Ann. Inst. Statist. Math.
Vol. 53, No. 1, 63–81 (2001)

NONLINEAR AND NON-GAUSSIAN STATE SPACE MODELING USING SAMPLING TECHNIQUES

Hisashi Tanizaki*

Faculty of Economics, Kobe University, Kobe 657-5801, Japan
(tanizaki@kobe-u.ac.jp)

(Received April 11, 2000; Revised October 12, 2000)

Abstract. In this paper, the nonlinear non-Gaussian filters and smoothers which are much less computational than the existing ones are proposed, where the sampling techniques such as rejection sampling (RS), importance resampling (IR) and the Metropolis-Hastings independence sampling (MH) are utilized. The existing density-based nonlinear and non-Gaussian filters and smoothers utilize the marginal densities, i.e., $p(\alpha_t|Y_t)$ for filtering and $p(\alpha_t|Y_T)$ for smoothing, but the algorithms proposed in this paper are based on the joint densities, i.e., $p(\alpha_t, \alpha_{t-1}|Y_t)$ for filtering and $p(\alpha_{t+1}, \alpha_t|Y_T)$ or $p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T)$ for smoothing. That is, in this paper, the random draws of α_t are generated from $p(\alpha_t, \alpha_{t-1}|Y_t)$ for filtering and $p(\alpha_{t+1}, \alpha_t|Y_T)$ or $p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T)$ for smoothing. By generating the random draws from the joint densities, much less computer-intensive algorithms on filtering and smoothing can be obtained. Furthermore, taking into account possibility of structural changes and outliers during the estimation period, the appropriately chosen sampling density is possibly introduced into the suggested nonlinear non-Gaussian filtering and smoothing procedures. Finally, through Monte Carlo simulation studies, the suggested filters and smoothers are examined.

Key words and phrases: State Space Modeling, Filtering, Smoothing, Marginal Density, Joint Density, Rejection Sampling, Importance Resampling, Metropolis-Hastings Independence Sampling.

1. Introduction

Various nonlinear non-Gaussian filters and smoothers have been proposed for the last decade in order to improve precision of the state estimates and reduce a computational burden. The state mean and variance are evaluated by generating random draws directly from the filtering density or the smoothing density. Clearly, precision of the state estimates is improved as number of random draws increases. Thus, the recent filters and smoothers have been developed by applying some sampling techniques such as Gibbs sampling, rejection sampling (RS), importance resampling (IR), the Metropolis-Hastings independence sampling (MH) and etc.

Carlin et al. (1992) and Carter and Kohn (1994, 1996) applied the Gibbs sampler to some specific state space models, which are extended to more general state space models by Geweke and Tanizaki (1999). The Gibbs sampler sometimes gives us the

*The author would like to acknowledge the constructive comments of the anonymous referees. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Encouragement of Young Scientists (# 12730022).

imprecise estimates of the state variables, depending on the underlying state space model (see Carter and Kohn (1994, 1996)). Especially when the random variables are highly correlated with each other, it is well known that convergence of the Gibbs sampler is unacceptably slow. In the case of state space models, the transition equation represents the relationship between the state variable α_t and the lagged one α_{t-1}. Accordingly, it is clear that the state variable at present time has high correlation with that at past time. As for the state space models, therefore, the Gibbs sampler is one of the sources of imprecise state estimates. In this paper, unlike Carlin et al. (1992) and Carter and Kohn (1994, 1996), the density-based recursive algorithms on filtering and smoothing are discussed, where they are compared with respect to the three sampling methods, i.e., RS, IR and MH, although any sampling technique can be applied.

Gordon et al. (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) proposed the nonlinear non-Gaussian state space modeling by IR, which can be applied to almost all the state space models. Both filtering and smoothing random draws are based on the one-step ahead prediction random draws. In the case where the past information is too different from the present sample, however, the obtained filtering and smoothing random draws become unrealistic. To avoid this situation, in this paper we take an appropriately chosen density as the sampling density for random number generation. Note that Kong et al. (1994), Liu and Chen (1995, 1998) and Doucet et al. (2000) also utilized the density other than the state estimation. In addition, the fixed-interval smoother proposed by Kitagawa (1996) and Kitagawa and Gersch (1996) does not give us the exact solution of the state estimate even when the number of random draws is large enough, because the fixed-interval smoother suggested by Kitagawa (1996) is approximated by the fixed-lag smoother. To improve these disadvantages, in this paper we propose the fixed-interval smoother which yields the exact solution of the state estimate. As an alternative smoother, furthermore, Kitagawa (1996) introduces the fixed-lag smoother based on the two-filter formula, where forward and backward filtering are performed and combined to obtain the smoothing density. The smoother based on the two-filter formula is discussed in Appendix A.

Moreover, the RS filter and smoother have been developed by Tanizaki (1996, 1999), Hürzeler and Künsch (1998) and Tanizaki and Mariano (1998). To implement RS for random number generation, we need to compute the supremum in the acceptance probability, which depends on the underlying functional form of the measurement and transition equations. RS cannot be applied in the case where the acceptance probability is equal to zero, i.e., when the supremum is infinity. Even if the supremum is finite, it takes a lot of computational time when the acceptance probability is close to zero. To improve the problems in rejection sampling, Liu, Chen and Wong (1998) suggested the rejection controlled sequential importance sampling algorithm, where rejection sampling and importance sampling are combined.

As computer progresses day by day, the computer-intensive nonlinear non-Gaussian estimators have been developed. However, it is clear that less computational estimators are preferred. To reduce the computational disadvantages for filtering and smoothing, we consider generating random draws from the joint densities (i.e., $p(\alpha_t, \alpha_{t-1}|Y_t)$ for filtering and $p(\alpha_{t+1}, \alpha_t|Y_T)$ or $p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T)$ for smoothing, where the notations are defined in Section 2.1), not from the marginal densities (i.e., $p(\alpha_t|Y_t)$ for filtering and $p(\alpha_t|Y_T)$ for smoothing). A lot of filters have been proposed based on the marginal density and accordingly the existing filters are computationally too intensive. Kong et al. (1994) and Liu and Chen (1995) also suggested drawing from the joint density of the state variable and the auxiliary variable. However, use of the auxiliary variable yields a very computer-intensive filtering algorithm. Therefore, for filtering, we consider drawing from the joint density of the state variables, i.e., (α_t, α_{t-1}), given Y_t. Furthermore, in a lot of literature on the IR and RS procedures, smoothing is not investigated because smoothing is much more computer-intensive than filtering. Dealing with the joint densities of the state variables yields less computational procedures and therefore we can obtain not only filtering but also smoothing in the IR and RS procedures. In this paper, thus, we propose much less computational nonlinear non-Gaussian filters and smoothers using the joint densities of the state variables, where RS, IR or MH may be utilized for random number generation.

2. Preliminaries

2.1 *State Space Model*

Kitagawa (1987), Harvey (1989), Kitagawa and Gersch (1996) and Tanizaki (1996, 2000) discuss the nonlinear non-Gaussian state space models, which are described by the following two equations:

$$\text{(2.1)} \qquad \text{(Measurement equation)} \qquad y_t = h_t(\alpha_t, \epsilon_t),$$

$$\text{(2.2)} \qquad \text{(Transition equation)} \qquad \alpha_t = f_t(\alpha_{t-1}, \eta_t),$$

where y_t represents the observed data at time t while α_t denotes the state vector at time t which is unobservable. ϵ_t and η_t are mutually independently distributed. $h_t(\cdot,\cdot)$ and $f_t(\cdot,\cdot)$ are assumed to be known. $\alpha_{t|s} \equiv \mathrm{E}(\alpha_t|Y_s)$ is called prediction if $t > s$, filtering if $t = s$ and smoothing if $t < s$, where Y_s denotes the information set up to time s, i.e., $Y_s = \{y_1, y_2, \cdots, y_s\}$. Moreover, there are three kinds of smoothing estimators, i.e., the fixed-point smoothing $\alpha_{L|t}$, the fixed-lag smoothing $\alpha_{t|t+L}$ and the fixed-interval smoothing $\alpha_{t|T}$ for fixed L and fixed T. In this paper, we focus on the filter and the fixed-interval smoother, i.e., $\alpha_{t|s}$ for $s = t, T$.

Define $p_y(y_t|\alpha_t)$ and $p_\alpha(\alpha_t|\alpha_{t-1})$ by the density functions derived from the measurement equation (2.1) and the transition equation (2.2). The density-based filtering algorithm is given by:

$$\text{(2.3)} \quad \text{(Prediction equation)} \qquad p(\alpha_t|Y_{t-1}) = \int p_\alpha(\alpha_t|\alpha_{t-1}) p(\alpha_{t-1}|Y_{t-1}) \mathrm{d}\alpha_{t-1},$$

$$\text{(2.4)} \quad \text{(Update equation)} \qquad p(\alpha_t|Y_t) = \frac{p_y(y_t|\alpha_t) p(\alpha_t|Y_{t-1})}{\int p_y(y_t|\alpha_t) p(\alpha_t|Y_{t-1}) \mathrm{d}\alpha_t},$$

for $t = 1, 2, \cdots, T$. The initial condition is given by: $p(\alpha_1|Y_0) = \int p_\alpha(\alpha_1|\alpha_0) p_\alpha(\alpha_0) \mathrm{d}\alpha_0$ if α_0 is stochastic and $p(\alpha_1|Y_0) = p_\alpha(\alpha_1|\alpha_0)$ otherwise. $p_\alpha(\alpha_0)$ denotes the unconditional density of α_0. The filtering algorithm takes the following two steps: (i) from equation (2.3), $p(\alpha_t|Y_{t-1})$ is obtained given $p(\alpha_{t-1}|Y_{t-1})$, and (ii) from equation (2.4), $p(\alpha_t|Y_t)$ is derived given $p(\alpha_t|Y_{t-1})$. Thus, $p(\alpha_t|Y_t)$ is recursively obtained for $t = 1, 2, \cdots, T$.

