

Computing Observation Weights for Signal Extraction and Filtering

Siem Jan Koopman and Andrew Harvey
Department of Econometrics, Free University Amsterdam
Faculty of Economics and Politics, University of Cambridge

September 1999

Abstract

We present algorithms for computing the weights implicitly assigned to observations when estimating unobserved components using a model in state space form. The algorithms are for both filtering and signal extraction. In linear time-invariant models such weights can sometimes be obtained analytically from the Wiener-Kolmogorov formulae. Our method is much more general, being applicable to any model with a linear state space form, including models with deterministic components and time-varying state matrices. It applies to multivariate models and it can be used when there are data irregularities, such as missing observations.

The algorithms can be useful for a variety of purposes in econometrics and statistics: (i) the weights for signal extraction can be regarded as equivalent kernel functions and hence the weight pattern can be compared with the kernels typically used in nonparametric trend estimation; (ii) the weight algorithm for filtering implicitly computes the coefficients of the vector error-correction model (VECM) representation of any linear time series model; (iii) as a by-product the mean square errors associated with estimators may be obtained; (iv) the algorithm can be incorporated within a Markov chain Monte Carlo (MCMC) method enabling computation of weights assigned to observations when computing the posterior mean of unobserved components within a Bayesian treatment.

A wide range of illustrations show how the algorithms may provide important insights in empirical analysis. The algorithms are provided and implemented for the software package *SsfPack* 2.3, that is a set of filtering, smoothing and simulation algorithms for models in state space form (see www.ssfpack.com). Some details of implementation and example programs are given in the appendix of the paper.

KEYWORDS: KALMAN FILTER AND SMOOTHER; KERNEL SMOOTHING; MARKOV CHAIN MONTE CARLO; MOVING AVERAGE; NONPARAMETRIC REGRESSION; SPLINE SMOOTHING; STATE SPACE MODEL; WIENER-KOLMOGOROV FILTER.

JEL classification: C15, C22.

1 Introduction

Time series models are used for prediction and signal extraction. These operations are carried out using a variety of algorithms. If a model can be put in state space form (SSF) the computations are done by the Kalman filter and smoother (KFS); the latest version of this algorithm is described in Durbin and Koopman (2000). The KFS implicitly assigns weights to the various observations in making predictions and forming smoothed estimates of the components at different points in time. Examining these weighting patterns can be very informative. It helps the user to understand what a model actually does, and it enables comparisons to be made with other methods. For example,

nonparametric procedures may be used to estimate trends and it is interesting to compare the weighting patterns of various kernels, or moving averages, with the weighting patterns implied by an unobserved components (UC) time series model.

For time-invariant UC models, the Wiener-Kolmogorov (WK) filter provides expressions for filtering and smoothing weights; see Whittle (1983). However, this analytic solution, while very elegant, is often difficult to implement. In such circumstances numerical values of the weights can often be computed by the well-known device of running the appropriate algorithm on a set of artificially constructed observations all of which are zero except for the one at the point of interest which is unity. This device, which we call the zero-one method, cannot be used when systems are time-varying, or there are data irregularities such as missing observations. Nor can it be used for time-invariant UC models with correlated components if interest centres on the weights for estimators near the beginning or end of the sample. Even with uncorrelated components filtering weights cannot be accurately obtained if a steady-state has not been reached. The aim of this paper is to present an algorithm that can be used to compute the exact weights for any model which has a linear state space form. Thus we can, amongst other things, deal with short time series, deterministic components, heteroscedasticity, missing observations, and multivariate models. The weights are optimal if the model is Gaussian. In the absence of Gaussianity they are optimal within the class of linear estimators.

Further aspects of the weight algorithms are as follows. Firstly, the predictive filtering weights can be interpreted as the coefficients of a (vector) autoregressive (VAR) representation of a time series model in state space form. This can be of considerable interest in econometrics; for example, if there is co-integration it is possible to obtain the vector error-correction model (VECM) representation. Secondly, within a Bayesian framework the algorithms can be used in conjunction with a Markov chain Monte Carlo (MCMC) procedure to compute the weights for the posterior mean of the state vector. Finally, it is informative to compare weight patterns with kernels typically used in nonparametric trend estimation. Because of the known connection between cubic spline smoothing and trend estimation in a UC model the spline weights may be obtained and interpreted as an implicit kernel.

Section 2 gives a brief review of the classical Wiener-Kolmogorov filter, while the Kalman filter and smoother (KFS) is discussed in section 3. The algorithms for computing weights for predictive and contemporaneous filtering and for smoothing are developed in section 4; the section begins with a discussion of when the zero-one method can be used. Mean square errors of estimators can be very useful for purposes of analysis and sub-section 4.7 shows how they can be obtained as a by-product of the weighting algorithms. Applications are given in section 5 for univariate models and in section 6 for multivariate models. The appendix includes some example programs in the Ox code of Doornik (1998) using the *SsfPack* package of Koopman, Shephard and Doornik (1999).

2 Classical Formulae for Prediction and Signal Extraction

Consider a model consisting of two stochastic components, μ_t and ε_t . The classical Wiener-Kolmogorov (WK) formula for finding the weights used to extract $m_{t|\infty}$, that is the minimum mean square linear estimator (MMSLE) of μ_t in a doubly infinite sample, is

$$m_{t|\infty} = w(L)y_t = \sum_{j=-\infty}^{\infty} w_{-j}L^j y_t = \sum_{j=-\infty}^{\infty} w_j y_{t+j}, \quad w(L) = \frac{\gamma_{\mu}(L) + \gamma_{\mu\varepsilon}(L)}{\gamma_y(L)}, \quad (1)$$

where L is the lag operator, $\gamma_{\mu}(L)$ is the autocovariance generating function (ACGF) of μ_t and $\gamma_{\mu\varepsilon}(L)$ is the cross-covariance generating function of μ_t and ε_t . The ACGF of y_t is

$$\gamma_y(L) = \gamma_{\mu}(L) + \gamma_{\mu\varepsilon}(L) + \gamma_{\varepsilon\mu}(L) + \gamma_{\varepsilon\varepsilon}(L), \quad (2)$$

though this is usually evaluated in terms of the reduced form parameters. For a stationary ARMA process, written $\phi^{-1}(L)\theta(L)\xi_t$, where $\phi(L)$ and $\theta(L)$ are polynomials in the lag operator and ξ_t is

white noise with variance σ^2 , the ACGF is given directly by

$$\gamma(L) = \{|\theta(L)|^2/|\phi(L)|^2\}\sigma^2, \quad (3)$$

where $|\theta(L)|^2 = \theta(L)\theta(L^{-1})$ and similarly for $\phi(L)$.

Although formula (1) is only proved for stationary models in Whittle (1983, pp.56-58), it is argued in Bell (1984) and Burridge and Wallis (1988) that it can still be used for nonstationary models even though expressions like (3) are no longer ACGFs.

The weights for filtering and for smoothing near the end of a sample may be obtained by modifying the above signal extraction formula. Expressions are most easily derived by assuming a semi-infinite sample, in other words an infinite number of past observations are available; see Whittle (1983, chapter 6). The expressions for finite samples may also be worked out, but are more complicated; see Whittle (1983, chapter 7). The initial conditions may have important effects, particularly for nonstationary models.

2.1 Local level model

As a simple illustration of the WK formula, we consider the local level (or the random walk plus noise) model

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim \text{WN}(0, \sigma_\varepsilon^2), \quad t = 1, \dots, T, \quad (4)$$

$$\mu_{t+1} = \mu_t + \eta_t, \quad \eta_t \sim \text{WN}(0, \sigma_\eta^2), \quad (5)$$

where $q = \sigma_\eta^2/\sigma_\varepsilon^2$ is the signal-noise ratio and WN denotes ‘white noise’, that is serially uncorrelated random variables. The disturbances may be correlated contemporaneously, that is

$$\begin{aligned} \text{E}(\varepsilon_t \eta_t) &= \rho q^{1/2} \sigma_\varepsilon^2, & t, s = 1, \dots, T, \\ \text{E}(\varepsilon_t \eta_s) &= 0, & t \neq s, \end{aligned} \quad (6)$$

with $|\rho| \leq 1$. An alternative model has the transition equation written at time t , that is

$$\mu_t = \mu_{t-1} + \eta_t, \quad \eta_t \sim \text{WN}(0, q\sigma_\varepsilon^2), \quad (7)$$

with the covariance still as in (6). When the correlation, ρ , is zero the two models are essentially the same. For non-zero ρ the weighting patterns display an interesting asymmetry. This is examined in Harvey and Koopman (1999) where (5) is referred to as the *future* state form and (7) as the *contemporaneous* state form. In this paper we present algorithms for the future state form but as shown in section 3 it can be used for the contemporaneous form straightforwardly by expressing the model in future state form.

The reduced form of the local level model is an ARIMA(0,1,1) model, that is

$$\Delta y_t = \xi_t + \theta \xi_t, \quad \xi_t \sim \text{WN}(0, \sigma^2). \quad (8)$$

Assuming a local level model with uncorrelated disturbances ($\rho = 0$), the relation between the MA parameter θ and the signal-to-noise ratio q is given by

$$\theta = \{-q - 2 + \sqrt{q^2 + 4q}\}/2, \quad (9)$$

and further $\sigma^2 = -\sigma_\varepsilon^2/\theta$ for $\theta \neq 0$. Thus, provided $q > 0$, the WK formula for estimating μ_t yields

$$m_{t|\infty} = w(L)y_t = \frac{\sigma_\eta^2}{\sigma^2|1 + \theta L|^2} y_t = \frac{(1 + \theta)^2}{|1 + \theta L|^2} y_t, \quad (10)$$

as $\sigma_\eta^2 = (1 + \theta)^2 \sigma^2$. Note that $|1 + \theta L|^2$ denotes $(1 + \theta L)(1 + \theta L^{-1})$ where L is the lag operator. The weights decline symmetrically and exponentially, that is

$$w_j = \{(1 + \theta)/(1 - \theta)\}(-\theta)^{|j|}, \quad j = 0, 1, 2, \dots \quad (11)$$

On setting $L = 1$ in (10) it can be immediately seen that the weights sum to unity. When the components are correlated, the weighting pattern is asymmetric; see Harvey and Koopman (1999).

At the end of a semi-infinite sample

$$w_{-j} = (1 + \theta)(-\theta)^j, \quad j = 0, 1, 2, \dots \quad (12)$$

This is the filtered estimator which here is an exponentially weighted moving average (EWMA). The weights for the smoothed estimator of μ_t near the end of a semi-infinite sample are given in Whittle (1983, p.69) as:

$$w_j = \{(1 + \theta)/(1 - \theta)\}[(-\theta)^{|-j|} + (-\theta)^{-j+2(T-t)+1}], \quad -\infty < j \leq T - t. \quad (13)$$

Setting $t = T$ gives the weights for the filtered estimator (12), while if $t \ll T$, the weights are as for a doubly infinite sample as given in (11).

2.2 Local linear trend

In the local linear trend model

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\sim \text{WN}(0, \sigma_\varepsilon^2), \\ \mu_{t+1} &= \mu_t + \beta_t + \eta_t, & \eta_t &\sim \text{WN}(0, \sigma_\eta^2), \\ \beta_{t+1} &= \beta_t + \zeta_t, & \zeta_t &\sim \text{WN}(0, \sigma_\zeta^2), \end{aligned} \quad (14)$$

we may be interested in estimating both the trend, μ_t , and the slope, β_t . The weights may be found from the WK formula by noting that the reduced form is $ARIMA(0, 2, 2)$:

$$\Delta^2 y_t = \xi_t + \theta_1 \xi_{t-1} + \theta_2 \xi_{t-2} = \theta_2(L)\xi_t, \quad \xi_t \sim \text{WN}(0, \sigma^2).$$

The weights for the slope are obtained by first writing

$$\Delta y_t = \beta_{t-1} + \eta_{t-1} + \Delta \varepsilon_t.$$

Application of the WK filter then gives a set of weights attached to first differences of the observations, that is

$$b_{t|\infty} = w_\beta(L)\Delta y_{t+1} = \frac{\sigma_\zeta^2}{\sigma^2|\theta_2(L)|^2}\Delta y_{t+1}.$$

Since the denominator is equal to $\sigma_\zeta^2 + |1 - L|^2 \sigma_\eta^2 + |1 - L|^4 \sigma_\varepsilon^2$, it can be seen immediately that $w_\beta(1) = 1$, in other words the weights sum to one. The weights attached to the observations themselves are given by $w_\beta(L)(1 - L)L^{-1}$ and these clearly sum to zero. On the other hand, the weights for estimating the level of the trend are given by

$$w(L) = \frac{\sigma_\zeta^2 + |1 - L|^2 \sigma_\eta^2}{\sigma_\zeta^2 + |1 - L|^2 \sigma_\eta^2 + |1 - L|^4 \sigma_\varepsilon^2},$$

and these sum to one.

The smooth trend, obtained by setting $\sigma_\eta^2 = 0$, is of particular interest, one reason being that it is equivalent to a cubic spline. Harvey and Koopman (1999) present some weighting patterns for

trend extraction. It follows immediately from the above formulae that $w_\beta(L) = w(L)$. It is interesting that this weighting function for extracting the level from the observations is the same as the one for extracting the slope from the first differences. However, this is not surprising since it follows that

$$b_{t|\infty} = w_\beta(L)(1-L)y_{t+1} = m_{t+1|\infty} - m_{t|\infty},$$

and this corresponds to the identity in the model. This identity still holds in finite samples.

The exact weights algorithm for smoothing that we present in section 4 confirms that weights for level and slope (the latter in terms of first differences) still sum to unity for finite samples even when observations are missing.

