6

State-Space Models

6.1 Introduction

A very general model that seems to subsume a whole class of special cases of interest in much the same way that linear regression does is the state-space model or the dynamic linear model, which was introduced in Kalman (1960) and Kalman and Bucy (1961). Although the model was originally introduced as a method primarily for use in aerospace-related research, it has been applied to modeling data from economics (Harrison and Stevens, 1976; Harvey and Pierse, 1984; Harvey and Todd, 1983; Kitagawa and Gersch 1984, Shumway and Stoffer, 1982), medicine (Jones, 1984) and the soil sciences (Shumway, 1988, §3.4.5). An excellent modern treatment of time series analysis based on the state space model is the text by Durbin and Koopman (2001).

The state-space model or dynamic linear model (DLM), in its basic form, employs an order one, vector autoregression as the state equation,

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \tag{6.1}$$

where the state equation determines the rule for the generation of the $p \times 1$ state vector $\boldsymbol{x}_t$ from the past $p \times 1$ state $\boldsymbol{x}_{t-1}$, for time points $t = 1, \ldots, n$. We assume the $\boldsymbol{w}_t$ are $p \times 1$ independent and identically distributed, zero-mean normal vectors with covariance matrix Q. In the DLM, we assume the process starts with a normal vector $\boldsymbol{x}_0$ that has mean $\boldsymbol{\mu}_0$ and $p \times p$ covariance matrix Σ_0.

The DLM, however, adds an additional component to the model in assuming we do not observe the state vector $\boldsymbol{x}_t$ directly, but only a linear transformed version of it with noise added, say

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \boldsymbol{v}_t, \tag{6.2}$$

where A_t is a $q \times p$ measurement or observation matrix; equation (6.2) is called the observation equation. The model arose originally in the space tracking setting, where the state equation defines the motion equations for the position

R.H. Shumway and D.S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, Springer Texts in Statistics, DOI 10.1007/978-1-4419-7865-3_6,
© Springer Science+Business Media, LLC 2011

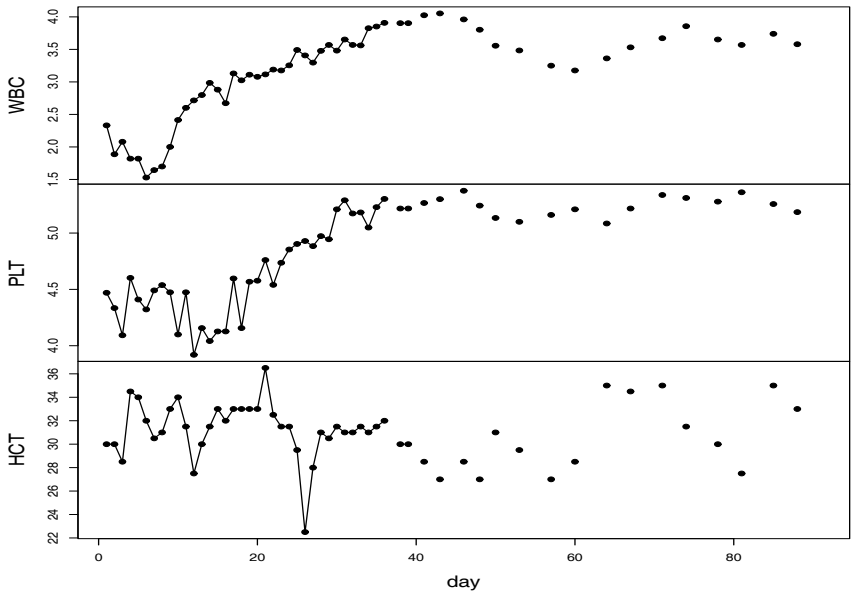

Fig. 6.1. Longitudinal series of blood parameter levels monitored, log (white blood count) [WBC; top], log (platelet) [PLT; middle], and hematocrit [HCT; bottom], after a bone marrow transplant ($n = 91$ days).

or state of a spacecraft with location $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ reflects information that can be observed from a tracking device such as velocity and azimuth. The observed data vector, $\boldsymbol{y}_t$, is q-dimensional, which can be larger than or smaller than p, the state dimension. The additive observation noise $\boldsymbol{v}_t$ is assumed to be white and Gaussian with $q \times q$ covariance matrix R. In addition, we initially assume, for simplicity, $\boldsymbol{x}_0$, $\{\boldsymbol{w}_t\}$ and $\{\boldsymbol{v}_t\}$ are uncorrelated; this assumption is not necessary, but it helps in the explanation of first concepts. The case of correlated errors is discussed in §6.6.

As in the ARMAX model of §5.8, exogenous variables, or fixed inputs, may enter into the states or into the observations. In this case, we suppose we have an $r \times 1$ vector of inputs $\boldsymbol{u}_t$, and write the model as

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \Upsilon \boldsymbol{u}_t + \boldsymbol{w}_t \tag{6.3}$$

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \Gamma \boldsymbol{u}_t + \boldsymbol{v}_t \tag{6.4}$$

where Υ is $p \times r$ and Γ is $q \times r$.

Example 6.1 A Biomedical Example

Suppose we consider the problem of monitoring the level of several biomedical markers after a cancer patient undergoes a bone marrow transplant. The data in Figure 6.1, used by Jones (1984), are measurements made for 91 days

on three variables, log(white blood count) [WBC], log(platelet) [PLT], and hematocrit [HCT], denoted y_{t1}, y_{t2}, and y_{t3}, respectively. Approximately 40% of the values are missing, with missing values occurring primarily after the 35th day. The main objectives are to model the three variables using the state-space approach, and to estimate the missing values. According to Jones, "Platelet count at about 100 days post transplant has previously been shown to be a good indicator of subsequent long term survival." For this particular situation, we model the three variables in terms of the state equation (6.1); that is,

$$\begin{pmatrix} x_{t1} \\ x_{t2} \\ x_{t3} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{pmatrix} \begin{pmatrix} x_{t-1,1} \\ x_{t-1,2} \\ x_{t-1,3} \end{pmatrix} + \begin{pmatrix} w_{t1} \\ w_{t2} \\ w_{t3} \end{pmatrix}. \tag{6.5}$$

The 3×3 observation matrix, A_t, is either the identity matrix, or the identity matrix with all zeros in a row when that variable is missing. The covariance matrices R and Q are 3×3 matrices with $R = \text{diag}\{r_{11}, r_{22}, r_{33}\}$, a diagonal matrix, required for a simple approach when data are missing.

The following R code was used to produce Figure 6.1. These data have zero as the missing data code; to produce a cleaner graphic, we set the zeros to `NA`.

```
blood = cbind(WBC, PLT, HCT)
blood = replace(blood, blood==0, NA)
plot(blood, type="o", pch=19, xlab="day", main="")
```

The model given in (6.1) involving only a single lag is not unduly restrictive. A multivariate model with m lags, such as the VAR(m) discussed in §5.8, could be developed by replacing the $p\times1$ state vector, $\boldsymbol{x}_t$, by the $pm\times1$ state vector $\boldsymbol{X}_t = (\boldsymbol{x}_t', \boldsymbol{x}_{t-1}', \ldots, \boldsymbol{x}_{t-m+1}')'$ and the transition matrix by

$$\Phi = \begin{pmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_{m-1} & \Phi_m \\ I & 0 & \ldots & 0 & 0 \\ 0 & I & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & I & 0 \end{pmatrix}. \tag{6.6}$$

Letting $\boldsymbol{W}_t = (\boldsymbol{w}_t', \boldsymbol{0}', \ldots, \boldsymbol{0}')'$ be the new $pm\times1$ state error vector, the new state equation will be

$$\boldsymbol{X}_t = \Phi\boldsymbol{X}_{t-1} + \boldsymbol{W}_t, \tag{6.7}$$

where $\text{var}(\boldsymbol{W}_t)$ is a $pm\times pm$ matrix with Q in the upper left-hand corner and zeros elsewhere. The observation equation can then be written as

$$\boldsymbol{y}_t = \left[A_t \,\middle|\, 0 \,\middle|\, \cdots \,\middle|\, 0\right] \boldsymbol{X}_t + \boldsymbol{v}_t. \tag{6.8}$$

This simple recoding shows one way of handling higher order lags within the context of the single lag structure. It is not necessary and often not desirable

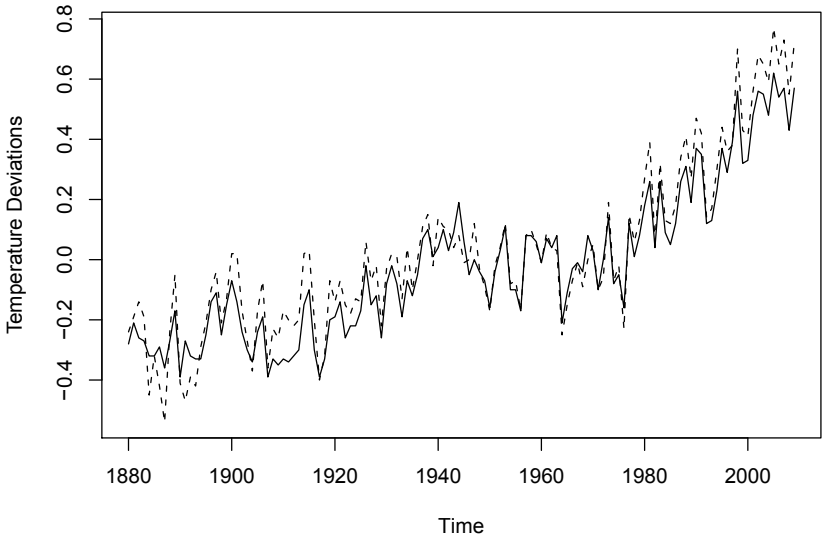

Fig. 6.2. Annual global temperature deviation series, measured in degrees centigrade, 1880–2009. The solid line is the land-marine series (`gtemp`) whereas the dashed line shows the land-based series (`gtemp2`).

to have a singular $\boldsymbol{W}_t$ process in the state equation, (6.7). Further discussion of this topic is given in §6.6.

The real advantages of the state-space formulation, however, do not really come through in the simple example given above. The special forms that can be developed for various versions of the matrix A_t and for the transition scheme defined by the matrix Φ allow fitting more parsimonious structures with fewer parameters needed to describe a multivariate time series. We will give some examples of structural models in §6.5, but the simple example shown below is instructive.

Example 6.2 Global Warming

Figure 6.2 shows two different estimators for the global temperature series from 1880 to 2009. The solid line is `gtemp`, which was considered in the first chapter, and are the global mean land-ocean temperature index data. The second series, `gtemp2`, are the surface air temperature index data using only meteorological station data. Precise details may be obtained from `http://data.giss.nasa.gov/gistemp/graphs/`. Conceptually, both series should be measuring the same underlying climatic signal, and we may consider the problem of extracting this underlying signal. The R code to generate the figure is

```
1 ts.plot(gtemp, gtemp2, lty=1:2, ylab="Temperature Deviations")
```

We suppose both series are observing the same signal with different noises; that is,

$$y_{t1} = x_t + v_{t1} \quad \text{and} \quad y_{t2} = x_t + v_{t2},$$

or more compactly as

$$\begin{pmatrix} y_{t1} \\ y_{t2} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} x_t + \begin{pmatrix} v_{t1} \\ v_{t2} \end{pmatrix}, \tag{6.9}$$

where

$$R = \text{var} \begin{pmatrix} v_{t1} \\ v_{t2} \end{pmatrix} = \begin{pmatrix} r_{11} \; r_{12} \\ r_{21} \; r_{22} \end{pmatrix}.$$

It is reasonable to suppose that the unknown common signal, x_t, can be modeled as a random walk with drift of the form

$$x_t = \delta + x_{t-1} + w_t, \tag{6.10}$$

with $Q = \text{var}(w_t)$. In this example, $p = 1, q = 2$, $\Phi = 1$, and $\Upsilon = \delta$ with $u_t \equiv 1$.

The introduction of the state-space approach as a tool for modeling data in the social and biological sciences requires model identification and parameter estimation because there is rarely a well-defined differential equation describing the state transition. The questions of general interest for the dynamic linear model (6.3) and (6.4) relate to estimating the unknown parameters contained in $\Phi, \Upsilon, Q, \Gamma, A_t$, and R, that define the particular model, and estimating or forecasting values of the underlying unobserved process $\boldsymbol{x}_t$. The advantages of the state-space formulation are in the ease with which we can treat various missing data configurations and in the incredible array of models that can be generated from (6.1) and (6.2). The analogy between the observation matrix A_t and the design matrix in the usual regression and analysis of variance setting is a useful one. We can generate fixed and random effect structures that are either constant or vary over time simply by making appropriate choices for the matrix A_t and the transition structure Φ. We give a few examples in this chapter; for further examples, see Durbin and Koopman (2001), Harvey (1993), Jones (1993), or Shumway (1988) to mention a few.

Before continuing our investigation of the more complex model, it is instructive to consider a simple univariate state-space model wherein an AR(1) process is observed using a noisy instrument.

Example 6.3 An AR(1) Process with Observational Noise

Consider a univariate state-space model where the observations are noisy,

$$y_t = x_t + v_t, \tag{6.11}$$

and the signal (state) is an AR(1) process,

$$x_t = \phi x_{t-1} + w_t, \tag{6.12}$$

for $t = 1, 2, \ldots, n$, where $v_t \sim$ iid $\mathrm{N}(0, \sigma_v^2)$, $w_t \sim$ iid $\mathrm{N}(0, \sigma_w^2)$, and $x_0 \sim \mathrm{N}\big(0, \sigma_w^2/(1-\phi^2)\big)$; $\{v_t\}, \{w_t\}, x_0$ are independent.

In Chapter 3, we investigated the properties of the state, x_t, because it is a stationary AR(1) process (recall Problem 3.2e). For example, we know the autocovariance function of x_t is

$$\gamma_x(h) = \frac{\sigma_w^2}{1-\phi^2}\,\phi^h, \quad h = 0, 1, 2, \ldots\ . \tag{6.13}$$

But here, we must investigate how the addition of observation noise affects the dynamics. Although it is not a necessary assumption, we have assumed in this example that x_t is stationary. In this case, the observations are also stationary because y_t is the sum of two independent stationary components x_t and v_t. We have

$$\gamma_y(0) = \mathrm{var}(y_t) = \mathrm{var}(x_t + v_t) = \frac{\sigma_w^2}{1-\phi^2} + \sigma_v^2, \tag{6.14}$$

and, when $h \geq 1$,

$$\gamma_y(h) = \mathrm{cov}(y_t, y_{t-h}) = \mathrm{cov}(x_t + v_t, x_{t-h} + v_{t-h}) = \gamma_x(h). \tag{6.15}$$

Consequently, for $h \geq 1$, the ACF of the observations is

$$\rho_y(h) = \frac{\gamma_y(h)}{\gamma_y(0)} = \left(1 + \frac{\sigma_v^2}{\sigma_w^2}(1-\phi^2)\right)^{-1} \phi^h. \tag{6.16}$$

It should be clear from the correlation structure given by (6.16) that the observations, y_t, are not AR(1) unless $\sigma_v^2 = 0$. In addition, the autocorrelation structure of y_t is identical to the autocorrelation structure of an ARMA(1,1) process, as presented in Example 3.13. Thus, the observations can also be written in an ARMA(1,1) form,

$$y_t = \phi y_{t-1} + \theta u_{t-1} + u_t,$$

where u_t is Gaussian white noise with variance σ_u^2, and with θ and σ_u^2 suitably chosen. We leave the specifics of this problem alone for now and defer the discussion to §6.6; in particular, see Example 6.11.

Although an equivalence exists between stationary ARMA models and stationary state-space models (see §6.6), it is sometimes easier to work with one form than another. As previously mentioned, in the case of missing data, complex multivariate systems, mixed effects, and certain types of nonstationarity, it is easier to work in the framework of state-space models; in this chapter, we explore some of these situations.

6.2 Filtering, Smoothing, and Forecasting

From a practical view, the primary aims of any analysis involving the state-space model as defined by (6.1)-(6.2), or by (6.3)-(6.4), would be to produce estimators for the underlying unobserved signal $\boldsymbol{x}_t$, given the data $Y_s = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s\}$, to time s. When $s < t$, the problem is called forecasting or prediction. When $s = t$, the problem is called filtering, and when $s > t$, the problem is called smoothing. In addition to these estimates, we would also want to measure their precision. The solution to these problems is accomplished via the Kalman filter and smoother and is the focus of this section.

Throughout this chapter, we will use the following definitions:

$$\boldsymbol{x}_t^s = E(\boldsymbol{x}_t \mid Y_s) \tag{6.17}$$

and

$$P_{t_1,t_2}^s = E\left\{(\boldsymbol{x}_{t_1} - \boldsymbol{x}_{t_1}^s)(\boldsymbol{x}_{t_2} - \boldsymbol{x}_{t_2}^s)'\right\}. \tag{6.18}$$

When $t_1 = t_2$ ($= t$ say) in (6.18), we will write P_t^s for convenience.

In obtaining the filtering and smoothing equations, we will rely heavily on the Gaussian assumption. Some knowledge of the material covered in Appendix B, §B.1, will be helpful in understanding the details of this section (although these details may be skipped on a casual reading of the material). Even in the non-Gaussian case, the estimators we obtain are the minimum mean-squared error estimators within the class of linear estimators. That is, we can think of E in (6.17) as the projection operator in the sense of §B.1 rather than expectation and Y_s as the space of linear combinations of $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s\}$; in this case, P_t^s is the corresponding mean-squared error. When we assume, as in this section, the processes are Gaussian, (6.18) is also the conditional error covariance; that is,

$$P_{t_1,t_2}^s = E\left\{(\boldsymbol{x}_{t_1} - \boldsymbol{x}_{t_1}^s)(\boldsymbol{x}_{t_2} - \boldsymbol{x}_{t_2}^s)' \mid Y_s\right\}.$$

This fact can be seen, for example, by noting the covariance matrix between $(\boldsymbol{x}_t - \boldsymbol{x}_t^s)$ and Y_s, for any t and s, is zero; we could say they are orthogonal in the sense of §B.1. This result implies that $(\boldsymbol{x}_t - \boldsymbol{x}_t^s)$ and Y_s are independent (because of the normality), and hence, the conditional distribution of $(\boldsymbol{x}_t - \boldsymbol{x}_t^s)$ given Y_s is the unconditional distribution of $(\boldsymbol{x}_t - \boldsymbol{x}_t^s)$. Derivations of the filtering and smoothing equations from a Bayesian perspective are given in Meinhold and Singpurwalla (1983); more traditional approaches based on the concept of projection and on multivariate normal distribution theory are given in Jazwinski (1970) and Anderson and Moore (1979).

First, we present the Kalman filter, which gives the filtering and forecasting equations. The name filter comes from the fact that $\boldsymbol{x}_t^t$ is a linear filter of the observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$; that is, $\boldsymbol{x}_t^t = \sum_{s=1}^t B_s \boldsymbol{y}_s$ for suitably chosen $p \times q$ matrices B_s. The advantage of the Kalman filter is that it specifies how to update the filter from $\boldsymbol{x}_{t-1}^{t-1}$ to $\boldsymbol{x}_t^t$ once a new observation $\boldsymbol{y}_t$ is obtained, without having to reprocess the entire data set $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$.

Property 6.1 The Kalman Filter

For the state-space model specified in (6.3) and (6.4), with initial conditions $\boldsymbol{x}_0^0 = \boldsymbol{\mu}_0$ *and* $P_0^0 = \Sigma_0$*, for* $t = 1, \ldots, n$*,*

$$\boldsymbol{x}_t^{t-1} = \Phi \boldsymbol{x}_{t-1}^{t-1} + \Upsilon \boldsymbol{u}_t, \tag{6.19}$$

$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi' + Q, \tag{6.20}$$

with

$$\boldsymbol{x}_t^t = \boldsymbol{x}_t^{t-1} + K_t(\boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t), \tag{6.21}$$

$$P_t^t = [I - K_t A_t] P_t^{t-1}, \tag{6.22}$$

where

$$K_t = P_t^{t-1} A_t' [A_t P_t^{t-1} A_t' + R]^{-1} \tag{6.23}$$

is called the Kalman gain. Prediction for $t > n$ *is accomplished via (6.19) and (6.20) with initial conditions* $\boldsymbol{x}_n^n$ *and* P_n^n*. Important byproducts of the filter are the innovations (prediction errors)*

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - E(\boldsymbol{y}_t \mid Y_{t-1}) = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t, \tag{6.24}$$

and the corresponding variance-covariance matrices

$$\Sigma_t \stackrel{\text{def}}{=} \operatorname{var}(\boldsymbol{\epsilon}_t) \ = \operatorname{var}[A_t(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}) + \boldsymbol{v}_t] = A_t P_t^{t-1} A_t' + R \tag{6.25}$$

for $t = 1, \ldots, n$

Proof. The derivations of (6.19) and (6.20) follow from straight forward calculations, because from (6.3) we have

$$\boldsymbol{x}_t^{t-1} = E(\boldsymbol{x}_t \mid Y_{t-1}) = E(\Phi \boldsymbol{x}_{t-1} + \Upsilon \boldsymbol{u}_t + \boldsymbol{w}_t \mid Y_{t-1}) = \Phi x_{t-1}^{t-1} + \Upsilon \boldsymbol{u}_t,$$

and thus

$$\begin{aligned} P_t^{t-1} &= E\left\{(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1})(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1})'\right\} \\ &= E\left\{\left[\Phi(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{t-1}) + \boldsymbol{w}_t\right]\left[\Phi(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{t-1}) + \boldsymbol{w}_t\right]'\right\} \\ &= \Phi P_{t-1}^{t-1} \Phi' + Q. \end{aligned}$$

To derive (6.21), we note that $E(\boldsymbol{\epsilon}_t \boldsymbol{y}_s') = 0$ for $s < t$, which in view of the fact the innovation sequence is a Gaussian process, implies that the innovations are independent of the past observations. Furthermore, the conditional covariance between $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}_t$ given Y_{t-1} is

$$\begin{aligned} \operatorname{cov}(\boldsymbol{x}_t,\ \boldsymbol{\epsilon}_t \mid Y_{t-1}) &= \operatorname{cov}(\boldsymbol{x}_t,\ \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t \mid Y_{t-1}) \\ &= \operatorname{cov}(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1},\ \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t \mid Y_{t-1}) \\ &= \operatorname{cov}[\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1},\ A_t(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}) + \boldsymbol{v}_t] \\ &= P_t^{t-1} A_t'. \end{aligned} \tag{6.26}$$

Using these results we have that the joint conditional distribution of $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}_t$ given Y_{t-1} is normal

$$\begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{\epsilon}_t \end{pmatrix} \Big| \; Y_{t-1} \sim \mathrm{N}\left(\begin{bmatrix} \boldsymbol{x}_t^{t-1} \\ \mathbf{0} \end{bmatrix}, \; \begin{bmatrix} P_t^{t-1} & P_t^{t-1} A_t' \\ A_t P_t^{t-1} & \Sigma_t \end{bmatrix} \right). \tag{6.27}$$

Thus, using (B.9) of Appendix B, we can write

$$\boldsymbol{x}_t^t = E(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}, \boldsymbol{y}_t) = E(\boldsymbol{x}_t \mid Y_{t-1}, \boldsymbol{\epsilon}_t) = \boldsymbol{x}_t^{t-1} + K_t \boldsymbol{\epsilon}_t, \tag{6.28}$$

where

$$K_t = P_t^{t-1} A_t' \Sigma_t^{-1} = P_t^{t-1} A_t' (A_t P_t^{t-1} A_t' + R)^{-1}.$$

The evaluation of P_t^t is easily computed from (6.27) [see (B.10)] as

$$P_t^t = \mathrm{cov}\left(\boldsymbol{x}_t \mid Y_{t-1}, \boldsymbol{\epsilon}_t\right) = P_t^{t-1} - P_t^{t-1} A_t' \Sigma_t^{-1} A_t P_t^{t-1},$$

which simplifies to (6.22). □

Nothing in the proof of Property 6.1 precludes the cases where some or all of the parameters vary with time, or where the observation dimension changes with time, which leads to the following corollary.

Corollary 6.1 Kalman Filter: The Time-Varying Case
If, in the DLM (6.3) and (6.4), any or all of the parameters are time dependent, $\Phi = \Phi_t$, $\Upsilon = \Upsilon_t$, $Q = Q_t$ in the state equation or $\Gamma = \Gamma_t$, $R = R_t$ in the observation equation, or the dimension of the observational equation is time dependent, $q = q_t$, Property 6.1 holds with the appropriate substitutions.

Next, we explore the model, prediction, and filtering from a density point of view. For the sake of brevity, consider the Gaussian DLM without inputs, as described in (6.1) and (6.2); that is,

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \boldsymbol{w}_t \quad \text{and} \quad \boldsymbol{y}_t = A_t \boldsymbol{x}_t + \boldsymbol{v}_t.$$

Recall $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are independent, white Gaussian sequences, and the initial state is normal, say, $\boldsymbol{x}_0 \sim \mathrm{N}(\boldsymbol{\mu}_0, \Sigma_0)$; we will denote the initial p-variate state normal density by $f_0(\boldsymbol{x}_0)$. Now, letting $p_\Theta(\cdot)$ denote a generic density function with parameters represented by Θ, we could describe the state relationship as

$$p_\Theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots, \boldsymbol{x}_0) = p_\Theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = f_w(\boldsymbol{x}_t - \Phi \boldsymbol{x}_{t-1}), \tag{6.29}$$

where $f_w(\cdot)$ denotes the p-variate normal density with mean zero and variance-covariance matrix Q. In (6.29), we are stating the process is Markovian, linear, and Gaussian. The relationship of the observations to the state process is written as

$$p_\Theta(\boldsymbol{y}_t \mid \boldsymbol{x}_t, Y_{t-1}) = p_\Theta(\boldsymbol{y}_t \mid \boldsymbol{x}_t) = f_v(\boldsymbol{y}_t - A_t \boldsymbol{x}_t), \tag{6.30}$$

where $f_v(\cdot)$ denotes the q-variate normal density with mean zero and variance-covariance matrix R. In (6.30), we are stating the observations are conditionally independent given the state, and the observations are linear and Gaussian.

Note, (6.29), (6.30), and the initial density, $f_0(\cdot)$, completely specify the model in terms of densities, namely,

$$p_\Theta(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = f_0(\boldsymbol{x}_0) \prod_{t=1}^{n} f_w(\boldsymbol{x}_t - \Phi \boldsymbol{x}_{t-1}) f_v(\boldsymbol{y}_t - A_t \boldsymbol{x}_t), \tag{6.31}$$

where $\Theta = \{\boldsymbol{\mu}_0, \Sigma_0, \Phi, Q, R\}$.

Given the data, $Y_{t-1} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}\}$, and the current filter density, $p_\Theta(\boldsymbol{x}_{t-1} \,|\, Y_{t-1})$, Property 6.1 tells us, via conditional means and variances, how to recursively generate the Gaussian forecast density, $p_\Theta(\boldsymbol{x}_t \,|\, Y_{t-1})$, and how to update the density given the current observation, $\boldsymbol{y}_t$, to obtain the Gaussian filter density, $p_\Theta(\boldsymbol{x}_t \,|\, Y_t)$. In terms of densities, the Kalman filter can be seen as a simple Bayesian updating scheme, where, to determine the forecast and filter densities, we have

$$\begin{aligned} p_\Theta(\boldsymbol{x}_t \mid Y_{t-1}) &= \int_{R^p} p_\Theta(\boldsymbol{x}_t, \boldsymbol{x}_{t-1} \mid Y_{t-1})\, d\boldsymbol{x}_{t-1} \\ &= \int_{R^p} p_\Theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) p_\Theta(\boldsymbol{x}_{t-1} \mid Y_{t-1})\, d\boldsymbol{x}_{t-1} \\ &= \int_{R^p} f_w(\boldsymbol{x}_t - \Phi \boldsymbol{x}_{t-1}) p_\Theta(\boldsymbol{x}_{t-1} \mid Y_{t-1}) d\boldsymbol{x}_{t-1}, \end{aligned} \tag{6.32}$$

which simplifies to the p-variate $\mathrm{N}(\boldsymbol{x}_t^{t-1}, P_t^{t-1})$ density, and

$$\begin{aligned} p_\Theta(\boldsymbol{x}_t \mid Y_t) &= p_\Theta(\boldsymbol{x}_t \mid \boldsymbol{y}_t, Y_{t-1}) \\ &\propto p_\Theta(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\; p_\Theta(\boldsymbol{x}_t \mid Y_{t-1}), \\ &= f_v(\boldsymbol{y}_t - A_t \boldsymbol{x}_t) p_\Theta(\boldsymbol{x}_t \mid Y_{t-1}), \end{aligned} \tag{6.33}$$

from which we can deduce $p_\Theta(\boldsymbol{x}_t \mid Y_t)$ is the p-variate $\mathrm{N}(\boldsymbol{x}_t^t, P_t^t)$ density. These statements are true for $t = 1, \ldots, n$, with initial condition $p_\Theta(\boldsymbol{x}_0 \mid Y_0) = f_0(\boldsymbol{x}_0)$. The prediction and filter recursions of Property 6.1 could also have been calculated directly from the density relationships (6.32) and (6.33) using multivariate normal distribution theory. The following example illustrates the Bayesian updating scheme.

Example 6.4 Bayesian Analysis of a Local Level Model

In this example, we suppose that we observe a univariate series y_t that consists of a trend component, μ_t, and a noise component, v_t, where

$$y_t = \mu_t + v_t \tag{6.34}$$

and $v_t \sim$ iid $\mathrm{N}(0, \sigma_v^2)$. In particular, we assume the trend is a random walk given by

$$\mu_t = \mu_{t-1} + w_t \tag{6.35}$$

where $w_t \sim$ iid $\mathrm{N}(0, \sigma_w^2)$ is independent of $\{v_t\}$. Recall Example 6.2, where we suggested this type of trend model for the global temperature series.

The model is, of course, a state-space model with (6.34) being the observation equation, and (6.35) being the state equation. For forecasting, we seek the posterior density $p(\mu_t \mid Y_{t-1})$. We will use the following notation introduced in Blight (1974) for the multivariate case. Let

$$\{x; \mu, \sigma^2\} = \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}, \tag{6.36}$$

then simple manipulation shows

$$\{x; \mu, \sigma^2\} = \{\mu; x, \sigma^2\} \tag{6.37}$$

and

$$\begin{aligned}\{x; \mu_1, \sigma_1^2\}\{x; \mu_2, \sigma_2^2\} = &\left\{x; \frac{\mu_1/\sigma_1^2 + \mu_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}, (1/\sigma_1^2 + 1/\sigma_2^2)^{-1}\right\} \\ &\times \left\{\mu_1; \mu_2, \sigma_1^2 + \sigma_2^2\right\}.\end{aligned} \tag{6.38}$$

Thus, using (6.32), (6.37) and (6.38) we have

$$\begin{aligned}p(\mu_t \mid Y_{t-1}) &\propto \int \left\{\mu_t; \mu_{t-1}, \sigma_w^2\right\} \left\{\mu_{t-1}; \mu_{t-1}^{t-1}, P_{t-1}^{t-1}\right\} d\mu_{t-1} \\ &= \int \left\{\mu_{t-1}; \mu_t, \sigma_w^2\right\} \left\{\mu_{t-1}; \mu_{t-1}^{t-1}, P_{t-1}^{t-1}\right\} d\mu_{t-1} \\ &= \left\{\mu_t; \mu_{t-1}^{t-1}, P_{t-1}^{t-1} + \sigma_w^2\right\}.\end{aligned} \tag{6.39}$$

From (6.39) we conclude that

$$\mu_t \mid Y_{t-1} \sim \mathrm{N}(\mu_t^{t-1}, P_t^{t-1}) \tag{6.40}$$

where

$$\mu_t^{t-1} = \mu_{t-1}^{t-1} \quad \text{and} \quad P_t^{t-1} = P_{t-1}^{t-1} + \sigma_w^2 \tag{6.41}$$

which agrees with the first part of Property 6.1.

To derive the filter density using (6.33) and (6.37) we have

$$\begin{aligned}p(\mu_t \mid Y_t) &\propto \left\{y_t; \mu_t, \sigma_v^2\right\} \left\{\mu_t; \mu_t^{t-1}, P_t^{t-1}\right\} \\ &= \left\{\mu_t; y_t, \sigma_v^2\right\} \left\{\mu_t; \mu_t^{t-1}, P_t^{t-1}\right\}.\end{aligned} \tag{6.42}$$

An application of (6.38) gives

$$\mu_t \mid Y_t \sim \mathrm{N}(\mu_t^t, P_t^t) \tag{6.43}$$

with

$$\mu_t^t = \frac{\sigma_v^2 \mu_t^{t-1}}{P_t^{t-1} + \sigma_v^2} + \frac{P_t^{t-1} y_t}{P_t^{t-1} + \sigma_v^2} = \mu_t^{t-1} + K_t(y_t - \mu_t^{t-1}), \tag{6.44}$$

where we have defined

$$K_t = \frac{P_t^{t-1}}{P_t^{t-1} + \sigma_v^2}, \tag{6.45}$$

and

$$P_t^t = \left(\frac{1}{\sigma_v^2} + \frac{1}{P_t^{t-1}}\right)^{-1} = \frac{\sigma_v^2 P_t^{t-1}}{P_t^{t-1} + \sigma_v^2} = (1 - K_t)P_t^{t-1}. \tag{6.46}$$

The filter for this specific case, of course, agrees with Property 6.1.

Next, we consider the problem of obtaining estimators for $\boldsymbol{x}_t$ based on the entire data sample $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, where $t \leq n$, namely, $\boldsymbol{x}_t^n$. These estimators are called smoothers because a time plot of the sequence $\{\boldsymbol{x}_t^n;\ t = 1, \ldots, n\}$ is typically smoother than the forecasts $\{\boldsymbol{x}_t^{t-1};\ t = 1, \ldots, n\}$ or the filters $\{\boldsymbol{x}_t^t;\ t = 1, \ldots, n\}$. As is obvious from the above remarks, smoothing implies that each estimated value is a function of the present, future, and past, whereas the filtered estimator depends on the present and past. The forecast depends only on the past, as usual.

