


Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models

Genshiro KITAGAWA

A new algorithm for the prediction, filtering, and smoothing of non-Gaussian nonlinear state space models is shown. The algorithm is based on a Monte Carlo method in which successive prediction, filtering (and subsequently smoothing), conditional probability density functions are approximated by many of their realizations. The particular contribution of this algorithm is that it can be applied to a broad class of nonlinear non-Gaussian higher dimensional state space models on the provision that the dimensions of the system noise and the observation noise are relatively low. Several numerical examples are shown.

Key Words: Fixed interval smoothing; Non-Gaussian state space model; Nonstationary time series; Recursive filtering; Time series modeling.

1. INTRODUCTION

This article shows a Monte Carlo method for non-Gaussian nonlinear filtering and smoothing. In this method, each distribution is expressed by many of its realizations, and the trajectory of each *particle* in successive prediction stages is simulated by using the assumed model. In the filtering stage, the resampling with a weight proportional to the *likelihood* is performed to get a set of *particles* that represents the filter distribution.

The use of the state space model and the Kalman filter analysis have become very popular in time series applications within the past two decades, and the advantages of using the state space model have been widely recognized. However, with the expansion of applications—for example, in the detection of structural changes of time series models, in the analysis of time series with outliers, and in nonlinear time series modeling—the necessity of non-Gaussian state space modeling became apparent. For example, West, Harrison, and Migon (1986) introduced a generalized dynamic linear model (see also Fahrmeir 1992; Smith and Miller 1986). Various extensions of Kalman filtering have been proposed (e.g., Fahrmeir and Kaufmann 1991, Meinhold and Singpurwalla 1989, Sage and Melsa 1971, and Schnatter 1992). Also, several approximations to non-Gaussian filter have been developed (e.g., Masreliez 1975; Sage and Melsa 1971). Kitagawa (1987) directly generalized the state space model to the case where either the system noise

Genshiro Kitagawa is Professor, The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106 Japan.

Journal of Computational and Graphical Statistics, Volume 5, Number 1, Pages 1–25

or the observation noise are non-Gaussian. In the method described there, recursive formulas for filtering and smoothing were derived and they were implemented using numerical computations. It was also demonstrated that many nonstandard situations in time series analysis, such as abrupt model parameter changes, time series with outliers and with skewed distributions, can be well handled with this modeling and computational technique. The method generalizes easily to discrete distribution models and to nonlinear models (Kitagawa 1987, 1991).

On the other hand, a problem with this numerical method is that it requires intensive use of the computer, both in memory and CPU time and therefore it is difficult to apply to models with high dimensional states. Considerable work has been done on the refinement of the numerical algorithm for low-state dimensional modeling. For example, Hodges and Hale (1993) used a computationally more efficient integration algorithm and Tanizaki (1993) used a Monte Carlo random placement of knots method. In a different approach to mitigate the high-state dimensional computational difficulty, Kitagawa (1989, 1994) used a Gaussian-sum approximation, in a non-Gaussian seasonal adjustment problem, where, typically a 13-dimensional state space model was required. (The Gaussian-sum approximation was introduced for low dimensional models in Alspach and Sorenson 1972.)

In this article, we present a new direct Monte Carlo method for state space filtering and smoothing. The algorithm is based on the approximation of successive prediction and filtering density functions by many of their realizations. The difference between the present algorithm and other Monte Carlo–Gibbs sampling methods (Carlin, Polson, and Stoffer 1992; Frühwirth-Schnatter 1994), is that we use the Monte Carlo method for the entire filtering and smoothing procedures whereas the other algorithms are used only for numerical integration. The virtue of this new algorithm is that it can be applied to very wide class of nonlinear non-Gaussian higher dimensional state space models, on the provision that the dimensions of the system noise and the observation noise are relatively low—less than five, for example—by merely specifying the functions and noise densities.

2. A NON-GAUSSIAN NONLINEAR STATE SPACE MODEL AND STATE ESTIMATION

Assume that the time series y_n is obtained by the following non-Gaussian nonlinear state space model

$$x_n = F(x_{n-1}, v_n) \tag{2.1}$$

$$y_n = H(x_n, w_n), \tag{2.2}$$

where x_n is a k-dimensional state vector, the system noise v_n and the observation noise w_n are ℓ-dimensional and 1-dimensional white noise sequences with densities $q(v)$ and $r(w)$, respectively. F and H are possibly nonlinear function $\mathrm{R}^k \times \mathrm{R}^\ell \to \mathrm{R}^k$, $\mathrm{R}^k \times \mathrm{R} \to \mathrm{R}$, respectively. It is also assumed that given the state x_n and the observation y_n, the observational noise w_n is uniquely determined by $w_n = G(y_n, x_n)$, where G has a derivative as a function of y which is denoted by $\frac{\partial G}{\partial y}$. The initial state vector x_0 is assumed to be distributed according to the density $p_0(x)$. This type of state space model

contains a broad class of linear, nonlinear, stationary, or nonstationary time series models. Some examples are given in Section 5.

The most important problem in state space modeling is the estimation of the state x_n from the observations. Many problems in time series such as the likelihood computation for parameter estimation and the prediction, the interpolation, and the decomposition of a stationary or nonstationary time series can be handled by estimating the state.

The problem of state estimation can be formulated as an evaluation of the conditional density $p(x_n|Y_t)$, where Y_t is the set of observations $\{y_1, \ldots, y_t\}$. Corresponding to the three distinct cases, $n > t$, $n = t$ and $n < t$, the state estimation problem can be classified into three corresponding categories where the conditional density $p(x_n|Y_t)$ is called the predictor, the filter, and the smoother, respectively.

For the standard linear-Gaussian state space model, each density can be expressed by a Gaussian density and its mean vector and the variance-covariance matrix can be obtained by computationally efficient recursive formulas such as the Kalman filter and smoothing algorithms (Anderson and Moore 1979).

For nonlinear or non-Gaussian state space models, however, the conditional distributions are non-Gaussian and various types of approximations to or assumptions on the densities are used to obtain recursive formulas for state estimation. Some of the examples are the extended Kalman filter (Anderson and Moore 1979), the Gaussian-sum filter (Alspach and Sorenson 1972), the dynamic generalized linear model (West, Harrison, and Migon 1985), and the non-Gaussian filter and smoother (Hodges and Hale 1993; Kitagawa 1987; Tanizaki 1993).

In this article, we take a different approach. We approximate or express each density function by many of the realizations from that distribution. Specifically, assume that each distribution is expressed by using m (say 1,000 or 10,000) particles as follows:

$$\begin{array}{lcll} \{p_n^{(1)}, \ldots, p_n^{(m)}\} & \sim & p(x_n|Y_{n-1}) & \text{predictor} \\ \{f_n^{(1)}, \ldots, f_n^{(m)}\} & \sim & p(x_n|Y_n) & \text{filter} \\ \{s_{n|N}^{(1)}, \ldots, s_{n|N}^{(m)}\} & \sim & p(x_n|Y_N) & \text{smoother.} \end{array}$$

In effect we approximate the distributions by the empirical distributions determined by the set of particles. For example, when m independent realizations $p_n^{(1)}, \ldots, p_n^{(m)}$ from $p(x_n|Y_{n-1})$ are given, the distribution function of $p(x_n|Y_{n-1})$ can be approximated by the empirical distribution function

$$P_n(x) = \frac{1}{m} \sum_{j=1}^{m} I(x, p_n^{(j)}), \tag{2.3}$$

where $I(x, a)$ is the indicator function defined by $I(x, a) = 0$ if $x < a$ and $I(x, a) = 1$ otherwise. This means that $p(x_n|Y_{n-1})$ is approximated by the probability function

$$\Pr(x_n = p_n^{(j)}|Y_{n-1}) = \frac{1}{m}, \qquad \text{for } j = 1, \ldots, m. \tag{2.4}$$

It will be shown that a set of realizations expressing the one step ahead predictor $p(x_n|Y_{n-1})$ and the filter $p(x_n|Y_n)$ can be obtained recursively. Namely, $\{p_n^{(1)}, \ldots, p_n^{(m)}\}$ is obtained from $\{f_{n-1}^{(1)}, \ldots, f_{n-1}^{(m)}\}$ and then $\{f_n^{(1)}, \ldots, f_n^{(m)}\}$ is from $\{p_n^{(1)}, \ldots, p_n^{(m)}\}$.

3. MONTE CARLO FILTERING

In the following two sections, we show a Monte Carlo based algorithm for recursive evaluation of the predictor and the filter, separately. Two different algorithms for smoothing are discussed in Section 4.

3.1 ONE STEP AHEAD PREDICTION

In this subsection, we show that a set of m realizations $\{p_n^{(1)}, \ldots, p_n^{(m)}\}$ can be obtained from $\{f_{n-1}^{(1)}, \ldots, f_{n-1}^{(m)}\}$, a set of realizations of $p(x_{n-1}|Y_{n-1})$. Let $\{v_n^{(1)}, \ldots, v_n^{(m)}\}$ be independent realizations of the system noise v_n. Namely, for $j = 1, \ldots, m$

$$f_{n-1}^{(j)} \sim p(x_{n-1}|Y_{n-1}), \quad v_n^{(j)} \sim q(v). \tag{3.1}$$

Here the predictive distribution $p(x_n|Y_{n-1})$ can be expressed by

$$\begin{aligned} p(x_n|Y_{n-1}) &= \int\int p(x_n, x_{n-1}, v_n|Y_{n-1})dv_n dx_{n-1} \\ &= \int\int p(x_n|x_{n-1}, v_n, Y_{n-1})p(v_n|x_{n-1}, Y_{n-1})p(x_{n-1}|Y_{n-1})dv_n dx_{n-1} \\ &= \int\int p(x_n|x_{n-1}, v_n)p(v_n)p(x_{n-1}|Y_{n-1})dv_n dx_{n-1} \\ &= \int\int \delta(x_n - F(x_{n-1}, v_n))p(v_n)p(x_{n-1}|Y_{n-1})dv_n dx_{n-1}, \end{aligned} \tag{3.2}$$

with $\delta(x)$ being the delta function. Therefore, if we define $p_n^{(j)}$, the jth particles, by

$$p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)}), \tag{3.3}$$

$\{p_n^{(1)}, \ldots, p_n^{(m)}\}$ can be considered as independent realizations of the one step ahead predictor density, $p(x_n|Y_{n-1})$. We note that since $f_{n-1}^{(j)}$ and $v_n^{(j)}$ are independent realization from $p(x_{n-1}|Y_{n-1})$ and $p(v_n)$, respectively the prediction formula is valid even for a non-linear model.

