A simple and efficient simulation smoother for state space time series analysis

BY J. DURBIN

Department of Statistics, London School of Economics and Political Science,
London WC2A 2AE, UK.

durbinja@aol.com

AND S.J. KOOPMAN

Department of Econometrics, Free University Amsterdam,
NL-1081 HV Amsterdam.

s.j.koopman@feweb.vu.nl

October 2001

SUMMARY

A simulation smoother in state space time series analysis is a procedure for drawing samples from the conditional distribution of state or disturbance vectors given the observations. We present a new technique for this which is both simple and computationally efficient. The treatment includes models with diffuse initial conditions and regression effects. Computational comparisons are made with the previous standard method. Two illustrations are provided using real data.

Some key words: Diffuse initialisation; Disturbance smoothing; Gibbs sampling; Importance sampling; Kalman filter; Markov chain Monte Carlo methods.

1. INTRODUCTION

State space models may be formulated in a variety of ways. In this paper we consider first the linear Gaussian form

$$\begin{aligned} y_t &= Z_t\alpha_t + \varepsilon_t, & \varepsilon_t &\sim \mathrm{N}(0, H_t), \\ \alpha_{t+1} &= T_t\alpha_t + R_t\eta_t, & \eta_t &\sim \mathrm{N}(0, Q_t), \qquad t = 1, \dots, n, \end{aligned} \tag{1}$$

where y_t is a $p \times 1$ vector of observations, α_t is an $m \times 1$ state vector and ε_t and η_t are vectors of disturbances. Matrices Z_t, T_t, R_t, H_t and Q_t are assumed to be known. To begin with we assume that $\alpha_1 \sim \mathrm{N}(a_1, P_1)$ where a_1 and P_1 are known; later we will investigate the case where elements of a_1 and P_1 are unknown. We will then consider the addition of a regression component of the form $X_t\beta$ to the first equation of (1).

We shall examine the problem of drawing samples from the conditional distributions of $\varepsilon = (\varepsilon_1', \dots, \varepsilon_n')'$, $\eta = (\eta_1', \dots, \eta_n')'$ and $\alpha = (\alpha_1', \dots, \alpha_n')'$ given $y = (y_1', \dots, y_n')'$. Such samples are needed for simulation studies of the properties of estimates arising in the analysis of model (1) and for the analysis of non-Gaussian and nonlinear variants of it from both classical and Bayesian inference perspectives.

Fruhwirth-Schnatter (1994b) and Carter and Kohn (1994) independently developed methods of drawing samples of $\alpha|y$ using a recursive technique consisting of first sampling $\alpha_n|y$, then sampling $\alpha_{n-1}|\alpha_n, y$, then $\alpha_{n-2}|\alpha_{n-1}, \alpha_n, y$, and so on. A significant advance was made by de Jong and Shephard (1995) for a model which is a generalisation of (1). They first considered recursive sampling of the disturbances and subsequently sampling of the states; this is generally more efficient than sampling the states directly when the dimension of η is smaller than the dimension of α. Their paper reviews previous work and describes the application of their simulation smoother to Bayesian Markov chain Monte Carlo (MCMC) analysis of Gaussian and non-Gaussian time series.

In this paper we present a new simulation smoother which is simple and is computationally efficient relative to that of de Jong and Shephard (1995). We achieve the improvements by avoiding generating random vectors recursively and employing instead a direct approach in which only mean corrections for unconditional vectors are required. The new simulation method can be adjusted straightforwardly to allow for diffuse initial conditions of the state vector and for the inclusion of a regression component in (1). To illustrate the use of the new method we apply it to classical and Bayesian analyses of structural time series models. Real data illustrations include monthly time series of number of car drivers killed or seriously injured in road accidents in Great Britain.

The next section presents the main result together with modifications for sampling state vectors, allowing for diffuse initial conditions and also for the inclusion of regression components. Section 3 discusses two applications which concern a Gaussian model and a Poisson model for counts. Our conclusions are presented in §4. The extension of our approach to the more general model employed by de Jong and Shephard (1995) is discussed in the Appendix.

2. THE NEW SIMULATION SMOOTHER

2·1 *Main result*

We first consider the construction of a simulation smoother for the disturbances ε and η. Let $w = (\varepsilon', \eta')'$ and let $\hat{w} = \mathrm{E}(w|y)$, $W = \mathrm{var}(w|y)$. Since the model is linear and Gaussian, the density of $w|y$ is $p(w|y) = \mathrm{N}(\hat{w}, W)$. The calculation of $\hat{w}$ is performed by means of the disturbance smoother as developed by Koopman (1993) based on work by de Jong (1988) and Kohn and Ansley (1989); for an elementary treatment see Durbin and Koopman (2001, §4.4.1). The matrix W has the important property that it does not depend upon y; this follows immediately from the general result that in a multivariate normal distribution the conditional variance matrix of a vector given that a second vector is fixed does not depend on the second vector; see, for example, Anderson (1984, Theorem 2.5.1). Since y is an exact linear function of the elements of w, the matrix W is singular; however, it turns out that this singularity has no effect on our calculations.

Our task is to draw random vectors $\tilde{w}$ from $p(w|y)$. We do this by drawing vectors from $\mathrm{N}(0, W)$ independently of y and adding these to the known vector $\hat{w}$. This is easily accomplished in the following way. The density of w is

$$p(w) = \mathrm{N}(0, \Omega), \qquad \Omega = \mathrm{diag}(H_1, \dots, H_n, Q_1, \dots, Q_n). \tag{2}$$

Let w^+ be a random vector drawn from $p(w)$. The process of drawing w^+ is straightforward, particularly since in most cases in practice the matrices H_t and Q_t, for $t = 1, \dots, n$, are scalars or diagonal. Denote by y^+ the stacked vector of values of y_t generated by drawing a vector

α_1^+ from $p(\alpha_1)$ and replacing α_1 and w in (1) by α_1^+ and w^+. Compute $\hat{w}^+ = \mathrm{E}(w^+|y^+)$ using the disturbance smoother given in (4) below. Since W is independent of y, $\mathrm{var}(w^+|y^+) = W$. Consequently, $w^+ - \hat{w}^+$ is the desired draw from $\mathrm{N}(0, W)$. Let $\tilde{w} = \hat{w} + w^+ - \hat{w}^+$. It follows that $\tilde{w}$ is a draw from density $p(w|y)$. In particular, we have

$$\mathrm{E}(\tilde{w}|y) = \mathrm{E}(\hat{w} + w^+ - \hat{w}^+|y) = \mathrm{E}(w^+ - \hat{w}^+|y) + \hat{w} = \hat{w},$$

and

$$\mathrm{var}(\tilde{w}|y) = \mathrm{E}\left[(w^+ - \hat{w}^+)(w^+ - \hat{w}^+)'|y\right] = W,$$

since $w^+ - \hat{w}^+$ is independent of y.

This result implies the validity of the following algorithm for selecting a draw $\tilde{w}$ from density $p(w|y)$.

Algorithm 1.

