mathcal{N}\left(0, \sigma^2_{\bar{\omega}}\right) \end{aligned}$$

where $\bar{\omega}_t$ and $\bar{\omega}^*_t$ terms are independent $\mathcal{N}\left(0, \sigma^2_{\bar{\omega}}\right)$ variables, $0 < \rho \leq 1$ is the damping factor of the cycle and $0 \leq \lambda_c \leq \pi$ is the frequency in radians, so that the period of the cycle is $2\pi/\lambda_c$. The damping factor and the frequency are unknown parameters of the model to be estimated. If $\rho = 1$ the cycle results to be non-stationary, while if $|\rho| < 1$ the cycle is a stationary process.

1.2.4 AR component

Generally, an autoregressive process of order p can be written as

$$\varphi_t = \sum_{j=1}^{p} \theta_j \varphi_{t-j} + \tau_t, \qquad \tau_t \sim \mathcal{N}\left(0, \sigma^2_{\varphi}\right).$$

In matricial form, it results

$$\begin{pmatrix} \varphi_t \\ \varphi_{t-1} \\ \vdots \\ \varphi_{t-p+1} \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_2 & \dots & \theta_p \\ 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} \varphi_{t-1} \\ \varphi_{t-2} \\ \vdots \\ \varphi_{t-p} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tau_t$$

where we call the matrix T_φ. Stationarity is ensured if T_φ spectral radius is less than 1. Finally, in our model θ_j coefficients, for $j = 1, \dots, p$, are parameters to be estimated.

1.2.5 Assembling system matrices

To represent the model in the state space form, we shall specify vectors and matrices presented in Section (1.1). Two features are to be taken into account: most of them depend on the seasonality s that could be an even or an odd number, and the AR order p may vary in the natural numbers range; since differences occur we have to distinguish cases.

Some elements of vectors and matrices do not change according to the cases, so we present them explicitly while we specify the differences later. We have

$$\begin{aligned}
\alpha_t &= \left(\mu_t\ \nu_t\ \ \gamma_{1,t}\ \gamma^*_{1,t}\ \ \dots\ \gamma_{s^*,t}\ \gamma^*_{s^*,t}\ \ c_t\ c^*_t\ \ \varphi_t\ \varphi_{t-1}\ \ \dots\ \varphi_{t-p+1}\right), \\
Z &= (1\ 0\ \ 1\ 0\ \dots\ 1\ 0\ \ 1\ 0\ \ 1\ 0\ \dots\ 0), \\
\eta_t &= \left(\xi_t\ \zeta_t\ \ \omega_{1,t}\ \omega^*_{1,t}\ \ \dots\ \omega_{s^*,t}\ \omega^*_{s^*,t}\ \ \bar{\omega}_t\ \bar{\omega}^*_t\ \ \tau_t\right)
\end{aligned}$$

where $s^* = \lfloor s/2 \rfloor$, so $s^* = (s-1)/2$ if s is odd, $s^* = s/2$ if s is even;

$$T = \operatorname{diag}\left(T_\mu\ T_\gamma\ T_c\ T_\varphi\right), \qquad R = \operatorname{diag}\left(R_\mu\ R_\gamma\ R_c\ R_\varphi\right),$$

$$Q = \operatorname{diag}\left(Q_\mu\ Q_\gamma\ Q_c\ Q_\varphi\right)$$

with

$$T_\mu = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad T_c = \rho \begin{pmatrix} \cos\lambda_c & \sin\lambda_c \\ -\sin\lambda_c & \cos\lambda_c \end{pmatrix}$$

$$R_\mu = I_2, \qquad R_\gamma = I_{s-1}, \qquad R_c = I_2$$

$$Q_\mu = \operatorname{diag}\left(\sigma^2_\xi\ \sigma^2_\zeta\right), \quad Q_\gamma = \sigma^2_\omega I_{s-1}, \quad Q_c = \sigma^2_{\bar{\omega}} I_2, \quad Q_\varphi = \sigma^2_\varphi.$$

We remind that $\sigma^2_\omega = 0$ so $Q_\gamma = 0_{s-1}$, because of what we said at the end of Section 1.2.2.

We now analyse differences in the cases announced above. According to parity of s, if it is odd we have

$$T_\gamma = \operatorname{diag}\left(C_1\ \dots\ C_{s^*}\right) \quad \text{with } C_j = \begin{pmatrix} \cos\lambda_j & \sin\lambda_j \\ -\sin\lambda_j & \cos\lambda_j \end{pmatrix},\ \lambda_j = \frac{2\pi j}{s}$$

while if s is even the matrix becomes

$$T_\gamma = \operatorname{diag}\left(C_1\ \dots\ C_{s^*-1} - 1\right);$$

moreover, when $s = s/2$ it results $\lambda_{s/2} = \pi$, thus

$$\begin{aligned}
\gamma_{s/2,t+1} &= \gamma_{s/2,t} \cos\pi = -\gamma_{s/2,t} \\
\gamma^*_{s/2,t+1} &= \gamma^*_{j,t} \cos\pi = -\gamma^*_{s/2,t}
\end{aligned}$$

which are redundant. As a consequence, $\gamma^*_{s^*,t}$ and $\omega^*_{s^*,t}$ are left out of α_t and η_t respectively, and so the corresponding 0 in vector Z.

Changes about the AR component depend on the value of p as it determines sizes of matrices:

$$T_\varphi \in \mathbb{R}^{p\times p}$$

$$R_\varphi = (1\ 0\ \ldots\ 0)' \in \mathbb{R}^{p\times 1}$$

where T_φ has already been presented in the previous Section.

In all cases, we remark that σ_ε^2, matrix Q, ρ, λ_c and θ_j, $j = 1, \ldots, p$ are parameters to be estimated.

1.3 Kalman filter

As the model is completely specified, it is simple setting arbitrary values for parameters and drawing a sample of observations using model (1.1) recursions. According to these observations, the model is obviously correctly specified. The problem is that usually we only have observations data and our aim is to fit the model to data. The Kalman filter recursions allow us to estimate mean and variance of α_t given $y_1, \ldots, y_{t-1}$.

1.3.1 Basic result in multivariate regression theory

Deriving Kalman filter recursions requires we present an important lemma of multivariate regression theory. Because of the variety of application contexts, we present here its general proof before exploring all its consequences. Moreover this form of presentation exposes the intrinsically simple nature of the mathematical theory underlying the state space approach to time series analysis.

Lemma 1. *Let x and y be jointly normally distributed random vectors with*

$$E\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \qquad Var\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma'_{xy} & \Sigma_{yy} \end{pmatrix}$$

where Σ_{yy} is assumed to be a nonsingular matrix.

Then the conditional distribution of x given y is normal with mean vector

$$E(x|y) = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y) \tag{1.4}$$

and variance matrix

$$Var(x|y) = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma'_{xy}. \tag{1.5}$$

Proof. Let $z = x - \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)$. Since the transformation from (x, y) to (y, z) is linear and (x, y) is normally distributed by hypothesis, the

joint distribution of y and z is normal. We have

$$\begin{aligned} E(z) &= \mu_x, \\ Var(z) &= E\left[(z-\mu_x)(z-\mu_x)'\right] \\ &= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}', \\ Cov(y,z) &= E\left[y(z-\mu_x)'\right] \\ &= E\left[y(x-\mu_x)' - y(y-\mu_y)'\Sigma_{yy}^{-1}\Sigma_{xy}'\right] \\ &= 0. \end{aligned}$$

Using the result that if two vectors are normal and uncorrelated they are independent, $Cov(y,z) = 0$ implies that z is distributed independently of y. Since the distribution of z does not depend on y, its conditional distribution given y is the same as its unconditional distribution, that is, it is normal with mean vector μ_x and variance matrix $\Sigma_{xx} - \Sigma_x y\Sigma_{yy}^{-1}\Sigma_{xy}'$ which is the same as the thesis. By definition of z, it follows that the conditional distribution of x given y is normal with mean vector and variance matrix as in the thesis. □

1.3.2 Derivation of the Kalman filter

In this section, we derive Kalman filter for model (1.1) for the case where the initial state α_1 is $\mathcal{N}(a_1, P_1)$ and a_1, P_1 are known. First of all, let denote $y_1, \ldots, y_t$ by Y_t. Our object is to obtain the conditional distributions of α_t and α_{t+1} given Y_t for $t = 1, \ldots, T$. Let

$$\begin{aligned} a_{t|t} &= E(\alpha_t|Y_t), & P_{t|t} &= Var(\alpha_t|Y_t), \\ a_{t+1} &= E(\alpha_{t+1}|Y_t), & P_{t+1} &= (\alpha_{t+1}|Y_t). \end{aligned}$$

Since all distributions are normal, it follows from the Lemma that conditional distributions of subsets of variables given other subsets of variables are also normal; the distributions of α_t given Y_t and α_{t+1} given Y_t are therefore given by $\mathcal{N}\left(a_{t|t}, P_{t|t}\right)$ and $\mathcal{N}(a_{t+1}, P_{t+1})$. We proceed inductively calculating $a_{t|t}$, a_{t+1}, $P_{t|t}$ and P_{t+1} from a_t and P_t recursively for $t = 1, \ldots, T$.

Let define the *one-step ahead prediction error* of y_t given Y_{t-1}

$$v_t = y_t - E(y_t|Y_{t-1}) = y_t - E(Z\alpha_t + \varepsilon_t|Y_{t-1}) = y_t - Za_t.$$

When Y_{t-1} and v_t are fixed, then Y_t is fixed, and viceversa; thus $E(\alpha_t|Y_t) = E(\alpha_t|Y_{t-1}, v_t)$. Moreover

$$E(v_t|Y_{t-1}) = E(y_t - Z\alpha_t|Y_{t-1}) = E(Z\alpha_t + \varepsilon_t - Za_t|Y_{t-1}) = 0$$

implies $E(v_t) = 0$ and $Cov(y_j, v_t) = E\left[y_j E(v_t|Y_{t-1})'\right] = 0$ for $j = 1, \ldots, t-1$. Also,

$$a_{t|t} = E(\alpha_t|Y_t) = E(\alpha_t|Y_{t-1}, v_t)$$

and

$$a_{t+1} = E\left(\alpha_{t+1}|Y_t\right) = E\left(\alpha_{t+1}|Y_{t-1}, v_t\right).$$

Now we apply the Lemma to the conditional joint distribution of α_t and v_t given Y_{t-1}, taking x and y in the Lemma as α_t and v_t here. We obtain

$$a_{t|t} = E\left(\alpha_t|Y_{t-1}\right) + Cov\left(\alpha_t, v_t\right)\left[Var\left(v_t\right)\right]^{-1} v_t$$

where Cov and Var are covariance and variance conditional to Y_{t-1}. Here, $E\left(\alpha_t|Y_{t-1}\right) = a_t$ by definition of a_t and

$$\begin{aligned} Cov\left(\alpha_t, v_t\right) &= E\left[\alpha_t\left(Z\alpha_t + \varepsilon_t - Za_t\right)'|Y_{t-1}\right] \\ &= E\left[\alpha_t\left(\alpha_t - a_t\right)'Z'|Y_{t-1}\right] \\ &= P_t Z' \end{aligned}$$

by definition of P_t. Let define the *one-step ahead prediction error variance*

$$F_t = Var\left(v_t|Y_{t-1}\right) = Var\left(Z\alpha_t + \varepsilon_t - Za_t|Y_{t-1}\right) = ZP_tZ' + \sigma_\varepsilon^2$$

Then

$$a_{t|t} = a_t + P_tZ'F_t^{-1}v_t. \tag{1.6}$$

By (1.5) of the Lemma, we have

$$\begin{aligned} P_{t|t} = Var\left(\alpha_t|Y_t\right) &= Var\left(\alpha_t|Y_{t-1}, v_t\right) \\ &= Var\left(\alpha_t|Y_{t-1}\right) - Cov\left(\alpha_t, v_t\right)\left[Var\left(v_t\right)\right]^{-1} Cov\left(\alpha_t, v_t\right)' \\ &= P_t - P_tZ'F_t^{-1}ZP_t. \end{aligned} \tag{1.7}$$

We assume that F_t is non-singular; this assumption is usually valid in well-formulated models. Relations (1.6) and (1.7) are sometimes referred to as the *updating step* of the Kalman filter.

We now derive recursions for a_{t+1} and P_{t+1}. Since $\alpha_{t+1} = T\alpha_t + R\,\eta_t$, we have

$$\begin{aligned} a_{t+1} &= E\left(T\alpha_t + R\,\eta_t|Y_t\right) \\ &= TE\left(\alpha_t|Y_t\right), \end{aligned} \tag{1.8}$$

$$\begin{aligned} P_{t+1} &= Var\left(T\alpha_t + R\,\eta_t|Y_t\right) \\ &= T\,Var\left(\alpha_t|Y_t\right)T' + RQR' \end{aligned} \tag{1.9}$$

for $t = 1, \ldots, T$. Substituting (1.6) into (1.8) gives

$$\begin{aligned} a_{t+1} &= Ta_{t|t} \\ &= Ta_t + K_tv_t \end{aligned} \tag{1.10}$$

for $t = 1, \ldots, T$, where

$$K_t = TP_tZ'F_t^{-1} \tag{1.11}$$

is reffered to as the *Kalman gain*. Finally, substituting from (1.7) and (1.11) in (1.9) gives

$$P_{t+1} = TP_t (T - K_t Z)' + RQR' \tag{1.12}$$

for $t = 1, \ldots, T$. Relations (1.10) and (1.12) are sometimes called the *prediction step* of the Kalman filter.

For convenience we now collect together the filtering equations:

$$\begin{aligned} v_t &= y_t - Za_t, & F_t &= ZP_tZ' + \sigma_\varepsilon^2, \\ a_{t|t} &= a_t + P_tZ'F_t^{-1}v_t, & P_{t|t} &= P_t - P_tZ'F_t^{-1}ZP_t, \\ a_{t+1} &= Ta_t + K_tv_t, & P_{t+1} &= TP_t (T - K_tZ)' + RQR', \end{aligned} \tag{1.13}$$

for $t = 1, \ldots, T$, where $K_t = TP_tZ'F_t^{-1}$, and a_1, P_1 are mean vector and variance matrix of the initial state vector α_1 and are supposed known. Indeed, once $a_{t|t}$ and $P_{t|t}$ are computed, it suffices to adopt the relations

$$a_{t+1} = Ta_{t|t}, \qquad P_{t+1} = TP_{t|t}T' + RQR'$$

for predicting the state vector α_{t+1}.

1.4 Maximum log-likelihood criterium

The maximum log-likelihood criterium joined with Kalman recursions allow us to estimate model parameters having observations $y_1, \ldots, y_T$.

Since

$$p(y_1, \ldots, y_t) = p(Y_{t-1})\, p(y_t|Y_{t-1})$$

for $t = 2, \ldots, T$, the joint density of $y_1, \ldots, y_T$ can be expressed as

$$p(y_1, \ldots, y_T) = \prod_{t=1}^{T} p(y_t|Y_{t-1})$$

where $p(y_1|Y_0) = p(y_1)$. By definition, the likelihood is $L = p(y_1, \ldots, y_t)$, so the log-likelihood may be written as

$$\log L = \sum_{t=1}^{T} \log p(y_t|Y_{t-1})$$

with the previous notation when $t = 1$. In model (1.1), we have $E(y_t|Y_{t-1}) = Za_t$, $v_t = y_t - Za_t$ and $F_t = Var(y_t|Y_{t-1})$, thus $p(y_t|Y_{t-1}) = \mathcal{N}(Za_t, F_t)$. As a consequence,

$$\log L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum \left(\log F_t + \frac{v_t^2}{F_t}\right).$$

We remark that v_t and F_t are computed with Kalman filter recursions, thus they depend inherently on the model parameters σ_ε^2, Q, ρ, λ_c and θ_j for $j = 1, \ldots, p$. Maximising $\log L$ an estimation of these parameters is obtained.

1.5 Model miss-specification tests

Let standardising the one-step ahead prediction error defining

$$e_t = \frac{v_t}{\sqrt{F_t}}, \quad t = 1, \dots T,$$

which are obviously $\sim \mathcal{N}(0, 1)$ and serially independent. We can check the following properties.

- Normality
  The first four moments of the standardised prediction errors are given by

$$m_1 = \frac{1}{T} \sum_{t=1}^{T} e_t,$$
$$m_q = \frac{1}{T} \sum_{t=1}^{T} (e_t - m_1)^q, \quad q = 2, 3, 4.$$

  Skewness and kurtosis are defined respectively as

$$S = \frac{m_3}{\sqrt{m_2^3}}, \qquad K = \frac{m_4}{m_2^2},$$

  and when the model assumptions are valid it can be shown that

$$S \sim \mathcal{N}\left(0, \frac{6}{T}\right), \qquad K \sim \mathcal{N}\left(3, \frac{24}{T}\right).$$

  An analysis of quantiles can be used to verify the null hypotesis of distribution.

- Heteroscedasticity
  A simple test for heteroscedasticity is obtained by comparing the sum of squares of two exclusive subsets of the sample. Let define

$$H(h) = \frac{\sum_{t=T-h+1}^{T} e_t^2}{\sum_{t=1}^{h} e_t^2};$$

  thus, for an arbitrary h, $H(h)$ is $F_{h,h}$-distributed under null hypothesis of homoscedasticity. Again, analysis of quantiles allows accepting or rejecting the assumption.

- Serial correlation
  Let c_j be the j-th correlogram value

$$c_j = \frac{1}{Tm_2} \sum_{t=j+1}^{T} (e_t - m_1)(e_{t-j} - m_1)$$

for $1 \leq j \leq T$. Under the null hypothesis of uncorrelation, the correlogram of the prediction errors should reveal serial correlation insignificant, that is $|c_j| \leq 1/\sqrt{T} \quad \forall j$. Moreover, Ljung and Box [11] have proven that the Box-Ljung statistic

$$Q(k) = T(T+2) \sum_{j=1}^{k} \frac{c_j^2}{T-j}$$

is χ^2_k-distributed (that is a χ^2-distribution with k degrees of freedom).

Finally, we may also check the t-statistics, which are obtained dividing the estimated parameters by approximated estimation errors; such approximation is achieved taking the diagonal of the inverse matrix of the Hessian matrix computed with the maximisation of the log-likelihood function. Generally, a t-statistic is highly significant if its value is higher than 3.

1.6 State smoothing

We now derive the conditional distribution of α_t given the entire series $y_1, \ldots, y_T$ for $t = 1, \ldots, T$; we calculate the conditional mean vector $\hat{\alpha}_t = E(\alpha_t | Y_T)$ and the conditional variance matrix $V_t = Var(\alpha_t | Y_T)$, respectively referred to as *smoothed state* vector and *smoothed state variance* matrix; the whole operation is called *state smoothing*. We construct recursion for $\hat{\alpha}_t$ and V_t on the assumption that $\alpha_1 \sim \mathcal{N}(a_1, P_1)$ where a_1 and P_1 are known, we shall also use the Lemma presented in Section 1.3.1.

1.6.1 Preliminary definitions

Before deriving state smoothing recursions, we need introduce some new elements. Let define the *state estimation error* as

$$x_t = \alpha_t - a_t \qquad \text{with} \quad Var(x_t) = P_t.$$

It follows immediately from the Kalman filter relations and the definition of x_t that

$$\begin{aligned} v_t &= y_t - Z a_t \\ &= Z\alpha_t + \varepsilon_t - Z a_t \\ &= Z x_t + \varepsilon_t \end{aligned} \tag{1.14}$$

and

$$\begin{aligned} x_{t+1} &= \alpha_{t+1} - a_{t+1} \\ &= T\alpha_t + R\,\eta_t - Ta_t - K_t v_t \\ &= Tx_t + R\,\eta_t - K_t Z x_t - k_t \varepsilon_t \\ &= L_t x_t + R\,\eta_t - K_t \varepsilon_t \end{aligned} \tag{1.15}$$

where $K_t = TP_t Z' F_t^{-1}$ and $L_t = T - K_t Z$.

By definition of the one-step ahead prediction error $v_t = y_t - E(y_t|Y_{t-1})$, v_t is the part of y_t that cannot be predicted from the past, thus the v_t's are sometimes referred to as *innovations*. Finally, let denote the vector $(v_t', \ldots, v_T')'$ by $v_{t:T}$ and note also that Y_T is fixed when Y_{t-1} and $v_{t:T}$ are fixed.