3.2 FILTERING

Given the observation y_n and the particle, $p_n^{(j)}$, compute $\alpha_n^{(j)}$, the likelihood of the particle $p_n^{(j)}$ based on the observation y_n. That is,

$$\alpha_n^{(j)} = p\left(y_n|p_n^{(j)}\right) = r\left(G\left(y_n, p_n^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_n}\right|$$

for $j = 1, \ldots, m$. Note that G is the inverse function of H, and r is the density of the observation noise w.

Next, we obtain m particles $\{f_n^{(1)}, \ldots, f_n^{(m)}\}$ by the resampling of $\{p_n^{(1)}, \ldots, p_n^{(m)}\}$, with the probabilities proportional to $\alpha_n^{(1)}, \ldots, \alpha_n^{(m)}$, respectively. Namely, define $f_n^{(j)}$

by

$$f_n^{(j)} = \begin{cases} p_n^{(1)} & \text{with probability } \alpha_n^{(1)} / \left(\alpha_n^{(1)} + \cdots + \alpha_n^{(m)}\right) \\ \vdots & \vdots \\ p_n^{(m)} & \text{with probability } \alpha_n^{(m)} / \left(\alpha_n^{(1)} + \cdots + \alpha_n^{(m)}\right). \end{cases} \tag{3.4}$$

Then $\{f_n^{(1)}, \ldots, f_n^{(m)}\}$ can be considered as the realizations of the filter, $p(x_n|Y_n)$.

This can be verified as follows. Given the observation y_n the posterior probability is obtained by

$$\begin{aligned} \Pr(x_n = p_n^{(i)}|Y_n) &= \Pr(x_n = p_n^{(i)}|Y_{n-1}, y_n) \\ &= \lim_{\Delta y \to 0} \frac{\Pr\left(x_n = p_n^{(i)}, y_n \le y \le y_n + \Delta y|Y_{n-1}\right)}{\Pr(y_n \le y \le y_n + \Delta y|Y_{n-1})} \\ &= \frac{p\left(y_n|p_n^{(i)}\right) \Pr\left(x_n = p_n^{(i)}|Y_{n-1}\right)}{\sum_{j=1}^m p\left(y_n|p_n^{(j)}\right) \Pr\left(x_n = p_n^{(j)}|Y_{n-1}\right)} \\ &= \frac{\alpha_n^{(i)} \cdot \frac{1}{m}}{\sum_{j=1}^m \alpha_n^{(j)} \cdot \frac{1}{m}} = \frac{\alpha_n^{(i)}}{\sum_{j=1}^m \alpha_n^{(j)}}. \end{aligned} \tag{3.5}$$

This means that the probability distribution function associated with $\Pr(x_n = p_n^{(i)}|Y_n)$ can be expressed by a step function

$$\frac{1}{\sum_{j=1}^m \alpha_n^{(j)}} \sum_{i=1}^m \alpha_n^{(i)} I\left(x, p_n^{(i)}\right) \tag{3.6}$$

which has jumps only at $p_1, \ldots, p_m$ with step sizes proportional to $\alpha_n^{(1)}, \ldots, \alpha_n^{(m)}$.

For the next step of prediction, it is necessary to represent this distribution function by an empirical distribution of the form

$$\frac{1}{m} \sum_{i=1}^m I\left(x, f_n^{(i)}\right). \tag{3.7}$$

This can be done by generating m realizations $\{f_n^{(1)}, \ldots, f_n^{(m)}\}$ by the resampling of $\{p_n^{(1)}, \ldots, p_n^{(m)}\}$ with probabilities

$$\Pr(f_n^{(i)} = p_n^{(j)}|Y_n) = \frac{\alpha_n^{(j)}}{\alpha_n^{(1)} + \cdots + \alpha_n^{(m)}}, \quad \text{for } i = 1, \ldots, m. \tag{3.8}$$

The details of the resampling scheme and some modifications are discussed in the Appendix.

3.3 AN ALGORITHM FOR FILTERING

Summarizing the previous two subsections, we obtain the following recursive algorithm for one step ahead prediction and filter:

1. Generate a k-dimensional random number $f_0^{(j)} \sim p_0(x)$ for $j = 1, \ldots, m$.
2. Repeat the following steps for $n = 1, \ldots, N$.
   (a) Generate an ℓ-dimensional random number $v_n^{(j)} \sim q(v)$ for $j = 1, \ldots, m$.
   (b) Compute $p_n^{(j)} = F\left(f_{n-1}^{(j)}, v_n^{(j)}\right)$ for $j = 1, \ldots, m$.
   (c) Compute $\alpha_n^{(j)} = r\left(G\left(y_n, p_n^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_n}\right|$ for $j = 1, \ldots, m$.
   (d) Generate $f_n^{(j)} \sim \left(\sum_{i=1}^m \alpha_n^{(i)}\right)^{-1} \sum_{i=1}^m \alpha_n^{(i)} I(x, p_n^{(i)})$ for $j = 1, \ldots, m$ by the resampling of $p_n^{(1)}, \ldots, p_n^{(m)}$.

3.4 LIKELIHOOD OF THE MODEL

The state space model, (2.1) and (2.2), usually contains several unknown parameters such as the variances of the noises and the coefficients of the functions F and H. The vector consisted of such unknown parameters is denoted by θ.

Given the observations $y_1, \ldots, y_N$, the likelihood of the parameter θ of the model is obtained by

$$L(\theta) = p(y_1, \ldots, y_N|\theta) = \prod_{n=1}^N p(y_n|y_1, \ldots, y_{n-1}, \theta) = \prod_{n=1}^N p(y_n|Y_{n-1}), \tag{3.9}$$

where $p(y_1|Y_0) = p_0(y_1)$. Therefore, by using the approximation,

$$\begin{aligned} p(y_n|Y_{n-1}) &= \int p(y_n|x_n)p(x_n|Y_{n-1})dx_n \\ &\cong \frac{1}{m}\sum_{j=1}^m p(y_n|p_n^{(j)}) \\ &= \frac{1}{m}\sum_{j=1}^m \alpha_n^{(j)}, \end{aligned} \tag{3.10}$$

the log-likelihood can be approximated by

$$\ell(\theta) = \sum_{n=1}^N \log p(y_n|Y_{n-1}) \cong \sum_{n=1}^N \log\left(\sum_{j=1}^m \alpha_n^{(j)}\right) - N\log m. \tag{3.11}$$

The maximum likelihood estimate $\hat{\theta}$ of the parameter vector θ can be estimated by maximizing the log-likelihood.

If there are several candidate models, the goodness of the fit of each model is evaluated by the AIC criterion (Akaike 1973; Sakamoto, Ishiguro, and Kitagawa 1986)

$$\text{AIC} = -2\ell(\hat{\theta}) + 2(\text{number of parameters}).$$

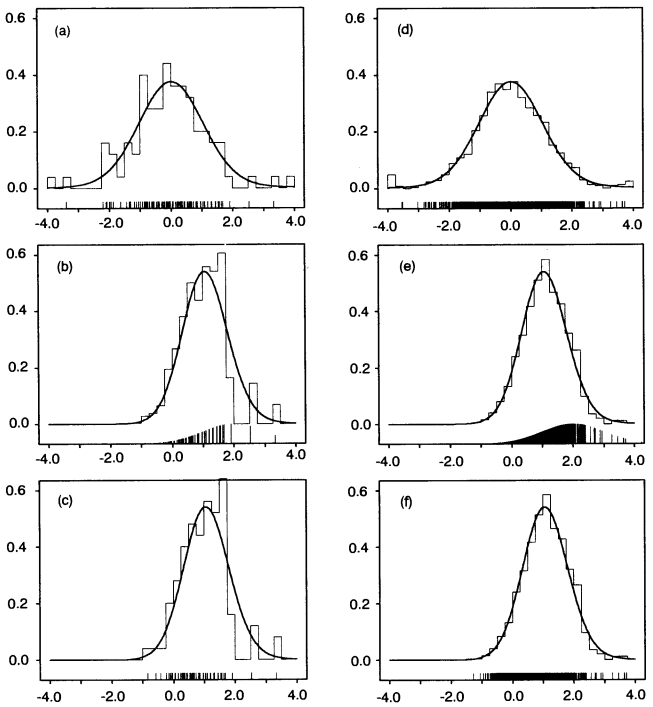

Figure 1. One Step of the Monte Carlo Filter. Bold curve = exact density, step function: histogram obtained from the particles. Tick mark = location of the particle. (a) Predictor, m = 100; (b) filter, m = 100; (c) resampled filter, m = 100; (d) predictor, m = 1,000; (e) filter, m = 1,000; (f) resampled filter, m = 1,000.

3.5 ILLUSTRATIVE EXAMPLES

To illustrate how the one step ahead predictor and the filter distributions are expressed by the particles, first, one cycle of the prediction and the filter steps is described (e.g., $n = 1$). Consider a one-dimensional non-Gaussian state space model

$$\begin{aligned} x_n &= x_{n-1} + v_n \\ y_n &= x_n + w_n, \end{aligned} \tag{3.12}$$

where v_n and w_n are white noises distributed as the Cauchy distribution, Cauchy(0,.01), with density $q(v) = .1\pi^{-1}(v^2 + .01)^{-1}$, and the standard normal distribution, $N(0, 1)$, respectively. We further assume that the initial distribution $p_0(x_0)$ is also the standard normal distribution, $N(0, 1)$. This assumption of normality is not essential for the Monte Carlo filter. Under these assumptions, we now consider how the filter distribution $p(x_1|Y_1)$ is obtained from the predictive distribution $p(x_1|Y_0)$ when the observation $y_1 = 2$. The number of particles was set to $m = 100$ in one case and to 1,000 in another case. $m = 100$ is for illustrative purpose and is smaller than what we usually use in actual computation.