1. Draw a random vector w^+ from density $p(w)$ and use it to generate y^+ by means of recursion (1) with w replaced by w^+, where the recursion is initialised by the draw $\alpha_1^+ \sim \mathrm{N}(a_1, P_1)$;

2. Compute $\hat{w} = \mathrm{E}(w|y)$ and $\hat{w}^+ = \mathrm{E}(w^+|y^+)$ by means of standard Kalman filtering and disturbance smoothing using (3) and (4) below;

3. Take $\tilde{w} = \hat{w} - \hat{w}^+ + w^+$.

The algorithm is applied as many times as is needed to obtain the desired sample of independent values of $\tilde{w}$. When a single draw $\tilde{w}$ is required, the amount of computing can be reduced by defining $y_t^* = y_t - y_t^+$ and putting y_t^* through the Kalman filter and disturbance smoother once instead of putting y_t and y_t^+ separately through the filter and smoother.

This algorithm for generating $\tilde{w}$ only requires standard Kalman filtering and disturbance smoothing applied to the constructed series y^+ and is therefore easy to incorporate in new software; special algorithms for simulation smoothing such as the ones developed by Fruhwirth-Schnatter (1994b), Carter and Kohn (1994) and de Jong and Shephard (1995) are not required. We do not regard the generation of y^+ by (1) as an algorithm since we are merely making straightforward use of the basic model. Thus, the result is not only mathematically simple, it is also computationally simple.

In §2·2 we present formulae for the Kalman filter and disturbance smoother that are needed for the implementation of Algorithm 1. We discuss in §2·3 a modified version of Algorithm 1 which is slightly more efficient computationally. Obviously, if we do not require the whole of $\tilde{w}$, but only the part consisting of either ε or η, Steps 2 and 3 of Algorithm 1 can be confined to the relevant part. The whole vector w^+ is, however, needed for Step 1. In §2·4 we obtain a simulation smoother for the state vector α. The case where at least part of the initial vector α_1 is diffuse is considered in §2·5. Finally, in §2·6 we discuss the computation of antithetic variables in our method.

It will be evident from the above treatment that the same approach could be employed to prove the following general proposition. Suppose that x and y are vectors which are jointly normally distributed with density $p(x, y)$ and that we wish to draw sample vectors from density $p(x|y)$. Denote a draw from density $p(x, y)$ by x^+, y^+ and let $\hat{x} = \mathrm{E}(x|y)$, $\hat{x}^+ = \mathrm{E}(x^+|y^+)$ and $\tilde{x} = \hat{x} + x^+ - \hat{x}^+$. Then $\tilde{x}$ is a draw from $p(x|y)$. We mention this generalisation in case there are situations other than state space applications where the device might be useful, particularly where drawing from $p(x, y)$ and calculation of $\mathrm{E}(x|y)$ are relatively easy, while direct drawing from $p(x|y)$ is relatively difficult.

2·2 *The Kalman filter and disturbance smoother*

The Kalman filter for model (1) is

$$\begin{aligned} v_t &= y_t - Z_t a_t, & F_t &= Z_t P_t Z_t' + H_t, \\ K_t &= T_t P_t Z_t' F_t^{-1}, & L_t &= T_t - K_t Z_t, \\ a_{t+1} &= T_t a_t + K_t v_t, & P_{t+1} &= T_t P_t L_t' + R_t Q_t R_t', \end{aligned} \tag{3}$$

for $t = 1, \ldots, n$ with a_1 and P_1 as the mean vector and variance matrix of the initial state vector α_1. Proofs are given by, for example, Anderson and Moore (1979, Chapter 3) and Durbin and Koopman (2001, §4.2.1)

The computation of $\hat{w}$ takes the form

$$\hat{w}_t = \begin{bmatrix} H_t F_t^{-1} & -H_t K_t' \\ 0 & Q_t R_t' \end{bmatrix} \begin{pmatrix} v_t \\ r_t \end{pmatrix}, \tag{4}$$

where r_t is evaluated by the backwards recursion

$$r_{t-1} = Z_t F_t^{-1} v_t + L_t' r_t, \tag{5}$$

for $t = n, n-1, \ldots, 1$ with $r_n = 0$. The two block elements obtained by multiplying out the right-hand side of (4) give the equations for $\hat{\varepsilon}_t = \mathrm{E}(\varepsilon_t|y)$ and $\hat{\eta}_t = \mathrm{E}(\eta_t|y)$, respectively. One or the other of these can be used when multiple draws of ε only or η only are required. Proofs of the formulae are given in Koopman (1993) and Durbin and Koopman (2001, §4.4).

It should be noted that in standard cases the matrices P_t, F_t, K_t and L_t in (3) and (4), as distinct from the vectors a_t, v_t and r_t, are all independent of y. However, some or all of them will in practical cases of interest depend on an unknown parameter vector, ψ say. Consequently, when the analysis is based on classical inference, an estimate $\hat{\psi}$ of ψ will be calculated at the beginning of the analysis, and the values of the matrices will be treated as if $\hat{\psi}$ was the true value of ψ. Thus when generating multiple draws using Algorithm 1, only the elements of vectors a_t, v_t and r_t need recalculation for each draw of $\tilde{w}$. On the other hand, when the analysis is Bayesian, the parameter vector ψ is treated as random and it will vary from one simulation to another. Thus the matrices that depend on ψ will need to be recalculated for each draw of $\tilde{w}$. The effect is that more calculation per draw is required when multiple samples are required within a Bayesian analysis than for a classical analysis.

2·3 *Modified version of the simulation smoothing algorithm*

We observe that the smoothing recursion (4) depends as a function of y only on $v = (v_1', \ldots, v_n')'$. This suggests that we can increase computational efficiency by generating v from w directly during the simulations without computing y as an intermediate step. Let $x_t = \alpha_t - a_t$. Then

$$\begin{aligned} v_t &= Z_t \alpha_t + \varepsilon_t - Z_t a_t \\ &= Z_t x_t + \varepsilon_t, \qquad\qquad t = 1, \ldots, n, \end{aligned} \tag{6}$$

and

$$\begin{aligned} x_{t+1} &= T_t \alpha_t + R_t \eta_t - T_t a_t - K_t v_t \\ &= T_t x_t + R_t \eta_t - K_t v_t, \qquad\qquad t = 1, \ldots, n-1, \end{aligned} \tag{7}$$

initialised with $x_1 \sim \mathrm{N}(0, P_1)$. Thus if we select x_1^+ from $\mathrm{N}(0, P_1)$ and substitute subvectors $\varepsilon_1^+, \ldots, \varepsilon_n^+, \eta_1^+, \ldots, \eta_{n-1}^+$ from w^+ into (6) and (7) we can obtain $v_1^+, \ldots, v_n^+$ directly rather than generate y^+ from (1) and then derive the v_t^+'s from the relevant parts of the Kalman filter. This process involves fewer numerical operations than are required in Algorithm 1. However, the computational gain is small since all operations in the simulation once w^+ has been drawn are linear so the computations based on them are already fast. Noting that when y is fixed, v is fixed, and vice-versa, we obtain a modified form of Algorithm 1 in which the subvectors ε^+ and η^+ are used in Step 1 to generate $v^+ = (v_1^{+\prime}, \ldots, v_n^{+\prime})'$ from (6) and (7). Steps 2 and 3 then proceed as before. Since $\mathrm{E}(w^+|v^+) = \mathrm{E}(w^+|y^+)$ where y^+ is the value that would have been obtained in Step 1 of Algorithm 1 from the same w^+ and the same value of $x_1 = \alpha_1 - a_1$, it follows that $\tilde{w} \sim p(w|y)$.

