2

Dynamic linear models

In this chapter we discuss the basic notions about state space models and their use in time series analysis. The dynamic linear model is presented as a special case of a general state space model, being linear and Gaussian. For dynamic linear models, estimation and forecasting can be obtained recursively by the well-known Kalman filter.

2.1 Introduction

In recent years there has been an increasing interest in the application of state space models in time series analysis; see, for example, Harvey (1989), West and Harrison (1997), Durbin and Koopman (2001), the recent overviews by Künsch (2001) and Migon et al. (2005), and the references therein. State space models consider a time series as the output of a dynamic system perturbed by random disturbances. They allow a natural interpretation of a time series as the combination of several components, such as trend, seasonal or regressive components. At the same time, they have an elegant and powerful probabilistic structure, offering a flexible framework for a very wide range of applications. Computations can be implemented by recursive algorithms. The problems of estimation and forecasting are solved by recursively computing the conditional distribution of the quantities of interest, given the available information. In this sense, they are quite naturally treated within a Bayesian framework.

State space models can be used to model univariate or multivariate time series, also in the presence of non-stationarity, structural changes, and irregular patterns. In order to develop a feeling for the possible applications of state space models in time series analysis, consider for example the data plotted in Figure 2.1. This time series appears fairly predictable, since it repeats quite regularly its behavior over time: we see a trend and a rather regular seasonal component, with a slightly increasing variability. For data of this kind, we would probably be happy with a fairly simple time series model, with a trend

G. Petris et al., *Dynamic Linear Models with R*, Use R, DOI: 10.1007/b135794_2,

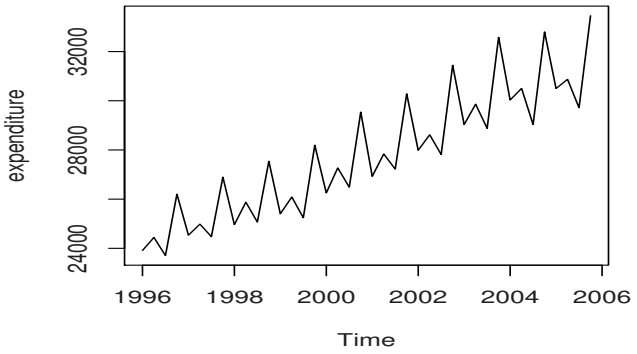

Fig. 2.1. Family food expenditure, quarterly data (1996Q1 to 2005Q4). Data available from `http://con.istat.it`

and a seasonal component. In fact, basic time series analysis relies on the possibility of finding a reasonable regularity in the behavior of the phenomenon under study: forecasting future behavior is clearly easier if the series tends to repeat a regular pattern over time. Things get more complex for time series

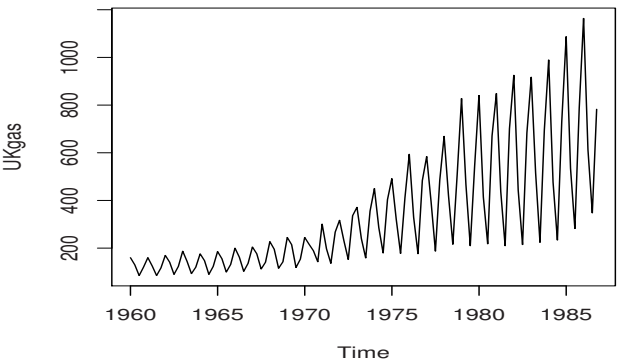

Fig. 2.2. Quarterly UK gas consumption from 1960Q1 to 1986Q4, in millions of therms

such as the ones plotted in Figures 2.2-2.4. Figure 2.2 shows the quarterly UK gas consumption from 1960 to 1986 (the data are available in R as *`UKgas`*). We clearly see a change in the seasonal component. Figure 2.3 shows a well-studied

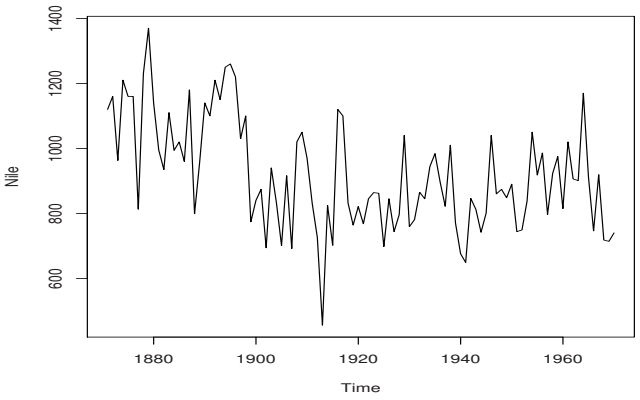

Fig. 2.3. Measurements of the annual flow of the river Nile at Ashwan, 1871-1970

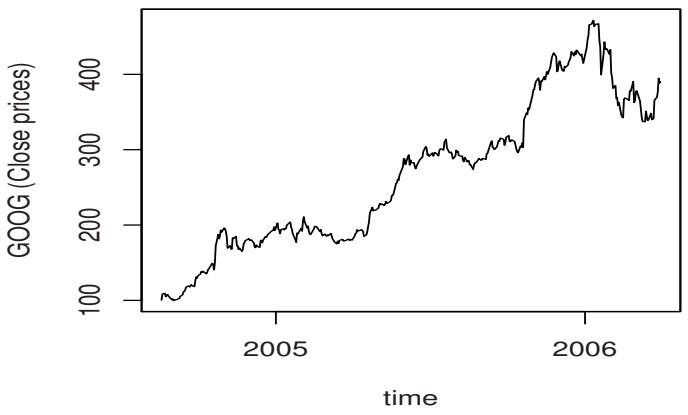

Fig. 2.4. Daily prices for Google Inc. (GOOG)

data set: the measurements of the annual flow of the river Nile at Ashwan from 1871 to 1970. The series shows level shifts. We know that the construction of the first dam of Ashwan started in 1898; the second big dam was completed in 1971: if you have ever seen these huge dams, you can easily understand the enormous changes that they caused on the Nile flow and in the vast surrounding area. Thus, we begin to feel the need for more flexible time series models, which do not assume a regular pattern and stability of the underlying system, but can include change points or structural breaks. Possibly more irregular is

the series plotted in Figure 2.4, showing daily prices of Google[1](close prices, 2004-08-19 to 2006-03-31). This series looks clearly nonstationary and in fact quite irregular: indeed, we know how unstable the market for the new economy has been in those years. The analysis of nonstationary time series with ARMA models requires at least a preliminary transformation of the data to get stationarity; but we might feel more natural to have models that allow us to analyze more directly data that show instability in the mean level and in the variance, structural breaks, and sudden jumps. State space models include ARMA models as a special case, but can be applied to nonstationary time series without requiring a preliminary transformation of the data. But there is a further basic issue. When dealing with economic or financial data, for example, a univariate time series model is often quite limited. An economist might want to gain a deeper understanding of the economic system, looking for example at relevant macroeconomic variables that influence the variable of specific interest. For the financial example of Figure 2.4, a univariate series model might be satisfying for high frequency data (the data in Figure 2.4 are daily prices), quickly *adapting* to irregularities, structural breaks or jumps; however, it will be hardly capable of *predicting* sudden changes without a further effort in a deeper and broader study of the economic and socio-political variables that influence the markets. Even then, forecasting sudden changes is clearly not at all an easy task! But we do feel that it is desirable to include regression terms in our model or use multivariate time series models. Including regression terms is quite natural in state space time series models. And state space models can in general be formulated for multivariate time series.

State space models originated in engineering in the early sixties, although the problem of forecasting has always been a fundamental and fascinating issue in the theory of stochastic processes and time series. Kolmogorov (1941) studied this problem for discrete time stationary stochastic processes, using a representation proposed by Wold (1938). Wiener (1949) studied continuous time stochastic processes, reducing the problem of forecasting to the solution of the so-called Wiener–Hopf integral equation. However, the methods for solving the Wiener problem were subject to several theoretical and practical limitations. A new look at the problem was given by Kalman (1960), using the Bode–Shannon representation of random processes and the "state transition" method of analsyis of dynamical systems. Kalman's solution, known as the Kalman filter (Kalman; 1960; Kalman and Bucy; 1963), applies to stationary and nonstationary random processes. These methods quickly gained popularity in other fields and were applied to a wide array of problems, from the determination of the orbits of the Voyager spacecraft to oceanographic problems, from agriculture to economics and speech recognition (see for instance the special issue of the IEEE Transactions on Automatic Control (1983) dedicated to applications of the Kalman filter). The importance of these methods

[1] Financial data can be easily downloaded in R using the function `get.hist.quote` in package `tseries`, or the function `priceIts` in package `its`.

was recognized by statisticians only later, although the idea of latent variables and recursive estimation can be found in the statistical literature at least as early as Thiele (1880) and Plackett (1950); see Lauritzen (1981). One reason for this delay is that the work on the Kalman filter was mostly published in the engineering literature. This means not only that the language of these works was not familiar to statisticians, but also that some issues that are crucial in applications in statistics and time series analysis were not sufficiently understood yet. Kalman himself, in his 1960 paper, underlines that the problem of obtaining the transition model, which is crucial in practical applications, was treated as a separate question and not solved. In the engineering literature, it was common practice to assume the structure of the dynamic system as known, except for the effects of random disturbances, the main problem being to find an optimal estimate of the state of the system, given the model. In time series analysis, the emphasis is somehow different. The physical interpretation of the underlying states of the dynamic system is often less evident than in engineering applications. What we have is the observable process, and even if it may be convenient to think of it as the output of a dynamic system, the problem of forecasting is often the most relevant. In this context, model building can be more difficult, and even when a state space representation is obtained, there are usually quantities or parameters in the model that are unknown and need to be estimated.

State space models appeared in the time series literature in the seventies (Akaike; 1974a; Harrison and Stevens; 1976) and became established during the eighties (Harvey; 1989; West and Harrison; 1997; Aoki; 1987). In the last decades they have become a focus of interest. This is due on one hand to the development of models well suited to time series analysis, but also to a wider range of applications, including, for instance, molecular biology or genetics, and on the other hand to the development of computational tools, such as modern Monte Carlo methods, for dealing with more complex nonlinear and non-Gaussian situations.

In the next sections we discuss the basic formulation of state space models and the structure of the recursive computations for estimation. Then, as a special case, we present the Kalman filter for Gaussian linear dynamic models.

2.2 A simple example

Before presenting the general formulation of state space models, it is useful to give an intuition of the basic ideas and of the recursive computations through a simple, introductory example. Let's think of the problem of determining the position θ of an object, based on some measurements $(Y_t : t = 1, 2, \ldots)$ affected by random errors. This problem is fairly intuitive, and dynamics can be incorporated into it quite naturally: in the static problem, the object does not move over time, but it is natural to extend the discussion to the case of a moving target. If you prefer, you may think of some economic problem, such as

forecasting the sales of a good; in short-term forecasting, the observed sales are often modeled as measurements of the unobservable average sales level plus a random error; in turn, the average sales are supposed to be constant or randomly evolving over time (this is the so-called random walk plus noise model, see page 42).

We have already discussed Bayesian inference in the static problem in Chapter 1 (page 7). There, you were lost at sea, on a small island, and θ was your unknown position (univariate: distance from the coast, say). The observations were modeled as

$$Y_t = \theta + \epsilon_t, \qquad \epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2);$$

that is, given θ, the Y_t's are conditionally independent and identically distributed with a $\mathcal{N}(\theta, \sigma^2)$ distribution; in turn, θ has a Normal prior $\mathcal{N}(m_0, C_0)$. As we have seen in Chapter 1, the posterior for θ is still Gaussian, with updated parameters given by (1.2), or by (1.3) if we compute them sequentially, as new data become available.

To be concrete, let us suppose that your prior guess about the position θ is $m_0 = 1$, with variance $C_0 = 2$; the prior density is plotted in the first panel of Figure 2.5. Note that m_0 is also your point forecast for the observation: $\mathrm{E}(Y_1) = \mathrm{E}(\theta + \epsilon_1) = \mathrm{E}(\theta) = m_0 = 1$.

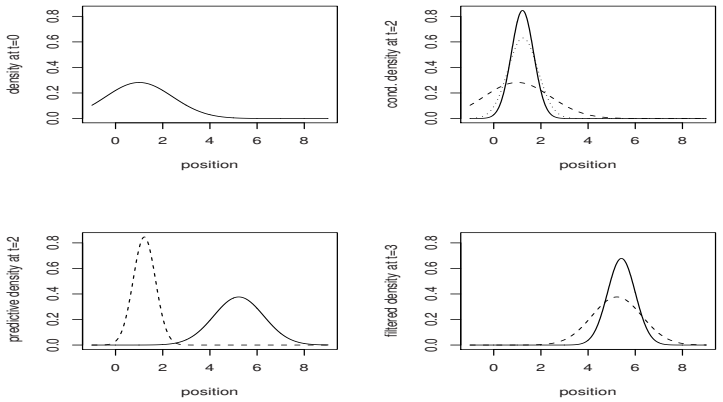

Fig. 2.5. Recursive updating of the density of θ_t

At time $t = 1$, we take a measurement, $Y_1 = 1.3$, say; from (1.3), the parameters of the posterior Normal density of θ are

$$m_1 = m_0 + \frac{C_0}{C_0 + \sigma^2}(Y_1 - m_0) = 1.24,$$

with precision $C_1^{-1} = \sigma^{-2} + C_0^{-1} = 0.4^{-1}$. We see that m_1 is obtained as our best guess at time zero, m_0, corrected by the forecast error $(Y_1 - m_0)$, weighted by a factor $K_1 = C_0/(C_0 + \sigma^2)$. The more precise the observation is, or the more vague our initial information was, the more we "trust the data": in the above formula, the smaller σ^2 is with respect to C_0, the bigger is the weight K_1 of the data-correction term in m_1. When a new observation, $Y_2 = 1.2$ say, becomes available at time $t = 2$, we can compute the density of $\theta|Y_{1:2}$, which is $\mathcal{N}(m_2, C_2)$, with $m_2 = 1.222$ and $C_2 = 0.222$, using again (1.3). The second panel in Figure 2.5 shows the updating from the prior density to the posterior density of θ, given $y_{1:2}$. We can proceed recursively in this manner as new data become available.

Let us introduce now a dynamic component to the problem. Suppose we know that at time $t = 2$ the object starts to move, so that its position changes between two consecutive measurements. Let us assume a motion of a simple form, say[2]

$$\theta_t = \theta_{t-1} + \nu + w_t, \qquad w_t \sim \mathcal{N}(0, \sigma_w^2). \tag{2.1}$$

where ν is a known nominal speed and w_t is a Gaussian random error with mean zero and known variance σ_w^2. Let, for example, $\nu = 4.5$ and $\sigma_w^2 = 0.9$. Thus, we have a process $(\theta_t : t = 1, 2, \ldots)$, which describes the unknown position of the target at successive time points. The observation equation is now

$$Y_t = \theta_t + \epsilon_t, \qquad \epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2), \tag{2.2}$$

and we assume that the sequences (θ_t) and (ϵ_t) are independent. To make inference about the unknown position θ_t, we proceed along the following steps.

Initial step. By the previous results, at time $t = 2$ we have

$$\theta_2|y_{1:2} \sim \mathcal{N}(m_2 = 1.222, C_2 = 0.222).$$

Prediction step. At time $t = 2$, we can predict where the object will be at time $t = 3$, based on the dynamics (2.1). We easily find that

[2] Equation (2.1) can be thought of as a discretization of a motion law in continuous time, such as

$$d\theta_t = \nu dt + dW_t$$

where ν is the nominal speed and dW_t is an error term. For simplicity, we consider a discretization in small intervals of time (t_{i-1}, t_i), as follows:

$$\frac{\theta_{t_i} - \theta_{t_{i-1}}}{t_i - t_{i-1}} = \nu + w_{t_i},$$

that is

$$\theta_{t_i} = \theta_{t_{i-1}} + \nu(t_i - t_{i-1}) + w_{t_i}(t_i - t_{i-1}),$$

where we assume that the random error w_{t_i} has density $\mathcal{N}(0, \sigma_w^2)$. With a further simplification, we take unitary time intervals, $(t_i - t_{i-1}) = 1$, so that the above expression is rewritten as (2.1).

$$\theta_3|y_{1:2} \sim \mathcal{N}(a_3, R_3),$$

with

$$a_3 = \mathrm{E}(\theta_2 + \nu + w_3|y_{1:2}) = m_2 + \nu = 5.722$$

and variance

$$R_3 = \mathrm{Var}(\theta_2 + \nu + w_3|y_{1:2}) = C_2 + \sigma_w^2 = 1.122.$$

The third plot in Figure 2.5 illustrates the prediction step, from the conditional distribution of $\theta_2|y_{1:2}$ to the "predictive" distribution of $\theta_3|y_{1:2}$. Note that even if we were fairly confident about the position of the target at time $t = 2$, we become more uncertain about its position at time $t = 3$. This is the effect of the random error w_t in the dynamics of θ_t: the larger σ_w^2 is, the more uncertain we are about the position at the time of the next measurement. We can also predict the next observation Y_3, given $y_{1:2}$. Based on the observation equation (2.2), we easily find that

$$Y_3|y_{1:2} \sim \mathcal{N}(f_3, Q_3),$$

where

$$f_3 = \mathrm{E}(\theta_3 + \epsilon_3|y_{1:2}) = a_3 = 5.722$$

and

$$Q_3 = \mathrm{Var}(\theta_3 + \epsilon_3|y_{1:2}) = R_3 + \sigma^2 = 1.622.$$

The uncertainty about Y_3 depends on the measurement error (the term σ^2 in Q_3) as well as the uncertainty about the position at time $t = 3$ (expressed by R_3).

