Independent Particle Filters

Ming T. LIN, Junni L. ZHANG, Qiansheng CHENG, and Rong CHEN

Sequential Monte Carlo methods, especially the particle filter (PF) and its various modifications, have been used effectively in dealing with stochastic dynamic systems. The standard PF samples the current state through the underlying state dynamics, then uses the current observation to evaluate the sample's importance weight. However, there is a set of problems in which the current observation provides significant information about the current state but the state dynamics are weak, and thus sampling using the current observation often produces more efficient samples than sampling using the state dynamics. In this article we propose a new variant of the PF, the independent particle filter (IPF), to deal with these problems. The IPF generates exchangeable samples of the current state from a sampling distribution that is conditionally independent of the previous states, a special case of which uses only the current observation. Each sample can then be matched with multiple samples of the previous states in evaluating the importance weight. We present some theoretical results showing that this strategy improves efficiency of estimation as well as reduces resampling frequency. We also discuss some extensions of the IPF, and use several synthetic examples to demonstrate the effectiveness of the method.

KEY WORDS: Discharging; Nonlinear filtering; Particle filter; Sequential Monte Carlo; State-space model; Target tracking.

1. INTRODUCTION

Sequential Monte Carlo (SMC) methods provide a general framework for tackling stochastic dynamic systems, which often arise in engineering, bioinformatics, economics, and other fields. They have been applied successfully in various real problems, including computer vision and target tracking (Gordon, Salmond, and Smith 1993; Avitzour 1995; Isard and Blake 1996; Blake, Bascle, Isard, and MacCormick 1998; Salmond and Gordon 2001; McGinnity and Irwin 2001; Hue, Le Cadre, and Perez 2002), economic time series analysis (Hendry and Richard 1991; Durbin and Koopman 1997; Pitt and Shephard 1999; Durham and Gallant 2002), protein structure simulation and energy minimization (Vasquez and Scheraga 1985; Grassberger 1997; Zhang and Liu 2002; Liang, Chen, and Zhang 2002; Zhang, Chen, Tang, and Liang 2003), communications and signal processing (Chen, Wang, and Liu 2000; Wang, Chen, and Guo 2002; Fong, Godsill, Doucet, and West 2002). Also see Liu 2001; Doucet, Gordon, and Krishnamurthy 2001, and references therein. Specially, Liu and Chen (1998) provided a general framework for SMC and unified various sequential simulation algorithms.

We focus our discussion herein on the discrete-time state-space model

$$\text{state equation:} \quad \mathbf{x}_t \sim q_t(\cdot|\mathbf{x}_{t-1}); \tag{1}$$

$$\text{observation equation:} \quad \mathbf{y}_t \sim f_t(\cdot|\mathbf{x}_t), \tag{2}$$

where $\mathbf{x}_t$ is the latent state and $\mathbf{y}_t$ is the observation at time t. Let $\mathbf{X}_t = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t)$ and $\mathbf{Y}_t = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$. Statistical inference about $\mathbf{X}_t$ can often be formulated as an estimation of $E_{\pi_t}[h(\mathbf{X}_t)]$, the expectation of $h(\mathbf{X}_t)$ with respect to the posterior distribution

$$\pi_t(\mathbf{X}_t|\mathbf{Y}_t) \propto q_0(\mathbf{x}_0) q_1(\mathbf{x}_1|\mathbf{x}_0) f_1(\mathbf{y}_1|\mathbf{x}_1) \cdots q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) f_t(\mathbf{y}_t|\mathbf{x}_t),$$

where $h(\cdot)$ is a square-integrable function. The SMC algorithm, in the framework of Liu and Chen (1998), draws samples of $\mathbf{X}_t$ from

$$g_t^*(\mathbf{X}_t) = g_0(\mathbf{x}_0) g_1(\mathbf{x}_1|\mathbf{x}_0, \mathbf{y}_1) \cdots g_t(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t),$$

where g_t is the conditional sampling distribution of $\mathbf{x}_t$ given $\mathbf{X}_{t-1}$ and $\mathbf{Y}_t$, and g_t^* is the joint sampling distribution of $\mathbf{X}_t$. Each draw of $\mathbf{X}_t$ needs to be weighted by

$$w_t(\mathbf{X}_t) = \frac{\pi_t(\mathbf{X}_t|\mathbf{Y}_t)}{g_t^*(\mathbf{X}_t)}.$$

It can be shown that such a sample, $\{(\mathbf{X}_t^{(j)}, w_t^{(j)}),\ j = 1, \ldots, m\}$, is *properly weighted* with respect to the distribution π_t. That is, for any square-integrable function $h(\cdot)$,

$$\frac{\sum_{j=1}^m w_t^{(j)} h(\mathbf{X}_t^{(j)})}{\sum_{j=1}^m w_t^{(j)}} \to E_{\pi_t}[h(\mathbf{X}_t)] \quad \text{as } m \to \infty.$$

In all of the discussions that follow, we assume that the sampling distribution g^* selected and the function h satisfy $\mathrm{var}[w(\mathbf{X}_t)] < \infty$ and $\mathrm{var}[h(\mathbf{X}_t) w(\mathbf{X}_t)] < \infty$.

We consider the situation when h is a function of $\mathbf{x}_t$ only. When SMC is used for filtering purposes and the inference is made directly using the samples (particles), it is also called the *particle filter* (PF) (Kitagawa 1996; Carpenter, Clifford, and Fearnhead 1999; Pitt and Shephard 1999). Note that SMC (PF) can be implemented in a sequential way; at stage t, for $j = 1, \ldots, m$:

1. Generate $\mathbf{x}_t^{(j)}$ from $g_t(\mathbf{x}_t^{(j)}|\mathbf{X}_{t-1}^{(j)}, \mathbf{Y}_t)$, and let $\mathbf{X}_t^{(j)} = (\mathbf{X}_{t-1}^{(j)}, \mathbf{x}_t^{(j)})$.
2. Compute incremental weight

$$u_t^{(j)} = \frac{\pi_t(\mathbf{X}_{t-1}^{(j)}, \mathbf{x}_t^{(j)}|\mathbf{Y}_t)}{\pi_{t-1}(\mathbf{X}_{t-1}^{(j)}|\mathbf{Y}_{t-1}) g_t(\mathbf{x}_t^{(j)}|\mathbf{X}_{t-1}^{(j)}, \mathbf{Y}_t)}$$
$$\propto \frac{q_t(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)}) f_t(\mathbf{y}_t|\mathbf{x}_t^{(j)})}{g_t(\mathbf{x}_t^{(j)}|\mathbf{X}_{t-1}^{(j)}, \mathbf{Y}_t)},$$

Ming T. Lin is a Graduate Student, LMAM, School of Mathematical Sciences (E-mail: *linming@uic.edu*), Junni L. Zhang is Assistant Professor, Department of Business Statistics and Econometrics, Guanghua School of Management (E-mail: *zjn@gsm.pku.edu.cn*), and Qiansheng Cheng is Professor, LMAM, School of Mathematical Sciences (E-mail: *qcheng@pku.edu.cn*), Peking University, Beijing, 100871, P.R. China. Rong Chen is Professor, Department of Business Statistics and Econometrics, Peking University and Department of Information and Decision Sciences, University of Illinois, Chicago, IL 60607 (E-mail: *rongchen@uic.edu*). Zhang's research is sponsored in part by Chinese National Science Foundation grant 10401003, and her work was partially done while visiting the Institute for Mathematical Sciences, National University of Singapore in 2004. Chen's research is sponsored in part by National Science Foundation grants DMS-00-73601, DMS-02-44541, CCR 9980599, and National Institutes of Health grant R01 Gm068958. The authors thank the associate editor and three referees for their detailed and constructive comments that led to significant improvement of the article.

© 2005 American Statistical Association
Journal of the American Statistical Association
December 2005, Vol. 100, No. 472, Theory and Methods
DOI 10.1198/016214505000000349

and let $w_t^{(j)} = w_{t-1}^{(j)} u_t^{(j)}$.

Sample generation and importance weight calculation are two key factors determining the efficiency of SMC methods. The bootstrap filter (Gordon et al. 1993) uses the state dynamics in (1) as the sampling distribution, that is, $g_t(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t) = q_t(\mathbf{x}_t|\mathbf{x}_{t-1})$. Fox et al. (2001), Torma and Szepesvari (2003), Isard and Blake (2004), and Rossi (2004), in several special applications, proposed using a sampling distribution depending only on the current observation,

$$g_t(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t) \propto f_t(\mathbf{y}_t|\mathbf{x}_t). \tag{3}$$

Kong, Liu, and Wong (1994) and Liu and Chen (1995, 1998) proposed using a sampling distribution that combines information from both the state dynamics and the current observation, such as

$$g_t(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t) \propto q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) f_t(\mathbf{y}_t|\mathbf{x}_t). \tag{4}$$

Various approximation methods have been developed for use when it is difficult to sample from (4), including auxiliary PFs (Pitt and Shephard 1999), unscented PFs (van der Merwe, Doucet, de Freitas, and Wan 2002), mixture Kalman filters (Chen and Liu 2000), and Gaussian sum PFs (Kotecha and Djuric 2003).

In this article we focus on a set of problems for which the current observation provides significant information about the current state but the state dynamics are weak. For example, in visual tracking and robot localization, the observations in the form of images provide significant information, whereas the motion dynamics are not stable (Fox et al. 2001; Blake et al. 1998; Torma and Szepesvari 2003). Such problems also arise in target tracking with fast and unstable maneuvering (Bar-Shalom and Fortmann 1988), fast flat-fading channels (Chen et al. 2000), and population biology. Also, more and more applications in communications are under a high signal-to-noise ratio environment.