2·4 *Simulation smoothing for state vector*

To construct an algorithm for generating draws of the state vector $\alpha = (\alpha_1', \ldots, \alpha_n')'$ from the conditional density $p(\alpha|y)$, we denote a draw from $p(\alpha)$ as α^+ and a draw from $p(\alpha|y)$ as $\tilde{\alpha}$. The smoothed mean $\hat{\alpha}_t = \mathrm{E}(\alpha_t|y)$ can be computed as suggested by Koopman (1993) by taking the conditional expectation given y of both sides of the second equation of (1), substituting for $\hat{\eta}_t$ from the second line of (4) and then applying the resulting forwards recursion

$$\hat{\alpha}_{t+1} = T_t\hat{\alpha}_t + R_tQ_tR_t'r_t, \qquad t = 1, \ldots, n, \tag{8}$$

with the initialisation $\hat{\alpha}_1 = a_1 + P_1r_0$, where r_t is obtained from (5); for details about the initialisation see Durbin and Koopman (2001, §4.4.2).

Based on this approach, the following algorithm for drawing random vectors $\tilde{\alpha}$ from $p(\alpha|y)$ is obtained by arguments similar to those used for drawing $\tilde{w}$ from $p(w|y)$ in Algorithm 1.

Algorithm 2.

1. Draw a random vector w^+ from density $p(w)$ and use it to generate α^+ and y^+ by means of recursion (1) with w replaced by w^+, where the recursion is initialised by the draw $\alpha_1^+ \sim \mathrm{N}(a_1, P_1)$;

2. Compute $\hat{\alpha} = \mathrm{E}(\alpha|y)$ and $\hat{\alpha}^+ = \mathrm{E}(\alpha^+|y^+)$ by means of standard filtering and smoothing using (3) forwards, (4) backwards and (8) forwards;

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

When a single draw $\tilde{\alpha}$ is required, it is computationally more efficient to compute $\tilde{\alpha}$ by constructing the artificial observations $y^* = y - y^+$ and using $\tilde{\alpha} = \hat{\alpha}^* + \alpha^+$ where $\hat{\alpha}^* = \mathrm{E}(\alpha|y^*)$.

2·5 *Modifications for diffuse initial conditions*

In situations where the initial state vector contains nonstationary elements or unknown fixed coefficients, we treat the corresponding initial elements as diffuse random variables, that is, as having infinite variances. Exact solutions have been developed by Ansley and Kohn (1985), de Jong (1991) and Koopman (1997) for filtering and smoothing the observed series under the assumption that some elements of P_1 go to infinity. A detailed treatment of diffuse initialisation is given by Durbin and Koopman (2001, Chapter 5), particularly in §§5.3 and 5.4 where explicit formulae are given for calculating $\hat{\alpha} = \mathrm{E}(\alpha|y)$ and $\hat{w} = \mathrm{E}(w|y)$. Smoothers obtained by formulae given in these sections we shall refer to as diffuse smoothers.

An outstanding question is the draw $\alpha_1^+ \sim \mathrm{N}(a_1, P_1)$ in Step 1 of Algorithm 1 since in the diffuse case some elements of P_1 will have variances going to infinity and a draw from a normal density with infinite variance is impossible. However, we now show that provided diffuse smoothers are used for the calculation of $\hat{w}^+$, the diffuse elements of α_1 can be set equal to arbitrary quantities, say zeros, when using Algorithms 1 and 2.

The initial state vector can be modelled generally by

$$\alpha_1 = A_1\delta + C_1\chi, \qquad \delta \sim \mathrm{N}(0, \kappa I), \qquad \chi \sim \mathrm{N}(\lambda, I),$$

where $\kappa \to \infty$ with δ and χ independent. It follows that $\alpha_1 \sim \mathrm{N}(a_1, P_1)$ with

$$a_1 = C_1\lambda, \qquad P_1 = \kappa A_1 A_1' + C_1 C_1'.$$

Substituting in model (1), it follows that

$$y = A\delta + Bw + C\chi, \qquad \alpha = H\delta + Gw + D\chi,$$

where y, w and α are defined in §2·1 and the matrices A, B, C, H, G and D are known functions of the system matrices. For a given value of κ we have

$$\hat{w} \quad = \quad \mathrm{Cov}(w, y)\Sigma^{-1}[y - \mathrm{E}(y)], \tag{9}$$

where

$$\mathrm{Cov}(w,y) = \Omega B', \qquad \Sigma = \kappa A A' + \Sigma_*, \qquad \Sigma_* = B\Omega B' + CC', \qquad \mathrm{E}(y) = C\lambda,$$

with $\Omega = \mathrm{var}(w)$ defined in (2). Applying a standard inversion lemma to Σ, see for example Rao (1973, p.33, Problem 2.9), gives

$$\Gamma = \Sigma^{-1} = \Sigma_*^{-1} - \Sigma_*^{-1} A (\frac{1}{\kappa} I + A'\Sigma_*^{-1} A)^{-1} A' \Sigma_*^{-1},$$

for $\kappa > 0$ and so

$$\hat{w} = \Omega B' \Gamma (y - C\lambda).$$

Letting $\kappa \to \infty$ we obtain

$$\hat{w} = \Omega B' \Gamma_\infty (y - C\lambda), \tag{10}$$

where

$$\Gamma_\infty = \Sigma_*^{-1} - \Sigma_*^{-1} A (A'\Sigma_*^{-1} A)^{-1} A' \Sigma_*^{-1}. \tag{11}$$

Equation (10) provides a general form for a value of $\hat{w}$ obtained by the use of a diffuse smoother.

Let δ^+ be an arbitrary value of δ and let χ^+ be a random draw of χ. Now apply Algorithm 1 to compute y^+, $\hat{w}^+$ and $\tilde{w}$, taking $\alpha_1^+ = A_1\delta^+ + C_1\chi^+$ and using the diffuse smoother to compute $\hat{w}^+$. Since (10) holds for any realised vector y which satisfies model (1), it holds for $y^+ = A\delta^+ + Bw^+ + C\chi^+$ so we have

$$\begin{aligned} \hat{w}^+ &= \Omega B' \Gamma_\infty (y^+ - C\lambda) \\ &= \Omega B' \Gamma_\infty [A\delta^+ + Bw^+ + C(\chi^+ - \lambda)]. \end{aligned} \tag{12}$$

Postmultiplying (11) by A gives $\Gamma_\infty A = 0$ so δ^+ disappears from (12) and we therefore have

$$\hat{w}^+ = \Omega B' \Gamma_\infty [Bw^+ + C(\chi^+ - \lambda)], \tag{13}$$

which does not depend on δ^+. It follows that we can take $\delta^+ = 0$ and $\alpha_1^+ = C_1\chi$, thus obtaining a finite series $y_1^+, \ldots, y_n^+$.