Estimation step (filtering). At time $t = 3$, the new observation $Y_3 = 5$ becomes available. Our point forecast of Y_3 was $f_3 = a_3 = 5.722$, so we have a forecast error $e_t = y_t - f_t = -0.722$. Intuitively, we have overestimated θ_3 and consequently Y_3; thus, our new estimate $\mathrm{E}(\theta_3|y_{1:3})$ of θ_3 will be smaller than $a_3 = \mathrm{E}(\theta_3|y_{1:2})$. For computing the posterior density of $\theta_3|y_{1:3}$, we use the Bayes formula, where the role of the prior is played by the density $\mathcal{N}(a_3, R_3)$ of θ_3 given $y_{1:2}$, and the likelihood is the density of Y_3 given (θ_3, y_1, y_2). Note that (2.2) implies that Y_3 is independent from the past observations given θ_3 (assuming independence among the error sequences), with

$$Y_3|\theta_3 \sim \mathcal{N}(\theta_3, \sigma^2).$$

Thus, by the Bayes formula (see (1.3)), we obtain

$$\theta_3|y_1, y_2, y_3 \sim \mathcal{N}(m_3, C_3),$$

where

$$m_3 = a_3 + \frac{R_3}{R_3 + \sigma^2}(y_3 - f_3) = 5.568$$

and

$$C_3 = \frac{\sigma^2 R_3}{\sigma^2 + R_3} = R_3 - \frac{R_3}{R_3 + \sigma^2} R_3 = 0.346.$$

We see again the estimation-correction structure of the updating mechanism in action. Our best estimate of θ_3 given the data $y_{1:3}$ is computed as our previous best estimate a_3, corrected by a fraction of the forecast error $e_3 = y_3 - f_3$, having weight $K_3 = R_3/(R_3 + \sigma^2)$. This weight is bigger the more uncertain we are about our forecast a_3 of θ_3 (that is, the larger R_3 is, which in turn depends on C_2 and σ_w^2) and the more precise the observation Y_3 is (i.e., the smaller σ^2 is). From these results we see that a crucial role in determining the effect of the data on estimation and forecasting is played by the magnitude of the system variance σ_w^2 relative to the observation variance σ^2, the so-called *signal-to-noise* ratio. The last plot in Figure 2.5 illustrates this estimation step. We can proceed repeating recursively the previous steps for updating our estimates and forecasts as new observations become available.

The previous simple example illustrates the basic aspects of dynamic linear models, which can be summarized as follows.

- The observable process $(Y_t : t = 1, 2, \ldots)$ is thought of as determined by a latent process $(\theta_t : t = 1, 2, \ldots)$, up to Gaussian random errors. If we knew the position of the object at successive time points, the Y_t's would be independent: what remains are only unpredictable measurement errors. Furthermore, the observation Y_t depends only on the position θ_t of the target at time t.
- The latent process (θ_t) has a fairly simple dynamics: θ_t does not depend on the entire past trajectory but only on the previous position θ_{t-1}, through a linear relationship, up to Gaussian random errors.
- Estimation and forecasting can be obtained sequentially, as new data become available.

The assumption of linearity and Gaussianity is specific to dynamic linear models, but the dependence structure of the processes (Y_t) and (θ_t) is part of the definition of a general state space model.

2.3 State space models

Consider a time series $(Y_t)_{t \geq 1}$. Specifying the joint finite-dimensional distributions of $(Y_1, \ldots, Y_t)$, for any $t \geq 1$, is not an easy task. In particular, in time series applications the assumptions of independence or exchangeability are seldom justified, since they would essentially make time irrelevant. Markovian dependence is arguably the simplest form of dependence among the Y_t's in which time has a definite role. We say that $(Y_t)_{t \geq 1}$ is a *Markov chain* if, for any $t > 1$,

$$\pi(y_t | y_{1:t-1}) = \pi(y_t | y_{t-1}).$$

This means that the information about Y_t carried by all the observations up to time $t-1$ is exactly the same as the information carried by y_{t-1} alone. Another way of saying the same thing is that Y_t and $Y_{1:t-2}$ are conditionally independent given y_{t-1}. For a Markov chain the finite-dimensional joint distributions can be written in the fairly simple form

$$\pi(y_{1:t}) = \pi(y_1) \cdot \prod_{j=2}^{t} \pi(y_j|y_{j-1}).$$

Assuming a Markovian structure for the observations is, however, not appropriate in many applications. State space models build on the relatively simple dependence structure of a Markov chain to define more complex models for the observations. In a state space model we assume that there is an unobservable Markov chain (θ_t), called the state process, and that Y_t is an imprecise measurement of θ_t. In engineering applications θ_t usually describes the state of a physically observable system that produced the output Y_t. On the other hand, in econometric applications θ_t is often a latent construct, which may, however, have a useful interpretation. In any case, one can think of (θ_t) as an auxiliary time series that facilitates the task of specifying the probability distribution of the observable time series (Y_t).

Formally, a state space model consists of an $\mathbb{R}^p$-valued time series $(\theta_t : t = 0, 1 \dots)$ and an $\mathbb{R}^m$-valued time series $(Y_t : t = 1, 2 \dots)$, satisfying the following assumptions.

(A.1) (θ_t) is a Markov chain.
(A.2) Conditionally on (θ_t), the Y_t's are independent and Y_t depends on θ_t only.

The consequence of (A.1)-(A.2) is that a state space model is completely specified by the initial distribution $\pi(\theta_0)$ and the conditional densities $\pi(\theta_t|\theta_{t-1})$ and $\pi(y_t|\theta_t)$, $t \geq 1$. In fact, for any $t > 0$,

$$\pi(\theta_{0:t}, y_{1:t}) = \pi(\theta_0) \cdot \prod_{j=1}^{t} \pi(\theta_j|\theta_{j-1})\pi(y_j|\theta_j). \tag{2.3}$$

From (2.3) one can derive, by conditioning or marginalization, any other distribution of interest. For example, the joint density of the observations $Y_{1:t}$ can be obtained by integrating out the θ_j's in (2.3); note however that in this way the simple product form of (2.3) is lost.

The information flow assumed by a state space model is represented in Figure 2.6. The graph in the figure is a special case of a directed acyclic graph (see Cowell et al.; 1999). The graphical representation of the model can be used to deduce conditional independence properties of the random variables occurring in a state space model. In fact, two sets of random variables, A and B, can be shown to be conditionally independent given a third set of variables, C, if and only if C separates A and B, i.e., if any path connecting

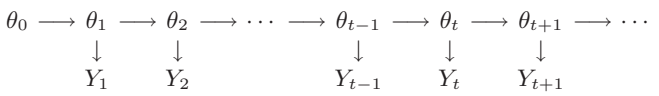

Fig. 2.6. Dependence structure for a state space model

one variable in A to one in B passes through C. Note that in the previous statement the arrows in Figure 2.6 have to be considered as undirected edges of the graph that can be transversed in both directions. For a proof, see Cowell et al. (1999, Section 5.3). As an example, we will use Figure 2.6 to show that Y_t and $(\theta_{0:t-1}, Y_{1:t-1})$ are conditionally independent given θ_t. The proof simply consists in observing that any path connecting Y_t with one of the previous Y_s $(s < t)$ or with one of the states θ_s, $s < t$, has to go through θ_t; hence, $\{\theta_t\}$ separates $\{\theta_{0:t-1}, Y_{1:t-1}\}$ and $\{Y_t\}$. It follows that

$$\pi(y_t|\theta_{0:t-1}, y_{1:t-1}) = \pi(y_t|\theta_t).$$

In a similar way, one can show that θ_t and $(\theta_{0:t-2}, Y_{1:t-1})$ are conditionally independent given θ_{t-1}, which can be expressed in terms of conditional distributions as

$$\pi(\theta_t|\theta_{0:t-1}, y_{1:t-1}) = \pi(\theta_t|\theta_{t-1}).$$

State space models in which the states are discrete-valued random variables are often called *hidden Markov models*.

2.4 Dynamic linear models.

The first, important class of state space models is given by Gaussian linear state space models, also called dynamic linear models. A *dynamic linear model* (DLM) is specified by a Normal prior distribution for the p-dimensional state vector at time $t = 0$,

$$\theta_0 \sim \mathcal{N}_p(m_0, C_0), \tag{2.4a}$$

together with a pair of equations for each time $t \geq 1$,

$$Y_t = F_t\theta_t + v_t, \qquad v_t \sim \mathcal{N}_m(0, V_t), \tag{2.4b}$$

$$\theta_t = G_t\theta_{t-1} + w_t, \qquad w_t \sim \mathcal{N}_p(0, W_t), \tag{2.4c}$$

where G_t and F_t are known matrices (of order $p\times p$ and $m\times p$ respectively) and $(v_t)_{t\geq 1}$ and $(w_t)_{t\geq 1}$ are two independent sequences of independent Gaussian random vectors with mean zero and known variance matrices $(V_t)_{t\geq 1}$ and $(W_t)_{t\geq 1}$, respectively. Equation (2.4b) is called the *observation equation*, while (2.4c) is the *state equation* or *system equation*. Furthermore, it is assumed that θ_0 is independent of (v_t) and (w_t). One can show that a DLM satisfies the

assumptions (A.1) and (A.2) of the previous section, with $Y_t|\theta_t \sim \mathcal{N}(F_t\theta_t, V_t)$ and $\theta_t|\theta_{t-1} \sim \mathcal{N}(G_t\theta_{t-1}, W_t)$ (see Problems 2.1 and 2.2).

In contrast to (2.4), a general state space model can be specified by a prior distribution for θ_0, together with the observation and evolution equations

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

for arbitrary functions g_t and h_t. *Linear* state space models specify g_t and h_t as linear functions, and *Gaussian* linear models add the assumptions of Gaussian distributions. The assumption of Normality is sensible in many applications, and it can be justified by central limit theorem arguments. However, there are many important extensions, such as heavy tailed errors for modeling outliers, or the dynamic generalized linear model for treating discrete time series. The price to be paid when removing the assumption of Normality is additional computational difficulties.

We introduce here some examples of DLMs for time series analysis, which will be treated more extensively in Chapter 3. The simplest model for a univariate time series $(Y_t : t = 1, 2, \ldots)$ is the so-called *random walk plus noise* model, defined by

$$\begin{aligned} Y_t &= \mu_t + v_t, & v_t &\sim \mathcal{N}(0, V) \\ \mu_t &= \mu_{t-1} + w_t, & w_t &\sim \mathcal{N}(0, W), \end{aligned} \tag{2.5}$$

where the error sequences (v_t) and (w_t) are independent, both within them and between them. This is a DLM with $m = p = 1$, $\theta_t = \mu_t$ and $F_t = G_t = 1$. It is the model used in the introductory example in Section 2.2, when there is no speed in the dynamics ($\nu = 0$ in the state equation (2.1)). Intuitively, it is appropriate for time series showing no clear trend or seasonal variation: the observations (Y_t) are modeled as noisy observations of a level μ_t which, in turn, is subject to random changes over time, described by a random walk. This is why the model is also called *local level* model. If $W = 0$, we are back to the constant mean model. Note that the random walk (μ_t) is nonstationary. Indeed, DLMs can be used for modeling nonstationary time series. On the contrary, the usual ARMA models require a preliminary transformation of the data to achieve stationarity.

A slightly more elaborated model is the *linear growth model*, or local linear trend, which has the same observation equation as the local level model, but includes a time-varying slope in the dynamics for μ_t:

$$\begin{aligned} Y_t &= \mu_t + v_t, & v_t &\sim \mathcal{N}(0, V), \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + w_{t,1}, & w_{t,1} &\sim \mathcal{N}(0, \sigma_\mu^2), \\ \beta_t &= \beta_{t-1} + w_{t,2}, & w_{t,2} &\sim \mathcal{N}(0, \sigma_\beta^2), \end{aligned} \tag{2.6}$$

with uncorrelated errors v_t, $w_{t,1}$ and $w_{t,2}$. This is a DLM with

$$\theta_t = \begin{bmatrix}\mu_t \\ \beta_t\end{bmatrix}, \quad G = \begin{bmatrix}1 & 1 \\ 0 & 1\end{bmatrix}, \quad W = \begin{bmatrix}\sigma_\mu^2 & 0 \\ 0 & \sigma_\beta^2\end{bmatrix}, \quad F = \begin{bmatrix}1 & 0\end{bmatrix}.$$

The system variances σ_μ^2 and σ_β^2 are allowed to be zero. We have used this model in the introductory example of Section 2.2; there, we had a constant nominal speed in the dynamics, that is $\sigma_\beta^2 = 0$.

Note that in these examples the matrices G_t and F_t and the covariance matrices V_t and W_t are constant; in this case the model is said to be *time invariant*. We will see other examples in Chapter 3. In particular, the popular Gaussian ARMA models can be obtained as special cases of DLM; in fact, it can be shown that Gaussian ARMA and DLM models are equivalent in the time-invariant case (see Hannan and Deistler; 1988).

DLMs can be regarded as a generalization of the linear regression model, allowing for time varying regression coefficients. The simple, static linear regression model describes the relationship between a variable Y and a nonrandom explanatory variable x as

$$Y_t = \theta_1 + \theta_2 x_t + \epsilon_t, \quad \epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

Here we think of $(Y_t, x_t), t = 1, 2, \ldots$ as observed over time. Allowing for time varying regression parameters, one can model nonlinearity of the functional relationship between x and y, structural changes in the process under study, omission of some variables. A simple dynamic linear regression model assumes

$$Y_t = \theta_{t,1} + \theta_{t,2} x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma_t^2),$$

with a further equation for describing the system evolution

$$\theta_t = G_t \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_2(0, W_t).$$

This is a DLM with $F_t = [1, x_t]$ and states $\theta_t = (\theta_{t,1}, \theta_{t,2})'$. As a particuar case, if $G_t = I$, the identity matrix, $\sigma_t^2 = \sigma^2$ and $w_t = 0$ for every t, we are back to the simple static linear regression model.

2.5 Dynamic linear models in package dlm

DLMs are represented in package dlm as named lists with a class attribute, which makes them into objects of class "`dlm`". Objects of class `dlm` can represent constant or time-varying DLMs. A constant DLM is completely specified once the matrices F, V, G, W, C_0, and the vector m_0 are given. In R, these components are stored in a `dlm` object as elements `FF`, `V`, `GG`, `W`, `C0`, and `m0`, respectively. Extractor and replacement functions are available to access and modify specific parts of the model in a user-friendly way. The package also provides several functions that create particular classes of DLMs from minimal

input; we will illustrate those functions in Chapter 3, where we discuss model specification. A general univariate or multivariate DLM can be specified using the function `dlm`. This function creates a `dlm` object from its components, performing some sanity checks on the input, such as testing the dimensions of the matrices for consistency. The input may be given as a list with named arguments or as individual arguments. Here is how to use `dlm` to create a `dlm` object corresponding to the random walk plus noise model and to the linear growth model introduced on page 42. We assume that $V = 1.4$ and $\sigma^2 = 0.2$. Note that 1×1 matrices can safely be passed to `dlm` as scalars, i.e., numerical vectors of length one.

R code

```
> rw <- dlm(m0 = 0, C0 = 10, FF = 1, V = 1.4, GG = 1, W = 0.2)
> unlist(rw)
  m0   C0   FF    V   GG    W
 0.0 10.0  1.0  1.4  1.0  0.2
> lg <- dlm(FF = matrix(c(1, 0), nr = 1),
+           V = 1.4,
+           GG = matrix(c(1, 0, 1, 1), nr = 2),
+           W = diag(c(0, 0.2)),
+           m0 = rep(0, 2),
+           C0 = 10 * diag(2))
> lg
$FF
     [,1] [,2]
[1,]    1    0

$V
     [,1]
[1,]  1.4

$GG
     [,1] [,2]
[1,]    1    1
[2,]    0    1

$W
     [,1] [,2]
[1,]    0  0.0
[2,]    0  0.2

$m0
[1] 0 0

$C0
```

```
     [,1] [,2]
[1,]   10    0
[2,]    0   10

> is.dlm(lg)
[1] TRUE
```

Suppose now that one wants to change the observation variance in the linear growth model `lg` to $V = 0.8$ and the system variance W so as to have $\sigma^2 = 0.5$. This can be easily achieved as illustrated in the following code.

