Università
Ca'Foscari
Venezia

Master's Degree programme – Second Cycle
(*D.M. 270/2004*)
in Economics and Finance

Final Thesis

Ca' Foscari
Dorsoduro 3246
30123 Venezia

Identification of a dynamic linear model for the American GDP

Supervisor
Ch. Prof. Andrea Pastore

Graduand
Prisca Primavera Piccoli
Matriculation Number 832672

Academic Year
2014 / 2015

Table of Contents

PREFACE

Economic forecasting has always held an important role as a decision-making tool for both businesses and governments as they formulate financial policy and strategy. Nevertheless, when dealing with real data (not only for what concerns economics, but also in sociology, biology, engineering, etc...) it rarely happens that we have perfect information about the phenomenon of interest. Even in the case in which a precise and deterministic model describing the system we want to analyse is available, there is always the possibility to omit some explicative variables, or to commit measurement errors. Thus, a given amount of uncertainty will be always present when analysing economic phenomena. This is why we cannot consider economic forecasting as an exact science.

According to the Bayesian paradigm, uncertainty should be described by means of *probability* by applying the rules of probability theory, which defines how to assign probabilities consistently, in order to avoid contradictions and undesired consequences. In fact, the problem of "learning from experience" about a specific phenomenon is solved thanks to the application of probability rules: in particular, it is necessary to compute the *conditional probability* of the event of interest, given the experimental information. For this purpose, the basic rule to be applied is expressed by Bayes' theorem, stating that, given two events A and B, the joint probability of them occurring is given by $P(A \cap B) = P(A|B)\,P(B) = P(B|A)\,P(A)$, where $P(A|B)$ is the conditional probability of A given B and $P(B)$ is the marginal probability of B. Bayes' theorem, also referred to as the theorem of inverse probability, is a consequence of the equalities reported above and states that

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad .$$

In the last few years there has been an increasing interest for the application of state-space models for time series analysis purposes. The models belonging to this category describe a given time series as the union of a dynamic system and a random disturbance component. Their powerful probabilistic structure provides us with a flexible framework which can be exploited for a wide range of applications. Estimation and forecasting problems are solved by means of recursive

algorithms, which iteratively compute the distribution of the quantities of interest, conditional on the available information. In this sense, they naturally follow the Bayesian approach.

Moreover, state-space models can be used for modelling both univariate and multivariate time series, even in situations of non-stationarity, structural changes or irregular patterns. This flexibility is what had led us to choose a subclass of state-space models, namely *dynamic linear models*, in order to identify a forecasting model for the nominal Gross Domestic Product of the United States.

In a paper entitled *Real-Time Nowcasting Nominal GDP Under Structural Break*, published in 2014 by William A. Barnett, Marcelle Chauvet and Danilo Leiva-Leon, the construction of a multivariate state space model is proposed, aiming at forecasting the trend of the American GDP. From the evaluation of the predictive capability of the different models proposed, including different sets of explicative variables, it is concluded that, if we want to create a model including three macroeconomic factors affecting the American GDP, we could select the Industrial Production Index, the Consumer Price Index and the 3-Month Treasury Bill interest rate as explicative variables, as these factors, together with the GDP itself, provide one of the best four-variable combinations which can be used in order to forecast the trend of the GDP. Consequently, the purpose of this paper is that of verifying whether the choice of these variables is suitable for the construction of a dynamic linear model aiming at providing forecasts for the American GDP.

The paper is structured as follows. In the first chapter we will present a general overview of Bayesian statistics, with the aim of describing the tools it provides for both inference and decision-making under conditions of uncertainty. In the second chapter we will illustrate the mathematical structure of the dynamic linear model. In the third chapter we will describe the procedures which can be utilised to derive the correct specification of a dynamic linear model for time series analysis purposes. The fourth chapter concludes the treatment with a practical application, where we will illustrate the procedure followed in order to identify a dynamic linear model for the American GDP.

Chapter 1

Bayesian statistics

In this chapter we will present an overview of the Bayesian methods for statistic analysis, which provide a complete set of tools for both inference and decision making in conditions of uncertainty. The main bibliographical references used for this topic are the Chapter *Bayesian Statistics*, by J. M. Bernardo, published in the volume *Probability and Statistics* of the *Encyclopedia of Life Support Systems* (Oxford, UK: UNESCO, 2003) and *Metodi MCMC nell'inferenza statistica* by Maria M. Barbieri (Centro d'Informazione e Stampa Universitaria, 1996).

1.1. Introduction

With the expression *Bayesian statistics* we indicate a subset of the field of statistics in which the evidence about the true state of the aspect we want to analyse is expressed in terms of degrees of belief or, more specifically, Bayesian probabilities. This interpretation of probability is what mainly distinguishes Bayesian methods from other statistical approaches, such as, for instance, the frequentist approach. One of the advantages of the Bayesian approach is constituted by its wider field of applicability, since it provides a solution to many of the problematic issues faced by conventional statistical methods. For example, by introducing the notion of prior distribution, Bayesian methods allow us to incorporate scientific hypotheses in our analysis, and the peculiar situation where only documented data provides us with acceptable information is dealt with as a particular case by *objective* Bayesian methods.

Coherently with the ordinary meaning of the word *probability*, the Bayesian paradigm conceives it as a *rational*, or better as a *conditional measure of uncertainty*. The aim of statistical inference about a specific parameter of interest is that of describing the modification in the uncertainty about its value on the basis of evidence, and the procedure to perform this modification is specified by the Bayes' theorem.

1.2. The Bayesian interpretation of probability

As we have already mentioned in the introduction, Bayesian statistics interprets probability as a measure of the uncertainty regarding the occurrence of a specific event, conditional on the available information and the accepted assumptions.

Thus, to introduce some notation, we will indicate with Pr($E|C$) the measure of the rational degree of belief in the occurrence of the event E given that the condition C is satisfied. It is important to emphasize that we never compute "unconditional" probabilities, because in any application there exists a condition, which is generally constituted by the set of the available data, the set of the assumptions concerning the process generating the data, and the relevant knowledge. In some particular situations, it is possible to estimate the probability of a certain event under given conditions by associating it to the relative frequency of comparable events under comparable conditions.

1.3. Exchangeability and representation theorem

The concept of *exchangeability* refers to the fact that the data set $\{x_1, \ldots, x_n\}$ consists of "homogeneous" observations, in the sense that the only thing that matters is their *value*, and not their *order* of appearance. Statistically speaking, the set of random vectors $\{x_1, \ldots, x_n\}$ is said to be "exchangeable" if and only if their joint distribution is invariant under permutations, and an infinite sequence $\{x_j\}$ of random vectors is exchangeable if all its finite subsequences are exchangeable. In particular, a typical example of an exchangeable sequence is constituted by a generic *random sample*. In fact, the *representation theorem* states that if we assume that a given set of observations is a subset of an exchangeable sequence, then it can be described as a random sample from some probability density function $\{p(x|\omega), \omega \in \Omega\}, x \in X$, depending on some parameter vector ω, defined as the limiting value as $n \to \infty$ of some function of the data. Of

course, the available information concerning the value of ω is also described by some probability distribution $p(\omega|C)$.

Thus, the concept of exchangeability can be considered as an extension of the notion of random sample. In fact, any i.i.d. random sample is exchangeable, since a generic probability distribution $p(x_1, \ldots, x_n|\omega)=\prod_{i=1}^{n} p(x_i|\omega)$ is invariant under permutations. Of course, when dealing with forecasting, the assumption of independence among the observations of a random sample would be of no use, for in such a case knowing x_n could not provide us with any information in order to predict x_{n+1}, thus this assumption only concerns the distribution of the observations conditional on the parameter ω.

Moreover, under conditions of exchangeability, the representation theorem has two main important consequences. The first one is the *existence theorem* for a probability distribution $p(\omega|C)$ on the parameter space Ω, and the second one is the formal definition of the parameter ω, which is identified as the limiting value, for $n\to\infty$, of some function $f(x_1, \ldots, x_n)$ of the observed sample.

1.4. The Bayesian paradigm

When doing a statistical analysis, one generally starts by suggesting a probability model $\{p(D|\omega), \omega\in\Omega\}$ with the aim of representing the probabilistic process which has generated the observations D, for some unknown value of the parameter ω. This implies the necessity of establishing a *prior* distribution $p(\omega|K)$ over the parameter space Ω, which describes the available knowledge K about the value of ω before obtaining any information coming from the data. If the probability model hypothesized is the correct one, then, after the data information D has become available it is possible to derive a *posterior* distribution $p(\omega|D, A, K)$ including all the knowledge about the value of the parameter ω that has been obtained from the observations. Such distribution can be easily derived from the Bayes' theorem as follows,

$$p(\omega|D, A, K)=\frac{p(D|\omega)\, p(\omega|K)}{\int_{\Omega} p(D|\omega)\, p(\omega|K)\, d\omega} \quad ,$$

where A indicates the assumptions made on the probability model. The application of this theorem in order to include in the model the information provided by the data justifies the qualification of *Bayesian* attributed to this paradigm. Of course, if the prior distribution attributes zero density to a

given value, such value will also have zero posterior density. Thus, a common assumption is that prior distributions must be strictly positive. If it was necessary to assign null probability to a given value of ω, one should appropriately restrict the parameter space Ω.

1.5. The learning process

In the Bayesian approach, the Bayes' theorem is systematically applied in order to implement a learning process where the available prior information is combined with the information derived from the data in order to obtain the posterior distribution. The computation of a generic posterior density is simplified by the possibility to express the Bayes' theorem in terms of proportionality, as follows:

$$p(\omega|D) \propto p(D|\omega)\, p(\omega) \quad ,$$

where we have omitted the acceptable assumptions A and the prior information K to simplify the notation. Indeed, the proportionality constant $[\int_{\Omega} p(D|\omega)\, p(\omega)\, d\omega]^{-1}$ can always be obtained thanks to the property that the integral of any probability density $p(\omega|D)$ must be equal to one. Thus, any posterior distribution can be completely specified by a *kernel*, which is a function $k(\omega)$ such that $p(\omega|D) = c(D)k(\omega)$ for some $c(D)$ which does not depend on ω.

A positive function $\pi(\omega)$ such that $\int_{\Omega} \pi(\omega)\, d\omega$ is not finite is defined as an *improper prior function*. The expression of the Bayes' theorem remains still valid in the cases in which $p(\omega)$ is an improper prior function, and it is then possible to obtain a *proper* posterior distribution $\pi(\omega|D) \propto p(D|\omega)\, \pi(\omega)$.

The *likelihood function*, defined as $l(\omega, D) = p(D|\omega)$, also plays a role in determining the posterior distribution of ω. In fact, if we assume that the same prior $p(\omega)$ is used, two different data sets D_1 and D_2 , coming from different probability models $p_1(D_1|\omega)$ and $p_2(D_2|\omega)$ but having proportional likelihood functions, will provide equal posterior distributions for ω. This property is also known as the *likelihood principle*.

Of course, the definitions of prior and posterior are relative to a specific data set, and if the data $D = \{x_1, \ldots, x_n\}$ are sequentially processed, so that the "posterior" distribution at a given time t becomes the "prior" at the next time, in accordance with the proportionality property that, for $i = 1, \ldots, n - 1$, the final result will be the same one would obtain if the data were globally processed.

In most cases, the posterior distribution is "sharper" than the prior, since the uncertainty about the true parameter value decreases after taking into account the observed data. However, there are

occasional situations in which an "unforeseen" observation may increase the parameter variance, instead of decreasing it.

For a given probability model, it is always possible to find some function of the data $t = t(D)$ such that t is a *sufficient* statistic, which means that $t(D)$ includes all information about ω which is contained in D. According to the formal definition, $t = t(D)$ is sufficient if and only if there exist non-negative functions f and g such that the likelihood function may be factorized in the form $p(D|\omega) = f(\omega|t)g(D)$. The simplest example of a sufficient statistic is, of course, $t(D) = D$. However, there are many situations in which it is possible to find a much simpler sufficient statistic, having a fixed dimensionality which does not depend on the sample size. This particularly happens when the probability model belongs to the *generalized exponential family*. It may be proved that if t is sufficient, then the posterior distribution of ω only depends on t itself, and may be calculated in terms of $p(t|\omega)$ as follows: $p(\omega|D) = p(\omega|t) \propto p(t|\omega)\, p(\omega)$.

It can be easily deduced that, for any given data set and model assumptions, different priors imply different posteriors, but it is important to evaluate the impact that a sensible change in the prior would induce in the posterior. This kind of evaluations constitutes what is commonly referred to as *sensitivity analysis*.

We will now consider the situation where the quantity of interest is not the parameter vector ω itself, but some function $\theta = \theta(\omega)$, possibly of lower dimension than ω. If we assume $\{p(D|\omega),\, \omega \in \Omega\}$ to be a probability model describing the statistical mechanism that has generated the data D, let $p(\omega)$ be the prior distribution of ω, describing the available information about its value, and let $\theta = \theta(\omega) \in \Omega$ be a function of the parameters over whose value inference has to be made, any deduction about the vector of interest θ will then be based on its posterior distribution $p(\theta|D)$, which will of course also depend on the accepted probability model $\{p(D|\omega),\, \omega \in \Omega\}$ and on the prior knowledge provided by $p(\omega)$. In fact, the posterior $p(\theta|D)$ can be derived by using standard probability calculus procedures. In particular, from the Bayes' theorem we obtain that $p(\omega|D) \propto p(D|\omega)\, p(\omega)$, and if we let $\lambda = \lambda(\omega) \in \Lambda$ be some function of ω such that $\psi = \{\theta, \lambda\}$ is a one-to-one transformation of the original parameter, and let $J(\omega) = (\partial \psi / \partial \omega)$ be the corresponding Jacobian matrix, it is possible to compute the posterior distribution of ψ by using the standard change-of-variable tool, as follows:

$$p(\psi|D) = p(\theta, \lambda|D) = \left[\frac{p(\omega|D)}{|J(\omega)|} \right]_{\omega = \omega(\psi)} ;$$

consequently, the posterior density of θ is the corresponding marginal distribution, obtained by integrating over the *nuisance parameter* λ, as follows:

$$p(\theta|D) = \int_{\Lambda} p(\theta, \lambda|D)\, d\lambda \quad .$$

In some cases it may be useful to restrict the range of possible values of ω. For example, if ω is known to belong to $\Omega_c \subset \Omega$, we will assume the prior distribution to be only positive in Ω_c, and the restricted posterior will be computed by exploiting the Bayes' theorem in the following way:

$$p(\omega|D, \omega \in \Omega_c) = \frac{p(\omega|D)}{\int_{\Omega_c} p(\omega|D)}, \qquad \omega \in \Omega_c .$$

Thus, in order to impose a constraint on the possible parameter values, it will be sufficient to renormalize the unrestricted posterior density to the set $\Omega_c \subset \Omega$ satisfying the required condition.

1.6. Predictive distributions

Let us suppose that $D = \{x_1, \ldots, x_n\}, x_i \in \chi$ is a set of exchangeable observations, and we want to predict the value of a future observation $x \in \chi$ generated by the same probabilistic mechanism that has produced the data set *D*. It can be naturally deduced that the solution to a problem of this kind is contained in the *predictive* distribution $p(x|D)$. If we assume that the data *D* constitute a random sample coming from a distribution in the family $\{p(x|\omega), \omega \in \Omega\}$, and that $p(\omega)$ is a prior density function describing the available information concerning the parameter ω, then, since $p(x|\omega, D) = p(x|\omega)$, it follows that the predictive density is given by $p(x|D) = \int_{\Omega} p(x|\omega)\, p(\omega|D)\, d\omega$, which corresponds to the average of the distributions of *x* conditional on the parameter ω, weighted with the posterior distribution of ω conditional on the data.

If the model is based on correct assumptions, the posterior predictive distribution $p(x|D)$ will converge to the distribution $p(x|\omega)$ generating the data, as the sample size increases. Thus, in order to evaluate the inferential capability of the posterior $p(\omega|D)$, it is possible to compare the results of a simulation produced by the predictive distribution $p(x|D)$ generated by $p(\omega|D)$ with the observed data.

1.7. Asymptotic behaviour

We will now consider the asymptotic behaviour of posterior distributions, whose importance is mainly due to two reasons:

1) when large sample sizes are available, it may be useful to exploit asymptotic results as approximations;
2) objective Bayesian methods typically depend on the asymptotic behaviour of the assumed model.

In formal terms, if we assume $D=\{x_1,\ldots,x_n\}, x\in\chi$ to be a random sample with n observations from $\{p(x|\omega),\omega\in\Omega\}$, it may be proved that, when $n\to\infty$, the posterior distribution $p(\omega|D)$ of a *discrete* parameter ω converges to a degenerate distribution where the true value of ω has probability one, and the posterior distribution of a *continuous* parameter ω converges to a normal distribution whose mean equals its *maximum likelihood estimate* $\hat{\omega}$ and whose variance decreases with the speed of the order of infinity $1/n$.

In particular, if we consider the case in which $\Omega=\{\omega_1,\omega_2,\ldots\}$ is constituted by a *countable* set of values, where the probability model corresponding to the true parameter ω_t is *distinguishable* from the others, which means that the discrepancy $\delta\{p(x|\omega_i),p(x|\omega_t)\}$ of each one of the $p(x|\omega_i)$ from $p(x|\omega_t)$ is strictly positive, it may be proved that

$$\lim_{n\to\infty} p(\omega_t|x_1,\ldots,x_n)=1 \quad , \qquad \lim_{n\to\infty} p(\omega_i|x_1,\ldots,x_n)=0 \quad , \quad i\neq t.$$

To express this in words, the posterior probability of the true parameter value converges to one as when the sample size tends to infinity.

Let us now take into account the case in which ω is a k-dimensional *continuous* parameter. If we express the Bayes' theorem as $p(\omega|x_1,\ldots,x_n)\propto\exp\left(\log[p(\omega)]+\sum_{j=1}^{n}\log[p(x_j|\omega)]\right)$ and expand $\sum_j \log[p(x_j|\omega)]$ about its maximum (the MLE $\hat{\omega}$), it can be shown that the posterior distribution of ω can be approximated by using a k-variate normal density function:

$$p(\omega|x_1,\ldots,x_n)\approx N_k\{\hat{\omega},S(D,\hat{\omega})\}, \qquad S^{-1}(D,\omega)=\left(-\sum_{l=1}^{n}\frac{\partial^2\log[p(x_l|\omega)]}{\partial\omega_i\partial\omega_j}\right) \quad .$$

It is also possible to use a simpler, even though poorer, approximation obtained by applying the law of large numbers to the sums contained in the previous expression. In this way we would obtain that $S^{-1}(D,\hat{\omega})\approx nF(\hat{\omega})$, where $F(\omega)$ is the *Fisher's information matrix*, whose general element is equal to

$$F_{ij}(\omega)=-\int_{\chi} p(x|\omega)\frac{\partial^2\log[p(x|\omega)]}{\partial\omega_i\partial\omega_j}dx \quad .$$

In conclusion, in the continuous case, assuming very general regularity conditions, the posterior probability density of the parameter vector ω converges, as the sample size grows, to the one of a multivariate normal density centred in the MLE $\hat{\omega}$, with a variance that decreases with n with the speed of $1/n$.

If we decompose the parameter vector into $\omega = (\theta, \lambda)$ and partition the Fisher's information matrix accordingly, obtaining:

$$F(\omega)=F(\theta,\lambda)=\begin{pmatrix} F_{\theta\theta}(\theta,\lambda) & F_{\theta\lambda}(\theta,\lambda) \\ F_{\lambda\theta}(\theta,\lambda) & F_{\lambda\lambda}(\theta,\lambda) \end{pmatrix} \quad ,$$

and

$$S(\theta,\lambda)=F^{-1}(\theta,\lambda)=\begin{pmatrix} S_{\theta\theta}(\theta,\lambda) & S_{\theta\lambda}(\theta,\lambda) \\ S_{\lambda\theta}(\theta,\lambda) & S_{\lambda\lambda}(\theta,\lambda) \end{pmatrix} \quad ,$$

then we will have that the marginal posterior distribution of θ will be given by

$$p(\theta|D)\approx N\{\theta|\hat{\theta}, n^{-1}S_{\theta\theta}(\hat{\theta},\hat{\lambda})\} \quad ,$$

and the posterior distribution of λ conditional on θ will be given by

$$p(\lambda|\theta, D)\approx N\{\lambda|\hat{\lambda}-F_{\lambda\lambda}^{-1}(\theta,\hat{\lambda})F_{\lambda\theta}(\theta,\hat{\lambda})(\hat{\theta}-\theta), n^{-1}F_{\lambda\lambda}^{-1}(\theta,\hat{\lambda})\} \quad .$$

In the situation where θ and λ are asymptotically independent, which would imply F to be block diagonal, we have that $F_{\lambda\lambda}^{-1}=S_{\lambda\lambda}$.

1.8. Bayesian estimation

When we are dealing with parameter vectors in one or two dimensions, a useful tool in order to convey an intuitive summary of what can be deduced about the quantity of interest is a graphical representation of its posterior distribution. Anyway, the main drawback of drawing one's conclusions uniquely from a graphical representation is that this is not easily feasible when we need to estimate parameters with more than two dimensions. Beyond this dimensionality, in fact, quantitative conclusions are often necessary.

Point estimation

If we assume D to be the available data sample, generated by a probability model $\{p(D|\omega), \omega\in\Omega\}$, and assume $\theta=\theta(\omega)\in\Theta$ to be the parameter of interest, some function $\tilde{\theta}=\tilde{\theta}(D)$ is defined a *point estimator* of θ if it can be considered as a proxy for the actual value of a θ. From a formal viewpoint, choosing a point estimate for θ can be regarded as a decision problem, where the action space corresponds to the parameter space Θ. In fact, the choice of a point estimate $\tilde{\theta}$ for some quantity θ determines the decision whether to act as if θ were equal to $\tilde{\theta}$ or not. Coherently with decision theory, to solve this decision problem we need to define a *loss* function $l(\tilde{\theta},\theta)$ with the aim of measuring the loss one could suffer if he acted as if $\tilde{\theta}$ were the true value of the parameter of interest, when it is instead equal to θ. Such function is computed as

$$l[\tilde{\theta}|D]=\int_{\Theta} l(\tilde{\theta},\theta)\,p(\theta|D)\,d\theta \quad ,$$

and the corresponding Bayesian estimator $\hat{\theta}$ is that function of the data, $\hat{\theta}=\hat{\theta}(D)$, which minimizes this expectation.

Region estimation

In order to get an intuition of the posterior distribution of the quantity of interest $p(\theta|D)$ it may be useful to identify regions $R\subset\Theta$ of specified probabilities under $p(\theta|D)$. Any region R_q such that

$$R_q\subset\Omega, \qquad \int_{R_q} p(\theta|D)\,d\theta=q, \qquad 0<q<1,$$

is defined as the *q-credible region* of θ. A credible region is invariant under reparametrization, which means that, for any q-credible region R_q of θ, $\phi(R_q)$ is a q-credible region of $\varphi = \varphi(\theta)$. Of course, for any given value of q there exist generally infinitely many credible regions.

The notion of credible region for a function $\theta = \theta(\omega)$ of the parameter vector can be easily extended to prediction problems, and in this case we will define as a posterior q-credible region for $x\in X$ any subset R_q of the outcome space X with posterior predictive probability q, so that $\int_{R_q} p(x|D)\,dx=q$.

1.9. Hypothesis testing

By analysing the posterior distribution $p(\theta|D)$ of the parameter of interest θ, we can get an immediate intuition about those values having a relatively high probability density, which may be considered *compatible* with the set of observations D. In several situations, in fact, we may identify a restriction $\theta\in\Theta_0\subset\Theta$ of the possible values of the quantity of interest which deserves a special consideration. Thus, the *hypothesis* $H_0\equiv\{\theta\in\Theta_0\}$ should be accepted as *compatible* with the observed data D if the subset Θ_0 contains values with a relatively high posterior density. From a formal viewpoint, testing the hypothesis $H_0\equiv\{\theta\in\Theta_0\}$ is a decision problem with an action space including only two elements: to accept (a_0) or to reject (a_1) the restriction considered. A decision problem of this kind can be solved by defining an appropriate loss function, $l(a_i,\theta)$, measuring the consequences of accepting or rejecting H_0 as a function of the actual value θ of the parameter of interest. The decision a_1 is defined as an *alternative* to a_0 , thus it is not necessarily the best possible decision, but it is better than anything else that has been imagined.

If we consider the loss function $l(a_i,\theta)$ for a given action a_i , we obtain that the optimal decision will be rejecting H_0 if and only if the expected posterior loss deriving from accepting the

null hypothesis, $\int_{\Theta} l(a_0,\theta)\,p(\theta|D)\,d\theta$, is larger than the expected posterior loss deriving from rejecting the null, $\int_{\Theta} l(a_1,\theta)\,p(\theta|D)\,d\theta$, which is equivalent to saying, if and only if $\int_{\Theta}\left[l(a_0,\theta)-l(a_1,\theta)\right]p(\theta|D)\,d\theta=\int_{\Theta}\Delta l(\theta)\,p(\theta|D)\,d\theta>0$.

As a consequence, we only need to specify the loss difference $\Delta l(\theta)=l(a_0,\theta)-l(a_1,\theta)$, representing the advantage deriving from the rejection of H_0 as a function of the parameter θ. In particular, if this difference is positive, one should reject the null H_0 .

It is important to emphasize that, while accepting a_0 by assumption means to act as if H_0 were true, that is to act as if $\theta\in\Theta_0$ in our case, there exist at least two possible interpretations of the alternative choice a_1 . In fact, this may either mean the negation of H_0 , which would imply that $\theta\notin\Theta_0$, or, differently, that the restriction proposed by the null hypothesis is rejected, and we keep the unrestricted model whose parameter space includes all $\theta\in\Theta$. Generally, however, hypothesis testing techniques are more suitable in the second case. Indeed, a restricted model where the null hypothesis $H_0\equiv\{\theta\in\Theta_0\}$ is realized is often embedded into a more general model, $\{\theta\in\Theta\,,\,\Theta_0\subset\Theta\}$ built in order to include the possibility of departures from H_0 , and the purpose of our analysis would be to verify whether the available data *D* present some compatibility with the hypothesis under which $\theta\in\Theta_0$.

1.10. Approximation techniques in Bayesian inference

It frequently happens that difficulties may be encountered in obtaining analytical solutions to inferential problems, which often leads to the adoption of simplification procedures, such as the usage of probability distributions belonging to the exponential family, or assumptions of linear dependence. Such a circumstance caused the necessity to develop some techniques which might be used as an alternative to analytical calculus, in the cases in which the model considered is particularly complex. All of the proposed techniques could be classified in one of the following three categories:

1) ***analytical approximation methods***;
2) ***efficient numerical integration methods***;
3) ***simulation-based methods***.

Generally, analytical approximation methods are the ones whose application is the most straightforward, even though they are often unable to provide particularly accurate solutions.

Numerical integration methods possess the drawback that they do not take into account the fact that the functions involved in the optimization or integration procedure we want to perform are related to probabilistic laws. On the contrary, simulation-based methods make inference about the parameters of interest on the basis of simulated numerical values, without considering the analytical properties of the functions involved. Numerical methods generally lead to reliable results when the function of interest has a regular shape and the size of the problem is quite reduced, while in complex situations the simulation-based approach is preferable.

Recently, a different paradigm was developed, which uses a sample of values obtained from a finite realization of a Markov chain in order to provide the required inferences. If the utilized method satisfies regularity conditions, the generated sample may be used for inferential purposes as if it was an independent sample. The methods belonging to this category, commonly known as **Markov Chain Monte Carlo** (**MCMC**) provide a tool for the analysis of extremely complex models, and their usage is generally quite simple and flexible.

In the last few years, MCMC methods became particularly popular thanks to their applications in the field of Bayesian inference, since they provide a solution to problems where the computation of complex integrals is needed in order to estimate the parameters of interest.

1.10.1. Markov chains

A discrete parameter stochastic process $\{Z^t; t=0,1,2,...\}$, whose values are comprised in a set *Z*, is said to be a *Markov chain* if and only if the conditional distribution of Z^t , given the values $z^0, z^1, ..., z^{t-1}$ previously presented by the process, is only dependent on z^{t-1} . Formally, a process can be defined as such if

$$Pr\{Z^t \in A | Z^0 = z^0, Z^1 = z^1, ..., Z^{t-1} = z^{t-1}\} = Pr\{Z^t \in A | Z^{t-1} = z^{t-1}\}$$

for any measurable $A \subseteq Z$.

This property is commonly referred to as *Markovianity* and expresses the circumstance that the future is independent on the past, conditional on the present. Each one of the values presented by the process at a given point in time is defined to be a *state* occupied by the chain, and the set *Z* is said to be *state space*.

In order to define a Markov chain it is necessary to define the probability distribution of Z^t conditional on Z^{t-1} , also called *transition probability* distribution.

1.10.2. Markov Chain Monte Carlo algorithms

As we have previously explained, in the real world it happens quite frequently that the posterior distribution of a parameter of interest is not easy to be determined. In order to face this issue, MCMC methods are commonly utilised. In fact, if we suppose that the posterior distribution of a given parameter is equal to π and we draw from it a random sample which we name $\psi_1,...,\psi_N$ (i.i.d.), then, by means of the Monte Carlo method, it is possible to estimate the mean of any given function $g(\psi)$ with a finite expected value of the posterior, by using the following approximation:

$$E_\pi(g(\psi)) \approx N^{-1}\sum_{j=1}^{N} g(\psi_j).$$

Even if independent samples from the posterior of the parameters cannot be easily obtained, this approximation method generally works quite effectively in the cases of dependent samples, and in particular for Markov chains.

If we consider a given Markov chain $\{\psi_t\}_{t\geq 1}$ distributed according to an invariant probabilistic law π, it is possible to show that for any initial value ψ_1 the distribution of ψ_t tends to π when t tends to infinity. In particular, for M large enough, the parameters $\psi_{M+1},...,\psi_{M+N}$ follow the distribution of π and their statistical properties are comparable to those possessed by an independent sample from π. It is useful to notice how, even if the Markov chain is not aperiodic, the law of large numbers mentioned above still applies.

Moreover, if we assume that ψ_1 is distributed according to π, we are allowed to use the traditional formula for the estimation of the variance to assess the amount of uncertainty associated to the calculation of the ergodic averages. If we denote with $\bar{g}_N$ the right-hand side of the equation reported before, we obtain that

$$Var(\bar{g}_N) \approx N^{-1}Var(g(\psi_1))\tau(g)$$

with $\tau(g) = \sum_{t=-\infty}^{+\infty}\rho_t$ and $\rho_t = corr(g(\psi_s), g(\psi_{s+t}))$. Thus, we can estimate the quantity $Var(g(\psi_1))$ by using the sample variance of $g(\psi_1),...,g(\psi_N)$. In order to provide an estimate of *τ(g)*, we must substitute theoretical correlations with empirical correlations for all of the N iterations, as follows:

$$\hat{\tau}_n = \sum_{|t|\leq n}\hat{\rho}_t .$$

In the paragraphs which follow we will illustrate the most frequently used MCMC algorithms for the simulation of ergodic means from a theoretical distribution π.

1.10.3. Gibbs sampling

There are many situations where the parameter whose distribution we are willing to determine is multidimensional, and consequently we have to derive a sample from a multivariate distribution. In such a case we depart from the expression $\psi = (\psi^{(1)}, \psi^{(2)})$ where $\psi^{(1)}$ and $\psi^{(2)}$ may be both one-dimensional and multidimensional. Furthermore, we assume that the target density is given by $\pi(\psi) = \pi(\psi^{(1)}, \psi^{(2)})$. The Gibbs sampling procedure starts from an arbitrary state denoted by $\psi_0 = (\psi_0^{(1)}, \psi_0^{(2)})$, and then alternates the updating of the two single parameters by simulating from the respective conditional distributions. We can summarise the algorithms in the following five steps.

1. Set $j = 1$.
2. Draw $\psi_j^{(1)}$ from the conditional density $\pi(\psi^{(1)} \left| \psi^{(2)} = \psi_{j-1}^{(2)} \right.)$.
3. Draw $\psi_j^{(2)}$ from the conditional density $\pi(\psi^{(2)} \left| \psi^{(1)} = \psi_j^{(1)} \right.)$.
4. Set $j = j + 1$.
5. If $j \leq N$ go back to step 2, otherwise stop.

1.10.4. Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm allows us to generate a simulation of the next state of a Markov chain departing from an arbitrary density function, and the invariance of the distribution is guaranteed by a step at which we are forced to decide whether to accept or to reject the result of the simulation itself.

Let us assume that the current state of a given chain is equal to ψ. We define *proposal distribution* the conditional probability of proposing a state $\tilde{\psi}$ given that we are in the state ψ, which we denote with $q(\psi, \cdot)$. Then we define *acceptance distribution* the conditional probability of accepting the proposed state $\tilde{\psi}$ given that we are in the state ψ, which we denote with $\alpha(\psi, \tilde{\psi})$. The proposed value will be accepted with a probability given by:

$$\alpha(\psi,\tilde{\psi}) = \min\left\{1, \frac{\pi(\tilde{\psi})q(\tilde{\psi},\psi)}{\pi(\psi)q(\psi,\tilde{\psi})}\right\}$$

where the numerator $\pi(\tilde{\psi})q(\tilde{\psi},\psi)$ indicates the probability that we are in the state $\tilde{\psi}$ and we pass to the state ψ, while the denominator $\pi(\psi)q(\psi,\tilde{\psi})$ indicates the probability that we are in the state ψ and we pass to the state $\tilde{\psi}$

Of course, the simulated parameter will be always accepted if the acceptance distribution presents a value larger than 1, and will be rejected accordingly to the chosen confidence level if this value is smaller than 1.

The choice of the proposal density and of the related rejection level is quite important for the performance of the Markov chain. In fact, a proposal distribution implying a high rejection level will lead to a higher rigidity of the chain, which means that the state will tend to remain at a constant level for a larger number of iterations. The disadvantage of such a situation is that the computation of ergodic averages by means of the formula provided previously will not probably result in satisfying approximations. On the contrary, if the distribution implies a low rejection level, this may result in a chain which does not correctly follow the target density.

The whole procedure of the Metropolis-Hastings algorithm is summarised in the following steps.

1. Set $j = 1$.
2. Draw $\tilde{\psi}_j$ from the proposal distribution $q(\psi_{j-1},\cdot)$.
3. Calculate $\alpha = \alpha(\psi_{j-1},\tilde{\psi}_j)$.
4. Draw an independent sample from the random variable $U_j \sim \text{Ber}(\alpha)$.
5. If $U_j = 1$ set $\psi_j = \tilde{\psi}_j$, otherwise set $\psi_j = \psi_{j-1}$.
6. Set $j = j + 1$.
7. If $j \leq N$ go back to step 2, otherwise stop.

Chapter 2

The dynamic linear model

In this chapter we will describe and analyse the mathematical structure of the general class of dynamic linear models, with the goal of providing a basis for the understanding of the special cases and generalisations that will be presented in the subsequent chapters. The main bibliographical source we referred to for what concerns this topic is *Bayesian Forecasting and Dynamic Models* by M. West and J. Harrison (Springer-Verlag New York, Inc., 1997).

2.1. Introduction

Dynamic linear models, commonly referred to as *DLMs*, constitute the most widely used subclass of *dynamic models*, where the word *dynamic* means that the parameters of such models are automatically modified in consequence of changes caused by the passage of time. In fact, with the evolving of time, the modeller is able to take into account information which was not available in the past, and can use it to revise his forecasts, whether such revision involve modifications at the quantitative, formal or conceptual level, which we shortly define hereafter.

The quantitative level involves the values assigned to the parameters, the formal level regards the selection of the appropriate variables and the definition of relationships between them, and the conceptual level provides an abstract view of the model describing for example the scientific or socio-economic laws and the behavioural features the model itself is based upon. The conceptual basis of a model is expected to be very stable and rarely modified, while the quantitative aspect is the one for which changes occur most frequently.

In order to exemplify the dynamic updating process, we suppose that $t = 0$ represents the present time, and the available information at that time is denoted by D_0. A forecaster will produce his views about the future time $t > 0$ on the basis of such information, by means of studying conditional distributions of the kind $Y_t|D_0$. As time evolves and t becomes the present time our existing information will be denoted by D_t. Consequently, when making projections at time t about any quantity of interest, a forecaster will have to take into account the information available at time t; for example, in order to produce a forecast concerning time $s > t$ we will need to consider the distribution of $Y_s|D_t$.

If we focus on the information available at a generic time t, we can see that it is composed of the information which already existed at time t -1 and the amount of information which was only acquired at time t and includes the observation of Y_t, which we indicate by I_t. In order to express this with formulas, we can write:

$$D_t = \{I_t, D_{t-1}\} \text{ for every } t = 1, 2, ...$$

Thus, forecasting will be performed thanks to the use of conditional distributions based on the knowledge of past information and on the estimation of specific parameters, which we will denote by θ_t. If we consider a forecast concerning the variable Y_t one step ahead, we can state that the beliefs of the forecaster are based on a parametric probability model of this kind:

$$p(Y_t|\theta_t, D_{t-1}).$$

This representation will allow decision makers to choose the behaviour that will lead to a favourable result with the highest probability.

The indexing of θ_t by t indicates that such parameter is subject to a dynamic updating process. Furthermore, though in the majority of cases the number and the meaning of the parameters are constant over time, there are situations in which such number will be modified or the variables will be changed in consequence of updates in the information about the actual time series. This is particularly frequent in the case of open systems, such as social, economic and biological environments, where the state of the system itself affects the factors influencing the time series process. In such cases, variations in the parameters are often necessary in order to reflect the effects of the system learning, but, since it is generally difficult to foresee such effects, modellers usually include such changes only after they have caused remarkable results in the system.

Thus, we can state that the learning process allows modellers to continually revise the state of their knowledge as regards the parameters. In fact, the Bayesian approach, consisting in a probabilistic representation of reality, concerns both observable quantities of interest, such as

consumption or the Gross Domestic Product, and unobservable quantities such as the parameters of a forecasting model.

For example, at time t the forecaster's knowledge can be summarised by a so called *prior* distribution of the future parameters, where the term *prior* indicates that such knowledge precedes the observation of Y_t. We indicate such distribution by $p(\theta_t|D_{t-1})$.

Of course, there will also be a *posterior* distribution, which takes into account the observation of Y_t and all of the information available at time t, which we denote by $p(\theta_t|D_t)$.

These two distributions symbolise the way in which the model improves its parameters over time thanks to the acquisition of new information. In particular, inference about the future developments of the analysed system is derived from the knowledge of such distributions by means of a standard statistical approach.

We present hereafter the formal definition of the simplest form of dynamic linear model.

Definition 2.1. For every t, we define univariate, uniparameter normal dynamic linear model, indicated by the quadruple $\{F_t, \lambda_t, V_t, W_t\}$ the set of the following expressions:

Observation equation:	$Y_t = F_t \mu_t + v_t,$	$v_t \sim \mathrm{N}[0, \mathrm{V_t}],$
System equation:	$\mu_t = \lambda \mu_{t-1} + \omega_t,$	$\omega_t \sim \mathrm{N}[0, \mathrm{W_t}],$
Initial information:	$(\mu_0 \mid D_0) \sim \mathrm{N}[\mathrm{m_0}, \mathrm{C_0}],$	

where the error terms v_t and ω_t are both singularly and mutually independent, and are also independent of $\mu_0|D_0$. λ identifies a known constant and F_t is a known sequence of values. Differently, the variance sequences V_t and W_t might be unknown.