Property 6.2 The Kalman Smoother

For the state-space model specified in (6.3) and (6.4), with initial conditions $\boldsymbol{x}_n^n$ and P_n^n obtained via Property 6.1, for $t = n, n-1, \ldots, 1$,

$$\boldsymbol{x}_{t-1}^n = \boldsymbol{x}_{t-1}^{t-1} + J_{t-1}\left(\boldsymbol{x}_t^n - \boldsymbol{x}_t^{t-1}\right), \tag{6.47}$$

$$P_{t-1}^n = P_{t-1}^{t-1} + J_{t-1}\left(P_t^n - P_t^{t-1}\right) J_{t-1}', \tag{6.48}$$

where

$$J_{t-1} = P_{t-1}^{t-1}\Phi'\left[P_t^{t-1}\right]^{-1}. \tag{6.49}$$

Proof. The smoother can be derived in many ways. Here we provide a proof that was given in Ansley and Kohn (1982). First, for $1 \leq t \leq n$, define

$$Y_{t-1} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}\} \quad \text{and} \quad \eta_t = \{\boldsymbol{v}_t, \ldots, \boldsymbol{v}_n, \boldsymbol{w}_{t+1}, \ldots, \boldsymbol{w}_n\},$$

with Y_0 being empty, and let

$$\boldsymbol{q}_{t-1} = E\{\boldsymbol{x}_{t-1} \mid Y_{t-1}, \boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}, \eta_t\}.$$

Then, because Y_{t-1}, $\{\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}\}$, and η_t are mutually independent, and $\boldsymbol{x}_{t-1}$ and η_t are independent, using (B.9) we have

$$\boldsymbol{q}_{t-1} = \boldsymbol{x}_{t-1}^{t-1} + J_{t-1}(\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}), \tag{6.50}$$

where

$$J_{t-1} = \mathrm{cov}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t - \boldsymbol{x}_t^{t-1})[P_t^{t-1}]^{-1} = P_{t-1}^{t-1}\Phi'[P_t^{t-1}]^{-1}.$$

Finally, because Y_{t-1}, $\boldsymbol{x}_t - \boldsymbol{x}_t^{t-1}$, and η_t generate $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$,

$$x_{t-1}^n = E\{x_{t-1} \mid Y_n\} = E\{\boldsymbol{q}_{t-1} \mid Y_n\} = \boldsymbol{x}_{t-1}^{t-1} + J_{t-1}(\boldsymbol{x}_t^n - \boldsymbol{x}_t^{t-1}),$$

which establishes (6.47).

The recursion for the error covariance, P_{t-1}^n, is obtained by straightforward calculation. Using (6.47) we obtain

$$\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^n = \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{t-1} - J_{t-1}\left(\boldsymbol{x}_t^n - \Phi \boldsymbol{x}_{t-1}^{t-1}\right),$$

or

$$\left(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^n\right) + J_{t-1}\boldsymbol{x}_t^n = \left(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{t-1}\right) + J_{t-1}\Phi \boldsymbol{x}_{t-1}^{t-1}. \tag{6.51}$$

Multiplying each side of (6.51) by the transpose of itself and taking expectation, we have

$$P_{t-1}^n + J_{t-1}E(\boldsymbol{x}_t^n \boldsymbol{x}_t^{n'})J_{t-1}' = P_{t-1}^{t-1} + J_{t-1}\Phi E(\boldsymbol{x}_{t-1}^{t-1}\boldsymbol{x}_{t-1}^{t-1'})\Phi' J_{t-1}', \tag{6.52}$$

using the fact the cross-product terms are zero. But,

$$E(\boldsymbol{x}_t^n \boldsymbol{x}_t^{n'}) = E(\boldsymbol{x}_t \boldsymbol{x}_t') - P_t^n = \Phi E(\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}')\Phi' + Q - P_t^n,$$

and

$$E(\boldsymbol{x}_{t-1}^{t-1}\boldsymbol{x}_{t-1}^{t-1'}) = E(\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}') - P_{t-1}^{t-1},$$

so (6.52) simplifies to (6.48). □

Example 6.5 Prediction, Filtering and Smoothing for the Local Level Model

For this example, we simulated $n = 50$ observations from the local level trend model discussed in Example 6.4. We generated a random walk

$$\mu_t = \mu_{t-1} + w_t \tag{6.53}$$

with $w_t \sim$ iid N(0, 1) and $\mu_0 \sim$ N(0, 1). We then supposed that we observe a univariate series y_t consisting of the trend component, μ_t, and a noise component, $v_t \sim$ iid N(0, 1), where

$$y_t = \mu_t + v_t. \tag{6.54}$$

The sequences $\{w_t\}$, $\{v_t\}$ and μ_0 were generated independently. We then ran the Kalman filter and smoother, Properties 6.1 and 6.2, using the actual parameters. The top panel of Figure 6.3 shows the actual values of μ_t as points, and the predictions μ_t^{t-1}, for $t = 1, 2, \ldots, 50$, superimposed on the graph as a line. In addition, we display $\mu_t^{t-1} \pm 2\sqrt{P_t^{t-1}}$ as dashed lines on the plot. The middle panel displays the filter, μ_t^t, for $t = 1, \ldots, 50$, as a line with $\mu_t^t \pm 2\sqrt{P_t^t}$ as dashed lines. The bottom panel of Figure 6.3 shows a similar plot for the smoother μ_t^n.

Table 6.1 shows the first 10 observations as well as the corresponding state values, the predictions, filters and smoothers. Note that in Table 6.1,

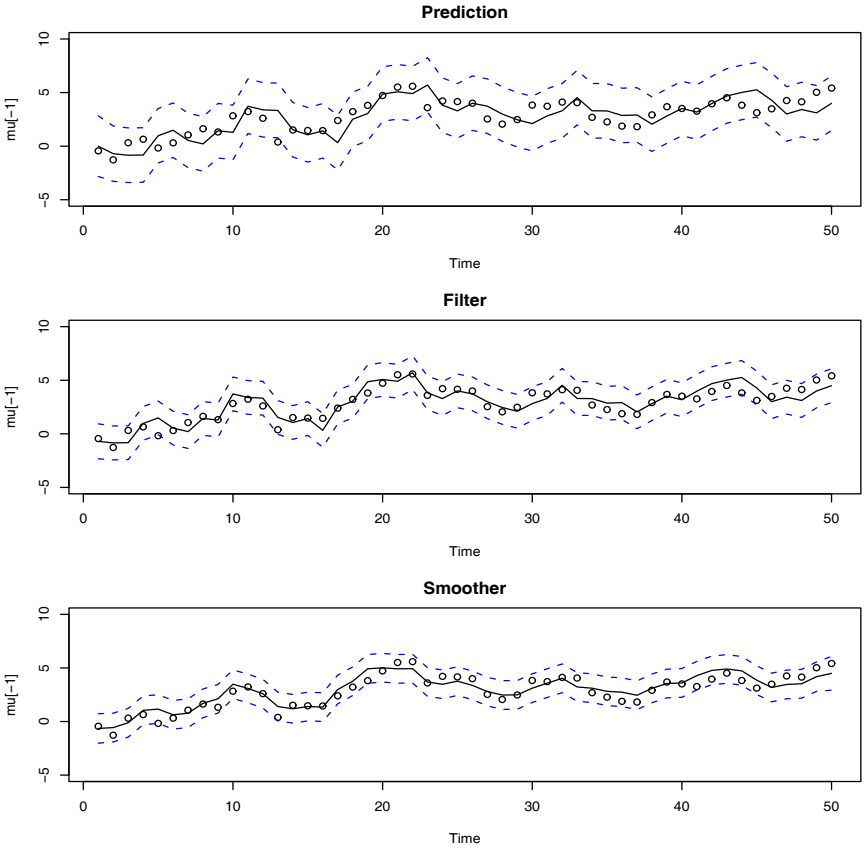

Fig. 6.3. Displays for Example 6.5. The simulated values of μ_t, for $t = 1, \ldots, 50$, given by (6.53) are shown as points. *Top:* The predictions μ_t^{t-1} obtained via the Kalman filter are shown as a line. Error bounds, $\mu_t^{t-1} \pm 2\sqrt{P_t^{t-1}}$, are shown as dashed lines. *Middle:* The filter μ_t^t obtained via the Kalman filter are shown as a line. Error bounds, $\mu_t^t \pm 2\sqrt{P_t^t}$, are shown as dashed lines. *Bottom:* The smoothers μ_t^n obtained via the Kalman smoother are shown as a line. Error bounds, $\mu_t^n \pm 2\sqrt{P_t^n}$, are shown as dashed lines.

one-step-ahead prediction is more uncertain than the corresponding filtered value, which, in turn, is more uncertain than the corresponding smoother value (that is $P_t^{t-1} \geq P_t^t \geq P_t^n$). Also, in each case, the error variances stabilize quickly.

The R code for this example is as follows. In the example we use `Ksmooth0`, which calls `Kfilter0` for the filtering part (see Appendix R). In the returned values from `Ksmooth0`, the letters `p, f, s` denote prediction, filter, and smooth, respectively (e.g., `xp` is x_t^{t-1}, `xf` is x_t^t, `xs` is x_t^n, and so on).

Table 6.1. Forecasts, Filters, and Smoothers for Example 6.5

t	y_t	μ_t	μ_t^{t-1}	P_t^{t-1}	μ_t^t	P_t^t	μ_t^n	P_t^n
0	—	−.63	—	—	.00	1.00	−.32	.62
1	−1.05	−.44	.00	2.00	−.70	.67	−.65	.47
2	−.94	−1.28	−.70	1.67	−.85	.63	−.57	.45
3	−.81	.32	−.85	1.63	−.83	.62	−.11	.45
4	2.08	.65	−.83	1.62	.97	.62	1.04	.45
5	1.81	−.17	.97	1.62	1.49	.62	1.16	.45
6	−.05	.31	1.49	1.62	.53	.62	.63	.45
7	.01	1.05	.53	1.62	.21	.62	.78	.45
8	2.20	1.63	.21	1.62	1.44	.62	1.70	.45
9	1.19	1.32	1.44	1.62	1.28	.62	2.12	.45
10	5.24	2.83	1.28	1.62	3.73	.62	3.48	.45

These scripts use the Cholesky decomposition[1] of Q and R; they are denoted by `cQ` and `cR`. Practically, the scripts only require that `Q` or `R` may be reconstructed as `t(cQ)%*%(cQ)` or `t(cR)%*%(cR)`, respectively, which allows more flexibility. For example, the model (6.7) - (6.8) does not pose a problem even though the state noise covariance matrix is not positive definite.

```
# generate data
set.seed(1);  num = 50
w = rnorm(num+1,0,1); v = rnorm(num,0,1)
mu = cumsum(w)  # state:  mu[0], mu[1],..., mu[50]
y = mu[-1] + v  #   obs:  y[1],..., y[50]
# filter and smooth
mu0 = 0; sigma0 = 1;  phi = 1; cQ = 1; cR = 1
ks = Ksmooth0(num, y, 1, mu0, sigma0, phi, cQ, cR)
# start figure
par(mfrow=c(3,1));  Time = 1:num
plot(Time, mu[-1], main="Prediction", ylim=c(-5,10))
  lines(ks$xp)
  lines(ks$xp+2*sqrt(ks$Pp), lty="dashed", col="blue")
  lines(ks$xp-2*sqrt(ks$Pp), lty="dashed", col="blue")
plot(Time, mu[-1], main="Filter", ylim=c(-5,10))
  lines(ks$xf)
  lines(ks$xf+2*sqrt(ks$Pf), lty="dashed", col="blue")
  lines(ks$xf-2*sqrt(ks$Pf), lty="dashed", col="blue")
plot(Time, mu[-1],  main="Smoother", ylim=c(-5,10))
  lines(ks$xs)
  lines(ks$xs+2*sqrt(ks$Ps), lty="dashed", col="blue")
  lines(ks$xs-2*sqrt(ks$Ps), lty="dashed", col="blue")
mu[1]; ks$x0n; sqrt(ks$P0n)   # initial value info
```

[1] Given a positive definite matrix A, its Cholesky decomposition is an upper triangular matrix U with strictly positive diagonal entries such that $A = U'U$. In R, use `chol(A)`. For the univariate case, it is simply the positive square root of A.

When we discuss maximum likelihood estimation via the EM algorithm in the next section, we will need a set of recursions for obtaining $P_{t,t-1}^n$, as defined in (6.18). We give the necessary recursions in the following property.

Property 6.3 The Lag-One Covariance Smoother

For the state-space model specified in (6.3) and (6.4), with K_t, J_t ($t = 1, \ldots, n$), and P_n^n obtained from Properties 6.1 and 6.2, and with initial condition

$$P_{n,n-1}^n = (I - K_n A_n)\Phi P_{n-1}^{n-1}, \tag{6.55}$$

for $t = n, n-1, \ldots, 2$,

$$P_{t-1,t-2}^n = P_{t-1}^{t-1} J_{t-2}' + J_{t-1}\left(P_{t,t-1}^n - \Phi P_{t-1}^{t-1}\right) J_{t-2}'. \tag{6.56}$$

Proof. Because we are computing covariances, we may assume $\boldsymbol{u}_t \equiv \mathbf{0}$ without loss of generality. To derive the initial term (6.55), we first define

$$\widetilde{\boldsymbol{x}}_t^s = \boldsymbol{x}_t - \boldsymbol{x}_t^s.$$

Then, using (6.21) and (6.47), we write

$$\begin{aligned} P_{t,t-1}^t &= E\left(\widetilde{\boldsymbol{x}}_t^t\, \widetilde{\boldsymbol{x}}_{t-1}^{t'}\right) \\ &= E\left\{[\widetilde{\boldsymbol{x}}_t^{t-1} - K_t(\boldsymbol{y}_t - A_t\boldsymbol{x}_t^{t-1})][\widetilde{\boldsymbol{x}}_{t-1}^{t-1} - J_{t-1}K_t(\boldsymbol{y}_t - A_t\boldsymbol{x}_t^{t-1})]'\right\} \\ &= E\left\{[\widetilde{\boldsymbol{x}}_t^{t-1} - K_t(A_t\widetilde{\boldsymbol{x}}_t^{t-1} + \boldsymbol{v}_t)][\widetilde{\boldsymbol{x}}_{t-1}^{t-1} - J_{t-1}K_t(A_t\widetilde{\boldsymbol{x}}_t^{t-1} + \boldsymbol{v}_t)]'\right\}. \end{aligned}$$

Expanding terms and taking expectation, we arrive at

$$P_{t,t-1}^t = P_{t,t-1}^{t-1} - P_t^{t-1}A_t'K_t'J_{t-1}' - K_tA_tP_{t,t-1}^{t-1} + K_t(A_tP_t^{t-1}A_t' + R)K_t'J_{t-1}',$$

noting $E(\widetilde{\boldsymbol{x}}_t^{t-1}\boldsymbol{v}_t') = 0$. The final simplification occurs by realizing that $K_t(A_tP_t^{t-1}A_t' + R) = P_t^{t-1}A_t'$, and $P_{t,t-1}^{t-1} = \Phi P_{t-1}^{t-1}$. These relationships hold for any $t = 1, \ldots, n$, and (6.55) is the case $t = n$.

We give the basic steps in the derivation of (6.56). The first step is to use (6.47) to write

$$\widetilde{\boldsymbol{x}}_{t-1}^n + J_{t-1}\boldsymbol{x}_t^n = \widetilde{\boldsymbol{x}}_{t-1}^{t-1} + J_{t-1}\Phi\boldsymbol{x}_{t-1}^{t-1} \tag{6.57}$$

and

$$\widetilde{\boldsymbol{x}}_{t-2}^n + J_{t-2}\boldsymbol{x}_{t-1}^n = \widetilde{\boldsymbol{x}}_{t-2}^{t-2} + J_{t-2}\Phi\boldsymbol{x}_{t-2}^{t-2}. \tag{6.58}$$

Next, multiply the left-hand side of (6.57) by the transpose of the left-hand side of (6.58), and equate that to the corresponding result of the right-hand sides of (6.57) and (6.58). Then, taking expectation of both sides, the left-hand side result reduces to

$$P_{t-1,t-2}^n + J_{t-1}E(\boldsymbol{x}_t^n\boldsymbol{x}_{t-1}^{n'})J_{t-2}' \tag{6.59}$$

and the right-hand side result reduces to

$$P_{t-1,t-2}^{t-2} - K_{t-1}A_{t-1}P_{t-1,t-2}^{t-2} + J_{t-1}\Phi K_{t-1}A_{t-1}P_{t-1,t-2}^{t-2} + J_{t-1}\Phi E(\boldsymbol{x}_{t-1}^{t-1}\boldsymbol{x}_{t-2}^{t-2'})\Phi' J_{t-2}'. \tag{6.60}$$

In (6.59), write

$$E(\boldsymbol{x}_t^n \boldsymbol{x}_{t-1}^{n'}) = E(\boldsymbol{x}_t \boldsymbol{x}_{t-1}') - P_{t,t-1}^n = \Phi E(\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-2}')\Phi' + \Phi Q - P_{t,t-1}^n,$$

and in (6.60), write

$$E(\boldsymbol{x}_{t-1}^{t-1}\boldsymbol{x}_{t-2}^{t-2'}) = E(\boldsymbol{x}_{t-1}^{t-2}\boldsymbol{x}_{t-2}^{t-2'}) = E(\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-2}') - P_{t-1,t-2}^{t-2}.$$

Equating (6.59) to (6.60) using these relationships and simplifying the result leads to (6.56). □

6.3 Maximum Likelihood Estimation

The estimation of the parameters that specify the state-space model, (6.3) and (6.4), is quite involved. We use $\Theta = \{\boldsymbol{\mu}_0, \Sigma_0, \Phi, Q, R, \Upsilon, \Gamma\}$ to represent the vector of parameters containing the elements of the initial mean and covariance $\boldsymbol{\mu}_0$ and Σ_0, the transition matrix Φ, and the state and observation covariance matrices Q and R and the input coefficient matrices, Υ and Γ. We use maximum likelihood under the assumption that the initial state is normal, $\boldsymbol{x}_0 \sim \mathrm{N}(\boldsymbol{\mu}_0, \Sigma_0)$, and the errors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are jointly normal and uncorrelated vector variables. We continue to assume, for simplicity, $\{\boldsymbol{w}_t\}$ and $\{\boldsymbol{v}_t\}$ are uncorrelated.

The likelihood is computed using the innovations $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_n$, defined by (6.24),

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t.$$

The innovations form of the likelihood function, which was first given by Schweppe (1965), is obtained using an argument similar to the one leading to (3.116) and proceeds by noting the innovations are independent Gaussian random vectors with zero means and, as shown in (6.25), covariance matrices

$$\Sigma_t = A_t P_t^{t-1} A_t' + R. \tag{6.61}$$

Hence, ignoring a constant, we may write the likelihood, $L_Y(\Theta)$, as

$$-\ln L_Y(\Theta) = \frac{1}{2}\sum_{t=1}^{n} \log|\Sigma_t(\Theta)| + \frac{1}{2}\sum_{t=1}^{n} \boldsymbol{\epsilon}_t(\Theta)'\Sigma_t(\Theta)^{-1}\boldsymbol{\epsilon}_t(\Theta), \tag{6.62}$$

where we have emphasized the dependence of the innovations on the parameters Θ. Of course, (6.62) is a highly nonlinear and complicated function of the unknown parameters. The usual procedure is to fix $\boldsymbol{x}_0$ and then develop a set of recursions for the log likelihood function and its first two derivatives (for

example, Gupta and Mehra, 1974). Then, a Newton–Raphson algorithm (see Example 3.29) can be used successively to update the parameter values until the negative of the log likelihood is minimized. This approach is advocated, for example, by Jones (1980), who developed ARMA estimation by putting the ARMA model in state-space form. For the univariate case, (6.62) is identical, in form, to the likelihood for the ARMA model given in (3.116).

The steps involved in performing a Newton–Raphson estimation procedure are as follows.

(i) Select initial values for the parameters, say, $\Theta^{(0)}$.
(ii) Run the Kalman filter, Property 6.1, using the initial parameter values, $\Theta^{(0)}$, to obtain a set of innovations and error covariances, say, $\{\boldsymbol{\epsilon}_t^{(0)};\ t = 1, \ldots, n\}$ and $\{\Sigma_t^{(0)};\ t = 1, \ldots, n\}$.
(iii) Run one iteration of a Newton–Raphson procedure with $-\ln L_Y(\Theta)$ as the criterion function (refer to Example 3.29 for details), to obtain a new set of estimates, say $\Theta^{(1)}$.
(iv) At iteration j, $(j = 1, 2, \ldots)$, repeat step 2 using $\Theta^{(j)}$ in place of $\Theta^{(j-1)}$ to obtain a new set of innovation values $\{\boldsymbol{\epsilon}_t^{(j)};\ t = 1, \ldots, n\}$ and $\{\Sigma_t^{(j)};\ t = 1, \ldots, n\}$. Then repeat step 3 to obtain a new estimate $\Theta^{(j+1)}$. Stop when the estimates or the likelihood stabilize; for example, stop when the values of $\Theta^{(j+1)}$ differ from $\Theta^{(j)}$, or when $L_Y(\Theta^{(j+1)})$ differs from $L_Y(\Theta^{(j)})$, by some predetermined, but small amount.

Example 6.6 Newton–Raphson for Example 6.3

In this example, we generated $n = 100$ observations, $y_1, \ldots, y_{100}$, using the model in Example 6.3, to perform a Newton–Raphson estimation of the parameters ϕ, σ_w^2, and σ_v^2. In the notation of §6.2, we would have $\Phi = \phi$, $Q = \sigma_w^2$ and $R = \sigma_v^2$. The actual values of the parameters are $\phi = .8$, $\sigma_w^2 = \sigma_v^2 = 1$.

In the simple case of an AR(1) with observational noise, initial estimation can be accomplished using the results of Example 6.3. For example, using (6.16), we set

$$\phi^{(0)} = \widehat{\rho}_y(2)/\widehat{\rho}_y(1).$$

Similarly, from (6.15), $\gamma_x(1) = \gamma_y(1) = \phi\sigma_w^2/(1-\phi^2)$, so that, initially, we set

$$\sigma_w^{2^{(0)}} = (1 - \phi^{(0)^2})\widehat{\gamma}_y(1)/\phi^{(0)}.$$

Finally, using (6.14) we obtain an initial estimate of σ_v^2, namely,

$$\sigma_v^{2^{(0)}} = \widehat{\gamma}_y(0) - [\sigma_w^{2^{(0)}}/(1 - \phi^{(0)^2})].$$

Newton–Raphson estimation was accomplished using the R program `optim`. The code used for this example is given below. In that program, we must provide an evaluation of the function to be minimized, namely,

$-\ln L_Y(\Theta)$. In this case, the function call combines steps 2 and 3, using the current values of the parameters, $\Theta^{(j-1)}$, to obtain first the filtered values, then the innovation values, and then calculating the criterion function, $-\ln L_Y(\Theta^{(j-1)})$, to be minimized. We can also provide analytic forms of the gradient or score vector, $-\partial \ln L_Y(\Theta)/\partial\Theta$, and the Hessian matrix, $-\partial^2 \ln L_Y(\Theta)/\partial\Theta\ \partial\Theta'$, in the optimization routine, or allow the program to calculate these values numerically. In this example, we let the program proceed numerically and we note the need to be cautious when calculating gradients numerically. For better stability, we can also provide an iterative solution for obtaining analytic gradients and Hessians of the log likelihood function; for details, see Problems 6.11 and 6.12 and Gupta and Mehra (1974).

```
1  # Generate Data
2  set.seed(999); num = 100; N = num+1
3  x = arima.sim(n=N, list(ar = .8, sd=1))
4  y = ts(x[-1] + rnorm(num,0,1))
5  # Initial Estimates
6  u = ts.intersect(y, lag(y,-1), lag(y,-2))
7  varu = var(u); coru = cor(u)
8  phi = coru[1,3]/coru[1,2]
9  q = (1-phi^2)*varu[1,2]/phi;  r = varu[1,1] - q/(1-phi^2)
10 (init.par = c(phi, sqrt(q), sqrt(r)))  # = .91, .51, 1.03
11 # Function to evaluate the likelihood
12 Linn=function(para){
13   phi = para[1]; sigw = para[2]; sigv = para[3]
14   Sigma0 = (sigw^2)/(1-phi^2); Sigma0[Sigma0<0]=0
15   kf = Kfilter0(num, y, 1, mu0=0, Sigma0, phi, sigw, sigv)
16   return(kf$like)   }
17 # Estimation   (partial output shown)
18 (est = optim(init.par, Linn, gr=NULL, method="BFGS", hessian=TRUE))
19 SE = sqrt(diag(solve(est$hessian)))
20 cbind(estimate=c(phi=est$par[1],sigw=est$par[2],sigv=est$par[3]),SE)
```

```
          estimate         SE
    phi  0.8137623 0.08060636
    sigw 0.8507863 0.17528895
    sigv 0.8743968 0.14293192
```

As seen from the output, the final estimates, along with their standard errors (in parentheses), are $\widehat{\phi} = .81$ (.08), $\widehat{\sigma}_w = .85$ (.18), $\widehat{\sigma}_v = .87$ (.14). Adding `control=list(trace=1, REPORT=1)` to the `optim` call in line 19 yielded the following results of the estimation procedure:

```
    initial  value 79.601468
    iter   2 value 79.060391
    iter   3 value 79.034121
    iter   4 value 79.032615
    iter   5 value 79.014817
    iter   6 value 79.014453
      final  value 79.014452
```

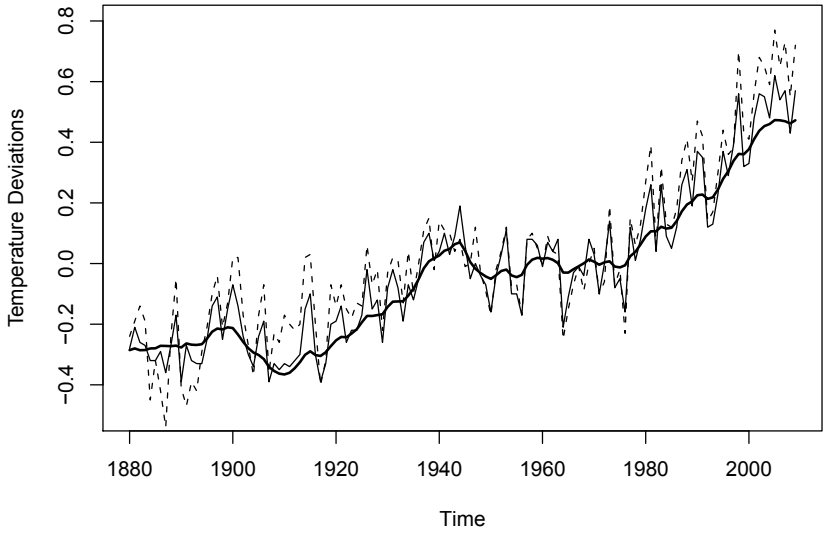

Fig. 6.4. Plot for Example 6.7. The thin solid and dashed lines are the two average global temperature deviations shown in Figure 6.2. The thick solid line is the estimated smoother $\widehat{x}_t^n$.

Note that the algorithm converged in six (or seven?) steps with the final value of the negative of the log likelihood being 79.014452. The standard errors are a byproduct of the estimation procedure, and we will discuss their evaluation later in this section, after Property 6.4.

Example 6.7 Newton–Raphson for Example 6.2

In Example 6.2, we considered two different global temperature series of $n = 130$ observations each, and they are plotted in Figure 6.2. In that example, we argued that both series should be measuring the same underlying climatic signal, x_t, which we model as a random walk with drift,

$$x_t = \delta + x_{t-1} + w_t.$$

Recall that the observation equation was written as

$$\begin{pmatrix} y_{t1} \\ y_{t2} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} x_t + \begin{pmatrix} v_{t1} \\ v_{t2} \end{pmatrix},$$

and the model covariance matrices are given by $Q = q_{11}$ and

$$R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}.$$

Hence, there are five parameters to estimate, δ, the drift, and the variance components, $q_{11}, r_{11}, r_{12}, r_{22}$, noting that $r_{21} = r_{12}$ We hold the the initial state parameters fixed in this example at $\mu_0 = -.26$ and $\Sigma_0 = .01$.

The final estimates are $\widehat{\delta} = .006, \widehat{q}_{11} = .033, \widehat{r}_{11} = .007, \widehat{r}_{12} = .01, \widehat{r}_{22} = .02$, with all values being significant. The observations and the smoothed estimate of the signal, $\widehat{x}_t^n$, are displayed in Figure 6.4. The R code, which uses `Kfilter1` and `Ksmooth1`, is as follows.

```
# Setup
y = cbind(gtemp,gtemp2); num = nrow(y); input = rep(1,num)
A = array(rep(1,2), dim=c(2,1,num))
mu0 = -.26; Sigma0 = .01;  Phi = 1
# Function to Calculate Likelihood
Linn=function(para){
 cQ = para[1]       # sigma_w
  cR1 = para[2]     # 11 element of chol(R)
  cR2 = para[3]     # 22 element of chol(R)
  cR12 = para[4]    # 12 element of chol(R)
 cR = matrix(c(cR1,0,cR12,cR2),2)  # put the matrix together
 drift = para[5]
 kf = Kfilter1(num,y,A,mu0,Sigma0,Phi,drift,0,cQ,cR,input)
 return(kf$like) }
# Estimation
init.par = c(.1,.1,.1,0,.05)  # initial values of parameters
(est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
     control=list(trace=1,REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
# display estimates
u = cbind(estimate=est$par, SE)
rownames(u)=c("sigw","cR11", "cR22", "cR12", "drift"); u
```

```
              estimate          SE
      sigw  0.032730315 0.007473594
      cR11  0.084752492 0.007815219
      cR22  0.070864957 0.005732578
      cR12  0.122458872 0.014867006
     drift  0.005852047 0.002919058
```

```
# Smooth (first set parameters to their final estimates)
cQ=est$par[1]
 cR1=est$par[2]
 cR2=est$par[3]
 cR12=est$par[4]
cR = matrix(c(cR1,0,cR12,cR2), 2)
(R = t(cR)%*%cR)     #  to view the estimated R matrix
drift = est$par[5]
ks = Ksmooth1(num,y,A,mu0,Sigma0,Phi,drift,0,cQ,cR,input)
# Plot
xsmooth = ts(as.vector(ks$xs), start=1880)
plot(xsmooth, lwd=2, ylim=c(-.5,.8), ylab="Temperature Deviations")
lines(gtemp, col="blue", lty=1)   # color helps here
lines(gtemp2, col="red", lty=2)
```

In addition to Newton–Raphson, Shumway and Stoffer (1982) presented a conceptually simpler estimation procedure based on the EM (expectation-maximization) algorithm (Dempster et al., 1977). For the sake of brevity, we ignore the inputs and consider the model in the form of (6.1) and (6.2). The basic idea is that if we could observe the states, $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, in addition to the observations $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, then we would consider $\{X_n, Y_n\}$ as the complete data, with the joint density

$$f_\Theta(X_n, Y_n) = f_{\mu_0,\Sigma_0}(\boldsymbol{x}_0) \prod_{t=1}^{n} f_{\Phi,Q}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \prod_{t=1}^{n} f_R(\boldsymbol{y}_t|\boldsymbol{x}_t). \tag{6.63}$$

Under the Gaussian assumption and ignoring constants, the complete data likelihood, (6.63), can be written as

$$\begin{aligned} -2\ln L_{X,Y}(\Theta) = {} & \ln|\Sigma_0| + (\boldsymbol{x}_0 - \boldsymbol{\mu}_0)'\Sigma_0^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_0) \\ & + n\ln|Q| + \sum_{t=1}^{n}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})'Q^{-1}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1}) \\ & + n\ln|R| + \sum_{t=1}^{n}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)'R^{-1}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t). \end{aligned} \tag{6.64}$$

Thus, in view of (6.64), if we did have the complete data, we could then use the results from multivariate normal theory to easily obtain the MLEs of Θ. We do not have the complete data; however, the EM algorithm gives us an iterative method for finding the MLEs of Θ based on the incomplete data, Y_n, by successively maximizing the conditional expectation of the complete data likelihood. To implement the EM algorithm, we write, at iteration j, $(j = 1, 2, \ldots)$,

$$Q\left(\Theta \mid \Theta^{(j-1)}\right) = E\left\{-2\ln L_{X,Y}(\Theta) \mid Y_n, \Theta^{(j-1)}\right\}. \tag{6.65}$$

Calculation of (6.65) is the expectation step. Of course, given the current value of the parameters, $\Theta^{(j-1)}$, we can use Property 6.2 to obtain the desired conditional expectations as smoothers. This property yields

$$\begin{aligned} Q\left(\Theta \mid \Theta^{(j-1)}\right) = {} & \ln|\Sigma_0| + \mathrm{tr}\left\{\Sigma_0^{-1}[P_0^n + (\boldsymbol{x}_0^n - \boldsymbol{\mu}_0)(\boldsymbol{x}_0^n - \boldsymbol{\mu}_0)']\right\} \\ & + n\ln|Q| + \mathrm{tr}\left\{Q^{-1}[S_{11} - S_{10}\Phi' - \Phi S_{10}' + \Phi S_{00}\Phi']\right\} \\ & + n\ln|R| \\ & + \mathrm{tr}\Big\{R^{-1}\sum_{t=1}^{n}[(\boldsymbol{y}_t - A_t\boldsymbol{x}_t^n)(\boldsymbol{y}_t - A_t\boldsymbol{x}_t^n)' + A_tP_t^nA_t']\Big\}, \end{aligned} \tag{6.66}$$

where

$$S_{11} = \sum_{t=1}^{n}(\boldsymbol{x}_t^n\boldsymbol{x}_t^{n\prime} + P_t^n), \tag{6.67}$$

$$S_{10} = \sum_{t=1}^{n} (\boldsymbol{x}_t^n \boldsymbol{x}_{t-1}^{n\prime} + P_{t,t-1}^n), \tag{6.68}$$

and

$$S_{00} = \sum_{t=1}^{n} (\boldsymbol{x}_{t-1}^n \boldsymbol{x}_{t-1}^{n\prime} + P_{t-1}^n). \tag{6.69}$$

In (6.66)-(6.69), the smoothers are calculated under the current value of the parameters $\Theta^{(j-1)}$; for simplicity, we have not explicitly displayed this fact.