Tick marks in Figure 1a show the 100 realizations of $p(x_1|Y_0)$ directly computed by (3.3). The histogram obtained from these realizations approximates the "exact" density

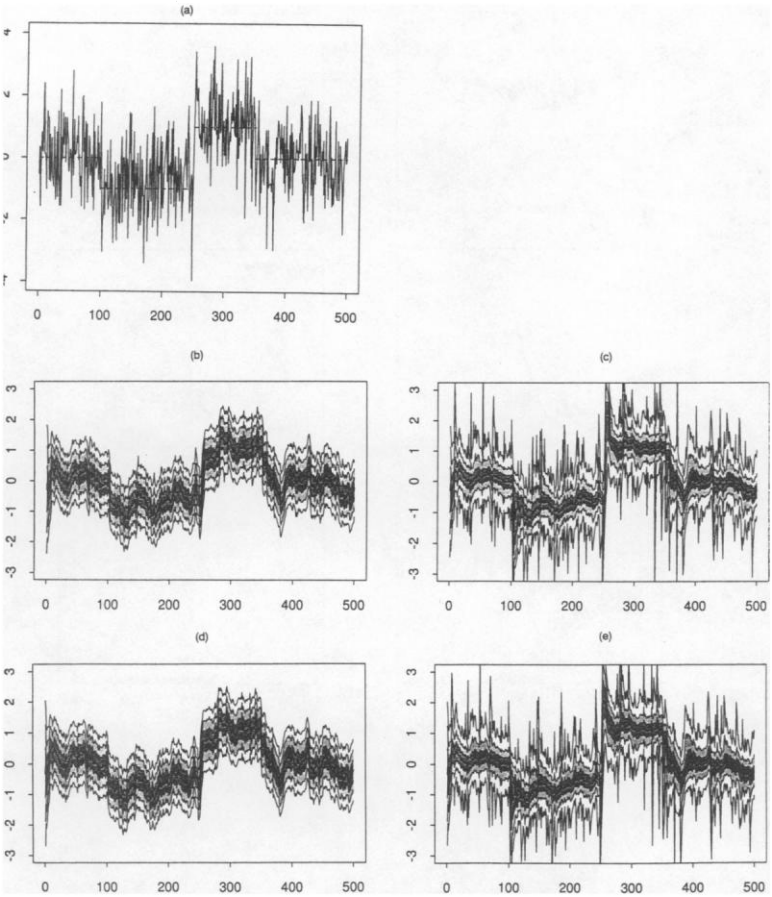

Figure 2. (a) Test data and the true trend function. (b)–(e) Posterior distributions of the filter. Bold curve = median; fine curve = .13%, 2.3%, 15.9%, 84.1%, 97.7%, and 99.87% points. (b) Monte Carlo filter with m = 10,000, Gaussian model; (c) Monte Carlo filter with m = 10,000, Cauchy model; (d) exact distribution obtained by the Kalman filter for the Gaussian model; (e) exact distribution obtained by the Cauchy filter for the non-Gaussian model.

function (bold curve) obtained by the numerical convolution of two densities $q(v)$ and $p_0(x_0)$. Figure 1b shows the filtered density $p(x_n|Y_n)$ and its Monte Carlo approximation. The locations of the 100 realizations in Figure 1b are the same as the ones in Figure 1a. However, the height of the jth each tick mark is proportional to $\alpha_n^{(j)}$, and the corresponding cumulative distribution function defined by (3.6) approximates the filtered density and is quite different from the one in Figure 1a, which has equal probability steps. Figure 1c shows 100 realizations obtained by the resampling of the data shown in Figure 1b according to the probability given by (3.8). It can be seen that the histogram shown in Figure 1c closely resembles the distribution in Figure 1b and approximates the exact one shown by the bold curve.

Figures 1d, 1e, and 1f show the case when the number of particles $m = 1{,}000$. Closer approximations to the exact densities are obtained when $m = 1{,}000$.

For a more complete illustrative example, we consider the problem shown in Kita-

Table 1. The Effect of the Number of Particles, m

m	D_1	D_2
100	.1060	.0471
200	.0544	.0372
400	.0276	.0281
800	.0145	.0201
1600	.0072	.0171
3200	.0038	.0096
6400	.0023	.0098
12800	.0011	.0060
25600	.0007	.0048
51200	.0005	.0029

gawa (1987, sec. 5.1). In that problem, a known abruptly changing mean value time series (or trend) is observed in the presence of additive Gaussian noise. A trend model establishes a prior distribution on the trend parameters, $t_1, \ldots, t_N, N = 500$. The problem is to obtain the smooth estimate of the trend. The quasi-Bayesian method of analysis yields the marginal posterior distribution of the trend. The data is analyzed by two different models for noise inputs. The first is a Gaussian system and Gaussian observation noise. The second is a Cauchy system and Gaussian observation noise. Figure 2a shows the known trend (dotted lines) and the observed trend plus noise data.

For the estimation of the trend by the Gaussian model we use

$$\begin{aligned} t_n &= t_{n-1} + v_n, \qquad v_n \sim N(0, 1.22 \times 10^{-2}) \\ y_n &= t_n + w_n, \qquad w_n \sim N(0, 1.043). \end{aligned} \tag{3.13}$$

The number of realizations, m, is set to 10,000.

Figure 2b shows the posterior mean (bold curve) and $\pm 1\sigma$, 2σ, 3σ intervals (fine curves) of the Monte Carlo filter for the Gaussian model. On the other hand, Figure 2c shows the Monte Carlo estimates obtained by a non-Gaussian model

$$v_n \sim \text{Cauchy}(0, 3.48 \times 10^{-5}), \quad w_n \sim N(0, 1.022). \tag{3.14}$$

In the figure, the bold curve shows the 50% points of the filter densities and the other six fine curves show .13%, 2.3%, 15.9%, 84.1%, 97.7%, and 99.87% points, respectively. These points correspond to the ± 1, 2, 3 standard errors in Gaussian distributions. Figures 2d and 2e show the exact estimates obtained by the Kalman filter and the non-Gaussian (numerical integration) filter described in Kitagawa (1987), respectively. It can be seen that the Monte Carlo filter densities agree with the ones obtained by the exact filter. In particular, in the Gaussian case, the two results are visually almost indistinguishable. It is remarkable that, in the Gaussian case, even with $m = 10$, we get a reasonable approximation to the exact posterior mean.

Table 1 shows the effects of the selection of the number of particles m on the approximation. The second column shows the discrepancy between the exact filter distribution and the Monte Carlo distribution measured by $D_1 = \sum_{n=1}^{N} \sum_{i=1}^{K} (g_{in} - f_{in})^2 g_{in}$, where g_{in} and f_{in} are relative frequencies obtained from the exact and Monte Carlo distributions, respectively, and K is the number of bins. On the other hand D_2 shows the average

Table 2. The Effect of the Initial Distribution

	Log-likelihood	
Initial distribution	*Normal*	*Cauchy*
$N(0,1)$	-748.20 (.30)	-742.11 (.18)
$U[-4,4]$	-749.25 (.26)	-743.27 (.25)
Cauchy(0,1/4)	-748.29 (.37)	-742.38 (.23)
δ_0	-747.08 (.27)	-740.89 (.18)
δ_2	-761.27 (.38)	-748.75 (.42)

difference of the means of the exact and Monte Carlo distributions. From the table, it is seen that the average difference of the means is less than .01 for $m \geq 3{,}200$. As a rule of thumb, $m = 10{,}000$ is recommended for the estimation of the filter distribution. However, if only the mean value is required, $m = 1{,}000$ is often sufficient.

We also considered how the selection of the initial state distribution affects the filtering. Table 2 shows the mean of the log-likelihoods obtained from 100 different random numbers for five distinct distributions: normal, uniform, Cauchy distributions, and two δ functions concentrated at $x_0 = 0$ and 2, respectively. The variances of the log-likelihoods in 100 runs are shown in parentheses. One can see that, for the case of alternative continuous distributions, the log-likelihood values are rather insensitive to the selection of the particular initial distribution. Obtained posterior distributions are visually indistinguishable except for initial several steps. On the other hand, if the initial state is concentrated on one point, the likelihood value and posterior distribution are very sensitive to its location of the delta function.

4. MONTE CARLO SMOOTHING

The Monte Carlo filter idea can be generalized to achieve smoothing. In the following two subsections, we present two different algorithms for smoothing.

4.1 SMOOTHING BY STORING THE STATE VECTOR

The first algorithm is just a simple modification of the filter shown in Section 3. In this section, $\left(s^{(j)}_{1|i}, \ldots, s^{(j)}_{n|i}\right)^t$ denotes the jth realization of the conditional joint density $p(x_1, \ldots, x_n|Y_i)$.

Assume that $\Pr\left(x_1 = s^{(j)}_{1|n-1}, \ldots, x_{n-1} = s^{(j)}_{n-1|n-1}|Y_{n-1}\right) = 1/m$ and $v^{(j)}_n \sim q(v)$, and define $(p^{(j)}_{1|n-1}, \ldots, p^{(j)}_{n|n-1})^t$ by

$$p^{(j)}_{i|n-1} = \begin{cases} s^{(j)}_{i|n-1} & \text{for } i = 1, \ldots, n-1 \\ F(s^{(j)}_{n-1|n-1}, v^{(j)}_n) & \text{for } i = n. \end{cases} \tag{4.1}$$

Then $\left(p^{(j)}_{1|n-1}, \ldots, p^{(j)}_{n|n-1}\right)^t$ can be considered as a realization from the conditional joint distribution of $(x_1, \ldots, x_n)^t$ given the observations Y_{n-1}.

