Nonlinear and non-gaussian state estimation: A quasi-optimal estimator

Tanizaki, Hisashi

(Citation)
Communications in Statistics: Theory and Methods,29(12):2805-2834

(Issue Date)
2000

(Resource Type)
journal article

(Version)
Accepted Manuscript

(URL)
https://hdl.handle.net/20.500.14094/90000568

Communications in Statistics, Theory and Methods, Vol.29, No.12.

NONLINEAR AND NON-GAUSSIAN STATE ESTIMATION: A QUASI-OPTIMAL ESTIMATOR

Hisashi Tanizaki
Faculty of Economics, Kobe University
Nadaku, Kobe 657-8501, Japan
(tanizaki@kobe-u.ac.jp)

Key Words: State-Space Model, Filtering, Smoothing, Nonlinear, Non-Gaussian, Rejection Sampling, Metropolis-Hastings Algorithm.

ABSTRACT

The rejection sampling filter and smoother, proposed by Tanizaki (1996, 1999), Tanizaki and Mariano (1998) and Hürzeler and Künsch (1998), take a lot of time computationally. The Markov chain Monte Carlo smoother, developed by Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996) and Geweke and Tanizaki (1999a, 1999b), does not show a good performance depending on nonlinearity and nonnormality of the system in the sense of the root mean square error criterion, which reason comes from slow convergence of the Gibbs sampler. Taking into account these problems, we propose the nonlinear and non-Gaussian filter and smoother which have much less computational burden and give us relatively better state estimates, although the proposed estimator does not yield the optimal state estimates in the sense of the minimum mean square error. The proposed filter and smoother are called the quasi-optimal filter and quasi-optimal smoother in this paper. Finally, through some Monte Carlo studies, the quasi-optimal filter and smoother are compared with the rejection sampling procedure and the Markov chain Monte Carlo procedure.

1 INTRODUCTION

Recent development of nonlinear and non-Gaussian state space modeling is based on sampling techniques. Gordon, Salmond and Smith (1993), Kitagawa

(1996, 1998) and Kitagawa and Gersch (1996) proposed a nonlinear and non-Gaussian filter and smoother by resampling. Tanizaki (1996, 1999), Tanizaki and Mariano (1998), Mariano and Tanizaki (1998) and Hürzeler and Künsch (1998) utilized rejection sampling. Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996), Chib and Greenberg (1996) and Geweke and Tanizaki (1999a, 1999b) considered using Markov chain Monte Carlo.

The importance resampling procedure suggested by Gordon, Salmond and Smith (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) yields an optimal estimator when number of random draws increases. However, smoothing needs a large amount of computation and sometimes it is impossible to obtain the smoothing solutions. Therefore, Kitagawa (1996) suggested approximating the smoothing estimates.

The acceptance/rejection sampling filter and smoother were proposed by Tanizaki (1996, 1999), Tanizaki and Mariano (1998), Mariano and Tanizaki (1998) and Hürzeler and Künsch (1998), which also give us very precise state estimation if it is implemented. For rejection sampling, however, we have the case where the acceptance probability is equal to zero. In this case, rejection sampling does not work. Also, when the acceptance probability is close to zero, it computationally takes a lot of time. Thus, rejection sampling requires the condition that the acceptance probability is greater than zero and is not close to zero.

Liu and Chen (1998) described a general framework for using sequential Monte Carlo methods in dynamic systems. The sequential importance sampling setting provides a good framework for understanding many existing methods such as the resampling procedure and the rejection sampling procedure. Their approach is still computer-intensive. In addition, they considered filtering, not smoothing. Also, for example, see Liu and Chen (1995), Berzuini, Best, Gilks and Larizza (1997), Liu, Chen and Wong (1998) and MacEachern, Clyde and Liu (1999) for sequential importance sampling.

Theoretically, the Markov chain Monte Carlo procedure proposed by Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996), Chib and Greenberg (1996) and Geweke and Tanizaki (1999a, 1999b) approaches the true value of the smoothing mean. For Gibbs sampler, however, when the random variables are highly correlated with each other, it is known that convergence speed is extremely slow, which depends on nonlinearity and nonnormality of the system. Moreover, Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996), Chib and Greenberg (1996) and Geweke and Tanizaki (1999a, 1999b) do not investigate filtering, because filtering is more computer-intensive than smoothing in this framework, which is different from the standard theory of filtering and smoothing. Note that usually smoothing is based on filtering and therefore smoothing has more computational burden than filtering. For example, see Anderson and Moore (1979), Kitagawa (1987), Harvey (1989) and Tanizaki (1996) for the standard filter and smoother.

Improving computational disadvantages of the rejection sampling procedure and the Markov chain Monte Carlo procedure, in this paper we propose a quasi-optimal nonlinear and non-Gaussian procedure, which can be applied to any nonlinear and nonnormal system. In addition, the quasi-optimal procedure yields much less computational estimator and relatively shows a good performance in the sense of the root mean square error criterion. Finally, the quasi-optimal procedure is compared with the rejection sampling procedure and the Markov chain Monte Carlo procedure through some Monte Carlo studies.

2 STATE-SPACE MODEL

As discussed in Anderson and Moore (1979), Kitagawa (1987), Harvey (1989) and Tanizaki (1996), the nonlinear and non-Gaussian state space model is described by the following two equations:

$$\text{(Measurement equation)} \qquad y_t = h_t(\alpha_t, \epsilon_t), \tag{1}$$

$$\text{(Transition equation)} \qquad \alpha_t = f_t(\alpha_{t-1}, \eta_t), \tag{2}$$

where y_t represents the observed data at time t while α_t denotes the state vector at time t which is an unobservable variable. ϵ_t and η_t are mutually independently distributed. y_t and ϵ_t are $l \times 1$ vectors and α_t and η_t are $k \times 1$ vectors. The problem is to estimate α_t given information set Y_s, where $Y_s = \{y_1, y_2, \cdots, y_s\}$. $a_{t|s} \equiv \mathrm{E}(\alpha_t|Y_s)$ is called prediction if $t > s$, filtering if $t = s$ and smoothing if $t < s$. Define $p_y(y_t|\alpha_t)$ and $p_\alpha(\alpha_t|\alpha_{t-1})$ by the density functions derived from the measurement equation (1), the transition equation (2) and the assumption of the distribution of the errors ϵ_t and η_t. The filtering algorithm is given by:

$$p(\alpha_t|Y_{t-1}) = \int p_\alpha(\alpha_t|\alpha_{t-1}) p(\alpha_{t-1}|Y_{t-1}) \mathrm{d}\alpha_{t-1}, \tag{3}$$

$$p(\alpha_t|Y_t) = \frac{p_y(y_t|\alpha_t) p(\alpha_t|Y_{t-1})}{\int p_y(y_t|\alpha_t) p(\alpha_t|Y_{t-1}) \mathrm{d}\alpha_t}, \tag{4}$$

for $t = 1, 2, \cdots, T$, where the initial condition is given by:

$$p(\alpha_1|Y_0) = \begin{cases} \displaystyle\int p_\alpha(\alpha_1|\alpha_0) p_\alpha(\alpha_0) \mathrm{d}\alpha_0, & \text{if } \alpha_0 \text{ is stochastic,} \\ p_\alpha(\alpha_1|\alpha_0), & \text{otherwise.} \end{cases}$$

Since Y_0 is an empty set, we have $p(\alpha_0|Y_0) = p_\alpha(\alpha_0)$, where $p_\alpha(\alpha_0)$ denotes the unconditional density of α_0. Equation (3) represents the one-step ahead prediction algorithm and equation (4) is called the update equation. The filtering algorithm takes the following two steps: (i) from equation (3), $p(\alpha_t|Y_{t-1})$ is obtained given $p(\alpha_{t-1}|Y_{t-1})$, and (ii) from equation (4), $p(\alpha_t|Y_t)$ is derived given $p(\alpha_t|Y_{t-1})$. Thus, prediction and updating are recursively repeated, where the filtering algorithm is based on the one-step ahead prediction density.

The smoothing algorithm is based on 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:

$$p(\alpha_t|Y_T) = p(\alpha_t|Y_t) \int \frac{p(\alpha_{t+1}|Y_T)p_\alpha(\alpha_{t+1}|\alpha_t)}{p(\alpha_{t+1}|Y_t)} \mathrm{d}\alpha_{t+1}. \tag{5}$$

Given the filtering density $p(\alpha_t|Y_t)$ and the one-step ahead prediction density $p(\alpha_{t+1}|Y_t)$, the smoothing algorithm represented by equation (5) is a backward recursion from $p(\alpha_{t+1}|Y_T)$ to $p(\alpha_t|Y_T)$. Thus, the smoothing density utilizes both the filtering density and the one-step ahead prediction density.

Let $g(\cdot)$ be a function, e.g., $g(\alpha_t) = \alpha_t$ or $g(\alpha_t) = (\alpha_t - a_{t|s})'(\alpha_t - a_{t|s})$. Once we have the density $p(\alpha_t|Y_s)$, the conditional expectation of $g(\alpha_t)$ given Y_s is given by:

$$\mathrm{E}(g(\alpha_t)|Y_s) = \int g(\alpha_t)p(\alpha_t|Y_s)\mathrm{d}\alpha_t. \tag{6}$$

Thus, prediction, filtering or smoothing solution, i.e., $\mathrm{E}(g(\alpha_t)|Y_s)$, can be computed.

When unknown parameters are included in equations (1) and (2), the following likelihood function is maximized with respect to the parameters:

$$p(Y_T) = \prod_{t=1}^{T} p(y_t|Y_{t-1}) = \prod_{t=1}^{T} \left(\int p(y_t|\alpha_t)p(\alpha_t|Y_{t-1})\mathrm{d}\alpha_t \right). \tag{7}$$

The denominator in equation (4) is utilized for the likelihood function (7). Accordingly, we do not need extra computation for evaluation of the likelihood function (7).

3 USE OF SAMPLING TECHNIQUES

3.1 Rejection Sampling Procedure

To evaluate equations (3) – (7), recently, generating random dares of α_t from $p(\alpha_t|Y_s)$ directly, the estimate of prediction, filtering or smoothing solution is computed. Especially, Tanizaki (1996, 1999), Tanizaki and Mariano (1998), Mariano and Tanizaki (1998) and Hürzeler and Künsch (1998) proposed the nonlinear and non-Gaussian filter and smoother using rejection sampling. See Appendix 1 for rejection sampling.