2.2. The first-order polynomial model

We believe it is important to introduce dynamic linear models by examining firstly their simplest and most widely used example, which is the **first-order polynomial model**.

In this DLM model the observed variable Y_t is represented as follows:

$$Y_t = \mu_t + v_t, \qquad \text{with } v_t \sim \mathrm{N}[0, \mathrm{V_t}],$$

where μ_t is referred to as the **level** of the variable at time *t* and v_t denotes the **observational error**. The evolution of the level over time can be simply modelled as a random walk expressed by the process which follows:

$$\mu_t = \mu_{t-1} + \omega_t, \qquad \text{with } \omega_t \sim \mathrm{N}[0, \mathrm{W_t}],$$

where ω_t is called **evolution error**.

We also assume that for all *t* and *s* such that $t \neq s$, v_t is independent of v_s, ω_t is independent of ω_s, v_t and ω_s are mutually independent. In the initial phase, we will also assume that the variances V_t and W_t are known for every time *t*. The conditional distributions of the observational and evolution equation are expressed as follows:

$$\begin{aligned} &(Y_t|\mu_t) \sim \mathrm{N}[\mu_t, \mathrm{V_t}], \\ &(\mu_t|\mu_{t\text{-}1}) \sim \mathrm{N}[\mu_{t\text{-}1}, \mathrm{W_t}]. \end{aligned} \qquad \text{for every } t = 1, 2, \ldots$$

At an intuitive level, the mean μ_t may be considered as a function of time with a smoothing effect, which can be represented by the following expression:

$$\mu(t + \delta t) = \mu(t) + \text{higher-order terms},$$

where the higher-order terms identify the white-noise distribution of the error term.

As it can be seen from the equations presented, in the DLM model the level is a locally constant approximation for the evolution underlying the fluctuations of the variable of interest. The expected value *k*-steps ahead of a given series conditionally to its present value is calculated as follows:

$$E[Y_{t+k}|\mu_t] = E[\mu_{t+k}|\mu_t] = \mu_t.$$

Consequently, at time *t*, conditionally to the available information D_t, the posterior distribution of μ_t will have a mean equal to m_t, and the forecast function $f_t(\cdot)$ will be constant and given by:

$$f_t(k) = E[Y_{t+k}|D_t] = E[\mu_t|D_t] = m_t$$

for every $k > 0$. We can deduce that the use of this kind of DLM can only provide good results for short-term forecasts, and particularly when the variability of the single observation, expressed by V_t is notably higher than the variability at the systematic level, indicated by W_t.

2.3. Definition and updating equations

According to Definition 2.1, the first-order polynomial model is identified by the quadruple $\{1,1,V_t,W_t\}$ and defined by the following set of equations:

Definition 2.2. For every *t*, the DLM expressed by the quadruple $\{1,1,V_t,W_t\}$ is defined by:

Observation equation:	$Y_t = \mu_t + v_t$	with $v_t \sim \mathrm{N}[0, \mathrm{V_t}]$,
System equation:	$\mu_t = \mu_{t-1} + \omega_t$	with $\omega_t \sim \mathrm{N}[0, \mathrm{W_t}]$,
Initial information:	$(\mu_0 \mid D_0) \sim \mathrm{N}[\mathrm{m_0}, \mathrm{C_0}]$,	

where we assume that the error terms v_t and ω_t are independent of time *t*, mutually independent and also independent of the conditional mean $\mu_0 \mid D_0$, whose distribution represents all the available information at the initial time, and the variance is a measure of the corresponding uncertainty.

We present hereafter the prior and posterior distribution of the level and the distribution of the one-step-ahead forecast for this kind of model.

1) Posterior for μ_{t-1}:	$(\mu_{t-1} \mid D_{t-1}) \sim \mathrm{N}[\mathrm{m_{t-1}}, \mathrm{C_{t-1}}]$
2) Prior for μ_t:	$(\mu_t \mid D_{t-1}) \sim \mathrm{N}[\mathrm{m_{t-1}}, \mathrm{R_t}]$
3) Posterior for μ_t:	$(\mu_t \mid D_t) \sim \mathrm{N}[\mathrm{m_t}, \mathrm{C_t}]$
4) 1-step-ahead forecast:	$(Y_t \mid D_{t-1}) \sim \mathrm{N}[\mathrm{f_t}, \mathrm{Q_t}]$

where we make the following assumptions:

- $R_t = C_{t-1} + W_t$
- $f_t = m_{t-1}$
- $Q_t = R_t + V_t$
- $m_t = m_{t-1} + A_t e_t$
- $C_t = A_t V_t$

- $A_t = R_t / Q_t$
- $e_t = Y_t - f_t$.

In particular, the distribution of the posterior for μ_t can be demonstrated through the application of the standard theory of bivariate normal distribution, and we present such proof hereafter.

We depart from the consideration that any linear function involving Y_t and μ_t can be expressed as a linear combination of the independent normal distributions of v_t, ω_t and μ_{t-1}; thus, any linear function such as this possesses a normal distribution conditionally on μ_{t-1}. As a consequence we obtain that the joint distribution $Y_t, \mu_t | D_{t-1}$ is a bivariate normal. By taking into account the expressions in 2) and 4) and the independence of μ_t and v_t we are able to calculate the covariance between our two normal quantities, and we obtain:

$$C[Y_t, \mu_t | D_{t-1}] = V[\mu_t | D_{t-1}] = R_t .$$

We deduce that the joint distribution is given by:

$$\begin{pmatrix} Y_t \\ \mu_t \end{pmatrix} \Big| D_{t-1} \sim \mathrm{N}\left[\begin{pmatrix} \mathrm{m_{t-1}} \\ \mathrm{m_{t-1}} \end{pmatrix}, \begin{pmatrix} Q_t & R_t \\ R_t & R_t \end{pmatrix} \right].$$

In order to derive the distribution of the posterior for μ_t it is necessary to apply the following property of the bivariate normal: if X_1 and X_2 are two normal random variables, whose respective means are μ_1 and μ_2, whose respective standard deviations are σ_1 and σ_2, and whose correlation coefficient is equal to ρ, the distribution of X_1 conditional on X_2 is given by:

$$X_1 | X_2 = x_2 \sim \mathrm{N}(\mu_1 + \frac{\sigma_1}{\sigma_2} \rho(\mathrm{x}_2 - \mu_2), (1 - \rho^2)\sigma_1^2) .$$

From this formula we derive that:

$$\mu_t | Y_t, D_{t-1} \sim \mathrm{N}[\mathrm{m_t}, \mathrm{C_t}]$$

where

$$m_t = m_{t-1} + \rho_t^2 (Y_t - m_{t-1})$$

and

$$C_t = (1 - \rho_t^2) R_t = R_t V_t / Q_t = A_t V_t$$

where A_t corresponds both to the squared correlation and to the regression coefficient of μ_t on Y_t.

2.4. General theorem of the DLMs

The general, univariate, dynamic linear model is exactly specified by the following two equations:

$$Y_t = F_t'\theta_t + v_t, \text{ with } v_t \sim \mathrm{N}_m(0, V_t)$$

$$\theta_t = G_t\theta_{t-1} + w_t, \text{ with } w_t \sim \mathrm{N}_p(0, W_t),$$

where we assume that both F_t and G_t are known matrices and that the error terms v_t and w_t are mutually independent white noise random variables, with zero mean and covariance matrices V_t and W_t respectively. The first equation is called the **observational equation**, while the second one is named the **state equation** or **system equation**. A further assumption we make is that θ_0 presents a Gaussian distribution,

$$\theta_0 \sim \mathrm{N_p}(m_0, C_0),$$

where m_0 is the vector of the mean values of the independent variables involved and C_0 is the covariance matrix. θ_0 is also assumed to be independent of v_t and w_t. It is easily visible that the general DLM satisfies the conditions corresponding to the features of the first-order polynomial model, which are $Y_t|\theta_t \sim \mathrm{N}(\mathrm{F_t}\theta_t, \mathrm{V_t})$ and $\theta_t|\theta_{t-1} \sim \mathrm{N}(\mathrm{G_t}\theta_{\mathrm{t\text{-}1}}, \mathrm{W_t})$.

Differently from the general dynamic linear model, the general state space model can be otherwise expressed in the form:

$$Y_t = h_t(\theta_t, v_t)$$
$$\theta_t = g_t(\theta_{t-1}, w_t)$$

where h_t and g_t are two given functions. The advantage of using a model such as this is its higher flexibility, because the utilisation of a DLM implies that we are assuming the functions h_t and g_t to be linear, and the data to have a Gaussian distribution. A possible drawback deriving from the use of a general state space model is the fact that removing the Normality assumption may cause additional computational difficulties.

Definition 2.3. We define hereby two particular subclasses of the DLMs.

1. A dynamic linear model is referred to as a **time series DLM**, or **TSDLM** if the pair of matrices $\{F, G\}_t$ is constant for all values of t.
2. A TSDLM having constant observation and evolution variances for all values of t is called a **constant TSDLM**.

Therefore, a constant DLM can be completely specified by a single quadruple of the kind

$$\{\mathbf{F, G, V, W}\}.$$

Any general univariate DLM is thus completely defined by the quadruple:

$$\{F_t, G_t, V_t, W_t\},$$

which leads to the following distributions for the observation

$$(Y_t|\theta_t) \sim \mathrm{N}[F_t'\theta_t, V_t]$$

and for the level

$$(\theta_t|\theta_{t-1}) \sim \mathrm{N}[G_t\theta_{t-1}, W_t].$$

From these distributions, we can derive the full definition of the model.

Observation equation:	$Y_t = F_t'\theta_t + v_t,$	$v_t \sim \mathrm{N_m}(0, V_t),$
System equation:	$\theta_t = G_t\theta_{t-1} + w_t,$	$w_t \sim \mathrm{N_p}(0, W_t),$
Initial information:	$(\theta_0\|D_0) \sim \mathrm{N}[m_0, C_0]$	

2.5. Updating equations: the univariate DLM

In order to simplify, we assume that at any time DLMs are closed in respect to external information, which means that given the initial information D_0 available at time $t = 0$, for any future time $t \geq 1$ the available amount of information is simply given by:

$$D_t = \{Y_t, D_{t-1}\}.$$

We also assume that D_0 incorporates in itself the knowledge of the quadruples $\{F_t, G_t, V_t, W_t\}$ for every t. The results of the analysis of such model are as follows.

1) Posterior at time t -1:

$(\theta_{t-1}|D_{t-1}) \sim \mathrm{N}[\mathrm{m}_{\mathrm{t-1}}, \mathrm{C}_{\mathrm{t-1}}]$

2) Prior at time *t*:

$(\theta_t|D_{t-1}) \sim \mathrm{N}[\mathrm{a}_{\mathrm{t}}, \mathrm{R}_{\mathrm{t}}]$

3) Posterior at time *t*:

$(\theta_t|D_t) \sim \mathrm{N}[\mathrm{m}_{\mathrm{t}}, \mathrm{C}_{\mathrm{t}}]$

4) One-step-ahead forecast:

$(Y_t|D_t) \sim \mathrm{N}[\mathrm{f}_{\mathrm{t}}, \mathrm{Q}_{\mathrm{t}}]$

where

- $a_t = G_t m_{t-1}$
- $R_t = G_t C_{t-1} G_t' + W_t$
- $f_t = F_t' a_t$
- $Q_t = F_t' R_t F_t + V_t$
- $m_t = a_t + A_t e_t$
- $C_t = R_t - A_t Q_t A_t'$
- $A_t = R_t F_t Q_t^{-1}$
- $e_t = Y_t - f_t$.

In order to prove statement 3) concerning the distribution of the posterior it is possible to utilise some theoretical rules applying in cases of normality of all distributions.

In particular, the system equation $\theta_t = G_t \theta_{t-1} + w_t$, where $w_t \sim \mathrm{N}_{\mathrm{p}}(0, W_t)$, and the prior distribution $(\theta_{t-1}|D_{t-1}) \sim \mathrm{N}[\mathrm{m}_{\mathrm{t-1}}, \mathrm{C}_{\mathrm{t-1}}]$ immediately lead to the conclusion that $(\theta_t|D_{t-1}) \sim \mathrm{N}[\mathrm{a}_{\mathrm{t}}, \mathrm{R}_{\mathrm{t}}]$.

The observation equation $Y_t = F_t' \theta_t + v_t$, where $v_t \sim \mathrm{N}_{\mathrm{m}}(0, V_t)$, jointly with the distribution of the prior lead to the conclusion that $(Y_t|D_{t-1}) \sim \mathrm{N}[F_t' a_t, F_t' R_t F_t + V_t]$ and consequently demonstrate that the joint distribution of Y_t and θ_t is Gaussian.

Moreover, we calculate the covariance between Y_t and θ_t, given by:

$$C[Y_t, \theta_t | D_{t-1}] = C[\theta_t, F_t'\theta_t + v_t | D_{t-1}] = V[\theta_t | D_{t-1}]F_t + 0 = R_t F_t = A_t Q_t .$$

On the basis of these results, we can derive the joint distribution of Y_t and θ_t.

$$\begin{pmatrix} \theta_t \\ Y_t \end{pmatrix} \Big| D_{t-1} \sim N\left[\begin{pmatrix} a_t \\ f_t \end{pmatrix}, \begin{pmatrix} R_t & A_t Q_t \\ Q_t A_t' & Q_t \end{pmatrix} \right].$$

By knowing that the regression vector of θ_t on Y_t is A_t, and by applying the standard normal theory in the same way we illustrated when dealing with the first order polynomial model, it is easily verifiable that the distribution of the posterior is the one presented in 3).

2.6. State estimation and forecasting

As in any statistical application, in the case of DLMs the most important step is constituted by the model specification; notwithstanding this, in order to simplify, in the following paragraphs we will focus on the phases involving estimation and forecasting, and we will therefore assume that the model is given, which means that the density functions $f(y_t|\theta_t)$ and $p(\theta_t|\theta_{t-1})$ have been correctly specified.

In order to solve problems involving statistical inference and prediction of future observations, we have to calculate the conditional distributions of the quantities of interest, departing from a known data sequence.

In particular, in order to estimate the state vector we have to calculate the conditional densities of the type: $p(\theta_s | y_1, ..., y_t)$. Such issues are usually distinguished in the following categories:

- problems of *filtering* (when $s = t$),
- problems of *state prediction* (when $s > t$),
- problems of *smoothing* (when $s < t$).

We will firstly analyse in detail the procedures which are commonly used in order to deal with problems of filtering, where our goal is to estimate the current value of the state vector based on the available information up to time *t*, and to update our estimates and predictions every time new observations become available. In the case of DLMs, the Kalman filter provides a procedure to pass

from the density $p(\theta_t|y_1,...,y_t)$ to $p(\theta_{t+1}|y_1,...,y_{t+1})$, which is equivalent to updating the current estimate on the state vector as new data are available. If a researcher is interested in estimating the evolution of the system *k* steps ahead from time *t*, which we assume to be the present time, he must compute the *k*-steps-ahead predictive density of the observations, $f(y_{t+k}|y_1,...y_t)$, on the basis of the *k*-steps-ahead predictive density of the level, $p(\theta_{t+k}|y_1,...,y_t)$.

We will show hereafter how to compute predictive densities by means of recursive procedure.

2.7. Filtering

As we have seen, one of the advantages of state space models is that the filtered and predictive densities can be computed by means of a recursive algorithm. For example, by departing from $\theta_0 \sim p_0(\theta_0) = p(\theta_0|D_0)$ it is possible to calculate recursively:

1. the one-step-ahead predictive density of θ_0 conditional on D_{t-1}, on the basis of the filtering density $p(\theta_{t-1}|D_{t-1})$ and of the DLM general theorem;
2. the one-step-ahead predictive density for the next observation;
3. the filtering density $p(\theta_t|D_t)$ by using the prior density $p(\theta_t|D_{t-1})$ and the likelihood function $f(y_t|\theta_t)$.

In particular, the recursive filtering procedure is explained hereafter.

1. The one-step-ahead predictive density for the level is computable as follows:

$$p(\theta_t|D_{t-1}) = \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|D_{t-1})dv(\theta_{t-1}).$$

2. The one-step-ahead predictive density for the observations is computable as follows:

$$f(y_t|D_{t-1}) = \int f(y_t|D_{t-1})p(\theta_t|D_{t-1})dv(\theta_t)$$

3. By applying the Bayes' theorem, the filtering density can be calculated starting from the densities reported above in the following way:

$$p(\theta_t|D_t) = \frac{f(y_t|\theta_t)p(\theta_t|D_{t-1})}{f(y_t|D_{t-1})}.$$

The proof departs from the conditional independence relationship $\theta_{t+1} \perp (Y_1,...,Y_t)|\theta_t$, from which we obtain:

$$p(\theta_t|D_{t-1}) = \int p(\theta_{t-1},\theta_t|D_{t-1})dv(\theta_{t-1}) = \int p(\theta_t|\theta_{t-1},D_{t-1})\,p(\theta_{t-1}|D_{t-1})dv(\theta_{t-1})$$
$$= \int p(\theta_t|\theta_{t-1})\,p(\theta_{t-1}|D_{t-1})dv(\theta_{t-1})$$

Then, from the conditional independence $Y_t \perp (Y_1,...,Y_{t-1})|\theta_t$ we derive:

$$f(y_t,\theta_t|D_{t-1})dv(\theta_t) = \int f(y_t|\theta_t,D_{t-1})\,p(\theta_t|D_{t-1})dv(\theta_t) = \int f(y_t|\theta_t)\,p(\theta_t|D_{t-1})dv(\theta_t)$$

Finally, by applying the Bayes' theorem and the conditional independence property $Y_t \perp (Y_1,...,Y_{t-1})|\theta_t$, we obtain:

$$p(\theta_t|D_t) = \frac{p(\theta_t|D_{t-1})\,f(y_t|\theta_t,D_{t-1})}{f(y_t|D_{\text{t-1}})} = \frac{p(\theta_t|D_{t-1})\,f(y_t|\theta_t)}{f(y_t|D_{t-1})}.$$

On the basis of these results, we are able to compute the k-steps ahead predictive densities by means of a recursive algorithm. In particular, we obtain the following formulas:

$$p(\theta_{t+k}|D_t) = \int p(\theta_{t+k}|\theta_{t+k-1})\,p(\theta_{t+k-1}|D_t)dv(\theta_{t+k-1}),$$

$$f(y_{t+k}|D_t) = \int f(y_{t+k}|\theta_{t+k})\,p(\theta_{t+k}|D_t)dv(\theta_{t+k}).$$

2.8. The Kalman filter for DLM

The procedure illustrated until now provide a solution for any generic filtering problem; nevertheless, in general, the computation of conditional densities is not straightforward. On the contrary, in the specific case of dynamic linear models, the recursive procedure can be notably simplified thanks to the assumptions concerning the Gaussian distribution of the random vector $(\theta_0,\theta_1,...,\theta_t,Y_1,...,Y_t)$, with variances w_t and v_t, for any $t \geq 1$. As all the marginal distributions are also Gaussian, it will be sufficient to calculate their mean values, variances and covariances.

We illustrate hereafter the **Kalman filter**.

Considering a generic DLM model, if

$$\theta_{t-1}|D_{t-1} \sim \text{N}(\text{m}_{\text{t-1}},\text{C}_{\text{t-1}}) \text{ with } t \geq 1,$$

then:

1. the one-step-ahead predictive density of θ_t conditional on D_{t-1} is Gaussian, and its parameters are as follows:

$$a_t = E(\theta_t | D_{t-1}) = G_t m_{t-1}$$
$$R_t = Var(\theta_t | D_{t-1}) = G_t C_{t-1} G_t' + W_t$$

2. the one-step-ahead predictive density of Y_t conditional on D_{t-1} is Gaussian, and its parameters are as follows:

$$f_t = E(Y_t | D_{t-1}) = F_t' a_t$$
$$Q_t = Var(Y_t | D_{t-1}) = F_t' R_t F_t + V_t$$

3. the filtering density of θ_t conditional on D_t presents a Gaussian distribution, with parameters:

$$m_t = E(\theta_t | D_t) = a_t + R_t F_t Q_t^{-1} e_t$$
$$C_t = Var(\theta_t | D_t) = R_t - R_t F_t Q_t^{-1} F_t' R_t$$

where $e_t = Y_t - f_t$ identifies the forecast error.

Therefore, the Kalman filter provides a recursive procedure in order to calculate the predictive and filtering densities; trying to summarise, by starting from the knowledge of the distribution $\theta_0 | D_0 \sim \mathrm{N}(\mathrm{m}_0, \mathrm{C}_0)$ and then computing the density $p(\theta_1 | D_1)$, it is possible to derive the subsequent densities as soon as new data are available, thanks to the use of the conditional distribution $\theta_t | D_t$. Of course, the point estimate of θ_t given the information D_t, with respect to a loss function of the type $L(\theta_t, a) = (\theta_t - a)' H (\theta_t - a)$, is equal to the conditional expected value $m_t = E(\theta_t | D_t)$, which is the quantity minimizing the conditional expected loss $E((\theta_t - a)' H (\theta_t - a) | D_{t-1})$. In particular, m_t may be expressed as the sum of the prediction mean and a correction component which refers to the amount of the discrepancy between the new observation and its prediction, where the weight of the correction term is expressed by the *gain matrix*

$$K_t = R_t F'_t Q_t^{-1} .$$

Thus, the weight of the available information Y_t depends on Q_t, which is built by departing from the observation covariance matrix V_t, and on R_t, calculated as:

$R_t = Var(\theta_t | D_{t-1}) = G_t C_{t-1} G'_t + W_t$.

One of the drawbacks deriving from the computation of C_t by using the iterative algorithms provided by the theorem of the Kalman filter, is that this method may lead to having a non-symmetric or even negative definite covariance matrix. The two strategies which are most widely

used to overcome this problem are the square root filter, providing a formula for the sequential updating of the square root of C_t , and the SVD-based filter, aiming at sequentially updating the singular value decomposition of C_t .

The authors G. Petris, S. Petrone and P. Campagnoli, in their volume entitled *Dynamic Linear Models with R* (2007), in order to illustrate the role of filtering, provide a graphical comparison between the actual level of the Nile river and the corresponding filtered levels obtained by means of two random walk plus noise models having different signal-to-noise ratios. We report this comparison in Figure 2.1.

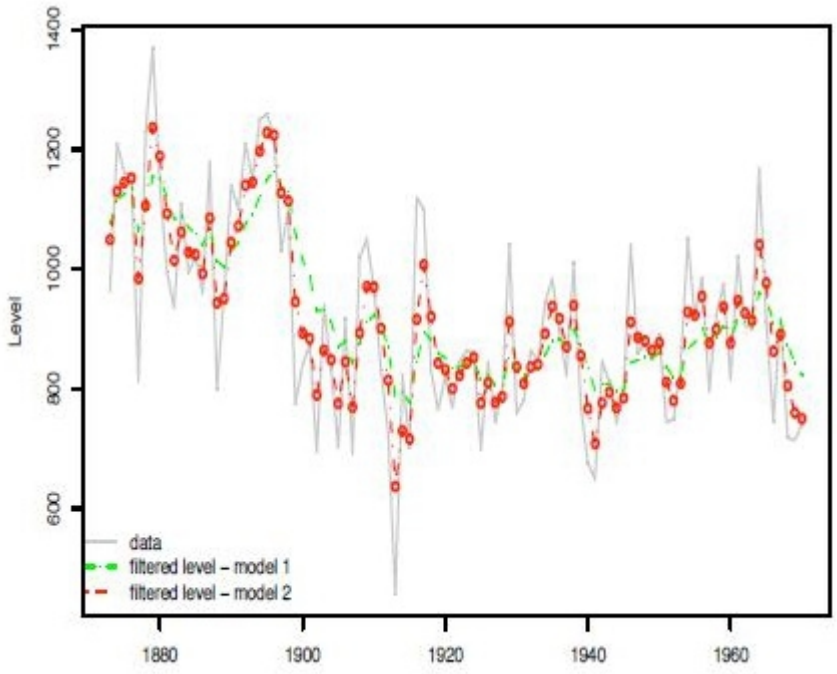

Figure 2.1. Filtered values of the Nile river level for two different signal-to-noise ratios

What clearly appears from the graphic is how in the second model, whose signal-to-noise ratio is ten times larger than it is in the first model, the filtered values generally follow more closely the actual data.

2.9. Smoothing

The aim of smoothing is that of retrospectively reconstructing the structure of the system on the basis of the available observations, in order to study, for example, the socio-economic phenomenon underlying the observations. Similarly to what happens in the case of filtering, even in this case we

can use a backward-recursive algorithm to compute the conditional densities of $\theta_t|D_T$ for each $t < T$, by starting from the filtering density $p(\theta_T|D_T)$.

In particular, the backward transition probability of the state sequence $(\theta_0, \ldots, \theta_T)$ conditional on D_T is given by:

$$p(\theta_t|\theta_{t+1}, D_T) = \frac{p(\theta_{t+1}|\theta_t)\, p(\theta_t|D_t)}{p(\theta_{t+1}|D_t)} \quad ,$$

and the smoothing densities of θ_t conditional on D_T are computed by applying the following backward recursion method starting from $p(\theta_T|D_T)$:

$$p(\theta_t|D_T) = p(\theta_t|D_t) \int \frac{p(\theta_{t+1}|\theta_t)}{p(\theta_{t+1}|D_t)}\, p(\theta_{t+1}|D_T)\, d\,\mu(\theta_{t+1}) \quad .$$

In the specific situation of a DLM, if $\theta_{t+1}|D_T \sim N(s_{t+1}, S_{t+1})$, then $\theta_t|D_T \sim N(s_t, S_t)$, where

$$s_t = m_t + C_t G'_{t+1} R_{t+1}^{-1}(s_{t+1} - a_{t+1})$$

$$S_t = C_t + C_t G'_{t+1} R_{t+1}^{-1}(S_{t+1} - R_{t+1}) R_{t+1}^{-1} G_{t+1} C_t \quad .$$

In conclusion, the smoothing procedure departing from the Kalman filter allows us to calculate the densities of θ_t given D_T in a recursive way, starting from $t = T - 1$, where we know that $\theta_T|D_T \simeq N(s_T = m_T, S_T = C_T)$, and then proceeding backward in order to compute the densities of $\theta_t|D_T$ where $t = T - 2$, $t = T - 3$, and so forth.

In Figure 2.2 we provide a visual evidence of the smoothing effect, by showing a comparison between the data concerning the actual level of the Nile river and the corresponding smoothed values, accompanied by their 95% confidence intervals.

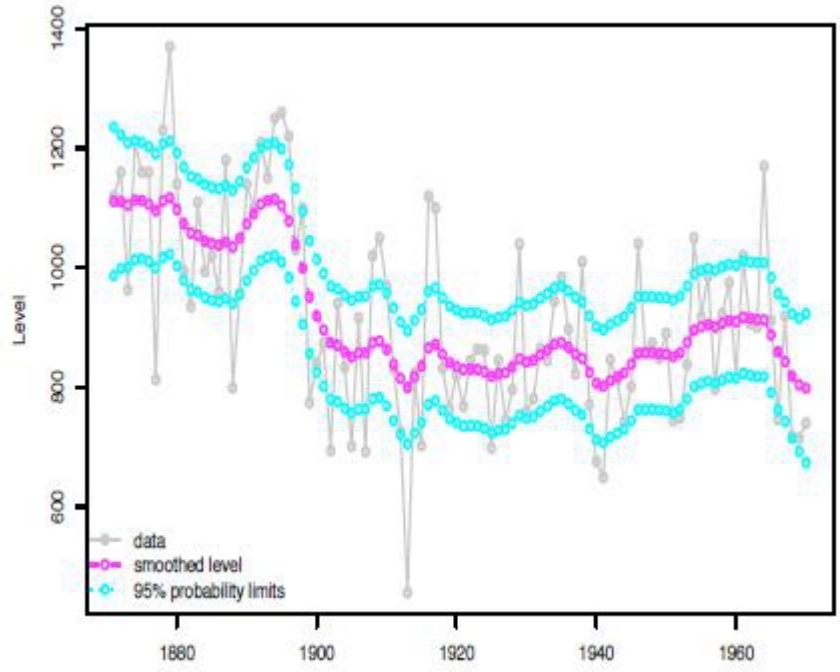

Figure 2.2. Smoothed values of the Nile river level, with 95% probability limits

2.10. Forecasting

The forecasting procedure allows us to predict the future values of the observations Y_{t+k} or of the state vectors θ_{t+k} based on the available data D_t .

One of the most frequent usages of forecasting is the computation of one-step-ahead forecasts, which are mainly useful in two situations: firstly, when we need to update our forecasts related to data which become available sequentially, such as in the case of the price of a stock, and secondly, when we want to check the performance of a model (in this case we will compute *in-sample* forecasts).

As we presented in the theorem of the Kalman filter, the one-step-ahead predictive densities for states and observations are obtained as a by-product of the filter itself.

In the case of forecasting, the meaning of the gain matrix K is the same that we had in the case of filtering. In fact, if we consider the one-step-ahead forecasts of the observations, given by $f_t = E(Y_t|D_t)$ for each t, since f_t is a function of the filtering mean m_{t-1} , the magnitude of K indicates the influence of an unexpected observation Y_{t-1} in determining the value of the forecast f_t .

In order to provide a visual evidence of the usage of forecasting for model checking purposes, we present in Figure 2.3 a graphical comparison between the actual data concerning the level of the

Nile river and two in-sample one-step-ahead forecasts deriving from two random walk plus noise models having different signal-to-noise ratios.

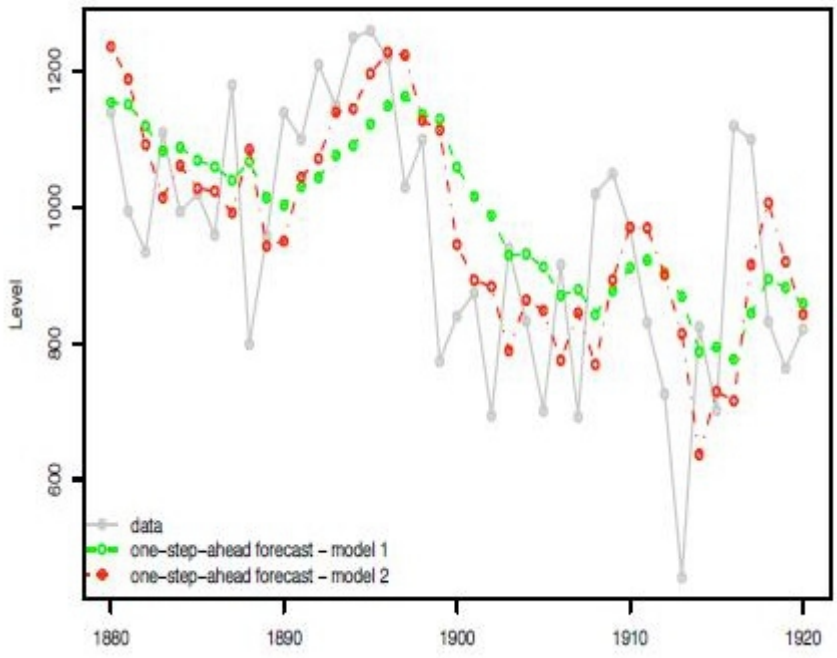

Figure 2.3. One-step-ahead forecasts of the Nile level by using different signal-to-noise ratios

When we are interested in providing forecasts concerning the *k*-steps-ahead behaviour of a time series, we need to exploit the property that the joint distribution of present and future states $(\theta_{t+k})_{k\geq 0}$ and future observations $(Y_{t+k})_{k\geq 1}$ is that of a DLM having initial distribution $p(\theta_t|D_t)$, and the available data only provide us with some knowledge related to this distribution, in particular the mean m_t. Consequently, we can state that knowing m_t is sufficient in order to realize predictions.

The flow of information we need to exploit works as follows: the data $(Y_1,\dots,Y_t)$ provide us with some information about θ_t, which allows us to obtain some insight on the state evolution until θ_{t+k} and subsequently about Y_{t+k}. Of course, the larger the value of *k*, the lower will be the predictive capability of the model.

At this point we present the recursive methods for computing the first and second moments of the forecast distributions. We firstly define:

$$a_t(k)=E(\theta_{t+k}|D_t)$$
$$R_t(k)=Var(\theta_{t+k}|D_t)$$
$$f_t(k)=E(Y_{t+k}|D_t)$$

$$Q_t(k)=Var(Y_{t+k}|D_t)$$

If we set $a_t(0)=m_t$ and $R_t(0)=C_t$, then for $k \geq 1$ we obtain the following distributions:

1. the predictive state θ_{t+k} conditional on D_t presents a Gaussian distribution, with moments

$$a_t(k)=G_{t+k}a_{t,k-1} \quad ,$$

$$R_t(k)=G_{t+k}R_{t,k-1}G'_{t+k}+W_{t+k} \quad ;$$

2. the predictive observation Y_{t+k} conditional on D_t presents a Gaussian distribution, with moments

$$f_t(k)=F_{t+k}a_t(k) \quad ,$$

$$Q_t(k)=F_{t+k}R_t(k)F'_{t+k}+V_t \quad .$$

Chapter 3

Model specification

In this chapter we will illustrate the procedures utilised in order to arrive to the correct specification of a dynamic linear model for a univariate or multivariate time series. The main bibliographical reference we used for this topic is the treatment *Dynamic linear models with R* by G. Petris, S. Petrone, P. Campagnoli (Springer, 2007).

3.1. Introduction

In this chapter we will illustrate some specific categories of DLMs that are commonly used in order to model both univariate and multivariate time series. In fact, as we will explain hereafter, a time series can be more easily analysed if we consider it as originated from the union of different components, such as, for example, the composition of a long term trend, a seasonal component and an error term presenting a white noise distribution.

In particular, we will firstly analyse some basic models which have to be combined by the researcher to derive a useful model specification; secondly, we will take into account the case in which the model matrices are dependent of a vector of unknown parameters, and thus an estimation process will be needed.

3.2. Superposition of models

In general, it can be useful to consider a time series as a sum of different elementary components, each one of which describes a specific characteristic of the series itself, such as, for example, its long term trend, its seasonal component and its relationship with some given explanatory variables. In particular, it is possible to model each individual component by means of a DLM and subsequently combine these single components into a unique DLM, which aims at providing a good explanation of the whole time series.

For example, if we take into account a univariate series which we denote by Y_t, we will be able to rewrite such series in the following form:

$$Y_t = Y_{1,t} + ... + Y_{h,t},$$

where all the $Y_{i,t}$ represent given independent components, such as trend, seasonality, and so forth.

Every $Y_{i,t}$ component might be modelled by a DLM as follows:

$$\begin{aligned} Y_{i,t} &= F_{i,t}\theta_{i,t} + v_{i,t}, & v_{i,t} &\sim \mathrm{N}(0, \mathrm{V_{i,t}}), \\ \theta_{i,t} &= G_{i,t}\theta_{i,t-1} + w_{i,t}, & w_{i,t} &\sim \mathrm{N}(0, \mathrm{W_{i,t}}), \end{aligned}$$

where all the vectors composed by the observation and its respective state vector, such as $(Y_{i,t}, \theta_{i,t})$ and $(Y_{j,t}, \theta_{j,t})$, are independent for any $i \neq j$. On the basis of the independence assumption, we obtain that $Y_t = \sum_{i=1}^{h} Y_{i,t}$ and that the given time series can be described by the following DLM:

$$\begin{aligned} Y_t &= F_t\theta_t + v_t, & v_t &\sim \mathrm{N}(0, \mathrm{V_t}), \\ \theta_t &= G_t\theta_{t-1} + w_t, & w_t &\sim \mathrm{N}(0, \mathrm{W_t}). \end{aligned}$$

The state vector and the F matrix are defined as follows:

$$\theta_t = \begin{pmatrix} \theta_{1,t} \\ \vdots \\ \theta_{h,t} \end{pmatrix}, \qquad F_t = \left(F_{1,t} \middle| \cdots \middle| F_{h,t}\right);$$

G_t and W_t are defined by the following diagonal matrices:

$$G_t = \begin{pmatrix} G_{1,t} & & \\ & \ddots & \\ & & G_{h,t} \end{pmatrix}, \qquad W_t = \begin{pmatrix} W_{1,t} & & \\ & \ddots & \\ & & W_{h,t} \end{pmatrix}.$$

Finally, the covariance matrix V_t is given by the following sum: $V_t = \sum_{i=1}^{j} V_{i,t}$.

3.3. Trend models

The trend of a time series indicates its smooth long term development over time, which can be considered as the expected future behaviour of Y_{t+k} for any $k \geq 1$, given the information available at the present time *t*. In general, any reasonable trend shape can be approximated by an *n*-order polynomial model, which consists of a DLM with steady matrices *F* and *G*, known covariance matrices V_t and W_t, and a forecast function defined as follows:

$$f_t(k) = E(Y_{t+k}|D_t) = a_{t,0} + a_{t,1}k + ... + a_{t,n-1}k^{n-1}, \qquad \text{with } k \geq 0,$$

where $a_{t,0},...,a_{t,n-1}$ indicate linear functions of $m_t = E(\theta|D_t)$ and are independent of the term *k*.

The larger the polynomial order *n*, the more precise the approximation; anyway, the most widely used polynomial models are the one with order $n = 1$ (random walk plus noise model) and the one with order $n = 2$ (linear growth model).

3.4. The random walk plus noise model

The random walk plus noise, also referred to as **local level model**, is defined by the following equations:

$$Y_t = \mu_t + v_t, \qquad v_t \sim \mathrm{N}(0, \mathrm{V_t}),$$

$$\mu_t = \mu_{t-1} + w_t, \qquad w_t \sim \mathrm{N}(0, \mathrm{W_t}).$$

It can be shown that the trajectory of the process is strongly affected by the ratio *W/V*, denominated *signal-to-noise* ratio.

The *k*-steps-ahead predictive distribution for this kind of model is given by:

$$Y_{t+k}|D_t \sim \mathrm{N}(\mathrm{m_t}, \mathrm{Q_t(k)}), \qquad \text{with } k \geq 1,$$

where the variance is given by $Q_t(k) = C_t + \sum_{j=1}^{k} W_{t+j} + V_{t+k} = C_t + kW + V$ and the forecast function is constant and given by $f_t(k) = E(Y_{t+k}|D_t) = m_t$. It can be easily noticed how the uncertainty related to the future observations increases as the time horizon *k* becomes further.

An advantage of utilising this kind of model is that it allows us to compute a lower bound for the variance $Q_t(k)$. In fact, as *t* tends to infinity, it is possible to derive the limit of K_t, which is given by:

$$\lim_{t\to\infty} K_t = \frac{r}{2}\left(\sqrt{1+\frac{4}{r}} - 1\right) = K$$

where $r = W/V$. This result implies that, when *t* is large enough, C_t converges to $C_t \approx KV$, which allows us to derive the lower bound for the variance of the predictive distribution, corresponding to an upper bound for the precision attainable in the estimation of the current value of μ_t.

3.5. The local linear trend model

The local linear trend model, also referred to as **linear growth model**, is defined by the following equations:

$$\begin{aligned} Y_t &= \mu_t + v_t, & v_t &\sim \mathrm{N}(0, \mathrm{V}), \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + w_{1,t}, & w_{1,t} &\sim \mathrm{N}(0, \sigma^2_{\mathrm{w1}}), \\ \beta_t &= \beta_{t-1} + w_{2,t}, & w_{2,t} &\sim \mathrm{N}(0, \sigma^2_{\mathrm{w2}}), \end{aligned}$$

where the error terms are assumed to be uncorrelated.

The state vector is constituted by $\theta_t = (\mu_t, \beta_t)'$, where the term μ_t usually indicates the local average level of the series and β_t identifies the local growth rate. In this model both the current level and the growth rate are assumed to evolve over time, which allows us to obtain a higher degree of flexibility if compared to the model presented previously.