Next, given the observation y_n, the distribution

$$\Pr(x_1 = p^{(j)}_{1|n-1}, \ldots, x_n = p^{(j)}_{n|n-1}|Y_{n-1})$$

is updated as follows:

$$\begin{aligned}
\Pr(x_1 &= p^{(j)}_{1|n-1}, \ldots, x_n = p^{(j)}_{n|n-1}|Y_n) \\
&= \Pr(x_1 = p^{(j)}_{1|n-1}, \ldots, x_n = p^{(j)}_{n|n-1}|Y_{n-1}, y_n) \\
&= \frac{p(y_n|x_1 = p^{(j)}_{1|n-1}, \ldots, x_n = p^{(j)}_{n|n-1}, Y_{n-1})\Pr(x_1 = p^{(j)}_{1|n-1}, \ldots, x_n = p^{(j)}_{n|n-1}|Y_{n-1})}{p(y_n|Y_{n-1})} \\
&= \frac{p(y_n|p^{(j)}_{n|n-1})\Pr(x_1 = p^{(j)}_{1|n-1}, \ldots, x_n = p^{(j)}_{n|n-1}|Y_{n-1})}{p(y_n|Y_{n-1})}. \qquad (4.2)
\end{aligned}$$

Since $p^{(j)}_{n|n-1}$ is identical to the $p^{(j)}_n$ used in the filtering algorithm (3.5), this indicates that realization of the fixed interval smoother, $p(x_1, \ldots, x_n|Y_n)$ can be obtained by storing and resampling m sets of vector realizations $(p^{(j)}_{1|n-1}, \ldots, p^{(j)}_{n|n-1})^t$, $j = 1, \ldots, m$, with the same probability as for the filtering.

Therefore, an algorithm for smoothing is obtained by replacing the Step 2d of the algorithm for filtering shown in section 3.3 with

(d-S) Generate $\{(s^{(j)}_{1|n}, \ldots, s^{(j)}_{n-1|n}, s^{(j)}_{n|n})^t,\ j = 1, \ldots, m\}$ by resampling $\{(s^{(j)}_{1|n-1}, \ldots, s^{(j)}_{n-1|n-1}, p^{(j)}_n)^t,\ j = 1, \ldots, m\}$.

In this modification, the past trajectories $s^{(j)}_{1|n-1}, \ldots s^{(j)}_{n-1|n-1}$, are preserved and the set $\{(s^{(j)}_{1|n-1}, \ldots, s^{(j)}_{n-1|n-1}, p^{(j)}_n)^t, j = 1, \ldots, m\}$ is resampled with the same weights as the one obtained by Step 2d of Section 3.3. This algorithm realizes fixed interval smoothing for the nonlinear non-Gaussian state space model. In practice, however, because the number of realizations is finite, the repetition of the resampling (d-S) will gradually decrease the number of different realizations in $\left\{s^{(1)}_{i|n}, \ldots, s^{(m)}_{i|n}\right\}$ and the shape of the distribution will deteriorate.

In Figures 3a–f, the fine curve shows the fixed point smoother, $p(x_1|Y_n)$, for six distinct time points $n = 1$, 5, 10, 20, 100, and 500, respectively. In comparison with the final smoothed estimate shown by the bold curve, $p(x_1|Y_{500})$, it can be seen that $p(x_1|Y_n)$ quickly converges to $p(x_1|Y_{500})$.

The histograms of the particles $\{s^{(1)}_{1|n}, \ldots, s^{(m)}_{1|n}\}$ generated by the Monte Carlo smoother with $m = 1{,}000$ are also shown. For $n = 1$ shown in Figure 3a, the histogram is a reasonable approximation to $p(x_1|Y_1)$. However, for larger n, such as $n = 100$ and 500, the histogram is significantly different from $p(x_1|Y_n)$. In fact, the histogram of Figure 3f is concentrated on a single bin. In Figure 3g, the fine curve shows the change of a measure of the discrepancy between $p(x_1|Y_n)$ and $p(x_1|Y_{500})$ as n increases from 1 to 10 0 (see (A.1) in the Appendix). It rapidly converges to 0. On the other hand, the bold curve shows discrepancy between $p(x_1|Y_n)$ and the empirical distribution defined by $\left\{s^{(1)}_{1|n}, \ldots, s^{(m)}_{1|n}\right\}$. Due to the overlap of the realizations, it increases as n increases. In one case, the number of different realizations, which was originally 1000 at $n = 1$, reduced to only 84 after 50 steps and to 51 at $n = 100$.

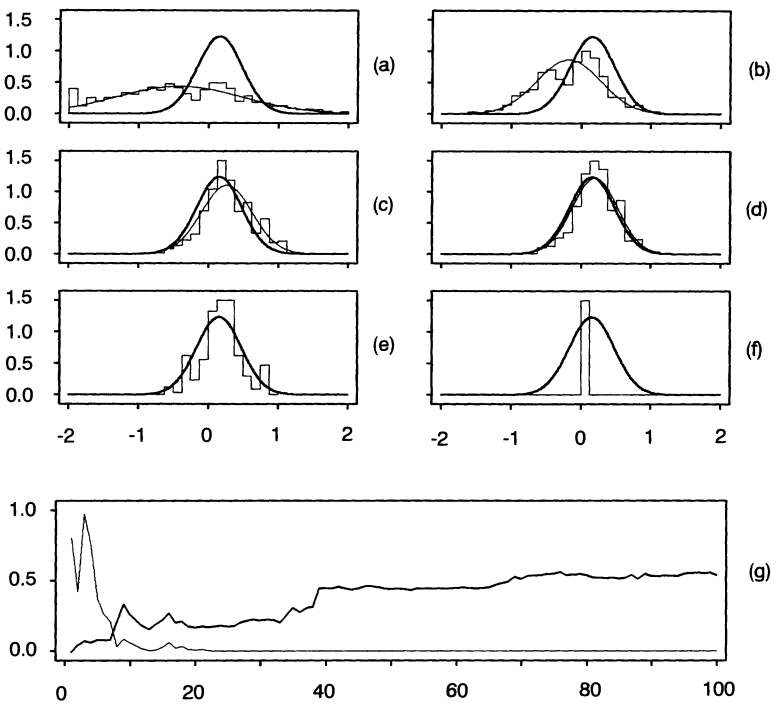

Figure 3. (a)–(f) Bold curve = $p(x_l|Y_{500})$; fine curve = $p(x_l|Y_n)$, step function: histogram obtained from $\{s^{(1)}_{l|n}, \ldots, s^{(m)}_{l|n}\}$. m = 1,000. (a) n = 1, (b) n = 5, (c) n = 10, (d) n = 20, (e) n = 100, (f) n = 500. (g) Fine curve = distance between $p(x_l|Y_n)$ and $p(x_l|Y_{500})$; dashed curve = distance between $p(x_l|Y_n)$ and $\{s^{(1)}_{l|n}, \ldots, s^{(m)}_{l|n}\}$.

Considering this, the step (d-S) needs to be modified to as follows:

(d-L) For fixed L, generate $\{(s^{(j)}_{n-L|n}, \ldots, s^{(j)}_{n-1|n}, s^{(j)}_{n|n}),\ j = 1, \ldots, m\}$ by the resampling of $\{(s^{(j)}_{n-L|n-1}, \ldots, s^{(j)}_{n-1|n-1}, p^{(j)}_n),\ j = 1, \ldots, m\}$ with $f^{(j)}_n = s^{(j)}_{n|n}$.

This is equivalent to applying the L-lag fixed lag smoother rather than the fixed interval smoother (Anderson and Moore 1979). The increase of lag, L, will improve the accuracy of the $p(x_n|Y_{n+L})$ as an approximation to $p(x_n|Y_N)$, while it is very likely to decrease the accuracy of $\left\{s^{(1)}_{n|N}, \ldots, s^{(m)}_{n|N}\right\}$ as representatives of $p(x_n|Y_{n+L})$. Because $p(x_n|Y_{n+L})$ usually converges quickly to $p(x_n|Y_N)$, it is recommended to take L not so large (say, 10 or 20, at the largest 50).

It is also advantageous in storage. The original algorithm requires $m \times k \times N$ storage and thus requires huge memory for large sample size N. On the other hand, the modified fixed-lag smoother algorithm needs only $m \times k \times L$ storage and is independent of the sample size N. For example, for m = 10,000, k = 10, N = 1,000 the necessary storage is 100M; necessary storage is 1M when L = 10.

Figures 4a and b show the exact fixed interval smoother, $p(x_n|Y_N)$, obtained by the Kalman smoother and the non-Gaussian smoother, respectively. The estimated curves

are much smoother than the ones by the filters shown in Figure 2 and the non-Gaussian estimate shown in Figure 4b detects the jumps more clearly than the ones in Figures 2b and 2d.

Figure 4 also shows the Monte Carlo 10-lag smoother obtained by using $m = 100$ (Figs. 4c and 4d), $m = 1{,}000$ (Figs. 4e and 4f), and $m = 10{,}000$ (Figs. 4g and 4h) particles. Note that for $m = 100$ and 1,000, the outermost .13% and 99.87% points are meaningless or very unreliable and only the inner 5 curves are shown. These figures suggest that, to get good approximations of the posterior distributions, we need at least $m = 1{,}000$ particles.