A similar result applies to state simulation smoothing. We have

$$\hat{\alpha} = \mathrm{E}(\alpha) + \mathrm{Cov}(\alpha, y)\Sigma^{-1}[y - \mathrm{E}(y)], \tag{14}$$

where

$$\mathrm{E}(\alpha) = D\lambda, \qquad \mathrm{Cov}(\alpha, y) = \kappa H A' + X, \qquad X = G\Omega B' + DC'.$$

Thus

$$\hat{\alpha} \;=\; D\lambda + X\Gamma(y - C\lambda) + \kappa H A'\Gamma(y - C\lambda), \tag{15}$$

with

$$\begin{aligned} \kappa A'\Gamma &= \kappa[I - A'\Sigma_*^{-1}A(\tfrac{1}{\kappa}I + A'\Sigma_*^{-1}A)^{-1}]A'\Sigma_*^{-1} \\ &= \kappa[(\tfrac{1}{\kappa}I + A'\Sigma_*^{-1}A) - A'\Sigma_*^{-1}A](\tfrac{1}{\kappa}I + A'\Sigma_*^{-1}A)^{-1}A'\Sigma_*^{-1} \\ &= (\tfrac{1}{\kappa}I + A'\Sigma_*^{-1}A)^{-1}A'\Sigma_*^{-1}, \end{aligned} \tag{16}$$

for $\kappa > 0$. As $\kappa \to \infty$ we have

$$\hat{\alpha} \;=\; D\lambda + X\Gamma_\infty(y - C\lambda) + H(A'\Sigma_*^{-1}A)^{-1}A'\Sigma_*^{-1}(y - C\lambda). \tag{17}$$

To obtain $\tilde{\alpha} = \hat{\alpha} + \alpha^+ - \hat{\alpha}^+$, we first compute y^+ and α^+ from (1) initialised with $\alpha_1 = A_1\delta^+ + C_1\chi^+$ where δ^+ is arbitrary and then calculate $\hat{\alpha}$ and $\hat{\alpha}^+$ using diffuse smoothers. Analogously to (13), we have

$$\hat{\alpha}^+ \;=\; D\lambda + (X\Gamma_\infty + H(A'\Sigma_*^{-1}A)^{-1}A'\Sigma_*^{-1})[Bw^+ + C(\chi^+ - \lambda)] + H\delta^+, \tag{18}$$

which includes the term $H\delta^+$. However, this term will be eliminated when computing $\tilde{\alpha}$ since it also appears in $\alpha^+ = H\delta^+ + Gw^+ + D\chi^+$. We can therefore take $\delta^+ = 0$.

If the observational vector y_t depends on a regressor matrix X_t with unknown constant regression coefficient vector β the first equation of (1) is replaced by the form

$$y_t = Z_t\alpha_t + X_t\beta + \varepsilon_t. \tag{19}$$

We can estimate β in the Kalman filter by redefining the state vector as $\alpha_t^* = (\alpha_t', \beta_t')'$, with the constraints $\beta_1 = \beta$ and $\beta_{t+1} = \beta_t$, $t = 1, \ldots, n$, and modifying the second equation of (1) accordingly. We then treat the vector β_1 as diffuse. It follows from the earlier results of this section that we can put $\beta = \beta_1 = 0$ when drawing (unconditional) simulation samples provided that we use diffuse smoothers for the expanded model to calculate $\hat{w}$, $\hat{w}^+$, $\hat{\alpha}$ and $\hat{\alpha}^+$. This has the computational advantage that we can exclude X_t and consequently employ the reduced model (1) when computing y^+. This solution is simpler than the treatment of fixed effects given by de Jong and Shephard (1995, §5).

2·6 *Antithetic variables*

When using the simulation smoother in practice, it is often advantageous to employ antithetic variables. An antithetic variable for a draw x is one which is equiprobable with x and which, when used together with x, increases simulation efficiency. It is easy to construct antithetic variables using the techniques of this paper. Thus for the draw $\tilde{w} = \hat{w} - \hat{w}^+ + w^+$, we note that $w^+ - \hat{w}^+$ and $-(w^+ - \hat{w}^+)$ have the same distribution $\mathrm{N}(0, W)$. It follows that if we define $\tilde{w}^- = \hat{w} + \hat{w}^+ - w^+$ then $\tilde{w}$ and $\tilde{w}^-$ have the same conditional distribution given y, that is, $\mathrm{N}(\hat{w}, W)$. The use of $\tilde{w}$ and $\tilde{w}^-$ together in the estimation process leads to an increase in efficiency for two reasons. First, estimates based on $\tilde{w}$ and $\tilde{w}^-$ separately are likely to be negatively correlated. Secondly, two draws of $w|y$ are obtained for a computational cost which is little more than the cost of $\tilde{w}$ alone. A second antithetic could be constructed along similar lines to the one described in Durbin and Koopman (2001, §11.9.3) but we shall not pursue this further here.

3. ILLUSTRATIONS

3·1 *Bayesian analysis based on Gibbs sampling*

For our first illustrative example, we consider the class of structural time series models as discussed in Harvey (1989) and Durbin and Koopman (2001, §3.2) for which a Bayesian analysis will be presented based on the Gibbs sampler as proposed by Fruhwirth-Schnatter (1994b) and Carter and Kohn (1994). For example, let us consider the local level model

$$y_t = \mu_t + \varepsilon_t, \qquad \mu_{t+1} = \mu_t + \eta_t, \tag{20}$$

where the disturbances ε_t and η_t are mutually and serially uncorrelated and generated by normal densities with zero mean and variances σ^2_ε and σ^2_η, respectively. The variances are treated as random variables and, as an example, a model for a variance σ^2 can be based on the inverse gamma distribution with logdensity

$$\log p(\sigma^2|c,s) = -\log\Gamma(\frac{c}{2}) - \frac{c}{2}\log\frac{s}{2} - \frac{c+2}{2}\log\sigma^2 - \frac{s}{2\sigma^2}, \qquad \text{for } \sigma^2 > 0,$$

and $p(\sigma^2|c,s) = 0$ for $\sigma^2 \leq 0$; see, for example, Poirier (1995, Table 3.3.1). We denote this density by $\sigma^2 \sim \mathrm{IG}(c/2, s/2)$ where c determines the shape and s determines the scale of the distribution. It has the convenient property that if we take this as the prior density of σ^2 and we take a sample $u_1, \ldots, u_n$ of independent $\mathrm{N}(0, \sigma^2)$ variables, the posterior density of σ^2 is

$$p(\sigma^2 | u_1, \ldots, u_n) = \mathrm{IG}[(c+n)/2, (s + \sum_{i=1}^{n} u_i^2)/(2\sigma^2)]; \tag{21}$$

for further details see, for example, Poirier (1995, Chapter 6).

The posterior means of $\mu = (\mu_1, \ldots, \mu_n)'$ and of the variances $\psi = (\sigma_\varepsilon^2, \sigma_\eta^2)'$ can be estimated by simulating from the joint density $p(\psi, \mu | y)$ and taking sample means. In a Markov chain Monte Carlo (MCMC) procedure, the sampling from this joint density is implemented as a Markov chain. After the initialisation $\psi = \psi^{(0)}$, we repeat the following simulation steps M^* times:

1. sample $\mu^{(i)}$ from $p(\mu | y, \psi^{(i-1)})$ using Algorithm 1 of §2·1 to obtain $\varepsilon^{(i)}$ and hence $\mu^{(i)}$ from (20);

2. sample $\psi^{(i)}$ from $p(\psi | y, \mu^{(i)})$ using the inverse gamma density;

for $i = 1, \ldots, M^*$. After the process has stabilised, we treat the last M samples from Step 2 as being generated from the density $p(\psi | y)$. Usually, sampling from conditional densities is easier than sampling from the marginal density $p(\psi | y)$. For the implementation of Step 2 a sample value of an element of ψ is chosen from the posterior density (21). We can take u_t in (21) as a standardised element of ε_t or η_t obtained by the simulation smoother of §2·1 in Step 1. Here, we are following standard practice in working with the marginal distributions of σ_ε^2 and σ_η^2 instead of their joint distributions.