2.3 Mean square errors of estimated components

Applying the formula in Whittle (1983, p.58) indicates that the error in estimating the signal in a model of the form (4) has an ACGF given by

$$[\gamma_\mu(L)\gamma_\varepsilon(L) - \gamma_{\mu\varepsilon}(L)\gamma_{\varepsilon\mu}(L)]/\gamma_y(L). \quad (15)$$

The MSE of the estimator of the signal is given by the variance of the estimation error which is obtained as the coefficient of L^0 in (15). The formula may be adapted to semi-infinite samples. Thus in the local level model with uncorrelated components the MSE associated with (13) is shown by Whittle (1983, p.70) to be

$$\sigma_\varepsilon^2[(1+\theta)/(1-\theta)][1-\theta(-\theta)^{2(T-t)}], \quad t \leq T. \quad (16)$$

The MSE of the filtered estimator, (12), is obtained when $t = T$, so it is $\sigma_\varepsilon^2(1+\theta)$, while if $T-t$ is large we get the MSE of the smoother in a doubly infinite sample, that is

$$\text{MSE}(m_{t|\infty}) = \sigma_\varepsilon^2[(1+\theta)/(1-\theta)]. \quad (17)$$

2.4 Multivariate models

The WK formula (1) can be generalized to a multivariate model as in Whittle (1983, ch9). If μ_t is a $K \times 1$ vector of unobserved components to be extracted from a series of observation vectors of length N , the $K \times N$ weighting matrix is given by

$$W(L) = \Gamma_{\mu y}(L)\Gamma_y^{-1}(L),$$

where $\Gamma(L)$ denotes a multivariate ACGF. Unfortunately, it is difficult to obtain expressions for the reduced form parameters in terms of the parameters of the unobserved components model. (Whittle refers to this as canonical factorisation). Consider the multivariate local level model

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim \text{WN}(0, \Sigma_\varepsilon), \quad t = 1, \dots, T, \quad (18)$$

$$\mu_{t+1} = \mu_t + \eta_t, \quad \eta_t \sim \text{WN}(0, \Sigma_\eta), \quad (19)$$

where y_t , μ_t , ε_t and η_t are all $N \times 1$ vectors so that, for example, $y_t = (y_{1t}, y_{2t}, \dots, y_{Nt})'$. The reduced form is

$$\Delta y_t = \xi_t + \Theta \xi_{t-1}, \quad \xi_t \sim \text{WN}(0, \Sigma) \quad t = 2, \dots, T,$$

but unless $N = 1$, it does not seem to be possible to find a formula expressing the $N \times N$ moving average matrix Θ in terms of Σ_ε and Σ_η . However, we can at least write

$$W(L) = \Sigma_\eta[\Sigma_\eta + |1-L|^2\Sigma_\varepsilon]^{-1},$$

from which it follows that $W(1) = I$. Thus in estimating the i -th element in μ_t , the weights on the observations in the i -th series sum to one while the weights on the other series all sum to zero. An analogous result holds in the multivariate local linear trend model, both for the level and the slope, that is $W(1) = I$ and $W_\beta(1) = I$.

3 Kalman filter and smoother

The future form of the linear state space model is given by

$$\begin{aligned} y_t &= Z_t \alpha_t + G_t \varepsilon_t, & \varepsilon_t &\sim \text{WN}(0, I), & t &= 1, \dots, T, \\ \alpha_{t+1} &= T_t \alpha_t + H_t \varepsilon_t, & \alpha_1 &\sim \text{WN}(a, P), \end{aligned} \quad (20)$$

where y_t is the $N \times 1$ vector of observations, α_t is the $p \times 1$ state vector and ε_t is the $q \times 1$ vector of disturbances. The equation for y_t is called the measurement equation and the equation for α_{t+1} is the transition equation. The standardized disturbance vector ε_t appears in both equations but the disturbances in the two equations are mutually uncorrelated if $H_t G_t' = 0$. The initial state vector is to have mean a and variance matrix P . The system matrices Z_t , G_t , T_t and H_t , with appropriate dimensions, are fixed. The state space model (20) is said to be time-invariant when the system matrices are constant over time t , that is $Z_t = Z$, $T_t = T$, $G_t = G$ and $H_t = H$, for $t = 1, \dots, T$. An introduction to statistical analysis based on state space models is given by, for example, Harvey (1989), Kitagawa and Gersch (1996), West and Harrison (1997) and Durbin and Koopman (2000).

The contemporaneous form of the state space model is

$$\begin{aligned} y_t &= Z_t^* \alpha_t^* + G_t^* \varepsilon_t, & \varepsilon_t &\sim \text{WN}(0, I), & t &= 1, \dots, T, \\ \alpha_t^* &= T_t \alpha_{t-1}^* + H_t \varepsilon_t, & \alpha_0^* &\sim \text{WN}(a, P). \end{aligned} \quad (21)$$

By setting $\alpha_t^* = \alpha_{t+1}$, the future state model is obtained with

$$y_t = Z_t^* T_t \alpha_t + (G_t^* + Z_t^* H_t) \varepsilon_t, \quad (22)$$

such that $Z_t = Z_t^* T_t$ and $G_t = G_t^* + Z_t^* H_t$.

A predictive filtered estimator is an estimator of (a function of) the state vector at time $t + 1$ based on observations up to and including time t . The Kalman filter computes the predictive filtered estimator $a_{t+1|t}$, the MMSLE of the state vector α_{t+1} conditional on the observations $Y_t = \{y_1, \dots, y_t\}'$, together with its MSE matrix, that is the covariance matrix of the estimation error, $P_{t+1|t}$. The Kalman filter is given by

$$\begin{aligned} v_t &= y_t - Z_t a_{t|t-1}, & F_t &= Z_t P_{t|t-1} Z_t' + G_t G_t', \\ & & M_t &= T_t P_{t|t-1} Z_t' + H_t G_t', & t &= 1, \dots, T, \\ a_{t+1|t} &= T_t a_{t|t-1} + K_t v_t, & P_{t+1|t} &= T_t P_{t|t-1} T_t' + H_t H_t' - K_t M_t', \end{aligned} \quad (23)$$

with Kalman gain matrix $K_t = M_t F_t^{-1}$ and initialisation $a_{1|0} = a$ and $P_{1|0} = P$. The one-step ahead prediction error is v_t with covariance matrix $\text{Var}(v_t) = F_t$. The filtered estimator $a_{t|t}$, the MMSLE of the state vector α_t conditional on Y_t and its MSE matrix $P_{t|t}$, can be computed by

$$a_{t|t} = a_{t|t-1} + P_{t|t-1} Z_t' F_t^{-1} v_t, \quad P_{t|t} = P_{t|t-1} - P_{t|t-1} Z_t' F_t^{-1} Z_t P_{t|t-1}, \quad t = 1, \dots, T. \quad (24)$$

For a time-invariant state space model, the Kalman recursion for P_{t+1} may converge to a constant matrix \bar{P} for a sufficiently large t . The steady state is the solution to the matrix equation

$$\bar{P} = T \bar{P} T' + H H' - \bar{K} \bar{M}',$$

where $\bar{M} = T \bar{P} Z' + H G'$, $\bar{F} = Z \bar{P} Z' + G G'$ and $\bar{K} = \bar{M} \bar{F}^{-1}$. Use of the steady state after convergence usually leads to considerable computational savings because the computations for F_t , M_t , K_t and P_{t+1} are no longer required.

Smoothed estimators of the state and disturbance vector, which are conditional on all observations Y_T , can be evaluated by the backwards recursion

$$r_{t-1} = Z_t' F_t^{-1} v_t + L_t' r_t, \quad N_{t-1} = Z_t' F_t^{-1} Z_t + L_t' N_t L_t, \quad t = T, \dots, 1, \quad (25)$$

where $L_t = T_t - K_t Z_t$ and with initialisation $r_T = 0$ and $N_T = 0$. The Kalman filter output of v_t , F_t^{-1} and K_t must be stored for $t = 1, \dots, T$. The smoothed state vector $a_{t|T}$ with MSE matrix $P_{t|T}$ can be evaluated by

$$a_{t|T} = a_{t|t-1} + P_{t|t-1} r_{t-1}, \quad P_{t|T} = P_{t|t-1} - P_{t|t-1} N_{t-1} P_{t|t-1}, \quad t = T, \dots, 1. \quad (26)$$

A substantive amount of additional memory space is required to store $a_{t|t-1}$ and $P_{t|t-1}$ of the Kalman filter. The smoothed disturbance vector is computed by

$$e_{t|T} = G_t' F_t^{-1} v_t + J_t' r_t, \quad \text{Var}(e_{t|T}) = G_t' F_t^{-1} G_t + J_t' N_t J_t, \quad t = T, \dots, 1, \quad (27)$$

where $J_t = H_t - K_t G_t'$. In the case where $H_t G_t' = 0$, we obtain

$$\begin{aligned} G_t e_{t|T} &= G_t G_t' e_t^*, & \text{Var}(G_t e_{t|T}) &= G_t G_t' D_t G_t G_t', \\ H_t e_{t|T} &= H_t H_t' r_t, & \text{Var}(H_t e_{t|T}) &= H_t H_t' N_t H_t H_t', \end{aligned} \quad (28)$$

with

$$e_t^* = F_t^{-1} v_t - K_t' r_t, \quad D_t = F_t^{-1} + K_t' N_t K_t, \quad t = T, \dots, 1.$$

Disturbance smoothing does not require the storage of $a_{t|t-1}$ and $P_{t|t-1}$ but it still enables us to compute $a_{t|T}$ by the forwards recursion

$$a_{t+1|T} = T_t a_{t|T} + H_t e_{t|T}, \quad t = 1, \dots, T,$$

where $a_{1|T} = a + P r_0$.

4 Computation of the observation weights

In this section we present general algorithms for computing the weights implicitly assigned to observations when estimating (linear combinations of) elements of the state vector. These quantities typically correspond to trends, seasonals, cycles and other components of interest. There are two filtering algorithms, one for predictive filtering, that is computing the vector of weights used to form $a_{t|t-1}$, and another for contemporaneous filtering, that is for $a_{t|t}$. The smoothing algorithm is for $a_{t|T}$, and associated with this is an algorithm for computing weights used to extract estimators of the disturbances. The algorithms are easy to implement and have been included in the *SsfPack 2.3* package of Koopman *et al* (1999) as the function `SsfWeights()`. *Ssfpack* is written for the *Ox 2.1* programming language of Doornik (1998) and the latest version can be obtained from www.ssfpack.com; some examples of the implementation of the algorithms are given in the appendix.

Before presenting the new KFS algorithms, we discuss the idea behind the zero-one method for computing weights and analyse the conditions under which it can be correctly applied.

4.1 The zero-one method

Consider a time-invariant UC model and suppose that we are interested in the trend component μ_t , as for example in the local level model of section 2. The weights for computing $m_{t|T}$, the MMSLE of μ_t based on all the observations, will be denoted as $w_j(m_{t|T})$ such that

$$m_{t|T} = \sum_{j=1}^T w_j(m_{t|T}) y_j, \quad t = 1, \dots, T. \quad (29)$$

Given an algorithm for computing $m_{t|T}$, it is often suggested that the weights can be obtained by running the algorithm on an artificial series consisting entirely of zeroes except for a one in the t -th

position; for, example, see Härdle (1991, p.59) or Riani (1998). We will call this the *zero-one method*. If the weights are independent of time then, we can write

$$w_j = w_{t+j}(m_{t|T}), \quad j = \dots, -2, -1, 0, 1, 2, \dots, \quad 1 \leq t \leq T,$$

and

$$m_{t+j|T} = \sum_i w_i y_{t+j+i} = w_{-j}, \quad j = \dots, -2, -1, 0, 1, 2, \dots \quad (30)$$

To illustrate this, take $T = 3$ so that

$$\begin{aligned} m_{1|T} &= w_0 y_1 + w_1 y_2 + w_2 y_3, \\ m_{2|T} &= w_{-1} y_1 + w_0 y_2 + w_1 y_3, \\ m_{3|T} &= w_{-2} y_1 + w_{-1} y_2 + w_0 y_3. \end{aligned}$$

For the zero-one method with $t = 2$, we have $y_1 = y_3 = 0$ and $y_2 = 1$ such that

$$m_{1|T} = w_1, \quad m_{2|T} = w_0, \quad m_{3|T} = w_{-1}.$$

Thus if the weighting pattern is asymmetric ($w_j \neq w_{-j}$), the weights appear in reverse order. The WK formula (1), which is for a doubly infinite sample and a time-invariant model, applies in this situation but it is not always easy to evaluate.

In order to investigate the extent to which the zero-one method applies more generally, we consider filtering and smoothing for the local level model as a matrix operation for a finite sample of size T , so we have

$$m = W y, \quad (31)$$

where $m = (m_{1|T}, \dots, m_{T|T})'$, W is a $T \times T$ matrix with the i, j -th element equal to $w_j(m_{i|T})$ and $y = (y_1, \dots, y_T)'$. The zero-one method amounts to replacing the $T \times 1$ vector y by a $T \times 1$ selection vector ℓ_t , where ℓ_t is the t -th column of the identity matrix I_T . However, this device can only be used when the matrix W is of a special structure because when we postmultiply W by ℓ_t we obtain the t -th column of W whereas what we want is its t -th row. Only when W is symmetric, is the t -th row the same as the t -th column. When W is not necessarily symmetric, but has a band structure, the t -th row is the same as the t -th column in reverse order. The band structure means the weights are independent of time.

When the components are mutually uncorrelated, W is symmetric for smoothing (in this case, $W = V_\mu V_y^{-1}$ where V_μ and V_y are the covariance matrices of the vectors μ and y respectively). Hence the KFS applied to $y = \ell_t$ computes the correct weights in the right order. The method not only works for the smoother near the end of the sample, it also gives the (contemporaneous) filter weights at time T since these are the same as the weights for the smoother at time T .

With correlated components, W is not symmetric but the correct weights are obtained in reverse order provided the smoother is in a steady-state, that is W has a band structure and the weights are time-invariant as in (30). This will not be the case near the end of the sample. When a filter is in a steady-state it implies a lower triangular band structure for W such that the t -th row is the reverse of the t -th column at least for elements close to (t, t) . Therefore, a Kalman filter applied to the observations $y = \ell_t$ produces the t -th row of W but in reverse order. The initialisation means that the zero-one method may only give the weights for large t , and then perhaps only approximately.