1.6.2 Smoothed state vector

To calculate $\hat{\alpha}_t$ we apply the Lemma to the conditional joint distributions of α_t and $v_{t:T}$ given Y_{t-1}, taking x and y of the Lemma as α_t and $v_{t:T}$ here. Using the fact that $v_t, \ldots, v_T$ are independent of Y_{t-1} and of each other, we therefore have from (1.4)

$$\begin{aligned} \hat{\alpha}_t &= E(\alpha_t|Y_n) = E(\alpha_t|Y_{t-1}, v_{t:T}) \\ &= a_t + \sum_{j=t}^{T} Cov(\alpha_t, v_j) F_j^{-1} v_j \end{aligned} \tag{1.16}$$

since $E(\alpha_t|Y_{t-1}) = a_t$ for $t = 1, \ldots, T$, where Cov refers to covariance in the conditional distribution given Y_{t-1} and $F_j = Var(v_j|Y_{t-1})$. It follows from (1.14) that

$$\begin{aligned} Cov(\alpha_t, v_j) &= E\left(\alpha_t v_j'|Y_{t-1}\right) \\ &= E\left[\alpha_t (Zx_j + \varepsilon_j)'|Y_{t-1}\right] \\ &= E\left(\alpha_t x_j'|Y_{t-1}\right) Z' \end{aligned} \tag{1.17}$$

for $j = t, \ldots, T$. Moreover,

$$\begin{aligned} E(\alpha_t x_t'|Y_{t-1}) &= E\left[\alpha_t (\alpha_t - a_t)|Y_{t-1}\right] = P_t \\ E\left(\alpha_t x_{t+1}'|Y_{t-1}\right) &= E\left[\alpha_t (L_t x_t + R\,\eta_t - K_t\varepsilon_t)'|Y_{t-1}\right] = P_t L_t' \\ E\left(\alpha_t x_{t+2}'|Y_{t-1}\right) &= P_t L_t' L_{t+1}' \\ &\vdots \\ E(\alpha_t x_T'|Y_{t-1}) &= P_t L_t' \cdots L_{T-1}' \end{aligned} \tag{1.18}$$

using (1.15) repetedly for $t+1, t+2, \ldots$. Note that we interpret $L_t' \cdots L_{T-1}'$ as I_m when $t = T$ and as L_{T-1}' when $t = T-1$.

Substituting into (1.16) gives

$$\begin{aligned}\hat{\alpha}_T &= a_T + P_T Z' F_T^{-1} v_T \\ \hat{\alpha}_{T-1} &= a_{T-1} + P_{T-1} Z' F_{T-1}^{-1} v_{T-1} + P_{T-1} L_{T-1}' Z' F_T^{-1} v_T \\ \hat{\alpha}_t &= a_t + P_t Z' F_t^{-1} v_t + P_t L_t' Z' F_{t+1}^{-1} v_{t+1} + \cdots + P_t L_t' \cdots L_{T-1}' Z' F_T^{-1} v_T\end{aligned}$$

for $t = T-2, T-3, \ldots, 1$. We can therefore express the smoothed state vector as

$$\hat{\alpha}_t = a_t + P_t r_{t-1}$$

where

$$\begin{aligned}r_{T-1} &= Z' F_T^{-1} v_T \\ r_{T-2} &= Z' F_{T-1}^{-1} v_{T-1} + L_{T-1}' Z' F_T^{-1} v_T \\ r_{t-1} &= Z' F_t^{-1} v_t + L_t' Z' F_{t+1}^{-1} v_{t+1} + \cdots + L_t' \cdots L_{T-1}' Z' F_T^{-1} v_T \qquad (1.19)\end{aligned}$$

for $t = T-2, T-3, \ldots, 1$. The vector r_{t-1} is a weighted sum of innovations v_j occurring after time $t-1$, that is for $j = t, \ldots, T$; the value at time t is

$$r_t = Z' F_{t+1}^{-1} v_{t+1} + L_{t+1}' Z' F_{t+2}^{-1} v_{t+2} + \cdots + L_{t+1}' \cdots L_{T-1}' Z' F_T^{-1} v_T \quad (1.20)$$

and $r_T = 0$ since no innovations are available after time T. Substituting from (1.20) into (1.19) we obtain the backwards recursion

$$r_{t-1} = Z' F_t^{-1} v_t + L_t' r_t, \quad t = T, \ldots, 1$$

with $r_T = 0$.

Collecting these results together gives the recursions for state smoothing:

$$\hat{\alpha}_t = a_t + P_t r_{t-1}, \qquad r_{t-1} = Z' F_t^{-1} v_t + L_t' r_t \qquad (1.21)$$

for $t = T, \ldots, 1$ with $r_T = 0$.

1.6.3 Smoothed state variance matrix

We now derive a recursion for calculating V_t. Applying the Lemma as at the beginning of previous Section, we obtain from (1.5)

$$\begin{aligned}V_t &= Var\left(\alpha_t | Y_T\right) = Var\left(\alpha_t | Y_{t-1}, v_{t:T}\right) \\ &= P_t - \sum_{j=t}^{T} Cov\left(\alpha_t, v_j\right) F_j^{-1} Cov\left(\alpha_t, v_j\right)' \qquad (1.22)\end{aligned}$$

where $Cov\left(\alpha_t, v_j\right)$ and F_j are as in (1.16). Using (1.17) and (1.18) we obtain immediately

$$\begin{aligned}V_t &= P_t - P_t Z' F_t^{-1} Z P_t - P_t L_t' Z' F_{t+1}^{-1} Z L_t P_t - \cdots \\ &\quad - P_t L_t' \cdots L_{T-1}' Z' F_T^{-1} Z L_{T-1} \cdots L_t P_t \\ &= P_t - P_t N_{t-1} P_t\end{aligned}$$

where

$$N_{t-1} = Z'F_t^{-1}Z - L_t'Z'F_{t+1}^{-1}ZL_t + \cdots + L_t' \cdots L_{T-1}'Z'F_T^{-1}ZL_{T-1} \cdots L_t. \tag{1.23}$$

We note that here $L_t' \cdots L_{T-1}'$ is interpreted as in the previous Section. Moreover, we have at time t

$$N_t = Z'F_{t+1}^{-1}Z - L_{t+1}'Z'F_{t+2}^{-1}ZL_{t+1} + \cdots + L_{t+1}' \cdots L_{T-1}'Z'F_T^{-1}ZL_{T-1} \cdots L_{t+1}. \tag{1.24}$$

Substituting (1.24) into (1.23), we obtain the backwards recursion

$$N_{t-1} = Z'F_t^{-1}Z + L_t'N_tL_t, \quad t = T, \ldots, 1.$$

Noting from (1.24) that $N_{T-1} = Z'F_T^{-1}Z$, we deduce that this recursion is initialised with $N_T = 0$.

Collecting these results together, we find that V_t can be computed with the recursions

$$N_{t-1} = Z'F_t^{-1}Z + L_t'N_tL_t, \qquad V_t = P_t - P_tN_{t-1}P_t$$

for $t = T, \ldots, 1$ with $N_T = 0$. Moreover, since $v_{t+1}, \ldots, v_T$ are independent, it follows from (1.20) and (1.24) that $N_t = Var\,(r_t)$.

1.7 Simulation smoothing

It is simple to draw samples generated by the linear Gaussian state space model (1.1): we first draw random samples ε^+ and η^+ of the disturbance vectors and then we generate observations y^+ using the model recursions. We now show how to draw samples of disturbance vectors and state vector conditional on the observed time series $y_1, \ldots, y_T$; this operation is called *simulation smoothing*. The difference between simulating a sample unconditionally and simulating a sample conditionally to the linear Gaussian state space model is that the conditional sample is closer to the smoothed state $\hat{\alpha}_t$ computed from the observations vector, while the unconditional sample has nothing in common with it.

Let show how to draw a conditional sample $\tilde{\alpha}$. We draw T random disturbance vectors $w_t^+ = (\varepsilon_t, \eta_t')'$, independent of each other, from a distribution $\mathcal{N}(0, \varphi_t)$ with $\varphi_t = \mathrm{diag}\,(\sigma_\varepsilon^2, Q)$. Then we generate an unconditional sample α^+ of the state vector using model (1.1) as a recursion initialised by $\alpha_1^+ \sim \mathcal{N}(a_1, P_1)$ with α and w replaced by α^+ and w^+ respectively. Next we compute $\hat{\alpha} = E\,(\alpha|y_1, \ldots, y_T)$ and $\hat{\alpha}^+ = E\left(\alpha|y_1^+, \ldots, y_T^+\right)$ by the Kalman filter and smoothing recursions. The required draw of $\tilde{\alpha}$ is given by the expression

$$\tilde{\alpha} = \alpha^+ - \hat{\alpha}^+ + \hat{\alpha}.$$

1.8 Initialisation

In previous Sections we have considered the operations of filtering and smoothing for the model (1.1) under the assumption that $\alpha_1 \sim \mathcal{N}(a_1, P_1)$ where a_1 and P_1 are known; in most applications, however, the situation is different. Process of starting up the series when initial state is unknown is called *initialisation*.

We consider the following model for α_1:

$$\alpha_1 = a + A\delta + R_0\,\eta_0, \quad \eta_0 \sim \mathcal{N}(0, Q_0)$$

where a is a $m \times 1$ zero vector (in general, it is known and usually is treated as a zero vector), δ is a $q \times 1$ unknown vector, and the $m \times q$ matrix A and the $m \times (m-q)$ matrix R_0 are selection matrices so that $A'R_0 = 0$; the idea is to separate out α_1 into a constant part a, a non-stationary part $A\delta$ and a stationary part $R_0\eta_0$. We assume that δ is a vector of random normal variables with infinite variances:

$$\delta \sim \mathcal{N}(0, \kappa I_q)$$

where we let $\kappa \to \infty$; a vector with such a distribution is said to be *diffuse* and then the entire procedure is called *diffuse initialisation*. We begin by considering the initial conditions

$$\begin{aligned} a_1 &= E(\alpha_1) = a \\ P_1 &= Var(\alpha_1) = \kappa P_\infty + P_* \end{aligned}$$

where $P_\infty = AA'$ and $P_* = R_0 Q_0 R_0'$; by definition of A, P_∞ is a $m \times m$ diagonal matrix. Consequently, modifications to Kalman filter are required.

A simple approximate technique is to replace κ by an arbitrary large number. However, this device is suitable for exploratory work but not recommended for general use since it can lead to large rounding errors. We therefore develop an exact treatment: we expand matrix products as power series in κ^{-1}, taking only the first two or three terms as required, and then we let $\kappa \to \infty$ to obtain the dominant term; the word *exact* emphasize the difference with the other approximating method.

1.8.1 Exact initial Kalman filter

Analogously to the decomposition of the initial matrix P_1, the mean square error matrix P_t may be decomposed as

$$P_t = \kappa P_{\infty,t} + P_{*,t} + O\left(\kappa^{-1}\right), \quad t = 2, \ldots, T$$

where $P_{\infty,t}$ and $P_{*,t}$ do not depend on κ. Since $F_t = ZP_tZ' + \sigma_\varepsilon^2$ and $M_t = P_tZ'$, this decomposition leads to the similar ones

$$F_t = \kappa F_{\infty,t} + F_{*,t} + O\left(\kappa^{-1}\right), \quad M_t = \kappa M_{\infty,t} + M_{*,t} + O\left(\kappa^{-1}\right)$$

where

$$\begin{aligned} F_{\infty,t} &= ZP_{\infty,t}Z', & M_{\infty,t} &= P_{\infty,t}Z', \\ F_{*,t} &= ZP_{*,t}Z' + \sigma_\varepsilon^2, & M_{*,t} &= P_{*,t}Z', \end{aligned}$$

for $t = 1, \ldots, T$. It is important to note that a zero matrix $M_{\infty,t}$ (whether $P_{\infty,t}$ is a zero matrix or not) implies that $F_{\infty,t} = 0$. As in Section 1.3.2 about Kalman filter, we assume that F_t is non-singular. The derivation of the exact initial Kalman filter is based on the expansion for $F_t^{-1} = \left[\kappa F_{\infty,t} + F_{*,t} + O\left(\kappa^{-1}\right)\right]^{-1}$ as a power series in κ^{-1}, that is

$$F_t^{-1} = F_t^{(0)} + \kappa^{-1}F_t^{(1)} + \kappa^{-2}F_t^{(2)} + O\left(\kappa^{-3}\right)$$

for large κ. Since $I_p = F_t F_t^{-1}$, we have

$$\begin{aligned} I_p = &\left(\kappa F_{\infty,t} + F_{*,t} + \kappa^{-1}F_{a,t} + \kappa^{-2}F_{b,t} + \cdots\right) \\ &\times \left(F_t^{(0)} + \kappa^{-1}F_t^{(1)} + \kappa^{-2}F_t^{(2)} + \cdots\right) \end{aligned}$$

and comparing the two sides the following equations are obtained for $F_t^{(0)}$, $F_t^{(1)}$ and $F_t^{(2)}$, while further terms are not required:

$$\begin{aligned} F_{\infty,t}F_t^{(0)} &= 0 \\ F_{*,t}F_t^{(0)} + F_{\infty,t}F_t^{(1)} &= I_p \\ F_{a,t}F_t^{(0)} + F_{*,t}F_t^{(1)} + F_{\infty,t}F_t^{(2)} &= 0 \end{aligned} \tag{1.25}$$

We shall consider only the cases where $F_{\infty,t}$ is non-singular or $F_{\infty,t} = 0$, because this limitation gives a complete solution for the case of univariate series; moreover, $F_{\infty,t}$ non-singular is the most common case, thus we deal only with this one here.

We have from (1.25)

$$F_t^{(0)} = 0, \quad F_t^{(1)} = F_{\infty,t}^{-1}, \quad F_t^{(2)} = -F_{\infty,t}^{-1}F_{*,t}F_{\infty,t}^{-1}.$$

Since the matrices $K_t = TM_tF_t^{-1}$ and $L_t = T - K_tZ$ depend on the inverse matrix F_t^{-1}, they also can be expressed as power series in κ^{-1}:

$$K_t = T\left[\kappa M_{\infty,t} + M_{*,t} + O\left(\kappa^{-1}\right)\right]\left[\kappa^{-1}F_t^{(1)} + \kappa^{-2}F_t^{(2)} + \cdots\right]$$

so

$$K_t = K_t^{(0)} + \kappa^{-1}K_t^{(1)} + O\left(\kappa^{-2}\right), \quad L_t = L_t^{(0)} + \kappa^{-1}L_t^{(1)} + O\left(\kappa^{-2}\right)$$

where

$$\begin{aligned} K_t^{(0)} &= TM_{\infty,t}F_t^{(1)}, & L_t^{(0)} &= T - K_t^{(0)}Z, \\ K_t^{(1)} &= TM_{*,t}F_t^{(1)} + TM_{\infty,t}F_t^{(2)}, & L_t^{(1)} &= -K_t^{(1)}Z. \end{aligned}$$

Referring to the recursion (1.10) for a_{t+1} starting with $t = 1$, we have that

$$a_t = a_t^{(0)} + \kappa^{-1} a_t^{(1)} + O\left(\kappa^{-2}\right)$$

where $a_1^{(0)} = a$ and $a_1^{(1)} = 0$. As a consequence v_t has the form

$$v_t = v_t^{(0)} + \kappa^{-1} v_t^{(1)} + O\left(\kappa^{-2}\right)$$

where $v_t^{(0)} = y_t - Z a_t^{(0)}$ and $v_t^{(1)} = -Z a_t^{(1)}$. Therefore,

$$\begin{aligned} a_{t+1} &= T a_t + K_t v_t \\ &= T\left[a_t^{(0)} + \kappa^{-1} a_t^{(1)} + O\left(\kappa^{-2}\right)\right] \\ &\quad + \left[K_t^{(0)} + \kappa^{-1} K_t^{(1)} + O\left(\kappa^{-2}\right)\right]\left[v_t^{(0)} + \kappa^{-1} v_t^{(1)} + O\left(\kappa^{-2}\right)\right] \end{aligned}$$

which becomes as $\kappa \to \infty$

$$a_{t+1}^{(0)} = T a_t^{(0)} + K_t^{(0)} v_t^{(0)}, \quad t = 1, \ldots, T. \tag{1.26}$$

The updating equation (1.12) for P_{t+1} becomes

$$\begin{aligned} P_{t+1} &= T P_t L_t' + RQR' \\ &= T\left[\kappa P_{\infty,t} + P_{*,t} + O\left(\kappa^{-1}\right)\right]\left[L_t^{(0)} + \kappa^{-1} L_t^{(1)} + O\left(\kappa^{-2}\right)\right]' + RQR'. \end{aligned}$$

Consequently, letting $\kappa \to \infty$ gives the following updates:

$$\begin{aligned} P_{\infty,t+1} &= T P_{\infty,t} L_t^{(0)\prime} \\ P_{*,t+1} &= T P_{\infty,t} L_t^{(1)\prime} + T P_{*,t} L_t^{(0)\prime} + RQR' \end{aligned} \tag{1.27}$$

for $t = 1, \ldots, T$. Actually, the matrix P_{t+1} also depends on terms in κ^{-1}, κ^{-2}, ecc. but these terms are not taken into account in the updating equations. Recursions (1.26) and (1.27) constitute the exact Kalman filter.

1.8.2 Transition to the usual Kalman filter

It can be shown that for non-degenerate models there exists a value d of t such that

$$\begin{aligned} P_{\infty,t} &\neq 0 \text{ for } t \leq d \\ P_{\infty,t} &= 0 \text{ for } t > d \end{aligned}$$

Thus, when $t > d$ we have $P_t = P_{*,t} + O\left(\kappa^{-1}\right)$ and letting $\kappa \to \infty$ we can use the usual Kalman filter (1.13) starting with $a_{d+1} = a_{d+1}^{(0)}$ and $P_{d+1} = P_{*,d+1}$.

If such a value of t does not exist the model is called *degenerate*, since a series of observations that does not even contain enough information to estimate the initial conditions is obviously useless.

1.8.3 Exact initial state smoothing

To obtain the recursions for the smoothing state vector equation $\hat{\alpha}_t = a_t + P_t r_{t-1}$ given in (1.21) for $t = d, \ldots, 1$, we return to the recursion for r_{t-1}, that is

$$r_{t-1} = Z' F_t^{-1} v_t + L_t' r_t, \quad t = T, \ldots, 1$$

with $r_T = 0$. Since r_{t-1} depends on F_t^{-1} and L_t, which can both be expressed as power series in κ^{-1}, it may be written

$$r_{t-1} = r_{t-1}^{(0)} + \kappa^{-1} r_{t-1}^{(1)} + O\left(\kappa^{-2}\right), \quad t = d, \ldots, 1.$$

Comparing the two expressions for r_{t-1} it results, for $F_{\infty,t}$ non-singular,

$$\begin{aligned} r_{t-1}^{(0)} + \kappa^{-1} r_{t-1}^{(1)} + \cdots = Z' &\left(\kappa^{-1} F_t^{(1)} + \kappa^{-2} F_t^{(2)} + \cdots\right)\left(v_t^{(0)} + \kappa^{-1} v_t^{(1)} + \cdots\right) \\ &+ \left(L_t^{(0)} + \kappa^{-1} L_t^{(1)} + \cdots\right)'\left(r_t^{(0)} + \kappa^{-1} r_t^{(1)} + \cdots\right) \end{aligned}$$

leading to the recursions

$$\begin{aligned} r_{t-1}^{(0)} &= L_t^{(0)\prime} r_0^{(0)} \\ r_{t-1}^{(1)} &= Z' F_t^{(1)} v_t^{(0)} + L_t^{(0)\prime} r_t^{(1)} + L_t^{(1)\prime} r_t^{(0)} \end{aligned}$$

for $t = d, \ldots, 1$ with $r_d^{(0)} = r_d$ and $r_t^{(1)} = 0$.

Thus, the smoothed state vector is

$$\begin{aligned} \hat{\alpha}_t &= a_t + P_t r_{t-1} \\ &= a_t + \left[\kappa P_{\infty,t} + P_{*,t} + O\left(\kappa^{-1}\right)\right]\left[r_{t-1}^{(0)} + \kappa^{-1} r_{t-1}^{(1)} + O\left(\kappa^{-2}\right)\right] \\ &= a_t + \kappa P_{\infty,t}\left(r_{t-1}^{(0)} + \kappa^{-1} r_{t-1}^{(1)}\right) + P_{*,t}\left(r_{t-1}^{(0)} + \kappa^{-1} r_{t-1}^{(1)}\right) + O\left(\kappa^{-1}\right) \\ &= a_t + \kappa P_{\infty,t} r_{t-1}^{(0)} + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)} + O\left(\kappa^{-1}\right) \end{aligned}$$

where $a_t = a_t^{(0)} + \kappa^{-1} a_t^{(1)} + \cdots$. It is clear that it must be $P_{\infty,t} r_{t-1}^{(0)} = 0$ for all t because this expression makes sense, but this is true by definition of d as $\kappa \to \infty$. So, letting $\kappa \to \infty$, we finally obtain

$$\hat{\alpha}_t = a_t^{(0)} + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)}, \quad t = d, \ldots, 1$$

with $r_d^{(0)} = r_d$ and $r_t^{(1)} = 0$.

Chapter 2

Numeric explorations

We built many different models in order to gain confidence with Kalman filter and log-likelihood maximisation aiming to reconstruct unobserved components of data. We started up drawing samples of data with Monte-Carlo simulation and adding components step by step; then we downloaded public data and fit models to them. In this second stage, we referred to a paper by Mills [12] and repeated his explorations.

2.1 Monte-Carlo analysis

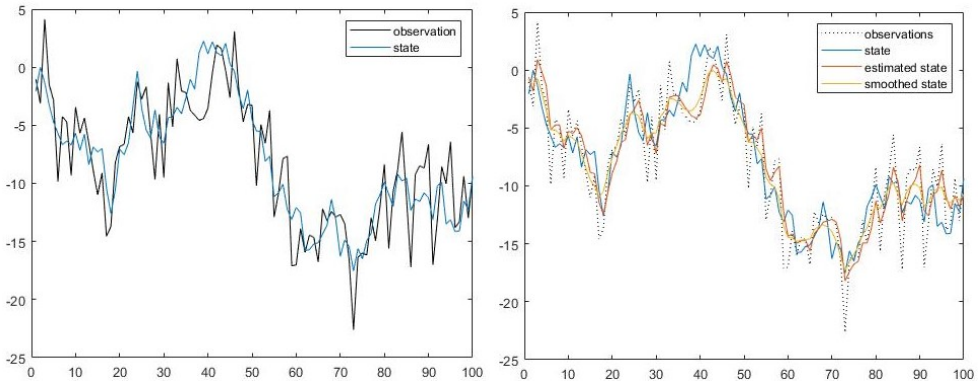
We began with the local level model, a one-dimensional model with trend component denoted by α_t and idiosyncratic term ε_t:

$$\begin{aligned} y_t &= \alpha_t + \varepsilon_t, & \varepsilon_t &\sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right) \\ \alpha_t &= \alpha_{t-1} + \eta_t, & \eta_t &\sim \mathcal{N}\left(0, \sigma_\eta^2\right). \end{aligned}$$

Although his very simple form, this model allows analysis of real problem thus it is not just an artificial special case (as highlighted by Durbin and Koopman [6]). We set initial conditions and parameters of the model as in the table below, then we generated a time series using model recursions in a Monte-Carlo simulation.

