Nonlinear and Non-Gaussian State-Space Modeling with Monte Carlo Techniques: A Survey and Comparative Study

Hisashi Tanizaki*
Faculty of Economics, Kobe University

January, 2000

Contents

*The author would like to acknowledge an anonymous referee for valuable comments. Correspondence: Hisashi Tanizaki, Faculty of Economics, Kobe University, Rokkodai, Nadaku, Kobe 657-8501, Japan. E-mail: tanizaki@kobe-u.ac.jp

Abstract: Since Kitagawa (1987) and Kramer and Sorenson (1988) proposed the filter and smoother using numerical integration, nonlinear and/or non-Gaussian state estimation problems have been developed. Numerical integration becomes extremely computer-intensive in the higher dimensional cases of the state vector. Therefore, to improve the above problem, the sampling techniques such as Monte Carlo integration with importance sampling, resampling, rejection sampling, Markov chain Monte Carlo and so on are utilized, which can be easily applied to multi-dimensional cases. Thus, in the last decade, several kinds of nonlinear and non-Gaussian filters and smoothers have been proposed using various computational techniques. The objective of this paper is to introduce the nonlinear and non-Gaussian filters and smoothers which can be applied to any nonlinear and/or non-Gaussian cases. Moreover, by Monte Carlo studies, each procedure is compared by the root mean square error criterion.

Key Words: Nonlinear, Non-Gaussian, State-Space Model, Prediction, Filtering, Smoothing, Numerical Integration, Monte Carlo Integration with Importance Sampling, Resampling, Rejection Sampling, Gibbs Sampling, Metropolis-Hastings Algorithm.

1 Introduction

The Kalman filter proposed by Kalman (1960) and Kalman and Bucy (1961) has been extended to nonlinear and nonnormal cases. The most heuristic and easiest nonlinear filters utilize the Taylor series expansions and the expanded nonlinear measurement and transition functions are directly applied to the standard linear recursive Kalman filter algorithm (see Appendix A for the standard linear recursive prediction, filtering and smoothing algorithms). One of the traditional nonlinear filters is known as the extended Kalman filter, which is discussed in Wishner, Tabaczynski and Athans (1969), Jazwinski (1970), Gelb (1974), Anderson and Moore (1979), Tanizaki (1993a, 1996) and Tanizaki and Mariano (1996). Since we have the inequality $\mathrm{E}(g(x)) \neq g(\mathrm{E}(x))$ for a nonlinear function $g(\cdot)$ and a random variable x (see Brown and Mariano (1984, 1989) and Mariano and Brown (1983, 1989)), the state vector estimated by the extended Kalman filter is clearly biased.

In order to avoid the biased estimates, we need to consider the filtering and smoothing algorithms based on the underlying distribution functions. The standard Kalman filter algorithm is represented by the first- and the second-moments (i.e., mean and variance) of the underlying density functions, provided that the measurement and transition equations are linear and normality is assumed for the error terms (see, for example, Harvey (1989) and Tanizaki (1996)). Unless the distributions of the error terms in the state-space model are normal and/or the measurement and transition equations are linear, we cannot derive an explicit linear recursive expression for the filtering and smoothing algorithms. If we approximate the nonlinear measurement and transition equations by the Taylor series expansions and apply the linearized nonlinear equations directly to the linear recursive filtering and

smoothing algorithm, the obtained filtering and smoothing estimates are not appropriate and plausible.

It is known that when the system is linear and normal the Kalman filter estimate is optimal in the sense that it minimizes the mean square error. When the normality assumption is dropped, there is no longer any guarantee that the Kalman filter gives the conditional mean of the state vector. However, it is still an optimal estimator in the sense that it minimizes the mean square error within the class of all linear estimators (see Harvey (1989)). In addition, as Meinhold and Singpurwalla (1989) pointed out, Kalman filter models based on normality assumption are known to be non-robust, which implies that when there is a large difference between the prior density and the observed data, the posterior density becomes unrealistic. Therefore, approximation of the underlying densities, rather than that of the nonlinear functions, is essential to the nonlinear and/or nonnormal filtering problem.

The recursive algorithms on prediction, filtering and smoothing can be obtained from the conditional density functions, which are derived from Bayes' formula. The nonlinear filters and smoothers based on the underlying density functions have been developed to obtain an optimal estimator. Alspach and Sorenson (1972) and Sorenson and Alspach (1971) approximated the densities by a sum of Gaussian distributions, called the Gaussian sum filter. The obtained algorithm is a weighted sum of the extended Kalman filters. Therefore, the sum of the biased estimators also leads to the biased estimator.

To improve the biased filtering and smoothing estimates, Kitagawa (1987) and Kramer and Sorenson (1988) proposed approximating the densities numerically by a piecewise linear function, where each density is represented as number of segments, location of nodes and the value at each node, and it is evaluated through numerical integration. According to the numerical integration approach, however, computational burden increases more than proportionally as the dimension of the state vector increases. Programming is also extremely tedious in multi-dimensional cases.

In order to resolve the problems of the numerical integration procedure, Tanizaki (1993, 1996), Tanizaki and Mariano (1994) and Mariano and Tanizaki (1995) suggested using Monte-Carlo integration with importance sampling to density evaluation, in which a recursive algorithm of the weight functions represented by the ratio of two densities is derived. Geweke (1988, 1989a, 1989b) and Shao (1989) developed an approximation of prior density in Bayesian framework, so-called importance sampling (see Appendix B for several sampling methods). From the point of programming and computational time, the nonlinear and nonnormal filter and smoother based on Monte Carlo integration with importance sampling can be easily extended to the higher-dimensional cases in practice, comparing with the numerical integration procedure, although the importance sampling procedure is inferior to the numerical integration approach for precision of the filtering and smoothing estimates. However, one of the problems in the importance sampling procedure is as follows. For approximation of the target density functions, a researcher has to choose another appropriate density function, called the importance density, which is quite *ad hoc*. Unless the importance density is plausible, the obtained filtering and smoothing estimates might be biased.

Because both the numerical integration procedure and the importance sampling approach are based on density approximation for each time, where the density approximation at

present time depends on that at the past time, accumulation of computational errors possibly become large as time goes by. Therefore, recently, some attempts are made to generate random draws directly from prediction, filtering and smoothing distribution functions. Gordon, Salmond and Smith (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) utilized the resampling method to generate random draws from prediction, filtering and smoothing densities. For generation of filtering and smoothing random draws, one-step ahead prediction random draws are chosen with the corresponding probabilities. Programming is very easy compared with the above two approaches.

Tanizaki (1996, 1999a), Tanizaki and Mariano (1998), Hürzeler and Künsch (1998), Mariano and Tanizaki (2000) proposed an nonlinear and nonnormal filter and smoother using rejection sampling. For a solution to nonlinear and nonnormal state-space model, we use the random draws to obtain filtering and smoothing estimates. By rejection sampling, a recursive algorithm of the random draws are obtained. Thus, the random draws of the state vector are directly generated from the filtering and smoothing densities. The rejection sampling procedure gives us more precise estimates than the resampling procedure. However, when the acceptance probability in rejection sampling is small, computational time increases.

Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996) and Chib and Greenberg (1996) suggested applying an adaptive Monte-Carlo integration technique known as the Gibbs sampler to the density approximation. Geweke and Tanizaki (1999a, 1999b) extended the approach to more general formulation. That is, the filter and smoother proposed by Geweke and Tanizaki (1999a, 1999b) can be applied to any nonlinear and non-Gaussian cases, using the Metropolis-Hastings algorithm within Gibbs sampling for random number generation. However, when we apply the Gibbs sampler to the filtering and smoothing framework, convergence of random draws is sometimes very slow, depending on the underlying state-space model, which implies that extremely large number of random draws are required for precision of the filtering and smoothing estimates.

Furthermore, improving computation time on rejection sampling and convergence on the Gibbs sampler, we introduce quasi-filter and quasi-smoother using the Metropolis-Hastings algorithm (see Tanizaki (1998)). As mentioned above, the rejection sampling procedure does not work when the acceptance probability is small and the Markov chain Monte Carlo procedure needs numerous numbers of random draws because of slow convergence of the Gibbs sampler. The quasi-filter and quasi-smoother improve both the problems. However, because the quasi-filter and quasi-smoother utilize some approximations, the obtained estimates are not exactly equal to the true state variables although the estimates are very close to the true values.

Thus, in the past, several kinds of nonlinear and non-Gaussian filters and smoothers have been proposed from various aspects. The objective of this paper is to introduce the nonlinear and non-Gaussian filters and smoothers which can be applied to any nonlinear and/or non-Gaussian cases. The outline of this paper is as follows. In the first half of Section 2, some economic applications of the state-space model are discussed and in the last half, two sorts of density-based prediction, filtering and smoothing algorithms are described, i.e., recursive and non-recursive algorithms. Section 3 summarizes recent work on the nonlinear and non-Gaussian filters and smoothers. In Section 4, all the procedures introduced in Section 3

are compared by Monte Carlo studies. Finally, Section 5 makes summary and concluding remarks.

2 State-Space Model

We consider the nonlinear and non-Gaussian state-space model, which is represented in the following general form:

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

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

for $t = 1, 2, \cdots, T$, where T denotes the sample size. A vector y_t is observable while a vector α_t, called the state variable, is unobserved. The error terms ϵ_t and η_t are mutually independently distributed, which are typically assumed to be normal but not necessarily. $h_t(\cdot, \cdot)$ and $f_t(\cdot, \cdot)$ are vector functions, which are assumed to be known and may depend on other exogenous variables (however, we omit them for simplicity). Equation (1) is called the measurement equation, which represents the relationship between the observed data y_t and the unobservable state variable α_t. Equation (2) indicates the movement of the state variable, which is called the transition equation. Let Y_s be the information set up to time s, i.e., $Y_s = \{y_1, y_2, \cdots, y_s\}$.

The purpose of the state-space model is to estimate the unobserved state variable α_t utilizing the measurement and transition equations. That is, we consider estimating the conditional expectation of α_t using information Y_s, i.e., $a_{t|s} \equiv \mathrm{E}(\alpha_t|Y_s)$. Depending on t and s, the conditional expectation $a_{t|s}$ is called prediction if $t > s$, filtering if $t = s$ and smoothing if $t < s$. Moreover, there are three kinds of smoothing by the relationship between t and s with $t < s$. Let L be the fixed nonnegative integer. Then, $a_{L|t}$ for and $t = L+1, L+2, \cdots, T$ is called fixed-point smoothing, which is useful to estimate the initial condition of the system. $a_{t|t+L}$ for $t = 1, 2, \cdots, T-L$ is known as fixed-lag smoothing. $a_{t|T}$ for $t = 1, 2, \cdots, T$ is called fixed-interval smoothing, which is helpful to investigate the past condition of the system.

In this paper, we focus on L-step ahead prediction (i.e., $a_{t+L|t}$), filtering (i.e., $a_{t|t}$) and fixed-interval smoothing (i.e., $a_{t|T}$) in nonlinear non-Gaussian cases.

2.1 Some Applications of State-Space Model in Economics

Some economic applications of the state-space model are discussed in this section, where we consider the following examples: Time Varying Parameter Model (Section 2.1.1), Autoregressive-Moving Average Process (Section 2.1.2), Seasonal Adjustment Models (Section 2.1.3), Prediction of Final Data Using Preliminary Data (Section 2.1.4), Estimation of Permanent Consumption (Section 2.1.5), Markov Switching Model (Section 2.1.6) and Stochastic Variance Models (Section 2.1.7).

2.1.1 Time Varying Parameter Model

In the case where we deal with time series data, the nonlinear regression model can be written as follows:

$$y_t = h_t(x_t, \alpha, \epsilon_t),$$

for $t = 1, 2, \cdots, T$, where y_t is a dependent variable, x_t denotes a $1 \times k$ vector of the explanatory variables, a $k \times 1$ vector of unknown parameters to be estimated is given by α, and ϵ_t is the error term. $h_t(\cdot, \cdot, \cdot)$ is assumed to be a known vector function, which is given by $h_t(x_t, \alpha, \epsilon_t) = x_t\alpha + \epsilon_t$ in a classical linear regression model. There are some methods to estimate the equation above, for example, the least squares method, the method of moments and so on. In any case, if the unknown parameters to be estimated are constant over time, the model is known as the fixed-parameter model. However, structural changes (for example, the first- and second oil crises), specification errors, proxy variables and aggregation are all the sources of parameter variation; see Sarris (1973), Belsley (1973), Belsley and Kuh (1973) and Cooley and Prescott (1976). Therefore, we need to consider the model such that the parameter is a function of time, which is called the time varying parameter model. Using the state-space form, the model is represented as the following two equations:

$$\text{(Measurement equation)} \quad y_t = h_t(x_t, \alpha_t, \epsilon_t), \tag{3}$$

$$\text{(Transition equation)} \quad \alpha_t = \Psi\alpha_{t-1} + \eta_t, \tag{4}$$

where the movement of the parameter α_t is assumed to be the first-order autoregressive (AR(1)) process, which can be extended to the AR(p) process. The error term η_t, independent of ϵ_t, is a white noise. Here, equations (3) and (4) are referred to as the measurement equation and the transition equation, respectively. The time varying parameter α_t is unobservable, which is estimated using the observed data y_t and x_t. There are numerous other papers which deal with the time varying parameter model, for example, Cooley (1977), Cooley, Rosenberg and Wall (1977), Cooper (1973), Nicholls and Pagan (1985), Tanizaki (1989, 1993b) and Sant (1977).

2.1.2 Autoregressive Moving Average Process

It is well known that any autoregressive-moving average (ARMA) process can be written in a state-space form. See, for example, Aoki (1987, 1990), Brockwell and Davis (1987), Burridge and Wallis (1988), Gardner, Harvey and Phillips (1980), Hannan and Deistler (1988), Harvey (1981, 1989) and Kirchen (1988).

First, consider the following ARMA(p, q) process.

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ \cdots \ + a_p y_{t-p} + \eta_t + b_1\eta_{t-1} + \ \cdots \ + b_q\eta_{t-q},$$

where η_t is a white noise. The ARMA(p, q) model above is rewritten as:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ \cdots \ + a_m y_{t-m} + \eta_t + b_1\eta_{t-1} + \ \cdots \ + b_{m-1}\eta_{t-m+1},$$

where $m = \max(p, q+1)$ and some of the coefficients $a_1, a_2, \cdots, a_m$, $b_1, b_2, \cdots, b_{m-1}$ can be zeros. As it is well known, the ARMA process above is represented as:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t = z\alpha_t, \\ &\text{(Transition equation)} \quad\quad \alpha_t = A\alpha_{t-1} + B\eta_t, \end{aligned}$$

where z, A and B are defined as:

$$z = (1, 0, \cdots, 0), \quad A = \left(\begin{array}{c|c} a_1 & \\ \vdots & I_{m-1} \\ a_{m-1} & \\ \hline a_m & 0 \end{array}\right), \quad B = \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_{m-1} \end{pmatrix}.$$

$$1 \times m \qquad m \times m \qquad m \times 1$$

Thus, the state-space model is constructed from the ARMA model, where the first element of α_t represents the time series data to be estimated.

2.1.3 Seasonal Adjustment Models

A time series consists of seasonal, cyclical, and irregular components. Each component is unobservable and therefore the Kalman filter is applied to estimate each component separately. In this section, two seasonal adjustment models are introduced; one is developed by Pagan (1975) and another is Kitagawa (1996) and Kitagawa and Gersch (1996).

The suggestion by Pagan (1975) is essentially a combination of an econometric model for the cyclical components with the filtering and estimation of the seasonal components formulated in a state-space form (see Chow (1983)). Assume, first, that an endogenous variable y_t is the sum of cyclical, seasonal, and irregular components, as given by:

$$y_t = y_t^c + y_t^s + \epsilon_t, \tag{5}$$

where y_t^c, y_t^s and ϵ_t denote the cyclical, seasonal, and irregular components, respectively. Second, that the cyclical component y_t^c is represented as the following model:

$$y_t^c = Ay_{t-1}^c + Cx_t + u_t, \tag{6}$$

where x_t is a $k \times 1$ vector of exogenous variables and u_t denotes a random disturbance. In equation (6), the AR(1) model is assumed for simplicity but the AR(p) model is also possible. Finally, an autoregressive seasonal model is assumed for the seasonal component, i.e.,

$$y_t^s = By_{t-m}^s + w_t, \tag{7}$$

where w_t represents a random residual and m can be 4 for a quarterly model and 12 for a monthly model. Combining the equations (5) – (7), we can construct the following state-space form:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t = z\alpha_t + \epsilon_t, \\ &\text{(Transition equation)} \quad\quad \alpha_t = M\alpha_{t-1} + Nx_t + \eta_t, \end{aligned} \tag{8}$$

where z, α_t, M, N and η_t are given by:

$$z = (1, 1, 0, \cdots, 0), \quad M = \left(\begin{array}{c|cc} A & & 0 \\ \hline & 0 & I_{m-1} \\ 0 & B & 0 \end{array}\right), \quad N = \begin{pmatrix} C \\ 0 \end{pmatrix}, \qquad \eta_t = \begin{pmatrix} u_t \\ w_t \\ 0 \end{pmatrix}.$$

$$1 \times (m+1) \qquad (m+1) \times (m+1) \qquad (m+1) \times k \qquad (m+1) \times 1$$

The first and second elements of α_t represent y_t^c and y_t^s, respectively.

Kitagawa (1996) and Kitagawa and Gersch (1996) suggested an alternative seasonal component model, which is represented by equation (5) and the following two equations:

$$y_t^c = a_1 y_{t-1}^c + a_2 y_{t-2}^c + \cdots + a_p y_{t-p}^c + u_t, \tag{9}$$

$$y_t^s = -y_{t-1}^s - y_{t-2}^s - \cdots - y_{t-m+1}^s + w_t, \tag{10}$$

where equation (9) may depend on the other exogenous variables x_t as in equation (6). Equations (5), (9) and (10) yield the following state-space model:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t = z\alpha_t + \epsilon_t, \\ &\text{(Transition equation)} \qquad \alpha_t = M\alpha_{t-1} + \eta_t, \end{aligned} \tag{11}$$

where z, α_t, M and η_t are given by:

$$z = (1, 0, \cdots, 0, 1, 0, \cdots, 0), \quad A = \left(\begin{array}{c|c} a_1 & \\ \vdots & I_{p-1} \\ a_{p-1} & \\ \hline a_p & 0 \end{array}\right), \quad B = \left(\begin{array}{c|c} -1 & \\ \vdots & I_{m-2} \\ -1 & \\ \hline -1 & 0 \end{array}\right),$$

$$1 \times (p+m-1) \qquad p \times p \qquad (m-1) \times (m-1)$$

$$M = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}, \qquad \eta_t = (u_t, 0, \cdots, 0, w_t, 0, \cdots, 0)'.$$

$$(p+m-1) \times (p+m-1) \qquad (p+m-1) \times 1$$

All the elements of z and η_t are zeros except for the first and $(p+1)$-th elements. The cyclical component y_t^c and the seasonal component y_t^s are given by the first element of α_t and the $(p+1)$-th element of α_t, respectively. Note that difference between the two systems (8) and (11) is a formulation in the seasonal component, which is described in equations (7) and (10).

Table 1: Revision Process of U.S. National Accounts (Nominal GDP)

	1984	1985	1986	1987	1988	1989	1990
1979	2375.2	2375.2	2464.4^r	2464.4	2464.4	2464.4	2464.4
1980	2586.4^r	2586.4	2684.4^r	2684.4	2684.4	2684.4	2684.4
1981	2904.5^r	2907.5^r	3000.5^r	3000.5	3000.5	3000.5	3000.5
1982	3025.7^r	3021.3^r	3114.8^r	3114.8	3114.8	3114.8	3114.8
1983	3263.4^p	3256.5^r	3350.9^r	3355.9^r	3355.9	3355.9	3355.9
1984	NA	3616.3^p	3726.7^r	3717.5^r	3724.8^r	3724.8	3724.8
1985	NA	NA	3951.8^p	3957.0^r	3970.5^r	3974.1^r	3974.1
1986	NA	NA	NA	4171.2^p	4201.3^r	4205.4^r	4197.2^r
1987	NA	NA	NA	NA	4460.2^p	4497.2^r	4493.8^r
1988	NA	NA	NA	NA	NA	4837.8^p	4847.3^r
1989	NA	NA	NA	NA	NA	NA	5199.6^p
	1991	1992	1993	1994	1995	1996	1997
1979	2464.4	2488.6^r	2488.6	2488.6	2488.6	2557.5^r	2557.5
1980	2684.4	2708.0^r	2708.0	2708.0	2708.0	2784.2^r	2784.2
1981	3000.5	3030.6^r	3030.6	3030.6	3030.6	3115.9^r	3115.9
1982	3114.8	3149.6^r	3149.6	3149.6	3149.6	3242.1^r	3242.1
1983	3355.9	3405.0^r	3405.0	3405.0	3405.0	3514.5^r	3514.5
1984	3724.8	3777.2^r	3777.2	3777.2	3777.2	3902.4^r	3902.4
1985	3974.1	4038.7^r	4038.7	4038.7	4038.7	4180.7^r	4180.7
1986	4197.2	4268.6^r	4268.6	4268.6	4268.6	4422.2^r	4422.2
1987	4486.7^r	4539.9^r	4539.9	4539.9	4539.9	4692.3^r	4692.3
1988	4840.2^r	4900.4^r	4900.4	4900.4	4900.4	5049.6^r	5049.6
1989	5163.2^r	5244.0^r	5250.8^r	5250.8	5250.8	5438.7^r	5438.7
1990	5424.4^p	5513.8^r	5522.2^r	5546.1^r	5546.1	5743.8^r	5743.8
1991	NA	5671.8^p	5677.5^r	5722.8^r	5724.8^r	5916.7^r	5916.7
1992	NA	NA	5945.7^p	6038.5^r	6020.2^r	6244.4^r	6244.4
1993	NA	NA	NA	6374.0^p	6343.3^r	6550.2^r	6553.0^r
1994	NA	NA	NA	NA	6736.9^p	6931.4^r	6935.7^r
1995	NA	NA	NA	NA	NA	7245.8^p	7253.8^r
1996	NA	NA	NA	NA	NA	NA	7580.0^p

2.1.4 Prediction of Final Data Using Preliminary Data

It is well known that economic indicators are usually reported according to the following two steps: (i) the preliminary data are reported and (ii) thereafter we can obtain the final or revised data (see Table 1). The problem is how to estimate the final data (or the revised data) while only the preliminary data are available.

In the case of annual data on the U.S. national accounts, the preliminary data at the present time are reported at the beginning of the next year. The revision process is performed over a few years and every decade, as shown in Table 1, where an example of the nominal gross domestic product data (GDP, billion dollars) is taken.