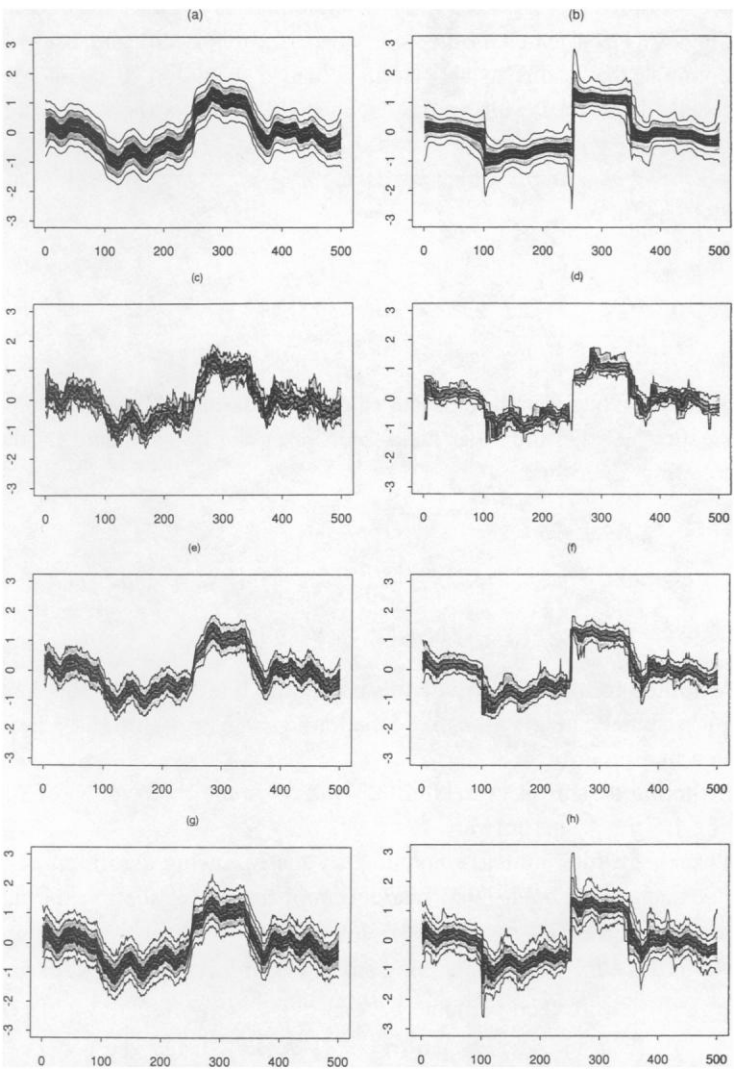

Figure 4. Posterior Distribution of the Smoother. Bold curve = median; fine curve = .13%, 2.3%, 15.9%, 84.1%, 97.7%, 99.87% points. (a) Exact distribution by the fixed interval smoother for the Gaussian model. (b) Exact distribution by the non-Gaussian smoother for the Cauchy model. (c)–(h) Monte Carlo smoother with lag L = 10. (c) m = 100, Gaussian; (d) m = 100, Cauchy; (e) m = 1,000, Gaussian; (f) m = 1,000, Cauchy; (g) m = 10,000, Gaussian; (h) m = 10,000, Cauchy.

One way to get a closer approximation to the smoothed distribution is to generate d random numbers $v_n^{(j,i)} \sim q(v)$, $i = 1, \ldots, d$, for each $f_{n-1}^{(j)}$. In smoothing, this may help to avoid a significant reduction of the different realizations. Figures 5a and 5b show, respectively, the 10-lag fixed lag smoothed estimate with $m = 10{,}000$, $L = 20$, and $d = 10$ for Gaussian and Cauchy system noise models. Closer approximations to the exact smoothed estimates shown in Figures 4a and 4b are obtained.

4.2 SMOOTHING BY THE TWO-FILTER FORMULA

Another way of achieving fixed interval smoothing is to apply the two-filter formula used in Kitagawa (1994). In this method, we perform forward and backward filtering and then combine the results to get the smoothed distribution. It is interesting that in combining two filters, the result by the backward filter plays the same role as the one point likelihood, $p(y_n|x_{n-1})$, in the forward filter.

Define $Y_N^n = \{y_n, \ldots, y_N\}$, then we have $Y_N = Y_{n-1} \cup Y_N^n$. Therefore, the smoother is divided into as follows:

$$\begin{aligned} p(x_n|Y_N) &= p(x_n|Y_{n-1}, Y_N^n) \\ &\propto p(x_n, Y_N^n|Y_{n-1}) \\ &= p(Y_N^n|x_n)p(x_n|Y_{n-1}). \end{aligned} \tag{4.3}$$

Here $p(x_n|Y_{n-1})$ is the forward filter and can be approximated by the Monte Carlo filter shown in Section 3.3. On the other hand, $p(Y_N^n|x_n)$ can be evaluated by the following backward filtering algorithm:

$$\begin{aligned} p(Y_N^N|x_N) &= p(y_N|x_N) \\ p(Y_N^{n+1}|x_n) &= \int p(Y_N^{n+1}|x_{n+1})p(x_{n+1}|x_n)dx_{n+1} \\ p(Y_N^n|x_n) &= p(y_n|x_n)p(Y_N^{n+1}|x_n). \end{aligned} \tag{4.4}$$

The two-filter formula (4.3) and the backward filtering (4.4) can be realized as follows: We assume that given x the function $z = F(x, v)$ in (2.1) has an inverse function such that $v = \gamma(z, x)$.

1. Do filtering as shown in Section 3.3 and store the realizations of the predictor, $(p_n^{(j)},\ j = 1, \ldots, m)$ for $n = 1, \ldots, N$.
2. Do backward filtering and smoothing by the following algorithm
   (a) Generate a k-dimensional random numbers $c_N^{(j)} \sim p_N(x)$, the unconditional distribution of x_N, and set $\delta_N^{(j)} = 1$ for $j = 1, \ldots, m$.
   (b) Repeat the following smoothing and backward filtering steps for $n = N, \ldots, 1$.
      i. If $n < N$ then compute $\delta_n^{(j)} = \frac{1}{d'} \sum_{i=1}^{d'} q\left(\gamma\left(c_{n+1}^{(j_i)}, p_n^{(j)}\right)\right) \beta_{n+1}^{(j_i)}$, where j_i is an integer selected from $\{1, \ldots, m\}$. It may be generated randomly or deterministically.
      ii. Compute $\beta_n^{(j)} = \delta_n^{(j)} r\left(G\left(y_n, p_n^{(j)}\right)\right)\left|\frac{\partial G}{\partial y_n}\right|$, for $j = 1, \ldots, m$.
      iii. Generate, $u_n^{(1)}, \ldots, u_n^{(m)}$ by the resampling of $p_n^{(1)}, \ldots, p_n^{(m)}$ with the weights proportional to $\beta_n^{(1)}, \ldots, \beta_n^{(m)}$.

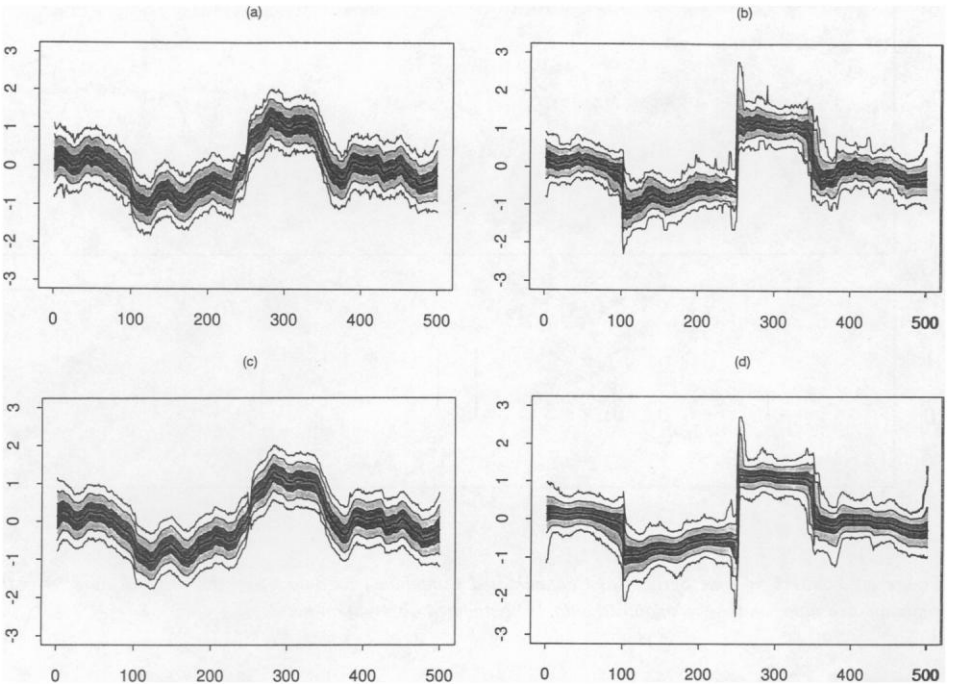

Figure 5. (a) Posterior distribution by the 10-lag Monte Carlo smoother with d = 10, Gaussian model. (b) Posterior distribution by the 10-lag Monte Carlo smoother with d = 10, Cauchy model. (c) Posterior distribution by the Monte Carlo smoother based on the two-filter formula, Gaussian model. (d) Posterior distribution by the Monte Carlo smoother based on the two-filter formula, Cauchy model.

iv. For the next step generate an ℓ-dimensional random number $w_n^{(j)} \sim q(v)$ and compute $e_n^{(j)} = \gamma(c_{n+1}^{(j)}, w_n^{(j)})$.

v. Compute $\varepsilon_n^{(j)} \sim r(G(y_n, e_n^{(j)}))\left|\frac{\partial G}{\partial y_n}\right|$ and generate $c_n^{(j)} \sim \left(\sum_{i=1}^m \varepsilon_n^{(i)}\right)^{-1} \sum_{i=1}^m \varepsilon_n^{(i)} I(x, e_n^{(i)})$ for $j = 1, \ldots, m$ by the resampling of $e_n^{(1)}, \ldots, e_n^{(m)}$.

Figures 5c and 5d show the smoothed estimates obtained by the two-filter formula with $d' = 1{,}000$. Much smoother and closer approximations to the "true" ones in Figures 4a and 4b than the other estimates are obtained.