If we denote with m_{t-1} the vector $(\mu_{t-1}, \beta_{t-1})'$, and if we indicate with a_t the matrix given by:

$$a_t = Gm_{t-1} = \begin{bmatrix} \hat{\mu}_{t-1} + \hat{\beta}_{t-1} \\ \hat{\beta}_{t-1} \end{bmatrix}$$

we obtain that the filtering state estimate is equal to:

$$f_t = F_t a_t = \hat{\mu}_{t-1} + \hat{\beta}_{t-1}$$

where

$$m_t = \begin{bmatrix} \hat{\mu}_t \\ \hat{\beta}_t \end{bmatrix} = a_t + K_t e_t = \begin{bmatrix} \hat{\mu}_{t-1} + \hat{\beta}_{t-1} + k_{t1} e_t \\ \hat{\beta}_{t-1} + k_{t2} e_t \end{bmatrix}.$$

The forecast function is given by:

$$f_t(k) = \hat{\mu}_t + k\hat{\beta}_t ;$$

since this is a linear function of k, we can say that the local linear trend model consists of a polynomial DLM of order 2.

Since the variances are assumed to be constant, it is possible to demonstrate that, as t tends to infinity, K_t converges to a given vector (k_1, k_2) which we denote with K. Thus, we obtain the asymptotic formulas of the state vector.

$$\hat{\mu}_t = \hat{\mu}_{t-1} + \hat{\beta}_{t-1} + k_1 e_t$$
$$\hat{\beta}_t = \hat{\beta}_{t-1} + k_2 e_t$$

3.6. The *n*th order polynomial model

The *n*th order polynomial model is based upon an *n*-dimensional state space and is specified by the following matrices:

$$F = (1,0,...,0)$$

$$G = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 1 \\ 0 & \dots & & 0 & 1 \end{bmatrix}$$

$$W = diag(W_1,...,W_n)$$

The equations composing this model are as follows:

$$\begin{cases} Y_t = \theta_{t,1} + v_t \\ \theta_{t,j} = \theta_{t-1,j} + \theta_{t-1,j+1} + w_{t,j} & \text{with } j = 1, \dots, n\text{-}1 \\ \theta_{t,n} = \theta_{t-1,n} + w_{t,n} \end{cases}$$

The first element of the state vector $\theta_{t,1}$ for any time *t* represents the average level of the series, and for *j* = 2, ... , *n* the *j*th element of the state vector indicates the change of the *j*-1th element during the subsequent time interval.

It can be proved that the forecast function consists of a polynomial of order *n* - 1 in *k*.

This kind of model includes the special case in which $W_1 = ... = W_{n-1} = 0$, also referred to as *integrated random walk model*. In this case the level satisfies the condition $\Delta^n \mu_t = \varepsilon_t$, where Δ is the determinant of the diagonal matrix *W* and ε_t identifies a white noise error term.

3.7. The representation of seasonality

In order to simplify our explanation, we will assume that the series we are analysing presents no trend component, thus it is stationary and has mean equal to zero. In particular, the effect of seasonality might be expressed by a different coefficient for every different quarter of the year, which we denote with α_i for *i* = 1, ... , 4. If we assume that *t*-1 is referred to the first quarter, *t* to the second quarter, and so on, we might describe the series by the following model.

$$Y_{t-1} = \alpha_1 + v_{t-1}$$
$$Y_t = \alpha_2 + v_t$$

If we suppose that the state vector at time *t* -1 is equal to $\theta_{t-1} = (\alpha_1, \alpha_4, \alpha_3, \alpha_2)'$ and the matrix F_t is given by *F* = (1,0,0,0) for every *t*, we obtain that the model can be written according to the DLM form as follows:

$$Y_t = F\theta_t + v_t .$$

Our goal is to create a permutation matrix which returns the following modification of the state vector: $\theta_t = (\alpha_2, \alpha_1, \alpha_4, \alpha_3)$, so that, for the second quarter of the year, we have the equation $Y_t = F\theta_t + v_t = \alpha_2 + v_t$. Such permutation can be obtained thanks to the use of the following *G* matrix:

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Thanks to this permutation, we can write the state equation as follows:

$$\theta_t = G\theta_{t-1} + w_t = (\alpha_2, \alpha_1, \alpha_4, \alpha_3)' + w_t .$$

If we assume that the seasonal effects may change over time, W_t is not a null matrix.

In general, we can state that it is possible to describe a seasonal time series by means of an *s*-dimensional state vector θ_t, an *F* matrix of the type *F* = (1,0, ... ,0) and an *s* by *s* permutation matrix *G*.

The seasonal coefficients are aimed at describing the variations of the series around a given mean value; for this reason they are usually imposed to sum to zero, with the constraint which follows:

$$\sum_{j=1}^{s} \alpha_j = 0 .$$

An alternative procedure to model seasonal variations is to consider one of the coefficients as the normal value in the absence of seasonal effect, and the remaining ones as indicators of periodical variations. In this way, our state vector will be an (*s* - 1)-dimensional one, and consequently the *F* and *G* matrices will also be (*s* - 1)-dimensional. For example, when the seasonal coefficient must pass from α_1 to α_2, we will need to transform the state vector $\theta_{t-1} = (\alpha_1, \alpha_4, \alpha_3)$ into $\theta_t = (\alpha_2, \alpha_1, \alpha_4)$. In order to do this, we must utilise the following permutation matrix:

$$G = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

In general, if the period of the seasonal model is equal to *s*, we must consider *F* = (1,0,...,0) and

$$G = \begin{bmatrix} -1 & -1 & \cdots & -1 & -1 \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & & 1 & 0 \end{bmatrix}.$$

The variance of the seasonal components is expressed by a system evolution error matrix, which we define as $W = diag(\sigma_w^2, 0, ..., 0)$.

3.8. Discount factors

Till now we have underlined several times how the influence that past observations have on the state estimation and forecasting is described by the matrix W_t. In particular, if this matrix includes large quantities, this means that passing from θ_{t-1} to θ_t causes a huge loss of information, because the estimation of $\theta_t|y_{1:t}$ is mainly determined by the real values of the current observation y_t.

In fact, the amount of uncertainty on θ_{t-1} conditional on the data $y_{1:t-1}$ is expressed by the conditional covariance matrix $C_{t-1} = V(\theta_{t-1}|y_{1:t-1})$. If we move to the subsequent state vector θ_t we obtain that the variance of its estimate is given by $V(\theta_t|y_{1:t-1}) = R_t = P_t + W_t$, where P_t corresponds to the variance component which is uniquely influenced by the previous state vector θ_{t-1}, and is given by:

$$P_t = V(G_t\theta_{t-1}|D_{t-1}) = G_t C_{t-1} G_t' .$$

3.9. Simulation-based Bayesian inference

If a DLM includes in its specification the observations $y_{1:T}$ and a multidimensional unknown parameter ψ, the posterior distribution of the parameters and the unobservable states is given by

$$\pi(\psi, \theta_{0:T}|y_{1:T}) \quad .$$

As it was mentioned in the first chapter, there exist many situations where it is not possible to compute this distribution analytically, and it is then necessary to resort to simulation-based methods, such as, for instance, MCMC methods. In order to come up with posterior summaries, in the MCMC approach a sample is generated from the posterior itself, and subsequently summaries are drawn from the simulated sample and evaluated. Even in the case in which one is only interested in the posterior distribution of the unknown parameter, $\pi(\psi|y_{1:T})$, it may be useful to include the states in the posterior, as it contributes to designing an efficient sampler. Indeed, drawing a random variable from $\pi(\psi|\theta_{0:T}, y_{1:T})$ is generally much simpler than drawing it from $\pi(\psi|y_{1:T})$. In addition, there also exist efficient algorithms allowing to generate the states conditionally on the data and the unknown parameter. In fact, a sample from the posterior distribution of the parameter and the unobservable states can be obtained by using a Gibbs sampler which alternates draws from

$\pi(\psi|\theta_{0:T}, y_{1:T})$ and $\pi(\theta_{0:T}|\psi, y_{1:T})$. The results of the simulation of the posterior may then be utilised as an input in order to generate a sample from the predictive distribution of states and observables, $\pi(\theta_{T+1:T+k}, y_{T+1:T+k}|y_{1:T})$. Indeed,

$$\pi(\theta_{T+1:T+k}, y_{T+1:T+k}, \psi, \theta_T|y_{1:T}) = \pi(\theta_{T+1:T+k}, y_{T+1:T+k}|\psi, \theta_T) \cdot \pi(\psi, \theta_T|y_{1:T}) \quad .$$

Thus, for every couple $(\psi, \theta - T)$ drawn from $\pi(\psi, \theta_T|y_{1:T})$, it is possible to generate the future values $\theta_{T+1:T+k}, y_{T+1:T+k}$ from $\pi(\theta_{T+1:T+k}, y_{T+1:T+k}|\psi, \theta_T)$ and obtain a sample from the predictive distribution.

This approach presents the remarkable advantage that it provides a simple solution to the filtering, smoothing and forecasting problem for a DLM with unknown parameters. However, it also has the drawback that, if we wanted to update the posterior after a new set of observations becomes available, we should apply the Gibbs sampling algorithm all over again, which could turn out to be quite inefficient.

3.10. Forward Filtering Backward Sampling

If our goal is doing a Gibbs sampling from $\pi(\theta_{0:T}, \psi|y_{1:T})$, we need to do a simulation from the conditional distributions $\pi(\psi|\theta_{0:T}, y_{1:T})$ and $\pi(\theta_{0:T}|\psi, y_{1:T})$, but while the first one is completely specified by the problem itself, the second one can only be determined by means of simulation techniques.

Thus, in order to determine such distribution, we depart from the information concerning the mean and the variance of the density of θ_t conditional on $y_{1:T}$ and ψ, for t = 0, 1, ..., T, which can be derived by means of the smoothing recursive algorithm. Then, as we are assuming that such density is Gaussian, we can consider this posterior conditional distribution as completely specified. Consequently, if one wanted to derive the joint posterior of $\theta_{0:T}$ conditional on $y_{1:T}$ and ψ, it would be also necessary to determine the covariances between θ_t and θ_s for every $t \neq s$.

Coherently with what we explained about Gibbs sampling, the distribution $\pi(\theta_{0:T}|\psi, y_{1:T})$ will periodically have the role of full conditional, thus we need to devise a procedure which allows us to generate a sample from such distribution, and the **Forward Filtering Backward Sampling** (**FFBS**) method was developed with this objective.

This methods departs from the formula for the joint distribution of $(\theta_0, \theta_1, ..., \theta_T)$ given D_T, which is easily obtainable by applying the Bayes' theorem:

$$\pi(\theta_0,\theta_1,...,\theta_T|D_T)=\prod_{t=0}^{T}\pi(\theta_t|\theta_{t+1},...,\theta_T,D_T)\,.$$

Of course, the last factor of the product is given by $\pi(\theta_T|D_T)$, whose distribution is $N(m_T,C_T)$. Thus, in order to draw a sample from the joint density, we could start from an estimation of θ_T and then recursively simulate the values of θ_t from $\pi(\theta_t|\theta_{t+1},...,\theta_T,D_T)$ for $t = T - 1, T - 2, ..., 0$. It can be shown that

$$\pi(\theta_t|\theta_{t+1},...,\theta_T,D_T)=\pi(\theta_t|\theta_{t+1},D_T)$$

and its distribution is $N(h_t,H_t)$ where

$$h_t=m_t+C_tG'_{t+1}R^{-1}_{t+1}(\theta_{t+1}-a_{t+1})\,,$$

$$H_t=C_t-C_tG'_{t+1}R^{-1}_{t+1}G_{t+1}C_t\,.$$

Thus, once we have obtained the simulated sample $(\theta_{t+1},...,\theta_T)$ we have to simulate θ_t from a $N(h_t,H_t)$ distribution, where the mean h_t depends on the value of θ_{t+1} generated previously.

In order to summarise, the FFBS procedure is composed of the following phases:

1. Apply Kalman filter;
2. Simulate θ_T from a $N(m_T,C_T)$ density function;
3. For values of t comprised between $T - 1$ and 0, simulate θ_t from a $N(m_t,C_t)$ density function.

Chapter 4

Identification of a DLM for the American GDP

As it was mentioned in the preface, W. A. Barnett, M. Chauvet and D. Leiva-Leon, in their paper entitled *Real-Time Nowcasting Nominal GDP Under Structural Break*, dating to 2014, proposed a multivariate state-space model with the goal of forecasting the future trend of the nominal GDP of the United States based on specific macroeconomic indicators. In this chapter we want to verify the validity of a model of this kind, by estimating a dynamic linear model for the American GDP taking into account the same indicators. For this purpose, we will utilise the R statistical software.

4.1. Introduction

Taking as a reference the above mentioned paper, as explicative variables for the American GDP we considered the following macroeconomic indicators: the Industrial Production Index (IPI), the Consumer Price Index (CPI) and the 3-Month Treasury Bill interest rate (TBILL).

We retrieved the numerical information from the website of the Federal Reserve Bank of St. Louis, http://research.stlouisfed.org/fred2, and chose to consider quarterly observations from the considered variables in the course of 51 years, from January 1963 to October 2013, obtaining

overall a sample of 204 instances. The values of the American GDP are expressed in billions of dollars.

We illustrate the graphs of the four time series considered in the Figures 4.1 to 4.4.

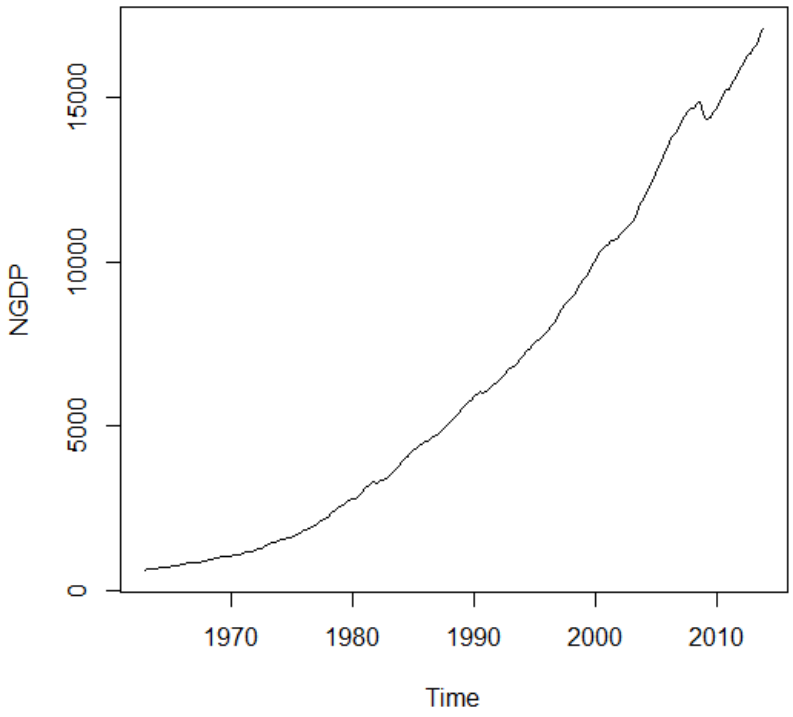

Figure 4.1. Plot of the American nominal GDP

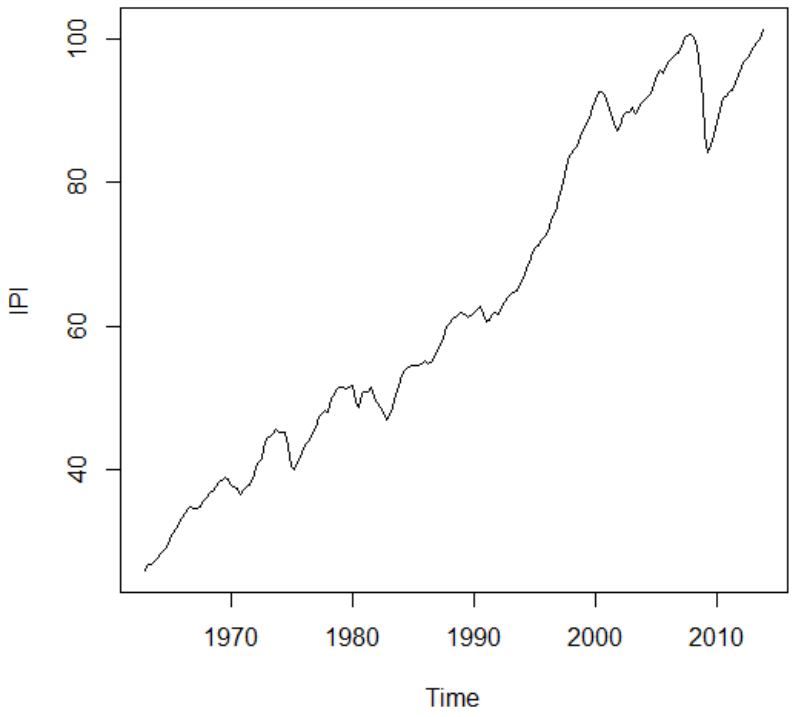

Figure 4.2. Plot of the American Industrial Production Index

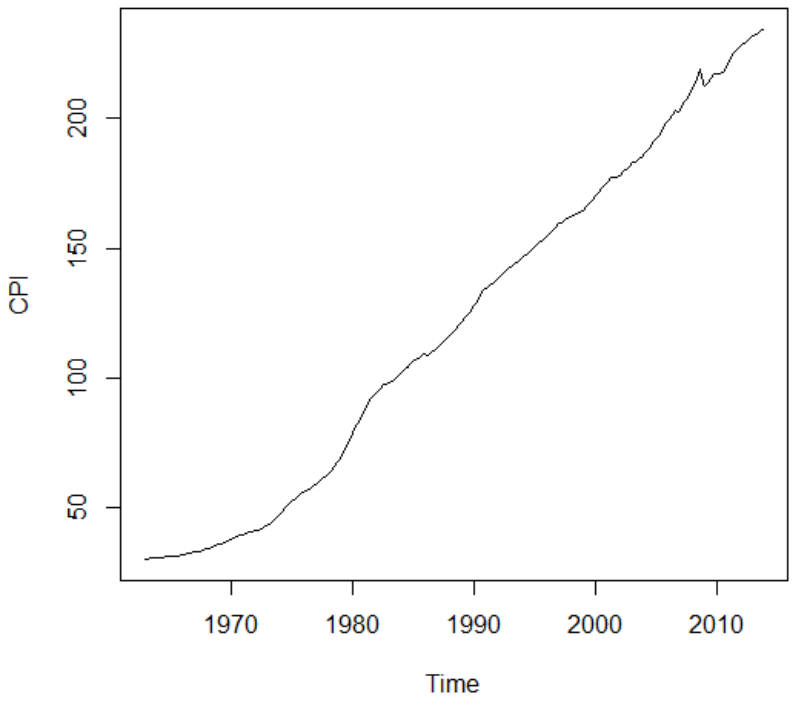

Figure 4.3. Plot of the American Consumer Price Index

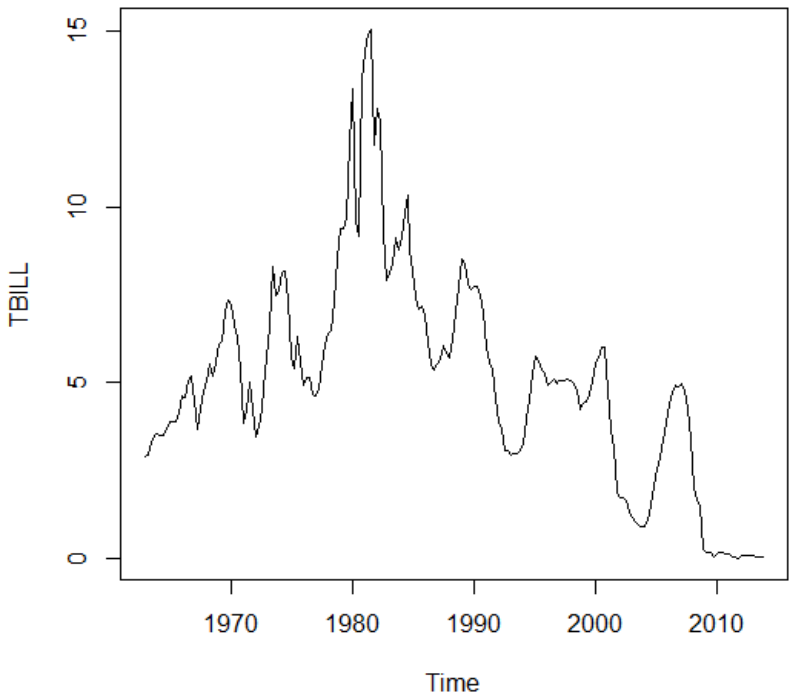

Figure 4.4. Plot of the 3-month Treasury-Bill rate

In the course of this chapter we will propose four different models, in particular: a 2nd order polynomial model without the inclusion of any seasonal component, a 2nd order polynomial model including a seasonal component related to the T-Bill rate, and two dynamic linear regression models, the first of which considers contemporary values of the interest variable and the regressors, while the second one considers lagged values of the regressors. The decision to include a seasonal component for the T-Bill rate in one of the models was due the fact that it was not possible to retrieve a seasonally adjusted time series for this variable, which was instead possible for all the other considered factors.

For our analysis purposes, we will exploit the R package `dlm`, which is available at the address: http://CRAN.R-project.org/package=dlm, and provides us with an integrated set of functions for Bayesian inference using DLMs. This package includes tools for filtering, smoothing and forecasting accordingly to the Kalman procedure, as well as maximum likelihood estimation and Markov Chain Monte Carlo simulation. The algorithms used in this package are based on the singular value decomposition of the relevant matrices, which has the advantage of guaranteeing a higher numerical stability if compared to other kinds of algorithms.

4.2. Acquisition of the numerical information and specification of the first model proposal

In the first part of the R program, whose code is reported in Appendix A, we import the numerical information from the dataset `database.txt`, reported in Appendix B, and create some graphs in order to establish whether a DLM could be suitable for modelling our data, and to decide which kinds of DLM constitute an acceptable model specification, by running the portion of code utilised for the acquisition of the numerical information.

In the first line we load the package `tseries`, which provides us with some tools for time series analysis purposes and includes some functions related to computational finance.

After importing the data, we run the `str` command, which allows us to see the types of objects which are contained in our data frame. The output of this command, reported hereafter, shows that our database is composed of one nominal vector, (called `factor` in the R terminology) corresponding to the array of the observation dates, and four numeric vectors, corresponding to the variables we wish to analyse.

```
'data.frame':   204 obs. of  5 variables:
 $ observation_date: Factor w/ 204 levels "1963-01-01","1963-04-01",..: 1 2 3
4 5 6 7 8 9 10 ...
 $ GDP             : num  623 632 645 655 671 ...
 $ INDPRO          : num  26.1 26.8 27 27.4 27.8 ...
 $ CPI             : num  30.5 30.5 30.7 30.8 30.9 ...
 $ TB3MS           : num  2.91 2.94 3.29 3.5 3.53 3.48 3.5 3.68 3.89 3.87 ...
```

For a large part of the analysis, we will need to deal with time-series objects. For this reason, we transform the vectors `GDP`, `INDPRO`, `CPI` and `TB3MS`, which are currently numeric vectors, into time-series objects by applying the function `ts`, which also allows us to specify the time range and the frequency of the observations.

From the time series plots, we can see how all of the series follow the same type of dynamics, and in particular for each one of them it is reasonable to assume that the state vector is composed of a level and a slope component. For this reason, as a first model proposal, we decided to analyse each series by using a linear growth model.

In order to evaluate whether we should assume the existence of some degree of dependence among the variables, we start by creating a matrix of scatterplots, using the function `pairs`. The output is showed in Figure 4.5.

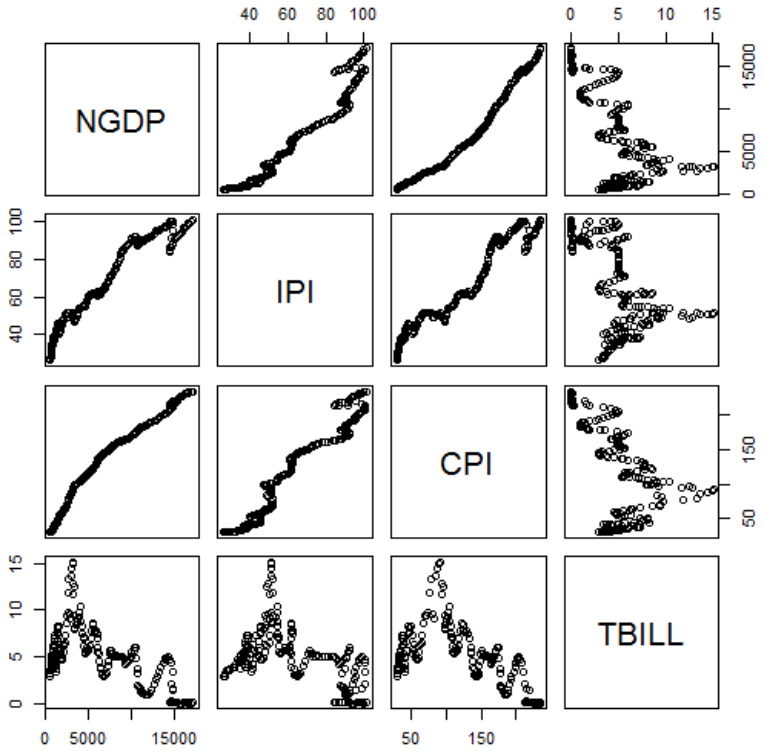

Figure 4.5. Scatterplot matrix of NGDP, IPI, CPI and TBILL

The graph shows an evident positive correlation between the GDP, the Industrial Production Index and the Consumer Price Index, while we can notice the presence of a negative correlation between all these variables and the T-Bill rate.

Such a circumstance leads us to suppose the possibility to estimate one of the variables we are analysing by using the information contained in the three other ones. In order to verify whether this is a reasonable assumption, since our purpose is that of building a forecasting model for the nominal GDP, we proceed with the estimation of a linear regression model where the GDP is the so-called response variable.

From the output of the estimation, reported hereafter, it appears that the p-values associated to the regression coefficients of the explicative variables are lower than 0.01. Consequently, we can state that all the regressors considered affect significantly the American GDP. Moreover, the value of the multiple R-squared is equal to 0.9802, which means that the proportion of the variability of the GDP explained by this model is equal to 98.02%, which is almost equivalent to the total variability.

```
Residuals:
     Min       1Q   Median       3Q      Max
-1357.15  -650.39   -10.01   508.77  1502.37

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2396.395    286.111  -8.376 9.54e-15 ***
IPI            66.320      9.582   6.922 5.93e-11 ***
CPI            48.139      3.372  14.276  < 2e-16 ***
TBILL        -226.006     19.106 -11.829  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 717.4 on 200 degrees of freedom
Multiple R-squared:  0.9802,    Adjusted R-squared:  0.9799
F-statistic:  3299 on 3 and 200 DF,  p-value: < 2.2e-16
```

In Figure 4.6 we present three graphs related to the regression residuals, from which it is evident how they do not appear centred around a constant mean value, but, on the contrary, there exist periods in which the residuals are clearly positive and other periods in which they are clearly negative. This is consistent with the gradually decreasing trend of the autocorrelation function and the p-values of the Ljung-Box test, which evidence how these residuals are non-stationary.

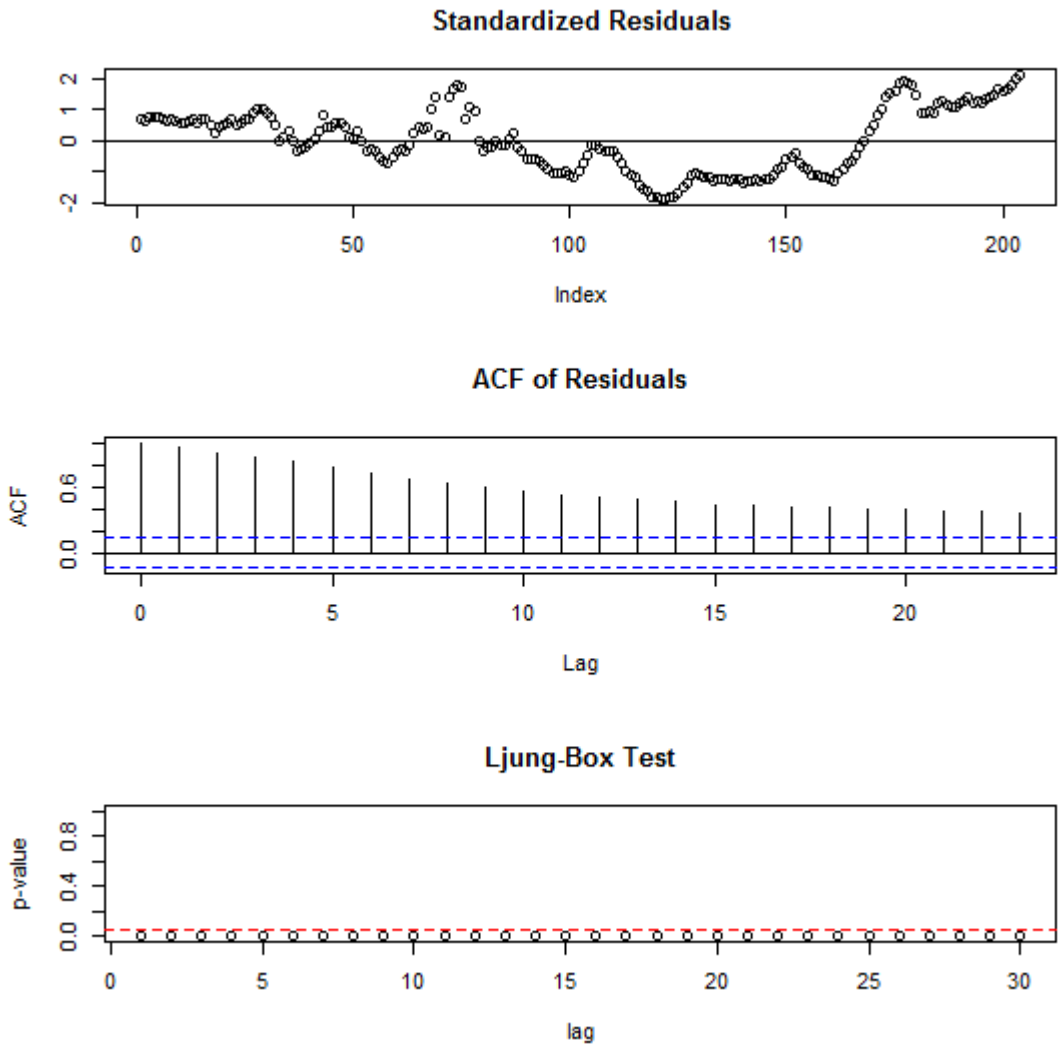

Figure 4.6. Diagnostic plots for the linear regression model

On the basis of these results, we decided to build a more sophisticated model in order to reduce the amount of residual autocorrelation presented by the linear regression model, keeping the assumption that, in our case, the evolution of the levels and of the slopes of the series are affected by correlated inputs. Consequently, for our first model proposal, we decided to apply the so-called **Seemingly Unrelated Time Series Equations** (**SUTSE**) paradigm, comprising a class of models which allow us to deal with dependent state vectors.

4.3. Seemingly unrelated time series equations

As we have just mentioned, SUTSE are a category of the dynamic linear models by means of which it is possible to specify the dependence structure among the state vectors $\theta_{1,t}, \ldots, \theta_{m,t}$, where *m* is equivalent to the number of the variables we are analysing.

If, as in our specific case, it is reasonable to study each series with a linear growth model, the assumption underlying this class of models is that, at any given time, the components of the system error corresponding to the levels of the different series may be correlated, and the same holds for the components of the system error corresponding to the different slopes. Thus, in order to describe the joint evolution of the state vectors it is convenient to group together both the levels and the slopes of each series in a single state vector $\theta_t=(\mu_{t1},\ldots,\mu_{tm},\beta_{t1},\ldots,\beta_{tm})'$. It will then be possible to express the system error of this overall state vector with a block-diagonal covariance matrix composed by two blocks, a first *m* by *m* block describing the correlation among the levels and a second *m* by *m* block accounting for the correlation among the slopes.

To go back to our specific situation, since we have $m = 4$ series, our common state vector will be $\theta_t=(\mu_{t1},\mu_{t2},\mu_{t3},\mu_{t4},\beta_{t1},\beta_{t2},\beta_{t3},\beta_{t4})'$ and the system equation will be expressed in the following way:

$$\begin{bmatrix}\mu_{t1}\\ \mu_{t2}\\ \mu_{t3}\\ \mu_{t4}\\ \beta_{t1}\\ \beta_{t2}\\ \beta_{t3}\\ \beta_{t4}\end{bmatrix}=\begin{bmatrix}1&0&0&0&1&0&0&0\\ 0&1&0&0&0&1&0&0\\ 0&0&1&0&0&0&1&0\\ 0&0&0&1&0&0&0&1\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}\mu_{t-1,1}\\ \mu_{t-1,2}\\ \mu_{t-1,3}\\ \mu_{t-1,4}\\ \beta_{t-1,1}\\ \beta_{t-1,2}\\ \beta_{t-1,3}\\ \beta_{t-1,4}\end{bmatrix}+\begin{bmatrix}w_{t1}\\ w_{t2}\\ w_{t3}\\ w_{t4}\\ w_{t5}\\ w_{t6}\\ w_{t7}\\ w_{t8}\end{bmatrix}$$

where $(w_{t1},w_{t2},w_{t3},w_{t4},w_{t5},w_{t6},w_{t7},w_{t8})'\sim N(0,W)$ and

$$W=\begin{bmatrix}&&&&0&0&0&0\\ &W_\mu&&&0&0&0&0\\ &&&&0&0&0&0\\ &&&&0&0&0&0\\ 0&0&0&0&&&&\\ 0&0&0&0&&W_\beta&&\\ 0&0&0&0&&&&\\ 0&0&0&0&&&&\end{bmatrix}.$$

The observation equation for the multivariate time series $((y_{t1},y_{t2},y_{t3},y_{t4}):t\geq 1)$ will be given by:

$$\begin{bmatrix} Y_{t1} \\ Y_{t2} \\ Y_{t3} \\ Y_{t4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \theta_t + \begin{bmatrix} v_{t1} \\ v_{t2} \\ v_{t3} \\ v_{t4} \end{bmatrix},$$

where $(v_{t1}, v_{t2}, v_{t3}, v_{t4})' \sim N(0, V)$. If the covariance matrix V is assumed to be non-diagonal, we are introducing a further source of correlation among the series.

Let us consider the general case of *m* univariate time series where Y_t denotes the multivariate observation at time *t*, and suppose that the observation equation of the *i*-th variable is described by the DLM

$$\begin{aligned} Y_{ti} &= F\theta_t^{(i)} + v_{ti} \\ \theta_t^{(i)} &= G\theta_{t-1}^{(i)} + w_t^{(i)} \end{aligned},$$

where $\theta_t^{(i)} = (\theta_{t1}^{(i)}, \ldots, \theta_{tp}^{(i)})'$, with $i = 1, \ldots, m$ and p equal to the number of parameters of the state vector. We obtain a SUTSE model for Y_t , whose observation and system equations have the following form:

$$\begin{aligned} Y_t &= (F \otimes I_m)\theta_t + v_t, & v_t &\sim N(0, V), \\ \theta_t &= (G \otimes I_m)\theta_{t-1} + w_t, & w_t &\sim N(0, W), \end{aligned}$$

where $\theta_t = (\theta_{t1}^{(1)}, \theta_{t1}^{(2)}, \ldots, \theta_{tp}^{(m-1)}, \theta_{tp}^{(m)})'$.

The block-diagonal structure of the covariance matrix W derives from the common assumption that for each variable the $w_t^{(i)}$ have diagonal variances. This circumstance implies that the forecasts of $\theta_{t+k}^{(i)}$ or $Y_{t+k,i}$ made at time t are uniquely based on the conditional distribution of $\theta_t^{(i)}$ given D_t .

4.4. SUTSE model

By running the piece of code concerning the SUTSE model, we set up a model of this kind in accordance with the structure just described.

As a first thing, we define the training set, which will be used for the estimation of the model parameters, and the test set, which will be used in order to evaluate the predictive capability of the model. In our case, we decided to include in the test set the last three years of the available data set, which were consequently excluded from the training set.

After this, we load the `dlm` package, and assign arbitrarily an initial value of 0 to the variance of the observational error for all of the three variables, by creating a vector of four zeros named

`varEpsilon`, and do the same for the variance of the evolution error of the level and of the slope, by creating respectively the vectors `varLevel` and `varSlope`. Then we initialize to zero the values of the covariances among the observational errors (`covEpsilon`), the levels (`covLevel`) and the slopes (`covSlope`) of the three variables. After that we join together all these vectors in a single vector named `parameters`. We will use this array in order to compute the maximum likelihood estimators for the parameters contained in it.

At this point, we need to program a function that we call `build` taking as an argument the parameter vector, in order to be able to construct a DLM based on the parameters assigned. Inside this function, we start by specifying that we wish to build a polynomial model of the 2nd order (`mod = dlmModPoly(2)`), which is equivalent to a linear growth model, and then redefine the *F* and *G* matrices according to the observation and system equations presented for the SUTSE models, by using the Kronecker products. Then we define the matrices `V`, `W1` and `W2`, corresponding respectively to the covariance matrices of the observational errors, of the levels and of the slopes, and initialize them by taking the exponential of the sections of the parameter vector referring to the respective variances, in order to ensure their positivity, and by taking the actual initial values of the parameter vector for the respective covariances. Moreover, we construct the matrix `W` of the evolution errors as a block-diagonal matrix including `W1` and `W2`. Finally, we specify our assumptions relatively to the mean (`m0`) and the variance (`C0`) of the prior distribution of the state vector.

After building the function, we organize the four series into a matrix and launch the command `dlmMLE` in order to optimize the parameters of the model according to the maximum likelihood criterion. In our case this command returns a convergence code equal to 0, which indicates that the algorithm has been completed successfully.

We are now able to construct the final SUTSE model by running again the `build` function, this time taking as an argument the parameter vector obtained thanks to the maximum likelihood estimator. The observational and state variance of the model obtained are showed hereafter.

```
V
1.009829e+03 0.06863831 0.0812761 0.02230113
6.863831e-02 0.17378985 0.1514390 0.19460946
8.127610e-02 0.15143902 0.1825120 0.15535255
2.230113e-02 0.19460946 0.1553526 0.22050462
```

```
W
0.501 0.075  0.041  0.037 0       0     0     0
0.075 0.351  0.309  0.350 0       0     0     0
0.041 0.309  0.348 -0.055 0       0     0     0
0.037 0.350 -0.055  0.406 0       0     0     0
0     0      0      0     488.105 0.246 0.079 0.088
0     0      0      0     0.246   0.333 0.166 0.256
0     0      0      0     0.079   0.166 0.266 0.273
0     0      0      0     0.088   0.256 0.273 0.309
```

4.5. Filtering

In this section of the analysis, we will compute the *filtering* distribution of the GDP by taking into account the model we have just illustrated, in order to be able to evaluate its predictive capability. We believe it is important to recall that, if for any pair of integers (i, j), with $i \leq j$, we denote by $y_{i:j}$ the observations from the ith to the jth inclusive, then the filtering distribution at a given time t is the conditional distribution of the state vector θ_t given $y_{i:t}$, for $t = 0, 1, \ldots, n$, where n is equal to the number of observations. The number of filtering distributions will then be equal to $n + 1$, and for $t = 0$ such density is conventionally assumed to be equal to the prior distribution of θ_0.