The density-based smoothing algorithm utilizes both the one-step ahead prediction density $p(\alpha_{t+1}|Y_t)$ and the filtering density $p(\alpha_t|Y_t)$, which is represented by:

$$\text{(2.5)} \qquad p(\alpha_t|Y_T) = p(\alpha_t|Y_t) \int \frac{p(\alpha_{t+1}|Y_T) p_\alpha(\alpha_{t+1}|\alpha_t)}{p(\alpha_{t+1}|Y_t)} \mathrm{d}\alpha_{t+1},$$

for $t = T-1, T-2, \cdots, 1$. Given $p(\alpha_t|Y_t)$ and $p(\alpha_{t+1}|Y_t)$, the smoothing algorithm represented by equation (2.5) is a backward recursion from $p(\alpha_{t+1}|Y_T)$ to $p(\alpha_t|Y_T)$.

For the fixed-interval smoother, Kitagawa (1996) developed an alternative smoothing algorithm, which is based on the two-filter formula. This alternative algorithm is discussed in Appendix A.

Let $g(\cdot)$ be a function, e.g., $g(\alpha_t) = \alpha_t$ for mean or $g(\alpha_t) = (\alpha_t - \alpha_{t|s})(\alpha_t - \alpha_{t|s})'$ for variance. Given $p(\alpha_t|Y_s)$, the conditional expectation of $g(\alpha_t)$ given Y_s is represented by:

$$\mathrm{E}\big(g(\alpha_t)|Y_s\big) = \int g(\alpha_t) p(\alpha_t|Y_s) \mathrm{d}\alpha_t. \tag{2.6}$$

When we have the unknown parameters in equations (2.1) and (2.2), the following likelihood function is maximized with respect to the parameters:

$$p(Y_T) = \prod_{t=1}^{T} p(y_t|Y_{t-1}) = \prod_{t=1}^{T} \left(\int p(y_t|\alpha_t) p(\alpha_t|Y_{t-1}) \mathrm{d}\alpha_t \right). \tag{2.7}$$

Since $p(y_t|Y_{t-1})$ in (2.7) corresponds to the denominator in equation (2.4), we do not need extra computation for evaluation of the likelihood function. Thus, the unknown parameter is obtained by maximum likelihood estimation (MLE). As for an alternative approach to estimate the unknown parameter, Kitagawa (1998) suggested taking the unknown parameter as the state variable, which is called the self-organizing filter.

Our goal is to obtain the expectation in (2.6), which is evaluated generating random draws of α_t. Therefore, in the next section, we overview some sampling techniques.

2.2 *Sampling Techniques*

We want to generate random draws from $p(x)$, called the target density, but we consider the case where it is hard to sample from $p(x)$. Suppose that it is easy to generate a random draw from another density $p_*(x)$, called the sampling density. In this case, random draws of x from the target density $p(x)$ are generated by utilizing the random draws sampled from the sampling density $p_*(x)$. Let x_i be the i-th random draw of x generated from the target density $p(x)$. We consider generating x_1, x_2, $\cdots$, x_N from the target density $p(x)$. Suppose that $q(x)$ is proportional to the ratio of the target density and the sampling density, i.e., $q(x) \propto p(x)/p_*(x)$. Then, the target density is rewritten as: $p(x) \propto q(x)p_*(x)$.

Based on $q(x)$, the acceptance probability is computed. The random draw is generated from the sampling density $p_*(x)$. Using the acceptance probability based on $q(x)$, we can obtain the random draw from the target density $p(x)$. Depending on the structure of the acceptance probability, we have three kinds of sampling techniques, i.e., RS, IR and MH. Thus, to generate random draws of x from the target density $p(x)$, the functional form of $q(x)$ should be known and random draws have to be easily generated from the sampling density $p_*(x)$.

Now we discuss the three sampling techniques, which are the random number generation methods in the case where it is not easy to generate random draws directly from the target density. See Liu (1996) for a comparison of the three sampling methods. For all the three sampling techniques, the sampling density $p_*(x)$ is utilized, i.e., x_i is generated through the sampling density $p_*(x)$.

2.2.1 *Rejection Sampling (RS)*

Let x^* be a random draw of x generated from the sampling density $p_*(x)$. Define the acceptance probability as: $\omega(x) = q(x)/\sup_z q(z)$, where the supremum is assumed to be finite. N random draws of x from the target density $p(x)$ are obtained as follows: (i) generate x^* from the sampling density $p_*(x)$ and compute $\omega(x^*)$, (ii) set $x_i = x^*$ with probability $\omega(x^*)$ and go back to (i) otherwise, and (iii) repeat (i) and (ii) for $i = 1, 2, \cdots, N$.

RS is the most efficient sampling method in the sense of precision of the random draws, because using RS we can generate mutually independently distributed random draws. However, in order to apply RS, we need to obtain the supremum of $q(x)$. If the supremum is infinite, the acceptance probability $\omega(x)$ is zero and accordingly the candidate x^* is never accepted in Steps (i) and (ii). Let N_R be the average number of the rejected random numbers. We need $1 + N_R$ random draws in average to generate one random number from the target density $p(x)$. In other words, the rejection rate is given by $1/(1 + N_R)$ in average. Therefore, to obtain N random draws from the target density $p(x)$, we have to generate $N(1 + N_R)$ random draws from the sampling density $p_*(x)$. See, for example, Boswell, Gore, Patil and Taillie (1993), O'Hagan (1994) and Geweke (1996) for rejection sampling.

To examine the condition that $\omega(x)$ is greater than zero, consider the case where $p(x)$ and $p_*(x)$ are distributed as $N(\mu, \sigma^2)$ and $N(\mu_*, \sigma_*^2)$, respectively. Then, $\sigma_*^2 > \sigma^2$ can be derived for existence of the supremum, which implies that the sampling density $p_*(x)$ should be more broadly distributed than the target density $p(x)$. Thus, it is known that RS has the disadvantages: we need to compute the supremum in $\omega(x)$, which sometimes does not exist, and it takes a long time in the case where $\omega(\cdot)$ is close to zero even if the supremum exists.

2.2.2 *Importance Resampling (IR)*

Let x_i^* be the i-th random draw of x generated from the sampling density $p_*(x)$. The acceptance probability is defined as: $\omega(x_i^*) = q(x_i^*)/\sum_{j=1}^N q(x_j^*)$. To obtain N random draws from the target density $p(x)$, we perform the following procedure: (i) generate x_j^* from the sampling density $p_*(x)$ and compute $\omega(x_j^*)$ for all $j = 1, 2, \cdots, N$, (ii) take $x_i = x_j^*$ with probability $\omega(x_j^*)$, and (iii) repeat (ii) for $i = 1, 2, \cdots, N$.

In Step (ii), practically we need to generate a uniform random draw between zero and one, denoted by u, and set $x_i = x_j^*$ when $\Omega_{j-1} \leq u < \Omega_j$, where $\Omega_j \equiv \Omega_{j-1} + \omega(x_j^*)$ and $\Omega_0 \equiv 0$. For example, see Smith and Gelfand (1992) for the resampling procedure.

For precision of the random draws, IR is inferior to RS under the assumption for RS that the supremum in the acceptance probability exists. According to IR, when we have N different random draws from the sampling density, we pick up one of them with the corresponding probability weight. Therefore, some of the random draws have the exactly same values for IR, while all the random draws take the different values for RS. In other words, to obtain N random draws from the target density $p(x)$, IR requires just N random draws from the sampling density $p_*(x)$, but RS needs more than N random draws from the sampling density $p_*(x)$. Remember that in order for RS to generate one random draw from the target density $p(x)$ we need one accepted random draw and some rejected random draws from the sampling density $p_*(x)$.

2.2.3 *Metropolis-Hastings Independence Sampling (MH)*

Let us define the acceptance probability by: $\omega(x_{i-1}, x^*) = \min\big(q(x^*)/q(x_{i-1}), 1\big)$. N random draws of x from the target density $p(x)$ are generated as: (i) take the initial value of x as x_{-M}, (ii) generate x^* from the sampling density $p_*(x)$ and compute $\omega(x_{i-1}, x^*)$, (iii) set $x_i = x^*$ with probability $\omega(x_{i-1}, x^*)$ and $x_i = x_{i-1}$ otherwise, and (iv) repeat (ii) and (iii) for $i = -M+1, -M+2, \cdots, N$.

For choice of the sampling density $p_*(x)$, the sampling density should not have too large variance and too small variance, compared with the target density. The sampling density $p_*(x)$ should be chosen so that the chain travels over the support of the target density $p(x)$. It is also possible to take $p_*(x^*) = p_*(x^*|x_{i-1})$. See, for example, Chib and Greenberg (1995) and Geweke (1996) for MH.

Note as follows. For MH, x_1 is taken as a random draw of x from the target density $p(x)$ for sufficiently large M. To obtain N random draws, thus we need to generate $M+N$ random draws. Moreover, clearly we have $\mathrm{Cov}(x_{i-1}, x_i) > 0$, because x_i is generated based on x_{i-1}. Therefore, for precision of the random draws, MH gives us the worst random number of the three sampling methods.

As an alternative random number generation method to avoid the positive correlation, we can perform the case of $N = 1$ in the above procedures (i) – (iv) N times in parallel, taking different initial values for x_{-M}. In this case, we need to generate $M+1$ random numbers to obtain one random draw from the target density $p(x)$. That is, N random draws from the target density $p(x)$ are based on $N(1+M)$ random draws from the sampling density $p_*(x)$. Thus, we can obtain mutually independently distributed random draws. For precision of the random draws, the alternative MH is similar to RS. However, this alternative method is too computer-intensive, compared with the above procedures (i) – (iv), which takes more time RS in the case of $M > N_R$. In simulation studies of Section 4., we do not utilize this alternative MH.

3. Use of the Sampling Techniques

As discussed in Section 2.2, in order to apply the sampling techniques, the filtering density and the smoothing density have to be written as the form $p(x) \propto q(x)p_*(x)$, where $q(x)$ is the known function and $p_*(x)$ denotes the sampling density.