Because of the weak state dynamics, the particles from the bootstrap filter do not follow the target distribution closely, resulting in excessive weight variation that will greatly reduce the accuracy of statistical inference (Kong et al. 1994). In contrast, the sampling distribution in (3) uses information from the current observation in the sampling distribution and presumably can work well. In this article, we extend this idea and propose using sampling distributions in the form of

$$g_t(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t) \propto g_t(\mathbf{x}_t|\mathbf{Y}_t). \tag{5}$$

We call a PF using (5) as the sampling distribution an *independent particle filter* (IPF), because each draw of $\mathbf{x}_t$ is conditionally independent of individual past particles and of each other. In this article we formulate a general framework of the IPF and study its theoretical and empirical properties. The most intriguing feature of IPF, as also noted by Torma and Szepesvari (2003) and Isard and Blake (2004), is that it allows the draws of $\mathbf{x}_t$ from the sampling distribution (5) to be matched arbitrarily with the past particles of $\mathbf{X}_{t-1}$. We show that for the set of problems on which we focus, an IPF can outperform a PF using full information, that is, using the sampling distribution in (4).

We proceed as follows. Section 2 formally presents IPF and a multiple-matching scheme that uses the independence feature. It also discusses some theoretical properties of the IPF with multiple matching. Section 3 discusses extensions of the IPF and practical guidance for designing IPFs. Section 4 presents several examples. The technical proofs of all theorems are presented in the Appendix.

2. THE INDEPENDENT PARTICLE FILTER

2.1 The Basic Form

Formally, the IPF can be described as follows: At stage t, for $j = 1, \dots, m$,

1. Generate $\mathbf{x}_t^{(j)}$ from $g_t(\mathbf{x}_t^{(j)}|\mathbf{X}_{t-1}^{(j)}, \mathbf{Y}_t) = g_t(\mathbf{x}_t^{(j)}|\mathbf{Y}_t)$, and let $\mathbf{X}_t^{(j)} = (\mathbf{X}_{t-1}^{(j)}, \mathbf{x}_t^{(j)})$.
2. Compute incremental weight

$$u_t^{(j)} \propto \frac{q_t(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)}) f_t(\mathbf{y}_t|\mathbf{x}_t^{(j)})}{g_t(\mathbf{x}_t^{(j)}|\mathbf{Y}_t)}$$

and let $w_t^{(j)} = w_{t-1}^{(j)} u_t^{(j)}$.

If $g_t(\mathbf{x}_t|\mathbf{Y}_t) \propto f_t(\mathbf{y}_t|\mathbf{x}_t)$ is used, then the incremental weight becomes $u_t^{(j)} \propto q_t(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)})$. This would require that $\int f_t(\mathbf{y}_t|\mathbf{x}_t)\, d\mathbf{x}_t < \infty$; that is, $f_t(\mathbf{y}_t|\mathbf{x}_t)$ as a distribution of $\mathbf{x}_t$ is proper.

2.2 Independent Particle Filter With Multiple Matching

Throughout this article, we let $S_t = \{(\mathbf{x}_t^{(j)}, w_t^{(j)}),\ j = 1, \dots, m\}$ denote a set of weighted random samples properly weighted with respect to $\pi_t(\mathbf{x}_t|\mathbf{Y}_t)$. Because the draws $\{\mathbf{x}_t^{(j)}, j = 1, \dots, m\}$ in the IPF are not coupled with any particular particles in S_{t-1}, they can be matched with the samples in S_{t-1} in any order. Specifically, if $\mathbf{x}_{t-1}^{(i)}$ is matched with $\mathbf{x}_t^{(j)}$, then the importance weight is

$$\begin{aligned} \lambda_t^{(i,j)} &= w_{t-1}^{(i)} u_t^{(i,j)}, \\ u_t^{(i,j)} &\propto \frac{q_t(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(i)}) f_t(\mathbf{y}_t|\mathbf{x}_t^{(j)})}{g_t(\mathbf{x}_t^{(j)}|\mathbf{Y}_t)}. \end{aligned} \tag{6}$$

Each different match will result in different weights for $\{\mathbf{x}_t^{(j)}, j = 1, \dots, m\}$, and thus different estimates of $E_{\pi_t}[h(\mathbf{x}_t)]$. It is natural to combine these estimates to produce more efficient ones.

Consider L different permutations of $(1, \dots, m)$: $\mathbf{K}_l \doteq (k_{l,1}, \dots, k_{l,m})$, $l = 1, \dots, L$. For each permutation, the past particles $\{\mathbf{x}_{t-1}^{(k_{l,j})}, j = 1, \dots, m\}$ are matched with the current particles $\{\mathbf{x}_t^{(j)}, j = 1, \dots, m\}$, and the importance weights are

$$w_{t,l}^{(j)} \equiv \lambda_t^{(k_{l,j},j)}. \tag{7}$$

The resulting set of weighted random samples, $\{(\mathbf{x}_t^{(j)}, w_{t,l}^{(j)}), j = 1, \dots, m\}$, is properly weighted with respect to $\pi(\mathbf{x}_t|\mathbf{Y}_t)$. Combining different permutations, we can also construct a new weight, $w_t^{(j)}$, for each $\mathbf{x}_t^{(j)}$,

$$w_t^{(j)} = \frac{\sum_{l=1}^{L} w_{t,l}^{(j)}}{L}, \tag{8}$$

and estimate $E_{\pi_t}[h(\mathbf{x}_t)]$ with

$$\widehat{H}_{t,L} = \frac{m^{-1}\sum_{j=1}^m w_t^{(j)} h(\mathbf{x}_t^{(j)})}{m^{-1}\sum_{j=1}^m w_t^{(j)}} = \frac{(Lm)^{-1}\sum_{j=1}^m \sum_{l=1}^L w_{t,l}^{(j)} h(\mathbf{x}_t^{(j)})}{(Lm)^{-1}\sum_{j=1}^m \sum_{l=1}^L w_{t,l}^{(j)}} \equiv \frac{A_{t,L}}{B_{t,L}}. \qquad (9)$$

Because $\lambda_t^{(i,j)} h(\mathbf{x}_t^{(j)})$ and $\lambda_t^{(i,j)}$ are exchangeable for different i and j, we can define two constants, $\mathcal{A}_t \equiv E[\lambda_t^{(i,j)} h(\mathbf{x}_t^{(j)})]$ and $\mathcal{B}_t \equiv E(\lambda_t^{(i,j)}) > 0$. We then have the following results.

Proposition 1. For any L permutations and any $1 \le l \le L$,

$$E(A_{t,L}) = \mathcal{A}_t, \qquad E(B_{t,L}) = \mathcal{B}_t, \qquad E_{\pi_t}[h(\mathbf{x}_t)] = \frac{\mathcal{A}_t}{\mathcal{B}_t}. \qquad (10)$$

Proposition 2. Under the assumption that $\{(\mathbf{x}_{t-1}^{(j)}, w_{t-1}^{(j)}),\ j = 1, \dots, m\}$ are exchangeable, (a) for any fixed $2 \le L \le m$, $\mathrm{var}(A_{t,L})$ and $\mathrm{var}(B_{t,L})$ achieve minimum when the L permutations are mutually exclusive, that is, for any l_1, l_2 $(1 \le l_1 \ne l_2 \le L)$, $k_{l_1,j} \ne k_{l_2,j},\ j = 1, \dots, m$; and (b) with mutually exclusive permutations, $\mathrm{var}(A_{t,L})$ and $\mathrm{var}(B_{t,L})$ decrease as L increases $(L \le m)$.

Proposition 3. When $\{(\mathbf{x}_{t-1}^{(j)}, w_{t-1}^{(j)}),\ j = 1, \dots, m\}$ are exchangeable, if $A_{t,m}$ and $B_{t,m}$ are obtained with m mutually exclusive permutations, then for any $L > m$, $\mathrm{var}(A_{t,L}) \ge \mathrm{var}(A_{t,m})$ and $\mathrm{var}(B_{t,L}) \ge \mathrm{var}(B_{t,m})$.

Remark 1. According to (9), (10), and Proposition 2, to reduce the variance of $\widehat{H}_{t,L}$, we should choose mutually exclusive permutations. The largest set containing such permutations is of size $L = m$. Proposition 3 says that using $L > m$ nonexclusive permutations does not help. The size of L controls a trade-off between estimation efficiency and computational cost.

Remark 2. There is no obvious way to select the "good" matches without first calculating the weights. Torma and Szepesvari (2003) and Isard and Blake (2004) proposed calculating all the weights in complete matching ($L = m$ mutually exclusive permutations) and then use these weights to sample a "good" match that is used for estimation. In our approach, all weights that have been calculated are used in estimation, in partial matching ($L < m$) or complete matching.

Remark 3. Torma and Szepesvari (2003) and Isard and Blake (2004) used complete matching for certain special applications. Although complete matching has the maximum benefit, it is often computationally intensive. As shown in the examples in Section 4, partial matching with $L = 5$ or 10 often performs better than complete matching, given the same amount of computation.