The first thing we do in the code is computing the filtering distribution by using the command `dlmFilter`, by taking the parameters previously estimated as known. In the list returned by this command, the components `U.C` and `D.C` correspond respectively to the U and D matrices constituting the singular value decomposition (SVD) of the variances $C_0, \ldots, C_n$ of the filtering distributions, where the SVD of a symmetric nonnegative definite covariance matrix Σ is given by $\Sigma = UDDU'$, where U is orthogonal and D is diagonal. In our particular situation, since we are considering 204 observations and the state vector is composed by 8 elements, `U.C` consists of a list of 205 8 by 8 matrices, and `D.C` consists of a matrix with 205 rows and 8 columns storing in each row the diagonal entries of all the successive D matrices. The function `dlmSvd2var` reconstructs the variances of the filtering distributions departing from their SVD, which allows us to compute confidence intervals for such densities.

In Figure 4.7 we present a graphical comparison between the actual GDP time series (in green) and its filtering distribution (in brown) accompanied by the relevant 90% confidence intervals (in brown).

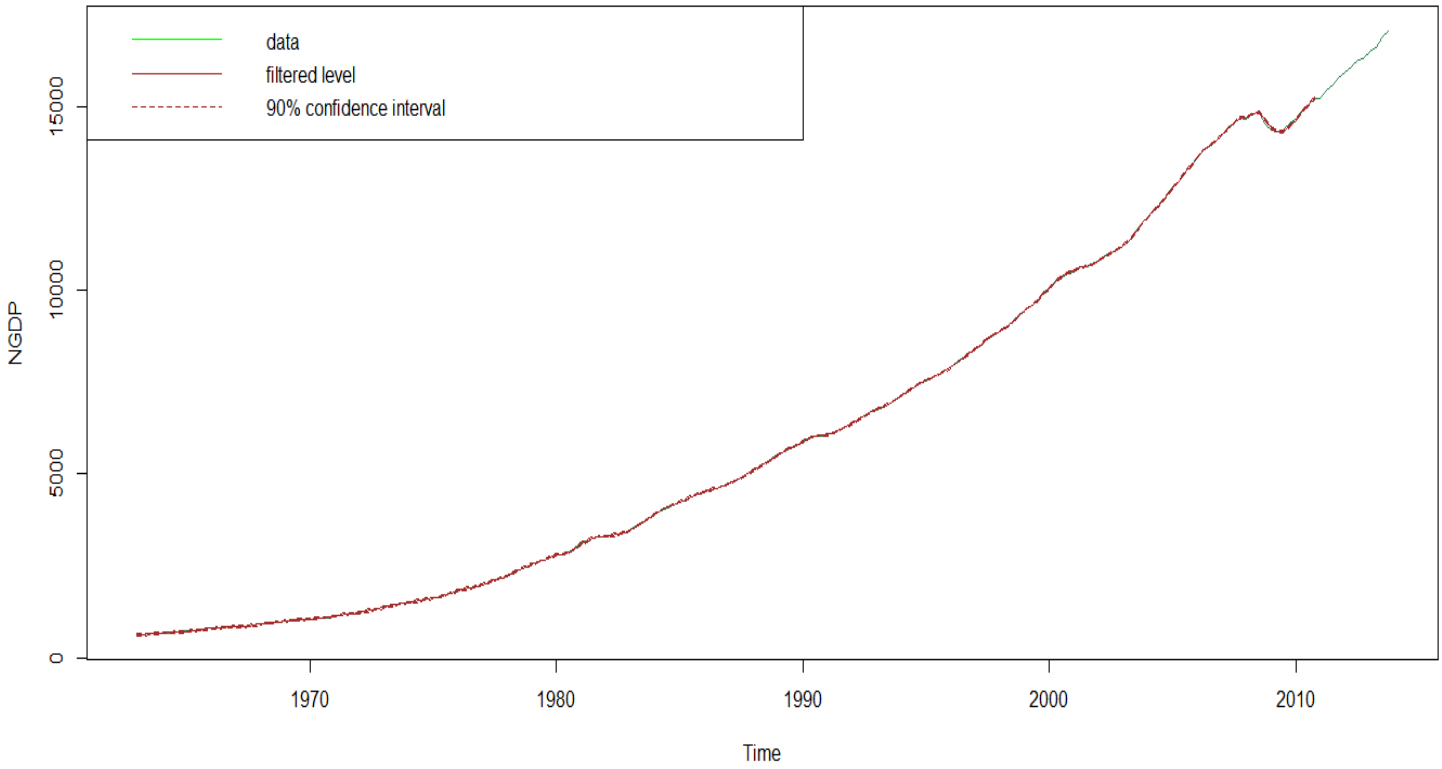

Figure 4.7. Nominal GDP with filtered level (SUTSE model)

From the graph it is easy to notice how an almost perfect identity exists between the actual data and the relevant filtered level. The probability intervals cannot be easily distinguished because of the reduced relative variance of the filtered level. Thus, we can state that the level of fitting obtained through the filtered series is, in the case of this model, very satisfying.

4.6. Evaluation of the predictive performance

As we have mentioned in the section dedicated to forecasting of the second chapter, it can be useful to compute one-step-ahead forecasts based on the filtered level of a time series as a tool for checking the predictive capability of a DLM model, which is exactly what we do in the relevant portion of code. In addition, we create a comparative graph of the actual data and of the relevant one-step-ahead forecast values, and evaluate the predictive capability of this model by computing some forecast performance measures which will be illustrated hereafter.

The R software returns the one-step-ahead forecasts $f_t = E(Y_t|D_t)$ in the matrix `f` contained in the output of the function `dlmFilter`, and we use the first column of this matrix to compare the results with the actual data of the nominal GDP. Since the forecast observation f_t is a linear function of the filtering mean m_{t-1}, what we need to evaluate on the basis of the graph is the degree of sensitivity of f_t to an unexpected observation Y_{t-1}. The result obtained is illustrated in Figure 4.8. It is easily visible how the one-step-ahead forecasts adapt very rapidly to the sudden changes of the observations, which leads us to obtain quite accurate forecasts.

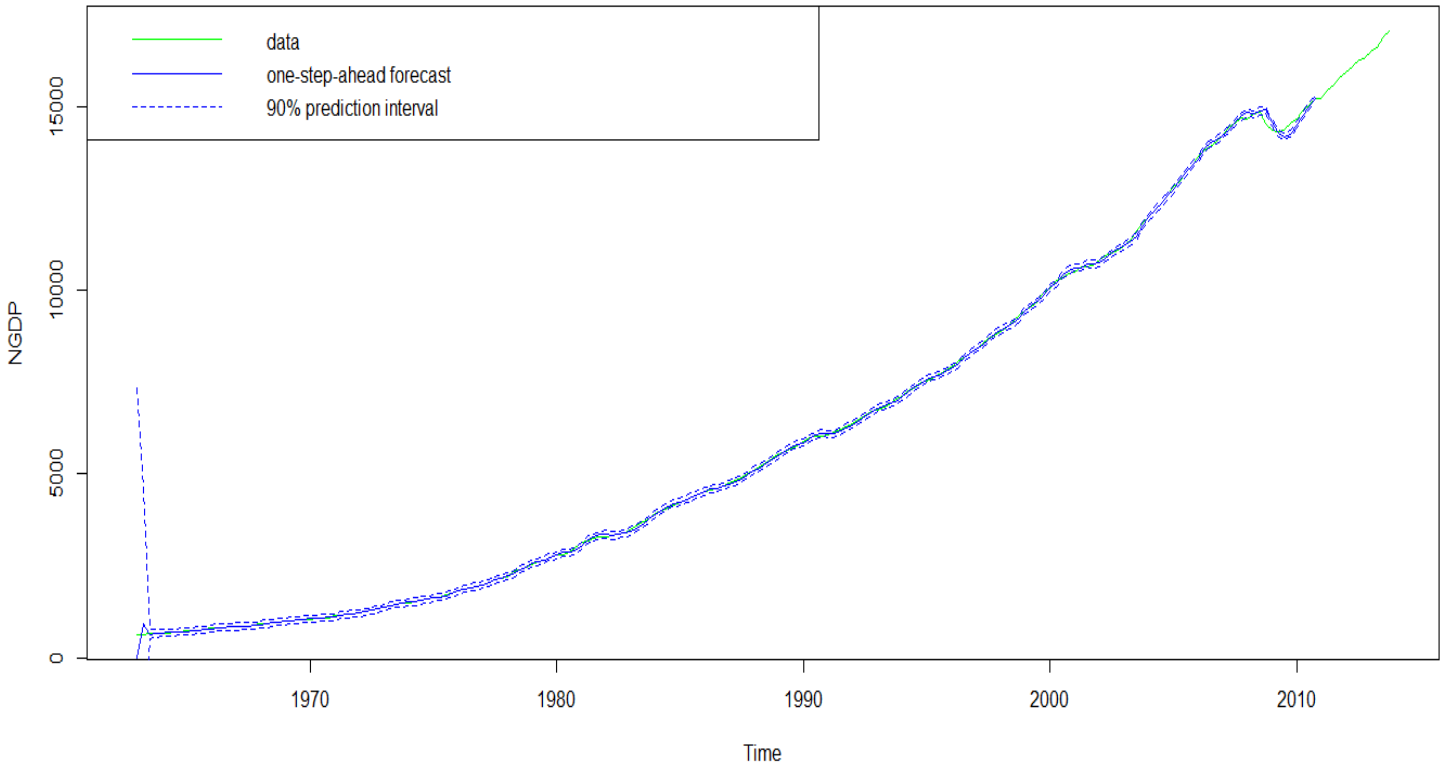

Figure 4.8. One-step-ahead forecasts of the nominal GDP (SUTSE model)

Let us now pursue with the analysis of the following forecast performance measures: the Root Mean Square Error, defined by:

$$\sqrt{\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h} \quad ,$$

the Mean Absolute Error, defined by:

$$\sqrt{\sum_{t=T+1}^{T+h} |\hat{y}_t - y_t| / h} \quad ,$$

and the Theil Inequality Coefficient, defined by:

$$\frac{\sqrt{\sum_{t=T+1}^{T+h}(\hat{y}_t - y_t)^2/h}}{\sqrt{\sum_{t=T+1}^{T+h}\hat{y}_t^{\,2}/h} + \sqrt{\sum_{t=T+1}^{T+h}y_t^2/h}}\,.$$

In the piece of code reported hereafter, we compute these indicators based on series of the one-step-ahead forecasts previously obtained, excluding from the computation the very first observations, whose forecast is still affected by the prior distribution of the model parameters.

```
IN-SAMPLE FORECAST
Root Mean Square Error
57.39284
Mean Absolute Error
5.731953
Theil Inequality Coefficient
0.003838805
```

The first two indicators are expressed in the measurement unit of the dependent variable y_t, whose forecasts we are interested in evaluating, and therefore do not provide us with much information if considered inside a single forecast, and will turn out to be more useful when comparing this forecast with the ones provided by the alternative models we will propose. Anyway, since the values of the American nominal GDP range from about 500 to more than 15000 (in billions of dollars), with a root mean square error equal to 57.39 and a mean absolute error of 5.73 we can reasonably deduce that this model presents a quite good predictive capability. On the contrary, the Theil inequality coefficient is comprised between 0 and 1, and the closer it is to 0, the better the forecast. Thus, it provides a measure which is invariant in respect to the scale of the interest variable. In our case, in particular, this coefficient is equal to 0.00384, which is very close to 0: this indicates that the one-step-ahead forecasts provided by this model only present slight discrepancies with the series of the actual data.

We will now take into account the forecast errors ϵ_t, also known as *innovations*. This terminology is justified by the expression $Y_t = f_t + \epsilon_t$, since a given observation Y_t can be represented as the sum of f_t, the component which is predictable based on past observations, and ϵ_t, the component which is independent of the past and consequently provides the actually new information which is contained in the observation Y_t.

If the assumptions underlying the model were correct, the sequence $\epsilon_1, \dots, \epsilon_t$ of the innovations computed from the data should look like a white noise. In order to verify this

circumstance, we report in Figure 4.9 a plot of the empirical autocorrelation function of the innovations.

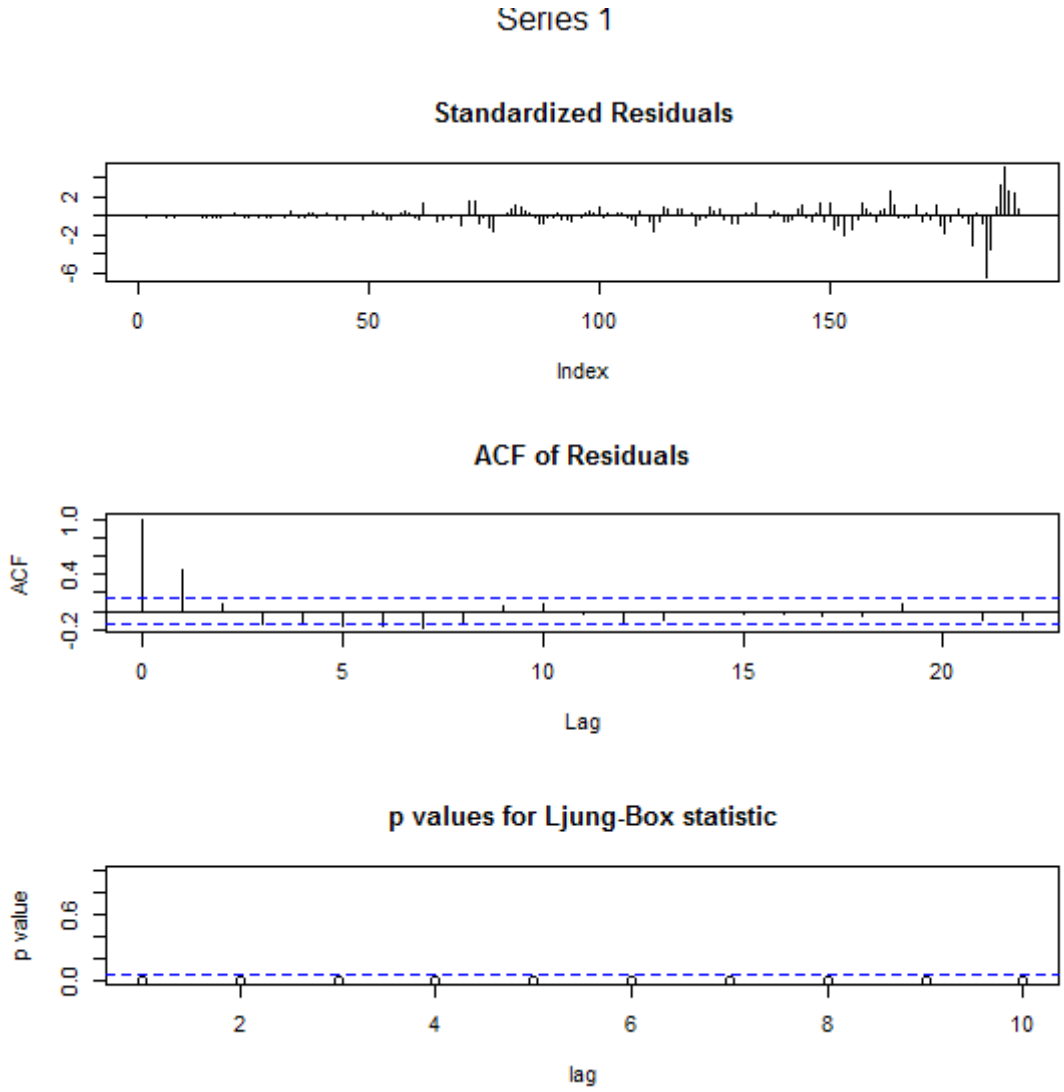

Figure 4.9. Diagnostic plots for the SUTSE model

The diagnostic plots clearly reveal the presence of residual heteroskedasticity, which can be justified by the fact the GDP itself presents a higher degree of variability in the most recent period considered, and autocorrelation, which might suggest the existence of a variability component that is not explained by the model.

We can now pursue with the analysis of the predictive capability of this model for what concerns the out-of-sample data. In order to do this, we will perform a forecasting of the behaviour of the nominal GDP in the last three years of the available data set, which were not included in the sample utilised for the estimation of the model parameters, and will compare the forecast values with the actual values. The fundamental function we have utilised for this purpose is

`dlmForecast`, which computes the expected value and the variance of the future system states and observations. In Figure 4.10 we present a plot illustrating the period of time going from January 2009 to October 2013: in particular, in the time range comprised in the test set (from October 2011 until October 2013) it shows a graphical comparison between the actual values of the nominal GDP during the last three years of the sample and the corresponding forecast values accompanied by the respective 90% probability intervals. Moreover, we report hereafter the forecast performance measures for the out-of-sample data associated to this model.

```
OUT-OF-SAMPLE FORECAST
Root Mean Square Error
289.0009
Mean Absolute Error
16.47265
Theil Inequality Coefficient
0.008867184
```

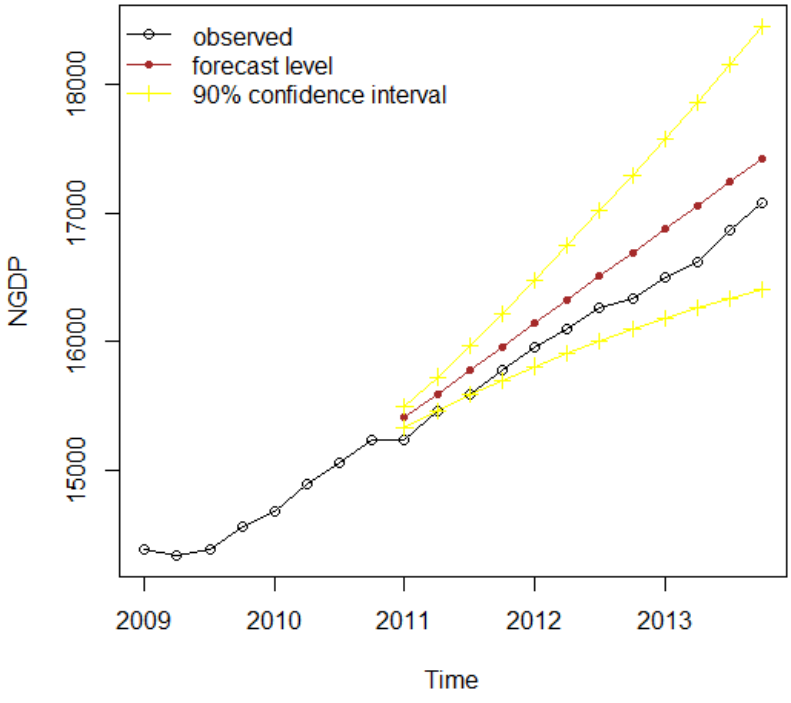

Figure 4.10. Forecast level for the nominal GDP (SUTSE model)

Both from the consideration of the forecast performance measures computed in the code and from a visual comparison of the forecast curve with the actual one it is possible to state that the predictive precision provided by this model is quite satisfying. In fact, a Root Mean Square Error of

289.00 and Mean Absolute Error of 16.47 denote a very slight discrepancy between the forecast curve and the actual one, since, in the portion of the sample under analysis, the values of the GDP vary approximately from 15500 to 17000, quite high values if compared with the forecast error. This is confirmed by the value of the Theil Inequality Coefficient, which is very close to 0.

Moreover, if we observe the graph, with the exception of the first three observations of the test set, all the actual values, represented by the black curve, are included inside the confidence intervals, which means that the hypothesis that the forecast values do not remarkably wander off the actual ones can be accepted with a significance level of 90%.

4.7. SUTSE model with seasonality

In this section we will present our second model proposal, consisting of a SUTSE model similar to the one just described, with the addition of a seasonal component concerning the T-Bill rate. As we have already mentioned in the introduction, we believed it was important to verify the presence of any seasonal component in the T-Bill rate because it was not possible to retrieve an already seasonally adjusted time series for this variable.

In this case, for all the variables considered, we assume the vector of the unobservable states to be equal to

$$\theta_t = (\mu_t, \beta_t, s_t^{(1)}, s_t^{(2)}, s_t^{(3)})' \quad ,$$

where μ_t is the current level, β_t is the slope of the trend, $s_t^{(1)}, s_t^{(2)}$ and $s_t^{(3)}$ are the seasonal components related to the current quarter, previous quarter, and two quarters back. Thus, the observation at time *t* is given by

$$y_t = \mu_t + s_t^{(1)} + v_t \quad , \qquad v_t \sim N(0, \sigma^2) \quad .$$

The dynamics of the unobservable states is described by the following equations:

$$\mu_t = \mu_{t-1} + \beta_{t-1} + w_t^{\mu} \quad , \qquad w_t^{\mu} \sim N(0, \sigma_{\mu}^2)$$

$$\beta_t = \beta_{t-1} + w_t^{\beta} \quad , \qquad w_t^{\beta} \sim N(0, \sigma_{\beta}^2)$$

$$s_t^{(1)} = -s_{t-1}^{(1)} - s_{t-1}^{(2)} - s_{t-1}^{(3)} + w_t^{s} \quad , \qquad w_t^{s} \sim N(0, \sigma_s^2)$$

$$s_t^{(2)} = s_{t-1}^{(1)} \quad ,$$

$$s_t^{(3)} = s_{t-1}^{(2)} \quad .$$

In the DLM representation of this model, we obtain that

$$V = [\sigma^2] \quad ,$$

$$W = \begin{bmatrix} \sigma_{\mu}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\beta}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

As we want the seasonal component to be only related to the T-Bill rate, we assume the variance σ_s^2 to be strictly positive for the T-Bill, and null for the other variables.

The code utilised for the construction follows the same structure of the SUTSE model without seasonality. The only difference consists in the inclusion of the variance related to the seasonal component of the T-Bill rate, that we have identified with `varSeasTbill`.

We report hereafter the variance parameters estimated by means of the maximum likelihood method.

```
V
367.27726413 0.08903564 0.14523862 0.02939057
0.08903564   0.14095736 0.08226716 0.20005605
0.14523862   0.08226716 0.16031025 0.13129468
0.02939057   0.20005605 0.13129468 0.32597611

W
0.492 0.081  0.082  0.015 0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0.081 0.337  0.303  0.271 0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0.082 0.303  0.341 -0.128 0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0.015 0.271 -0.128  0.486 0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     2.72e+03 0.125 0.059 0.005 0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     1.25e-01 0.375 0.140 0.247 0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     5.92e-02 0.140 0.230 0.211 0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     5.25e-03 0.247 0.211 0.244 0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0.138 0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
0     0      0      0     0        0     0     0     0 0 0 0     0 0 0 0 0 0 0 0
```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

4.8. Filtering

In this section we compute the filtering distribution based on the DLM including a seasonal component, and, in Figure 4.11, we report a graph of the filtered level of the GDP time series.

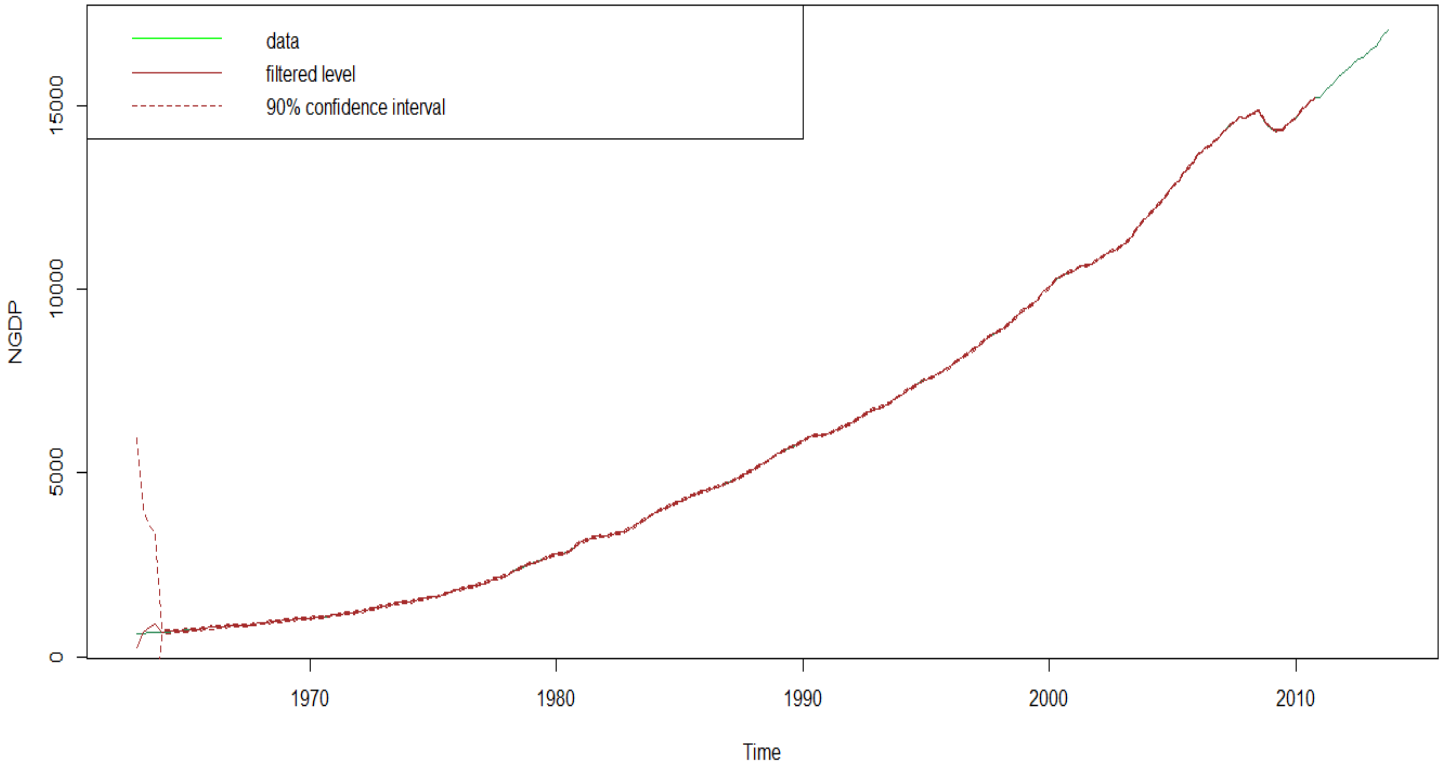

Figure 4.11. Nominal GDP with filtered level (SUTSE + seasonal model)

Similarly to what we obtained with the model previously analysed, we can notice how, even in this case, the series of the filtered levels possesses a quite good adherence to the actual data.

4.9. Evaluation of the predictive performance

In this section we will evaluate the predictive capability of the SUTSE model including a seasonal component, by exploiting the same indicators used in the case of the simple SUTSE model.

An analysis of the one-step-ahead forecasts provided the following results.

```
IN-SAMPLE FORECAST
Root Mean Square Error
51.43722
Mean Absolute Error
5.660481
Theil Inequality Coefficient
0.003413109
```

If we compare these performance measures with the ones related to the one-step-ahead forecasts provided by the simple SUTSE model, previously analysed, we can notice that all the indicators considered are in favour of the model including a seasonal component, even though the difference may be judged quite negligible.

In Figure 4.12 a graphical representation of the one-step-ahead forecasts is provided, from which it appears clearly that, even in this case, the model seems to possess a very good predictive capability.

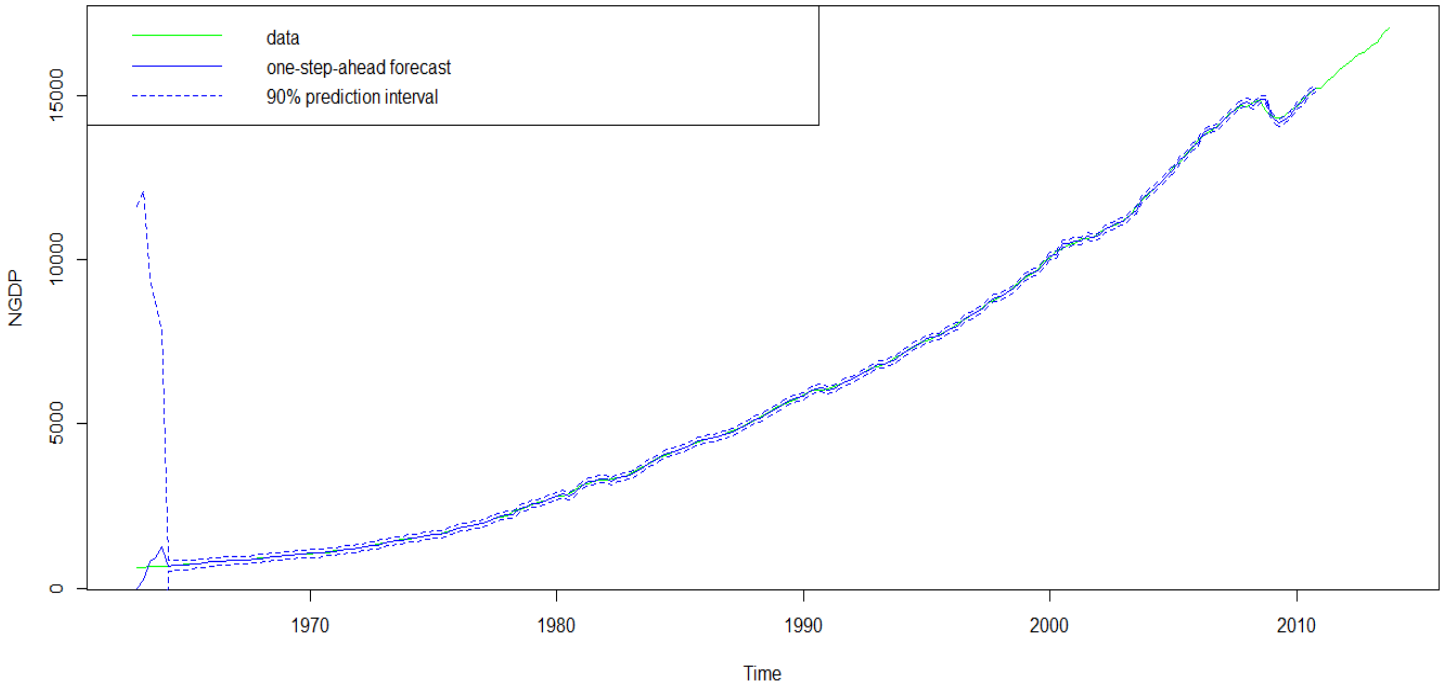

Figure 4.12. One-step-ahead forecasts of the nominal GDP (SUTSE + seasonal model)

In Figure 4.13 we present some residual diagnostic plots, similarly to what we did in the case of the previous model.

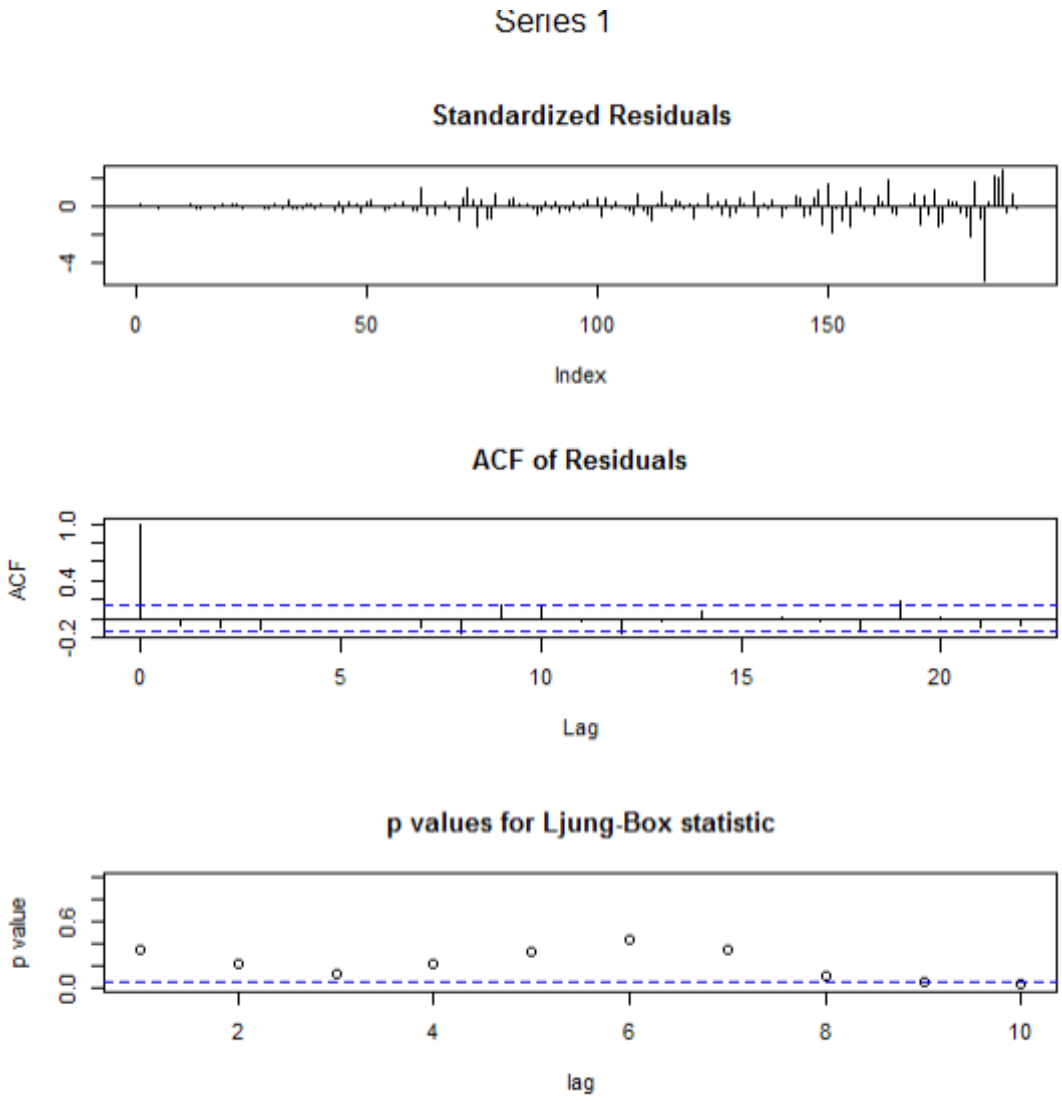

Figure 4.13. Diagnostic plots for the SUTSE model including a seasonal component

Similarly to what happened in the case of the simple SUTSE model, the plots reveal a clear presence of heteroskedasticity. As compensation, differently from the model previously proposed, this one does not seem to present relevant departures from uncorelatedness.

Figure 4.14 illustrates a plot of the out-of-sample forecasts provided by the SUTSE model including a seasonal component, and the forecast performance measures associated to this model are reported hereafter.

```
OUT-OF-SAMPLE FORECAST
Root Mean Square Error
232.1847
Mean Absolute Error
14.82537
Theil Inequality Coefficient
```

0.007135454

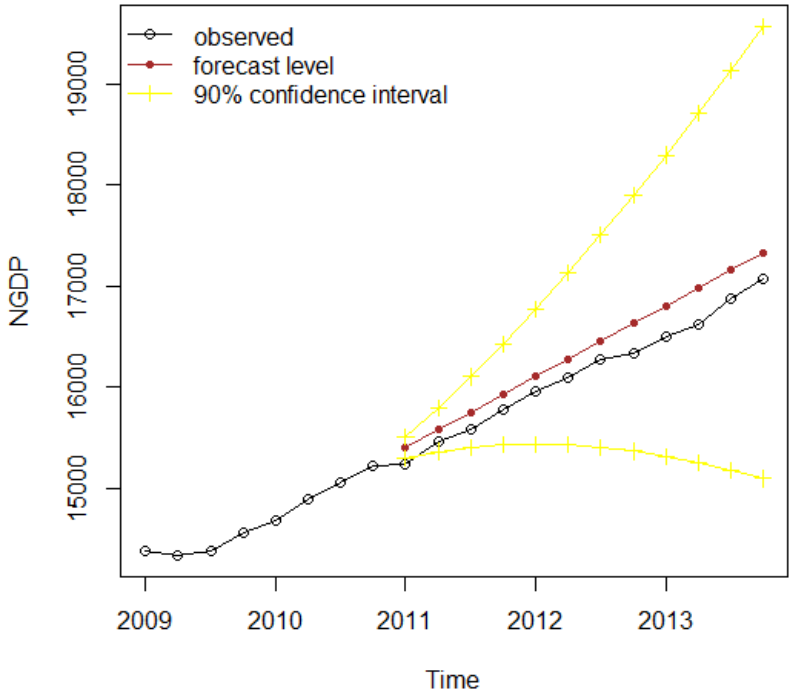

Figure 4.14. Forecast level for the nominal GDP (SUTSE + seasonal model)

From the plot we can notice how, in this case, the credible region includes the actual GDP values even with the only exception of the first observation, while, with the simple SUTSE model, the credible region did not include the first three observations. Furthermore, if we consider the forecast performance measures related to the out-of-sample forecast, we can notice how all of them denote that the forecast error of the SUTSE model including a seasonal component is smaller, if we compare these results with the ones obtained from the previous model.

4.10. Multiple regression DLM

Since the relationship between the explicative variables we have chosen and the American GDP has been verified in previous studies, and has been confirmed by the significance level of these variables in the static linear regression model, as a third model proposal, we decided to construct a dynamic linear regression model, whose main feature is that the coefficients are allowed to change over time.

The DLM representation of the model we want to build is constituted by the following set of equations:

$$\begin{aligned} y_t &= \alpha_t + X_t \beta_t + v_t, & v_t &\sim N(0, V) \\ \alpha_t &= \alpha_{t-1} + w_{\alpha,t}, & w_{\alpha,t} &\sim N(0, W_\alpha) \\ \beta_t &= \beta_{t-1} + w_{\beta,t}, & w_{\beta,t} &\sim N(0, W_\beta) \end{aligned}$$

where the system state is represented by the vector $\theta_t = (\alpha_t, \beta_t)'$. In particular, β_t is a vector containing the regression coefficients related to the three explicative variables at time *t*, and X_t is the matrix of the covariates $(x_{1,t}, x_{2,t}, x_{3,t})$ at time *t*. In this case, the matrix F_t is time-varying and is given by $F_t = [1, x_{1,t}, x_{2,t}, x_{3,t}]$.