Minimizing (6.66) with respect to the parameters, at iteration j, constitutes the maximization step, and is analogous to the usual multivariate regression approach, which yields the updated estimates

$$\Phi^{(j)} = S_{10} S_{00}^{-1}, \tag{6.70}$$

$$Q^{(j)} = n^{-1} \left(S_{11} - S_{10} S_{00}^{-1} S_{10}' \right), \tag{6.71}$$

and

$$R^{(j)} = n^{-1} \sum_{t=1}^{n} [(\boldsymbol{y}_t - A_t \boldsymbol{x}_t^n)(\boldsymbol{y}_t - A_t \boldsymbol{x}_t^n)' + A_t P_t^n A_t']. \tag{6.72}$$

The updates for the initial mean and variance–covariance matrix are

$$\boldsymbol{\mu}_0^{(j)} = \boldsymbol{x}_0^n \quad \text{and} \quad \Sigma_0^{(j)} = P_0^n \tag{6.73}$$

obtained from minimizing (6.66).

The overall procedure can be regarded as simply alternating between the Kalman filtering and smoothing recursions and the multivariate normal maximum likelihood estimators, as given by (6.70)–(6.73). Convergence results for the EM algorithm under general conditions can be found in Wu (1983). We summarize the iterative procedure as follows.

(i) Initialize the procedure by selecting starting values for the parameters $\Theta^{(0)} = \{\boldsymbol{\mu}_0, \Sigma_0, \Phi, Q, R\}$.

On iteration j, $(j = 1, 2, \ldots)$:

(ii) Compute the incomplete-data likelihood, $-\ln L_Y(\Theta^{(j-1)})$; see equation (6.62).

(iii) Perform the E-Step. Use Properties 6.1, 6.2, and 6.3 to obtain the smoothed values $\boldsymbol{x}_t^n, P_t^n$ and $P_{t,t-1}^n$, for $t = 1, \ldots, n$, using the parameters $\Theta^{(j-1)}$. Use the smoothed values to calculate S_{11}, S_{10}, S_{00} given in (6.67)–(6.69).

(iv) Perform the M-Step. Update the estimates, $\boldsymbol{\mu}_0, \Sigma_0, \Phi, Q$, and R using (6.70)–(6.73), to obtain $\Theta^{(j)}$.

(v) Repeat Steps (ii) – (iv) to convergence.

Example 6.8 EM Algorithm for Example 6.3

Using the same data generated in Example 6.6, we performed an EM algorithm estimation of the parameters ϕ, σ_w^2 and σ_v^2 as well as the initial parameters μ_0 and Σ_0 using the script `EM0`. The convergence rate of the EM algorithm compared with the Newton–Raphson procedure is slow. In this example, with convergence being claimed when the relative change in the log likelihood is less than .00001; convergence was attained after 41 iterations. The final estimates, along with their standard errors (in parentheses), were

$$\widehat{\phi} = .80\ (.08), \quad \widehat{\sigma}_w = .87\ (.17), \quad \widehat{\sigma}_v = .84\ (.14),$$

with $\widehat{\mu}_0 = -1.98$ and $\widehat{\Sigma}_0 = .03$. Evaluation of the standard errors used a call to `fdHess` in the `nlme` R package to evaluate the Hessian at the final estimates. The `nlme` package must be loaded prior to the call to `fdHess`.

```
library(nlme)   # loads package nlme
# Generate data (same as Example 6.6)
set.seed(999); num = 100; N = num+1
x = arima.sim(n=N, list(ar = .8, sd=1))
y = ts(x[-1] + rnorm(num,0,1))
# Initial Estimates
u = ts.intersect(y, lag(y,-1), lag(y,-2))
varu = var(u); coru = cor(u)
phi = coru[1,3]/coru[1,2]
q = (1-phi^2)*varu[1,2]/phi
r = varu[1,1] - q/(1-phi^2)
cr = sqrt(r); cq = sqrt(q); mu0 = 0; Sigma0 = 2.8
# EM procedure - output not shown
(em = EM0(num, y, 1, mu0, Sigma0, phi, cq, cr, 75, .00001))
# Standard Errors  (this uses nlme)
phi = em$Phi; cq = chol(em$Q); cr = chol(em$R)
mu0 = em$mu0; Sigma0 = em$Sigma0
para = c(phi, cq, cr)
Linn = function(para){  # to evaluate likelihood at estimates
  kf = Kfilter0(num, y, 1, mu0, Sigma0, para[1], para[2], para[3])
  return(kf$like)  }
emhess = fdHess(para, function(para) Linn(para))
SE = sqrt(diag(solve(emhess$Hessian)))
# Display Summary of Estimation
estimate = c(para, em$mu0, em$Sigma0); SE = c(SE, NA, NA)
u = cbind(estimate, SE)
rownames(u) = c("phi","sigw","sigv","mu0","Sigma0"); u
```

```
              estimate         SE
phi     0.80639903 0.07986272
sigw    0.86442634 0.16719703
sigv    0.84276381 0.13805072
mu0    -1.96010956         NA
Sigma0  0.03638596         NA
```

Asymptotic Distribution of the MLEs

The asymptotic distribution of estimators of the model parameters, say, $\widehat{\Theta}_n$, is studied extensively in Caines (1988, Chapters 7 and 8), and in Hannan and Deistler (1988, Chapter 4). In both of these references, the consistency and asymptotic normality of the estimators is established under general conditions. Although we will only state the basic result, some crucial elements are needed to establish large sample properties of the estimators. An essential condition is the stability of the filter. Stability of the filter assures that, for large t, the innovations $\boldsymbol{\epsilon}_t$ are basically copies of each other (that is, independent and identically distributed) with a stable covariance matrix Σ that does not depend on t and that, asymptotically, the innovations contain all of the information about the unknown parameters. Although it is not necessary, for simplicity, we shall assume here that $A_t \equiv A$ for all t. Details on departures from this assumption can be found in Jazwinski (1970, Sections 7.6 and 7.8). We also drop the inputs and use the model in the form of (6.1) and (6.2).

For stability of the filter, we assume the eigenvalues of Φ are less than one in absolute value; this assumption can be weakened (for example, see Harvey, 1991, Section 4.3), but we retain it for simplicity. This assumption is enough to ensure the stability of the filter in that, as $t \to \infty$, the filter error covariance matrix P_t^t converges to P, the steady-state error covariance matrix, the gain matrix K_t converges to K, the steady-state gain matrix, from which it follows that the innovation variance–covariance matrix Σ_t converges to Σ, the steady-state variance–covariance matrix of the stable innovations; details can be found in Jazwinski (1970, Sections 7.6 and 7.8) and Anderson and Moore (1979, Section 4.4). In particular, the steady-state filter error covariance matrix, P, satisfies the Riccati equation:

$$P = \Phi[P - PA'(APA' + R)^{-1}AP]\Phi' + Q;$$

the steady-state gain matrix satisfies $K = PA'[APA' + R]^{-1}$. In Example 6.5, for all practical purposes, stability was reached by the fourth observation.

When the process is in steady-state, we may consider $\boldsymbol{x}_{t+1}^t$ as the steady-state predictor and interpret it as $\boldsymbol{x}_{t+1}^t = E(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \ldots)$. As can be seen from (6.19) and (6.21), the steady-state predictor can be written as

$$\begin{aligned} \boldsymbol{x}_{t+1}^t &= \Phi[I - KA]\boldsymbol{x}_t^{t-1} + \Phi K y_t \\ &= \Phi \boldsymbol{x}_t^{t-1} + \Phi K \boldsymbol{\epsilon}_t, \end{aligned} \tag{6.74}$$

where $\boldsymbol{\epsilon}_t$ is the steady-state innovation process given by

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - E(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \boldsymbol{y}_{t-2}, \ldots).$$

In the Gaussian case, $\boldsymbol{\epsilon}_t \sim$ iid $\mathrm{N}(\mathbf{0}, \Sigma)$, where $\Sigma = APA' + R$. In steady-state, the observations can be written as

$$\boldsymbol{y}_t = A\boldsymbol{x}_t^{t-1} + \boldsymbol{\epsilon}_t. \tag{6.75}$$

Together, (6.74) and (6.75) make up the steady-state innovations form of the dynamic linear model.

In the following property, we assume the Gaussian state-space model (6.1) and (6.2), is time invariant, i.e., $A_t \equiv A$, the eigenvalues of Φ are within the unit circle and the model has the smallest possible dimension (see Hannan and Diestler, 1988, Section 2.3 for details). We denote the true parameters by Θ_0, and we assume the dimension of Θ_0 is the dimension of the parameter space. Although it is not necessary to assume $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are Gaussian, certain additional conditions would have to apply and adjustments to the asymptotic covariance matrix would have to be made (see Caines, 1988, Chapter 8).

Property 6.4 Asymptotic Distribution of the Estimators

Under general conditions, let $\widehat{\Theta}_n$ be the estimator of Θ_0 obtained by maximizing the innovations likelihood, $L_Y(\Theta)$, as given in (6.62). Then, as $n \to \infty$,

$$\sqrt{n}\left(\widehat{\Theta}_n - \Theta_0\right) \xrightarrow{d} \mathrm{N}\left[0,\ \mathcal{I}(\Theta_0)^{-1}\right],$$

where $\mathcal{I}(\Theta)$ is the asymptotic information matrix given by

$$\mathcal{I}(\Theta) = \lim_{n\to\infty}\ n^{-1}E\left[-\partial^2 \ln L_Y(\Theta)/\partial\Theta\ \partial\Theta'\right].$$

Precise details and the proof of Property 6.4 are given in Caines (1988, Chapter 7) and in Hannan and Deistler (1988, Chapter 4). For a Newton procedure, the Hessian matrix (as described in Example 6.6) at the time of convergence can be used as an estimate of $n\mathcal{I}(\Theta_0)$ to obtain estimates of the standard errors. In the case of the EM algorithm, no derivatives are calculated, but we may include a numerical evaluation of the Hessian matrix at the time of convergence to obtain estimated standard errors. Also, extensions of the EM algorithm exist, such as the SEM algorithm (Meng and Rubin, 1991), that include a procedure for the estimation of standard errors. In the examples of this section, the estimated standard errors were obtained from the numerical Hessian matrix of $-\ln L_Y(\widehat{\Theta})$, where $\widehat{\Theta}$ is the vector of parameters estimates at the time of convergence.

6.4 Missing Data Modifications

An attractive feature available within the state-space framework is its ability to treat time series that have been observed irregularly over time. For example, Jones (1980) used the state-space representation to fit ARMA models to series with missing observations, and Palma and Chan (1997) used the model for estimation and forecasting of ARFIMA series with missing observations. Shumway and Stoffer (1982) described the modifications necessary to fit multivariate state-space models via the EM algorithm when data are missing. We will discuss the procedure in detail in this section. Throughout this section, for notational simplicity, we assume the model is of the form (6.1) and (6.2).

Suppose, at a given time t, we define the partition of the $q \times 1$ observation vector $\boldsymbol{y}_t = (\boldsymbol{y}_t^{(1)'}, \boldsymbol{y}_t^{(2)'})'$, where the first $q_{1t} \times 1$ component is observed and the second $q_{2t} \times 1$ component is unobserved, $q_{1t} + q_{2t} = q$. Then, write the partitioned observation equation

$$\begin{pmatrix} \boldsymbol{y}_t^{(1)} \\ \boldsymbol{y}_t^{(2)} \end{pmatrix} = \begin{bmatrix} A_t^{(1)} \\ A_t^{(2)} \end{bmatrix} \boldsymbol{x}_t + \begin{pmatrix} \boldsymbol{v}_t^{(1)} \\ \boldsymbol{v}_t^{(2)} \end{pmatrix}, \tag{6.76}$$

where $A_t^{(1)}$ and $A_t^{(2)}$ are, respectively, the $q_{1t} \times p$ and $q_{2t} \times p$ partitioned observation matrices, and

$$\operatorname{cov} \begin{pmatrix} \boldsymbol{v}_t^{(1)} \\ \boldsymbol{v}_t^{(2)} \end{pmatrix} = \begin{bmatrix} R_{11t} & R_{12t} \\ R_{21t} & R_{22t} \end{bmatrix} \tag{6.77}$$

denotes the covariance matrix of the measurement errors between the observed and unobserved parts.

In the missing data case where $\boldsymbol{y}_t^{(2)}$ is not observed, we may modify the observation equation in the DLM, (6.1)-(6.2), so that the model is

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \boldsymbol{w}_t \quad \text{and} \quad \boldsymbol{y}_t^{(1)} = A_t^{(1)} \boldsymbol{x}_t + \boldsymbol{v}_t^{(1)}, \tag{6.78}$$

where now, the observation equation is q_{1t}-dimensional at time t. In this case, it follows directly from Corollary 6.1 that the filter equations hold with the appropriate notational substitutions. If there are no observations at time t, then set the gain matrix, K_t, to the $p \times q$ zero matrix in Property 6.1, in which case $\boldsymbol{x}_t^t = \boldsymbol{x}_t^{t-1}$ and $P_t^t = P_t^{t-1}$.

Rather than deal with varying observational dimensions, it is computationally easier to modify the model by zeroing out certain components and retaining a q-dimensional observation equation throughout. In particular, Corollary 6.1 holds for the missing data case if, at update t, we substitute

$$\boldsymbol{y}_{(t)} = \begin{pmatrix} \boldsymbol{y}_t^{(1)} \\ \mathbf{0} \end{pmatrix}, \quad A_{(t)} = \begin{bmatrix} A_t^{(1)} \\ 0 \end{bmatrix}, \quad R_{(t)} = \begin{bmatrix} R_{11t} & 0 \\ 0 & I_{22t} \end{bmatrix}, \tag{6.79}$$

for $\boldsymbol{y}_t$, A_t, and R, respectively, in (6.21)–(6.23), where I_{22t} is the $q_{2t} \times q_{2t}$ identity matrix. With the substitutions (6.79), the innovation values (6.24) and (6.25) will now be of the form

$$\boldsymbol{\epsilon}_{(t)} = \begin{pmatrix} \boldsymbol{\epsilon}_t^{(1)} \\ \mathbf{0} \end{pmatrix}, \qquad \Sigma_{(t)} = \begin{bmatrix} A_t^{(1)} P_t^{t-1} A_t^{(1)'} + R_{11t} & 0 \\ 0 & I_{22t} \end{bmatrix}, \tag{6.80}$$

so that the innovations form of the likelihood given in (6.62) is correct for this case. Hence, with the substitutions in (6.79), maximum likelihood estimation via the innovations likelihood can proceed as in the complete data case.

Once the missing data filtered values have been obtained, Stoffer (1982) also established the smoother values can be processed using Properties 6.2

and 6.3 with the values obtained from the missing data-filtered values. In the missing data case, the state estimators are denoted

$$\boldsymbol{x}_t^{(s)} = E\left(\boldsymbol{x}_t \mid \boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_s^{(1)}\right), \tag{6.81}$$

with error variance–covariance matrix

$$P_t^{(s)} = E\left\{\left(\boldsymbol{x}_t - \boldsymbol{x}_t^{(s)}\right)\left(\boldsymbol{x}_t - \boldsymbol{x}_t^{(s)}\right)'\right\}. \tag{6.82}$$

The missing data lag-one smoother covariances will be denoted by $P_{t,t-1}^{(n)}$.

The maximum likelihood estimators in the EM procedure require further modifications for the case of missing data. Now, we consider

$$Y_n^{(1)} = \{\boldsymbol{y}_1^{(1)}, \ldots, \boldsymbol{y}_n^{(1)}\} \tag{6.83}$$

as the incomplete data, and X_n, Y_n, as defined in (6.63), as the complete data. In this case, the complete data likelihood, (6.63), or equivalently (6.64), is the same, but to implement the E-step, at iteration j, we must calculate

$$\begin{aligned} Q\big(\Theta \mid \Theta^{(j-1)}\big) &= E\big\{-2\ln L_{X,Y}(\Theta) \mid Y_n^{(1)}, \Theta^{(j-1)}\big\} \\ &= E_*\Big\{\ln|\Sigma_0| + \operatorname{tr}\ \Sigma_0^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_0)(\boldsymbol{x}_0 - \boldsymbol{\mu}_0)' \mid Y_n^{(1)}\Big\} \\ &+ E_*\Big\{n\ln|Q| + \sum_{t=1}^{n} \operatorname{tr}\ \left[Q^{-1}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})'\right] \mid Y_n^{(1)}\Big\} \\ &+ E_*\Big\{n\ln|R| + \sum_{t=1}^{n} \operatorname{tr}\ \left[R^{-1}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)'\right] \mid Y_n^{(1)}\Big\}, \end{aligned} \tag{6.84}$$

where E_* denotes the conditional expectation under $\Theta^{(j-1)}$ and tr denotes trace. The first two terms in (6.84) will be like the first two terms of (6.66) with the smoothers $\boldsymbol{x}_t^n$, P_t^n, and $P_{t,t-1}^n$ replaced by their missing data counterparts, $\boldsymbol{x}_t^{(n)}$, $P_t^{(n)}$, and $P_{t,t-1}^{(n)}$. What changes in the missing data case is the third term of (6.84), where we must evaluate $E_*(\boldsymbol{y}_t^{(2)} \mid Y_n^{(1)})$ and $E_*(\boldsymbol{y}_t^{(2)}\boldsymbol{y}_t^{(2)'} \mid Y_n^{(1)})$. In Stoffer (1982), it is shown that

$$\begin{aligned} &E_*\left\{(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)' \mid Y_n^{(1)}\right\} \\ &\quad= \begin{pmatrix} \boldsymbol{y}_t^{(1)} - A_t^{(1)}\boldsymbol{x}_t^{(n)} \\ R_{*21t}R_{*11t}^{-1}(\boldsymbol{y}_t^{(1)} - A_t^{(1)}\boldsymbol{x}_t^{(n)}) \end{pmatrix}\begin{pmatrix} \boldsymbol{y}_t^{(1)} - A_t^{(1)}\boldsymbol{x}_t^{(n)} \\ R_{*21t}R_{*11t}^{-1}(\boldsymbol{y}_t^{(1)} - A_t^{(1)}\boldsymbol{x}_t^{(n)}) \end{pmatrix}' \\ &\quad+ \begin{pmatrix} A_t^{(1)} \\ R_{*21t}R_{*11t}^{-1}A_t^{(1)} \end{pmatrix} P_t^{(n)} \begin{pmatrix} A_t^{(1)} \\ R_{*21t}R_{*11t}^{-1}A_t^{(1)} \end{pmatrix}' \\ &\quad+ \begin{pmatrix} 0 & 0 \\ 0 & R_{*22t} - R_{*21t}R_{*11t}^{-1}R_{*12t} \end{pmatrix}. \end{aligned} \tag{6.85}$$

In (6.85), the values of R_{*ikt}, for $i, k = 1, 2$, are the current values specified by $\Theta^{(j-1)}$. In addition, $\boldsymbol{x}_t^{(n)}$ and $P_t^{(n)}$ are the values obtained by running the smoother under the current parameter estimates specified by $\Theta^{(j-1)}$.

In the case in which observed and unobserved components have uncorrelated errors, that is, R_{*12t} is the zero matrix, (6.85) can be simplified to

$$E_*\big\{(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)' \mid Y_n^{(1)}\big\}$$
$$= \big(\boldsymbol{y}_{(t)} - A_{(t)}\boldsymbol{x}_t^{(n)}\big)\big(\boldsymbol{y}_{(t)} - A_{(t)}\boldsymbol{x}_t^{(n)}\big)' + A_{(t)}P_t^{(n)}A_{(t)}' + \begin{pmatrix} 0 & 0 \\ 0 & R_{*22t} \end{pmatrix}, \quad (6.86)$$

where $\boldsymbol{y}_{(t)}$ and $A_{(t)}$ are defined in (6.79).

In this simplified case, the missing data M-step looks like the M-step given in (6.67)-(6.73). That is, with

$$S_{(11)} = \sum_{t=1}^{n}(\boldsymbol{x}_t^{(n)}\boldsymbol{x}_t^{(n)\prime} + P_t^{(n)}), \quad (6.87)$$

$$S_{(10)} = \sum_{t=1}^{n}(\boldsymbol{x}_t^{(n)}\boldsymbol{x}_{t-1}^{(n)\prime} + P_{t,t-1}^{(n)}), \quad (6.88)$$

and

$$S_{(00)} = \sum_{t=1}^{n}(\boldsymbol{x}_{t-1}^{(n)}\boldsymbol{x}_{t-1}^{(n)\prime} + P_{t-1}^{(n)}), \quad (6.89)$$

where the smoothers are calculated under the present value of the parameters $\Theta^{(j-1)}$ using the missing data modifications, at iteration j, the *maximization step* is

$$\Phi^{(j)} = S_{(10)}S_{(00)}^{-1}, \quad (6.90)$$

$$Q^{(j)} = n^{-1}\left(S_{(11)} - S_{(10)}S_{(00)}^{-1}S_{(10)}'\right), \quad (6.91)$$

and

$$R^{(j)} = n^{-1}\sum_{t=1}^{n} D_t\Big\{\Big(\boldsymbol{y}_{(t)} - A_{(t)}\boldsymbol{x}_t^{(n)}\Big)\Big(\boldsymbol{y}_{(t)} - A_{(t)}\boldsymbol{x}_t^{(n)}\Big)' + A_{(t)}P_t^{(n)}A_{(t)}' + \begin{pmatrix} 0 & 0 \\ 0 & R_{22t}^{(j-1)} \end{pmatrix}\Big\}D_t, \quad (6.92)$$

where D_t is a permutation matrix that reorders the variables at time t in their original order and $\boldsymbol{y}_{(t)}$ and $A_{(t)}$ are defined in (6.79). For example, suppose $q = 3$ and at time t, y_{t2} is missing. Then,

$$\boldsymbol{y}_{(t)} = \begin{pmatrix} y_{t1} \\ y_{t3} \\ 0 \end{pmatrix}, \quad A_{(t)} = \begin{bmatrix} A_{t1} \\ A_{t3} \\ \mathbf{0}' \end{bmatrix}, \quad \text{and} \quad D_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

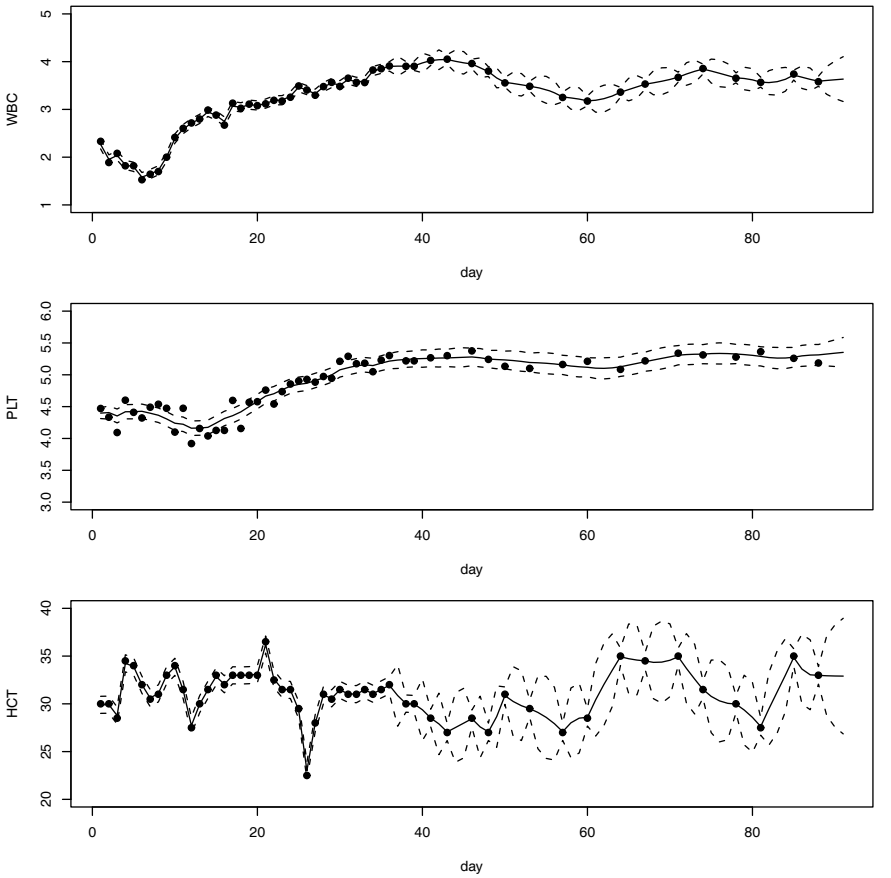

Fig. 6.5. Smoothed values for various components in the blood parameter tracking problem. The actual data are shown as points, the smoothed values are shown as solid lines, and ±3 standard error bounds are shown as dashed lines.

where A_{ti} is the ith row of A_t and $\mathbf{0}'$ is a $1 \times p$ vector of zeros. In (6.92), only R_{11t} gets updated, and R_{22t} at iteration j is simply set to its value from the previous iteration, $j-1$. Of course, if we cannot assume $R_{12t} = 0$, (6.92) must be changed accordingly using (6.85), but (6.90) and (6.91) remain the same. As before, the parameter estimates for the initial state are updated as

$$\boldsymbol{\mu}_0^{(j)} = \boldsymbol{x}_0^{(n)} \quad \text{and} \quad \Sigma_0^{(j)} = P_0^{(n)}. \tag{6.93}$$

Example 6.9 Longitudinal Biomedical Data

We consider the biomedical data in Example 6.1, which have portions of the three-dimensional vector missing after the 40th day. The maximum likelihood procedure yielded the estimators

$$\widehat{\Phi} = \begin{pmatrix} .970 & -.022 & .007 \\ .057 & .927 & .006 \\ -1.342 & 2.190 & .792 \end{pmatrix}, \quad \widehat{Q} = \begin{pmatrix} .018 & -.002 & .018 \\ -.002 & .003 & .028 \\ .018 & .028 & 4.10 \end{pmatrix},$$

and $\widehat{R} = \text{diag}\{.003, .017, .342\}$ for the transition, state error covariance and observation error covariance matrices, respectively. The coupling between the first and second series is relatively weak, whereas the third series HCT is strongly related to the first two; that is,

$$\widehat{x}_{t3} = -1.342x_{t-1,1} + 2.190x_{t-1,2} + .792x_{t-1,3}.$$

Hence, the HCT is negatively correlated with white blood count (WBC) and positively correlated with platelet count (PLT). Byproducts of the procedure are estimated trajectories for all three longitudinal series and their respective prediction intervals. In particular, Figure 6.5 shows the data as points, the estimated smoothed values $\widehat{x}_t^{(n)}$ as solid lines, and error bounds, $\widehat{x}_t^{(n)} \pm 2\sqrt{\widehat{P}_t^{(n)}}$ as dotted lines, for critical post-transplant platelet count.

In the following R code we use the script `EM1`. In this case the observation matrices A_t are either the identity or the zero matrix because either all the series are observed or none of them are observed.

```
y = cbind(WBC, PLT, HCT)
num = nrow(y)
A = array(0, dim=c(3,3,num))  # make array of obs matrices
 for(k in 1:num) if (y[k,1] > 0) A[,,k]= diag(1,3)
# Initial values
mu0 = matrix(0, 3, 1)
Sigma0 = diag(c(.1, .1, 1), 3)
Phi = diag(1, 3)
cQ = diag(c(.1, .1, 1), 3)
cR = diag(c(.1, .1, 1), 3)
# EM procedure - output not shown
(em = EM1(num, y, A, mu0, Sigma0, Phi, 0, 0, cQ, cR, 0, 100, .001))
# Graph smoother
ks = Ksmooth1(num, y, A, em$mu0, em$Sigma0, em$Phi, 0, 0,
     chol(em$Q), chol(em$R), 0)
y1s = ks$xs[1,,]; y2s = ks$xs[2,,]; y3s = ks$xs[3,,]
p1 = 2*sqrt(ks$Ps[1,1,])
p2 = 2*sqrt(ks$Ps[2,2,])
p3 = 2*sqrt(ks$Ps[3,3,])
par(mfrow=c(3,1), mar=c(4,4,1,1)+.2)
plot(WBC, type="p", pch=19, ylim=c(1,5), xlab="day")
 lines(y1s); lines(y1s+p1, lty=2); lines(y1s-p1, lty=2)
plot(PLT, type="p",  ylim=c(3,6), pch=19, xlab="day")
 lines(y2s); lines(y2s+p2, lty=2); lines(y2s-p2, lty=2)
plot(HCT, type="p", pch=19, ylim=c(20,40), xlab="day")
 lines(y3s); lines(y3s+p3, lty=2); lines(y3s-p3, lty=2)
```

6.5 Structural Models: Signal Extraction and Forecasting

In order to develop computing techniques for handling a versatile cross section of possible models, it is necessary to restrict the state-space model somewhat, and we consider one possible class of specializations in this section. The components of the model are taken as linear processes that can be adapted to represent fixed and disturbed trends and periodicities as well as classical autoregressions. The observed series is regarded as being a sum of component signal series. To illustrate the possibilities, we consider an example that shows how to fit a sum of trend, seasonal, and irregular components to the quarterly earnings data that we have considered before.

Example 6.10 Johnson & Johnson Quarterly Earnings

Consider the quarterly earnings series from the U.S. company Johnson & Johnson as given in Figure 1.1. The series is highly nonstationary, and there is both a trend signal that is gradually increasing over time and a seasonal component that cycles every four quarters or once per year. The seasonal component is getting larger over time as well. Transforming into logarithms or even taking the nth root does not seem to make the series stationary, as there is a slight bend to the transformed curve. Suppose, however, we consider the series to be the sum of a trend component, a seasonal component, and a white noise. That is, let the observed series be expressed as

$$y_t = T_t + S_t + v_t, \tag{6.94}$$

where T_t is trend and S_t is the seasonal component. Suppose we allow the trend to increase exponentially; that is,

$$T_t = \phi T_{t-1} + w_{t1}, \tag{6.95}$$

where the coefficient $\phi > 1$ characterizes the increase. Let the seasonal component be modeled as

$$S_t + S_{t-1} + S_{t-2} + S_{t-3} = w_{t2}, \tag{6.96}$$

which corresponds to assuming the seasonal component is expected to sum to zero over a complete period or four quarters. To express this model in state-space form, let $\boldsymbol{x}_t = (T_t, S_t, S_{t-1}, S_{t-2})'$ be the state vector so the observation equation (6.2) can be written as

$$y_t = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{pmatrix} + v_t,$$

with the state equation written as

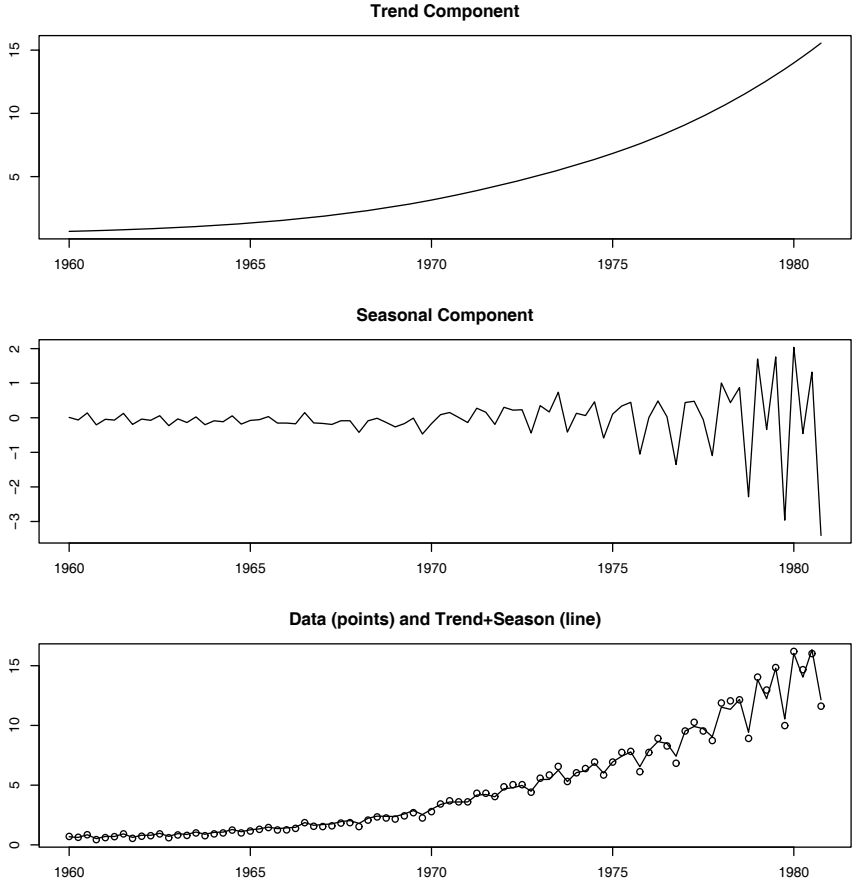

Fig. 6.6. Estimated trend component, T_t^n (top), estimated seasonal component, S_t^n (middle), and the Johnson and Johnson quarterly earnings series with $T_t^n + S_t^n$ superimposed (bottom).

$$\begin{pmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{pmatrix} = \begin{pmatrix} \phi & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} T_{t-1} \\ S_{t-1} \\ S_{t-2} \\ S_{t-3} \end{pmatrix} + \begin{pmatrix} w_{t1} \\ w_{t2} \\ 0 \\ 0 \end{pmatrix},$$

where $R = r_{11}$ and

$$Q = \begin{pmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The model reduces to state-space form, (6.1) and (6.2), with $p = 4$ and $q = 1$. The parameters to be estimated are r_{11}, the noise variance in the measurement equations, q_{11} and q_{22}, the model variances corresponding to

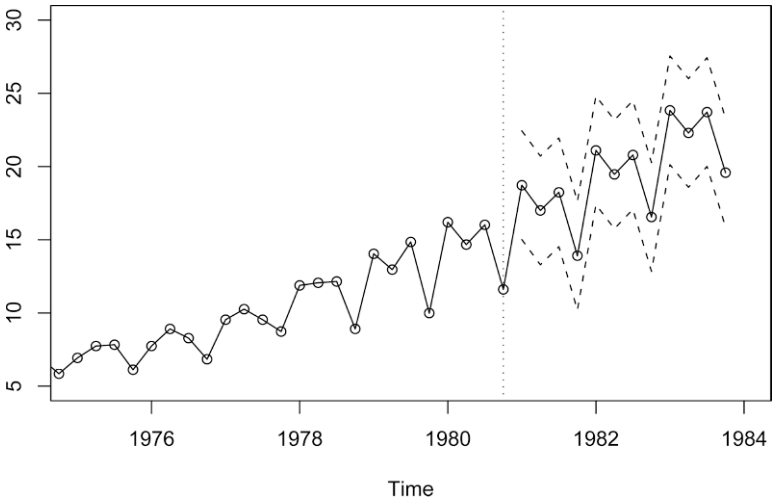

Fig. 6.7. A 12-quarter forecast for the Johnson & Johnson quarterly earnings series. The forecasts are shown as a continuation of the data (points connected by a solid line). The dashed lines indicate the upper and lower 95% prediction intervals.

the trend and seasonal components and ϕ, the transition parameter that models the growth rate. Growth is about 3% per year, and we began with $\phi = 1.03$. The initial mean was fixed at $\boldsymbol{\mu}_0 = (.7, 0, 0, 0)'$, with uncertainty modeled by the diagonal covariance matrix with $\Sigma_{0ii} = .04$, for $i = 1, \ldots, 4$. Initial state covariance values were taken as $q_{11} = .01, q_{22} = .01,$. The measurement error covariance was started at $r_{11} = .25$.