Let $\alpha_{i,t|s}$ be the i-th random draw of α_t from $p(\alpha_t|Y_s)$. Using the sampling techniques such as RS, IR and MH, in this section we consider generating $\alpha_{i,t|s}$. If the random draws ($\alpha_{1,t|s}$, $\alpha_{2,t|s}$, $\cdots$, $\alpha_{N,t|s}$) for $s = t, T$ and $t = 1, 2, \cdots, T$ are available, equation (2.6) is evaluated by $\mathrm{E}\big(g(\alpha_t)|Y_s\big) \approx (1/N)\sum_{i=1}^N g(\alpha_{i,t|s})$. Similarly, equation (2.7) is given by:

$$p(Y_T) \approx \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} p(y_t|\alpha_{i,t|t-1})\right), \tag{3.8}$$

where $\alpha_{i,t|t-1} = f_t(\alpha_{i,t-1|t-1}, \eta_{i,t})$ and $\eta_{i,t}$ denotes the i-th random draw of η_t. Utilizing equation (3.8), MLE is performed for estimation of the unknown parameter.

3.1 *Filtering*

Based on ($\alpha_{1,t-1|t-1}$, $\alpha_{2,t-1|t-1}$, $\cdots$, $\alpha_{N,t-1|t-1}$), an attempt is made to generate ($\alpha_{1,t|t}$, $\alpha_{2,t|t}$, $\cdots$, $\alpha_{N,t|t}$). Depending on whether the initial value α_0 is stochastic or not, $\alpha_{i,0|0}$ for $i = 1, 2, \cdots, N$ are assumed to be generated from $p_\alpha(\alpha_0)$ or to be fixed for all i.

We have two representations on the filtering density (2.4). First, as shown from equation (2.4), $p(\alpha_t|Y_t)$ is immediately rewritten as follows:

$$p(\alpha_t|Y_t) \propto q_1(\alpha_t)p(\alpha_t|Y_{t-1}), \tag{3.9}$$

where $q_1(\alpha_t)$ is given by:

$$q_1(\alpha_t) \propto p_y(y_t|\alpha_t).$$

In this case, $p_*(x)$ and $q(x)$ in Section 2.2 correspond to $p(\alpha_t|Y_{t-1})$ and $q_1(\alpha_t)$, respectively. $q_1(\alpha_t)$ is known because $p_y(y_t|\alpha_t)$ is obtained from the measurement equation (2.1), and given $\alpha_{i,t-1|t-1}$ for $i = 1, 2, \cdots, N$ a random draw of α_t from $p(\alpha_t|Y_{t-1})$ is easily generated through the transition equation (2.2). Accordingly, using the sampling techniques shown in Section 2.2, $\alpha_{i,t|t}$ can be generated based on $\alpha_{i,t|t-1}$. Note that Gordon et al. (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) proposed the IR filter based on (3.9).

When we have a structural change or an outlier at time t, the present sample $p(y_t|\alpha_t)$ is far from the one-step ahead prediction density $p(\alpha_t|Y_{t-1})$. In this case, for IR and MH the random draws of α_t from $p(\alpha_t|Y_t)$ become unrealistic because the reasonable random draws of α_t cannot be obtained from the sampling density $p(\alpha_t|Y_{t-1})$, and for RS it takes a lot of time computationally because the acceptance probability becomes very small. In addition, when a random draw of η_t is not easily obtained, it might be difficult to generate a random draw of α_t from $p(\alpha_t|Y_{t-1})$. As for the second representation of the filtering density, therefore, we explicitly introduce the importance sampling density of α_t, i.e., $p_*(\alpha_t|\alpha_{t-1})$, to obtain more plausible random draws. Furthermore, to reduce computational disadvantages, we consider generating random draws of α_t from the joint density $p(\alpha_t, \alpha_{t-1}|Y_t)$. Substituting equation (2.3) into equation (2.4) and eliminating the integration with respect to α_{t-1}, the joint density of α_t and α_{t-1} given Y_t, i.e., $p(\alpha_t, \alpha_{t-1}|Y_t)$, is written as:

$$(3.10) \qquad p(\alpha_t, \alpha_{t-1}|Y_t) \propto q_2(\alpha_t, \alpha_{t-1}) p_*(\alpha_t|\alpha_{t-1}) p(\alpha_{t-1}|Y_{t-1}),$$

where $q_2(\alpha_t, \alpha_{t-1})$ is represented by:

$$q_2(\alpha_t, \alpha_{t-1}) \propto \frac{p_y(y_t|\alpha_t) p_\alpha(\alpha_t|\alpha_{t-1})}{p_*(\alpha_t|\alpha_{t-1})}.$$

In equation (3.10), $p_*(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})$ is taken as the sampling density. When N random draws of α_{t-1} given Y_{t-1}, i.e., $\alpha_{i,t-1|t-1}$ for $i = 1, 2, \cdots, N$, are available, generating a random draw of α_{t-1} from $p(\alpha_{t-1}|Y_{t-1})$ is equivalent to choosing one out of the N random draws ($\alpha_{1,t-1|t-1}$, $\alpha_{2,t-1|t-1}$, $\cdots$, $\alpha_{N,t-1|t-1}$) with equal probability weight. Given $\alpha_{i,t-1|t-1}$, a random draw of α_t (i.e., $\alpha^*_{i,t}$) is generated from $p_*(\alpha_t|\alpha_{i,t-1|t-1})$. Thus, since the functional form of $q_2(\alpha_t, \alpha_{t-1})$ is known and the random draw of (α_t, α_{t-1}) is generated from the sampling density $p_*(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})$, the random draws of (α_t, α_{t-1}) from the target density $p(\alpha_t, \alpha_{t-1}|Y_t)$ can be obtained through RS, IR or MH. The i-th random draw of (α_t, α_{t-1}) from $p(\alpha_t, \alpha_{t-1}|Y_t)$ is denoted by $(\alpha_{i,t|t}, \alpha_{i,t-1|t})$. The random draw which we want at this stage is $\alpha_{i,t|t}$, not $\alpha_{i,t-1|t}$. Note that a random draw of α_t from $p(\alpha_t, \alpha_{t-1}|Y_t)$ is equivalent to that of α_t from $p(\alpha_t|Y_t)$. Furthermore, we point out that the appropriately chosen sampling density might be taken as $p_*(\alpha_t|\alpha_{t-1}) = p_*(\alpha_t)$, which does not depend on α_{t-1}.

To obtain the marginal density $p(\alpha_t|Y_t)$ based on (3.10), we have to integrate (3.10) with respect to α_{t-1}. The marginal density $p(\alpha_t|Y_t)$ based on (3.10) reduces to $p(\alpha_t|Y_t) \propto (1/N)\sum_{j=1}^N q_2(\alpha_t, \alpha_{j,t-1|t-1}) p_*(\alpha_t|\alpha_{j,t-1|t-1})$. Therefore, random number generation from the marginal density $p(\alpha_t|\ Y_t)$ based on (3.10) is N times as computer-intensive as that from the joint density $p(\alpha_t,\ \alpha_{t-1}|Y_t)$. This approach is adopted in numerous literature, e.g., Hürzeler and Künsch (1998), Liu and Chen (1998), Tanizaki (1996, 1999, 2000) and Tanizaki and Mariano (1998). Clearly, use of $p(\alpha_t, \alpha_{t-1}|Y_t)$ leads to much reduction in computational burden, rather than evaluation of $\int p(\alpha_t, \alpha_{t-1}|Y_t)\mathrm{d}\alpha_{t-1}$. For RS, we need to compute the supremum of q_2 with respect to α_t and α_{t-1}. Therefore, sometimes, RS is not feasible if (3.10) is utilized. For (3.10), IR or MH is recommended, rather than RS.

3.2 *Smoothing*

Given $(\alpha_{1,t+1|T}, \alpha_{2,t+1|T}, \cdots, \alpha_{N,t+1|T})$, we consider generating $(\alpha_{1,t|T}, \alpha_{2,t|T}, \cdots, \alpha_{N,t|T})$. Note that the smoothing random draws at time T are equivalent to the filtering random draws at time T, where both are represented by $\alpha_{i,T|T}$.

Based on equation (2.5), we have three representations on the smoothing density. By eliminating the integration with respect to α_{t+1} from equation (2.5), the first and second representations of $p(\alpha_{t+1}, \alpha_t|Y_T)$ are as follows:

$$
\begin{aligned}
&(3.11) \qquad & p(\alpha_{t+1}, \alpha_t|Y_T) &\propto q_3(\alpha_{t+1}, \alpha_t)p(\alpha_t|Y_t)p(\alpha_{t+1}|Y_T) \\
&(3.12) \qquad & &\propto q_4(\alpha_{t+1}, \alpha_t)p(\alpha_t|Y_{t-1})p(\alpha_{t+1}|Y_T),
\end{aligned}
$$

where q_3 and q_4 are represented by:

$$
\begin{aligned}
q_3(\alpha_{t+1}, \alpha_t) &\propto \frac{p_\alpha(\alpha_{t+1}|\alpha_t)}{p(\alpha_{t+1}|Y_t)}, \\
q_4(\alpha_{t+1}, \alpha_t) &\propto q_1(\alpha_t)q_3(\alpha_{t+1}, \alpha_t) \propto \frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_{t+1}|\alpha_t)}{p(\alpha_{t+1}|Y_t)}.
\end{aligned}
$$

In equation (3.12), $p(\alpha_t|Y_t)$ in equation (3.11) is replaced by equation (3.9). For evaluation of $p(\alpha_{t+1}|Y_t)$ in $q_3(\alpha_{t+1}, \alpha_t)$ and $q_4(\alpha_{t+1}, \alpha_t)$ of equations (3.11) and (3.12), from equation (2.3) we can use the following Monte Carlo integration:

$$
\begin{aligned}
(3.13) \qquad p(\alpha_{t+1}|Y_t) &= \int p_\alpha(\alpha_{t+1}|\alpha_t)p(\alpha_t|Y_t)\mathrm{d}\alpha_t \\
&\approx \frac{1}{N'}\sum_{j=1}^{N'} p_\alpha(\alpha_{t+1}|\alpha_{j,t|t}),
\end{aligned}
$$

where N' is not necessarily equal to N. To reduce the computational disadvantage, it might be appropriate for N' to take the number which is less than N. Because $\alpha_{1,t|t}$, $\alpha_{2,t|t}$, $\cdots$, $\alpha_{N,t|t}$ are in random order, the first N' random draws may be chosen for evaluation of the integration in equation (3.13). In any case, smoothing is N' times as computer-intensive as filtering, because of evaluation of $p(\alpha_{t+1}|Y_t)$. Thus, at each time period t, the order of computation is given by $N \times N'$ for smoothing (remember that the order of computation is N for filtering).