When deterministic components are present, the zero-one method for smoothing still works because of the symmetric weight patterns, but the filter doesn't because it never gets to a steady-state with non-zero weights; see Harvey (1989, p.108).

Thus the validity of the zero-one method is limited even for time-invariant models. For models with time-varying system matrices and for data irregularities, such as missing observations, the zero-one method does not work because the t -th column of W is not equal to the t -th row, even in reverse order.

4.2 Computing weights for predictive filtering

The filtered estimator of the state vector in the next time period, that is the estimator of α_t based on information available at time $t - 1$, can be written as

$$a_{t|t-1} = \sum_{j=1}^{t-1} w_j(a_{t|t-1}) y_j. \quad (32)$$

After the Kalman filter has been applied up to time $t - 1$ and the matrices K_j stored for $j = 1, \dots, t - 1$, the weight vectors can be computed by the backward recursion

$$w_j(x_t) = B_{t,j}K_j, \quad B_{t,j-1} = B_{t,j}T_j - w_j(x_t)Z_j, \quad j = t - 1, t - 2, \dots, 1, \quad (33)$$

with $x_t = a_{t|t-1}$ and $B_{t,t-1} = I$. We have introduced the variable x_t because (33) will also be used for computing the weights of other estimated variables. The weights for the one-step ahead estimator of the observation y_t , denoted by $\hat{y}_{t|t-1}$ for which $\hat{y}_{t|t-1} = Z_t a_{t|t-1}$ and

$$\hat{y}_{t|t-1} = \sum_{j=1}^{t-1} w_j(\hat{y}_{t|t-1}) y_j,$$

are given simply by $w_j(\hat{y}_{t|t-1}) = Z_t w_j(a_{t|t-1})$ for $j = t - 1, t - 2, \dots, 1$. This leads to the backward recursion

$$w_j(x_t) = b_{t,j}K_j, \quad b_{t,j-1} = b_{t,j}T_j - w_j(x_t)Z_j, \quad j = t - 1, t - 2, \dots, 1, \quad (34)$$

with $x_t = \hat{y}_{t|t-1}$ and $b_{t,t-1} = Z_t$. Note that $b_{t,j} = Z_t B_{t,j}$. In the case of univariate observations y_t , $b_{t,j}$ is a vector.

Vector autoregressive representation

Since $\hat{y}_{t|t-1} + v_t = y_t$, the weights give the (vector) autoregressive (VAR) representation for a time-invariant model when the filter is in a steady state; see section 3. In more familiar notation

$$y_t = \sum_k^{\infty} \Phi_k y_{t-k} + v_t, \quad \text{Var}(v_t) = \bar{F}, \quad (35)$$

where $\Phi_k = w_{t-k}(\hat{y}_{t|t-1})$. The lag length will usually be infinite for a UC model, but the weights will tend to die away as k increases. If deterministic components, such as means, are present, some modification to the computed weights may be required in small samples. The relevance of the VAR representation for the (vector) error-correction model (VECM) is discussed in section 6.3.

Derivation

The Kalman recursion for $a_{t+1|t}$ can be re-formulated as

$$a_{t+1|t} = L_t a_{t|t-1} + K_t y_t,$$

where $L_t = T_t - K_t Z_t$. By simple substitution, it follows that

$$\begin{aligned} a_{t|t-1} &= K_{t-1}y_{t-1} + L_{t-1}K_{t-2}y_{t-2} + L_{t-1}L_{t-2}K_{t-3}y_{t-3} \\ &\quad + L_{t-1}L_{t-2}L_{t-3}K_{t-4}y_{t-4} + \dots + L_{t-1} \dots L_2 K_1 y_1 \\ &= \sum_{j=1}^{t-1} B_{t,j} K_j y_j \end{aligned}$$

where

$$B_{t,t-1} = I, \quad B_{t,j} = L_{t-1}L_{t-2} \dots L_{j+1}, \quad j = 1, \dots, t-2. \quad (36)$$

Thus the weight vectors in (32) are given by $w_j(a_{t|t-1}) = B_{t,j}K_j$. They can be computed efficiently by the backwards recursion (33) since it follows from (36) that

$$B_{t,j-1} = B_{t,j}L_j, \quad j = t-1, t-2, \dots, 1.$$

Minor manipulation, using the definition $L_j = T_j - K_jZ_j$, gives (33).

4.3 Computing weights for (contemporaneous) filtering

The estimator of the state vector α_t given the observations up to and including time t is referred to as the (contemporaneous) filtered estimator, $a_{t|t}$, and it can be represented as

$$a_{t|t} = \sum_{j=1}^t w_j(a_{t|t}) y_j. \quad (37)$$

After the Kalman filter is applied together with the backwards recursion (33), the weight vector at time t is computed by $w_t(a_{t|t}) = P_{t|t-1}Z'_tF_t^{-1}$ and the weight vectors for $j = t-1, t-2, \dots, 1$ are the same as for $a_{t|t-1}$ but premultiplied by the matrix $I - P_{t|t-1}Z'_tF_t^{-1}Z_t$. The filtering weights for $j < t$ can be computed by the backwards recursion (33) with $x_t = a_{t|t}$ and replacing the initialisation $B_{t,t-1} = I$ with $B_{t,t-1} = I - P_{t|t-1}Z'_tF_t^{-1}Z_t$.

The filtered estimator of the observation y_t is given by $\hat{y}_{t|t}$ and can be represented as

$$\hat{y}_{t|t} = \sum_{j=1}^t w_j(\hat{y}_{t|t}) y_j.$$

The weight at time t is computed by $w_t(\hat{y}_{t|t}) = I - G_tG'_tF_t^{-1}$ and the weights $w_j(\hat{y}_{t|t})$ for $j = t-1, t-2, \dots, 1$ are the same as $w_j(\hat{y}_t)$ but need to be premultiplied by the matrix $G_tG'_tF_t^{-1}$. Thus we can use recursion (34) with $x_t = \hat{y}_{t|t}$ and initialisation $b_{t,t-1} = G_tG'_tF_t^{-1}Z_t$.

Derivation

The filtered estimator of the state vector at time t is given by

$$\begin{aligned} a_{t|t} &= a_{t|t-1} + P_{t|t-1}Z'_tF_t^{-1}v_t \\ &= (I - P_{t|t-1}Z'_tF_t^{-1}Z_t)a_{t|t-1} + P_{t|t-1}Z'_tF_t^{-1}y_t. \end{aligned}$$

The weights for contemporaneous filtering are closely related to the weights for predictive filtering, $w_j(a_{t|t-1})$, since we have

$$\begin{aligned} w_t(a_{t|t}) &= P_{t|t-1}Z'_tF_t^{-1}, \\ w_j(a_{t|t}) &= (I - P_{t|t-1}Z'_tF_t^{-1}Z_t)w_j(a_{t|t-1}), \quad j = t-1, t-2, \dots, 1. \end{aligned}$$

Therefore, recursion (33) can be used with $x_t = a_{t|t}$ and $B_{t,t-1} = I - P_{t|t-1}Z'_tF_t^{-1}Z_t$.

The weights for the filtered estimator $\hat{y}_{t|t}$ are given by $w_j(\hat{y}_{t|t}) = Z_t w_j(a_{t|t})$, for $j = t, t-1, \dots, 1$. Since

$$Z_t P_{t|t-1} Z'_t F_t^{-1} = I - G_t G'_t F_t^{-1}, \quad Z_t (I - P_{t|t-1} Z'_t F_t^{-1} Z_t) = G_t G'_t F_t^{-1} Z_t,$$

we obtain the result as stated.

4.4 Computing weights for smoothing

The smoothed state vector, $a_{t|T}$, can be represented as a weighted sum of all observations, that is

$$a_{t|T} = \sum_{j=1}^T w_j(a_{t|T}) y_j. \quad (38)$$

The weight matrices $w_j(a_{t|T})$ can be computed after the Kalman filter is applied for $t = 1, \dots, T$ and the smoothing recursion (25) is applied backwards up to time t . The Kalman filter stores K_j , for $j = 1, \dots, T$, and the smoother stores

$$C_j = Z_j' D_j - T_j' N_j K_j, \quad (39)$$

where $D_j = F_j^{-1} + K_j' N_j K_j$ for $j = T, T-1, \dots, t$. The weight matrices are calculated from time t in the two opposite directions: (i) backwards in time ($j = t-1, \dots, 1$); (ii) forwards in time ($j = t, t+1, \dots, T$).

The weights $w_j(a_{t|T})$ for $j < t$ are given by $w_j(a_{t|T}) = (I - P_{t|t-1} N_{t-1}) w_j(a_{t|t-1})$ and they can be computed by the backwards recursion (33) with $x_t = a_{t|T}$ and initialisation $B_{t,t-1} = I - P_{t|t-1} N_{t-1}$. The weights $w_j(a_{t|T})$ for $j \geq t$ are computed by the forwards recursion

$$w_j(x_t) = B_{t,j}^* C_j, \quad B_{t,j+1}^* = B_{t,j}^* L_j', \quad j = t+1, \dots, T, \quad (40)$$

with $x_t = a_{t|T}$ and $B_{t,t+1}^* = P_{t|t-1} L_t'$.

The smoothed estimator of the observation y_t is given by $\hat{y}_{t|T} = Z_t a_{t|T}$ and can be represented as

$$\hat{y}_{t|T} = \sum_{j=1}^n w_j(\hat{y}_{t|T}) y_j,$$

where $w_j(\hat{y}_{t|T}) = Z_t w_j(a_{t|T})$. The weight vectors $w_j(\hat{y}_{t|T})$ can be obtained directly by using the same kind of backwards and forwards operation. The backwards recursion is the same as (34) but with $x_t = \hat{y}_{t|T}$ and $b_{t,t-1} = G_t G_t' C_t'$. The forwards recursion is similar as (40), that is

$$w_j(x_t) = b_{t,j}^* C_j, \quad b_{t,j+1}^* = b_{t,j}^* L_j', \quad j = t+1, \dots, T, \quad (41)$$

with $x_t = \hat{y}_{t|T}$ and $b_{t,t+1}^* = Z_t P_{t|t-1} L_t' = G_t G_t' K_t'$. Further, note that $w_t(\hat{y}_{t|T}) = Z_t P_{t|t-1} C_t = I - G_t G_t' D_t$.

Derivation

We develop the algorithm for computing the weights for smoothing in three steps: (1) the weights for r_{t-1} ; (2) the weights for $a_{t|T}$; (3) the weights for $Z_{t|T} a_{t|T}$.

Step 1

Define

$$u_t = Z_t F_t^{-1} v_t, \quad U_t = Z_t F_t^{-1} Z_t'.$$

It follows that $U_t = \text{Var}(u_t)$ and, since $v_t = y_t - Z_t a_{t|t-1}$,

$$w_t(u_t) = Z_t F_t^{-1} - U_t w_t(a_{t|t-1}), \quad w_j(u_t) = -U_t w_j(a_{t|t-1}), \quad (42)$$

for $j = t - 1, \dots, 1$. We can rewrite the smoothing recursions (25) as

$$\begin{aligned} r_{t-1} &= u_t + L'_t r_t \\ &= u_t + L'_t u_{t+1} + L'_t L'_{t+1} u_{t+2} + \dots + L'_t \dots L'_{n-1} u_n \\ &= u_t + B'_{t+1,t-1} u_{t+1} + B'_{t+2,t-1} u_{t+2} + \dots + B'_{n,t-1} u_n, \end{aligned} \quad (43)$$

$$\begin{aligned} N_{t-1} &= U_t + L'_t N_t L_t \\ &= U_t + L'_t U_{t+1} L_t + L'_t L'_{t+1} U_{t+2} L_{t+1} L_{t+2} + \dots + L'_t \dots L'_{T-1} U_T L_{T-1} \dots L_t \\ &= U_t + B'_{t+1,t-1} U_{t+1} B_{t+1,t-1} + B'_{t+2,t-1} U_{t+2} B_{t+2,t-1} + \dots + B'_{T,t-1} U_T B_{T,t-1}, \end{aligned} \quad (44)$$

where matrix $B_{t,j}$ is defined in (36). Note some properties of matrix $B_{j,t-1}$:

$$\begin{aligned} B_{j,t-1} &= L_{j-1} B_{j-1,t-1}, & j &= t, \dots, n, \\ B_{j,t-1} &= B_{j,k} B_{k+1,t-1}, & j &= t, \dots, n, \quad k = t - 1, \dots, j - 1, \\ B_{j,t-1} &= 0, & j &= 1, \dots, t - 1. \end{aligned} \quad (45)$$

The weights for r_{t-1} are obtained by

$$w_j(r_{t-1}) = w_j(u_t) + B'_{t+1,t-1} w_j(u_{t+1}) + B'_{t+2,t-1} w_j(u_{t+2}) + \dots + B'_{T,t-1} w_j(u_T).$$

for $j = 1, \dots, n$. By substitution of (42), we obtain

$$\begin{aligned} w_j(r_{t-1}) &= B'_{j,t-1} Z_j F_j^{-1} \\ &\quad - [U_t w_j(a_{t|t-1}) + B'_{t+1,t-1} U_{t+1} w_j(a_{t+1|t}) + \dots + B'_{T,t-1} U_T w_j(a_{T|T-1})]. \end{aligned}$$

From (33) we have $w_j(a_{t|t-1}) = B_{t,j} K_j$ and substitution leads to

$$\begin{aligned} w_j(r_{t-1}) &= B'_{j,t-1} Z_j F_j^{-1} \\ &\quad - [U_t B_{t,j} + B'_{t+1,t-1} U_{t+1} B_{t+1,j} + \dots + B'_{T,t-1} U_T B_{T,j}] K_j. \end{aligned}$$