R code

```
> V(lg) <- 0.8
> W(lg)[2,2] <- 0.5
> V(lg)
[1] 0.8
> W(lg)
     [,1] [,2]
[1,]    0  0.0
[2,]    0  0.5
```

In a similar way we can modify or view the other components of the model, including the mean and variance of the state at time zero, m_0 and C_0.

Let us turn now on time-varying DLMs and how they are represented in R. Most often, in a time-invariant DLM, only a few entries (possibly none) of each matrix change over time, while the remaining are constant. Therefore, instead of storing the entire matrices F_t, V_t, G_t, W_t for all values of t that one wishes to consider, we opted to store a template of each of them, and save the time-varying entries in a separate matrix. This matrix is the component `X` of a `dlm` object. Taking this approach, one also needs to know to which entry of which matrix each column of `X` corresponds. To this aim one has to specify one or more of the components `JFF`, `JV`, `JGG`, and `JW`. Let us focus on the first one, `JFF`. This should be a matrix of the same dimension of `FF`, with integer entries: if `JFF[i,j]` is `k`, a positive integer, that means that the value of `FF[i,j]` at time `s` is `X[s,k]`. If, on the other hand, `JFF[i,j]` is zero then `FF[i,j]` is taken to be constant in time. `JV`, `JGG`, and `JW` are used in the same way, for `V`, `GG`, and `W`, respectively. Consider, for example, the dynamic regression model introduced on page 43. The only time-varying element is the $(1, 2)$-entry of F_t; therefore, `X` will be a one-column matrix (although `X` is allowed to have extra, unused, columns). The following code shows how a dynamic regression model can be defined in R.

R code

```
> x <- rnorm(100) # covariates
> dlr <- dlm(FF = matrix(c(1, 0), nr = 1),
+            V = 1.3,
+            GG = diag(2),
+            W = diag(c(0.4, 0.2)),
+            m0 = rep(0, 2), C0 = 10 * diag(2),
+            JFF = matrix(c(0, 1), nr = 1),
+            X = x)
> dlr
$FF
     [,1] [,2]
[1,]    1    0

$V
     [,1]
[1,]  1.3

$GG
     [,1] [,2]
[1,]    1    0
[2,]    0    1

$W
     [,1] [,2]
[1,]  0.4  0.0
[2,]  0.0  0.2

$JFF
     [,1] [,2]
[1,]    0    1

$X
     [,1]
[1,] 0.4779
[2,] 0.5414
[3,] ...

$m0
[1] 0 0

$C0
     [,1] [,2]
[1,]   10    0
[2,]    0   10
```

Note that the dots on line 36 of the display above were produced by the *print* method function for objects of class *dlm*. If you want the entire *X* component to be printed, you need to extract it as *X(dlr)*, or use *print.default*. When modifying individual components of a *dlm* object, the user must ensure that the new components are compatible with the rest of the *dlm* object, as the replacement functions do not perform any check. This is a precise design choice, reflecting the fact that one may want to modify a *dlm* object one component at a time in such a way that, while the intermediate steps result in an invalid specification, the final result is a well-defined *dlm* object. For example, suppose one wants to use *rw* with a time series of length 30, and one wants to specify a time-varying observation variance as

$$V_t = \begin{cases} 0.75 & \text{if } t = 1, \ldots, 10, \\ 1.25 & \text{if } t = 11, \ldots, 30. \end{cases}$$

Assuming the researcher is satisfied with the constant system variance previously specified, she has to add to *rw* the two components *JV* and *X*. Adding *JV* first temporarily produces an invalid *dlm* object, which is then made into a valid one by the further addition of the *X* component. To stay on the safe side, one can make sure that a model obtained from another one by changing, adding, or removing components "by hand" is a valid *dlm* object by calling the function *dlm* on the modified model. In this case *is.dlm* is not useful, as it only looks at the class attribute of the object. The original value of *V* is still present in the new model but will never be used. For this reason *V(rw)* gives back the old value of *V*, at the same time warning the user that in *rw* the component *V* is now time-varying. The code below illustrates the previous discussion.

R code

```
> JV(rw) <- 1
> is.dlm(rw)
[1] TRUE
> dlm(rw)
Error in dlm(rw) : Component X must be provided for time-varying
models
> X(rw) <- rep(c(0.75, 1.25), c(10, 20))
> rw <- dlm(rw)
> V(rw)
     [,1]
[1,]  1.4
Warning message:
In V.dlm(rw) : Time varying V
```

2.6 Examples of nonlinear and non-Gaussian state space models

Specification and estimation of DLMs for time series analysis will be treated in Chapters 3 and 4. Here we briefly present some important classes of nonlinear and non-Gaussian state space models. Although in this book we will limit ourself to the linear Gaussian case, this section should give the reader an idea of the extensions that are possible in state space modeling when dropping those assumptions.

Exponential family state space models

Dynamic linear models can be generalized by removing the assumption of Gaussian distributions. This generalization is required for modeling discrete time series; for example, if Y_t represents the presence/absence of a characteristic in the problem under study over time, we would use a Bernoulli distribution; if Y_t are counts, we might use a Poisson model, etc. *Dynamic Generalized Linear Models* (West et al.; 1985) assume that the conditional distribution $\pi(y_t|\theta_t)$ of Y_t given θ_t is a member of the exponential family, with natural parameter $\eta_t = F_t\theta_t$. The state equation is as for Gaussian linear models, $\theta_t = G_t\theta_{t-1} + w_t$. Inference for generalized DLMs presents computational difficulties, which can, however, be solved by MCMC techniques.

Hidden Markov models

State space models in which the state θ_t is discrete are usually referred to as *hidden Markov models*. Hidden Markov models are used extensively in speech recognition (see for example Rabiner and Juang; 1993). In economics and finance, they are often used to model a time series with structural breaks. The dynamics of the series and the change points are thought as determined by a latent Markov chain (θ_t), with state space $\{\theta_1^*, \ldots, \theta_k^*\}$ and transition probabilities

$$\pi(i|j) = P(\theta_t = \theta_i^* | \theta_{t-1} = \theta_j^*).$$

Consequently, Y_t can come from a different distribution depending on the state of the chain at time t, in the sense that

$$Y_t|\{\theta_t = \theta_j^*\} \sim \pi(y_t|\theta_j^*), \qquad j = 1, \ldots, k.$$

Although state space models and hidden Markov models have evolved as separate subjects, their basic assumptions and recursive computations are closely related. MCMC methods for hidden Markov models have been developed, see for example Rydén and Titterington (1998), Kim and Nelson (1999), Cappé et al. (2005), and the references therein.

Stochastic volatility models

Stochastic volatility models are widely used in financial applications. Let Y_t be the log-return of an asset at time t (i.e., $Y_t = \log P_t/P_{t-1}$, where P_t is the asset price at time t). Under the assumption of efficient markets, the log-returns have null conditional mean: $\mathrm{E}(Y_{t+1}|y_{1:t}) = 0$. However, the conditional variance, called volatility, varies over time. There are two main classes of models for analyzing volatility of returns. The popular ARCH and GARCH models (Engle; 1982; Bollerslev; 1986) describe the volatility as a function of the past values of the returns. Stochastic volatility models, instead, consider the volatility as an exogenous random process. This leads to a state space model where the volatility is (part of) the state vector, see for example Shephard (1996). The simplest stochastic volatility model has the following form:

$$\begin{aligned} Y_t &= \exp\left\{\frac{1}{2}\theta_t\right\} w_t, & w_t &\sim \mathcal{N}(0,1), \\ \theta_t &= \eta + \phi\theta_{t-1} + v_t, & v_t &\sim \mathcal{N}(0,\sigma^2), \end{aligned}$$

that is, θ_t follows an autoregressive model of order one. These models are nonlinear and non-Gaussian, and computations are usually more demanding than for ARCH and GARCH models; however, MCMC approximations are available (Jacquier et al.; 1994). On the other hand, stochastic volatility models seem easier to generalize to the case of returns of a collection of assets, while for multivariate ARCH and GARCH models the number of parameters quickly becomes too large. Let $Y_t = (Y_{t,1}, \ldots, Y_{t,m})$ be the log-returns for m assets. A simple multivariate stochastic volatility model might assume that

$$Y_{t,i} = \exp\left(z_t + x_{t,i}\right) v_{t,i}, \qquad i = 1, \ldots, m,$$

where z_t describes a common market volatility factor and the $x_{t,i}$'s are individual volatilities. The state vector is $\theta_t = (z_t, x_{t,1}, \ldots, x_{t,m})'$, and a simple state equation might assume that the components of θ_t are independent AR(1) processes.

2.7 State estimation and forecasting

The great flexibility of state space models is one reason for their extensive application in an enormous range of applied problems. Of course, as in any statistical application, a crucial and often difficult step is a careful model specification. In many problems, the statistician and the experts together can build a state space model where the states have an intuitive meaning, and expert knowledge can be used to specify the transition probabilities in the state equation, determine the dimension of the state space, etc. However, often the model building can be a major difficulty: there might be no clear identification of physically interpretable states, or the state space representation could

be non unique, or the state space is too big and poorly identifiable, or the model is too complicated. We will discuss some issues about model building for time series analysis with DLMs in Chapter 3. Here, to get started, we consider the model as given; that is, we assume that the densities $\pi(y_t|\theta_t)$ and $\pi(\theta_t|\theta_{t-1})$ have been specified, and we present the basic recursions for estimation and forecasting. In Chapter 4, we will let these densities depend on unknown parameters ψ and discuss their estimation.

For a given state space model, the main tasks are to make inference on the unobserved states or predict future observations based on a part of the observation sequence. Estimation and forecasting are solved by computing the conditional distributions of the quantities of interest, given the available information.

To estimate the state vector we compute the conditional densities $\pi(\theta_s|y_{1:t})$. We distinguish between problems of *filtering* (when $s = t$), *state prediction* ($s > t$) and *smoothing* ($s < t$). It is worth underlining the difference between filtering and smoothing. In the filtering problem, the data are supposed to arrive sequentially in time. This is the case in many applied problems: think for example of the problem of tracking a moving object, or of financial applications where one has to estimate, day by day, the term structure of interest rates, updating the current estimates as new data are observed on the markets the following day. In these cases, we want a procedure to estimate the current value of the state vector, based on the observations up to time t ("now"), and to update our estimates and forecasts as new data become available at time $t+1$. To solve the filtering problem, we compute the conditional density $\pi(\theta_t|y_{1:t})$. In a DLM, the Kalman filter provides the formulae for updating our current inference on the state vector as new data become available, that is for passing from the filtering density $\pi(\theta_t|y_{1:t})$ to $\pi(\theta_{t+1}|y_{1:t+1})$.

The problem of smoothing, or retrospective analysis, consists instead in estimating the state sequence at times $1, \ldots, t$, given the data $y_1, \ldots, y_t$. In many applications, one has observations on a time series for a certain period, and wants to retrospectively study the behavior of the system underlying the observations. For example, in economic studies, the researcher might have the time series of consumption, or of the gross domestic product of a country, for a certain number of years, and she might be interested in retrospectively understanding the socio-economic behavior of the system. The smoothing problem is solved by computing the conditional distribution of $\theta_{1:t}$ given $y_{1:t}$. As for filtering, smoothing can be implemented as a recursive algorithm.

As a matter of fact, in time series analysis forecasting is often the main task; the state estimation is then just a step for predicting the value of future observations. For one-step-ahead forecasting, that is, predicting the next observation Y_{t+1} based on the data $y_{1:t}$, one first estimates the next value θ_{t+1} of the state vector, and then, based on this estimate, one computes the forecast for Y_{t+1}. The one-step-ahead state predictive density is $\pi(\theta_{t+1}|y_{1:t})$ and it is based on the filtering density of θ_t. From this, one obtains the one-step-ahead predictive density $\pi(y_{t+1}|y_{1:t})$.

One might be interested in looking a bit further ahead, estimating the evolution of the system, represented by the state vector θ_{t+k} for some $k \geq 1$, and making k-steps-ahead forecasts for Y_{t+k}. The state prediction is solved by computing the k-steps-ahead state predictive density $\pi(\theta_{t+k}|y_{1:t})$. Based on this density, one can compute the k-steps-ahead predictive density $\pi(y_{t+k}|y_{1:t})$ for the future observation at time $t+k$. Of course, forecasts become more and more uncertain as the time horizon $t+k$ gets farther away in the future, but note that we can anyway quantify the uncertainty through a probability density, namely the predictive density of Y_{t+1} given $y_{1:t}$. We will show how to compute the predictive densities in a recursive fashion. In particular, the conditional mean $\mathrm{E}(Y_{t+1}|y_{1:t})$ provides an optimal one-step-ahead point forecast of the value of Y_{t+1}, minimizing the conditional expected square prediction error. As a function of k, $\mathrm{E}(Y_{t+k}|y_{1:t})$ is usually called the *forecast function*.

2.7.1 Filtering

We first describe the recursive steps needed to compute the filtering densities $\pi(\theta_t|y_{1:t})$ in general state space models. Even if we will not make extensive use of these formulae, it is useful to look now at the general recursions to better understand the role of the conditional independence assumptions that have been introduced. Then we move to the DLM case, for which the filtering problem is solved by the well-known Kalman filter.

One of the advantages of state space models is that, due to the Markovian structure of the state dynamics (A.1) and the assumptions on the conditional independence for the observables (A.2), the filtered and predictive densities can be computed using a recursive algorithm. As we have seen in the introductory example of Section 2.2, starting from $\theta_0 \sim \pi(\theta_0)$ one can recursively compute, for $t = 1, 2, \ldots$:

(i) the one-step-ahead predictive distribution for θ_t given $y_{1:t-1}$, based on the filtering density $\pi(\theta_{t-1}|y_{1:t-1})$ and the conditional distribution of θ_t given θ_{t-1} specified by the model;

(ii) the one-step-ahead predictive distribution for the next observation;

(iii) the filtering distribution $\pi(\theta_t|y_{1:t})$, using the Bayes rule with $\pi(\theta_t|y_{1:t-1})$ as the prior distribution and likelihood $\pi(y_t|\theta_t)$.

The following proposition contains a formal presentation of the filtering recursions for a general state space model.

Proposition 2.1 (Filtering recursions). *For a general state space model defined by* (A.1)-(A.2) *(p.40), the following statements hold.*

(i) The one-step-ahead predictive density for the states can be computed from the filtered density $\pi(\theta_{t-1}|y_{1:t-1})$ *according to*

$$\pi(\theta_t|y_{1:t-1}) = \int \pi(\theta_t|\theta_{t-1})\pi(\theta_{t-1}|y_{1:t-1})\,\mathrm{d}\theta_{t-1}. \tag{2.7a}$$

(ii) The one-step-ahead predictive density for the observations can be computed from the predictive density for the states as

$$\pi(y_t|y_{1:t-1}) = \int \pi(y_t|\theta_t)\pi(\theta_t|y_{1:t-1})\,\mathrm{d}\theta_t. \tag{2.7b}$$

(iii) The filtering density can be computed from the above densities as

$$\pi(\theta_t|y_{1:t}) = \frac{\pi(y_t|\theta_t)\pi(\theta_t|y_{1:t-1})}{\pi(y_t|y_{1:t-1})}. \tag{2.7c}$$

Proof. The proof relies heavily on the conditional independence properties of the model, which can be deduced from the graph in Figure 2.6.

To prove (i), note that θ_t is conditionally independent of $Y_{1:t-1}$, given θ_{t-1}. Therefore,

$$\begin{aligned}\pi(\theta_t|y_{1:t-1}) &= \int \pi(\theta_{t-1}, \theta_t|y_{1:t-1})\,\mathrm{d}\theta_{t-1} \\ &= \int \pi(\theta_t|\theta_{t-1}, y_{1:t-1})\pi(\theta_{t-1}|y_{1:t-1})\,\mathrm{d}\theta_{t-1} \\ &= \int \pi(\theta_t|\theta_{t-1})\pi(\theta_{t-1}|y_{1:t-1})\,\mathrm{d}\theta_{t-1}.\end{aligned}$$

To prove (ii), note that Y_t is conditionally independent of $Y_{1:t-1}$ given θ_t. Therefore,

$$\begin{aligned}\pi(y_t|y_{1:t-1}) &= \int \pi(y_t, \theta_t|y_{1:t-1})\,\mathrm{d}\theta_t \\ &= \int \pi(y_t|\theta_t, y_{1:t-1})\pi(\theta_t|y_{1:t-1})\,\mathrm{d}\theta_t \\ &= \int \pi(y_t|\theta_t)\pi(\theta_t|y_{1:t-1})\,\mathrm{d}\theta_t.\end{aligned}$$

Part (iii) follows from Bayes' rule and the conditional independence of Y_t and $Y_{1:t-1}$ given θ_t:

$$\pi(\theta_t|y_{1:t}) = \frac{\pi(\theta_t|y_{1:t-1})\,\pi(y_t|\theta_t, y_{1:t-1})}{\pi(y_t|y_{1:t-1})} = \frac{\pi(\theta_t|y_{1:t-1})\,\pi(y_t|\theta_t)}{\pi(y_t|y_{1:t-1})}.$$

□

From the one-step-ahead predictive distribution provided by the previous proposition, k-steps ahead predictive distributions for the state and for the observation can be computed recursively according to the formulae

$$\pi(\theta_{t+k}|y_{1:t}) = \int \pi(\theta_{t+k}|\theta_{t+k-1})\,\pi(\theta_{t+k-1}|y_{1:t})\,\mathrm{d}\theta_{t+k-1}$$

and

$$\pi(y_{t+k}|y_{1:t}) = \int \pi(y_{t+k}|\theta_{t+k})\,\pi(\theta_{t+k}|y_{1:t})\,\mathrm{d}\theta_{t+k}.$$

Incidentally, these recursions also show that $\pi(\theta_t|y_{1:t})$ summarizes the information contained in the past observations $y_{1:t}$, which is sufficient for predicting Y_{t+k}, for any $k > 0$.