Parameters	Set values	Estimated values	t-statistics
T	100	/	/
a_1	0	/	/
P_1	10	/	/
σ_ε^2	8	6.9	4.71
σ_η^2	3	2.5	2.36

Thanks to model specification, all recursions were really simple: Kalman filter, state smoothing and simulation smoothing ones. Estimate emerged to be really good, as we may deduce from t-statistics in the table above and graphs in Figure 2.1. Thanks to this simple model we actually took confidence with the 'estimate philosophy'.



(a) Monte-Carlo simulation of observations and related states (b) Estimation and smoothing results

Figure 2.1: Local level model simulation and estimate.

Afterwards, we developed an univariate model with trend, slope, seasonal and cycle components and idiosyncratic term:

$$y_t = \mu_t + \sum_{j=1}^{\lfloor s/2 \rfloor} \gamma_{j,t} + c_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right) \tag{2.1}$$

where

$$\begin{aligned}
\mu_{t+1} &= \mu_t + \nu_t + \xi_t, & \xi_t &\sim \mathcal{N}\left(0, \sigma_\xi^2\right) \\
\nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim \mathcal{N}\left(0, \sigma_\zeta^2\right) \\
\gamma_{j,t+1} &= \gamma_{j,t} \cos\lambda_j + \gamma_{j,t}^* \sin\lambda_j + \omega_{j,t}, & \omega_{j,t} &\sim \mathcal{N}\left(0, \sigma_\omega^2\right) \\
\gamma_{j,t+1}^* &= -\gamma_{j,t} \sin\lambda_j + \gamma_{j,t}^* \cos\lambda_j + \omega_{j,t}^*, & \omega_{j,t}^* &\sim \mathcal{N}\left(0, \sigma_\omega^2\right) \\
c_{t+1} &= \rho\left(c_t \cos\lambda_c + c_t^* \sin\lambda_c\right) + \bar{\omega}_t, & \bar{\omega}_t &\sim \mathcal{N}\left(0, \sigma_{\bar{\omega}}^2\right) \\
c_{t+1}^* &= \rho\left(-c_t \sin\lambda_c + c_t^* \cos\lambda_c\right) + \bar{\omega}_t^*, & \bar{\omega}_t^* &\sim \mathcal{N}\left(0, \sigma_{\bar{\omega}}^2\right).
\end{aligned} \tag{2.2}$$

An overview of the work done on this model will be now provided; then details will be specified and graphs showed, since further models followed precisely from this. For the univariate model, as in the previous one, we set initial conditions and parameters besides of model dimensions, then we drew a sample of data and we estimated parameters using Kalman filter recursions and log-likelihood maximisation. We compared simulated components and estimated ones; we tested the results with quantiles analysis of skewness, kurtosis, homoscedasticity and serial correlation, as well as t-statistics. We computed state smoothing and simulation smoothing.

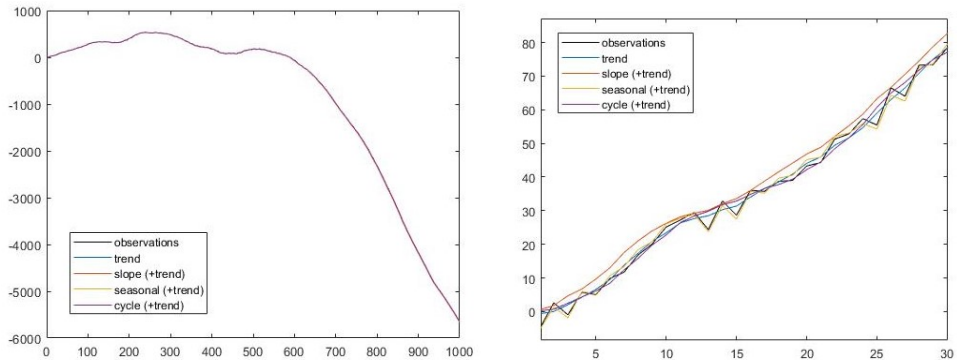
We now give some details. In the following tables, we reported all model dimensions, settings and parameters with estimates.

Dimensions	Set values	Initial conds	Set values
T	1000	a_1	$(0\ 0\ \ldots\ 0)'$ $[m \times 1$ vect$]$
s	12	P_1	I_m
m	$s + 3 = 15$		

Parameters	Set values	Estimated values	t-statistics
σ^2_ε	0.16	0.1496	8.38
σ^2_ξ	0.16	0.1743	3.11
σ^2_ζ	0.25	0.2108	10.7
σ^2_ω	0	3.3e-6	16.7
$\sigma^2_{\bar{\omega}}$	0.02	0.0274	11.8
λ_c	0.5	0.5014	256

Seasonality was fixed at $s = 12$, with the purpose of repeating Li et al.'s [10] and Petrova et al.'s [13] works; that means we considered monthly data and consequentely state vector lenght was 15. Observation error variance of seasonal component was fixed at 0 when generating observations but it was not when parameters shall be estimated, in other words it was a parameter to be estimated indeed. Cycle damping factor ρ was set at value 1 and did not required estimation. It should be clear that autoregressive component was not included yet. With the aim of maximising log-likelihood, MatLab function `fminunc` (*f-min-unconditional*) was employed: instead of maximising $\log L$, we minimised $-\log L$, and initial point was set arbitrarily as a vector of ones. To ensure positivity of estimated parameters, we introduced exponential link functions: $\sigma^2_\bullet = e^{x_\bullet}$, where $x_\bullet$ were the parameters estimated with `fminunc`; also, $\lambda_c = e^{x_\lambda}$ and x_λ was actually estimated.

The drawn sample is showed in Figure 2.2 with a focus in order to appreciate latent components decomposition.

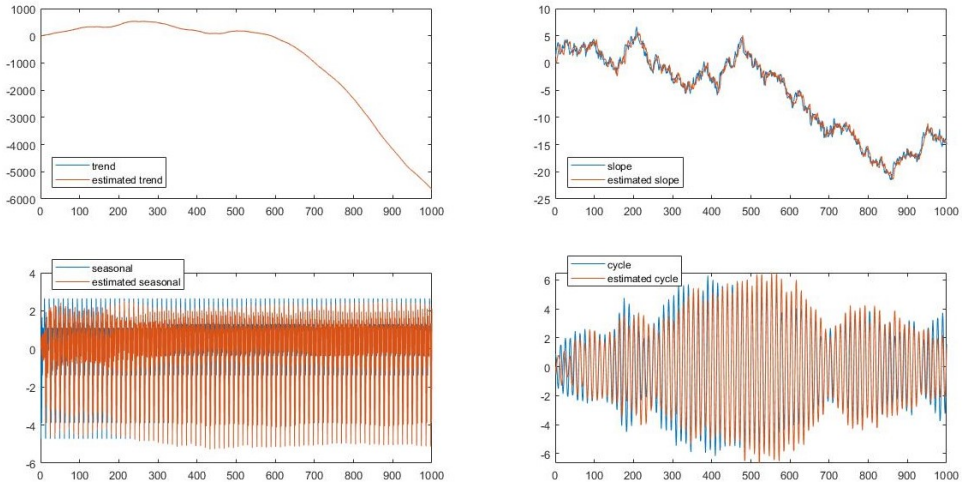


(a) Whole data sample (b) Focus

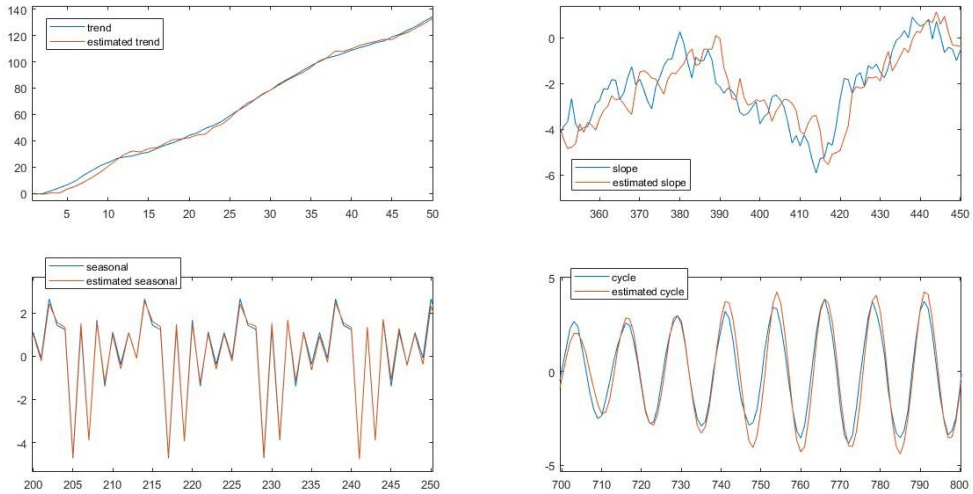
Figure 2.2: Univariate model: Monte-Carlo simulation of observations and related latent components.

Concerning results, we refer to Figure 2.3. We could observe that the estimated seasonal component does not seem to be deterministic; we may blame estimation of its observation error variance for this, although we believed more likely it might be a matter of learning: we hypothesized the model needs some iterations of Kalman filter recursions to gain the right

deterministic seasonal pattern. Other component estimation is really good.



(a) Whole data sample



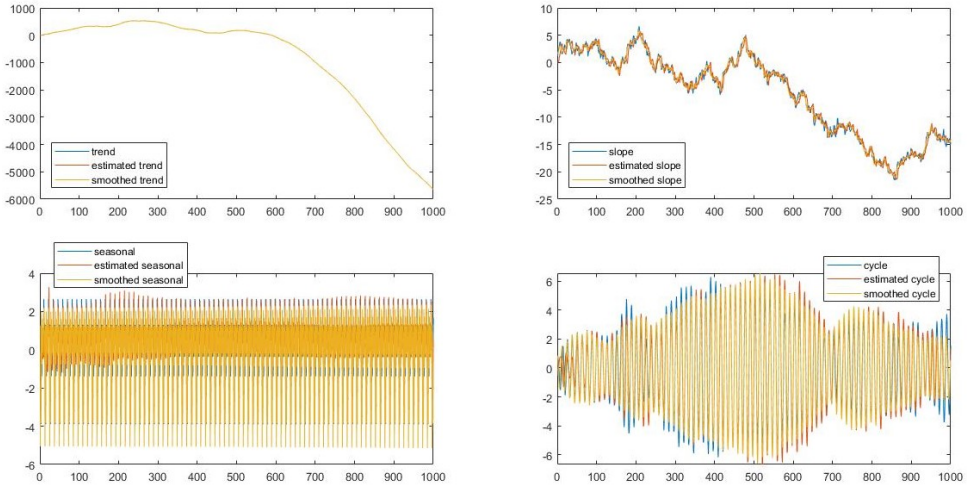
(b) Focus

Figure 2.3: Univariate model: comparison between simulated and estimated components.

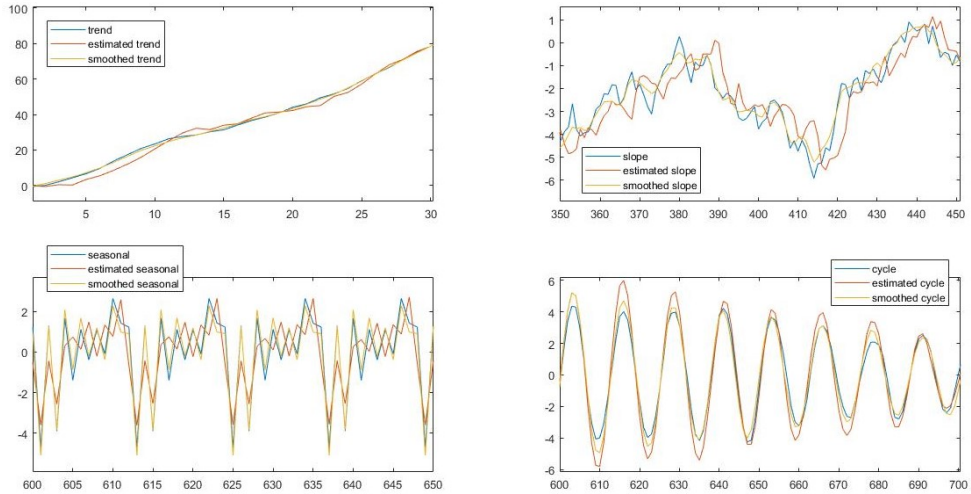
According to Figure 2.4, we remark that smoothed components are indeed smoother than the simply-estimated ones, as asserted in Chapter 1: those ones result a middle between simulated and estimated components, in particular seasonal smoothed component seems to be deterministic, even if with little lower magnitude than the simulated component.

Simulation smoothing also works as expected: unconditional sample is far from simulated observations while the conditional drawn one is consistent with them, as shown in Figure 2.5.

Models presented so far referred to Sections 1.1 to 1.7 of Chapter 1



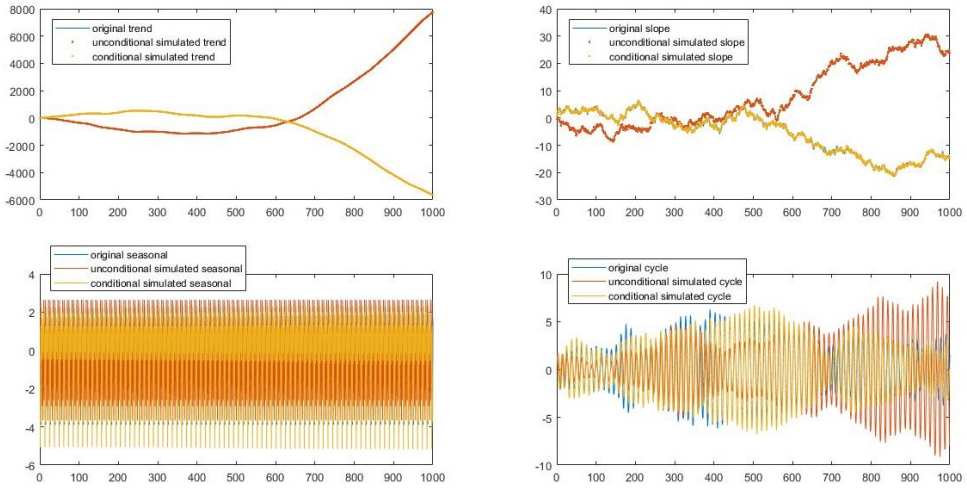
(a) Whole data sample



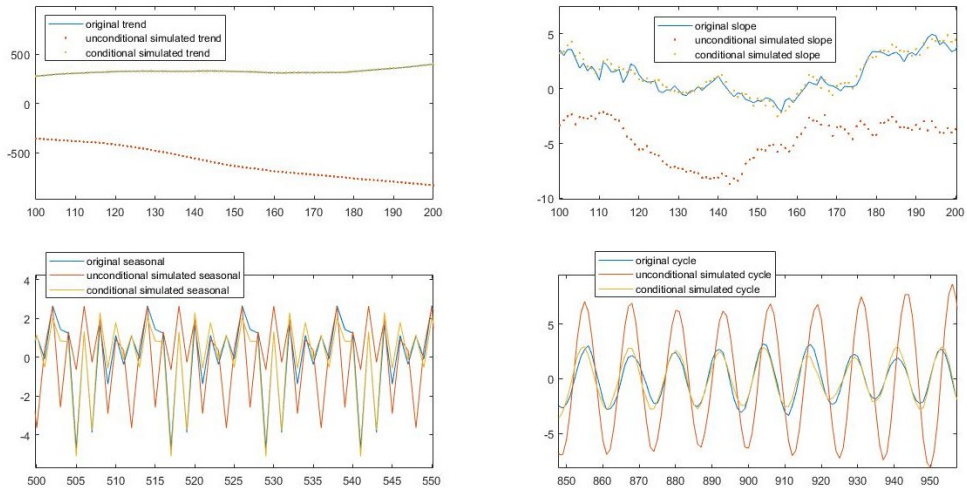
(b) Focus

Figure 2.4: Univariate model: state smoothing, comparison between simulated, estimated and smoothed components.

(without autoregressive component). We concluded Monte-Carlo analysis referring to Section 1.8: we introduced the exact initial recursions to both the Kalman filter and the state smoothing. Obviously, in this case initial mean vector a_1 and initial variance matrix P_1 were no more known parameters of the auxiliary functions we implemented, as they were in previous models. It is remarkable that d, number of needed iterations to initialise the Kalman filter, is approximately equal to s, the seasonality; intuitively, it makes sense: the model requires about a whole seasonality to learn the seasonal pattern. In our Monte-Carlo simulation, d results being 15. We also remark that estimated components vectors and matrices lack of elements at first d-th indices because of exact initialisation technique, as it can be seen in Figures 2.6 and 2.7; moreover, graphs show



(a) Whole data sample



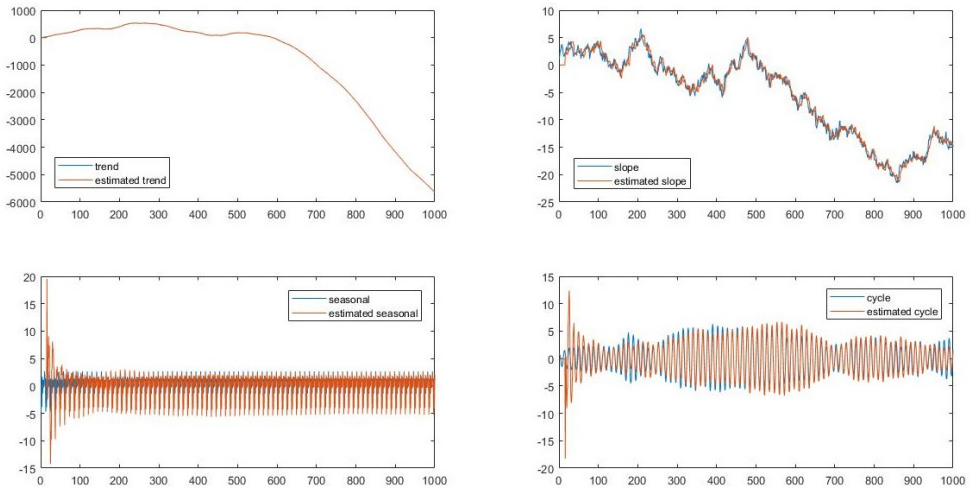
(b) Focus

Figure 2.5: Univariate model: simulation smoothing, comparison between unconditional and conditional drawn samples.

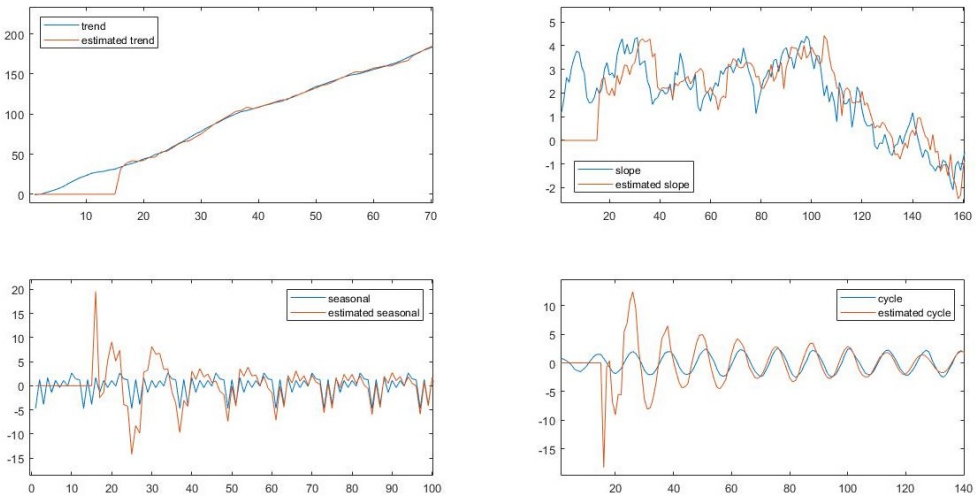
that some iterations over the "exact ones" are still needed to obtain a good components estimation.