Convenient expressions for $w_j(r_{t-1})$ can be obtained by using (44) for N_{t-1} and using the properties of $B_{j,t-1}$ in (45), we have

$$w_j(r_{t-1}) = \begin{cases} -N_{t-1} B_{t,j} K_j, & j = t - 1, t - 2, \dots, 1, \\ Z_t F_t^{-1} - L'_t N_t K_t, & j = t, \\ B'_{j,t-1} Z_j F_j^{-1} - B'_{j+1,t-1} N_j K_j, & j = t + 1, t + 2, \dots, T, \end{cases}$$

Using definition $D_j = F_j^{-1} + K'_j N_j K_j$ we further define

$$C_j = Z'_j F_j^{-1} - L'_j N_j K_j = Z'_j D_j - T'_j N_j K_j, \quad j = t, t + 1, \dots, T.$$

From the fact that $B_{j+1,t-1} = L_j B_{j,t-1}$, we obtain finally

$$w_j(r_{t-1}) = \begin{cases} -N_{t-1} B_{t,j} K_j, & j = t - 1, t - 2, \dots, 1, \\ C_t, & j = t, \\ B'_{j,t-1} C_j, & j = t + 1, t + 2, \dots, T, \end{cases}$$

Step 2

For the state smoothing recursions (25) and (26), it can be shown that $a_{t|T}$ is the weighted average

$$a_{t|T} = \sum_{j=1}^T w_j(a_{t|T}) y_j.$$

The weights for $a_{t|T}$ depend on the position of j with respect to t . For example, for $j < t$ we have

$$\begin{aligned} w_j(a_{t|T}) &= w_j(a_{t|t-1}) + P_{t|t-1} w_j(r_{t-1}) \\ &= B_{t,j} K_j - P_{t|t-1} N_{t-1} B_{t,j} K_j \\ &= (I - P_{t|t-1} N_{t-1}) B_{t,j} K_j, \quad j = t-1, t-2, \dots, 1. \end{aligned}$$

It follows that the weights for the smoothed state vector $a_{t|T}$ are given by

$$w_j(a_{t|T}) = \begin{cases} (I - P_{t|t-1} N_{t-1}) w_j(a_{t|t-1}), & j = t-1, t-2, \dots, 1, \\ P_{t|t-1} C_t, & j = t, \\ P_{t|t-1} B'_{j,t-1} C_j, & j = t+1, t+2, \dots, T. \end{cases}$$

Note that $w_j(a_{t|t-1}) = 0$ for $j \geq t$.

Evaluation of $B'_{j,t-1}$ for $j = t+1, t+2, \dots, T$ can be done using the forwards recursion (40) since it follows that

$$B'_{j+1,t-1} = B'_{j,t-1} L'_j, \quad j = t+1, \dots, T,$$

and $B'_{t+1,t-1} = L'_t$. The matrix $B^*_{t,j}$ in (40) for $x_t = a_{t|T}$ is defined as $B^*_{t,j} = P_{t|t-1} B'_{j,t-1}$ for $j = t+1, \dots, T$.

Step 3

The weights for the smoothed estimator of the signal are given by $w_j(Z_t(a_{t|T})) = Z_t w_j(a_{t|T})$, for $j = 1, \dots, T$. This leads to the following simplification

$$Z_t(a_{t|T}) = \sum_{j=1}^T w_j(\hat{y}_{t|T}) y_j,$$

where

$$w_j(Z_t(a_{t|T})) = \begin{cases} G_t G'_t C'_t w_j(a_{t|t-1}), & j = t-1, t-2, \dots, 1, \\ I - G_t G'_t D_t, & j = t, \\ G_t G'_t K'_t B'_{j,t} C_j, & j = t+1, t+2, \dots, T. \end{cases}$$

Note that $Z_t(I - P_{t|t-1} N_{t-1}) = G_t G'_t C'_t$ and $Z_t P_{t|t-1} L'_t = G_t G'_t K_t$. Note also that $Z_t(a_{t|T})$ is the estimator of y_t when it is missing.

4.5 Computing weights for disturbance smoothing

The smoothed estimator of the disturbance vector ε_t conditional on all available observations, that is $e_{t|T}$, is given by (27) and is a linear estimator which can be expressed as a weighted average of all observations, that is

$$e_{t|T} = \sum_{j=1}^T w_j(e_{t|T}) y_j.$$

The weight matrices $w_j(e_{t|T})$ can be computed by applying the Kalman filter and smoother and storing K_j ($j = 1, \dots, T$) and C_j ($j = T, \dots, t$). The weight at time t is $w_t(e_{t|T}) = G'_t F_t^{-1} - J'_t N_t K_t$. The backwards recursion (33) with initialisation $B_{t,t-1} = -(G'_t C'_t + H'_t N_t L_t)$ is used to compute the weights $w_j(e_{t|T})$ for $j = t-1, t-2, \dots, 1$. The forwards recursion (40) with initialisation $B^*_{t,t+1} = J'_t$ is used to compute the weights for $j = t+1, t+2, \dots, T$.

Derivation

The smoothed estimator for $e_{t|T}$ can be expressed as

$$\begin{aligned} e_{t|T} &= G'_t F_t^{-1} v_t + J'_t r_t \\ &= G'_t F_t^{-1} y_t + J'_t r_t - G'_t F_t^{-1} Z_t a_{t|t-1}. \end{aligned}$$

It follows that

$$w_t(e_{t|T}) = G'_t F_t^{-1} + J'_t w_t(r_t),$$

where $w_t(r_t) = -N_t K_t$. Further,

$$w_j(e_{t|T}) = J'_t w_j(r_t) - G'_t F_t^{-1} Z_t w_j(a_{t|t-1}), \quad j = t-1, t-2, \dots, 1,$$

where $w_j(a_{t|t-1}) = B_{t,j} K_j$, $w_j(r_t) = -N_t B_{t+1,j} K_j$ and $B_{t+1,j} = L_t B_{t,j}$. Minor manipulation, using (39), gives

$$w_j(e_{t|T}) = -(G'_t C'_t + H'_t N_t L_t) w_j(a_{t|t-1}), \quad j = t-1, t-2, \dots, 1.$$

Therefore, we can use recursion (33) with $x_t = e_{t|T}$ and initialisation $B_{t,t-1} = -(G'_t C'_t + H'_t N_t L_t)$. Further,

$$w_j(e_{t|T}) = J'_t w_j(r_t), \quad j = t+1, t+2, \dots, T.$$

where $w_j(r_t) = B'_{j,t} C_j$. In this case we can use recursion (40) with $x_t = e_{t|T}$ and initialisation $B^*_{t,t+1} = J'_t$.

4.6 An overview

The same backwards recursion (33) can be used to compute the weights $w_j(x_t)$, with $j = t-1, t-2, \dots, 1$, for predictive filtering ($x_t = a_{t|t-1}$), contemporaneous filtering ($x_t = a_{t|t}$) and smoothing ($x_t = a_{t|T}$). Table 1 shows that only the initialisation of the recursion differs when computing weights for filtering and smoothing. For smoothing, the weights for $j = t+1, \dots, T$ are computed by the forwards recursion (40). The initialisations of both recursions (33) and (40) differ for different type of estimators. For example, the initialisations for computing the weights for the smoothed state and the smoothed disturbance vector ε_t are clearly different. However, these different initialisations lead to consistent results. We give a few examples:

- The initialisations of (33) for contemporaneous filtering and smoothing at $t = T$ are consistent with each other since $N_{T-1} = Z'_T F_T^{-1} Z_T = U_T$.
- The weights for the estimator of the observation y_t are equivalent to the weights for the state α_t when they are premultiplied by Z_t . For example, $\hat{y}_{t|T} = Z_t a_{t|T}$ and it can be shown that $Z_t - Z_t P_{t|t-1} N_{t-1} = G_t G'_t C'_t$.
- The relationship $G_t e_{t|T} = y_t - \hat{y}_{t|T}$ leads to similar relationships for the weights obtained by our algorithms.

4.7 Computing mean squared errors

The WK approach to signal extraction also provides formulae for the MSEs of the estimated signals. Similarly, the KFS computes the MSEs. The new algorithms, which are summarized in table 1, provide direct ways to compute the MSE of particular estimators at time t . It follows immediately that

$$\begin{aligned} \text{MSE}(a_{t|\tau}) &= B_{t,t-1} P_{t|t-1}, \\ \text{MSE}(\hat{y}_{t|\tau}) &= Z_t B_{t,t-1} P_{t|t-1} Z'_t = b_{t|t-1} P_{t|t-1} Z'_t, \end{aligned}$$

| x_t | operation | initialisation (33) | weight $w_t(x_t)$ | initialisation (40) |
|-------------------|------------------|----------------------------|---|---------------------------------------|
| $a_{t t-1}$ | filtering (pred) | I | 0 | 0 |
| $a_{t t}$ | filtering (cont) | $I - P_{t t-1}U_t$ | $P_{t t-1}Z_t'F_t^{-1}$ | 0 |
| $a_{t T}$ | smoothing | $I - P_{t t-1}N_{t-1}$ | $P_{t t-1}C_t$ | $P_{t t-1}L_t'$ |
| $\hat{y}_{t t-1}$ | filtering (pred) | Z_t | 0 | 0 |
| $\hat{y}_{t t}$ | filtering (cont) | $G_tG_t'F_t^{-1}Z_t$ | $I - G_tG_t'F_t^{-1}$ | 0 |
| $\hat{y}_{t T}$ | smoothing | $G_tG_t'C_t'$ | $Z_tP_{t t-1}C_t =$ $I - G_tG_t'D_t$ | $Z_tP_{t t-1}L_t' =$ $G_tG_t'K_t'$ |
| $e_{t T}$ | smoothing | $-(G_t'C_t' + H_t'N_tL_t)$ | $G_t'F_t^{-1} - J_t'N_tK_t$ | J_t' |
| $G_t e_{t T}$ | smoothing | $-G_tG_t'C_t'$ | $G_tG_t'D_t$ | $-G_tG_t'K_t'$ |
| $H_t e_{t T}$ | smoothing | $-H_tH_t'N_tL_t$ | $-H_tH_t'N_tK_t$ | H_tH_t' |

Table 1: Computing Weights

where $B_{t,t-1}$ associated with $\tau = t-1, t, T$ is given in table 1 under the header *initialisation (33)*. Note that $b_{t|t-1} = B_{t|t-1}Z_t$. For the disturbance estimators, it can be shown that

$$\text{MSE}(e_{t|T}) = w_t(e_{t|T})G_t + B_{t,t+1}^*N_tH_t, \quad (46)$$

where $B_{t,t+1}^* = J_t'$; see table 1 under the header *initialisation (40)*.

4.8 Computing weights when diffuse initial conditions apply

The general algorithms for computing weights are developed for state space models with a properly defined initial state vector as given in (20). However, when models contain nonstationary and deterministic components the initial specification of the state vector is adjusted to allow for diffuse elements; see Ansley and Kohn (1985, 1990), de Jong (1991) and Koopman (1997). This requires an augmented KFS for the initial period of the data set. The length of the initial period is usually equal to the number of diffuse elements in the state vector and is therefore relatively small. When using the approach of Ansley and Kohn (1990) and Koopman (1997), the KFS equations for the post-initial period remains unaltered. Therefore, the algorithms in this section for computing the weights assigned to observations after the initial period remain valid. The weights for the observations within the initial period can be computed straightforwardly by following similar arguments as in this section. However, notation and presentation of these results are somewhat elaborate and therefore we do not give them here. In any case, weights near the beginning of the sample are rarely required.

5 Univariate illustrations

5.1 Local level: small samples and deterministic components

For the local level model (4) and (5) with uncorrelated components, the simple zero-one method described in sub-section 4.1 can be used for the smoother, but not for the predictive filter if the sample size is too small to allow sufficient convergence to the steady-state; this is illustrated in table 2 where, for a sample of size eleven, the weights for $m_{6|6}$ (contemporaneous filtering) and $m_{6|11}$ (smoothing) are presented. The weights could, in principle, be obtained by the WK formulae adapted for finite samples as in Whittle (1983, chapter 7). However, these formulae cannot be used if q is zero since then the level is deterministic (so $\theta = -1$ in the reduced form). The zero-one method works for a smoother if the weights are taken in the natural order (because they are all the same, that is $1/T$).

| | | $q = 0.01 \quad \rho = 0$ | | | | $q = 4 \quad \rho = 1$ | | | |
|----|-----|---------------------------|----------|-----------|----------|------------------------|----------|-----------|----------|
| | | Filtering | | Smoothing | | Filtering | | Smoothing | |
| i | j | exact | zero-one | exact | zero-one | exact | zero-one | exact | zero-one |
| -5 | 1 | 0.157 | 0.0865 | 0.0865 | 0.0865 | 0.1667 | 1.0 | 0.091 | -1.0 |
| -4 | 2 | 0.159 | 0.098 | 0.0874 | 0.0874 | -0.5 | -1.1 | -0.273 | 1.0 |
| -3 | 3 | 0.162 | 0.113 | 0.0891 | 0.0891 | 0.8333 | 1.222 | 0.455 | -1.0 |
| -2 | 4 | 0.167 | 0.130 | 0.0918 | 0.0918 | -1.1667 | -1.375 | -0.636 | 1.0 |
| -1 | 5 | 0.173 | 0.152 | 0.0953 | 0.0953 | 1.5 | 1.571 | 0.818 | -1.0 |
| 0 | 6 | 0.1815 | 0.1815 | 0.0998 | 0.0998 | 0.1667 | 0.1667 | 1.0 | 1.0 |
| 1 | 7 | 0.0 | 0.0 | 0.0953 | 0.0953 | 0.0 | 0.0 | -0.818 | 1.0 |
| 2 | 8 | 0.0 | 0.0 | 0.0918 | 0.0918 | 0.0 | 0.0 | 0.636 | -1.0 |
| 3 | 9 | 0.0 | 0.0 | 0.0891 | 0.0891 | 0.0 | 0.0 | -0.455 | 1.0 |
| 4 | 10 | 0.0 | 0.0 | 0.0874 | 0.0874 | 0.0 | 0.0 | 0.273 | -1.0 |
| 5 | 11 | 0.0 | 0.0 | 0.0865 | 0.0865 | 0.0 | 0.0 | -0.091 | 1.0 |
| | sum | 1.0 | 0.761 | 1.0 | 1.0 | 1.0 | 1.485 | 1.0 | 1.0 |

Table 2: Weights for estimator of μ_6 of local level model ($T = 11$).