2.7.2 Kalman filter for dynamic linear models

The previous results solve in principle the filtering and the forecasting problems; however, in general the actual computation of the relevant conditional distributions is not at all an easy task. DLMs are one important case where the general recursions simplify considerably. In this case, using standard results about the multivariate Gaussian distribution, it is easily proved that the random vector $(\theta_0, \theta_1, \ldots, \theta_t, Y_1, \ldots, Y_t)$ has a Gaussian distribution for any $t \geq 1$. It follows that the marginal and conditional distributions are also Gaussian. Since all the relevant distributions are Gaussian, they are completely determined by their means and variances. The solution of the filtering problem for DLMs is given by the celebrated Kalman filter.

Proposition 2.2 (Kalman filter). *Consider the DLM specified by* (2.4) *(p.41). Let*

$$\theta_{t-1}|y_{1:t-1} \sim \mathcal{N}(m_{t-1}, C_{t-1}).$$

Then the following statements hold.

(i) The one-step-ahead predictive distribution of θ_t given $y_{1:t-1}$ is Gaussian, with parameters

$$\begin{aligned} a_t &= \mathrm{E}(\theta_t|y_{1:t-1}) = G_t m_{t-1}, \\ R_t &= \mathrm{Var}(\theta_t|y_{1:t-1}) = G_t C_{t-1} G_t' + W_t. \end{aligned} \tag{2.8a}$$

(ii) The one-step-ahead predictive distribution of Y_t given $y_{1:t-1}$ is Gaussian, with parameters

$$\begin{aligned} f_t &= \mathrm{E}(Y_t|y_{1:t-1}) = F_t a_t, \\ Q_t &= \mathrm{Var}(Y_t|y_{1:t-1}) = F_t R_t F_t' + V_t. \end{aligned} \tag{2.8b}$$

(iii) The filtering distribution of θ_t given $y_{1:t}$ is Gaussian, with parameters

$$\begin{aligned} m_t &= \mathrm{E}(\theta_t|y_{1:t}) = a_t + R_t F_t' Q_t^{-1} e_t, \\ C_t &= \mathrm{Var}(\theta_t|y_{1:t}) = R_t - R_t F_t' Q_t^{-1} F_t R_t, \end{aligned} \tag{2.8c}$$

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

Proof. The random vector $(\theta_0, \theta_1, \ldots, \theta_t, Y_1, \ldots, Y_t)$ has joint distribution given by (2.3), where the marginal and conditional distributions involved are Gaussian. From standard results on the multivariate Normal distribution (see Appendix A), it follows that the joint distribution of $(\theta_0, \theta_1, \ldots, \theta_t, Y_1, \ldots, Y_t)$ is Gaussian, for any $t \geq 1$. Consequently, the distribution of any subvector is also Gaussian, as is the conditional distribution of some components given some other components. Therefore the predictive distributions and the filtering distributions are Gaussian, and it suffices to compute their means and variances.

To prove (i), let $\theta_t|y_{1:t-1} \sim \mathcal{N}(a_t, R_t)$. Using (2.4c), a_t and R_t can be obtained as follows:

$$\begin{aligned} a_t &= \mathrm{E}(\theta_t|y_{1:t-1}) = \mathrm{E}(\mathrm{E}(\theta_t|\theta_{t-1}, y_{1:t-1})|y_{1:t-1}) \\ &= \mathrm{E}(G_t\theta_{t-1}|y_{1:t}) = G_t m_{t-1} \end{aligned}$$

and

$$\begin{aligned} R_t &= \mathrm{Var}(\theta_t|y_{1:t-1}) \\ &= \mathrm{E}(\mathrm{Var}(\theta_t|\theta_{t-1}, y_{1:t-1})|y_{1:t-1}) + \mathrm{Var}(\mathrm{E}(\theta_t|\theta_{t-1}, y_{1:t-1})|y_{1:t-1}) \\ &= W_t + G_t C_{t-1} G_t'. \end{aligned}$$

To prove (ii), let $Y_t|y_{1:t-1} \sim \mathcal{N}(f_t, Q_t)$. Using (2.4b), f_t and Q_t can be obtained as follows:

$$f_t = \mathrm{E}(Y_t|y_{1:t-1}) = \mathrm{E}(\mathrm{E}(Y_t|\theta_t, y_{1:t-1})|y_{1:t-1}) = \mathrm{E}(F_t\theta_t|y_{1:t-1}) = F_t a_t$$

and

$$\begin{aligned} Q_t &= \mathrm{Var}(Y_t|y_{1:t-1}) \\ &= \mathrm{E}(\mathrm{Var}(Y_t|\theta_t, y_{1:t-1})|y_{1:t-1}) + \mathrm{Var}(\mathrm{E}(Y_t|\theta_t, y_{1:t-1})|y_{1:t-1}) \\ &= V_t + F_t R_t F_t'. \end{aligned}$$

Let us prove (iii) next. We can adapt Proposition 2.1(iii) to the present special case. There, we showed that, in order to compute the filtering distribution at time t, we have to apply the Bayes formula to combine the prior $\pi(\theta_t|y_{1:t-1})$ and the likelihood $\pi(y_t|\theta_t)$. In the DLM case all the distributions are Gaussian and the problem is the same as the Bayesian inference problem for the linear model

$$Y_t = F_t\theta_t + v_t, \qquad v_t \sim \mathcal{N}(0, V_t),$$

with a regression parameter θ_t following a conjugate Gaussian prior $\mathcal{N}(a_t, R_t)$. (Here V_t is known.) From the results in Section 1.5 we have that

$$\theta_t|y_{1:t} \sim \mathcal{N}(m_t, C_t),$$

where, by (1.10),

$$m_t = a_t + R_t F_t' Q_t^{-1}(Y_t - F_t a_t)$$

and, by (1.9),

$$C_t = R_t - R_t F_t' Q_t^{-1} F_t R_t.$$

□

The Kalman filter allows us to compute the predictive and filtering distributions recursively, starting from $\theta_0 \sim \mathcal{N}(m_0, C_0)$, then computing $\pi(\theta_1|y_1)$, and proceeding recursively as new data become available.

The conditional distribution of $\theta_t|y_{1:t}$ solves the filtering problem. However, in many cases one is interested in a point estimate. As we have discussed in Section 1.3, the Bayesian point estimate of θ_t given the information $y_{1:t}$, with respect to the quadratic loss function $L(\theta_t, a) = (\theta_t - a)'H(\theta_t - a)$, is the conditional expected value $m_t = \mathrm{E}(\theta_t|y_{1:t})$. This is the optimal estimate since it minimizes the conditional expected loss $\mathrm{E}((\theta_t - a)'H(\theta_t - a)|y_{1:t-1})$ with respect to a. If $H = I_p$, the minimum expected loss is the conditional variance matrix $\mathrm{Var}(\theta_t|y_{1:t})$.

As we noted in the introductory example in Section 2.2, the expression of m_t has the intuitive estimation-correction form "filter mean equals the prediction mean a_t plus a correction depending on how much the new observation differs from its prediction". The weight of the correction term is given by the gain matrix

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

Thus, the weight of current data point Y_t depends on the observation variance V_t (through Q_t) and on $R_t = \mathrm{Var}(\theta_t|y_{1:t-1}) = G_t C_{t-1} G_t' + W_t$.

As an example, consider the local level model (2.5). The Kalman filter gives

$$\begin{aligned}
\mu_t|y_{1:t-1} &\sim \mathcal{N}(m_{t-1}, R_t = C_{t-1} + W), \\
Y_t|y_{1:t-1} &\sim \mathcal{N}(f_t = m_{t-1}, Q_t = R_t + V), \\
\mu_t|y_{1:t} &\sim \mathcal{N}(m_t = m_{t-1} + K_t e_t, C_t = K_t V),
\end{aligned}$$

where $K_t = R_t/Q_t$ and $e_t = Y_t - f_t$. It is worth underlining that the behavior of the process (Y_t) is greatly influenced by the ratio between the two error variances, $r = W/V$, which is usually called the *signal-to-noise* ratio (a good exercise for seeing this is to simulate some trajectories of (Y_t), for different values of V and W). This is reflected in the structure of the estimation and forecasting mechanism. Note that $m_t = K_t y_t + (1 - K_t)m_{t-1}$, a weighted average of y_t and m_{t-1}. The weight $K_t = R_t/Q_t = (C_{t-1}+W)/(C_{t-1}+W+V)$ of the current observation y_t is also called *adaptive coefficient*, and it satisfies

$0 < K_t < 1$. For any given C_0, if the signal-to-noise r is small, K_t is small and y_t receives little weight. If, at the opposite extreme, $V = 0$, we have $K_t = 1$ and $m_t = y_t$, that is, the one-step-ahead forecast is given by the most recent data point. A practical illustration of how different relative magnitudes of W and V affect the mean of the filtered distribution and the one-step-ahead forecasts is given on pages 57 and 67.

The evaluation of the posterior variances C_t (and consequently also of R_t and Q_t) using the iterative updating formulae contained in Proposition 2.2, as simple as it may appear, suffers from numerical instability that may lead to nonsymmetric and even negative definite calculated variance matrices. Alternative, stabler, algorithms have been developed to overcome this issue. Apparently, the most widely used, at least in the Statistics literature, is the square root filter, which provides formulae for the sequential update of a square root[3] of C_t. References for the square root filter are Morf and Kailath (1975) and Anderson and Moore (1979, Ch. 6)

In our work we have found that occasionally, in particular when the observational noise has a small variance, even the square root filter incurs numerical stability problems, leading to negative definite calculated variances. A more robust algorithm is the one based on sequentially updating the singular value decomposition[4] (SVD) of C_t. The details of the algorithm can be found in Oshman and Bar-Itzhack (1986) and Wang et al. (1992). Strictly speaking, the SVD-based filter can be seen as a square root filter: in fact if $A = UD^2U'$ is the SVD of a variance matrix, then DU' is a square root of A. However, compared to the standard square root filtering algorithms, the SVD-based one is typically more stable (see the references for further discussion).

The Kalman filter is performed in package dlm by the function `dlmFilter`. The arguments are the data, `y`, in the form of a numerical vector, matrix, or time series, and the model, `mod`, an object of class `dlm` or a list that can be coerced to a `dlm` object. For the reasons of numerical stability mentioned above, the calculations are performed on the SVD of the variance matrices C_t and R_t. Accordingly, the output provides, for each t, an orthogonal matrix $U_{C,t}$ and a vector $D_{C,t}$ such that $C_t = U_{C,t}\,\mathrm{diag}(D_{C,t}^2)\,U_{C,t}'$, and similarly for R_t.

The output produced by `dlmFilter`, a list with class attribute "`dlmFiltered`," includes, in addition to the original data and the model (components `y` and `mod`), the means of the predictive and filtered distributions (components `a` and `m`) and the SVD of the variances of the predictive and filtered distributions (components `U.R`, `D.R`, `U.C`, and `D.C`). For convenience, the component `f` of the output list provides the user with one-step-ahead forecasts. The component `U.C` is a list of matrices, the $U_{C,t}$ above, while `D.C`

[3] We define a square root of variance matrix A to be any square matrix N such that $A = N'N$.

[4] See Appendix B for a definition.

is a matrix containing, stored by row, the vectors $D_{C,t}$ of the SVD of the C_t's. Similarly for `U.R` and `D.R`. The utility function `dlmSvd2var` can be used to reconstruct the variances from their SVD. In the display below we use a random walk plus noise model with the Nile data (Figure 2.3). The variances $V = 15100$ and $W = 1468$ are maximum likelihood estimates. To set up the model we use, instead of `dlm`, the more convenient `dlmModPoly`, which will be discussed in Chapter 3.

R code

```
> NilePoly <- dlmModPoly(order = 1, dV = 15100, dW = 1468)
> unlist(NilePoly)
      m0       C0       FF        V       GG        W
       0 10000000        1    15100        1     1468
> NileFilt <- dlmFilter(Nile, NilePoly)
> str(NileFilt, 1)
List of 9
 $ y  : Time-Series [1:100] from 1871 to 1970: 1120 1160 ...
 $ mod:List of 10
  ..- attr(*, "class")= chr "dlm"
 $ m  : Time-Series [1:101] from 1870 to 1970:    0 1118 ...
 $ U.C:List of 101
 $ D.C: num [1:101, 1] 3162  123 ...
 $ a  : Time-Series [1:100] from 1871 to 1970:    0 1118 ...
 $ U.R:List of 100
 $ D.R: num [1:100, 1] 3163  129 ...
 $ f  : Time-Series [1:100] from 1871 to 1970:    0 1118 ...
 - attr(*, "class")= chr "dlmFiltered"
> n <- length(Nile)
> attach(NileFilt)
> dlmSvd2var(U.C[[n + 1]], D.C[n + 1, ])
         [,1]
[1,] 4031.035
```

The last number in the display is the variance of the filtering distribution of the 100-th state vector. Note that m_0 and C_0 are included in the output, which is the reason why `U.C` has one element more than `U.R`, and `m` and `U.D` one row more than `a` and `D.R`.

As we already noted on page 55, the relative magnitude of W and V is an important factor that enters the gain matrix, which, in turn, determines how sensitive the state prior-to-posterior updating is to unexpected observations. To illustrate the role of the signal-to-noise ratio W/V in the local level model, we use two models, with a significantly different signal-to-noise ratio, to estimate the true level of the Nile River. The filtered values for the two models can then be compared.

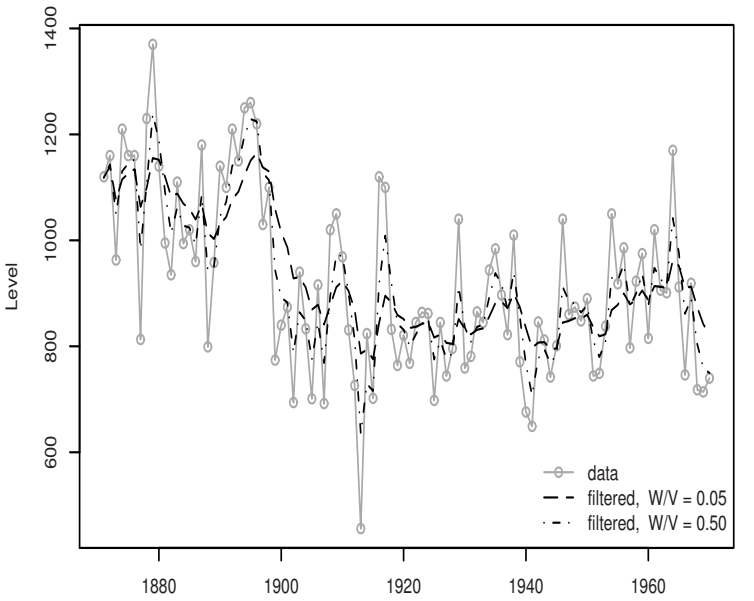

Fig. 2.7. Filtered values of the Nile River level for two different signal-to-noise ratios

R code

```
> plot(Nile, type='o', col = c("darkgrey"),
+      xlab = "", ylab = "Level")
> mod1 <- dlmModPoly(order = 1, dV = 15100, dW = 755)
> NileFilt1 <- dlmFilter(Nile, mod1)
> lines(dropFirst(NileFilt1$m), lty = "longdash")
> mod2 <- dlmModPoly(order = 1, dV = 15100, dW = 7550)
> NileFilt2 <- dlmFilter(Nile, mod2)
> lines(dropFirst(NileFilt2$m), lty = "dotdash")
> leg <- c("data", paste("filtered,  W/V =",
+                        format(c(W(mod1) / V(mod1),
+                                 W(mod2) / V(mod2)))))
> legend("bottomright", legend = leg,
+        col=c("darkgrey", "black", "black"),
+        lty = c("solid", "longdash", "dotdash"),
+        pch = c(1, NA, NA), bty = "n")
```

Figure 2.7 displays the filtered levels resulting from the two models. It is appearent that for model 2, which has a signal-to-noise ratio ten times larger than model 1, the filtered values tend to follow more closely the data.