In Table 1, The preliminary data of 1988, 1992, 1995 and 1996 are taken from *Survey of Current Business* (January, 1989, January, 1993, July, 1996 and February, 1997), while the rest of the preliminary data and all the revised data are from *Economic Report of*

the President (ERP), published from 1984 to 1997. Each column indicates the year when ERP is published, while each row represents the reported data of the corresponding year. The superscripts p and r denote the preliminary data and the data revised in the year corresponding to each column. NA indicates that the data are not available, which implies that the data have not been published yet. For instance, take the GDP data of 1984 (see the corresponding row in Table 1). The preliminary GDP data of 1984 was reported in 1985 (i.e., 3616.3), and it was revised in 1986 for the first time (i,e., 3726.7). In 1987 and 1988, the second and third revised data were published, respectively (i.e., 3717.5 and 3724.8). Since it was not revised in 1989, the GDP data of 1984 published in 1989 is given by 3724.8. Moreover, the GDP data of 1984 was revised as 3777.2 in 1992 and 3902.4 in 1996.

Thus, each data series is revised every year for the first few years and thereafter less frequently. This implies that we cannot really know the true final data, because the data are revised forever while the preliminary data are reported only once. Therefore, it might be possible to consider that the final data are unobservable, which leads to estimation of the final data given the preliminary data.

There is a wide literature dealing with the data revision process. Conrad and Corrado (1979) applied the Kalman filter to improve upon published preliminary estimates of monthly retail sales, using an ARIMA model. Howrey (1978, 1984) used the preliminary data in econometric forecasting and obtained the substantial improvements in forecast accuracy if the preliminary and revised data are used optimally.

In the context of the revision process, the filtering and smoothing techniques are used as follows. There is some relationship between the final and preliminary data, because they are originally same data (see, for example, Conrad and Corrado (1979)). This relationship is referred to as the measurement equation, where the final data is unobservable but the preliminary data is observed. The equation obtained by the underlying economic theory is related to the final data, rather than the preliminary data. This equation is taken as the transition equation. Therefore, we can represent the revision problem with the following state-space form:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t^p = h_t(y_t^f, \epsilon_t), \\ &\text{(Transition equation)} \qquad y_t^f = f_t(y_{t-1}^f, x_t, \eta_t), \end{aligned}$$

where y_t^p and y_t^f denote the preliminary data and the final data, respectively. The unobserved state variable is given by y_t^f, while y_t^p is observable. Thus, the state-space model is utilized to estimate y_t^f (see Mariano and Tanizaki (1995) and Tanizaki and Mariano (1994)).

2.1.5 Estimation of Permanent Consumption

The next application is concerned with estimation of permanent consumption. Total consumption consists of permanent and transitory consumption. This relationship is represented by an identity equation, which corresponds to the measurement equation. Permanent consumption depends on life-time income expected in the future, i.e., permanent income. The following expected utility function of the representative agent is maximized with respect to

permanent consumption (see Hall (1978, 1990)):

$$\max_{\{c_t^p\}} \quad \mathrm{E}_0\left(\sum_t \beta^t u(c_t^p)\right), \text{ subject to } A_{t+1} = R_t(A_t + y_t - c_t),$$

where $0 < \beta < 1$ and $c_t = c_t^p + c_t^T$. c_t, c_t^p, c_t^f, R_t, A_t, y_t, β, $u(\cdot)$ and $\mathrm{E}_t(\cdot)$ denote per capita total consumption, per capita permanent consumption, per capita transitory consumption, the real gross rate of return on savings between periods t and $t+1$, the stock of assets at the beginning of period t, per capita labor income, the discount rate, the representative utility function and the mathematical expectation given information up to t, respectively.

Solving the above maximization problem, we can obtain the transition equation which represents the relationship between c_t^p and c_{t-1}^p. Transitory consumption is assumed to be a random shock with mean zero and variance σ_ϵ.

Under the above setup, the model to this problem is given by:

$$\text{(Measurement equation)} \quad c_t = c_t^p + \epsilon_t,$$

$$\text{(Transition equation)} \qquad \frac{\beta R_{t-1} u'(c_t^p)}{u'(c_{t-1}^p)} = 1 + \eta_t,$$

Note that $c_t^T = \epsilon_t$, which is assumed to be independent of η_t. c_t is observable while both c_t^p and c_t^T are unobservable, where c_t^p is regarded as the state variable to be estimated by the nonlinear filtering and smoothing technique. Thus, we can estimate permanent and transitory consumption separately. Tanizaki (1993a, 1996) and Mariano and Tanizaki (2000) consider the above example, where the utility function of the representative agent is assumed to be a constant relative risk aversion type of utility function. Also see Diebold and Nerlove (1989) for a concise survey of testing the permanent income hypothesis.

2.1.6 Markov Switching Model

The Markov switching model was developed by Hamilton (1989, 1990, 1991, 1993, 1994), where the discrete random variable is taken for the state variable. Consider the k-dimensional discrete state variable, i.e., $\alpha_t = (\alpha_{1t}, \alpha_{2t}, \cdots, \alpha_{kt})'$, where we assume that one of the k elements of α_t is one and the others are zeros.

First, consider the following model:

$$y_t = h_t(\mu_t^*, \epsilon_t), \tag{12}$$

where μ_t^* is a discrete random variable and $h_t(\cdot,\cdot)$ may depend on the other exogenous variable x_t. Assume that μ_t^* depends on the unobserved random variable s_t^*, which is called the state or regime. Suppose that we have k states (or regimes). If $s_t^* = j$, then the process is in regime j and $\mu_t^* = \mu_j$ is taken. We assume that one of the k states at time t occurs depending on time $t-1$.

Define $p = (p_1', p_2', \cdots, p_k')'$ as the transition probability matrix, where $p_i = (p_{i1}, p_{i2}, \cdots, p_{ik})$ for $i = 1, 2, \cdots, k$. Note that $\sum_{i=1}^k p_{ij} = 1$ should be satisfied for all $j = 1, 2, \cdots, k$. p_{ij} implies the conditional probability of $s_t^* = j$ given $s_{t-1}^* = i$, i.e., $p_{ij} \equiv \mathrm{Prob}(s_t^* = j | s_{t-1}^* = i)$. Such a

process is described as an k-state Markov chain with transition probabilities $\{p_{ij}\}_{i,j=1,2,\cdots,k}$. The transition probability p_{ij} gives the probability that state i is followed by state j. Under the above setup, each element of the $k \times 1$ multivariate discrete state variable α_t takes a binary number, i.e.,

$$\alpha_t = \begin{cases} (1,0,0,\cdots,0)', & \text{when } s_t^* = 1, \\ (0,1,0,\cdots,0)', & \text{when } s_t^* = 2, \\ \quad\vdots & \quad\vdots \\ (0,0,0,\cdots,1)', & \text{when } s_t^* = k. \end{cases}$$

Let us define $\mu_t = (\mu_1, \mu_2, \cdots, \mu_k)$, where each element depends on the regime. Then, μ_t^* is rewritten as: $\mu_t^* = \mu\alpha_t$. Accordingly, the model described in equation (12) is represented by the following state-space model:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t = h_t(\mu\alpha_t, \epsilon_t), \\ &\text{(Transition equation)} \qquad \alpha_t = p\alpha_{t-1} + \eta_t, \end{aligned}$$

for $t = 1, 2, \cdots, T$. μ is a $k \times 1$ vector of unknown parameter to be estimated. η_t is distributed as a k-dimensional discrete random variable. The conditional density of α_t given α_{t-1} is represented by $P_\alpha(\alpha_t|\alpha_{t-1}) = \prod_{i=1}^{k}(p_i\alpha_{t-1})^{\alpha_{it}}$, which implies that the probability which event i occurs at time t is $p_i\alpha_{t-1}$.

Thus, it is assumed in the Markov switching model that the economic situation is stochastically switched from one to another for each time. The Markov switching model is similar to the time varying parameter model in Section 2.1.1 in the sense that the parameter changes over time. From specification of the transition equation, however, the time varying parameter model takes into account a gradual shift in the economic structural change but the Markov switching model deals with a sudden shift because μ_t^* is a discrete random variable which depends on state or regime.

2.1.7 Stochastic Variance Models

In this section, we introduce two stochastic variance models (see Taylor (1994) for the stochastic variance models). One is called the autoregressive conditional heteroskedasticity (ARCH) model proposed by Engle (1982) and another is the stochastic volatility model (see Ghysels, Harvey, and Renault (1996)).

Let β be a $k \times 1$ vector of unknown parameters to be estimated. y_t and x_t are assumed to be observable variables. The first-order ARCH model is given by the following two equations:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t = x_t\beta + \alpha_t, \\ &\text{(Transition equation)} \qquad \alpha_t = (\delta_0 + \delta_1\alpha_{t-1}^2)^{1/2}\eta_t, \end{aligned}$$

for $t = 1, 2, \cdots, T$, where $\eta_t \sim N(0,1)$, $0 < \delta_0$ and $0 \leq \delta_1 < 1$ have to be satisfied. The conditional variance of α_t is represented by $\delta_0 + \delta_1\alpha_{t-1}^2$ while the unconditional variance is

given by $\delta_0/(1-\delta_1)$. It might be possible to put the error term (say, ϵ_t) in the measurement equation, i,e., $y_t = x_t\beta + \alpha_t + \epsilon_t$.

As an alternative stochastic variance model, we can consider the stochastic volatility model, which is defined as follows:

$$\begin{aligned} &\text{(Measurement equation)} \quad y_t = x_t\beta + \exp\left(\frac{1}{2}\alpha_t\right)\epsilon_t, \\ &\text{(Transition equation)} \qquad \alpha_t = \delta\alpha_{t-1} + \eta_t, \end{aligned}$$

for $t = 1, 2, \cdots, T$, where $0 \leq \delta < 1$ has to be satisfied. The error terms ϵ_t and η_t are mutually independently distributed.

For the other applications of the state-space model in economics, we can find estimation of the rational expectation models (for example, see Burmeister and Wall (1982), Engle and Watson (1987) and McNelis and Neftci (1983)). See Harvey (1987) for a survey of applications of the Kalman filter model.

2.2 Prediction, Filtering and Smoothing Algorithms

In the state-space model shown in equations (1) and (2), the state variable α_t is the unobserved variable to be estimated. As mentioned above, there are three estimation problems, viz., prediction $a_{t+L|t}$, filtering $a_{t|t}$ and smoothing $a_{t|T}$. In this section, we introduce derivation of $a_{t+L|t}$, $a_{t|t}$ and $a_{t|T}$.

Let $P_y(y_t|\alpha_t)$ be the conditional density function derived from equation (1) and $P_\alpha(\alpha_t|\alpha_{t-1})$ be the conditional density function obtained from equation (2). For prediction, filtering and smoothing, two kinds of algorithms are introduced in this section, i.e, recursive algorithm and non-recursive algorithm, which are described in Sections 2.2.1 and 2.2.2.

2.2.1 Recursive Algorithm

The relationships among density-based recursive algorithms on prediction, filtering and smoothing are as follows: (i) the initial value of prediction is given by filtering, (ii) one-step ahead prediction is utilized to obtain filtering, and (iii) smoothing is derived based on one-step ahead prediction and filtering.

Prediction: The L-step ahead prediction algorithm based on the density function is given by the following recursion:

$$P(\alpha_{t+L}|Y_t) = \int P_\alpha(\alpha_{t+L}|\alpha_{t+L-1})P(\alpha_{t+L-1}|Y_t)\mathrm{d}\alpha_{t+L-1}, \tag{13}$$

for $L = 1, 2, \cdots$. See Kitagawa (1987) and Harvey (1989) for the density-based prediction algorithm. The filtering density $P(\alpha_t|Y_t)$ is assumed to be known, which is the initial density of the density-based L-step ahead prediction algorithm (13). Given $P(\alpha_t|Y_t)$, $P(\alpha_{t+L}|Y_t)$ is recursively obtained given $P(\alpha_{t+L-1}|Y_t)$ for $L = 1, 2, \cdots$.

Filtering: The density-based filtering algorithm is represented by the following two equations:

$$P(\alpha_t|Y_{t-1}) = \int P_\alpha(\alpha_t|\alpha_{t-1})P(\alpha_{t-1}|Y_{t-1})\mathrm{d}\alpha_{t-1}, \tag{14}$$

$$P(\alpha_t|Y_t) = \frac{P_y(y_t|\alpha_t)P(\alpha_t|Y_{t-1})}{\int P_y(y_t|\alpha_t)P(\alpha_t|Y_{t-1})\mathrm{d}\alpha_t}, \tag{15}$$

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

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

where $P_\alpha(\alpha_0)$ denotes the unconditional density of α_0.

Equation (14) corresponds to one-step ahead prediction, which plays a role of predicting α_t using the past information Y_{t-1}. Equation (15) combines the present sample y_t with the past information Y_{t-1}. Equation (14) is called the prediction equation while equation (15) is known as the update equation. Based on the two densities $P_\alpha(\alpha_t|\alpha_{t-1})$ and $P_y(y_t|\alpha_t)$, the density-based filtering algorithm is represented as the following two steps: (i) equation (14) yields $P(\alpha_t|Y_{t-1})$ given $P(\alpha_{t-1}|Y_{t-1})$ and (ii) $P(\alpha_t|Y_t)$ is obtained given $P(\alpha_t|Y_{t-1})$ from equation (15). Thus, repeating predicting and updating for all t, the filtering densities $P(\alpha_t|Y_t)$, $t = 1, 2, \cdots, T$, can be recursively computed.

Smoothing: The density-based fixed-interval smoothing algorithm is represented as (see, for example, Kitagawa (1987, 1996) and Harvey (1989)):[1]

$$P(\alpha_t|Y_T) = P(\alpha_t|Y_t)\int \frac{P(\alpha_{t+1}|Y_T)P_\alpha(\alpha_{t+1}|\alpha_t)}{P(\alpha_{t+1}|Y_t)}\mathrm{d}\alpha_{t+1}, \tag{16}$$

for $t = T-1, T-2, \cdots, 1$, which is a backward recursion. Using equation (16), $P(\alpha_t|Y_T))$ is obtained from $P(\alpha_{t+1}|Y_T)$, given $P_\alpha(\alpha_{t+1}|\alpha_t)$, $P(\alpha_t|Y_t)$ and $P(\alpha_{t+1}|Y_t)$. Note that the smoothing density at time T (i.e., the endpoint case in the smoothing algorithm) is equivalent to the filtering density at time T. Thus, the fixed-interval smoothing algorithm (16) is derived together with the filtering algorithm given by equations (14) and (15).

Mean, Variance and Likelihood Function: Once the density $P(\alpha_r|Y_s)$ is obtained, the conditional expectation of a function $g(\alpha_r)$ is given by:

$$\mathrm{E}(g(\alpha_r)|Y_s) = \int g(\alpha_r)P(\alpha_r|Y_s)\mathrm{d}\alpha_r, \tag{17}$$

[1] As mentioned above, there are three kinds of smoothing algorithms, i.e., fixed-point smoothing, fixed-lag smoothing and fixed-interval smoothing (see, for example, Anderson and Moore (1979) and Harvey (1989)). In a field of economics, the economic situations in the past are often analyzed using the data available at the present time. Accordingly, the fixed-interval smoothing algorithm is taken in this section. That is, we consider evaluating the conditional mean $\mathrm{E}(\alpha_t|Y_T) = a_{t|T}$ for $t = 1, 2, \cdots, T$.

for $(r, s) = (t + L, t), (t, t), (t, T)$. The function $g(\cdot)$ is typically specified as: $g(\alpha_r) = \alpha_r$ for mean or $g(\alpha_r) = (\alpha_r - a_{r|s})(\alpha_r - a_{r|s})'$ for variance, where $a_{r|s} = \mathrm{E}(\alpha_r|Y_s)$.

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

$$P(Y_T) = \prod_{t=1}^{T} P(y_t|Y_{t-1}) = \prod_{t=1}^{T} \left(\int P(y_t|\alpha_t) P(\alpha_t|Y_{t-1}) \mathrm{d}\alpha_t \right), \tag{18}$$

which is called the innovation form of the likelihood function. Recall that $P(y_t|Y_{t-1})$ corresponds to the denominator of equation (15) in the filtering algorithm. Therefore, we do not need any extra computation for evaluation of the likelihood function (18).

In the case where the functions $h_t(\alpha_t, \epsilon_t)$ and $g_t(\alpha_{t-1}, \eta_t)$ are linear and the error terms ϵ_t and η_t are normally distributed, equations (13) – (16) and (18) reduce to the standard linear recursive algorithms on prediction, filtering and smoothing and the likelihood function. See Appendix A for the standard linear recursive algorithms and the likelihood function.

2.2.2 Non-Recursive Algorithm

Usually, the prediction, filtering and smoothing formulas are represented as the recursive algorithms based on the density functions. The non-recursive formulas on prediction, filtering and smoothing are described in Tanizaki (1996, 1997) and Tanizaki and Mariano (1998). However, we can easily show equivalence between both the algorithms (see Appendix C for the proof). In this section, we introduce an alternative solution to a nonlinear and nonnormal prediction, filter and smoothing, which are not conventional recursive algorithms.

Let us define $A_t = \{\alpha_0, \alpha_1, \cdots, \alpha_t\}$, which is a set consisting of the state variables up to time t. Suppose that $P_\alpha(A_t)$ and $P_y(Y_t|A_t)$ are represented as:

$$P_\alpha(A_t) = \begin{cases} P_\alpha(\alpha_0) \prod_{s=1}^{t} P_\alpha(\alpha_s|\alpha_{s-1}), & \text{if } \alpha_0 \text{ is stochastic,} \\ \prod_{s=1}^{t} P_\alpha(\alpha_s|\alpha_{s-1}), & \text{otherwise,} \end{cases} \tag{19}$$

$$P_y(Y_t|A_t) = \prod_{s=1}^{t} P_y(y_s|\alpha_s). \tag{20}$$

Based on the two densities $P(A_t)$ and $P(Y_t|A_t)$, prediction, filtering and smoothing formulas can be derived.[2]

[2]Tanizaki (1996, 1997) and Tanizaki and Mariano (1998) made an attempt to evaluate the prediction, filtering and smoothing estimates, generating random draws of A_T from $P_\alpha(A_T)$, where $P_y(Y_T|A_T)$ is not utilized to generate the random draws.

Prediction: To derive L-step ahead prediction mean $a_{t+L|t}$, we have to obtain the conditional density of A_{t+L} given Y_t, i.e., $P(A_{t+L}|Y_t)$. First, using equations (19) and (20), note that the joint density of A_{t+L} and Y_t is written as: $P_\alpha(A_{t+L}, Y_t) = P_\alpha(A_{t+L})P_y(Y_t|A_t)$. Accordingly, $P(A_{t+L}|Y_t)$ is given by:

$$\begin{aligned} P(A_{t+L}|Y_t) &= \frac{P(A_{t+L}, Y_t)}{\int P(A_{t+L}, Y_t)\mathrm{d}A_{t+L}} \\ &= \frac{P_\alpha(A_{t+L})P_y(Y_t|A_t)}{\int P_\alpha(A_{t+L})P_y(Y_t|A_t)\mathrm{d}A_{t+L}}. \end{aligned} \tag{21}$$

Thus, from equation (21), the L-step ahead prediction density function is represented by:

$$P(\alpha_{t+L}|Y_t) = \int P(A_{t+L}|Y_t)\mathrm{d}A_{t+L-1}, \tag{22}$$

which is not a recursive algorithm. However, it is easily shown that equation (13) is equivalent to equation (22). See Appendix C for equivalence between equations (13) and (22).

Filtering: The conditional density of A_t given Y_t is represented as:

$$\begin{aligned} P(A_t|Y_t) &= \frac{P(A_t, Y_t)}{\int P(A_t, Y_t)\mathrm{d}A_t} \\ &= \frac{P_\alpha(A_t)P_y(Y_t|A_t)}{\int P_\alpha(A_t)P_y(Y_t|A_t)\mathrm{d}A_t}, \end{aligned} \tag{23}$$

where $P(A_t, Y_t) = P_\alpha(A_t)P_y(Y_t|A_t)$ is utilized in the second line of equation (23).

Therefore, integrating $P(A_t|Y_t)$ with respect to A_{t-1}, the filtering density is written as:

$$P(\alpha_t|Y_t) = \int P(A_t|Y_t)\mathrm{d}A_{t-1}. \tag{24}$$

Equation (24) can be derived from equations (14) and (15). See Appendix C.

Smoothing: The conditional density of A_T given Y_T is obtained as follows:

$$\begin{aligned} P(A_T|Y_T) &= \frac{P(A_T, Y_T)}{\int P(A_T, Y_T)\mathrm{d}A_T} \\ &= \frac{P_\alpha(A_T)P_y(Y_T|A_T)}{\int P_\alpha(A_T)P_y(Y_T|A_T)\mathrm{d}A_T}. \end{aligned} \tag{25}$$

Let us define $A_t^* = \{\alpha_t, \alpha_{t+1}, \cdots, \alpha_T\}$, where A_t^* satisfies the following properties: (i) $A_T = A_0^*$ and (ii) $A_T = \{A_t, A_{t+1}^*\}$ for $t = 0, 1, \cdots, T-1$. From equation (25), using A_{t+1}^* the smoothing density at time t, i.e., $P(\alpha_t|Y_T)$, is given by:

$$P(\alpha_t|Y_T) = \frac{\iint P_\alpha(A_T)P_y(Y_T|A_T)\mathrm{d}A_{t-1}\mathrm{d}A_{t+1}^*}{\int P_\alpha(A_T)P_y(Y_T|A_T)\mathrm{d}A_T}, \tag{26}$$

for $t = 1, 2, \cdots, T$. Again, note that it is easy to derive the standard density-based smoothing algorithm (16) from equation (26). See Appendix C for the proof.

Mean, Variance and Likelihood Function: Using equation (21) for prediction, equation (23) for filtering and equation (25) for smoothing, evaluation of the conditional expectation of a function $g(\alpha_t)$ is given by:

$$\begin{aligned}
\mathrm{E}(g(\alpha_t)|Y_s) &= \int g(\alpha_t)P(A_r|Y_s)\mathrm{d}A_r \\
&= \frac{\int g(\alpha_t)P(A_r, Y_s)\mathrm{d}A_r}{\int P(A_r, Y_s)\mathrm{d}A_r} \\
&= \frac{\int g(\alpha_t)P_\alpha(A_r)P_y(Y_s|A_s)\mathrm{d}A_r}{\int P_\alpha(A_r)P_y(Y_s|A_s)\mathrm{d}A_r},
\end{aligned}$$

for $(r, s) = (t + L, t), (t, t), (T, T)$.

In the case where equations (1) and (2) depends on an unknown parameter, the likelihood function to be maximized is written as:

$$P(Y_T) = \int P(A_T, Y_T)\mathrm{d}A_T = \int P_y(Y_T|A_T)P_\alpha(A_T)\mathrm{d}A_T,$$

which corresponds to the denominator of equation (23) in the filtering formula and moreover it is equivalent to the innovation form of the likelihood function given by equation (18).

An alternative estimation method of an unknown parameter is known as the EM algorithm (Expectation-Maximization algorithm), where the expected log-likelihood function is maximized with respect to the parameter, given all the observed data Y_T (see Dempster, Laird and Rubin (1977), Rund (1991) and Laird (1993) for the EM algorithm). That is, for the EM algorithm, the following expected log-likelihood function is maximized:

$$\begin{aligned}
\mathrm{E}\Big(\log(P(A_T, Y_T))|Y_T\Big) &= \mathrm{E}\Big(\log(P_y(Y_T|A_T)P_\alpha(A_T))|Y_T\Big) \\
&= \int \log(P_y(Y_T|A_T)P_\alpha(A_T))P(A_T|Y_T)\mathrm{d}A_T. \qquad (27)
\end{aligned}$$

As for the features of the EM algorithm, it is known that the convergence speed is very slow but it quickly searches the neighborhood of the true parameter value. Shumway and Stoffer (1982) and Tanizaki (1989) applied the EM algorithm to the state-space model in linear and normal case.