The weights for contemporaneous filtering are associated with $m_{6|6} = \sum_{i=-5}^0 w_i y_{6+i} = \sum_{j=1}^6 w_j(m_{6|6}) y_j$, and the weights for smoothing are associated with $m_{6|11} = \sum_{i=-5}^5 w_i y_{6+i} = \sum_{j=1}^{11} w_j(m_{6|11}) y_j$.

A less obvious case where WK cannot be applied is in a model with perfectly correlated components, $\rho = 1$ in (6), and $q = 4$, since then the reduced form is strictly noninvertible with $\theta = 1$. The weights computed by the exact algorithm are shown in table 2. The zero-one method gives the wrong answer because of the asymmetry in the W matrix in (31).

5.2 Seasonality

Consider a model consisting of trend, seasonal and irregular components, that is

$$y_t = \mu_t + \gamma_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2), \quad (47)$$

where μ_t is a random walk, (5), and γ_t is the stochastic seasonal component which can be either constructed using dummy variables or trigonometric terms; see Harvey (1989, chapter 2). The randomness of the seasonal component is governed by Gaussian disturbances with variance σ_ω^2 . A knowledge of the implied weighting patterns for smoothed components is useful for making comparisons between a model-based seasonal adjustment procedure based on (47) and procedures based on moving averages. The weighting patterns are studied in Riani (1998) using the zero-one method. Here we simply emphasize that the weight functions can all be obtained directly using the exact algorithm even if some components are deterministic (that is the corresponding disturbance variances are zero). If the seasonal component is deterministic, the zero-one method cannot give the filter weights. Even if, as often happens, the seasonal changes, but does so slowly, convergence to the steady-state is slow, rendering the zero-one method for filtering inappropriate in small samples. Note that the weights for the estimated irregular are computed using the algorithm of section 4.5.

In figure 1 we present the weighting patterns column-wise for the estimated level, seasonal and irregular, respectively. The row-wise panels reflect four different model specifications: in row 1, the

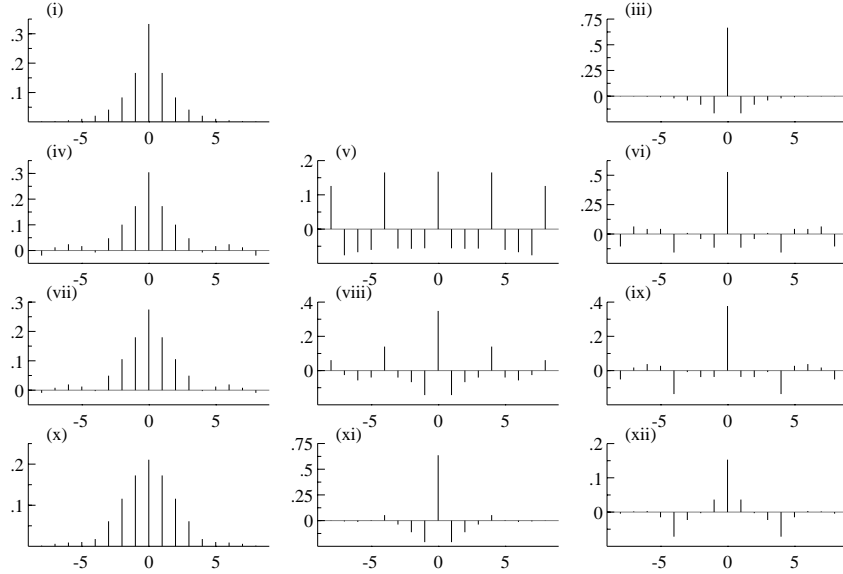


Figure 1: Weighting patterns for smoothed components

Weights are for model (47) and for middle of series ($T = 17$): the weights $w_j = w_{9+j}(m_{9|T})$ for $j = -8, -7, \dots, 8$ are associated with level, seasonal and irregular, respectively (column-wise); with $\sigma_\eta^2/\sigma_\varepsilon^2 = 0.5$ and (i)-(iii) no seasonal; (iv)-(vi) fixed seasonal ($\sigma_\omega^2 = 0$); (vii)-(ix) $\sigma_\omega^2/\sigma_\varepsilon^2 = 0.5$; (x)-(xii) $\sigma_\omega^2/\sigma_\varepsilon^2 = 5$.

seasonal component is removed and the signal-to-noise ratio for the level, $\sigma_\eta^2/\sigma_\varepsilon^2$, is 0.5 (as for the other cases); in row 2, the seasonal component is included but fixed ($\sigma_\omega^2 = 0$); in row 3, $\sigma_\omega^2/\sigma_\varepsilon^2 = 0.5$; and in row 4, $\sigma_\omega^2/\sigma_\varepsilon^2 = 5$. The weights are for the estimated components in the middle of the series of length $T = 17$. We note that the weights for the level and the irregular of a model without a seasonal are very similar compared to a model with a fixed seasonal. Even when the seasonal component is time-varying the level weights are not affected very much; the weights for the irregular however do change when $\sigma_\omega^2/\sigma_\varepsilon^2$ becomes larger. The end-of-sample effects are visible for the seasonal weights.

5.3 Missing observations

Missing observations can be estimated by smoothing. It may often be of interest to look at the weighting patterns implied for surrounding observations and to compute the MSE. By following the argument in Whittle (1983, pp.76-7), analytic expressions can be obtained but only in special cases. The zero-one method does not work. When the i th observation is missing, the i th column of W in (31) is a vector of zeroes and this is exactly the column which is obtained by computing the weights for $\hat{y}_{i|T}$ using the zero-one method. Thus the symmetry and band structure of W are lost. To illustrate our point we repeat the computations for table 2 but treat observations 3, 6, 8 and 9 as missing. The results are reported in table 3. Note that the correct weights still sum to unity.

5.4 Multi-step predictions

Multi-step forecasts can be obtained by extending the sample, applying the Kalman filter beyond time T and treating the observations after T as missing. The vector $a_{T+k|T}$ and its MSE matrix $P_{T+k|T}$, for $k = 1, 2, \dots$, are the multi-step forecast of the state vector and its mean square forecast error matrix, respectively. The weights assigned to observations at the end of the sample for multi-step predictions can be obtained using the exact method, simply by adding missing values at the end of the sample.

The local linear trend model, (14), serves to illustrate the kind of weighting patterns which can arise. In this case there is a straightforward relationship between the forecasting weights and the

| | | $q = 0.01 \quad \rho = 0$ | | $q = 4 \quad \rho = 1$ | |
|----|-----|---------------------------|-----------|------------------------|-----------|
| i | j | Filtering | Smoothing | Filtering | Smoothing |
| -5 | 1 | 0.2415 | 0.1374 | -0.05 | -0.04579 |
| -4 | 2 | 0.2439 | 0.1387 | 0.15 | 0.1374 |
| | | 0.0000 | 0.0000 | 0.0 | 0.0000 |
| -2 | 4 | 0.2536 | 0.1442 | -0.65 | -0.5953 |
| -1 | 5 | 0.2610 | 0.1485 | 1.55 | 1.420 |
| | | 0.0 | 0.0000 | 0.0 | 0.0 |
| 1 | 7 | 0.0 | 0.1498 | 0.0 | 0.09306 |
| | | 0.0 | 0.0000 | 0.0 | 0.0 |
| | | 0.0 | 0.0000 | 0.0 | 0.0 |
| 4 | 10 | 0.0 | 0.1414 | 0.0 | -0.01329 |
| 5 | 11 | 0.0 | 0.14 | 0.0 | 0.004431 |
| | sum | 1.0 | 1.0 | 1.0 | 1.0 |

Table 3: Exact weights for estimator of μ_6 of local level model ($T = 11$).

The weights have the same interpretation as for table 2. Observations 3, 6, 8 and 9 are missing. The zero-one method gives all weights equal to zero.

filtering weights since

$$\hat{y}_{T+k|T} = \hat{\mu}_{T+k|T} = \hat{\mu}_{T|T} + k\hat{\beta}_{T|T}, \quad k = 1, 2, \dots \quad (48)$$

and so

$$w_j(\hat{y}_{T+k|T}) = w_j(\hat{\mu}_{T|T}) + kw_j(\hat{\beta}_{T|T}), \quad k = 1, 2, \dots$$

Thus as k becomes larger the weights for the forecasts depend more and more on the weights of the slope estimator $\hat{\beta}_{T|T}$. This can be seen in figure 2 with $\sigma_\varepsilon^2 = 1$, $\sigma_\eta^2 = .04$ and $\sigma_\zeta^2 = .01$.

5.5 Irregularly spaced observations

In this sub-section and the next sub-section, we consider 133 observations of acceleration against time (measured in milliseconds) for a simulated motorcycle accident which is originally analysed by Silverman (1985). The observations are not a time series, and they are not equally spaced over time and at certain time points multiple observations are recorded; see figure 3. These data are often used as an illustration for nonparametric regression using cubic splines. The connection between cubic splines and the state space framework has been known for many years; see Wecker and Ansley (1983). Given the data set y_1, \dots, y_T for which the t -th observation is recorded at time τ_t , the time series model for a cubic spline is (14) with

$$\varepsilon_t \sim \text{WN}(0, \sigma_\varepsilon^2), \quad \eta_t \sim \text{WN}(0, \sigma_\zeta^2 \delta_t^3 / 3), \quad \zeta_t \sim \text{WN}(0, \sigma_\zeta^2 \delta_t), \quad \text{E}(\eta_t \zeta_t) = \sigma_\zeta^2 \delta_t^2 / 2,$$

ε_t is uncorrelated with both η_t and ζ_t and $\delta_t = \tau_{t+1} - \tau_t$; see Harvey (1989, sections 9.1.2 and 9.2.1) and Koopman *et al* (1999) for further details. The weighting pattern for the estimated signal μ_t can be interpreted as the equivalent kernel function and can be compared with kernel functions used in nonparametric regression.

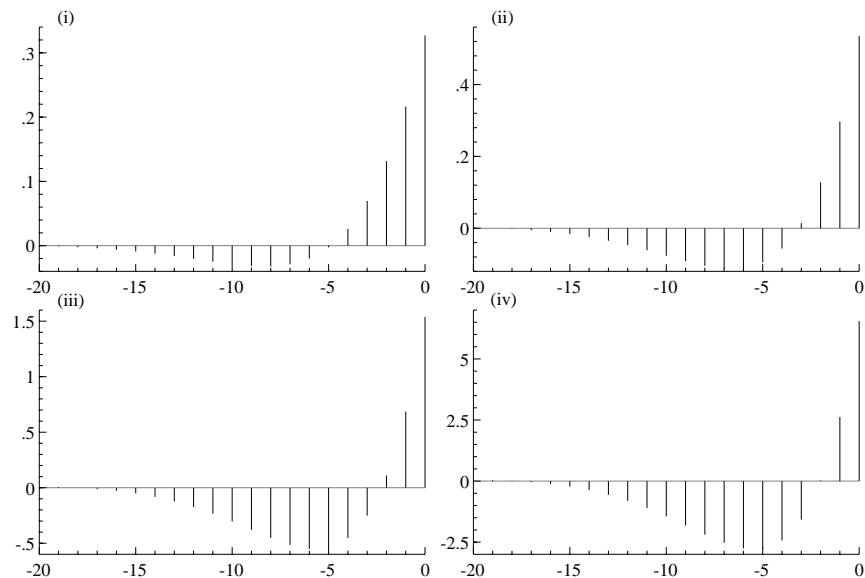


Figure 2: Weights for forecasting

The weights $w_j = w_{T+j}(\hat{y}_{T+k|T})$ are for local linear trend model with $\sigma_\varepsilon^2 = 1$, $\sigma_\eta^2 = .04$ and $\sigma_\zeta^2 = .01$: (i) $k = 1$, (ii) $k = 6$, (iii) $k = 30$, (iv) $k = 150$.

The smoothing parameter, $q_\zeta = \sigma_\zeta^2 / \sigma_\varepsilon^2$, can be estimated by maximum likelihood using the Kalman filter and is given by $\hat{q}_\zeta = .0275$. Figure 3 (i) presents the cubic spline (solid line). The KFS is used to compute the estimates of μ_{τ_t} but it also computes the root mean square error (RMSE) of these estimates. The error bounds based on the RMSEs for the fitted spline are also given in figure 3 (i) (dotted lines).

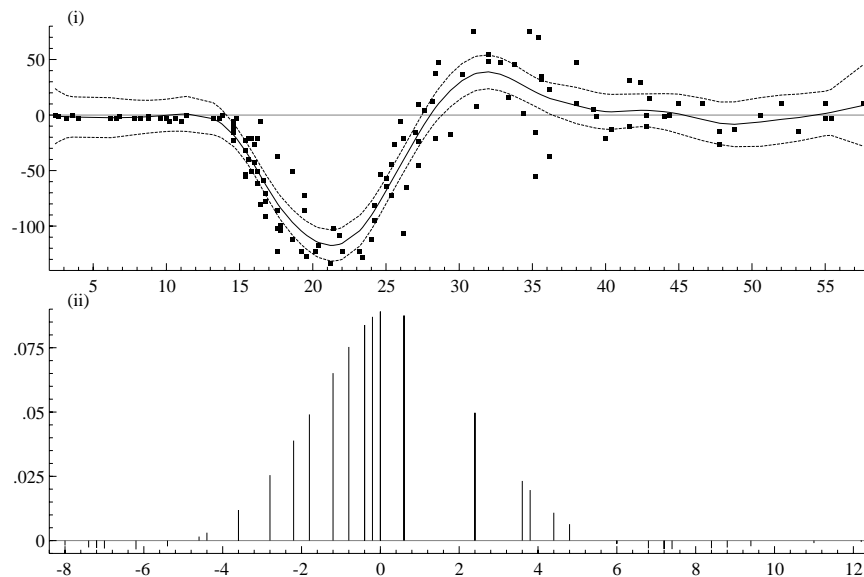


Figure 3: Motorcycle acceleration data analysed by a cubic spline

(i) observations against time with spline and 95% confidence intervals and (ii) the weights for the spline at time $\tau_{105} = 35.6$ against time distance $\tau_{105} - \tau_j$.