2.3 Multiple Matching as a Discharging Tool for the Independent Particle Filter

2.3.1 Discharging in Standard Importance Sampling. Suppose that the target distribution is $\pi(\mathbf{x})$. To draw a set of random samples from a trial distribution $g(\mathbf{x})$, we can generate $\{(\mathbf{x}^{(j)}, \mathbf{z}^{(j)}),\ j = 1, \dots, m\}$ from $g^*(\mathbf{x}, \mathbf{z})$ with $\int g^*(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = g(\mathbf{x})$ and consider only $\{\mathbf{x}^{(j)},\ j = 1, \dots, m\}$. The importance weight can be calculated in two ways: through the joint trial distribution, $w_j^* = \frac{\pi(\mathbf{x}^{(j)}, \mathbf{z}^{(j)})}{g^*(\mathbf{x}^{(j)}, \mathbf{z}^{(j)})}$, where $\pi(\mathbf{x}, \mathbf{z})$ is any distribution satisfying $\int \pi(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \pi(\mathbf{x})$, and by direct calculation through the marginal distribution $w_j = \frac{\pi(\mathbf{x}^{(j)})}{g(\mathbf{x}^{(j)})}$. Both $\{(\mathbf{x}^{(j)}, w_j^*),\ j = 1, \dots, m\}$ and $\{(\mathbf{x}^{(j)}, w_j),\ j = 1, \dots, m\}$ are properly weighted with respect to $\pi(\mathbf{x})$, which can be seen from

$$\int\int h(\mathbf{x}) \frac{\pi(\mathbf{x}, \mathbf{z})}{g^*(\mathbf{x}, \mathbf{z})} g^*(\mathbf{x}, \mathbf{z})\, d\mathbf{z}\, d\mathbf{x} = \int h(\mathbf{x}) \frac{\pi(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x})\, d\mathbf{x} = \int h(\mathbf{x}) \pi(\mathbf{x})\, d\mathbf{x}. \qquad (11)$$

Proposition 4. With the foregoing w_j^* and w_j, we have

$$\mathrm{var}_g\left[\frac{1}{m}\sum_{j=1}^m h(\mathbf{x}^{(j)}) w_j\right] \le \mathrm{var}_{g^*}\left[\frac{1}{m}\sum_{j=1}^m h(\mathbf{x}^{(j)}) w_j^*\right].$$

Hence the weight based on the marginal distribution is more efficient than the weight based on the joint distribution, even though the sample was actually generated using the joint distribution. We call such a procedure *discharging*. It removes the inefficiency due to the additional sampling of $\mathbf{z}$.

2.3.2 Discharging in the Independent Particle Filter. We demonstrate that multiple matching serves as an approximation of discharging for the IPF. Suppose that we perform complete matching (i.e., using $L = m$ mutually exclusive permutations) in the IPF; then, according to (6)–(8),

$$w_t^{(j)} = \frac{1}{m}\sum_{i=1}^m w_{t-1}^{(i)} \frac{\pi_t(\mathbf{X}_{t-1}^{(i)}, \mathbf{x}_t^{(j)} | \mathbf{Y}_t)}{\pi_{t-1}(\mathbf{X}_{t-1}^{(i)} | \mathbf{Y}_{t-1}) g_t(\mathbf{x}_t^{(j)} | \mathbf{Y}_t)}.$$

Because $\{(\mathbf{X}_{t-1}^{(i)}, w_{t-1}^{(i)})\}_{i=1}^m$ is properly weighted with respect to $\pi_{t-1}(\mathbf{X}_{t-1} | \mathbf{Y}_{t-1})$, we have

$$w_t^{(j)} \approx \int \frac{\pi_t(\mathbf{X}_{t-1}, \mathbf{x}_t^{(j)} | \mathbf{Y}_t)}{\pi_{t-1}(\mathbf{X}_{t-1} | \mathbf{Y}_{t-1}) g_t(\mathbf{x}_t^{(j)} | \mathbf{Y}_t)} \pi_{t-1}(\mathbf{X}_{t-1} | \mathbf{Y}_{t-1})\, d\mathbf{X}_{t-1} = \frac{\pi_t(\mathbf{x}_t^{(j)} | \mathbf{Y}_t)}{g_t(\mathbf{x}_t^{(j)} | \mathbf{Y}_t)}, \qquad (12)$$

which is the weight as if we had performed discharging. With partial matching ($L < m$), the foregoing calculation still holds, although the approximation in (12) is less accurate.

Remark 4. It should be clear that the IPF with discharging is not necessarily better than the PF without discharging, due to the use of different sampling distributions. However, discharging provides a justification that the IPF with multiple matching performs better than the IPF without multiple matching. In some cases, this improvement can be significant enough so that the IPF outperforms the PF using full information in the sampling distribution (4), as shown by the examples in Section 4.

2.4 Multiple Matching and Resampling

Resampling is an important step in all SMC algorithms, because the importance weight w_t can be increasingly skewed, resulting in many unrepresentative samples of $\mathbf{x}_t$. Suppose that we have obtained S_t. By drawing a new set of samples from $\{\mathbf{X}_t^{(j)},\ j = 1, \dots, m\}$ with probabilities proportional to $w_t^{(j)}$ and reassigning the weights to 1, we can effectively discard the samples with small weights and duplicate the more important samples. Liu and Chen (1998) proposed engaging the resampling step when the effective sample size (Kong et al. 1994), $ESS = \frac{m}{1+C^2(w)}$, is less than a certain threshold, where $C(w)$ is the coefficient of variation of the weights.

A shortcoming of resampling is that it reduces sample diversity. The following proposition shows that multiple matching in the IPF reduces the coefficient of variation of the weights with increasing L, and thus reduces the frequency of resampling, maintaining sample diversity.

Proposition 5. $E(w_t^{(j)}) = \mathcal{B}_t$ is constant for different L, while $\mathrm{var}(w_t^{(j)})$ decreases as L increases.

The following proposition further shows that if complete matching is to be performed at time t, then resampling should not be performed at time $t-1$, because it cannot improve the efficiency of statistical inference about $\mathbf{x}_t$.

Proposition 6. Suppose that at time $t-1$ we have obtained S_{t-1}, where (without loss of generality) the weights are standardized such that $\sum_{j=1}^m w_{t-1}^{(j)} = m$. Let $A_{t,m}$ and $B_{t,m}$ be obtained with m mutually exclusive permutation. Suppose that we have performed a resampling step and obtained $\{(\mathbf{x}_{t-1}^{*(j)}, w_{t-1}^{*(j)} = 1),\ j = 1, \ldots, m\}$ at time $t-1$. Let $A_{t,m}^*$ and $B_{t,m}^*$ be obtained with m mutually exclusive permutations using the resampled samples and weights. Then

$$E(A_{t,m}^*) = E(A_{t,m}), \qquad E(B_{t,m}^*) = E(B_{t,m}),$$
$$\mathrm{var}(A_{t,m}^*) \geq \mathrm{var}(A_{t,m}), \qquad \mathrm{var}(B_{t,m}^*) \geq \mathrm{var}(B_{t,m}).$$

3. EXTENSIONS OF INDEPENDENT PARTICLE FILTERS

3.1 Using the Information From Past Particles

The efficiency of the IPF could be improved if the information from past particles is also used. To achieve this and at the same time obtain exchangeable samples of $\mathbf{x}_t$ as required by the IPF, we can use the *general* information from all samples of $\mathbf{x}_{t-1}$. More formally, suppose that we have obtained S_{t-1}. Through the state equation (1), we can construct a distribution $\boldsymbol{\xi}(\mathbf{x}_t|\mathbf{Y}_{t-1})$ for $\mathbf{x}_t$ using the overall information carried by S_{t-1}. Then the sampling distribution for $\mathbf{x}_t$ becomes

$$g_t(\mathbf{x}_t|\mathbf{Y}_t) \propto f_t(\mathbf{y}_t|\mathbf{x}_t)\boldsymbol{\xi}(\mathbf{x}_t|\mathbf{Y}_{t-1}).$$

In particular, suppose that the state equation is in the form of $\mathbf{x}_t = \mathbf{Q}(\mathbf{x}_{t-1}) + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Then we can use $\boldsymbol{\xi}(\mathbf{x}_t|\mathbf{Y}_{t-1}) \sim \mathrm{N}(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$, where

$$\widehat{\boldsymbol{\mu}} = \frac{\sum_{j=1}^m w_{t-1}^{(j)}\mathbf{Q}(\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^m w_{t-1}^{(j)}} \qquad \text{and}$$
$$\widehat{\boldsymbol{\Sigma}} = \frac{\sum_{j=1}^m w_{t-1}^{(j)}\mathbf{Q}(\mathbf{x}_{t-1}^{(j)})(\mathbf{Q}(\mathbf{x}_{t-1}^{(j)}))^T}{\sum_{j=1}^m w_{t-1}^{(j)}} - \widehat{\boldsymbol{\mu}}\widehat{\boldsymbol{\mu}}^T + \sigma^2\mathbf{I}. \tag{13}$$

Suppose further that $f_t(\mathbf{y}_t|\mathbf{x}_t)$ can be well approximated by $\hat{f}_t(\mathbf{y}_t|\mathbf{x}_t)$, a Gaussian (or mixture Gaussian) approximation of $f_t(\mathbf{y}_t|\mathbf{x}_t)$ as a distribution of $\mathbf{x}_t$, we form the sampling distribution $g_t(\mathbf{x}_t|\mathbf{Y}_t) \propto \hat{f}_t(\mathbf{y}_t|\mathbf{x}_t)\xi(\mathbf{x}_t|\mathbf{Y}_{t-1})$, which is a Gaussian (or mixture Gaussian) distribution. A similar approach has been used in other cases (e.g., Chopin 2002).