3 Nonlinear and Non-Gaussian State-Space Modeling

Kitagawa (1987) and Kramer and Sorenson (1988) proposed the nonlinear and non-Gaussian filter and smoother using numerical integration. It is well known that the numerical integration procedure is not suitable to the multi-dimensional cases from computational point of view. In order to improve this disadvantage of the numerical integration approach, recently, simulation-based nonlinear and nonnormal filters and smoothers have been investigated. Mariano and Tanizaki (1995), Tanizaki (1993a, 1996) and Tanizaki and Mariano (1994) applied Monte Carlo integration with importance sampling to evaluate each integration. Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996) and Chib and Greenberg

(1996) utilized the Gibbs sampler. Gordon, Salmond and Smith (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) proposed a nonlinear filter using a resampling procedure. Hürzeler and Künsch (1998), Mariano and Tanizaki (2000), Tanizaki (1996, 1999a) and Tanizaki and Mariano (1998) introduced a nonlinear and nonnormal filter with rejection sampling. Furthermore, using the Metropolis-Hastings algorithm within Gibbs sampling, a nonlinear and non-Gaussian smoother was proposed by Geweke and Tanizaki (1999a, 1999b). See Appendix B for the sampling techniques such as Monte Carlo integration with importance sampling, rejection sampling, Gibbs sampling and Metropolis-Hastings algorithm. In any case, their nonlinear and non-Gaussian filters and smoothers can be applied to any nonlinear and non-Gaussian cases. Generating random draws from the filtering and smoothing densities at each time, the filtering and smoothing means are evaluated. The obtained estimates clearly go to the true means as number of random draws increases.

3.1 Numerical Integration

In this section, we introduce the nonlinear and non-Gaussian prediction, filtering and smoothing proposed by Kitagawa (1987) and Kramer and Sorenson (1988), which utilizes numerical integration.

Let $\alpha_{i,t}$, $i = 0, 1, \cdots, N$, be the nodes at time t, which are assumed to be sorted in order of size, i.e., $\alpha_{0,t}$ is the smallest value and $\alpha_{N,t}$ is the largest one. For numerical integration, there are some integration methods such as sum of rectangles, sum of trapezoids, Simpson's formula and so on. In this section, for simplicity of discussion, we introduce the numerical integration prediction, filtering and smoothing with the rectangle rule.[3] However, it can be easily extended to the other integration techniques.

Prediction: Equation (13) is approximated as[4]:

$$P(\alpha_{i,t+L}|Y_t) \approx \sum_{j=1}^{N} P_\alpha(\alpha_{i,t+L}|\alpha_{j,t+L-1}) P(\alpha_{j,t+L-1}|Y_t)(\alpha_{j,t+L-1} - \alpha_{j-1,t+L-1}).$$

Thus, a recursive algorithm of the L-step ahead prediction density is obtained.

[3] In Monte Carlo experiments of Section 4, numerical integration is performed by the trapezoid rule.

[4] Note on the expression of numerical integration as follows. Let x be a k-dimensional vector, i.e., $x = (x_1.x_2, \cdots, x_k)$, and x_i be the i-th node, i.e., $x_i = (x_{i,1}.x_{i,2}, \cdots, x_{i,k})$. Suppose that we have N nodes, which are sorted by size. Numerical integration of $g(x)$ with respect to x is expressed as:

$$\begin{aligned}\int g(x)\mathrm{d}x &\approx \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_k=1}^{N} g(x_{i,1}, x_{i,2}, \cdots, x_{i,k}) \\ &\qquad \times (x_{i,1} - x_{i-1,1})(x_{i,2} - x_{i-1,2})\cdots(x_{i,k} - x_{i-1,k}) \\ &\equiv \sum_{i=1}^{N} g(x_i)(x_i - x_{i-1}).\end{aligned}$$

That is, in this paper we write the multivariate numerical integration as the second line of the above equation.

Filtering: Equations (14) and (15) are numerically integrated as follows:

$$P(\alpha_{i,t}|Y_{t-1}) \approx \sum_{j=1}^{N} P_\alpha(\alpha_{i,t}|\alpha_{j,t-1})P(\alpha_{j,t-1}|Y_{t-1})(\alpha_{j,t-1} - \alpha_{j-1,t-1}), \tag{28}$$

$$P(\alpha_{i,t}|Y_t) \approx \frac{P_y(y_t|\alpha_{i,t})P(\alpha_{i,t}|Y_{t-1})}{\sum_{j=1}^{N} P_y(y_t|\alpha_{j,t})P(\alpha_{j,t}|Y_{t-1})(\alpha_{j,t} - \alpha_{j-1,t})}, \tag{29}$$

where the initial condition is given by:

$$P(\alpha_{i,1}|Y_0) \approx \sum_{j=1}^{N} P_\alpha(\alpha_{i,1}|\alpha_{j,0})P_\alpha(\alpha_{j,0})(\alpha_{j,0} - \alpha_{j-1,0}),$$

if α_0 is stochastic and

$$P(\alpha_{i,1}|Y_0) = P_\alpha(\alpha_{i,1}|\alpha_0),$$

otherwise. $P(\alpha_{i,t}|Y_{t-1})$ for $i = 1, 2, \cdots, N$ is obtained given $P(\alpha_{j,t-1}|Y_{t-1})$ for $j = 1, 2, \cdots, N$ from equation (28), while $P(\alpha_{i,t}|Y_t)$ for $i = 1, 2, \cdots, N$ is computed given $P(\alpha_{j,t}|Y_{t-1})$ for $j = 1, 2, \cdots, N$ using equation (29). Thus, equations (28) and (29) give us the recursive algorithm.

Smoothing: Equation (16) is approximated as:

$$P(\alpha_{i,t}|Y_T) \approx P(\alpha_{i,t}|Y_t) \sum_{j=1}^{N} \frac{P(\alpha_{j,t+1}|Y_T)P_\alpha(\alpha_{j,t+1}|\alpha_{i,t})}{P(\alpha_{j,t+1}|Y_t)}(\alpha_{j,t+1} - \alpha_{j-1,t+1}),$$

which is a backward recursive algorithm. That is, given the two densities $P_\alpha(\alpha_{j,t+1}|\alpha_{i,t})$ and $P(\alpha_{j,t+1}|Y_t)$, the smoothing density at time t (i.e., $P(\alpha_{i,t}|Y_T)$) is computed from the smoothing density at time $t+1$ (i.e., $P(\alpha_{j,t+1}|Y_T)$).

Mean, Variance and Likelihood Function: Equation (17) is evaluated as:

$$\mathrm{E}(g(\alpha_r)|Y_s) \approx \sum_{j=1}^{N} g(\alpha_{j,r})P(\alpha_{j,r}|Y_s)(\alpha_{j,r} - \alpha_{j-1,r}),$$

for $(r, s) = (t + L, t), (t, t), (t, T)$.

The likelihood function (18) is computed as:

$$P(Y_T) \approx \prod_{t=1}^{T} \left(\sum_{j=1}^{N} P(y_t|\alpha_{j,t})P(\alpha_{j,t}|Y_{t-1})(\alpha_{j,t} - \alpha_{j-1,t}) \right),$$

which is obtained from the denominator of the filtering recursion (29).

Some Comments: According to the numerical integration approach, clearly we can obtain asymptotically unbiased estimate of $\mathrm{E}(g(\alpha_r)|Y_s)$. Note that an asymptotically unbiased estimate in this case implies that the estimate goes to the true state variable as N increases for all t. However, some problems are encountered. First, when we compute the densities numerically, computation errors are accumulated and therefore integration of $P(\alpha_r|Y_s)$ is not equal to one in practice. In such situations, as L or t increases, we have poor density approximation. To improve this problem, we need *ad hoc* modification which satisfies the following condition: $\sum_{i=1}^{N} P(\alpha_{i,r}|Y_s)(\alpha_{i,r} - \alpha_{i-1,r}) = 1$, which comes from $\int P(\alpha_r|Y_s)\mathrm{d}\alpha_r = 1$. Thus, $P(\alpha_r|Y_s)$ should be re-computed for each time r.

Another problem is choice of the nodes. Density approximation is imprecise when number of nodes is small and/or location of the nodes is not appropriate. If the nodes $\alpha_{i,t}$ for $i = 1, 2, \cdots, N$ are distributed away from the true distribution of α_r given Y_s, the estimate of $\mathrm{E}(g(\alpha_r)|Y_s)$ becomes unrealistic. The nodes $\alpha_{i,r}$ for $i = 1, 2, \cdots, N$ have to be overlapped with $P(\alpha_r|Y_s)$. However, the true distribution of α_r given Y_s is unknown in practice. Let $a^*_{t|s}$ be the extended Kalman filtering mean and $D^*_{t|s}$ be the vector which consists of the diagonal elements of the extended Kalman filtering variance $\Sigma^*_{t|s}$. In the case of filtering, Tanizaki (1993, 1996) Tanizaki and Mariano (1994) suggested taking the nodes $\alpha_{i,t}$ from the interval: $[a^*_{t|s} - (cD^*_{t|s})^{1/2}, a^*_{t|s} + (cD^*_{t|s})^{1/2}]$, where half of the nodes are taken from $s = t$ and the rest of the nodes are from $s = t - 1$. c is a fixed value and $c \geq 1$ is preferable (remember that the probability is 99% when $c = 9$ in the case of normal distribution). Thus, the numerical integration procedure is obtained based on the extended Kalman filter. It might be plausible to consider that mean and variance from the extended Kalman filter are not far from those from the true distribution.

Moreover, when the numerical integration approach is applied to the higher-dimensional cases, it takes an extraordinarily long time and also results in tedious programming. Computational time for $P(\alpha_{i,r}|Y_s)$, $i = 1, 2, \cdots, N$, is proportional to $L \times kN \times N^k$ for L-step ahead prediction, $T \times (kN \times N^k + kN \times N^k)$ for filtering, and $T \times kN \times N^k$ for smoothing.[5] The procedures shown in the proceeding sections can be easily extended to the higher-dimensional cases in the sense of both computational time and programming.

Thus, the numerical integration approach gives us the asymptotically unbiased estimator as number of the nodes increases, but we should keep in mind the following two problems: first, location of nodes has to be set by a researcher, and second, computational time increases more than proportionally as the dimension of the state variable is high.

3.2 Monte Carlo Integration with Importance Sampling

Mariano and Tanizaki (1995), Tanizaki (1993a, 1996, 1999b) and Tanizaki and Mariano (1994) proposed the nonlinear filtering and smoothing algorithms with importance sampling, where recursive algorithms of density functions are converted by those of weight functions.

[5]Note that we need $kN \times N^k$ computation to obtain each integration in equations (13) – (16), where k denotes number of elements of the state variable α_t.

Define the weight function as:

$$\omega(\alpha_r|Y_s) \equiv \frac{P(\alpha_r|Y_s)}{P_*(\alpha_r)},$$

for $(r,s) = (t+L,t),(t,t),(t,T)$. The density function $P_*(\alpha_r)$ has to be appropriately specified by a researcher, which is called the importance density. Let $\alpha_{i,r}$, $i = 1,2,\cdots,N$, be the random numbers generated from the importance density $P_*(\alpha_r)$ for $r = 1,2,\cdots,T$.

Prediction: Using the weight function, the prediction equation (13) is transformed into:

$$\begin{aligned}\omega(\alpha_{i,t+L}|Y_t) &= \int \frac{P_\alpha(\alpha_{i,t+L}|\alpha_{t+L-1})}{P_*(\alpha_{i,t+L})}\omega(\alpha_{t+L-1}|Y_t)P_*(\alpha_{t+L-1})\mathrm{d}\alpha_{t+L-1} \\ &\approx \frac{1}{N}\sum_{j=1}^{N}\frac{P_\alpha(\alpha_{i,t+L}|\alpha_{j,t+L-1})}{P_*(\alpha_{i,t+L})}\omega(\alpha_{j,t+L-1}|Y_t),\end{aligned}$$

for $L = 1,2,\cdots$. In the second line, integration is approximated using $\alpha_{j,t+L-1}$, $j = 1,2,\cdots,N$, which are generated from $P_*(\alpha_{t+L-1})$. Therefore, given $\omega(\alpha_{j,t+L-1}|Y_t)$, we can compute $\omega(\alpha_{j,t+L}|Y_t)$. Thus, the weight functions $\omega(\alpha_{j,t+L}|Y_t)$, $L = 1,2,\cdots$, are recursively obtained.

Filtering: Using the weight functions $\omega(\alpha_t|Y_s)$ for $s = t-1,t$, the density-based filtering algorithm given by equations (14) and (15) is rewritten as follows:

$$\begin{aligned}\omega(\alpha_{i,t}|Y_{t-1}) &= \int \frac{P_\alpha(\alpha_{i,t}|\alpha_{t-1})}{P_*(\alpha_{i,t})}\omega(\alpha_{t-1}|Y_{t-1})P_*(\alpha_{t-1})\mathrm{d}\alpha_{t-1} \\ &\approx \frac{1}{N}\sum_{j=1}^{N}\frac{P_\alpha(\alpha_{i,t}|\alpha_{j,t-1})}{P_*(\alpha_{i,t})}\omega(\alpha_{j,t-1}|Y_{t-1}),\end{aligned} \tag{30}$$

$$\begin{aligned}\omega(\alpha_{i,t}|Y_t) &= \frac{P_y(y_t|\alpha_{i,t})\omega(\alpha_{i,t}|Y_{t-1})}{\int P_y(y_t|\alpha_t)\omega(\alpha_t|Y_{t-1})P_*(\alpha_t)\mathrm{d}\alpha_t} \\ &\approx \frac{P_y(y_t|\alpha_{i,t})\omega(\alpha_{i,t}|Y_{t-1})}{(1/N)\sum_{j=1}^{N}P_y(y_t|\alpha_{j,t})\omega(\alpha_{j,t}|Y_{t-1})},\end{aligned} \tag{31}$$

where the initial condition of the weight function is given by:

$$\omega(\alpha_{i,1}|Y_0) \approx \frac{1}{N}\sum_{j=1}^{N}\frac{P_\alpha(\alpha_{i,1}|\alpha_{j,0})}{P_*(\alpha_{i,1})}\omega(\alpha_{j,0}|Y_0),$$

when α_0 is stochastic and

$$\omega(\alpha_{i,1}|Y_0) = \frac{P_\alpha(\alpha_{i,1}|\alpha_0)}{P_*(\alpha_{i,1})},$$

when α_0 is nonstochastic. Note that $\omega(\alpha_{j,0}|Y_0)$ is given by: $\omega(\alpha_{j,0}|Y_0) \equiv P_\alpha(\alpha_{j,0})/P_*(\alpha_{j,0})$. The recursive algorithm shown in equations (30) and (31) is implemented by the following two steps: (i) given $\omega(\alpha_{j,t-1}|Y_{t-1})$ for $j = 1, 2, \cdots, N$, $\omega(\alpha_{i,t}|Y_{t-1})$ for $i = 1, 2, \cdots, N$ are obtained from equation (30), and (ii) given $\omega(\alpha_{j,t}|Y_{t-1})$ for $j = 1, 2, \cdots, N$, $\omega(\alpha_{i,t}|Y_t)$ for $i = 1, 2, \cdots, N$ are given by equation (31). Thus, the weight functions $\omega(\alpha_{j,t}|Y_t)$ are recursively computed for $t = 1, 2, \cdots, T$ and $j = 1, 2, \cdots, N$.

Smoothing: Using Monte Carlo integration with importance sampling, equation (16) is rewritten as:

$$\begin{aligned}\omega(\alpha_{i,t}|Y_T) &= \omega(\alpha_{i,t}|Y_t)\int \frac{\omega(\alpha_{t+1}|Y_T)P_\alpha(\alpha_{t+1}|\alpha_{i,t})}{\omega(\alpha_{t+1}|Y_t)P_*(\alpha_{t+1})}P_*(\alpha_{t+1})\mathrm{d}\alpha_{t+1} \\ &\approx \omega(\alpha_{i,t}|Y_t)\frac{1}{N}\sum_{j=1}^{N}\frac{\omega(\alpha_{j,t+1}|Y_T)P_\alpha(\alpha_{j,t+1}|\alpha_{i,t})}{\omega(\alpha_{j,t+1}|Y_t)P_*(\alpha_{j,t+1})}, \end{aligned} \tag{32}$$

which is a backward recursive algorithm of the weight functions. Thus, based on the filtering formula (30) and (31), $\omega(\alpha_{i,t}|Y_T)$ is computed from $\omega(\alpha_{j,t+1}|Y_T)$ from equation (32).

Mean, Variance and Likelihood Function: Using the weight functions $\omega(\alpha_{i,r}|Y_s)$ for $t = 1, 2, \cdots, T$, equation (17) is approximated as follows:

$$\begin{aligned}\mathrm{E}(g(\alpha_r)|Y_s) &= \int g(\alpha_r)\omega(\alpha_r|Y_s)P_*(\alpha_r)\mathrm{d}\alpha_r \\ &\approx \frac{1}{N}\sum_{i=1}^{N} g(\alpha_{i,r})\omega(\alpha_{i,r}|Y_s).\end{aligned}$$

From definition of the weight function, we can evaluate the conditional density $P(\alpha_r|Y_s)$ at $\alpha_r = \alpha_{i,r}$, which is represented by:

$$P(\alpha_{i,r}|Y_s) = \omega(\alpha_{i,r}|Y_s)P_*(\alpha_{i,r}),$$

for $(r, s) = (t + L, t), (t, t), (t, T)$.

The likelihood function (18) can be transformed into:

$$\begin{aligned}P(Y_T) &= \prod_{t=1}^{T}\left(\int P_y(y_t|\alpha_t)\omega(\alpha_t|Y_{t-1})P_*(\alpha_t)\mathrm{d}\alpha_t\right) \\ &\approx \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} P_y(y_t|\alpha_{i,t})\omega(\alpha_{i,t}|Y_{t-1})\right),\end{aligned}$$

which corresponds to the denominator of equation (31).

Some Comments: Compared with the numerical integration procedure, the most attractive features of the importance sampling procedure is less computational burden, especially, in the higher-dimensional cases of the state vector. That is, for numerical integration, the nodes have to be sorted for all elements of α_t and sum of the rectangles has to be performed for all elements of α_t. For Monte Carlo integration, however, computational time does not depend on the dimension of the state vector, which is proportional to $L \times N \times N$ for L-step ahead prediction, $T \times (N \times N + N)$ for filtering, and $T \times N \times N$ for smoothing.[6] Thus, from computational point of view, the importance sampling procedure is more applicable in practice.

The disadvantage of the importance sampling approach is choice of the importance density. It is important that the importance density should not be too different from the target density (see Geweke (1988, 1989a, 1989b)). If the importance density $P_*(\alpha_r)$ is chosen away from $P(\alpha_r|Y_s)$, the weight function $\omega(\alpha_r|Y_s)$ becomes unrealistic. Accordingly, the obtained estimates are biased in such a case. In the case of filtering, the importance density $P_*(\alpha_t)$ needs to cover both $P(\alpha_t|Y_{t-1})$ and $P(\alpha_t|Y_t)$ over the range of α_t. Usually, peak and range of $P(\alpha_t|Y_{t-1})$ is different from those of $P(\alpha_t|Y_t)$. In addition, range of $P(\alpha_t|Y_{t-1})$ is larger than that of $P(\alpha_t|Y_t)$ in general. For the importance sampling filter, it is important that the two densities $P(\alpha_t|Y_s)$, $s = t-1, t$, have to be approximated by one importance density $P_*(\alpha_t)$. From some simulation studies, Tanizaki (1996) pointed out that we should choose the importance density with the following conditions: (i) the importance density should have a wide range of distribution, compared with the original distribution and (ii) center of the importance density should be close to that of the original density but we do not have to pay too much attention to center of the importance density. Tanizaki and Mariano (1994) and Mariano and Tanizaki (1995) suggested using the bimodal distribution for $P_*(\alpha_t)$, i.e.,

$$P_*(\alpha_t) = \frac{1}{2}N(a^*_{t|t-1}, c\Sigma^*_{t|t-1}) + \frac{1}{2}N(a^*_{t|t}, c\Sigma^*_{t|t}),$$

which denotes the average of two normal densities, where $a^*_{t|s}$ and $\Sigma^*_{t|s}$ for $s = t-1, t$ are the extended Kalman filter estimates and c is constant. Since $P_*(\alpha_t)$ should be broader than $P(\alpha_t|Y_{t-1})$ and $P(\alpha_t|Y_t)$, $c \geq 1$ must be chosen. The peak and range of $P(\alpha_t|Y_s)$ are not known in practice, but mean and variance of the state variable can be estimated by the extended Kalman filter even if it is the biased estimator. It is appropriate to consider that the extended Kalman filter estimates are not too far from the true values. Therefore, it might be plausible to take the bimodal distribution for the importance density. Thus, the importance sampling estimator would be improved by utilizing the importance density based on the extended Kalman filter.

Clearly, the Monte Carlo integration yields the simulation errors. Therefore, the numerical integration gives us a better approximation than the Monte Carlo integration. In order to avoid the problem, Geweke (1988) proposed the antithetic Monte Carlo integration and Tanizaki (1999b) suggested using the antithetic Monte Carlo integration in the nonlinear filtering framework.

[6]Note that we need $N \times N$ order of computation for each integration in equations (13), (14) and (16), and N order of computation for equation (15).

Moreover, another problem, which is also found in the numerical integration procedure, is that we need to pay attention to accumulation of the computation errors. In order to reduce unreliability of the approximation, we must have the restriction of $(1/N)\sum_{t=1}^{T}\omega(\alpha_{i,r}|Y_s)=1$ for all r and s, which comes from one of the properties of the density function.

Thus, each integration in the density-based algorithms is evaluated by numerical integration or Monte-Carlo integration with importance sampling, where a researcher has to assume the nodes or the importance density. In the next two sections, we introduce the nonlinear filters and smoothers which do not require *ad hoc* assumption such as choice of the nodes or choice of the importance density.

3.3 Resampling

Using numerical integration or Monte Carlo integration, an attempt has been made to evaluate each integration in equations (13) – (16). Recently, as computer progresses day by day, expectation (17) is evaluated generating random draws directly from the prediction, filtering and smoothing densities, where each integration is not computed explicitly. Gordon, Salmond and Smith (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) proposed nonlinear filter and smoother using a resampling procedure (see, for example, Smith and Gelfand (1992) for the resampling procedure). Let us define $\alpha_{i,r|s}$ as the i-th random draw of α_r generated from the conditional density $P(\alpha_r|Y_s)$. We consider how to generate random draws $\alpha_{i,r|s}$, $i=1,2,\cdots,N$.

Prediction: The prediction estimate is very simple and easy (see, for example, Tanizaki and Mariano (1998)). Suppose that $\alpha_{i,t+L-1|t}$, $i=1,2,\cdots,N$, are available, which are the random draws generated from $(L-1)$-step ahead prediction density. Consider generating the random draws from L-step ahead prediction density, given $\alpha_{i,t+L-1|t}$, $i=1,2,\cdots,N$.