Figure 3 (ii) shows the weights assigned to observations at adjacent time points for computing $\tilde{\mu}_{\tau_t}$, with $\tau_t = 35.6$ corresponding to the 105th observation. The gaps are caused by time periods for

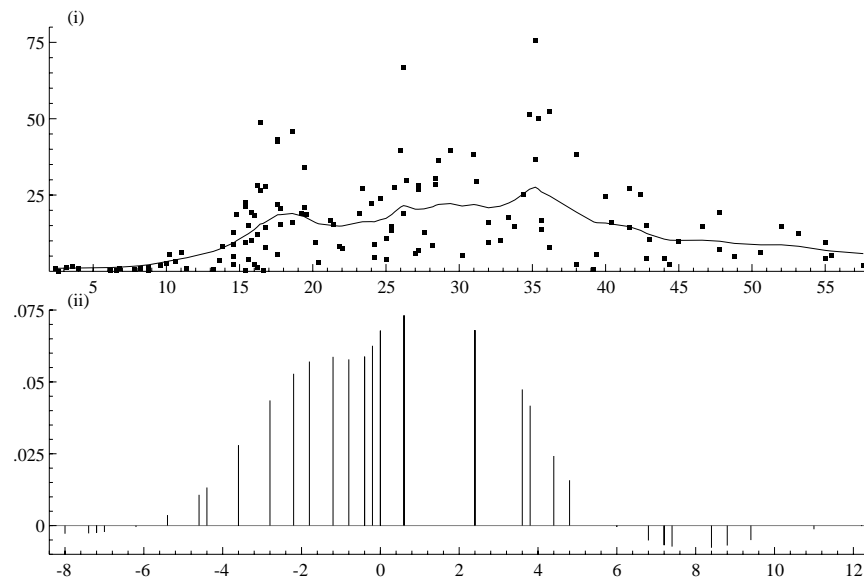


Figure 4: Motorcycle acceleration data analysed by a cubic spline corrected for heteroscedasticity

(i) absolute values of smoothed irregular and h_t^* , (ii) the weights for the local level signal at time $\tau_{105} = 35.6$ against time distance $\tau_{105} - \tau_j$.

which no observations are recorded. The weighting pattern in figure 3 (ii) is *not* symmetric. This is in contrast to the nonparametric approach where the weighting pattern is symmetric in that observations which are at the same ‘distance’ from the time point τ_t receive the same weight; see, for example, Green and Silverman (1994). The reason the optimal weights, obtained from the model, are not symmetric is that the number of data points observed around a particular observation is taken into account; if this number is large, the observation receives less weight.

5.6 Heteroscedasticity

Figure 4 (i) shows the standardised smoothed estimates of the irregular disturbance from the analysis of the previous sub-section. There appears to be heteroscedasticity. This may be corrected for by the simple device of fitting a local level signal through the absolute values of the smoothed irregular. The variance σ_ε^2 of the original cubic spline model is then replaced by $\sigma_\varepsilon^2 h_t^{*2}$ where h_t^* is the smoothed estimate of the local level signal but scaled such that $h_1^* = 1$; see Harvey and Koopman (1999) for more details. The absolute values of the smoothed irregular and the h_t^* ’s are presented in figure 4 (i). Figure 4 (ii) presents the weighting pattern at time $\tau_{105} = 35.6$ and it can be seen that the adjusted spline is locally more smooth than before because the weights decline at a slower rate. Since the variance of the irregular is changing over time, we need to apply the exact method of computing the weights.

5.7 Bayesian estimation

The weights we have considered so far apply to estimators of the state vector when the parameters of the model are known. However, these parameters, such as the variances σ_ε^2 and σ_η^2 in the local level model, are usually unknown. For Gaussian models they can be estimated by maximum likelihood, using the Kalman filter to compute the likelihood function. The weights assigned to observations for filtering and smoothing can then be computed using the algorithms of section 4 with parameters replaced by estimates.

In a Bayesian approach the parameters are treated as random variables with some prior distribution. To obtain the posterior distribution of the parameter and state vector we may employ Markov chain Monte Carlo (MCMC) techniques; see, for example, Shephard and Pitt (1997). In the case of a linear Gaussian model the MCMC techniques are relatively easy to implement; see Frühwirth-Schnatter (1994). In this sub-section, we show how our weighting algorithm can be used to calculate weights for the posterior mean of the state vector.

Consider the local level model (4) and (5) where the disturbances are Gaussian and mutually independent. To compute moment characteristics of the posterior densities $p(\sigma_\varepsilon^2|y_1, \dots, y_T)$ and $p(\sigma_\eta^2|y_1, \dots, y_T)$ we can use the following MCMC algorithm:

1. Give some prior values for σ_ε^2 and σ_η^2 .
2. Sample from the multivariate Gaussian distribution of $\mu|y, \sigma_\varepsilon^2, \sigma_\eta^2$, where $\mu = (\mu_1, \dots, \mu_T)'$ and $y = (y_1, \dots, y_T)'$, using the simulation smoother of de Jong and Shephard (1995) as implemented in *SsfPack* 2.2.
3. Sample from the distribution of $\sigma_\varepsilon^2|y, \mu$ and $\sigma_\eta^2|y, \mu$.
4. Goto 2.

It can be shown that after some “burning-in” iterations, the draws generated in step 3 can be regarded as draws from the posterior distributions of $\sigma_\varepsilon^2|y$ and $\sigma_\eta^2|y$, respectively.

Simulation samples from the distributions of $\sigma_\varepsilon^2|y, \mu$ and $\sigma_\eta^2|y, \mu$ for a given y , can be obtained as follows. First, the appropriate prior density for the variance is given by

$$\sigma_a^2 \sim \mathcal{IG}\left(\frac{c_a}{2}, \frac{s_a}{2}\right),$$

where \mathcal{IG} is the inverse gamma distribution with shape parameter c_a and scale parameter s_a and a refers to disturbance ε_t or η_t . The prior mean and variance of σ_a^2 are given by

$$\frac{s_a}{c_a - 2}, \quad \frac{2s_a}{(c_a - 2)^2(c_a - 4)},$$

respectively. Then, the corresponding posteriors are

$$\sigma_\varepsilon^2|y, \mu \sim \mathcal{IG}\left(\frac{c_\varepsilon + n}{2}, \frac{s_\varepsilon + \sum \varepsilon_t^2}{2}\right), \quad \sigma_\eta^2|y, \mu \sim \mathcal{IG}\left(\frac{c_\eta + n}{2}, \frac{s_\eta + \sum \eta_t^2}{2}\right), \quad (49)$$

where

$$\begin{aligned} \eta_t &= \mu_{t+1} - \mu_t \sim \text{NID}(0, \sigma_\eta^2), \\ \varepsilon_t &= y_t - \mu_t \sim \text{NID}(0, \sigma_\varepsilon^2). \end{aligned}$$

It is easy to sample from these distributions as shown in the Ox program `bayesw.ox` of the appendix; further computing details are given by Koopman *et al* (1999). The posterior means for σ_ε^2 and σ_η^2 are simply obtained by taking the averages of the generated samples in step 3 excluding the samples from the “burning-in” period. In the same way the posterior mean of the level μ_t is obtained using the samples generated in step 2.

The weights of the smoothed posterior mean of the level can simply be obtained by averaging the weight patterns of the smoothed level implied by a model with parameter values obtained from step 3 of the MCMC algorithm. This approach can be straightforwardly generalised to more complicated models.

To illustrate our method of computing weights for the local level model, we use the Nile data analysed by Balke (1993) and Harvey, Koopman and Penzer (1998). The maximum likelihood estimates of

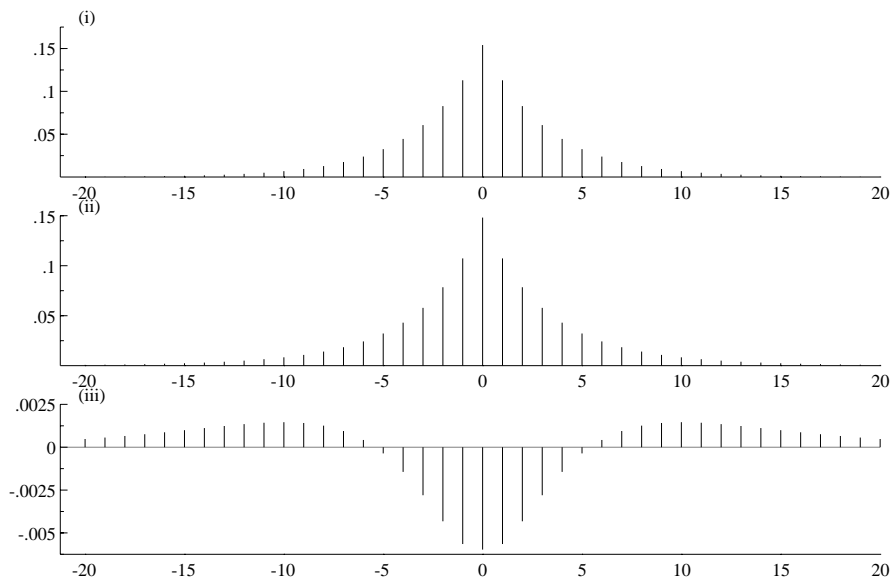


Figure 5: Weighting patterns for local level model

(i) classical estimator of level; (ii) Bayesian smoothed posterior mean; (iii) difference.

the variances, calculated using STAMP 5.0 of Koopman *et al* (1995), are $\hat{\sigma}_\varepsilon^2 = 15099$ and $\hat{\sigma}_\eta^2 = 1469.2$ and the consequent “classical” weighting pattern for the smoothed level in the middle of the series is shown in figure 5. The Bayesian weights are computed using the MCMC algorithm with 10,000 iterations and 1,000 “burning-in” iterations. The posterior means of the variances are $\bar{\sigma}_\varepsilon^2 = 15165$ and $\bar{\sigma}_\eta^2 = 1407.4$. The differences between the classical and Bayesian weights are also presented in figure 5; the tails of the Bayesian weighting pattern are heavier because the Bayesian approach takes into account the uncertainty of the variances. The Ox program `wbayes.ox` is used to generate these results and the listing of the program is given in the appendix.

6 Multivariate models

Suppose we have a multivariate model for N time series and we wish to investigate how the observations in all N series are weighted in forming an estimator of a component in the first series. This is of considerable interest in understanding how the interactions in a dynamic model work and whether they result in weighting patterns which have a useful interpretation. We illustrate this by looking at multivariate unobserved components models which generalise the local level and local linear trend models introduced in section 2.

6.1 Filtering and signal extraction using related series

The multivariate local level model can be written in the contemporaneous state space form

$$\begin{aligned} y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\sim \text{WN}(0, \Sigma_\varepsilon), & t &= 1, \dots, T, \\ \mu_t &= \mu_{t-1} + \eta_t, & \eta_t &\sim \text{WN}(0, \Sigma_\eta), \end{aligned} \quad (50)$$

where WN now denotes multivariate white noise. For simplicity we concentrate on a bivariate model for the series of interest, the target series, and a single related series. The aim is to examine how the observations in the two series are weighted when the level in the target series is estimated. This provides insight into what exactly the model does. In general the weights cannot be deduced analytically for the reasons given in sub-section 2.3. However, there are two special cases – homogeneity

and common trends – where some theoretical insight is possible and we examine these cases before presenting weights computed for a range of specifications by our exact numerical algorithm.

If the model is homogeneous, that is $\Sigma_\eta = q\Sigma_\varepsilon$, where q is a scalar, then there is no gain from using a second series (except in the estimation of q) and so the weighting pattern is as in the univariate case; see Harvey (1989, chapter 8). Thus the filtered estimator of each level component is an EWMA of its own past observations only. Similarly the smoother shows a two-sided exponential decline as in figure 1 (i).

If the rank of the covariance matrix Σ_η in (50) is $K < N$, an appropriate ordering of the series enables the model to be written in the common trends (or levels) form

$$\begin{aligned} y_{1t} &= \mu_t^\dagger + \varepsilon_{1t} \\ y_{2t} &= \Pi\mu_t^\dagger + \bar{\mu} + \varepsilon_{2t} \end{aligned} \quad (51)$$

where y_{1t} is a $K \times 1$ vector, y_{2t} is an $r \times 1$ vector with $r = N - K$, Π is an $r \times K$ matrix of coefficients and the $K \times 1$ vector μ_t^\dagger follows a multivariate random walk

$$\mu_t^\dagger = \mu_{t-1}^\dagger + \eta_t^\dagger, \quad \eta_t^\dagger \sim \text{WN}(0, \Sigma_\eta^\dagger), \quad (52)$$

where η_t^\dagger is a $K \times 1$ vector and Σ_η^\dagger is a $K \times K$ positive definite matrix. The $r \times 1$ vector $\bar{\mu}$ is deterministic. In terms of (50), $\Sigma_\eta = \Pi^\dagger \Sigma_\eta^\dagger \Pi$, where $\Pi^\dagger = (I, \Pi')'$.