5. EXAMPLES

5.1 FITTING AR MODEL WITH OUTLIERS

In this section we consider an outlier problem. Figure 6a shows the log-transformed version of the famous Canadian lynx (e.g., Tong 1977) data which we have artificially contaminated at nine randomly chosen time points with the addition of the quantity 1.0, to each of the selected data. The symbol + identifies the original uncontaminated data and ∘ the contaminated ones. To detect the above mentioned additive outliers, we consider

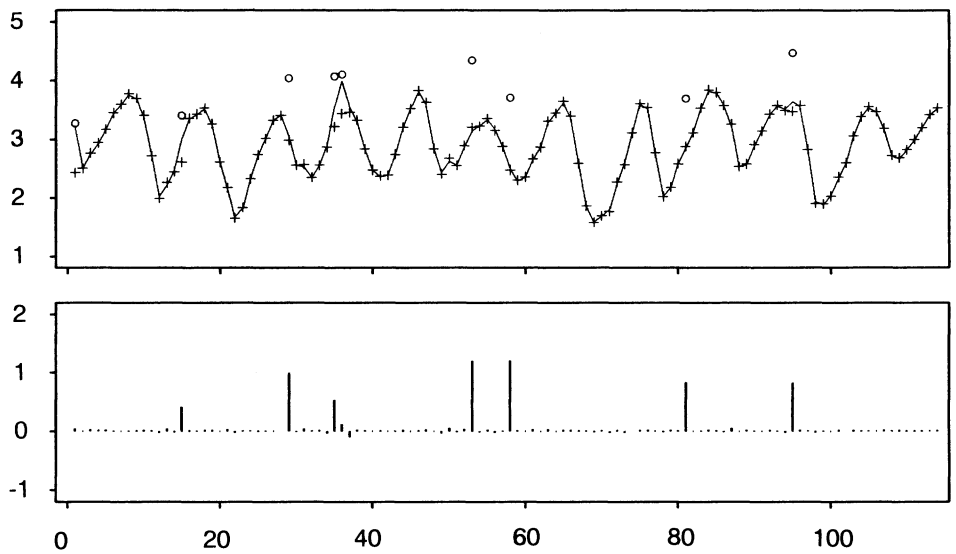

Figure 6. Outliers in Time Series. (a) Contaminated Canadian lynx data: + = the original data; ○ = the contaminated data; and —- = smoothed data. (b) Estimated observation noise.

the following AR model with observation error

$$\begin{aligned} p_n &= \sum_{i=1}^{M} a_i p_{n-i} + v_n \\ y_n &= p_n + w_n. \end{aligned} \tag{5.1}$$

To simplify the discussion, the AR order M and the AR coefficients a_i are assumed to be known. In the present example, $M = 11$ and thus the dimension of the state vector is 11. The number of particles, m, is set to 1,000.

For this model, the general state space model (2.1)–(2.2) is specified by

$$x_n = \begin{bmatrix} p_n \\ p_{n-1} \\ \vdots \\ p_{n-10} \end{bmatrix},$$

$$F(x_{n-1}, v_n) = \begin{bmatrix} a_1 & a_2 & \cdots & a_{11} \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix} x_{n-1} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_n, \text{ and}$$

$$H(x_n, w_n) = [\,1\ 0\ \ldots\ 0\,]x_n + w_n. \tag{5.2}$$

In this case the inverse function G is given by $G(y_n, x_n) = y_n - p_n$ and $|\frac{\partial G}{\partial y_n}| = 1$.

Four different models were fitted. Their associated AIC's are:

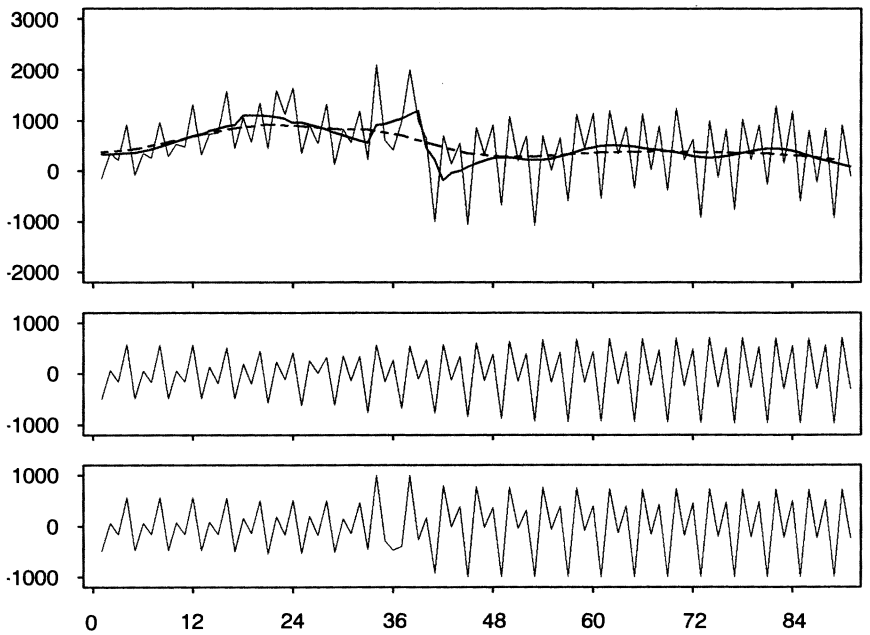

Figure 7. Seasonal Adjustment of Inventory of Private Companies in Japan. (a) Original data and the estimated trend. Dashed curve = Gaussian model; fine curve = non-Gaussian model. (b) Estimated seasonal component by a Gaussian model. (c) Estimated seasonal component by a Gaussian-mixture model.

Model(G,G):	$v_n \sim N(0, \tau^2)$,	$w_n \sim N(0, \sigma^2)$,	AIC=94 .46
Model(G,C):	$v_n \sim N(0, \tau^2)$,	$w_n \sim \text{Cauchy}(0, \sigma^2)$,	AIC=47 .32
Model(C,G):	$v_n \sim \text{Cauchy}(0, \tau^2)$,	$w_n \sim N(0, \sigma^2)$,	AIC=13 7.04
Model(C,C):	$v_n \sim \text{Cauchy}(0, \tau^2)$,	$w_n \sim \text{Cauchy}(0, \sigma^2)$,	AIC=10 1.48

As expected, Model(G,C) is the AIC best model. In Figure 6a, the line identifies the smoothed estimate of the AR process. Figure 6b shows the plot of the estimated observation noise sequence. Seven out of the nine noise contaminated data are properly detected and for the most part, recovered the original data. The noise contamination at the first observation was not detected at all and the contaminated data at $n = 36$ was considerably underestimated.

5.2 SEASONAL ADJUSTMENT

In Figure 7a, the variable curve shows the quarterly series of the inventory of private companies in Japan from 1965 to 1987. The standard state space model for the seasonal adjustment is (Kitagawa and Gersch 1984; Harvey 1989)

$$y_n = T_n + S_n + w_n, \tag{5.3}$$

where T_n is a second-order trend component given by $t_n = 2t_{n-1} - t_{n-2} + v_n$, and S_n is a seasonal component defined by

$$S_n = -(S_{n-1} + \cdots + S_{n-p+1}) + u_n. \tag{5.4}$$

p is the number of observations in one period. The state space model for this model for the decomposition of seasonal time series is

$$x_n = \begin{bmatrix} t_n \\ t_{n-1} \\ \hline S_n \\ S_{n-1} \\ \vdots \\ S_{n-p+2} \end{bmatrix},$$

$$F = \left[\begin{array}{cc|ccccc} 2 & -1 & & & & & \\ 1 & 0 & & & & & \\ \hline & & -1 & -1 & \cdots & -1 & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & & 1 & \end{array}\right],$$

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

$$H = [1\,0\,1\,0\,\cdots\,0]. \tag{5.5}$$

(See Kitagawa and Gersch 1984 for the details and extensions of the model.) Usually, each of u_n, v_n, and w_n is assumed to be Gaussian, and the Kalman filter and smoother are applied for the estimation of the trend and seasonal components and for the computation of the likelihood. However, if the trend or seasonal component changes abruptly, as in the case of Figure 7a, the non-Gaussian model is preferable since that model achieves automatic detection of abrupt changes, (Kitagawa 1987). The direct application of the non-Gaussian smoother is difficult for seasonal adjustment because it requires estimation of a $p+1$ dimensional state vector. The Gaussian-sum filter and smoother shown in Kitagawa (1989, 1994), does satisfactorily cope with this problem.

In this example, our Monte Carlo smoother with $m = 40{,}000$ was applied to the same problem as in the articles cited. We compared the Gaussian model $w_n \sim N(0, \sigma^2)$, $v_n \sim N(0, \tau_1^2)$, $u_n \sim N(0, \tau_2^2)$, and the following non-Gaussian (Gaussian-mixture) model:

$$\begin{aligned} w_n &\sim N(0, \sigma^2) \\ v_n &\sim \alpha N(0, \tau_1^2) + (1-\alpha) N(0, \zeta_1^2) \\ u_n &\sim \beta N(0, \tau_2^2) + (1-\beta) N(0, \zeta_2^2). \end{aligned} \tag{5.6}$$

In (5.6) w_n, v_n, and u_n are respectively the observation, the trend and the seasonal noises. The maximum likelihood estimates of the Gaussian model are $\hat{\sigma}^2 = .874 \times 10^5$,

$\hat{\tau}_1^2 = .146 \times 10^3$ and $\hat{\tau}_2^2 = .365 \times 10^4$. The log-likelihood of the model is -670.39 and the AIC= 1346.8. On the other hand the maximum likelihood estimates of the non-Gaussian model are $\hat{\sigma}^2 = 0.288 \times 10^5$, $\hat{\tau}_1^2 = .528 \times 10^3$, $\hat{\tau}_2^2 = .362 \times 10^3$ and $\hat{\alpha} = \hat{\beta} = .98$. In the estimation of these parameters, for simplicity we let $\alpha = \beta$ and arbitrarily set ζ_1^2 and ζ_2^2 to the variance of the original series. The log-likelihood of that model is -667.01 and the AIC= 1342.0. According to the AIC criterion, the non-Gaussian model is better than the Gaussian model.

Figure 7 shows the quarterly series of the inventories of private companies in Japan 1965–1983 and the estimated trend and seasonal components of the Gaussian and non-Gaussian models. In Figure 7a, the dashed line and smooth curve show the estimated trends by the Gaussian and non-Gaussian models, respectively. Figures 7b and 7c respectively show the estimated seasonal component by the Gaussian and the non-Gaussian models. The 1973–1974 energy crisis occurred in the middle of the series, and the level of the series changed suddenly. The estimated trend (dashed line in Fig. 7a) and seasonal component (Fig. 7b) obtained by the Gaussian model is insensitive to the abrupt change of the series. On the other hand, the non-Gaussian model, without any explicit specification of the change points, clearly detects the abrupt changes of both the trend (line in Fig. 7a) and the seasonal pattern (Fig. 7c).