After about 20 iterations of a Newton–Raphson, the transition parameter estimate was $\widehat{\phi} = 1.035$, corresponding to exponential growth with inflation at about 3.5% per year. The measurement uncertainty was small at $\sqrt{\widehat{r}_{11}} = .0005$, compared with the model uncertainties $\sqrt{\widehat{q}_{11}} = .1397$ and $\sqrt{\widehat{q}_{22}} = .2209$. Figure 6.6 shows the smoothed trend estimate and the exponentially increasing seasonal components. We may also consider forecasting the Johnson & Johnson series, and the result of a 12-quarter forecast is shown in Figure 6.7 as basically an extension of the latter part of the observed data.

This example uses the `Kfilter0` and `Ksmooth0` scripts as follows.

```
num = length(jj); A = cbind(1, 1, 0, 0)
# Function to Calculate Likelihood
Linn=function(para){
 Phi = diag(0,4); Phi[1,1] = para[1]
 Phi[2,]=c(0,-1,-1,-1); Phi[3,]=c(0, 1, 0, 0); Phi[4,]=c(0, 0, 1, 0)
 cQ1 = para[2]; cQ2 = para[3]; cR = para[4]  # sqrt of q11, q22, r11
 cQ=diag(0,4); cQ[1,1]=cQ1; cQ[2,2]=cQ2;
 kf = Kfilter0(num, jj, A, mu0, Sigma0, Phi, cQ, cR)
 return(kf$like)  }
```

```
# Initial Parameters
mu0 = c(.7, 0, 0, 0);  Sigma0 = diag(.04, 4)
init.par = c(1.03, .1, .1, .5)   # Phi[1,1], the 2 Qs and R
# Estimation
est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
     control=list(trace=1,REPORT=1))
SE = sqrt(diag(solve(est$hessian)))
u = cbind(estimate=est$par,SE)
rownames(u)=c("Phi11","sigw1","sigw2","sigv"); u
# Smooth
Phi = diag(0,4); Phi[1,1] = est$par[1]
Phi[2,]=c(0,-1,-1,-1); Phi[3,]=c(0,1,0,0); Phi[4,]=c(0,0,1,0)
cQ1 = est$par[2]; cQ2 = est$par[3]; cR = est$par[4]
cQ = diag(1,4); cQ[1,1]=cQ1; cQ[2,2]=cQ2
ks = Ksmooth0(num, jj, A, mu0, Sigma0, Phi, cQ, cR)
# Plot
Tsm = ts(ks$xs[1,,], start=1960, freq=4)
Ssm = ts(ks$xs[2,,], start=1960, freq=4)
p1 = 2*sqrt(ks$Ps[1,1,]); p2 = 2*sqrt(ks$Ps[2,2,])
par(mfrow=c(3,1))
plot(Tsm, main="Trend Component", ylab="Trend")
 lines(Tsm+p1, lty=2, col=4); lines(Tsm-p1,lty=2, col=4)
plot(Ssm, main="Seasonal Component", ylim=c(-5,4), ylab="Season")
 lines(Ssm+p2,lty=2, col=4); lines(Ssm-p2,lty=2, col=4)
plot(jj, type="p", main="Data (points) and Trend+Season (line)")
 lines(Tsm+Ssm)
```

For forecasting, we use the first part of the filter recursions directly and store the predictions in `y` and the root mean square prediction errors in `rmspe`.

```
n.ahead=12; y = ts(append(jj, rep(0,n.ahead)), start=1960, freq=4)
rmspe = rep(0,n.ahead); x00 = ks$xf[,,num]; P00 = ks$Pf[,,num]
Q=t(cQ)%*%cQ;  R=t(cR)%*%(cR)  # see footnote and discussion below
for (m in 1:n.ahead){
 xp = Phi%*%x00; Pp = Phi%*%P00%*%t(Phi)+Q
 sig = A%*%Pp%*%t(A)+R; K = Pp%*%t(A)%*%(1/sig)
 x00 = xp; P00 = Pp-K%*%A%*%Pp
 y[num+m] = A%*%xp; rmspe[m] = sqrt(sig)  }
plot(y, type="o", main="", ylab="", ylim=c(5,30), xlim=c(1975,1984))
 upp = ts(y[(num+1):(num+n.ahead)]+2*rmspe, start=1981, freq=4)
 low = ts(y[(num+1):(num+n.ahead)]-2*rmspe, start=1981, freq=4)
 lines(upp, lty=2);  lines(low, lty=2);  abline(v=1980.75, lty=3)
```

Note that the Cholesky decomposition of `Q` does not exist here, however, the diagonal form allows us to use standard deviations for the first two diagonal elements of `cQ`. Also when we perform the smoothing part of the example, we set the lower 2×2 diagonal block of the `Q` matrix equal to the identity matrix; this is done for inversions in the script and it is only a device, the values are not used. These technicalities can be avoided using a form of the model that we present in the next section.

6.6 State-Space Models with Correlated Errors

Sometimes it is advantageous to write the state-space model in a slightly different way, as is done by numerous authors; for example, Anderson and Moore (1979) and Hannan and Deistler (1988). Here, we write the state-space model as

$$\boldsymbol{x}_{t+1} = \Phi \boldsymbol{x}_t + \Upsilon \boldsymbol{u}_{t+1} + \Theta \boldsymbol{w}_t \qquad t = 0, 1, \ldots, n \tag{6.97}$$

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \Gamma \boldsymbol{u}_t + \boldsymbol{v}_t \qquad t = 1, \ldots, n \tag{6.98}$$

where, in the state equation, $\boldsymbol{x}_0 \sim \mathrm{N}_p(\boldsymbol{\mu}_0, \Sigma_0)$, Φ is $p \times p$, and Υ is $p \times r$, Θ is $p \times m$ and $\boldsymbol{w}_t \sim$ iid $\mathrm{N}_m(\boldsymbol{0}, Q)$. In the observation equation, A_t is $q \times p$ and Γ is $q \times r$, and $\boldsymbol{v}_t \sim$ iid $\mathrm{N}_q(\boldsymbol{0}, R)$. In this model, while $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are still white noise series (both independent of $\boldsymbol{x}_0$), we also allow the state noise and observation noise to be correlated at time t; that is,

$$\mathrm{cov}(\boldsymbol{w}_s, \boldsymbol{v}_t) = S\, \delta_s^t, \tag{6.99}$$

where δ_s^t is Kronecker's delta; note that S is an $m \times q$ matrix. The major difference between this form of the model and the one specified by (6.3)–(6.4) is that this model starts the state noise process at $t = 0$ in order to ease the notation related to the concurrent covariance between $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$. Also, the inclusion of the matrix Θ allows us to avoid using a singular state noise process as was done in Example 6.10.

To obtain the innovations, $\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t$, and the innovation variance $\Sigma_t = A_t P_t^{t-1} A_t' + R$, in this case, we need the one-step-ahead state predictions. Of course, the filtered estimates will also be of interest, and they will be needed for smoothing. Property 6.2 (the smoother) as displayed in §6.2 still holds. The following property generates the predictor $\boldsymbol{x}_{t+1}^t$ from the past predictor $\boldsymbol{x}_t^{t-1}$ when the noise terms are correlated and exhibits the filter update.

Property 6.5 The Kalman Filter with Correlated Noise

For the state-space model specified in (6.97) and (6.98), with initial conditions $\boldsymbol{x}_1^0$ and P_1^0, for $t = 1, \ldots, n$,

$$\boldsymbol{x}_{t+1}^t = \Phi \boldsymbol{x}_t^{t-1} + \Upsilon \boldsymbol{u}_{t+1} + K_t \boldsymbol{\epsilon}_t \tag{6.100}$$

$$P_{t+1}^t = \Phi P_t^{t-1} \Phi' + \Theta Q \Theta' - K_t \Sigma_t K_t' \tag{6.101}$$

where $\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t$ and the gain matrix is given by

$$K_t = [\Phi P_t^{t-1} A_t' + \Theta S][A_t P_t^{t-1} A_t' + R]^{-1}. \tag{6.102}$$

The filter values are given by

$$\boldsymbol{x}_t^t = \boldsymbol{x}_t^{t-1} + P_t^{t-1} A_t' \left[A_t P_t^{t-1} A_t' + R\right]^{-1} \boldsymbol{\epsilon}_{t+1}, \tag{6.103}$$

$$P_t^t = P_t^{t-1} - P_t^{t-1} A_{t+1}' \left[A_t P_t^{t-1} A_t' + R\right]^{-1} A_t P_t^{t-1}. \tag{6.104}$$

The derivation of Property 6.5 is similar to the derivation of the Kalman filter in Property 6.1 (Problem 6.18); we note that the gain matrix K_t differs in the two properties. The filter values, (6.103)–(6.104), are symbolically identical to (6.19) and (6.20). To initialize the filter, we note that

$$\boldsymbol{x}_1^0 = E(\boldsymbol{x}_1) = \Phi\boldsymbol{\mu}_0 + \Upsilon\boldsymbol{u}_1, \quad \text{and} \quad P_1^0 = \text{var}(\boldsymbol{x}_1) = \Phi\Sigma_0\Phi' + \Theta Q\Theta'.$$

In the next two subsections, we show how to use the model (6.97)-(6.98) for fitting ARMAX models and for fitting (multivariate) regression models with autocorrelated errors. To put it succinctly, for ARMAX models, the inputs enter in the state equation and for regression with autocorrelated errors, the inputs enter in the observation equation. It is, of course, possible to combine the two models and we give an example of this at the end of the section.

6.6.1 ARMAX Models

Consider a k-dimensional ARMAX model given by

$$\boldsymbol{y}_t = \Upsilon\boldsymbol{u}_t + \sum_{j=1}^{p} \Phi_j \boldsymbol{y}_{t-j} + \sum_{k=1}^{q} \Theta_k \boldsymbol{v}_{t-k} + \boldsymbol{v}_t. \tag{6.105}$$

The observations $\boldsymbol{y}_t$ are a k-dimensional vector process, the Φs and Θs are $k \times k$ matrices, Υ is $k \times r$, $\boldsymbol{u}_t$ is the $r \times 1$ input, and $\boldsymbol{v}_t$ is a $k \times 1$ white noise process; in fact, (6.105) and (5.98) are identical models, but here, we have written the observations as $\boldsymbol{y}_t$. We now have the following property.

Property 6.6 A State-Space Form of ARMAX
For $p \geq q$, let

$$F = \begin{bmatrix} \Phi_1 & I & 0 \cdots & 0 \\ \Phi_2 & 0 & I \cdots & 0 \\ \vdots & \vdots & \vdots \ddots & \vdots \\ \Phi_{p-1} & 0 & 0 \cdots & I \\ \Phi_p & 0 & 0 \cdots & 0 \end{bmatrix} \quad G = \begin{bmatrix} \Theta_1 + \Phi_1 \\ \vdots \\ \Theta_q + \Phi_q \\ \Phi_{q+1} \\ \vdots \\ \Phi_p \end{bmatrix} \quad H = \begin{bmatrix} \Upsilon \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.106}$$

where F is $kp \times kp$, G is $kp \times k$, and H is $kp \times r$. Then, the state-space model given by

$$\boldsymbol{x}_{t+1} = F\boldsymbol{x}_t + H\boldsymbol{u}_{t+1} + G\boldsymbol{v}_t, \tag{6.107}$$

$$\boldsymbol{y}_t = A\boldsymbol{x}_t + \boldsymbol{v}_t, \tag{6.108}$$

where $A = \big[I, 0, \cdots, 0\big]$ is $k \times pk$ and I is the $k \times k$ identity matrix, implies the ARMAX model (6.105). If $p < q$, set $\Phi_{p+1} = \cdots = \Phi_q = 0$, in which case $p = q$ and (6.107)–(6.108) still apply. Note that the state process is kp-dimensional, whereas the observations are k-dimensional.

This form of the model is somewhat different than the form suggested in §6.1, equations (6.6)-(6.8). For example, in (6.8), by setting A_t equal to the $p \times p$ identity matrix (for all t) and setting $R = 0$ implies the data y_t in (6.8) follow a VAR(m) process. In doing so, however, we do not make use of the ability to allow for correlated state and observation error, so a singularity is introduced into the system in the form of $R = 0$. The method in Property 6.6 avoids that problem, and points out the fact that the same model can take many forms. We do not prove Property 6.6 directly, but the following example should suggest how to establish the general result.

Example 6.11 Univariate ARMAX$(1,1)$ in State-Space Form
Consider the univariate ARMAX$(1,1)$ model

$$y_t = \alpha_t + \phi y_{t-1} + \theta v_{t-1} + v_t,$$

where $\alpha_t = \boldsymbol{\Upsilon} \boldsymbol{u}_t$ to ease the notation. For a simple example, if $\boldsymbol{\Upsilon} = (\beta_0, \beta_1)$ and $\boldsymbol{u}_t = (1, t)'$, the model for y_t would be ARMA(1,1) with linear trend, $y_t = \beta_0 + \beta_1 t + \phi y_{t-1} + \theta v_{t-1} + v_t$. Using Property 6.6, we can write the model as

$$x_{t+1} = \phi x_t + \alpha_{t+1} + (\theta + \phi) v_t, \tag{6.109}$$

and

$$y_t = x_t + v_t. \tag{6.110}$$

In this case, (6.109) is the state equation with $w_t \equiv v_t$ and (6.110) is the observation equation. Consequently, $\mathrm{cov}(w_t, v_t) = \mathrm{var}(v_t) = R$, and $\mathrm{cov}(w_t, v_s) = 0$ when $s \neq t$, so Property 6.5 would apply. To verify (6.109) and (6.110) specify an ARMAX$(1,1)$ model, we have

$$\begin{aligned} y_t &= x_t + v_t && \text{from (6.110)} \\ &= \phi x_{t-1} + \alpha_t + (\theta + \phi) v_{t-1} + v_t && \text{from (6.109)} \\ &= \alpha_t + \phi(x_{t-1} + v_{t-1}) + \theta v_{t-1} + v_t && \text{rearrange terms} \\ &= \alpha_t + \phi y_{t-1} + \theta v_{t-1} + v_t, && \text{from (6.110).} \end{aligned}$$

Together, Properties 6.5 and 6.6 can be used to accomplish maximum likelihood estimation as described in §6.3 for ARMAX models. The ARMAX model is only a special case of the model (6.97)–(6.98), which is quite rich, as will be discovered in the next subsection.

6.6.2 Multivariate Regression with Autocorrelated Errors

In regression with autocorrelated errors, we are interested in fitting the regression model

$$\boldsymbol{y}_t = \boldsymbol{\Gamma} \boldsymbol{u}_t + \boldsymbol{\varepsilon}_t \tag{6.111}$$

to a $k \times 1$ vector process, $\boldsymbol{y}_t$, with r regressors $\boldsymbol{u}_t = (u_{t1}, \ldots, u_{tr})'$ where $\boldsymbol{\varepsilon}_t$ is vector ARMA(p, q) and $\boldsymbol{\Gamma}$ is a $k \times r$ matrix of regression parameters. We

note that the regressors do not have to vary with time (e.g., $u_{t1} \equiv 1$ includes a constant in the regression) and that the case $k = 1$ was treated in §5.6.

To put the model in state-space form, we simply notice that $\boldsymbol{\varepsilon}_t = \boldsymbol{y}_t - \Gamma \boldsymbol{u}_t$ is a k-dimensional ARMA(p, q) process. Thus, if we set $H = 0$ in (6.107), and include $\Gamma \boldsymbol{u}_t$ in (6.108), we obtain

$$\boldsymbol{x}_{t+1} = F\boldsymbol{x}_t + G\boldsymbol{v}_t, \tag{6.112}$$

$$\boldsymbol{y}_t = \Gamma \boldsymbol{u}_t + A\boldsymbol{x}_t + \boldsymbol{v}_t, \tag{6.113}$$

where the model matrices A, F, and G are defined in Property 6.6. The fact that (6.112)–(6.113) is multivariate regression with autocorrelated errors follows directly from Property 6.6 by noticing that together, $\boldsymbol{x}_{t+1} = F\boldsymbol{x}_t + G\boldsymbol{v}_t$ and $\boldsymbol{\varepsilon}_t = A\boldsymbol{x}_t + \boldsymbol{v}_t$ imply $\boldsymbol{\varepsilon}_t = \boldsymbol{y}_t - \Gamma \boldsymbol{u}_t$ is vector ARMA(p, q).

As in the case of ARMAX models, regression with autocorrelated errors is a special case of the state-space model, and the results of Property 6.5 can be used to obtain the innovations form of the likelihood for parameter estimation.

Example 6.12 Mortality, Temperature and Pollution

In this example, we fit an ARMAX model to the detrended mortality series `cmort`. As in Examples 5.10 and 5.11, we let M_t denote the weekly cardiovascular mortality series, T_t as the corresponding temperature series `tempr`, and P_t as the corresponding particulate series. A preliminary analysis suggests the following considerations (no output is shown):

- An AR(2) model fits well to detrended M_t:
  ```
  fit = arima(cmort, order=c(2,0,0), xreg=time(cmort))
  ```
- The CCF between the mortality residuals, the temperature series and the particulates series, shows a strong correlation with temperature lagged one week (T_{t-1}), concurrent particulate level (P_t) and the particulate level about one month prior (P_{t-4}).
  ```
  acf(cbind(dmort <- resid(fit), tempr, part))
  lag.plot2(tempr, dmort, 8)
  lag.plot2(part, dmort, 8)
  ```

From these results, we decided to fit the ARMAX model

$$\widetilde{M}_t = \phi_1 \widetilde{M}_{t-1} + \phi_2 \widetilde{M}_{t-2} + \beta_1 T_{t-1} + \beta_2 P_t + \beta_3 P_{t-4} + v_t \tag{6.114}$$

to the detrended mortality series, $\widetilde{M}_t = M_t - (\alpha + \beta_4 t)$, where $v_t \sim$ iid $\mathrm{N}(0, \sigma_v^2)$. To write the model in state-space form using Property 6.6, let

$$\boldsymbol{x}_{t+1} = \Phi \boldsymbol{x}_t + \Upsilon \boldsymbol{u}_{t+1} + \Theta v_t \qquad t = 0, 1, \ldots, n$$

$$y_t = \alpha + A\boldsymbol{x}_t + \Gamma \boldsymbol{u}_t + v_t \qquad t = 1, \ldots, n$$

with

$$\Phi = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \quad \Upsilon = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \Theta = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

$A = [\ 1\ \ 0\ ]$, $\Gamma = [\ 0\ 0\ 0\ \beta_4\ ]$, $\boldsymbol{u}_t = (T_t, P_t, P_{t-4}, t)'$, $y_t = M_t$. Note that the state process is bivariate and the observation process is univariate. We could have included α in Γ; however, to reduce the number of parameters to be estimated numerically, we centered M_t by its sample mean and removed the constant α from the model. In addition, t was centered by its mean. Initial values of the parameters were taken from the preliminary investigation. We note that P_t and P_{t-4} are highly correlated, so orthogonalizing these two inputs would be advantageous (although we did not do it here).

The estimates and standard errors are displayed along with the following R code; investigation of the residuals shows that the model fits well.

```
dm = cmort - mean(cmort)                  # center mortality
trend = time(cmort) - mean(time(cmort))   # center time
ded = ts.intersect(dm, u1=lag(tempr,-1), u2=part, u3=lag(part,-4),
    u4=trend, dframe=TRUE)
y = ded$dm; input = cbind(ded$u1, ded$u2, ded$u3, ded$u4)
num = length(y); A = array(c(1,0), dim = c(1,2,num))
# Function to Calculate Likelihood
Linn=function(para){
  phi1=para[1]; phi2=para[2]; cR=para[3]
  b1=para[4]; b2=para[5]; b3=para[6]; b4=para[7]
  mu0 = matrix(c(0,0), 2, 1); Sigma0 = diag(100, 2)
  Phi = matrix(c(phi1, phi2, 1, 0), 2)
  Theta = matrix(c(phi1, phi2), 2)
  Ups = matrix(c(b1, 0, b2, 0, b3, 0, 0, 0), 2, 4)
  Gam = matrix(c(0, 0, 0, b4), 1, 4); cQ = cR;  S = cR^2
  kf = Kfilter2(num, y, A, mu0, Sigma0, Phi, Ups, Gam, Theta, cQ,
    cR, S, input)
  return(kf$like) }
# Estimation
phi1=.4;  phi2=.4;  cR=5; b1=-.1; b2=.1; b3=.1; b4=-1.5
init.par = c(phi1, phi2, cR, b1, b2, b3, b4)  # initial parameters
est = optim(init.par, Linn, NULL, method="L-BFGS-B", hessian=TRUE,
    control=list(trace=1,REPORT=1))
SE = sqrt(diag(solve(est$hessian)))
# Results
u = cbind(estimate=est$par, SE)
rownames(u)=c("phi1","phi2","sigv","TL1","P","PL4","trnd"); u
```

```
           estimate         SE
phi1    0.31437053 0.03712001
phi2    0.31777254 0.03825371
sigv    5.05662192 0.15920440
TL1    -0.11929669 0.01106674   (beta 1)
P       0.11935144 0.01746386   (beta 2)
PL4     0.06715402 0.01844125   (beta 3)
trnd   -1.34871992 0.21921715   (beta 4)
```

The residuals can be obtained by running `Kfilter2` again at the final estimates; they are returned as `innov`.

6.7 Bootstrapping State-Space Models

Although in §6.3 we discussed the fact that under general conditions (which we assume to hold in this section) the MLEs of the parameters of a DLM are consistent and asymptotically normal, time series data are often of short or moderate length. Several researchers have found evidence that samples must be fairly large before asymptotic results are applicable (Dent and Min, 1978; Ansley and Newbold, 1980). Moreover, as we discussed in Example 3.35, problems occur if the parameters are near the boundary of the parameter space. In this section, we discuss an algorithm for bootstrapping state-space models; this algorithm and its justification, including the non-Gaussian case, along with numerous examples, can be found in Stoffer and Wall (1991) and in Stoffer and Wall (2004). In view of §6.6, anything we do or say here about DLMs applies equally to ARMAX models.

Using the DLM given by (6.97)–(6.99) and Property 6.5, we write the innovations form of the filter as

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t, \tag{6.115}$$

$$\Sigma_t = A_t P_t^{t-1} A_t' + R, \tag{6.116}$$

$$K_t = [\Phi P_t^{t-1} A_t' + \Theta S] \Sigma_t^{-1}, \tag{6.117}$$

$$\boldsymbol{x}_{t+1}^t = \Phi \boldsymbol{x}_t^{t-1} + \Upsilon \boldsymbol{u}_{t+1} + K_t \boldsymbol{\epsilon}_t, \tag{6.118}$$

$$P_{t+1}^t = \Phi P_t^{t-1} \Phi' + \Theta Q \Theta' - K_t \Sigma_t K_t'. \tag{6.119}$$

This form of the filter is just a rearrangement of the filter given in Property 6.5.

In addition, we can rewrite the model to obtain its innovations form,

$$\boldsymbol{x}_{t+1}^t = \Phi \boldsymbol{x}_t^{t-1} + \Upsilon \boldsymbol{u}_{t+1} + K_t \boldsymbol{\epsilon}_t, \tag{6.120}$$

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t^{t-1} + \Gamma \boldsymbol{u}_t + \boldsymbol{\epsilon}_t. \tag{6.121}$$

This form of the model is a rewriting of (6.115) and (6.118), and it accommodates the bootstrapping algorithm.

As discussed in Example 6.5, although the innovations $\boldsymbol{\epsilon}_t$ are uncorrelated, initially, Σ_t can be vastly different for different time points t. Thus, in a resampling procedure, we can either ignore the first few values of $\boldsymbol{\epsilon}_t$ until Σ_t stabilizes or we can work with the standardized innovations

$$\boldsymbol{e}_t = \Sigma_t^{-1/2} \boldsymbol{\epsilon}_t, \tag{6.122}$$

so we are guaranteed these innovations have, at least, the same first two moments. In (6.122), $\Sigma_t^{1/2}$ denotes the unique square root matrix of Σ_t defined by $\Sigma_t^{1/2} \Sigma_t^{1/2} = \Sigma_t$. In what follows, we base the bootstrap procedure on the standardized innovations, but we stress the fact that, even in this case, ignoring startup values might be necessary, as noted by Stoffer and Wall (1991).

The model coefficients and the correlation structure of the model are uniquely parameterized by a $k \times 1$ parameter vector Θ_0; that is, $\Phi = \Phi(\Theta_0)$, $\Upsilon = \Upsilon(\Theta_0)$, $Q = Q(\Theta_0)$, $A_t = A_t(\Theta_0)$, $\Gamma = \Gamma(\Theta_0)$, and $R = R(\Theta_0)$. Recall the innovations form of the Gaussian likelihood (ignoring a constant) is

$$\begin{aligned} -2 \ln L_Y(\Theta) &= \sum_{t=1}^{n} \left[\ln |\Sigma_t(\Theta)| + \boldsymbol{\epsilon}_t(\Theta)' \Sigma_t(\Theta)^{-1} \boldsymbol{\epsilon}_t(\Theta) \right] \\ &= \sum_{t=1}^{n} \left[\ln |\Sigma_t(\Theta)| + \boldsymbol{e}_t(\Theta)' \boldsymbol{e}_t(\Theta) \right]. \end{aligned} \tag{6.123}$$

We stress the fact that it is not necessary for the model to be Gaussian to consider (6.123) as the criterion function to be used for parameter estimation.

Let $\widehat{\Theta}$ denote the MLE of Θ_0, that is, $\widehat{\Theta} = \operatorname{argmax}_{\Theta} L_Y(\Theta)$, obtained by the methods discussed in §6.3. Let $\boldsymbol{\epsilon}_t(\widehat{\Theta})$ and $\Sigma_t(\widehat{\Theta})$ be the innovation values obtained by running the filter, (6.115)–(6.119), under $\widehat{\Theta}$. Once this has been done, the bootstrap procedure is accomplished by the following steps.

(i) Construct the standardized innovations

$$\boldsymbol{e}_t(\widehat{\Theta}) = \Sigma_t^{-1/2}(\widehat{\Theta}) \boldsymbol{\epsilon}_t(\widehat{\Theta}).$$

(ii) Sample, with replacement, n times from the set $\{\boldsymbol{e}_1(\widehat{\Theta}), \ldots, \boldsymbol{e}_n(\widehat{\Theta})\}$ to obtain $\{\boldsymbol{e}_1^*(\widehat{\Theta}), \ldots, \boldsymbol{e}_n^*(\widehat{\Theta})\}$, a bootstrap sample of standardized innovations.

(iii) Construct a bootstrap data set $\{\boldsymbol{y}_1^*, \ldots, \boldsymbol{y}_n^*\}$ as follows. Define the $(p+q) \times 1$ vector $\boldsymbol{\xi}_t = (\boldsymbol{x}_{t+1}^{t'}, \boldsymbol{y}_t')'$. Stacking (6.120) and (6.121) results in a vector first-order equation for $\boldsymbol{\xi}_t$ given by

$$\boldsymbol{\xi}_t = F_t \boldsymbol{\xi}_{t-1} + G \boldsymbol{u}_t + H_t \boldsymbol{e}_t, \tag{6.124}$$

where

$$F_t = \begin{bmatrix} \Phi & 0 \\ A_t & 0 \end{bmatrix}, \quad G = \begin{bmatrix} \Upsilon \\ \Gamma \end{bmatrix}, \quad H_t = \begin{bmatrix} K_t \Sigma_t^{1/2} \\ \Sigma_t^{1/2} \end{bmatrix}.$$

Thus, to construct the bootstrap data set, solve (6.124) using $\boldsymbol{e}_t^*(\widehat{\Theta})$ in place of $\boldsymbol{e}_t$. The exogenous variables $\boldsymbol{u}_t$ and the initial conditions of the Kalman filter remain fixed at their given values, and the parameter vector is held fixed at $\widehat{\Theta}$.

(iv) Using the bootstrap data set $\{\boldsymbol{y}_t^*; t = 1, \ldots, n\}$, construct a likelihood, $L_{Y^*}(\Theta)$, and obtain the MLE of Θ, say, $\widehat{\Theta}^*$.

(v) Repeat steps 2 through 4, a large number, B, of times, obtaining a bootstrapped set of parameter estimates $\{\widehat{\Theta}_b^*;\ b = 1, \ldots, B\}$. The finite sample distribution of $\widehat{\Theta} - \Theta_0$ may be approximated by the distribution of $\widehat{\Theta}_b^* - \widehat{\Theta}$, $b = 1, \ldots, B$.

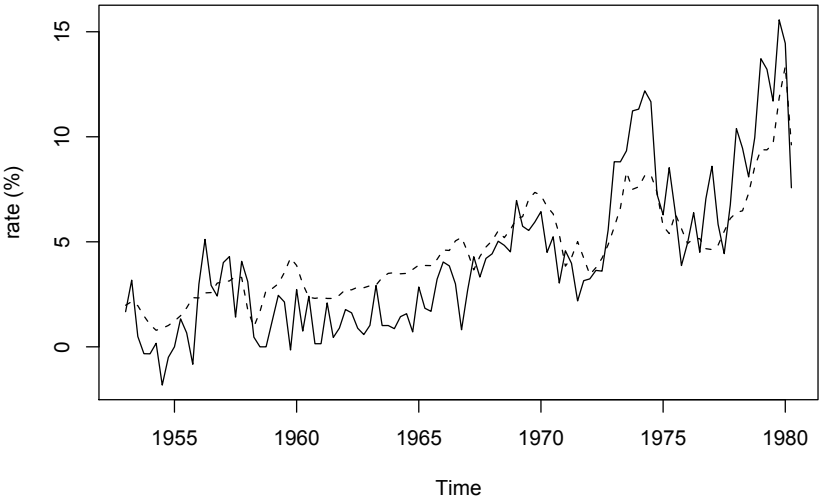

Fig. 6.8. Quarterly interest rate for Treasury bills (dashed line) and quarterly inflation rate (solid line) in the Consumer Price Index.

In the next example, we discuss the case of a linear regression model, but where the regression coefficients are stochastic and allowed to vary with time. The state-space model provides a convenient setting for the analysis of such models.

Example 6.13 Stochastic Regression

Figure 6.8 shows the quarterly inflation rate (solid line), y_t, in the Consumer Price Index and the quarterly interest rate recorded for Treasury bills (dashed line), z_t, from the first quarter of 1953 through the second quarter of 1980, $n = 110$ observations. These data are taken from Newbold and Bos (1985).

In this example, we consider one analysis that was discussed in Newbold and Bos (1985, pp. 61-73), that focused on the first 50 observations and where quarterly inflation was modeled as being stochastically related to quarterly interest rate,

$$y_t = \alpha + \beta_t z_t + v_t,$$

where α is a fixed constant, β_t is a stochastic regression coefficient, and v_t is white noise with variance σ_v^2. The stochastic regression term, which comprises the state variable, is specified by a first-order autoregression,

$$(\beta_t - b) = \phi(\beta_{t-1} - b) + w_t,$$

where b is a constant, and w_t is white noise with variance σ_w^2. The noise processes, v_t and w_t, are assumed to be uncorrelated.