Filtering: Let $\alpha_{i,t|s}$ be the i-th random draw of α_t generated from $p(\alpha_t|Y_s)$. Consider generating $\alpha_{i,t|t}$, $i=1,2,\cdots,N$, given $\alpha_{i,t-1|t-1}$, $i=1,2,\cdots,N$. Using $\alpha_{i,t-1|t-1}$, the filtering density (4) is rewritten as follows:

$$\begin{aligned} p(\alpha_t|Y_t) &= \frac{1}{\gamma_t}\int p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})\mathrm{d}\alpha_{t-1} \\ &\approx \sum_{i=1}^{N}\frac{\gamma_{i,t}}{\gamma_t}\frac{1}{N}\left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}}\right) \\ &\approx \sum_{i=1}^{N}\frac{\widehat{\gamma}_{i,t}}{\widehat{\gamma}_t}\frac{1}{N}\left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}}\right) \\ &\equiv \sum_{i=1}^{N}q_{i,t}\left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}}\right), \end{aligned} \tag{8}$$

where γ_t, $\widehat{\gamma}_t$, $\gamma_{i,t}$, $\widehat{\gamma}_{i,t}$ and $q_{i,t}$ are represented as:

$$\begin{aligned} \gamma_t &\equiv \iint p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})\mathrm{d}\alpha_{t-1}\mathrm{d}\alpha_t \\ &\approx \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}p_y(y_t|\alpha_{ji,t|t-1}) \equiv \widehat{\gamma}_t, \end{aligned}$$

$$\begin{aligned} \gamma_{i,t} &\equiv \int p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\mathrm{d}\alpha_t \\ &\approx \frac{1}{N}\sum_{j=1}^{N}p_y(y_t|\alpha_{ji,t|t-1}) \equiv \widehat{\gamma}_{i,t}, \end{aligned}$$

$$q_{i,t} \equiv \frac{\widehat{\gamma}_{i,t}}{N\widehat{\gamma}_t}.$$

Note that $\alpha_{ji,t|t-1}$ denotes the j-th random draw of α_t generated from the density $p(\alpha_t|\alpha_{i,t-1|t-1})$, i.e., $\alpha_{ji,t|t-1}=f_t(\alpha_{i,t-1|t-1},\eta_{j,t})$, where $\eta_{j,t}$ denotes the j-th random draw of η_t. In the right hand side of the first equality of equation (8), equation (3) is substituted into equation (4). In the second line, the integration in the first line is approximated using $\alpha_{i,t-1|t-1}$ and in the third line, γ_t and $\gamma_{i,t}$ are approximately replaced by $\widehat{\gamma}_t$ and $\widehat{\gamma}_{i,t}$. Equation (8) implies that a random draw of α_t is generated from $p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{i,t}$ with probability $q_{i,t}$, where Tanizaki (1999) suggested using rejection sampling for random number generation from the density $p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{i,t}$. Equation (8) is a mixture of N distributions $p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{i,t}$, $i=1,2,\cdots,N$, with probability $q_{i,t}$. Therefore, based on $\alpha_{i,t-1|t-1}$, we can generate $\alpha_{i,t|t}$ recursively.

Let $p_*(z)$ be the sampling distribution, which is properly specified by a researcher. Define the acceptance probability $\omega_f(z)$ as:

$$\omega_f(z) = \frac{p_y(y_t|z)p_\alpha(z|\alpha_{i,t-1|t-1})/p_*(z)}{\sup_z\, p_y(y_t|z)p_\alpha(z|\alpha_{i,t-1|t-1})/p_*(z)}.$$

The condition $\sup_z p_y(y_t|z)p_\alpha(z|\alpha_{i,t-1|t-1})/p_*(z) < \infty$ has to be satisfied for rejection sampling. Representatively, we may choose $p_*(z) = p_\alpha(z|\alpha_{i,t-1|t-1})$.

The estimation procedure for the rejection sampling filter is as follows:

(i) Pick $\alpha_{i,t-1|t-1}$ for i with probability $q_{i,t}$.

(ii) Generate a random draw z from $p_*(\cdot)$ and a uniform random draw u from the interval between zero and one.

(iii) Take z as $\alpha_{j,t|t}$ if $u \le \omega_f(z)$ and go back to (ii) otherwise.

(iv) Repeat (i) – (iii) N times for $j = 1, 2, \cdots, N$.

(v) Repeat (i) – (iv) T times for $t = 1, 2, \cdots, T$.

Note that rejection sampling is utilized in procedures (ii) and (iii). Even though the acceptance probability is greater than zero (i.e., even though the supremum in the denominator of $\omega_f(z)$ is finite), rejection sampling is very inefficient in the sense of computational time (i.e., number of rejection increases) when the acceptance probability $\omega_f(z)$ is close to zero. In this case, (ii) and (iii) are repeated forever.

Smoothing: To generate a random draw of α_t from the smoothing density $p(\alpha_t|Y_T)$, note that each component in equation (5) is approximated as:

$$p(\alpha_t|Y_t) \approx \frac{1}{\gamma_t}\left(\frac{1}{N}\sum_{i=1}^N p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\right), \tag{9}$$

$$\int \frac{p(\alpha_{t+1}|Y_T)p(\alpha_{t+1}|\alpha_t)}{p(\alpha_{t+1}|Y_t)}\mathrm{d}\alpha_{t+1} \approx \frac{1}{N}\sum_{j=1}^N \frac{p_\alpha(\alpha_{j,t+1|T}|\alpha_t)}{p(\alpha_{j,t+1|T}|Y_t)}. \tag{10}$$

Equation (9) is equivalent to the second line in equation (8). In equation (10), suppose that $\alpha_{j,t+1|T}$, $j = 1, 2, \cdots, N$, are available, which are the random draws of α_{t+1} generated from $p(\alpha_{t+1}|Y_T)$.

Given the smoothing random draws at time $t+1$ (i.e., $\alpha_{j,t+1|T}$ for $j = 1, 2, \cdots, N$), we consider generating the smoothing random draws at time t (i.e., $\alpha_{j,t|T}$ for $j = 1, 2, \cdots, N$) using the smoothing algorithm (5). Substituting equations (9) and (10) into equation (5), the following expression is easily obtained:

$$\begin{aligned}
&p(\alpha_t|Y_T) \\
&\approx \frac{1}{\gamma_t}\left(\frac{1}{N}\sum_{i=1}^N p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\right)\left(\frac{1}{N}\sum_{j=1}^N \frac{p_\alpha(\alpha_{j,t+1|T}|\alpha_t)}{p(\alpha_{j,t+1|T}|Y_t)}\right) \\
&= \sum_{i=1}^N\sum_{j=1}^N \frac{\gamma^*_{ij,t}}{N^2\gamma_t p(\alpha_{j,t+1|T}|Y_t)}\left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma^*_{ij,t}}\right)
\end{aligned}$$

$$\begin{aligned}
&\approx \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\widehat{\gamma}^{*}_{ij,t}}{N\sum_{i=1}^{N}\widehat{\gamma}^{*}_{ij,t}} \left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma^{*}_{ij,t}}\right) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{q^{*}_{ij,t}}{N} \left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma^{*}_{ij,t}}\right), \qquad (11)
\end{aligned}$$

where $\gamma^{*}_{ij,t}$, $p(\alpha_{j,t+1|T}|Y_t)$ and $q^{*}_{ij,t}$ are given by:

$$\begin{aligned}
\gamma^{*}_{ij,t} &\equiv \int p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\mathrm{d}\alpha_t \\
&\approx \frac{1}{N}\sum_{m=1}^{N} p_y(y_t|\alpha_{mi,t|t-1})p_\alpha(\alpha_{j,t+1|T}|\alpha_{mi,t|t-1}) \equiv \widehat{\gamma}^{*}_{ij,t},
\end{aligned}$$

$$\begin{aligned}
&p(\alpha_{j,t+1|T}|Y_t) \\
&= \int p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p(\alpha_t|Y_t)\mathrm{d}\alpha_t \\
&= \frac{1}{\gamma_t}\iint p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{t-1})p(\alpha_{t-1}|Y_{t-1})\mathrm{d}\alpha_{t-1}\mathrm{d}\alpha_t \\
&\approx \frac{1}{\gamma_t}\frac{1}{N}\sum_{i=1}^{N}\int p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\mathrm{d}\alpha_t \\
&\approx \frac{1}{\gamma_t}\frac{1}{N^2}\sum_{i=1}^{N}\sum_{m=1}^{N} p_y(y_t|\alpha_{mi,t|t-1})p_\alpha(\alpha_{j,t+1|T}|\alpha_{mi,t|t-1}) \\
&= \frac{1}{\gamma_t}\frac{1}{N}\sum_{i=1}^{N}\widehat{\gamma}^{*}_{ij,t},
\end{aligned}$$

$$q^{*}_{ij,t} \equiv \frac{\widehat{\gamma}^{*}_{ij,t}}{\sum_{i=1}^{N}\widehat{\gamma}^{*}_{ij,t}}.$$

Choosing $\alpha_{i,t-1|t-1}$ and $\alpha_{j,t+1|T}$ with probabilities $q^{*}_{ij,t}$ and $1/N$, respectively, a random draw of α_t is generated from $p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})$ $/\gamma^{*}_{ij,t}$ using rejection sampling.

Define the acceptance probability $\omega_s(z)$ as:

$$\omega_s(z) = \frac{p_y(y_t|z)p_\alpha(\alpha_{j,t+1|T}|z)p_\alpha(z|\alpha_{i,t-1|t-1})/p_*(z)}{\sup\limits_{z}\, p_y(y_t|z)p_\alpha(\alpha_{j,t+1|T}|z)p_\alpha(z|\alpha_{i,t-1|t-1})/p_*(z)}.$$

the condition $\sup\limits_{z}\, p_y(y_t|z)p_\alpha(\alpha_{j,t+1|T}|z)p_\alpha(z|\alpha_{i,t-1|t-1}) < \infty$ has to be satisfied for rejection sampling. Under the setup, the estimation procedure for the rejection sampling smoother is as follows:

(i) Pick $\alpha_{j,t+1|T}$ for j with probability $1/N$ and choose $\alpha_{i,t-1|t-1}$ for i with probability $q^{*}_{ij,t}$.

(ii) Generate a random draw z from $p_*(\cdot)$ and a uniform random draw u from the interval between zero and one.

(iii) Take z as $\alpha_{m,t|T}$ if $u \leq \omega_s(z)$ and go back to (ii) otherwise.

(iv) Repeat (i) – (iii) N times for $m = 1, 2, \cdots, N$.

(v) Repeat (i) – (iv) T times for $t = 1, 2, \cdots, T$.

In equations (8) and (11), $\widehat{\gamma}_t$, $\widehat{\gamma}_{i,t}$ and $\widehat{\gamma}^*_{ij,t}$ are too computer-intensive. The computational reduction method is discussed later in Remark 1.