Such a sampling distribution needs to be constructed only once at each stage t, hence bearing minimal additional computational cost. The draws of $\mathbf{x}_t$ are exchangeable, so partial or complete matching can still be conducted, and the theoretical results in Section 2 will hold.

3.2 When $\mathbf{y}_t$ Is Directly Related to Only Part of $\mathbf{x}_t$

In many applications, the observation is related directly only to part of the state vector $\mathbf{x}_t$. For example, in target tracking problems the state vector includes speed and acceleration, which are not directly related to the observations. In state-space models, if there are some unknown parameters that are involved only in the state equation but not in the observation equation, then they are often included as part of the state vector and are not directly related to the observations.

In these cases, the state space-models can be written as

$$\begin{aligned} \text{state-equation:}\quad \mathbf{x}_t &= (x_{t,1}, x_{t,2}) \\ &\sim q_t(\cdot|\mathbf{x}_{t-1}) = q_{t,1}(x_{t,1}|\mathbf{x}_{t-1}) \\ &\qquad \times q_{t,2}(x_{t,2}|\mathbf{x}_{t-1}, x_{t,1}); \\ \text{observation equation:}\quad \mathbf{y}_t &\sim f_t(\cdot|x_{t,1}). \end{aligned}$$

The first component, $x_{t,1}$, can be generated using a sampling distribution, $g_{t,1}(x_{t,1}|\mathbf{Y}_t)$, constructed as before. These samples can be matched arbitrarily with multiple previous particles of $\mathbf{x}_{t-1}$ and properly weighted. When $x_{t,1}^{(j)}$ is matched with $\mathbf{x}_{t-1}^{(i)}$, the partial incremental weight is

$$u_{t,1}^{(i,j)} \propto \frac{f_t(\mathbf{y}_t|x_{t,1}^{(j)})q_{t,1}(x_{t,1}^{(j)}|\mathbf{x}_{t-1}^{(i)})}{g_{t,1}(x_{t,1}^{(j)}|\mathbf{Y}_t)}.$$

Generating the second component, $x_{t,2}$, based on a second sampling distribution, $g_{t,2}(x_{t,2}|\mathbf{x}_{t-1}, x_{t,1})$, will require a one-to-one match of $x_{t,1}$ and $\mathbf{x}_{t-1}$. Given a match between $\mathbf{x}_{t-1}^{(i)}$ and $x_{t,1}^{(j)}$, we can generate $x_{t,2}^{(i,j)} \sim g_{t,2}(x_{t,2}|\mathbf{x}_{t-1}^{(i)}, x_{t,1}^{(j)})$. The corresponding partial incremental weight for this stage is

$$u_{t,2}^{(i,j)} \propto \frac{q_{t,2}(x_{t,2}^{(i,j)}|\mathbf{x}_{t-1}^{(i)}, x_{t,1}^{(j)})}{g_{t,2}(x_{t,2}^{(i,j)}|\mathbf{x}_{t-1}^{(i)}, x_{t,1}^{(j)})}.$$

Under this setting, we modify the IPF as follows.

3.2.1 MIPF-1. Suppose that at stage $t-1$, we have obtained S_{t-1}; then, at stage t:

1. Generate $x_{t,1}^{(j)}, j = 1, \ldots, m$, from $g_{t,1}(x_{t,1}|\mathbf{Y}_t)$.
2. For each permutation $\mathbf{K}_l = (k_{l,1}, \ldots, k_{l,m}),\ l = 1, \ldots, L$, match $\mathbf{x}_{t-1}^{(k_{l,j})}$ with $x_{t,1}^{(j)}$:
    - Calculate the partial incremental weight $u_{t,1}^{(k_{l,j},j)}, j = 1, \ldots, m$.
    - Generate $x_{t,2}^{(k_{l,j},j)} \sim g_{t,2}(x_{t,2}|\mathbf{x}_{t-1}^{(k_{l,j})}, x_{t,1}^{(j)})$, and calculate the partial incremental weight $u_{t,2}^{(k_{l,j},j)}$; let $w_{t,l}^{(j)} = w_{t-1}^{(k_{l,j})} u_{t,1}^{(k_{l,j},j)} u_{t,2}^{(k_{l,j},j)}$.

3. Combine all of the weighted samples of $\mathbf{x}_t$, that is, $\{((x_{t,1}^{(j)}, x_{t,2}^{(k_{l,j})}), w_{t,l}^{(j)}),\ j = 1, \dots, m,\ l = 1, \dots, L\}$, to estimate $E_{\pi_t}[h(\mathbf{x}_t)]$,

$$\widehat{H}_{t,L} = \frac{\sum_{l=1}^{L} \sum_{j=1}^{m} w_{t,l}^{(j)} h(x_{t,1}^{(j)}, x_{t,2}^{(k_{l,j},j)})}{\sum_{l=1}^{L} \sum_{j=1}^{m} w_{t,l}^{(j)}}. \qquad (14)$$

4. Construct the set of weighted random samples, $S_t = \{(\mathbf{x}_t^{(j)}, w_t^{(j)}),\ j = 1, \dots, m\}$, for propagation to stage $t+1$:
   - For each $x_{t,1}^{(j)}$, select a final match $x_{t,2}^{(j)}$ from the set $\{x_{t,2}^{(k_{l,j},j)},\ l = 1, \dots, L\}$ with probabilities proportional to $w_{t,l}^{(j)}$.
   - Set the weight for each $\mathbf{x}_t^{(j)} = (x_{t,1}^{(j)}, x_{t,2}^{(j)})$ as $w_t^{(j)} = \sum_{l=1}^{L} w_{t,l}^{(j)}/L$.

Proposition 7. For MIPF-1, Propositions 1–3, 5, and 6 hold.

Remark 5. Although estimation can also be done after step 4, the estimator (14) has smaller variation under the Rao–Blackwellization principle (Liu and Chen 1998). If h involves only $x_{t,1}$, then the estimation should be done using

$$\widehat{H}_{t,L} = \frac{\sum_{l=1}^{L} \sum_{j=1}^{m} w_{t-1}^{(k_{l,j})} u_{t,1}^{(k_{l,j},j)} h(x_{t,1}^{(j)})}{\sum_{l=1}^{L} \sum_{j=1}^{m} w_{t-1}^{(k_{l,j})} u_{t,1}^{(k_{l,j},j)}}.$$

This estimator further reduces variation under the Rao–Blackwellization principle.

In MIPF-1, for each $\mathbf{x}_t^{(j)}$, we need to generate $x_{t,2}$ and calculate $u_{t,2}$ L times, increasing the computational cost. Next we present another modification of the IPF that has lower computational cost but may be less statistically efficient than MIPF-1.

3.2.2 MIPF-2. Suppose that at stage $t-1$, we have obtained S_{t-1}; then at stage t:

1. Generate $x_{t,1}^{(j)}, j = 1, \dots, m$, from $g_{t,1}(x_{t,1}|\mathbf{Y}_t)$.
2. For each permutation $\mathbf{K}_l = (k_{l,1}, \dots, k_{l,m})$, $l = 1, \dots, L$, match $\mathbf{x}_{t-1}^{(k_{l,j})}$ with $x_{t,1}^{(j)}$, and calculate the partial incremental weight $u_{t,1}^{(k_{l,j},j)}$.
3. Construct the set of weighted random samples, $S_t = \{(\mathbf{x}_t^{(j)}, w_t^{(j)}),\ j = 1, \dots, m\}$, to be used for estimation of $E_{\pi_t}[h(\mathbf{x}_t)]$ and for propagation to stage $t+1$.
   - For each $x_{t,1}^{(j)}$, select a final match $\mathbf{x}_{t-1}^{(s_j)}$ from the set $\{\mathbf{x}_{t-1}^{(k_{l,j},j)},\ l = 1, \dots, L\}$ with probabilities proportional to $w_{t-1}^{(k_{l,j})} u_{t,1}^{(k_{l,j},j)}$.
   - Generate $x_{t,2}^{(j)}$ from $x_{t,2}^{(j)} \sim g_{t,2}(x_{t,2}|\mathbf{x}_{t-1}^{(s_j)}, x_{t,1}^{(j)})$, and calculate $u_{t,2}^{(s_j,j)}$.
   - Calculate the weight for each $\mathbf{x}_t^{(j)} = (x_{t,1}^{(j)}, x_{t,2}^{(j)})$ as $w_t^{(j)} = (\sum_{l=1}^{L} w_{t-1}^{(k_{l,j})} u_{t,1}^{(k_{l,j},j)}/L) u_{t,2}^{(s_j,j)}$.
   - The estimate of $E_{\pi_t}[h(\mathbf{x}_t)]$ is then $\widehat{H}_{t,L} = (\sum_{j=1}^{m} w_t^{(j)} \times h(\mathbf{x}_t^{(j)}))/\sum_{j=1}^{m} w_t^{(j)}$.

Remark 6. If the second sampling distribution, $g_{t,2}$, is chosen as the state dynamics, $q_{t,2}(x_{t,2}|\mathbf{x}_{t-1}, x_{t,1})$, then the matching selection probabilities in step 4 of MIPF-1 and those in step 3 of MIPF-2 become the same, because $u_{t,2}^{(i,j)}$ is constant.