2.2 Models on real data: Mills' model

In order to test our model specification, we tried to replicate Mills' work [12]. In that paper the author has monthly global temperature data from 1850 to 2007 and sets a model with trend, deterministic seasonal, cycle and autoregressive components and an idiosyncratic term, exactly like the model with latent components explained in Section 1.2 except for the slope component which Mills does not include. New elements



(a) Whole data sample

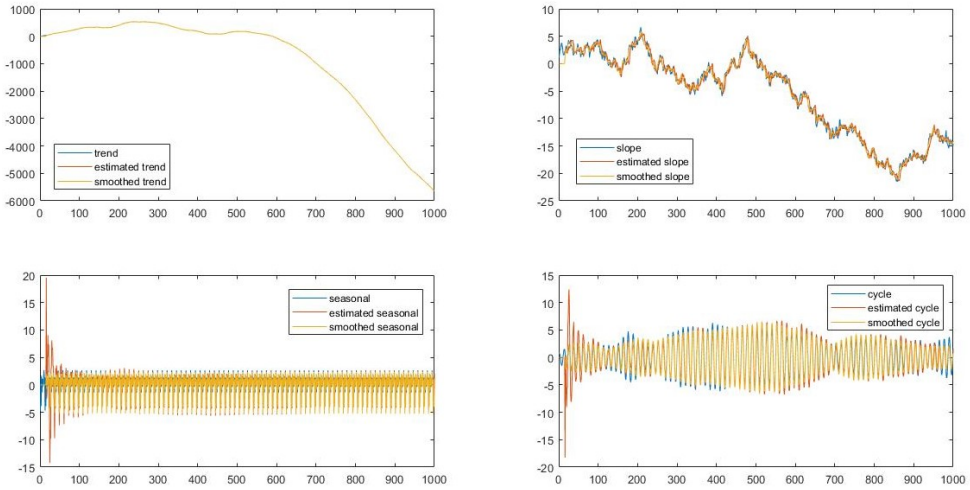


(b) Focus

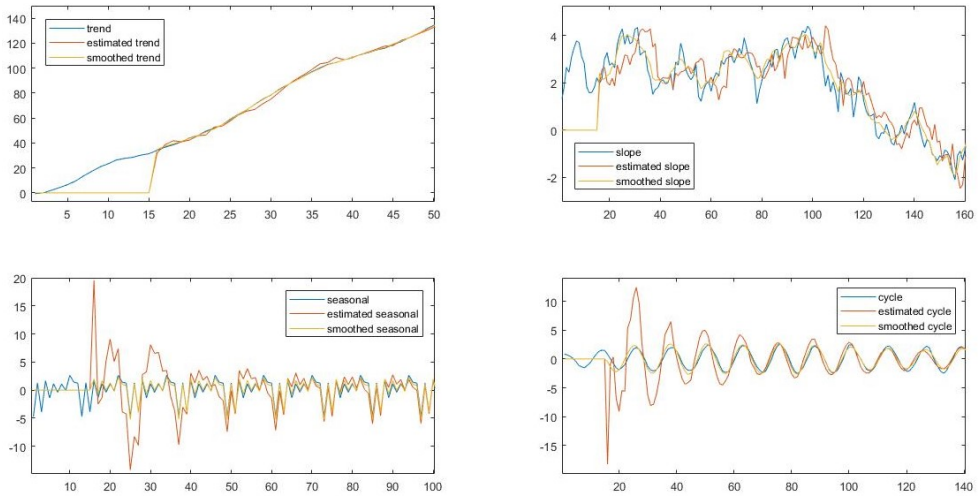
Figure 2.6: Univariate model with exact initial technique: comparison between simulated and estimated components.

comparing with our previous models presented so far are AR component and cycle damping factor ρ as a parameter to be estimated; the AR component is second-ordered with parameters θ_1 and θ_2 implemented as zeros of the T_φ matrix characteristic polynomial. One of the reasons why we referred to that work is the great similarity between our and Mills' model specifications, expecially regarding seasonal and cycle components.

Commonalities between models presented in this and next Sections are the great number of explorations required to fit models to data, which include adding and removing components, and the tough dependency of the estimate goodness from the initial point set for the `fminunc` MatLab function. Moreover, in order to increase computing efficiency we implemented, for each model, two different versions of the auxiliary function



(a) Whole data sample



(b) Focus

Figure 2.7: Univariate model with exact initial technique: state smoothing, comparison between simulated, estimated and smoothed components.

employed in log-likelihood maximisation: one of them computes vectors and matrices to calculate $-\log L$ without saving values, and this is the target function of `fminunc`, while the other saves computed values to allow model miss-specification tests and components plotting; the first is evaluated hundreds of times, the latter just one. For these models on real data, we limited numeric explorations to parameters estimate, without proceeding to state smoothing and simulation smoothing.

To replicate Mills' work, we firstly downloaded the same data the author used, which are available from the Hadley Centre for Climate Prediction and Research [2] (see Figure 2.8 for a plot). Then we started up building a model with trend, deterministic seasonal and cycle com-

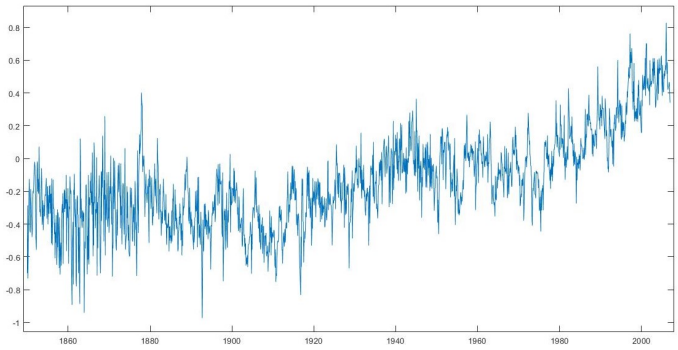


Figure 2.8: Mills' data: monthly average global temperatures in years 1850-2007 scaled of average computed from years 1960-1990 data.

ponents and idiosyncratic term, like the univariate model of previous Section (see equations (2.1) and (2.2)) but excluding slope component. Further changes to that model were different link function for λ_c, so that cycle period was forced to be major of $s = 12$, and introduction of ρ as a parameter required to be in $(0, 1)$ interval; explicitly, we set

$$\lambda_c = \frac{2\pi}{s + e^{x_\lambda}} \quad \text{and} \quad \rho = \frac{e^{x_\rho}}{1 + e^{x_\rho}}$$

where x_λ and x_ρ were the actual parameters of the model. In log-likelihood maximisation we used Mills' estimated parameters as initial point. The biggest challenge regarding implementation has been initialisation of exact initial procedure: matrix $P_{\infty,1}$ took trend and seasonal components, while matrix $P_{*,1}$ was related to cycle component and required some extra calculations. As explained in Section 1.8, by definition $P_{*,1} = R_0 Q_0 R_0'$ where R_0 is a selection matrix for non-stationary components of the model and Q_0 is the variance matrix of state vector. Since $\alpha_{t+1} = T\alpha_t + R\eta_t$, in order to compute Q_0 we had to solve the equation $Q_0 = TQ_0T' + RQR'$ where T, Q and R are model matrices specified in Section 1.2.5. As illustrated in Durbin and Koopman [6], it can be shown that a solution to above equation can be obtained solving the linear equation $(I_{m^2} - T \otimes T)\, vec\,(Q_0) = vec\,(RR')$ for Q_0, where $vec\,(M)$ is the stacked columns of matrix M and $T \otimes T$ is the Kronecker product of matrices T, that is

$$T \otimes T = \begin{pmatrix} t_{11}T & \dots & t_{1m}T \\ t_{21}T & \dots & t_{2m}T \\ \vdots & \ddots & \vdots \\ t_{m1}T & \dots & t_{mm}T \end{pmatrix}$$

with t_{ij} denoting the (i, j) element of matrix T. Obviously, this calculation bothers only part of matrices related to non-stationary components, specifically cycle component; consequently, instead of T, Q and R we considered T_c, Q_c and R_c and computed $Q_{0,c}$.

Now results deserve some attention: referring to Figure 2.9, trend component does not seem as smooth as in Mills' work, as it has lots of little fluctuations; seasonal component does not seem deterministic at all, it has big variations in magnitude that decrease gradually and stabilise just near the end of the sample data; cycle component seems to be very noisy. Aiming to smooth trend estimation and improve cycle estimation, we added a slope component but the result was worst cycle estimation, which was under-estimated in magnitude, and insignificant values for slope, that we soon excluded.

Then we added a second-order autoregressive component and estimation was very closer to Mills' result (as could be seen in Figure 2.10), although trend still showed unexpected fluctuations and seasonal pattern shown by Mills was reached just at the end of the data sample, like in the previous model specification. On the other hand, cycle component and AR component plus idiosyncratic term were reconstructed perfectly. It is remarkable that second-order AR was actually a first-order component because parameter θ_2 was estimated indistinguishable from zero; thus we explored the case with AR component of first order and results were exactly the same. From the implementation point of view, autoregressive component is stationary therefore it contribute to $P_{*,1}$ matrix with analogous calculations to previously presented ones for cycle component.

The fact that Mills analyses autoregressive component together with idiosyncratic term led us to reflect on the reason behind this choice; as a result we introduced in our model an idiosyncratic autoregressive term instead of an autoregressive component plus an independent and identically normally distributed idiosyncratic term:

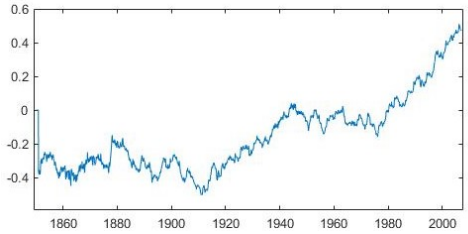
$$y_t = \mu_t + \sum_{j=1}^{\lfloor s/2 \rfloor} \gamma_{j,t} + c_t + \varphi_t \tag{2.3}$$

where

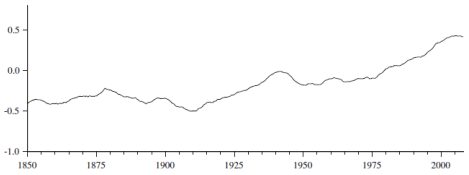
$$\begin{aligned}
\mu_{t+1} &= \mu_t + \nu_t + \xi_t, & \xi_t &\sim \mathcal{N}\left(0, \sigma_\xi^2\right) \\
\gamma_{j,t+1} &= \gamma_{j,t} \cos \lambda_j + \gamma_{j,t}^* \sin \lambda_j & & \\
\gamma_{j,t+1}^* &= -\gamma_{j,t} \sin \lambda_j + \gamma_{j,t}^* \cos \lambda_j & & \\
c_{t+1} &= \rho \left(c_t \cos \lambda_c + c_t^* \sin \lambda_c\right) + \bar{\omega}_t, & \bar{\omega}_t &\sim \mathcal{N}\left(0, \sigma_{\bar{\omega}}^2\right) \\
c_{t+1}^* &= \rho \left(-c_t \sin \lambda_c + c_t^* \cos \lambda_c\right) + \bar{\omega}_t^*, & \bar{\omega}_t^* &\sim \mathcal{N}\left(0, \sigma_{\bar{\omega}}^2\right) \\
\varphi_{t+1} &= \theta \varphi_t + \tau_t, & \tau_t &\sim \mathcal{N}\left(0, \sigma_\varphi^2\right).
\end{aligned}$$

Obviously, we expected observation error variance of AR term, σ_φ^2, to assimilate observation error variance of idiosyncratic term σ_ε^2, and this happened, as proved by data in the following table.

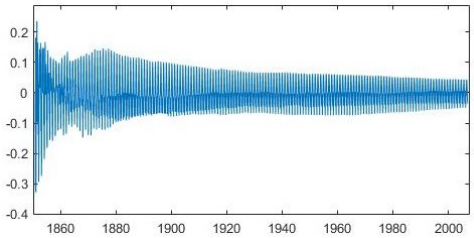
AR comp. + idiosyncratic		AR idiosyncratic	
σ_φ^2	0.005124	σ_φ^2	0.010528
σ_ε^2	0.005850		



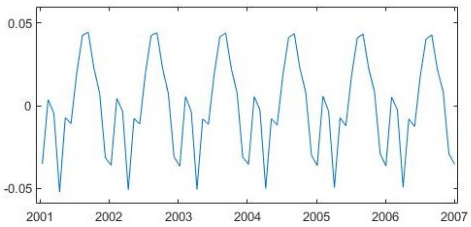
(a) Our trend estimate



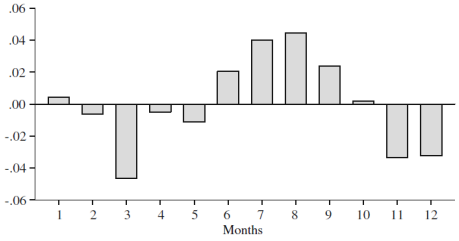
(b) Mills' trend estimate



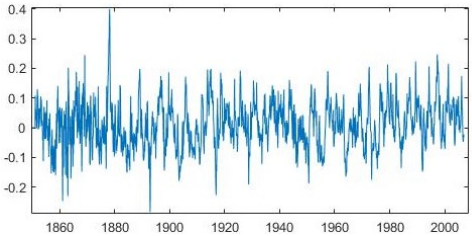
(c) Our seasonal estimate



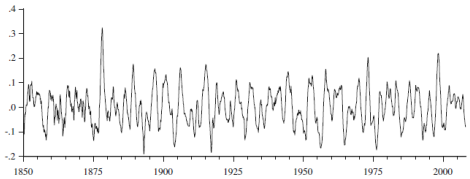
(d) Our seasonal estimate, focus



(e) Mills' seasonal estimate, pattern

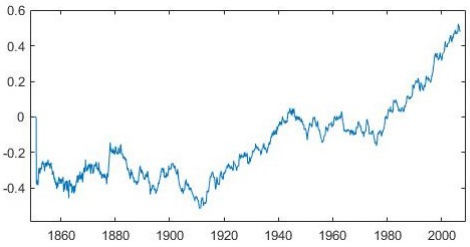


(f) Our cycle estimate

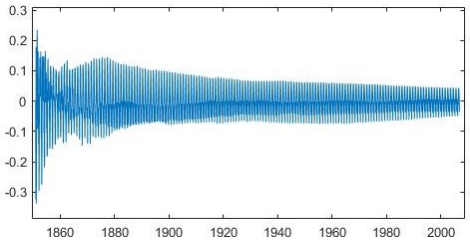


(g) Mills' cycle estimate

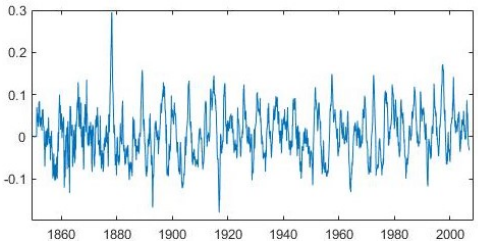
Figure 2.9: Mills' model without AR component: comparison between our components estimate and Mills' one.



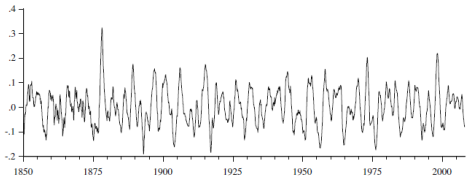
(a) Our trend estimate



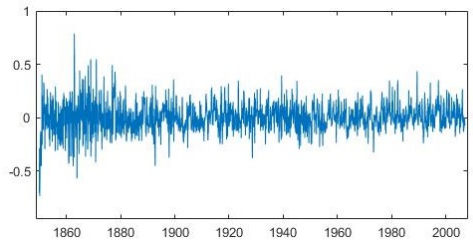
(b) Our seasonal estimate



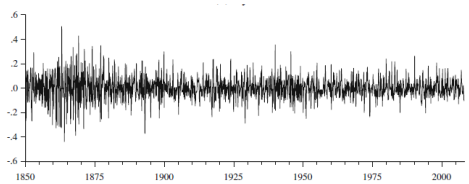
(c) Our cycle estimate



(d) Mills' cycle estimate



(e) Our AR plus idiosyncratic estimate



(f) Mills' AR plus idiosyncratic estimate

Figure 2.10: Mills' model with AR component: comparison between our components estimate and Mills' one.

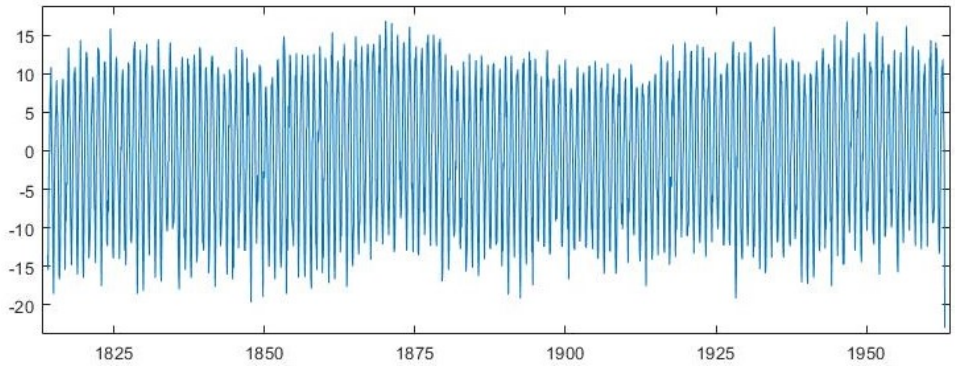
Other components were estimated like before, particularly seasonal one still decreases in magnitude along the process, but we became confindent with the idea it was just a matter of model training, as introduced in the previous Section for univariate model.

Mills' results were then replicated quite accurately and we gained decent confidence with monthly data and models, thus we went over.

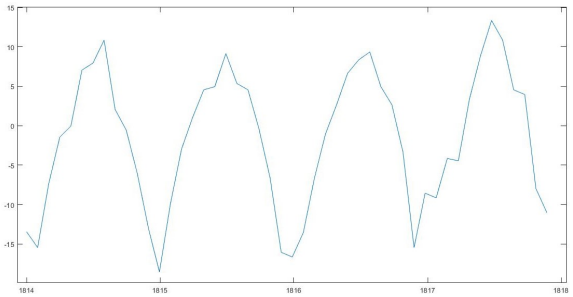
2.3 Models on real data: Bologna station

We downloaded daily maximum temperature data for the city of Bologna from the site of European Climate Assessment and Dataset [9]. Since the time serie was very long, we left choice to set number of years to be taken into account when collecting data, later we have had explorations varying also this parameter.

Aiming to fit the last model presented above to these new data, we aggregated them on a monthly basis: all data referred to the same solar month have been merged in their maximum. We also scaled data with respect to their average value, in order to obtain results as close as possible to Mills' ones. In Figure 2.11 manipulated data are plotted.



(a) 150 years data sample



(b) Focus

Figure 2.11: Bologna station data: monthly maximum temperatures in years 1814-1963 scaled of their average value.

About estimate, Figure 2.12 shows results even better than Mills' model ones expecially for seasonal component, which is nearly deterministic since first iterations. Obvious difference concerns the scale of estimated components but this is due to difference in data scale itself.

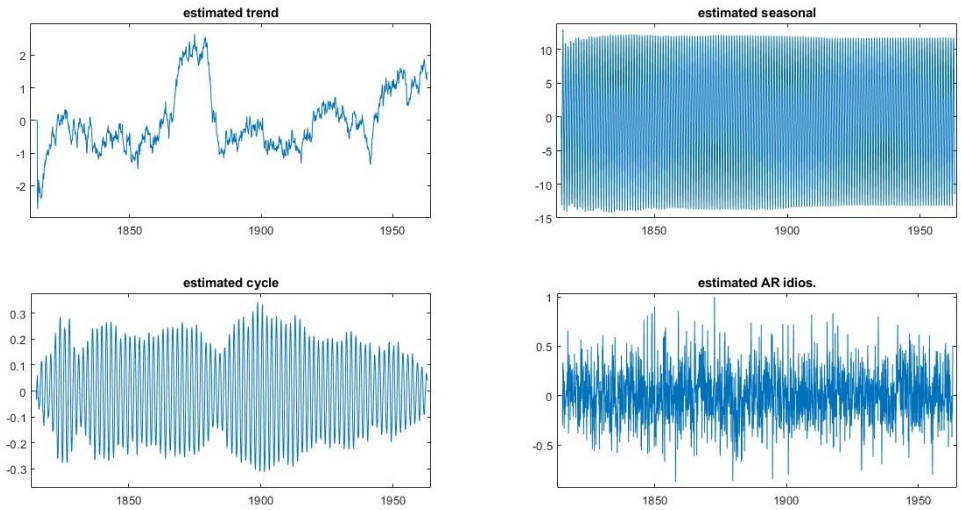
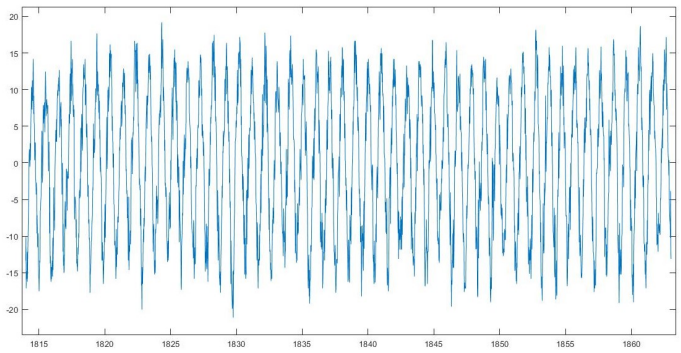


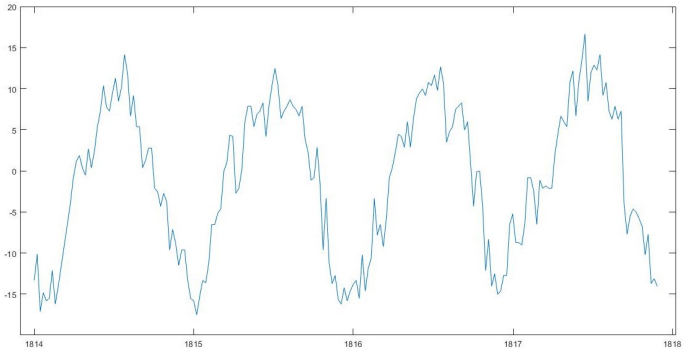
Figure 2.12: Monthly model for Bologna data: estimated components.