2.7.3 Filtering with missing observations

In applied data analysis it is not infrequent to have to deal with a time series containing one or more missing observations. In multivariate time series, missing observations can be of two different types: totally missing and partially missing observations. The first type is the one that occurs when the observation vector at some time t is not available. In the second case only some of the components of the observation vector are not available. This may happen for example when considering a daily time series of closing prices of a set of stock indices in several countries: if day t is a holiday in country A but not country B, then for that day the closing price for the index of country A is not even defined, i.e., it is missing, while the closing price of the index of country B is normally recorded. Clearly, for a univariate time series an observation is either missing or not missing. Luckily, the structure of state space models is such that missing observations can be easily accomodated in the filtering recursion. We will first consider the case of totally missing observations. Following R convention, we will consider a missing observation as one having the special value `NA`. If the observation at time t is missing, then $y_t =$ `NA` and y_t does not carry any information, so that

$$\pi(\theta_t|y_{1:t}) = \pi(\theta_t|y_{1:t-1}). \tag{2.9}$$

This means that in this case the filtering distribution at time t is just the one-step-ahead predictive distribution at time $t-1$. Operationally, in the filtering recursion (Proposition 2.1) one has to replace (2.7c) with (2.9). In particular, for a DLM, since $\theta_t|y_{1:t-1} \sim \mathcal{N}(a_t, R_t)$, all one needs to do is to set $m_t = a_t$ and $C_t = R_t$. From time $t+1$ the standard filtering recursion resumes as usual, provided y_{t+1} is nonmissing. Note that formally, in a DLM, having $y_t =$ `NA` is the same as setting $F_t = 0$ or $V_t = \infty$. In the first case y_t is not linked to θ_t in any way, in the second the observation is so noisy as to be totally unreliable in providing meaningful information about θ_t. Either way leads to a gain matrix $K_t = 0$ and consequently $m_t = a_t$ and $C_t = R_t$.

Consider now a state space model with m-dimensional observation vectors, $m > 1$. Suppose that some, but not all, of the components of y_t are missing. The vector y_t in this case provides some information about θ_t, but all this information is contained in the nonmissing components. Let $\tilde{y}_t$ be the vector comprising only the nonmissing components of y_t. Then in the filtering recursion (2.7), $\pi(y_t|\theta_t)$ should be replaced by $\pi(\tilde{y}_t|\theta_t)$ and $\pi(y_t|y_{1:t-1})$ by $\pi(\tilde{y}_t|y_{1:t-1})$. Let us take a closer look at the DLM case. Denote by $\tilde{m}_t$ the dimension of $\tilde{y}_t$ and consider the $\tilde{m}_t$ by m matrix M_t obtained by removing from an m by m identity matrix the rows corresponding to the missing components of y_t, so that $\tilde{y}_t = M_t y_t$. The fact that we observed $\tilde{y}_t$ instead of y_t implies that in updating the prior $\mathcal{N}(a_t, R_t)$ to the posterior $\mathcal{N}(m_t, C_t)$, the correct observation equation to consider is

$$\tilde{y}_t = \tilde{F}_t\theta_t + \tilde{v}_t \qquad \tilde{v}_t \sim \mathcal{N}(0, \tilde{V}_t),$$

with $\tilde{F}_t = M_t F_t$ and $\tilde{V}_t = M_t V_t M_t'$. In practice, this implies that when computing the Kalman filter (Proposition 2.2), one has simply to replace F_t and V_t with $\tilde{F}_t$ and $\tilde{V}_t$ in (2.8b) and (2.8c).

The function `dlmFilter` accepts data containing `NA`'s, computing the moments of the correct filtering distributions.

2.7.4 Smoothing

One of the attractive features of state space models is that estimation and forecasting can be applied sequentially, as new data become available. However, in time series analysis one often has observations on Y_t for a certain period, $t = 1, \ldots, T$, and wants to retrospectively reconstruct the behavior of the system, to study the socio-economic construct or physical phenomenon underlying the observations. In this case, one can use a backward-recursive algorithm to compute the conditional distributions of θ_t given $y_{1:T}$, for any $t < T$, starting from the filtering distribution $\pi(\theta_T|y_{1:T})$ and estimating backward all the states' history. The result for general state space models is contained in the following proposition.

Proposition 2.3 (Smoothing recursion). *For a general state space model defined by* (A.1)-(A.2) *(p. 40), the following statements hold.*

(i) Conditional on $y_{1:T}$, the state sequence $(\theta_0, \ldots, \theta_T)$ has backward transition probabilities given by

$$\pi(\theta_t|\theta_{t+1}, y_{1:T}) = \frac{\pi(\theta_{t+1}|\theta_t)\pi(\theta_t|y_{1:t})}{\pi(\theta_{t+1}|y_{1:t})}.$$

(ii) The smoothing distributions of θ_t given $y_{1:T}$ can be computed according to the following backward recursion in t, starting from $\pi(\theta_T|y_{1:T})$:

$$\pi(\theta_t|y_{1:T}) = \pi(\theta_t|y_{1:t}) \int \frac{\pi(\theta_{t+1}|\theta_t)}{\pi(\theta_{t+1}|y_{1:t})} \pi(\theta_{t+1}|y_{1:T}) \, \mathrm{d}\theta_{t+1}.$$

Proof. To prove (i), note that θ_t and $Y_{t+1:T}$ are conditionally independent given θ_{t+1}; moreover, θ_{t+1} and $Y_{1:T}$ are conditionally independent given θ_t. (Use the DAG in Figure 2.6 to show this.) Using the Bayes formula, one has

$$\begin{aligned}
\pi(\theta_t|\theta_{t+1}, y_{1:T}) &= \pi(\theta_t|\theta_{t+1}, y_{1:t}) \\
&= \frac{\pi(\theta_t|y_{1:t})\pi(\theta_{t+1}|\theta_t, y_{1:t})}{\pi(\theta_{t+1}|y_{1:t})} \\
&= \frac{\pi(\theta_t|y_{1:t})\pi(\theta_{t+1}|\theta_t)}{\pi(\theta_{t+1}|y_{1:t})}.
\end{aligned}$$

To prove (ii), marginalize $\pi(\theta_t, \theta_{t+1}|y_{1:T})$ with respect to θ_{t+1}:

$$\begin{aligned}
\pi(\theta_t|y_{1:T}) &= \int \pi(\theta_t, \theta_{t+1}|y_{1:T})\,\mathrm{d}\theta_{t+1} \\
&= \int \pi(\theta_{t+1}|y_{1:T})\pi(\theta_t|\theta_{t+1}, y_{1:T})\,\mathrm{d}\theta_{t+1} \\
&= \int \pi(\theta_{t+1}|y_{1:T})\frac{\pi(\theta_{t+1}|\theta_t)\pi(\theta_t|y_{1:t})}{\pi(\theta_{t+1}|y_{1:t})}\,\mathrm{d}\theta_{t+1} \\
&= \pi(\theta_t|y_{1:t}) \int \pi(\theta_{t+1}|\theta_t)\frac{\pi(\theta_{t+1}|y_{1:T})}{\pi(\theta_{t+1}|y_{1:t})}\,\mathrm{d}\theta_{t+1}.
\end{aligned}$$

□

For a DLM, the smoothing recursion can be stated more explicitely in terms of means and variances of the smoothing distributions.

Proposition 2.4 (Kalman smoother). *For a DLM defined by* (2.4)*, if* $\theta_{t+1}|y_{1:T} \sim \mathcal{N}(s_{t+1}, S_{t+1})$*, then* $\theta_t|y_{1:T} \sim \mathcal{N}(s_t, S_t)$*, where*

$$\begin{aligned}
s_t &= m_t + C_t G'_{t+1} R^{-1}_{t+1}(s_{t+1} - a_{t+1}) \\
S_t &= C_t - C_t G'_{t+1} R^{-1}_{t+1}(R_{t+1} - S_{t+1}) R^{-1}_{t+1} G_{t+1} C_t.
\end{aligned}$$

Proof. It follows from the properties of the multivariate Gaussian distribution that the conditional distribution of θ_t given $y_{1:T}$ is Gaussian; thus, it suffices to compute its mean and variance. We have

$$s_t = \mathrm{E}(\theta_t|y_{1:T}) = \mathrm{E}(\mathrm{E}(\theta_t|\theta_{t+1}, y_{1:T})|y_{1:T})$$

and

$$S_t = \mathrm{Var}(\theta_t|y_{1:T}) = \mathrm{Var}(\mathrm{E}(\theta_t|\theta_{t+1}, y_{1:T})|y_{1:T}) + \mathrm{E}(\mathrm{Var}(\theta_t|\theta_{t+1}, y_{1:T})|y_{1:T}).$$

As shown in the proof of Proposition 2.3, θ_t and $Y_{t+1:T}$ are conditionally independent given θ_{t+1}, so that $\pi(\theta_t|\theta_{t+1}, y_{1:T}) = \pi(\theta_t|\theta_{t+1}, y_{1:t})$. We can use the Bayes formula to compute this distribution. Note that the likelihood $\pi(\theta_{t+1}|\theta_t, y_{1:t}) = \pi(\theta_{t+1}|\theta_t)$ is expressed by the state equation (2.4c), that is,

$$\theta_{t+1}|\theta_t \sim \mathcal{N}(G_{t+1}\theta_t, W_{t+1}).$$

The prior is $\pi(\theta_t|y_{1:t})$, which is $\mathcal{N}(m_t, C_t)$. Using (1.10) and (1.9), we find that

$$\begin{aligned}
\mathrm{E}(\theta_t|\theta_{t+1}, y_{1:t}) &= m_t + C_t G'_{t+1}(G_{t+1} C_t G'_{t+1} + W_{t+1})^{-1}(\theta_{t+1} - G_{t+1} m_t) \\
&= m_t + C_t G'_{t+1} R^{-1}_{t+1}(\theta_{t+1} - a_{t+1}) \\
\mathrm{Var}(\theta_t|\theta_{t+1}, y_{1:t}) &= C_t - C_t G'_{t+1} R^{-1}_{t+1} G_{t+1} C_t,
\end{aligned}$$

from which it follows that

$$
\begin{aligned}
s_t &= \mathrm{E}(\mathrm{E}(\theta_t|\theta_{t+1}, y_{1:t})|y_{1:T}) = m_t + C_t G'_{t+1} R^{-1}_{t+1}(s_{t+1} - a_{t+1}) \\
S_t &= \mathrm{Var}(\mathrm{E}(\theta_t|\theta_{t+1}, y_{1:t})|y_{1:T}) + \mathrm{E}(\mathrm{Var}(\theta_t|\theta_{t+1}, y_{1:t})|y_{1:T}) \\
&= C_t - C_t G'_{t+1} R^{-1}_{t+1} G_{t+1} C_t + C_t G'_{t+1} R^{-1}_{t+1} S_{t+1} R^{-1}_{t+1} G_{t+1} C_t \\
&= C_t - C_t G'_{t+1} R^{-1}_{t+1}(R_{t+1} - S_{t+1}) R^{-1}_{t+1} G_{t+1} C_t,
\end{aligned}
$$

being $\mathrm{E}(\theta_{t+1}|y_{1:T}) = s_{t+1}$ and $\mathrm{Var}(\theta_{t+1}|y_{1:T}) = S_{t+1}$ by assumption. □

The Kalman smoother allows us to compute the distributions of $\theta_t|y_{1:T}$, starting from $t = T-1$, in which case $\theta_T|y_{1:T} \sim \mathcal{N}(s_T = m_T, S_T = C_T)$, and then proceeding backward to compute the distributions of $\theta_t|y_{1:T}$ for $t = T-2$, $t = T-3$, etc. Note that the smoothing recursion depends on the data only through the filtering and one-step-ahead predictive moments obtained using the Kalman filter. Therefore, if a time series contains missing observations, this should be accounted for when performing the filtering recursion, but no additional adjustment is required in the smoothing recursion.

About the numerical stability of the smoothing algorithm, the same caveat holds as for the filtering recursions. The formulae of Proposition 2.4 are subject to numerical instability, and more robust square root and SVD-based smoothers are available (see Zhang and Li; 1996). The function `dlmSmooth` performs the calculations in R, starting from an object of class `dlmFiltered`, typically the output produced by `dlmFilter`. Alternatively, the user can provide the data and the model, in which case `dlmFilter` is called internally. `dlmSmooth` returns a list with components `s`, the means of the smoothing distributions, and `U.S`, `D.S`, their variances, given in terms of their SVD. The following display illustrates the use of `dlmSmooth` on the Nile data.

R code

```
> NileSmooth <- dlmSmooth(NileFilt)
> str(NileSmooth, 1)
List of 3
 $ s  : Time-Series [1:101] from 1870 to 1970: 1111 1111 ...
 $ U.S:List of 101
 $ D.S: num [1:101, 1] 74.1 63.5 ...
> attach(NileSmooth)
> drop(dlmSvd2var(U.S[[n + 1]], D.S[n + 1,]))
[1] 4031.035
> drop(dlmSvd2var(U.C[[n + 1]], D.C[n + 1,]))
[1] 4031.035
> drop(dlmSvd2var(U.S[[n / 2 + 1]], D.S[n / 2 + 1,]))
[1] 2325.985
> drop(dlmSvd2var(U.C[[n / 2 + 1]], D.C[n / 2 + 1,]))
[1] 4031.035
```

In the display above, `n` is 100, the number of observations, so, accounting for time $t = 0$, `n/2 + 1` corresponds to time 50. Observe that the smoothing and filtering variances are equal at the end of the observation period – time T (lines 9 and 11); but the smoothing variance at time 50 (line 13) is much smaller than the filtering variance at the same time (line 15). This is due to the fact that in the filtering distribution at time 50 we are conditioning on the first fifty observations only, while in the smoothing distribution the conditioning is with respect to all the one hundred observations available. Note also, incidentally, that the filtering variance at time 50 is the same as the filtering variance at time 100. It is the case for many constant models that the filtering variance, C_t, tends to a limiting value as t increases. In very informal terms, the explanation of this behavior is the following. In DLMs the learning process about the state of the system occurs in a dynamic environment, that is, one in which the state changes as one gains information about it. Therefore, in the updating of the filtering variance from time $t-1$ to time t, there are two conflicting processes going on: on one hand, the observation y_t brings new information about θ_{t-1}, but in the meanwhile the state of the system has changed to θ_t, with the additional uncertainty carried by w_t. This additional uncertainty is represented by the variance $W_t = W$, say. If C_0 is large – typically one does not have much confidence in his prior guess about the state – then the first observations are very informative and their impact on C_t is much more important than that of the dynamics of the state, resulting in an overall decrease of the filtering variance. However, as more data are collected, the impact of one additional observation on the information about the state of the system decreases and, at some point, it will be exactly balanced by the loss of information represented by the additional variance W. From that time on, C_t will essentially stay constant.