Likelihood Function: From the definition of γ_t, the likelihood function is evaluated as:

$$p(Y_T) = \prod_{t=1}^{T} \gamma_t \approx \prod_{t=1}^{T} \widehat{\gamma}_t.$$

Remark 1 (Computation Reduction Method): The procedure shown above involves some computational difficulties. First, it is well known that rejection sampling takes a lot of time computationally when the acceptance probability is too small. In order to reduce the computational burden, for one strategy, we may switch random number generation from rejection sampling to the Metropolis-Hastings algorithm. That is, while generating a random draw by rejection sampling, we perform the Metropolis-Hastings algorithm in parallel and take the random draw as $\alpha_{j,t|t}$ or $\alpha_{m,t|T}$ for enough large number of iteration if any generated random draw is not accepted by rejection sampling. Second, it takes a lot of time to compute $\widehat{\gamma}_{i,t}$ and $\widehat{\gamma}^*_{ij,t}$ in equations (8) and (11). Therefore, another strategy is that we may approximately use $\alpha_{i,t|t-1} = f_t(\alpha_{i,t-1|t-1}, \eta_{i,t})$ to evaluate $\widehat{\gamma}_t$, $\widehat{\gamma}_{i,t}$ and $\widehat{\gamma}^*_{ij,t}$, i.e.,

$$\begin{cases} \widehat{\gamma}_t \approx \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} p_y(y_t|\alpha_{i,t|t-1}), \\ \widehat{\gamma}_{i,t} \approx p_y(y_t|\alpha_{i,t|t-1}), \\ \widehat{\gamma}^*_{ij,t} \approx p_y(y_t|\alpha_{i,t|t-1}) p_\alpha(\alpha_{j,t+1|T}|\alpha_{i,t|t-1}). \end{cases} \tag{12}$$

Even though we take these strategies to reduce computational burden, however, the nonlinear and non-Gaussian procedure shown in this section is still too computer-intensive. Therefore, we consider less computational filter and smoother in Section 4.

3.2 Markov Chain Monte Carlo Procedure

Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996) and Chib and Greenberg (1996) introduced the nonlinear and non-Gaussian state-space models with Gibbs sampling. See Appendix 2 for the Gibbs sampler. They investigated the nonlinear state-space models in the Bayesian framework and they studied the special state-space models such that it is easy to generate random draws from the underlying assumptions. To improve these problems, Geweke and Tanizaki (1999a) proposed the nonlinear and non-Gaussian smoother using both Gibbs sampling and the Metropolis-Hastings algorithm, which would be suitable to any nonlinear and non-Gaussian state-space models. See Appendix 3 for the Metropolis-Hastings algorithm. The nonlinear and non-Gaussian smoother proposed by Geweke and Tanizaki (1999a) is discussed in this section.

Filtering and Smoothing: Let us define A_s and A_s^* as $A_s = \{\alpha_0,\ \alpha_1,\ \cdots,\ \alpha_s\}$ and $A_s^* = \{\alpha_s,\ \alpha_{s+1},\ \cdots,\ \alpha_t\}$, respectively, where $s \le t$. We consider generating random draws of A_t directly from $p(A_t|Y_t)$. According to the Gibbs sampling theory, random draws of A_t from $p(A_t|Y_t)$ are based on those of α_s from $p(\alpha_s|A_{s-1}, A_{s+1}^*, Y_t)$ for $s = 1, 2, \cdots, t$, which is given by the following conditional distribution function:

$$
\begin{aligned}
&p(\alpha_s|A_{s-1}, A_{s+1}^*, Y_t) \\
&= \frac{p_y(Y_t|A_t)p_\alpha(A_t)}{\int p_y(Y_t|A_t)p_\alpha(A_t)\mathrm{d}\alpha_s} \\
&\propto \begin{cases} p_y(y_s|\alpha_s)p_\alpha(\alpha_s|\alpha_{s-1})p_\alpha(\alpha_{s+1}|\alpha_s), & \text{if } s = 1, 2, \cdots, t-1, \\ p_y(y_s|\alpha_s)p_\alpha(\alpha_s|\alpha_{s-1}), & \text{if } s = t \text{ (i.e., endpoint)}, \end{cases}
\end{aligned} \tag{13}
$$

where $p_y(Y_t|A_t)$ and $p_\alpha(A_t)$ are given by:

$$
\begin{aligned}
p_y(Y_t|A_t) &= \prod_{s=1}^{t} p_y(y_s|\alpha_s), \\
p_\alpha(A_t) &= \begin{cases} p_\alpha(\alpha_0)\displaystyle\prod_{s=1}^{t} p_\alpha(\alpha_s|\alpha_{s-1}), & \text{if } \alpha_0 \text{ is stochastic}, \\ \displaystyle\prod_{s=1}^{t} p_\alpha(\alpha_s|\alpha_{s-1}), & \text{otherwise}. \end{cases}
\end{aligned}
$$

Equation (13) implies that a kernel of $p(\alpha_s|A_{s-1}, A_{s+1}^*, Y_t)$ is given by $p_y(y_s|\alpha_s)$ $p_\alpha(\alpha_s|\alpha_{s-1})p_\alpha(\alpha_{s+1}|\alpha_s)$ when $s = 1, 2, \cdots, t-1$ and $p_y(y_s|\alpha_s)p_\alpha(\alpha_s|\alpha_{s-1})$ when $s = t$ (i.e., endpoint).

Using a kernel of $p(\alpha_s|A_{s-1}, A_{s+1}^*, Y_t)$, we generate random draws of A_t directly from $p(A_t|Y_t)$. Here, the Gibbs sampler is applied to random number generation. Let $\alpha_{i,s|t}$ be the i-th random draw of the state vector at time s given information Y_t. Denote $A_{i,s|t}$ and $A_{i,s|t}^*$ by $A_{i,s|t} = \{\alpha_{i,0|t},\ \alpha_{i,1|t},\ \cdots,\ \alpha_{i,s|t}\}$ and $A_{i,s|t}^* = \{\alpha_{i,s|t},\ \alpha_{i,s+1|t},\ \cdots,\ \alpha_{i,t|t}\}$, respectively, which are the i-th random draws of A_s and A_s^* given Y_t. Let $p_*(z|x)$ be the proposal density, which is the conditional distribution of z given x. Define the acceptance probability $\omega(x,z)$ as follows:

$$\omega(x,z)=\begin{cases}\min\left(\dfrac{p(z|A_{i,s-1|t},A^*_{i-1,s+1|t},Y_t)p_*(x|z)}{p(x|A_{i,s-1|t},A^*_{i-1,s+1|t},Y_t)p_*(z|x)},1\right), & \\ \qquad\qquad\qquad\qquad \text{if } p(x|A_{i,s-1|t},A^*_{i-1,s+1|t},Y_t)p_*(z|x)>0, & \\ 1, \qquad\qquad\qquad\qquad \text{otherwise.} & \end{cases}$$

To generate random draws of α_t from $p(A_t|Y_t)$, the following procedure is taken:

(i) Pick up appropriate values for $\alpha_{0,s|t}$, $s=1,2,\cdots,t$. For example, we may take $\alpha_{0,s|t}=\alpha_{N,s|t-1}$ for $s=1,2,\cdots,t-1$ and $\alpha_{0,s|t}=a^*_{t|t}$ for $s=t$, where $a^*_{t|t}$ denotes the state estimate from the extended Kalman filter.

(ii) Generate a random draw of α_0 (i.e., $\alpha_{i,0|t}$) from the initial density $p_\alpha(\alpha_0)$ if α_0 is stochastic or take $\alpha_{i,0|t}=\alpha_0$ if α_0 is fixed.

(iii) Generate a random draw z from $p_*(\cdot|\alpha_{i-1,s|t})$ and a uniform random draw u from the uniform distribution between zero and one.

(iv) Set $\alpha_{i,s|t}=z$ if $u\le\omega(\alpha_{i-1,s|t},z)$ and $\alpha_{i,s|t}=\alpha_{i-1,s|t}$ otherwise.

(v) Repeat (iii) and (iv) for $s=1,2,\cdots,t$.

(vi) Repeat (ii) – (v) for $i=1,2,\cdots,N$.

(vii) Repeat (i) – (vi) for $t=1,2,\cdots,T$.

In Step (i), the extended Kalman filter is one of the traditional nonlinear filters, where the nonlinear measurement and transition equations are linearized by the first-order Taylor series expansion and the linearized system is directly applied to the standard linear recursive Kalman filter algorithm. See, for example, Tanizaki (1993, 1996), Tanizaki and Mariano (1996) and Wishner, Tabaczynski and Athans (1969). The Metropolis-Hastings algorithm is used in Steps (iii) and (iv) to generate a random draw of α_s from $p(\alpha_s|A_{i,s-1|t},A^*_{i-1,s+1|t},Y_t)$. We can obtain the i-th random draw of α_t given Y_t (i.e., the i-th filtering random draw $\alpha_{i,t|t}$) when $s=t$ in Step (v) and the i-th random draw of α_t given Y_T (i.e., the i-th smoothing random draw $\alpha_{i,t|T}$) when $t=T$ in Step (vii). Thus, $\alpha_{i,t|t}$ and $\alpha_{i,t|T}$ for $i=1,2,\cdots,N$ are obtained from the above procedure (i) – (vii). From the basic results of the Gibbs sampler and the Metropolis-Hastings algorithm, $\alpha_{N,t|t}$ and $\alpha_{N,t|T}$ are equivalent to the random draws based on $p(\alpha_t|Y_t)$ and $p(\alpha_t|Y_T)$ for sufficiently large N,

For smoothing, in order to reduce computational burden, we may erase Step (vii) and replace Step (v) as follows:

(v) Repeat (iii) and (iv) for $s=1,2,\cdots,T$.

Note that we cannot obtain the filtering estimates in this case.

The Markov chain Monte Carlo procedure needs a numerous number of random draws, compared with the independence Monte Carlo such as rejection sampling. For one reason, it is well known that a random draw is positively correlated with the next random draw in general. For another reason, the first M random draws are discarded from consideration, where M denotes the number of random draws to be required for convergence to those from the target density and usually 10% – 20% of N is taken for M. In addition, convergence of the Gibbs sampler is very slow, depending on the underlying joint distribution, especially in the case where there is high correlation between α_t and α_{t-1} (see Chib and Greenberg (1995)).