In equation (3.11), the sampling density is given by $p(\alpha_{t+1}|Y_T)p(\alpha_t|Y_t)$. That is, the random draw of α_t is sampled from $p(\alpha_t|Y_t)$, while that of α_{t+1} is from $p(\alpha_{t+1}|Y_T)$. Similarly, in equation (3.12), the sampling density becomes $p(\alpha_{t+1}|Y_T)p(\alpha_t|Y_{t-1})$. Thus, (3.11) is different from (3.12) with respect to the sampling density of α_t, i.e., the former is based on $p(\alpha_t|Y_t)$ while the latter is $p(\alpha_t|Y_{t-1})$. From equation (3.11) or (3.12), we can generate the random draw of (α_{t+1}, α_t) from $p(\alpha_{t+1}, \alpha_t|Y_T)$, which is denoted by $(\alpha_{i,t+1|T}, \alpha_{i,t|T})$. The random draw which we need at this stage is $\alpha_{i,t|T}$ because we already have $\alpha_{i,t+1|T}$. Thus, given $(\alpha_{1,t+1|T}, \alpha_{2,t+1|T}, \cdots, \alpha_{N,t+1|T})$, $\alpha_{i,t|T}$ is generated. Repeating the procedure for $i = 1, 2, \cdots, N$, we can obtain $(\alpha_{1,t|T}, \alpha_{2,t|T}, \cdots, \alpha_{N,t|T})$ by the backward recursion.

Again, we compare the marginal density $p(\alpha_t|Y_T)$ and the joint density $p(\alpha_{t+1}, \alpha_t|Y_T)$ from computational point of view. Random number generation from $p(\alpha_t|Y_T)$ yields N times more computational burden than that from $p(\alpha_{t+1}, \alpha_t|Y_T)$ shown in equation (3.11) or (3.12), because we have to compute $p(\alpha_t|Y_T) \propto (1/N)\sum_{j=1}^{N} q_3(\alpha_{j,t+1|T}, \alpha_t)p(\alpha_t|Y_t)$ from equation (3.11) and $p(\alpha_t|Y_T) \propto (1/N)\sum_{j=1}^{N} q_4(\alpha_{j,t+1|T}, \alpha_t)p(\alpha_t|Y_{t-1})$ from equation (3.12). Therefore, use of $p(\alpha_{t+1}, \alpha_t|Y_T)$ is much less computer-intensive than that of $p(\alpha_t|Y_T)$.

In general, filtering is approximately close to smoothing when t approaches T (i.e., the end point), because Y_t approaches Y_T as t goes to T. Therefore, in order to obtain the smoothing random draws around the end point, it might be plausible to take $p(\alpha_t|Y_t)$ for (3.11) and $p(\alpha_t|Y_{t-1})$ for (3.12) as the sampling density of α_t. However, when t goes to the starting point, possibly $p(\alpha_t|Y_t)$ or $p(\alpha_t|Y_{t-1})$ is quite different from $p(\alpha_t|Y_T)$. In the third representation, therefore, another sampling density $p_*(\alpha_t|\alpha_{t-1}, \alpha_{t+1})$ is introduced to improve the smoothing random draws especially around the starting point. Substituting equation (2.3) into equation (2.5) and eliminating the two integrations with respect to α_{t+1} and α_{t-1}, the joint density of α_{t+1}, α_t and α_{t-1} given Y_T, i.e., $p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T)$, is obtained as:

$$\begin{aligned} (3.14) \qquad & p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T) \\ & \quad \propto q_5(\alpha_{t+1}, \alpha_t, \alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})p_*(\alpha_t|\alpha_{t-1}, \alpha_{t+1})p(\alpha_{t+1}|Y_T), \end{aligned}$$

where q_5 is given by:

$$q_5(\alpha_{t+1}, \alpha_t, \alpha_{t-1}) \propto \frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{t-1})p_\alpha(\alpha_{t+1}|\alpha_t)}{p_*(\alpha_t|\alpha_{t-1}, \alpha_{t+1})p(\alpha_{t+1}|Y_t)}.$$

In equation (3.14), the sampling density is taken as $p(\alpha_{t+1}|Y_T)p_*(\alpha_t|\alpha_{t-1}, \alpha_{t+1})p(\alpha_{t-1}|Y_{t-1})$. After random draws of α_{t+1} and α_{t-1} are mutually independently generated from $p(\alpha_{t+1}|Y_T)$ and $p(\alpha_{t-1}|Y_{t-1})$, respectively, we may generate a random draw of α_t from another sampling density $p_*(\alpha_t|\alpha_{t-1}, \alpha_{t+1})$. That is, first, $\alpha_{i,t+1|T}$ and $\alpha_{i,t-1|t-1}$ are generated from $p(\alpha_{t+1}|Y_T)$ and $p(\alpha_{t-1}|Y_{t-1})$, and second, α_i^* is generated from $p(\alpha_t|\alpha_{i,t-1|t-1}, \alpha_{i,t+1|T})$. Note that a random draw of α_t from $p(\alpha_t|Y_T)$ is equivalent to that of α_t from $p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T)$. That is, $(\alpha_{i,t+1|T}, \alpha_{i,t|T}, \alpha_{i,t-1|T})$ is generated from equation (3.14), but the random draw which we want is $\alpha_{i,t|T}$ because $\alpha_{i,t+1|T}$ is already available and $\alpha_{i,t-1|T}$ can be obtained at the next stage. Moreover, note that $p_*(\alpha_t|\alpha_{t-1}, \alpha_{t+1}) = p_*(\alpha_t)$ is also a possible candidate of the appropriately chosen sampling density, where the sampling density is not a function of α_{t+1} and α_{t-1}.

3.3 *Discussion*

Both (3.9) and (3.10) are related to filtering while (3.11), (3.12) and (3.14) correspond to smoothing. Using the sampling techniques such as RS, IR and MH, the random draws of α_t are generated from (3.9) – (3.12) and (3.14). The correspondence between Sections 2.2 and 3. is summarized in Table 1, where x denotes the random variable, $p(x)$ is the target density and $q(x)$ represents the ratio of the kernel and the sampling density (see Section 2.2). Our purpose is to generate random draws of α_t from each target density.

Table 1. Correspondence between Section 2.2 and Densities (3.9) – (3.12) and (3.14)

	x	$p(x)$	$q(x)$
(3.9)	α_t	$p(\alpha_t\|Y_t)$	$p_y(y_t\|\alpha_t)$
(3.10)	(α_t, α_{t-1})	$p(\alpha_t, \alpha_{t-1}\|Y_t)$	$\dfrac{p_y(y_t\|\alpha_t)p_\alpha(\alpha_t\|\alpha_{t-1})}{p_*(\alpha_t\|\alpha_{t-1})}$
(3.11)	(α_{t+1}, α_t)	$p(\alpha_{t+1}, \alpha_t\|Y_T)$	$\dfrac{p_\alpha(\alpha_{t+1}\|\alpha_t)}{p(\alpha_{t+1}\|Y_t)}$
(3.12)	(α_{t+1}, α_t)	$p(\alpha_{t+1}, \alpha_t\|Y_T)$	$\dfrac{p_y(y_t\|\alpha_t)p_\alpha(\alpha_{t+1}\|\alpha_t)}{p(\alpha_{t+1}\|Y_t)}$
(3.14)	$(\alpha_{t+1}, \alpha_t, \alpha_{t-1})$	$p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}\|Y_T)$	$\dfrac{p_y(y_t\|\alpha_t)p_\alpha(\alpha_t\|\alpha_{t-1})p_\alpha(\alpha_{t+1}\|\alpha_t)}{p_*(\alpha_t\|\alpha_{t-1}, \alpha_{t+1})p(\alpha_{t+1}\|Y_t)}$

The IR filter which uses (3.9) has been already proposed by Gordon et al. (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) and accordingly it is not the new proposal in this paper. However, for comparison with the other estimators, we have discussed the IR filter based on (3.9) in Section 3.. The IR filter based on (3.10) and the IR smoothers with (3.11), (3.12) and (3.14) are proposed in this paper to improve the existing procedures from computational point of view. In this paper, the filtering and smoothing procedures with much less computational burden than the existing ones are derived by utilizing the joint density. The RS filters and smoothers proposed by Tanizaki (1996, 1999), Tanizaki and Mariano (1998) are substantially extended to much less computational estimators. Moreover, in this paper the Markov chain Monte Carlo smoothers discussed in Geweke and Tanizaki (1999) are developed without using the Gibbs sampler which is source of imprecise estimators. In addition, filtering is not discussed in Carter and Kohn (1994, 1996) and Geweke and Tanizaki (1999), but the MH filters are also discussed in this paper

The advantages of RS are that (i) we can generate random numbers from any density function when the supremum in the acceptance probability exists and (ii) precision of the random draws does not depend on choice of the sampling density (computational time depends on choice of the sampling density). For RS, however, the supremum has to be computed. We sometimes have the case where the supremum is not finite or the case where it is not easy to compute the supremum. Practically, it is difficult to obtain the supremums of q_2, q_3, q_4 and q_5 except for special cases. We cannot implement RS in this case. However, we can expect that there exist the supremums based on (3.9) in almost all cases. Therefore, applying RS to (3.9) might be recommended, rather than (3.10) – (3.12) and (3.14).

For all the three sampling techniques, the state estimate goes to the true value as N increases. However, under the same number of random draws, it is easily expected that RS gives us the best estimates of the three sampling techniques while MH yields the worst estimates. Using RS we can generate mutually independently distributed random draws, but the feature of MH is that a random draw is correlated with the next random draw (or the last random draw). See Section 2.2 for the three sampling techniques.