The display below illustrates how the variance of the smoothing distribution can be used to construct pointwise probability intervals for the state components – only one in this example. The plot produced by the code below is shown in Figure 2.8

R code

```
> hwid <- qnorm(0.025, lower = FALSE) *
+     sqrt(unlist(dlmSvd2var(U.S, D.S)))
> smooth <- cbind(s, as.vector(s) + hwid %o% c(-1, 1))
> plot(dropFirst(smooth), plot.type = "s", type = "l",
+      lty = c(1, 5, 5), ylab = "Level", xlab = "",
+      ylim = range(Nile))
> lines(Nile, type = "o", col = "darkgrey")
> legend("bottomleft", col = c("darkgrey", rep("black", 2)),
+        lty = c(1, 1, 5), pch = c(1, NA, NA), bty = "n",
+        legend = c("data", "smoothed level",
+        "95% probability limits"))
```

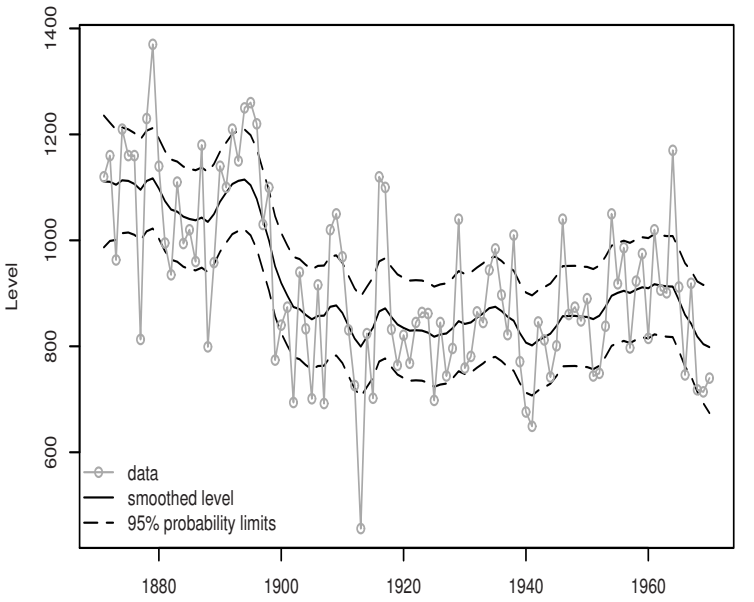

Fig. 2.8. Smoothed values of the Nile River level, with 95% probability limits

As an additional example, we consider a quarterly time series of consumer expenditure on durable goods in the UK, in 1958£, from the first quarter of 1957 to the last quarter of 1967[5]. A DLM including a local level plus a quarterly seasonal component was fitted to the data. This kind of model will be discussed in Chapter 3; here we focus on filtering and smoothing. In the model the state vector is 4-dimensional. Two of its components have a particularly relevant interpretation: the first one can be thought of as the true, deseasonalized, level of the series; the second is a dynamic seasonal component. According to the model, the observations are obtained by adding observational noise to the sum of the first and second component of the state vector, as can be deduced from the matrix *FF*. Figure 2.9 shows the data, together with the deseasonalized filtered and smoothed level. These values are just the first components of the series of filtered and smoothed state vectors. In addition to the level of the series, one can also estimate the seasonal component, which is just the second component of the smoothed or filtered state vector. Figure 2.10 shows the smoothed seasonal component. It is worth stressing that the model is dynamic, hence the seasonal component is allowed to vary as time goes by. This is clearly the case in the present example: from an alternating of positive and negative values at the beginning of the observation period, the series moves to a two-positive two-negative pattern in the second half. The display below shows how filtered and smoothed values have been obtained in R, as

[5] Source: Hyndman (n.d.).

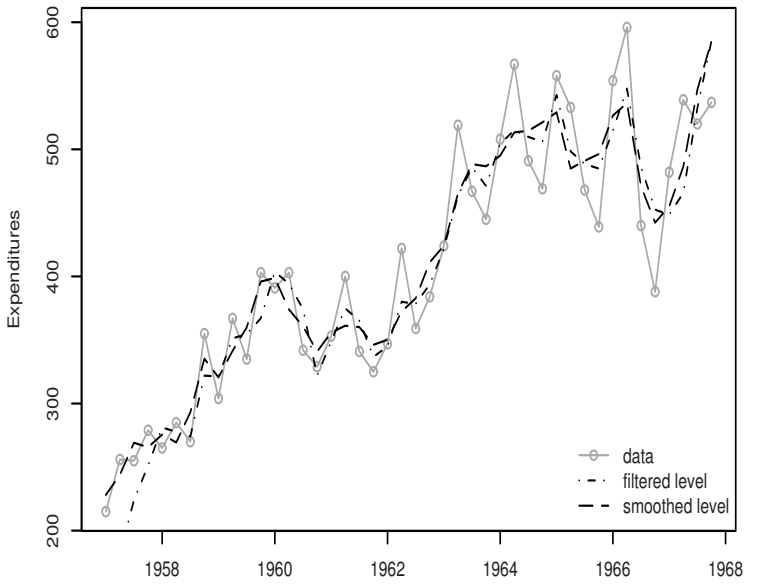

Fig. 2.9. Quarterly expenditure on durable goods, with filtered and smoothed level

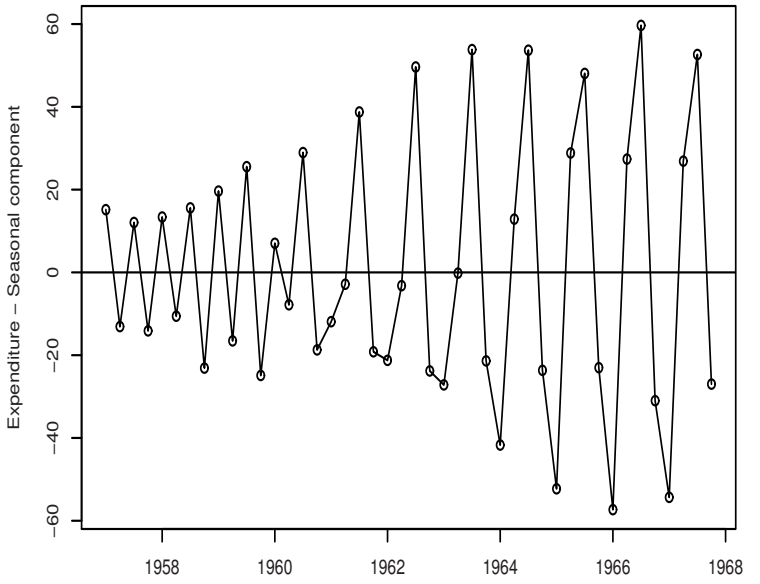

Fig. 2.10. Quarterly expenditure on durable goods: smoothed seasonal component

well as how the plots were created. The function `bdiag` is a utility function in package dlm that creates a block diagonal matrix from the individual blocks, or from a list containing the blocks.

R code

```
> expd <- ts(read.table("Datasets/qconsum.dat", skip = 4,
+                       colClasses = "numeric")[, 1],
+            start = c(1957, 1), frequency = 4)
> expd.dlm <- dlm(m0 = rep(0,4), C0 = 1e8 * diag(4),
+                 FF = matrix(c(1, 1, 0, 0), nr = 1),
+                 V = 1e-3,
+                 GG = bdiag(matrix(1),
+                 matrix(c(-1, -1, -1, 1, 0, 0, 0, 1, 0),
+                        nr = 3, byrow = TRUE)),
+                 W = diag(c(771.35, 86.48, 0, 0), nr = 4))
> plot(expd, xlab = "", ylab = "Expenditures", type = 'o',
+      col = "darkgrey")
> ### Filter
> expdFilt <- dlmFilter(expd, expd.dlm)
> lines(dropFirst(expdFilt$m[, 1]), lty = "dotdash")
> ### Smooth
> expdSmooth <- dlmSmooth(expdFilt)
> lines(dropFirst(expdSmooth$s[,1]), lty = "longdash")
> legend("bottomright", col = c("darkgrey", rep("black", 2)),
+        lty = c("solid", "dotdash", "longdash"),
+        pch = c(1, NA, NA), bty = "n",
+        legend = c("data", "filtered level", "smoothed level"))
> ### Seasonal component
> plot(dropFirst(expdSmooth$s[, 3]), type = 'o', xlab = "",
+      ylab = "Expenditure - Seasonal component")
> abline(h = 0)
```

2.8 Forecasting

With $y_{1:t}$ at hand, one can be interested in forecasting future values of the observations, Y_{t+k}, or of the state vectors, θ_{t+k}. For state space models, the recursive form of the computations makes it natural to compute the one-step-ahead forecasts and to update them sequentially as new data become available. This is clearly of interest in applied problems where the data do arrive sequentially, such as in day-by-day forecasting stock prices, or in tracking a moving target; but one-step-ahead forecasts are often also computed "in-sample", as a tool for checking the performance of the model.

For a DLM, the one-step-ahead predictive distributions, for states and observations, are obtained as a byproduct of the Kalman filter, as presented in Proposition 2.2.

In R, the one-step-ahead forecasts $f_t = \mathrm{E}(Y_t|y_{1:t-1})$ are provided in the output of the function `dlmFilter`. Since for each t the one-step-ahead forecast of the observation, f_t, is a linear function of the filtering mean m_{t-1}, the magnitude of the gain matrix plays the same role in determining how sensitive f_t is to an unexpected observation y_{t-1} as it did for m_{t-1}. In the case of the random walk plus noise model this is particularly evident, since in this case $f_t = m_{t-1}$. Figure 2.11, produced with the code below, contains the one-step-ahead forecasts obtained from the local level models with the different signal-to-noise ratios defined in the display on page 57.

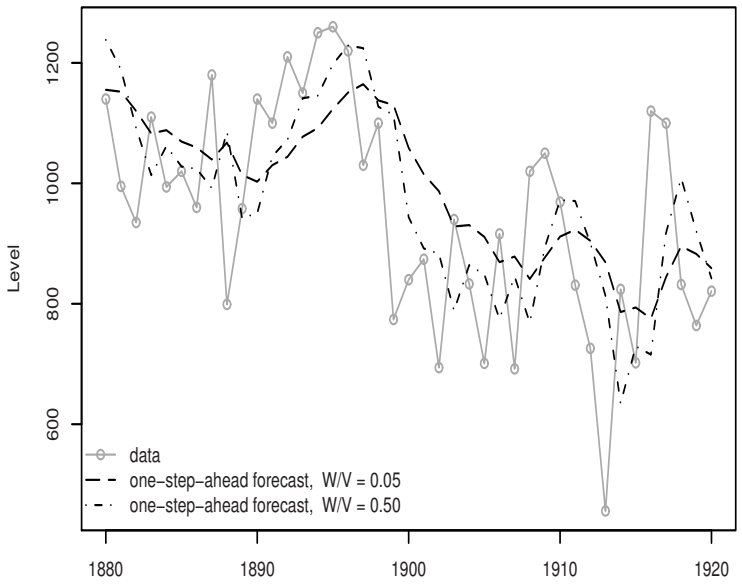

Fig. 2.11. One-step-ahead forecasts for the Nile level using different signal-to-noise ratios

R code

```
> a <- window(cbind(Nile, NileFilt1$f, NileFilt2$f),
+             start = 1880, end = 1920)
> plot(a[, 1], type = 'o', col = "darkgrey",
+      xlab = "", ylab = "Level")
> lines(a[, 2], lty = "longdash")
> lines(a[, 3], lty = "dotdash")
> leg <- c("data", paste("one-step-ahead forecast,  W/V =",
```

```
+                         format(c(W(mod1) / V(mod1),
+                                  W(mod2) / V(mod2)))))
> legend("bottomleft", legend = leg,
+        col = c("darkgrey", "black", "black"),
+        lty = c("solid", "longdash", "dotdash"),
+        pch = c(1, NA, NA), bty = "n")
```

To further elaborate on the same example, we note that the signal-to-noise ratio need not be constant in time. The construction of the Ashwan dam in 1898, for instance, can be expected to produce a major change in the level of the Nile River. A simple way to incorporate this expected level shift in the model is to assume a system evolution variance W_t larger than usual (12 times larger in the display below) for that year and the following one. In this way the estimated true level of the river will quickly recognize the new regime, leading in turn to more accurate one-step-ahead forecasts. The code below illustrates this idea.

R code

```
> mod0 <- dlmModPoly(order = 1, dV = 15100, dW = 1468)
> X <- ts(matrix(mod0$W, nc = 1, nr = length(Nile)),
+         start = start(Nile))
> window(X, 1898, 1899) <- 12 * mod0$W
> modDam <- mod0
> modDam$X <- X
> modDam$JW <- matrix(1, 1, 1)
> damFilt <- dlmFilter(Nile, modDam)
> mod0Filt <- dlmFilter(Nile, mod0)
> a <- window(cbind(Nile, mod0Filt$f, damFilt$f),
+             start = 1880, end = 1920)
> plot(a[, 1], type = 'o', col = "darkgrey",
+      xlab="", ylab="Level")
> lines(a[, 2], lty = "longdash")
> lines(a[, 3], lty = "dotdash")
> abline(v=1898, lty=2)
> leg <- c("data", paste("one-step-ahead forecast -",
+                        c("mod0", "modDam")))
> legend("bottomleft", legend = leg,
+        col = c("darkgrey", "black", "black"),
+        lty = c("solid", "longdash", "dotdash"),
+        pch = c(1, NA, NA), bty = "n")
```

Note (see Figure 2.12) how, using the modified model *modDam*, the forecast for the level of the river in 1900 is already around what the new river level actually is, while for the other model this happens only around 1907. On a

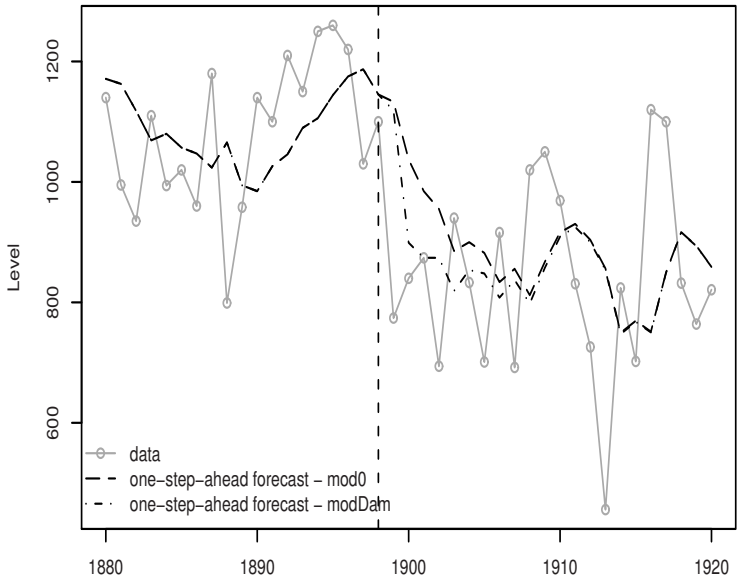

Fig. 2.12. One-step-ahead forecasts of Nile River level with and without change point

more technical note, it is instructive to note how we define the time varying model `modDam` by adding the components `X` and `JW` (lines 6 and 7) to the constant model `mod0`.

In many applications one is interested in looking a bit further in the future, and provide possible scenarios of the behavior of the series for k steps ahead. We present here the recursive formulae for the means and variances of the conditional distributions of states and observations at a future time $t + k$, given the data up to time t. In view of the Markovian nature of the model, the filtering distribution at time t acts like an initial distribution for the future evolution of the model. To be more precise, 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 state space model having conditional distributions $\pi(\theta_{t+k}|\theta_{t+k-1})$ and $\pi(y_{t+k}|\theta_{t+k})$, and initial distribution $\pi(\theta_t|y_{1:t})$. The information about the future provided by the data is all contained in this distribution. For a DLM, in particular, since the data are only used to obtain m_t, the mean of $\pi(\theta_t|y_{1:t})$, it follows that m_t provides a summary of the data that is sufficient for predictive purposes. You can have a further intuition about that by looking at the DAG representing the dependence structure among the variables (Figure 2.6). We see that the path from $Y_{1:t}$ to Y_{t+k} is as in Figure 2.13, showing that the data $Y_{1:t}$ provide information about θ_t, which in turn gives information about the future state evolution up to θ_{t+k} and consequently on Y_{t+k}. Of course, as k

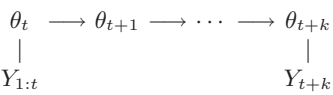

Fig. 2.13. Flow of information from $Y_{1:t}$ to Y_{t+k}

gets larger, more uncertainty enters in the system, and the forecasts will be less and less precise.

Proposition 2.5 provides recursive formulae to compute the forecast distributions for states and observations for a general state space model.

Proposition 2.5 (Forecasting recursion). *For a general state space model defined by* (A.1)-(A.2) *(p.40), the following statements hold for any* $k > 0$.