Proposal Density: The Metropolis-Hastings algorithm has the problem of specifying the proposal density, which is the crucial criticism. We should choose the proposal density $p_*(z|x)$ such that random draws can be easily and quickly generated. Several generic choices of the proposal density are discussed by Tierney (1994) and Chib and Greenberg (1995). We may take the following several candidates for the proposal density function $p_*(z|x)$. First, it might be natural to take the density function obtained from the transition equation (2), i.e., $p_*(z|x) = p_\alpha(z|\alpha_{i,s-1|t})$, which is examined in Monte Carlo experiments of Section 5, In this case, $p_*(z|x)$ does not depend on x, i.e., $p_*(z|x) = p_*(z)$, which is called the independence chain. Second, it is also possible to utilize the extended Kalman smoothed estimates, i.e., $p_*(z|x) = N(a^*_{s|T}, c\Sigma^*_{s|T})$ (this is also the independence chain), where $a^*_{s|T}$ and $\Sigma^*_{s|T}$ denote the first- and the second-moments (i.e., mean and variance) based on the extended Kalman smoothed estimates at time t and c is an appropriate constant value, e.g., $c = 1, 2, \cdots$. Third, we may take the proposal density called the random walk chain, i.e., $p_*(z|x) = p_*(z - x)$, which is written as $p_*(z|x) = N(x, c\Sigma^*_{s|T})$. Fourth, in the case where the state variable α_s lies on an interval, a uniform distribution between the interval might be taken as the proposal density.

4 QUASI-OPTIMAL PROCEDURE ON FILTERING AND SMOOTHING

For the rejection sampling filter and smoother, it takes a lot of time to choose i even though we use the computation reduction method as in (12) for $\widehat{\gamma}_t$, $\widehat{\gamma}_{i,t}$ and $\widehat{\gamma}^*_{ij,t}$. Moreover, when the rejection sampling filter and smoother are implemented, Steps (ii) and (iii) in Section 3.1 are repeated until z is accepted with probability $\omega_f(z)$ or $\omega_s(z)$. Accordingly it takes much computational time to be accepted, especially in the case where the acceptance probability is close to zero.

For the Markov chain Monte Carlo procedure, the Gibbs sampler and the Metropolis-Hastings algorithm are simultaneously utilized. In the case where the random variables are correlated with each other, it is known that convergence of Gibbs sampling is very slow. Since the transition equation (2) indicates the relationship between α_t and α_{t-1}, it is easily expected that α_t is highly correlated with α_{t-1}, Therefore, α_1, α_2, $\cdots$, α_T are correlated with each other. Under this situation, it takes a long time computationally to obtain the random draws of A_T generated from $p(A_T|Y_T)$, because we need to generate a numerous number of random draws for convergence. Thus, we should keep in mind that we have the case where the random draws extremely slowly converge to those from the target density function.

To improve these computational problems, the quasi-optimal filter and the quasi-optimal smoother are proposed in this paper. As N goes to infinity, the state variables estimated by rejection sampling and Markov chain Monte Carlo approach the true values. However, the state estimates by the quasi-optimal filter and quasi-optimal smoother do not go to the true values even though N is large, which is the reason why we use the word "quasi" in this paper. Large N results in extremely large computational burden for both the rejection sampling procedure and the Markov chain Monte Carlo procedure. Accordingly we turn out to choose the finite number of random draws (i.e., N). Therefore, the satisfactory results are not necessarily obtained from the two procedures. We consider much less computer-intensive but relatively efficient estimator in the root mean square error criterion.

Filtering: Approximating $q_{i,t}$ as $1/N$ in equation (8), the filtering density $p(\alpha_t|Y_t)$ is given by:

$$p(\alpha_t|Y_t) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}}\right). \tag{14}$$

Implications of $q_{i,t} \approx 1/N$ are discussed later in Remark 2. Given i, consider generating a random draw of α_t (i.e., denoted by $\alpha_{i,t|t}$) from the target density $p_y(y_t|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{i,t}$ by the Metropolis-Hastings algorithm. Define the acceptance probability $\omega_f(x,z)$ as:

$$\omega_f(x,z) = \begin{cases} \min\left(\dfrac{p_y(y_t|z)p_\alpha(z|\alpha_{i,t-1|t-1})p_*(x|z)}{p_y(y_t|x)p_\alpha(x|\alpha_{i,t-1|t-1})p_*(z|x)}, 1\right), \\ \qquad\qquad\qquad\qquad \text{if } p_y(y_t|x)p_\alpha(x|\alpha_{i,t-1|t-1})p_*(z|x) > 0, \\ 1, \qquad\qquad\qquad\quad\; \text{otherwise.} \end{cases}$$

Based on equation (14), the estimation procedure for the quasi-optimal filter is as follows:

(i) Pick up appropriate values for $\alpha_{1,0|0}$ and $\alpha_{0,t|t}$, $t = 1, 2, \cdots, T$.

(ii) Generate a random draw z from $p_*(\cdot|\alpha_{i-1,t|t})$ and a uniform random draw u from the uniform distribution between zero and one.

(iii) Set $\alpha_{i,t|t} = z$ if $u \le \omega_f(\alpha_{i-1,t|t}, z)$ and set $\alpha_{i,t|t} = \alpha_{i-1,t|t}$ otherwise.

(iv) Repeat (ii) and (iii) for $t = 1, 2, \cdots, T$.

(v) Repeat (ii) – (iv) for $i = 1, 2, \cdots, N$.

Thus, given $\alpha_{i,t-1|t-1}$, we can generate $\alpha_{i,t|t}$, which implies a recursive algorithm on random draws. For the rejection sampling filter, first the procedure procedure is implemented for $i = 1, 2, \cdots, N$ to compute $\widehat{\gamma}$ and $\widehat{\gamma}_{i,t}$ and second it is performed for $t = 1, 2, \cdots, T$. However, for the quasi-optimal filter, the estimation procedure is implemented by the order of $t = 1, 2, \cdots, T$ and $i = 1, 2, \cdots, N$.

Remark 2 ($q_{i,t} \approx 1/N$): For simplicity of discussion, we consider $p(x) = \sum_{i=1}^N w_i p_i(x)$ as the target density, where each component is represented as: $p(x) = p(x|Y_t)$, $w_i = q_{i,t}$ and $p_i(x) = p_y(y_t|x)p_\alpha(x|\alpha_{i,t-1|t-1})/\gamma_{i,t}$ in this case.

Under the setup, the approximation $q_{i,t} \approx 1/N$ shown in equation (14) implies that $p(x) = \sum_{i=1}^N w_i p_i(x)$ is approximated as $p(x) \approx (1/N)\sum_{i=1}^N p_i(x)$, where $\sum_{i=1}^N w_i = 1$. Let us define $\mu_i = \int x p_i(x)\mathrm{d}x$ and $\sigma_i^2 = \int (x-\mu)^2 p_i(x)\mathrm{d}x$.

The expectation of x based on the true density $p(x) = \sum_{i=1}^N w_i p_i(x)$ is given by:

$$\sum_{i=1}^N w_i \int x p_i(x)\mathrm{d}x = \sum_{i=1}^N w_i \mu_i \equiv \mu.$$

The expectation of x from the approximated density $p(x) \approx (1/N)\sum_{i=1}^N p_i(x)$ is obtained as:

$$\frac{1}{N}\sum_{i=1}^N \int x p_i(x)\mathrm{d}x = \frac{1}{N}\sum_{i=1}^N \mu_i,$$

which is not necessarily equal to $\sum_{i=1}^N w_i\mu_i$. When $\mu_i = \mu$ for all i, mean from the true density is equal to that from the approximated one.

Variance obtained from the true density is given by:

$$\sum_{i=1}^N w_i \int (x-\mu)^2 p_i(x)\mathrm{d}x = \sum_{i=1}^N w_i \sigma_i^2 \equiv \sigma^2,$$

and that from the approximated density is:

$$\frac{1}{N}\sum_{i=1}^N \int (x-\mu)^2 p_i(x)\mathrm{d}x = \frac{1}{N}\sum_{i=1}^N \sigma_i^2.$$

When $\sigma_i^2 = \sigma^2$ for all i, variance from the true density is equal to that from the approximated one. If σ_i^2 is small when w_i is large, it can be easily verified that $\sum_{i=1}^N w_i\sigma_i^2$ is smaller than $(1/N)\sum_{i=1}^N \sigma_i^2$. In this paper, the former corresponds to variance from the rejection sampling approach while the latter is associated with the quasi-optimal approach. Thus, the rejection sampling filter is more efficient than the quasi-optimal filter.

From the above results, the true density is equivalent to the approximated one when $p_i(x) = p(x)$ for all i. This case corresponds to the condition that (i) $\alpha_{i,t-1|t-1}$ is constant for all i or (ii) $p_y(y_t|x)p_\alpha(x|\alpha_{i,t-1|t-1})$ is flat with respect to $\alpha_{i,t-1|t-1}$. For (i), if $p(\alpha_{t-1}|Y_{t-1})$ is distributed within narrow range, i.e., if variance of α_{t-1} given Y_{t-1} is small, $\alpha_{i,t-1|t-1}$, $i = 1, 2, \cdots, N$, are close enough to each other. (ii) implies the approximation: $p(\alpha_t|Y_t) \approx p(\alpha_t|y_t)$.

Smoothing: In equation (11), approximating $q_{ij,t}^*$ as $1/N$, the smoothing density $p(\alpha_t|Y_T)$ is represented as:

$$\begin{aligned} p(\alpha_t|Y_T) &\approx \frac{1}{N^2}\sum_{i=1}^N\sum_{j=1}^N \left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{ij,t}^*}\right) \\ &\approx \frac{1}{N}\sum_{i=1}^N \left(\frac{p_y(y_t|\alpha_t)p_\alpha(\alpha_{i,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{ii,t}^*}\right) \end{aligned} \tag{15}$$

Implications of the second line in equation (15) and $q_{ij,t}^* \approx 1/N$ are discussed later in Remark 3. From equation (15), given i, consider generating a random draw of α_t (i.e., denoted by $\alpha_{i,t|T}$) from the target density $p_y(y_t|\alpha_t)$ $p_\alpha(\alpha_{i,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{ii,t}^*$. Define the acceptance probability $\omega_s(x,z)$ as:

$$\omega_s(x,z) = \begin{cases} \min\left(\dfrac{p_y(y_t|z)p_\alpha(\alpha_{i,t+1|T}|z)p_\alpha(z|\alpha_{i,t-1|t-1})p_*(x|z)}{p_y(y_t|x)p_\alpha(\alpha_{i,t+1|T}|x)p_\alpha(x|\alpha_{i,t-1|t-1})p_*(z|x)}, 1\right), \\ \qquad\qquad \text{if } p_y(y_t|x)p_\alpha(\alpha_{i,t+1|T}|x)p_\alpha(x|\alpha_{i,t-1|t-1})p_*(z|x) > 0, \\ 1, \qquad\quad \text{otherwise.} \end{cases}$$

From equation (15), the estimation procedure for the quasi-optimal smoother is implemented as follows:

(i) Pick up appropriate values for $\alpha_{0,t|T}$, $t = 1, 2, \cdots, T$.