A rough measure of computing time is shown in Table 2, which represents the number of actually generated random draws for each time period t. In RS, N_R denotes

Table 2. Order of Computing Time

Sampling Method	F(8) and F(9)	S(10) – S(12)
RS	$N(1+N_R)$	$N(1+N_R)\times N'$
IR	N	$N\times N'$
MH	$N+M$	$(N+M)\times N'$
MH*	$N(1+M)$	$N(1+M)\times N'$

the average number of the rejected random draws, which implies that in average we need N_R rejected random numbers to generate one random draw from the target density. N_R depends on the functional form of both target and sampling densities. If the supremum of $q(x)$ in Section 2.2 is large, the computing time increases because the acceptance rate $\omega(x)$ is small and N_R becomes large. MH* denotes the alternative MH in which the generated random draws are mutually uncorrelated, as discussed in Section 2.2,

Furthermore, it might be possible to use different sampling techniques for filtering and smoothing. In other words, possibly we may obtain the RS filter based on (3.9) and the IR smoother based on (3.11), where computational burden can be reduced by using IR (3.11) for smoothing. Moreover, for different time period, we may combine (3.9) and (3.10) for filtering and (3.11), (3.12) and (3.14) for smoothing. To show an example, suppose that we have a structural change or an outlier at time period t', which implies that $p(\alpha_{t'}|Y_{t'-1})$ is far from $p(\alpha_{t'}|Y_{t'})$. In this case, if $p(\alpha_{t'}|Y_{t'-1})$ in equation (3.9) is taken as the sampling density, for IR and MH we cannot obtain the plausible random draws of $\alpha_{t'}$ given $Y_{t'}$ and for RS we extremely take a lot of computational time to have the random draws of $\alpha_{t'}$ given $Y_{t'}$. Therefore, as shown in equation (3.10), we can introduce another sampling density $p_*(\alpha_{t'}|\alpha_{t'-1})$ at time t' to avoid this problem. Depending on the situation which we have, we can switch the sampling density at time t' from $p(\alpha_{t'}|Y_{t'-1})$ in (3.9) to $p_*(\alpha_{t'}|\alpha_{t'-1})p(\alpha_{t'-1}|Y_{t'-1})$ in (3.10). By combining different sampling techniques between filtering and smoothing or utilizing different sampling densities at different time periods, it might be expected that the obtained filtering and smoothing solutions give us more precise and less computational state estimates.

In addition, it is also useful for filtering to take another sampling density $p_*(\alpha_t|\alpha_{t-1})$ when it is not easy to generate a random draw of α_t from $p(\alpha_t|Y_{t-1})$. That is, even though the density function of η_t is known, we have the case where it is difficult to obtain random draws of η_t. In this case, we can easily deal with this problem by utilizing $p_*(\alpha_t|\alpha_{t-1})$.

Thus, the filtering and smoothing procedures suggested in this paper is very flexible and easy to use in practice.

4. Monte Carlo Studies

4.1 *Simulation Procedure*

In this section, we examine the filters and smoothers suggested in this paper. $T = 100$ and $N = 200, 500, 1000$ are taken. See Appendix B for a discussion on the number of random draws, i.e., N. The simulation procedure is: (i) generating random numbers of ϵ_t and η_t for $t = 1, 2, \cdots, T$, compute a set of data $(y_t,\ \alpha_t)$ from equations (2.1) and (2.2), (ii) given the data set, obtain the filtering and smoothing estimates, and (iii) repeat (i) and (ii) G times and compare the root mean square error (RMSE), defined as: RMSE =

$(1/T)\ \sum_{t=1}^{T}\ \mathrm{MSE}_{t|s}^{1/2}$ for $s = t, T$, where $\mathrm{MSE}_{t|s} = (1/G)\ \sum_{g=1}^{G}\ (\overline{\alpha}_{t|s}^{(g)} - \alpha_t^{(g)})^2$ and $\overline{\alpha}_{t|s}$ takes the state estimate while α_t denotes the artificially simulated state value in (i). The superscript (g) denotes the g-th simulation run and $G = 1000$ is taken. Simulations I – V are univariate cases while Simulation VI is a multivariate case. In Simulations I – III and V, ϵ_t, η_t and α_0 are assumed to be mutually independently distributed as: $\epsilon_t \sim N(0,1)$, $\eta_t \sim N(0,1)$ and $\alpha_0 \sim N(0,1)$. The true parameter value is set to be $\delta = 0.5, 0.9, 1.0$ in Simulation I and $\delta = 0.5, 0.9$ in Simulations II and III.

Simulation I (Linear and Normal Model): Consider the univariate system: $y_t = \alpha_t + \epsilon_t$ and $\alpha_t = \delta\alpha_{t-1} + \eta_t$.

Simulation II (ARCH Model): The model is given by: $y_t = \alpha_t + \epsilon_t$ and $\alpha_t = (\delta_0 + \delta\alpha_{t-1}^2)^{1/2}\eta_t$ for $\delta_0 > 0$ and $0 \le \delta < 1$. We take $\delta_0 = 1 - \delta$, which implies that the unconditional variance of α_t is normalized to be one. (2.1) consists of the ARCH(1) process α_t and the error term ϵ_t. See Engle (1982) and Bollerslev et al. (1994) for the ARCH model.

Simulation III (Stochastic Volatility Model): Take the state space model: $y_t = \exp(0.5\alpha_t)\epsilon_t$ and $\alpha_t = \delta\alpha_{t-1} + \eta_t$ for $0 \le \delta < 1$. See Ghysels et al. (1996) for the stochastic volatility model.

Simulation IV (Nonstationary Growth Model): The system is: $y_t = \alpha_t^2/20 + \epsilon_t$ and $\alpha_t = \alpha_{t-1}/2 + 25\alpha_{t-1}\ /(1 + \alpha_{t-1}^2) + 8\cos(1.2(t-1)) + \eta_t$, where ϵ_t, η_t and α_0 are mutually independently distributed as: $\epsilon_t \sim N(0,1)$, $\eta_t \sim N(0,10)$ and $\alpha_0 \sim N(0,10)$. This model is examined in Kitagawa (1987, 1996, 1998) and Carlin et al. (1992), where the Gibbs sampler suggested by Carlin et al. (1992) does not work at all (see, for example, Tanizaki (2000)).

Simulation V (Structural Change): The data generating process is given by: $y_t = d_t + \alpha_t + \epsilon_t$ and $\alpha_t = \alpha_{t-1} + \eta_t$, but the estimated system is: $y_t = \alpha_t + \epsilon_t$ and $\alpha_t = \alpha_{t-1} + \eta_t$, where $d_t = 1$ for $t = 21, 22, \cdots, 40$, $d_t = -1$ for $t = 61, 62, \cdots, 80$ and $d_t = 0$ otherwise. This model corresponds to the case where the sudden shifts occur at time periods 21, 41, 61 and 81.

Simulation VI (Bivariate Non-Gaussian Model): We consider the following bivariate state space model: $y_t = \alpha_{1t}x_t + \alpha_{2t} + \epsilon_t$ and $\alpha_t = \alpha_{t-1} + \eta_t$, where $\alpha_t = (\alpha_{1t}, \alpha_{2t})'$ and $\eta_t = (\eta_{1t}, \eta_{2t})'$. Each density is assumed to be: $\epsilon_t \sim$ Logistic (i.e., the logistic cumulative distribution function is given by: $F(x) = (\exp(-x)+1)^{-1}$), $\eta_{1t} \sim N(0,1)$, $\eta_{2t} \sim t(3)$, and $x_t \sim U(0,1)$. For the initial value $\alpha_0 = (\alpha_{10}, \alpha_{20})'$, we take the assumptions that $\alpha_{10} \sim N(0,1)$ and $\alpha_{20} \sim t(3)$. Moreover, ϵ_t, η_{1t}, η_{2t}, x_t, α_{10} and α_{20} are assumed to be mutually independent.

4.2 *Results and Discussion*

The results are in Tables 3 – 7, where δ in Simulations I – III is assumed to be known. The values in each table represent the RMSEs defined above. The small RMSE indicates a good estimator, because RMSE represents a measure of precision of the state estimates. It might be expected that under the same number of random draws RS shows the best performance and MH indicates the worst estimator. RMSE decreases as the number of random draws (i.e., N) increases, because the simulation errors disappear as N goes to infinity. For all the tables, F and S denote filtering and smoothing, respectively.

In Table 3, F(3.9) shows the RMSE obtained from the filtering estimates based on (3.9), while S(3.11)+F(3.9) represents the RMSE from the smoothing estimates based on (3.11) with filtering density (3.9). As shown in Table 3, the filtering estimates are more volatile than the smoothing estimates, because smoothing uses more information

Table 3. F(3.9) and S(3.11)+F(3.9)

Simu-			F(3.9)			S(3.11)+F(3.9)		
lation	δ	N	RS	IR	MH	RS	IR	MH
I	0.5	200	0.7305	0.7328	0.7368	0.7088	0.7101	0.7170
		500	0.7293	0.7301	0.7316	0.7065	0.7069	0.7096
		1000	0.7289	0.7293	0.7301	0.7058	0.7060	0.7077
	0.9	200	0.7747	0.7782	0.7840	0.6880	0.6915	0.7017
		500	0.7733	0.7743	0.7768	0.6851	0.6867	0.6912
		1000	0.7729	0.7735	0.7747	0.6844	0.6851	0.6874
	1.0	200	0.7881	0.7910	0.7972	0.6769	0.6806	0.6911
		500	0.7865	0.7875	0.7908	0.6738	0.6751	0.6809
		1000	0.7861	0.7867	0.7876	0.6730	0.6743	0.6764
II	0.5	200	0.6894	0.6944	0.6999	0.6815	0.6861	0.6941
		500	0.6882	0.6907	0.6930	0.6794	0.6815	0.6852
		1000	0.6877	0.6889	0.6901	0.6783	0.6795	0.6811
	0.9	200	0.5346	0.5475	0.5505	0.5168	0.5338	0.5382
		500	0.5325	0.5389	0.5399	0.5140	0.5223	0.5239
		1000	0.5322	0.5347	0.5376	0.5135	0.5170	0.5202
III	0.5	200	0.9348	0.9360	0.9396	0.9063	0.9084	0.9149
		500	0.9332	0.9339	0.9347	0.9031	0.9036	0.9068
		1000	0.9327	0.9329	0.9338	0.9022	0.9024	0.9035
	0.9	200	1.1087	1.1105	1.1188	0.9295	0.9419	0.9547
		500	1.1064	1.1067	1.1110	0.9249	0.9319	0.9370
		1000	1.1054	1.1054	1.1076	0.9233	0.9277	0.9299
IV		200	4.6446	4.8462	5.0560	4.2119	4.3384	4.4870
		500	4.6388	4.7316	4.8166	4.2101	4.3040	4.2727
		1000	4.6377	4.6787	4.7358	4.2101	4.3179	4.2453