Similar methods can be applied to the local linear trend model, which incorporates a stochastic slope in μ_t, and to the basic structural time series model, which includes slope and seasonal components; see Harvey (1989, §2.3) or Durbin and Koopman (2001, §3.2) for details of these models. The Gibbs sampler requires the application of a simulation smoother M times. We now investigate the computational efficiency of Algorithm 1 compared to the simulation smoother of de Jong and Shephard (1995), hereafter referred to as method JS, for a general class of models. We accept the claim in their paper that for most cases method JS is computationally more efficient than the methods of Fruhwirth-Schnatter (1994b) and Carter and Kohn (1994). Method JS and Algorithm 1 both require the Kalman filter although method JS applies it to the observed series y_t whereas in effect our method applies it to the constructed series $y_t^* = y_t - y_t^+$; for this we need to draw random values of disturbances from univariate normal densities and then apply the state space recursion (1). After the Kalman filter, Algorithm 1 applies standard disturbance smoothing whereas method JS applies either equation (3) or (4) or (5) in de Jong and Shephard (1995) which is similar to backwards disturbance smoothing but is computationally more involved.

Table 1 presents the numbers of multiplications required for a single draw of univariate ($p = 1$) state space models with different state vector dimensions. It is assumed that the elements of Z_t, T_t and R_t are either zero or one and variance matrices H_t and Q_t are diagonal. Method JS clearly involves more computations for all state dimensions although when a draw from $p(\varepsilon|y)$ only is required, differences with Algorithm 1 are smaller. Method JS further requires for each time period t an inversion of a symmetric $m \times m$ matrix and a draw from a multivariate normal distribution whereas Algorithm 1 does not require matrix inversions and draws are from univariate densities. Both methods require the same storage from the Kalman filter, that is, storage of v_t, F_t and K_t for $t = 1, \ldots, n$. We conclude that computational gains are achieved using our simulation smoothing Algorithm 1 compared to method JS. The computational gains for the modified algorithm of §2·3 are virtually the same since the main difference from Algorithm 1 is that the Kalman filter equation for a_{t+1} in (3) is replaced by the equation for x_{t+1} in (7) and the resulting difference is negligible.

[Table 1 about here]

3·2 *Linear Gaussian illustration*

In this illustration we follow Fruhwirth-Schnatter (1994a) in considering data from the study of Harvey and Durbin (1986) on the effect of the seat belt law on road accidents in Great Britain using a Bayesian analysis based on a structural time series model. A graph of the log of monthly number of car drivers killed or seriously injured shows a seasonal pattern due primarily to weather conditions and festive celebrations. The overall trend of the series is basically constant over the years with breaks in the mid-seventies, probably due to the oil crisis, and in February 1983 after the introduction of the seat belt law. The model that we consider is

$$y_t = \mu_t + \gamma_t + \varepsilon_t,$$

where μ_t is the local level component as given by (20) and the seasonal component γ_t is modelled by the equation

$$\gamma_t + \gamma_{t-1} + \ldots + \gamma_{t-s+1} = \omega_t,$$

where s is the seasonal length and ω_t is a disturbance with mean zero and variance σ^2_ω. The irregular ε_t is treated as a disturbance term with mean zero and variance σ^2_ε. All disturbances are mutually and serially uncorrelated and generated by normal densities. For the purpose of this illustration we do not include an intervention component to measure the effect of the seat belt law.

[Table 2 about here]

We have applied the Gibbs sampler as described in the previous subsection with $M = 2000$. The computations were implemented in the matrix language `Ox` of Doornik (1998) using `SsfPack` of Koopman, Shephard, and Doornik (1999). Let $\psi = (\sigma^2_\eta, \sigma^2_\omega, \sigma^2_\varepsilon)'$. Figure 1 shows the realised draws from $p(\psi|y)$, the correlogram of the series of draws and a histogram of the realised draws which is smoothed nonparametrically to provide estimates of the posterior densities of the elements of ψ. The estimated posterior means and standard deviations of elements of ψ are reported in Table 2. We observe that the histogram of the seasonal variance σ^2_ω has a lot of mass at zero and its posterior mean is also close to zero. This is consistent with the fact that in a classical analysis the maximum likelihood estimate of σ^2_ω is zero. We therefore repeat the Gibbs sampler with σ^2_ω fixed at zero and the same results as in Figure 1 are presented in Figure 2 for $\psi^* = (\sigma^2_\eta, \sigma^2_\varepsilon)'$. In the second panel of Table 2 the posterior mean and standard deviation of ψ^* are reported. Comparisons of Figures 1 and 2 supports the view that the Gibbs sampler works better for ψ^* than for ψ. The computer time required for the Gibbs sampler using Algorithm 1 was about 30% less compared to using method JS for both ψ and ψ^*.

[Figures 1 and 2 about here]

3·3 *Illustration of the use of importance sampling for non-Gaussian models*

We now consider classical inference for a class of models in which the normal density of the observation equation in (1) is replaced by the more general class of the exponential family densities, that is, we generalise the density

$$p(y_t|\theta_t) = \mathrm{N}(\theta_t, \sigma_\varepsilon^2),$$

where $\theta_t = Z_t y_t$, to densities of the form

$$p(y_t|\theta_t) = \exp[y_t'\theta_t - b_t(\theta_t) + c_t(y_t)], \qquad -\infty < \theta_t < \infty, \tag{22}$$

where $b_t(\theta_t)$ is a twice differentiable function and $c_t(y_t)$ is a function of y_t only. Examples of such densities include the Poisson, binomial and exponential densities.

For this more general class of models, smoothed estimates of the state vector cannot be evaluated analytically so we adopt simulation techniques. Using methods developed in Shephard and Pitt (1997), Durbin and Koopman (1997) and Durbin and Koopman (2000), we evaluate the smoothed state vector by means of importance sampling based on the use of an approximating linear Gaussian model with observational density denoted by $g(y_t|\theta_t)$. The approximating model is based on the standard state space model (1) and is obtained by solving the equations

$$\frac{\partial p(y_t|\theta_t)}{\partial \theta_t} = \frac{\partial g(y_t|\theta_t)}{\partial \theta_t}, \qquad \frac{\partial^2 p(y_t|\theta_t)}{\partial \theta_t \partial \theta_t'} = \frac{\partial^2 g(y_t|\theta_t)}{\partial \theta_t \partial \theta_t'};$$

see Durbin and Koopman (2001, Chapter 11) for further details. The smoothed estimator of the state vector α_t for exponential family models can be computed by

$$\hat{\alpha}_t = \frac{\sum_{i=1}^M \alpha_t^i w_i}{\sum_{i=1}^M w_i},$$

where $w_i = \prod_{t=1}^n p(y_t|\theta_t^i)/g(y_t|\theta_t^i)$, with $\theta_t^i = Z_t\alpha_t^i$ where α_t^i is a draw from the conditional Gaussian density $g(\alpha_t|y)$ for the approximating linear Gaussian model.