(i) The k-steps-ahead forecast distribution of the state is

$$\pi(\theta_{t+k}|y_{1:t}) = \int \pi(\theta_{t+k}|\theta_{t+k-1})\pi(\theta_{t+k-1}|y_{1:t})\,\mathrm{d}\theta_{t+k-1}.$$

(ii) The k-steps-ahead forecast distribution of the observation is

$$\pi(y_{t+k}|y_{1:t}) = \int \pi(y_{t+k}|\theta_{t+k})\pi(\theta_{t+k}|y_{1:t})\,\mathrm{d}\theta_{t+k}.$$

Proof. Using the conditional independence properties of the model, we have:

$$\begin{aligned}\pi(\theta_{t+k}|y_{1:t}) &= \int \pi(\theta_{t+k}, \theta_{t+k-1}|y_{1:t})\,\mathrm{d}\theta_{t+k-1} \\ &= \int \pi(\theta_{t+k}|\theta_{t+k-1}, y_{1:t})\pi(\theta_{t+k-1}|y_{1:t})\,\mathrm{d}\theta_{t+k-1} \\ &= \int \pi(\theta_{t+k}|\theta_{t+k-1})\pi(\theta_{t+k-1}|y_{1:t})\,\mathrm{d}\theta_{t+k-1},\end{aligned}$$

which is (i). The proof of (ii) is again based on the conditional independence properties of the models. We have that

$$\begin{aligned}\pi(y_{t+k}|y_{1:t}) &= \int \pi(y_{t+k}, \theta_{t+k}|y_{1:t})\,\mathrm{d}\theta_{t+k} \\ &= \int \pi(y_{t+k}|\theta_{t+k}, y_{1:t})\pi(\theta_{t+k}|y_{1:t})\,\mathrm{d}\theta_{t+k} \\ &= \int \pi(y_{t+k}|\theta_{t+k})\pi(\theta_{t+k}|y_{1:t})\,\mathrm{d}\theta_{t+k},\end{aligned}$$

which is (ii). □

For DLMs, Proposition 2.5 takes a more specific form, since all the integrals can be computed explicitly. However, as is the case for filtering and smoothing,

since all the forecast distributions are Gaussian, it is enough to compute their means and variances. Proposition 2.6 provides recursive formulae to compute them. We need to introduce some notation first. For $k \geq 1$, define

$$a_t(k) = \mathrm{E}(\theta_{t+k}|y_{1:t}), \tag{2.10a}$$
$$R_t(k) = \mathrm{Var}(\theta_{t+k}|y_{1:t}), \tag{2.10b}$$
$$f_t(k) = \mathrm{E}(Y_{t+k}|y_{1:t}), \tag{2.10c}$$
$$Q_t(k) = \mathrm{Var}(Y_{t+k}|y_{1:t}). \tag{2.10d}$$

Proposition 2.6. *For a DLM defined by* (2.4)*, let* $a_t(0) = m_t$ *and* $R_t(0) = C_t$*. Then, for* $k \geq 1$*, the following statements hold.*

(i) The distribution of θ_{t+k} *given* $y_{1:t}$ *is Gaussian, with*

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

(ii) The distribution of Y_{t+k} *given* $y_{1:t}$ *is Gaussian, with*

$$f_t(k) = F_{t+k}a_t(k),$$
$$Q_t(k) = F_{t+k}R_t(k)F'_{t+k} + V_t.$$

Proof. As we have already noted, all conditional distributions are Gaussian. Therefore, we only need to prove the formulae giving the means and variances. We proceed by induction. The result holds for $k = 1$ in view of Proposition 2.2. For $k > 1$,

$$\begin{aligned}
a_t(k) &= \mathrm{E}(\theta_{t+k}|y_{1:t}) = \mathrm{E}(\mathrm{E}(\theta_{t+k}|y_{1:t}, \theta_{t+k-1})|y_{1:t}) \\
&= \mathrm{E}(G_{t+k}\theta_{t+k-1}|y_{1:t}) = G_{t+k}a_{t,k-1}, \\
R_t(k) &= \mathrm{Var}(\theta_{t+k}|y_{1:t}) = \mathrm{Var}(\mathrm{E}(\theta_{t+k}|y_{1:t}, \theta_{t+k-1})|y_{1:t}) \\
&\quad + \mathrm{E}(\mathrm{Var}(\theta_{t+k}|y_{1:t}, \theta_{t+k-1})|y_{1:t}) \\
&= G_{t+k}R_{t,k-1}G'_{t+k} + W_{t+k}, \\
f_t(k) &= \mathrm{E}(Y_{t+k}|y_{1:t}) = \mathrm{E}(\mathrm{E}(Y_{t+k}|y_{1:t}, \theta_{t+k})|y_{1:t}) \\
&= \mathrm{E}(F_{t+k}\theta_{t+k}|y_{1:t}) = F_{t+k}a_t(k), \\
Q_t(k) &= \mathrm{Var}(Y_{t+k}|y_{1:t}) = \mathrm{Var}(\mathrm{E}(Y_{t+k}|y_{1:t}, \theta_{t+k})|y_{1:t}) \\
&\quad + \mathrm{E}(\mathrm{Var}(Y_{t+k}|y_{1:t}, \theta_{t+k})|y_{1:t}) \\
&= F_{t+k}R_t(k)F'_{t+k} + V_{t+k},
\end{aligned}$$

□

Note that the data only enter the predictive distributions through the mean of the filtering distribution at the time the last observation was taken. The function `dlmForecast` computes the means and variances of the predictive distributions of the observations and the states. Optionally, it can be used to draw a sample of future states and observations. The principal argument of `dlmForecast` is an object of class `dlmFiltered`. Alternatively, it can be a object of class `dlm` (or a list with the appropriate named components), where the components `m0` and `C0` are interpreted as being the mean and variance of the state vector at the end of the observation period, given the data, i.e., they are the mean and variance of the last (most recent) filtering distribution. The code below shows how to obtain predicted values of the expenditure series (Figure 2.9, p.65) for the three years following the last observation, together with a sample from their distribution. Figure 2.14 shows the forecasted and simulated future values of the series.

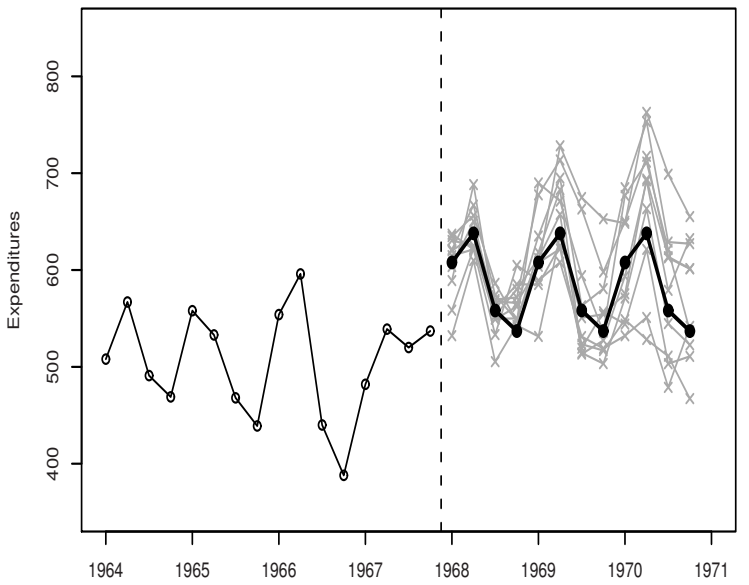

Fig. 2.14. Quarterly expenditure on durable goods: forecasts

R code

```
> set.seed(1)
> expdFore <- dlmForecast(expdFilt, nAhead = 12, sampleNew = 10)
> plot(window(expd, start = c(1964,1)), type = 'o',
+      xlim = c(1964,1971), ylim = c(350, 850),
+      xlab = "", ylab = "Expenditures")
> names(expdFore)
```

```
[1] "a"          "R"          "f"          "Q"
[5] "newStates" "newObs"
> attach(expdFore)
> invisible(lapply(newObs, function(x)
+                  lines(x, col = "darkgrey",
+                        type = 'o', pch = 4)))
> lines(f, type = 'o', lwd = 2, pch = 16)
> abline(v = mean(c(time(f)[1], time(expd)[length(expd)])),
+        lty = "dashed")
> detach()
```

2.9 The innovation process and model checking

As we have seen, for DLMs we can compute the one-step-ahead forecasts $f_t = \mathrm{E}(Y_y|y_{1:t-1})$, and we defined the forecast error as

$$e_t = Y_t - \mathrm{E}(Y_t|y_{1:t-1}) = Y_t - f_t.$$

The forecast errors can alternatively be written in terms of the one-step-ahead estimation errors as follows:

$$\begin{aligned} e_t &= Y_t - F_t a_t = F_t\theta_t + v_t - F_t a_t \\ &= F_t(\theta_t - a_t) + v_t. \end{aligned}$$

The sequence $(e_t)_{t\geq 1}$ of forecast errors enjoys some interesting properties, the most important of which are collected in the following proposition.

Proposition 2.7. *Let $(e_t)_{t\geq 1}$ be the sequence of forecast errors of a DLM. Then the following properties hold.*

(i) The expected value of e_t is zero.
(ii) The random vector e_t is uncorrelated with any function of $Y_1, \ldots, Y_{t-1}$.
(iii) For any $s < t$, e_t and Y_s are uncorrelated.
(iv) For any $s < t$, e_t and e_s are uncorrelated.
(v) e_t is a linear function of $Y_1, \ldots, Y_t$.
(vi) $(e_t)_{t\geq 1}$ is a Gaussian process.

Proof. (i) By taking iterated expected values,

$$\mathrm{E}(e_t) = \mathrm{E}(\mathrm{E}(Y_t - f_t|Y_{1:t-1})) = 0.$$

(ii) Let $Z = g(Y_1, \ldots, Y_{t-1})$. Then

$$\begin{aligned} \mathrm{Cov}(e_t, Z) &= \mathrm{E}(e_t Z) = \mathrm{E}(\mathrm{E}(e_t Z|Y_{1:t-1})) \\ &= \mathrm{E}(\mathrm{E}(e_t|Y_{1:t-1})Z) = 0. \end{aligned}$$

(iii) If the observations are univariate, this follows from (ii), taking $Z = Y_s$. Otherwise, apply (ii) to each component of Y_s.
(iv) This follows again from (ii), taking $Z = e_s$ if the observations are univariate. Otherwise, apply (ii) componentwise.
(v) Since $Y_1, \ldots, Y_t$ have a joint Gaussian distribution, $f_t = \mathrm{E}(Y_t|Y_{1:t-1})$ is a linear function of $Y_1, \ldots, Y_{t-1}$. Hence, e_t is a linear function of $Y_1, \ldots, Y_t$.
(vi) For any t, in view of (v), $(e_1, \ldots, e_t)$ is a linear transformation of $(Y_1, \ldots, Y_t)$, which has a joint Normal distribution. It follows that also $(e_1, \ldots, e_t)$ has a joint Normal distribution. Hence, since all finite-dimensional distributions are Gaussian, the process $(e_t)_{t\geq 1}$ is Gaussian. □

The forecast errors e_t are also called *innovations*. The representation $Y_t = f_t + e_t$ justifies this terminology, since one can think of Y_t as the sum of a component, f_t, which is predictable from past observations, and another component, e_t, which is independent of the past and therefore contains the really new information provided by the observation Y_t.

Sometimes it may be convenient to work with the so-called *innovation form* of a DLM. This is obtained by choosing as new state variables the vectors $a_t = \mathrm{E}(\theta_t|y_{1:t-1})$. Then the observation equation is derived from $e_t = Y_t - f_t = Y_t - F_t a_t$:

$$Y_t = F_t a_t + e_t \tag{2.11a}$$

and, being $a_t = G_t m_{t-1}$, where m_{t-1} is given by the Kalman filter:

$$a_t = G_t m_{t-1} = G_t a_{t-1} + G_t R_{t-1} F'_{t-1} Q^{-1}_{t-1} e_t;$$

so, the new state equation is

$$a_t = G_t a_{t-1} + w^*_t, \tag{2.11b}$$

with $w^*_t = G_t R_{t-1} F'_{t-1} Q^{-1}_{t-1} e_t$. The system (2.11) is the innovation form of the DLM. Note that, in this form, the observation errors and the system errors are no longer independent, that is, the dynamics of the states is no longer independent of the observations. The main advantage is that in the innovation form all components of the state vector on which we cannot obtain any information from the observations are automatically removed. It is thus in some sense a minimal model.

When the observations are univariate, the sequence of standardized innovations, defined by $\tilde{e}_t = e_t/\sqrt{Q_t}$, is a Gaussian white noise, i.e., a sequence of independent identically distributed zero-mean normal random variables. This property can be exploited to check model assumptions: if the model is correct, the sequence $\tilde{e}_1, \ldots, \tilde{e}_t$ computed from the data should look like a sample of size t from a standard normal distribution. Many statistical tests, several of them readily available in R, can be carried out on the standardized innovations. Such tests fall into two broad categories: those aimed at checking

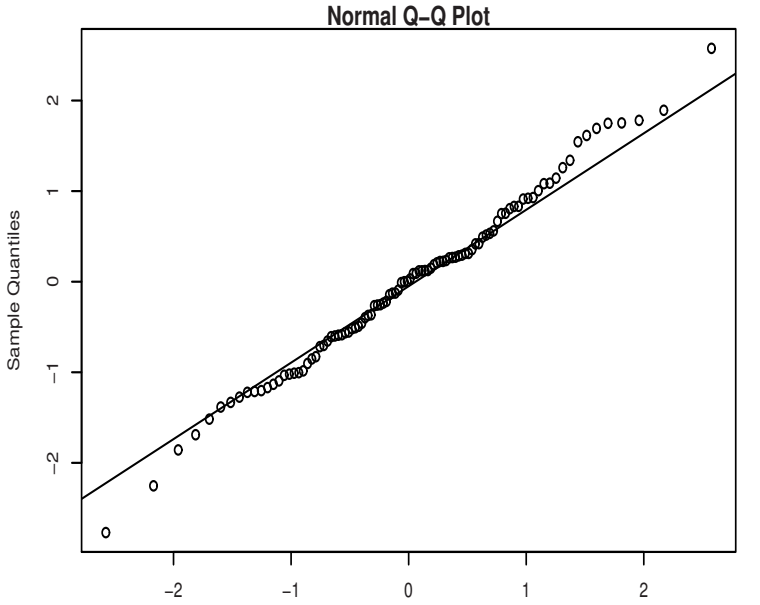

Fig. 2.15. Nile River: QQ-plot of standardized innovations

if the distribution of the $\tilde{e}_t$'s is standard normal, and those aimed at checking whether the $\tilde{e}_t$'s are uncorrelated. We will illustrate the use of some of these tests in Chapter 3. However, most of the time we take a more informal approach to model checking, based on the subjective assessment of selected diagnostic plots. The most useful are, in the opinion of the authors, a QQ-plot and a plot of the empirical autocorrelation function of the standardized innovations. The former can be used to assess normality, while the latter reveals departures from uncorrelatedness. A time series plot of the standardized innovations may prove useful in detecting outliers, change points, and other unexpected patterns.

In R, the standardized innovations can be extracted from an object of class `dlmFiltered` using the function `residuals`. Package dlm also provides a method function for `tsdiag` for objects of class `dlmFiltered`. This function, modeled after `tsdiag.Arima`, exctracts the standardized innovations and plots them, together with their empirical autocorrelation function and the p-values for Ljung-Box test statistics up to a specific lag (the default is 10). For the DLM `modDam` (p.68) used to model Nile River level data, Figure 2.9 shows a QQ-plot of the standardized innovations, while Figure 2.9 displays the plots produced by a call to `tsdiag`. The two figures were obtained with the code below.

R code

```
> qqnorm(residuals(damFilt, sd = FALSE))
> qqline(residuals(damFilt, sd = FALSE))
> tsdiag(damFilt)
```

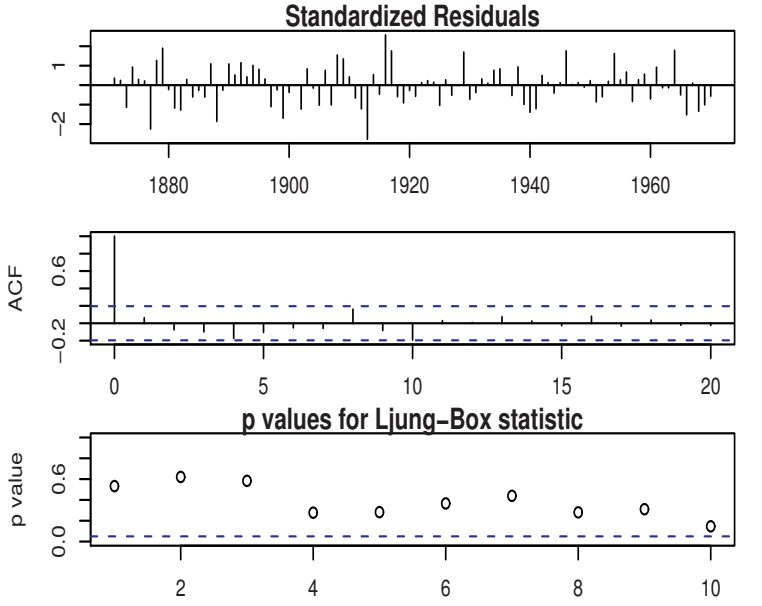

Fig. 2.16. Nile River: diagnostic plots produced by *tsdiag*

For multivariate observations we usually apply the same univariate graphical diagnostic tools component-wise to the innovation sequence. A further step would be to adopt the vector standardization $\tilde{e}_t = B_t e_t$, where B_t is a $p \times p$ matrix such that $B_t Q_t B_t' = I$. This makes the components of $\tilde{e}_t$ independent and identically distributed according to a standard normal distribution. Using this standardization, the sequence $\tilde{e}_{1,1}, \tilde{e}_{1,2} \ldots, \tilde{e}_{1,p}, \ldots, \tilde{e}_{t,p}$ should look like a sample of size tp from a univariate standard normal distribution. This approach, however, is not very popular in applied work and we will not employ it in this book.