The estimated observational and system variances are as follows.

```
V
0.7330774

W
0.9013384  0.00000000  0.00000000  0.00000000
0.0000000  0.02140005 -0.06606129  0.09459783
0.0000000 -0.06606129  0.22187471 -0.22540229
0.0000000  0.09459783 -0.22540229  0.64054191
```

4.11. Evaluation of the predictive performance

It is important to evidence that, in the case of the multiple regression DLM, the state vector is not composed by the means and the slopes of the considered variables at a given time *t*, but consists of the intercept and the regression coefficients related to the different explicative variables. Thus, for our purposes, it would not be so meaningful to report the results of the filtering distribution, because these results could only be expressed in terms of coefficients. Therefore, we decided to analyse directly the distribution of the one-step-ahead forecasts of the GDP.

```
IN-SAMPLE FORECAST
Root Mean Square Error
58.32701
Mean Absolute Error
6.223159
Theil Inequality Coefficient
0.003918419
```

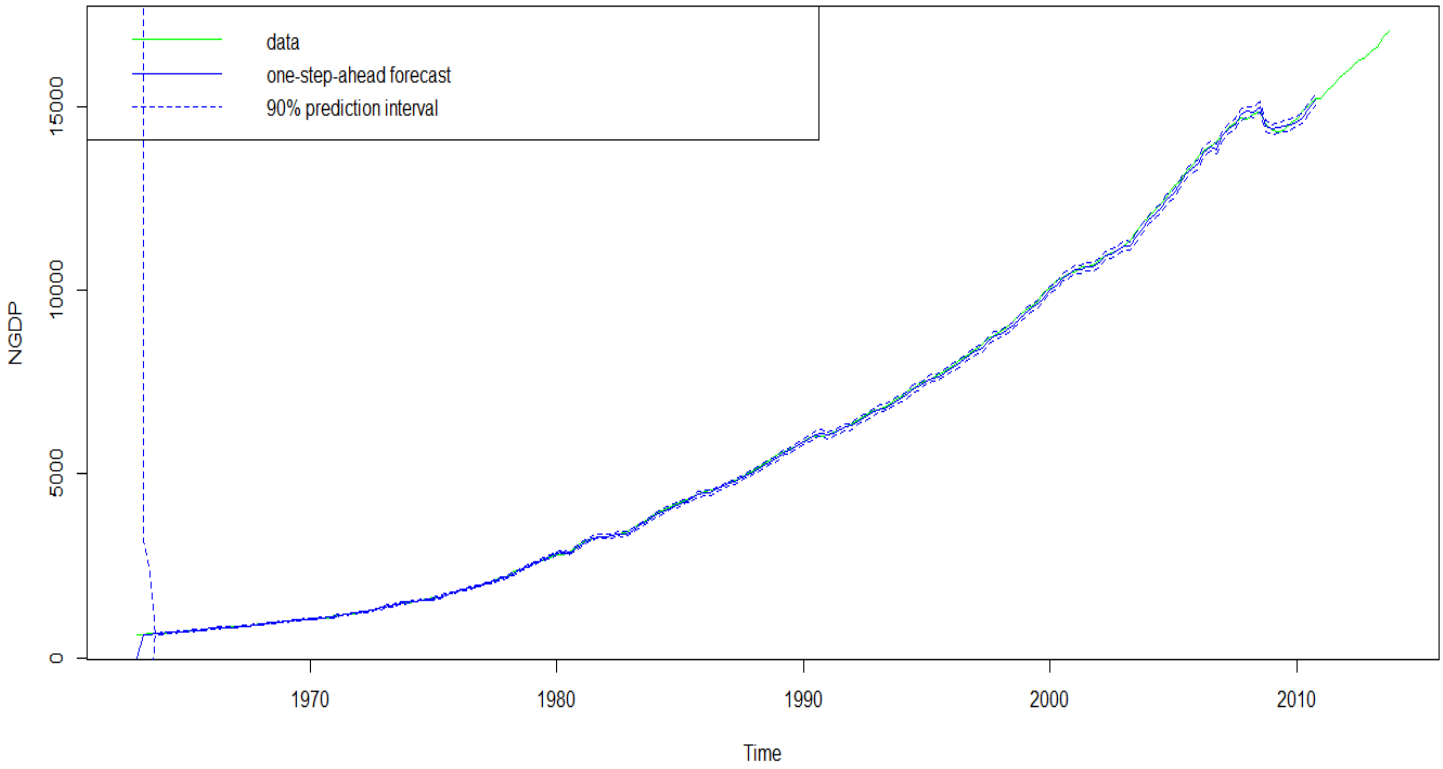

Figure 4.15. One-step-ahead forecasts of the nominal GDP (multiple regression model)

If we observe the series of the forecast values provided in Figure 4.15, this model seems to possess a predictive capability quite similar to the one of the models previously proposed. Nevertheless, if we consider the forecast performance measures related to the one-step-ahead forecast, we can see how the results obtained from this model reveal a lower predictive precision.

Moreover, if we analyse the plot of the residuals illustrated in Figure 4.16, we can notice a remarkable presence of autocorrelation.

The results obtained from the out-of-sample forecast are presented in Figure 4.17. The plot reveals that, even for what concerns the out-of-sample data, this model provides forecast values which differ from the actual values of the nominal GDP in larger measure if compared to what happened in the case of the two SUTSE models. This is also confirmed by the performance measures provided hereafter.

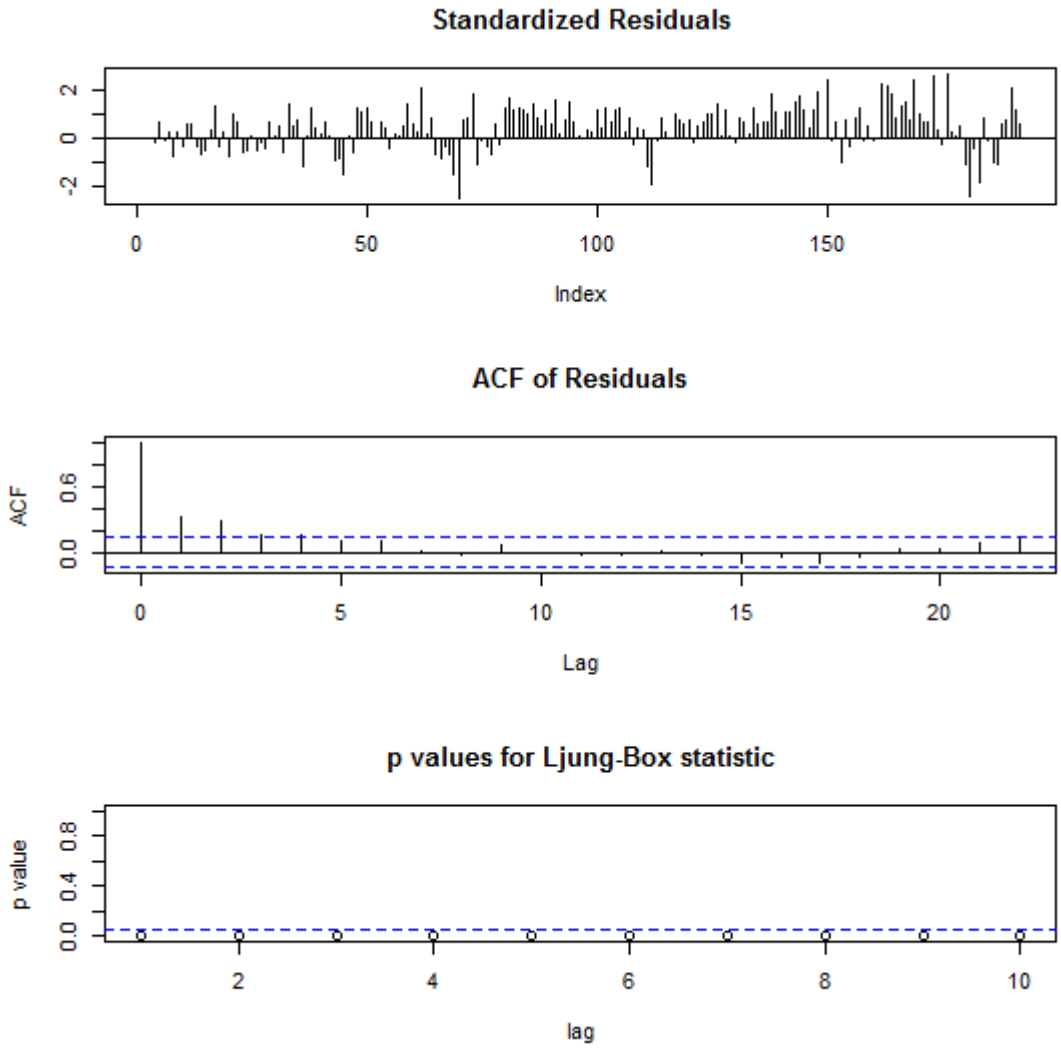

Figure 4.16. Diagnostic plots for the multiple regression DLM

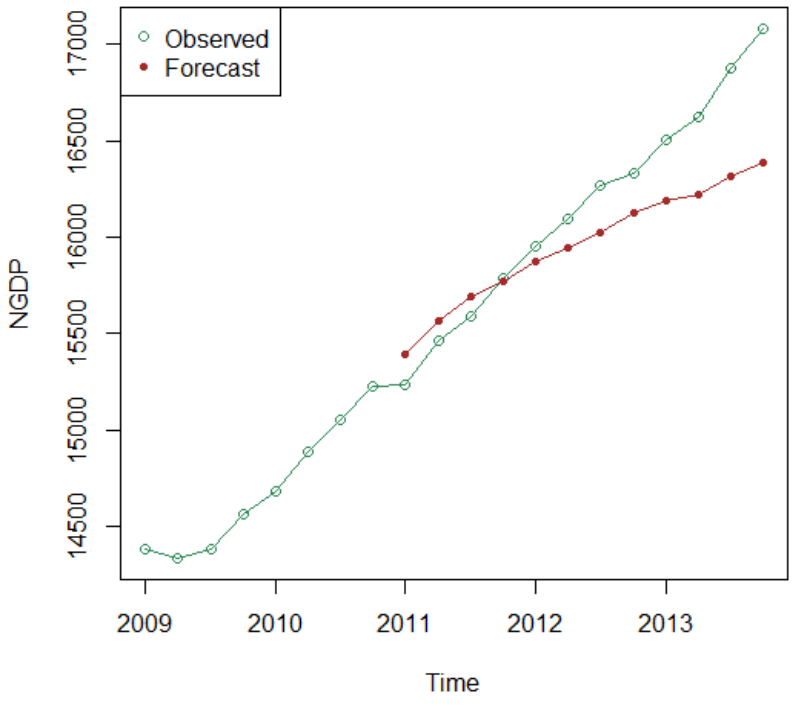

Figure 4.17. Forecast level for the nominal GDP (multiple regression DLM)

```
OUT-OF-SAMPLE FORECAST
Root Mean Square Error
321.0583
Mean Absolute Error
15.9085
Theil Inequality Coefficient
0.009995773
```

4.12. Multiple regression DLM with lagged regressors

We now want to verify whether the inclusion of lagged values of the explicative variables allows us to improve the performance of the multiple regression model. Therefore, in this section we will construct a DLM having a similar structure to the one previously described, with the difference that the matrix of regressors X_t is substituted by the corresponding lagged matrix X_{t-1}. Formally speaking, the DLM representation of the model we want to build is as follows.

$$\begin{aligned} y_t &= \alpha_t + X_{t-1}\beta_t + v_t, & v_t &\sim N(0, V) \\ \alpha_t &= \alpha_{t-1} + w_{\alpha,t}, & w_{\alpha,t} &\sim N(0, W_\alpha) \\ \beta_t &= \beta_{t-1} + w_{\beta,t}, & w_{\beta,t} &\sim N(0, W_\beta) \end{aligned}$$

In this case, being X_{t-1} the matrix of the covariates $(x_{1,t-1}, x_{2,t-1}, x_{3,t-1})$ at time $t-1$, the matrix F_t will be given by $[1, x_{1,t-1}, x_{2,t-1}, x_{3,t-1}]$.

The estimated observational and systematic variances, similarly to the ones obtained from the previous model, are as follows.

```
V
0.7330774

W
0.9013384  0.00000000  0.00000000  0.00000000
0.0000000  0.02140005 -0.06606129  0.09459783
0.0000000 -0.06606129  0.22187471 -0.22540229
0.0000000  0.09459783 -0.22540229  0.64054191
```

4.13. Evaluation of the predictive performance

Similarly to what we did for the previous model, we report in Figure 4.18 a plot of the one-step-ahead forecasts based on the multiple regression DLM based on the lagged regressors. The performance measures based on the one-step-ahead ahead forecasts are presented hereafter.

```
IN-SAMPLE FORECAST
Root Mean Square Error
82.31051
Mean Absolute Error
7.379575
Theil Inequality Coefficient
0.005531399
```

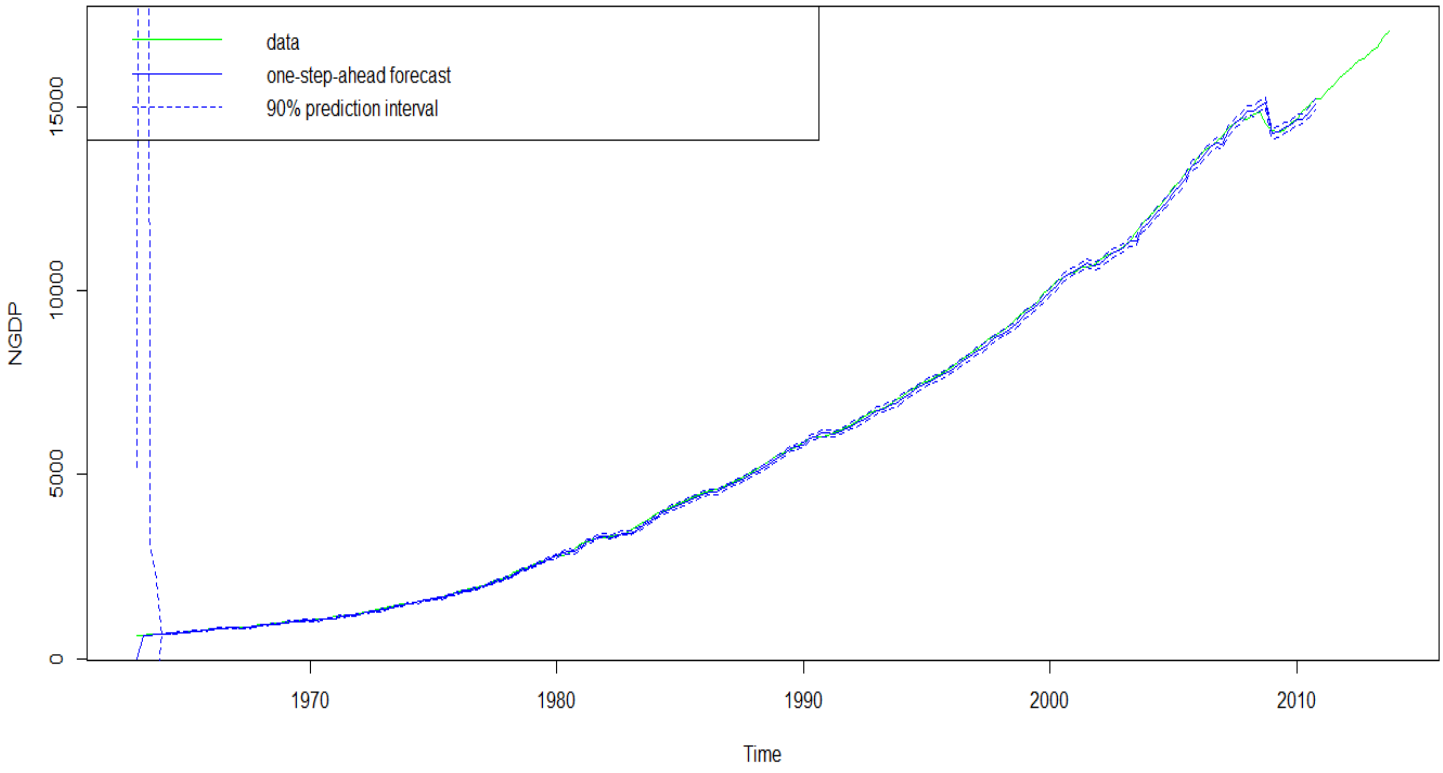

Figure 4.18. One-step-ahead forecasts for nominal GDP (lagged multiple regression DLM)

The plot does not present relevant discordances with the regression model based on contemporary regressors, even though the performances measures reveal a slightly lower predictive precision of the model based on lagged regressors.

Moreover, if we observe the residual diagnostic plot in Figure 4.19, it is possible to notice how this model does not solve the autocorrelation problem presented by the multiple regression model previously proposed.

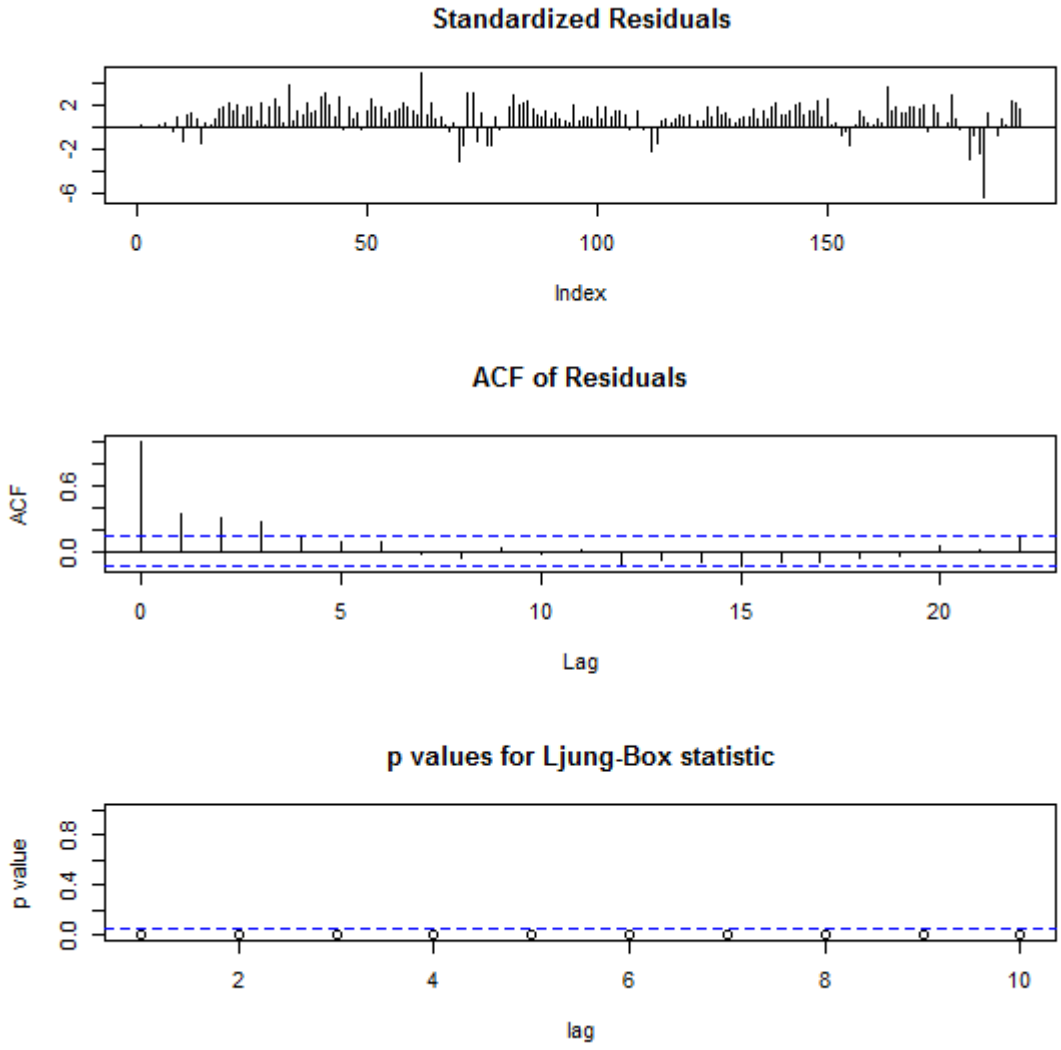

Figure 4.19. Diagnostic plots for the lagged multiple regression DLM

The results obtained form the out-of-sample forecast confirm this behaviour. In fact, both the graph in Figure 4.20 and the performance measures reported hereafter reveal a weaker predictive capability of this model, if compared to the previous multiple regression DLM.

```
OUT-OF-SAMPLE FORECAST
Root Mean Square Error
371.1428
Mean Absolute Error
16.83304
Theil Inequality Coefficient
```

0.01157599

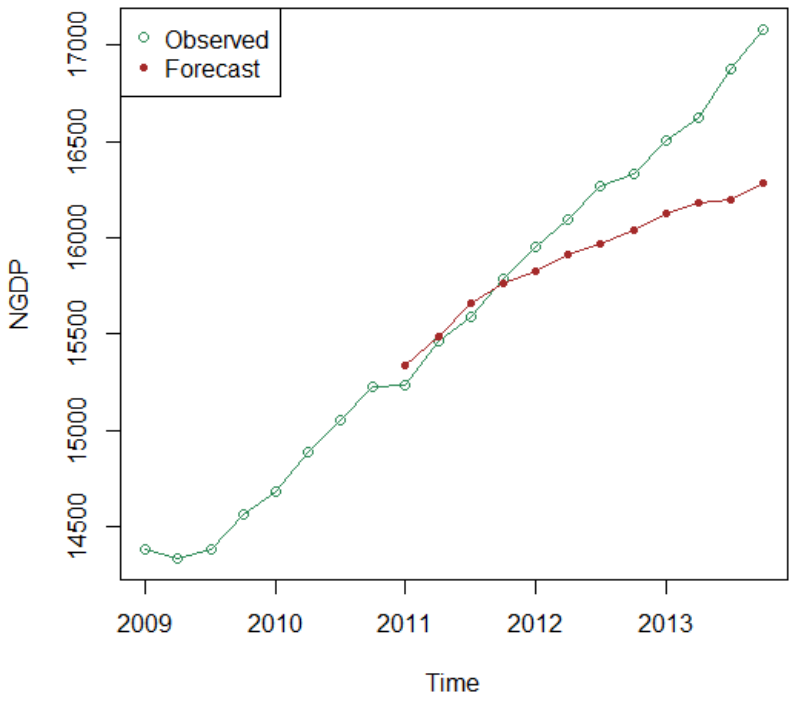

Figure 4.20. Forecast level for the nominal GDP (lagged multiple regression DLM)

4.14. Model selection and filtering

On the basis of the forecast performance measures considered for the three models we have proposed, we will choose as a benchmark for our further analyses the SUTSE model including a seasonal component.

In order to evaluate the accurateness of this model in relation to the other variables included in it, we provide in Figures 4.21 to 4.23 a graphical comparison between the actual data concerning the Industrial Production Index, the Consumer Price Index and the T-Bill rate and their respective filtered states, accompanied by the relevant 90% credible interval.

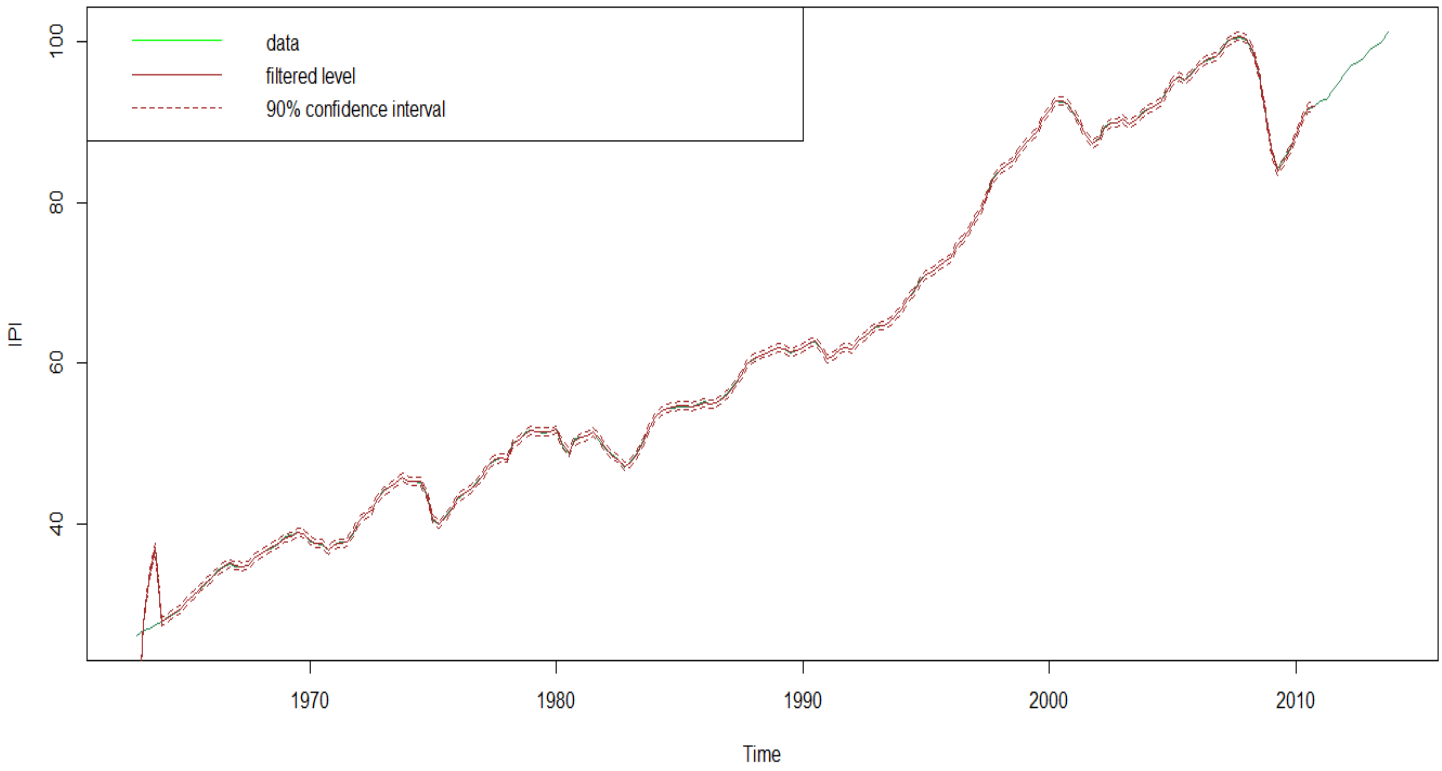

Figure 4.21. Industrial Production Index with filtered level

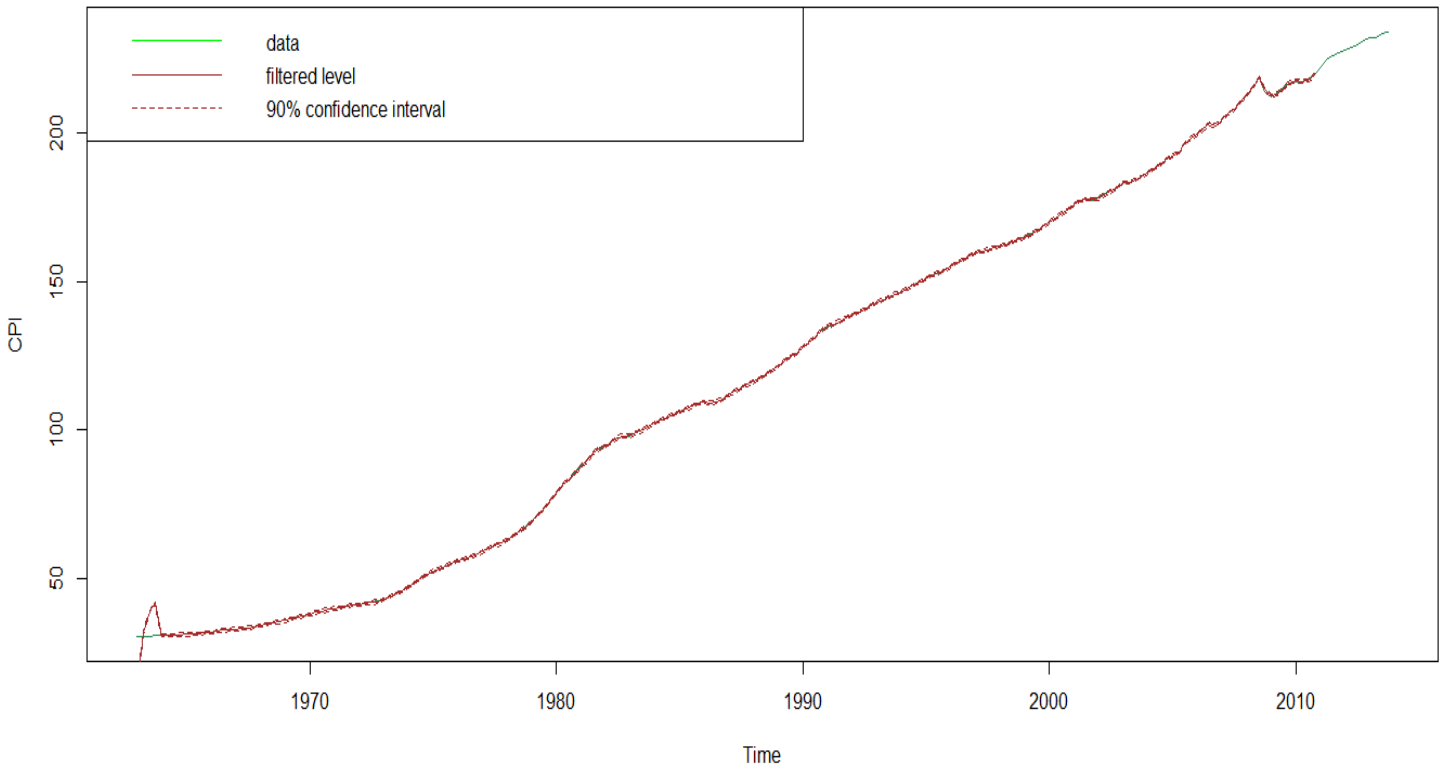

Figure 4.22. Consumer Price Index with filtered level

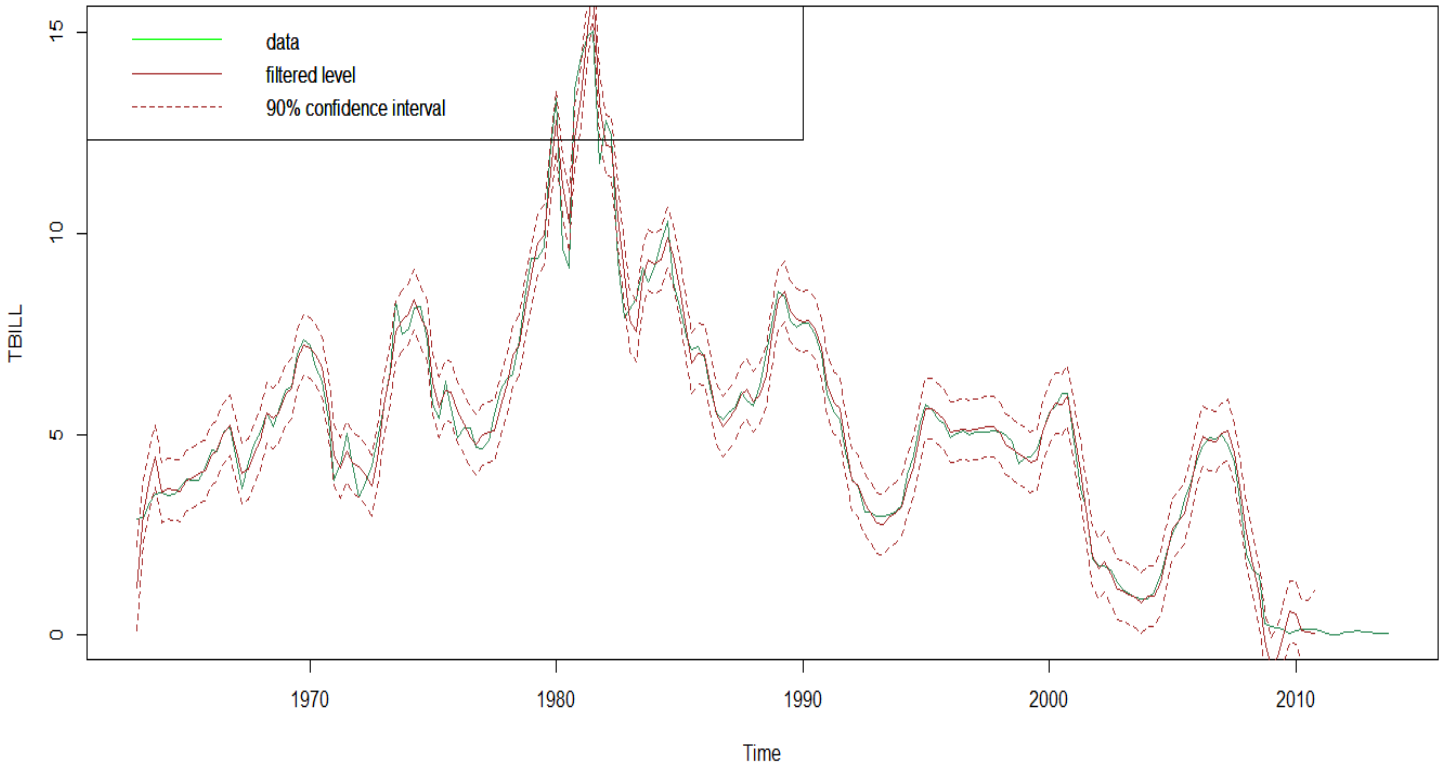

Figure 4.23. T-Bill rate with filtered level

From the graphs it appears clearly how in all the cases there is a good level of adherence between the series of the actual data and the relevant filtered level. The main difference among the three series resides in the width of the credible intervals.

In particular, in the case of the Industrial Production Index and of the Consumer Price Index, the actual series and the filtered levels overlap almost perfectly, and it is not easy to distinguish the credible interval, probably because of the reduced variance of the filtering distribution. In the case of the T-Bill rate, instead, this interval seems to be notably wider, even though the actual data series remains comprised between the upper and lower limit of the interval almost throughout the whole period considered.

4.15. Smoothing

Similarly to what we did as regards the filtering, in this section of the program, we will perform a *smoothing* for the American GDP and the three explicative variables under analysis. We recall

that the smoothing distribution at time t is equal to the conditional distribution of $\theta_{0:t}$ given the observations $y_{1:t}$, or, in an equivalent way, any one of its marginals, which means the conditional distribution of θ_s given $y_{1:t}$, for each $s \leq t$. As a natural consequence, when $s = t$ this marginal density coincides with the filtering density.

In the relevant code utilised, we exploit the function `dlmSmooth` in order to compute means and variances of the smoothing distributions, by using as an argument the "filtered DLM" previously obtained thanks to the function `dlmFilter`. Similarly to what we did previously, we compute the covariance matrices of the smoothing errors by reconstructing them departing from their singular value decomposition, which we are provided with by the function `dlmSmooth` thanks to the matrices `U.S` and `D.S`.

We illustrate in the Figures 4.24 to 4.27 the results obtained by means of the Kalman smoother.