Given $\alpha_{j,t+L-1|t}$, $j=1,2,\cdots,N$, the L-step ahead prediction algorithm (13) is approximately represented as:

$$P(\alpha_{t+L}|Y_t)\approx\frac{1}{N}\sum_{j=1}^{N}P(\alpha_{t+L}|\alpha_{j,t+L-1|t}). \tag{33}$$

Therefore, $\alpha_{i,t+L|t}$ is generated as follows. Pick up $\alpha_{j,t+L-1|t}$ randomly (i.e., pick j with equal probability) and generate a random number of η_{t+L} (i.e., $\eta_{i,t+L}$), and we have a random draw of α_{t+L} (i.e., $\alpha_{i,t+L|t}$) from the transition equation: $\alpha_{i,t+L|t}=f_{t+L}(\alpha_{j,t+L-1|t},\eta_{i,t+L})$.

Thus, $\alpha_{i,t+L|t}$ is recursively obtained based on $\alpha_{j,t+L-1|t}$ for $L=1,2,\cdots$. The initial random draws of the prediction algorithm (i.e., $\alpha_{i,t|t}$, $i=1,2,\cdots,N$) are generated in the filtering algorithm shown below.

Finally, note that the resampling procedure is not used to generate random draws from the prediction density (i.e., $\alpha_{i,t+L|t}$, $i=1,2,\cdots,N$), which is utilized in the following filtering and smoothing procedures.

Filtering: As discussed above, when $\alpha_{j,t-1|t-1}$ for $j = 1, 2, \cdots, N$ are available, $\alpha_{i,t|t-1}$ for $i = 1, 2, \cdots, N$ can be obtained from the prediction equation: $\alpha_{i,t|t-1} = f_t(\alpha_{j,t-1|t-1}, \eta_{i,t})$, where $\alpha_{j,t-1|t-1}$ is chosen with probability $1/N$ and $\eta_{i,t}$ denotes the i-th random draw of η_t.

For filtering, we consider generating random draws α_t from the filtering density $P(\alpha_t|Y_t)$. Based on the random draws $\alpha_{i,t|t-1}$ for $i = 1, 2, \cdots, N$, equation (15) is approximately rewritten as follows:

$$P(\alpha_{i,t|t-1}|Y_t) \approx \frac{P_y(y_t|\alpha_{i,t|t-1})}{\sum_{j=1}^{N} P_y(y_t|\alpha_{j,t|t-1})}. \tag{34}$$

Note that $P(\alpha_{i,t|t-1}|Y_{t-1}) \approx 1/N$ is used to derive equation (34).[7] Equation (34) is interpreted as follows. The probability which α_t takes $\alpha_{j,t|t-1}$ is approximately given by $P(\alpha_{j,t|t-1}|Y_t)$. Accordingly, $\alpha_{i,t|t}$ is chosen from $\alpha_{j,t|t-1}$, $j = 1, 2, \cdots, N$, with probability $P(\alpha_{j,t|t-1}|Y_t)$. That is, the i-th random draw of α_t from $P(\alpha_t|Y_t)$, i.e., $\alpha_{i,t|t}$, is resampled as:

$$\alpha_{i,t|t} = \begin{cases} \alpha_{1,t|t-1}, & \text{with probability } P(\alpha_{1,t|t-1}|Y_t), \\ \alpha_{2,t|t-1}, & \text{with probability } P(\alpha_{2,t|t-1}|Y_t), \\ \quad \vdots & \quad \vdots \\ \alpha_{N,t|t-1}, & \text{with probability } P(\alpha_{N,t|t-1}|Y_t). \end{cases} \tag{35}$$

Thus, in order to obtain random draws from the filtering density, first we have to compute $P(\alpha_{j,t|t-1}|Y_t)$ for all $j = 1, 2, \cdots, N$ and next obtain $\alpha_{i,t|t}$ for $i = 1, 2, \cdots, N$ by resampling. In practice, a uniform random draw between zero and one (say, u) is generated and $\alpha_{j,t|t-1}$ is taken as $\alpha_{i,t|t}$ when $\omega_{j-1} \leq u < \omega_j$, where $\omega_j \equiv \sum_{m=1}^{j} P(\alpha_{m,t|t-1}|Y_t)$ and $\omega_o \equiv 0$.

Smoothing: Assume that $\alpha_{j,t+1|T}$, $j = 1, 2, \cdots, N$, are available, which denote the random numbers of α_{t+1} generated from $P(\alpha_{t+1}|Y_T)$. The problem is to generate random draws of α_t from $P(\alpha_t|Y_T)$, given $\alpha_{j,t+1|T}$, $j = 1, 2, \cdots, T$. Equation (16) is approximated as follows:

$$\begin{aligned} P(\alpha_t|Y_T) &\approx P(\alpha_t|Y_t) \frac{1}{N} \sum_{j=1}^{N} \frac{P_\alpha(\alpha_{j,t+1|T}|\alpha_t)}{P(\alpha_{j,t+1|T}|Y_t)} \\ &\approx P(\alpha_t|Y_t) \sum_{j=1}^{N} \frac{P_\alpha(\alpha_{j,t+1|T}|\alpha_t)}{\sum_{m=1}^{N} P_\alpha(\alpha_{j,t+1|T}|\alpha_{m,t|t})} \end{aligned} \tag{36}$$

Note that in the first line the integration is approximated by random numbers $\alpha_{j,t+1|T}$ and that in the second approximation we utilize $P(\alpha_{j,t+1|T}|Y_t) \approx (1/N) \sum_{m=1}^{N} P_\alpha(\alpha_{j,t+1|T}|\alpha_{m,t|t})$.

Moreover, $P(\alpha_{m,t|t}|Y_t) = 1/N$ is obtained because $\alpha_{m,t|t}$ is a random number of α_t from $P(\alpha_t|Y_t)$. Therefore, equation (36) is rewritten as:

$$P(\alpha_{i,t|t}|Y_T) \approx \frac{1}{N} \sum_{j=1}^{N} \frac{P_\alpha(\alpha_{j,t+1|T}|\alpha_{i,t|t})}{\sum_{m=1}^{N} P_\alpha(\alpha_{j,t+1|T}|\alpha_{m,t|t})}. \tag{37}$$

[7] Let x_i, $i = 1, 2, \cdots, N$, be the random draws from the density $P_x(x)$. Then, for all $i \neq j$, the probability which we have x_i should be equal to the probability which x_j occurs. In the case where we have N random draws of x, therefore, $P_x(x_i) \approx 1/N$ is obtained.

Thus, it is shown from equation (37) that $\alpha_{i,t|T}$ is chosen from $\alpha_{j,t|t}$, $j = 1, 2, \cdots, N$, with probability $P(\alpha_{j,t|t}|Y_T)$. That is, the i-th random draw of α_t from $P(\alpha_t|Y_T)$, i.e., $\alpha_{i,t|T}$, is resampled as:

$$\alpha_{i,t|T} = \begin{cases} \alpha_{1,t|t}, & \text{with probability } P(\alpha_{1,t|t}|Y_T), \\ \alpha_{2,t|t}, & \text{with probability } P(\alpha_{2,t|t}|Y_T), \\ \quad \vdots & \qquad \vdots \\ \alpha_{N,t|t}, & \text{with probability } P(\alpha_{N,t|t}|Y_T). \end{cases} \tag{38}$$

To obtain random draws from the smoothing density, first we compute $P(\alpha_{j,t|t}|Y_T)$ for all $j = 1, 2, \cdots, N$ and next obtain $\alpha_{i,t|T}$ for $i = 1, 2, \cdots, N$ by resampling. Note that $\alpha_{i,T|T}$ is a random draw from the filtering density at time T. As discussed above, a uniform random number has to be compared with $\sum_{m=1}^{i} P(\alpha_{i,t|t}|Y_T)$ when we obtain $\alpha_{i,t|T}$ in practice.

From resampling procedures (35) and (38), both $\alpha_{i,t|t}$ and $\alpha_{i,t|T}$ are chosen from $\alpha_{i,t|t-1}$, $i = 1, 2, \cdots, N$, with different probabilities.

Mean, Variance and Likelihood Function: When the random draws (i.e., $\alpha_{i,r|s}$) are available, the conditional mean (17) is simply computed as:

$$\mathrm{E}(g(\alpha_r)|Y_s) \approx \frac{1}{N} \sum_{i=1}^{N} g(\alpha_{i,r|s}), \tag{39}$$

for $(r, s) = (t + L, t), (t, t), (t, T)$.

Using one-step ahead prediction random draws, the likelihood function (18) is approximately rewritten as:

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

Some Comments: In both numerical integration and importance sampling procedures, each integration in equations (13) – (16) is evaluated to compute prediction, filtering and smoothing means. However, in the resampling procedure proposed by Gordon, Salmond and Smith (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996), prediction, filtering and smoothing means are computed using the random draws which are directly generated from each appropriate density, where we do not need to evaluate any integration. The obtained algorithm is based on the recursive algorithm of random draws. The resampling procedure improves over the nonlinear filters based on the numerical integration and importance sampling procedures from simplicity of computer programming and no *ad hoc* assumptions such as choice of the nodes for numerical integration (Kitagawa (1987) and Kramer and Sorenson (1988)), choice of the importance density for Monte Carlo integration (Tanizaki (1993a), Tanizaki and Mariano (1994) and Mariano and Tanizaki (1995)). It might be expected that the estimates obtained from the resampling procedure go to the true state vector values as number of the random draws increases.

The disadvantage of the resampling procedure is that computational time extremely increases, especially for smoothing, as number of random draws (i.e., N) increases. Computational burden is given by the order of $L \times N$ for prediction, $T \times (N + N^2)$ for filtering, and $T \times (N^3 + N^2)$ for smoothing.[8]

3.4 Rejection Sampling

An alternative approach to nonlinear and nonnormal prediction, filtering and smoothing algorithms is proposed by Tanizaki (1996, 1999a), Tanizaki and Mariano (1998) and Mariano and Tanizaki (2000). Given random draws of the state vector which are directly generated from the filtering or smoothing density, the filtering or smoothing mean is recursively obtained, where we do not evaluate any integration included in the density-based algorithms (13) – (16). The procedure introduced in this section is similar to the resampling procedure in Section 3.4 However,for random number generation, rejection sampling is adopted in this section.

L-step ahead prediction is derived in the exactly same fashion as in Section 3.3. Therefore, we start with filtering in this section.

Filtering: Suppose that the random draws $\alpha_{i,t-1|t-1}$, $i = 1, 2, \cdots, N$, are available. Then, we consider generating $\alpha_{i,t|t}$, $i = 1, 2, \cdots, N$. By substituting equation (14) into equation (15), the filtering density at time t, $P(\alpha_t|Y_t)$, is approximated as:

$$\begin{aligned}
P(\alpha_t|Y_t) &= \frac{1}{\gamma_t} \int P_y(y_t|\alpha_t) P_\alpha(\alpha_t|\alpha_{t-1}) P(\alpha_{t-1}|Y_{t-1}) \mathrm{d}\alpha_{t-1} \\
&\approx \sum_{i=1}^{N} \frac{\gamma_{i,t}}{\gamma_t} \frac{1}{N} \left(\frac{P_y(y_t|\alpha_t) P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}} \right) \\
&\approx \sum_{i=1}^{N} \frac{\widehat{\gamma}_{i,t}}{\widehat{\gamma}_t} \frac{1}{N} \left(\frac{P_y(y_t|\alpha_t) P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}} \right) \\
&\equiv \sum_{i=1}^{N} q_{i,t} \left(\frac{P_y(y_t|\alpha_t) P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma_{i,t}} \right), \qquad (40)
\end{aligned}$$

where γ_t denotes the denominator of equation (15). In the second line, approximation is used by generating $\alpha_{i,t-1|t-1}$ from $P(\alpha_{t-1}|Y_{t-1})$. In the third line, γ_t is approximately equal to $\widehat{\gamma}_t$, which is rewritten as follows:

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

[8] In the case of filtering, we need the following computation: N for (34) and N^2 for comparison between the uniform random draw u and $\sum_{m=1}^{j} P(\alpha_{m,t|t-1}|Y_t)$. In smoothing case, the orders of computational burden are N^3 for equation (37) and N^2 for comparison between the uniform random draw u and $\sum_{m=1}^{i} P(\alpha_{m,t|t}|Y_T)$.

where $\alpha_{ji,t|t-1}$ denotes the j-th random draw of α_t generated from $P(\alpha_t|\alpha_{i,t-1|t-1})$. Moreover, $\gamma_{i,t}$ and $\widehat{\gamma}_{i,t}$ are represented as:

$$\gamma_{i,t} \equiv \int P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\mathrm{d}\alpha_t \approx \frac{1}{N}\sum_{j=1}^{N} P_y(y_t|\alpha_{ji,t|t-1}) \equiv \widehat{\gamma}_{i,t}.$$

Furthermore, in the fourth line, $q_{i,t}$ is defined as $q_{i,t} \equiv \widehat{\gamma}_{i,t}/N\widehat{\gamma}_t$.

Thus, from equation (40), $P(\alpha_t|Y_t)$ is approximated as a mixture of N distributions with probability $q_{i,t}$, $i = 1, 2, \cdots, N$. That is, $\alpha_{i,t-1|t-1}$ is chosen with probability $q_{i,t}$. Therefore, given $\alpha_{i,t-1|t-1}$, the next problem is how to generate a random draw of α_t from the density:

$$P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma_{i,t}. \tag{41}$$

Rejection sampling is applied to obtain the random draw.

Let $P_*(z)$ be the proposal density. The acceptance probability $\omega(z)$ is defined as:

$$\omega(z) = \frac{P_y(y_t|z)P_\alpha(z|\alpha_{i,t-1|t-1})/P_*(z)}{\sup_z P_y(y_t|z)P_\alpha(z|\alpha_{i,t-1|t-1})/P_*(z)}, \tag{42}$$

where we require the assumption which the denominator in equation (42) exists.

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 $u \leq \omega(z)$ and go back to (ii) otherwise, (iv) repeat (i) – (iii) N times for $j = 1, 2, \cdots, N$, and (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 denominator in equation (42) exists (i.e., $\sup_z P_y(y_t|z)P_\alpha(z|\alpha_{i,t-1|t-1})/P_*(z) < \infty$), rejection sampling is very inefficient in the sense of computational time (i.e., number of rejection increases) if the acceptance probability $\omega(z)$ is close to zero. In such a case, (ii) and (iii) are repeated forever.

For choice of the proposal density, we might consider the following candidates, i.e., (i) one is $P_*(\alpha_t) = P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})$ and (ii) another is $P_*(\alpha_t) = N(a^*_{t|t}, c\Sigma^*_{t|t})$, where c is a constant and $a^*_{t|t}$ and $\Sigma^*_{t|t}$ denote the first- and second-moments obtained from the extended Kalman filter. Note that c should be greater than one, because the proposal density should have larger variance than the target density (see Appendix B). However, as mentioned above, because the denominator in equation (42) has to exist, it might be easier and better to use the candidate (i) rather than the candidate (ii).[9]

Smoothing: Hürzeler and Künsch (1998), Tanizaki (1996) and Tanizaki and Mariano (1998) proposed the fixed-interval smoothing algorithm using rejection sampling, where a backward recursive algorithm of the random draws is derived. Suppose that $\alpha_{i,t+1|T}$ for $i = 1, 2, \cdots, N$ are available. Then we consider how to generate $\alpha_{i,t|T}$ for $i = 1, 2, \cdots, N$.

[9]Even if we take the candidate (i) as the proposal density, we possibly have the case where the supremum does not exit, depending on the functional form of $P_y(\alpha_t|y_t)$.

In order to obtain the rejection sampling smoothing, first, note that each component in the smoothing algorithm (16) is transformed as follows:

$$P(\alpha_t|Y_t) \approx \frac{1}{\gamma_t}\left(\frac{1}{N}\sum_{i=1}^{N} P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\right), \tag{43}$$

$$\int \frac{P(\alpha_{t+1}|Y_T)P(\alpha_{t+1}|\alpha_t)}{P(\alpha_{t+1}|Y_t)}\mathrm{d}\alpha_{t+1} \approx \frac{1}{N}\sum_{j=1}^{N}\frac{P_\alpha(\alpha_{j,t+1|T}|\alpha_t)}{P(\alpha_{j,t+1|T}|Y_t)}. \tag{44}$$

In equation (43), the integration in the first equality of equation (40) is approximated by using $\alpha_{i,t-1|t-1}$, $i = 1, 2, \cdots, N$, which are generated from $P(\alpha_{t-1}|Y_{t-1})$. Moreover, in equation (44), the integration is approximated using $\alpha_{j,t+1|T}$, $j = 1, 2, \cdots, N$, which are generated from $P(\alpha_{t+1}|Y_T)$.

Using equations (43) and (44), the smoothing density (16) is approximated as:

$$\begin{aligned}
&P(\alpha_t|Y_T)\\
\approx&\frac{1}{\gamma_t}\left(\frac{1}{N}\sum_{i=1}^{N} P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\right)\left(\frac{1}{N}\sum_{j=1}^{N}\frac{P_\alpha(\alpha_{j,t+1|T}|\alpha_t)}{P(\alpha_{j,t+1|T}|Y_t)}\right)\\
=&\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\gamma^*_{ij,t}}{N^2\gamma_t P(\alpha_{j,t+1|T}|Y_t)}\left(\frac{P_y(y_t|\alpha_t)P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma^*_{ij,t}}\right)\\
\approx&\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\widehat{\gamma}^*_{ij,t}}{N\sum_{i=1}^{N}\widehat{\gamma}^*_{ij,t}}\left(\frac{P_y(y_t|\alpha_t)P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma^*_{ij,t}}\right)\\
=&\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q^*_{ij,t}}{N}\left(\frac{P_y(y_t|\alpha_t)P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})}{\gamma^*_{ij,t}}\right),
\end{aligned} \tag{45}$$

for $t = T-1, T-2, \cdots, 1$, which is a backward recursion. $\gamma^*_{ij,t}$ is approximated as $\widehat{\gamma}^*_{ij,t}$, which is represented by:

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

$P(\alpha_{j,t+1|T}|Y_t)$ is approximated as follows:

$$\begin{aligned}
&P(\alpha_{j,t+1|T}|Y_t)\\
=&\int P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P(\alpha_t|Y_t)\mathrm{d}\alpha_t\\
=&\frac{1}{\gamma_t}\iint P_y(y_t|\alpha_t)P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})P(\alpha_{t-1}|Y_{t-1})\mathrm{d}\alpha_{t-1}\mathrm{d}\alpha_t\\
\approx&\frac{1}{\gamma_t}\frac{1}{N}\sum_{i=1}^{N}\int P_y(y_t|\alpha_t)P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})\mathrm{d}\alpha_t
\end{aligned}$$

$$\approx \frac{1}{\gamma_t}\frac{1}{N^2}\sum_{i=1}^{N}\sum_{m=1}^{N} P_y(y_t|\alpha_{mi,t|t-1})P_\alpha(\alpha_{j,t+1|T}|\alpha_{mi,t|t-1})$$
$$= \frac{1}{\gamma_t}\frac{1}{N}\sum_{i=1}^{N}\widehat{\gamma}^*_{ij,t}.$$

Moreover, in the fourth line of equation (45), $q^*_{ij,t}$ is defined as: $q^*_{ij,t} \equiv \widehat{\gamma}^*_{ij,t}/\sum_{i=1}^{N}\widehat{\gamma}^*_{ij,t}$. Accordingly, we have the equality: $\sum_{i=1}^{N} q^*_{ij,t} = 1$ for all j.

Thus, in equation (45), given $\alpha_{i,t-1|t-1}$ and $\alpha_{j,t+1|T}$, the smoothing density $P(\alpha_t|Y_T)$ is approximated as a mixture of N^2 distributions. $\alpha_{i,t-1|t-1}$ and $\alpha_{j,t+1|T}$ are chosen with probabilities $q^*_{ij,t}$ and $1/N$. The next problem is how to generate a random draw of α_t from the density:

$$P_y(y_t|\alpha_t)P_\alpha(\alpha_{j,t+1|T}|\alpha_t)P_\alpha(\alpha_t|\alpha_{i,t-1|t-1})/\gamma^*_{ij,t}. \tag{46}$$

The random draws from the smoothing density (45) are generated by rejection sampling.

The acceptance probability $\omega(z)$ is defined as:

$$\omega(z) = \frac{P_y(y_t|z)P_\alpha(\alpha_{j,t+1|T}|z)P_\alpha(z|\alpha_{i,t-1|t-1})/P_*(z)}{\sup_z P_y(y_t|z)P_\alpha(\alpha_{j,t+1|T}|z)P_\alpha(z|\alpha_{i,t-1|t-1})/P_*(z)}. \tag{47}$$

As discussed above, we need the assumption that the denominator in equation (47) is bounded.

The following procedure is taken for rejection sampling smoother: (i) pick one of $\alpha_{j,t+1|T}$, $j = 1, 2, \cdots, N$ with probability $1/N$ and one of $\alpha_{i,t-1|t-1}$, $i = 1, 2, \cdots, N$ with probability $q^*_{ij,t}$, (ii) generate a random draw z from the proposal density $P_*(\cdot)$ and a uniform random draw u from the interval between zero and one, (iii) take z as $\alpha_{m,t|T}$ if $u \leq \omega(z)$ and go back to (ii) otherwise, (iv) repeat (i) – (iii) N times for $m = 1, 2, \cdots, N$, and (v) repeat (i) – (iv) T times for $t = T-1, T-2, \cdots, 1$.

Mean, Variance and Likelihood Function: As discussed in Section 3.3, utilizing the random draws $\alpha_{i,r|s}$ for $i = 1, 2, \cdots, N$ and $(r, s) = (t+L, t), (t, t), (t, T)$, the expectation of a function $g(\cdot)$ is given by equation (39).

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

$$P(Y_T) = \prod_{t=1}^{T}\gamma_t \approx \prod_{t=1}^{T}\widehat{\gamma}_t = \prod_{t=1}^{T}\left(\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N} P_y(y_t|\alpha_{ji,t|t-1})\right).$$

Some Comments: In the rejection sampling procedure, as in Section 3.3, we do not evaluate any integration included in the density-based filtering algorithm, where we utilize the random draws only and the recursive algorithm of random draws are derived. Compared with the numerical integration procedure and the importance sampling approach, the rejection sampling procedure has the advantages from simplicity of computer programming and no *ad hoc* assumptions. As number of random draws increases, the obtained estimates approach the true state vector values. The rejection sampling procedure does not need as

many random draws as the resampling procedure to obtain the same precision of the filtering and smoothing estimates, from difference between the random number generation methods.

Computational burden is proportional to $L \times N$ for prediction, $T \times (N^2 + N \times (N + A))$ for filtering, and $T \times N \times (N^2 + N + A)$ for smoothing, where A denotes the average number of rejection to obtain one random draw.[10] Rejection sampling depends on A, which is related to the acceptance probability. Small acceptance probability implies large number of A. Thus, the random number generator by rejection sampling is inefficient when the acceptance probability $\omega(\cdot)$ is close to zero. That is, for rejection sampling, it sometimes takes a long time, especially when the acceptance probability $\omega(\cdot)$ is small. See, for example, Carlin and Polson (1991) and Carlin, Polson and Stoffer (1992). Thus, the rejection sampling has the disadvantage that we cannot exactly predict computation time.