Using the notation of the state-space model (6.97) and (6.98), we have in the state equation, $\boldsymbol{x}_t = \beta_t$, $\Phi = \phi$, $\boldsymbol{u}_t \equiv 1$, $\Upsilon = (1 - \phi)b$, $Q = \sigma_w^2$, and

Table 6.2. Comparison of Standard Errors

Parameter	MLE	Asymptotic Standard Error	Bootstrap Standard Error
ϕ	.865	.223	.463
α	−.686	.487	.557
b	.788	.226	.821
σ_w	.115	.107	.216
σ_v	1.135	.147	.340

in the observation equation, $A_t = z_t$, $\Gamma = \alpha$, $R = \sigma_v^2$, and $S = 0$. The parameter vector is $\Theta = (\phi, \alpha, b, \sigma_w, \sigma_v)'$. The results of the Newton–Raphson estimation procedure are listed in Table 6.2. Also shown in the Table 6.2 are the corresponding standard errors obtained from $B = 500$ runs of the bootstrap. These standard errors are simply the standard deviations of the bootstrapped estimates, that is, the square root of $\sum_{b=1}^{B}(\Theta_{ib}^* - \bar{\Theta}_i^*)^2/(B-1)$, where Θ_i, represents the ith parameter, $i = 1, \ldots, 5$, and $\bar{\Theta}_i^* = \sum_{b=1}^{B} \Theta_{ib}^*/B$.

The asymptotic standard errors listed in Table 6.2 are typically much smaller than those obtained from the bootstrap. For most of the cases, the bootstrapped standard errors are at least 50% larger than the corresponding asymptotic value. Also, asymptotic theory prescribes the use of normal theory when dealing with the parameter estimates. The bootstrap, however, allows us to investigate the small sample distribution of the estimators and, hence, provides more insight into the data analysis.

For example, Figure 6.9 shows the bootstrap distribution of the estimator of ϕ in the upper left-hand corner. This distribution is highly skewed with values concentrated around .8, but with a long tail to the left. Some quantiles are −.09 (5%), .11 (10%), .34 (25%), .73 (50%), .86 (75%), .96 (90%), .98 (95%), and they can be used to obtain confidence intervals. For example, a 90% confidence interval for ϕ would be approximated by (−.09, .96). This interval is ridiculously wide and includes 0 as a plausible value of ϕ; we will interpret this after we discuss the results of the estimation of σ_w.

Figure 6.9 shows the bootstrap distribution of $\widehat{\sigma}_w$ in the lower right-hand corner. The distribution is concentrated at two locations, one at approximately $\widehat{\sigma}_w = .25$ (which is the median of the distribution of values away from 0) and the other at $\widehat{\sigma}_w = 0$. The cases in which $\widehat{\sigma}_w \approx 0$ correspond to deterministic state dynamics. When $\sigma_w = 0$ and $|\phi| < 1$, then $\beta_t \approx b$ for large t, so the approximately 25% of the cases in which $\widehat{\sigma}_w \approx 0$ suggest a fixed state, or constant coefficient model. The cases in which $\widehat{\sigma}_w$ is away from zero would suggest a truly stochastic regression parameter. To investigate this matter further, the off-diagonals of Figure 6.9 show the joint bootstrapped estimates, $(\widehat{\phi}, \widehat{\sigma}_w)$, for positive values of $\widehat{\phi}^*$. The joint distribution suggests $\widehat{\sigma}_w > 0$ corresponds to $\widehat{\phi} \approx 0$. When $\phi = 0$, the state dynamics are given by $\beta_t = b + w_t$. If, in addition, σ_w is small relative to b, the system is nearly de-

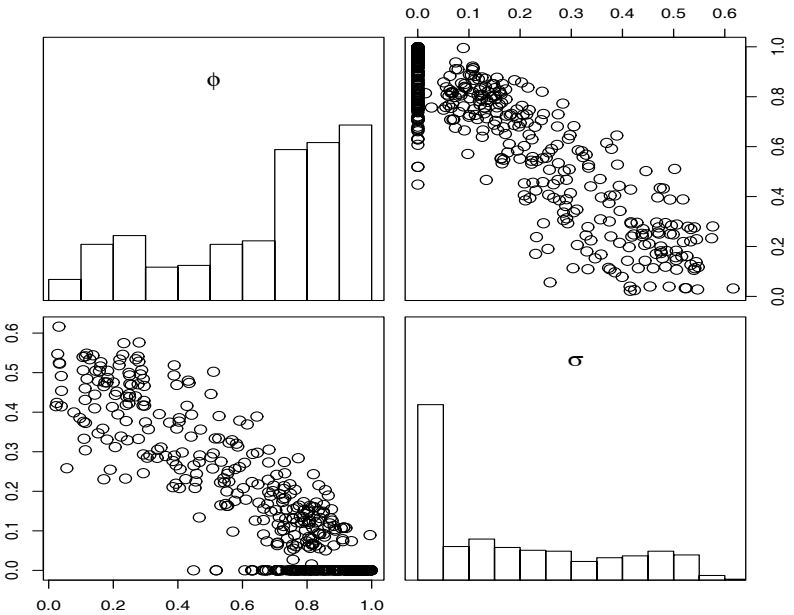

Fig. 6.9. Bootstrap distribution, $B = 500$, of (i) the estimator of ϕ (upper left), (ii) the estimator of σ_w (lower right), and (iii) the estimators jointly (off-diagonals). Only the values corresponding to $\widehat{\phi}^* \geq 0$ are shown.

terministic; that is, $\beta_t \approx b$. Considering these results, the bootstrap analysis leads us to conclude the dynamics of the data are best described in terms of a fixed regression effect.

The following R code was used for this example. We note that the first line of the code sets the relative tolerance for determining convergence for the numerical optimization. *Using the current default setting may result in a long run time of the algorithm* and we suggest the value be decreased on slower machines or for demonstration purposes. Also, decreasing the number of bootstrap replicates will improve computation time; for example, setting `tol=.001` and `nboot=200` yields reasonable results. In this example, we fix the first three values of the data for the resampling scheme.

```
tol = sqrt(.Machine$double.eps)  # convergence tolerance
nboot = 500  # number of bootstrap replicates
y = window(qinfl, c(1953,1), c(1965,2))  # inflation
z = window(qintr, c(1953,1), c(1965,2))  # interest
num = length(y); input = matrix(1, num, 1)
A = array(z, dim=c(1,1,num))
# Function to Calculate Likelihood
Linn=function(para){
  phi=para[1]; alpha=para[2]; b=para[3]; Ups=(1-phi)*b
  cQ=para[4]; cR=para[5]
```

```
  kf=Kfilter2(num,y,A,mu0,Sigma0,phi,Ups,alpha,1,cQ,cR,0,input)
  return(kf$like)    }
# Parameter Estimation
mu0=1; Sigma0=.01; phi=.84; alpha=-.77; b=.85; cQ=.12; cR=1.1
init.par = c(phi, alpha, b, cQ, cR)    # initial parameters
est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
     control=list(trace=1,REPORT=1,reltol=tol))
SE = sqrt(diag(solve(est$hessian)))
phi = est$par[1]; alpha=est$par[2]; b=est$par[3]; Ups=(1-phi)*b
cQ=est$par[4]; cR=est$par[5]
rbind(estimate=est$par, SE)
```

```
                 phi      alpha         b   sigma_w   sigma_v
estimate 0.8653348 -0.6855891 0.7879308 0.1145682 1.1353139
SE       0.2231382  0.4865775 0.2255649 0.1071838 0.1472067
```

```
# BEGIN BOOTSTRAP
Linn2=function(para){  # likelihood for bootstrapped data
 phi=para[1]; alpha=para[2]; b=para[3]; Ups=(1-phi)*b
 cQ=para[4]; cR=para[5]
 kf=Kfilter2(num,y.star,A,mu0,Sigma0,phi,Ups,alpha,1,cQ,cR,0,input)
 return(kf$like)  }
# Run the filter at the estimates
kf=Kfilter2(num,y,A,mu0,Sigma0,phi,Ups,alpha,1,cQ,cR,0,input)
# Pull out necessary values from the filter and initialize
xp=kf$xp; innov=kf$innov; sig=kf$sig; K=kf$K; e=innov/sqrt(sig)
e.star=e; y.star=y; xp.star=xp; k=4:50
para.star = matrix(0, nboot, 5)  # to store estimates
init.par=c(.84,-.77,.85,.12,1.1)
for (i in 1:nboot){cat("iteration:", i, "\n")
 e.star[k] = sample(e[k], replace=TRUE)
 for (j in k){
 xp.star[j] = phi*xp.star[j-1]+Ups+K[j]*sqrt(sig[j])*e.star[j] }
y.star[k] = z[k]*xp.star[k]+alpha+sqrt(sig[k])*e.star[k]
est.star = optim(init.par, Linn2, NULL, method="BFGS",
     control=list(reltol=tol))
para.star[i,] = cbind(est.star$par[1], est.star$par[2],
     est.star$par[3], abs(est.star$par[4]), abs(est.star$par[5]))}
rmse = rep(NA,5)  # compute bootstrapped SEs (Table 6.2)
for(i in 1:5){rmse[i]=sqrt(sum((para.star[,i]-est$par[i])^2)/nboot)
  cat(i, rmse[i],"\n")   }
```

```
  1  0.46294 | 2  0.55698 | 3  0.82148 | 4  0.21595 | 5  0.34011
```

```
# Plot for phi vs sigma_w
phi = para.star[,1]; sigw = abs(para.star[,4])
phi = ifelse(phi<0, NA, phi)  # any phi < 0 not plotted
panel.hist <- function(x, ...){
   usr <- par("usr"); on.exit(par(usr))
   par(usr = c(usr[1:2], 0, 1.5) )
   h <- hist(x, plot = FALSE)
   breaks <- h$breaks; nB <- length(breaks)
```

```
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, ...)}
u = cbind(phi, sigw);  colnames(u) = c("f","s")
pairs(u, cex=1.5, pch=1, diag.panel=panel.hist, cex.labels=1.5,
      font.labels=5)
```

6.8 Dynamic Linear Models with Switching

The problem of modeling changes in regimes for vector-valued time series has been of interest in many different fields. In §5.5, we explored the idea that the dynamics of the system of interest might change over the course of time. In Example 5.6, we saw that pneumonia and influenza mortality rates behave differently when a flu epidemic occurs than when no epidemic occurs. As another example, some authors (for example, Hamilton, 1989, or McCulloch and Tsay, 1993) have explored the possibility the dynamics of the quarterly U.S. GNP series (say, y_t) analyzed in Example 3.35 might be different during expansion ($\nabla \log y_t > 0$) than during contraction ($\nabla \log y_t < 0$). In this section, we will concentrate on the method presented in Shumway and Stoffer (1991). One way of modeling change in an evolving time series is by assuming the dynamics of some underlying model changes discontinuously at certain undetermined points in time. Our starting point is the DLM given by (6.1) and (6.2), namely,

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \tag{6.125}$$

to describe the $p \times 1$ state dynamics, and

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \boldsymbol{v}_t \tag{6.126}$$

to describe the $q \times 1$ observation dynamics. Recall $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are Gaussian white noise sequences with $\mathrm{var}(\boldsymbol{w}_t) = Q$, $\mathrm{var}(\boldsymbol{v}_t) = R$, and $\mathrm{cov}(\boldsymbol{w}_t, \boldsymbol{v}_s) = 0$ for all s and t.

Generalizations of (6.125) and (6.126) to include the possibility of changes occurring over time have been approached by allowing changes in the error covariances (Harrison and Stevens, 1976, Gordon and Smith, 1988, 1990) or by assigning mixture distributions to the observation errors $\boldsymbol{v}_t$ (Peña and Guttman, 1988). Approximations to filtering were derived in all of the aforementioned articles. An application to monitoring renal transplants was described in Smith and West (1983) and in Gordon and Smith (1990). Changes can also be modeled in the classical regression case by allowing switches in the design matrix, as in Quandt (1972).

Switching via a stationary Markov chain with independent observations has been developed by Lindgren (1978) and Goldfeld and Quandt (1973). In the Markov chain approach, we declare the dynamics of the system at time t are generated by one of m possible regimes evolving according to a Markov

chain over time. As a simple example, suppose the dynamics of a univariate time series, y_t, is generated by either the model (1) $y_t = \beta_1 y_{t-1} + w_t$ or the model (2) $y_t = \beta_2 y_{t-1} + w_t$. We will write the model as $y_t = \phi_t y_{t-1} + w_t$ such that $\Pr(\phi_t = \beta_j) = \pi_j$, $j = 1, 2$, $\pi_1 + \pi_2 = 1$, and with the Markov property

$$\Pr(\phi_t = \beta_j \mid \phi_{t-1} = \beta_i, \phi_{t-2} = \beta_{i_2}, \ldots) = \Pr(\phi_t = \beta_j \mid \phi_{t-1} = \beta_i) = \pi_{ij},$$

for $i, j = 1, 2$ (and $i_2, \ldots = 1, 2$). As previously mentioned, Markov switching for dependent data has been applied by Hamilton (1989) to detect changes between positive and negative growth periods in the economy. Applications to speech recognition have been considered by Juang and Rabiner (1985). The case in which the particular regime is unknown to the observer comes under the heading of hidden Markov models, and the techniques related to analyzing these models are summarized in Rabiner and Juang (1986). An application of the idea of switching to the tracking of multiple targets has been considered in Bar-Shalom (1978), who obtained approximations to Kalman filtering in terms of weighted averages of the innovations.

Example 6.14 Tracking Multiple Targets

The approach of Shumway and Stoffer (1991) was motivated primarily by the problem of tracking a large number of moving targets using a vector $\boldsymbol{y}_t$ of sensors. In this problem, we do not know at any given point in time which target any given sensor has detected. Hence, it is the structure of the measurement matrix A_t in (6.126) that is changing, and not the dynamics of the signal $\boldsymbol{x}_t$ or the noises, $\boldsymbol{w}_t$ or $\boldsymbol{v}_t$. As an example, consider a 3×1 vector of satellite measurements $\boldsymbol{y}_t = (y_{t1}, y_{t2}, y_{t3})'$ that are observations on some combination of a 3×1 vector of targets or signals, $\boldsymbol{x}_t = (x_{t1}, x_{t2}, x_{t3})'$. For the measurement matrix

$$A_t = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

for example, the first sensor, y_{t1}, observes the second target, x_{t2}; the second sensor, y_{t2}, observes the first target, x_{t1}; and the third sensor, y_{t3}, observes the third target, x_{t3}. All possible detection configurations will define a set of possible values for A_t, say, $\{M_1, M_2, \ldots, M_m\}$, as a collection of plausible measurement matrices.

Example 6.15 Modeling Economic Change

As another example of the switching model presented in this section, consider the case in which the dynamics of the linear model changes suddenly over the history of a given realization. For example, Lam (1990) has given the following generalization of Hamilton's (1989) model for detecting positive and negative growth periods in the economy. Suppose the data are generated by

$$y_t = z_t + n_t, \tag{6.127}$$

where z_t is an autoregressive series and n_t is a random walk with a drift that switches between two values α_0 and $\alpha_0 + \alpha_1$. That is,

$$n_t = n_{t-1} + \alpha_0 + \alpha_1 S_t, \tag{6.128}$$

with $S_t = 0$ or 1, depending on whether the system is in state 1 or state 2. For the purpose of illustration, suppose

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + w_t \tag{6.129}$$

is an AR(2) series with $\text{var}(w_t) = \sigma_w^2$. Lam (1990) wrote (6.127) in a differenced form

$$\nabla y_t = z_t - z_{t-1} + \alpha_0 + \alpha_1 S_t, \tag{6.130}$$

which we may take as the observation equation (6.126) with state vector

$$\boldsymbol{x}_t = (z_t, z_{t-1}, \alpha_0, \alpha_1)' \tag{6.131}$$

and

$$M_1 = [1, -1, 1, 0] \quad \text{and} \quad M_2 = [1, -1, 1, 1] \tag{6.132}$$

determining the two possible economic conditions. The state equation, (6.125), is of the form

$$\begin{pmatrix} z_t \\ z_{t-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} z_{t-1} \\ z_{t-2} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} + \begin{pmatrix} w_t \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.133}$$

The observation equation, (6.130), can be written as

$$\nabla y_t = A_t \boldsymbol{x}_t + v_t, \tag{6.134}$$

where we have included the possibility of observational noise, and where $\Pr(A_t = M_1) = 1 - \Pr(A_t = M_2)$, with M_1 and M_2 given in (6.132).

To incorporate a reasonable switching structure for the measurement matrix into the DLM that is compatible with both practical situations previously described, we assume that the m possible configurations are states in a nonstationary, independent process defined by the time-varying probabilities

$$\pi_j(t) = \Pr(A_t = M_j), \tag{6.135}$$

for $j = 1, \ldots, m$ and $t = 1, 2, \ldots, n$. Important information about the current state of the measurement process is given by the filtered probabilities of being in state j, defined as the conditional probabilities

$$\pi_j(t|t) = \Pr(A_t = M_j | Y_t), \tag{6.136}$$

which also vary as a function of time. In (6.136), we have used the notation $Y_s = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s\}$. The filtered probabilities (6.136) give the time-varying estimates of the probability of being in state j given the data to time t.

It will be important for us to obtain estimators of the configuration probabilities, $\pi_j(t|t)$, the predicted and filtered state estimators, $\boldsymbol{x}_t^{t-1}$ and $\boldsymbol{x}_t^t$, and the corresponding error covariance matrices P_t^{t-1} and P_t^t. Of course, the predictor and filter estimators will depend on the parameters, Θ, of the DLM. In many situations, the parameters will be unknown and we will have to estimate them. Our focus will be on maximum likelihood estimation, but other authors have taken a Bayesian approach that assigns priors to the parameters, and then seeks posterior distributions of the model parameters; see, for example, Gordon and Smith (1990), Peña and Guttman (1988), or McCulloch and Tsay (1993).

We now establish the recursions for the filters associated with the state $\boldsymbol{x}_t$ and the switching process, A_t. As discussed in §6.3, the filters are also an essential part of the maximum likelihood procedure. The predictors, $\boldsymbol{x}_t^{t-1} = E(\boldsymbol{x}_t|Y_{t-1})$, and filters, $\boldsymbol{x}_t^t = E(\boldsymbol{x}_t|Y_t)$, and their associated error variance–covariance matrices, P_t^{t-1} and P_t^t, are given by

$$\boldsymbol{x}_t^{t-1} = \Phi \boldsymbol{x}_{t-1}^{t-1}, \tag{6.137}$$

$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi' + Q, \tag{6.138}$$

$$\boldsymbol{x}_t^t = \boldsymbol{x}_t^{t-1} + \sum_{j=1}^{m} \pi_j(t|t) K_{tj} \boldsymbol{\epsilon}_{tj}, \tag{6.139}$$

$$P_t^t = \sum_{j=1}^{m} \pi_j(t|t)(I - K_{tj} M_j) P_t^{t-1}, \tag{6.140}$$

$$K_{tj} = P_t^{t-1} M_j' \Sigma_{tj}^{-1}, \tag{6.141}$$

where the innovation values in (6.139) and (6.141) are

$$\boldsymbol{\epsilon}_{tj} = \boldsymbol{y}_t - M_j \boldsymbol{x}_t^{t-1}, \tag{6.142}$$

$$\Sigma_{tj} = M_j P_t^{t-1} M_j' + R, \tag{6.143}$$

for $j = 1, \ldots, m$.

Equations (6.137)-(6.141) exhibit the filter values as weighted linear combinations of the m innovation values, (6.142)-(6.143), corresponding to each of the possible measurement matrices. The equations are similar to the approximations introduced by Bar-Shalom and Tse (1975), by Gordon and Smith (1990), and Peña and Guttman (1988).

To verify (6.139), let the indicator $I(A_t = M_j) = 1$ when $A_t = M_j$, and zero otherwise. Then, using (6.21),

$$\begin{aligned}
\boldsymbol{x}_t^t &= E(\boldsymbol{x}_t|Y_t) = E[E(\boldsymbol{x}_t|Y_t, A_t) \mid Y_t] \\
&= E\Big\{\sum_{j=1}^{m} E(\boldsymbol{x}_t|Y_t, A_t = M_j) I(A_t = M_j) \mid Y_t\Big\} \\
&= E\Big\{\sum_{j=1}^{m} [\boldsymbol{x}_t^{t-1} + K_{tj}(\boldsymbol{y}_t - M_j \boldsymbol{x}_t^{t-1})] I(A_t = M_j) \mid Y_t\Big\} \\
&= \sum_{j=1}^{m} \pi_j(t|t)[\boldsymbol{x}_t^{t-1} + K_{tj}(\boldsymbol{y}_t - M_j \boldsymbol{x}_t^{t-1})],
\end{aligned}$$

where K_{tj} is given by (6.141). Equation (6.140) is derived in a similar fashion; the other relationships, (6.137), (6.138), and (6.141), follow from straightforward applications of the Kalman filter results given in Property 6.1.

Next, we derive the filters $\pi_j(t|t)$. Let $f_j(t|t-1)$ denote the conditional density of $\boldsymbol{y}_t$ given the past $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}$, and $A_t = M_j$, for $j = 1, \ldots, m$. Then,

$$\pi_j(t|t) = \frac{\pi_j(t) f_j(t|t-1)}{\sum_{k=1}^{m} \pi_k(t) f_k(t|t-1)}, \tag{6.144}$$

where we assume the distribution $\pi_j(t)$, for $j = 1, \ldots, m$ has been specified before observing $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$ (details follow as in Example 6.16 below). If the investigator has no reason to prefer one state over another at time t, the choice of uniform priors, $\pi_j(t) = m^{-1}$, for $j = 1, \ldots, m$, will suffice. Smoothness can be introduced by letting

$$\pi_j(t) = \sum_{i=1}^{m} \pi_i(t-1|t-1)\pi_{ij}, \tag{6.145}$$

where the non-negative weights π_{ij} are chosen so $\sum_{i=1}^{m} \pi_{ij} = 1$. If the A_t process was Markov with transition probabilities π_{ij}, then (6.145) would be the update for the filter probability, as shown in the next example.

Example 6.16 Hidden Markov Chain Model

If $\{A_t\}$ is a hidden Markov chain with stationary transition probabilities $\pi_{ij} = \Pr(A_t = M_j | A_{t-1} = M_i)$, for $i, j = 1, \ldots, m$, letting $p(\cdot)$ denote a generic probability function, we have

$$\begin{aligned}
\pi_j(t|t) &= \frac{p(A_t = M_j, \boldsymbol{y}_t, Y_{t-1})}{p(\boldsymbol{y}_t, Y_{t-1})} \\
&= \frac{p(Y_{t-1}) p(A_t = M_j \mid Y_{t-1}) p(\boldsymbol{y}_t \mid A_t = M_j, Y_{t-1})}{p(Y_{t-1}) p(\boldsymbol{y}_t \mid Y_{t-1})} \\
&= \frac{\pi_j(t|t-1) f_j(t|t-1)}{\sum_{k=1}^{m} \pi_k(t|t-1) f_k(t|t-1)}.
\end{aligned} \tag{6.146}$$

In the Markov case, the conditional probabilities

$$\pi_j(t|t-1) = \Pr(A_t = M_j \big| Y_{t-1})$$

in (6.146) replace the unconditional probabilities, $\pi_j(t) = \Pr(A_t = M_j)$, in (6.144).

To evaluate (6.146), we must be able to calculate $\pi_j(t|t-1)$ and $f_j(t|t-1)$. We will discuss the calculation of $f_j(t|t-1)$ after this example. To derive $\pi_j(t|t-1)$, note,

$$\begin{aligned}
\pi_j(t|t-1) &= \Pr(A_t = M_j \big| Y_{t-1}) \\
&= \sum_{i=1}^{m} \Pr(A_t = M_j, A_{t-1} = M_i \big| Y_{t-1}) \\
&= \sum_{i=1}^{m} \Pr(A_t = M_j \big| A_{t-1} = M_i) \Pr(A_{t-1} = M_i \big| Y_{t-1}) \\
&= \sum_{i=1}^{m} \pi_{ij} \pi_i(t-1|t-1). \qquad (6.147)
\end{aligned}$$

Expression (6.145) comes from equation (6.147), where, as previously noted, we replace $\pi_j(t|t-1)$ by $\pi_j(t)$.

The difficulty in extending the approach here to the Markov case is the dependence among the $\boldsymbol{y}_t$, which makes it necessary to enumerate over all possible histories to derive the filtering equations. This problem will be evident when we derive the conditional density $f_j(t|t-1)$. Equation (6.145) has $\pi_j(t)$ as a function of the past observations, Y_{t-1}, which is inconsistent with our model assumption. Nevertheless, this seems to be a reasonable compromise that allows the data to modify the probabilities $\pi_j(t)$, without having to develop a highly computer-intensive technique.

As previously suggested, the computation of $f_j(t|t-1)$, without some approximations, is highly computer-intensive. To evaluate $f_j(t|t-1)$, consider the event

$$A_1 = M_{j_1}, \ \ldots \ , A_{t-1} = M_{j_{t-1}}, \qquad (6.148)$$

for $j_i = 1, \ldots, m$, and $i = 1, \ldots, t-1$, which specifies a specific set of measurement matrices through the past; we will write this event as $A_{(t-1)} = M_{(\ell)}$. Because m^{t-1} possible outcomes exist for $A_1, \ldots, A_{t-1}$, the index ℓ runs through $\ell = 1, \ldots, m^{t-1}$. Using this notation, we may write

$$\begin{aligned}
&f_j(t|t-1) \\
&\quad = \sum_{\ell=1}^{m^{t-1}} \Pr\{A_{(t-1)} = M_{(\ell)} \mid Y_{t-1}\} f(\boldsymbol{y}_t \mid Y_{t-1}, A_t = M_j, A_{(t-1)} = M_{(\ell)}) \\
&\quad \equiv \sum_{\ell=1}^{m^{t-1}} \alpha(\ell) \ \mathrm{N}\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{\mu}_{tj}(\ell), \Sigma_{tj}(\ell)\right), \quad j = 1, \ldots, m, \qquad (6.149)
\end{aligned}$$

where the notation $\mathrm{N}(\cdot \mid \boldsymbol{b}, B)$ represents the normal density with mean vector $\boldsymbol{b}$ and variance–covariance matrix B. That is, $f_j(t|t-1)$ is a mixture of normals with non-negative weights $\alpha(\ell) = \Pr\{A_{(t-1)} = M_{(\ell)} \mid Y_{t-1}\}$ such that $\sum_\ell \alpha(\ell) = 1$, and with each normal distribution having mean vector

$$\boldsymbol{\mu}_{tj}(\ell) = M_j \boldsymbol{x}_t^{t-1}(\ell) = M_j E[\boldsymbol{x}_t \mid Y_{t-1}, A_{(t-1)} = M_{(\ell)}] \tag{6.150}$$

and covariance matrix

$$\Sigma_{tj}(\ell) = M_j P_t^{t-1}(\ell) M_j' + R. \tag{6.151}$$

This result follows because the conditional distribution of $\boldsymbol{y}_t$ in (6.149) is identical to the fixed measurement matrix case presented in §4.2. The values in (6.150) and (6.151), and hence the densities, $f_j(t|t-1)$, for $j = 1, \ldots, m$, can be obtained directly from the Kalman filter, Property 6.1, with the measurement matrices $A_{(t-1)}$ fixed at $M_{(\ell)}$.

Although $f_j(t|t-1)$ is given explicitly in (6.149), its evaluation is highly computer intensive. For example, with $m = 2$ states and $n = 20$ observations, we have to filter over $2 + 2^2 + \cdots + 2^{20}$ possible sample paths; note, $2^{20} = 1,048,576$. One remedy is to trim (remove), at each t, highly improbable sample paths; that is, remove events in (6.148) with extremely small probability of occurring, and then evaluate $f_j(t|t-1)$ as if the trimmed sample paths could not have occurred. Another alternative, as suggested by Gordon and Smith (1990) and Shumway and Stoffer (1991), is to approximate $f_j(t|t-1)$ using the closest (in the sense of Kulback–Leibler distance) normal distribution. In this case, the approximation leads to choosing normal distribution with the same mean and variance associated with $f_j(t|t-1)$; that is, we approximate $f_j(t|t-1)$ by a normal with mean $M_j \boldsymbol{x}_t^{t-1}$ and variance Σ_{tj} given in (6.143).

To develop a procedure for maximum likelihood estimation, the joint density of the data is

$$\begin{aligned} f(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) &= \prod_{t=1}^{n} f(\boldsymbol{y}_t | Y_{t-1}) \\ &= \prod_{t=1}^{n} \sum_{j=1}^{m} \Pr(A_t = M_j | Y_{t-1}) f(\boldsymbol{y}_t | A_t = M_j, Y_{t-1}), \end{aligned}$$

and hence, the likelihood can be written as

$$\ln L_Y(\Theta) = \sum_{t=1}^{n} \ln \left(\sum_{j=1}^{m} \pi_j(t) f_j(t|t-1) \right). \tag{6.152}$$

For the hidden Markov model, $\pi_j(t)$ would be replaced by $\pi_j(t|t-1)$. In (6.152), we will use the normal approximation to $f_j(t|t-1)$. That is, henceforth, we will consider $f_j(t|t-1)$ as the normal, $\mathrm{N}(M_j \boldsymbol{x}_t^{t-1}, \Sigma_{tj})$, density,

where $\boldsymbol{x}_t^{t-1}$ is given in (6.137) and Σ_{tj} is given in (6.143). We may consider maximizing (6.152) directly as a function of the parameters $\Theta = \{\boldsymbol{\mu}_0, \Phi, Q, R\}$ using a Newton method, or we may consider applying the EM algorithm to the complete data likelihood.

To apply the EM algorithm as in §6.3, we call $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, $A_1, \ldots, A_n$, and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, the complete data, with likelihood given by

$$\begin{aligned} -2\ln L_{X,A,Y}(\Theta) &= \ln|\Sigma_0| + (\boldsymbol{x}_0 - \boldsymbol{\mu}_0)'\Sigma_0^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_0) \\ &\quad + n\ln|Q| + \sum_{t=1}^{n}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})'Q^{-1}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1}) \\ &\quad - 2\sum_{t=1}^{n}\sum_{j=1}^{m} I(A_t = M_j)\ln\pi_j(t) + n\ln|R| \\ &\quad + \sum_{t=1}^{n}\sum_{j=1}^{m} I(A_t = M_j)(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)'R^{-1}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t). \end{aligned} \tag{6.153}$$

As discussed in §6.3, we require the minimization of the conditional expectation

$$Q\left(\Theta \mid \Theta^{(k-1)}\right) = E\left\{-2\ln L_{X,A,Y}(\Theta) \;\middle|\; Y_n, \Theta^{(k-1)}\right\}, \tag{6.154}$$

with respect to Θ at each iteration, $k = 1, 2, \ldots$. The calculation and maximization of (6.154) is similar to the case of (6.65). In particular, with

$$\pi_j(t|n) = E[I(A_t = M_j) \mid Y_n], \tag{6.155}$$

we obtain on iteration k,

$$\pi_j^{(k)}(t) = \pi_j(t|n), \tag{6.156}$$

$$\boldsymbol{\mu}_0^{(k)} = \boldsymbol{x}_0^n, \tag{6.157}$$

$$\Phi^{(k)} = S_{10}S_{00}^{-1}, \tag{6.158}$$

$$Q^{(k)} = n^{-1}\left(S_{11} - S_{10}S_{00}^{-1}S_{10}'\right), \tag{6.159}$$

and

$$R^{(k)} = n^{-1}\sum_{t=1}^{n}\sum_{j=1}^{m}\pi_j(t|n)\left[(\boldsymbol{y}_t - M_j\boldsymbol{x}_t^n)(\boldsymbol{y}_t - M_j\boldsymbol{x}_t^n)' + M_jP_t^nM_j'\right]. \tag{6.160}$$

where S_{11}, S_{10}, S_{00} are given in (6.67)-(6.69). As before, at iteration k, the filters and the smoothers are calculated using the current values of the parameters, $\Theta^{(k-1)}$, and Σ_0 is held fixed. Filtering is accomplished by using (6.137)-(6.141). Smoothing is derived in a similar manner to the derivation of the filter, and one is led to the smoother given in Properties 6.2 and 6.3, with one exception, the initial smoother covariance, (6.55), is now

$$P^n_{n,n-1} = \sum_{j=1}^{m} \pi_j(n|n)(I - K_{tj}M_j)\Phi P^{n-1}_{n-1}. \tag{6.161}$$

Unfortunately, the computation of $\pi_j(t|n)$ is excessively complicated, and requires integrating over mixtures of normal distributions. Shumway and Stoffer (1991) suggest approximating the smoother $\pi_j(t|n)$ by the filter $\pi_j(t|t)$, and find the approximation works well.

Example 6.17 Analysis of the Influenza Data

We use the results of this section to analyze the U.S. monthly pneumonia and influenza mortality data presented in §5.4, Figure 5.7. Letting y_t denote the mortality caused by pneumonia and influenza at month t, we model y_t in terms of a structural component model coupled with a hidden Markov process that determines whether a flu epidemic exists.

The model consists of three structural components. The first component, x_{t1}, is an AR(2) process chosen to represent the periodic (seasonal) component of the data,

$$x_{t1} = \alpha_1 x_{t-1,1} + \alpha_2 x_{t-2,1} + w_{t1}, \tag{6.162}$$

where w_{t1} is white noise, with $\text{var}(w_{t1}) = \sigma_1^2$. The second component, x_{t2}, is an AR(1) process with a nonzero constant term, which is chosen to represent the sharp rise in the data during an epidemic,

$$x_{t2} = \beta_0 + \beta_1 x_{t-1,2} + w_{t2}, \tag{6.163}$$

where w_{t2} is white noise, with $\text{var}(w_{t2}) = \sigma_2^2$. The third component, x_{t3}, is a fixed trend component given by,

$$x_{t3} = x_{t-1,3} + w_{t3}, \tag{6.164}$$

where $\text{var}(w_{t3}) = 0$. The case in which $\text{var}(w_{t3}) > 0$, which corresponds to a stochastic trend (random walk), was tried here, but the estimation became unstable, and lead to us fitting a fixed, rather than stochastic, trend. Thus, in the final model, the trend component satisfies $\nabla x_{t3} = 0$; recall in Example 5.6 the data were also differenced once before fitting the model.