To employ this approach we require multiple samples of the state vectors using simulation smoothing algorithms. To sample from $p(\alpha|y)$ we use Algorithm 2 in §2·4. The method JS is different in the sense that it first samples from $p(\eta|y)$ and then computes draws for α_t using the second equation of (1). Table 3 presents the numbers of multiplications required for multiple draws of univariate ($p = 1$) state space models with different state vector dimensions. It is assumed that the elements of Z_t, T_t and R_t are either zero or one and variance matrices H_t and Q_t are diagonal. Since matrices such as F_t, K_t and P_t in (3) and U_t in (4) of JS depend only on the parameter vector ψ which is kept fixed for all draws, we only need to repeat the calculation of v_t in (3), $\hat{\eta}_t$ in (4) and $\hat{\alpha}_t$ in (8) for our method while method JS only requires to repeat the computation of η_t using (4) in JS and α_t using (1). The number of multiplications required to draw from $p(\eta|y)$ in our method is smaller when the state vector dimension is larger than one; the computational gains become more evident when the state size increases. However, when drawing from the density $p(\varepsilon|y)$, method JS is more efficient by one multiplication irrespective of the state dimension. For the implementation of both Algorithms 1 and 2, the storage of F_t and K_t in (3) only is required whereas method JS requires the extra storage of C_t and V_t in (4) of JS. Table 4 presents the number of values to be stored when multiple draws need to be selected and it confirms that the required storage space for our method is small relative to that required for method JS.

[Tables 3 and 4 about here]

3·4 *Poisson illustration*

We now apply these ideas to the special case of the monthly numbers of light goods vehicle (van) drivers killed in road accidents from 1969 to 1984, which was previously considered by us in Durbin and Koopman (2000, §6.1). The numbers of deaths of van drivers were too small to justify the use of the linear Gaussian model. A better model for the data is based on the Poisson distribution with mean $\exp(\theta_t)$ and density

$$p(y_t|\theta_t) = \exp\{\theta_t' y_t - \exp(\theta_t) - \log y_t!\}, \qquad t = 1, \ldots, n. \tag{23}$$

We model θ_t by the relation

$$\theta_t = \mu_t + \gamma_t,$$

where the level μ_t and seasonal γ_t have the same specification as in §3·2. For simplicity we do not include an intervention component to measure the effect of the seat belt law. The smoothed estimates of the two components were computed using importance sampling as decribed in the previous section. The computations were implemented in the matrix language `Ox` of Doornik (1998) using `SsfPack` of Koopman, Shephard, and Doornik (1999). Figure 3 presents the smoothed estimates of μ_t and γ_t. A reduction of about 15% computing time was achieved when we replaced method JS by our Algorithm 1.

[Figure 3 about here]

4. CONCLUSIONS

In this paper we have presented a new simulation smoother for drawing samples from the conditional distribution of the disturbances given the observations. We obtained this by exploiting elementary properties of the multivariate normal distribution. Our main algorithm only involves the application of standard filtering and smoothing methods and does not require special recursions. An extension of the algorithm is given for the case in which draws of the state vector are required. Some of the advantages of our algorithms in relation to existing methods are:

- derivation is simple;
- the method requires only the generation of simulated observations from the model together with the Kalman filter and standard smoothing algorithms;
- no inversions of matrices are needed beyond those in the standard Kalman filter;
- our algorithms involve smaller numbers of multiplications than other methods;
- our approach solves problems arising from the singularity of the conditional variance matrix W automatically;
- for many practical models, draws from multivariate normal distributions are not needed;
- when multiple samples are needed, required storage space is smaller than with other methods;
- diffuse initialisation of the state vector is handled simply.

ACKNOWLEDGEMENT

We thank the referees for some perceptive comments which led to a significant improvement of the paper.

APPENDIX A

In this Appendix we consider the application of the basic technique of this paper to the state space model used in de Jong and Shephard (1995), which was originally proposed by de Jong (1991). This model is

$$\begin{aligned} y_t &= X_t\beta + Z_t\alpha_t + G_t u_t, & t &= 1, \ldots, n, \\ \alpha_{t+1} &= W_t\beta + T_t\alpha_t + H_t u_t, & t &= 0, 1, \ldots, n, \end{aligned} \tag{24}$$

where $\alpha_0 = 0$, the u_t's are independent $\mathrm{N}(0, \sigma^2 I)$ vectors, and the coefficient matrices may depend, implicitly, on some random vector ω drawn from a specified distribution. We use the notation of de Jong and Shephard (1995) in this Appendix in order to facilitate comparison with their paper; their use of symbols G_t, H_t and others should not be confused with our use of these symbols in the main paper. Model (24) is more general than our model (1) since it allows overtly for regression effects and for correlation between the disturbances $G_t u_t$ and $H_t u_t$ in the two equations of (24). However, we prefer our formulation since it appears in most textbooks and since we regard it as more transparent than (24) while at the same time covering most practical applications. We believe that it is preferable to treat regression effects and correlation between disturbances as optional extras that can be dealt with separately.

A general form of simulation smoothing is considered in de Jong and Shephard (1995) in which they draw samples of η from density $p(\eta|y, \omega)$ with $\eta = (\eta_0', \eta_1', \ldots \eta_n')'$ and $\eta_t = F_t u_t$, where the F_t are, with some qualifications, arbitrary matrices. Following the approach of our §2, let u_t^+ be a random draw from $\mathrm{N}(0, \sigma^2 I)$, generate $y_1^+, \ldots, y_n^+$ from $u_0^+, u_1^+, \ldots, u_n^+$ using (24), let $\eta_t^+ = F_t u_t^+$, $\eta^+ = (\eta_0^{+\prime}, \eta_1^{+\prime}, \ldots \eta_n^{+\prime})'$, $\hat{\eta} = \mathrm{E}(\eta|y, \omega)$, $\hat{\eta}^+ = \mathrm{E}(\eta^+|y^+, \omega)$ and $\tilde{\eta} = \hat{\eta} + \eta^+ - \hat{\eta}^+$. It follows by applying the steps of the proof in §2·1 that $\tilde{\eta} \sim p(\eta|y, \omega)$. The smoothed vector $\hat{\eta}$ is obtained by taking $\hat{\eta}_t = F_t(G_t' D_t^{-1} e_t + J_t' r_t)$, where D_t and e_t are given by (2) and r_t is given by (3) of de Jong and Shephard (1995) with $\varepsilon_t = 0$ and $V_t = 0$. It is worth mentioning that our technique handles cases where $\mathrm{var}(\eta|y, \omega)$ is singular without difficulty.

References

Anderson, B. D. O. and J. B. Moore (1979). *Optimal Filtering.* Englewood Cliffs: Prentice-Hall.

Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis* (2nd ed.). New York: John Wiley & Sons.

Ansley, C. F. and R. Kohn (1985). Estimation, filtering and smoothing in state space models with incompletely specified initial conditions. *Annals of Statistics 13*, 1286–1316.

Carter, C. K. and R. Kohn (1994). On Gibbs sampling for state space models. *Biometrika 81*, 541–53.

de Jong, P. (1988). A cross validation filter for time series models. *Biometrika 75*, 594–600.

de Jong, P. (1991). The diffuse Kalman filter. *Annals of Statistics 19*, 1073–83.

de Jong, P. and N. Shephard (1995). The simulation smoother for time series models. *Biometrika 82*, 339–50.