To improve the rejection sampling procedure in the sense of computation time, we have the following strategies. One is that we may pick another j and/or i in procedure (i) and repeat procedures (ii) and (iii) again when the acceptance probability $\omega(\cdot)$ is too small. Alternatively, we may switch random number generation from rejection sampling to the Metropolis-Hastings algorithm when $\omega(\cdot)$ is too small (see Appendix B for the Metropolis-Hastings algorithm). That is, when repeating procedures (ii) and (iii), 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. Furthermore, 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.,

$$\begin{cases} \widehat{\gamma}_t \equiv \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} P_y(y_t|\alpha_{i,t|t-1}), \\ \widehat{\gamma}_{i,t} \equiv P_y(y_t|\alpha_{i,t|t-1}), \\ \widehat{\gamma}^*_{ij,t} \equiv P_y(y_t|\alpha_{i,t|t-1}) P_\alpha(\alpha_{j,t+1|T}|\alpha_{i,t|t-1}). \end{cases} \tag{48}$$

Under the above re-definitions, computational burden reduces to the order of $T \times (N + N \times (N + A))$ for filtering, and $T \times N \times (N + N + A)$ for smoothing.

3.5 Markov Chain Monte Carlo (Metropolis-Hastings Algorithm within Gibbs Sampling)

Carlin, Polson and Stoffer (1992), Carter and Kohn (1994, 1996) and Chib and Greenberg (1996) introduced the nonlinear and/or non-Gaussian state-space models with Gibbs sampling. They investigated the nonlinear state-space models in the Bayesian framework. Moreover, the state-space models which they used are quite restricted to some functional forms, because 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, 1999b) proposed the nonlinear and non-Gaussian smoother using both Gibbs sampling and the Metropolis-Hastings algorithm, which would be suitable to any

[10] Filtering computes N^2 to obtain $\widehat{\gamma}_t$ and $N \times (N + A)$ to generate N filtering random draws, while smoothing uses $(N^2 + N + A)$ to obtain one smoothing random draw at each time.

nonlinear and non-Gaussian state-space model. In this section, the nonlinear and/or non-Gaussian smoother proposed by Geweke and Tanizaki (1999a, 1999b) is introduced, where the measurement and transition equations are specified in any general formulation and the error terms are not necessarily normal.

Smoothing: We generate random draws of A_T directly from $P(A_T|Y_T)$, shown in equation (25). According to the Gibbs sampling theory, random draws of A_T from $P(A_T|Y_T)$ are based on those of α_t from $P(\alpha_t|A_{t-1}, A^*_{t+1}, Y_T)$ for $t = 1, 2, \cdots, T$, which is derived from equations (19) and (20) and represented as the following equation:

$$\begin{aligned}
&P(\alpha_t|A_{t-1}, A^*_{t+1}, Y_T) \\
=&\frac{P(A_T|Y_T)}{P(A_{t-1}, A^*_{t+1}|Y_T)} \\
=&\frac{P_y(Y_T|A_T)P_\alpha(A_T)}{\int P_y(Y_T|A_T)P_\alpha(A_T)\mathrm{d}\alpha_t} \\
\propto&\begin{cases} P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})P_\alpha(\alpha_{t+1}|\alpha_t), & \text{if } t = 1, 2, \cdots, T-1, \\ P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1}), & \text{if } t = T \text{ (i.e., endpoint),} \end{cases}
\end{aligned} \tag{49}$$

where the third line of equation (49) utilizes equations (19) and (20). Thus, equation (49) implies that a kernel of $P(\alpha_t|A_{t-1}, A^*_{t+1}, Y_T)$ is given by $P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})P_\alpha(\alpha_{t+1}|\alpha_t)$ when $t = 1, 2, \cdots, T-1$ and $P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})$ when $t = T$ (i.e., endpoint).

Using a kernel of $P(\alpha_t|A_{t-1}, A^*_{t+1}, Y_T)$, we consider generating random draws of A_T directly from $P(A_T|Y_T)$. Here, the Gibbs sampler is applied to random number generation. Let $\alpha_{i,t}$ be the i-th random draw of the state vector at time t. Define $A_{i,t}$ and $A^*_{i,t}$ as $A_{i,t} = \{\alpha_{i,0}, \alpha_{i,1}, \cdots, \alpha_{i,t}\}$ and $A^*_{i,t} = \{\alpha_{i,t}, \alpha_{i,t+1}, \cdots, \alpha_{i,T}\}$, respectively, which are the i-th random draws of A_t and A^*_t.

Let $P_*(z|x)$ be the proposal density, which is the conditional distribution of z given x. We should choose the proposal density $P_*(z|x)$ such that random draws can be easily and quickly generated. Define the acceptance probability $\omega(x, z)$ as follows:

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

To generate random draws from $P(A_T|Y_T)$, the following procedure is taken: (i) pick up appropriate values for $\alpha_{1,0}$ and $\alpha_{0,t}$, $t = 1, 2, \cdots, T$, (ii) generate a random draw z from $P_*(\cdot|\alpha_{i-1,t})$ and a uniform random draw u from the uniform distribution between zero and one, (iii) set $\alpha_{i,t} = z$ if $u \leq \omega(\alpha_{i-1,t}, z)$ and set $\alpha_{i,t} = \alpha_{i-1,t}$ otherwise, (iv) repeat (ii) and (iii) for $t = 1, 2, \cdots, T$, and (v) repeat (ii) – (iv) for $i = 1, 2, \cdots, N$.

Note that the Metropolis-Hastings algorithm is used in procedures (ii) and (iii). In procedure (i), typically, the smoothing estimates based on the extended Kalman filter are

taken for $\alpha_{0,t}$, $t = 1, 2, \cdots, T$. $\alpha_{i,0}$ for $i = 1, 2, \cdots, N$ depend on the underlying assumption of α_0. That is, $\alpha_{i,0}$ for $i = 1, 2, \cdots, N$ are generated from $P_\alpha(\alpha_0)$ if α_0 is stochastic and they are fixed as α_0 for all i if α_0 is nonstochastic.

Mean, Variance and Likelihood Function: Based on the random draws $\alpha_{i,t}$ for $i = 1, 2, \cdots, N$, evaluation of $\mathrm{E}(g(\alpha_t)|Y_T)$ is simply obtained as the arithmetic average of $g(\alpha_{i,t})$, $i = 1, 2, \cdots, N$, which is represented by:

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

Usually, 10 – 20% of N is taken for M, which implies that the first M random draws are discarded.

For estimation of unknown parameters, the conditional expectation of the log-likelihood function given by equation (27) is maximized (i.e., EM algorithm). Using the random draws generated from $P(A_T|Y_T)$, equation (27) is evaluated as follows:

$$\mathrm{E}\Big(\log(P(A_T, Y_T))|Y_T\Big) \approx \frac{1}{N-M} \sum_{i=M+1}^{N} \log(P_y(Y_T|A_{i,T})P_\alpha(A_{i,T})).$$

Some Comments: For the Markov chain Monte Carlo method, numerous number of random draws have to be generated to obtain the same precision of the smoothing estimates as both the resampling procedure and the rejection sampling procedure. Generally, it is intractable to generate $\alpha_{i,t}$ from $P(\alpha_t|A_{i,t-1}, A^*_{i-1,t+1}, Y_T)$. In such a case, there two ways to generate random draws, i.e., one is rejection sampling and another is the Metropolis-Hastings algorithm. It is known that rejection sampling sometimes takes a long time computationally or it is not feasible in the case where the acceptance probability does not exists. Therefore, in order to generate numerous random draws very quickly, we apply the Metropolis-Hastings algorithm in procedures (ii) and (iii).

The Metropolis-Hastings algorithm has the problem of specifying the proposal density, which is the crucial criticism. 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., $P_*(z|x) = P_\alpha(z|\alpha_{i,t-1})$. 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., $P_*(z|x) = N(a^*_{t|T}, c\Sigma^*_{t|T})$ (this is also the independence chain), where $a^*_{t|T}$ and $\Sigma^*_{t|T}$ 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. Third, we may take the proposal density called the random walk chain, i.e., $P_*(z|x) = P_*(z - x)$, which is written as $P_*(z|x) = N(x, c\Sigma^*_{t|T})$. Fourth, in the case where the state variable α_t lies on an interval, a uniform distribution between the interval might be taken as the proposal density. In Monte Carlo experiments of Section 4, $P_\alpha(z|\alpha_{i,t-1})$ and $N(a^*_{t|T}, c\Sigma^*_{t|T})$ are examined for the proposal density $P_*(z|x)$.

In this section, the filtering problem has not been discussed until now. The filtering procedure might be implemented as follows. Simply, replacing T by t in procedures (i) – (v), the random draws from the filtering density $P(\alpha_t|Y_t)$ are given by $\alpha_{i,t}$, $i = 1, 2, \cdots, N$, where t corresponds to the endpoint in the procedure (i) – (v). Recall that the random draws obtained at the endpoint represent the filtering random draws. In addition to procedures (i) – (v), we should put the following procedure: (vi) repeat (i) – (v) for $t = 1, 2, \cdots, T$. Accordingly, filtering is more computer-intensive than smoothing.[11] For prediction, equation (33) is utilized given the filtering random draws. Computational burden is as follows. Number of iteration is given by $T \times N$ for smoothing and $\sum_{t=1}^{T} Nt = NT(T-1)/2$ for filtering. It seems that the Markov chain Monte Carlo procedure is less computational than any other estimators. However, the Markov chain Monte-Carlo methods need a lot of random draws, compared with the independence Monte-Carlo methods such as importance sampling and rejection sampling, because in the Markov chain Monte-Carlo methods we usually discard the first 10% – 20% random draws and a random draw is positively correlated with the next random draw in general. Moreover, it is known that convergence of the Gibbs sampler is very slow especially in the case where there is high correlation between α_t and α_{t-1} (see Chib and Greenberg (1995)).

3.6 Quasi-Filter and Quasi-Smoother

The resampling procedure and the rejection sampling approach takes a lot of time and the Markov chain Monte Carlo procedure sometimes has the feature of slow convergence. Improving the problems, Tanizaki (1998) proposed a quasi approach to nonlinear and/or non-Gaussian state estimation.

In this section, let $\alpha_{i,r|s}$ be the i-th random draw of α_r from $P(\alpha_r|Y_s)$, which is the same notation as in Sections 3.3 and 3.4. The procedure in this section has the same computational burden as that in Section 3.5. Convergence speed is faster because the Gibbs sampler is not applied in this section.

For L-step ahead prediction, as discussed in Section 3.3, equation (33) is utilized based on filtering random draws $\alpha_{j,t|t}$, $j = 1, 2, \cdots, N$.

Filtering: We generate $\alpha_{i,t|t}$ based on $\alpha_{i,t-1|t-1}$, as in Section 3.4. In the filtering algorithm (i) – (v) of Section 3.4, when $q_{i,t} \approx 1/N$ is approximately taken, (iv) can be exchanged with (v). In (ii) and (iii), we can apply the Metropolis-Hastings algorithm to the random number generation.

Therefore, given $\alpha_{i,t-1|t-1}$ and $q_{i,t} \approx 1/N$, we generate a random draw of α_t from the

[11] In the standard density-based smoothing algorithm, $P(\alpha_t|Y_{t-1})$ and $P(\alpha_t|Y_t)$ are required. After $P(\alpha_t|Y_{t-1})$ and $P(\alpha_t|Y_t)$ are computed for $t = 1, 2, \cdots, T$, $P(\alpha_t|Y_T)$ are obtained by the backward recursive algorithm. See Section 2.2.1 for the standard algorithms. Thus, clearly smoothing is more computer-intensive than filtering in the conventional density-based recursive algorithm. However, according to the Markov chain Monte Carlo procedure, it is easier to compute smoothing, rather than filtering.

density (41) by the Metropolis-Hastings algorithm. Define the acceptance probability as:

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

The estimation procedure is as follows: (i) given $\alpha_{i,t-1|t-1}$, 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, (ii) set $\alpha_{i,t|t} = z$ if $u \leq \omega(\alpha_{i-1,t|t}, z)$ and set $\alpha_{i,t|t} = \alpha_{i-1,t|t}$ otherwise, (iii) repeat (i) and (ii) for $t = 1, 2, \cdots, T$, and (iv) repeat (i) – (iii) for $i = 1, 2, \cdots, N$. The extended Kalman filter estimates might be chosen for the initial random draws, i.e., $\alpha_{0,t|t} = a^*_{t|t}$ for $t = 1, 2, \cdots, T$. Thus, $\alpha_{i,t|t}$, $i = 1, 2, \cdots, N$, are obtained recursively.

Smoothing: Similarly, consider generating $\alpha_{i,t|T}$ based on $\alpha_{i,t-1|t-1}$ and $\alpha_{i,t+1|T}$. In the smoothing algorithm (i) – (v) of Section 3.4, when $q^*_{ij,t} \approx 1/N$ is approximately taken, (iv) is exchanged with (v). Moreover, in (ii) and (iii), the Metropolis-Hastings algorithm is applied.

Thus, given $\alpha_{i,t-1|t-1}$, $\alpha_{i,t+1|T}$ and $q^*_{ij,t} \approx 1/N$, we generate a random draw of α_t from the density (46) by the Metropolis-Hastings algorithm. Define the acceptance probability as:

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

The smoothing procedure is implemented as the following backward recursive algorithm: (i) given $\alpha_{i,t+1|T}$ and $\alpha_{i,t-1|t-1}$, 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, (ii) set $\alpha_{i,t|T} = z$ if $u \leq \omega(\alpha_{i-1,t|T}, z)$ and set $\alpha_{i,t|T} = \alpha_{i-1,t|T}$ otherwise, (iii) repeat (i) and (ii) for $t = T-1, T-2, \cdots, 1$ (i.e., backward recursion), and (iv) repeat (i) – (iii) for $i = 1, 2, \cdots, N$. The extended Kalman smoothed estimates might be chosen for the initial random draws, i.e., $\alpha_{0,t|T} = a^*_{t|T}$ for $t = 1, 2, \cdots, T$

Mean, Variance and Likelihood Function: Evaluation of $\mathrm{E}(g(\alpha_r)|Y_s)$ is given by:

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

for all $(r,s) = (t+L,t), (t,t), (t,T)$. As discussed in Section 3.5, the first M random draws are excluded because of stability of the random draws.

The likelihood function (18) 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., $\eta_{i,t}$).

Some Comments: The rejection sampling procedure discussed in Section 3.4 is slightly modified from the following two points: we impose the approximations of $q_{i,t} = 1/N$ and $q^*_{ij,t} = 1/N$ for all i and j, and the Metropolis-Hastings algorithm is utilized instead of rejection sampling in order to reduce computational burden. Computational time is the order of $T \times N$ for filtering, which implies the least computational burden of all the filters introduced in this paper, and $T \times N$ for smoothing.[12]

For precision the estimates, the quasi-filter and quasi-smoother might be inferior to the resampling and rejection sampling procedures, because the approximations of $q_{i,t} = 1/N$ and $q^*_{ij,t} = 1/N$ are taken. If the approximations are appropriate, the obtained estimates become plausible. However, it is important to note as follows. For smoothing, the numerical integration, the importance sampling, the resampling and the rejection sampling procedures require a great amount of data storage (i.e., the order of $T \times N$) but the Markov chain Monte Carlo procedure and the quasi-filter and quasi-smoother do not need too much storage (i.e., the order of T). From capacity of computer memory, the estimators discussed in Sections 3.5 and 3.6 are more useful than the other estimators.[13]

As pointed out in Section 3.5, we might consider several candidates of the proposal density. In Monte Carlo studies of Section 4, the following two types of the proposal density are examined: $P_*(z|x) = P_\alpha(z|\alpha_{i,t-1|t-1})$ and $P_*(z|x) = N(a^*_{t|s}, c\Sigma^*_{t|s})$ ($s = t$ for filtering and $s = T$ for smoothing).

4 Monte Carlo Studies

In this section, by Monte Carlo studies, we compare numerical accuracy for all the estimators introduced in Section 3. 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 (Stochastic Volatility Model): The system is represented as: $y_t = \exp\left(\frac{1}{2}\alpha_t\right)\epsilon_t$ and $\alpha_t = \delta\alpha_{t-1} + \eta_t$ for $0 \leq \delta < 1$.

[12]For smoothing, the Markov chain Monte Carlo procedure in Section 3.5 is less computational then the quasi-smoother in this section. To obtain the smoothing random draws, the quasi-smoother requires the filtering random draws while the Markov chain Monte Carlo procedure does not utilize them.

[13]However, we should keep in mind that the Markov chain Monte Carlo approach uses a large number of random draws from the convergence property of the Gibbs sampler and that the quasi-filter and the quasi-smoother do not give us the exact solution.

Table 2: Extended Kalman Filter and Smoother (δ known)

$T \backslash \delta$	I			II		III		IV
	0.5	0.9	1.0	0.5	0.9	0.5	0.9	
Filtering								
20	0.7292	0.7760	0.7897	0.7016	0.6748	1.1487	2.0909	23.351
40	0.7334	0.7793	0.7928	0.7039	0.6503	1.1577	2.2135	21.275
100	0.7307	0.7747	0.7878	0.7061	0.6439	1.1609	2.2656	22.336
Smoothing								
20	0.7054	0.6855	0.6746	0.7016	0.6748	1.1487	2.0909	19.597
40	0.7096	0.6876	0.6761	0.7039	0.6503	1.1577	2.2135	18.685
100	0.7057	0.6822	0.6705	0.7061	0.6439	1.1609	2.2656	19.079

Simulation III (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 $0 \leq \delta < 1$. In this simulation study, $\delta_0 = 1 - \delta$ is taken.[14]

Simulation IV (Nonstationary Growth Model): Take the univariate system[15]: $y_t = \alpha_t^2/20 + \epsilon_t$ 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.

We compare the extended Kalman filter and smoother[16] and the nonlinear and non-Gaussian filters and smoothers introduced in Section 3. 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 $T = 20, 40, 100$ 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: $\text{RMSE} = (1/T)\sum_{t=1}^{T} \text{MSE}_{t|s}^{1/2}$, 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 $\widehat{\alpha}_{t|t}$ takes the state variable estimated by each estimator while α_t denotes the artificially simulated state variable.[17] Note that the superscript (g) denotes the g-th simulation run, where $G = 1000$ is taken.

All the values in Tables 2 – 8 indicate the RMSE's, defined above, The RMSE's for both filtering and smoothing estimates are reported in all the tables except for Table 7.[18] In all the tables, I, II, III and IV denote the corresponding simulation study. T, N and δ denote the sample size, number of random draws and the unknown parameter which is included in

[14] $\delta_0 = 1 - \delta$ in the transition equation implies that the unconditional variance of α_t is normalized to be one.

[15] This system is examined in Kitagawa (1987, 1996) and Carlin, Polson and Stoffer (1992), which is called the nonstationary growth model in Carlin, Polson and Stoffer (1992).

[16] In the case where the system is linear and normal (i.e., Simulation I), the extended Kalman filter and smoother reduce to the conventional Kalman filter and smoother.

[17] Note that $\text{MSE}_{t|s}$ goes to $\Sigma_{t|s} \equiv \text{Var}(\alpha_t|Y_s)$, as number of random draws (i.e., N) is large for $s = t, T$.

[18] Table 7 represents the RMSE's for smoothing and not for filtering.

Table 3: Numerical Integration (δ known)

T \ δ	N	c	I 0.5	I 0.9	I 1.0	II 0.5	II 0.9	III 0.5	III 0.9	IV
Filtering										
20	50	9	0.7292	0.7760	0.7897	0.6855	0.5821	0.9212	1.0931	11.868
		16	0.7292	0.7760	0.7897	0.6854	0.5687	0.9212	1.0930	9.452
		25	0.7291	0.7761	0.7898	0.6855	0.5626	0.9212	1.0930	8.336
	100	9	0.7292	0.7760	0.7897	0.6855	0.5810	0.9212	1.0931	11.745
		16	0.7292	0.7760	0.7897	0.6854	0.5681	0.9212	1.0931	8.910
		25	0.7292	0.7760	0.7897	0.6854	0.5621	0.9212	1.0931	7.796
40	50	9	0.7334	0.7793	0.7928	0.6889	0.5644	0.9329	1.1077	11.122
		16	0.7334	0.7793	0.7928	0.6889	0.5523	0.9329	1.1076	8.241
		25	0.7334	0.7794	0.7928	0.6890	0.5467	0.9329	1.1077	7.591
	100	9	0.7334	0.7793	0.7928	0.6889	0.5635	0.9329	1.1077	10.811
		16	0.7334	0.7793	0.7928	0.6889	0.5518	0.9329	1.1077	7.835
		25	0.7334	0.7793	0.7928	0.6889	0.5463	0.9329	1.1077	6.881
100	50	9	0.7307	0.7748	0.7878	0.6903	0.5579	0.9353	1.1138	11.589
		16	0.7307	0.7748	0.7879	0.6903	0.5463	0.9353	1.1138	9.157
		25	0.7307	0.7748	0.7879	0.6903	0.5414	0.9353	1.1138	8.141
	100	9	0.7307	0.7747	0.7878	0.6903	0.5570	0.9353	1.1138	11.271
		16	0.7307	0.7747	0.7878	0.6903	0.5458	0.9353	1.1138	8.711
		25	0.7307	0.7748	0.7878	0.6903	0.5411	0.9353	1.1138	7.566
Smoothing										
20	50	9	0.7054	0.6855	0.6746	0.6784	0.5663	0.8928	0.9303	12.932
		16	0.7054	0.6855	0.6746	0.6784	0.5506	0.8928	0.9303	9.865
		25	0.7053	0.6856	0.6747	0.6784	0.5434	0.8928	0.9303	7.869
	100	9	0.7054	0.6855	0.6746	0.6784	0.5651	0.8928	0.9303	12.642
		16	0.7054	0.6855	0.6746	0.6784	0.5498	0.8928	0.9303	9.555
		25	0.7054	0.6855	0.6746	0.6784	0.5429	0.8928	0.9303	7.437
40	50	9	0.7096	0.6877	0.6761	0.6804	0.5494	0.9039	0.9308	12.297
		16	0.7096	0.6877	0.6761	0.6804	0.5353	0.9039	0.9308	8.999
		25	0.7096	0.6878	0.6762	0.6805	0.5288	0.9039	0.9308	7.622
	100	9	0.7096	0.6876	0.6761	0.6804	0.5484	0.9039	0.9308	11.988
		16	0.7096	0.6876	0.6761	0.6804	0.5347	0.9039	0.9308	8.514
		25	0.7096	0.6876	0.6761	0.6804	0.5285	0.9039	0.9308	7.039
100	50	9	0.7057	0.6823	0.6705	0.6805	0.5416	0.9062	0.9322	12.582
		16	0.7057	0.6823	0.6705	0.6806	0.5279	0.9062	0.9322	9.345
		25	0.7057	0.6823	0.6705	0.6806	0.5221	0.9062	0.9323	7.743
	100	9	0.7057	0.6822	0.6705	0.6805	0.5405	0.9062	0.9322	12.227
		16	0.7057	0.6822	0.6705	0.6805	0.5273	0.9062	0.9322	8.764
		25	0.7057	0.6823	0.6705	0.6805	0.5218	0.9062	0.9322	7.160