4. SYNTHETIC EXAMPLES

4.1 Example 1: Target Tracking With Random Acceleration

Consider a two-dimensional target tracking model with random acceleration,

$$\text{state equation:} \quad \begin{pmatrix} \mathbf{z}_t \\ \mathbf{v}_t \end{pmatrix} = \begin{pmatrix} \mathbf{I}_2 & T_0\mathbf{I}_2 \\ 0 & \mathbf{I}_2 \end{pmatrix} \begin{pmatrix} \mathbf{z}_{t-1} \\ \mathbf{v}_{t-1} \end{pmatrix} + \begin{pmatrix} .5T_0^2 \boldsymbol{\varepsilon}_t \\ T_0 \boldsymbol{\varepsilon}_t \end{pmatrix}; \qquad (15)$$

$$\text{observation equation:} \quad \mathbf{y}_t = \mathbf{z}_t + \boldsymbol{\eta}_t, \qquad (16)$$

where $\boldsymbol{\varepsilon}_t \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I}_2)$ and $\boldsymbol{\eta}_t \sim \mathrm{N}(\mathbf{0}, \delta^2\mathbf{I}_2)$. Here $\mathbf{z}_t$, $\mathbf{v}_t$, and $\boldsymbol{\varepsilon}_t$ are the position, velocity, and random acceleration vectors for a target moving on a two-dimensional plane. $\mathbf{I}_2$ is the 2×2 identity matrix, and T_0 is the time interval between observations. The state vector is $\mathbf{x}_t = (\mathbf{z}_t, \mathbf{v}_t)^T$, in which $\mathbf{z}_t$ is observed with noise $\boldsymbol{\eta}_t$. The variances σ^2 and δ^2 specify the strength of information for the current state in the state dynamics and the current observation.

Because the model in (15) and (16) is linear and Gaussian, a Kalman filter can be used to obtain the exact value of $E(\mathbf{z}_t|\mathbf{Y}_t)$, and thus this example is used purely for illustration. To compare different PF methods, we use the root mean squared error (RMSE), defined as

$$RMSE = \left[\frac{1}{T}\sum_{t=1}^{T} \|\widehat{\mathbf{z}}_t - E(\mathbf{z}_t|\mathbf{Y}_t)\|_2^2\right]^{1/2}, \qquad (17)$$

where $\widehat{\mathbf{z}}_t$ is the estimate of $E(\mathbf{z}_t|\mathbf{Y}_t)$ and T is the total number of observations.

We compared the following PF methods. PF1 is the bootstrap filter, using $g_t(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t) \propto q_t(\mathbf{x}_t|\mathbf{x}_{t-1})$. PF2 uses the sampling distribution (4), which combines information from both the state dynamics and the current observation. APF is the auxiliary PF (Pitt and Shephard 1999). Specifically, at each step t, we first resample from the set $\{(\mathbf{x}_{t-1}^{(j)}, w_{t-1}^{(j)}),\ j = 1, \dots, m\}$ with probabilities proportional to $\widetilde{w}_{t-1}^{(j)} = w_{t-1}^{(j)} f_t(\mathbf{y}_t|E(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)}))$ then draw $\mathbf{x}_t^{(j)}$ from $q_t(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})$ for $j = 1, \dots, m$. So, through an approximation, APF also uses information from both $\mathbf{x}_{t-1}$ and $\mathbf{y}_t$. MIPF is the MIPF-2 algorithm described in Section 3.2. According to the state equation, $\mathbf{v}_t$ can be determined exactly by $\mathbf{z}_{t-1}$, $\mathbf{v}_{t-1}$, and $\mathbf{z}_t$ as $\mathbf{v}_t = \mathbf{v}_{t-1} + 2(\mathbf{z}_t - \mathbf{z}_{t-1} - T_0\mathbf{v}_{t-1})/T_0$, so the state dynamic, $q_{t,2}(\mathbf{v}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$, and the sampling distribution, $g_{t,2}(\mathbf{v}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$, in MIPF-2 are degenerate.

In the simulation that follows, we set the initial position vector and velocity vector as $\mathbf{z}_0 = (0, 0)'$ and $\mathbf{v}_0 = (1, 0)'$, and let $T_0 = 5$ and $T = 100$. A dynamic resampling schedule is used in which a resampling step is engaged if the effective sample size is smaller than $.1m$. The experiment is repeated 100 times, and the average RMSEs are reported.

We fix $\sigma^2 = .5^2$ and let δ^2 vary, so that δ controls the relative importance of the current observation and the state dynamics. When the same number of particles, $m = 500$, are used, we observed (detailed results not shown) that as L increases, MIPF performs better (confirming Prop. 2) with longer computational time, and that as δ decreases or as the information from the observations becomes stronger relative to that from the state dynamics, MIPF performs better than PF1, PF2, and APF. We also noticed that as L increases, the number of resampling steps in MIPF decreases (confirming Prop. 5); also, PF1 requires a significant amount of resampling, due partially to the fact that samples of $\mathbf{x}_t$ are generated with weak state dynamics.

For a fairer comparison, Table 1 reports the average RMSEs of the PF methods when the numbers of particles were chosen so that each method used approximately the same CPU time. It is seen that when the state dynamics is strong (i.e., large δ), the standard PF is very efficient, because of the ease of computation while enjoying the sufficient information contained in the state dynamics. In contrast, when the current observation contains more information (i.e., small δ), MIPF performs better than all of the other methods considered. In addition, we see that with the same computation time, L should not be too small or too large to achieve small RMSE. Therefore, in practice, L should be chosen considering the trade-off between estimation efficiency and computational effort, as suggested in Remark 1 in Section 2.2.

4.2 Example 2: Target Tracking With Maneuvering

Consider a two-dimensional maneuvering mobility model used by Ikoma et al. (2001),

$$\text{state equation:}\quad \begin{pmatrix}\mathbf{z}_t\\ \mathbf{v}_t\\ \mathbf{r}_t\end{pmatrix} = \mathbf{F}\begin{pmatrix}\mathbf{z}_{t-1}\\ \mathbf{v}_{t-1}\\ \mathbf{r}_{t-1}\end{pmatrix} + \mathbf{G}\boldsymbol{\varepsilon}_t;$$

$$\text{observation equation:}\quad \begin{pmatrix}y_{t,1}\\ y_{t,2}\end{pmatrix} = \begin{pmatrix}\tan^{-1}(z_{t,1}/z_{t,2})\\ \sqrt{z_{t,1}^2 + z_{t,2}^2}\end{pmatrix} + \delta\boldsymbol{\eta}_t,$$

where $\mathbf{z}_t$, $\mathbf{v}_t$, and $\mathbf{r}_t$ are the position, velocity, and acceleration of the target moving on a two-dimensional plane; $\boldsymbol{\varepsilon}_t$ is the random change of acceleration vector, and $\mathbf{F}$ and $\mathbf{G}$ are two known matrices (see Ikoma, Ichimura, Higuchi, and Maeda 2001) with $\alpha = 1{,}000$ (Zaidi and Mark 2003).

The state equation is a discretization of a continuous-time mobility model. Because the acceleration of the target may change abruptly with maneuvering, $\varepsilon_{t,i}$ $(i = 1, 2)$ are assumed to follow Cauchy distributions independently,

$$p(\varepsilon_{t,i}) = \frac{q}{\pi(\varepsilon_{t,i}^2 + q^2)}.$$

The observation vector consists of the angle, $y_{t,1}$, and the radius, $y_{t,2}$, of the target, with the observation noise following a Gaussian distribution,

$$\begin{pmatrix}\eta_{t,1}\\ \eta_{t,2}\end{pmatrix} \sim \mathrm{N}(\mathbf{0}, \mathbf{R}), \qquad \mathbf{R} = \begin{pmatrix}\delta_1^2 & 0\\ 0 & \delta_2^2\end{pmatrix}.$$

In comparing the PF methods, the RMSE in (17) is used, with $E(\mathbf{z}_t|\mathbf{Y}_t)$ estimated by a bootstrap filter with 100,000 particles. In the simulation, we set $T_0 = 3.75$, $T = 100$, and $q = 5$. Following Ikoma et al. (2001), we set $\delta_1^2 = 10^{-10}$ and $\delta_2^2 = 10^{-2}$. The prior distribution for the initial state of the target is $\mathrm{N}((50{,}000, -5{,}000, 0, 10, 0, 0)', \mathbf{I}_6)$. A dynamic resampling schedule is used, as in Example 1. The experiment is repeated 100 times.

Note that δ in the observation equation again controls the relative importance between the current observation and the state dynamic. For different values of δ, Table 2 reports the average RMSEs of the bootstrap filter (PF), APF, and MIPF-2 with different values of L, using approximately the same computation time. It reveals the same information as in Example 1. Also note that when $L = m$ in the MIPF, resampling actually increases the average RMSEs, which confirms Proposition 6.