(ii) Generate a random draw z from $p_*(\cdot|\alpha_{i-1,t|T})$ and a uniform random draw u from the uniform distribution between zero and one.

(iii) Set $\alpha_{i,t|T} = z$ if $u \le \omega_s(\alpha_{i-1,t|T}, z)$ and $\alpha_{i,t|T} = \alpha_{i-1,t|T}$ otherwise.

(iv) Repeat (ii) and (iii) for $t = T-1, T-2, \cdots, 1$.

(v) Repeat (ii) – (iv) for $i = 1, 2, \cdots, N$.

Thus, given $\alpha_{i,t+1|T}$, we can generate $\alpha_{i,t|T}$, which implies a backward recursive algorithm on random draws. Note that the filtering random draw $\alpha_{i,t-1|t-1}$ is utilized to obtain the smoothing random draw $\alpha_{i,t|T}$. Therefore, $\alpha_{i,t|T}$ is based on $\alpha_{i,t-1|t-1}$ and $\alpha_{i,t+1|T}$. For the rejection sampling smoother, first the procedure is implemented for $i,j = 1,2,\cdots,N$ to compute $\widehat{\gamma}^*_{ij,t}$ and second it is performed for $t = 1,2,\cdots,T$. However, for the quasi-optimal smoother, the estimation procedure is implemented by the order of $t = 1,2,\cdots,T$ and $i = 1,2,\cdots,N$, which is different from the rejection sampling smoother in the sense of computation order.

Remark 3 (the 2nd Line in Eq. (15) and $q^*_{ij,t} \approx 1/N$): If we generate α_{t+1} and α_{t-1} separately from the two marginal densities $p(\alpha_{t+1}|Y_T)$ and $p(\alpha_{t-1}|Y_{t-1})$, we obtain $(\alpha_{j,t+1|T},\ \alpha_{i,t-1|t-1})$. However, by generating α_{t+1} and α_{t-1} simultaneously from the joint density $p(\alpha_{t+1}|Y_T)p(\alpha_{t-1}|Y_{t-1})$, we can obtain $(\alpha_{i,t+1|T},\ \alpha_{i,t-1|t-1})$. Thus, we have the second line in equation (15) by generating random draws from the joint density.

As in Remark 2, if $\alpha_{j,t+1|T}$ is constant for all j and $\alpha_{i,t-1|t-1}$ is also constant for i, we have $q^*_{ij,t} \approx 1/N$. Or if $p_y(y_t|\alpha_t)p_\alpha(\alpha_{j,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma^*_{ij,t}$ is flat with respect to both $\alpha_{j,t+1|T}$ and $\alpha_{i,t-1|t-1}$, we have $q^*_{ij,t} \approx 1/N$.

Mean, Variance and Likelihood Function: When the random draws (i.e., $\alpha_{i,t|s}$) are available, evaluation of $\mathrm{E}(g(\alpha_r)|Y_s)$ is given by:

$$\mathrm{E}(g(\alpha_t)|Y_s) \approx \frac{1}{N-M}\sum_{i=M+1}^{N} g(\alpha_{i,t|s}).$$

Usually, 10 – 20% of N is taken for M, which implies that the first M random draws are excluded because of stability of the random draws. In Monte Carlo studies of Section 5, we take $M = 0.2N$.

When the unknown parameters are included in the system, the likelihood function (7) to be estimated is represented as:

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

where $\alpha_{i,t|t-1}$ is obtained from the transition equation $\alpha_{i,t|t-1} = f_t(\alpha_{i,t-1|t-1}, \eta_{i,t})$ given a random draw of η_t (i.e., $\eta_{i,t}$).

Proposal Density: For the quasi-optimal filter, we may choose the following proposal densities: (A) $p_\alpha(z|\ \alpha_{i,t-1|t-1})$, (B) $N(a^*_{t|t}, c\Sigma^*_{t|t})$ and (C) $N(x, c\Sigma^*_{t|t})$ as the candidates of the proposal density $p_*(z|x)$. $c = 1,2,4,9,16,25$ is taken in Monte Carlo studies of Section 5. $a^*_{t|t}$ and $\Sigma^*_{t|t}$ denote the mean and variance estimated by the extended Kalman filter.

Table 1: Comparison of Computational Time (Number of Iteration)

	Filtering	Smoothing
EK	T	$T+(T-1)$
RS	$(N^2+(N+A+1)N)T$	$(N^2+(N+A+1)N)T$ $+(N^2+(N+A+1))N(T-1)$
RS*	$(N+(N+A+1)N)T$	$(N+(N+A+1)N)T$ $+(N+(N+A+1))N(T-1)$
MCMC	$NT(T-1)/2$	$NT(T-1)/2$
QUASI	NT	$NT+N(T-1)$

For the quasi-optimal smoother, similarly, as discussed above, (A) $p_\alpha(z|\alpha_{i,t-1|T})$, (B) $N(a^*_{t|T}, c\Sigma^*_{t|T})$ and (C) $N(x, c\Sigma^*_{t|T})$ might be plausible for the candidates of the proposal density. $a^*_{t|T}$ and $\Sigma^*_{t|T}$ denote the mean and variance estimated by the extended Kalman smoother.

(A) and (B) are called the independence chain while (C) is called the random walk chain. See Chib and Greenberg (1995).

Order of Computation: For the rejection sampling filter, the order of computation is $(N^2+(N+A+1)N)T$, because we need the order of N^2 in computing $\widehat{\gamma}_{i,t}$ for each t, and that of N in choosing i and that of $(A+1)$ in implementing rejection sampling for each i and t, where A denotes the average number of rejection. The Markov chain Monte Carlo filter requires the order of $\sum_{t=1}^T tN = NT(T-1)/2$. However, for the quasi-optimal filter, the order of computation is only TN. Therefore, the quasi-optimal filter is $N+(N+A+1)$ times faster than the rejection sampling filter in computational time and $(T-1)/2$ times faster than the Markov chain Monte Carlo filter. According to the computation reduction method shown in (12), since computation of $\widehat{\gamma}_{i,t}$ is the order of N for each t, the computational burden of the rejection sampling filter reduces to the order of $(N+(N+A+1)N)T$. In this case, the rejection sampling filter is $1+(N+A+1)$ times slower than the quasi-optimal filter in computational time.

For the rejection sampling smoother, in addition to computation for filtering, the order of computation is $(N^2+(N+A+1))N(T-1)$, because for each i and t we need the order of N^2 in computing $\widehat{\gamma}^*_{ij,t}$, that of N in choosing i and that of $(A+1)$ in implementing rejection sampling. The Markov chain Monte Carlo smoother requires the order of $\sum_{t=1}^T tN = NT(T-1)/2$, where smoothing is not based on filtering in this case. However, for the quasi-optimal smoother, in addition to computation for filtering, the order of computation is $(T-1)N$. Therefore, except for computation of filtering, the rejection sampling smoother is $N^2 + (N + A + 1)$ times faster than the quasi-optimal smoother in computational time and $T/2$ times faster than the Markov chain Monte Carlo smoother. By the computation reduction method, since computation of $\widehat{\gamma}^*_{ij,t}$ reduces to the order of N for each i and each t, the computational burden of the rejection sampling smoother becomes the order of $(N + (N + A + 1))N(T - 1)$. In this case, the rejection sampling smoother is $N + (N + A + 1)$ times slower than the quasi-optimal smoother in computational time. Thus, the quasi-optimal filter and the quasi-optimal smoother have much less computational burden than the rejection sampling procedure and the Markov chain Monte Carlo approach.

A comparison of computation is summarized in Table 1. EK, RS, MCMC and QUASI denote the extended Kalman filter, the rejection sampling procedure, the Markov chain Monte Carlo procedure and the quasi-optimal procedure, respectively. RS* implies the RS with the computation reduction method shown in (12). Except for MCMC, since smoothing is based on filtering, the order of computation becomes sum of filtering and smoothing. That is, in Table 1, computation on filtering is equivalent to the first term of computation on smoothing for EK, RS, RS* and QUASI.

Data Storage: We compare the quasi-optimal procedure and the rejection sampling procedure from the data storage. Note that implementation of the quasi-optimal filter and quasi-optimal smoother is summarized as:

(i) Generate a random draw of α_t (i.e., denoted by $\alpha_{i,t|t}$) from $p_y(y_t|\alpha_t)$ $p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{i,t}$ for $t = 1, 2, \cdots, T$.

(ii) Generate a random draw of α_t (i.e., denoted by $\alpha_{i,t|T}$) from $p_y(y_t|\alpha_t)$ $p_\alpha(\alpha_{i,t+1|T}|\alpha_t)p_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma^*_{ii,t}$ for $t = T - 1, T - 2, \cdots, 1$.

(iii) Repeat (i) and (ii) for $i = 1, 2, \cdots, N$.

Thus, for the quasi-optimal procedure, filtering is represented together with smoothing as in Steps (i) – (iii). After computing filtering and smoothing for t, we repeat (i) and (ii) with respect to i. Therefore, the i-th filtering random draws $\alpha_{i,t|t}$ for $t = 1, 2, \cdots, T$ (i.e., T random draws) are used for the i-th smoothing random draws $\alpha_{i,t|T}$ for $t = T - 1, T - 2, \cdots, 1$. However, for the rejection sampling smoother, all the filtering random draws $\alpha_{i,t|t}$ for $i = 1, 2, \cdots, N$ and $t = 1, 2, \cdots, T$ (i.e., $N \times T$ random draws) should be stored for smoothing. That is, after all the filtering random draws are computed, the smoothing random draws should be generated using the backward recursive algorithm in Section 3.1. In order to obtain the i-th random draw of α_t from the smoothing density (i.e., $\alpha_{i,t|T}$), the quasi-optimal smoother uses the i-th random draw of α_{t-1} from the filtering density (i.e., $\alpha_{i,t-1|t-1}$) while the rejection sampling smoother utilizes all the filtering random draws ($\alpha_{1,t-1|t-1}$,

$\alpha_{2,t-1|t-1}$, $\cdots$, $\alpha_{N,t-1|t-1}$). Accordingly, it is clear that the quasi-optimal filter and quasi-optimal smoother have the advantage from the data storage, compared with the rejection sampling filter and smoother.