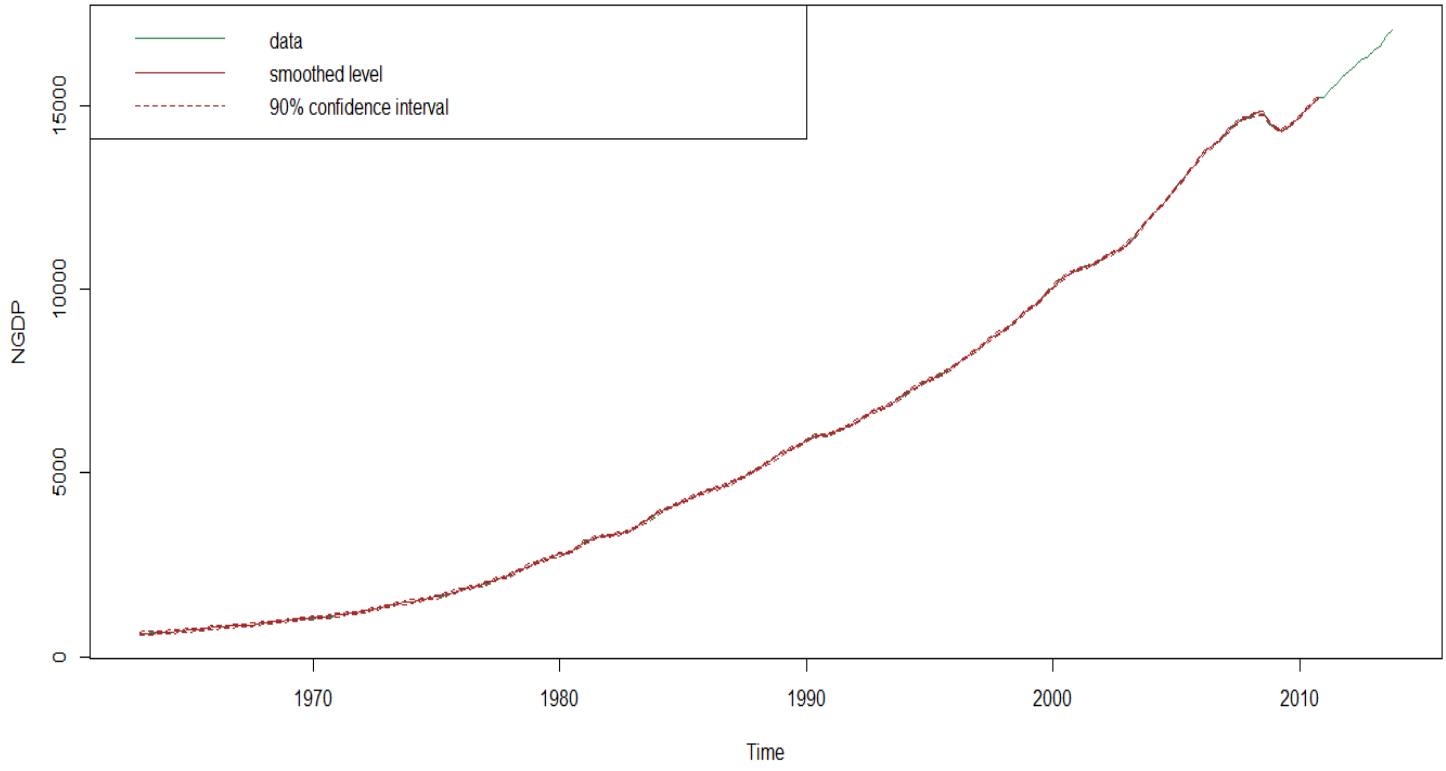

Figure 4.24. Nominal GDP with smoothed level

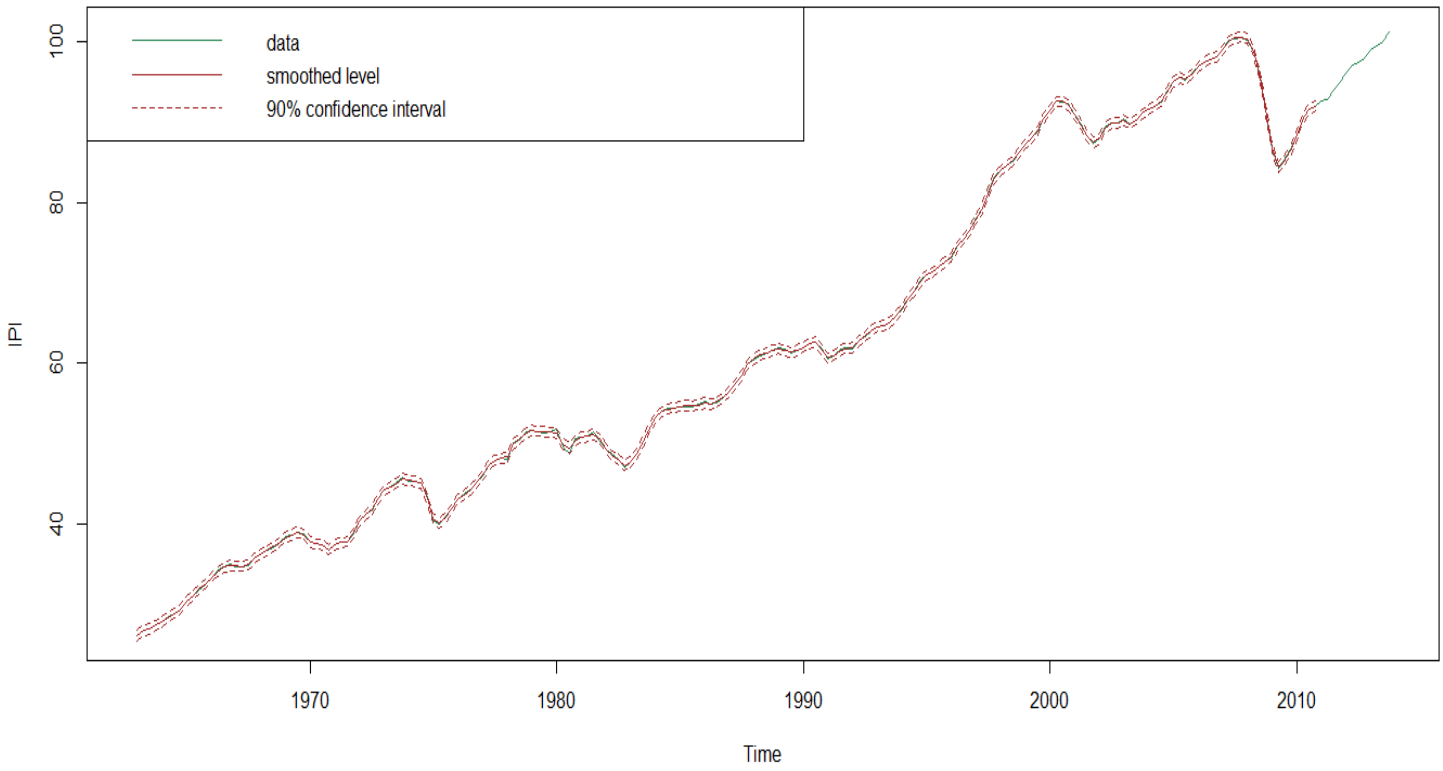

Figure 4.25. Industrial Production Index with smoothed level

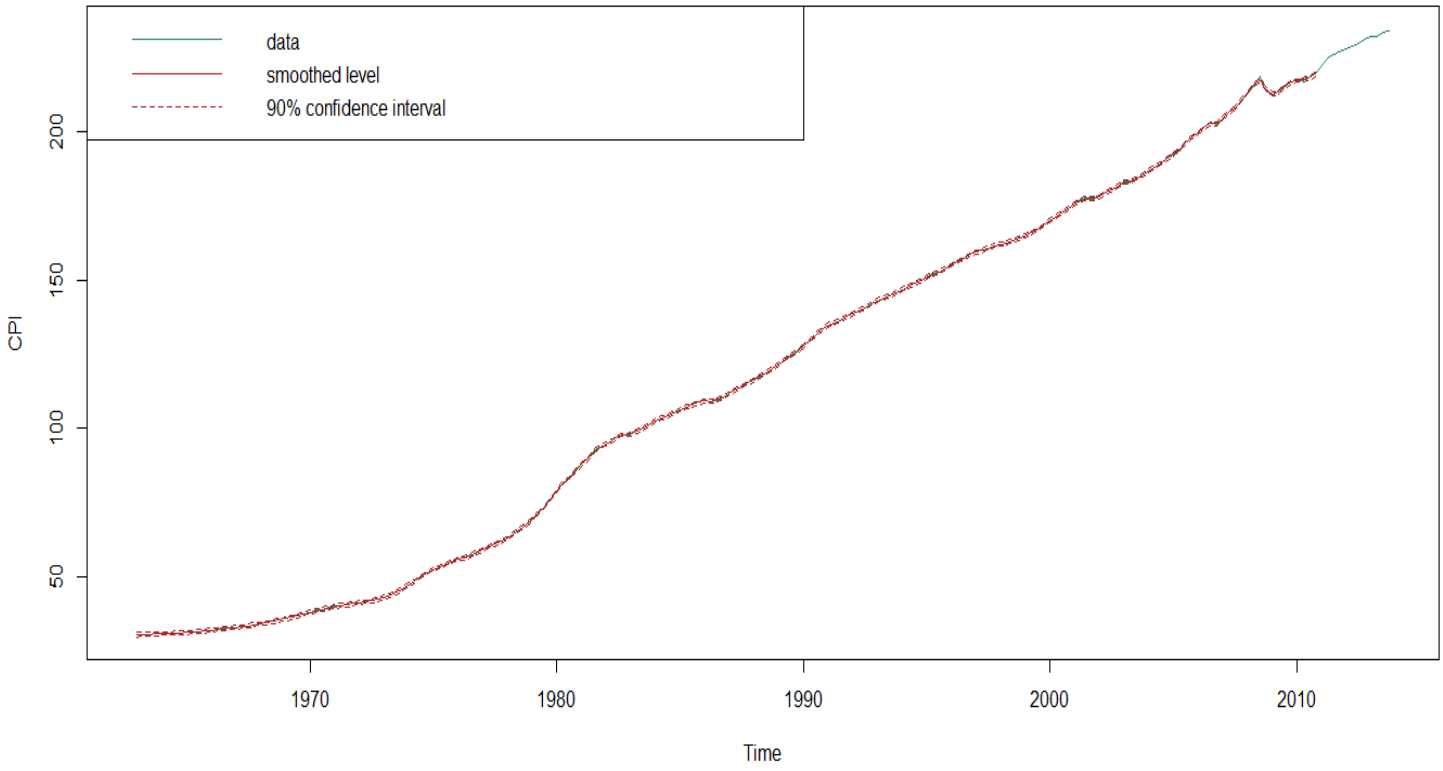

Figure 4.26. Consumer Price Index with smoothed level

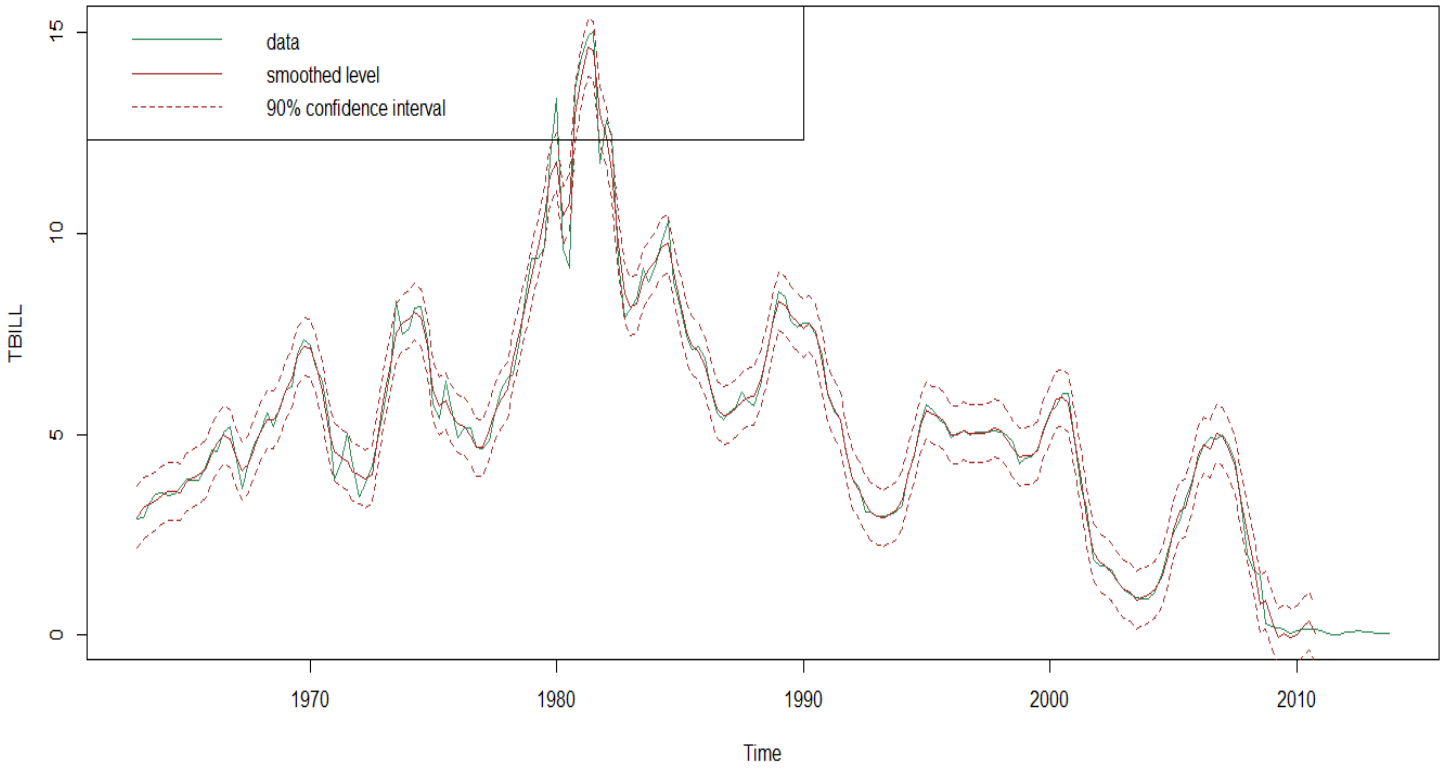

Figure 4.27. T-Bill rate with smoothed level

It is visible from the plots of all the series how there is an optimal fit of the smoothed level to the actual data. Furthermore, similarly to what happened in the case of filtering, the smoothed level of the American GDP and the indices related to the industrial production and the consumer price present very narrow probability intervals, while we have obtained quite the large intervals in the case of the 3-month T-Bill rate, due to the higher degree of volatility of this variable.

4.16. Forecasting

In the code which follows, for each one of the explicative variables considered we construct a similar graph to the ones utilised in order to evaluate the predictive capability of the different models as far as the GDP in concerned, and compute some forecast performance measures related to these variables. It is important to say that, in this case, some "absolute" performance measures must be utilised, which are not affected by the scale of the interest variable, because we cannot compare these results with the ones provided by different models for these variables. Therefore, we decided not to use the Root Mean Square Error and the Mean Absolute Error, as we have done for

the previous forecast evaluations, and to substitute these indicators with the Mean Absolute Percentage Error, given by:

$$100 \sum_{t=T+1}^{T+h} \left| \frac{\hat{y}_t - y_t}{y_t} \right| / h \quad ,$$

returning a percentage measure of the forecast error.

```
IPI
Mean Absolute Percentage Error
0.3680098
Theil Inequality Coefficient
0.002087563

CPI
Mean Absolute Percentage Error
1.583648
Theil Inequality Coefficient
0.008136327

TBILL
Mean Absolute Percentage Error
1379.146
Theil Inequality Coefficient
0.8367708
```

As regards the Industrial Production Index and the Consumer Price Index, both the performance indicators and the graphs illustrated in the Figures 4.28 and 4.29 show that the model provides quite reliable forecasts of their future trend. Instead, if we consider the T-Bill rate, we can notice how, even though the graph reported in Figure 4.30 seems to demonstrate ad adequate predictive precision, the performance measures denote a remarkable discrepancy between the series of the forecast values and the actual data. However, this circumstance might be justified by the reduced scale of the values of this time series.

In conclusion, we can state that the model possesses a satisfying predictive performance, both for what concerns the nominal GDP itself and the macroeconomic factors affecting its behaviour.

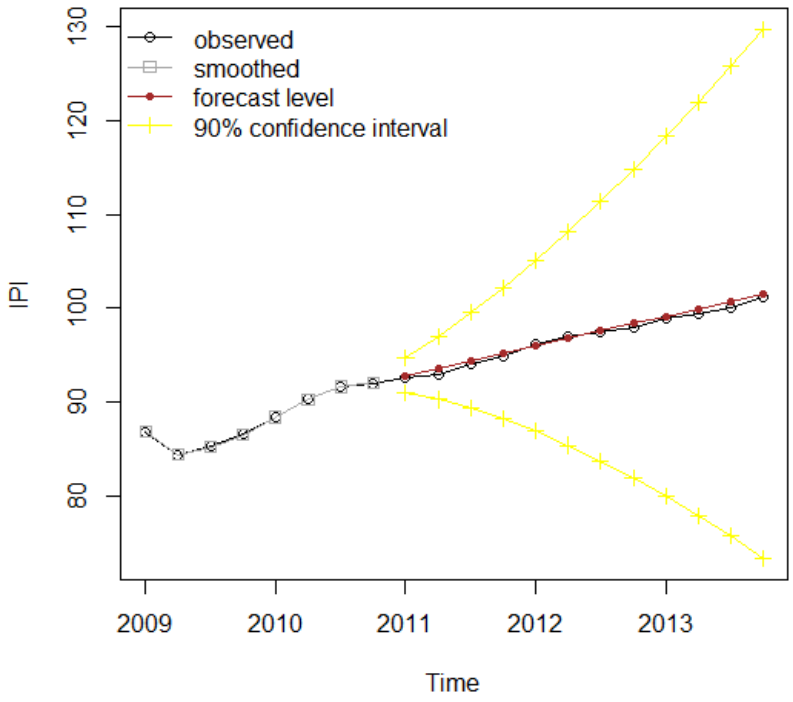

Figure 4.28. Forecast level of the Industrial Production Index

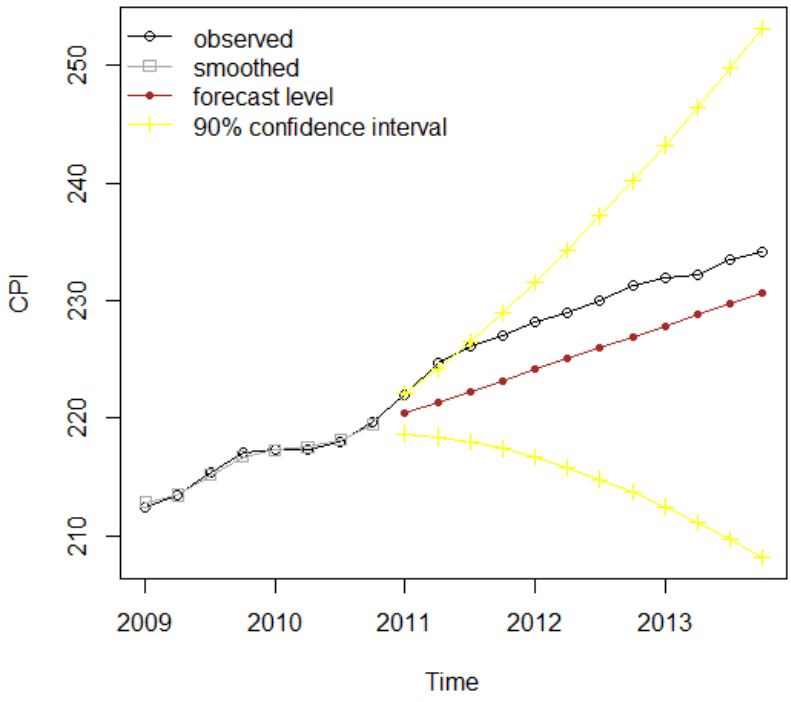

Figure 4.29. Forecast level of the Consumer Price Index

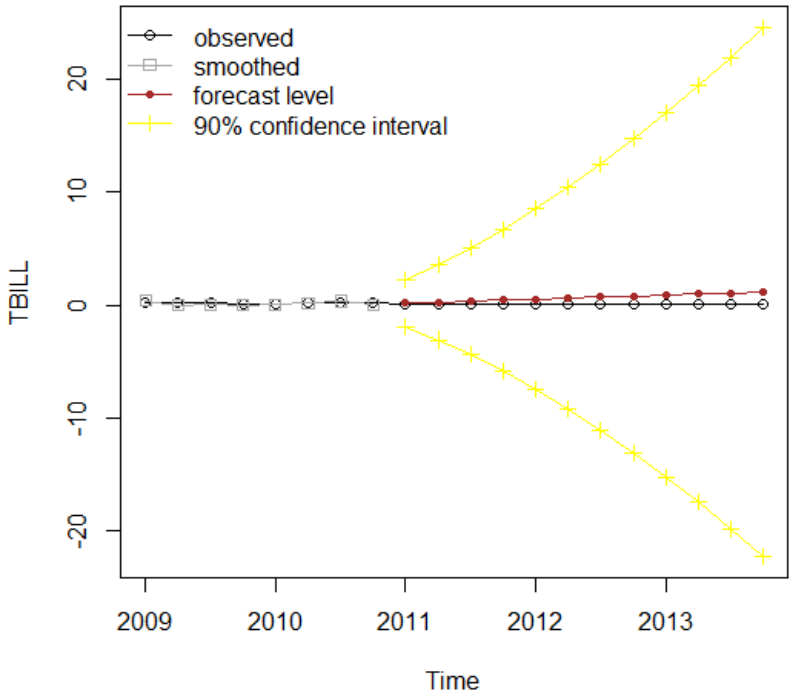

Figure 4.30. Forecast level of the T-Bill rate

4.17. Markov Chain Monte Carlo simulations

In this section we will provide an example of the usage of Gibbs sampling, which we will apply to the four time series we have considered, and will compare the predictive capability of the model we have previously analysed with the results obtained by estimating the unknown parameters by using a simulation algorithm.

In fact, since the objective of MCMC simulations consists in making inference about the joint posterior distribution of the unobservable states and the unknown parameters, we will now assume that the variances of the DLM are unknown, and will not take into account the values previously obtained by means of the maximum likelihood estimator. This kind of model is commonly referred to as **d-inverse-gamma** model.

We will anyway retain the assumption according to which the nominal GDP and the indices related to the industrial production and the consumer price can be modelled by using a simple linear growth DLM, while the T-Bill interest rate can be described by a linear growth model containing a seasonal component. Of course, all of these models should include a noise component.

As we have described in the section dedicated to the SUTSE model with seasonality, the unknown parameters in a DLM of this kind are constituted by the observational variance, indicated with

$$V=[\sigma^2] \quad ,$$

and the system variance, given by

$$W=\begin{bmatrix} \sigma_\mu^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\beta^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad ,$$

where $\sigma_\mu^2, \sigma_\beta^2$ and σ_s^2 respectively identify the variances of the level, the slope and the seasonal component of the variable considered.

Therefore, in our specific case, the unknown parameters correspond to the variances $\sigma^2, \sigma_\mu^2, \sigma_\beta^2$ and σ_s^2 . For the relevant precision measures, corresponding to the inverses of the variances themselves, we assume independent gamma prior distributions with means $a, a_{\theta,1}, a_{\theta,2}, a_{\theta,3}$ and variances $b, b_{\theta,1}, b_{\theta,2}, b_{\theta,3}$ respectively.

It is possible to verify that, by using the unobservable states as *latent* variables, a Gibbs sampler can be applied on the basis of the following full conditional densities:

$$\theta_{0:n} \sim N() \quad ,$$

$$\sigma^2 \sim IG\left(\frac{a^2}{b}+\frac{n}{2}, \frac{a}{b}+\frac{1}{2}SS_y\right) \quad ,$$

$$\sigma_\mu^2 \sim IG\left(\frac{a_{\theta,1}^2}{b_{\theta,1}}+\frac{n}{2}, \frac{a_{\theta,1}}{b_{\theta,1}}+\frac{1}{2}SS_{\theta,1}\right) \quad ,$$

$$\sigma_\beta^2 \sim IG\left(\frac{a_{\theta,2}^2}{b_{\theta,2}}+\frac{n}{2}, \frac{a_{\theta,2}}{b_{\theta,2}}+\frac{1}{2}SS_{\theta,2}\right) \quad ,$$

$$\sigma_s^2 \sim IG\left(\frac{a_{\theta,3}^2}{b_{\theta,3}}+\frac{n}{2}, \frac{a_{\theta,3}}{b_{\theta,3}}+\frac{1}{2}SS_{\theta,3}\right) \quad ,$$

where

$$SS_y=\sum_{t=1}^{t=n}(y_t-F_t\theta_t)^2 \quad ,$$

$$SS_{\theta,i}=\sum_{t=1}^{t=T}(\theta_{t,i}-(G_t\theta_{t-1})_i)^2 \quad , \qquad i = 1, 2, 3.$$

At the moment the R package for dynamic linear models does not include a function which is able to simulate the unknown parameters for a multivariate model. For this reason, in the piece of code reported below, we had to implement the Gibbs sampler by using the function `dlmGibbsDIG`, which can only deal with constant univariate DLMs. This function runs the sampling procedure on the basis of the full conditional densities just described, and includes inside its algorithm the forward filtering backward sampling phase, which implies that it is not necessary to derive explicitly the mean and the variance of the full conditional density of the states $\theta_{0:n}$, because they are automatically generated from their appropriate distribution thanks to this sampling procedure.

In the final part of the code, for each of the four series considered, we perform a 1100-iterations Gibbs sampling procedure, where the first 100 iterations are discarded as *burn-in*[1]. Subsequently, we provide a graphical illustration of the simulated values of the observation and system variances, accompanied by the plots of their relevant ergodic means, a graph of the simulated levels of the state vector for what concerns the training set, and an analysis of the predictive capability of a DLM built by using the parameter values resulting from this simulation algorithm.

For what concerns the nominal GDP, the results obtained from the simulation are as follows.

```
NGDP - GIBBS SAMPLING
AVERAGE OBSERVATIONAL AND SYSTEM VARIANCES
      V          W_μ         W_β
  0.022835   0.001694   0.144844

FORECAST PERFORMANCE MEASURES
Root Mean Square Error
217.6508
Mean Absolute Error
14.365
Theil Inequality Coefficient
0.006691617
```

The simulated observation and system variances, with their relevant ergodic means, and the simulated levels are respectively represented in the Figures 4.31 and 4.32.

[1] *Burn-in* is a colloquial term that refers to the practice of discarding a given number of iterations at the beginning of an MCMC algorithm. The meaning of the notion is that, for the burn-in period, the results obtained from the simulation are not retained, but after that period the simulations are run normally, using every iteration of the MCMC procedure.

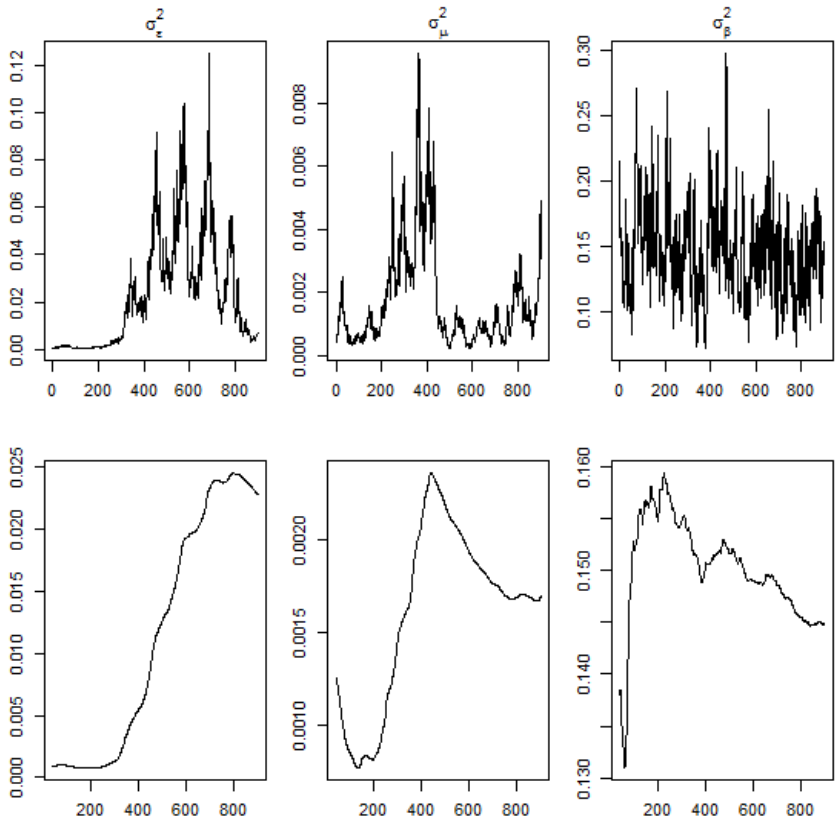

Figure 4.31. Plots of simulated variances (top) and ergodic means (bottom) for the nominal GDP

If we compare the average observational and system variances estimated by means of the Gibbs sampler, returned by the function `mcmcMean`, with the ones computed by using the maximum likelihood method in the treatment of the SUTSE model with seasonality, we can notice how the Gibbs sampler provides remarkably smaller values for these variances, which can be interpreted as a signal of the higher precision of this algorithm in estimating the states. However, if we consider the levels simulated by means of the Gibbs sampler and the ones computed by the Kalman filter (Figure 4.11), there does not appear to be any remarkable discrepancy between two state estimation methods, and both of them seem to provide an adequate data fitting.

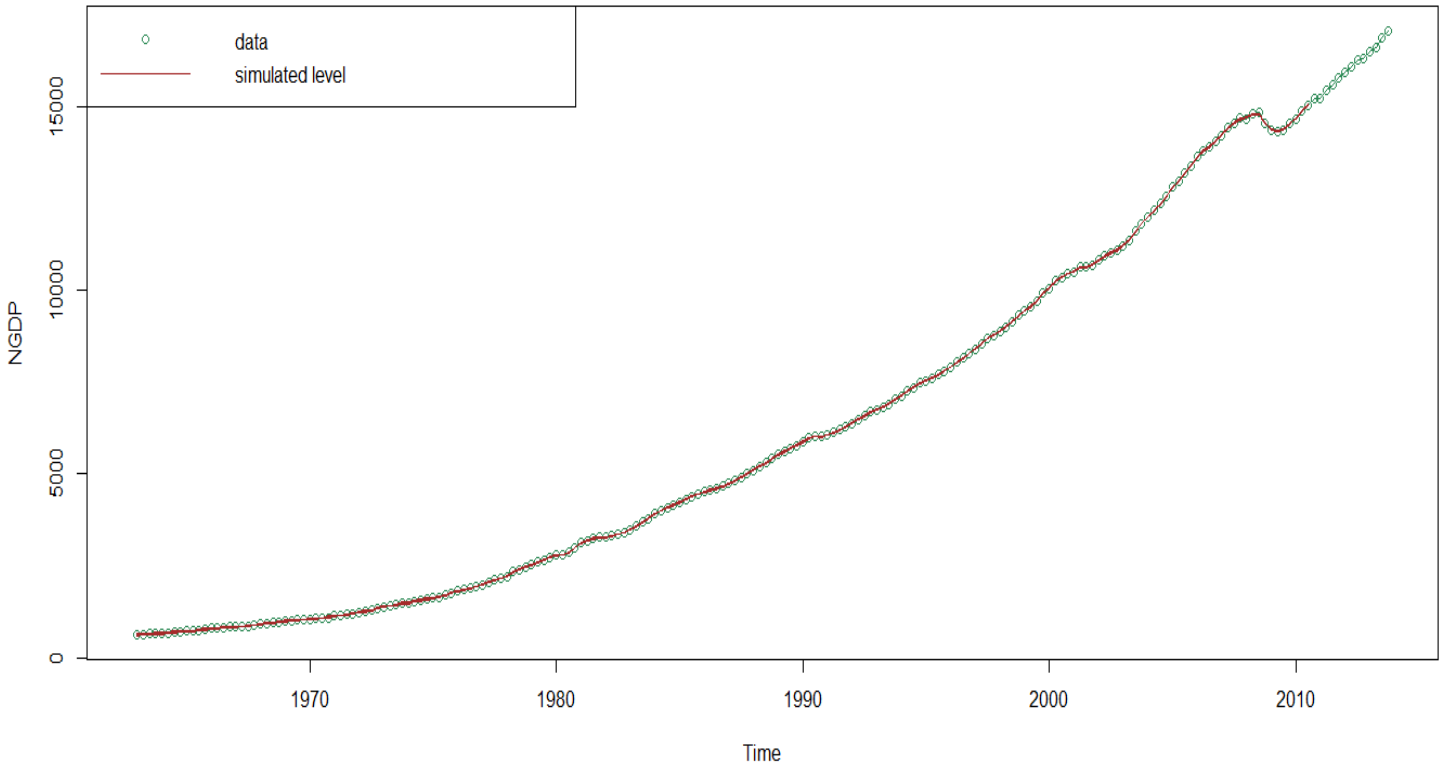

Figure 4.32. Nominal GDP levels simulated by the Gibbs sampler

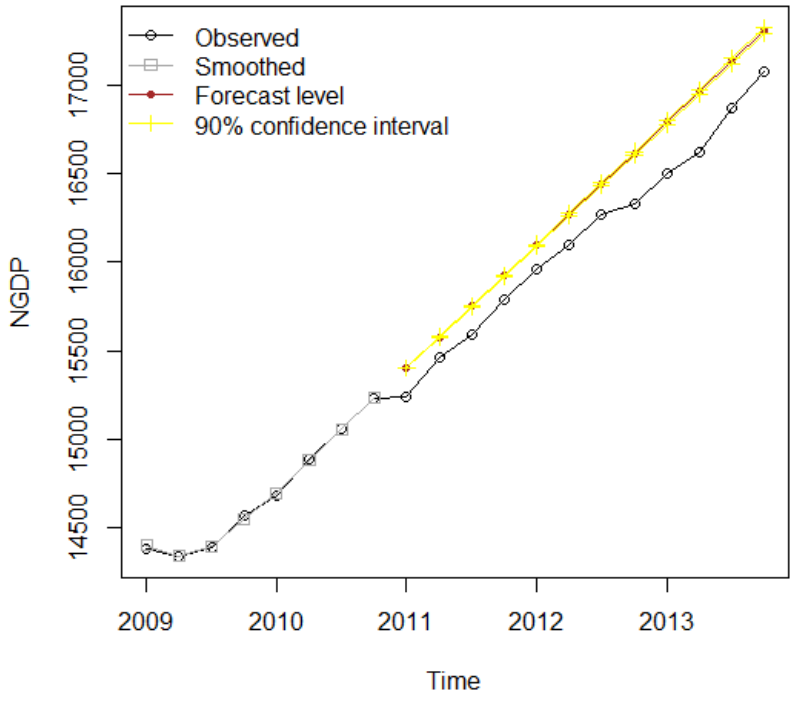

Figure 4.33. Forecast level of the nominal GDP (Gibbs sampler)

As far as the predictive capability is concerned, if we look at the graph in Figure 4.33, and compare it with the one presented in Figure 4.14, there does not appear to exist any substantial difference between the two models, for what concerns the forecast values. Both of them, in fact, provide a slightly overestimated forecast of the future trend of the GDP. The main discrepancy between the two models, in fact, resides in the width of the confidence intervals, which is remarkably greater in the case of the multivariate SUTSE model, allowing the credible region to include the actual values of the GDP. If we consider the performance measures, we can notice how all of them seem to be in favour of the model whose variance parameters have been computed by means of the Gibbs sampler, but the fact that the credible region provided by the SUTSE model contains the real GDP values should lead us to judge that model as preferable.

We will now consider the results provided by the Gibbs sampler for what concerns the Industrial Production Index.

```
IPI – GIBBS SAMPLING
AVERAGE OBSERVATIONAL AND SYSTEM VARIANCES
      V          W_μ          W_β
  0.00500    0.02298    0.56242

FORECAST PERFORMANCE MEASURES
Mean Absolute Percentage Error
2.265835
Theil Inequality Coefficient
0.01357749
```

In the Figures 4.34 and 4.35 the simulated variances, together with the ergodic means, and the simulated levels are respectively illustrated.

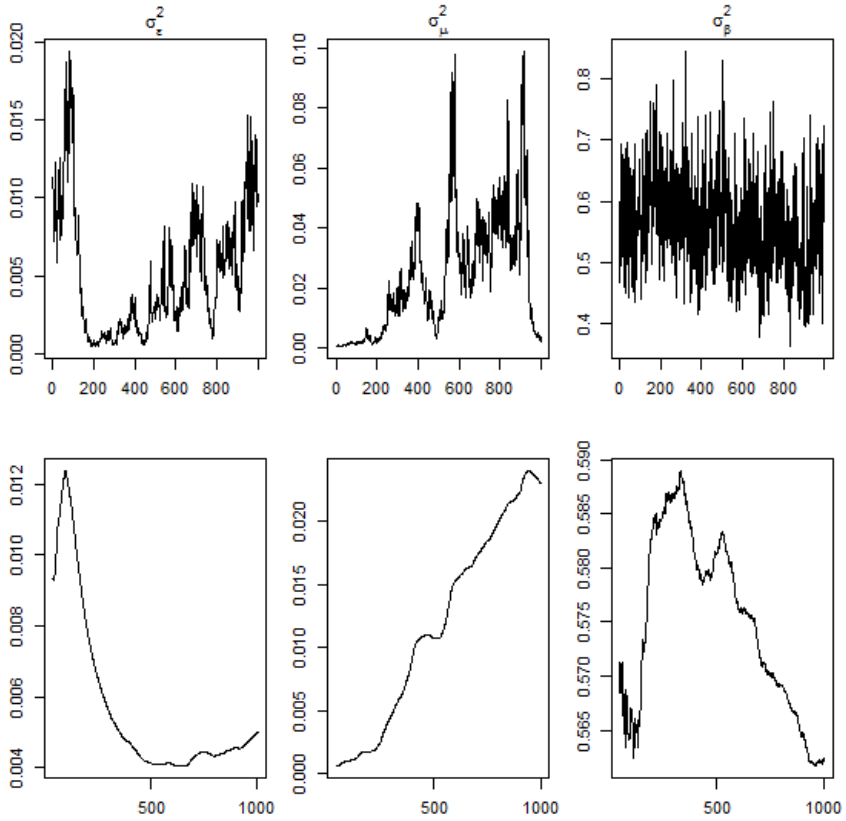

Figure 4.34. Plots of simulated variances (top) and ergodic means (bottom) for the IPI

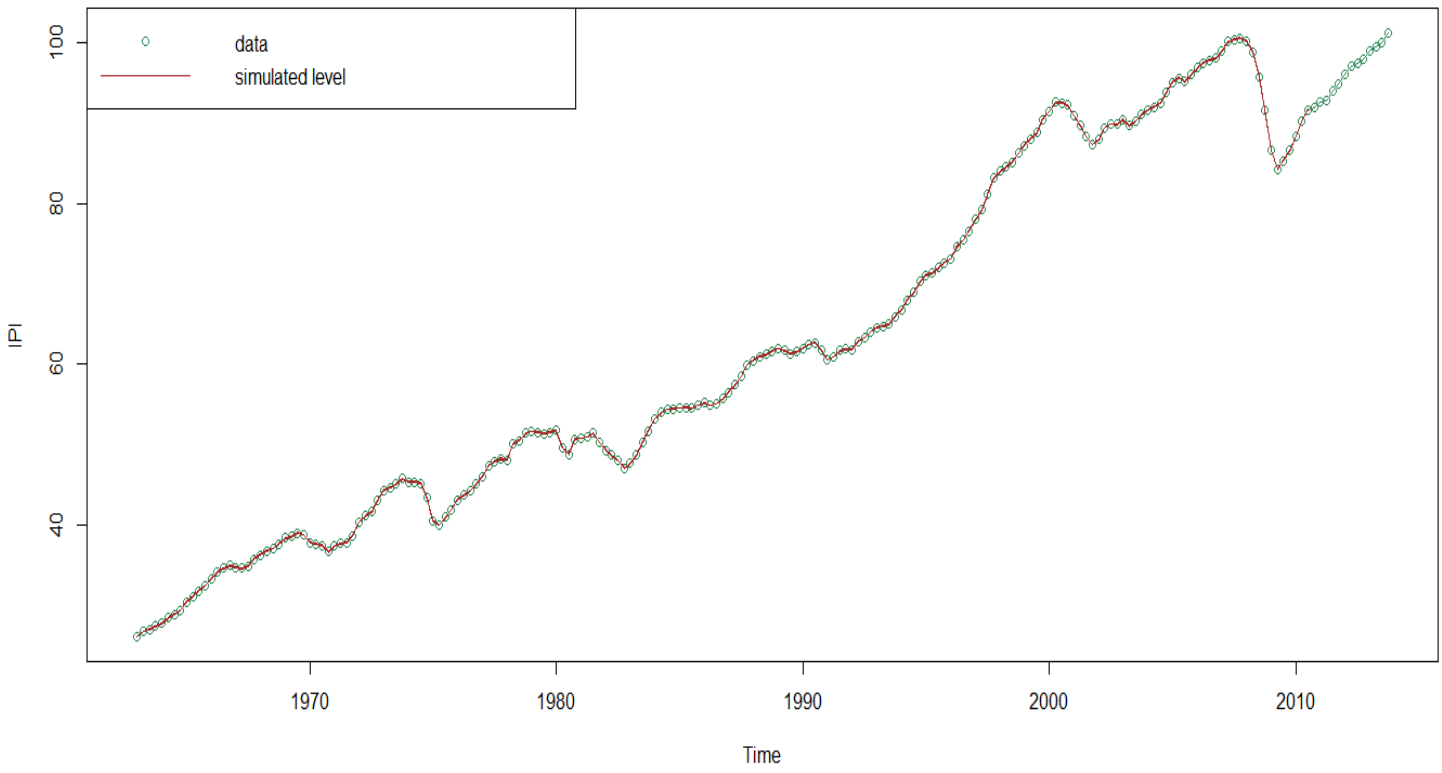

Figure 4.35. IPI levels simulated by the Gibbs sampler

Similarly to what we have observed for the GDP, even in the case of the IPI the variances estimated by the Gibbs sampler possess overall smaller values than the ones computed for the multivariate SUTSE model. Then, if we compare the simulated levels with the filtered levels illustrated in Figure 4.21, the SUTSE model and the Gibbs sampler seem to provide substantially indifferent results from the viewpoint of the data fitting.

As regards the predictive capability, instead, the graph reported in Figure 4.36 demonstrates that the Gibbs sampling procedure has led to an underestimation of the future values of the index, which did not happen in the case of the previous model (Figure 4.28). A comparison between the performance measures of the two models confirms these results.

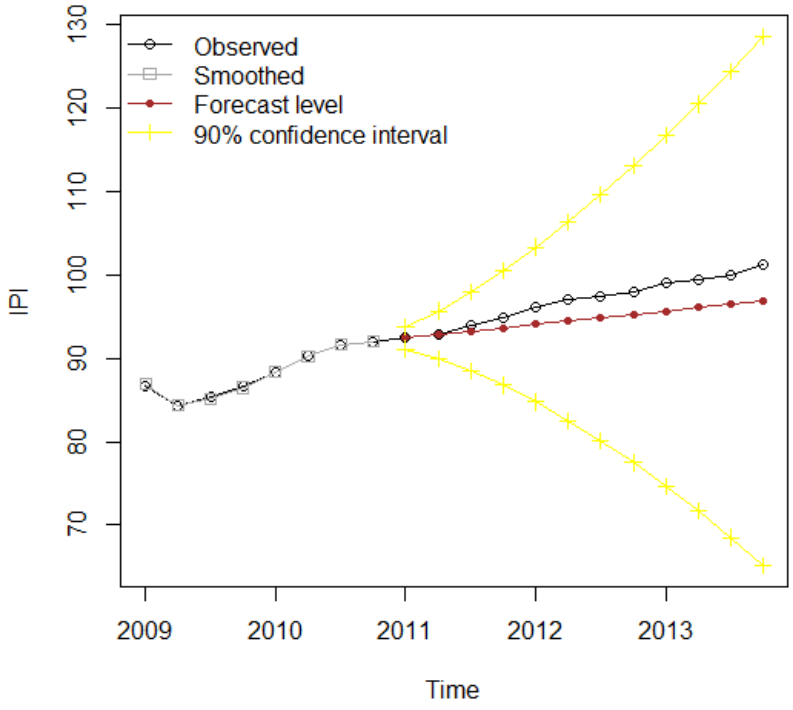

Figure 4.36. Forecast level of the IPI (Gibbs sampler)

We will now continue with the analysis of the results obtained as regards the Consumer Price Index. The code we used for the Gibbs sampling is the following.

```
CPI - GIBBS SAMPLING
AVERAGE OBSERVATIONAL AND SYSTEM VARIANCES
      V          W_μ          W_β
  0.13188    0.00392    0.23855

FORECAST PERFORMANCE MEASURES
```

```
Mean Absolute Percentage Error
1.035389
Theil Inequality Coefficient
0.005438035
```

We present in the Figures 4.37 and 4.38 the results of the simulation for what concerns the variances and the levels.