2.10 Controllability and observability of time-invariant DLMs

In the engineering literature, DLMs are widely used in *control* problems; indeed, optimal control was one main objective in Kalman's contributions. See, for example, Kalman (1961), Kalman et al. (1963), and Kalman (1968). Here, the interest is in the state of the system, θ_t, which one wants to regulate by means of so-called *control variables* u_t. Problems of this nature are clearly of great relevance in many applied fields, besides engineering; for example, in economics, the monetary authority might want to regulate the *state* of macroeconomic variables, for example the inflation and the unemployment rates, by means of monetary instruments u_t under its control. A DLM including control variables will be referred to as a *controlled* DLM and will be written in the form

$$\begin{aligned} y_t &= F_t\theta_t + v_t, \\ \theta_t &= G_t\theta_{t-1} + H_t u_t + w_t \end{aligned}$$

where u_t is an r-dimensional vector of control variables, i.e., variables whose value can be regulated by the researcher, in order to obtain a desired level of the state θ_t, and H_t is a known $p \times r$ matrix; the usual assumptions are made for the stochastic errors v_t and w_t. Control problems have been first studied for deterministic systems (i.e., systems with no stochastic terms v_t and w_t); in most applications, however, a further difficulty is the presence of stochastic errors in the relationship between θ_t and y_t and in the state evolution. A comprehensive treatment of control problems is beyond the scope of this book; in this section we will only briefly recall some basic notions, limiting our attention to the case of a *time-invariant* controlled DLM, i.e., a controlled DLM where the matrices F_t, G_t, V_t, W_t, and H_t, are constant over time:

$$\begin{aligned} y_t &= F\theta_t + v_t, \\ \theta_t &= G\theta_{t-1} + Hu_t + w_t. \end{aligned}$$

Good references are Anderson and Moore (1979), Harvey (1989), Maybeck (1979), and Jazwinski (1970).

At a basic level, the goal of a control problem is to drive the state of a DLM from the initial value θ_0 to a target value θ^* in a finite time T, setting appropriately the control variables $u_1, \ldots, u_T$. Two issues immediately arise: the first is that the states of a DLM are not observed directly, so, in particular, θ_0 is not known exactly in general; the second is that, even if θ_0 were known, there is no guarantee that one can drive the system to the desired state θ^*. Let us take a closer look at the second problem first, considering the ideal case of a deterministic system equation, i.e., one in which $w_t = 0$ for every t. The system equation reduces in this case to

$$\theta_t = G\theta_{t-1} + Hu_t \tag{2.12}$$

Starting at θ_0 at time zero and applying (2.12) repeatedly, we have

$$\begin{aligned}
\theta_1 &= G\theta_0 + Hu_1, \\
\theta_2 &= G\theta_1 + Hu_2 = G^2\theta_0 + GHu_1 + Hu_2, \\
&\vdots \\
\theta_T &= G^T\theta_0 + \sum_{j=0}^{T-1} G^j H u_{T-j}.
\end{aligned}$$

Therefore, if we want the system to be in state θ^* at time T, we need to solve the equation $\theta_T = \theta^*$ with respect to the control variables $u_1, \ldots, u_T$. More explicitely, let $\mathcal{C}_T$ be the $p \times rT$ matrix defined by

$$\mathcal{C}_T = \left[G^{T-1}H \mid \cdots \mid GH \mid H \right].$$

Stacking the vectors $u_1, \ldots, u_T$, we obtain the following system of linear equations:

$$\mathcal{C}_T \begin{bmatrix} u_1 \\ \vdots \\ u_T \end{bmatrix} = \theta^* - G^T\theta_0. \tag{2.13}$$

If (2.13) has to have a solution for any arbitrary θ^* and θ_0, then $\mathcal{C}_T$ must be of rank p, and vice versa. In other words, a DLM with system equation (2.12) can be driven from an arbitrary initial state θ_0 to another arbitrary state θ^* in a finite time T through an appropriate choice of the control variables $u_1, \ldots, u_T$ if and only if $\mathcal{C}_T$ has full rank p. Moreover, using elementary linear algebra arguments, it can be shown that if $\mathcal{C}_T$ has rank p for some T, then $\mathcal{C}_p$ has rank p. For this reason the matrix $\mathcal{C}_p$ is called the *controllability* matrix of the DLM, and we will denote it $\mathcal{C}$, without subscript. A DLM is said to be *controllable* if its controllability matrix $\mathcal{C}$ has full rank p.

The definition of controllability given above can be transported to a standard time-invariant DLM with system equation

$$\theta_t = G\theta_{t-1} + w_t, \qquad w_t \sim \mathcal{N}(0, W). \tag{2.14}$$

After all, the only difference between (2.12) and (2.14) is that the control term Hu_t in the former is replaced by the system noise w_t in the latter. To carry the analogy one step further, we can write the noise as $w_t = B\eta_t$, where η_t is an r-dimensional random vector having independent standard normal components, and B is a full-rank $p \times r$ matrix. Note that $W = BB'$. When $r < p$, the rank of W is r and the possible values of w_t lie on an r-dimensional linear subspace of $\mathbb{R}^p$ – in this sense we can think of w_t as being essentially r-dimensional, and we can represent it via η_t. We define the *controllability* matrix of a DLM with system equation (2.14) to be

$$\mathcal{C} = \left[G^{p-1}B \mid \cdots \mid GB \mid B\right],$$

and the DLM to be *controllable* if its controllability matrix has full rank p.

Note that the decomposition $W = BB'$ does not identify B uniquely, since for any orthogonal matrix O of order r, the matrix $\tilde{B} = BO$ provides the representation $W = \tilde{B}\tilde{B}'$. However, the particular choice of B does not matter. In fact, one can also avoid computing the decomposition $W = BB'$ altogether. Note that the linear subspace of $\mathbb{R}^p$ spanned by the columns of B is the same as the one spanned by the columns of W. Hence, $\mathcal{C}$ and the matrix

$$\mathcal{C}^W = \left[G^{p-1}W \mid \cdots \mid GW \mid W\right]$$

have the same rank, although $\mathcal{C}^W$ has p^2 columns instead of rp.

As an example, consider an integrated random walk of order 2 (cf. p. 100), which is a DLM whose system equation is defined by the two matrices

$$\begin{aligned} G &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \\ W &= \begin{bmatrix} 0 & 0 \\ 0 & \sigma_\beta^2 \end{bmatrix}, \end{aligned} \tag{2.15}$$

with $\sigma_\beta^2 > 0$. Here $p = 2$ and

$$\mathcal{C}^W = [GW \mid W] = \begin{bmatrix} 0 & \sigma_\beta^2 & 0 & 0 \\ 0 & \sigma_\beta^2 & 0 & \sigma_\beta^2 \end{bmatrix}.$$

Since $\mathcal{C}^W$ has rank 2, the DLM is controllable.

Clearly for a standard DLM, since the noise (w_t) cannot be set by the observer, the notion of controllability has a different interpretation than in the case of a controlled DLM. A controllable DLM with system equation (2.14) is one for which, by effect of the noise sequence (w_t), the state vector θ_t can reach any point in $\mathbb{R}^p$, no matter what the initial value of the state vector is. In other words, there are no inaccessible regions for the state of the system. In the general theory of Markov chains, this property is called *irreducibility* of the Markov chain (θ_t).

Let us turn now to the first issue raised at the beginning of the discussion, related to the observability of the states. Clearly, if the system or observation noises are nonzero, there is little hope of determining exactly the value of θ_t based solely on the observation y_t, or even on a finite number T of observations $y_{t:t+T-1}$. Therefore we will focus on the idealized situation of a time-invariant DLM in which we can set $V = 0$ and $W = 0$. The observation and system equation reduce to

$$\begin{aligned} y_t &= F\theta_t, \\ \theta_t &= G\theta_{t-1}. \end{aligned} \tag{2.16}$$

Applying repeatedly (2.16) we obtain

$$\begin{aligned} y_t &= F\theta_t, \\ y_{t+1} &= F\theta_{t+1} = FG\theta_t, \\ &\vdots \\ y_{t+T-1} &= FG^{T-1}\theta_t. \end{aligned}$$

Defining the matrix

$$\mathcal{O}_T = \begin{bmatrix} F \\ FG \\ \vdots \\ FG^{T-1} \end{bmatrix}$$

and stacking the observation vectors, the system above can be written as

$$\begin{bmatrix} y_t \\ \vdots \\ y_{t+T-1} \end{bmatrix} = \mathcal{O}_T \theta_t.$$

Therefore, the state θ_t can be determined from the data $y_{t:t+T-1}$ if and only if the previous system of linear equations has a unique solution (in θ_t). This is the case if and only if the $mT \times p$ matrix $\mathcal{O}_T$ has rank p. Also in this case, it can be shown that, if $\mathcal{O}_T$ has rank p for some T, then $\mathcal{O}_p$ has rank p. The matrix $\mathcal{O}_p$ is called the *observability* matrix of the given DLM and it will be denoted by $\mathcal{O}$, without subscript. A time-invariant DLM is said to be observable if its observability matrix $\mathcal{O}$ has full rank p.

Consider again, for example, the 2nd-order integrated random walk whose system equation is defined by (2.15). The observation matrix for this DLM is

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

Therefore the observability matrix is

$$\mathcal{O} = \begin{bmatrix} F \\ FG \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

This matrix has rank 2, hence the DLM is observable.

In the next section we will link controllability and observability to the asymptotic behavior of the Kalman filter.

2.11 Filter stability

Consider a time-invariant DLM. As shown in Section 2.7, for any t we have that

$$\theta_t|y_{1:t-1} \sim \mathcal{N}_p(a_t, R_t),$$

where a_t and R_t are given by Proposition 2.2. Note that, if the matrices F, G, V and W are known, then the covariance matrix $R_t = \mathrm{Var}(\theta_t|y_{1:t-1})$ does not depend on the data, but only on the initial conditions m_0, C_0, on the system matrices F and G, and on the covariance matrices V and W. In this sense, the asymptotic behavior of R_t is intrinsic to the model, and it can be studied on the basis of the properties of the matrices F, G, V and W. In particular, one can study whether the conditional variance of θ_t given $y_{1:t-1}$ or $y_{1:t}$, tends to become stable for t increasing to infinity, forgetting the initial conditions m_0 and C_0.

Note that, by substituting the expressions of $m_{t-1}, C_{t-1}, f_{t-1}$ in the formulae given by (i) of Proposition 2.2 for a_t and R_t, the latter can be written in the form

$$a_t = (G - A_{t-1}F)a_{t-1} + A_{t-1}y_{t-1},$$

where we denoted by $A_{t-1} = GK_{t-1} = GR_{t-1}F'[V + FR_{t-1}F']^{-1}$ the gain matrix for the state forecast, and

$$R_t = GR_{t-1}G' - A_{t-1}FR_{t-1}G' + W. \tag{2.17}$$

The previous expression, when seen as an equation in the unknown matrix R_t, is called Riccati equation. Note that in (2.17), $A_t = A_t(R_{t-1})$. If there exists a constant positive semi-definite matrix R that satisfies

$$R = GRG' - GRF'[V + FRF']^{-1}FRG' + W \tag{2.18}$$

(which is called the *steady-state (or algebric) Riccati equation*), we say that the DLM has a *steady state* solution.

In the steady state,

$$\theta_t|y_{1:t-1} \sim \mathcal{N}_p(a_t, R),$$

where

$$a_t = (G - AF)a_{t-1} + Ay_t, \tag{2.19}$$

while $R = \mathrm{Var}(\theta_t|y_{1:t-1})$ is time-invariant. In this sense, R represents a bound, intrinsic to the system, to the information one can get in the state forecast. A sufficient condition for R_t to approach R as t increases can be given in terms of the eigenvalues of the matrix $G - AF$: the Kalman filter is asymptotically stable if all the eigenvalues of $G - AF$ are in modulus less than one.

Similarly, the filtering distribution is

$$\theta_t|y_{1:t} \sim \mathcal{N}_p(m_t, C),$$

where $m_t = a_t + K(y_t - Fa_{t-1})$ is recursively updated, while $C = R - KFR$, where $K = RF'[V + FRF']^{-1}$, is time-invariant, giving a bound to the information one can get in filtering.

Note that a solution of (2.18) – i.e., a steady state – does not always exist; and even when a solution is known to exist, it is not simple to show that it

is unique nor that it is a positive semi-definite matrix. However, it can be proved (see Anderson and Moore; 1979) that, if the DLM is observable and controllable, then:

1. For any initial conditions m_0, C_0, we have $R_t \to R$ for $t \to \infty$, and R satisfies the algebraic Riccati equation (2.18);
2. All the eigenvalues of $G - AF$ are smaller than one in modulus, so the Kalman filter is asymptotically stable.

Problems

2.1. Show that

(i) w_t and $(Y_1, \ldots, Y_{t-1})$ are independent;
(ii) w_t and $(\theta_1, \ldots, \theta_{t-1})$ are independent;
(iii) v_t and $(Y_1, \ldots, Y_{t-1})$ are independent;
(iv) v_t and $(\theta_1, \ldots, \theta_t)$ are independent.

2.2. Show that a DLM satisfies the conditional independence assumptions A.1 and A.2 of state space models.

2.3. Give an alternative proof of Proposition 2.2, exploiting the independence properties of the error sequences (see Problem 2.1) and using the state equation directly:

$$\mathrm{E}(\theta_t|y_{1:t-1}) = \mathrm{E}(G_t\theta_{t-1} + w_t|y_{1:t-1}) = G_t m_{t-1}$$

$$\mathrm{Var}(\theta_t|y_{1:t-1}) = \mathrm{Var}(G_t\theta_{t-1} + w_t|y_{1:t-1}) = G_t C_{t-1} G_t' + W_t.$$

Analogously for (ii).

2.4. Give an alternative proof of Proposition 2.6 exploiting the independence properties of the error sequences (see Problem 2.1) and using the state equation directly:

$$\begin{aligned} a_t(k) &= \mathrm{E}(\theta_{t+k}|y_{1:t}) = \mathrm{E}(G_{t+k}\theta_{t+k-1} + w_{t+k}|y_{1:t}) = G_{t+k}a_{t,k-1}, \\ R_t(k) &= \mathrm{Var}(\theta_{t+k}|y_{1:t}) = \mathrm{Var}(G_{t+k}\theta_{t+k-1} + w_{t+k}|y_{1:t}) \\ &= G_{t+k}R_{t,k-1}G_{t+k}' + W_{t+k} \end{aligned}$$

and analogously, from the observation equation:

$$\begin{aligned} f_t(k) &= \mathrm{E}(Y_{t+k}|y_{1:t}) = \mathrm{E}(F_{t+k}\theta_{t+k} + v_{t+k}|y_{1:t}) = F_{t+k}a_t(k), \\ Q_t(k) &= \mathrm{Var}(Y_{t+k}|y_{1:t}) = \mathrm{Var}(F_{t+k}\theta_{t+k} + v_{t+k}|y_{1:t}) \\ &= F_{t+k}R_t(k)F_{t+k}' + V_{t+k}. \end{aligned}$$

2.5. Plot the following data:

$$(Y_t, t = 1, \ldots, 10) = (17, 16.6, 16.3, 16.1, 17.1, 16.9, 16.8, 17.4, 17.1, 17).$$

Consider the random walk plus noise model

$$\begin{aligned} Y_t &= \mu_t + v_t, & v_t &\sim N(0, 0.25), \\ \mu_t &= \mu_{t-1} + w_t, & w_t &\sim N(0, 25), \end{aligned}$$

with $V = 0.25$, $W = 25$, and $\mu_0 \sim N(17, 1)$.
(a) Compute the filtering states estimates.
(b) Compute the one-step ahead forecasts $f_t, t = 1, \ldots, 10$ and plot them,

together with the observations. Comment briefly.
(c) What is the effect of the observation variance V and of the system variance W on the forecasts? Repeat the exercise with different choices of V and W.
(d) Discuss the choice of the initial distribution.
(e) Compute the smoothing state estimates and plot them.

2.6. This requires maximum likelihood estimates (see Chapter 4). For the data and model of Problem 2.5, compute the maximum likelihood estimates of the variances V and W (since these must be positive, write them as $V = \exp(u_1)$, $W = \exp(u_2)$ and compute the MLE of the parameters (u_1, u_2)). Then repeat Problem 2.5, using the MLE of V and W.

2.7. Let $R_{t,h,k} = \mathrm{Cov}(\theta_{t+h}, \theta_{t+k}|y_{1:t})$ and $Q_{t,h,k} = \mathrm{Cov}(Y_{t+h}, Y_{t+k}|y_{1:t})$ for $h, k > 0$, so that $R_{t,k,k} = R_t(k)$ and $Q_{t,k,k} = Q_t(k)$, according to definition (2.10b) and (2.10d).

(i) Show that $R_{t,h,k}$ can be computed recursively via the formula:

$$R_{t,h,k} = G_{t+h} R_{t,h-1,k}, \qquad h > k.$$

(ii) Show that $Q_{t,h,k}$ is equal to $F_{t+h} R_{t,h,k} F'_{t+k}$.
(iii) Find explicit formulae for $R_{t,h,k}$ and $Q_{t,h,k}$ for the random walk plus noise model.

2.8. Derive the filter formulae for the DLM with intercepts:

$$v_t \sim \mathcal{N}(\delta_t, V_t),\ w_t \sim \mathcal{N}(\lambda_t, W_t).$$