Table 4: Importance Sampling (δ known)

T \ δ	N	c	I 0.5	I 0.9	I 1.0	II 0.5	II 0.9	III 0.5	III 0.9	IV
Filtering										
20	50	4	0.7365	0.7848	0.7990	0.6920	0.5995	0.9326	1.1216	14.270
		9	0.7398	0.7890	0.8035	0.6939	0.5801	0.9382	1.1366	12.420
		16	0.7437	0.7940	0.8089	0.6965	0.5712	0.9439	1.1493	10.923
	100	4	0.7333	0.7813	0.7954	0.6890	0.5921	0.9280	1.1091	13.517
		9	0.7354	0.7837	0.7978	0.6900	0.5727	0.9308	1.1186	10.831
		16	0.7373	0.7860	0.8003	0.6914	0.5654	0.9334	1.1251	8.977
40	50	4	0.7418	0.7894	0.8035	0.6966	0.5810	0.9446	1.1388	13.808
		9	0.7447	0.7930	0.8074	0.6989	0.5629	0.9495	1.1541	12.053
		16	0.7478	0.7970	0.8117	0.7013	0.5554	0.9544	1.1697	10.602
	100	4	0.7373	0.7839	0.7975	0.6926	0.5731	0.9397	1.1245	13.063
		9	0.7384	0.7852	0.7989	0.6933	0.5562	0.9414	1.1298	10.154
		16	0.7400	0.7872	0.8012	0.6942	0.5494	0.9439	1.1363	8.288
100	50	4	0.7388	0.7847	0.7984	0.6976	0.5745	0.9478	1.1495	14.117
		9	0.7414	0.7880	0.8020	0.6991	0.5573	0.9520	1.1635	12.485
		16	0.7449	0.7924	0.8067	0.7017	0.5506	0.9567	1.1829	11.158
	100	4	0.7348	0.7801	0.7936	0.6942	0.5665	0.9413	1.1289	13.395
		9	0.7359	0.7812	0.7948	0.6949	0.5506	0.9434	1.1345	10.604
		16	0.7373	0.7829	0.7965	0.6958	0.5450	0.9456	1.1422	8.888
Smoothing										
20	50	4	0.7130	0.6962	0.6864	0.6855	0.5855	0.9048	0.9635	14.851
		9	0.7163	0.7009	0.6913	0.6877	0.5633	0.9108	0.9832	13.584
		16	0.7203	0.7066	0.6974	0.6903	0.5527	0.9171	0.9978	11.811
	100	4	0.7096	0.6910	0.6806	0.6821	0.5758	0.9000	0.9477	14.613
		9	0.7119	0.6940	0.6838	0.6833	0.5545	0.9031	0.9571	12.001
		16	0.7140	0.6967	0.6866	0.6846	0.5464	0.9060	0.9659	9.983
40	50	4	0.7187	0.6986	0.6876	0.6887	0.5669	0.9165	0.9691	14.638
		9	0.7217	0.7025	0.6915	0.6911	0.5466	0.9216	0.9842	13.335
		16	0.7249	0.7071	0.6967	0.6937	0.5384	0.9267	1.0008	11.824
	100	4	0.7137	0.6928	0.6814	0.6843	0.5580	0.9122	0.9489	14.192
		9	0.7147	0.6942	0.6831	0.6850	0.5392	0.9138	0.9555	11.596
		16	0.7164	0.6964	0.6855	0.6860	0.5317	0.9166	0.9655	9.458
100	50	4	0.7141	0.6929	0.6818	0.6881	0.5594	0.9200	0.9716	14.872
		9	0.7168	0.6967	0.6860	0.6895	0.5396	0.9250	0.9888	13.592
		16	0.7203	0.7012	0.6908	0.6923	0.5321	0.9303	1.0097	11.947
	100	4	0.7101	0.6880	0.6766	0.6847	0.5500	0.9127	0.9509	14.523
		9	0.7112	0.6897	0.6784	0.6854	0.5321	0.9151	0.9568	11.969
		16	0.7129	0.6920	0.6810	0.6862	0.5258	0.9178	0.9648	9.739

Table 5: Resampling (δ known)

T \ δ / N		I			II		III		IV
		0.5	0.9	1.0	0.5	0.9	0.5	0.9	
Filtering									
20	50	0.7508	0.8067	0.8209	0.7145	0.5934	0.9485	1.1378	5.645
	100	0.7387	0.7907	0.8057	0.7003	0.5655	0.9320	1.1191	5.038
	500	0.7313	0.7794	0.7924	0.6880	0.5491	0.9230	1.0986	4.629
	1000	0.7303	0.7774	0.7912	0.6859	0.5451	0.9220	1.0949	4.547
40	50	0.7557	0.8104	0.8249	0.7182	0.6094	0.9571	1.1537	5.859
	100	0.7440	0.7933	0.8080	0.7043	0.5725	0.9455	1.1289	5.167
	500	0.7355	0.7819	0.7967	0.6924	0.5420	0.9355	1.1129	4.723
	1000	0.7342	0.7806	0.7941	0.6903	0.5384	0.9343	1.1098	4.643
100	50	0.7523	0.8044	0.8206	0.7190	0.5951	0.9588	1.1606	5.882
	100	0.7413	0.7888	0.8042	0.7050	0.5668	0.9477	1.1379	5.227
	500	0.7329	0.7778	0.7911	0.6938	0.5423	0.9382	1.1186	4.740
	1000	0.7320	0.7761	0.7987	0.6919	0.5389	0.9369	1.1166	4.653
Smoothing									
20	50	0.7350	0.7278	0.7195	0.7147	0.5838	0.9310	0.9975	5.186
	100	0.7196	0.7066	0.6964	0.6973	0.5535	0.9107	0.9653	4.514
40	50	0.7404	0.7307	0.7224	0.7183	0.6029	0.9411	0.9990	5.322
	100	0.7236	0.7069	0.6987	0.7010	0.5616	0.9233	0.9694	4.554
100	50	0.7353	0.7230	0.7152	0.7184	0.5865	0.9413	1.0034	5.416
	100	0.7208	0.7020	0.6926	0.7000	0.5539	0.9260	0.9681	4.681

Table 6: Rejection Sampling (δ known)

T \ δ / N		I			II		III		IV
		0.5	0.9	1.0	0.5	0.9	0.5	0.9	
Filtering									
20	50	0.7366	0.7870	0.8014	0.6941	0.5534	0.9319	1.1165	5.233
	100	0.7331	0.7820	0.7964	0.6904	0.5491	0.9281	1.1042	4.864
	500	0.7304	0.7777	0.7910	0.6862	0.5438	0.9224	1.0952	4.590
	1000	0.7296	0.7767	0.7900	0.6854	0.5425	0.9221	1.0940	4.542
40	50	0.7410	0.7891	0.8022	0.6968	0.5476	0.9432	1.1288	5.412
	100	0.7375	0.7851	0.7977	0.6937	0.5429	0.9381	1.1175	4.999
	500	0.7344	0.7804	0.7938	0.6901	0.5384	0.9342	1.1097	4.680
	1000	0.7337	0.7798	0.7936	0.6894	0.5372	0.9333	1.1085	4.640
100	50	0.7381	0.7853	0.7990	0.6988	0.5476	0.9467	1.1337	5.397
	100	0.7351	0.7801	0.7939	0.6948	0.5418	0.9412	1.1243	5.001
	500	0.7315	0.7756	0.7892	0.6912	0.5371	0.9367	1.1158	4.657
	1000	0.7311	0.7753	0.7882	0.6909	0.5363	0.9358	1.1151	4.618
Smoothing									
20	50	0.7137	0.6956	0.6826	0.6887	0.5374	0.9061	0.9512	4.292
	100	0.7098	0.6902	0.6798	0.6846	0.5344	0.8991	0.9407	3.927
40	50	0.7166	0.6975	0.6853	0.6895	0.5328	0.9151	0.9525	4.457
	100	0.7132	0.6927	0.6811	0.6866	0.5274	0.9090	0.9422	3.971
100	50	0.7134	0.6928	0.6806	0.6916	0.5300	0.9172	0.9544	4.414
	100	0.7095	0.6869	0.6753	0.6874	0.5257	0.9126	0.9422	3.989

Table 7: Markov Chain Monte Carlo (δ known)

T \ δ	N	c	I 0.5	I 0.9	I 1.0	II 0.5	II 0.9	III 0.5	III 0.9	IV
Smoothing										
$\boldsymbol{P_*(z\|x) = P_\alpha(z\|\alpha_{i,t-1})}$										
20	1000		0.7081	0.6883	0.6778	0.6802	0.5296	0.8945	0.9351	12.944
	5000		0.7061	0.6859	0.6751	0.6786	0.5267	0.8931	0.9318	13.069
40	1000		0.7112	0.6900	0.6787	0.6826	0.5225	0.9062	0.9360	13.009
	5000		0.7099	0.6880	0.6765	0.6810	0.5199	0.9044	0.9321	13.179
100	1000		0.7078	0.6845	0.6734	0.6831	0.5191	0.9082	0.9368	13.038
	5000		0.7060	0.6825	0.6708	0.6808	0.5166	0.9067	0.9334	13.189
$\boldsymbol{P_*(z\|x) = N(a^*_{t\|t}, c\Sigma^*_{t\|t})}$										
20	1000	4	0.7072	0.6896	0.6797	0.6794	0.5635	0.8949	0.9512	15.235
		9	0.7074	0.6903	0.6799	0.6801	0.5436	0.8975	0.9548	14.852
		16	0.7080	0.6910	0.6810	0.6802	0.5392	0.8993	0.9587	14.675
	5000	4	0.7059	0.6878	0.6774	0.6791	0.5538	0.8938	0.9371	15.090
		9	0.7060	0.6877	0.6775	0.6789	0.5405	0.8945	0.9378	14.771
		16	0.7060	0.6880	0.6777	0.6790	0.5364	0.8955	0.9406	14.614
40	1000	4	0.7108	0.6905	0.6793	0.6818	0.5434	0.9069	0.9489	15.008
		9	0.7115	0.6916	0.6807	0.6821	0.5297	0.9079	0.9509	14.649
		16	0.7122	0.6921	0.6816	0.6825	0.5259	0.9092	0.9553	14.488
	5000	4	0.7101	0.6889	0.6776	0.6807	0.5380	0.9053	0.9363	14.846
		9	0.7100	0.6891	0.6780	0.6807	0.5268	0.9056	0.9377	14.563
		16	0.7105	0.6895	0.6785	0.6809	0.5234	0.9057	0.9394	14.448
100	1000	4	0.7068	0.6845	0.6732	0.6815	0.5390	0.9087	0.9503	15.137
		9	0.7074	0.6855	0.6740	0.6818	0.5243	0.9103	0.9544	14.797
		16	0.7079	0.6863	0.6751	0.6823	0.5206	0.9120	0.9652	14.615
	5000	4	0.7059	0.6830	0.6714	0.6808	0.5307	0.9066	0.9354	14.992
		9	0.7060	0.6831	0.6716	0.6808	0.5208	0.9069	0.9365	14.718
		16	0.7062	0.6834	0.6721	0.6809	0.5188	0.9071	0.9388	14.575

Table 8: Quasi-Filter and Quasi-Smoother (δ known)

T \ δ	N	c	I 0.5	I 0.9	I 1.0	II 0.5	II 0.9	III 0.5	III 0.9	IV
Filtering										
$\boldsymbol{P_*(z\|x) = P_\alpha(z\|\alpha_{i,t-1\|t-1})}$										
20	1000		0.7310	0.7805	0.7954	0.6879	0.5463	0.9227	1.1033	4.708
	5000		0.7300	0.7786	0.7930	0.6855	0.5437	0.9217	1.1014	4.604
40	1000		0.7364	0.7842	0.7981	0.6924	0.5438	0.9346	1.1192	4.774
	5000		0.7341	0.7813	0.7954	0.6899	0.5391	0.9336	1.1180	4.668
100	1000		0.7330	0.7796	0.7942	0.6935	0.5412	0.9374	1.1271	4.785
	5000		0.7313	0.7770	0.7910	0.6911	0.5376	0.9359	1.1250	4.666
$\boldsymbol{P_*(z\|x) = N(a^*_{t\|t}, c\Sigma^*_{t\|t})}$										
		4	0.7316	0.7956	0.8193	0.6923	0.6337	0.9281	1.2819	17.087
	1000	9	0.7320	0.7974	0.8225	0.6916	0.6170	0.9276	1.2826	15.669
20		16	0.7329	0.7988	0.8244	0.6918	0.6071	0.9286	1.2922	14.838
		4	0.7310	0.7945	0.8178	0.6917	0.6336	0.9265	1.2763	16.455
	5000	9	0.7311	0.7963	0.8213	0.6910	0.6170	0.9255	1.2729	15.151
		16	0.7315	0.7971	0.8227	0.6911	0.6064	0.9256	1.2728	14.468
		4	0.7364	0.7985	0.8215	0.6961	0.6158	0.9410	1.3221	15.954
	1000	9	0.7368	0.8005	0.8253	0.6960	0.6021	0.9409	1.3249	14.959
40		16	0.7374	0.8019	0.8270	0.6961	0.5938	0.9416	1.3308	14.383
		4	0.7353	0.7971	0.8198	0.6953	0.6158	0.9395	1.3167	15.606
	5000	9	0.7354	0.7989	0.8230	0.6948	0.6019	0.9384	1.3173	14.716
		16	0.7354	0.7994	0.8240	0.6948	0.5932	0.9385	1.3220	14.232
		4	0.7332	0.7934	0.8157	0.6975	0.6106	0.9435	1.3579	16.503
	1000	9	0.7339	0.7956	0.8195	0.6973	0.5971	0.9433	1.3563	15.210
100		16	0.7345	0.7971	0.8216	0.6976	0.5891	0.9434	1.3640	14.513
		4	0.7326	0.7927	0.8150	0.6970	0.6103	0.9423	1.3512	16.032
	5000	9	0.7329	0.7944	0.8182	0.6965	0.5965	0.9411	1.3487	14.885
		16	0.7331	0.7952	0.8195	0.6964	0.5884	0.9407	1.3521	14.317
Smoothing										
$\boldsymbol{P_*(z\|x) = P_\alpha(z\|\alpha_{i,t-1\|t-1})}$										
20	1000		0.7083	0.6934	0.6838	0.6817	0.5346	0.8978	0.9592	4.329
	5000		0.7067	0.6912	0.6814	0.6791	0.5299	0.8964	0.9567	4.114
40	1000		0.7125	0.6959	0.6848	0.6853	0.5281	0.9106	0.9657	4.340
	5000		0.7107	0.6927	0.6823	0.6817	0.5241	0.9088	0.9633	4.128
100	1000		0.7083	0.6900	0.6796	0.6844	0.5258	0.9120	0.9678	4.356
	5000		0.7070	0.6873	0.6767	0.6822	0.5213	0.9110	0.9656	4.065
$\boldsymbol{P_*(z\|x) = N(a^*_{t\|T}, c\Sigma^*_{t\|T})}$										
		4	0.7083	0.7072	0.7068	0.6863	0.6275	0.9034	1.1922	15.229
	1000	9	0.7092	0.7103	0.7109	0.6858	0.6085	0.9019	1.1854	14.816
20		16	0.7098	0.7114	0.7131	0.6857	0.5973	0.9035	1.1933	14.577
		4	0.7076	0.7065	0.7060	0.6857	0.6273	0.9026	1.1876	15.063
	5000	9	0.7077	0.7086	0.7093	0.6845	0.6083	0.9003	1.1775	14.715
		16	0.7079	0.7095	0.7108	0.6845	0.5963	0.9000	1.1761	14.522
		4	0.7122	0.7084	0.7066	0.6889	0.6087	0.9168	1.2274	14.950
	1000	9	0.7131	0.7104	0.7100	0.6882	0.5925	0.9151	1.2219	14.574
40		16	0.7132	0.7122	0.7125	0.6883	0.5826	0.9155	1.2262	14.363
		4	0.7115	0.7072	0.7053	0.6882	0.6087	0.9154	1.2214	14.807
	5000	9	0.7117	0.7091	0.7085	0.6871	0.5923	0.9129	1.2144	14.498
		16	0.7119	0.7098	0.7097	0.6870	0.5821	0.9123	1.2178	14.320
		4	0.7086	0.7025	0.7001	0.6899	0.6033	0.9193	1.2604	15.079
	1000	9	0.7094	0.7050	0.7041	0.6892	0.5876	0.9176	1.2512	14.707
100		16	0.7101	0.7068	0.7064	0.6893	0.5783	0.9173	1.2558	14.484
		4	0.7077	0.7015	0.6991	0.6892	0.6029	0.9178	1.2540	14.941
	5000	9	0.7079	0.7035	0.7023	0.6881	0.5868	0.9153	1.2434	14.623
		16	0.7082	0.7044	0.7037	0.6878	0.5771	0.9146	1.2443	14.429

Table 9: Comparison of Computation Times ($T = 100$)

	N	Prediction	Filtering	Smoothing
Numerical Integration		LkN^{k+1}	$2TkN^{k+1}$	TkN^{k+1}
	50		(0.37)	(0.65)
	100		(1.43)	(2.53)
Importance Sampling		LN^2	$TN(N+1)$	TN^2
	50		(0.28)	(1.03)
	100		(1.03)	(4.02)
Resampling		LN	$TN(N+1)$	$TN^2(N+1)$
	50		(0.05)	(12.2)
	100		(0.10)	(96.9)
Rejection Sampling		LN	$TN(2N+A)$	$TN(N^2+N+A)$
(*)		LN	$TN(N+1+A)$	$TN(2N+A)$
	50		(0.24)	(1.95)
	100		(0.52)	(6.58)
Markov Chain		LN	$NT(T-1)/2$	TN
Monte Carlo	1000		—	(0.98)
	5000		—	(4.88)
Quasi-Filter and		LN	TN	TN
Quasi-Smoother	1000		(0.91)	(1.91)
	5000		(4.54)	(9.51)

Simulations I – III, respectively. c represents range of the nodes in Table 3, variance of the importance density in Table 4, and variance of the proposal density in Tables 7 and 8. In Tables 7 and 8, M is taken as 20% of N and moreover two types of proposal densities are examined: one is based on the transition equation (2) and another is use of the extended Kalman filter and smoother estimates. In Tables 6, the transition equation is utilized for the proposal density.

Since Simulation I represents the linear and normal case, the RMSE's in Simulation I of Table 2 give us the minimum values, compared with those in Tables 3 – 8. However, for Simulations II – IV, it is easily expected that Table 2 shows the worst RMSE's. Note in Simulations II and III of Table 2 that the filtering estimates are exactly equivalent to the smoothing estimates (i.e., $a^*_{t|s} = 0,\ s = t, T$) because of the functional form of the underlying state-space model (in addition, the initial value of $a_{0|0} = 0$ causes this situation for Simulation III).

The results obtained from the numerical integration procedure are in Table 3, where the trapezoid rule is taken for evaluation of numerical integration (in Section 3.1 each integration is evaluated by the rectangle rule for simplicity of discussion). For Simulation I, the case $\delta = 0.5$ of Simulation II and Simulation III, the RMSE's are unchanged for $N = 50, 100$ and $c = 9, 16, 25$, which implies that the obtained RMSE's are very close to the true RMSE's. However, for the case $\delta = 0.9$ of Simulation II and Simulation IV, the RMSE's are small when N and c increase, i.e., we should take more nodes and larger range of the nodes to obtain the true RMSE's.

Table 4 shows the RMSE's obtained from the importance sampling procedure in Section 3.2. In the nonlinear cases of Simulations II – IV, the importance sampling procedure performs much better than the extended Kalman filter and smoother but worse than the numerical integration procedure. To obtain the same precision as the numerical integration approach, more random draws are necessary.

The results of the resampling procedure are in Table 5. Smoothing requires an extremely large computational burden.[19] Therefore, the resampling procedure does not have as small RMSE's as the numerical integration approach. However, for filtering in Simulation I, the case $\delta = 0.5$ of Simulation II and Simulation III, the resampling procedure is very close to the numerical integration approach as N is large. Especially, for the case $\delta = 0.9$ of Simulation II and Simulation IV, the filtering results in Table 5 are much better than those in Tables 2 – 4.

In Table 6, similarly, smoothing takes an extremely long time computationally, compared with filtering. The approximations based on (48) are taken for less computational burden and the Metropolis-Hastings algorithm are utilized in parallel to avoid repeating procedures (ii) and (iii) forever. Accordingly, when the rejection sampling does not work, a random draw (i.e., $\alpha_{j,t|t}$ or $\alpha_{m,t|T}$) is generated by the Metropolis-Hastings algorithm. We sometimes have the case where it is not clear whether the supremum exists. Since $P_*(z) = P_\alpha(z|\alpha_{i,t-1|t-1})$ is taken in this Monte Carlo study, the denominators of the acceptance probabilities (42) and (47) are given by $\sup_z P_y(y_t|z)$ and $\sup_z P_y(y_t|z)P_\alpha(\alpha_{j,t+1|T}|z)$, respectively. In the case where we cannot obtain the explicit solution, the supremum is computed by the Newton-Raphson optimization procedure, which implies that the solution is possibly the local supremum. In Simulations II and IV, the acceptance probability is numerically evaluated to obtain smoothing. For all simulation studies, however, the rejection sampling procedure shows a good performance even when N is small. Especially, the resampling procedure requires about two times more random draws to obtain the same RMSE as the rejection sampling procedure.

The Markov chain Monte Carlo approach is taken in Table 7. All the values indicate the RMSE's obtained from smoothing. Two types of the proposal densities are examined. Both proposal densities yield the similar results, but the proposal density of the transition equation is slightly better than that of the extended Kalman smoother. The estimator performs well in Simulations I – III but not in Simulation IV. Convergence of the Gibbs sampler is very slow in the case of Simulation IV.

In Table 8, both quasi-filter and quasi-smoother perform better when the proposal density is based on the transition equation. However, the proposal density based on the extended Kalman filter and smoother does not work in spite of c. Accordingly, from the results in Tables 7 and 8, the transition equation should be utilized for the proposal density.

The resampling procedure is the easiest estimator in the sense of programming but it takes an extraordinarily long time for smoothing. The Markov chain Monte Carlo procedure has the least computational burden although it sometimes shows a poor performance. For all the simulation studies, the most accurate estimator is the rejection sampling approach but we

[19]The case $N = 1000$ is feasible for filtering but even the case $N = 100$ takes a lot of time for smoothing.