Throughout the years, periods of normal influenza mortality (state 1) are modeled as

$$y_t = x_{t1} + x_{t3} + v_t, \tag{6.165}$$

where the measurement error, v_t, is white noise with $\text{var}(v_t) = \sigma_v^2$. When an epidemic occurs (state 2), mortality is modeled as

$$y_t = x_{t1} + x_{t2} + x_{t3} + v_t. \tag{6.166}$$

The model specified in (6.162)–(6.166) can be written in the general state-space form. The state equation is

Table 6.3. Estimation Results for Influenza Data

Parameter	Initial Model Estimates	Final Model Estimates
α_1	1.422 (.100)	1.406 (.079)
α_2	−.634 (.089)	−.622 (.069)
β_0	.276 (.056)	.210 (.025)
β_1	-.312 (.218)	—
σ_1	.023 (.003)	.023 (.005)
σ_2	.108 (.017)	.112 (.017)
σ_v	.002 (.009)	—

Estimated standard errors in parentheses

$$\begin{pmatrix} x_{t1} \\ x_{t-1,1} \\ x_{t2} \\ x_{t3} \end{pmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \beta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{t-1,1} \\ x_{t-2,1} \\ x_{t-1,2} \\ x_{t-1,3} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \beta_0 \\ 0 \end{pmatrix} + \begin{pmatrix} w_{t1} \\ 0 \\ w_{t2} \\ 0 \end{pmatrix}. \tag{6.167}$$

Of course, (6.167) can be written in the standard state-equation form as

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \Upsilon u_t + \boldsymbol{w}_t, \tag{6.168}$$

where $\boldsymbol{x}_t = (x_{t1}, x_{t-1,1}, x_{t2}, x_{t3})'$, $\Upsilon = (0, 0, \beta_0, 0)'$, $u_t \equiv 1$, and Q is a 4×4 matrix with σ_1^2 as the (1,1)-element, σ_2^2 as the (3,3)-element, and the remaining elements set equal to zero. The observation equation is

$$y_t = A_t \boldsymbol{x}_t + v_t, \tag{6.169}$$

where A_t is 1×4, and v_t is white noise with $\mathrm{var}(v_t) = R = \sigma_v^2$. We assume all components of variance w_{t1}, w_{t2}, and v_t are uncorrelated.

As discussed in (6.165) and (6.166), A_t can take one of two possible forms

$$\begin{aligned} A_t &= M_1 = [1, 0, 0, 1] \quad \text{no epidemic}, \\ A_t &= M_2 = [1, 0, 1, 1] \quad \text{epidemic}, \end{aligned}$$

corresponding to the two possible states of (1) no flu epidemic and (2) flu epidemic, such that $\Pr(A_t = M_1) = 1 - \Pr(A_t = M_2)$. In this example, we will assume A_t is a hidden Markov chain, and hence we use the updating equations given in Example 6.16, (6.146) and (6.147), with transition probabilities $\pi_{11} = \pi_{22} = .75$ (and, thus, $\pi_{12} = \pi_{21} = .25$).

Parameter estimation was accomplished using a quasi-Newton–Raphson procedure to maximize the approximate log likelihood given in (6.152), with initial values of $\pi_1(1|0) = \pi_2(1|0) = .5$. Table 6.3 shows the results of the estimation procedure. On the initial fit, two estimates are not significant, namely, $\widehat{\beta}_1$ and $\widehat{\sigma}_v$. When $\sigma_v^2 = 0$, there is no measurement error, and the variability in data is explained solely by the variance components of the

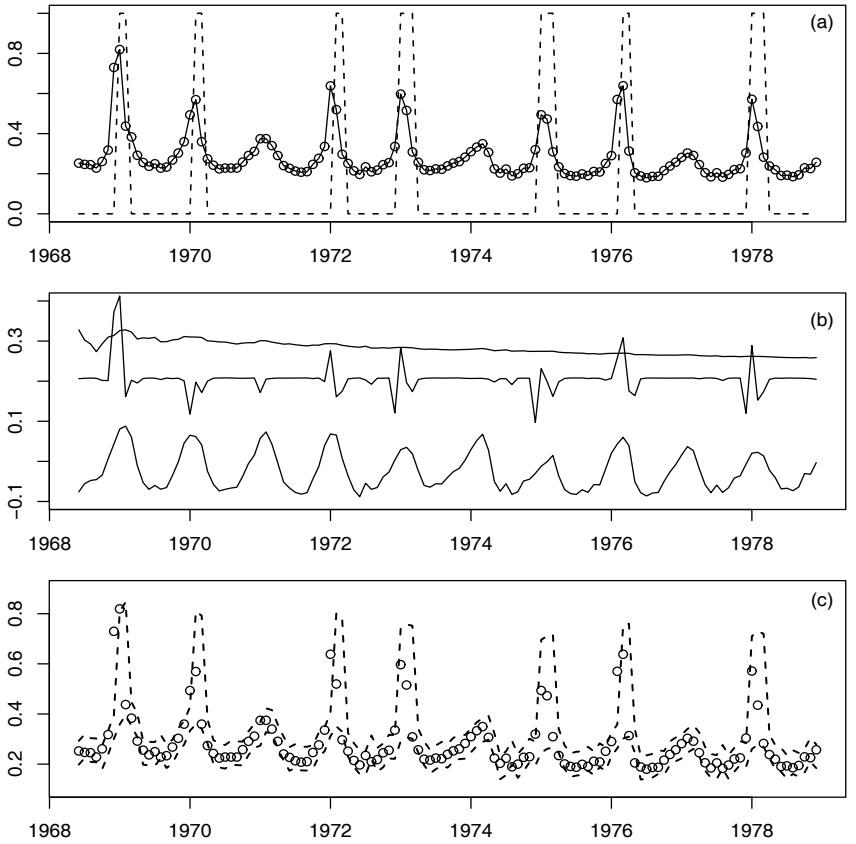

Fig. 6.10. (a) Influenza data, y_t, (line–points) and a prediction indicator (0 or 1) that an epidemic occurs in month t given the data up to month $t-1$ (dashed line). (b) The three filtered structural components of influenza mortality: $\widehat{x}^t_{t1}$ (*cyclic trace*), $\widehat{x}^t_{t2}$ (*spiked trace*), and $\widehat{x}^t_{t3}$ (*negative linear trace*). (c) One-month-ahead predictions shown as upper and lower limits $\widehat{y}^{t-1}_t \pm 2\sqrt{P^{t-1}_t}$ (dashed lines), of the number of pneumonia and influenza deaths, and y_t (points).

state system, namely, σ_1^2 and σ_2^2. The case in which $\beta_1 = 0$ corresponds to a simple level shift during a flu epidemic. In the final model, with β_1 and σ_v^2 removed, the estimated level shift ($\widehat{\beta}_0$) corresponds to an increase in mortality by about .2 per 1000 during a flu epidemic. The estimates for the final model are also listed in Table 6.3.

Figure 6.10(a) shows a plot of the data, y_t, for the ten-year period of 1969-1978 as well as an indicator that takes the value of 1 if the estimated approximate conditional probability exceeds .5, i.e., $\widehat{\pi}_2(t|t-1) > .5$. The estimated prediction probabilities do a reasonable job of predicting a flu epidemic, although the peak in 1972 is missed.

Figure 6.10(b) shows the estimated filtered values (that is, filtering is done using the parameter estimates) of the three components of the model, x_{t1}^t, x_{t2}^t, and x_{t3}^t. Except for initial instability (which is not shown), $\widehat{x}_{t1}^t$ represents the seasonal (cyclic) aspect of the data, $\widehat{x}_{t2}^t$ represents the spikes during a flu epidemic, and $\widehat{x}_{t3}^t$ represents the slow decline in flu mortality over the ten-year period of 1969-1978.

One-month-ahead prediction, say, $\widehat{y}_t^{t-1}$, is obtained as

$$\widehat{y}_t^{t-1} = M_1 \widehat{\boldsymbol{x}}_t^{t-1} \quad \text{if} \quad \widehat{\pi}_1(t|t-1) > \widehat{\pi}_2(t|t-1),$$

$$\widehat{y}_t^{t-1} = M_2 \widehat{\boldsymbol{x}}_t^{t-1} \quad \text{if} \quad \widehat{\pi}_1(t|t-1) \leq \widehat{\pi}_2(t|t-1).$$

Of course, $\widehat{\boldsymbol{x}}_t^{t-1}$ is the estimated state prediction, obtained via the filter presented in (6.137)-(6.141) (with the addition of the constant term in the model) using the estimated parameters. The results are shown in Figure 6.10(c). The precision of the forecasts can be measured by the innovation variances, Σ_{t1} when no epidemic is predicted, and Σ_{t2} when an epidemic is predicted. These values become stable quickly, and when no epidemic is predicted, the estimated standard error of the prediction is approximately .02 (this is the square root of Σ_{t1} for t large); when a flu epidemic is predicted, the estimated standard error of the prediction is approximately .11.

The results of this analysis are impressive given the small number of parameters and the degree of approximation that was made to obtain a computationally simple method for fitting a complex model. In particular, as seen in Figure 6.10(a), the model is never fooled as to when a flu epidemic will occur. This result is particularly impressive, given that, for example, in 1971, it appeared as though an epidemic was about to begin, but it never was realized, and the model predicted no flu epidemic that year. As seen in Figure 6.10(c), the predicted mortality tends to be underestimated during the peaks, but the true values are typically within one standard error of the predicted value. Further evidence of the strength of this technique can be found in the example given in Shumway and Stoffer (1991).

The R code for the final model estimation is as follows.

```
y = as.matrix(flu); num = length(y); nstate = 4;
M1 = as.matrix(cbind(1,0,0,1))  # obs matrix normal
M2 = as.matrix(cbind(1,0,1,1))  # obs matrix flu epi
prob = matrix(0,num,1); yp = y  # to store pi2(t|t-1) & y(t|t-1)
xfilter = array(0, dim=c(nstate,1,num)) # to store x(t|t)
# Function to Calculate Likelihood
Linn = function(para){
  alpha1=para[1]; alpha2=para[2]; beta0=para[3]
  sQ1=para[4];  sQ2=para[5];  like=0
  xf=matrix(0, nstate, 1)  # x filter
  xp=matrix(0, nstate, 1)  # x pred
  Pf=diag(.1, nstate)      # filter cov
  Pp=diag(.1, nstate)      # pred cov
  pi11 <- .75 -> pi22;  pi12 <- .25 -> pi21; pif1 <- .5 -> pif2
```

```
  phi=matrix(0,nstate,nstate)
  phi[1,1]=alpha1; phi[1,2]=alpha2; phi[2,1]=1; phi[4,4]=1
  Ups = as.matrix(rbind(0,0,beta0,0))
  Q = matrix(0,nstate,nstate)
  Q[1,1]=sQ1^2; Q[3,3]=sQ2^2; R=0  # R=0 in final model
  # begin filtering #
    for(i in 1:num){
    xp = phi%*%xf + Ups; Pp = phi%*%Pf%*%t(phi) + Q
    sig1 = as.numeric(M1%*%Pp%*%t(M1) + R)
    sig2 = as.numeric(M2%*%Pp%*%t(M2) + R)
    k1 = Pp%*%t(M1)/sig1; k2 = Pp%*%t(M2)/sig2
    e1 = y[i]-M1%*%xp; e2 = y[i]-M2%*%xp
    pip1 = pif1*pi11 + pif2*pi21; pip2 = pif1*pi12 + pif2*pi22;
    den1 = (1/sqrt(sig1))*exp(-.5*e1^2/sig1);
    den2 = (1/sqrt(sig2))*exp(-.5*e2^2/sig2);
    denom = pip1*den1 + pip2*den2;
    pif1 = pip1*den1/denom; pif2 = pip2*den2/denom;
    pif1=as.numeric(pif1); pif2=as.numeric(pif2)
    e1=as.numeric(e1); e2=as.numeric(e2)
    xf = xp + pif1*k1*e1 + pif2*k2*e2
    eye = diag(1, nstate)
    Pf = pif1*(eye-k1%*%M1)%*%Pp + pif2*(eye-k2%*%M2)%*%Pp
    like = like - log(pip1*den1 + pip2*den2)
    prob[i]<<-pip2; xfilter[,,i]<<-xf; innov.sig<<-c(sig1,sig2)
    yp[i]<<-ifelse(pip1 > pip2, M1%*%xp, M2%*%xp)  }
 return(like)   }
# Estimation
alpha1=1.4; alpha2=-.5; beta0=.3; sQ1=.1; sQ2=.1
init.par = c(alpha1, alpha2, beta0, sQ1, sQ2)
(est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
    control=list(trace=1,REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
u = cbind(estimate=est$par, SE)
rownames(u)=c("alpha1","alpha2","beta0","sQ1","sQ2"); u
```

```
                estimate          SE
       alpha1  1.40570967 0.078587727
       alpha2 -0.62198715 0.068733109
       beta0   0.21049042 0.024625302
       sQ1     0.02310306 0.001635291
       sQ2     0.11217287 0.016684663
```

```
# Graphics
predepi = ifelse(prob<.5,0,1); k = 6:length(y)
Time = time(flu)[k]
par(mfrow=c(3,1), mar=c(2,3,1,1)+.1, cex=.9)
plot(Time, y[k], type="o", ylim=c(0,1),ylab="")
 lines(Time, predepi[k],  lty="dashed", lwd=1.2)
 text(1979,.95,"(a)")
plot(Time, xfilter[1,,k], type="l", ylim=c(-.1,.4), ylab="")
```

```
lines(Time, xfilter[3,,k]); lines(Time, xfilter[4,,k])
text(1979,.35,"(b)")
plot(Time, y[k], type="p", pch=1, ylim=c(.1,.9),ylab="")
prde1 = 2*sqrt(innov.sig[1]); prde2 = 2*sqrt(innov.sig[2])
prde = ifelse(predepi[k]<.5, prde1,prde2)
lines(Time, yp[k]+prde, lty=2, lwd=1.5)
lines(Time, yp[k]-prde, lty=2, lwd=1.5)
text(1979,.85,"(c)")
```

6.9 Stochastic Volatility

Recently, there has been considerable interest in stochastic volatility models. These models are similar to the ARCH models presented in Chapter 5, but they add a stochastic noise term to the equation for σ_t. Recall from §5.4 that a GARCH$(1,1)$ model for a return, which we denote here by r_t, is given by

$$r_t = \sigma_t \epsilon_t \tag{6.170}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \tag{6.171}$$

where ϵ_t is Gaussian white noise. If we define

$$h_t = \log \sigma_t^2 \quad \text{and} \quad y_t = \log r_t^2,$$

then (6.170) can be written as

$$y_t = h_t + \log \epsilon_t^2. \tag{6.172}$$

Equation (6.172) is considered the observation equation, and the stochastic variance h_t is considered to be an unobserved state process. Instead of (6.171), however, the model assumes the volatility process follows, in its basic form, an autoregression,

$$h_t = \phi_0 + \phi_1 h_{t-1} + w_t, \tag{6.173}$$

where w_t is white Gaussian noise with variance σ_w^2.

Together, (6.172) and (6.173) make up the stochastic volatility model due to Taylor (1982). If ϵ_t^2 had a log-normal distribution, (6.172)–(6.173) would form a Gaussian state-space model, and we could then use standard DLM results to fit the model to data. Unfortunately, $y_t = \log r_t^2$ is rarely normal, so we typically keep the ARCH normality assumption on ϵ_t; in which case, $\log \epsilon_t^2$ is distributed as the log of a chi-squared random variable with one degree of freedom. This density is given by

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(e^x - x\right)\right\} \quad -\infty < x < \infty, \tag{6.174}$$

and its mean and variance are -1.27 and $\pi^2/2$, respectively; the density (6.174) is highly skewed with a long tail on the left (see Figure 6.12).

Various approaches to the fitting of stochastic volatility models have been examined; these methods include a wide range of assumptions on the observational noise process. A good summary of the proposed techniques, both Bayesian (via MCMC) and non-Bayesian approaches (such as quasi-maximum likelihood estimation and the EM algorithm), can be found in Jacquier et al. (1994), and Shephard (1996). Simulation methods for classical inference applied to stochastic volatility models are discussed in Danielson (1994) and Sandmann and Koopman (1998).

Kim, Shephard and Chib (1998) proposed modeling the log of a chi-squared random variable by a mixture of seven normals to approximate the first four moments of the observational error distribution; the mixture is fixed and no additional model parameters are added by using this technique. The basic model assumption that ϵ_t is Gaussian is unrealistic for most applications. In an effort to keep matters simple but more general (in that we allow the observational error dynamics to depend on parameters that will be fitted), our method of fitting stochastic volatility models is to retain the Gaussian state equation (6.173), but to write the observation equation, with $y_t = \log r_t^2$, as

$$y_t = \alpha + h_t + \eta_t, \tag{6.175}$$

where η_t is white noise, whose distribution is a mixture of two normals, one centered at zero. In particular, we write

$$\eta_t = I_t z_{t0} + (1 - I_t) z_{t1}, \tag{6.176}$$

where I_t is an iid Bernoulli process, $\Pr\{I_t = 0\} = \pi_0$, $\Pr\{I_t = 1\} = \pi_1$ $(\pi_0 + \pi_1 = 1)$, $z_{t0} \sim$ iid $\mathrm{N}(0, \sigma_0^2)$, and $z_{t1} \sim$ iid $\mathrm{N}(\mu_1, \sigma_1^2)$.

The advantage to this model is that it is easy to fit because it uses normality. In fact, the model equations (6.173) and (6.175)-(6.176) are similar to those presented in Peña and Guttman (1988), who used the idea to obtain a robust Kalman filter, and, as previously mentioned, in Kim et al. (1998). The material presented in §6.8 applies here, and in particular, the filtering equations for this model are

$$h_{t+1}^t = \phi_0 + \phi_1 h_t^{t-1} + \sum_{j=0}^{1} \pi_{tj} K_{tj} \epsilon_{tj}, \tag{6.177}$$

$$P_{t+1}^t = \phi_1^2 P_t^{t-1} + \sigma_w^2 - \sum_{j=0}^{1} \pi_{tj} K_{tj}^2 \Sigma_{tj}, \tag{6.178}$$

$$\epsilon_{t0} = y_t - \alpha - h_t^{t-1}, \qquad \epsilon_{t1} = y_t - \alpha - h_t^{t-1} - \mu_1, \tag{6.179}$$

$$\Sigma_{t0} = P_t^{t-1} + \sigma_0^2, \qquad \Sigma_{t1} = P_t^{t-1} + \sigma_1^2, \tag{6.180}$$

$$K_{t0} = \phi_1 P_t^{t-1} \,/\, \Sigma_{t0}, \qquad K_{t1} = \phi_1 P_t^{t-1} \,/\, \Sigma_{t1}. \tag{6.181}$$

To complete the filtering, we must be able to assess the probabilities $\pi_{t1} = \Pr(I_t = 1 \mid y_1, \ldots, y_t)$, for $t = 1, \ldots, n$; of course, $\pi_{t0} = 1 - \pi_{t1}$. Let $f_j(t \mid t-1)$

denote the conditional density of y_t given the past $y_1, \ldots, y_{t-1}$, and $I_t = j$ $(j = 0, 1)$. Then,

$$\pi_{t1} = \frac{\pi_1 f_1(t \mid t-1)}{\pi_0 f_0(t \mid t-1) + \pi_1 f_1(t \mid t-1)}, \tag{6.182}$$

where we assume the distribution π_j, for $j = 0, 1$ has been specified *a priori*. If the investigator has no reason to prefer one state over another the choice of uniform priors, $\pi_1 = 1/2$, will suffice. Unfortunately, it is computationally difficult to obtain the exact values of $f_j(t \mid t-1)$; although we can give an explicit expression of $f_j(t \mid t-1)$, the actual computation of the conditional density is prohibitive. A viable approximation, however, is to choose $f_j(t \mid t-1)$ to be the normal density, $\mathrm{N}(h_t^{t-1} + \mu_j, \Sigma_{tj})$, for $j = 0, 1$ and $\mu_0 = 0$; see §6.8 for details.

The innovations filter given in (6.177)–(6.182) can be derived from the Kalman filter by a simple conditioning argument; e.g., to derive (6.177), write

$$\begin{aligned}
\mathrm{E}\left(h_{t+1} \,\middle|\, y_1, \ldots, y_t\right) &= \sum_{j=0}^{1} \mathrm{E}\left(h_{t+1} \,\middle|\, y_1, \ldots, y_t, I_t = j\right) \Pr(I_t = j \,|\, y_1, \ldots, y_t) \\
&= \sum_{j=0}^{1} \left(\phi_0 + \phi_1 h_t^{t-1} + K_{tj}\epsilon_{tj}\right) \pi_{tj} \\
&= \phi_0 + \phi_1 h_t^{t-1} + \sum_{j=0}^{1} \pi_{tj} K_{tj} \epsilon_{tj}.
\end{aligned}$$

Estimation of the parameters, $\Theta = (\phi_0, \phi_1, \sigma_0^2, \mu_1, \sigma_1^2, \sigma_w^2)'$, is accomplished via MLE based on the likelihood given by

$$\ln L_Y(\Theta) = \sum_{t=1}^{n} \ln \left(\sum_{j=0}^{1} \pi_j \; f_j(t \mid t-1) \right), \tag{6.183}$$

where the density $f_j(t \mid t-1)$ is approximated by the normal density, $\mathrm{N}(h_t^{t-1} + \mu_j, \sigma_j^2)$, previously mentioned. We may consider maximizing (6.183) directly as a function of the parameters Θ using a Newton method, or we may consider applying the EM algorithm to the complete data likelihood.

Example 6.18 Analysis of the New York Stock Exchange Returns

The top of Figure 6.11 shows the log of the squares of returns, $y_t = \log r_t^2$, for 200 of the 2000 daily observations of the NYSE previously displayed in Figure 1.4. Model (6.173) and (6.175)–(6.176), with π_1 fixed at .5, was fit to the data using a quasi-Newton–Raphson method to maximize (6.183). The results are given in Table 6.4. Figure 6.12 compares the density of the log of a χ_1^2 with the fitted normal mixture; we note the data indicate a substantial amount of probability in the upper tail that the log-χ_1^2 distribution misses.

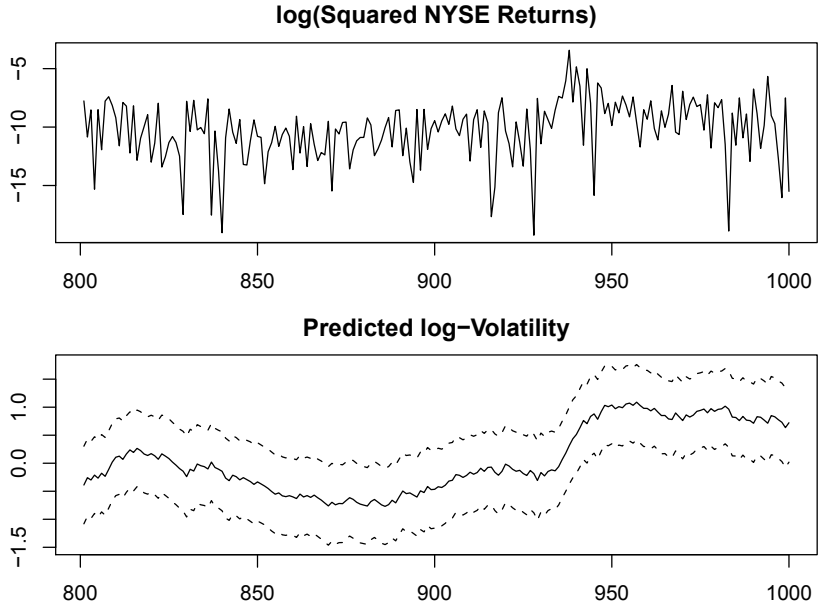

Fig. 6.11. Two hundred observations of $y_t = \log r_t^2$, for $801 \leq t \leq 1000$, where r_t is the daily return of the NYSE (top); the crash of October 19, 1987 occurs at $t = 938$. Corresponding one-step-ahead predicted log volatility, $\log \sigma_t^2$, with ± 2 standard prediction errors (bottom).

Finally, the bottom of Figure 6.11 shows y_t for $800 \leq t \leq 1000$, which includes the crash of October 19, 1987, with $y_t^{t-1} = \widehat{\alpha} + h_t^{t-1}$ superimposed on the graph; compare with Figure 5.6. Also displayed are error bounds.

The R code when ϕ_0 is included in the model is as follows.

```
y = log(nyse^2)
num = length(y)
# Initial Parameters
phi0=0; phi1=.95; sQ=.2; alpha=mean(y); sR0=1; mu1=-3; sR1=2
init.par = c(phi0, phi1, sQ, alpha, sR0, mu1, sR1)
# Innovations Likelihood
Linn = function(para){
  phi0=para[1]; phi1=para[2]; sQ=para[3]; alpha=para[4]
  sR0=para[5]; mu1=para[6]; sR1=para[7]
  sv = SVfilter(num,y,phi0,phi1,sQ,alpha,sR0,mu1,sR1)
  return(sv$like)    }
# Estimation
(est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
     control=list(trace=1,REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
u = cbind(estimates=est$par, SE)
rownames(u)=c("phi0","phi1","sQ","alpha","sigv0","mu1","sigv1"); u
```

Table 6.4. Estimation Results for the NYSE Fit

Parameter	Estimate	Estimated Standard Error
ϕ_0	−.006	.016
ϕ_1	.988	.007
σ_w	.091	.027
α	−9.613	1.269
σ_0	1.220	.065
μ_1	−2.292	.205
σ_1	2.683	.105

```
# Graphics   (need filters at the estimated parameters)
phi0=est$par[1]; phi1=est$par[2]; sQ=est$par[3]; alpha=est$par[4]
sR0=est$par[5]; mu1=est$par[6]; sR1=est$par[7]
sv = SVfilter(num,y,phi0,phi1,sQ,alpha,sR0,mu1,sR1)
# densities plot (f is chi-sq, fm is fitted mixture)
x = seq(-15,6,by=.01)
f = exp(-.5*(exp(x)-x))/(sqrt(2*pi))
f0 = exp(-.5*(x^2)/sR0^2)/(sR0*sqrt(2*pi))
f1 = exp(-.5*(x-mu1)^2/sR1^2)/(sR1*sqrt(2*pi))
fm = (f0+f1)/2
plot(x, f, type="l"); lines(x, fm, lty=2,lwd=2)
dev.new(); par(mfrow=c(2,1)); Time=801:1000
plot(Time, y[Time], type="l", main="log(Squared NYSE Returns)")
plot(Time, sv$xp[Time],type="l", main="Predicted log-Volatility",
     ylim=c(-1.5,1.8), ylab="", xlab="")
 lines(Time, sv$xp[Time]+2*sqrt(sv$Pp[Time]), lty="dashed")
 lines(Time, sv$xp[Time]-2*sqrt(sv$Pp[Time]), lty="dashed")
```

It is possible to use the bootstrap procedure described in §6.7 for the stochastic volatility model, with some minor changes. The following procedure was described in Stoffer and Wall (2004). We develop a vector first-order equation, as was done in (6.124). First, using (6.179), and noting that $y_t = \pi_{t0} y_t + \pi_{t1} y_t$, we may write

$$y_t = \alpha + h_t^{t-1} + \pi_{t0}\epsilon_{t0} + \pi_{t1}(\epsilon_{t1} + \mu_1). \tag{6.184}$$

Consider the standardized innovations

$$e_{tj} = \Sigma_{tj}^{-1/2}\epsilon_{tj}, \qquad j = 0, 1, \tag{6.185}$$

and define the 2×1 vector

$$\boldsymbol{e}_t = \begin{bmatrix} e_{t0} \\ e_{t1} \end{bmatrix}.$$

Also, define the 2×1 vector

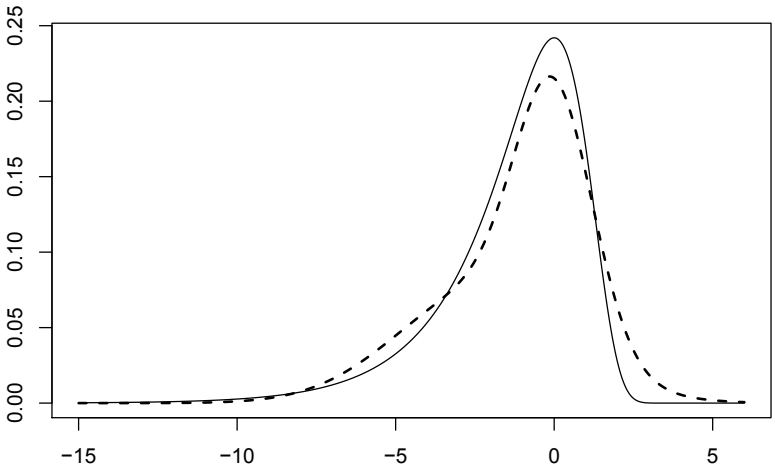

Fig. 6.12. Density of the log of a χ_1^2 as given by (6.174) (solid line) and the fitted normal mixture (dashed line) form the NYSE example.

$$\boldsymbol{\xi}_t = \begin{bmatrix} h_{t+1}^t \\ y_t \end{bmatrix}.$$

Combining (6.177) and (6.184) results in a vector first-order equation for $\boldsymbol{\xi}_t$ given by

$$\boldsymbol{\xi}_t = F\boldsymbol{\xi}_{t-1} + G_t + H_t\boldsymbol{e}_t, \tag{6.186}$$

where

$$F = \begin{bmatrix} \phi_1 & 0 \\ 1 & 0 \end{bmatrix}, \quad G_t = \begin{bmatrix} \phi_0 \\ \alpha + \pi_{t1}\mu_1 \end{bmatrix}, \quad H_t = \begin{bmatrix} \pi_{t0}K_{t0}\Sigma_{t0}^{1/2} & \pi_{t1}K_{t1}\Sigma_{t1}^{1/2} \\ \pi_{t0}\Sigma_{t0}^{1/2} & \pi_{t1}\Sigma_{t1}^{1/2} \end{bmatrix}.$$

Hence, the steps in bootstrapping for this case are the same as steps 1 through 5 described in §5.8, but with (6.124) replaced by the following first-order equation:

$$\boldsymbol{\xi}_t^* = F(\widehat{\Theta})\boldsymbol{\xi}_{t-1}^* + G_t(\widehat{\Theta}; \widehat{\pi}_{t1}) + H_t(\widehat{\Theta}; \widehat{\pi}_{t1})\boldsymbol{e}_t^*, \tag{6.187}$$

where $\widehat{\Theta} = (\widehat{\phi}_0, \widehat{\phi}_1, \widehat{\sigma}_0^2, \widehat{\alpha}, \widehat{\mu}_1, \widehat{\sigma}_1^2, \widehat{\sigma}_w^2)'$ is the MLE of Θ, and $\widehat{\pi}_{t1}$ is estimated via (6.182), replacing $f_1(t \mid t-1)$ and $f_0(t \mid t-1)$ by their respective estimated normal densities ($\widehat{\pi}_{t0} = 1 - \widehat{\pi}_{t1}$).

Example 6.19 Analysis of the U.S. GNP Growth Rate

In Example 5.4, we fit an ARCH model to the U.S. GNP growth rate. In this example, we will fit a stochastic volatility model to the residuals from the MA(2) fit on the growth rate (see Example 3.38). Figure 6.13 shows the log of the squared residuals, say y_t, from the MA(2) fit on the U.S. GNP series. The stochastic volatility model (6.172)–(6.176) was then fit to y_t.

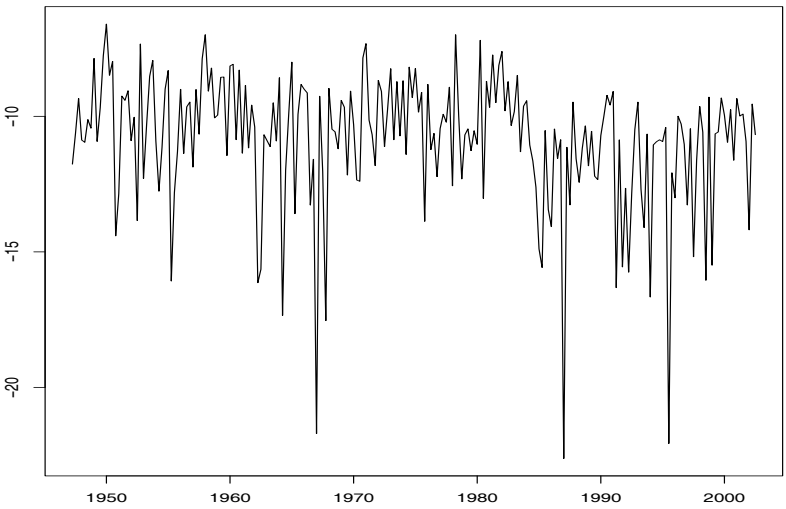

Fig. 6.13. Log of the squared residuals from an MA(2) fit on GNP growth rate.

Table 6.5 shows the MLEs of the model parameters along with their asymptotic SEs assuming the model is correct. Also displayed in Table 6.5 are the means and SEs of $B = 500$ bootstrapped samples. There is some amount of agreement between the asymptotic values and the bootstrapped values. The interest here, however, is not so much in the SEs, but in the actual sampling distribution of the estimates. For example, Figure 6.14 compares the bootstrap histogram and asymptotic normal distribution of $\widehat{\phi}_1$. In this case, the bootstrap distribution exhibits positive kurtosis and skewness which is missed by the assumption of asymptotic normality.