Doornik, J. A. (1998). *Object-Oriented Matrix Programming using Ox 2.0.* London: Timberlake Consultants Press.

Durbin, J. and S. J. Koopman (1997). Monte Carlo maximum likelihood estimation of non-Gaussian state space model. *Biometrika 84*, 669–84.

Durbin, J. and S. J. Koopman (2000). Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). *J. Royal Statistical Society B 62*, 3–56.

Durbin, J. and S. J. Koopman (2001). *Time Series Analysis by State Space Methods.* Oxford: Oxford University Press.

Fruhwirth-Schnatter, S. (1994a). Applied state space modelling of non-Gaussian time series using integration-based Kalman filtering. *Statistics and Computing 4*, 259–69.

Fruhwirth-Schnatter, S. (1994b). Data augmentation and dynamic linear models. *J. Time Series Analysis 15*, 183–202.

Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge: Cambridge University Press.

Harvey, A. C. and J. Durbin (1986). The effects of seat belt legislation on British road casualties: A case study in structural time series modelling, (with discussion). *J. Royal Statistical Society A 149*, 187–227.

Kohn, R. and C. F. Ansley (1989). A fast algorithm for signal extraction, influence and cross-validation. *Biometrika 76*, 65–79.

Koopman, S. J. (1993). Disturbance smoother for state space models. *Biometrika 80*, 117–26.

Koopman, S. J. (1997). Exact initial Kalman filtering and smoothing for non-stationary time series models. *J. American Statistical Association 92*, 1630–8.

Koopman, S. J., N. Shephard, and J. A. Doornik (1999). Statistical algorithms for models in state space using SsfPack 2.2. *Econometrics Journal 2*, 113–66.

Poirier, D. J. (1995). *Intermediate Statistics and Econometrics*. Cambridge: MIT.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications* (2nd ed.). New York: John Wiley & Sons.

Shephard, N. and M. K. Pitt (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika 84*, 653–67.

Table 1: Number of multiplications for a (univariate) single draw

Eqs	(1)	(3)	(4) & (5)		(3) of JS		Alg 1		Method JS	
(a)	$m{+}1$	$\frac{m^2+5m}{2}$	$m{+}1$		$m^2{+}m$		$\frac{m^2+9m+4}{2}$		$\frac{3m^2+9m+2}{2}$	
			ε	η	ε	η	ε	η	ε	η
(b)			1	m	$\frac{m^2+7m+6}{2}$	$\frac{3m^3+3m^2+8m}{2}$	1	m	$\frac{m^2+7m+8}{2}$	$\frac{3m^3+3m^2+10m}{2}$
m										
1	2	3	3	3	9	9	8	8	15	15
2	3	7	4	5	18	32	14	15	29	44
5	6	25	7	11	63	275	38	42	95	311
10	11	75	12	21	198	1800	98	107	285	1896
20	21	250	22	41	693	13100	293	312	965	13391

Number of multiplications are reported for each time period t: (a) common to both ε and η; (b) specific to ε and η. Total number is (a) plus (b). It is assumed that Z_t, T_t and R_t only contain zeros and ones and H_t and Q_t are diagonal. Reported values for equation (3) of JS are additional to equations (4) & (5). Algorithm 1 requires equations (1), (3), (4) and (5); Method JS requires equations (3) of this paper and (3) of JS.

Table 2: Posterior results for variances based on Gibbs sampler with $M = 2000$

all variances stochastic

variance	posterior mean	posterior stand.dev
σ_{ε}^2	0.003398	0.0006047
σ_{η}^2	0.001151	0.0003957
σ_{ω}^2	0.00001603	0.00002450

seasonal variance fixed at zero

variance	posterior mean	posterior stand.dev
σ_{ε}^2	0.003560	0.0005806
σ_{η}^2	0.001039	0.0003712

Table 3: Number of multiplications for (univariate) multiple draws

Eqs	(1)	(3)	(4) & (5)		(3) of JS		Alg 1		Method JS	
(a)	$m+1$	m	$m+1$		0		$3m+2$		$2m+1$	
			ε	η	ε	η	ε	η	ε	η
(b)			1	m	m	m^2	1	m	$m+1$	m^2+m
m										
1	2	1	3	3	1	1	6	6	5	5
2	3	2	4	5	2	4	9	10	8	11
5	6	5	7	11	5	25	18	22	17	41
10	11	10	12	21	10	100	33	42	32	131
20	21	20	22	41	20	400	63	82	62	461

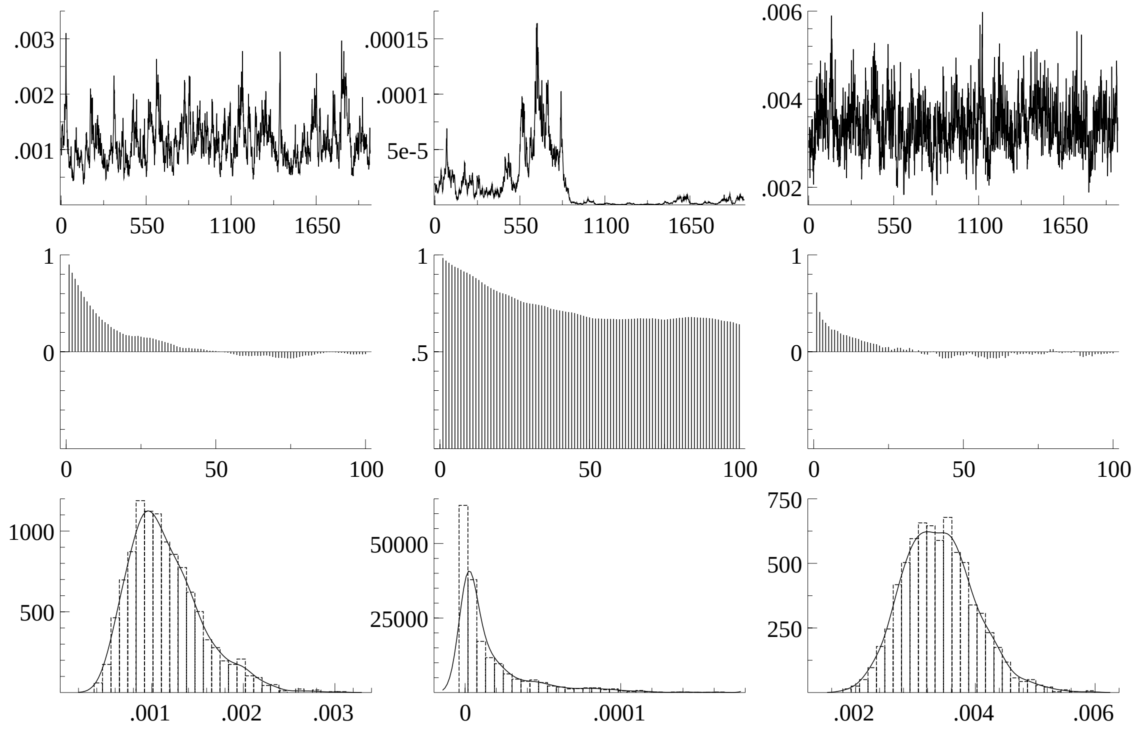
Number of multiplications are reported for each time period t: (a) common to both ε and η; (b) specific to ε and η. Total number is (a) plus (b). It is assumed that Z_t, T_t and R_t only contain zeros and ones and H_t and Q_t are diagonal. Reported values for equation (3) of JS are additional to equations (4) & (5). Algorithm 1 requires equations (1), (3), (4) and (5); Method JS requires equations (3) of this paper and (3) of JS.