5.3 NONLINEAR MODEL

We consider the data artificially generated by the following model that was originally used by Andrede Netto, Gimeno, and Mendes (1978) and reconsidered in Kitagawa (1987, 1991),

$$\begin{aligned} x_n &= \frac{1}{2}x_{n-1} + \frac{25x_{n-1}}{1+x_{n-1}^2} + 8\cos(1.2n) + v_n \\ y_n &= \frac{x_n^2}{20} + w_n, \end{aligned} \tag{5.7}$$

with $v_n \sim N(0, 0.1)$ and $w_n \sim N(0, 1)$. Figures 8a and 8b show the state signal x_n and the observed data y_n. The problem is to estimate the unobserved true signal x_n only from the sequence of observations $\{y_n\}$ assuming that the model (5.7) is known. It should be emphasized here that because the value of the state x_n is squared in the observation equation in (5.7), it is quite difficult to identify whether the state x_n is positive or negative. The Monte Carlo filter and smoother can be easily applied to this problem. For comparison, the well-known extended Kalman filter based smoother (Sage and Melsa 1971) and a nonlinear smoother based on numerical integration, (Kitagawa 1991) were also applied.

Figure 8c shows the estimate of x_n by the Monte Carlo method with $m = 10{,}000$. The 2.7%, 50%, and 97.7% probability points are shown. It can be seen that, except for $n = 3$, 24, 36, 64, and 98, the median of the Monte Carlo smoother correctly identifies the sign of the signal. Figures 8d and 8e show the results by the exact, (numerical integration) nonlinear smoother and the extended Kalman smoother, respectively. They show that the Monte Carlo method provides a close approximation to the exact smoother and that its performance is greatly superior to the extended Kalman smoother which almost diverges.

5.4 TIME SERIES WITH TREND AND TIME-VARYING VARIANCE

Figure 9a shows the record of daily series of Japanese stock price in Yen (Nikkei 225) from January 1987 to August 1990. It can be seen that this series has a generally increasing trend and the volatility increases after a significant decrease of trend around at $n = 230$, 840, and 940. For modeling this series, consider the trend model

$$y_n = t_n + w_n, \tag{5.8}$$

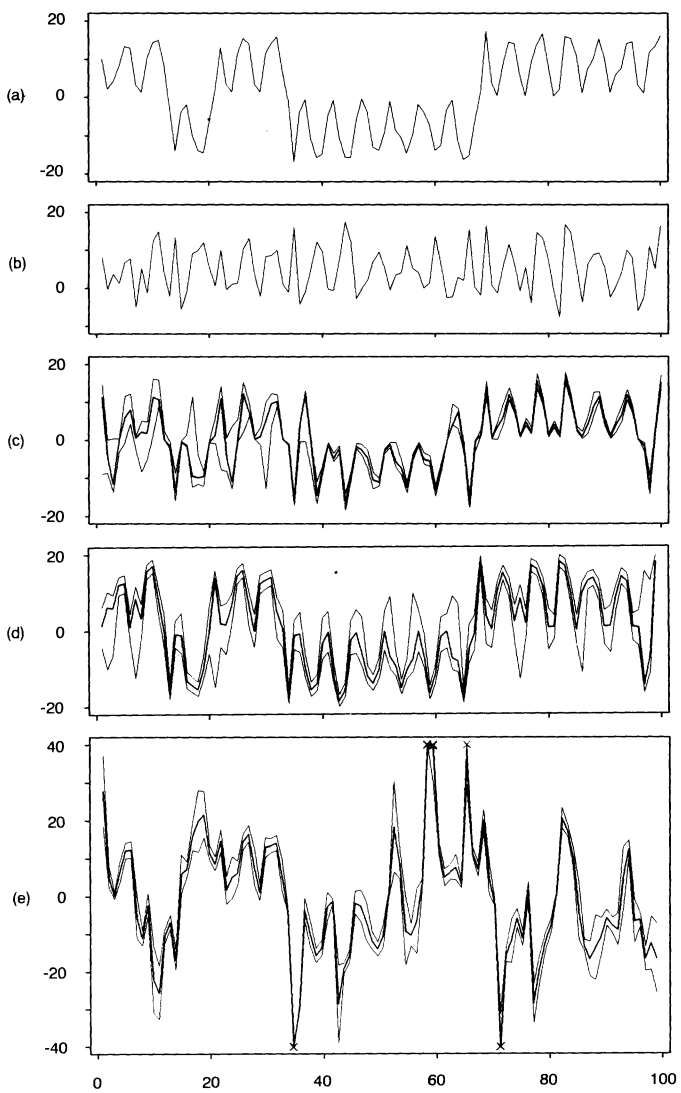

Figure 8. Nonlinear Smoothing. (c)–(e), posterior distribution, bold curve = median, fine curve = 2.3% and 97.7% points. (a) Unobserved signal. (b) Observed time series. (c) Monte Carlo smoother. (d) Exact non-Gaussian smoother. (e) Extended Kalman smoother.

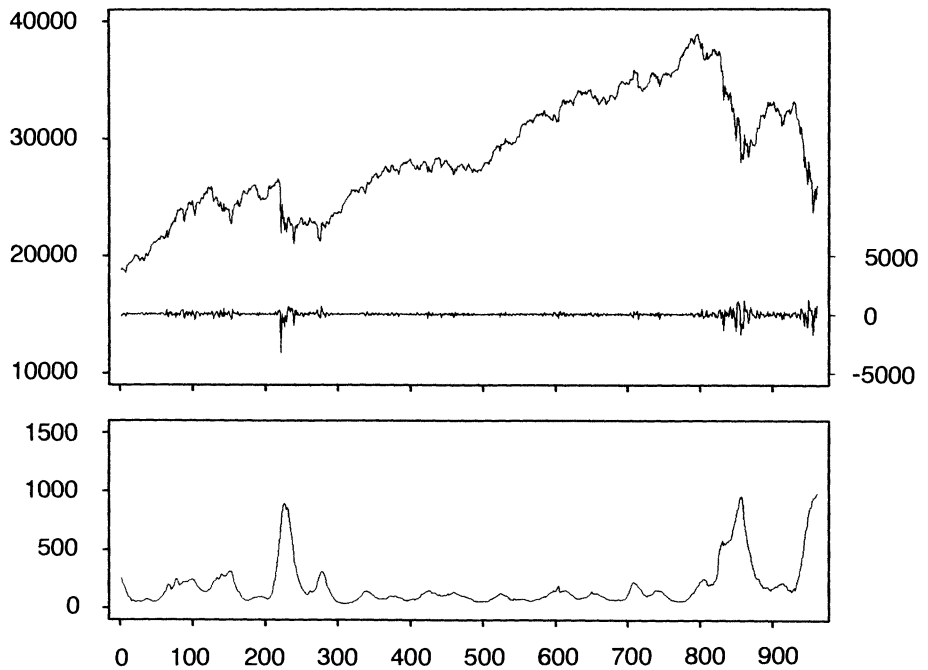

Figure 9. Daily Stock Price Data. (a) Nikkei 225 data (Jan. 1987–Aug. 1990) and the estimated noise sequence (lower). (b) Estimated time-varying standard deviation.

where t_n is a trend and w_n is a white noise sequence with unknown time-varying variance

$$w_n \sim N(0, \sigma_n^2). \tag{5.9}$$

For the simultaneous estimation of trend and variance, we introduce the following smoothness prior models

$$\begin{aligned} t_n &= 2t_{n-1} - t_{n-2} + \varepsilon_n \\ p_n &= 2p_{n-1} - p_{n-2} + \delta_n, \end{aligned} \tag{5.10}$$

with $\sigma_n^2 = \exp(p_n)$. ε_n and δ_n are white noise sequences specified in the following. The exponential transformation is used to assure the positivity of the variance σ_n^2. The models (5.8), (5.9), and (5.10), can be expressed in a state space model (2.1) and (2.2) with the four-dimensional state vector $x_n = (t_n, t_{n-1}, p_n, p_{n-1})^t$, two-dimensional system noise $v_n = (\varepsilon_n, \delta_n)^t$, and a nonlinear function $H(x_n, w_n) = H((t_n, t_{n-1}, p_n, p_{n-1})^t, w_n) = t_n + w_n = t_n + \exp(p_n/2)e_n$ with $e_n \sim N(0,1)$ and

$$\begin{aligned} F(x_n, v_n) &= F((t_n, t_{n-1}, p_n, p_{n-1})^t, (\varepsilon_n, \delta_n)^t) \\ &= \begin{bmatrix} 2t_{n-1} - t_{n-2} + \varepsilon_n \\ t_{n-1} \\ 2p_{n-1} - p_{n-2} + \delta_n \\ p_{n-1} \end{bmatrix} \\ G(y_n, x_n) &= e^{-p_n/2}(y_n - t_n). \end{aligned} \tag{5.11}$$

We considered the following two models:

$$\text{Model(G,G):} \quad \varepsilon_n \sim N(0, \tau_1^2), \qquad \delta_n \sim N(0, \tau_2^2),$$
$$\text{Model(C,G):} \quad \varepsilon_n \sim \text{Cauchy}(0, \tau_1^2), \quad \delta_n \sim N(0, \tau_2^2).$$

We used $m = 10{,}000$ particles for the Monte Carlo filter and smoother. The approximate maximum likelihood estimates obtained by discrete parameter search are:

$$\text{Model(G,G):} \quad \tau_1^2 = 9000, \quad \tau_2^2 = .0026, \quad \text{AIC} = 13726.02$$
$$\text{Model(C,G):} \quad \tau_1^2 = 700, \quad \tau_2^2 = .0005, \quad \text{AIC} = 13765.84.$$

The AIC criterion suggests that the Model(G,G) is better than the other.

In Figure 9a the lower series shows the estimated noise series w_n. The increase of the volatility around at n =230, 840, and 940 is seen. The trend is only a smoothed version of the original series and is not shown. Figure 9b shows the square root of the estimated time-varying variance, $\sqrt{\sigma_n^2}$. Corresponding to the increase of the variance of w_n seen in Figure 9a, the estimated $\sqrt{\sigma_n^2}$ clearly shows the increase of volatility after the sudden drops in the trend.

6. CONCLUSION

By approximating the density functions by many particles and by simulating the time evolution of the particles and resampling, we can develop algorithms for the filtering and smoothing of the general non-Gaussian nonlinear state space model. By using a simple example, for which an almost exact numerical integration based non-Gaussian filter and smoother can be applied, it was shown that the proposed Monte Carlo method can reasonably approximate the distribution of the state.