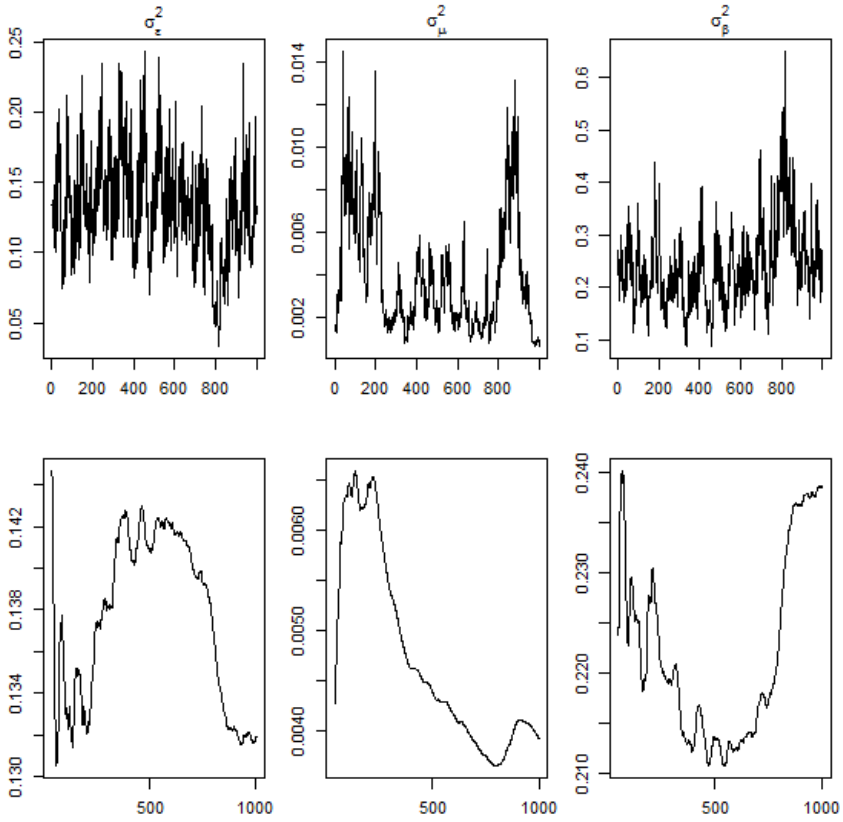

Figure 4.37. Plots of simulated variances (top) and ergodic means (bottom) for the CPI

Even for what concerns the CPI, the variances computed by the Gibbs sampler are overall smaller than the ones estimated with the maximum likelihood method for the SUTSE model. Notwithstanding this, if we compare the levels derived from the simulation algorithm with the filtered levels based on the SUTSE model (Figure 4.22), it is possible to notice how the two methods do not present important discrepancies.

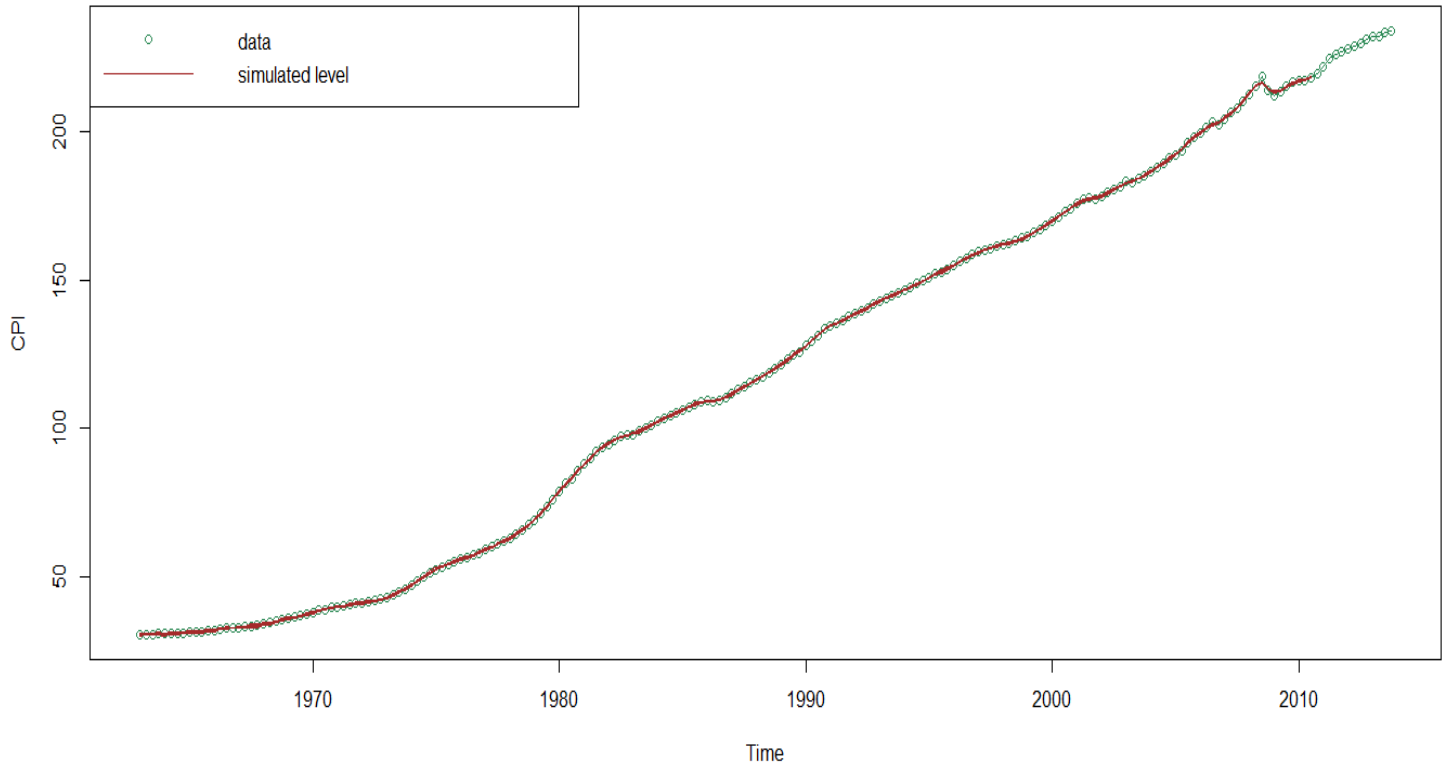

Figure 4.38. CPI levels simulated by the Gibbs sampler

In fact, if we evaluate the predictive capability, from the graph reported in Figure 4.39 it is visible how, similarly to what happened in the case of the SUTSE model (Figure 4.29), the model obtained by means of the Gibbs sampling underestimates the future trend of the CPI. Anyway, the performance measures computed in the code demonstrate that the model based on the Gibbs sampling provides a slightly more accurate forecast of this macroeconomic factor.

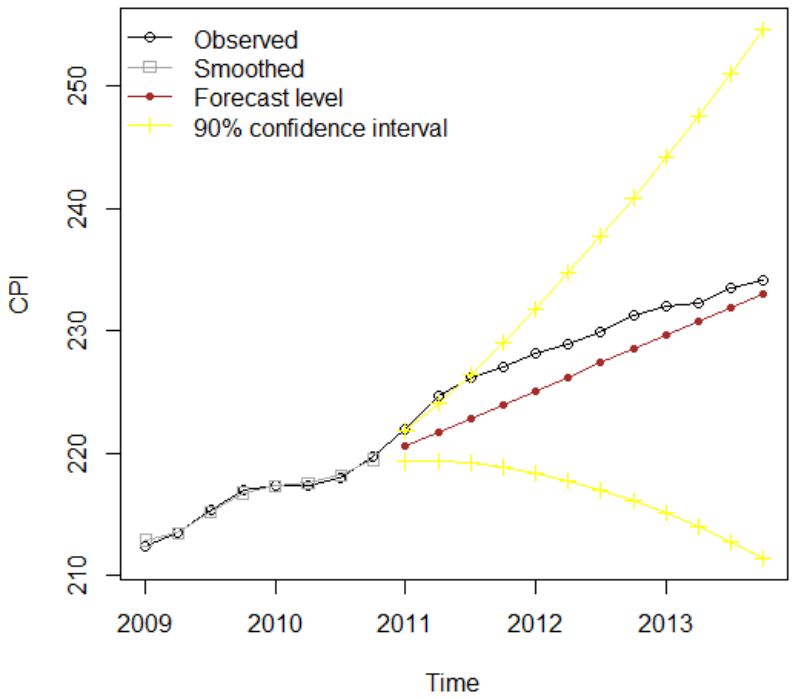

Figure 4.39. Forecast level of the CPI (Gibbs sampler)

In conclusion, we present the results obtained for what concerns the T-Bill rate.

```
TBILL - GIBBS SAMPLING
AVERAGE OBSERVATIONAL AND SYSTEM VARIANCES
```

V	W_μ	W_β	W_s
2.26e-01	7.79e-03	1.35e-01	7.62e-04

```
FORECAST PERFORMANCE MEASURES
Mean Absolute Percentage Error
624.754
Theil Inequality Coefficient
0.6511368
```

The simulated variances, accompanied by the relevant ergodic means, and the simulated levels are reported in the Figures 4.40 and 4.41.

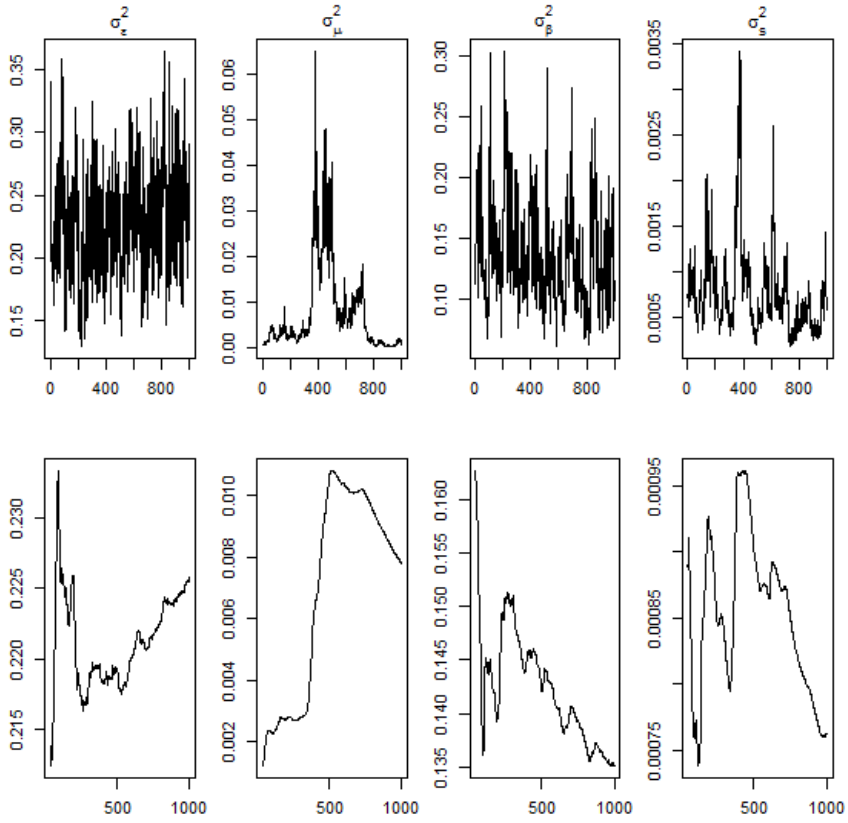

Figure 4.40. Plots of simulated variances (top) and ergodic means (bottom) for the 3-month T-Bill rate

Similarly to what we have verified about the other three variables analysed, even in the case of the T-Bill interest rate the variances estimated by means of the Gibbs sampler are lower than the ones previously obtained when estimating the SUTSE model. Anyway, the graph illustrating the levels does not remarkably differ from the one reported in Figure 4.23, representing the filtered levels based on the SUTSE model.

Finally, if we observe the graph reported in Figure 4.42, illustrating the forecast level for the last three years of the available data set, we can notice that the Gibbs-sampler-based model slightly overestimates the future trend of the interest rate. The same thing happens in the case of the multivariate SUTSE model (Figure 4.30), but, in that case, the forecast error is remarkably greater. This circumstance is confirmed by the performance measures, which are all in favour of the Gibbs-sampler-based model.

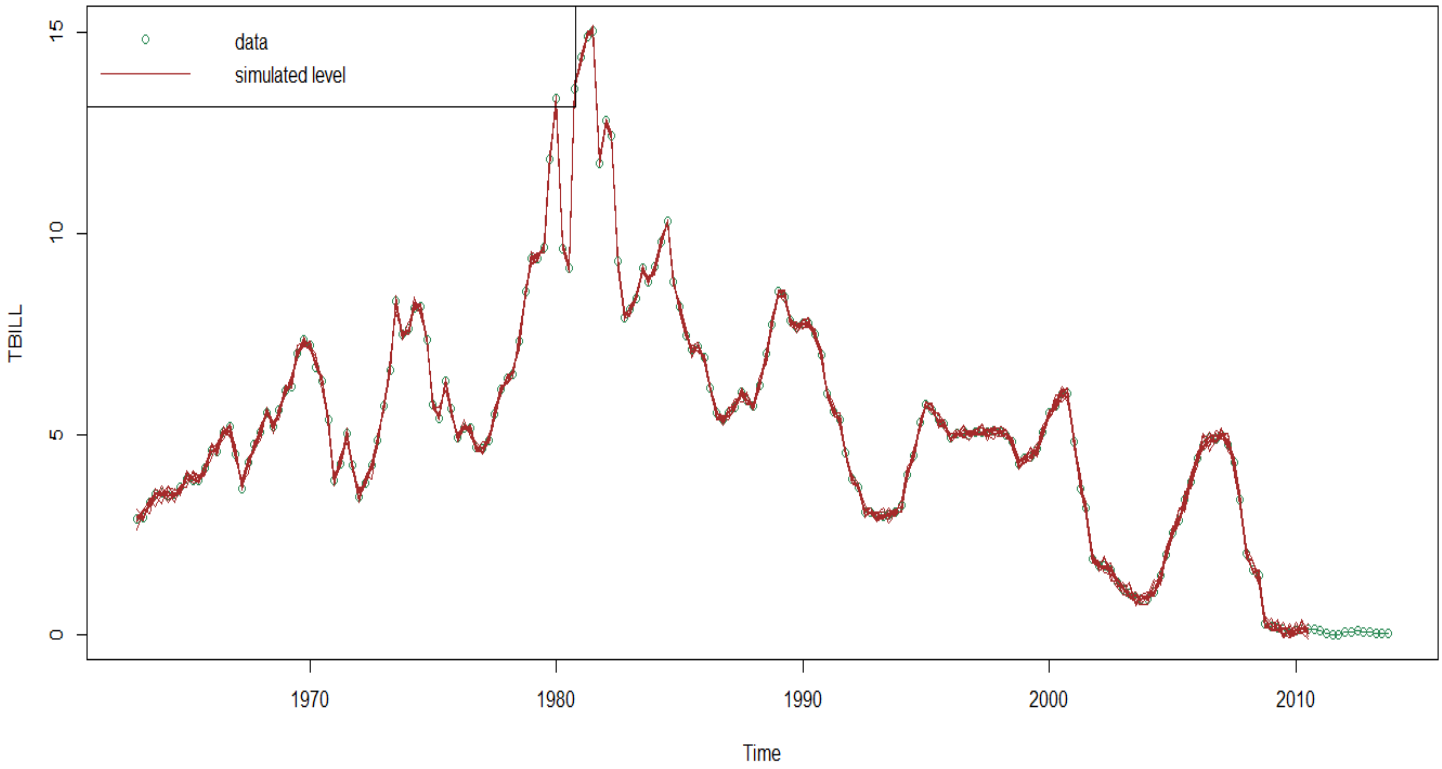

Figure 4.41. 3-month T-Bill rate with levels simulated by the Gibbs sampler

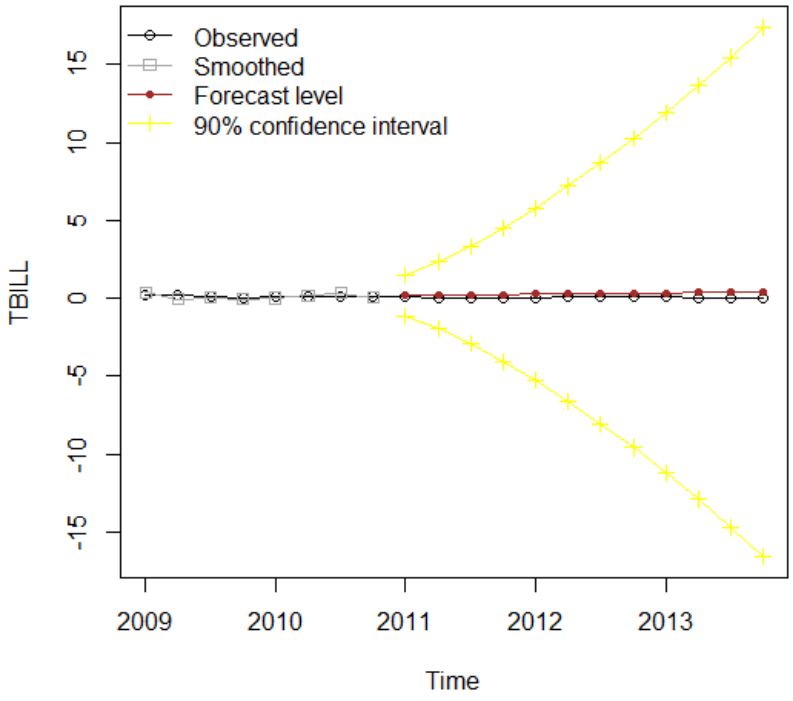

Figure 4.42. Forecast level of the 3-month T-Bill rate (Gibbs sampler)

4.18. Summary of the models proposed

IN-SAMPLE FORECAST PERFORMANCE MEASURES

SUTSE MODEL

Rmse = 57.39284

Mae = 5.73195

Theil = 0.00384

SUTSE + SEASONAL MODEL

Rmse = 51.43722

Mae = 5.66048

Theil = 0.00341

REGRESSION DLM

Rmse = 58.32701

Mae = 6.22316

Theil = 0.00392

REGRESSION DLM WITH LAGS

Rmse = 82.31051

Mae = 7.37958

Theil = 0.00553

OUT-OF-SAMPLE FORECAST PERFORMANCE MEASURES

SUTSE MODEL

Rmse = 289.00090

Mae = 16.47265

Theil = 0.00887

SUTSE + SEASONAL MODEL

Rmse = 232.18470

Mae = 14.82537

Theil = 0.00714

REGRESSION DLM

Rmse = 321.05830

Mae = 15.90850

Theil = 0.01000

REGRESSION DLM WITH LAGS

Rmse = 371.14280

Mae = 16.83304

Theil = 0.01158

MCMC SIMULATION

Rmse = 217.65080 Mae = 14.36500 Theil = 0.00669

CONCLUDING REMARKS

In this paper we have proposed four different dynamic linear models with the goal to identify the one possessing the highest predictive capability for what concerns the nominal GDP of the United States. The structure of the first two models proposed (the multivariate SUTSE models) is quite similar to the Mixed Frequency Dynamic Factor Model proposed by Barnett, Chauvet and Leiva in the paper we have taken as a reference for this thesis, while the other two models we have analysed (the multiple regression DLMs) were proposed in order to provide an example of a DLM with time-varying parameters. The results obtained have shown that the model presenting the best forecast performance is constituted by a multivariate SUTSE model including the four variables considered (nominal GDP, Industrial Production Index, Consumer Price Index and 3-month T-Bill rate) allowing for the presence of a seasonal component as regards the T-Bill rate.

Furthermore, for each one of the four variables under control, we have examined the predictive capability of a univariate DLM whose parameters have been estimated by using the Gibbs sampling simulation technique, and have verified that, for what concerns the Consumer Price Index and the T-Bill interest rate, a model of this kind is able to provide more accurate forecasts, while it returns quite unreliable results as far as the GDP and the Industrial Production Index are concerned, if compared with the results obtained for the same variables by using the multivariate SUTSE model.

A possible explanation of the inefficiencies we have encountered as regards the Gibbs-sampling-based models may be that, at present, the R package for DLM only allows to apply the Gibbs sampling algorithm to estimate the parameters of a constant and univariate DLM. This is the reason why we could not use this method with respect to the multivariate SUTSE model we had selected as the best-performing one. Consequently, among all the models proposed, we believe that the multivariate SUTSE including a seasonal component can be judged as the most useful one, as a forecasting tool, because, differently from the corresponding model for the GDP estimated by means of the Gibbs sampler, it possesses the advantage that it allows to account for the relationship

between the GDP itself and the macroeconomic factors considered. Such a relationship, in fact, has been confirmed by the forecast performance of all the multivariate models analysed, which is overall quite satisfying.

It is, anyway, strongly probable that, in the next few years, it will be possible to exploit simulation techniques for the estimation of multivariate models including time-varying parameters, which could allow to obtain even more accurate forecast results than the ones provided in this paper.

References

Barbieri, M. M. (1996). *Metodi MCMC nell'Inferenza Statistica*, Centro d'Informazione e Stampa Universitaria.

Barnett, W. A., Chauvet, M. and Leiva-Leon, D. (2014). *Real-Time Nowcasting Nominal GDP under Structural Break*.

Bawens, L., Lubrano, M. and Richard, J. F. (2004). *Bayesian Inference in Dynamic Econometric Models*, Oxford University Press.

Bernardo, J. M. (2003). *Bayesian Statistics*, R. Viertl, ed.

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (2009). *Time Series Analysis: Forecasting and Control*, Wiley Classics Library.

Box, G. E. P. and Tiao, G. C. (1992). *Bayesian Inference in Statistical Analysis*, Wiley Classics Library.

Broemeling, L. D. (1985). *Bayesian Analysis of Linear Models*, Marcel Dekker, Inc.

Commandeur, J. J. F. and Koopman, S. J. (2007). *An Introduction to State Space Time Series Analysis*, Oxford University Press.

Dey, D. K., Ghosh, S. K. and Mallick, B. K. (2000). *Generalized Linear Models: A Bayesian Perspective*, Marcel Dekker, Inc.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods*, Oxford University Press.

Franses, P. H., van Dijk, D. and Opschoor, A. (2014). *Time Series Models for Business and Economic Forecasting*, Cambridge University Press.

Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, Chapman & Hall/CRC.

Harney, H. L. (2003). *Bayesian Inference: Parameter Estimation and Decisions*, Springer-Verlag.

Harvey, A., Koopman, S. J. and Shephard, N. (2004). *State Space and Unobserved Component Models: Theory and Applications*, Cambridge University Press.

Hyndman, R. J. and Athanasopoulos, G. (2014). *Forecasting: Principles and Practice*, OTexts.com.

Iacus, S. M. and Masarotto, G. (2007). *Laboratorio di Statistica con R*, McGraw-Hill.

Kenkel, J. L. (1974). *Dynamic Linear Economic Models*, Gordon & Breach Publishing Group.

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

Petris, G., Petrone, S. and Campagnoli, P. (2007). *Dynamic Linear Models with R*, Springer.

Ruppert, D. (2010). *Statistics and Data Analysis for Financial Engineering*, Springer.

Shim, J. K. and Siegel, J. G. (2007). *Handbook of Financial Analysis, Forecasting and Modeling*, CCH Editions.

West, M. and Harrison, J. (1997) *Bayesian Forecasting and Dynamic Models*, Springer.

Yu, J., Duan, J. C., Hardle, W. and James, E. G. (2010). *Simulation-Based Estimation Methods for Financial Time Series Models: Handbook of Computational Finance*, Research Collection School of Economics.