The R code for this example is as follows. We held ϕ_0 at 0 for this analysis because it was not significantly different from 0 in an initial analysis.

```
n.boot = 500   # number of bootstrap replicates
tol = sqrt(.Machine$double.eps)  # convergence tolerance
gnpgr = diff(log(gnp))
fit = arima(gnpgr, order=c(1,0,0))
y = as.matrix(log(resid(fit)^2))
num = length(y)
plot.ts(y, ylab="")
# Initial Parameters
phi1 = .9; sQ = .5; alpha = mean(y); sR0 = 1; mu1 = -3; sR1 = 2.5
init.par = c(phi1, sQ, alpha, sR0, mu1, sR1)
# Innovations Likelihood
Linn=function(para){
  phi1 = para[1]; sQ = para[2]; alpha = para[3]
  sR0 = para[4]; mu1 = para[5]; sR1 = para[6]
  sv = SVfilter(num, y, 0, phi1, sQ, alpha, sR0, mu1, sR1)
  return(sv$like)    }
```

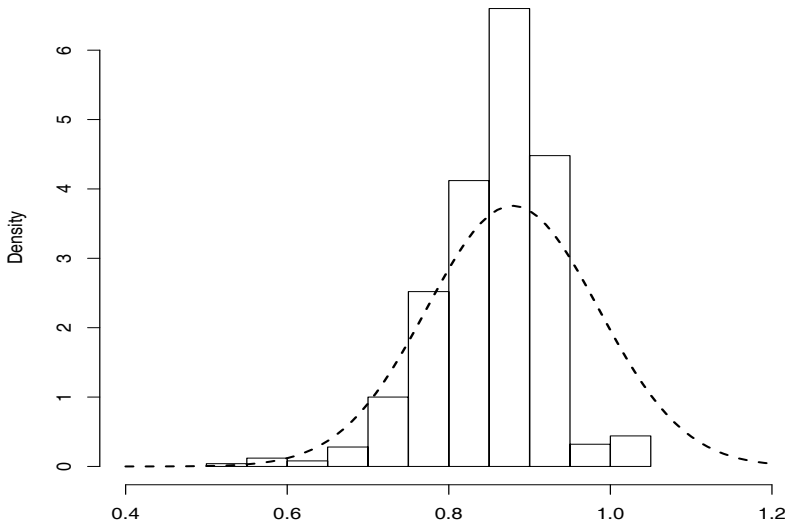

Fig. 6.14. Bootstrap histogram and asymptotic distribution of $\widehat{\phi}_1$ for the U.S. GNP example.

Table 6.5. Estimates and Standard Errors for GNP Example

Parameter	MLE	Asymptotic SE	Bootstrap† SE
ϕ_1	.879	.106	.074
σ_w	.388	.217	.428
α	−9.662	.339	1.536
σ_0	.833	.203	.389
μ_1	−2.341	.495	.437
σ_1	2.452	.293	.330

† Based on 500 bootstrapped samples.

```
# Estimation
(est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
      control=list(trace=1,REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
u = rbind(estimates=est$par, SE)
colnames(u)=c("phi1","sQ","alpha","sig0","mu1","sig1"); u
```

```
               phi1     sQ   alpha   sig0     mu1   sig1
estimates  0.8790 0.3878 -9.6624 0.8325 -2.3412 2.4516
SE         0.1061 0.2172  0.3386 0.2034  0.4952 0.2927
```

```
# Bootstrap
para.star = matrix(0, n.boot, 6)  # to store parameter estimates
```

```
Linn2 = function(para){   # calculate likelihood
  phi1 = para[1]; sQ = para[2]; alpha = para[3]
  sR0 = para[4]; mu1 = para[5]; sR1 = para[6]
  sv = SVfilter(num, y.star, 0, phi1, sQ, alpha, sR0, mu1, sR1)
  return(sv$like)    }
for (jb in 1:n.boot){
 cat("iteration:", jb, "\n")
 phi1 = est$par[1]; sQ = est$par[2]; alpha = est$par[3]
 sR0 = est$par[4]; mu1 = est$par[5]; sR1 = est$par[6]
 Q = sQ^2; R0 = sR0^2; R1 = sR1^2
 sv = SVfilter(num, y, 0, phi1, sQ, alpha, sR0, mu1, sR1)
 sig0 = sv$Pp+R0; sig1 = sv$Pp+R1;
 K0 = sv$Pp/sig0; K1 = sv$Pp/sig1
 inn0 = y-sv$xp-alpha; inn1 = y-sv$xp-mu1-alpha
 den1 = (1/sqrt(sig1))*exp(-.5*inn1^2/sig1)
 den0 = (1/sqrt(sig0))*exp(-.5*inn0^2/sig0)
 fpi1 = den1/(den0+den1)
 # start resampling at t=4
 e0 = inn0/sqrt(sig0); e1 = inn1/sqrt(sig1)
 indx = sample(4:num, replace=TRUE)
 sinn = cbind(c(e0[1:3], e0[indx]), c(e1[1:3], e1[indx]))
 eF = matrix(c(phi1, 1, 0, 0), 2, 2)
 xi = cbind(sv$xp,y) # initialize
   for (i in 4:num){    # generate boot sample
   G = matrix(c(0, alpha+fpi1[i]*mu1), 2, 1)
   h21 = (1-fpi1[i])*sqrt(sig0[i]); h11 = h21*K0[i]
   h22 = fpi1[i]*sqrt(sig1[i]); h12 = h22*K1[i]
   H = matrix(c(h11,h21,h12,h22),2,2)
   xi[i,] = t(eF%*%as.matrix(xi[i-1,],2) + G +
     H%*%as.matrix(sinn[i,],2))}
# Estimates from boot data
y.star = xi[,2]
phi1=.9; sQ=.5; alpha=mean(y.star); sR0=1; mu1=-3; sR1=2.5
init.par = c(phi1, sQ, alpha, sR0, mu1, sR1)   # same as for data
est.star = optim(init.par, Linn2, NULL, method="BFGS",
     control=list(reltol=tol))
para.star[jb,] = cbind(est.star$par[1], abs(est.star$par[2]),
     est.star$par[3], abs(est.star$par[4]), est.star$par[5],
     abs(est.star$par[6])) }
# Some summary statistics and graphics
rmse = rep(NA,6)  # SEs from the bootstrap
for(i in 1:6){
  rmse[i] = sqrt(sum((para.star[,i]-est$par[i])^2)/n.boot)
  cat(i, rmse[i],"\n") }
dev.new(); phi = para.star[,1]
hist(phi, 15, prob=TRUE, main="", xlim=c(.4,1.2), xlab="")
u = seq(.4, 1.2, by=.01)
lines(u,dnorm(u, mean=.8790267, sd=.1061884), lty="dashed", lwd=2)
```

6.10 Nonlinear and Non-normal State-Space Models Using Monte Carlo Methods

Most of this chapter has focused on linear dynamic models assumed to be Gaussian processes. Historically, these models were convenient because analyzing the data was a relatively simple matter. These assumptions cannot cover every situation, and it is advantageous to explore departures from these assumptions. As seen in §6.8, the solution to the nonlinear and non-Gaussian case will require computer-intensive techniques currently in vogue because of the availability of cheap and fast computers. In this section, we take a Bayesian approach to forecasting as our main objective; see West and Harrison (1997) for a detailed account of Bayesian forecasting with dynamic models. Prior to the mid-1980s, a number of approximation methods were developed to filter non-normal or nonlinear processes in an attempt to circumvent the computational complexity of the analysis of such models. For example, the extended Kalman filter and the Gaussian sum filter (Alspach and Sorensen, 1972) are two such methods described in detail in Anderson and Moore (1979). As in the previous section, these techniques typically rely on approximating the non-normal distribution by one or several Gaussian distributions or by some other parametric function.

With the advent of cheap and fast computing, a number of authors developed computer-intensive methods based on numerical integration. For example, Kitagawa (1987) proposed a numerical method based on piecewise linear approximations to the density functions for prediction, filtering, and smoothing for non-Gaussian and nonstationary state-space models. Pole and West (1988) used Gaussian quadrature techniques in a Bayesian analysis of nonlinear dynamic models; West and Harrison (1997, Chapter 13) provide a detailed explanation of these and similar methods. Markov chain Monte Carlo (MCMC) methods refer to Monte Carlo integration methods that use a Markovian updating scheme. We will describe the method in more detail later. The most common MCMC method is the Gibbs sampler, which is essentially a modification of the Metropolis algorithm (Metropolis et al., 1953) developed by Hastings (1970) in the statistical setting and by Geman and Geman (1984) in the context of image restoration. Later, Tanner and Wong (1987) used the ideas in their substitution sampling approach, and Gelfand and Smith (1990) developed the Gibbs sampler for a wide class of parametric models. This technique was first used by Carlin et al. (1992) in the context of general nonlinear and non-Gaussian state-space models. Frühwirth-Schnatter (1994) and Carter and Kohn (1994) built on these ideas to develop efficient Gibbs sampling schemes for more restrictive models.

If the model is linear, that is, (6.1) and (6.2) hold, but the distributions are not Gaussian, a non-Gaussian likelihood can be defined by (6.31) in §6.2, but where $f_0(\cdot)$, $f_w(\cdot)$ and $f_v(\cdot)$ are not normal densities. In this case, prediction and filtering can be accomplished using numerical integration techniques (e.g., Kitagawa, 1987; Pole and West, 1988) or Monte Carlo techniques

(e.g. Frühwirth-Schnatter, 1994; Carter and Kohn, 1994) to evaluate (6.32) and (6.33). Of course, the prediction and filter densities $p_\Theta(\boldsymbol{x}_t \mid Y_{t-1})$ and $p_\Theta(\boldsymbol{x}_t \mid Y_t)$ will no longer be Gaussian and will not generally be of the location-scale form as in the Gaussian case. A rich class of non-normal densities is given in (6.198).

In general, the state-space model can be given by the following equations:

$$\boldsymbol{x}_t = F_t(\boldsymbol{x}_{t-1}, \boldsymbol{w}_t) \quad \text{and} \quad \boldsymbol{y}_t = H_t(\boldsymbol{x}_t, \boldsymbol{v}_t), \tag{6.188}$$

where F_t and H_t are known functions that may depend on parameters Θ and $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are white noise processes. The main component of the model retained by (6.188) is that the states are Markov, and the observations are conditionally independent, but we do not necessarily assume F_t and H_t are linear, or $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are Gaussian. Of course, if $F_t(\boldsymbol{x}_{t-1}, \boldsymbol{w}_t) = \Phi_t \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$ and $H_t(\boldsymbol{x}_t, \boldsymbol{v}_t) = A_t \boldsymbol{x}_t + \boldsymbol{v}_t$ and $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are Gaussian, we have the standard DLM (exogenous variables can be added to the model in the usual way). In the general model, (6.188), the complete data likelihood is given by

$$L_{X,Y}(\Theta) = p_\Theta(\boldsymbol{x}_0) \prod_{t=1}^{n} p_\Theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) p_\Theta(\boldsymbol{y}_t \mid \boldsymbol{x}_t), \tag{6.189}$$

and the prediction and filter densities, as given by (6.32) and (6.33) in §6.2, still hold. Because our focus is on simulation using MCMC methods, we first describe the technique in a general context.

Example 6.20 MCMC Techniques and the Gibbs Sampler

The goal of a Monte Carlo technique is to simulate a pseudo-random sample of vectors from a desired density function $p_\Theta(\boldsymbol{z})$. In Markov chain Monte Carlo, we simulate an ordered sequence of pseudo-random vectors, $\boldsymbol{z}_0 \mapsto \boldsymbol{z}_1 \mapsto \boldsymbol{z}_2 \mapsto \cdots$ by specifying a starting value, $\boldsymbol{z}_0$ and then sampling successive values from a transition density $\pi(\boldsymbol{z}_t | \boldsymbol{z}_{t-1})$, for $t = 1, 2, \ldots$. In this way, conditional on $\boldsymbol{z}_{t-1}$, the t-th pseudo-random vector, $\boldsymbol{z}_t$, is simulated independent of its predecessors. This technique alone does not yield a pseudo-random sample because contiguous draws are dependent on each other (that is, we obtain a first-order dependent sequence of pseudo-random vectors). If done appropriately, the dependence between the pseudo-variates $\boldsymbol{z}_t$ and $\boldsymbol{z}_{t+m}$ decays exponentially in m, and we may regard the collection $\{\boldsymbol{z}_{t+\ell m}; \ell = 1, 2, \ldots\}$ for t and m suitably large, as a pseudo-random sample. Alternately, one may repeat the process in parallel, retaining the m-th value, on run $g = 1, 2, \ldots$, say, $\boldsymbol{z}_m^{(g)}$, for large m. Under general conditions, the Markov chain converges in the sense that, eventually, the sequence of pseudo-variates appear stationary and the individual $\boldsymbol{z}_t$ are marginally distributed according to the stationary "target" density $p_\Theta(\boldsymbol{z})$. Technical details may be found in Tierney (1994).

For Gibbs sampling, suppose we have a collection $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\}$ of random vectors with complete conditional densities denoted generically by

$$p_\Theta(\boldsymbol{z}_j \mid \boldsymbol{z}_i, i \neq j) \equiv p_\Theta(\boldsymbol{z}_j \mid \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{j-1}, \boldsymbol{z}_{j+1}, \ldots, \boldsymbol{z}_k),$$

for $j = 1, \ldots, k$, available for sampling. Here, available means pseudo-samples may be generated by some method given the values of the appropriate conditioning random vectors. Under mild conditions, these complete conditionals uniquely determine the full joint density $p_\Theta(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k)$ and, consequently, all marginals, $p_\Theta(\boldsymbol{z}_j)$ for $j = 1, \ldots, k$; details may be found in Besag (1974). The Gibbs sampler generates pseudo-samples from the joint distribution as follows.

(i) Start with an arbitrary set of starting values, say, $\{\boldsymbol{z}_{1[0]}, \ldots, \boldsymbol{z}_{k[0]}\}$.
(ii) Draw $\boldsymbol{z}_{1[1]}$ from $p_\Theta(\boldsymbol{z}_1 | \boldsymbol{z}_{2[0]}, \ldots, \boldsymbol{z}_{k[0]})$;
(iii) Draw $\boldsymbol{z}_{2[1]}$ from $p_\Theta(\boldsymbol{z}_2 | \boldsymbol{z}_{1[1]}, \boldsymbol{z}_{3[0]}, \ldots, \boldsymbol{z}_{k[0]})$;
(iv) Repeat until step k, which draws $\boldsymbol{z}_{k[1]}$ from $p_\Theta(\boldsymbol{z}_k | \boldsymbol{z}_{1[1]}, \ldots, \boldsymbol{z}_{k-1[1]})$.
(v) Repeat steps (i)-(iv) ℓ times obtaining a collection $\{\boldsymbol{z}_{1[\ell]}, \ldots, \boldsymbol{z}_{k[\ell]}\}$.

Geman and Geman (1984) showed that under mild conditions, $\{\boldsymbol{z}_{1[\ell]}, \ldots, \boldsymbol{z}_{k[\ell]}\}$ converges in distribution to a random observation from $p_\Theta(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k)$ as $\ell \to \infty$. For this reason, we typically drop the subscript $[\ell]$ from the notation, assuming ℓ is sufficiently large for the generated sample to be thought of as a realization from the joint density; hence, we denote this first realization as $\{\boldsymbol{z}_{1[\ell]}^{(1)}, \ldots, \boldsymbol{z}_{k[\ell]}^{(1)}\} \equiv \{\boldsymbol{z}_1^{(1)}, \ldots, \boldsymbol{z}_k^{(1)}\}$. This entire process is replicated in parallel, a large number, G, of times providing pseudo-random iid collections $\{\boldsymbol{z}_1^{(g)}, \ldots, \boldsymbol{z}_k^{(g)}\}$, for $g = 1, \ldots, G$ from the joint distribution. These simulated values can the be used to estimate the marginal densities. In particular, if $p_\Theta(\boldsymbol{z}_j | \boldsymbol{z}_i, i \neq j)$ is available in closed form, then[2]

$$\widehat{p}_\Theta(\boldsymbol{z}_j) = G^{-1} \sum_{g=1}^{G} p_\Theta(\boldsymbol{z}_j \mid \boldsymbol{z}_i^{(g)}, i \neq j). \tag{6.190}$$

Because of the relatively recent appearance of Gibbs sampling methodology, several important theoretical and practical issues are under investigation. These issues include the diagnosis of convergence, modification of the sampling order, efficient estimation, and sequential sampling schemes (as opposed to the parallel processing described above) to mention a few. At this time, the best advice can be obtained from the texts by Gelman et al. (1995) and Gilks et al. (1996), and we are certain that many more will follow.

Finally, it may be necessary to nest rejection sampling within the Gibbs sampling procedure. The need for rejection sampling arises when we want to sample from a density, say, $f(\boldsymbol{z})$, but $f(\boldsymbol{z})$ is known only up to a proportionality constant, say, $p(\boldsymbol{z}) \propto f(\boldsymbol{z})$. If a density $g(\boldsymbol{z})$ is available, and there is a constant c for which $p(\boldsymbol{z}) \leq cg(\boldsymbol{z})$ for all $\boldsymbol{z}$, the rejection algorithm generates pseudo-variates from $f(\boldsymbol{z})$ by generating a value, $\boldsymbol{z}^*$ from $g(\boldsymbol{z})$ and

[2] Approximation (6.190) is based on the fact that, for random vectors x and y with joint density $p(x, y)$, the marginal density of x is obtained by integrating y out of the joint density, i.e., $p(x) = \int p(x, y)dy = \int p(x|y)p(y)dy$.

accepting it as a value from $f(\boldsymbol{z})$ with probability $\pi(\boldsymbol{z}^*) = p(\boldsymbol{z}^*)/[cg(\boldsymbol{z}^*)]$. This algorithm can be quite inefficient if $\pi(\cdot)$ is close to zero; in such cases, more sophisticated envelope functions may be needed. Further discussion of these matters in the case of nonlinear state-space models can be found in Carlin et al. (1992, Examples 1.2 and 3.2).

In Example 6.20, the generic random vectors $\boldsymbol{z}_j$ can represent parameter values, such as components of Θ, state values $\boldsymbol{x}_t$, or future observations $\boldsymbol{y}_{n+m}$, for $m \geq 1$. This will become evident in the following examples. Before discussing the general case of nonlinear and non-normal state-space models, we briefly introduce MCMC methods for the Gaussian DLM, as presented in Frühwirth-Schnatter (1994) and Carter and Kohn (1994).

Example 6.21 Parameter Assessment for the Gaussian DLM

Consider a Gaussian DLM given by

$$\boldsymbol{x}_t = \Phi_t \boldsymbol{x}_{t-1} + \boldsymbol{w}_t \quad \text{and} \quad y_t = \boldsymbol{a}_t' \boldsymbol{x}_t + v_t. \tag{6.191}$$

The observations are univariate, and the state process is p-dimensional; this DLM includes the structural models presented in §6.5. The prior on the initial state is $\boldsymbol{x}_0 \sim \mathrm{N}(\boldsymbol{\mu}_0, \Sigma_0)$, and we assume that $\boldsymbol{w}_t \sim$ iid $\mathrm{N}(\boldsymbol{0}, Q_t)$, independent of $v_t \sim$ iid $\mathrm{N}(0, r_t)$. The collection of unknown model parameters will be denoted by Θ.

To explore how we would assess the values of Θ using an MCMC technique, we focus on the problem obtaining the posterior distribution, $p(\Theta \mid Y_n)$, of the parameters given the data, $Y_n = \{y_1, \ldots, y_n\}$ and a prior $\pi(\Theta)$. Of course, these distributions depend on "hyperparameters" that are assumed to be known. (Some authors consider the states $\boldsymbol{x}_t$ as the first level of parameters because they are unobserved. In this case, the values in Θ are regarded as the hyperparameters, and the parameters of their distributions are regarded as hyper-hyperparameters.) Denoting the entire set of state vectors as $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the posterior can be written as

$$p(\Theta \mid Y_n) = \int p(\Theta \mid X_n, Y_n)\, p(X_n, \Theta^* \mid Y_n)\, dX_n\, d\Theta^*. \tag{6.192}$$

Although the posterior, $p(\Theta \mid Y_n)$, may be intractable, conditioning on the states can make the problem manageable in that

$$p(\Theta \mid X_n, Y_n) \propto \pi(\Theta)\, p(x_0 \mid \Theta) \prod_{t=1}^{n} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \Theta)\, p(y_t \mid \boldsymbol{x}_t, \Theta) \tag{6.193}$$

can be easier to work with (either as members of conjugate families or using some rejection scheme); we will discuss this in more detail when we present the nonlinear, non-Gaussian case, but we will assume for the present $p(\Theta \mid X_n, Y_n)$ is in closed form.

Suppose we can obtain G pseudo-random draws, $X_n^{(g)} \equiv (X_n, \Theta^*)^{(g)}$, for $g = 1, \ldots, G$, from the joint posterior density $p(X_n, \Theta^* \mid Y_n)$. Then (6.192) can be approximated by

$$\widehat{p}(\Theta \mid Y_n) = G^{-1} \sum_{g=1}^{G} p(\Theta \mid X_n^{(g)}, Y_n).$$

A sample from $p(X_n, \Theta^* \mid Y_n)$ is obtained using two different MCMC methods. First, the Gibbs sampler is used, for each g, as follows: sample $X_{n[\ell]}$ given $\Theta^*_{[\ell-1]}$ from $p(X_n \mid \Theta^*_{[\ell-1]}, Y_n)$, and then a sample $\Theta^*_{[\ell]}$ from $p(\Theta \mid X_{n[\ell]}, Y_n)$ as given by (6.193), for $\ell = 1, 2, \ldots$. Stop when ℓ is sufficiently large, and retain the final values as $X_n^{(g)}$. This process is repeated G times.

The first step of this method requires simultaneous generation of the state vectors. Because we are dealing with a Gaussian linear model, we can rely on the existing theory of the Kalman filter to accomplish this step. This step is conditional on Θ, and we assume at this point that Θ is fixed and known. In other words, our goal is to sample the entire set of state vectors, $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, from the multivariate normal posterior density $p_\Theta(X_n \mid Y_n)$, where $Y_n = \{y_1, \ldots, y_n\}$ represents the observations. Because of the Markov structure, we can write,

$$p_\Theta(X_n \mid Y_n) = p_\Theta(\boldsymbol{x}_n \mid Y_n) p_\Theta(\boldsymbol{x}_{n-1} \mid \boldsymbol{x}_n, Y_{n-1}) \cdots p_\Theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1). \quad (6.194)$$

In view of (6.194), it is possible to sample the entire set of state vectors, X_n, by sequentially simulating the individual states backward. This process yields a simulation method that Frühwirth-Schnatter (1994) called the forward-filtering, backward-sampling algorithm. In particular, because the processes are Gaussian, we need only obtain the conditional means and variances, say, $\boldsymbol{m}_t = E_\Theta(\boldsymbol{x}_t \mid Y_t, \boldsymbol{x}_{t+1})$, and $V_t = \mathrm{var}_\Theta(\boldsymbol{x}_t \mid Y_t, \boldsymbol{x}_{t+1})$. This conditioning argument is akin to having $\boldsymbol{x}_{t+1}$ as an additional observation on state $\boldsymbol{x}_t$. In particular, using standard multivariate normal distribution theory,

$$\begin{aligned} \boldsymbol{m}_t &= \boldsymbol{x}_t^t + J_t(\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1}^t), \\ V_t &= P_t^t - J_t P_{t+1}^t J_t', \end{aligned} \quad (6.195)$$

for $t = n-1, n-2, \ldots, 0$, where J_t is defined in (6.49). To verify (6.195), the essential part of the Gaussian density (that is, the exponent) of $\boldsymbol{x}_t \mid Y_t, \boldsymbol{x}_{t+1}$ is

$$(\boldsymbol{x}_{t+1} - \Phi_{t+1}\boldsymbol{x}_t)'[Q_{t+1}]^{-1}(\boldsymbol{x}_{t+1} - \Phi_{t+1}\boldsymbol{x}_t) + (\boldsymbol{x}_t - \boldsymbol{x}_t^t)'[P_t^t]^{-1}(\boldsymbol{x}_t - \boldsymbol{x}_t^t),$$

and we simply complete the square; see Frühwirth-Schnatter (1994) or West and Harrison (1997, §4.7). Hence, the algorithm is to first sample $\boldsymbol{x}_n$ from a $\mathrm{N}(\boldsymbol{x}_n^n, P_n^n)$, where $\boldsymbol{x}_n^n$ and P_n^n are obtained from the Kalman filter, Property 6.1, and then sample $\boldsymbol{x}_t$ from a $\mathrm{N}(\boldsymbol{m}_t, V_t)$, for $t = n-1, n-2, \ldots, 0$,

where the conditioning value of $\boldsymbol{x}_{t+1}$ is the value previously sampled; $\boldsymbol{m}_t$ and V_t are given in (6.195).

Next, we address an MCMC approach to nonlinear and non-Gaussian state-space modeling that was first presented in Carlin et al. (1992). We consider the general model given in (6.188), but with additive errors:

$$\boldsymbol{x}_t = F_t(\boldsymbol{x}_{t-1}) + \boldsymbol{w}_t \quad \text{and} \quad \boldsymbol{y}_t = H_t(\boldsymbol{x}_t) + \boldsymbol{v}_t, \tag{6.196}$$

where F_t and H_t are given, but may also depend on unknown parameters, say, Φ_t and A_t, respectively, the collection of which will be denoted by Θ. The errors are independent white noise sequences with $\text{var}(\boldsymbol{w}_t) = Q_t$ and $\text{var}(\boldsymbol{v}_t) = R_t$. Although time-varying variance–covariance matrices are easily incorporated in this framework, to ease the discussion we focus on the case $Q_t \equiv Q$ and $R_t \equiv R$. Also, although it is not necessary, we assume the initial state condition $\boldsymbol{x}_0$ is fixed and known; this is merely for notational convenience, so we do not have to carry along the additional terms involving $\boldsymbol{x}_0$ throughout the discussion.

In general, the likelihood specification for the model is given by

$$L_{X,Y}(\Theta, Q, R) = \prod_{t=1}^{n} f_1(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \Theta, Q)\ f_2(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \Theta, R), \tag{6.197}$$

where it is assumed the densities $f_1(\cdot)$ and $f_2(\cdot)$ are scale mixtures of normals. Specifically, for $t = 1, \ldots, n$,

$$\begin{aligned} f_1(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \Theta, Q) &= \int f(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \Theta, Q, \lambda_t)\ p_1(\lambda_t)\ d\lambda_t, \\ f_2(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \Theta, R) &= \int f(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \Theta, R, \omega_t)\ p_2(\omega_t)\ d\omega_t, \end{aligned} \tag{6.198}$$

where conditional on the independent sequences of nuisance parameters $\boldsymbol{\lambda} = (\lambda_t; t = 1, \ldots, n)$ and $\boldsymbol{\omega} = (\omega_t; t = 1, \ldots, n)$,

$$\begin{aligned} \boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \Theta, Q, \lambda_t &\sim \text{N}\Big(F_t(\boldsymbol{x}_{t-1}; \Theta),\ \lambda_t Q\Big), \\ \boldsymbol{y}_t \mid \boldsymbol{x}_t, \Theta, R, \omega_t &\sim \text{N}\Big(H_t(\boldsymbol{x}_t; \Theta),\ \omega_t R\Big). \end{aligned} \tag{6.199}$$

By varying $p_1(\lambda_t)$ and $p_2(\omega_t)$, we can have a wide variety of non-Gaussian error densities. These densities include, for example, double exponential, logistic, and t distributions in the univariate case and a rich class of multivariate distributions; this is discussed further in Carlin et al. (1992). The key to the approach is the introduction of the nuisance parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ and the structure (6.199), which lends itself naturally to the Gibbs sampler and allows for the analysis of this general nonlinear and non-Gaussian problem.

According to Example 6.20, to implement the Gibbs sampler, we must be able to sample from the following complete conditional distributions:

(i) $\boldsymbol{x}_t \mid \boldsymbol{x}_{s\neq t}, \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, R, Y_n \quad t = 1, \ldots, n,$
(ii) $\lambda_t \mid \lambda_{s\neq t}, \boldsymbol{\omega}, \Theta, Q, R, Y_n, X_n \sim \lambda_t \mid \Theta, Q, \boldsymbol{x}_t, \boldsymbol{x}_{t-1} \quad t = 1, \ldots, n,$
(iii) $\omega_t \mid \omega_{s\neq t}, \boldsymbol{\lambda}, \Theta, Q, R, Y_n, X_n \sim \omega_t \mid \Theta, R, \boldsymbol{y}_t, \boldsymbol{x}_t \quad t = 1, \ldots, n,$
(iv) $Q \mid \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, R, Y_n, X_n \sim Q \mid \boldsymbol{\lambda}, Y_n, X_n,$
(v) $R \mid \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, Y_n, X_n \sim R \mid \boldsymbol{\omega}, Y_n, X_n,$
(vi) $\Theta \mid \boldsymbol{\lambda}, \boldsymbol{\omega}, Q, R, Y_n, X_n \sim \Theta \mid Y_n, X_n,$

where $X_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$. The main difference between this method and the linear Gaussian case is that, because of the generality, we sample the states one-at-a-time rather than simultaneously generating all of them. As discussed in Carter and Kohn (1994), if possible, it is more efficient to generate the states simultaneously as in Example 6.21.

We will discuss items (i) and (ii) above. The third item follows in a similar manner to the second, and items (iv)-(vi) will follow from standard multivariate normal distribution theory and from Wishart distribution theory because of the conditioning on $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$. We will discuss this matter further in the next example. First, consider the linear model, $F_t(\boldsymbol{x}_{t-1}) = \Phi_t \boldsymbol{x}_{t-1}$, and $H_t(\boldsymbol{x}_t) = A_t \boldsymbol{x}_t$ in (6.196). In this case, for $t = 1, \ldots, n$, $\boldsymbol{x}_t \mid \boldsymbol{x}_{s\neq t}, \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, R, Y_n$ has a p-dimensional $\mathrm{N}_p(B_t \boldsymbol{b}_t, B_t)$ distribution, with

$$\begin{aligned} B_t^{-1} &= \frac{Q^{-1}}{\lambda_t} + \frac{A_t' R^{-1} A_t}{\omega_t} + \frac{\Phi_{t+1}' Q^{-1} \Phi_{t+1}}{\lambda_{t+1}}, \\ \boldsymbol{b}_t &= \frac{\boldsymbol{x}_{t-1} \Phi_t' Q^{-1}}{\lambda_t} + \frac{\boldsymbol{y}_t R^{-1} A_t}{\omega_t} + \frac{\boldsymbol{x}_{t+1} Q^{-1} \Phi_{t+1}}{\lambda_{t+1}}, \end{aligned} \tag{6.200}$$

where, when $t = n$ in (6.200), terms in the sum with elements having a subscript of $n + 1$ are dropped (this is assumed to be the case in what follows, although we do not explicitly state it). This result follows by noting the essential part of the multivariate normal distribution (that is, the exponent) of $\boldsymbol{x}_t \mid \boldsymbol{x}_{s\neq t}, \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, R, Y_n$ is

$$\begin{aligned} &(\boldsymbol{x}_t - \Phi_t \boldsymbol{x}_{t-1})'(\lambda_t Q)^{-1}(\boldsymbol{x}_t - \Phi_t \boldsymbol{x}_{t-1}) + (\boldsymbol{y}_t - A_t \boldsymbol{x}_t)'(\omega_t R)^{-1}(\boldsymbol{y}_t - A_t \boldsymbol{x}_t) \\ &\quad + (\boldsymbol{x}_{t+1} - \Phi_{t+1} \boldsymbol{x}_t)'(\lambda_{t+1} Q)^{-1}(\boldsymbol{x}_{t+1} - \Phi_{t+1} \boldsymbol{x}_t), \end{aligned} \tag{6.201}$$

which upon manipulation yields (6.200).

Example 6.22 Nonlinear Models

In the case of nonlinear models, we can use (6.200) with slight modifications. For example, consider the case in which F_t is nonlinear, but H_t is linear, so the observations are $\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \boldsymbol{v}_t$. Then,

$$\boldsymbol{x}_t \mid \boldsymbol{x}_{s\neq t}, \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, R, Y_n \propto \eta_1(\boldsymbol{x}_t) \mathrm{N}_p(B_{1t} \boldsymbol{b}_{1t}, B_{1t}), \tag{6.202}$$

where

$$B_{1t}^{-1} = \frac{Q^{-1}}{\lambda_t} + \frac{A_t' R^{-1} A_t}{\omega_t},$$
$$\boldsymbol{b}_{1t} = \frac{F_t'(\boldsymbol{x}_{t-1}) Q^{-1}}{\lambda_t} + \frac{\boldsymbol{y}_t R^{-1} A_t}{\omega_t},$$

and

$$\eta_1(\boldsymbol{x}_t) = \exp\left\{ -\frac{1}{2\lambda_{t+1}} \Big(\boldsymbol{x}_{t+1} - F_{t+1}(\boldsymbol{x}_t)\Big)' Q^{-1} \Big(\boldsymbol{x}_{t+1} - F_{t+1}(\boldsymbol{x}_t)\Big) \right\}.$$

Because $0 \le \eta_1(\boldsymbol{x}_t) \le 1$, for all $\boldsymbol{x}_t$, the distribution we want to sample from is dominated by the $\mathrm{N}_p(B_{1t}\boldsymbol{b}_{1t}, B_{1t})$ density. Hence, we may use rejection sampling as discussed in Example 6.20 to obtain an observation from the required density. That is, we generate a pseudo-variate from the $\mathrm{N}_p(B_{1t}\boldsymbol{b}_{1t}, B_{1t})$ density and accept it with probability $\eta_1(\boldsymbol{x}_t)$.