The most significant merit of the method is that it can be applied to a state space model with higher dimension. The numerical examples in Sections 5.1 and 5.2 have state dimensions 11 and 5, respectively. In those examples, the numerical integration based non-Gaussian filter and smoother cannot be applied. Another merit of the method is that its application to various types of nonlinear or non-Gaussian models is very simple and only minor modifications for specifying the transition and observation functions F and H and the noise densities q and r are required.

APPENDIX: RESAMPLING ALGORITHMS

In this appendix, we consider the resampling algorithms for the filtering step.

First we show the basic algorithm.

1. Rearrange $p_n^{(1)}, \ldots, p_n^{(m)}$ in order of magnitude. The results are expressed as $\tilde{p}_n^{(1)}, \ldots, \tilde{p}_n^{(m)}$.
2. For $j = 1, \ldots, m$, repeat the following steps (a)–(c).
   (a) Generate a uniform random number $u_n^{(j)} \in U[0, 1]$.
   (b) Find i such that $\frac{1}{C}\sum_{\ell=1}^{i-1} \tilde{\alpha}_n^{(\ell)} < u_n^{(j)} \leq \frac{1}{C}\sum_{\ell=1}^{i} \tilde{\alpha}_n^{(\ell)}$, where $C = \sum_{\ell=1}^{m} \tilde{\alpha}_n^{(\ell)}$ and $\tilde{\alpha}_n^{(\ell)}$ is the posterior weight corresponding to $\tilde{p}_n^{(\ell)}$.
   (c) Define $f_n^{(j)} = \tilde{p}_n^{(i)}$.

A variety of modifications of this basic algorithm are possible. They are related to the sorting and the generation of random numbers. Because the purpose of the resampling is to obtain an empirical distribution function that mimics the distribution function defined in (3.6), it is not essential to do random resampling. A possible modification is stratified sampling. This scheme is devised so as not to draw more than one realization from a group of particles with total weight $1/m$ and is given as follows: Divide the interval [0, 1) to m subintervals with identical width. In this case, Step (a) can be modified to either of the following two methods:

(a-S) Generate a uniform random number $u_n^{(j)} \sim U\left[\frac{j-1}{m}, \frac{j}{m}\right)$.

(a-D) $u_n^{(j)} = \frac{j-\alpha}{m}$, for fixed $\alpha \in [0, 1)$.

The sorting in Step 1 above is the most time-consuming job, especially for large m. It takes .0001, .0005, .020, and 1.736 seconds for $m = 10$, 100, 1,000 and 10,000, respectively, by the HEAPSORT algorithm, (Press et al. 1986), on a 40 MFLOPS computer. If the random sampling without stratification is used, sorting is unnecessary.

Table A.1 shows the Monte Carlo evaluation of the approximation of various resampling methods using the same model as the one for Figure 1 in Section 3.5. One step of filtering was repeated for 1,000 times and the discrepancy between the distribution functions given in (3.6) and (3.7) were evaluated by the following measure

$$\begin{aligned} J(S,T) &= \int_{-\infty}^{\infty} (D_p(x) - D_f(x))^2 dx \\ &= \int_{-\infty}^{\infty} \left(\frac{1}{c} \sum_{i=1}^{m} \alpha_n^{(i)} I(x, p_n^{(i)}) - \frac{1}{m} \sum_{i=1}^{m} I(x, f_n^{(i)}) \right)^2 dx, \end{aligned} \tag{A.1}$$

where $c = \sum_{i=1}^{m} \alpha_n^{(i)}$ and $D_p(x)$ and $D_f(x)$ are distribution f unctions defined by (3.6) and (3.7), respectively. It can be seen that the deterministic method (a-D) and stratified method (a-S) outperform the random one. Within these two methods, the deterministic one is better than the stratified. When the realizations are ordered, stratified and deterministic ones show $O(m^{-2})$ convergence, while in the unordered case, they show only $O(m^{-1})$ convergence. Obviously, the deterministic algorithm with sorting is the best. On

Table A.1. Comparison of the Resampling Methods

		$J(D_f, D_p)$			$J(D_p, D_f^*)$
		Random	*Stratified*	*Deterministic*	
m = 10	sort	.0326	.00379	.00176	.1021
	no-sort	.0321	.00859	.00513	.1021
m = 100	sort	.00381	$.636 \times 10^{-3}$	$.311 \times 10^{-4}$	$.860 \times 10^{-2}$
	no-sort	.00405	$.939 \times 10^{-3}$	$.612 \times 10^{-3}$	$.860 \times 10^{-2}$
m = 1,000	sort	$.398 \times 10^{-3}$	$.838 \times 10^{-6}$	$.407 \times 10^{-6}$	$.102 \times 10^{-2}$
	no-sort	$.379 \times 10^{-3}$	$.982 \times 10^{-4}$	$.611 \times 10^{-4}$	$.102 \times 10^{-2}$
m =10,000	sort	$.387 \times 10^{-4}$	$.101 \times 10^{-7}$	$.498 \times 10^{-8}$	$.248 \times 10^{-3}$
	no-sort	$.406 \times 10^{-4}$	$.936 \times 10^{-5}$	$.636 \times 10^{-5}$	$.248 \times 10^{-3}$

the other hand, the rightmost column in Table A.1 shows the discrepancy between $\{p_n^{(j)}\}$ and the "exact" filter distribution D_f^*. This shows that, at least for large m, $J(D_f, D_p)$ is negligible compared with $J(D_p, D_f^*)$. Therefore, considering the significant computational cost in sorting, the deterministic algorithm without sorting might be a reasonable choice.

ACKNOWLEDGMENTS

The author thanks Will Gersch of University of Hawaii, for his valuable comments on the manuscript and the work. Careful reading and helpful comments by the editors and the referees are gratefully acknowledged. This research was partially supported by the Grant-in-Aid for Scientific Research by the Japanese Ministry of Education, Science and Culture.

[Received December 1994. Revised September 1995.]

REFERENCES

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in *Second International Symposium on Information Theory*, eds. B. N. Petrov and F. Csaki, Budapest: Akademiai Kiado, pp. 267–281.

Alspach, D. L., and Sorenson, H. W. (1972), "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, AC-17, 439–448.

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

Andrede Netto, M. L., Gimeno, L., and Mendes, M. J. (1978), "On the Optimal and Suboptimal Nonlinear Filtering Problem for Discrete-Time Systems," *IEEE Transactions on Automatic Control*, 23, 1062–1067.

Carlin, B.P., Polson, N.G., and Stoffer, D.S. (1992), "A Monte Carlo Approach to Nonnormal and Nonlinear State-Space Modeling," *Journal of the American Statistical Association*, 87, 439–500.

Fahrmeir, L. (1992), "Posterior Mode Estimation by Extended Kalman Filtering for Multivariate Dynamic Generalized Linear Models," *Journal of the American Statistical Association*, 87, 501–509.

Fahrmeir, L., and Kaufmann, H. (1991), "On Kalman Filtering, Posterior Mode Estimation and Fisher Scoring in Dynamic Exponential Family Regression," *Metrika*, 38, 37–60.

Frühwirth-Schnatter, S. (1994), "Data Augmentation and Dynamic Linear Models," *Journal of Time Series Analysis*, 15, 183–202.

Harrison, P. J., and Stevens, C. F. (1976), "Bayesian Forecasting" (with discussion), *Journal of the Royal Statistical Society*, Ser. B, 34, 1–41.

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

Hodges, P. E., and Hale, D. F. (1993), "A Computational Method for Estimating Densities of non-Gaussian Nonstationary Univariate Time Series," *Journal of Time Series Analysis*, 14, 163–178.

Kitagawa, G. (1987), "Non-Gaussian State-Space Modeling of Nonstationary Time Series" (with discussion), *Journal of the American Statistical Association,* 82, 1032–1063.

——— (1989), "Non-Gaussian Seasonal Adjustment," *Computers & Mathematics with Applications*, 18, 503–514.

——— (1991), "A Nonlinear Smoothing Method for Time Series Analysis," *Statistica Sinica*, 1, 371–388.

——— (1993), "A Monte Carlo Filtering and Smoothing Method for Non-Gaussian Nonlinear State Space Models," *Proceedings of the 2nd U.S.–Japan Joint Seminar on Statistical Time Series Analysis*, pp. 110–131.

——— (1994), "The Two-Filter Formula for Smoothing and an Implementation of the Gaussian-Sum Smoother," *Annals of the Institute of Statistical Mathematics,* 46, 605–623.

Kitagawa, G., and Gersch, W. (1984), "A Smoothness Priors-State Space Modeling of Time Series With Trend and Seasonality," *Journal of the American Statistical Association*, 79, 378–389.

Masreliez, C. J. (1975), "Approximate Non-Gaussian Filtering With Linear State and Observation Relations," *IEEE Transactions on Automatic Control*, AC-20, 107–110.

Meinhold, R. J., and Singpurwalla, N. D. (1989), "Robustification of Kalman Filter Models," *Journal of the American Statistical Association*, 84, 479–486.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992), *Numerical Recipes in FORTRAN, The Art of Scientific Computing,* (2nd ed.), Cambridge, UK: Cambridge University Press.

Sage, A. P., and Melsa, J. L. (1971), *Estimation Theory with Application to Communication and Control*, New York: McGraw-Hill.

Sakamoto, Y., Ishiguro, M., and Kitagawa, G (1986), *Akaike Information Criterion Statistics*, Tokyo: D. Reidel.

Schnatter, S. (1992), "Integration-Based Kalman-Filtering for a Dynamic Generalized Linear Trend Model," *Computational Statistics and Data Analysis*, 13, 447–459.

Smith, R. L., and Miller, J. E. (1986), "A Non-Gaussian State Space Model and Application to Prediction of Records," *Journal of the Royal Statistical Society*, Ser. B, 48, 79–88.

Tanizaki, H. (1993), "Nonlinear Filters," *Lecture Notes in Economics and Mathematical Systems*, No. 400, New York: Springer-Verlag.

Tong, H. (1977), "Some Comments on the Canadian Lynx Data," *Journal of the Royal Statistical Society*, A-140, 432–436.

West, M., Harrison, P.J., and Migon, H. S. (1985), "Dynamic Generalized Linear Models and Bayesian Forecasting" (with discussion), *Journal of the American Statistical Association,* 80, 73–97.