Appendix A

```
> # ACQUISITION OF THE NUMERICAL INFORMATION
> library(tseries)
> data = read.table('database.txt', header = TRUE)
> str(data)
> data.ts = data[, -1]
> attach(data.ts)
> NGDP = ts(GDP, start = c(1963, 1), end = c(2013, 4), freq = 4)
> IPI = ts(INDPRO, start = c(1963, 1), end = c(2013, 4), freq = 4)
> CPI = ts(CPI, start = c(1963, 1), end = c(2013, 4), freq = 4)
> TBILL = ts(TB3MS, start = c(1963, 1), end = c(2013, 4), freq = 4)
> ts.plot(NGDP)
> ts.plot(IPI)
> ts.plot(CPI)
> ts.plot(TBILL)
> pairs(~ NGDP + IPI + CPI + TBILL)
>
> # LINEAR REGRESSION MODEL
> lm0 = lm(NGDP ~ IPI + CPI + TBILL)
> summary(lm0)
> lm0.stdres = rstandard(lm0)
> plot(lm0.stdres, ylab = '', main = 'Standardized Residuals')
> abline(0, 0)
> acf(lm0.stdres, main = 'ACF of Residuals')
> library(FitAR)
> LBQPlot(lm0.stdres)
>
> # DEFINITION OF THE TRAINING AND TEST SET
> data.tr = data.ts[1:(dim(data.ts)[1]-12), ]
> NGDP.tr = NGDP[1:dim(data.tr)[1]]
> IPI.tr = IPI[1:dim(data.tr)[1]]
> CPI.tr = CPI[1:dim(data.tr)[1]]
> TBILL.tr = TBILL[1:dim(data.tr)[1]]
> NGDP.test = NGDP[(dim(data.tr)[1]+1):(length(NGDP))]
```

```
> IPI.test = IPI[(dim(data.tr)[1]+1):(length(IPI))]
> CPI.test = CPI[(dim(data.tr)[1]+1):(length(CPI))]
> TBILL.test = TBILL[(dim(data.tr)[1]+1):(length(TBILL))]
>
> # DYNAMIC LINEAR MODEL
>
> # MODEL 1: SUTSE MODEL
> library(dlm)
> varEpsilon = rep(0, 4)
> varLevel = rep(0, 4)
> varSlope = rep(0, 4)
> covEpsilon = rep(0, 6)
> covLevel = rep(0, 6)
> covSlope = rep(0, 6)
> parameters = c(varEpsilon, covEpsilon, varLevel, covLevel, varSlope, covSlope)
> build = function(x) {
+ mod = dlmModPoly(2)
+ mod$FF = mod$FF %x% diag(4)
+ mod$GG = mod$GG %x% diag(4)
+ V = diag(exp(x[1:4]))
+ k = 5
+ for (i in 1:3) {
+ for (j in (i+1):4) {
+ V[i,j] = V[j,i] = x[k]
+ k = k + 1
+ }
+ }
+ mod$V = V
+ W.mu = diag(exp(x[11:14]))
+ k = 15
+ for (i in 1:3) {
+ for (j in (i+1):4) {
+ W.mu[i,j] = W.mu[j,i] = x[k]
+ k = k + 1
+ }
+ }
+ W.beta = diag(exp(x[21:24]))
+ k = 25
+ for (i in 1:3) {
+ for (j in (i+1):4) {
+ W.beta[i,j] = W.beta[j,i] = x[k]
+ k = k + 1
```

```
+ }
+ }
+ mod$W = bdiag(W.mu, W.beta)
+ mod$m0 = rep(0, 8)
+ mod$C0 = diag(8) * 1e7
+ return (mod)
+ }
> matrix.tr = matrix(c(NGDP.tr, IPI.tr, CPI.tr, TBILL.tr), nrow = dim(data.tr)
[1], ncol = dim(data.tr)[2])
> fit = dlmMLE(matrix.tr, parameters, build)
> fit$conv
[1] 0
> modSUTSE = build(fit$par)
>
> # FILTERING
> filter = dlmFilter(matrix.tr, modSUTSE)
> attach(filter)
> v = dlmSvd2var(U.C, D.C)
> v.ngdp = rep(0, length(v))
> plot(NGDP, type = 'l', col = 'seagreen')
> ngdpFilt = ts(dropFirst(m[, 1]), start = c(1963, 1), freq = 4)
> lines(ngdpFilt, col = 'brown')
> for (i in 1:length(v)) {
+ v.ngdp[i] = v[[i]][1, 1]
+ }
> lwr.ngdp = ngdpFilt + qnorm(0.05, sd = sqrt(v.ngdp[-1]))
> upr.ngdp = ngdpFilt + qnorm(0.95, sd = sqrt(v.ngdp[-1]))
> lines(lwr.ngdp, lty = 2, col = 'brown')
> lines(upr.ngdp, lty = 2, col = 'brown')
> legend('topleft', legend = c('data', 'filtered level', '90% confidence
interval'), col = c('green', 'brown', 'brown'), lty = c(1, 1, 2))
> detach(filter)
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (IN-SAMPLE DATA)
> sdevSUTSE = residuals(filter)$sd[, 1]
> ngdpFore = ts(filter$f[, 1], start = c(1963, 1), freq = 4)
> lwrSUTSE = ngdpFore + qnorm(0.05) * sdevSUTSE
> uprSUTSE = ngdpFore + qnorm(0.95) * sdevSUTSE
> plot(NGDP, type = 'l', col = 'green')
> lines(ngdpFore, type = 'l', col = 'blue')
> lines(lwrSUTSE, lty = 2, col = 'blue')
> lines(uprSUTSE, lty = 2, col = 'blue')
```

```
> legend('topleft', legend = c('data', 'one-step-ahead forecast', '90%
prediction interval'), col = c('green', 'blue', 'blue'), lty = c(1, 1, 2))
> rmse = sqrt(mean((ngdpFore[-(1:2)] - NGDP.tr[-(1:2)])^2))
> mae = sqrt(mean(abs(ngdpFore[-(1:2)] - NGDP.tr[-(1:2)])))
> tic = sqrt(mean((ngdpFore[-(1:2)] - NGDP.tr[-(1:2)])^2))/(sqrt(mean(ngdpFore[-
(1:2)]^2)) + sqrt(mean(NGDP.tr[-(1:2)]^2)))
> tsdiag(filter)
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (OUT-OF-SAMPLE DATA)
> forecast = dlmForecast(filter, nAhead = 12)
> attach(forecast)
> sqrtRngdp = sapply(R, function(x) sqrt(x[1, 1]))
> ngdpFore = ts(a[, 1], start = c(2011, 1), freq = 4)
> lwr.ngdp = ngdpFore + qnorm(0.05, sd = sqrtRngdp)
> upr.ngdp = ngdpFore + qnorm(0.95, sd = sqrtRngdp)
> x.ngdp = ts.union(window(NGDP, start = c(2009, 1)),
+ ngdpFore, lwr.ngdp, upr.ngdp)
> plot(x.ngdp, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'), ylab = 'NGDP')
> legend('topleft', legend = c('observed', 'forecast level', '90% confidence
interval'),
+ bty = 'n', pch = c(1, 20, 3, 3), lty = 1,
+ col = c('black', 'brown', 'yellow', 'yellow'))
> detach(forecast)
> rmse.ngdp = sqrt(mean((ngdpFore - NGDP.test)^2))
> mae.ngdp = sqrt(mean(abs(ngdpFore - NGDP.test)))
> tic.ngdp = sqrt(mean((ngdpFore - NGDP.test)^2))/(sqrt(mean(ngdpFore^2)) +
sqrt(mean(NGDP.test^2)))
>
> # SUTSE WITH SEASONAL COMPONENT ON TBILL
> varSeasTbill = 0
> paramSeas = c(varEpsilon, covEpsilon, varLevel, covLevel, varSlope, covSlope,
varSeasTbill)
> buildSeas = function(x) {
+ mod = dlmModPoly(2) + dlmModSeas(4)
+ mod$FF = mod$FF %x% diag(4)
+ mod$GG = mod$GG %x% diag(4)
+ V = diag(exp(x[1:4]))
+ k = 5
+ for (i in 1:3) {
+ for (j in (i+1):4) {
+ V[i,j] = V[j,i] = x[k]
```

```
+ k = k + 1
+ }
+ }
+ mod$V = V
+ W.mu = diag(exp(x[11:14]))
+ k = 15
+ for (i in 1:3) {
+ for (j in (i+1):4) {
+ W.mu[i,j] = W.mu[j,i] = x[k]
+ k = k + 1
+ }
+ }
+ W.beta = diag(exp(x[21:24]))
+ k = 25
+ for (i in 1:3) {
+ for (j in (i+1):4) {
+ W.beta[i,j] = W.beta[j,i] = x[k]
+ k = k + 1
+ }
+ }
+ W.s1 = diag(c(0, 0, 0, exp(x[31])))
+ W.s2 = diag(rep(0, 4))
+ W.s3 = diag(rep(0, 4))
+ mod$W = bdiag(W.mu, W.beta, W.s1, W.s2, W.s3)
+ mod$m0 = rep(0, 20)
+ mod$C0 = diag(20) * 1e7
+ return (mod)
+ }
> fitSeas = dlmMLE(matrix.tr, paramSeas, buildSeas)
> fitSeas$conv
[1] 0
> modSUTSEseas = buildSeas(fitSeas$par)
>
> # FILTERING
> filterSeas = dlmFilter(matrix.tr, modSUTSEseas)
> attach(filterSeas)
> vSeas = dlmSvd2var(U.C, D.C)
> vSeas.ngdp = rep(0, length(vSeas))
> plot(NGDP, type = 'l', col = 'seagreen')
> ngdpFiltSeas = ts(dropFirst(m[, 1]), start = c(1963, 1), freq = 4)
> lines(ngdpFiltSeas, col = 'brown')
> for (i in 1:length(vSeas)) {
```

```
+ vSeas.ngdp[i] = vSeas[[i]][1, 1]
+ }
> lwrSeas.ngdp = ngdpFiltSeas + qnorm(0.05, sd = sqrt(vSeas.ngdp[-1]))
> uprSeas.ngdp = ngdpFiltSeas + qnorm(0.95, sd = sqrt(vSeas.ngdp[-1]))
> lines(lwrSeas.ngdp, lty = 2, col = 'brown')
> lines(uprSeas.ngdp, lty = 2, col = 'brown')
> legend('topleft', legend = c('data', 'filtered level', '90% confidence
interval'), col = c('green', 'brown', 'brown'), lty = c(1, 1, 2))
> detach(filterSeas)
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (IN-SAMPLE DATA)
> sdevSeas = residuals(filterSeas)$sd[, 1]
> ngdpForeSeas = ts(filterSeas$f[, 1], start = c(1963, 1), freq = 4)
> lwrSeas = ngdpForeSeas + qnorm(0.05) * sdevSeas
> uprSeas = ngdpForeSeas + qnorm(0.95) * sdevSeas
> plot(NGDP, type = 'l', col = 'green')
> lines(ngdpForeSeas, type = 'l', col = 'blue')
> lines(lwrSeas, lty = 2, col = 'blue')
> lines(uprSeas, lty = 2, col = 'blue')
> legend('topleft', legend = c('data', 'one-step-ahead forecast', '90%
prediction interval'), col = c('green', 'blue', 'blue'), lty = c(1, 1, 2))
> rmseSeas = sqrt(mean((ngdpForeSeas[-(1:5)] - NGDP.tr[-(1:5)])^2))
> maeSeas = sqrt(mean(abs(ngdpForeSeas[-(1:5)] - NGDP.tr[-(1:5)])))
> ticSeas = sqrt(mean((ngdpForeSeas[-(1:5)] - NGDP.tr[-(1:5)])^2))/
(sqrt(mean(ngdpForeSeas[-(1:5)]^2)) + sqrt(mean(NGDP.tr[-(1:5)]^2)))
> tsdiag(filterSeas)
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (OUT-OF-SAMPLE DATA)
> forecastSeas = dlmForecast(filterSeas, nAhead = 12)
> attach(forecastSeas)
> sqrtRngdpSeas = sapply(R, function(x) sqrt(x[1, 1]))
> ngdpForeSeas = ts(a[, 1], start = c(2011, 1), freq = 4)
> lwrSeas.ngdp = ngdpForeSeas + qnorm(0.05, sd = sqrtRngdpSeas)
> uprSeas.ngdp = ngdpForeSeas + qnorm(0.95, sd = sqrtRngdpSeas)
> xSeas.ngdp = ts.union(window(NGDP, start = c(2009, 1)),
+ ngdpForeSeas, lwrSeas.ngdp, uprSeas.ngdp)
> plot(xSeas.ngdp, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'), ylab = 'NGDP')
> legend('topleft', legend = c('observed', 'forecast level', '90% confidence
interval'),
+ bty = 'n', pch = c(1, 20, 3, 3), lty = 1,
+ col = c('black', 'brown', 'yellow', 'yellow'))
```

```
> detach(forecastSeas)
> rmseSeas.ngdp = sqrt(mean((ngdpForeSeas - NGDP.test)^2))
> maeSeas.ngdp = sqrt(mean(abs(ngdpForeSeas - NGDP.test)))
> ticSeas.ngdp = sqrt(mean((ngdpForeSeas - NGDP.test)^2))/
(sqrt(mean(ngdpForeSeas^2)) + sqrt(mean(NGDP.test^2)))
>
> # MULTIPLE REGRESSION DLM
> X.tr = data.tr[, -1]
> varObs = 0
> varInt = 0
> varSlope = rep(0, 3)
> covSlope = rep(0, 3)
> paramReg = c(varObs, varInt, varSlope, covSlope)
> buildReg = function(x) {
+ mod = dlmModReg(X.tr)
+ V = exp(x[1])
+ W.int = exp(x[2])
+ W.beta = diag(exp(x[3:5]))
+ k = 6
+ for (i in 1:2) {
+ for (j in (i+1):3) {
+ W.beta[i,j] = W.beta[j,i] = x[k]
+ k = k + 1
+ }
+ }
+ mod$V = V
+ mod$W = bdiag(W.int, W.beta)
+ mod$m0 = rep(0, 4)
+ mod$C0 = diag(4) * 1e7
+ return(mod)
+ }
> fitReg = dlmMLE(NGDP.tr, paramReg, buildReg)
> fitReg$conv
[1] 0
> modReg = buildReg(fitReg$par)
>
> # FILTERING
> modRegFilt = dlmFilter(NGDP.tr, modReg)
> sdev = residuals(modRegFilt)$sd
> modReg1Ahead = ts(modRegFilt$f, start = c(1963, 1), freq = 4)
> lwr = modReg1Ahead + qnorm(0.05) * sdev
> upr = modReg1Ahead + qnorm(0.95) * sdev
```

```
> plot(NGDP, type = 'l', col = 'green')
> lines(modReg1Ahead, type = 'l', pch = 20, col = 'blue')
> lines(lwr, lty = 2, col = 'blue')
> lines(upr, lty = 2, col = 'blue')
> legend('topleft', legend = c('data', 'one-step-ahead forecast', '90%
prediction interval'), col = c('green', 'blue', 'blue'), lty = c(1, 1, 2))
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (IN-SAMPLE DATA)
> rmseReg = sqrt(mean((modReg1Ahead[-1] - NGDP.tr[-1])^2))
> maeReg = sqrt(mean(abs(modReg1Ahead[-1] - NGDP.tr[-1])))
> ticReg = sqrt(mean((modReg1Ahead[-1] - NGDP.tr[-1])^2))/
(sqrt(mean(modReg1Ahead[-1]^2)) + sqrt(mean(NGDP.tr[-1]^2)))
> tsdiag(modRegFilt)
>
> # MULTIPLE REGRESSION DLM WITH LAGGED REGRESSORS
> IPI.L = c(0, IPI[-length(IPI)])
> CPI.L = c(0, CPI[-length(CPI)])
> TBILL.L = c(0, TBILL[-length(TBILL)])
> IPI.L.tr = IPI.L[1:length(IPI.tr)]
> CPI.L.tr = CPI.L[1:length(CPI.tr)]
> TBILL.L.tr = TBILL.L[1:length(TBILL.tr)]
> XL.tr = matrix(c(IPI.L.tr, CPI.L.tr, TBILL.L.tr), nrow = dim(data.tr)[1], ncol
= 3)
> buildRegL = function(x) {
+ mod = dlmModReg(XL.tr)
+ V = exp(x[1])
+ W.int = exp(x[2])
+ W.beta = diag(exp(x[3:5]))
+ k = 6
+ for (i in 1:2) {
+ for (j in (i+1):3) {
+ W.beta[i,j] = W.beta[j,i] = x[k]
+ k = k + 1
+ }
+ }
+ mod$V = V
+ mod$W = bdiag(W.int, W.beta)
+ mod$m0 = rep(0, 4)
+ mod$C0 = diag(4) * 1e7
+ return(mod)
+ }
> fitRegL = dlmMLE(NGDP.tr, paramReg, buildRegL)
```

```
> fitRegL$conv
[1] 0
> modRegL = buildRegL(fitReg$par)
>
> # FILTERING
> modRegFiltL = dlmFilter(NGDP.tr, modRegL)
> sdevL = residuals(modRegFiltL)$sd
> modReg1AheadL = ts(modRegFiltL$f, start = c(1963, 1), freq = 4)
> lwrL = modReg1AheadL + qnorm(0.05) * sdevL
> uprL = modReg1AheadL + qnorm(0.95) * sdevL
> plot(NGDP, type = 'l', col = 'green')
> lines(modReg1AheadL, type = 'l', pch = 20, col = 'blue')
> lines(lwrL, lty = 2, col = 'blue')
> lines(uprL, lty = 2, col = 'blue')
> legend('topleft', legend = c('data', 'one-step-ahead forecast', '90%
prediction interval'), col = c('green', 'blue', 'blue'), lty = c(1, 1, 2))
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (IN-SAMPLE DATA)
> rmseRegL = sqrt(mean((modReg1AheadL[-1] - NGDP.tr[-1])^2))
> maeRegL = sqrt(mean(abs(modReg1AheadL[-1] - NGDP.tr[-1])))
> ticRegL = sqrt(mean((modReg1AheadL[-1] - NGDP.tr[-1])^2))/
(sqrt(mean(modReg1AheadL[-1]^2)) + sqrt(mean(NGDP.tr[-1]^2)))
> ticRegL
> tsdiag(modRegFiltL)
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (OUT-OF-SAMPLE DATA)
> XL.whole = matrix(c(IPI.L, CPI.L, TBILL.L), nrow = dim(data.ts)[1], ncol = 3)
> NGDP.whole = c(NGDP.tr, rep(NA, 12))
> dlmRegL.test = dlmModReg(XL.whole, dV = modReg$V, dW = diag(modReg$W))
> dlmForeL = dlmFilter(NGDP.whole, dlmRegL.test)
> ngdpForeRegL = dlmForeL$f[-(1:length(NGDP.tr))]
> ngdpForeRegL = ts(ngdpForeRegL, start = c(2011, 1), freq = 4)
> plot(window(NGDP, start = c(2009, 1)), type = 'o', col = 'seagreen', ylab =
'NGDP')
> lines(ngdpForeRegL, type = 'o', pch = 20, col = 'brown')
> legend('topleft', legend = c('Observed', 'Forecast'), col = c('seagreen',
'brown'), pch = c(1, 20))
> rmseRegL.ngdp = sqrt(mean((ngdpForeRegL - NGDP.test)^2))
> maeRegL.ngdp = sqrt(mean(abs(ngdpForeRegL - NGDP.test)))
> ticRegL.ngdp = sqrt(mean((ngdpForeRegL - NGDP.test)^2))/
(sqrt(mean(ngdpForeRegL^2)) + sqrt(mean(NGDP.test^2)))
>
```

```
> # We select second model (SUTSE with seasonal component on T-Bill)
>
> # FILTERING
> attach(filterSeas)
>
> # IPI
> v.ipi = rep(0, length(v))
> plot(IPI, type = 'l', col = 'seagreen')
> ipiFilt = ts(dropFirst(m[, 2]), start = c(1963, 1), freq = 4)
> lines(ipiFilt, col = 'brown')
> for (i in 1:length(v)) {
+ v.ipi[i] = v[[i]][2, 2]
+ }
> lwr.ipi = ipiFilt + qnorm(0.05, sd = sqrt(v.ipi[-1]))
> upr.ipi = ipiFilt + qnorm(0.95, sd = sqrt(v.ipi[-1]))
> lines(lwr.ipi, lty = 2, col = 'brown')
> lines(upr.ipi, lty = 2, col = 'brown')
> legend('topleft', legend = c('data', 'filtered level', '90% confidence
interval'), col = c('seagreen', 'brown', 'brown'), lty = c(1, 1, 2))
>
> # CPI
> v.cpi = rep(0, length(v))
> plot(CPI, type = 'l', col = 'seagreen')
> cpiFilt = ts(dropFirst(m[, 3]), start = c(1963, 1), freq = 4)
> lines(cpiFilt, col = 'brown')
> for (i in 1:length(v)) {
+ v.cpi[i] = v[[i]][3, 3]
+ }
> lwr.cpi = cpiFilt + qnorm(0.05, sd = sqrt(v.cpi[-1]))
> upr.cpi = cpiFilt + qnorm(0.95, sd = sqrt(v.cpi[-1]))
> lines(lwr.cpi, lty = 2, col = 'brown')
> lines(upr.cpi, lty = 2, col = 'brown')
> legend('topleft', legend = c('data', 'filtered level', '90% confidence
interval'), col = c('seagreen', 'brown', 'brown'), lty = c(1, 1, 2))
>
> # TBILL
> v.tbill = rep(0, length(v))
> plot(TBILL, type = 'l', col = 'seagreen')
> tbillFilt = ts(dropFirst(m[, 4]), start = c(1963, 1), freq = 4)
> lines(tbillFilt, col = 'brown')
> for (i in 1:length(v)) {
+ v.tbill[i] = v[[i]][4, 4]
```

```
+ }
> lwr.tbill = tbillFilt + qnorm(0.05, sd = sqrt(v.tbill[-1]))
> upr.tbill = tbillFilt + qnorm(0.95, sd = sqrt(v.tbill[-1]))
> lines(lwr.tbill, lty = 2, col = 'brown')
> lines(upr.tbill, lty = 2, col = 'brown')
> legend('topleft', legend = c('data', 'filtered level', '90% confidence
interval'), col = c('seagreen', 'brown', 'brown'), lty = c(1, 1, 2))
>
> detach(filterSeas)
>
> # SMOOTHING
> smooth = dlmSmooth(filterSeas)
> attach(smooth)
> v = dlmSvd2var(U.S, D.S)
>
> # NGDP
> v.ngdp = rep(0, length(v))
> plot(NGDP, type = 'l', col = 'seagreen')
> ngdpSmooth = ts(dropFirst(s[, 1]), start = c(1963, 1), freq = 4)
> lines(ngdpSmooth, col = 'brown')
> for (i in 1:length(v)) {
+ v.ngdp[i] = v[[i]][1, 1]
+ }
> lwr.ngdp = ngdpSmooth + qnorm(0.05, sd = sqrt(v.ngdp[-1]))
> upr.ngdp = ngdpSmooth + qnorm(0.95, sd = sqrt(v.ngdp[-1]))
> lines(lwr.ngdp, lty = 2, col = 'brown')
> lines(upr.ngdp, lty = 2, col = 'brown')
> legend('topleft',
+ legend = c('data', 'smoothed level', '90% confidence interval'),
+ col = c('seagreen', 'brown', 'brown'),
+ pch = c(1, 20, NA),
+ lty = c(NA, NA, 2))
>
> # IPI
> v.ipi = rep(0, length(v))
> plot(IPI, type = 'l', col = 'seagreen')
> ipiSmooth = ts(dropFirst(s[, 2]), start = c(1963, 1), freq = 4)
> lines(ipiSmooth, col = 'brown')
> for (i in 1:length(v)) {
+ v.ipi[i] = v[[i]][2, 2]
+ }
> lwr.ipi = ipiSmooth + qnorm(0.05, sd = sqrt(v.ipi[-1]))
```

```
> upr.ipi = ipiSmooth + qnorm(0.95, sd = sqrt(v.ipi[-1]))
> lines(lwr.ipi, lty = 2, col = 'brown')
> lines(upr.ipi, lty = 2, col = 'brown')
> legend('topleft',
+ legend = c('data', 'smoothed level', '90% confidence interval'),
+ col = c('seagreen', 'brown', 'brown'),
+ pch = c(1, 20, NA),
+ lty = c(NA, NA, 2))
>
> # CPI
> v.cpi = rep(0, length(v))
> plot(CPI, type = 'l', col = 'seagreen')
> cpiSmooth = ts(dropFirst(s[, 3]), start = c(1963, 1), freq = 4)
> lines(cpiSmooth, col = 'brown')
> for (i in 1:length(v)) {
+ v.cpi[i] = v[[i]][3, 3]
+ }
> lwr.cpi = cpiSmooth + qnorm(0.05, sd = sqrt(v.cpi[-1]))
> upr.cpi = cpiSmooth + qnorm(0.95, sd = sqrt(v.cpi[-1]))
> lines(lwr.cpi, lty = 2, col = 'brown')
> lines(upr.cpi, lty = 2, col = 'brown')
> legend('topleft',
+ legend = c('data', 'smoothed level', '90% confidence interval'),
+ col = c('seagreen', 'brown', 'brown'),
+ pch = c(1, 20, NA),
+ lty = c(NA, NA, 2))
>
> # TBILL
> v.tbill = rep(0, length(v))
> plot(TBILL, type = 'l', col = 'seagreen')
> tbillSmooth = ts(dropFirst(s[, 4]), start = c(1963, 1), freq = 4)
> lines(tbillSmooth, col = 'brown')
> for (i in 1:length(v)) {
+ v.tbill[i] = v[[i]][4, 4]
+ }
> lwr.tbill = tbillSmooth + qnorm(0.05, sd = sqrt(v.tbill[-1]))
> upr.tbill = tbillSmooth + qnorm(0.95, sd = sqrt(v.tbill[-1]))
> lines(lwr.tbill, lty = 2, col = 'brown')
> lines(upr.tbill, lty = 2, col = 'brown')
> legend('topleft',
+ legend = c('data', 'smoothed level', '90% confidence interval'),
+ col = c('seagreen', 'brown', 'brown'),
```

```
+ pch = c(1, 20, NA),
+ lty = c(NA, NA, 2))
>
> detach(smooth)
>
> # FORECASTING
> forecast = dlmForecast(filterSeas, nAhead = 12)
> attach(forecast)
>
> # IPI
> sqrtRipi = sapply(R, function(x) sqrt(x[2, 2]))
> ipiFore = ts(a[, 2], start = c(2011, 1), freq = 4)
> lwr.ipi = ipiFore + qnorm(0.05, sd = sqrtRipi)
> upr.ipi = ipiFore + qnorm(0.95, sd = sqrtRipi)
> x.ipi = ts.union(window(IPI, start = c(2009, 1)),
+ ipiFore, lwr.ipi, upr.ipi)
> plot(x.ipi, plot.type = 'single', type = 'o', pch = c(1, 0, 20, 3, 3),
+ col = c('black', 'darkgrey', 'brown', 'yellow', 'yellow'), ylab = 'IPI')
> legend('topleft', legend = c('observed', 'smoothed', 'forecast level', '90%
confidence interval'),
+ bty = 'n', pch = c(1, 0, 20, 3, 3), lty = 1,
+ col = c('black', 'darkgrey', 'brown', 'yellow', 'yellow'))
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (OUT-OF-SAMPLE DATA)
> mape.ipi = 100 * mean(abs((ipiFore - IPI.test)/IPI.test))
> tic.ipi = sqrt(mean((ipiFore - IPI.test)^2))/(sqrt(mean(ipiFore^2)) +
sqrt(mean(IPI.test^2)))
>
> # CPI
> sqrtRcpi = sapply(R, function(x) sqrt(x[3, 3]))
> cpiFore = ts(a[, 3], start = c(2011, 1), freq = 4)
> lwr.cpi = cpiFore + qnorm(0.05, sd = sqrtRcpi)
> upr.cpi = cpiFore + qnorm(0.95, sd = sqrtRcpi)
> x.cpi = ts.union(window(CPI, start = c(2009, 1)),
+ cpiFore, lwr.cpi, upr.cpi)
> plot(x.cpi, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'), ylab = 'CPI')
> legend('topleft', legend = c('observed', 'forecast level', '90% confidence
interval'),
+ bty = 'n', pch = c(1, 20, 3, 3), lty = 1,
+ col = c('black', 'brown', 'yellow', 'yellow'))
>
```

```
> # EVALUATION OF THE PREDICTIVE CAPABILITY (OUT-OF-SAMPLE DATA)
> mape.cpi = 100 * mean(abs((cpiFore - CPI.test)/CPI.test))
> tic.cpi = sqrt(mean((cpiFore - CPI.test)^2))/(sqrt(mean(cpiFore^2)) +
sqrt(mean(CPI.test^2)))
>
> # TBILL
> sqrtRtbill = sapply(R, function(x) sqrt(x[4, 4]))
> tbillFore = ts(a[, 4], start = c(2011, 1), freq = 4)
> lwr.tbill = tbillFore + qnorm(0.05, sd = sqrtRtbill)
> upr.tbill = tbillFore + qnorm(0.95, sd = sqrtRtbill)
> x.tbill = ts.union(window(TBILL, start = c(2009, 1)),
+ tbillFore, lwr.tbill, upr.tbill)
> plot(x.tbill, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'), ylab = 'TBILL')
> legend('topleft', legend = c('observed', 'forecast level', '90% confidence
interval'),
+ bty = 'n', pch = c(1, 20, 3, 3), lty = 1,
+ col = c('black', 'brown', 'yellow', 'yellow'))
>
> # EVALUATION OF THE PREDICTIVE CAPABILITY (OUT-OF-SAMPLE DATA)
> mape.tbill = 100 * mean(abs((tbillFore - TBILL.test)/TBILL.test))
> tic.tbill = sqrt(mean((tbillFore - TBILL.test)^2))/(sqrt(mean(tbillFore^2)) +
sqrt(mean(TBILL.test^2)))
>
> detach(forecast)
>
> # MCMC SIMULATION
>
> # NGDP - GIBBS SAMPLING
> ngdpGibbs = dlmGibbsDIG(NGDP.tr, dlmModPoly(2), a.y = 1, b.y = 1000,
+ a.theta = 1, b.theta = 1000, n.sample = 1100, ind = c(1, 2))
> burn = 100
> attach(ngdpGibbs)
> dV = dV[-(1:burn)]
> dW = dW[-(1:burn), ]
> detach()
> par(mfrow = c(2, 3), mar = c(3.1, 2.1, 2.1, 1.1))
> plot(dV, type = 'l', xlab = "", ylab = "", main =
expression(sigma[epsilon]^2))
> plot(dW[, 1], type = 'l', xlab = "", ylab = "", main =
expression(sigma[mu]^2))
```

```
> plot(dW[, 2], type = 'l', xlab = "", ylab = "", main =
expression(sigma[beta]^2))
> use = length(dV) - burn
> from = 0.05 * use
> at = pretty(c(0, use), n = 3)
> at = at[at >= from]
> plot(ergMean(dV, from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 1], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 2], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> mcmcMean(cbind(dV, dW))
> graphics.off()
> plot(NGDP, type = 'o', col = 'seagreen')
> ngdpMCMC = ts(dropFirst(ngdpGibbs$theta[1:length(NGDP.tr), 1, 1091:1100]),
start = c(1963, 1), freq = 4)
> for (i in 1:10)
+ lines(ngdpMCMC[, i], col = 'brown')
> legend('topleft', legend = c('data', 'simulated level'), col = c('seagreen',
'brown'), pch = c(1, NA), lty = c(NA, 1))
>
> # FORECASTING
> parmMCMC = mcmcMean(cbind(dV, dW))[c(1, 3, 5)]
> ngdpModMCMC = dlmModPoly(2, dV = parmMCMC[1], dW = parmMCMC[c(2:3)])
> ngdpFiltMCMC = dlmFilter(NGDP.tr, mod = ngdpModMCMC)
> forecastMCMC = dlmForecast(ngdpFiltMCMC, nAhead = 12)
> sqrtR = sapply(forecastMCMC$R, function(x) sqrt(x[1, 1]))
> ngdpForeMCMC = ts(forecastMCMC$a[, 1], start = c(2011, 1), freq = 4)
> pl = ngdpForeMCMC + qnorm(0.05, sd = sqrtR)
> pu = ngdpForeMCMC + qnorm(0.95, sd = sqrtR)
> x = ts.union(window(NGDP, start = c(2009, 1)),
+ ngdpForeMCMC, pl, pu)
> plot(x, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'),
+ ylab = 'NGDP')
> legend('topleft', legend = c('Observed', 'Forecast level', '90% confidence
interval'), bty = 'n', pch = c(1, 20, 3, 3), lty = 1, col = c('black', 'brown',
'yellow', 'yellow'))
>
> # FORECAST PERFORMANCE MEASURES
> rmse.ngdpMCMC = sqrt(mean((ngdpForeMCMC - NGDP.test)^2))
```

```
> mae.ngdpMCMC = sqrt(mean(abs(ngdpForeMCMC - NGDP.test)))
> tic.ngdpMCMC = sqrt(mean((ngdpForeMCMC - NGDP.test)^2))/
(sqrt(mean(ngdpForeMCMC^2)) + sqrt(mean(NGDP.test^2)))
> rm(dV)
> rm(dW)
>
> # IPI
> ipiGibbs = dlmGibbsDIG(IPI.tr, dlmModPoly(2), a.y = 1, b.y = 1000,
+ a.theta = 1, b.theta = 1000, n.sample = 1100, ind = c(1, 2))
> burn = 100
> attach(ipiGibbs)
> dV = dV[-(1:burn)]
> dW = dW[-(1:burn), ]
> detach()
> par(mfrow = c(2, 3), mar = c(3.1, 2.1, 2.1, 1.1))
> plot(dV, type = 'l', xlab = "", ylab = "", main =
expression(sigma[epsilon]^2))
> plot(dW[, 1], type = 'l', xlab = "", ylab = "", main =
expression(sigma[mu]^2))
> plot(dW[, 2], type = 'l', xlab = "", ylab = "", main =
expression(sigma[beta]^2))
> use = length(dV) - burn
> from = 0.05 * use
> at = pretty(c(0, use), n = 3)
> at = at[at >= from]
> plot(ergMean(dV, from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 1], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 2], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> mcmcMean(cbind(dV, dW))
> graphics.off()
> plot(IPI, type = 'o', col = 'seagreen')
> ipiMCMC = ts(dropFirst(ipiGibbs$theta[1:length(IPI.tr), 1, 1091:1100]), start
= c(1963, 1), freq = 4)
> for (i in 1:10)
+ lines(ipiMCMC[, i], col = 'brown')
> legend('topleft', legend = c('data', 'simulated level'), col = c('seagreen',
'brown'), pch = c(1, NA), lty = c(NA, 1))
>
> # FORECASTING
```

```
> parmMCMC = mcmcMean(cbind(dV, dW))[c(1, 3, 5)]
> ipiModMCMC = dlmModPoly(2, dV = parmMCMC[1], dW = parmMCMC[c(2:3)])
> ipiFiltMCMC = dlmFilter(IPI.tr, mod = ipiModMCMC)
> forecastMCMC = dlmForecast(ipiFiltMCMC, nAhead = 12)
> sqrtR = sapply(forecastMCMC$R, function(x) sqrt(x[1, 1]))
> ipiForeMCMC = ts(forecastMCMC$a[, 1], start = c(2011, 1), freq = 4)
> pl = ipiForeMCMC + qnorm(0.05, sd = sqrtR)
> pu = ipiForeMCMC + qnorm(0.95, sd = sqrtR)
> x = ts.union(window(IPI, start = c(2009, 1)),
+ ipiForeMCMC, pl, pu)
> plot(x, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'),
+ ylab = 'IPI')
> legend('topleft', legend = c('Observed', 'Forecast level', '90% confidence
interval'), bty = 'n', pch = c(1, 20, 3, 3), lty = 1, col = c('black', 'brown',
'yellow', 'yellow'))
>
> # FORECAST PERFORMANCE MEASURES
> mape.ipiMCMC = 100 * mean(abs((ipiForeMCMC - IPI.test)/IPI.test))
> tic.ipiMCMC = sqrt(mean((ipiForeMCMC - IPI.test)^2))/
(sqrt(mean(ipiForeMCMC^2)) + sqrt(mean(IPI.test^2)))
> rm(dV)
> rm(dW)
>
> # CPI
> cpiGibbs = dlmGibbsDIG(CPI.tr, dlmModPoly(2), a.y = 1, b.y = 1000,
+ a.theta = 1, b.theta = 1000, n.sample = 1100, ind = c(1, 2))
> burn = 100
> attach(cpiGibbs)
> dV = dV[-(1:burn)]
> dW = dW[-(1:burn), ]
> detach()
> par(mfrow = c(2, 3), mar = c(3.1, 2.1, 2.1, 1.1))
> plot(dV, type = 'l', xlab = "", ylab = "", main =
expression(sigma[epsilon]^2))
> plot(dW[, 1], type = 'l', xlab = "", ylab = "", main =
expression(sigma[mu]^2))
> plot(dW[, 2], type = 'l', xlab = "", ylab = "", main =
expression(sigma[beta]^2))
> use = length(dV) - burn
> from = 0.05 * use
> at = pretty(c(0, use), n = 3)
```

```
> at = at[at >= from]
> plot(ergMean(dV, from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 1], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 2], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> mcmcMean(cbind(dV, dW))
> graphics.off()
> plot(CPI, type = 'o', col = 'seagreen')
> cpiMCMC = ts(dropFirst(cpiGibbs$theta[1:length(CPI.tr), 1, 1091:1100]), start
= c(1963, 1), freq = 4)
> for (i in 1:10)
+ lines(cpiMCMC[, i], col = 'brown')
> legend('topleft', legend = c('data', 'simulated level'), col = c('seagreen',
'brown'), pch = c(1, NA), lty = c(NA, 1))

>
> # FORECASTING
> parmMCMC = mcmcMean(cbind(dV, dW))[c(1, 3, 5)]
> cpiModMCMC = dlmModPoly(2, dV = parmMCMC[1], dW = parmMCMC[c(2:3)])
> cpiFiltMCMC = dlmFilter(CPI.tr, mod = cpiModMCMC)
> forecastMCMC = dlmForecast(cpiFiltMCMC, nAhead = 12)
> sqrtR = sapply(forecastMCMC$R, function(x) sqrt(x[1, 1]))
> cpiForeMCMC = ts(forecastMCMC$a[, 1], start = c(2011, 1), freq = 4)
> pl = cpiForeMCMC + qnorm(0.05, sd = sqrtR)
> pu = cpiForeMCMC + qnorm(0.95, sd = sqrtR)
> x = ts.union(window(CPI, start = c(2009, 1)),
+ cpiForeMCMC, pl, pu)
> plot(x, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'),
+ ylab = 'CPI')
> legend('topleft', legend = c('Observed', 'Forecast level', '90% confidence
interval'), bty = 'n', pch = c(1, 20, 3, 3), lty = 1, col = c('black', 'brown',
'yellow', 'yellow'))
>
> # FORECAST PERFORMANCE MEASURES
> mape.cpiMCMC = 100 * mean(abs((cpiForeMCMC - CPI.test)/CPI.test))
> tic.cpiMCMC = sqrt(mean((cpiForeMCMC - CPI.test)^2))/
(sqrt(mean(cpiForeMCMC^2)) + sqrt(mean(CPI.test^2)))
> rm(dV)
> rm(dW)
```

```
>
> # TBILL
> tbillGibbs = dlmGibbsDIG(TBILL.tr, dlmModPoly(2) + dlmModSeas(4), a.y = 1, b.y
= 1000,
+ a.theta = 1, b.theta = 1000, n.sample = 1100, ind = c(1, 2, 3))
> burn = 100
> attach(tbillGibbs)
> dV = dV[-(1:burn)]
> dW = dW[-(1:burn), ]
> detach()
> par(mfrow = c(2, 4), mar = c(3.1, 2.1, 2.1, 1.1))
> plot(dV, type = 'l', xlab = "", ylab = "", main =
expression(sigma[epsilon]^2))
> plot(dW[, 1], type = 'l', xlab = "", ylab = "", main =
expression(sigma[mu]^2))
> plot(dW[, 2], type = 'l', xlab = "", ylab = "", main =
expression(sigma[beta]^2))
> plot(dW[, 3], type = 'l', xlab = "", ylab = "", main = expression(sigma[s]^2))
> use = length(dV) - burn
> from = 0.05 * use
> at = pretty(c(0, use), n = 3)
> at = at[at >= from]
> plot(ergMean(dV, from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 1], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 2], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> plot(ergMean(dW[, 3], from), type = 'l', xaxt = 'n', xlab = "", ylab = "")
> axis(1, at = at - from, labels = format(at))
> mcmcMean(cbind(dV, dW))
> graphics.off()
> plot(TBILL, type = 'o', col = 'seagreen')
> tbillMCMC = ts(dropFirst(tbillGibbs$theta[1:length(TBILL.tr), 1, 1091:1100]),
start = c(1963, 1), freq = 4)
> for (i in 1:10)
+ lines(tbillMCMC[, i], col = 'brown')
> legend('topleft', legend = c('data', 'simulated level'), col = c('seagreen',
'brown'), pch = c(1, NA), lty = c(NA, 1))
>
> # FORECASTING
> parmMCMC = mcmcMean(cbind(dV, dW))[c(1, 3, 5, 7)]
```

```
> tbillModMCMC = dlmModPoly(2, dV = parmMCMC[1], dW = parmMCMC[c(2:3)]) +
dlmModSeas(4, dV = 0, dW = c(parmMCMC[4], rep(0, 2)))
> tbillFiltMCMC = dlmFilter(TBILL.tr, mod = tbillModMCMC)
> forecastMCMC = dlmForecast(tbillFiltMCMC, nAhead = 12)
> sqrtR = sapply(forecastMCMC$R, function(x) sqrt(x[1, 1]))
> tbillForeMCMC = ts(forecastMCMC$a[, 1], start = c(2011, 1), freq = 4)
> pl = tbillForeMCMC + qnorm(0.05, sd = sqrtR)
> pu = tbillForeMCMC + qnorm(0.95, sd = sqrtR)
> x = ts.union(window(TBILL, start = c(2009, 1)),
+ tbillForeMCMC, pl, pu)
> plot(x, plot.type = 'single', type = 'o', pch = c(1, 20, 3, 3),
+ col = c('black', 'brown', 'yellow', 'yellow'),
+ ylab = 'TBILL')
> legend('topleft', legend = c('Observed', 'Forecast level', '90% confidence
interval'), bty = 'n', pch = c(1, 20, 3, 3), lty = 1, col = c('black', 'brown',
'yellow', 'yellow'))
>
> # FORECAST PERFORMANCE MEASURES
> mape.tbillMCMC = 100 * mean(abs((tbillForeMCMC - TBILL.test)/TBILL.test))
> tic.tbillMCMC = sqrt(mean((tbillForeMCMC - TBILL.test)^2))/
(sqrt(mean(tbillForeMCMC^2)) + sqrt(mean(TBILL.test^2)))
> rm(dV)
> rm(dW)
```

Appendix B

File `database.txt`

observation_date	GDP	INDPRO	CPI	TB3MS
1963-01-01	622.700	26.125	30.477	2.910
1963-04-01	631.800	26.814	30.533	2.940
1963-07-01	645.000	26.993	30.720	3.290
1963-10-01	654.800	27.449	30.803	3.500
1964-01-01	671.100	27.825	30.930	3.530
1964-04-01	680.800	28.478	30.980	3.480
1964-07-01	692.800	28.934	31.050	3.500
1964-10-01	698.400	29.373	31.193	3.680
1965-01-01	719.200	30.482	31.290	3.890
1965-04-01	732.400	31.189	31.490	3.870
1965-07-01	750.200	31.842	31.583	3.870
1965-10-01	773.100	32.486	31.750	4.170
1966-01-01	797.300	33.417	32.047	4.610
1966-04-01	807.200	34.115	32.337	4.590
1966-07-01	820.800	34.643	32.617	5.040
1966-10-01	834.900	34.974	32.883	5.210
1967-01-01	846.000	34.781	32.967	4.510
1967-04-01	851.100	34.639	33.167	3.660
1967-07-01	866.600	34.877	33.500	4.300
1967-10-01	883.200	35.811	33.867	4.750
1968-01-01	911.100	36.322	34.200	5.050
1968-04-01	936.300	36.812	34.533	5.520
1968-07-01	952.300	37.097	35.000	5.200
1968-10-01	970.100	37.660	35.433	5.590
1969-01-01	995.400	38.393	35.867	6.090
1969-04-01	1011.400	38.561	36.433	6.200
1969-07-01	1032.000	39.024	36.933	7.020
1969-10-01	1040.700	38.779	37.500	7.350
1970-01-01	1053.500	37.839	38.100	7.210
1970-04-01	1070.100	37.631	38.633	6.680

1970-07-01	1088.500	37.497	39.033	6.330
1970-10-01	1091.500	36.686	39.600	5.350
1971-01-01	1137.800	37.393	39.933	3.840
1971-04-01	1159.400	37.732	40.300	4.250
1971-07-01	1180.300	37.848	40.700	5.010
1971-10-01	1193.600	38.726	41.000	4.230
1972-01-01	1233.800	40.368	41.333	3.440
1972-04-01	1270.100	41.138	41.600	3.770
1972-07-01	1293.800	41.659	41.933	4.220
1972-10-01	1332.000	43.122	42.367	4.860
1973-01-01	1380.700	44.349	43.033	5.700
1973-04-01	1417.600	44.710	43.933	6.600
1973-07-01	1436.800	45.098	44.800	8.320
1973-10-01	1479.100	45.771	45.933	7.500
1974-01-01	1494.700	45.355	47.300	7.620
1974-04-01	1534.200	45.391	48.567	8.150
1974-07-01	1563.400	45.190	49.933	8.190
1974-10-01	1603.000	43.402	51.467	7.360
1975-01-01	1619.600	40.532	52.567	5.750
1975-04-01	1656.400	40.010	53.200	5.390
1975-07-01	1713.800	41.008	54.267	6.330
1975-10-01	1765.900	41.887	55.267	5.630
1976-01-01	1824.500	43.173	55.900	4.920
1976-04-01	1856.900	43.719	56.400	5.160
1976-07-01	1890.500	44.289	57.300	5.150
1976-10-01	1938.400	45.083	58.133	4.670
1977-01-01	1992.500	46.016	59.200	4.630
1977-04-01	2060.200	47.434	60.233	4.840
1977-07-01	2122.400	47.964	61.067	5.500
1977-10-01	2168.700	48.274	61.967	6.110
1978-01-01	2208.700	48.136	63.033	6.390
1978-04-01	2336.600	50.069	64.467	6.480
1978-07-01	2398.900	50.508	65.967	7.310
1978-10-01	2482.200	51.419	67.500	8.570
1979-01-01	2531.600	51.644	69.200	9.380
1979-04-01	2595.900	51.557	71.400	9.380
1979-07-01	2670.400	51.394	73.700	9.670
1979-10-01	2730.700	51.581	76.033	11.840
1980-01-01	2796.500	51.787	79.033	13.350
1980-04-01	2799.900	49.618	81.700	9.620
1980-07-01	2860.000	48.834	83.233	9.150
1980-10-01	2993.500	50.700	85.567	13.610

1981-01-01	3131.800	50.832	87.933	14.390
1981-04-01	3167.300	50.994	89.767	14.910
1981-07-01	3261.200	51.472	92.267	15.050
1981-10-01	3283.500	50.336	93.767	11.750
1982-01-01	3273.800	49.313	94.600	12.810
1982-04-01	3331.300	48.696	95.967	12.420
1982-07-01	3367.100	48.016	97.633	9.320
1982-10-01	3407.800	47.120	97.933	7.910
1983-01-01	3480.300	47.655	98.000	8.110
1983-04-01	3583.800	48.712	99.133	8.400
1983-07-01	3692.300	50.381	100.100	9.140
1983-10-01	3796.100	51.695	101.100	8.800
1984-01-01	3912.800	53.215	102.533	9.170
1984-04-01	4015.000	54.039	103.500	9.800
1984-07-01	4087.400	54.422	104.400	10.320
1984-10-01	4147.600	54.464	105.300	8.800
1985-01-01	4237.000	54.616	106.267	8.180
1985-04-01	4302.300	54.666	107.233	7.460
1985-07-01	4394.600	54.574	107.900	7.110
1985-10-01	4453.100	54.902	109.000	7.170
1986-01-01	4516.300	55.222	109.567	6.900
1986-04-01	4555.200	54.898	109.033	6.140
1986-07-01	4619.600	55.121	109.700	5.520
1986-10-01	4669.400	55.730	110.467	5.350
1987-01-01	4736.200	56.470	111.800	5.540
1987-04-01	4821.500	57.460	113.067	5.660
1987-07-01	4900.500	58.515	114.267	6.040
1987-10-01	5022.700	59.950	115.333	5.860
1988-01-01	5090.600	60.493	116.233	5.720
1988-04-01	5207.700	60.994	117.567	6.210
1988-07-01	5299.500	61.237	119.000	7.010
1988-10-01	5412.700	61.669	120.300	7.730
1989-01-01	5527.400	61.946	121.667	8.540
1989-04-01	5628.400	61.728	123.633	8.410
1989-07-01	5711.600	61.319	124.600	7.840
1989-10-01	5763.400	61.567	125.867	7.650
1990-01-01	5890.800	61.985	128.033	7.760
1990-04-01	5974.700	62.470	129.300	7.750
1990-07-01	6029.500	62.720	131.533	7.480
1990-10-01	6023.300	61.745	133.767	6.990
1991-01-01	6054.900	60.565	134.767	6.020
1991-04-01	6143.600	60.934	135.567	5.560

1991-07-01	6218.400	61.749	136.600	5.380
1991-10-01	6279.300	61.874	137.733	4.540
1992-01-01	6380.800	61.794	138.667	3.890
1992-04-01	6492.300	62.872	139.733	3.680
1992-07-01	6586.500	63.342	140.800	3.080
1992-10-01	6697.600	64.018	142.033	3.070
1993-01-01	6748.200	64.591	143.067	2.960
1993-04-01	6829.600	64.724	144.100	2.970
1993-07-01	6904.200	64.984	144.767	3.000
1993-10-01	7032.800	65.966	145.967	3.060
1994-01-01	7136.300	66.794	146.700	3.240
1994-04-01	7269.800	68.020	147.533	3.990
1994-07-01	7352.300	68.906	148.900	4.480
1994-10-01	7476.700	70.308	149.767	5.280
1995-01-01	7545.300	71.067	150.867	5.740
1995-04-01	7604.900	71.328	152.100	5.600
1995-07-01	7706.500	72.019	152.867	5.370
1995-10-01	7799.500	72.622	153.700	5.260
1996-01-01	7893.100	73.105	155.067	4.930
1996-04-01	8061.500	74.613	156.400	5.020
1996-07-01	8159.000	75.530	157.300	5.100
1996-10-01	8287.100	76.548	158.667	4.980
1997-01-01	8402.100	78.013	159.633	5.060
1997-04-01	8551.900	79.208	160.000	5.050
1997-07-01	8691.800	81.077	160.800	5.050
1997-10-01	8788.300	83.126	161.667	5.090
1998-01-01	8889.700	84.041	162.000	5.050
1998-04-01	8994.700	84.565	162.533	4.980
1998-07-01	9146.500	85.143	163.367	4.820
1998-10-01	9325.700	86.319	164.133	4.250
1999-01-01	9447.100	87.230	164.733	4.410
1999-04-01	9557.000	88.045	165.967	4.450
1999-07-01	9712.300	88.872	167.200	4.650
1999-10-01	9926.100	90.493	168.433	5.040
2000-01-01	10031.000	91.446	170.100	5.520
2000-04-01	10278.300	92.629	171.433	5.710
2000-07-01	10357.400	92.541	173.000	6.020
2000-10-01	10472.300	92.253	174.233	6.020
2001-01-01	10508.100	90.927	175.900	4.820
2001-04-01	10638.400	89.652	177.133	3.660
2001-07-01	10639.500	88.340	177.633	3.170
2001-10-01	10701.300	87.325	177.500	1.910

2002-01-01	10834.400	87.952	178.067	1.720
2002-04-01	10934.800	89.352	179.467	1.720
2002-07-01	11037.100	89.879	180.433	1.640
2002-10-01	11103.800	89.824	181.500	1.330
2003-01-01	11230.100	90.400	183.367	1.160
2003-04-01	11370.700	89.695	183.067	1.040
2003-07-01	11625.100	90.283	184.433	0.930
2003-10-01	11816.800	91.040	185.133	0.920
2004-01-01	11988.400	91.606	186.700	0.920
2004-04-01	12181.400	92.013	188.167	1.080
2004-07-01	12367.700	92.495	189.367	1.490
2004-10-01	12562.200	93.779	191.400	2.010
2005-01-01	12813.700	95.093	192.367	2.540
2005-04-01	12974.100	95.604	193.667	2.860
2005-07-01	13205.400	95.187	196.600	3.360
2005-10-01	13381.600	95.989	198.433	3.830
2006-01-01	13648.900	96.895	199.467	4.390
2006-04-01	13799.800	97.472	201.267	4.700
2006-07-01	13908.500	97.848	203.167	4.910
2006-10-01	14066.400	98.041	202.333	4.900
2007-01-01	14233.200	98.925	204.317	4.980
2007-04-01	14422.300	100.119	206.631	4.740
2007-07-01	14569.700	100.388	207.939	4.300
2007-10-01	14685.300	100.567	210.490	3.390
2008-01-01	14668.400	100.223	212.770	2.040
2008-04-01	14813.000	98.863	215.538	1.630
2008-07-01	14843.000	95.740	218.861	1.490
2008-10-01	14549.900	91.648	213.849	0.300
2009-01-01	14383.900	86.689	212.378	0.210
2009-04-01	14340.400	84.243	213.507	0.170
2009-07-01	14384.100	85.313	215.344	0.160
2009-10-01	14566.500	86.626	217.030	0.060
2010-01-01	14681.100	88.369	217.341	0.110
2010-04-01	14888.600	90.240	217.320	0.150
2010-07-01	15057.700	91.667	217.990	0.160
2010-10-01	15230.200	92.010	219.668	0.140
2011-01-01	15238.400	92.578	221.951	0.130
2011-04-01	15460.900	92.850	224.655	0.050
2011-07-01	15587.100	94.026	226.125	0.020
2011-10-01	15785.300	94.920	226.997	0.010
2012-01-01	15956.500	96.132	228.179	0.070
2012-04-01	16094.700	97.024	228.964	0.090

2012-07-01	16268.900	97.402	229.939	0.100
2012-10-01	16332.500	97.976	231.314	0.090
2013-01-01	16502.400	98.980	231.998	0.090
2013-04-01	16619.200	99.445	232.230	0.050
2013-07-01	16872.300	100.053	233.476	0.030
2013-10-01	17078.300	101.250	234.136	0.060