4.3 Example 3: Nonlinear Filtering

Consider the following nonlinear state-space model (Gordon et al. 1993):

$$\text{state equation:}\quad x_t = .5x_{t-1} + \frac{25x_{t-1}}{1 + x_{t-1}^2} + 8\cos(1.2(t-1)) + \varepsilon_t; \qquad (18)$$

$$\text{observation equation:}\quad y_t = x_t^2/20 + \eta_t,$$

where $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$, and $\eta_t \sim \mathrm{N}(0, \delta^2)$ are Gaussian white noise. Because it is difficult to draw x_t directly from

$$f_t(y_t|x_t) \propto \exp\left\{-\frac{1}{2\delta^2}\left(\frac{x_t^2}{20} - y_t\right)^2\right\}, \qquad (19)$$

we sample x_t from a trial distribution that is close to (19) but easier to sample from. Specifically, we linearize $x_t^2/20$ at the points of x_t that maximize the likelihood $f_t(y_t|x_t)$ and set the sampling distribution as a mixture Gaussian distribution. When $y_t > 0$, the points of x_t that maximize $f_t(y_t|x_t)$ are $\overline{x}_t^{(k)} = \pm\sqrt{20y_t}$ $(k = 1, 2)$, and the first derivative of $x_t^2/20$ at

Table 1. For Different Values of δ in Example 1, Average RMSEs of PF1, PF2, APF, and MIPF With Different Values of L, and CPU Time

	δ					
	1	*2*	*4*	*8*	*16*	*CPU time (sec.)*
PF1 ($m = 8{,}000$)	.2669	.2823	.4138	.7983	1.3860	1.5470
PF2 ($m = 2{,}000$)	.1384	.2126	.5136	1.3229	2.6052	1.3400
APF ($m = 6{,}000$)	.6757	.8715	.9015	.5768	.6500	1.5470
MIPF ($L = 1$, $m = 4{,}000$)	.0460	.1189	.4027	1.4302	5.3018	1.3290
MIPF ($L = 5$, $m = 2{,}100$)	.0464	.1085	.3377	1.1560	4.0246	1.3260
MIPF ($L = 10$, $m = 1{,}400$)	.0474	.1112	.3407	1.1422	3.8801	1.4060
MIPF ($L = 20$, $m = 900$)	.0512	.1161	.3541	1.1689	3.9932	1.5470
MIPF ($L = 50$, $m = 400$)	.0739	.1653	.4596	1.4347	4.4516	1.5780
MIPF ($L = 100$, $m = 200$)	.1038	.2279	.6271	1.9387	5.4115	1.8910
MIPF ($L = 150$, $m = 150$)	.1185	.2665	.7297	2.1086	5.8297	1.8250

NOTE: The numbers of particles were chosen so that each method used approximately the same CPU time.

Table 2. For Different Values of δ in Example 2, Average RMSE of PF, APF, and MIPF With Different Values of L, and CPU Time

	δ					
	.625	1.25	2.5	5	10	CPU time (sec.)
PF ($m = 5{,}000$)	.6528	.5467	.6424	1.5377	2.4073	1.1090
APF ($m = 3{,}000$)	.2324	.3396	.4785	1.0696	1.4373	1.0780
MIPF ($L = 1$, $m = 2{,}400$)	.1068	.3076	.8389	2.1268	4.8148	1.0320
MIPF ($L = 5$, $m = 1{,}600$)	.0900	.2531	.6797	1.7708	4.4829	1.0310
MIPF ($L = 10$, $m = 1{,}200$)	.0952	.2580	.6361	1.7559	4.1170	1.1250
MIPF ($L = 20$, $m = 800$)	.0918	.2568	.6949	1.6992	3.8907	1.1410
MIPF ($L = 50$, $m = 400$)	.0877	.2413	.5957	1.5327	3.6006	.1250
MIPF ($L = 100$, $m = 200$)	.0784	.2398	.6304	1.6985	3.7970	1.1190
MIPF ($L = 150$, $m = 150$, r)	.0998	.2543	.6203	1.5282	3.5762	1.0940
MIPF ($L = 150$, $m = 150$, nr)	.0922	.2385	.6210	1.4980	3.4641	1.0780

NOTE: The last two rows are for MIPF with resampling and without resampling. The numbers of particles were chosen so that each method used approximately the same CPU time.

these points are $d_k = \bar{x}_t^{(k)}/10 = \pm\sqrt{y_t/5}$ $(k = 1, 2)$; thus we have

$$\frac{x_t^2}{20} \approx \frac{(\bar{x}_t^{(k)})^2}{20} + \big(x_t - \bar{x}_t^{(k)}\big)d_k = y_t + \big(x_t - \bar{x}_t^{(k)}\big)d_k.$$

Then

$$\begin{aligned} f_t(y_t|x_t) &\approx \widehat{f_t}(y_t|x_t) \\ &\propto .5\exp\left\{-\frac{1}{2\delta^2}\big(\big(x_t - \bar{x}_t^{(1)}\big)d_1\big)^2\right\} \\ &\quad + .5\exp\left\{-\frac{1}{2\delta^2}\big(\big(x_t - \bar{x}_t^{(2)}\big)d_2\big)^2\right\}. \end{aligned}$$

Consequently, we set the sampling distribution for x_t as

$$\begin{aligned} g_t^{(1)}(x_t) &\sim .5\mathrm{N}\big(\bar{x}_t^{(1)}, \delta^2/d_1^2\big) + .5\mathrm{N}\big(\bar{x}_t^{(2)}, \delta^2/d_2^2\big) \\ &= .5\mathrm{N}(\sqrt{20y_t}, 5\delta^2/y_t) + .5\mathrm{N}(-\sqrt{20y_t}, 5\delta^2/y_t), \end{aligned}$$

in which the maximum variance is set at $25\delta^2$ to avoid very large variances when y_t is close to 0. When $y_t \le 0$, we linearize the observation equation at $\bar{x}_t = 0$ and set the variance at $25\delta^2$. In summary, we have

$$g_t^{(1)}(x_t) \sim \begin{cases} .5\mathrm{N}(c, s^2) + .5\mathrm{N}(-c, s^2), & y_t > 0 \\ \mathrm{N}(0, 25\delta^2), & y_t \le 0, \end{cases}$$

where $c = \sqrt{20y_t}$ and $s^2 = \min(5\delta^2/y_t, 25\delta^2)$.

As described in Section 3.1, we can incorporate information from the past particles to improve our sampling distribution. Specifically, we first let the past particles $\{x_{t-1}^{(j)}, j = 1, \dots, m\}$ propagate to x_t, using the mean state dynamic in (18) without the noise. Then we try to "summarize" the propagated particles into a continuous distribution for x_t.

There are two different cases based on the past observation y_{t-1}. When $y_{t-1} > 0$, the past particles x_{t-1} were generated from a mixture distribution; as a result, they tend to propagate into a mixture distribution. Hence we attempt to summarize the propagated particles using a mixture distribution. Specifically, let

$$\xi(x_t) \sim p_1\mathrm{N}(\mu_1, \tau_1^2) + p_2\mathrm{N}(\mu_2, \tau_2^2),$$

where p_1 and p_2 are estimated mixing proportions that satisfy $p_1 + p_2 = 1$. Because every particle at time $t - 1$ is generated from one of the two mixture components, (13) is used separately for the two components to estimate μ_1, τ_1^2, and μ_2, τ_2^2. When $y_{t-1} \le 0$, x_{t-1} was generated from a Gaussian distribution, and, correspondingly, $\xi(x_t)$ is a Gaussian distribution: $\xi(x_t) \sim \mathrm{N}(\mu, \tau^2)$, using (13) to estimate μ and τ^2. After $\xi(x_t)$ is obtained, the sampling distribution at time t can be set as

$$g_t^{(2)}(x_t) \propto g_t^{(1)}(x_t)\xi(x_t),$$

which is a Gaussian distribution or a mixture Gaussian distribution.

In the simulation, we use the prior distribution $p(x_0) = \mathrm{N}(0, 2)$ and set $T = 50$ and $\sigma = \sqrt{10}$. A dynamic resampling schedule is used as in the previous examples, and the experiment is repeated 100 times. Again, δ controls the relative importance between the current observation and the state dynamic. For different values of δ, Table 3 reports the average RMSEs of the bootstrap filter (PF1), a PF using the sampling distribution $g_t(x_t|X_{t-1}, Y_t) \propto q_t(x_t|x_{t-1})g_t^{(1)}(x_t)$ to approximate (4) (PF2), an IPF with $g_t^{(1)}(x_t)$ as the sampling distribution (IPF1), and an IPF with $g_t^{(2)}(x_t)$ as the sampling distribution (IPF2), using approximately the same computational time. For each IPF method, Table 3 shows similar results as the previous examples. In addition, it is also seen that when L becomes large and/or δ becomes large, IPF2 performs slightly better than IPF1, due to the incorporation of information from x_{t-1}.

APPENDIX: PROOFS

Proof of Proposition 1

The proof is trivial.