Table 10: Estimation of Unknown Parameter ($T = 100$)

		I			II		III	
	δ	0.5	0.9	1.0	0.5	0.9	0.5	0.9
Extended Kalman Filter	AVE	0.472	0.878	0.981	0.301	0.652	—	—
	RMSE	0.144	0.065	0.040	0.285	0.313	—	—
	25%	0.400	0.850	0.970	0.140	0.530	—	—
	50%	0.490	0.890	0.990	0.310	0.660	—	—
	75%	0.570	0.920	1.000	0.450	0.790	—	—
Numerical Integration	AVE	0.472	0.878	0.981	0.444	0.850	0.440	0.878
	RMSE	0.144	0.065	0.040	0.299	0.168	0.218	0.071
	25%	0.400	0.850	0.970	0.170	0.810	0.330	0.850
	50%	0.490	0.890	0.990	0.490	0.890	0.490	0.890
	75%	0.570	0.920	1.000	0.680	0.950	0.600	0.920
Importance Sampling	AVE	0.470	0.877	0.980	0.447	0.845	0.434	0.887
	RMSE	0.146	0.065	0.040	0.304	0.188	0.221	0.078
	25%	0.400	0.850	0.970	0.160	0.810	0.320	0.850
	50%	0.490	0.890	0.990	0.490	0.900	0.490	0.900
	75%	0.570	0.920	1.000	0.695	0.950	0.590	0.940
Resampling	AVE	0.505	0.905	1.002	0.505	0.907	0.505	0.905
	RMSE	0.013	0.013	0.011	0.013	0.018	0.013	0.013
	25%	0.500	0.900	1.000	0.500	0.900	0.500	0.900
	50%	0.500	0.910	1.000	0.510	0.910	0.510	0.910
	75%	0.510	0.910	1.010	0.510	0.910	0.510	0.910
Rejection Sampling	AVE	0.505	0.905	1.003	0.506	0.906	0.505	0.904
	RMSE	0.013	0.014	0.011	0.013	0.013	0.013	0.013
	25%	0.500	0.900	1.000	0.500	0.900	0.500	0.900
	50%	0.510	0.910	1.000	0.510	0.910	0.510	0.900
	75%	0.510	0.910	1.010	0.510	0.910	0.510	0.910
Markov Chain Monte Carlo	AVE	0.539	0.915	1.003	0.915	0.983	0.515	0.909
	RMSE	0.059	0.024	0.012	0.445	0.086	0.027	0.016
	25%	0.510	0.900	1.000	0.990	0.990	0.500	0.900
	50%	0.530	0.910	1.000	0.990	0.990	0.510	0.910
	75%	0.560	0.920	1.010	0.990	0.990	0.520	0.920
Quasi-Filter and Quasi-Smoother	AVE	0.482	0.903	0.997	0.494	0.907	0.497	0.904
	RMSE	0.106	0.021	0.015	0.138	0.057	0.073	0.016
	25%	0.470	0.890	0.990	0.480	0.890	0.490	0.900
	50%	0.500	0.900	1.000	0.500	0.910	0.500	0.900
	75%	0.530	0.910	1.010	0.530	0.930	0.520	0.910

should keep in mind that existence of the supremum is required for rejection sampling. The quasi-filter and quasi-smoother gives us the relatively good estimator, which can be applied to any case in the sense of no restriction on the supremum such as rejection sampling.

Next, for each estimator, the order of computation and the computation times are compared in Table 9. The first line in each cell represents the order of computation, which is proportional to computation time. The values in the parentheses indicate the CPU times (the averages from 100 simulation runs, i.e., $G = 100$), which are given in seconds. The case $\delta = 1.0$ in Simulation I are used for comparison of computation times, where $T = 100$ is taken. Computations were performed by 300MHz Pentium II Processor and WATCOM Fortran 77/32 Compiler (Version 10.6), using double precision. (*) under Rejection Sampling denotes the reduction method of computational burden using (48). Computation times in Rejection Sampling were implemented by (*). Except for Markov Chain Monte Carlo, computations of filtering are necessary for those of smoothing. A denotes number of rejection in Rejection Sampling. The proposal density is based on the transition equation (1) for Rejection Sampling, Markov Chain Monte Carlo and Quasi-Filter and Quasi-Smoother. We choose $c = 16$ for Numerical Integration and $c = 4$ for Importance Sampling.

In Resampling, as expected, smoothing takes an extremely long time computationally, while filtering indicates much less computation. Computation times of Rejection Sampling depend on the functional form of the state-space model and the distribution of the error terms, because the acceptance probability is based on the density function $P_y(y_t|\alpha_t)$ or $P_y(y_t|\alpha_t)P_\alpha(\alpha_{t+1}|\alpha_t)$, while those of the other estimators are not influenced by nonlinear equations and non-Gaussian errors.[20]

In Tables 10, comparison between the true parameter and the estimate of δ is shown for each procedure. Note that δ in the table indicates the true value. Given observed data Y_T, the parameter δ in Simulations I – III is estimated using the appropriate likelihood function. AVE, RMSE, 25%, 50% and 75% represent the arithmetic average, the root mean square error, the 0.25th, 0.50th and 0.75th quantiles from 1000 estimates of δ (i.e., $G = 1000$). We take $N = 100$ for Numerical Integration, Importance Sampling, Resampling and Rejection Sampling and $N = 1000$ for Markov Chain Monte Carlo and Quasi-Filter and Quasi-Smoother. Furthermore, we assume $c = 16$ for Numerical Integration and $c = 4$ for Importance Sampling. As in Table 9, the proposal density is based on the transition equation (1) for Rejection Sampling, Markov Chain Monte Carlo and Quasi-Filter and Quasi-Smoother. 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. — in Simulation III of Extended Kalman Filter indicates that the maximum likelihood estimation cannot be performed, because the innovation form of the likelihood function (see equation (50) for the likelihood function) does not depend on the unknown parameter δ under the assumption of $\mathrm{E}(\alpha_0) = a_{0|0} = 0$.

[20] As for Rejection Sampling, Simulation IV computationally takes much more time than Simulation I. For Simulation IV in the case of $N = 50$ and $T = 100$, filtering and smoothing take 1.19 and 4.15 seconds, respectively.

Resampling, Rejection Sampling and Quasi-Filter and Quasi-Smoother perform better for all the simulation studies I – III. That is, AVE is very close to the true parameter value and RMSE is very small. In Simulation II, Extended Kalman Filter shows the worst estimator and moreover Markov Chain Monte Carlo does not perform good, compared with the other estimators. For all the simulation studies, Numerical Integration is similar to Importance Sampling, while Resampling and Rejection Sampling are close to Quasi-Filter and Quasi-Smoother.

Finally, we can conclude from the simulation studies that Quasi-Filter and Quasi-Smoother which utilize the transition equation for the proposal density might be recommended at the present time (i.e., see Tanizaki (1998)), because precision of the state estimates is quite good (Table 8), computational burden is very small (Table 9) and maximum likelihood estimation works well (Table 10). However, we should keep in mind as follows. The estimator which shows the best performance is Rejection Sampling. Resampling is inferior to Rejection Sampling from computational time and precision of the state estimates. On the other hand, Resampling can be applied to any nonlinear and non-Gaussian cases while Rejection Sampling does not work under some conditions. In the future, therefore, it might be expected that Resampling would be the best estimator because CPU speed of computer becomes faster and faster.

5 Summary and Concluding Remarks

In this paper, several nonlinear and non-Gaussian filters and smoothers have been introduced and compared through Monte Carlo studies. Each nonlinear and non-Gaussian estimator has both advantages and disadvantages, which are summarized as follows.

The numerical integration procedure proposed in Kitagawa (1987) and Kramer and Sorenson (1988) has the problems: (i) location of nodes has to be set by a researcher (unless the range of the nodes cover the density $P(\alpha_r|Y_s)$, the density approximation becomes poor), (ii) it is possible that computational errors accumulate because density functions are evaluated at each time, and (iii) computational burden increases more than proportionally as the dimension of the state variable is high. However, the advantage of the estimator is that we can obtain precise estimates of the state variable when the nodes are correctly chosen.

The problems of the Monte-Carlo integration procedure with importance sampling developed by Tanizaki (1993, 1996, 1999b), Tanizaki and Mariano (1994) and Mariano and Tanizaki (1995) are: (i) the importance density has to be appropriately chosen by a researcher (if the random draws generated from the importance density are away from the density $P(\alpha_r|Y_s)$, the approximation of the weight function becomes poor), and (ii) it might be possible that computational errors accumulate because weight functions are evaluated at each time. However, the merit of the importance sampling procedure is that computational burden does not increases too much even in the high dimensional cases of the state vector.

The resampling procedure by Gordon, Salmond, and Smith (1993), Kitagawa (1996, 1998) and Kitagawa and Gersch (1996) has the disadvantage that it requires heavy computation,

especially for smoothing. However, this problem will be improved in the future as computer progresses.

The disadvantages of the rejection sampling procedure (Mariano and Tanizaki (2000), Tanizaki and Mariano (1998) and Tanizaki (1999a)) are: (i) the proposal density has to be appropriately chosen by a researcher (use of the transition equation might be recommended, but not necessarily), (ii) it takes a long time computationally when the acceptance probability is small (i.e., we cannot predict how long the computer program will run), and (iii) sometimes the supremum of the ratio of the target density and the proposal density does not exists (we cannot apply rejection sampling in such a case). If rejection sampling works, i.e., if the supremum exists, the rejection sampling procedure shows the best performance of all the procedures introduced in this paper, because rejection sampling is the random number generation method which generate an exact random draw from any distribution function.

The Markov chain Monte Carlo procedure proposed by Geweke and Tanizaki (1999a, 1999b) has the following problems: (i) the proposal density has to be appropriately chosen by a researcher (it might be plausible to take the transition equation for the proposal density), and (ii) convergence is very slow because the Gibbs sampler and the Metropolis-Hastings are simultaneously used (remember that the random draw generated by the Markov chain Monte Carlo method is correlated with the next one). To obtain the smoothing estimates from the numerical integration approach, importance sampling procedure, the resampling procedure and the rejection sampling approach, we need at least $N \times T$ data storage, because smoothing is implemented after prediction and filtering. That is, we need store both prediction and filtering random draws before smoothing. However, the Markov chain Monte Carlo approach uses only T data storage to obtain the smoothing estimates.

The quasi-filter and quasi-smoother using the Metropolis-Hastings (Tanizaki (1998)) have the disadvantages: (i) the proposal density has to be appropriately chosen by a researcher (it is recommended to use the transition equation for the proposal density), and (ii) the obtained estimates do not give us the exact true values, because the filtering and smoothing densities are approximated. This procedure also uses only T data storage. Although the filtering and smoothing estimates are different from the true values, it is too small to ignore the difference between the estimates and the true values, which results come from the simulation studies in Section 4.

Thus, the rejection sampling approach might be the best estimator under the two conditions: (i) the supremum exists and (ii) the acceptance probability is not too small. In the sense of no *ad hoc* assumptions such as choice of the nodes, choice of the importance density and choice of the proposal density, the resampling procedure in Section 3.3 might be taken as the estimator which gives us an optimal solution for sufficiently large N. However, the resampling procedure takes an extremely long time for smoothing, although it does not take too much time for filtering. For the Markov chain Monte Carlo method, we have the case where the random draws are not correctly generated from the distribution function we want to sample, depending on the underlying nonlinear equations and non-Gaussian error terms. Accordingly, at the present time, the second best estimator might be the quasi-filter and quasi-smoother from computational point of view and getting precise estimates of the state variable.

Recently, the nonlinear non-Gaussian filters and smoothers which are much less computational than the existing ones are proposed by Tanizaki (2000), where the sampling techniques such as rejection sampling, resampling and the Metropolis-Hastings algorithm are utilized. The conventional density-based nonlinear algorithms require the random draws generated from 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 Tanizaki (2000) 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, 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.

Appendix A: Linear and Normal System

State-Space Model: Consider the case where the system is linear and normal, i.e.,

$$\begin{aligned}
&\text{(Measurement equation)} \quad y_t = Z_t\alpha_t + d_t + S_t\epsilon_t,\\
&\text{(Transition equation)} \quad \alpha_t = T_t\alpha_{t-1} + c_t + R_t\eta_t,\\
&\qquad\qquad \begin{pmatrix}\epsilon_t\\ \eta_t\end{pmatrix} \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}H_t & 0\\ 0 & Q_t\end{pmatrix}\right),
\end{aligned}$$

where Z_t, d_t, S_t, T_t, c_t, R_t, H_t and Q_t are assumed to be known for all time $t = 1, 2, \cdots, T$. Define conditional mean and variance as $a_{r|s} \equiv \mathrm{E}(\alpha_r|Y_s)$ and $\Sigma_{r|s} \equiv \mathrm{Var}(\alpha_r|Y_s)$ for $(r, s) = (t + L, t), (t, t), (t, T)$. Under the above setup, optimal prediction, filtering and smoothing are represented as the standard linear recursive algorithms, which are easily derived from the first- and second-moments of density functions (13) – (18). See, for example, Tanizaki (1996).

Prediction: From the density-based L-step ahead prediction algorithm (13), the following prediction algorithm can be obtained:

$$\begin{aligned}
&a_{t+L|t} = T_{t+L}a_{t+L-1|t} + c_{t+L},\\
&\Sigma_{t+L|t} = T_{t+L}\Sigma_{t+L-1|t}T'_{t+L} + R_{t+L}Q_{t+L}R'_{t+L},
\end{aligned}$$

for $L = 1, 2, \cdots$. Given filtering mean and variance (i.e., $a_{t|t}$ and $\Sigma_{t|t}$), prediction mean and variance (i.e., $a_{t+L|t}$ and $\Sigma_{t+L|t}$) are obtained recursively.

Filtering: The density-based filtering algorithm given by equations (14) and (15) reduces to the following standard linear recursive algorithm:

$$a_{t|t-1} = T_t a_{t-1|t-1} + c_t,$$

$$
\begin{aligned}
&\Sigma_{t|t-1} = T_t \Sigma_{t-1|t-1} T_t' + R_t Q_t R_t', \\
&y_{t|t-1} = Z_t a_{t|t-1} + d_t, \\
&F_{t|t-1} = Z_t \Sigma_{t|t-1} Z_t' + S_t H_t S_t', \\
&K_t = \Sigma_{t|t-1} Z_t' F_{t|t-1}^{-1}, \\
&a_{t|t} = a_{t|t-1} + K_t (y_t - y_{t|t-1}), \\
&\Sigma_{t|t} = \Sigma_{t|t-1} - K_t F_{t|t-1} K_t',
\end{aligned}
$$

for $t = 1, 2, \cdots, T$. Given the initial values $a_{0|0}$ and $\Sigma_{0|0}$, the filtering mean and variance at time t (i.e., $a_{t|t}$ and $\Sigma_{t|t}$) are recursively computed. See Anderson and Moore (1979), Gelb (1974), Jazwinski (1970) and Tanizaki (1996) for the Kalman filter algorithm.

Smoothing: The first- and the second-moments of the smoothing density (16) give us the following backward recursive algorithm:

$$
\begin{aligned}
&C_t = \Sigma_{t|t} T_{t+1}' \Sigma_{t+1|t}^{-1}, \\
&a_{t|T} = a_{t|t} + C_t (a_{t+1|T} - a_{t+1|t}), \\
&\Sigma_{t|T} = \Sigma_{t|t} + C_t (\Sigma_{t+1|T} - \Sigma_{t+1|t}) C_t',
\end{aligned}
$$

for $t = T-1, T-2, \cdots, 1$. Given $a_{t|t}$, $\Sigma_{t|t}$, $a_{t+1|t}$ and $\Sigma_{t+1|t}$, smoothing mean and variance at time t (i.e., $a_{t|T}$ and $\Sigma_{t|T}$) is obtained recursively.

Likelihood Function: When Z_t, d_t, S_t, T_t, c_t, R_t, H_t and Q_t depends on an unknown parameter, the following log of the likelihood function is maximized with respect to the parameter:

$$
\begin{aligned}
\log P(Y_T) =& -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T} \log |F_{t|t-1}| \\
& -\frac{1}{2}\sum_{t=1}^{T} (y_t - y_{t|t-1})' F_{t|t-1}^{-1} (y_t - y_{t|t-1}),
\end{aligned} \tag{50}
$$

which is also obtained from equation (18). Note that the conditional distribution of y_t given Y_{t-1} is represented as $y_t | Y_{t-1} \sim N(y_{t|t-1}, F_{t|t-1})$, where both $y_{t|t-1}$ and $F_{t|t-1}$ are obtained from the above standard filtering algorithm.

Appendix B: Sampling Methods

Monte Carlo Integration with Importance Sampling: When we want to evaluate the expectation of a function $g(\cdot)$, the problem is how to evaluate integration. There are two integration methods: one is numerical integration and another is Monte Carlo integration. To perform Monte Carlo integration, we usually take another appropriate distribution function $P_*(x)$, called the importance density, which is chosen by a researcher. Let x_i, $i = 1, 2, \cdots, N$,

be the random draws of x generated from $P_x(x)$. Define $\omega(x)$ as $\omega(x) \equiv P_x(x)/P_*(x)$. Then, in order to approximate integration, importance sampling is performed as follows:

$$\begin{aligned}\mathrm{E}(g(x)) &= \int g(x)P_x(x)\mathrm{d}x = \int g(x)\omega(x)P_*(x)\mathrm{d}x \\ &\approx \frac{1}{N}\sum_{i=1}^{N} g(x_i)\omega(x_i) \equiv \overline{g}_N,\end{aligned}$$

which is called the Monte-Carlo integration method with importance sampling.

Because x_i is a random variable from $P_*(x)$ while x is a random variable from $P_x(x)$, $\mathrm{E}(g(x_i)\omega(x_i))$ implies taking the expectation with respect to x_i but $\mathrm{E}(g(x))$ is the expectation taken with respect to x. Now, define μ and Σ as:

$$\begin{aligned}&\mu = \mathrm{E}(g(x_i)\omega(x_i)) = \mathrm{E}(g(x)), \\ &\Sigma = \mathrm{Var}(g(x_i)\omega(x_i)) = \mathrm{E}(g(x_i)\omega(x_i))^2 - \mu^2 = \mathrm{E}\Big((g(x))^2\omega(x)\Big) - \mu^2.\end{aligned}$$

$g(x_i)\omega(x_i)$ are mutually independent for $i = 1, 2, \cdots, N$. Therefore, by the central limit theorem, we can easily show:

$$\sqrt{N}(\overline{g}_N - \mu) \longrightarrow N(0, \Sigma).$$

Moreover, let us define $\overline{\Sigma}_N$ as:

$$\overline{\Sigma}_N = \frac{1}{N}\sum_{i=1}^{N}(g(x_i) - \overline{g}_N)(g(x_i) - \overline{g}_N)'\omega(x_i).$$

Then, we have the following:

$$\overline{\Sigma}_N \longrightarrow \Sigma.$$

Thus, $\overline{g}_N$ gives us an asymptotically unbiased estimator as N goes to infinity. However, it is shown from the above results that convergence is quite slow as $\sqrt{N}$. See, for example, Geweke (1988, 1989a, 1989b), Koop (1994) and Shao (1989).

Rejection Sampling: Rejection sampling is the method which generates random draws from any distribution function under some conditions. In the case where it is not easy to generate a random number from $P_x(x)$, suppose that we want to generate a random draw from $P_x(x)$, called the target density. In such a case, we take another distribution function $P_*(x)$, called the proposal density, which is appropriately chosen by a researcher.

Denote the acceptance probability by $\omega(x) = P_x(x)/aP_*(x)$, where a is defined as $a \equiv \sup_x P_x(x)/P_*(x)$ and the assumption of $a < \infty$ is required for rejection sampling. Under the setup, rejection sampling is implemented as: (i) generate a random draw of x (say, x_0) from $P_*(x)$ and (ii) accept it with probability $\omega(x_0)$. The accepted random draw is taken as a random draw of x generated from $P_x(x)$.

In the case where both $P_x(x)$ and $P_*(x)$ are normally distributed as $N(\mu, \sigma^2)$ and $N(\mu_*, \sigma_*^2)$, it is easily shown that $\sigma_*^2 > \sigma^2$ is required for the condition $a < \infty$, which implies that $P_*(x)$ has to be distributed with larger variance than $P_x(x)$.

Note that $P_x(x)$ is not necessarily a probability density function, i.e., it is possibly a kernel of the target density function, which implies that the target density is proportional to $P_x(x)$, i.e., $cP_x(x)$. Since the supremum a in $\omega(x)$ also includes $P_x(x)$, the constant c is canceled out from the acceptance probability $\omega(x)$.

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 the acceptance probability $\omega(\cdot)$ is close to zero. See, for example, Knuth (1981), Boswell, Gore, Patil and Taillie (1993), O'Hagan (1994) and Geweke (1996) for rejection sampling.

Gibbs Sampling: Geman and Geman (1984), Tanner and Wong (1987), Gelfand and Smith (1990), Gelfand, Hills, Racine-Poon and Smith (1990) and so on developed the Gibbs sampling theory, which is concisely described as follows (also see Geweke (1996, 1997)).

Consider two random variables x and y. Let $P_{x|y}(x|y)$, $P_{y|x}(y|x)$ and $P_{xy}(x, y)$ be the conditional density of x given y, the conditional density of y given x and the joint density of x and y, which are assumed to be known. Pick up an arbitrary initial value for x (i.e., x_0). Given x_{i-1}, generate a random number of y (i.e., y_i) from the density $P_{y|x}(y|x_{i-1})$. Again, given y_i, generate a random number of x (i.e., x_i) from the density $P_{x|y}(x|y_i)$. Thus, we can generate y_i from $P_{y|x}(y|x_{i-1})$ and x_i from $P_{x|y}(x|y_i)$ for $i = 1, 2, \cdots, N$. From the convergence theory of the Gibbs sampler, as N goes to infinity, we can regard x_N and y_N as random draws from $P_{xy}(x, y)$.

The basic result of the Gibbs sampler is:

$$\frac{1}{N-M}\sum_{i=M+1}^{N} g(x_i, y_i) \longrightarrow \mathrm{E}(g(x, y)) = \iint g(x, y)P_{xy}(x, y)\mathrm{d}x\mathrm{d}y,$$

as $N \longrightarrow \infty$, where $g(\cdot, \cdot)$ is a function. We may take $M = 0$ but usually 10 – 20% of N is taken for M. That is, the first M random draws are discarded from consideration.

Finally, note as follows. it is known that convergence is very slow if correlation between x and y is very high. See Chib and Greenberg (1995).

Metropolis-Hastings Algorithm: Smith and Roberts (1993), Tierney (1994), Chib and Greenberg (1995, 1996) and Geweke (1996) discussed the Metropolis-Hastings algorithm, which is also the random number generation method such that we can generate random draws from any density function.

Suppose that we want to generate a random draw of x from $P_x(x)$, which is called the target density function. When it is hard to generate random draws from the target density $P_x(\cdot)$, we can apply the Metropolis-Hastings algorithm to random number generation. The Metropolis-Hastings algorithm utilizes another appropriate distribution function $P_*(z|x)$, which is called the proposal density.

In order to perform the Metropolis-Hastings algorithm, first let us define the acceptance probability $\omega(x, z)$ as:

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

Using the acceptance probability $\omega(x, z)$ defined above, the Metropolis-Hastings algorithm can be implemented as follows: (i) take an initial value of x, which is denoted by 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 $u \leq \omega(x_{i-1}, z)$ and set $x_i = x_{i-1}$ otherwise, and (iv) repeat (ii) and (iii) for $i = 1, 2, \cdots, N$. Then, x_N is taken as a random draw from $P_x(x)$ for sufficiently large N. The basic result of the Metropolis-Hastings algorithm is as follows:

$$\frac{1}{N-M}\sum_{i=M+1}^{N} g(x_i) \longrightarrow \mathrm{E}(g(x)) = \int g(x)P_x(x)\mathrm{d}x,$$

as $N \longrightarrow \infty$, where $g(\cdot)$ is a function.