Afterwards, we turned our attention to weekly data models. For this purpose data manipulation was required: we firstly excluded data relative to February 29^{th}, then we aggregated data on a weekly basis taking the maximum; since the number of weeks in a year is not an integer, we excluded also December 31^{st} data. Resulting sample is plotted in Figure 2.13.

Remembering previous reasonings about AR component in monthly data models, we wondered what autoregressivity order we should set in a weekly data model, thinking fourth and eighth the most plausible; we concluded implementing model with parametric AR order, with the purpose of testing and choosing the best one. Moreover, since number of AR parameters increased remarkably, we applyed a step function to reduce it: thus the first parameter is called θ_1 and refers to the first step behind in the past, while others (3 or 7 according to the set AR order) are all indicated as θ_2. As a further condition on θ_1 and θ_2 was required to ensure AR component stationarity, we decided to employ `fmincon` (*f-min-conditional*) MatLab function in log-likelihood maximisation to be able to constraint more easily model parameters, which consequently did not need link functions any more. We tested different initial points (firstly set with estimated monthly parameters values), fourth and eighth order of autoregressivity and even different length of data sample. Comparing log-likelihood values, the model suggested AR(8) to be the best choice and gave good (in the sense of t-statistics) parameters estimate which was consistent with Mills' monthly one. Estimated components are shown in



(a) 50 years data sample



(b) Focus

Figure 2.13: Bologna station data: weekly maximum temperatures in years 1814-1863 scaled of their average value.

Figure 2.14, while Figure 2.15 compare observations and signal, that is the sum of estimated components except for idiosyncratic term, in other words the reconstructed signal of observation without noise.

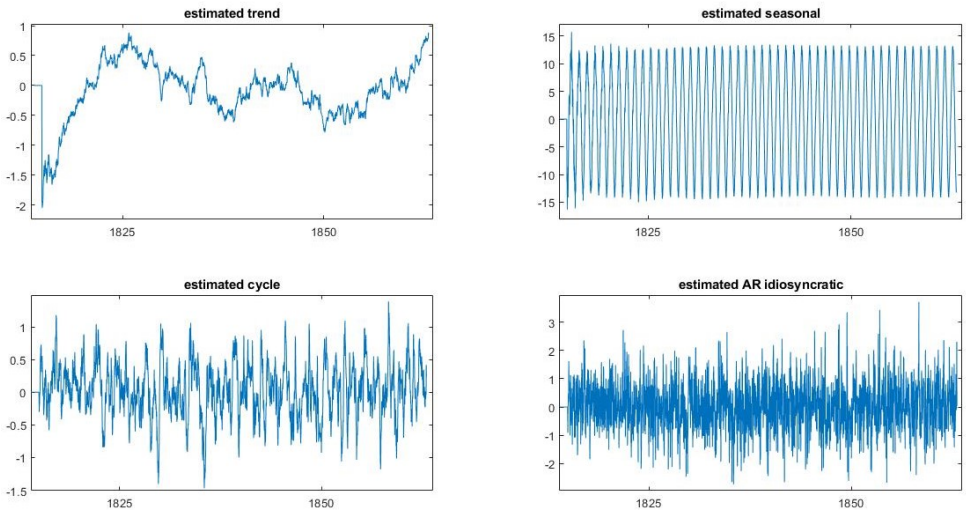
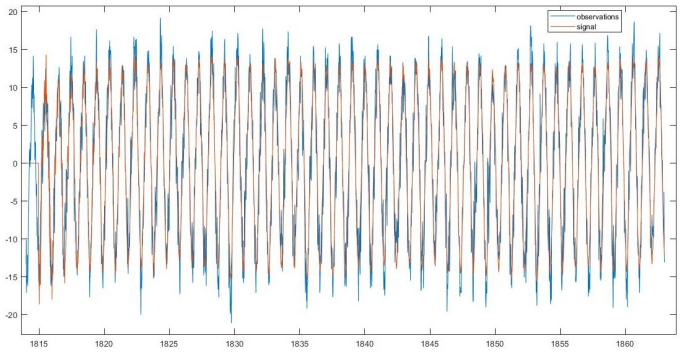
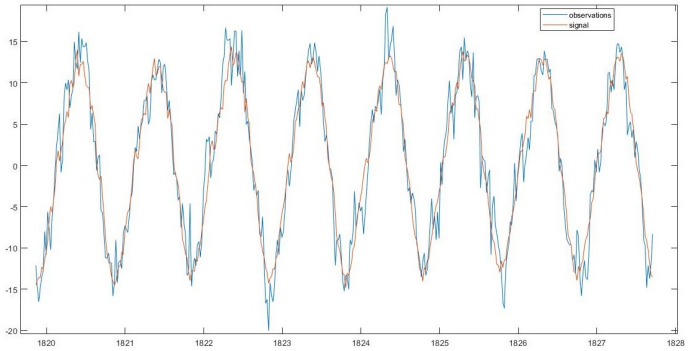


Figure 2.14: Weekly model for Bologna data: estimated components.



(a) 50 years data sample



(b) Focus

Figure 2.15: Bologna station data: comparison between observations and signal.

We lacked of reporting miss-specification tests results for previous models, but now we should illustrate them to conclude our analysis. Figure 2.16 displays a plot of the standardised one-step ahead prediction error, e in Chapter 1 notation; we could observe that the cloud of points is well-distributed and only very few of them lays out of the highlighted region, which means e validates the hypothesis of normality. This hypothesis is also confirmed by the QQ-plot in Figure 2.17: comparing quantiles of e and of a standand normal we could deduce their great similarity in distribution. QQ-plot indicates other features of the standardised prediction error, too: firstly, the simmetry in distribution, which is also supported by the Skewness value of 0.001998, verifing distributional hypothesis presented in Section 1.5; secondly, a probability excess on the tails, supported by the Kurtosis value of 3.35, which slightly rejects the distributional hypotesis. Other specification tests are available. A p-value of 0.1095 allows accepting the hypothesis of homoscedasticity of e and a p-value of 0.0982 in Box-Ljung test confirms what claimed by correlogram of e, shown in Figure 2.18: values are mostly within the region of non-significance and, above all, no residual correlation is detected.

We were satisfied of obtained results, so we considered appropriate to

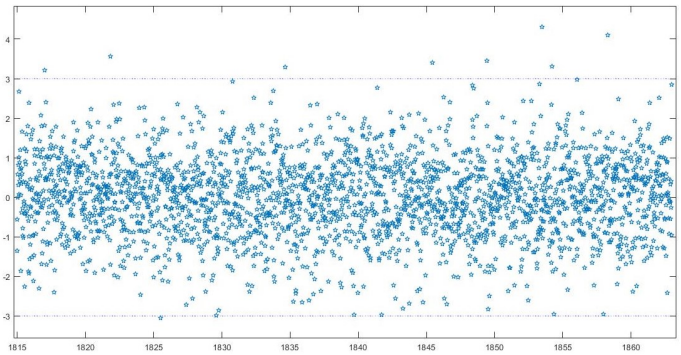


Figure 2.16: Weekly model for Bologna data: standardised one-step ahead prediction error.

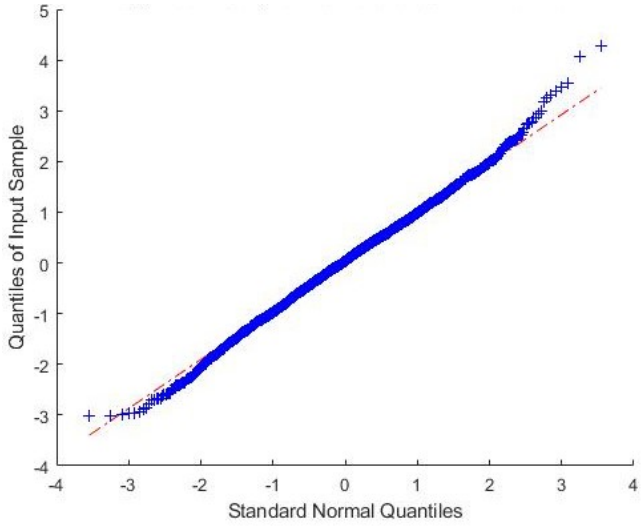


Figure 2.17: Weekly model for Bologna data: QQ-plot of standardised prediction error versus standard normal.

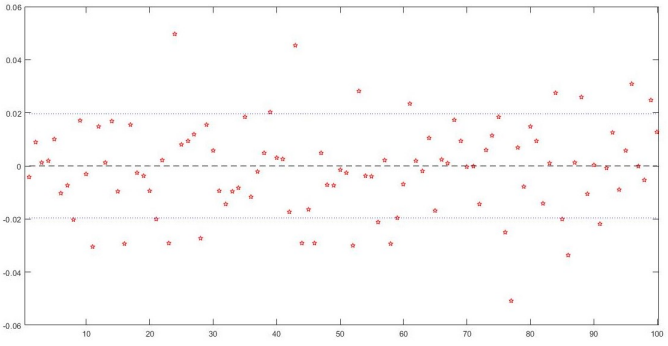


Figure 2.18: Weekly model for Bologna data: correlogram for standardised prediction error.

proceed with the next step of our work.

Chapter 3

Dynamic factor models

Thanks to the state space model we are able to compute prediction errors for each time series downloaded from the dataset [9], then we wonder if there may be residual information in such errors and how to extract it. Factor analysis is a dimension reduction technique summarising the sources of variation among variables; typically, a few factors are sufficient to explain correlations among high-dimensional panel of time series. As previous models, parameters estimation would be required, therefore we will present different methods and discuss the best one for our purpose. The second chapter in Doz and Fuleky [3] is referred for material and further information.

3.1 Factor models

As introduced, in a factor model correlations among N variables, $x_1, \ldots, x_N$, for which T observations are available, are assumed to be entirely due to a few, $r < N$, latent unobservable variables, called *factors*. The link between observable variables and factors is assumed to be linear, thus observations could be decomposed as

$$x_t = \Lambda f_t + e_t$$

where x_t denotes the observed $N \times 1$ vector of time series at time t, Λ is an $N \times r$ loading matrix of full column rank (otherwise fewer factors would suffice), f_t is the $r \times 1$ vector of common latent factors and e_t is the $N \times 1$ idiosyncratic vector. It is remarkable that each observation in x_t is thus decomposed into the sum of two mutually orthogonal unobserved components: the common component and the idiosyncratic component; while the factors drive the correlation between observations belonging to different time series at any time lag $1 \leq t \leq T$, the idiosyncratic term arises from features that are specific to an individual time series. As in most of the factor models literature, we consider stationary variables,

particularly we assume stationarity of f_t and e_t processes. Further assumptions on unobserved component result in factor models of different types.

3.1.1 Exact factor models

Exact factor models assume idiosyncratic terms not to be correlated at any leads and lags, that is e_s and e_t are mutually orthogonal for all vectorial components and any s and t. Consequently, all correlation among observations is solely due to the common factors and, analytically, the covariance matrix of e_t is diagonal. Moreover, all the components of f_t and e_t are assumed to be white noises, for all t.

So far static factor models were presented, but there exist also dynamic factor models which involve s time lags of the factors in the observation equation of x_t:

$$x_t = \Lambda_0 f_t + \Lambda_1 f_{t-1} + \ldots + \Lambda_s f_{t-s} + e_t.$$

In this kind of models, f_t and e_t are allowed to be autocorrelated dynamic processes involving *iid* errors. The dimension of f_t is therefore denoted by q, which is called *number of dynamic factors*. The dynamic model admits a static representation

$$x_t = \Lambda F_t + e_t$$

where $F_t = \left(f_t', f_{t-1}', \ldots, f_{t-s}'\right)'$ is a $r = q\,(s+1)$ dimensional vector of static factors and $\Lambda = (\Lambda_0, \Lambda_1, \ldots, \Lambda_s)$ is a $N \times r$ matrix of loading coefficients.

Exact dynamic factor models are usually employed with small cross-sectional dimension of the dataset, $N \ll T$.

3.1.2 Approximate factor models

Exact factor models rely on a very strict assumption of no cross-correlation among idiosyncratic components. In contrast, approximate factor models allow the idiosyncratic components to be mildly cross-correlated; moreover, cross-sectional dimension N is allowed to go to infinity.

Let $x_t^N = (x_{1t}, \ldots, x_{Nt})'$ denote the vector containing the t-th observation of the first N variables as $N \to \infty$, and let $\Sigma_N = cov\left(x_t^N\right)$ be the covariance matrix of x_t^N. Denoting by $\lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_n(A)$ the ordered eigenvalues of any symmetric matrix A with size $n \times n$, the assumptions underlying approximate factor models are the following:

- $\sup_N \lambda_r(\Sigma_N) = \infty$
- $\sup_N \lambda_{r+1}(\Sigma_N) < \infty$

- $\inf_N \lambda_N(\Sigma_N) > 0$

Respectively, these assumptions mean that the r largest eigenvalues of Σ_N are diverging while the remaining eigenvalues are bounded, and Σ_N does not approach to singularity. Under these assumptions, Chamberlain and Rothschild [1] proved there exists a unique decomposition

$$\Sigma_N = \Lambda_N \Lambda_N' + \Psi_N$$

where Λ_N is a sequence of nested $N \times r$ matrices with rank r and $\lambda_1(\Lambda_N \Lambda_N') \to \infty$, $\forall i = 1, \ldots, r$, and $\lambda_1(\Psi_N) < \infty$.

Alternatively, x_t^N can be decomposed using a pair of mutually orthogonal random vector process f_t and e_t^N:

$$x_t^N = \Lambda_N f_t + e_t^N$$

where unique identification of the model is ensured by imposing $cov(f_t) = I_r$ and $cov\left(e_t^N\right) = \Psi$.

Similarly to exact dynamic factor models, approximate dynamic factor models rely on an equation linking the observable series to several lags of the common factors, whose static representation is

$$x_t^N = \Lambda_N F_t + e_t^N.$$

(For semplicity, in the following we will drop the sub- and superscript N in Λ_N, x_t^N and e_t^N, assuming that in the exact factor models the number of series taken under account is finite, while in approximate factor models $N \to \infty$.)

3.2 Estimation methods

According to the type of factor model and the cross-sectional dimension of the time series panel, different methods are available in order to estimate the model. Consistency has been demonstrated for all methods.

3.2.1 Maximum likelihood estimate of small factor models

The static exact factor model is generally estimated by maximum likelihood criterium under the assumption that (f_t) and (e_t) are two orthogonal *iid* Gaussian processes. Unique identification of the model requires to impose some restrictions: first of all, since idiosyncratic components are set to be mutually orthogonal processes, their variance matrix $Var(e_t)$ should be diagonal; secondly, the variance of the factors is set to be the identity matrix, $Var(f_t) = I_r$.

For small values of N the number of parameters is small, therefore estimates can be obtained throught any numerical optimization procedure. Given the parameter estimates $\hat{\Lambda}$ and $\hat{\Psi}$, factors may be computed with the formula

$$\hat{f}_t = \left(\hat{\Lambda}'\hat{\Psi}^{-1}\hat{\Lambda}\right)^{-1}\hat{\Lambda}'\hat{\Psi}^{-1}x_t$$

which is the FGLS (Feasible Generalised Least Squares) estimator of f_t.

A dynamic exact factor model could also be estimated by maximum likelihood criterium under the assumption of Gaussian (f_t, e_t). In this case, factors are assumed to follow a vector autoregressive process, VAR(p), and the model can be easly cast in state space form using the companion form; more precisely, let consider the model

$$\begin{aligned} x_t &= \Lambda_0 f_t + \Lambda_1 f_{t-1} + \ldots + \Lambda_s f_{t-s} + e_t \\ f_t &= \Phi_1 f_{t-1} + \ldots + \Phi_p f_{t-p} + u_t \end{aligned}$$

where the coefficient matrices Φ_j, $j = 1, \ldots, p$ capture the dynamics of the factors. A commonly used identification restriction sets the variance of the innovations to the identity matrix, that is $cov\,(u_t) = I_r$, and additional restrictions are imposed on the factor loadings. The model written in state space form allows likelihood to be computed with the Kalman filter recursions; then, with parameter estimates $\hat{\theta}$, the Kalman smoother provides an approximation of f_t using information from all observations:

$$\hat{f}_{t|T} = E\left(f_t | x_1, \ldots, x_T, \hat{\theta}\right).$$

3.2.2 Principal component analysis of large approximate factor models

Chamberlain and Rothschild [1] suggested to use principal component analysis (PCA) to estimate approximate static factor models. Assuming that the number of factors r is known, PCA allows to simultaneously estimate the factors and the loading coefficients by solving the least squares problem

$$\min_{\Lambda, F} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(x_{it} - \lambda_i f_t\right)^2$$

where x_{it} is the i-th component of x_t and λ_i is the i-th row of Λ.

With the purpose of avoiding indeterminancy of factors and loadings, the following normalisation condition is imposed (when $T > N$):

$$\frac{\hat{\Lambda}'\hat{\Lambda}}{N} = I_r.$$

3.2.3 Two-step estimation of large approximate factor models

Doz, Giannone and Reichlin [5] proposed a two-step estimator that takes into account the dynamics of the factors. They consider an approximate dynamic factor model with VAR factors, and they allow idiosyncratic terms to be autocorrelated even if do not specify their dynamics:

$$x_t = \Lambda f_t + e_t$$
$$f_t = \Phi_1 f_{t-1} + \ldots + \Phi_p f_{t-p} + u_t$$

As said before, this model could easily be cast in state space form.

The first step of the estimation procedure starts obtaining preliminary estimators of loadings $\hat{\Lambda}$ and factors $\hat{f}_t$ by PCA. Particularly, let denote by d_j and v_j the j-th eigenvalue and relative unitary eigenvector respectively of the empirical covariance matrix of data, and define $D = diag(d_1, \ldots, d_r)$ and $V = (v_1, \ldots, v_r)$; then $\hat{\Lambda} = VD^{1/2}$ and $\hat{f}_t = D^{1/2}V'x_t$. Afterwards the idiosyncratic terms are estimated by $\hat{e}_t = x_t - \hat{\Lambda}\hat{f}_t$ and their variance is estimated by the associated empirical variance $\hat{\Psi}$. The estimated factors $\hat{f}_t$ are used in a VAR model to obtain the estimates $\hat{\Phi}_j$ for $j = 1, \ldots, p$; it is proved that maximum likelihood and ordinary least squares (OLS) give exactly the same results, therefore the latter estimate is chosen and coefficients $\hat{\Phi}_j$ are analytically known.

In the second step the model is cast in state space form with the variance of the common shocks set to identity matrix, that is $cov\,(u_t) = I_r$, and $cov\,(e_t)$ defined in diagonal shape as $\Psi = diag\left(\hat{\psi}_{11} \ldots \hat{\psi}_{nn}\right)$. Using the parameter estimates $\hat{\Lambda}$, $\hat{\Psi}$ and $\hat{\Phi}_j$, $j = 1, \ldots, p$ obtained in the first step of the procedure, one step of the Kalman smoother is then applied to the data; result is a new estimate of the factor $\hat{f}_{t|T} = E\left(f_t | x_1, \ldots, x_T, \hat{\theta}\right)$, where $\hat{\theta}$ is a vector containing the first step estimates of all parameters.

It is remarkable that assumptions imposed so far allow good estimation of the factors despite of multiplication by an orthogonal matrix, which means that factors are reconstructed unless the signs and a permutation of themself.

Summing up, in the first step parameters and factors are estimated throught PCA, then dynamics of the factors are estimated throught OLS from these preliminary estimates of the factors; thus numerical optimization is not required so far. In the second step common factors are re-estimated throught Kalman recursions, that have been discussed thoroughtly in previous Chapters.

Proof of the estimation consistency is central issue of the paper [5]; we will now summarise the results obtained by the authors and refer to their work for further information and proofs. Firstly, under the assumptions presented for approximate factor models, principal components give consistent estimators of the span of the common factors, and of associated

factor loadings, when both the cross-section and the sample size go to infinity. Under the same assumptions, the consistency of the autoregressive parameters estimates is also proven. Consequentely, authors could conclude that estimators obtained with the whole two-step procedure are consistent with the true common factors.

3.2.4 Quasi-maximum likelihood estimation of large approximate factor models

Doz, Giannone and Reichlin [4] also proposed to estimate a large approximate dynamic factor model by quasi-maximum likelihood (QML). As in the paper [5] discussed above, the quasi-likelihood is based on the assumption of mutually orthogonal *iid* Gaussian idiosyncratic terms and a Gaussian VAR model for the factors. The corresponding log-likelihood can be obtained from Kalman filter recursions for given values of the parameters, and the authors apply an EM algorithm to compute the maximum likelihood estimator.

The EM algorithm alternates an (E)xpectation step relying on a pass of the Kalman smoother for the current parameters values and a (M)aximisation step relying on multivariate regression; practically, iterations of the algorithm are equivalent to successive applications of the two-step procedure. As a consequence, calculations are feasible even when N is large thanks to the low computational cost of each iteration.