We proceed analogously in the case in which $F_t(\boldsymbol{x}_{t-1}) = \Phi_t \boldsymbol{x}_{t-1}$ is linear and $H_t(\boldsymbol{x}_t)$ is nonlinear. In this case,

$$\boldsymbol{x}_t \mid \boldsymbol{x}_{s\neq t}, \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, R, Y_n \propto \eta_2(\boldsymbol{x}_t) \mathrm{N}_p(B_{2t}\boldsymbol{b}_{2t}, B_{2t}), \tag{6.203}$$

where

$$B_{2t}^{-1} = \frac{Q^{-1}}{\lambda_t} + \frac{\Phi_{t+1}' Q^{-1} \Phi_{t+1}}{\lambda_{t+1}},$$
$$\boldsymbol{b}_{2t} = \frac{\boldsymbol{x}_{t-1} \Phi_t' Q^{-1}}{\lambda_t} + \frac{\boldsymbol{x}_{t+1} Q^{-1} \Phi_{t+1}}{\lambda_{t+1}},$$

and

$$\eta_2(\boldsymbol{x}_t) = \exp\left\{ -\frac{1}{2\omega_t} \Big(\boldsymbol{y}_t - H_t(\boldsymbol{x}_t)\Big)' R^{-1} \Big(\boldsymbol{y}_t - H_t(\boldsymbol{x}_t)\Big) \right\}.$$

Here, we generate a pseudo-variate from the $\mathrm{N}_p(B_{2t}\boldsymbol{b}_{2t}, B_{2t})$ density and accept it with probability $\eta_2(\boldsymbol{x}_t)$.

Finally, in the case in which both F_t and H_t are nonlinear, we have

$$\boldsymbol{x}_t \mid \boldsymbol{x}_{s\neq t}, \boldsymbol{\lambda}, \boldsymbol{\omega}, \Theta, Q, R, Y_n \propto \eta_1(\boldsymbol{x}_t)\eta_2(\boldsymbol{x}_t) \mathrm{N}_p(F_t(\boldsymbol{x}_{t-1}), \lambda_t Q), \tag{6.204}$$

so we sample from a $\mathrm{N}_p(F_t(\boldsymbol{x}_{t-1}), \lambda_t Q)$ density and accept it with probability $\eta_1(\boldsymbol{x}_t)\eta_2(\boldsymbol{x}_t)$.

Determination of (ii), $\lambda_t \mid \Theta, Q, \boldsymbol{x}_t, \boldsymbol{x}_{t-1}$ follows directly from Bayes theorem; that is, $p(\lambda_t \mid \Theta, Q, \boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \propto p_1(\lambda_t) p(\boldsymbol{x}_t \mid \lambda_t, \boldsymbol{x}_{t-1}, \Theta, Q)$. By (6.198), however, we know the normalization constant is given by $f_1(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \Theta, Q)$, and thus the complete conditional density for λ_t is of a known functional form.

Many examples of these techniques are given in Carlin et al. (1992), including the problem of model choice. In the next example, we consider a univariate nonlinear model in which the state noise process has a t-distribution.

As noted in Meinhold and Singpurwalla (1989), using t-distributions for the error processes is a way of robustifying the Kalman filter against outliers. In this example we present a brief discussion of a detailed analysis presented in Carlin et al. (1992, Example 4.2); readers interested in more detail may find it in that article.

Example 6.23 A Nonlinear, Non-Gaussian State-Space Model

Kitagawa (1987) considered the analysis of data generated from the following univariate nonlinear model:

$$x_t = F_t(x_{t-1}) + w_t \quad \text{and} \quad y_t = H_t(x_t) + v_t \quad t = 1, \ldots, 100, \tag{6.205}$$

with

$$\begin{aligned} F_t(x_{t-1}) &= \alpha x_{t-1} + \beta x_{t-1}/(1 + x_{t-1}^2) + \gamma \cos[1.2(t-1)], \\ H_t(x_t) &= x_t^2/20, \end{aligned} \tag{6.206}$$

where $x_0 = 0$, w_t are independent random variables having a central t-distribution with $\nu = 10$ degrees and scaled so $\text{var}(w_t) = \sigma_w^2 = 10$ [we denote this generically by $t(0, \sigma, \nu)$], and v_t is white standard Gaussian noise, $\text{var}(v_t) = \sigma_v^2 = 1$. The state noise and observation noise are mutually independent. Kitagawa (1987) discussed the analysis of data generated from this model with $\alpha = .5$, $\beta = 25$, and $\gamma = 8$ assumed known. We will use these values of the parameters in this example, but we will assume they are unknown. Figure 6.15 shows a typical data sequence y_t and the corresponding state process x_t.

Our goal here will be to obtain an estimate of the prediction density $p(x_{101} \mid Y_{100})$. To accomplish this, we use $n = 101$ and consider y_{101} as a latent variable (we will discuss this in more detail shortly). The priors on the variance components are chosen from a conjugate family, that is, $\sigma_w^2 \sim \text{IG}(a_0, b_0)$ independent of $\sigma_v^2 \sim \text{IG}(c_0, d_0)$, where IG denotes the inverse (reciprocal) gamma distribution [z has an inverse gamma distribution if $1/z$ has a gamma distribution; general properties can be found, for example, in Box and Tiao (1973, Section 8.5)]. Then,

$$\begin{aligned} &\sigma_w^2 \mid \boldsymbol{\lambda}, Y_n, X_n \sim \\ &\quad \text{IG}\left(a_0 + \frac{n}{2}, \left\{\frac{1}{b_0} + \frac{1}{2}\sum_{t=1}^{n}[x_t - F(x_{t-1})]^2/\lambda_t\right\}^{-1}\right), \\ &\sigma_v^2 \mid \boldsymbol{\omega}, Y_n, X_n \sim \\ &\quad \text{IG}\left(c_0 + \frac{n}{2}, \left\{\frac{1}{d_0} + \frac{1}{2}\sum_{t=1}^{n}[y_t - H(x_t)]^2/\omega_t\right\}^{-1}\right). \end{aligned} \tag{6.207}$$

Next, letting $\nu/\lambda_t \sim \chi_\nu^2$, we get that, marginally, $w_t \mid \sigma_w \sim t(0, \sigma_w, \nu)$, as required, leading to the complete conditional $\lambda_t \mid \sigma_w, \alpha, \beta, \gamma, Y_n, X_n$, for $t = 1, \ldots, n$, being distributed as

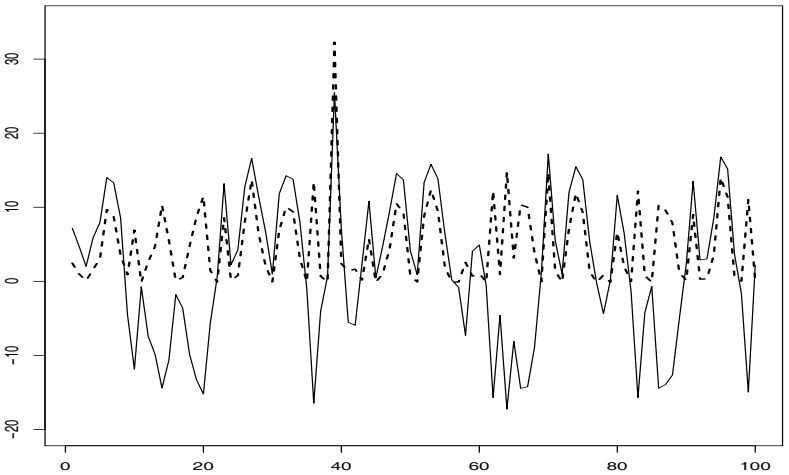

Fig. 6.15. The state process, x_t (solid line), and the observations, y_t (dashed line), for $t = 1, \ldots, 100$ generated from the model (6.205).

$$\mathrm{IG}\left(\frac{\nu+1}{2},\ 2\left\{\frac{[x_t - F(x_{t-1})]^2}{\sigma_w^2} + \nu\right\}^{-1}\right). \tag{6.208}$$

We take $\omega_t \equiv 1$ for $t = 1, \ldots, n$, because the observation noise is Gaussian.

For the states, x_t, we take a normal prior on the initial state, $x_0 \sim \mathrm{N}(\mu_0, \sigma_0^2)$, and then we use rejection sampling to conditionally generate a state value x_t, for $t = 1, \ldots, n$, as described in Example 6.22, equation (6.204). In this case, $\eta_1(x_t)$ and $\eta_2(x_t)$ are given in (6.202) and (6.203), respectively, with F_t and H_t given by (6.206), $\Theta = (\alpha, \beta, \gamma)'$, $Q = \sigma_w^2$ and $R = \sigma_v^2$. Endpoints take some special consideration; we generate x_0 from a $\mathrm{N}(\mu_0, \sigma_0^2)$ and accept it with probability $\eta_1(x_0)$, and we generate x_{101} as usual and accept it with probability $\eta_2(x_{101})$. The last complete conditional depends on y_{101}, a latent data value not observed but instead generated according to its complete conditional, which is $\mathrm{N}(x_{101}^2/20, \sigma_v^2)$, because $\omega_{101} = 1$.

The prior on $\Theta = (\alpha, \beta, \gamma)'$ is taken to be trivariate normal with mean $(\mu_\alpha, \mu_\beta, \mu_\gamma)'$ and diagonal variance–covariance matrix $\mathrm{diag}\{\sigma_\alpha^2, \sigma_\beta^2, \sigma_\gamma^2\}$. The necessary conditionals can be found using standard normal theory, as done in (6.200). For example, the complete conditional distribution of α is of the form $\mathrm{N}(Bb, B)$, where

$$B^{-1} = \frac{1}{\sigma_\alpha^2} + \frac{1}{\sigma_w^2}\sum_{t=1}^{n}\frac{x_{t-1}^2}{\lambda_t},$$

$$b = \frac{\mu_\alpha}{\sigma_\alpha^2} + \frac{1}{\sigma_w^2}\sum_{t=1}^{n}\frac{x_{t-1}}{\lambda_t}\left(x_t - \beta\frac{x_{t-1}}{1+x_{t-1}^2} - \gamma\cos[1.2(t-1)]\right).$$

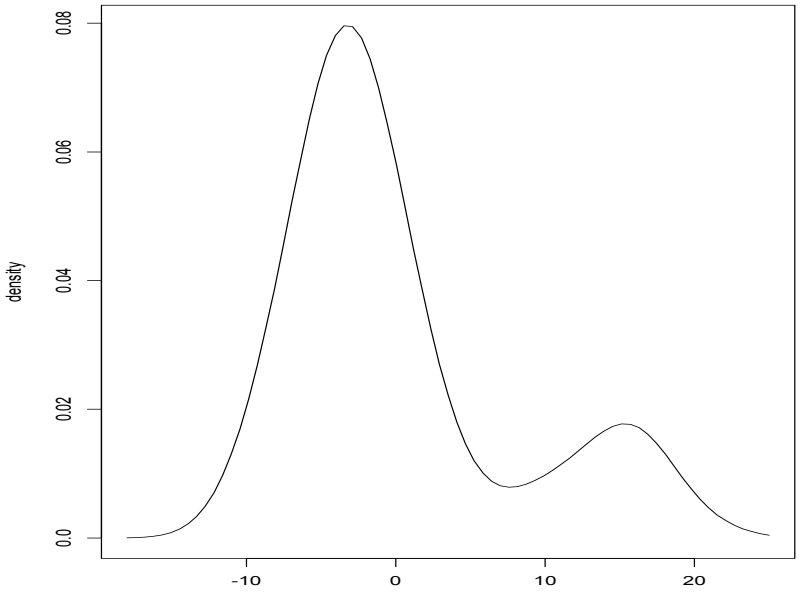

Fig. 6.16. Estimated one-step-ahead prediction posterior density $\widehat{p}(x_{101}|Y_{100})$ of the state process for the nonlinear and non-normal model given by (6.205) using Gibbs sampling, $G = 500$.

The complete conditional for β has the same form, with

$$B^{-1} = \frac{1}{\sigma_\beta^2} + \frac{1}{\sigma_w^2} \sum_{t=1}^{n} \frac{x_{t-1}^2}{\lambda_t (1 + x_{t-1}^2)^2},$$

$$b = \frac{\mu_\beta}{\sigma_\beta^2} + \frac{1}{\sigma_w^2} \sum_{t=1}^{n} \frac{x_{t-1}}{\lambda_t (1 + x_{t-1}^2)} \left(x_t - \alpha x_{t-1} - \gamma \cos[1.2(t-1)]\right),$$

and for γ the values are

$$B^{-1} = \frac{1}{\sigma_\gamma^2} + \frac{1}{\sigma_w^2} \sum_{t=1}^{n} \frac{\cos^2[1.2(t-1)]}{\lambda_t},$$

$$b = \frac{\mu_\gamma}{\sigma_\gamma^2} + \frac{1}{\sigma_w^2} \sum_{t=1}^{n} \frac{\cos[1.2(t-1)]}{\lambda_t} \left(x_t - \alpha x_{t-1} - \beta \frac{x_{t-1}}{1 + x_{t-1}^2}\right).$$

In this example, we put $\mu_0 = 0$, $\sigma_0^2 = 10$, and $a_0 = 3$, $b_0 = .05$ (so the prior on σ_w^2 has mean and standard deviation equal to 10), and $c_0 = 3$, $d_0 = .5$ (so the prior on σ_v^2 has mean and standard deviation equal to one). The normal prior on $\Theta = (\alpha, \beta, \gamma)'$ had corresponding mean vector equal to $(\mu_\alpha = .5, \mu_\beta = 25, \mu_\gamma = 8)'$ and diagonal variance matrix equal

to diag$\{\sigma_\alpha^2 = .25, \sigma_\beta^2 = 10, \sigma_\gamma^2 = 4\}$. The Gibbs sampler ran for $\ell = 50$ iterations for $G = 500$ parallel replications per iteration. We estimate the marginal posterior density of x_{101} as

$$\widehat{p}(x_{101} \mid Y_{100}) = G^{-1} \sum_{g=1}^{G} \mathrm{N}\left(x_{101} \;\middle|\; [F_t(x_{t-1})]^{(g)}, \lambda_{101}^{(g)} \sigma_w^{2(g)}\right), \qquad (6.209)$$

where $\mathrm{N}(\cdot|a, b)$ denotes the normal density with mean a and variance b, and

$$[F_t(x_{t-1})]^{(g)} = \alpha^{(g)} x_{t-1}^{(g)} + \beta^{(g)} x_{t-1}^{(g)} / (1 + x_{t-1}^{2(g)}) + \gamma^{(g)} \cos[1.2(t-1)].$$

The estimate, (6.209), with $G = 500$, is shown in Figure 6.16. Other aspects of the analysis, for example, the marginal posteriors of the elements of Θ, can be found in Carlin et al. (1992).

Problems

Section 6.1

6.1 Consider a system process given by

$$x_t = -.9x_{t-2} + w_t \quad t = 1, \ldots, n$$

where $x_0 \sim \mathrm{N}(0, \sigma_0^2)$, $x_{-1} \sim N(0, \sigma_1^2)$, and w_t is Gaussian white noise with variance σ_w^2. The system process is observed with noise, say,

$$y_t = x_t + v_t,$$

where v_t is Gaussian white noise with variance σ_v^2. Further, suppose x_0, x_{-1}, $\{w_t\}$ and $\{v_t\}$ are independent.

(a) Write the system and observation equations in the form of a state space model.
(b) Find the values of σ_0^2 and σ_1^2 that make the observations, y_t, stationary.
(c) Generate $n = 100$ observations with $\sigma_w = 1$, $\sigma_v = 1$ and using the values of σ_0^2 and σ_1^2 found in (b). Do a time plot of x_t and of y_t and compare the two processes. Also, compare the sample ACF and PACF of x_t and of y_t.
(d) Repeat (c), but with $\sigma_v = 10$.

6.2 Consider the state-space model presented in Example 6.3. Let $x_t^{t-1} = E(x_t|y_{t-1}, \ldots, y_1)$ and let $P_t^{t-1} = E(x_t - x_t^{t-1})^2$. The innovation sequence or residuals are $\epsilon_t = y_t - y_t^{t-1}$, where $y_t^{t-1} = E(y_t|y_{t-1}, \ldots, y_1)$. Find $\mathrm{cov}(\epsilon_s, \epsilon_t)$ in terms of x_t^{t-1} and P_t^{t-1} for (i) $s \neq t$ and (ii) $s = t$.

Section 6.2

6.3 Simulate $n = 100$ observations from the following state-space model:

$$x_t = .8x_{t-1} + w_t \quad \text{and} \quad y_t = x_t + v_t$$

where $x_0 \sim \text{N}(0, 2.78)$, $w_t \sim$ iid $\text{N}(0, 1)$, and $v_t \sim$ iid $\text{N}(0, 1)$ are all mutually independent. Compute and plot the data, y_t, the one-step-ahead predictors, y_t^{t-1} along with the root mean square prediction errors, $E^{1/2}(y_t - y_t^{t-1})^2$ using Figure 6.3 as a guide.

6.4 Suppose the vector $\boldsymbol{z} = (\boldsymbol{x}', \boldsymbol{y}')'$, where $\boldsymbol{x}$ $(p \times 1)$ and $\boldsymbol{y}$ $(q \times 1)$ are jointly distributed with mean vectors $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$ and with covariance matrix

$$\text{cov}(\boldsymbol{z}) = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}.$$

Consider projecting $\boldsymbol{x}$ on $\mathcal{M} = \overline{\text{sp}}\{\mathbf{1}, \boldsymbol{y}\}$, say, $\widehat{\boldsymbol{x}} = \boldsymbol{b} + B\boldsymbol{y}$.

(a) Show the orthogonality conditions can be written as

$$E(\boldsymbol{x} - \boldsymbol{b} - B\boldsymbol{y}) = 0,$$

$$E[(\boldsymbol{x} - \boldsymbol{b} - B\boldsymbol{y})\boldsymbol{y}'] = 0,$$

leading to the solutions

$$\boldsymbol{b} = \boldsymbol{\mu}_x - B\boldsymbol{\mu}_y \quad \text{and} \quad B = \Sigma_{xy}\Sigma_{yy}^{-1}.$$

(b) Prove the mean square error matrix is

$$MSE = E[(\boldsymbol{x} - \boldsymbol{b} - B\boldsymbol{y})\boldsymbol{x}'] = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}.$$

(c) How can these results be used to justify the claim that, in the absence of normality, Property 6.1 yields the best linear estimate of the state $\boldsymbol{x}_t$ given the data Y_t, namely, $\boldsymbol{x}_t^t$, and its corresponding MSE, namely, P_t^t?

6.5 *Projection Theorem Derivation of Property 6.2.* Throughout this problem, we use the notation of Property 6.2 and of the Projection Theorem given in Appendix B, where $\mathcal{H}$ is L^2. If $\mathcal{L}_{k+1} = \overline{\text{sp}}\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{k+1}\}$, and $\mathcal{V}_{k+1} = \overline{\text{sp}}\{\boldsymbol{y}_{k+1} - \boldsymbol{y}_{k+1}^k\}$, for $k = 0, 1, \ldots, n-1$, where $\boldsymbol{y}_{k+1}^k$ is the projection of $\boldsymbol{y}_{k+1}$ on $\mathcal{L}_k$, then, $\mathcal{L}_{k+1} = \mathcal{L}_k \oplus \mathcal{V}_{k+1}$. We assume $P_0^0 > 0$ and $R > 0$.

(a) Show the projection of $\boldsymbol{x}_k$ on $\mathcal{L}_{k+1}$, that is, $\boldsymbol{x}_k^{k+1}$, is given by

$$\boldsymbol{x}_k^{k+1} = \boldsymbol{x}_k^k + H_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{k+1}^k),$$

where H_{k+1} can be determined by the orthogonality property

$$E\left\{\left(\boldsymbol{x}_k - H_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{k+1}^k)\right)\left(\boldsymbol{y}_{k+1} - \boldsymbol{y}_{k+1}^k\right)'\right\} = 0.$$

Show

$$H_{k+1} = P_k^k \Phi' A_{k+1}' \left[A_{k+1} P_{k+1}^k A_{k+1}' + R\right]^{-1}.$$

(b) Define $J_k = P_k^k \Phi' [P_{k+1}^k]^{-1}$, and show

$$\boldsymbol{x}_k^{k+1} = \boldsymbol{x}_k^k + J_k(\boldsymbol{x}_{k+1}^{k+1} - \boldsymbol{x}_{k+1}^k).$$

(c) Repeating the process, show

$$\boldsymbol{x}_k^{k+2} = \boldsymbol{x}_k^k + J_k(\boldsymbol{x}_{k+1}^{k+1} - \boldsymbol{x}_{k+1}^k) + H_{k+2}(\boldsymbol{y}_{k+2} - \boldsymbol{y}_{k+2}^{k+1}),$$

solving for H_{k+2}. Simplify and show

$$\boldsymbol{x}_k^{k+2} = \boldsymbol{x}_k^k + J_k(\boldsymbol{x}_{k+1}^{k+2} - \boldsymbol{x}_{k+1}^k).$$

(d) Using induction, conclude

$$\boldsymbol{x}_k^n = \boldsymbol{x}_k^k + J_k(\boldsymbol{x}_{k+1}^n - \boldsymbol{x}_{k+1}^k),$$

which yields the smoother with $k = t - 1$.

Section 6.3

6.6 Consider the univariate state-space model given by state conditions $x_0 = w_0$, $x_t = x_{t-1} + w_t$ and observations $y_t = x_t + v_t$, $t = 1, 2, \ldots$, where w_t and v_t are independent, Gaussian, white noise processes with $\mathrm{var}(w_t) = \sigma_w^2$ and $\mathrm{var}(v_t) = \sigma_v^2$.

(a) Show that y_t follows an IMA(1,1) model, that is, ∇y_t follows an MA(1) model.
(b) Fit the model specified in part (a) to the logarithm of the glacial varve series and compare the results to those presented in Example 3.32.

6.7 Let y_t represent the global temperature series (`gtemp`) shown in Figure 1.2.

(a) Fit a smoothing spline using GCV (the default) to y_t and plot the result superimposed on the data. Repeat the fit using `spar=.7`; the GCV method yields `spar=.5` approximately. (Example 2.14 on page 75 may help. Also in R, see the help file `?smooth.spline`.)
(b) Write the model $y_t = x_t + v_t$ with $\nabla^2 x_t = w_t$, in state-space form. [*Hint:* The state will be a 2×1 vector, say, $\boldsymbol{x}_t = (x_t, x_{t-1})'$.] Assume w_t and v_t are independent Gaussian white noise processes, both independent of $\boldsymbol{x}_0$. Fit this state-space model to y_t, and exhibit a time plot the estimated smoother, $\widehat{x}_t^n$ and the corresponding error limits, $\widehat{x}_t^n \pm 2\sqrt{\widehat{P}_t^n}$ superimposed on the data.
(c) Superimpose all the fits from parts (a) and (b) [include the error bounds] on the data and briefly compare and contrast the results.

6.8 *Smoothing Splines and the Kalman Smoother.* Consider the discrete time version of the smoothing spline argument given in (2.56); that is, suppose we observe $y_t = x_t + v_t$ and we wish to fit x_t, for $t = 1, \ldots, n$, constrained to be smooth, by minimizing

$$\sum_{t=1}^{n} [y_t - x_t]^2 + \lambda \sum_{t=1}^{n} \left(\nabla^2 x_t\right)^2 . \tag{6.210}$$

Show that this problem is identical to obtaining $\widehat{x}_t^n$ in Problem 6.7(b), with $\lambda = \sigma_v^2/\sigma_w^2$, assuming $\boldsymbol{x}_0 = \mathbf{0}$. *Hint:* Using the notation surrounding equation (6.63), the goal is to find the MLE of X_n given Y_n, i.e., maximize $\log f(X_n|Y_n)$. Because of the Gaussianity, the maximum (or mode) of the distribution is when the states are estimated by x_t^n, the conditional means. But $\log f(X_n|Y_n) = \log f(X_n, Y_n) - \log f(Y_n)$, so maximizing $\log f(X_n, Y_n)$ with respect to X_n is an equivalent problem. Now, ignore the initial state and write $-2 \log f(X_n, Y_n)$ based on the model, which should look like (6.210); use (6.64) as a guide.

6.9 Consider the model

$$y_t = x_t + v_t,$$

where v_t is Gaussian white noise with variance σ_v^2, x_t are independent Gaussian random variables with mean zero and $\mathrm{var}(x_t) = r_t \sigma_x^2$ with x_t independent of v_t, and $r_1, \ldots, r_n$ are known constants. Show that applying the EM algorithm to the problem of estimating σ_x^2 and σ_v^2 leads to updates (represented by hats)

$$\widehat{\sigma}_x^2 = \frac{1}{n} \sum_{t=1}^{n} \frac{\sigma_t^2 + \mu_t^2}{r_t} \quad \text{and} \quad \widehat{\sigma}_v^2 = \frac{1}{n} \sum_{t=1}^{n} [(y_t - \mu_t)^2 + \sigma_t^2],$$

where, based on the current estimates (represented by tildes),

$$\mu_t = \frac{r_t \widetilde{\sigma}_x^2}{r_t \widetilde{\sigma}_x^2 + \widetilde{\sigma}_v^2}\, y_t \quad \text{and} \quad \sigma_t^2 = \frac{r_t \widetilde{\sigma}_x^2 \widetilde{\sigma}_v^2}{r_t \widetilde{\sigma}_x^2 + \widetilde{\sigma}_v^2}.$$

6.10 To explore the stability of the filter, consider a univariate state-space model. That is, for $t = 1, 2, \ldots$, the observations are $y_t = x_t + v_t$ and the state equation is $x_t = \phi x_{t-1} + w_t$, where $\sigma_w = \sigma_v = 1$ and $|\phi| < 1$. The initial state, x_0, has zero mean and variance one.

(a) Exhibit the recursion for P_t^{t-1} in Property 6.1 in terms of P_{t-1}^{t-2}.
(b) Use the result of (a) to verify P_t^{t-1} approaches a limit ($t \to \infty$) P that is the positive solution of $P^2 - \phi^2 P - 1 = 0$.
(c) With $K = \lim_{t\to\infty} K_t$ as given in Property 6.1, show $|1 - K| < 1$.
(d) Show, in steady-state, the one-step-ahead predictor, $y_{n+1}^n = E(y_{n+1} \mid y_n, y_{n-1}, \ldots)$, of a future observation satisfies

$$y_{n+1}^n = \sum_{j=0}^{\infty} \phi^j K (1 - K)^{j-1} y_{n+1-j}.$$

6.11 In §6.3, we discussed that it is possible to obtain a recursion for the gradient vector, $-\partial \ln L_Y(\Theta)/\partial\Theta$. Assume the model is given by (6.1) and (6.2) and A_t is a known design matrix that does not depend on Θ, in which case Property 6.1 applies. For the gradient vector, show

$$\partial \ln L_Y(\Theta)/\partial\Theta_i = \sum_{t=1}^{n} \left\{ \boldsymbol{\epsilon}_t' \Sigma_t^{-1} \frac{\partial \boldsymbol{\epsilon}_t}{\partial \Theta_i} - \frac{1}{2} \boldsymbol{\epsilon}_t' \Sigma_t^{-1} \frac{\partial \Sigma_t}{\partial \Theta_i} \Sigma_t^{-1} \boldsymbol{\epsilon}_t + \frac{1}{2} \mathrm{tr} \left(\Sigma_t^{-1} \frac{\partial \Sigma_t}{\partial \Theta_i} \right) \right\},$$

where the dependence of the innovation values on Θ is understood. In addition, with the general definition $\partial_i g = \partial g(\Theta)/\partial\Theta_i$, show the following recursions, for $t = 2, \ldots, n$ apply:

(i) $\partial_i \boldsymbol{\epsilon}_t = -A_t \; \partial_i \boldsymbol{x}_t^{t-1}$,
(ii) $\partial_i \boldsymbol{x}_t^{t-1} = \; \partial_i \Phi \; \boldsymbol{x}_{t-1}^{t-2} + \Phi \; \partial_i \boldsymbol{x}_{t-1}^{t-2} + \; \partial_i K_{t-1} \; \boldsymbol{\epsilon}_{t-1} + K_{t-1} \; \partial_i \boldsymbol{\epsilon}_{t-1}$,
(iii) $\partial_i \Sigma_t = A_t \; \partial_i P_t^{t-1} A_t' + \; \partial_i R$,
(iv) $\partial_i K_t = \left[\; \partial_i \Phi \; P_t^{t-1} A_t' + \Phi \; \partial_i P_t^{t-1} \; A_t' - K_t \; \partial_i \Sigma_t \right] \; \Sigma_t^{-1}$,
(v) $\partial_i P_t^{t-1} = \; \partial_i \Phi \; P_{t-1}^{t-2} \Phi' + \Phi \; \partial_i P_{t-1}^{t-2} \; \Phi' + \Phi \; P_{t-1}^{t-2} \; \partial_i \Phi' + \; \partial_i Q,$
$\quad - \; \partial_i K_{t-1} \; \Sigma_t K_{t-1}' - K_{t-1} \; \partial_i \Sigma_t \; K_{t-1}' - K_{t-1} \Sigma_t \; \partial_i K_{t-1}',$

using the fact that $P_t^{t-1} = \Phi P_{t-1}^{t-2} \Phi' + Q - K_{t-1} \Sigma_t K_{t-1}'$.

6.12 Continuing with the previous problem, consider the evaluation of the Hessian matrix and the numerical evaluation of the asymptotic variance–covariance matrix of the parameter estimates. The information matrix satisfies

$$E \left\{ -\frac{\partial^2 \ln L_Y(\Theta)}{\partial\Theta \; \partial\Theta'} \right\} = E \left\{ \left(\frac{\partial \ln L_Y(\Theta)}{\partial\Theta} \right) \left(\frac{\partial \ln L_Y(\Theta)}{\partial\Theta} \right)' \right\};$$

see Anderson (1984, Section 4.4), for example. Show the (i, j)-th element of the information matrix, say, $\mathcal{I}_{ij}(\Theta) = E\left\{ -\partial^2 \ln L_Y(\Theta)/\partial\Theta_i \; \partial\Theta_j \right\}$, is

$$\mathcal{I}_{ij}(\Theta) = \sum_{t=1}^{n} E \Big\{ \; \partial_i \boldsymbol{\epsilon}_t' \; \Sigma_t^{-1} \; \; \partial_j \boldsymbol{\epsilon}_t + \frac{1}{2} \mathrm{tr} \big(\Sigma_t^{-1} \; \partial_i \Sigma_t \; \Sigma_t^{-1} \; \partial_j \Sigma_t \big) + \frac{1}{4} \mathrm{tr} \left(\Sigma_t^{-1} \; \partial_i \Sigma_t \right) \mathrm{tr} \big(\Sigma_t^{-1} \; \partial_j \Sigma_t \big) \Big\}.$$

Consequently, an approximate Hessian matrix can be obtained from the sample by dropping the expectation, E, in the above result and using only the recursions needed to calculate the gradient vector.

Section 6.4

6.13 As an example of the way the state-space model handles the missing data problem, suppose the first-order autoregressive process

$$x_t = \phi x_{t-1} + w_t$$

has an observation missing at $t = m$, leading to the observations $y_t = A_t x_t$, where $A_t = 1$ for all t, except $t = m$ wherein $A_t = 0$. Assume $x_0 = 0$ with variance $\sigma_w^2/(1-\phi^2)$, where the variance of w_t is σ_w^2. Show the Kalman smoother estimators in this case are

$$x_t^n = \begin{cases} \phi y_1 & t = 0, \\ \frac{\phi}{1+\phi^2}(y_{m-1} + y_{m+1}) & t = m, \\ y_, & t \neq 0, m, \end{cases}$$

with mean square covariances determined by

$$P_t^n = \begin{cases} \sigma_w^2 & t = 0, \\ \sigma_w^2/(1+\phi^2) & t = m, \\ 0 & t \neq 0, m. \end{cases}$$

6.14 The data set `ar1miss` is $n = 100$ observations generated from an AR(1) process, $x_t = \phi x_{t-1} + w_t$, with $\phi = .9$ and $\sigma_w = 1$, where 10% of the data has been zeroed out at random. Considering the zeroed out data to be missing data, use the results of Problem 6.13 to estimate the parameters of the model, ϕ and σ_w, using the EM algorithm, and then estimate the missing values.

Section 6.5

6.15 Using Example 6.10 as a guide, fit a structural model to the Federal Reserve Board Production Index data and compare it with the model fit in Example 3.46.

Section 6.6

6.16 (a) Fit an AR(2) to the recruitment series, R_t in `rec`, and consider a lag-plot of the residuals from the fit versus the SOI series, S_t in `soi`, at various lags, S_{t-h}, for $h = 0, 1, \ldots$. Use the lag-plot to argue that S_{t-5} is reasonable to include as an exogenous variable.

(b) Fit an ARX(2) to R_t using S_{t-5} as an exogenous variable and comment on the results; include an examination of the innovations.

6.17 Use Property 6.6 to complete the following exercises.

(a) Write a univariate AR(1) model, $y_t = \phi y_{t-1} + v_t$, in state-space form. Verify your answer is indeed an AR(1).
(b) Repeat (a) for an MA(1) model, $y_t = v_t + \theta v_{t-1}$.
(c) Write an IMA(1,1) model, $y_t = y_{t-1} + v_t + \theta v_{t-1}$, in state-space form.

6.18 Verify Property 6.5.

6.19 Verify Property 6.6.

Section 6.7

6.20 Repeat the bootstrap analysis of Example 6.13 on the entire three-month Treasury bills and rate of inflation data set of 110 observations. Do the conclusions of Example 6.13—that the dynamics of the data are best described in terms of a fixed, rather than stochastic, regression—still hold?

Section 6.8

6.21 Fit the switching model described in Example 6.15 to the growth rate of GNP. The data are in `gnp` and, in the notation of the example, y_t is log-GNP and ∇y_t is the growth rate. Use the code in Example 6.17 as a guide.

Section 6.9

6.22 Use the material presented in Example 6.21 to perform a Bayesian analysis of the model for the Johnson & Johnson data presented in Example 6.10.

6.23 Verify (6.194) and (6.195).

6.24 Verify (6.200) and (6.207).

Section 6.10

6.25 Fit a stochastic volatility model to the returns of one (or more) of the four financial time series available in the R datasets package as `EuStockMarkets`.