For a bivariate model we can write

$$\Sigma_\eta = \begin{pmatrix} \sigma_{1\eta}^2 & \rho_\eta \sigma_{1\eta} \sigma_{2\eta} \\ \rho_\eta \sigma_{1\eta} \sigma_{2\eta} & \sigma_{2\eta}^2 \end{pmatrix} \quad (53)$$

where ρ_η is the correlation. If $\rho_\eta = \pm 1$, there is a single common trend and so

$$\begin{aligned} y_{1t} &= \mu_{1t} + \varepsilon_{1t} = \mu_t^\dagger + \varepsilon_{1t}, & t = 1, \dots, T, \\ y_{2t} &= \mu_{2t} + \varepsilon_{2t} = \pi\mu_t^\dagger + \bar{\mu} + \varepsilon_{2t} \end{aligned} \quad (54)$$

where $\pi = \text{sgn}(\rho_\eta)\sigma_{2\eta}/\sigma_{1\eta}$. Some insight into weighting patterns can be achieved by transforming this model to

$$\begin{aligned} (y_{1t} + y_{2t}/\pi)/2 &= \mu_t^\dagger + \bar{\mu}/2\pi + (\varepsilon_{1t} + \varepsilon_{2t}/\pi)/2 \\ -\pi y_{1t} + y_{2t} &= \bar{\mu} + (-\pi\varepsilon_{1t} + \varepsilon_{2t}) \end{aligned}$$

The Jacobian of the transformation is unity. The second equation does not contain the common trend while in the first equation $\bar{\mu}$ can be absorbed into the initial conditions for the trend. If we make the assumption that $\text{Var}(\varepsilon_{2t})/\text{Var}(\varepsilon_{1t}) = \pi^2 (= \sigma_{2\eta}^2/\sigma_{1\eta}^2)$ the irregular disturbances in the two transformed equations are uncorrelated with each other whatever the correlation, ρ_ε , between the original irregulars and so the weights for estimating the stochastic trend in the first equation will be as for a univariate local level with signal-noise ratio, q , equal to $(1 + \rho_\varepsilon)/2$ times the signal-noise ratio in the first untransformed equation. If $\bar{\mu}$ is known, this is the pattern which applies to μ_t^\dagger , and hence to μ_{1t} , and so, unless $\rho_\varepsilon = 1$, the weights associated with $(y_{1t} + y_{2t}/\pi)/2$ will show a more rapid exponential decline than the weights for y_{1t} in the univariate model. The weights on y_{1t} sum to one-half while those on y_{2t} sum to $1/2\pi$. If $\bar{\mu}$ is not known, its estimator is obtained from the second transformed equation as $\bar{y}_2 - \pi\bar{y}_1$. The weights for the estimator of $\bar{\mu}/2\pi$ need to be subtracted from the weighting pattern for μ_{1t} obtained when $\bar{\mu}$ is known. This term contributes $1/2T$ to each observation in the first series and $-1/2T\pi$ to each observation in the second series. Thus, for extracting μ_{1t} , the weights on the first series now sum to one while those on the second sum to zero. The converse is

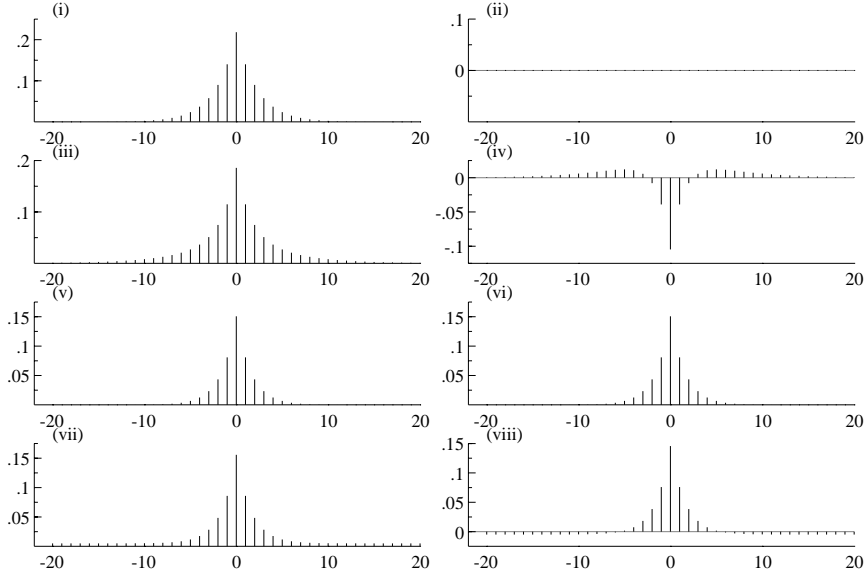


Figure 6: Weighting patterns for estimated level of first (target) series

Multivariate local level model with $\text{Var}(\varepsilon_t) = I_2$ and $T=100$: (i) weights for y_{1t} in case (a), (ii) corresponding weights for y_{2t} , (iii) weights for y_{1t} in case (b), (iv) corresponding weights for y_{2t} , (v) weights for y_{1t} in case (c) with known constant, (vi) corresponding weights for y_{2t} , (vii) weights for y_{1t} in case (c) with unknown constant, (viii) corresponding weights for y_{2t} .

true for μ_{2t} , thus illustrating that the weighting matrix sums to the identity matrix even if Σ_η is not of full rank.

We now consider three different specifications, (a), (b) and (c). In all cases $\Sigma_\varepsilon = I_2$ while

$$\Sigma_\eta^{(a)} = \begin{pmatrix} .2 & 0 \\ 0 & .2 \end{pmatrix}, \quad \Sigma_\eta^{(b)} = \begin{pmatrix} .2 & -.2 \\ -.2 & .3 \end{pmatrix}, \quad \Sigma_\eta^{(c)} = \begin{pmatrix} .2 & .2 \\ .2 & .2 \end{pmatrix}.$$

Figure 6 presents the weighting patterns used to construct smoothed estimators of the level in the first series in terms of its own values and those of the related series. The sample size is 100. The results for $\Sigma_\eta^{(a)}$ are in panels (i) and (ii), while for $\Sigma_\eta^{(b)}$ is in panels (iii) and (iv). Because the level disturbances in (b) are negatively correlated, the weights from the second series are negative. Case (c) has a common level as matrix $\Sigma_\eta^{(c)}$ is of rank one; π is equal to one. Weights when $\bar{\mu}$ in (54) is known are shown in panels (v) and (vi). The analysis of the previous paragraph is confirmed in that the weights on the target series decline faster than in the corresponding univariate model - these weights are as in panel (i). Panels (vii) and (viii) show that when $\bar{\mu}$ is estimated $1/2T$ is added to the weights for the first series while the same quantity is subtracted from the weights for the second series. Note that the presence of an unknown deterministic component in this case makes the use of the zero-one method problematic.

6.2 Missing observations

When a series contains missing observations, using a related series can provide better estimates of the missing values than can the target series alone. In such cases, it is of interest to compute the weights assigned to observations in both series and this can be done by the weighting algorithm. In figure 7 the same weighting patterns are presented as in figure 6 but with a missing observation in the target series at the time point for which the weights are produced. The patterns are similar to those in figure 6, except that the largest weight is now on the observation in the related series corresponding to the missing observation in the target series. With the exception of panels (vii) and (viii), the weights in the left-hand panels sum to one and those in the right-hand panels sum to zero.

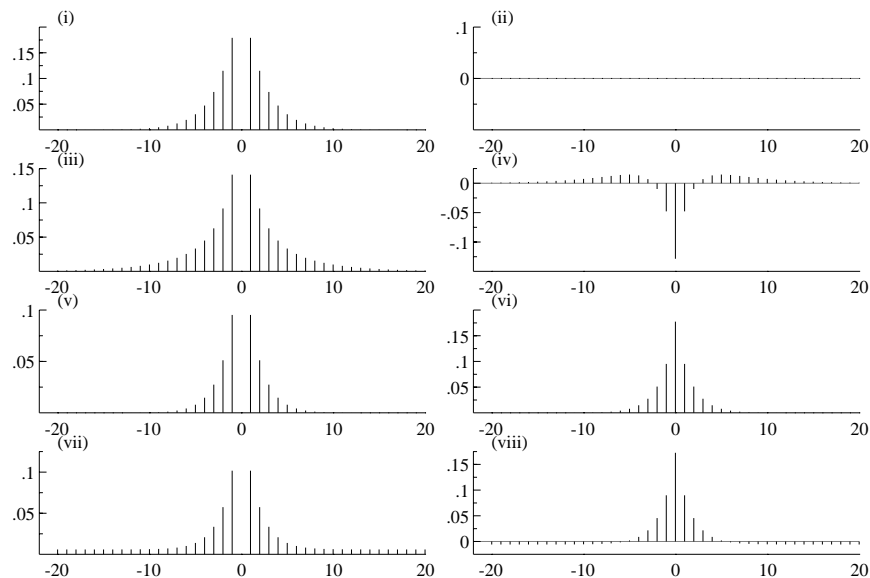


Figure 7: Weighting patterns for estimated level of first (target) series

Weighting patterns as in figure 6 but with observation in the target series y_{1t} missing.

A related problem is using a monthly series to estimate the underlying trend of a quarterly series. The state space form can be modified to handle flows as well as stocks. In such cases the exact method of computing weights is the only practical method.

6.3 Vector error correction representation

The presence of common trends implies co-integration. In the local level model (51), there exist $r = N - K$ co-integrating vectors. Let A be an $r \times N$ matrix partitioned as $A = (A_1, A_2)$. Then pre-multiplying (51) by A gives

$$Ay_t = A_1 y_{1t} + A_2 y_{2t} = (A_1 + A_2 \Pi) \mu_t^\dagger + A_2 \bar{\mu} + A_1 \varepsilon_{1t} + A_2 \varepsilon_{2t}, \quad t = 1, \dots, T. \quad (55)$$

The r series in Ay_t are stationary, and hence A consists of co-integrating vectors, if $A_1 + A_2 \Pi = 0$.

Pre-multiplying the observation vector in the bivariate model (54) by the vector $A = (-\pi, 1)$ gives

$$y_{2t} = \pi y_{1t} + \bar{\mu} + \varepsilon_t, \quad (56)$$

where $\varepsilon_t = \varepsilon_{2t} - \pi \varepsilon_{1t}$. Since the linear combination $y_{2t} - \pi y_{1t}$ is stationary, the series are co-integrated.

Econometricians often handle co-integrated systems by a vector error-correction model (VECM) and so it is interesting to compute the coefficients in the VECM representation of the common trends model. The VECM can be written as

$$\Delta y_t = \Phi^* y_{t-1} + \delta + \sum_{j=1}^{\infty} \Phi_j^* \Delta y_{t-j} + v_t, \quad (57)$$

where the relationship between the $N \times N$ parameter matrices and those in the VAR model of (35) is

$$\Phi^* = \sum_{k=1}^{\infty} \Phi_k - I, \quad \Phi_j^* = - \sum_{k=j+1}^{\infty} \Phi_k, \quad j = 1, 2, \dots \quad (58)$$

The rank of Φ^* is r . It is normally expressed as $\Phi^* = \Gamma A$, where A is the $r \times N$ matrix of co-integrating vectors and Γ is a $N \times r$ matrix of coefficients. When there is no co-integration, $\Phi^* = 0$.

If we were to have a homogeneous model, (50) with $\Sigma_\eta = q\Sigma_\varepsilon$, then there is no co-integration, so $\Phi^* = 0$, and the model is just a VAR in first differences. The parameter matrices are $\Phi_j^* = (-\theta)^j I$, $j = 1, 2, \dots$, where θ is obtained from (9). If q is small, θ is close to minus one and there is a slow decline in the Φ_j^* 's. Obtaining more general results is difficult. This provides a motivation for the use of our exact predictive filtering weighting algorithm. As already noted, the zero-one method cannot be used when a steady-state is obtained too slowly or when co-integration is present.

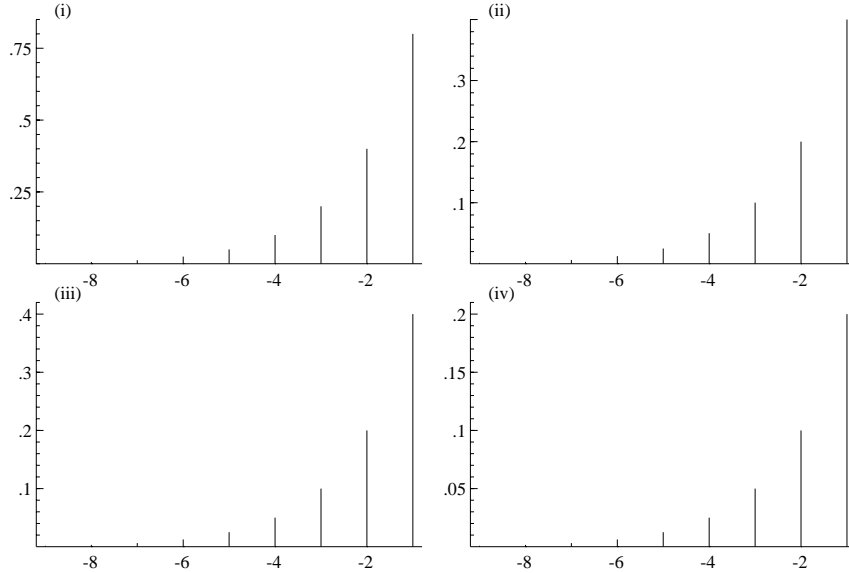


Figure 8: VAR representation weights for multivariate local level model

Model with $\text{Var}(\eta_{1t}) = .4$, $\text{Var}(\eta_{2t}) = .1$ and $\text{Cov}(\eta_{1t}, \eta_{2t}) = .2$: elements of Φ_j^* for $j = 1, \dots, 9$, (i) (1,1); (ii) (1,2); (iii) (2,1); (iv) (2,2).

Given a common trends model, (51), the weighting algorithm is used to compute the coefficient matrices in the corresponding VAR. The vector $\bar{\mu}$ is set to zero. The VECM matrices, Φ^* and the Φ_j^* 's, are then obtained from (58). As is well known, the matrix A is not unique for $r > 1$ but it can be set to $(-\Pi, I)$. The Γ matrix is then computed so as to satisfy $\Phi^* = \Gamma A$, while $\delta = -\Phi^*(0, \bar{\mu})'$. If Σ_η is dominated by Σ_ε , a slow decline in the Φ_j^* 's can be expected. This has implications for using a VECM to model a system for which a common trends model is appropriate.