Table 4: Storage space for (univariate) multiple draws

Eqs	(1)	(3)	(4) & (5)		(3) of JS		Alg 1		Method JS	
(a)	0	$m+1$	0		0		$m+1$		$m+1$	
			ε	η	ε	η	ε	η	ε	η
(b)			0	0	$m+1$	$\frac{3m^2+m}{2}$	0	0	$m+1$	$\frac{3m^2+m}{2}$
m										
1	0	2	0	0	2	2	2	2	4	4
2	0	3	0	0	3	7	3	3	6	10
5	0	6	0	0	6	40	6	6	12	46
10	0	11	0	0	11	155	11	11	22	166
20	0	21	0	0	21	610	21	21	42	631

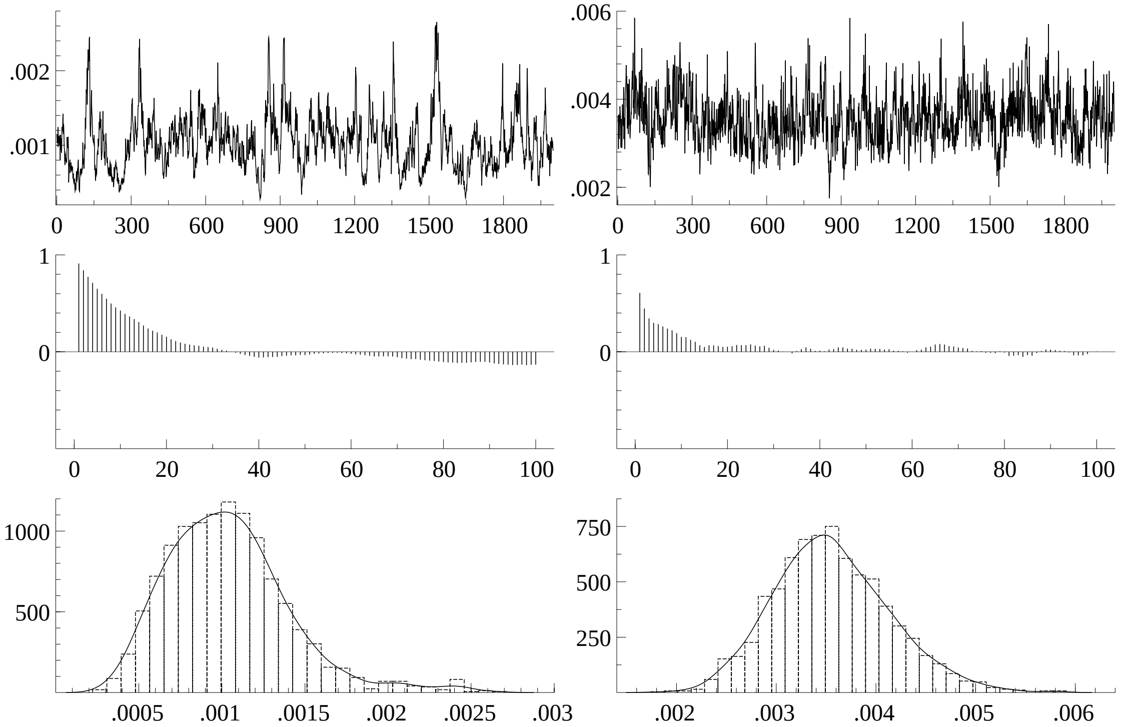
Storage space is reported for each time period t: (a) common to both ε and η; (b) specific to ε and η. Total number is (a) plus (b). It is assumed that Z_t, T_t and R_t only contain zeros and ones and H_t and Q_t are diagonal. Reported values for equation (3) of JS are additional to equations (4) & (5). Algorithm 1 requires equations (1), (3), (4) and (5); Method JS requires equations (3) of this paper and (3) of JS.

Figure 1: Gibbs sampler diagnostics for seatbelt model



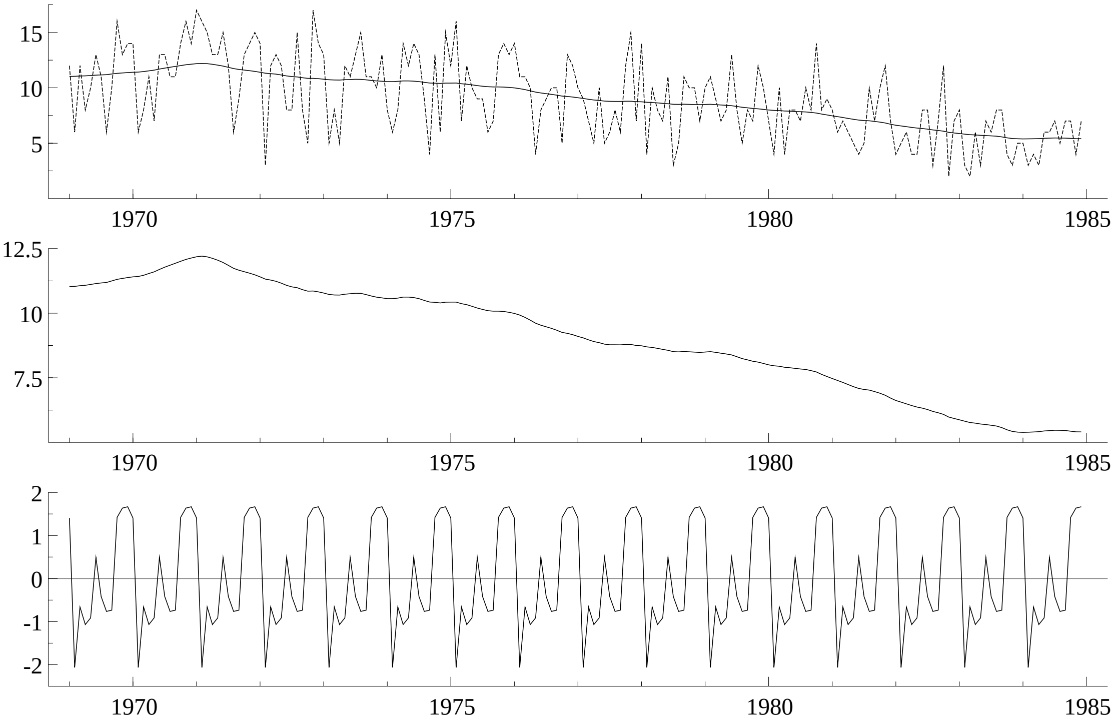
Columns represent elements of ψ and rows represent realised draws of ψ, correlogram of series of draws and histogram of draws, respectively.

Figure 2: Gibbs sampler diagnostics for seatbelt model with seasonal variance fixed at zero



Columns represent elements of ψ^* and rows represent realised draws of ψ^*, correlogram of series of draws and histogram of draws, respectively.

Figure 3: Data and estimated level and seasonal components for the van model.


15
10
5
1970
1975
1980
1985
12.5
10
7.5
1970
1975
1980
1985
2
1
0
-1
-2
1970
1975
1980
1985