Table 3. F(3.9) and S(3.11)+F(3.9) — Continued

Simu-			F(3.9)			S(3.11)+F(3.9)		
lation	δ	N	RS	IR	MH	RS	IR	MH
V	0.9	200	0.8683	0.8841	0.8922	0.6998	0.7214	0.7366
		500	0.8667	0.8735	0.8775	0.6961	0.7047	0.7140
		1000	0.8662	0.8699	0.8719	0.6951	0.7000	0.7051
	1.0	200	0.8763	0.8961	0.9069	0.6868	0.7121	0.7286
		500	0.8745	0.8827	0.8876	0.6833	0.6936	0.7027
		1000	0.8739	0.8789	0.8820	0.6815	0.6881	0.6932
VI	α_{1t}	200	2.8347	2.9340	3.1353	2.2318	2.5645	2.6803
		500	2.7993	2.8585	2.9570	2.1540	2.4692	2.3945
		1000	2.7880	2.8303	2.8888	2.1083	2.4009	2.2837
	α_{2t}	200	1.9553	2.1047	2.2035	1.5639	1.8401	1.9011
		500	1.9290	2.0229	2.0812	1.5209	1.7340	1.7167
		1000	1.9220	1.9893	2.0227	1.5004	1.6869	1.6333

Table 4. Number of Rejections (N_R) in RS: $N = 1000$

Simu-		F(3.9)	S(3.11)
lation	δ		
I	0.5	3.97	2.69
	0.9	4.25	3.15
	1.0	4.36	3.31
II	0.5	4.64	0.71
	0.9	4.98	1.14
III	0.5	5.76	2.82
	0.9	6.07	3.76
IV		12.87	95.47
V	0.9	13.11	16.01
	1.0	13.55	18.99

- Note that S(3.11) is based on F(3.9).

Table 5. F(3.10) and S(3.11)+F(3.10)

Simu-			F(3.10)		S(3.11)	
lation	δ	N	IR	MH	IR	MH
I	0.9	200	0.7770	0.7829	0.6898	0.7023
		500	0.7743	0.7766	0.6867	0.6910
		1000	0.7731	0.7747	0.6848	0.6873
	1.0	200	0.7907	0.7977	0.6795	0.6932
		500	0.7879	0.7904	0.6756	0.6809
		1000	0.7866	0.7880	0.6738	0.6766
II	0.9	200	0.5476	0.5521	0.5323	0.5387
		500	0.5431	0.5457	0.5259	0.5298
		1000	0.5407	0.5414	0.5226	0.5234
III	0.9	200	1.1246	1.1587	0.9576	1.0013
		500	1.1123	1.1280	0.9384	0.9599
		1000	1.1086	1.1165	0.9320	0.9411
V	0.5	200	0.8258	0.8314	0.7549	0.7650
		500	0.8238	0.8260	0.7518	0.7563
		1000	0.8233	0.8245	0.7512	0.7537
	0.9	200	0.8706	0.8790	0.7039	0.7186
		500	0.8675	0.8709	0.6977	0.7044
		1000	0.8665	0.8684	0.6956	0.7000
	1.0	200	0.8798	0.8900	0.6930	0.7087
		500	0.8760	0.8804	0.6864	0.6938
		1000	0.8745	0.8765	0.6834	0.6874

- Note that S(3.11) is based on F(3.10).
- For F(3.10), we take $p_*(\alpha_t|\alpha_{t-1}) = N(\alpha^*_{t|t}, 9\Sigma^*_{t|t})$ in Simulations I – III and $p_*(\alpha_t|\alpha_{t-1}) = N(d_t + \delta\alpha_{t-1}, 1)$ in Simulation V.

Table 6. IR S(3.11) Based on RS F(3.9): $N = 1000$

Simu-		S(3.11)
lation	δ	IR
I	0.5	0.7054
	0.9	0.6843
	1.0	0.6732

Simu-		S(3.11)
lation	δ	IR
II	0.5	0.6780
	0.9	0.5134
III	0.5	0.9020
	0.9	0.9247

Table 7. IR S(3.11) Based on IR F(3.9): $N = 1000$

Simu- lation	$\delta \setminus N'$	1000	250	100	50	10
I	0.5	0.7060	0.7059	0.7059	0.7059	0.7060
	0.9	0.6851	0.6851	0.6853	0.6854	0.6869
	1.0	0.6743	0.6744	0.6745	0.6748	0.6764
II	0.5	0.6795	0.6795	0.6796	0.6798	0.6813
	0.9	0.5170	0.5173	0.5177	0.5198	0.5517
III	0.5	0.9024	0.9025	0.9027	0.9028	0.9045
	0.9	0.9277	0.9305	0.9326	0.9393	0.9846
IV		4.3179	4.3392	4.4116	4.5086	4.9619

Table 8. Estimation of δ Using IR F(3.9): $N = 1000$

Simu- Lation	δ	AVE	SER	10%	25%	50%	75%	90%
I	0.5	0.481	0.129	0.30	0.41	0.50	0.57	0.63
	0.9	0.881	0.059	0.80	0.85	0.89	0.92	0.94
	1.0	0.983	0.033	0.94	0.97	0.99	1.00	1.01
II	0.5	0.313	0.202	0.00	0.16	0.32	0.45	0.58
	0.9	0.670	0.201	0.44	0.54	0.67	0.81	0.98
III	0.5	0.503	0.022	0.49	0.50	0.50	0.51	0.52
	0.9	0.902	0.019	0.88	0.89	0.90	0.91	0.92

than filtering. Moreover, as it is expected, RS shows the smallest RMSE in almost all the cases. Taking an example of $\delta = 0.9$ in Simulation I, $N = 200$ of RS is equal to $N = 1000$ of MH, which implies that MH needs 5 times more random draws than RS to keep the same precision, or equivalently the acceptance rate in RS is about 20% on average (note that this is a rough interpretation because RSME is not a linear function of N). For $\delta = 0.9$ of Simulation I, $N = 500$ of RS is almost equal to $N = 1000$ of IR, which implies that IR needs twice as many random draws as RS. Taking Simulation IV, $N = 200$ of RS is better than $N = 1000$ of IR. IR needs more than 5 times as many random draws as RS. Simulation VI represents a multivariate non-Gaussian case, where RS in S(3.11)+F(3.9) represents IR S(3.11) based on RS F(3.9). Thus, in Simulation VI, we utilize IR for smoothing because it is not easy to compute the supremum of $q_3(\alpha_{t+1}, \alpha_t)$. The RMSEs are shown for both α_{1t} and α_{2t} in Simulation VI of Table 3. For both filtering and smoothing, RMSEs of RS are the smallest.

In Table 5, $p_*(\alpha_t|\alpha_{t-1})$ is introduced for filtering, but not for smoothing, where $p_*(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})$ is used for the sampling density in Table 5 while $p(\alpha_t|Y_{t-1})$ is used for sampling density in Table 3. For F(3.10) in Simulations I – III, $p_*(\alpha_t|\alpha_{t-1}) = N(\alpha^*_{t|t}, c\Sigma^*_{t|t})$ and $c = 9$ are taken. $(\alpha^*_{t|t},\ \Sigma^*_{t|t})$ denotes the mean and variance estimated by the extended Kalman filter, which is obtained by applying the linearized nonlinear measurement and transition equations directly to the standard Kalman filter formula (see, for example, Tanizaki (1996) and Tanizaki and Mariano (1996)). For F(3.10) in Simulation V, $p_*(\alpha_t|\alpha_{t-1}) = N(d_t + \delta\alpha_{t-1}, 1)$ is taken. Since it is difficult to compute the supremum of q_2, RS is not shown in Table 5. We can compare F(3.9) in Table 3 with F(3.10) in Table 5 for filtering, and S(3.11)+F(3.9) in Table 3 with S(3.11)+F(3.10) in Table 5 for smoothing. For Simulation I, the RMSEs in Table 5 are very close to those in Table 3. For Simulations II and III, however, the RMSEs in Table 3 are slightly smaller than those in Table 5 in almost all the cases. For Simulation V, Tables 3 and 5 are compared. We consider the case where the data generating process is different from the estimated state space model. We often have this case in practice, because nobody knows the ture model. For F(3.10) of Simulation V in Table 5, $p_*(\alpha_t|\alpha_{t-1}) = N(d_t + \delta\alpha_{t-1}, 1)$ is taken. In other words, in Table 5 the sampling density is appropriately specified taking into account the sudden shifts. It is expected for Simulation V that RS in Table 3 should be close to IR and MH in Table 5, rather than those in Table 3. As a result, it is shown that for IR and MH we obtain a small RMSE if the plausible sampling density $p_*(\alpha_t|\alpha_{t-1})$ is chosen, because the RMSEs of IR and MH in Table 3 are larger than those in Table 5. Thus, two types of the sampling density $p_*(\alpha_t|\alpha_{t-1})$ are shown in Table 5, although we can consider the other kinds of the sampling density.

In Table 6, S(3.11)+F(3.9) is investigated, where we utilize RS for filtering and IR for smoothing. Therefore, Table 6 should be compared with RS or IR in S(3.11)+F(3.9) of Table 3. Theoretically, the RMSEs in Table 6 should be between RS and IR in S(3.11)+F(3.9) of Table 3, since the most precise sampling method is used for filtering but the second best one is utilized for smoothing. As a result, the RMSEs in Table 6 are very close to those of RS in Table 3. It is sometimes difficult for the RS smoothers to compute the supremums based on (3.11), (3.12) and (3.14). In addition, RS takes a lot of time computationally although it is a very efficient random number generation method. The IR smoothers can be applied to almost all the nonlinear non-Gaussian state space models, which give us much less computational burden than RS. Therefore, a combination of the RS filter based on (3.9) and the IR smoother might be a useful tool, judging from computation and efficiency.