The Markov chain Monte Carlo procedure shown in Section 3.2 requires the exactly same amount of data storage as the quasi-optimal procedure shown in this section. In the estimation procedure (i) – (vii) of the Markov chain Monte Carlo approach, $\alpha_{i,s|t}$ for $i = 1, 2, \cdots$ and $s = 1, 2, \cdots, t$ are generated, which are utilized to compute the smoothing means $a_{s|t}$, $s = 1, 2, \cdots, t$. That is, the Markov chain Monte Carlo approach computes all the possible smoothing random draws in the algorithm (i) – (vii), although we need $\alpha_{i,t|s}$ for $s = t, T$. Thus, the Markov chain Monte Carlo procedure needs a lot of unnecessary computation.

5 MONTE CARLO STUDIES

In this section, by Monte Carlo studies, we compare numerical accuracy for all the estimators introduced in Sections 3 and 4, i.e., the rejection sampling procedure (Section 3.1), the Markov chain Monte Carlo procedure (Section 3.2) and the quasi-optimal filter and smoother (Section 4). The following state-space models are examined.

Simulation I (Linear and Normal Model): Consider the scalar system: $y_t = \alpha_t + \epsilon_t$ and $\alpha_t = \delta\alpha_{t-1} + \eta_t$. The initial value α_0 and the error terms ϵ_t and η_t, $t = 1, 2, \cdots, T$, are assumed to be distributed as: $\alpha_0 \sim N(0, 1)$ and $(\epsilon_t, \eta_t)' \sim N(0, I_2)$, where I_2 denotes a 2×2 identity matrix. The exactly same assumptions on the initial value and the error terms are taken in Simulations II and III.

Simulation II (ARCH Model): Consider the state-space model: $y_t = \alpha_t + \epsilon_t$ and $\alpha_t = (\delta_0 + \delta\alpha_{t-1}^2)^{1/2}\eta_t$ for $\delta_0 > 0$ and $0 \leq \delta < 1$, α_t follows an ARCH model and y_t consists of the ARCH process α_t and the error term ϵ_t. In this simulation study, $\delta_0 = 1 - \delta$ is taken. $\delta_0 = 1 - \delta$ in the transition equation implies that the unconditional variance of α_t is normalized to be one. See Engle (1982) and Bollerslev, Engle and Nelson (1994) for the ARCH model.

Simulation III (Stochastic Volatility Model): The scalar system is represented as: $y_t = \exp(0.5\alpha_t)\epsilon_t$ and $\alpha_t = \delta\alpha_{t-1} + \eta_t$ for $0 \leq \delta < 1$. See Ghysels, Harvey and Renault (1996) for the stochastic volatility model.

Simulation IV (Logistic Model): Next, consider the following non-linear scalar system: $y_t = \exp(\alpha_t)/\ (\exp(\alpha_t) + \exp(\epsilon_t))$ and $\alpha_t = \exp(\alpha_{t-1})/\ (\exp(\alpha_{t-1}) + \exp(\eta_t))$, where we assume that $\alpha_0 \sim U(0, 1)$ and $(\epsilon_t, \eta_t)' \sim N(0, I_2)$. Tanizaki (1993, 1996) took the logistic model as an example.

Simulation V (Nonstationary Growth Model): Take the univariate system: $y_t = \alpha_t^2/20 + \epsilon_t$ 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 $\alpha_0 \sim N(0,10)$, $\epsilon_t \sim N(0,1)$, and $\eta_t \sim N(0,10)$. ϵ_t and η_t are assumed to be mutually independent. The nonstationary growth model is examined in Kitagawa (1987, 1996) and Carlin, Polson and Stoffer (1992).

Simulation VI (Multivariate Non-Gaussian Cases): Finally, we examine 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: $F(x) = (\exp(-x)+1)^{-1}$), $\eta_{1t} \sim N(0,1)$ and $x_t \sim U(0,1)$. For the initial value $\alpha_0 = (\alpha_{10}, \alpha_{20})'$, we take the assumption that $\alpha_{10} \sim N(0,1)$. Moreover, ϵ_t, η_{1t}, η_{2t}, x_t, α_{10} and α_{20} are assumed to be mutually independent. Under the model, we take the following two cases:

Case I: $\eta_{2t} \sim t(3)$ and $\alpha_{20} \sim t(3)$.

Case II: $\eta_{2t} \sim N(0,1)$ and $\alpha_{20} \sim N(0,1)$. The data generating process is $\epsilon_t \sim$ Logistic as shown above, but the estimated model is $\epsilon_t \sim N(0,1)$.

We compare the extended Kalman filter and smoother and the nonlinear and non-Gaussian filters and smoothers introduced in Sections 3 and 4. Note that the extended Kalman filter and smoother reduce to the conventional Kalman filter and smoother in the case where the system is linear and normal (i.e., Simulation I). The simulation procedure is as follows:

(i) Generating random numbers of ϵ_t and η_t for $t = 1, 2, \cdots, T$, we obtain a set of data y_t and α_t, $t = 1, 2, \cdots, T$, from equations (1) and (2), where $T = 100$ is taken.

(ii) Given Y_T, perform each estimator.

(iii) Repeat (i) and (ii) G times and compare the root mean square error (RMSE) for each estimator.

RMSE is defined as: $\text{RMSE} = (1/T)\sum_{t=1}^{T} \text{MSE}_{t|s}^{1/2}$ for $s = t, T$, where the mean square error (MSE) is given by: $\text{MSE}_{t|s} \equiv (1/G)\sum_{g=1}^{G}(\widehat{\alpha}_{t|s}^{(g)} - \alpha_t^{(g)})^2$ and $\widehat{\alpha}_{t|s}$ takes the state variable estimated by each estimator while α_t denotes the artificially simulated state variable in (i). The superscript (g) denotes the g-th simulation run, where $G = 1000$ is taken.

Results and Discussion: The results are in Tables 2 – 5. In Tables 2 and 3, the parameter δ is assumed to be known in Simulations I – III. In

Table 5, estimation of δ is examined. The univariate cases are investigated in Tables 2, 3 and 5, while the multivariate cases are examined in Table 4. In each table, I, II, III, IV, V and VI denote Simulations I, II, III, IV, V and VI, respectively. Table 2 shows difference between four sorts of estimators, where EK, RS*, MCMC and QUASI denote the extended Kalman filter and smoother, the rejection sampling filter and smoother using the computation reduction method, the Markov chain Monte Carlo procedure with proposal density (A), and the quasi-optimal procedure shown in Section 4. For QUASI, (A), (B) and (C) in Tables 2 – 5 indicate three kinds of the proposal density discussed in the previous section. For MCMC, we do not examine proposal densities (B) and (C).

First, we compare each estimator in Table 2. As N is large, RMSE becomes small for all the cases. For Simulation I, since the system is linear and normal, it is easily expected that EK is the best estimator of all the other estimators. For EK of Simulations II and III, filtering is exactly equivalent to smoothing because the first derivative of $h_t(\alpha_t, \epsilon_t)$ with respect to α_t is evaluated as zero. MCMC is very close to EK in the sense of RMSE, compared with RS* and QUASI (A). For Simulations II – IV, MCMC gives us the smallest RMSE's, but for Simulation V it is the worst estimator of the simulation-based estimators. One of the reasons why MCMC is the worst estimator in Simulation V is due to slow convergence of the Gibbs sampler, depending on nonlinearity and nonnormality of the system. Although RS* and QUASI are inferior to MCMC for Simulations I – IV, difference between them is very small. Moreover, for Simulation V, the best estimator is RS* and the second best is QUASI (A), where both RS* and QUASI (A) give us much smaller RMSE's than MCMC. Especially as far as we focus on filtering, RS* is very close to QUASI (A). As the number of random draws increases, RS* surely goes to the true state values under the condition that the acceptance probability is greater than zero. Similarly, MCMC certainly approaches the true state values as the number of random draws is large. However, we should keep in mind that both RS* and MCMC have serious computational disadvantages. RS* cannot be implemented when the acceptance probability is very small. MCMC has the case where convergence is extremely slow. On the other hand, QUASI can be applied to any nonlinear and non-Gaussian cases, where the amount of computation does not depend on the distribution and the functional form. Furthermore, QUASI shows a quite good performance in the RMSE criterion for all the cases. Accordingly, QUASI might be recommended in practice, rather than RS* and MCMC.

Next, we examine QUASI (A) in Table 2 and QUASI (B) and QUASI (C) in Table 3, where QUASI is compared with various candidates of the proposal density, i.e., (A), (B) and (C). That is, we take (A) $p_*(z|x) = p_\alpha(z|\alpha_{i,t-1|s})$ in Table 2, and (B) $p_*(z|x) = N(a^*_{t|s}, c\Sigma_{t|s})$ and (C) $p_*(z|x) = N(x, c\Sigma_{t|s})$ in

Table 2: Univariate Cases: Simulations I – V

	N	I			II		III		IV	V
	δ	0.5	0.9	1.0	0.5	0.9	0.5	0.9		
(a) Filtering										
EK		.7307	.7747	.7878	.7061	.6439	1.161	2.266	.2138	22.34
	100	.7345	.7803	.7937	.6952	.5434	.941	1.123	.1978	4.97
RS*	200	.7326	.7778	.7910	.6927	.5395	.938	1.119	.1972	4.80
	500	.7316	.7758	.7890	.6913	.5386	.936	1.116	.1969	4.66
	1000	.7326	.7768	.7902	.6928	.5388	.937	1.118	.1970	10.31
MCMC	2000	.7313	.7758	.7890	.6912	.5366	.936	1.116	.1969	10.32
	5000	.7311	.7752	.7883	.6907	.5363	.936	1.115	.1968	10.04
QUASI	1000	.7330	.7796	.7942	.6935	.5412	.937	1.127	.1970	4.79
(A)	5000	.7313	.7770	.7910	.6911	.5376	.936	1.125	.1968	4.67
	10000	.7312	.7769	.7907	.6910	.5370	.936	1.124	.1968	4.64
(b) Smoothing										
EK		.7057	.6822	.6705	.7061	.6439	1.161	2.266	.2157	19.08
	100	.7095	.6874	.6755	.6847	.5225	.912	.942	.1976	3.95
RS*	200	.7080	.6850	.6729	.6828	.5192	.909	.938	.1971	3.77
	500	.7064	.6833	.6713	.6813	.5172	.908	.934	.1968	3.67
	1000	.7076	.6847	.6731	.6828	.5196	.908	.937	.1968	10.54
MCMC	2000	.7067	.6837	.6717	.6818	.5175	.908	.935	.1967	10.48
	5000	.7062	.6829	.6712	.6811	.5169	.907	.934	.1967	10.13
QUASI	1000	.7083	.6900	.6796	.6844	.5258	.912	.968	.1968	4.36
(A)	5000	.7070	.6873	.6767	.6822	.5213	.911	.966	.1967	4.06
	10000	.7069	.6872	.6763	.6819	.5207	.911	.965	.1967	4.01

Table 3, where $s = t-1$ for filtering and $s = T$ for smoothing. Thus, the three candidates are compared for QUASI. Note that Simulation IV is not computed in Table 3 because the state variable lies on the interval between zero and one (for (B) and (C), the state variable takes any value). In Table 3, the cases of $c = 1, 2, 4, 9, 16, 25$ are examined and $N = 10000$ is taken. The case $N = 10000$ of QUASI (A) in Table 2 should be compared with Table 3. As δ is large (i.e., as correlation between α_t and α_{t-1} is high), difference between proposal densities (A) and (B) (or (C)) increases in the sense of RMSE. Especially, for Simulation V and $\delta = 0.9$ of Simulations II and III, proposal densities (B) and (C) do not work well, compared with proposal density (A). Moreover, the RMSE's are not improved by c. Therefore, in these simulation studies, (A) might be recommended, rather than (B) and (C).