3.3 Collapsing technique

When the common factors f_t and the idiosyncratic terms e_t are Gaussian, the likelihood function of the dynamic factor model $x_t = \Lambda f_t + e_t$ can be easily evaluated by applying Kalman filter recursions, once the model is cast in state space form; and when they are not Gaussian, the Gaussian likelihood is treated as a quasi-likelihood. As a consequence, the Gaussian likelihood function may be numerically maximised in order to obtain maximum likelihood or quasi-maximum likelihood parameter estimates.

However, the high-dimensional panel of time series and the resulting large number of parameters could make such an approach infeasible: most of the computational problems are related to the inversion of the matrix F_t, which requires more and more computing time and, over all, compromises numerical precision. In the paper [8] Jungbacker and Koopman propose a solution to this problem splitting the observed time series into a low-dimensional vector series and a high-dimensional vector series, and proving that it suffices applying the Kalman filter to the low-dimensional set to obtain the evaluation of the likelihood function.

Let define $x_t^* = Ax_t$ for any non-singular $N \times N$ matrix A and $t = 1, \ldots, T$. Suppose matrix A is partitioned as

$$A = \begin{pmatrix} A^L \\ A^H \end{pmatrix}$$

where A^L and A^H are respectively $m \times N$ and $(N-m) \times N$ matrices, and $0 < m \leq p$ is the rank of matrix Λ. Then

$$x_t^* = \begin{pmatrix} x_t^L \\ x_t^H \end{pmatrix} \qquad \text{with} \quad x_t^L = A^L x_t \quad \text{and} \quad x_t^H = A^H x_t$$

where the observation vectors have dimensions $m \times 1$ and $(N-m) \times 1$ respectively.

The aim is to choose matrix A such that x_t^L and x_t^H are not correlated with each other and only x_t^L depends on f_t. More specifically, the model for x_t^* will be of the form

$$\begin{aligned} x_t^L &= A^L \Lambda f_t + e_t^L \\ x_t^H &= e_t^H \end{aligned} \tag{3.1}$$

where $e_t^L = A^L e_t$ and $e_t^H = A^H e_t$. Moreover, we have that

$$E\left(e_t^L\right) = 0, \quad E\left(e_t^H\right) = 0,$$

$$Var\left(e_t^L\right) = \Sigma^L, \quad Var\left(e_t^H\right) = \Sigma^H, \quad E\left(e_t^H e_t^{L\prime}\right) = 0$$

for $t = 1, \ldots, T$, where $\Sigma^L = A^L \Sigma_e A^{L\prime}$ and $\Sigma^H = A^H \Sigma_e A^{H\prime}$.

A suitable matrix A needs to fulfil the following assumptions:

- A is full rank
- $A^H \Sigma_e A^{L\prime} = 0$
- $Row\left(A^H\right) = Col\left(\Lambda\right)^{\perp}$

The first assumption prevents any loss of information as a result of the transformation Ax_t, the second ensures that e_t^L and e_t^H are uncorrelated, and the last implies that the latter equation in (3.1) does not depend on f_t. These assumptions define a closed form for A^L, which could be written as

$$A^L = \widetilde{\Lambda}' \Sigma_e^{-1}$$

where the columns of the $N \times m$ matrix $\widetilde{\Lambda}$ form a basis for the column space of Λ. When Λ is of full column rank and $p = m$, a typical decomposition is

$$\widetilde{\Lambda} = \Lambda \left(\Lambda' \Sigma_e^{-1} \Lambda\right)^{-1}.$$

Remarkably, a closed-form expression for A^H is generally not available, but the authors proved that matrix A^H and, consequentely, x_t^H are not required for any of the computations we are interested in.

Finally, the Kalman filter and smoothing could be applyed to the low-dimensional model

$$x_t^L = A^L \Lambda f_t + e_t^L$$

with $E\left(e_t^L\right) = 0$, $Var\left(e_t^L\right) = \Sigma^L$ and $t = 1, \ldots, T$, obtaining identical results and considerably shorter computing times of the case with recursions applied to the high-dimensional complete model.

Chapter 4

Numeric explorations

In Chapter 2 we implemented what discussed in previous Chapter 1. Analogously, we are now going to present numeric explorations referred to dynamic factor models just introduced in Chapter 3.

As previously, we will simulate data with Monte-Carlo generation to fit the model and acquire parameters sensitivity; nevertheless we refer to the next Chapter for the handing of real data time series.

4.1 Monte-Carlo analysis

We downloaded codes directly from Giannone's web site [7]: he and his collegues wrote a main script, a function `sim-mod` to simulate model and a function `dynFA` to estimate factors using PCA, 2-step method and QML.

From their codes, we developed our own modifing and adding features. All main scripts are structured as follow: setting of the model parameters, model simulation, parameter value estimation, assessment of the goodness-of-fit, running of misspecification tests, and graph drawing. We looked at the model

$$\begin{aligned} x_t &= \Lambda f_t + e_t, & e_t &\sim \mathcal{N}(0, \Psi) \\ f_t &= \Phi_1 f_{t-1} + \ldots + \Phi_p f_{t-p} + u_t, & u_t &\sim \mathcal{N}(0, Q) \end{aligned}$$

where x_t is a row vector of lenght N, Λ is a $N \times r$ matrix, f_t is $r \times 1$ and Φ_j is $r \times r$ for all $j = 1, \ldots, p$. Actually, we store observations x_t in a matrix X with size $T \times N$ and factors f_t in a $T \times r$ matrix F.

In all explorations we set $N = 100$ and $T = 1000$, and we gradually archived generality about matrices Φ_j, $j = 1, \ldots, p$, starting from $p = 1$ and $\Phi_1 = \alpha I$ and reaching $p < 7$ and Φ_j of general shape.

About model simulation, it is remarkable that requiring $cov(f_t) = I$ implies calculations to compute Q, so that

$$I = \Phi I \Phi' + Q$$

although 2-step method assumes $cov(u_t) = I$; on the other hand, the hypothesis of $cov(f_t) = I$ is essential to prevent indeterminancy of estimated factors.

Regarding factors estimate with `dynFA` function, we applied collapsing technique to reduce computations each time Kalman filter was employed.

In order to evaluate goodness of estimation, we considered *trace statistics* as in paper [4]. Let indicate F the storage matrix of simulated factors and $\hat{F}$ the storage matrix of estimated factors with one of the methods named before, then we compute the statistic

$$\frac{tr\left(F'\hat{F}\left(\hat{F}'\hat{F}\right)^{-1}\hat{F}'F\right)}{tr\left(F'F\right)}.$$

The trace statistic is smaller than 1 and tends to 1 if the empirical canonical correlations between the true factors and their estimates tend to 1; the closer the statistic is to 1, better the approximation of the factors.

Moreover, we compared vectors and matrices of the model with their estimates plotting values.

We could now analyse in detail some model specifications, which differ mostly for the value of p and the shape of the matrices Φ_j, $j = 1, \ldots, p$. In all models, we consider $r = 3$ common factors.

4.1.1 Starting model

First of all, we tested the case with $p = 1$ and $\Phi_1 = \alpha I$, where $0 < \alpha < 1$ ensures stationarity; we set $\alpha = 0.9$. Results are presented below.

In this simple case, we compared running time of `dynFA` function with and without the collapsing technique:

	time (s)
w/o collapsing	35.22
w/ collapsing	3.82

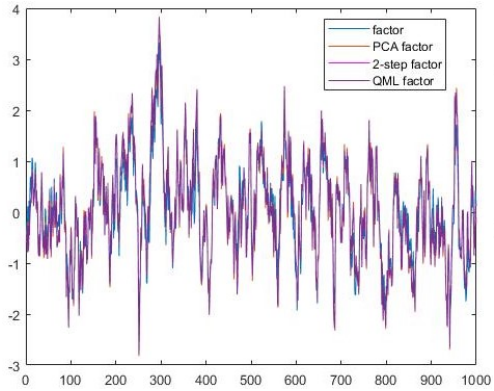
It is clear that collapsing reduces calculation time of several times. Moreover, it is interesting to underline that the QML method is the more lenghty, while PCA and 2-step methods are quite immediate.

Results obtained with and without collapsing were identical, but actually we report the first ones. Trace statistics were really good:

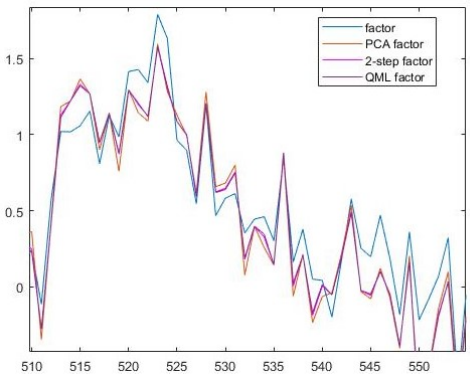
PCA	2-step	QML
0.9942	0.9958	0.9959

Obviously they increase along the different methods, as each method is more accurate that the previous one.

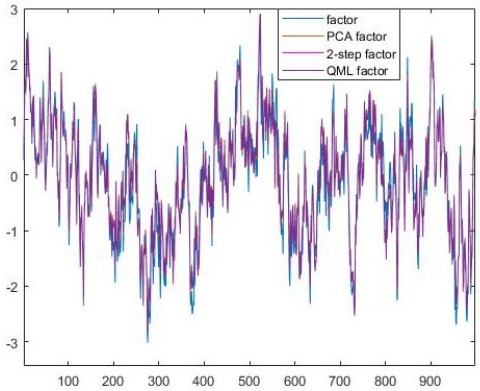
Hence we compared simulated and estimated vectors and matrices of the model, starting from factors. As said in Section 3.2.3, factors and matrices may be reconstructed despite of signs and permutation; computing correlation between each simulated factor and each estimated factor gave us the right matching of them. Results of matched estimations are shown in Figure 4.1. It is remarkable that estimates with different methods are really close among them, even when they are not so close to the simulated factor, and that more complex method gives better estimate. This is quite obvious, and the similarity among estimates is also clearly explained as each method is an improving of the previous and all of them start from the PCA.



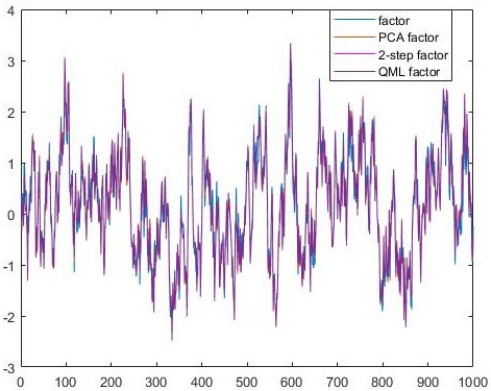
(a) 1st simulated factor and 3rd estimated factor



(b) Focus



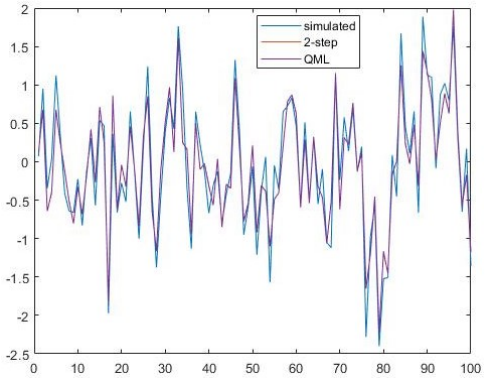
(c) 2nd sim. factor and 1st est. factor



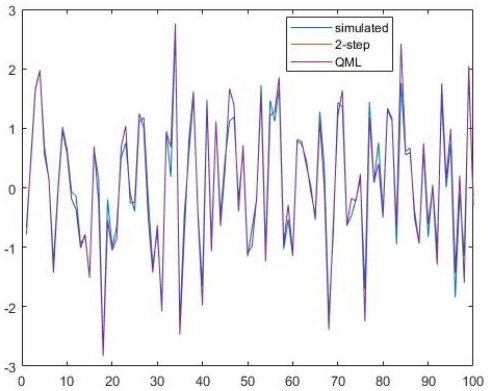
(d) 3rd sim. factor and 2nd est. factor

Figure 4.1: Comparison among simulated factors and estimated ones, matched according to correlations.

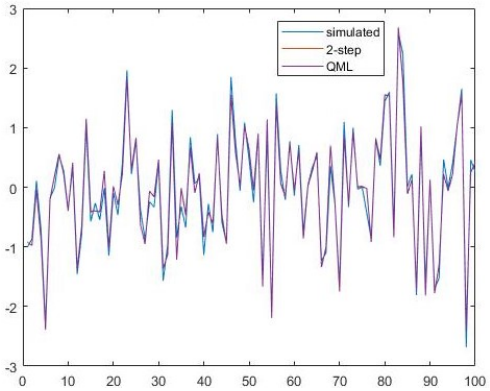
Simulated matrix of loading Λ and its estimated $\hat{\Lambda}$ suffered the same mis-matching of the factors along their columns, but after matching estimate was really good, as it could be seen in Figure 4.2. Comparing to factors estimate, in this case different methods gave indistinguishable results.



(a) 1st simulated column of Λ and 3rd estimated column of $\hat{\Lambda}$



(b) 2nd sim. column and 1st est. column



(c) 3rd sim. column and 2nd est. column

Figure 4.2: Comparison among simulated column of Λ and estimated ones, matched according to correlations.

Proceeding with order, it was the turn of comparing Ψ and its estimates $\hat{\Psi}$. All matrices are diagonal and Figure 4.3 compares the diagonals.

The estimation of α was also great:

	estimated values	
set value	2-step	QML
0.900	0.888	0.894

As introduced at the beginning of this Section 4.1, matrix Q had to be computed in order to ensure that $cov(f_t) = I$. In this simple case with $p = 1$ and $\Phi_1 = \alpha I$, we had

$$\begin{aligned} & I = \alpha I \alpha + Q \\ \Rightarrow \quad & Q = (1 - \alpha^2) I \end{aligned}$$

where $1 - \alpha^2 = 0.19$ as we chose $\alpha = 0.9$. Model estimation gave as a

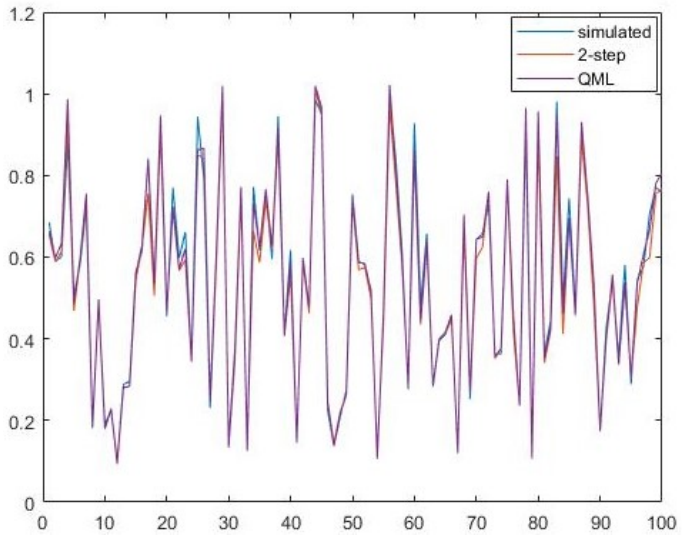


Figure 4.3: Comparison among diagonal of simulated Ψ and its estimates.

result

$$\hat{Q} = \begin{pmatrix} 0.184 & -0.004 & 0.002 \\ -0.004 & 0.219 & 0.007 \\ 0.002 & 0.007 & 0.239 \end{pmatrix}$$

which is a good estimation of $Q = diag(0.19, 0.19, 0.19)$.

Summing up, in this case estimation worked very well and all vectors and matrices had been recostructed accurately.

4.1.2 More complex model

When p is greater than 1, the model becomes more challenging and so estimation. Particularly, we need to arrange the factors in the companion form. For semplicity we chose $p = 2$, hence the model is

$$\begin{aligned} x_t &= \Lambda f_t + e_t, & e_t &\sim \mathcal{N}(0, \Psi) \\ f_t &= \Phi_1 f_{t-1} + \Phi_2 f_{t-2} + u_t, & u_t &\sim \mathcal{N}(0, Q). \end{aligned}$$

Let define

$$F_t = \begin{pmatrix} f_t \\ f_{t-1} \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \Phi_1 & \Phi_2 \\ I & 0 \end{pmatrix}$$

then the model may be written in the form

$$\begin{aligned} x_t &= \Lambda f_t + e_t, & e_t &\sim \mathcal{N}(0, \Psi) \\ F_t &= \Phi F_{t-1} + u_t, & u_t &\sim \mathcal{N}(0, Q) \end{aligned}$$

adding necessary zeros in vectors and matrices in order to match dimensions.

The biggest difference comparing with codes of previous model is about the model simulation and particularly about the matrix Q. Another difference regards the employed methods: since QML tried to compute inverse of singular matrices, we dropped this method out, and this choice is supported from the knowledge that for very greater dimension of the problem the 2-step method is more suitable.

In order to derive an expression for Q, we want to impose $E\left(f_t f_t'\right) = cov(f_t) = I$. Since $f_t f_t' = \left(\Phi_1 f_{t-1} + \Phi_2 f_{t-2} + u_t\right)\left(\Phi_1 f_{t-1} + \Phi_2 f_{t-2} + u_t\right)$ and all variables are assumed to be stationary in covariance,

$$I = \Phi_1 I \Phi_1' + \Phi_2 I \Phi_2' + \Phi_1 E\left(f_{t-1} f_{t-2}'\right)\Phi_2' + \Phi_2 E\left(f_{t-2} f_{t-1}'\right)\Phi_1' + Q.$$

Applying the expected value to $f_t f_{t-1}' = \Phi_1 f_{t-1} f_{t-1}' + \Phi_2 f_{t-2} f_{t-1}' + u_t f_{t-1}'$ and assuming symmetry of $\Psi_1 = E\left(f_t f_{t-1}'\right)$, we have

$$\Psi_1 = \Phi_1 I + \Phi_2 \Psi_1 \qquad \Rightarrow \qquad \Psi_1 = \left(I - \Phi_2\right)^{-1}\Phi_1$$

and then

$$Q = I - \Phi_1 \Phi_1' - \Phi_2 \Phi_2' - K - K' \quad \text{with} \quad K = \Phi_1\left(I - \Phi_2\right)^{-1}\Phi_1 \Phi_2'.$$

With this model specification and parameters set as before when not specified, results are still very good. Particularly, computation time is $1.8822s$ and trace statistics for PCA and 2-step method are 0.9943 and 0.9954 respectively. About Φ matrix, we set values of Φ_1 and Φ_2 "randomly but not too much": we focused on the diagonal and assigned small values to other elements, assuring that spectral radius of Φ keeps less than 1. Simulated and estimated matrices follow:

$$Q = \begin{pmatrix} 0.809 & -0.183 & -0.157 \\ -0.183 & 0.671 & -0.284 \\ -0.157 & -0.284 & 0.561 \end{pmatrix}, \quad \hat{Q} = \begin{pmatrix} 0.564 & -0.277 & -0.166 \\ -0.277 & 0.626 & -0.120 \\ -0.166 & -0.120 & 0.874 \end{pmatrix}.$$

It is evident that rows and columns of $\hat{Q}$ compared to Q suffer same permutation of columns of factors and $\hat{\Lambda}$; nevertheless, estimate is good. Also estimation of $\hat{\Phi}$ is good despite of permutation, as it could be seen:

$$\Phi_1 = \begin{pmatrix} 0.320 & 0.160 & 0.080 \\ 0.080 & 0.400 & 0.160 \\ 0.080 & 0.240 & 0.480 \end{pmatrix}, \quad \hat{\Phi}_1 = \begin{pmatrix} 0.450 & 0.202 & 0.190 \\ 0.155 & 0.460 & 0.040 \\ -0.027 & 0.075 & 0.229 \end{pmatrix}$$

$$\Phi_2 = \begin{pmatrix} 0.080 & 0.064 & 0.008 \\ 0.056 & 0.160 & 0.032 \\ 0.008 & 0.016 & 0.160 \end{pmatrix}, \quad \hat{\Phi}_2 = \begin{pmatrix} 0.166 & 0.006 & 0.051 \\ 0.047 & 0.147 & 0.046 \\ 0.152 & 0.041 & 0.126 \end{pmatrix}.$$

Graphs are really similar to previous case ones, so we avoid reporting them to not bore reader.