Proof of Proposition 2

Because of exchangeability, we have the following constants:

$$\begin{aligned} C_1 &= \mathrm{var}\big[\lambda_t^{(i,j)}h\big(\mathbf{x}_t^{(j)}\big)\big]; \\ C_2 &= \mathrm{cov}\big[\lambda_t^{(i_1,j)}h\big(\mathbf{x}_t^{(j)}\big), \lambda_t^{(i_2,j)}h\big(\mathbf{x}_t^{(j)}\big)\big], \qquad i_1 \ne i_2; \\ C_3 &= \mathrm{cov}\big[\lambda_t^{(i,j_1)}h\big(\mathbf{x}_t^{(j_1)}\big), \lambda_t^{(i,j_2)}h\big(\mathbf{x}_t^{(j_2)}\big)\big], \qquad j_1 \ne j_2; \end{aligned}$$

and

$$C_4 = \mathrm{cov}\big[\lambda_t^{(i_1,j_1)}h\big(\mathbf{x}_t^{(j_1)}\big), \lambda_t^{(i_2,j_2)}h\big(\mathbf{x}_t^{(j_2)}\big)\big], \qquad i_1 \ne i_2,\ j_1 \ne j_2.$$

Table 3. For Different Values of δ in Example 3, Average RMSE for PF1, PF2, and IPF1 and IPF2 With Different Values of L, and CPU Time

	δ				
	1/8	1/4	1/2	1	CPU time (sec.)
PF1 ($m = 5{,}000$)	.6676	.4767	.3263	.3062	.3170
PF2 ($m = 1{,}200$)	.3672	.3407	.3401	.3371	.3180
IPF1 ($L = 1$, $m = 2{,}600$)	.2879	.2848	.3193	.3921	.3130
IPF1 ($L = 5$, $m = 1{,}300$)	.2697	.2784	.3093	.3713	.3120
IPF1 ($L = 10$, $m = 850$)	.3077	.3134	.3554	.3852	.3440
IPF1 ($L = 20$, $m = 500$)	.3639	.3441	.3741	.4467	.3290
IPF1 ($L = 100$, $m = 100$)	.6595	.6957	.7644	.9021	.3120
IPF2 ($L = 1$, $m = 2{,}000$)	.3130	.3281	.2996	.3856	.3280
IPF2 ($L = 5$, $m = 1{,}100$)	.2880	.3009	.3006	.3794	.3120
IPF2 ($L = 10$, $m = 750$)	.3342	.3068	.3117	.3692	.3280
IPF2 ($L = 20$, $m = 500$)	.3814	.3510	.3638	.4199	.3440
IPF2 ($L = 100$, $m = 100$)	.6381	.6359	.6661	.8270	.3280

NOTE: The number of particles were chosen so that each method used approximately the same CPU time.

For $i_1 \neq i_2, j_1 \neq j_2$, we have

$$\operatorname{var}\left[\frac{(\lambda_t^{(i_1,j_1)}+\lambda_t^{(i_2,j_1)})h(\mathbf{x}_t^{(j_1)})+(\lambda_t^{(i_1,j_2)}+\lambda_t^{(i_2,j_2)})h(\mathbf{x}_t^{(j_2)})}{2}\right] \le \operatorname{var}\big[\lambda_t^{(i_1,j_1)}h\big(\mathbf{x}_t^{(j_1)}\big)+\lambda_t^{(i_2,j_2)}h\big(\mathbf{x}_t^{(j_2)}\big)\big].$$

This gives $C_1 + C_2 + C_3 + C_4 \le 2C_1 + 2C_4$, hence

$$C_1 + C_4 - C_2 - C_3 \ge 0. \qquad \text{(A.1)}$$

Define $a_{t,l} = m^{-1}\sum_{j=1}^m w_{t,l}^{(j)} h(\mathbf{x}_t^{(j)})$. Consider any pair of permutations, $\mathbf{K}_{l_1}$ and $\mathbf{K}_{l_2}$ ($l_1 \neq l_2$). Suppose that there are m_1 j's that satisfy $k_{l_1,j} = k_{l_2,j}$ and m_2 j's that satisfy $k_{l_1,j} \neq k_{l_2,j}$ ($m_1 + m_2 = m$). Using (A.1), we can easily show that when $m_1 = 0$, $\operatorname{cov}(a_{t,l_1}, a_{t,l_2})$ is minimized, and

$$\operatorname{cov}\big(a_{t,l_1}, a_{t,l_2}\big) = \frac{C_2 + C_3 + (m-2)C_4}{m}. \qquad \text{(A.2)}$$

At the same time, for $l = 1, \ldots, L$, $\operatorname{var}(a_{t,l}) = m^{-1}[C_1 + (m-1)C_4]$. Because $A_{t,L} = \sum_{l=1}^L a_{t,l}/L$, $\operatorname{var}(A_{t,L})$ is minimized when for any $1 \le l_1 \neq l_2 \le L$, $k_{l_1,j} \neq k_{l_2,j}$, $j = 1, \ldots, m$. This proves part (a).

With mutually exclusive permutations, by plugging (A.2) into the formula of $\operatorname{var}(A_{t,L})$ and using (A.1), we get that $\operatorname{var}(A_{t,L})$ decreases as L increases. This proves part (b).

By setting $h(\mathbf{x}_t) = 1$, we can prove similar conclusions for $\operatorname{var}(B_{t,L})$.

Proof of Proposition 3

According to (9), for any $L > m$ permutations,

$$A_{t,L} = \frac{1}{m}\sum_{l=1}^{L}\sum_{j=1}^{m}\frac{1}{L}\lambda_t^{(k_{l,j},j)}h\big(\mathbf{x}_t^{(j)}\big) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} d_{i,j}\lambda_t^{(i,j)}h\big(\mathbf{x}_t^{(j)}\big),$$

where $d_{i,j}$ satisfy

$$\sum_{i=1}^{m}\sum_{j=1}^{m} d_{i,j} = m, \qquad d_{i,j} \ge 0. \qquad \text{(A.3)}$$

The Lagrangian expression of minimizing $\operatorname{var}(A_{t,L})$ is

$$\begin{aligned} J &= \operatorname{var}(A_{t,L}) + \mu\left(\sum_{i=1}^{m}\sum_{j=1}^{m} d_{i,j} - m\right) \\ &= \frac{1}{m^2}\Bigg\{C_1\sum_{i=1}^{m}\sum_{j=1}^{m} d_{i,j}^2 + C_2\sum_{1\le i_1\neq i_2\le m}\sum_{j=1}^{m} d_{i_1,j}d_{i_2,j} \\ &\quad + C_3\sum_{i=1}^{m}\sum_{1\le j_1\neq j_2\le m} d_{i,j_1}d_{i,j_2} \\ &\quad + C_4\sum_{1\le i_1\neq i_2\le m}\sum_{1\le j_1\neq j_2\le m} d_{i_1,j_1}d_{i_2,j_2}\Bigg\} \\ &\quad + \mu\left(\sum_{i=1}^{m}\sum_{j=1}^{m} d_{i,j} - m\right), \end{aligned}$$

where C_1, C_2, C_3, and C_4 are as defined in the proof of Proposition 2. Setting the derivative of J with respect to a particular d_{i_0,j_0} to 0, we get, for any i_0 and j_0,

$$(C_1 - C_2 - C_3 + C_4)d_{i_0,j_0} + (C_2 - C_4)\sum_i d_{i,j_0} + (C_3 - C_4)\sum_j d_{i_0,j} + C_4 m = -.5\mu m^2. \qquad \text{(A.4)}$$

By summing (A.4) over i_0, j_0, and (i_0, j_0) jointly, we find that d_{i_0,j_0} is constant for different i_0 and j_0. By (A.3), $\operatorname{var}(A_{t,L})$ achieves its minimum when $d_{i_0,j_0} = 1/m$, and thus $A_{t,L}$ equals $A_{t,m}$ for m mutually exclusive permutations. Because when $h(\mathbf{x}_t) = 1$, $A_{t,L} = B_{t,L}$, similar conclusion holds for $\operatorname{var}(B_{t,L})$.

Proof of Proposition 4

It can be seen by

$$\begin{aligned} \operatorname{var}_{g^*}\left[h(\mathbf{x})\frac{\pi(\mathbf{x},\mathbf{z})}{g^*(\mathbf{x},\mathbf{z})}\right] &\ge \operatorname{var}_g\left[E_{g^*}\left\{h(\mathbf{x})\frac{\pi(\mathbf{x},\mathbf{z})}{g^*(\mathbf{x},\mathbf{z})}\Big|\mathbf{x}\right\}\right] \\ &= \operatorname{var}_g\left[h(\mathbf{x})\int\frac{\pi(\mathbf{x},\mathbf{z})}{g^*(\mathbf{x},\mathbf{z})}\frac{g^*(\mathbf{x},\mathbf{z})}{g(\mathbf{x})}\,d\mathbf{z}\right] \\ &= \operatorname{var}_g\left[h(\mathbf{x})\frac{\pi(\mathbf{x})}{g(\mathbf{x})}\right]. \end{aligned}$$

Proof of Proposition 5

According to (8), $E(w_t^{(i)}) = \mathcal{B}_t$ is a constant for different L's. Because of exchangeability, we have two constants, $D_1 = \operatorname{var}(\lambda_t^{(i,j)})$ and $D_2 = \operatorname{cov}(\lambda_t^{(i_1,j)}, \lambda_t^{(i_2,j)})$, $i_1 \neq i_2$:

$$\operatorname{var}\big(\lambda_t^{(i_1,j)} - \lambda_t^{(i_2,j)}\big) = 2D_1 - 2D_2 \ge 0 \quad \Longrightarrow \quad D_1 \ge D_2$$

and

$$\operatorname{var}\big(w_t^{(j)}\big) = \operatorname{var}\bigg(\frac{\sum_{l=1}^{L} \lambda_t^{(k_{l,j},j)}}{L}\bigg)$$
$$= \frac{LD_1 + L(L-1)D_2}{L^2} = \frac{D_1 - D_2}{L} + D_2,$$

which decrease as L increases.