Table 3: Quasi-Optimal Estimator: Proposal Densities (B) and (C)
— $N = 10000$ —

		I			II		III		V
	$c \setminus \delta$	0.5	0.9	1.0	0.5	0.9	0.5	0.9	
(a) Filtering									
QUASI (B)	1	.7321	.7865	.8053	.7011	.6324	.9497	1.4094	18.23
	2	.7324	.7900	.8106	.6985	.6227	.9447	1.3705	17.10
	4	.7326	.7925	.8149	.6970	.6103	.9421	1.3512	15.92
	9	.7327	.7942	.8180	.6964	.5966	.9409	1.3470	14.81
	16	.7327	.7950	.8193	.6963	.5885	.9406	1.3511	14.28
	25	.7328	.7952	.8197	.6962	.5833	.9404	1.3563	13.96
QUASI (C)	1	.7330	.7958	.8207	.6969	.5872	.9397	1.2748	14.04
	2	.7328	.7956	.8206	.6964	.5815	.9397	1.2894	13.85
	4	.7328	.7955	.8206	.6961	.5769	.9398	1.3075	13.34
	9	.7328	.7955	.8207	.6961	.5734	.9399	1.3299	11.92
	16	.7328	.7956	.8208	.6961	.5721	.9400	1.3444	10.91
	25	.7329	.7958	.8209	.6962	.5715	.9402	1.3556	10.20
(b) Smoothing									
QUASI (B)	1	.7071	.6950	.6896	.6951	.6282	.9315	1.3395	15.82
	2	.7074	.6985	.6946	.6915	.6169	.9226	1.2850	15.29
	4	.7076	.7013	.6989	.6892	.6029	.9175	1.2536	14.90
	9	.7078	.7032	.7021	.6880	.5868	.9150	1.2416	14.61
	16	.7079	.7040	.7034	.6877	.5770	.9142	1.2429	14.42
	25	.7080	.7044	.7040	.6875	.5709	.9139	1.2470	14.22
QUASI (C)	1	.7081	.7049	.7048	.6887	.5788	.9120	1.1498	14.04
	2	.7079	.7047	.7047	.6881	.5724	.9121	1.1669	13.85
	4	.7079	.7048	.7048	.6877	.5667	.9124	1.1878	13.39
	9	.7079	.7050	.7049	.6873	.5617	.9126	1.2137	12.15
	16	.7080	.7047	.7050	.6874	.5594	.9130	1.2306	11.17
	25	.7081	.7049	.7051	.6873	.5584	.9133	1.2433	10.38

Table 4: Multivariate Cases: Simulation VI

		Case I		Case II	
		α_{1t}	α_{2t}	α_{1t}	α_{2t}
(a) Filtering					
EK		2.840	1.983	2.670	1.780
RS*	100	3.031	2.025	2.861	1.818
	200	2.913	1.976	2.819	1.798
	500	2.828	1.934	2.778	1.780
MCMC	1000	3.192	2.114	2.922	1.849
	2000	3.057	2.049	2.873	1.825
	5000	2.917	1.981	2.814	1.799
QUASI (A)	1000	3.205	2.132	2.804	1.793
	5000	3.112	2.065	2.757	1.770
	10000	3.117	2.065	2.767	1.772
(b) Smoothing					
EK		2.083	1.549	1.950	1.347
RS*	100	2.353	1.647	2.272	1.478
	200	2.206	1.583	2.176	1.426
	500	2.095	1.521	2.093	1.386
MCMC	1000	2.352	1.599	2.168	1.415
	2000	2.245	1.550	2.115	1.392
	5000	2.147	1.512	2.056	1.363
QUASI (A)	1000	2.726	1.857	2.320	1.482
	5000	2.623	1.798	2.284	1.465
	10000	2.636	1.802	2.299	1.468

Table 5: Estimation of Unknown Parameter δ

	N		I			II		III	
	δ		0.5	0.9	1.0	0.5	0.9	0.5	0.9
EK		AVE	0.47	0.88	0.98	0.30	0.65	—	—
		RMSE	0.14	0.06	0.04	0.28	0.31	—	—
		25%	0.40	0.85	0.97	0.14	0.53	—	—
		50%	0.49	0.89	0.99	0.31	0.66	—	—
		75%	0.57	0.92	1.00	0.45	0.79	—	—
QUASI (A)	1000	AVE	0.48	0.90	1.00	0.49	0.91	0.50	0.90
		RMSE	0.11	0.02	0.01	0.14	0.06	0.07	0.02
		25%	0.47	0.89	0.99	0.48	0.89	0.49	0.90
		50%	0.50	0.90	1.00	0.50	0.91	0.50	0.90
		75%	0.53	0.91	1.01	0.53	0.93	0.52	0.91
	5000	AVE	0.48	0.90	0.99	0.47	0.90	0.45	0.89
		RMSE	0.13	0.04	0.03	0.21	0.09	0.20	0.05
		25%	0.43	0.89	0.98	0.45	0.88	0.37	0.88
		50%	0.50	0.90	1.00	0.51	0.91	0.50	0.90
		75%	0.55	0.91	1.00	0.55	0.95	0.58	0.92

In Table 4, two multivariate cases are examined. In Case I, $\eta_{2t} \sim t(3)$ and $\alpha_{20} \sim t(3)$ are assumed. In Case II, $\eta_{2t} \sim N(0,1)$ and $\alpha_{20} \sim N(0,1)$ are taken, and moreover the data generating process is $\epsilon_t \sim$ Logistic but the estimated model uses $\epsilon_t \sim N(0,1)$. We obtain the almost same results as Simulations I – V in Table 2. We see from Table 4 that RS* performs the best estimator.

Finally, in Table 5, the estimation results of unknown parameter δ are shown. The estimates of δ are compared for EK and QUASI (A). AVE and RMSE represent the arithmetic average and the root mean square error from the G estimates of δ. 25%, 50% and 75% give us the 0.25th, 0.50th and 0.75th percentiles from the G estimates of δ, where $G = 1000$ in this paper. The maximization of the likelihood function is performed by a simple grid search, in which the likelihood function is maximized by changing the parameter value of δ by 0.01. From Table 5, we obtain the result that QUASI (A) is superior to EK in the sense of AVE, RMSE and 50%, because for all the cases (Simulations I and II, $\delta = 0.5, 0.9, 1.0$), both AVE and 50% of QUASI are closer to each true parameter value than those of EK and QUASI has smaller RMSE than EK. Moreover, we have the case where EK does not work for maximization of the likelihood function (see Simulation III). For the extended Kalman filter in Simulation III, note that the likelihood function does not depend on δ when evaluation of the first derivative of $h_t(\alpha_t, \epsilon_t)$ with respect to α_t is zero and the initial value is given by $a^*_{0|0} = 0$.

6 SUMMARY

In this paper, we have proposed QUASI, which features are as follows: QUASI does not give us the exact filtering and smoothing solutions, but QUASI yields much less computer-intensive and relatively efficient estimators in the root mean square error criterion. Rejection sampling has the disadvantages which it takes an extremely lot of time when the acceptance probability is close to zero and it cannot be applied when the acceptance probability is equal to zero. The disadvantages of the Gibbs sampler are: the random draws generated by the Gibbs sampler converge slowly to the random draws generated from the true target density when the random variables are highly correlated with each other, i.e., a numerous number of random draws have to be generated in this case. Even though both RS and MCMC give us the true estimates of the state variables as the number of random draws is large enough, it is not realistic in practice to generate a great number of random draws from computational point of view and accordingly precision of the two estimators is not so reliable under the finite number of random draws. Therefore, we have proposed the quasi-optimal procedure as the second best estimator, which can be applied to any nonlinear and non-Gaussian type of the state space models while the rejection sampling procedure requires the condition that the acceptance probability is greater than zero.

Furthermore, through the Monte Carlo studies, the followings are examined: (i) QUASI is compared with EK, RS* and MCMC, (ii) the three candidates of the proposal density are examined for QUASI, and (iii) in the case where the unknown parameter is included in the system, estimation of the parameter is performed by the maximum likelihood method. We have obtained the results as follows. QUASI shows a good performance, which is very close to RS*. MCMC performs very bad for Simulation V because of slow convergence of the Gibbs sampler depending on nonlinearity and nonnormality of the system. For QUASI, in the case where we compare the three candidates of the proposal density, proposal density (A) is superior to proposal densities (B) and (C).

For the simulation-based estimator, precision of the state estimates and the computation cost are trade-off in general. As the number of random draws increases, precision of the estimates increases. However, at the same time, it is not realistic to generate infinitely many random draws. Therefore, the quasi-optimal estimator proposed in this paper might be useful, which yields much less computational burden and relatively efficient estimator. Moreover, we should point out that the quasi-optimal approach requires much less data storage than RS*. Thus, QUASI might be recommended in practice, rather than RS* and MCMC, from computation burden and data storage.

APPENDICES

APPENDIX 1: Rejection Sampling

Let $p(x)$ and $p_*(x)$ be the target density and the proposal density. When we want to generate a random draw from $p(x)$, we utilize another distribution function $p_*(x)$, which is appropriately chosen by a researcher. Define the acceptance probability as $\omega(x) = p(x)/ap_*(x)$, where the assumption $a \equiv \sup_x p(x)/p_*(x) < \infty$ is required. Rejection sampling is implemented as: (i) generate a random draw from $p_*(x)$, (ii) accept it with probability $\omega(x)$. The accepted random draw is taken as a random draw from $p(x)$. Note that $p(x)$ is not necessarily a probability density function, i.e., it is possibly a kernel of the target density function, because of the form of the acceptance probability $\omega(x)$.