4.1.3 Autoregressive model of order *p=6*

We considered $p = 6$ and unconstrained autoregressive coefficients Φ. Analogously to before, the model is

$$\begin{aligned} x_t &= \Lambda f_t + e_t, & e_t &\sim \mathcal{N}(0, \Psi) \\ f_t &= \Phi_1 f_{t-1} + \Phi_2 f_{t-2} + \ldots + \Phi_p f_{t-p} + u_t, & u_t &\sim \mathcal{N}(0, Q) \end{aligned}$$

For semplicity, we considered matrices $\Phi_2 = \Phi_3 = \ldots = \Phi_p$, as we did in Section 2.3 for the autoregressive term. Hence, defining

$$F_t = \begin{pmatrix} f_t \\ f_{t-1} \\ \vdots \\ f_{t-p+1} \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_2 \\ I & & & 0 \\ & \ddots & & \vdots \\ & & I & 0 \end{pmatrix},$$

the model could be written as

$$\begin{aligned} x_t &= \Lambda f_t + e_t, & e_t &\sim \mathcal{N}(0, \Psi) \\ F_t &= \Phi F_{t-1} + u_t, & u_t &\sim \mathcal{N}(0, Q). \end{aligned}$$

Computing of Q for ensure $cov(f_t) = I$ is analogous to previous model, but we should explicit it. From recursion for f_t it follows

$$\begin{aligned} f_t f_t' = (&\Phi_1 f_{t-1} + \Phi_2 f_{t-2} + \ldots + \Phi_2 f_{t-p} + u_t) \cdot \\ &\cdot (\Phi_1 f_{t-1} + \Phi_2 f_{t-2} + \ldots + \Phi_2 f_{t-p} + u_t). \end{aligned}$$

Applying the expected value, deriving Q and indicating $\Psi_j = E(f_t f_{t-j})$, we obtain

$$\begin{aligned} Q = I - [\ &\Phi_1\Phi_1' + (p-1)\,\Phi_2\Phi_2' + \Phi_1\Psi_1\Phi_2' + \Phi_2\Psi_1'\Phi_1' + \\ &+ (p-2)\,\Phi_2\Psi_1\Phi_2' + (p-2)\,\Phi_2'\Psi_1'\Phi_2 + \Phi_1\Psi_2\Phi_2' + \Phi_2\Psi_2'\Phi_1' + \\ &+ (p-3)\,\Phi_2\Psi_2\Phi_2' + (p-3)\,\Phi_2'\Psi_2'\Phi_2 + \Phi_1\Psi_3\Phi_2' + \Phi_2\Psi_3'\Phi_1' + \\ &+ \quad \ldots \quad + \\ &+ \Phi_2\Psi_{p-2}\Phi_2' + \Phi_2'\Psi_{p-2}'\Phi_2 + \Phi_1\Psi_{p-1}\Phi_2' + \Phi_2\Psi_{p-1}'\Phi_1'\ ]. \end{aligned}$$

It is obvious that the value of p determines the length of this sum, and consequentely the number of Ψ_j terms needed; let now compute these terms in the case of our interest: $p = 6$.

$$\begin{aligned} \Psi_1 &= E(f_t f_{t-1}) = E\left(\Phi_1 f_{t-1} f_{t-1}' + \Phi_2 f_{t-2} f_{t-1}' + \ldots + \Phi_2 f_{t-p} f_{t-1}' + u_t f_{t-1}'\right) = \\ &= \Phi_1 + \Phi_2\Psi_1 + \Phi_2\Psi_2 + \ldots + \Phi_2\Psi_5, \\ \Psi_2 &= E(f_t f_{t-2}) = E\left(\Phi_1 f_{t-1} f_{t-2}' + \Phi_2 f_{t-2} f_{t-2}' + \ldots + \Phi_2 f_{t-p} f_{t-2}' + u_t f_{t-2}'\right) = \\ &= \Phi_1\Psi_1 + \Phi_2 + \Phi_2\Psi_1 + \Phi_2\Psi_2 + \ldots + \Phi_2\Psi_4, \\ \Psi_3 &= E(f_t f_{t-3}) = E\left(\Phi_1 f_{t-1} f_{t-3}' + \Phi_2 f_{t-2} f_{t-3}' + \ldots + \Phi_2 f_{t-p} f_{t-3}' + u_t f_{t-3}'\right) = \\ &= \Phi_1\Psi_2 + \Phi_2\Psi_1 + \Phi_2 + \Phi_2\Psi_1 + \Phi_2\Psi_2 + \ldots + \Phi_2\Psi_3, \\ \Psi_4 &= E(f_t f_{t-4}) = \Phi_1\Psi_3 + \Phi_2\Psi_2 + \Phi_2\Psi_1 + \Phi_2 + \Phi_2\Psi_1 + \Phi_2\Psi_2, \\ \Psi_5 &= E(f_t f_{t-5}) = \Phi_1\Psi_4 + \Phi_2\Psi_3 + \Phi_2\Psi_2 + \Phi_2\Psi_1 + \Phi_2 + \Phi_2\Psi_1. \end{aligned}$$

Clearly, a linear system with variables Ψ_j, $j = 1, \ldots, 5$ has to be solved in order to obtain expressions to substitute in above equation for Q.

The results from the estimation exercise are very good. We do not show them because the computation times, the summary statistics, and the goodness-of-fit are very similar to the ones shown for previous model specifications.

Chapter 5

From observations to forecasting

In this chapter we will finally put together all the work done so far in order to fit models to real data and achive the goal of forecasting. We will follow the procedure presented by Li et al. in [10], that consists of three steps employing univariate state-space model, dynamic factor model and simulation smoothing.

5.1 Three-step forecasting method

Li et al. [10] proposed a forecasting exercise consisting of three stages:

Step 1: analysis, modelling and prediction of the variable of interest, in our case maximum temperature in different stations, using a univariate state-space model;

Step 2: joint analysis of the prediction errors from Step 1 using a multivariate state-space model like the dynamic factor model;

Step 3: simulation of the prediction errors series from Step 1 conditionally to the data simulated with the dynamic factor model in Step 2. The simulated errors series can be transformed to an ensemble time series of the temperature which can be forecasted using the univariate model of Step 1 with the parameters re-estimated. The sample average of these ensemble forecasts is the final forecast.

We anticipate that we took into account weekly time series coming from different stations in the region of Emilia Romagna (Italy).

5.1.1 Step 1: univariate state space model

The weekly time series recorded by each station is decomposed using the model given by

$$y_t = \mu_t + \gamma_t + c_t + \varphi_t$$

where the latent components are trend μ_t, seasonal γ_t and cyclical c_t components and autoregressive disturbances ϕ_t. It is the kind of decomposition presented in equation (1.2) of Section 1.2 and perfected with equation (2.3) in Section 2.2. Parameters of each component were discussed in Chapters 1 and 2, and so the maximum likelihood estimation using Kalman filter recursions, which compute the one-step ahead prediction errors

$$v_t = y_t - E\left(y_t | Y_{t-1}\right).$$

It is remarkable that v_t is computed from y_t with an invertible transformation, that is the Kalman filter recursions, which we will invert explicitly in Step 3 (Section 5.1.3).

5.1.2 Step 2: dynamic factor model

Repeating Step 1 for all the stations, N prediction errors time series are obtained: a joint analysis of them allow exctracting residual information. Defining x_t the vector of those prediction errors at time t, the dynamic factor model (DFM) is given by

$$\begin{aligned} x_t &= \Lambda f_t + e_t \\ f_t &= \Phi_1 f_{t-1} + \ldots + \Phi_p f_{t-p} + u_t \end{aligned}$$

where f_t is the vector of factors, Λ the factor loading matrix, e_t and u_t are disturbances, and $\Phi_1, \ldots, \Phi_p$ autoregressive coefficient matrices. Again, model discussion was addressed in Chapters 3 and 4, along with the two-step method by Doz et al. [5] for parameters estimation.

Then, the simulation smoother by Durbin and Koopman [6] permits sampling a set of M simulated prediction errors series v_t^i, $i = 1, \ldots, M$ conditionally to the estimated series v_t. For simulation smoother explanation we refer to Section 1.7, of which we recall the highlights:

1. first of all, a series of prediction errors and relative factors is generated unconditionally via the function `sim-mod` implemented in Section 4.1.3;

2. Kalman filter and state smoothing recursions are applied to the series of unconditional factors;

3. a conditional sample of factors f_t^i is obtained by the algebraic sum of the estimated factors, unconditional factors and smoothed unconditional factors;

4. finally, a conditional sample of prediction errors is computed thanks to the equation $v_t^i = \hat{\Lambda} f_t^i + e_t^i$, where e_t^i is randomly sampled with variance matrix $\hat{\Psi}$ and hatted matrices resulted from two-step method by Doz et al. [5] for parameters estimation in the DFM.

5.1.3 Step 3: forecasting via simulation and estimation

Based on the set of simulated series v_t^i, a set of artificial time series y_t^i could be constructed via the inverse trasformation hinted in Step 1. This sequence of M time series y_t^i, $i = 1, \ldots, M$ is denoted by *ensemble time series* and in fact is the result from an interaction between the univariate state-space model and the DFM. Now we will illustrate the transformation and, for this purpose, let us forget momentarily the index i. Since $v_t = y_t - Za_t$, obviously we have

$$y_t = v_t + Za_t.$$

The following recursions allow to compute all the needed elements to extract y_t supposing v_t to be known:

$$\begin{aligned} F_t &= ZP_tZ' \\ K_t &= TP_tZ'F_t^{-1} \\ a_{t+1} &= Ta_t + K_tv_t \\ P_{t+1} &= TP_t\left(T - K_tZ\right)' + RQR' \end{aligned}$$

where vector Z and matrix R are known from model specification, while matrices Q, T, a_1 and P_1 are fixed equal to the estimated values from Step 1.

Hence, for each ensemble time series, forecasts $\hat{y}_{T+h}^i$, $h = 1, \ldots, H$ are computed iterating Kalman filter recursions. Before computing the forecast values, the parameter values of the univariate model are re-estimated from the simulated time series. To clarify this point, let us go step by step:

1. for each ensemble time series y_t^i, a new estimation of the univariate state-space model parameters is performed;

2. re-estimated parameters are employed to compute the model matrices, in particular a_t^i, P_t^i for $t = 1, \ldots, T$ and T^i;

3. Kalman filter recursions are iterated over last index $t = T$ obtaining forecasts, precisely

$$\begin{aligned} a_{T+h} &= T^i a_{T+h-1} \\ y_{T+h}^i &= Za_{T+h} \end{aligned}$$

for $h = 1, \ldots, H$.

The idea behind our approach, which is inspired by Li et al., is that while generating M conditional series of observation is necessary to catch

variability of data itself, in a similar way re-estimation of parameters allow to seize their own statistical uncertainty.

The final forecast from this three-step procedure is simply obtained averaging

$$\hat{y}_{T+h} = \frac{1}{M} \sum_{i=1}^{M} y^i_{T+h}$$

for $h = 1, \ldots, H$. Actually, under our assumptions of normality, the distribution of the final forecast is easly computed with the standard deviation of the ensemble forecasts.

5.2 Forecasts in Emilia-Romagna

We downloaded daily maximum temperature data from the site of European Climate Assessment and Dataset [9] and, as introduced before, we accounted data recorded in some stations sited in Emilia-Romagna. We selected $N = 20$ stations which are more or less evenly distributed over the territory of the region, some on the plain, others on the hills, as shown in the following table and Figure 5.1.

station	altitude	temperatures		
		min	mean	max
Bagno di Romagna	500	1.3	21.0	37.3
Bardi	625	1.2	20.2	35.2
Bettola	329	1.8	20.5	35.5
Bologna	54	3.2	22.8	38.9
Castelnovo ne'Monti	750	-0.2	18.3	34.5
Cesena	44	2.9	22.5	38.5
Civitella di Romagna	219	2.0	21.4	38.0
Faenza	35	1.3	21.0	37.3
Forlí	34	3.2	22.7	38.5
Imola	47	3.2	24.1	42.7
Mirandola	9	2.2	22.9	39.9
Modena	34	3.1	22.7	39.3
Monghidoro	841	-0.3	18.9	34.2
Montefiorino	797	0.7	20.7	36.3
Piacenza	61	2.7	22.3	37.5
Ravenna	4	2.7	22.2	37.2
Reggio Emilia	56	3.8	22.9	38.3
Rimini	6	3.5	21.8	38.4
Vignola	125	2.9	22.8	39.0
Volano	1	3.9	22.4	38.5

In each time series we selected data from 01/01/2006 to 31/12/2014, and we manipulated them analogously to what we done in Section 2.3:

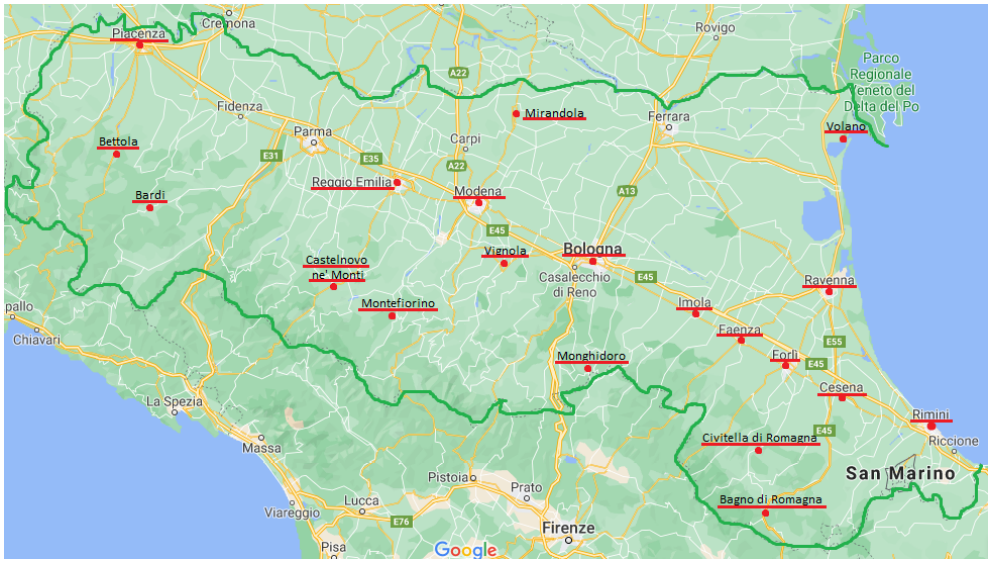


Figure 5.1: Selected stations in Emilia Romagna.

data relative to February 29^{th} and December 31^{st} were excluded and remaining data were aggregated on a weekly basis computing the maximum; in fact, any aggregated data is the maximum of maximum temperatures recorded in a week. Some statistics of the manipulated data are provided in the table above and Figure 5.2 is referred for an example of resulted time series.

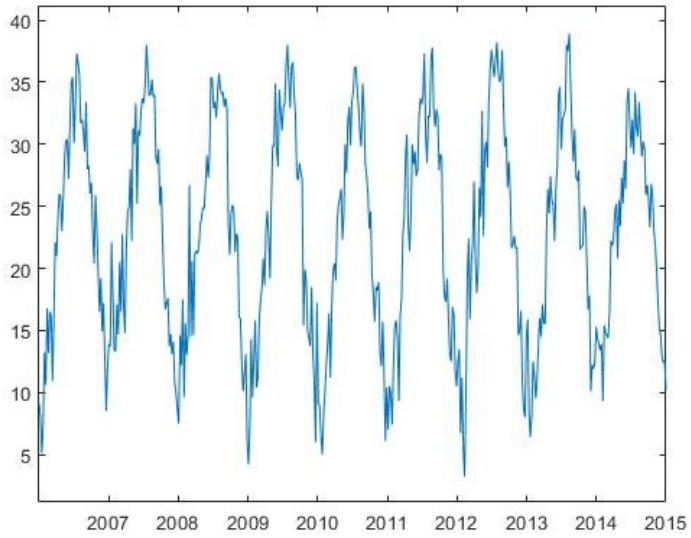
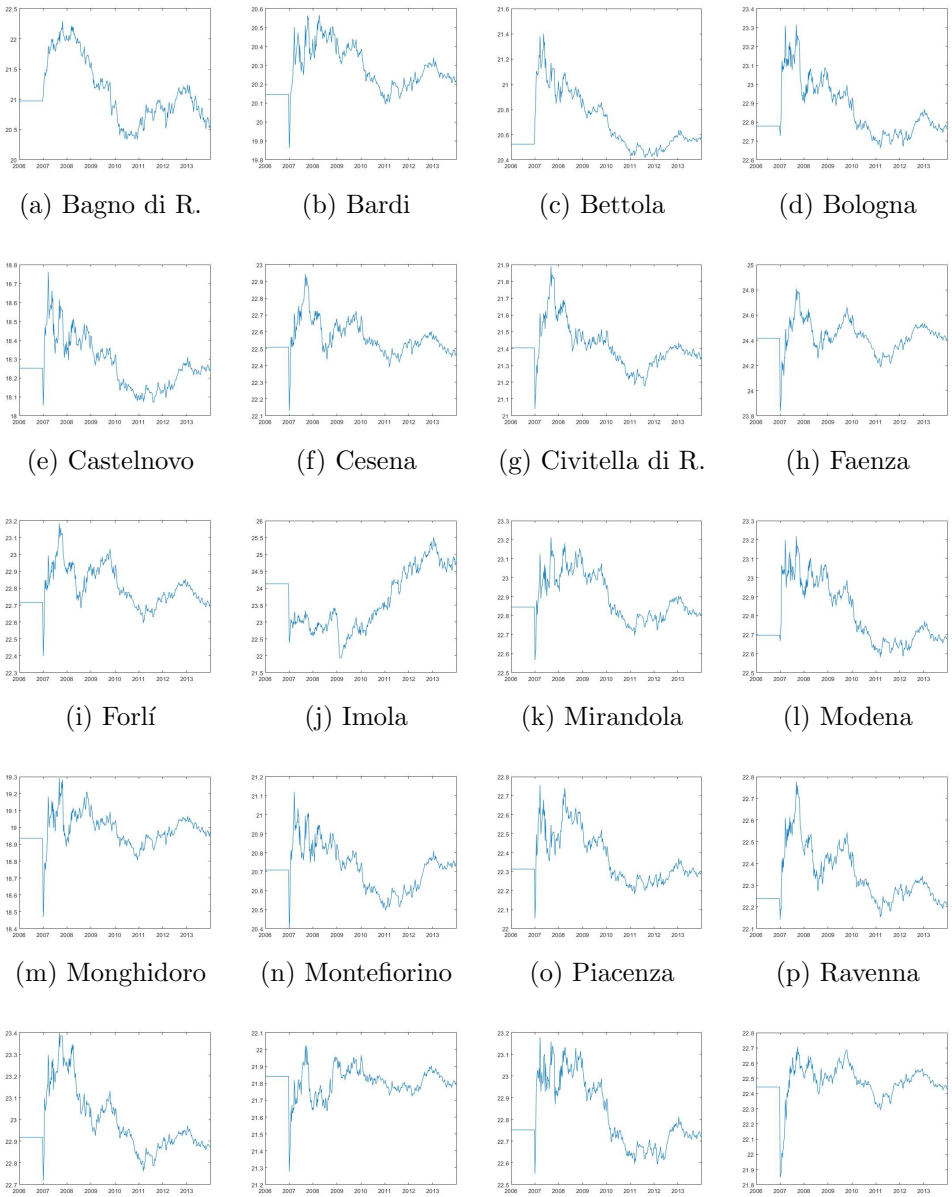


Figure 5.2: Manipulated time series of the station of Bologna.

Hence the univariate state-space model was fitted to each time series on data from 01/01/2006 to 31/12/2013 with kind of analysis employed for station of Bologna in Section 2.3; data relative to year 2014 were preserved for testing the forecast ability at the end of the three-step

procedure. Results of components estimation are shown in Figures 5.3, 5.4 and 5.5. As could be seen in ghaphs, trend components are quite different among the different stations, while seasonal components are pretty similar to each other; about cycle components, two main patterns are observed: the one with 3 valleys and the one with 5 valleys, which corresponds to frequency parameter λ_c with values about 0.50 and 0.87 respectively. Obviously, each estimated component of each station starts with a null seasonality because of the parameters exact initialisation.



(a) Bagno di R. (b) Bardi (c) Bettola (d) Bologna

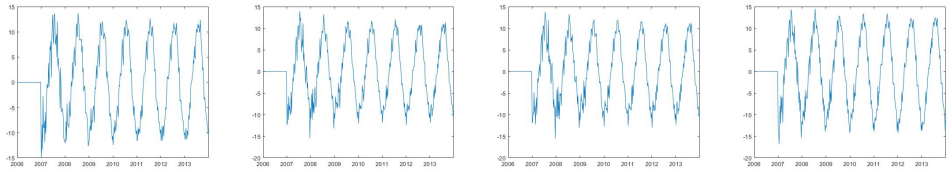
(e) Castelnovo (f) Cesena (g) Civitella di R. (h) Faenza

(i) Forlí (j) Imola (k) Mirandola (l) Modena

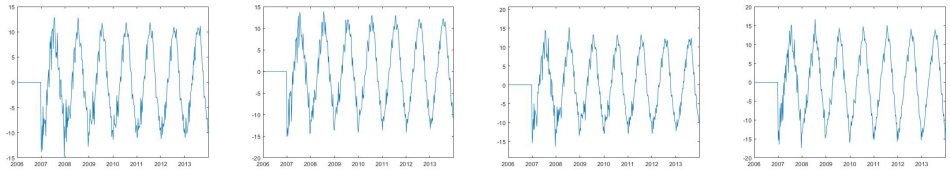
(m) Monghidoro (n) Montefiorino (o) Piacenza (p) Ravenna

(q) Reggio Emilia (r) Rimini (s) Vignola (t) Volano

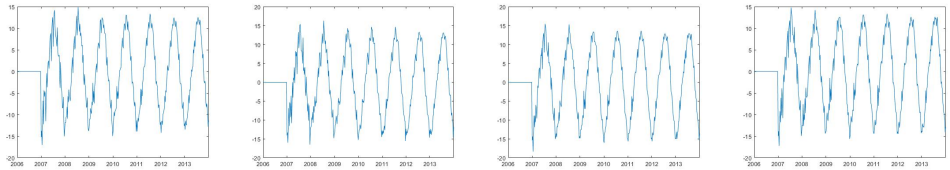
Figure 5.3: Trend component estimates from the univariate model.