In Table 7, we investigate how sensitive the approximation of $p(\alpha_{t+1}|Y_t)$ in equation (3.13) is, where $N' = 10, 50, 100, 250, 1000$ and $N = 1000$ are taken. IR is used for the sampling method. $N' = 1000$ in Table 7 is equivalent to $N = 1000$ of IR in Table 3. We have the result that $N' = 1000$ is very close to $N' = 100, 250$ in the RMSE criterion. Since for smoothing the order of computation is $N \times N'$, we can reduce the computational burden by taking N' less than N, where we may take $N' = 0.1N - 0.25N$ from Table 7.

In Table 8, we show an example to estimate the unknown parameter maximizing the likelihood function (3.8), where IR is used for the sampling method. The likelihood function is maximized by a simple grid search. AVE, SER, 10%, 25%, 50%, 75% and 90% denote the arithmetic average, the standard error, 10th, 25th, 50th, 75th and 90th percentiles from 1000 estimates of δ. For Simulations I and III, MLE shows a good performance because AVE and 50% are close to δ. However, for Simulation II, δ is underestimated and SER is large.

Thus, in this section, we have shown some examples of RS, IR and MH for filtering and smoothing.

5. Summary

In this paper, we have shown the nonlinear non-Gaussian filtering and smothering procedures in general formulation, where RS, IR and MH are applied to generate random draws of α_t given Y_s. The existing simulation-based procedures are very computer-intensive, because conventionally they are based on the marginal density, i.e., $p(\alpha_t|Y_t)$ for filtering and $p(\alpha_t|Y_T)$ for smoothing. However, our proposal is based on the joint density, i.e., $p(\alpha_t, \alpha_{t-1}|Y_t)$ for filtering and $p(\alpha_{t+1}, \alpha_t|Y_T)$ or $p(\alpha_{t+1}, \alpha_t, \alpha_{t-1}|Y_T)$ for smoothing. To reduce computational disadvantages, in this paper we have suggested sampling from the joint densities.

It might be expected that RS gives us the most precise state estimates and that MH yields the worst of the three sampling techniques, which results are consistent with the simulation results from the Monte Carlo studies. For RS, however, we need to compute the supremum in the acceptance probability. Especially, as for (3.10) – (3.12) and (3.14), we often have the case where the supremum does not exist or the case where it is difficult to compute the supremum. Therefore, for (3.10) – (3.12) and (3.14), it is better to utilize IR, rather than RS and MH. Moreover, even though the supremum exists, computational time of RS depends on the acceptance probability. When the acceptance probability is close to zero, it takes a lot of time computationally to obtain the random draws of the state variable α_t. Both MH and IR can be applied to almost all the nonlinear non-Gaussian cases, which is one of the advantages over RS, although MH and IR are inferior to RS in the sense of precision of the state estimates. Moreover, computational burden of IR and MH does not depend on the acceptance probability. Accordingly, in the case of IR and MH, (3.9) is computationally equivalent to (3.10) for filtering and similarly (3.11), (3.12) and (3.14) give us the same computational burden for smoothing.

It is possible to take different sampling methods between filtering and smoothing, i.e., for example, RS may be taken for filtering while IR is used for smoothing (see RS in S(3.11) of Simulation VI of Table 3 and Table 6). Or at different time periods we can adopt different sampling densities. That is, taking an example of filtering, the sampling density is taken as $p_*(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})$ if $t = t'$ and $p(\alpha_t|Y_{t-1})$ otherwise. It might be useful to introduce $p_*(\alpha_t|\alpha_{t-1})$ when $p(\alpha_t|Y_t)$ is far from from $p(\alpha_t|Y_{t-1})$. See Simulation V in Table 5 for this exercise. Thus, the proposed filters and smoothers are very flexible. Taking the advantages of each sampling method, we can obtain the less computational and precise state estimates for both filtering and smoothing.

Moreover, we need to point out as follows. Smoothing is much more computer-intensive than filtering. That is, at each time period, the order of computation is N for filtering and $N \times N'$ for smoothing. Accordingly, smoothing is N' times more computer-intensive than filtering. In equation (3.13), we do not necessarily choose $N' = N$. To reduce the computational disadvantage for smoothing, from the Monte Carlo studies (i.e., Table 7) we have obtained the result that we may take $N' = 0.1N - 0.25N$.

Finally, note as follows. For comparison with the procedure suggested in this paper, the smoother based on the two-filter formula, which is developed by Kitagawa (1996), is discussed in Appendix A. We have shown that using the sampling density the smoother is also rewritten in the same fashion. For Simulations I – III, the simulations studies are examined. As a result, it is shown that the smoother based on the two-filter formula shows a good performance.

Appendix A: Fixed-Interval Smoother based on the Two-Filter Formula

Kitagawa (1996) discusses the Monte Carlo smoother based on the two-filter formula, where the same approach shown in this paper can be applied. Define $Y_t^+ \equiv \{y_t, y_{t+1}, \cdots, y_T\}$, where we have $Y_T = Y_{t-1} \cup Y_t^+$. The fixed-interval smoothing density $p(\alpha_t|Y_T)$ is represented as:

$$p(\alpha_t|Y_T) \propto p(Y_t^+|\alpha_t)p(\alpha_t|Y_{t-1}), \tag{A.1}$$

where $p(Y_t^+|\alpha_t)$ is recursively obtained as follows:

$$p(Y_t^+|\alpha_t) = p_y(y_t|\alpha_t)\int p(Y_{t+1}^+|\alpha_{t+1})p_\alpha(\alpha_{t+1}|\alpha_t)\mathrm{d}\alpha_{t+1}, \tag{A.2}$$

for $t = T-1, T-2, \cdots, 1$. The initial condition is given by: $p(Y_T^+|\alpha_T) = p_y(y_T|\alpha_T)$.

First, we consider evaluating $p(Y_t^+|\alpha_t)$ in the backward recursion. Let $p_*(\alpha_t)$ be the importance sampling density and $\alpha_{i,t}^*$ be the i-th random draw of α_t generated from $p_*(\alpha_t)$. From equation (A.2), the density $p(Y_t^+|\alpha_t)$ evaluated at $\alpha_t = \alpha_{i,t}^*$ is rewritten as:

$$\begin{aligned} p(Y_t^+|\alpha_{i,t}^*) &= p_y(y_t|\alpha_{i,t}^*)\int \frac{p(Y_{t+1}^+|\alpha_{t+1})p_\alpha(\alpha_{t+1}|\alpha_{i,t}^*)}{p_*(\alpha_{t+1})}p_*(\alpha_{t+1})\mathrm{d}\alpha_{t+1} \\ &\approx p_y(y_t|\alpha_{i,t}^*)\frac{1}{N''}\sum_{j=1}^{N''}\frac{p(Y_{t+1}^+|\alpha_{j,t+1}^*)p_\alpha(\alpha_{j,t+1}^*|\alpha_{i,t}^*)}{p_*(\alpha_{j,t+1}^*)}, \end{aligned} \tag{A.3}$$

for $t = T-1, T-2, \cdots, 1$. In the second line of the above equation, the integration is evaluated by $\alpha_{j,t+1}^*$, $j = 1, 2, \cdots, N''$, where N'' is not necessarily equal to N. Thus, $p(Y_t^+|\alpha_{i,t}^*)$ is recursively obtained for $t = T-1, T-2, \cdots, 1$. Note that the importance sampling density $p_*(\alpha_t)$ may depend on the state variable at time $t-1$, where the sampling density is given by $p_*(\alpha_t|\alpha_{t-1})$

Next, given $p(Y_t^+|\alpha_{i,t}^*)$, we generate random draws of α_t from $p(\alpha_t|Y_T)$. We can rewrite equation (A.1) as follows:

$$p(\alpha_t|Y_T) \propto q_6(\alpha_t)p(\alpha_t|Y_{t-1}), \tag{A.4}$$

where $q_6(\alpha_t) \propto p(Y_t^+|\alpha_t)$. In this case, we have to take the importance sampling density as $p_*(\alpha_t) = p(\alpha_t|Y_{t-1})$, i.e., $\alpha_{i,t}^* = \alpha_{i,t|t-1}$, where we need to evaluate $p(\alpha_{j,t+1|t}|Y_t)$ in the denominator of equation (A.3). As shown in equation (3.13), however, evaluation of $p(\alpha_{j,t+1|t}|Y_t)$ becomes N' times more computer-intensive. Therefore, it is not realistic to take the sampling density as $p_*(\alpha_t) = p(\alpha_t|Y_{t-1})$.

Alternatively, as discussed in Section 3., we can consider generating random draws from the joint density of α_t and α_{t-1} given Y_T, which is represented by:

$$p(\alpha_t, \alpha_{t-1}|Y_T) \propto q_7(\alpha_t, \alpha_{t-1}) p_*(\alpha_t) p(\alpha_{t-1}|Y_{t-1}), \tag{A.5}$$

where $q_7(\alpha_t, \alpha_{t-1}) \propto \dfrac{p(Y_t^+|\alpha_t) p_\alpha(\alpha_t|\alpha_{t-1})}{p_*(\alpha_t)}$.

As shown above, we can evaluate $p(Y_t^+|\alpha_t)$ at $\alpha_t = \alpha_{i,t}^*$. However, an explicit functional form of $p(Y_t^+|\alpha_t)$ is not obtained and it is not possible to compute the supremum of $q_6(\alpha_t)$ and $q_7(\alpha_t, \alpha_{t-1})$. Therefore, RS cannot be applied to this smoother. Thus, we may apply IR and MH to the Monte Carlo smoother based on the two-filter formula.

Taking an example of the IR smoother based on (A.5), a random number of α_t from $p(\alpha_t|Y_T)$ is generated as follows. Define the probability weight $\omega(\alpha_{i,t}^*,\ \alpha_{i,t-1|t-1})$ which satisfies $\omega(\alpha_{i,t}^*,\ \alpha_{i,t-1|t-1}) \propto q_7(\alpha_{i,t}^*,\ \alpha_{i,t-1|t-1})$ and $\sum_{i=1}^N \omega(\alpha_{i,t}^*, \alpha_{i,t-1|t-1}) = 1$. Thus, from equation (A.1), the j-th smoothing random draw $\alpha_{j,t|T}$ is resampled from $\alpha_{1,t}^*$, $\alpha_{2,t}^*$, $\cdots$, $\alpha_{N,t}^*$ with the corresponding probability weights $\omega(\alpha_{1,t}^*, \alpha_{1,t-1|t-1})$, $\omega(\alpha_{2,t}^*, \alpha_{2,t-1|t-1})$, $\cdots$, $\omega(\alpha_{N,t}^*, \alpha_{N,t-1|t-1})$. Computing time of the IR smoother based on (A.5) is the order of $N \times N''$, which is equal to IR in Table 2, while that of the IR smoother with (A.4) is $N \times N' \times N''$. Thus, for reduction of computational burden, use of (A.5) is superior to that of (A.4).