Proof of Proposition 6

A complete matching ($L = m$) between $\{\mathbf{x}_t^{(j)},\ j = 1, \dots, m\}$ and the resampled set $\{\mathbf{x}_{t-1}^{*(i)},\ i = 1, \dots, m\}$ results in the new weight (under $w_{t-1}^{*(i)} = 1$),

$$w_t^{*(j)} = \frac{\sum_{l=1}^{m} w_{t,l}^{*(j)}}{m} = \frac{\sum_{i=1}^{m} \lambda_t^{*(i,j)}}{m}$$
$$= \frac{1}{m} \sum_{i=1}^{m} \frac{q_t(\mathbf{x}_t^{(j)} | \mathbf{x}_{t-1}^{*(i)}) f_t(\mathbf{y}_t | \mathbf{x}_t^{(j)})}{g_t(\mathbf{x}_t^{(j)} | \mathbf{Y}_t)}. \qquad \text{(A.5)}$$

Recall that

$$A_{t,m}^* = \frac{1}{m} \sum_{j=1}^{m} w_t^{*(j)} h\big(\mathbf{x}_t^{(j)}\big) \qquad \text{and} \qquad B_{t,m}^* = \frac{1}{m} \sum_{j=1}^{m} w_t^{*(j)}. \qquad \text{(A.6)}$$

Because the resampling probability for $\mathbf{x}_{t-1}^{(k')}$ is $w_{t-1}^{(k')}/m$, we have

$$E\big[\lambda_t^{*(i,j)} \big| \{\mathbf{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\}_{k=1}^{m}, \mathbf{x}_t^{(j)}\big]$$
$$= \sum_{k'=1}^{m} \Bigg[\frac{w_{t-1}^{(k')}}{m} \frac{q_t(\mathbf{x}_t^{(j)} | \mathbf{x}_{t-1}^{(k')}) f_t(\mathbf{y}_t | \mathbf{x}_t^{(j)})}{g_t(\mathbf{x}_t^{(j)} | \mathbf{Y}_t)}\Bigg]$$
$$= \frac{\sum_{k'=1}^{m} \lambda_t^{(k',j)}}{m} = w_t^{(j)}. \qquad \text{(A.7)}$$

From (A.5) and (A.7), we have

$$E\big[w_t^{*(j)} \big| \{\mathbf{x}_{t-1}^{(k)}, w_{t-1}^{(k)}\}_{k=1}^{m}, \mathbf{x}_t^{(j)}\big] = w_t^{(j)}. \qquad \text{(A.8)}$$

According to (A.6) and (A.8),

$$E\big(A_{t,m}^* \big| \{\mathbf{x}_{t-1}^{(j)}, w_{t-1}^{(j)}, \mathbf{x}_t^{(j)}\}_{j=1}^{m}\big) = A_{t,m} \qquad \text{and}$$
$$E\big(B_{t,m}^* \big| \{\mathbf{x}_{t-1}^{(j)}, w_{t-1}^{(j)}, \mathbf{x}_t^{(j)}\}_{j=1}^{m}\big) = B_{t,m}.$$

So we have

$$E(A_{t,m}^*) = E(A_{t,m}), \qquad E(B_{t,m}^*) = E(B_{t,m}),$$
$$\operatorname{var}(A_{t,m}^*) \geq \operatorname{var}(A_{t,m}), \qquad \operatorname{var}(B_{t,m}^*) \geq \operatorname{var}(B_{t,m}).$$

Proof of Proposition 7

The proof is similar to the proofs for Propositions 1–3, 5, and 6.

[Received August 2003. Revised February 2005.]

REFERENCES

Avitzour, D. (1995), "A Stochastic Simulation Bayesian Approach to Multitarget Tracking," *IEE Proceedings on Radar, Sonar and Navigation*, 142, 41–44.

Bar-Shalom, Y., and Fortmann, T. (1988), *Tracking and Data Association*, Boston: Academic Press.

Blake, A., Bascle, B., Isard, M., and MacCormick, J. (1998), "Statistical Models of Visual Shape and Motion," *Philosophical Transactions of the Royal Society*, Ser. A, 356, 1283–1302.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999), "Improved Particle Filter for Nonlinear Problems," *IEE Proceedings on Radar, Sonar, and Navigation*, 146, 2–7.

Chen, R., and Liu, J. S. (2000), "Mixture Kalman Filters," *Journal of the Royal Statistical Society*, Ser. B, 62, 493–508.

Chen, R., Wang, X., and Liu, J. (2000), "Adaptive Joint Detection and Decoding in Flat-Fading Channels via Mixture Kalman Filtering," *IEEE Transactions on Information Theory*, 46, 2079–2094.

Chopin, N. (2002), "A Sequential Particle Filter Method for Static Models," *Biometrica*, 89, 539–552.

Doucet, A., Gordon, N. J., and Krishnamurthy, V. (2001), "Particle Filters for State Estimation of Jump Markov Linear Systems," *IEEE Transactions on Signal Processing*, 49, 613–624.

Durbin, J., and Koopman, S. (1997), "Monte Carlo Maximum Likelihood Estimation for Non-Gaussian State-Space Models," *Biometrika*, 84, 669–684.

Durham, G., and Gallant, A. R. (2002), "Numerical Techniques for Maximum Likelihood Estimation of Continuous-Time Diffusion," *Journal of Business & Economic Statistics*, 20, 297–338.

Fong, W., Godsill, S., Doucet, A., and West, M. (2002), "Monte Carlo Smoothing With Application to Audio Signal Enhancement," *IEEE Transactions on Signal Processing*, 50, 438–449.

Fox, D., Thrun, S., Burgar, W., and Dellaert, F. (2001), "Particle Filters for Mobile Robot Localization," in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. de Freitas, and N. Gordon, New York: Springer-Verlag, pp. 401–428.

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

Grassberger, P. (1997), "Pruned-Enriched Rosenbluth Method: Simulation of θ Polymers of Chain Length up to 1,000,000," *Physics Review*, Ser. E, 56, 3682–3693.

Hendry, D., and Richard, J.-F. (1991), "Likelihood Evaluation for Dynamic Latent Variable Models," in *Computational Economics and Econometrics*, eds. H. Amman, D. Belsley, and L. Pau, Dordrecht: Kluwer, pp. 3–17.

Hue, C., Le Cadre, J.-P., and Perez, P. (2002), "Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion," *IEEE Transactions on Signal Processing*, 50, 309–315.

Ikoma, N., Ichimura, N., Higuchi, T., and Maeda, H. (2001), "Maneuvering Target Tracking by Using Particle Filter," *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 4, 2223–2228.

Isard, M., and Blake, A. (1996), "Contour Tracking by Stochastic Propagation of Conditional Density," in *Computer Vision—ECCV'96*, eds. B. Buxton and R. Cipolla, New York: Springer-Verlag, pp. 343–356.

——— (2004), "ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Framework," technical report, Oxford University, available at *http://robots.ox.ac.uk/~ab*.

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

Kong, A., Liu, J. S., and Wong, W. H. (1994), "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, 90, 567–576.

Kotecha, J., and Djuric, P. (2003), "Gaussian Sum Particle Filters," *IEEE Transactions on Signal Processing*, 51, 2602–2612.

Liang, J., Chen, R., and Zhang, J. (2002), "On Statistical Geometry of Packing Defects of Lattice Chain Polymer," *Journal of Chemical Physics*, 117, 3511–3521.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag.

Liu, J. S., and Chen, R. (1995), "Blind Deconvolution via Sequential Imputations," *Journal of the American Statistical Association*, 90, 567–576.

——— (1998), "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 93, 1032–1044.

McGinnity, S., and Irwin, G. (2001), "Manoeuvring Target Tracking Using a Multiple-Model Bootstrap Filter," in *Sequential Monte Carlo in Practice*, eds. A. Doucet, J. F. G. de Freitas, and N. J. Gordon, New York: Springer-Verlag, pp. 479–498.

Pitt, M. K., and Shephard, N. (1999), "Filtering via Simulation: Auxiliary Particle Filters," *Journal of the American Statistical Association*, 94, 590–599.

Rossi, G. D. (2004), "The Two-Factor Cox–Ingersoll–Ross Model as a Self-Organizing State Space," working paper.

Salmond, D., and Gordon, N. (2001), "Particles and Mixtures for Tracking and Guidance," in *Sequential Monte Carlo in Practice*, eds. A. Doucet, J. F. G. de Freitas, and N. J. Gordon, New York: Springer-Verlag, pp. 517–532.

Torma, P., and Szepesvari, P. (2003), "Sequential Importance Sampling for Visual Tracking Reconsidered," in *Proceedings of the AI and Statistics*, pp. 198–205.

van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E. (2002), "The Unscented Particle Filter," in *Advances in Neural Information Processing Systems (NIPS13)*, eds. T. K. Leen, T. G. Dietterich, and V. Tresp, Cambridge, MA: MIT Press.

Vasquez, M., and Scheraga, H. (1985), "Use of Build-up and Energy-Minimization Procedures to Compute Low-Energy Structures of the Backbone of Enkephalin," *Biopolymers*, 24, 1437–1447.

Wang, X., Chen, R., and Guo, D. (2002), "Delayed Pilot Sampling for Mixture Kalman Filter With Application in Fading Channels," *IEEE Transactions on Signal Processing*, 50, 241–264.

Zaidi, Z. R., and Mark, B. L. (2003), "A Mobility Tracking Model for Wireless ad hoc Networks," *IEEE Wireless Communication and Networking Conference (WCNC'2003)*, 3, 1790–1795.

Zhang, J., Chen, R., Tang, C., and Liang, J. (2003), "Origin of Scaling Behavior of Protein Packing Density: A Sequential Monte Carlo Study of Compact Long-Chain Polymer," *Journal of Chemical Physics*, 118, 6102–6109.

Zhang, J. L., and Liu, J. S. (2002), "A New Sequential Importance Sampling Method With Application to the 2D Hydrophobic-Hydrophilic Model," *Journal of Chemical Physics*, 117, 3492–3498.