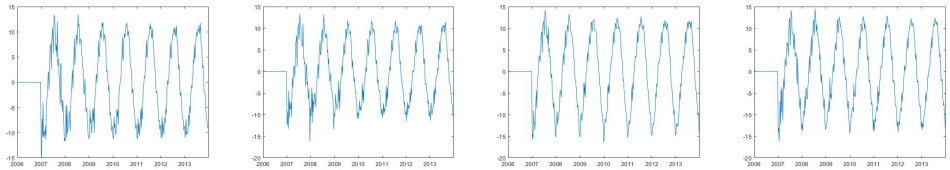
(a) Bagno di R. (b) Bardi (c) Bettola (d) Bologna



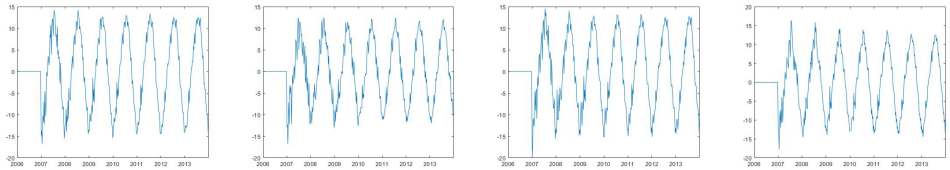
(e) Castelnovo (f) Cesena (g) Civitella di R. (h) Faenza



(i) Forlí (j) Imola (k) Mirandola (l) Modena



(m) Monghidoro (n) Montefiorino (o) Piacenza (p) Ravenna

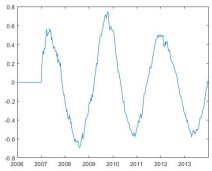


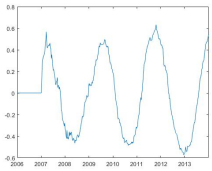
(q) Reggio Emilia (r) Rimini (s) Vignola (t) Volano

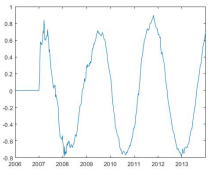
Figure 5.4: Seasonal component estimates from the univariate model.

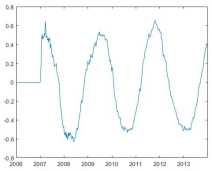
We stored all the prediction error series computed with univariate model in a single matrix, in alphabetical order according to stations names and removed of the initial null values, and feed it to the function `dynFA` presented in Chapter 4 and modified in order to apply the two-step method presented by Doz et al. [5]. Estimated factors are plotted in Figure 5.6.

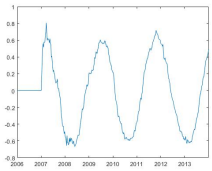
A look at the estimated matrix $\hat{\Phi}$ suggests a low persistence of factors as confirmed by the spectrum of the autoregressive matrix in the

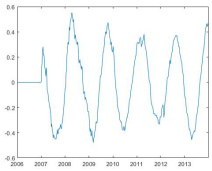

(a) Bagno di R.

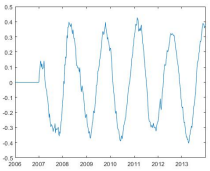

(b) Bardi

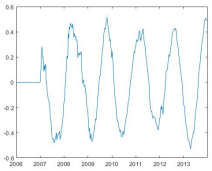

(c) Bettola

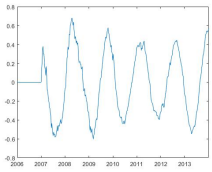

(d) Bologna

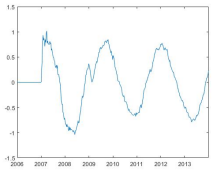

(e) Castelnovo

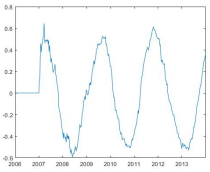

(f) Cesena

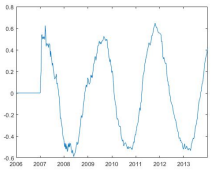

(g) Civitella di R.

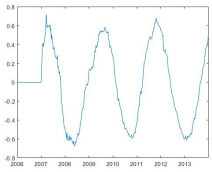

(h) Faenza

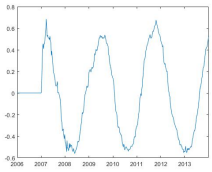

(i) Forlí

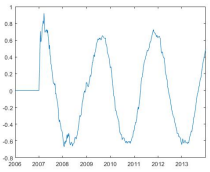

(j) Imola

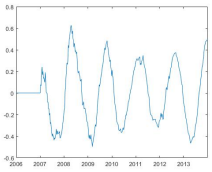

(k) Mirandola

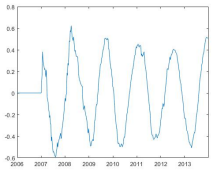

(l) Modena

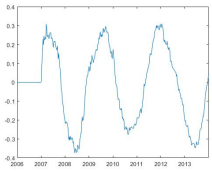

(m) Monghidoro

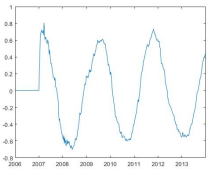

(n) Montefiorino


(o) Piacenza


(p) Ravenna


(q) Reggio Emilia


(r) Rimini


(s) Vignola

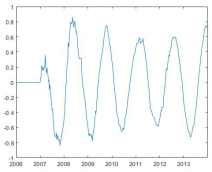
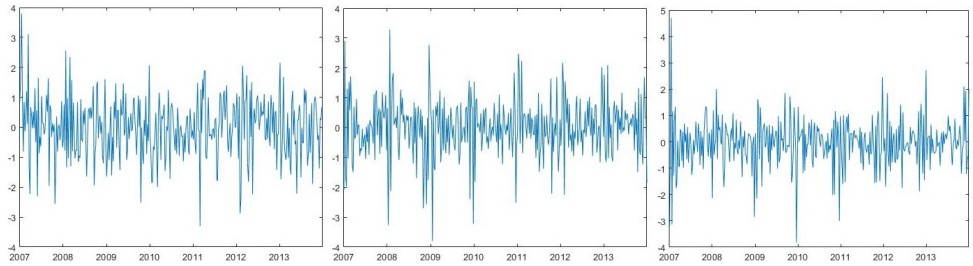

(t) Volano

Figure 5.5: Cycle component estimates from the univariate model.

companion form:

$$\hat{\Phi}_1 = \begin{pmatrix} -0.008 & 0.055 & 0.024 \\ -0.093 & -0.168 & 0.084 \\ -0.102 & -0.019 & -0.118 \end{pmatrix}, \quad \hat{\Phi}_2 = \begin{pmatrix} -0.070 & 0.023 & 0.053 \\ -0.022 & -0.042 & 0.049 \\ 0.047 & 0.005 & -0.059 \end{pmatrix}.$$

As a consequence the value of p may be not so relevant; nevertheless, aiming to confirm this intuition and choose the best value for p, we analysed the prediction errors obtained with the Kalman filter recursions



(a) 1st factor (b) 2nd factor (c) 3rd factor

Figure 5.6: Estimated common factors of prediction errors time series.

in the second step of factors estimation procedure. In other words, we performed a miss-specification test by analysing the correlation structure of the prediction errors of the factor model estimated from the prediction errors of the univariate time series. No residual correlation is visible, as could be verified in Figure 5.7 for the station of Bologna. Consequentely, we set $p = 4$.

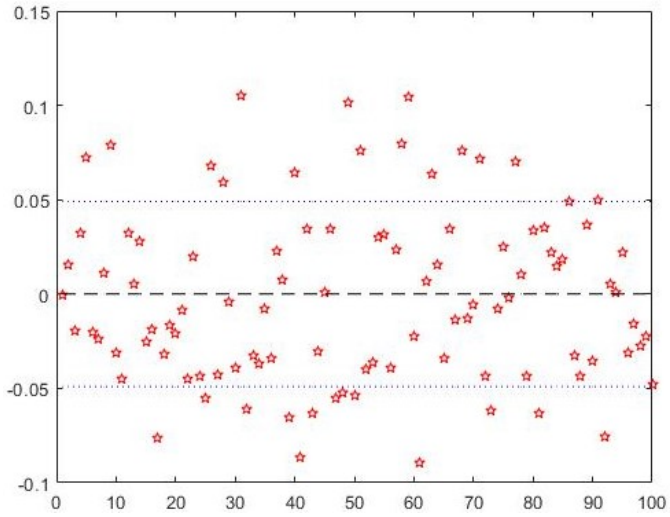


Figure 5.7: Auto-correlogram for prediction errors of prediction errors of Bologna time series.

With matrices of the DFM just estimated, we managed to sample a set of $M = 50$ simulated prediction error series thanks to the simulation smoother of Durbin and Koopman [6] and then compute from them the set of ensemble time series. This last computation required parameters of the univariate model for each observed time series, that is for each station; moreover, according to the three-step procedure of Li et al. [10], in this third stage parameters of the univariate model had to be re-estimated for each simulated time series of each station. Hence, because of computing complexity, we will show final results for the station of Bologna only.

We remark that parameters re-estimation implies again a maximisation of log-likelihood and consequentely an exact initial procedure, which causes the 'loss' of d data, where $d = 52$ for our model; in the first step of Li et al.'s method we already lost data regarding the whole year 2006, and likewise it would happen now (for year 2007). In our exploration, with just a few years of available data, it would be unpleasant. We solved this problem initialising the optimization with the previously estimated parameters. In other words, instead of employing the exact initial procedure with data from 2007 to 2013 'wasting' data of year 2007, we set a_1 and P_1 of the univariate model equal to their estimated value computed with data from 2006 to 2013 at the beginning of all this exploration.

Re-estimated parameters have been used in Kalman filter recursions to obtain ensemble forecasts, whose distribution was computed in order to conclude the forecasting procedure. Aiming to one-year ahead predictions, we set $H = 52$; furthermore, as hinted in Section 5.1.3, since we assumed the normality hypothesis when possible, the named distribution is univocally determined by mean and standard deviation of data. Observations of year 2014 were finally taken into account in order to evaluate forecasting goodness. In Figure 5.8 the blue line is the observed time series while the red one requires some explanation:

- from year 2007 to the whole 2013, the red line is computed averaging the ensemble time series y_t^i for $i = 1, \ldots, M$, consequentely it is the mean of conditional samples of observed time series;

- in year 2014, the red line is computed averaging predictions of the ensemble time series, that are y_{T+h}^i for $i = 1, \ldots, M$, in other words it is the forecast itself and it is clearly out-of-sample with respect to the data which were accounted for analysis and simulation.

It is remarkable that the maximum data for year 2014 (approximately 34° C) is lower than maximum data of previous years (approximately 38° C), nevertheless our model managed to predict it quite well.

In Figure 5.9 the same series of Figure 5.8 are shown with same colours, but for year 2014 only. The red line plots computed forecasts and the black dotted line signs 1.96 times the standard deviation of data: this means that the 95% of observed data should be within this threshold. It seems so, taking into account possible statistical fluctuations.

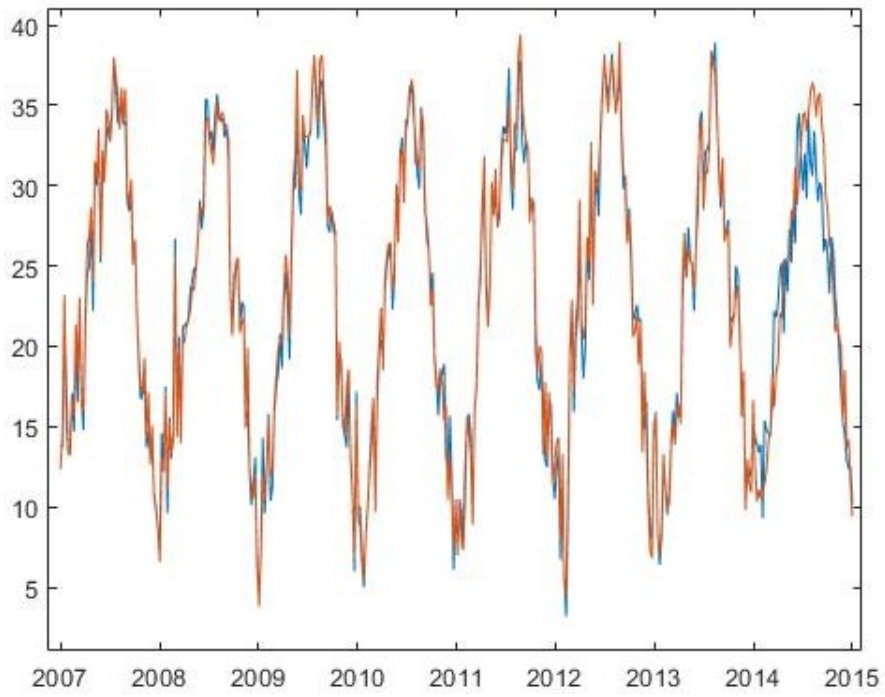


Figure 5.8: Simulations and forecasts (red line) for the station of Bologna compared to observations (blue line).

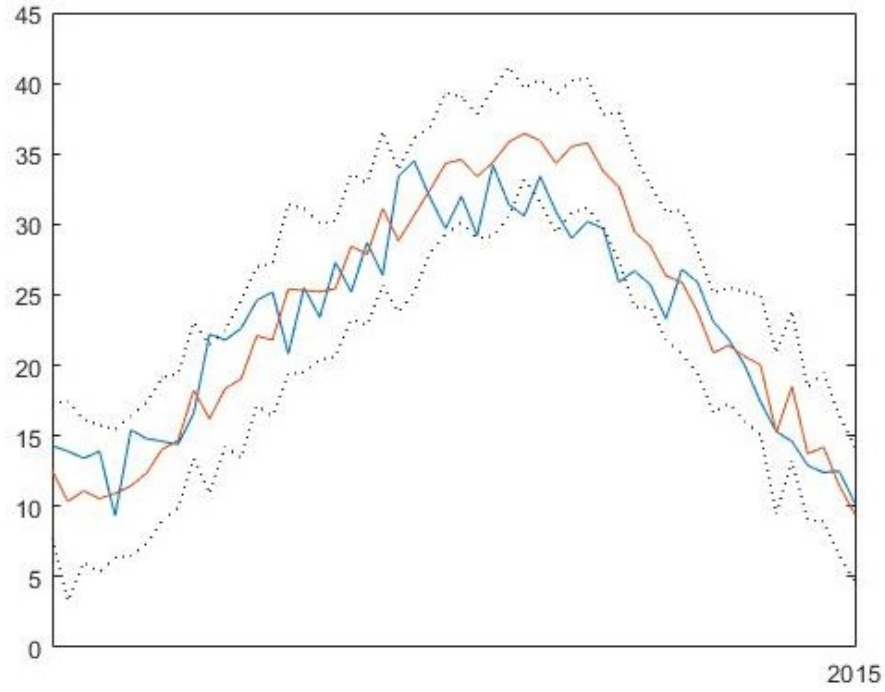


Figure 5.9: Forecasts (red line) for the station of Bologna compared to observations (blue line), and the 1.96-times standard deviation region (black dotted lines)

Conclusions

Time series had been the center of our studies, both generated with Monte-Carlo simulations and live recorded. We initially took confidence with the state space model, particularly with the state vector and latent components; afterwards it was the turn of the dynamic factor model (that can be cast itself in a state space model form), with factors, loadings and autoregressive order of the factors.

The whole procedure of analysis and forecasting of the climatic time series goes throught the state space model, then the dynamic factor model and after that comes back: once factors had been estimated, they are simulated, permitting simulation of prediction errors and therefore of observations; holding simulated observations, forecasting is computed.

The results we obtained are promising. They demonstrate the feasibility of the model presented by Li et al. [10]. Our exploration involved just a climatic parameter (the maximum temperature) and a few data (a weekly basis on 8 years and 20 stations) but we very preliminarily tested the whole approach on much longer time series (150 years on a weekly basis) with a richer cross-sectional data (up to 10.000 nodes) finding feasible computational times and very promising estimation results.

As a summary, in Chapter 1 we presented the state space model: the decomposition in latent components, the Kalman filter recursions, the simulation smoothing, the miss-specification tests and the exact initialisation; and in Chapter 2 we applied these techniques in order to analyse some time series obtained both with Monte-Carlo simulation and live recorded. Similarly, in Chapter 3 we introduced dynamic factor models and particularly the two-step estimation method by Doz, Giannone and Reichlin [5], which we employed in Chapter 4 to estimate model parameters of some Monte-Carlo simulated series. Finally, in Chapter 5 we applied a sort of Li et al.'s [10] forecasting method to time series of maximum temperatures recorded in Emilia-Romagna (Italy): throught state space and dynamic factor models, the analysis of such time series and the simulation smoothing technique by Durbin and Koopman [6] allowed us to compute forecasts.

Bibliography

[1] Gary Chamberlain and Michael Rothschild. "Arbitrage, Factor Structure, and Mean-Variance Analysis on Large Asset Markets". In: *Econometrica* 51.5 (1983), pp. 1281–1304.

[2] Hadley Centre for Climate Prediction and Research. *Global Temperature Dataset*. 2020. URL: `https://crudata.uea.ac.uk/cru/data/temperature/`.

[3] Catherine Doz and Peter Fuleky. "Dynamic Factor Models". In: *Macroeconomic Forecasting in the Era of Big Data, Theory And Practice*. Vol. 52. Advanced Studies In Theoretical And Applied Econometrics. Springer, 2020. Chap. 2.

[4] Catherine Doz, Domenico Giannone, and Lucrezia Reichlin. "A Quasi-Maximum Likelihood Approach for Large Approximate Dynamic Factor Models". In: *The Review of Economics and Statistics* 94.4 (Nov. 2012), pp. 1014–1024.

[5] Catherine Doz, Domenico Giannone, and Lucrezia Reichlin. "A Two-Step Estimator for Large Approximate Dynamic Factor Models Based on Kalman Filtering". In: *Journal of Econometrics* 164.1 (2011), pp. 188–205.

[6] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods, Second Edition*. Oxford Statistical Science Series. OUP Oxford, 2012.

[7] Domenico Giannone. *Personal homepage at* Universite Libre de Bruxelles. URL: `http://homepages.ulb.ac.be/~dgiannon/`.

[8] Borus Jungbacker and Siem Jan Koopman. "Likelihood-based dynamic factor analysis for measurement and forecasting". In: *The Econometrics Journal* 18.2 (2015), pp. C1–C21.

[9] A.M.G. Klein Tank and Coauthors. *Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment*. 2002. URL: `https://eca.knmi.nl//dailydata/predefinedseries.php`.

[10] Mengheng Li et al. "Long-term Forecasting of El Nino Events via Dynamic Factor Simulations". In: *Journal of Econometrics* 214 (July 2019).

[11] Greta M. Ljung and G. E. P. Box. “On a Measure of Lack of Fit in Time Series Models”. In: *Biometrika* 65 (Aug. 1978), pp. 67–72.

[12] Terence C. Mills. “’Skinning a cat’: Alternative Models of Representing Temperature Trends”. In: *Climatic Change* 101 (Aug. 2010), pp. 415–426.

[13] Desislava B. Petrova et al. “Multiyear Statistical Prediction of ENSO Enhanced by the Tropical Pacific Observing System”. In: *Journal of Climate* 33 (Oct. 2019).

Ringraziamenti

Il mio "grazie" più sentito va al mio relatore, il prof. Bormetti, con cui è stato un piacere lavorare in questi mesi. Nella prossima vita sceglierei sicuramente una tesi meno complessa, ma è stata davvero un'esperienza molto interessante nonchè formativa. Grazie, prof, per avermi seguita così tanto e per aver messo impegno e passione in questa tesi; non ci siamo mai visti di persona, ma resteranno indimenticati i nostri incontri da due ore su Teams.

Ovviamente non posso non ringraziare il mio co-relatore, il prof. Pascucci, che ho seguito in tutti questi anni di università e che mi ha fatto appassionare al suo campo di studi, grazie anche al suo metodo didattico; se non fosse per l'ennesimo corso tenuto da lui (Analisi stocastica 2) non mi sarei mai convinta a chiedergli la tesi.

Segue un sentito ringraziamento ai colleghi di Kiwitron, che mi hanno accettata per il tirocinio curricolare (senza il quale non avrei potuto laurearmi) e poi accolta come una famiglia; siamo solo all'inizio della nostra avventura insieme e spero durerà a lungo perchè grazie a voi è davvero una gioia alzarsi alla mattina per andare al lavoro! In particolare, ringrazio Federico, che mi ha preso sotto la sua ala per il tirocinio, Venus, Antonio e Davide, che mi fanno "giocare" con loro, e Andrea, che è il capo migliore del mondo perchè si fa rimproverare sempre con il sorriso sulle labbra e soprattutto perchè crede in me; poi ci sono Adriano, Valentina, Elena, Daniele, Carlo, ecc ecc (perchè ormai siamo davvero tanti in Kiwitron e non posso citarvi tutti).

Un meritatissimo "grazie" a Davide, che in questi mesi di tesi e lavoro mi ha supportata e sopportata, e a Martina, con cui ci siamo sostenute e fatte forza a vicenda nello scrivere le rispettive tesi e che si mostra sempre entusiasta per ciò che mi capita di bello (in cambio del fatto che io prendo le decisioni difficili riguardanti la sua vita).

Un ringraziamento speciale a Marcello, senza il quale non avrei mai superato certi esami, e alle ragazze del gruppo 'Belle anche di spalle', in particolare Giulia, Benedetta e Martina B., che non mancano mai di dimostrarmi il loro affetto e la loro stima.

Infine, ma solo perchè il suo supporto è sempre costante e quindi meno eclatante, la mia mamma, che si vanta con tutti della sua "bambina" e senza la quale non sarei quella che sono. Tutti dovrebbero avere una super-mamma come te!