In the case where both $p(x)$ and $p_*(x)$ are normally distributed as $N(\mu, \sigma^2)$ and $N(\mu_*, \sigma_*^2)$, it is easily shown that we need $\sigma_*^2 > \sigma^2$ for the condition $a < \infty$, which implies that $p_*(x)$ has to be distributed with larger variance than $p(x)$. Using rejection sampling, we can generate a random draw from any distribution function under the condition that $a < \infty$ is satisfied. However, the disadvantages of rejection sampling are: (i) we need to compute a, which sometimes does not exist and (ii) it takes a long time when $\omega(\cdot)$ is close to zero. See, for example, Boswell, Gore, Patil and Taillie (1993), O'Hagan (1994) and Geweke (1996) for rejection sampling.

APPENDIX 2: Gibbs Sampler

When the two conditional densities $p(x|y)$ and $p(y|x)$ are available, we consider generating random draws of (x, y) from the joint density $p(x, y)$. The Gibbs sampler is performed as follows: (i) pick up an appropriate value for the initial value of y (i.e., y_0), (ii) generate a random draw of x (i.e., x_i) from $p(x|y_{i-1})$, (iii) generate a random draw of y (i.e., y_i) from $p(y|x_i)$, and (iv) repeat (ii) and (iii) for $i = 1, 2, \cdots, N$.

From the basic result of the Gibbs sampler, (x_N, y_N) converges to a random draw from $p(x, y)$ for sufficiently large N. We take an example of the bivariate case here, but it is easily extended to the multivariate cases. In addition, it is also possible that x and y are vectors. See Geman and Geman (1984), Tanner and Wong (1987), Gelfand and Smith (1990), Arnold (1993) and Geweke (1996, 1997) for the Gibbs sampler.

APPENDIX 3: Metropolis-Hastings Algorithm

Consider generating a random draw of z from $p(z)$. The Metropolis-Hastings algorithm utilizes the proposal density $p_*(z|x)$, which has to be appropriately specified by a researcher. Define the acceptance probability $\omega(x,z)$ as:

$$\omega(x,z) = \begin{cases} \min\left(\dfrac{p(z)p_*(x|z)}{p(x)p_*(z|x)}, 1\right), & \text{if } p(x)p_*(z|x) > 0, \\ 1, & \text{otherwise.} \end{cases}$$

The Metropolis-Hastings algorithm is the random number generation method such that we can generate random draws from any density function, which can be implemented as follows: (i) take an initial value of x as x_0, (ii) given x_{i-1}, generate a random draw z from $p_*(\cdot|x_{i-1})$ and a uniform random draw u from the interval between zero and one, (iii) set $x_i = z$ if $u \le \omega(x_{i-1}, z)$ and $x_i = x_{i-1}$ otherwise, and (iv) repeat (ii) and (iii) for $i = 1, 2, \cdots, N$.

We can take x_N as a random draw from $p(x)$ for sufficiently large N. For choice of the proposal density $p_*(z|x)$, the proposal density should not have too large variance and too small variance, compared with the target density (see, for example, Chib and Greenberg (1995)). That is, the proposal density should be chosen so that the chain travels over the support of the target density. This may fail to occur, with a consequent undersampling of low probability regions, if the chain is near the mode and if candidates are drawn too close to the current value (see Chib and Greenberg (1996)). For a functional form of the proposal density $p_*(z|x)$, we may take $p_*(z|x) = p_*(z-x)$, called the random walk chain, or $p_*(z|x) = p_*(z)$, called the independence chain. Note that $p(z)$ is not necessarily a probability density function, i.e., it is possibly a kernel of the target density function, because of the form of the acceptance probability $\omega(x,z)$. Remember that we need the ratio of the target and proposal densities to derive $\omega(x,z)$. Smith and Roberts (1993), Tierney (1994), Chib and Greenberg (1995, 1996) and Geweke (1996) discussed the Metropolis-Hastings algorithm.

References

Anderson, B.D.O. and Moore, J.B., 1979, *Optimal Filtering*, Prentice-Hall, New York.

Arnold, S.F., 1993, “ Gibbs Sampling, ” in *Handbook of Statistics*, Vol.9, edited by Rao, C.R., pp.599 – 625, North-Holland.

Berzuini, C., Best, N.G., Gilks, W.R. and Larizza, C., 1997, “ Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods, ” *Journal of the American Statistical Association*, Vol.92, pp.1403 – 1412.

Bollerslev, T., Engle, R.F. and Nelson, D.B., 1994, " ARCH Models, " in *Handbook of Econometrics*, Vol.4, edited by Engle, R.F. and McFadden, D.L., pp.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, " in *Handbook of Statistics*, Vol.9, edited by Rao, C.R., pp.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, " *Journal of the American Statistical Association*, Vol.87, No.418, pp.493 – 500.

Carter, C.K. and Kohn, R., 1994, " On Gibbs Sampling for State Space Models, " *Biometrika*, Vol.81, No.3, pp.541 – 553.

Carter, C.K. and Kohn, R., 1996, " Markov Chain Monte Carlo in Conditionally Gaussian State Space Models, " *Biometrika*, Vol.83, No.3, pp.589 – 601.

Chib, S. and Greenberg, E., 1995, " Understanding the Metropolis-Hastings Algorithm, " *The American Statistician*, Vol.49, No.4, pp.327 – 335.

Chib, S. and Greenberg, E., 1996, " Markov Chain Monte Carlo Simulation Methods in Econometrics, " *Econometric Theory*, Vol.12, No.4, pp.409 – 431.

Durbin, J. and Koopman, S.J., 1997, " Monte Carlo Maximum Likelihood Estimation for Non-Gaussian State Space Models, " *Biometrika*, Vol.84, No.3, pp.669 – 684.

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

Gelb, A. (Ed.), 1974, *Applied Optimal Estimation*, MIT Press.

Gelfand, A.E. and Smith, A.F.M., 1990, " Sampling-Based Approaches to Calculating Marginal Densities, " *Journal of the American Statistical Association*, Vol.85, No.410, pp.398 – 409.

Geman, S. and Geman D., 1984, " Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, " *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.Pami-6, No.6, pp.721 – 741.

Geweke, J., 1996, " Monte Carlo Simulation and Numerical Integration, " in *Handbook of Computational Economics*, Vol.1, edited by Amman, H.M., Kendrick, D.A. and Rust, J., pp.731 – 800, North-Holland.

Geweke, J., 1997, " Posterior Simulators in Econometrics, " in *Advances in Economics and Econometrics: Theory and Applications*, Vol.3, edited by Kreps, D. and Wallis, K.F., pp.128 – 165, Cambridge University Press.

Geweke, J. and Tanizaki, H., 1999a, “ On Markov Chain Monte-Carlo Methods for Nonlinear and Non-Gaussian State-Space Models, ” *Communications in Statistics, Simulation and Computation*, Vol.28, No.4, pp.867 - 894.

Geweke, J. and Tanizaki, H., 1999b, “ Bayesian Estimation of State-Space Model Using the Metropolis-Hastings Algorithm within Gibbs Sampling, ” Unpublished Manuscript (http://ht. econ. kobe-u.ac. jp/ ~tanizaki/ cv/ papers/ bayes.pdf).

Ghysels, E., Harvey, A.C. and Renault, E., 1996, “ Stochastic Volatility, ” in *Handbook of Statistics*, Vol.14, edited by Maddala, G.S. and Rao, C.R., pp.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*, Vol.140, No.2, pp.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, ” *Journal of Computational and Graphical Statistics*, Vol.7, pp.175 – 193.

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

Kitagawa, G., 1996, “ Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State-Space Models, ” *Journal of Computational and Graphical Statistics*, Vol.5, No.1, pp.1 – 25.

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

Kitagawa, G., 1998, “ A Self-Organizing State-Space Model, ” *Journal of the American Statistical Association*, Vol.93, pp.1203 – 1215.

Liu, J.S. and Chen, R., 1995, “ Blind Deconvolution via Sequential Imputations, ” *Journal of the American Statistical Association*, Vol.90, pp.567 – 576.

Liu, J.S. and Chen, R., 1998, “ Rejection Control and Sequential Importance Sampling, ” *Journal of the American Statistical Association*, Vol.93, pp.1022 – 1031.

Liu, J.S. and Chen, R., 1998, “ Sequential Monte Carlo Methods for Dynamic Systems, ” *Journal of the American Statistical Association*, Vol.93, pp.1032 – 1044.

MacEachern, S.N., Clyde, M. and Liu, J.S., 1999, “ Sequential Importance Sampling for Nonparametric Bayes Models: The Next Generation, ” *The Canadian Journal of Statistics*, Vol.27, No.2, pp.251 – 267.

Mariano, R.S. and Tanizaki, H., 1998, “ Simulation-Based Inference in Nonlinear State-Space Models: Application to Testing Permanent Income Hypothesis, ” forthcoming in *Simulation-Based Inference in Econometrics: Methods and Applications*, edited by Mariano, R.S., Weeks, M. and Schuermann, T., Cambridge University Press.

O’Hagan, A., 1994, *Kendall’s Advanced Theory of Statistics*, Vol.2B (Bayesian Inference), Edward Arnold.

Smith, A.F.M. and Roberts, G.O., 1993, “ Bayesian Computation via Gibbs Sampler and Related Markov Chain Monte Carlo Methods, ” *Journal of the Royal Statistical Society*, Ser.B, Vol.55, No.1, pp.3 – 23.

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

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 Transactions on Automatic Control*, Vol.44, No.2, pp.315 – 319.

Tanizaki, H. and Mariano, R.S., 1996, “ Nonlinear Filters Based on Taylor Series Expansions, ” *Communications in Statistics, Theory and Methods*, Vol.25, No.6, pp.1261 – 1282.

Tanizaki, H. and Mariano, R.S., 1998, “ Nonlinear and Non-Gaussian State-Space Modeling with Monte Carlo Simulations, ” *Journal of Econometrics*, Vol.83, No.1,2, pp.263 – 290.

Tanner, M.A. and Wong, W.H., 1987, “ The Calculation of Posterior Distributions by Data Augmentation, ” *Journal of the American Statistical Association*, Vol.82, No.398, pp.528 – 550 (with discussion).

Tierney, L., 1994, “ Markov Chains for Exploring Posterior Distributions, ” *The Annals of Statistics*, Vol.22, No.4, pp.1701 – 1762.

Wishner, R.P., Tabaczynski, J.A. and Athans, M., 1969, “ A Comparison of Three Non-Linear Filters, ” *Automatica*, Vol.5, pp.487 – 496.