Consider the bivariate local level model (51) with $\Sigma_\varepsilon = I_2$ and $\Sigma_\eta = 0.1(2, 1)(2, 1)'$. The implied co-integrating vector is equal to $(-.5, 1)$ and we find

$$\Phi^* = \begin{pmatrix} -.2 & .4 \\ .4 & -.8 \end{pmatrix} = \begin{pmatrix} .4 \\ -.8 \end{pmatrix} \begin{pmatrix} -.5 & 1 \end{pmatrix}.$$

Figure 8 shows the weights attached to $\Delta y_{1,t-j}$ and $\Delta y_{2,t-j}$, for $j = 1, 2, \dots$, for the two series. It is interesting that the weights still decline exponentially as in the simple homogeneous case analysed earlier. The Ox program `vecm.ox` is used to generate these results and the listing of the program is given in the appendix.

6.4 Using a related series to estimate level and change

By constructing a bivariate local linear trend model, generalising (14), it may be possible to use a related series to obtain an improved estimator of the underlying change or slope in a target series. It was shown in sub-section 2.2 how the WK formula could be used in the univariate case to obtain weights for the smoothed estimator of slope. Weights for the filtered estimator could be similarly

obtained. In fact it is the filtered estimator which is usually required. Harvey and Chung (1999) give an example in connection with UK unemployment, where the series of interest, based on an ILO definition, is obtained from a survey which is subject to sampling error. The related series, on monthly claimant counts, is exact.

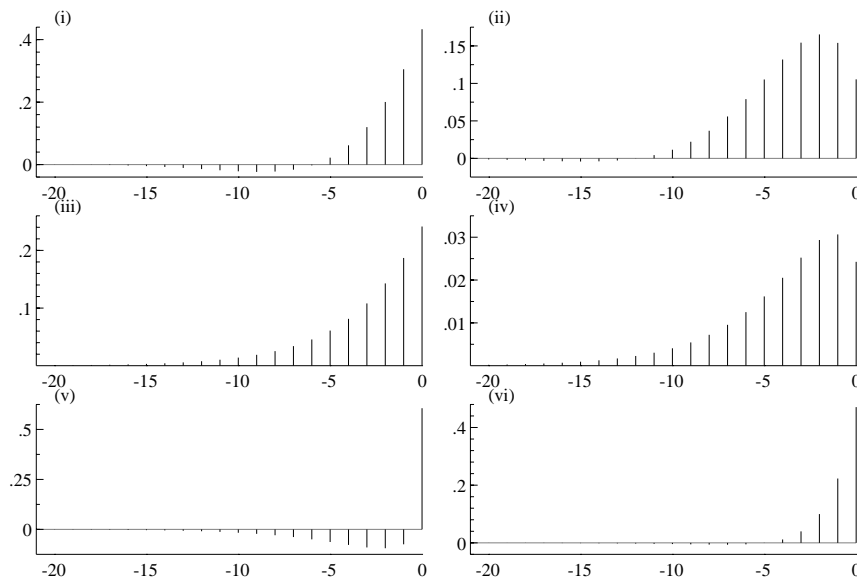


Figure 9: Weighting patterns for the Harvey and Chung model

(i) univariate $\hat{\mu}_{t|t}$ in terms of y_{t+j} ; (ii) univariate $\hat{\beta}_{t|t}$ in terms of Δy_{t+j} ; (iii) multivariate $\hat{\mu}_{1,t|t}$ in terms of $y_{1,t+j}$; (iv) multivariate $\hat{\beta}_{1,t|t}$ in terms of $\Delta y_{1,t+j}$; (v) multivariate $\hat{\mu}_{1,t|t}$ in terms of $y_{2,t+j}$; (vi) multivariate $\hat{\beta}_{1,t|t}$ in terms of $\Delta y_{2,t+j}$.

Parameter values close to those reported in Harvey and Chung (1999) are as follows

$$\Sigma_{\varepsilon} = \begin{pmatrix} 2500 & 0 \\ 0 & 0 \end{pmatrix}, \quad \Sigma_{\eta} = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}, \quad \Sigma_{\zeta} = \begin{pmatrix} 49 & 49 \\ 49 & 49 \end{pmatrix}.$$

The first series is the ILO series with an irregular component taken to represent the survey error, the variance of which is 2500. The level and slope disturbance variances are the same and the slope disturbances are perfectly correlated, so the model is co-integrated of order (2,1). Harvey and Chung argue further that the slopes in the two series are proportional so the deterministic slope component, analogous to $\bar{\mu}$ in the local level model, is set to zero. The zero-one method could not, therefore, be used. The weights for the current estimator of both the level and the slope of the first series are presented in the first two panels of figure 9. The weights implied by the bivariate model are presented in graphs (iii) and (iv) for the first series and in graphs (v) and (vi) for the second series. Because the second series has no sampling error it actually plays a dominant role in estimating the level and slope in the first series. Furthermore, in estimating the slope, relatively more weight is assigned to the most recent observations, so the bivariate estimator is faster to respond.

Actually the model in Harvey and Chung (1999) is more complex than the one above since the survey uses a rotating sample which leads to a serial correlation. Furthermore the observations are constructed from the survey by averaging over a three month period. Harvey and Chung (1999) build an unobserved components model to capture these features and estimate it using *SsfPack*. Adding the subroutine to calculate the weights requires only one line of extra code.

Acknowledgements

During the main part of this research, the first author was a Research Fellow of the Royal Netherlands Academy of Arts and Sciences and its financial support is gratefully acknowledged. Further, we would like to thank Neil Shephard and Marius Ooms for discussions and suggestions. Neil Shephard suggested the idea of computing the Bayesian weights.

Appendix

Some *Ox* code listings are given below. The software and the documentation of the *SsfPack 2.3* package can be freely downloaded from the Internet at www.ssfpack.com; for further details, see Koopman, Shephard and Doornik (1999). All *Ox* code used in this paper can be downloaded from <http://www.econ.vu.nl/koopman/weights/>.

Code for table 2

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack.h>

main()
{
    decl ct, loc, my, mphi, momega, q, r, mw, mw_zo;

    // data (zeroes with unity on position where weights are required)
    ct = 11; loc = 5;
    my = zeros(1,ct); my[0][loc] = 1;

    // local level model
    mphi = <1;1>; momega = unit(2);
    q = 0.01; r = 0.0;
    momega[0][0] = q; momega[1][0] = momega[0][1] = sqrt(q) * r;

    // exact method
    mw = SsfWeights(ST_SMO, my, mphi, momega);

    // zero one method
    SsfMomentEst(ST_SMO, &mw_zo, my, mphi, momega);

    // table output
    decl index = range(-5, 5);
    println(index'~mw'~reverser(mw_zo[:1][:])');
    println(sumc(index'~mw'~reverser(mw_zo[:1][:])'));
}
```

Comments: (i) by replacing SMO with FIL the program computes the weights for filtering rather than for smoothing (note that in *Ox* indices for arrays start counting at 0, not 1); (ii) the functions *SsfMomentEst* and *SsfWeights* are from *SsfPack*.

Code for computing Bayesian weights

```
#include <oxstd.h>
#include <oxprob.h>
#include <packages/ssfpack/ssfpack.h>
```

```

main()
{
    // data and model
    decl myt, mzt, loc = 50, mphi, momega, msigma;
    myt = loadmat("Nile.dat");
    mzt = zeros(myt); mzt[0][loc] = 1;
    GetSsfStsm(<CMP_LEVEL, 1.0, 0, 0; CMP_IRREG, 1.0, 0, 0>, &mphi, &momega, &msigma);

    // prior distribution: inverse gamma
    decl s_eta, s_xi, c_eta, c_xi;
    s_eta = 5000; s_xi = 50000;
    c_eta = c_xi = 2.5 + (0.5 * columns(myt));

    // posterior analysis by MCMC
    decl i, crep=10000, cinit=1000, mbw, mpsi, md;
    mbw = zeros(2, columns(myt));
    for (i = 0, mpsi = zeros(2, crep); i < crep; i++)
    {
        if (i >= cinit) mbw += SsfWeights(ST_SMO, mzt, mphi, momega, msigma);
        md = SsfCondDens(DS_SIM, myt, mphi, momega, msigma);
        md = md * md';
        mpsi[0][i] = 1.0 / rangamma(1,1, c_eta, (s_eta + md[0][0])/2);
        mpsi[1][i] = 1.0 / rangamma(1,1, c_xi, (s_xi + md[1][1])/2);
        momega = diag(mpsi[][i]);
    }
    mpsi = mpsi[][cinit:]; // drop first cinit draws
    print("\nPosterior analysis\n");
    print("%r", {"var_eta", "var_xi"}, "%c", {"mean", "st.dev."}, "%15.3f",
        meanr(mpsi)~sqrt(varr(mpsi)));
    mbw /= (crep-cinit); // bayesian weights

    // classical analysis
    momega = diag(<1469.2;15099>); // ml estimates obtained from STAMP
    decl mw = SsfWeights(ST_SMO, mzt, mphi, momega, msigma);

    // graphs for mw, mbw and mbw-mw
}

```

Comment: the functions `GetSsfStsm`, `SsfWeights` and `SsfCondDens` are from *SsfPack*.

Code for vector error-correction model weights

```

#include <oxstd.h>
#include <oxdraw.h>
#include <packages/ssfpack/ssfpack.h>

decl s_idraw = 0, s_ct = 400, s_loc = 200;
decl s_mphi, s_momega, s_msigma, s_op = ST_PRED;

DrawVECM(const i, const j)
{
    decl my = zeros(2,s_ct); my[j][s_loc - 1] = 1;
    decl mw = SsfWeights(s_op, my, s_mphi, s_momega, s_msigma);
    mw = cumulate(mw' )';

    decl d = mw[i][s_loc - 1]; if (i==j) d -= 1.0;
    println(" phi* ", i+1, " ", j+1, " = ", d);
}

```

```

    // graph for mw[i] []
}

main()
{
    // bivariate LL (common) model
    s_mphi = <1,0;0,1;1,0;0,1>;
    s_momega = unit(4);
    s_momega[:1][:1] = <.4,.2;.2,.1>;
    s_msigma = 10^9 * s_momega[:1][:1] | 0;

    DrawVECM(0,0); DrawVECM(0,1); DrawVECM(1,0); DrawVECM(1,1);

    // show graphs
}

```

Comment: the Ox function `cumulate()` is used to compute the VECM matrices defined in (58).

References

- Ansley, C.F. and R. Kohn (1985) Estimation, filtering and smoothing in state space models with incompletely specified initial conditions. *Annals of Statistics*, 13, pp.1286-1316.
- Ansley, C.F. and R. Kohn (1990) Filtering and smoothing in state space models with partially diffuse initial conditions. *Journal of Time Series Analysis*, 11, pp.275-293.
- Balke, N.S. (1993) Detecting level shifts in time series. *Journal of Business and Economic Statistics*, 11, pp.81-92.
- Bell, W.R. (1984) Signal extraction for nonstationary time series, *Annals of Statistics*, 13, pp.646-664.
- Burridge, P. and K.F. Wallis (1988) Prediction theory for autoregressive-moving average processes. *Econometric Reviews*, 7, pp.65-69.
- de Jong, P. (1991) The Diffuse Kalman filter. *Annals of Statistics*, 19, pp.1073-1083.
- de Jong, P. and Shephard, N. (1995) The simulation smoother for time series models. *Biometrika*, 82, pp.339-350.
- Doornik, J.A. (1998) *Object-Oriented Matrix Programming using Ox 2.0*. London: Timberlake Consultants Press.
- Durbin, J. and S.J. Koopman (2000) *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press, forthcoming.
- Frühwirth-Schnatter, S. (1994) Applied state space modelling of non-Gaussian time series using integration-based Kalman filtering. *Statistics and Computing*, 4, pp.259-269.
- Green, P.G. and B.W. Silverman (1994) *Nonparametric regression and generalized linear models*. London: Chapman and Hall.
- Harvey, A.C. (1989) *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press.

- Harvey, A.C. and C-H. Chung (1999) Estimating the Underlying Change in UK Unemployment. Discussion paper, University of Cambridge.
- Harvey, A.C. and S.J. Koopman (1999) Signal Extraction and the Formulation of Unobserved Components Models. CentER Discussion paper, Tilburg University.
- Harvey, A.C., S.J. Koopman and J. Penzer (1998) Messy Time Series. In T.B. Fomby and R.C. Hill (eds), *Advances in Econometrics. Volume 13*. New York: JAI Press.
- Kitagawa, G., and W. Gersch (1996). *Smoothness priors analysis of time series*. Berlin: Springer-Verlag.
- Koopman, S.J. (1997) Exact initial Kalman filtering and smoothing for nonstationary time series models. *Journal of the American Statistical Association*, 92, pp.1630-1638.
- Koopman, S.J, N. Shephard and J.A. Doornik (1999) Statistical algorithms for models in state space using SsfPack 2.2. *Econometrics Journal*, 2, pp.113-166.
- Riani, M. (1998) Weights and Robustness of Model-based Seasonal Adjustment. *Journal of Forecasting*, 17, pp.19-34.
- Shephard, N. and Pitt, M.K. (1997) Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84, pp.653-667.
- Silverman, B.W. (1985) Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society, Series B*, 47, pp.1-52.
- Wecker, W.E. and C.F. Ansley (1983) The signal extraction approach to nonlinear regression and spline smoothing. *Journal of the American Statistical Association*, 78, pp.81-89.
- West, M. and P.J. Harrison (1997) *Bayesian Forecasting and Dynamic Models*, 2nd ed. New York: Springer-Verlag.
- Whittle, P. (1983) *Prediction and Regulation*, 2nd ed. Oxford: Blackwell.