For choice of the proposal density $P_*(z|x)$, note as follows. The proposal density $P_*(z|x)$ should not have too large variance and too small variance (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)). Moreover, we should take the proposal density such that we can easily and quickly generate random draws. For a functional form of the proposal density, 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 $P_x(x)$ is not necessarily a probability density function, i.e., it is possibly a kernel of the target density function (remember that we need the ratio of the target and proposal densities to derive $\omega(x, z)$). It is also possible to apply the Metropolis-Hastings algorithm to generate random numbers from $P_{x|y}(x|y)$ and $P_{y|x}(y|x)$ in the Gibbs sampler, when it is difficult to generate a random draw of x from $P_{x|y}(x|y)$ and/or a random draw of y from $P_{y|x}(y|x)$ (see Chib and Greenberg (1995) for the Metropolis-Hastings algorithm within Gibbs sampling).

Appendix C: Recursive versus Non-Recursive Algorithms

In Sections 2.2.1 and 2.2.2, we introduce two density-based algorithms on prediction, filtering and smoothing The conventional recursive algorithms are represented by equations (13) – (16) of Section 2.2.1. Equations (22), (24) and (26) of Section 2.2.2 indicate the non-recursive algorithms. In this appendix, it is shown that both algorithms are equivalent (i.e., we can derive equation (13) from equation (22), equations (14) and (15) from equation (24) and equation (16) from equation (26), respectively).

Prediction: Equation (22) is rewritten as:

$$\begin{aligned}
P(\alpha_{t+L}|Y_t) &= \int P(A_{t+L}|Y_t)\mathrm{d}A_{t+L-1} \\
&= \int P_\alpha(\alpha_{t+L}|\alpha_{t+L-1})P(A_{t+L-1}|Y_t)\mathrm{d}A_{t+L-1} \\
&= \iint P_\alpha(\alpha_{t+L}|\alpha_{t+L-1})P(A_{t+L-1}|Y_t)\mathrm{d}A_{t+L-2}\mathrm{d}\alpha_{t+L-1} \\
&= \int P_\alpha(\alpha_{t+L}|\alpha_{t+L-1})\left(\int P(A_{t+L-1}|Y_t)\mathrm{d}A_{t+L-2}\right)\mathrm{d}\alpha_{t+L-1} \\
&= \int P_\alpha(\alpha_{t+L}|\alpha_{t+L-1})P(\alpha_{t+L-1}|Y_t)\mathrm{d}\alpha_{t+L-1}, \qquad (51)
\end{aligned}$$

where the second line in equation (51) uses the following two equations: $P(A_{t+L}, Y_t) = P_\alpha(A_{t+L})P_y(Y_t|A_t) = P_\alpha(\alpha_{t+L}|\alpha_{t+L-1})P_\alpha(A_{t+L-1})P_y(Y_t|A_t)$ and $P(A_{t+L}|Y_t) = P_\alpha(A_{t+L})$ $P_y(Y_t|A_t)/P(Y_t)$ while the fifth line utilizes the first equality. Thus, it can be easily shown that equation (13) is equivalent to equation (22).

Filtering: Equation (24) is transformed as:

$$\begin{aligned}
P(\alpha_t|Y_t) &= \frac{\int P(A_t, Y_t)\mathrm{d}A_{t-1}}{\int P(A_t, Y_t)\mathrm{d}A_t} \\
&= \frac{\int P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})P(A_{t-1}, Y_{t-1})\mathrm{d}A_{t-1}}{\iint P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})P(A_{t-1}, Y_{t-1})\mathrm{d}A_{t-1}\mathrm{d}\alpha_t} \\
&= \frac{P_y(y_t|\alpha_t)\left(\int P_\alpha(\alpha_t|\alpha_{t-1})P(A_{t-1}|Y_{t-1})\mathrm{d}A_{t-1}\right)}{\int P_y(y_t|\alpha_t)\left(\int P_\alpha(\alpha_t|\alpha_{t-1})P(A_{t-1}|Y_{t-1})\mathrm{d}A_{t-1}\right)\mathrm{d}\alpha_t} \\
&= \frac{P_y(y_t|\alpha_t)P(\alpha_t|Y_{t-1})}{\int P_y(y_t|\alpha_t)P(\alpha_t|Y_{t-1})\mathrm{d}\alpha_t} \qquad (52)
\end{aligned}$$

Note that $P(A_t, Y_t) = P_y(y_t|\alpha_t)P_\alpha(\alpha_t|\alpha_{t-1})P(A_{t-1}, Y_{t-1})$ in the second line of equation (52). From the second equality of equation (51), $P(\alpha_t|Y_{t-1}) = \int P_\alpha(\alpha_t|\alpha_{t-1})P(A_{t-1}|Y_{t-1})\mathrm{d}A_{t-1}$ is substituted into the fourth line. Thus, equation (15) is derived from equation (24).

Smoothing: Let us define $Y_t^* = \{y_t, y_{t+1}, \cdots, y_T\}$. Suppose that the joint density function of A_{t+1}^* and Y_{t+1}^* is given by:

$$P(A_{t+1}^*, Y_{t+1}^*) = \prod_{s=t+2}^{T} P_\alpha(\alpha_s|\alpha_{s-1}) \prod_{s=t+1}^{T} P_y(y_s|\alpha_s),$$

which implies that the joint density of A_T and Y_T is represented as follows:

$$P(A_T, Y_T) = P(A_t, Y_t)P_\alpha(\alpha_{t+1}|\alpha_t)P(A_{t+1}^*, Y_{t+1}^*),$$

which is utilized in the second and eighth equalities of equation (53). Equation (26) is represented as:

$$
\begin{aligned}
P(\alpha_t|Y_T)=&\frac{1}{P(Y_T)}\iint P(A_T,Y_T)\mathrm{d}A_{t-1}\mathrm{d}A^*_{t+1}\\
=&\frac{1}{P(Y_T)}\iint P(A_t,Y_t)P_\alpha(\alpha_{t+1}|\alpha_t)P(A^*_{t+1},Y^*_{t+1})\mathrm{d}A_{t-1}\mathrm{d}A^*_{t+1}\\
=&\frac{1}{P(Y_T)}\int P(A_t,Y_t)\mathrm{d}A_{t-1}\int P_\alpha(\alpha_{t+1}|\alpha_t)P(A^*_{t+1},Y^*_{t+1})\mathrm{d}A^*_{t+1}\\
=&\frac{P(Y_t)}{P(Y_T)}P(\alpha_t|Y_t)\int P_\alpha(\alpha_{t+1}|\alpha_t)P(A^*_{t+1},Y^*_{t+1})\mathrm{d}A^*_{t+1}\\
=&\frac{P(Y_t)}{P(Y_T)}P(\alpha_t|Y_t)\int \frac{P_\alpha(\alpha_{t+1}|\alpha_t)P(A^*_{t+1},Y^*_{t+1})}{\int P_\alpha(\alpha_{t+1}|\alpha_t)P(A_t,Y_t)\mathrm{d}A_t}\\
&\qquad\qquad\times\left(\int P_\alpha(\alpha_{t+1}|\alpha_t)P(A_t,Y_t)\mathrm{d}A_t\right)\mathrm{d}A^*_{t+1}\\
=&P(\alpha_t|Y_t)\iint \frac{P_\alpha(\alpha_{t+1}|\alpha_t)P(A^*_{t+1},Y^*_{t+1})}{P(Y_T)P(\alpha_{t+1}|Y_t)}\\
&\qquad\qquad\times\left(\int P_\alpha(\alpha_{t+1}|\alpha_t)P(A_t,Y_t)\mathrm{d}A_t\right)\mathrm{d}A^*_{t+2}\mathrm{d}\alpha_{t+1}\\
=&P(\alpha_t|Y_t)\int \frac{\iint P(A^*_{t+1},Y^*_{t+1})P_\alpha(\alpha_{t+1}|\alpha_t)P(A_t,Y_t)\mathrm{d}A_t\mathrm{d}A^*_{t+2}}{P(Y_T)}\\
&\qquad\qquad\times\frac{P_\alpha(\alpha_{t+1}|\alpha_t)}{P(\alpha_{t+1}|Y_t)}\mathrm{d}\alpha_{t+1}\\
=&P(\alpha_t|Y_t)\int \frac{\iint P(A_T,Y_T)\mathrm{d}A_t\mathrm{d}A^*_{t+2}}{P(Y_T)}\frac{P_\alpha(\alpha_{t+1}|\alpha_t)}{P(\alpha_{t+1}|Y_t)}\mathrm{d}\alpha_{t+1}\\
=&P(\alpha_t|Y_t)\int \frac{P(\alpha_{t+1}|Y_T)P_\alpha(\alpha_{t+1}|\alpha_t)}{P(\alpha_{t+1}|Y_t)}\mathrm{d}\alpha_{t+1}. \qquad (53)
\end{aligned}
$$

In the fourth equality, the first equality of equation (52) is utilized (note that $P(Y_t) = \int P(A_t,Y_t)\mathrm{d}A_t$). The denominator of the sixth equality is obtained from the case $L=1$ in the second equality of equation (51). The ninth equality comes from the first equality. Thus, it is shown that equation (26) is exactly equivalent to equation (16).

References

Alspach, D.L. and Sorenson, H.W., 1972, " Nonlinear Bayesian Estimation Using Gaussian Sum Approximations, " *IEEE Transactions on Automatic Control*, Vol.AC-17, No.4, pp.439 – 448.

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

Aoki, M., 1987, *State Space Modeling of Time Series*, Springer-Verlag.

Aoki, M., 1990, *State Space Modeling of Time Series* (Second, Revised and Enlarged Edition), Springer-Verlag.

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

Belsley, D.A., 1973, " On the determination of Systematic Parameter Variation in the Linear Regression Model, " *Annals of Economic and Social Measurement*, Vol.2, pp.487 – 494.

Belsley, D.A. and Kuh, E., 1973, " Time-Varying Parameter Structures: An Overview, " *Annals of Economic and Social Measurement*, Vol.2, No.4, pp.375 – 379.

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.

Brockwell, P.A. and Davis, R.A., 1987, *Time Series Theory and Models*, Springer-Verlag.

Brown, B.W. and Mariano, R.S., 1984, " Residual-based Procedures for Prediction and Estimation in a Nonlinear Simultaneous System, " *Econometrica*, Vol.52, No.2, pp.321 – 343.

Brown, B.W. and Mariano, R.S., 1989, " Measures of Deterministic Prediction Bias in Nonlinear Models, " *International Economic Review*, Vol.30, No.3, pp.667 – 684.

Burmeister, E. and Wall, K.D., 1982, " Kalman Filtering Estimation of Unobserved Rational Expectations with an Application to the German Hyperinflation, " *Journal of Econometrics*, Vol.20, pp.255 – 284.

Burridge, P. and Wallis, K.F., 1988, " Prediction Theory for Autoregressive Moving Average Processes, " *Econometric Reviews*, Vol.7, No.1, pp.65 – 95.

Carlin, B.P. and Polson, N.G., 1991, " Inference for Nonconjugate Bayesian Models Using the Gibbs Sampler, " *Canadian Journal of Statistics*, Vol.19, pp.399–405.

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, 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.

Chow, G.C., 1983, *Econometrics*, McGraw-Hill Book Company.

Conrad, W. and Corrado, C., 1979, " Application of the Kalman Filter to Revisions in Monthly Retail Sales Estimates, " *Journal of Economic Dynamic and Control*, Vol.1, pp.177 – 198.

Cooley, T.F., 1977, " Generalized Least Squares Applied to Time Varying Parameter Models: A Comment, " *Annals of Economic and Social Measurement*, Vol.6, No.3, pp.313 – 314.

Cooley, T.F. and Prescott, E.C., 1976, " Estimation in the presence of stochastic parameter variation, " *Econometrica*, Vol.44, pp.167 – 183.

Cooley, T.F., Rosenberg, B. and Wall, K.D., 1977, " A Note on Optimal Smoothing for Time Varying Coefficient Problems, " *Annals of Economic and Social Measurement*, Vol.6, No.4, pp.453 – 456.

Cooper, J.P., 1973, " Time-Varying Regression Coefficients: A Mixed Estimation Approach and Operational Limitations of the General Markov Structure, " *Annals of Economic and Social Measurement*, Vol.2, No.4, pp.525 – 530.

Dempster, A.P., Laird, N.M. and Rubin, D.B., 1977, " Maximum Likelihood from Incomplete Data via the EM Algorithm, " *Journal of the Royal Statistical Society*, Ser.B, Vol.39, pp.1 – 38 (with discussion) .

Diebold, F.X. and Nerlove, M., 1989, " Unit Roots in Economic Time Series: A Selective Survey, " in *Advances in Econometrics*, Vol.8, pp.3 – 69, JAI Press.

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

Engle, R.F. and Watson, M.W., 1987, " The Kalman Filter: Applications to Forecasting and Rational expectations Models, " in *Advances in Econometrics, Fifth World Congress, Vol.I*, Cambridge University Press.

Gardner, G., Harvey, A.C. and Phillips, G.D.A., 1980, " An Algorithm for Maximum Likelihood Estimation Autoregressive-Moving Average Models by means of Kalman Filtering, " *Applied Statistics*, Vol.29, No.3, pp.311 – 322.

Gelb, A., 1974, *Applied Optimal Estimation*, MIT Press.

Gelfand, A.E., Hills, S.E., Racine-Poon, H.A., and Smith, A.F.M., 1990, " Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling, " *Journal of the American Statistical Association*, Vol.85, No.412, pp.972 – 985.

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., 1988, " Antithetic Acceleration of Monte Carlo Integration in Bayesian Inference, " *Journal of Econometrics*, Vol.38, pp.73 – 90.

Geweke, J., 1989a, " Modeling with Normal Polynomial Expansions, " in *Economic Complexity: Chaos, Sunspots, Bubbles and Nonlinearity*, edited by Barnett, W.A., Geweke, J. and Shell, K., Cambridge University Press.

Geweke, J., 1989b, " Bayesian Inference in Econometric Models Using Monte Carlo Integration, " *Econometrica*, Vol.57, pp.1317 – 1339.

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 – pp.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.

Hall, R.E., 1978, " Stochastic Implications of the Life Cycle-Permanent Income Hypothesis: Theory and Evidence, " *Journal of Political Economy*, Vol.86, No.6, pp.971 – 987.

Hall, R.E. 1990, *The Rational Consumer*, The MIT Press.

Hamilton, J.D., 1989, " A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle, " *Econometrica*, Vol.57, pp.357 – 384.

Hamilton, J.D., 1990, " Analysis of Time Series Subject to Changes in Regime, " *Journal of Econometrics*, Vol.45, pp.39 – 70.

Hamilton, J.D., 1991, " A Quasi-Bayesian Approach to Estimating Parameters for Mixtures of Normal Distributions, " *Journal of Business and Economic Statistics*, Vol.9, pp.27 – 39.

Hamilton, J.D., 1993, " Estimation, Inference and Forecasting of Time Series Subject to Changes in Regime, " in *Handbook of Statistics*, Vol.11, edited by Maddala, G.S., Rao, C.R. and Vinod, H.D., pp.231 – 260, North-Holland.

Hamilton, J.D., 1994, *Time Series Analysis*, Princeton University Press.

Hannan, E.J. and Deistler, M., 1988, *The Statistical Theory of Linear System*, John Wiley & Sons.

Harvey, A.C., 1981, *Time Series Models*, Philip Allen Publishers Limited, Oxford.

Harvey, A.C., 1987, " Applications of the Kalman Filter in Econometrics, " in *Advances in Econometrics, Fifth World Congress, Vol.I*, Cambridge University Press.

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

Howrey, E.P., 1978, " The Use of Preliminary Data in Econometric Forecasting, " *The Review of Economics and Statistics*, Vol.60, pp.193 – 200.

Howrey, E.P., 1984, " Data Revision, Reconstruction, and Prediction: An Application to Inventory Investment, " *The Review of Economics and Statistics*, Vol.66, pp.386 – 393.

Hürzeler, M. and Künsch, H.R., 1998, " Monte Carlo Approximations for General State-Space Models, " *Journal of Computational and Graphical Statistics*, Vol.7, No.2, pp.175 – 193.

Jazwinski, A.H., 1970, *Stochastic Processes and Filtering Theory*, Academic Press, New York.

Kalman, R.E., 1960, " A New Approach to Linear Filtering and Prediction Problems, " *Journal of Basic Engineering, Transactions ASME*, Ser.D, Vol.82, pp.35 – 45.

Kalman, R.E. and Bucy, R.S., 1961, " New Results in Linear Filtering and Prediction Theory, " *Journal of Basic Engineering, Transactions ASME*, Ser.D, Vol.83, pp.95 – 108.

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., 1998, " A Self-Organizing State-Space Model, " *Journal of the American Statistical Association*, Vol.93, No.443, pp.1203 – 1215.

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

Kirchen, A., 1988, *Schäzung zeitveränderlicher Strukturparameter in ökonometrischen Prognosemodellen*, Frankfurt/Main: Athenäum.

Kramer, S.C. and Sorenson, H.W., 1988, " Recursive Bayesian Estimation Using Piece-wise Constant Approximations, " *Automatica*, Vol.24, No.6, pp.789 – 801.

Knuth, D.E., 1981, *The Art of Computer Programming*, Vol.2 (Seminumerical Algorithms, 2nd ed.), Addison-Wesley.

Koop, G., 1994, " Recent Progress in Applied Bayesian Econometrics, " *Journal of Economic Surveys*, Vol.8, No.1, pp.1 – 34.

Laird, N., 1993, " The EM Algorithm, " in *Handbook of Statistics*, Vol.9, edited by Rao, C.R., pp.661 – 721, North-Holland.

Mariano, R.S. and Brown, B.W., 1983, " Asymptotic Behavior of Predictors in a Nonlinear Simultaneous System, " *International Economic Review*, Vol.24, No.3, pp.523 – 536.

Mariano, R.S. and Brown, B.W., 1989, " Stochastic Simulation, Prediction and Validation of Nonlinear Models, " in *Economics in Theory and Practice: An Eclectic Approach*, edited by Klein, L.R. and Marquez, J., pp.17 – 36, Kluwer Academic Publishers.

Mariano, R.S. and Tanizaki, H., 1995, " Prediction of Final Data with Use of Preliminary and/or Revised Data, " *Journal of Forecasting*, Vol.14, No.4, pp.351 – 380.

Mariano, R.S. and Tanizaki, H., 2000, " Simulation-Based Inference in Nonlinear State-Space Models: Application to Testing the Permanent Income Hypothesis, " *Simulation-Based Inference in Econometrics: Methods and Applications*, edited by Mariano, R.S., Weeks, M. and Schuermann, T., Cambridge University Press, forthcoming.

McNelis, P.D. and Neftci, S.N., 1983, " Policy-dependent Parameters in the Presence of Optimal Learning: An Application of Kalman Filtering to the Fair and Sargent Supply-side Equations, " *The Review of Economics and Statistics*, Vol.65, pp.296 – 306.

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

Nicholls, D.F. and Pagan, A.R., 1985, " Varying Coefficient Regression, " in *Handbook of Statistics*, Vol.5, edited by Hannan, E.J., Krishnaiah, P.R. and Rao, M.M., pp.413 – 449, Elsevier Science Publishers B.V.

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

Pagan, A.R., 1975, " A Note on the Extraction of Components from Time Series, " *Econometrica*, Vol.43, pp.163 – 168.

Rund, P.A., 1991, " Extensions of Estimation Methods Using the EM Algorithm, " *Journal of Econometrics*, Vol.49, pp.305 – 341.

Sant, D.T., 1977, " Generalized Least Squares Applied to Time Varying Parameter Models, " *Annals of Economic and Measurement*, Vol.6, No.3, pp.301 – 311.

Sarris, A.H., 1973, " A Bayesian Approach to Estimation of Time Varying Regression Coefficients, " *Annals of Economic and Social Measurement*, Vol.2, No.4, pp.501 – 523.

Shao, J., 1989, " Monte Carlo Approximations in Bayesian Decision Theory, " *Journal of the American Statistical Association*, Vol.84, No.407, pp.727 – 732.

Shumway, R.H. and Stoffer, D.S., 1982, " An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm, " *Journal of Time Series Analysis*, Vol.3, PP.253 – 264.

Smith, A.F.M. and Gelfand, A.E., 1992, " Bayesian Statistics without Tears: A Sampling-Resampling Perspective, " *The American Statistician*, Vol.46, No.2, pp.84 – 88.

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.

Sorenson, H.W. and Alspach, D.L., 1971, " Recursive Bayesian Estimation Using Gaussian Sums, " *Automatica*, Vol.7, pp.465 – 479.

Tanizaki, H., 1989, " The Kalman Filter Model under the Assumption of the First-Order Autoregressive Process in the Disturbance Terms, " *Economics Letters*, Vol.31, No.2, pp.145 – 149.

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

Tanizaki, H., 1993b, " Kalman Filter Model with Qualitative Dependent Variable, " *The Review of Economics and Statistics*, Vol.75, No.4, pp.747 – 752.

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

Tanizaki, H., 1997, " Nonlinear and Nonnormal Filters Using Monte-Carlo Methods, " *Computational Statistics and Data Analysis*, No.25, No.4, pp.417 – 439.

Tanizaki, H., 1998, " Nonlinear and Non-Gaussian State Estimation: A Quasi-Optimal Estimator, " Unpublished Manuscript (http:// ht.econ.kobe-u.ac.jp/ ~tanizaki/ cv/ papers/ quasi.pdf).

Tanizaki, H., 1999a, " On the Nonlinear and Nonnormal Filter Using Rejection Sampling, " *IEEE Transactions on Automatic Control*, Vol.44, No.2, pp.314 – 319.

Tanizaki, H., 1999b, " Nonlinear and Nonnormal Filter Using Importance Sampling: Antithetic Monte-Carlo Integration, " *Communications in Statistics, Simulation and Computation*, Vol.28, No.2, pp.463 – 486.

Tanizaki, H. and Mariano, R.S., 1994, " Prediction, Filtering and Smoothing in Nonlinear and Nonnormal Cases Using Monte-Carlo Integration, " *Journal of Applied Econometrics*, Vol.9, No.2, pp.163 – 179 (in *Econometric Inference Using Simulation Techniques*, Chap.12, edited by van DijK, H.K., Manfort, A. and Brown, B.W., pp.245 – 261, (1995), John Wiley & Sons).

Tanizaki, H., 2000, " Nonlinear and Non-Gaussian State Space Modeling Using Sampling Techniques, " Unpublished Manuscript (http:// ht.econ.kobe-u.ac.jp/ ~tanizaki/ cv/ papers/ sfil.pdf).

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).

Taylor, S.J., 1994, " Modeling Stochastic Volatility: A Review and Comparative Study, " *Mathematical Finance*, Vol.4, No.2, pp.183 – 204.

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.

Index