One of the computational techniques is shown as follows. The dimension of Y_t^+ increases as t is small. That is, $p(Y_t^+|\alpha_{i,t}^*)$ for all i decreases as t goes to the initial time period. Therefore, practically we have some computational difficulties such as underflow errors. To avoid the computational difficulties, we can modify equation (A.3) as follows:

$$s_t(\alpha_{i,t}^*) \propto p_y(y_t|\alpha_{i,t}^*) \sum_{j=1}^{N''} \frac{s_{t+1}(\alpha_{j,t+1}^*) p_\alpha(\alpha_{j,t+1}^*|\alpha_{i,t}^*)}{p_*(\alpha_{j,t+1}^*)},$$

where $s_t(\alpha_{i,t}^*) \propto p(Y_t^+|\alpha_{i,t}^*)$. For instance, $s_t(\alpha_t)$ may be restricted to $\sum_{i=1}^N s_t(\alpha_{i,t}^*) = 1$. We need to compute $s_t(\alpha_{i,t}^*)$ whenever we update from $t+1$ to t. Note that the proportional relation $q_7(\alpha_{i,t}^*, \alpha_{i,t-1|t-1}) \propto \dfrac{s_t(\alpha_{i,t}^*) p_\alpha(\alpha_{i,t}^*|\alpha_{i,t-1|t-1})}{p_*(\alpha_{i,t}^*)}$ still holds.

Thus, the fixed-interval smoother based on the two-filter formula, proposed by Kitagawa (1996), can be also discussed in the same context. In Table 9, the smoother based on the two-filter formula is examined for Simulations I – III. We take $p_*(\alpha_t) = N(\alpha_{t|t}, 1)$, where $\alpha_{t|t}$ represents the filtering estimate obtained from IR F(3.9). After implementing IR F(3.9), we perform IR S(A.5). Each value in Table 9 is compared with that in IR S(3.11) of Table 3. As a result, IR S(A.5) performs much better than IR S(3.11) in almost all the cases.

Table 9. IR S(A.5) Based on IR F(3.9)

Simu-			S(A.5)	Simu-			S(A.5)
lation	δ	N	IR	lation	δ	N	IR
I	0.5	200	0.7073	II	0.5	200	0.6805
		500	0.7059			500	0.6787
		1000	0.7055			1000	0.6781
	0.9	200	0.6876		0.9	200	0.5190
		500	0.6852			500	0.5160
		1000	0.6843			1000	0.5137
	1.0	200	0.6765	III	0.5	200	0.9061
		500	0.6740			500	0.9033
		1000	0.6734			1000	0.9020
					0.9	200	0.9420
						500	0.9331
						1000	0.9284

- $p_*(\alpha_t) = N(\alpha_{t|t}, 1)$ is taken, where $\alpha_{t|t}$ is obtained from IR F(3.9).

Appendix B: A Comment on the Number of Random Draws

In Section 4., $N = 200, 500, 1000$ is examined. $N = 200, 500$ is taken for comparison with $N = 1000$. Some people think that $N = 1000$ is too small. In the past research, Carlin et al. (1992) takes $N = 2500$ in Example 3.1 and $N = 500$ in Example 3.2. Gordon et al. (1993) uses $N = 500$ in Section 4.1 and $N = 4000$ in Section 4.2. Carter and Kohn (1994) is $N = 2000, 20000$, while Carter and Kohn (1996) is $N = 10000$. Kitagawa (1996) examines $N = 100, 200, \cdots, 51200$ and concludes that $N = 1000$ is recommended for a point estimation. Even in the case of the Gibbs sampling and IR, $N = 500$ is sometimes taken (see Carlin et al. (1992) and Gordon et al. (1993)). In this paper, $N = 1000$ is taken and the three sampling methods RS, MH and IR are compared with respect to the RMSE. For Simulation I to check whether $N = 1000$ is too small or not, the cases of $N = 2000$ are examined in Table 10, which should be compared with the cases of $N = 200, 500, 1000$ in Table 3. For RS and IR, the RMSEs of $N = 1000$ are quite close to those of $N = 2000$, which implies that $N = 1000$ is enough large for a point estimate. Since RMSE is the point estimate which shows a measure of precision of the state estimates, we can conclude that $N = 1000$ is not too small. However, we should keep in mind that $N = 1000$ might be too small in the case where we want to obtain a functional form of $p(\alpha_t|Y_t)$ or $p(\alpha_t|Y_T)$ by the random numbers.

References

Bollerslev, T., Engle, R.F. and Nelson, D.B. (1994). ARCH Models, *Handbook of Econometrics*, **4** (ed. R.F. Engle and D.L. McFadden), 2959–3038, North-Holland.

Boswell, M.T., Gore, S.D., Patil, G.P. and Taillie, C. (1993). The Art of Computer Generation of Random Variables, *Handbook of Statistics*, **9** (ed. C.R. Rao), 661–721, North-Holland.

Carlin, B.P., Polson, N.G. and Stoffer, D.S. (1992). A Monte Carlo Approach to Nonnormal and Nonlinear State Space Modeling, *J. Amer. Statist. Assoc.*, **87**, 493–500.

Table 10. F(3.9) and S(3.11)+F(3.9): $N = 2000$

Simu-		F(3.9)			S(3.11)+F(3.9)		
lation	δ	RS	IR	MH	RS	IR	MH
I	0.5	0.7287	0.7289	0.7294	0.7053	0.7054	0.7063
	0.9	0.7724	0.7728	0.7737	0.6838	0.6842	0.6854
	1.0	0.7858	0.7862	0.7868	0.6725	0.6731	0.6743

Carter, C.K. and Kohn, R. (1994). On Gibbs Sampling for State Space Models, *Biometrika*, **81**, 541–553.

Carter, C.K. and Kohn, R. (1996). Markov Chain Monte Carlo in Conditionally Gaussian State Space Models, *Biometrika*, **83**, 589–601.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm, *The American Statistician*, **49**, 327–335.

Doucet, A., Godsill, S. and Andrieu, C., (2000) On Sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, **10**, 197–208.

Engle, R.F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of Variance of U.K. Inflation,, *Econometrica*, **50**, 987–1008.

Geweke, J. (1996). Monte Carlo Simulation and Numerical Integration, *Handbook of Computational Economics*, **1** (ed. H.M. Amman, D.A. Kendrick and J. Rust), 731–800, North-Holland.

Geweke, J. and Tanizaki, H. (1999). On Markov Chain Monte-Carlo Methods for Nonlinear and Non-Gaussian State-Space Models, *Communications in Statistics, Simulation and Computation*, **28**, 867–894.

Ghysels, E., Harvey, A.C. and Renault, E. (1996). Stochastic Volatility, *Handbook of Statistics*, **14** (ed. G.S. Maddala and C.R. Rao), 119–191, North-Holland.

Gordon, N.J., Salmond, D.J. and Smith, A.F.M. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, *IEE Proceedings-F*, **140**, 107–113.

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

Hürzeler, M. and Künsch, H.R. (1998). Monte Carlo Approximations for General State-Space Models, *J. of Computational and Graphical Statistics*, **7**, 175–193.

Kitagawa, G. (1987). Non-Gaussian State-Space Modeling of Nonstationary Time Series (with discussion), *J. Amer. Statist. Assoc.*, **82**, 1032–1063.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State-Space Models, *J. of Computational and Graphical Statistics*, **5**, 1–25.

Kitagawa, G. (1998). A Self-Organizing State-Space Model, *J. Amer. Statist. Assoc.*, **93**, 1203–1215.

Kitagawa, G. and Gersch, W. (1996). *Smoothness Priors Analysis of Time Series* (Lecture Notes in Statistics, No.116), Springer-Verlag.

Kong, A., Liu, J.S. and Chen, R. (1998). Sequential Imputations and Bayesian Missing Data Problems, *J. Amer. Statist. Assoc.*, **89**, 278–288.

Liu, J.S. (1996). Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling, *Statistics and Computing*, **6**, 113–119.

Liu, J.S. and Chen, R. (1995). Blind Deconvolution via Sequential Imputations, *J. Amer. Statist. Assoc.*, **90**, 567–576.

Liu, J.S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems, *J. Amer. Statist. Assoc.*, **93**, 1032–1044.

Liu, J.S., Chen, R. and Wong, W.HG. (1998). Rejection Control and Sequential Importance Sampling, *J. Amer. Statist. Assoc.*, **93**, 1022–1031.

O'Hagan, A. (1994). *Kendall's Advanced Theory of Statistics*, **2B**, Edward Arnold.

Smith, A.F.M. and Gelfand, A.E. (1992). Bayesian Statistics without Tears: A Sampling-Resampling Perspective, *The American Statistician*, **46**, 84–88.

Tanizaki, H. (1996). *Nonlinear Filters: Estimation and Applications* (Second, Revised and Enlarged Edition), Springer-Verlag.

Tanizaki, H. (1999). On the Nonlinear and Nonnormal Filter Using Rejection Sampling, *IEEE Trans. Automat. Control*, **44**, 314–319.

Tanizaki, H. (2000). Nonlinear and Non-Gaussian State-Space Modeling with Monte Carlo Techniques: A Survey and Comparative Study, *Handbook of Statistics (Stochastic Processes: Modeling and Simulation)* (ed. C.R. Rao and D.N. Shanbhag), North-Holland, forthcoming.

Tanizaki, H. and Mariano, R.S. (1996). Nonlinear Filters Based on Taylor Series Expansions, *Communications in Statistics, Theory and Methods*, **25**, 1261–1282.

Tanizaki, H. and Mariano, R.S. (1998). Nonlinear and Non-Gaussian State-Space Modeling with Monte Carlo Simulations, *J. of Econometrics*, **83**, 263–290.