Statistics and Computing **15**: 135–147, 2005
© 2005 Springer Science + Business Media, Inc. *Manufactured in The Netherlands.*

*New sequential Monte Carlo methods for nonlinear dynamic systems**

DONG GUO[†], XIAODONG WANG[†] and RONG CHEN[‡,§]

[†]*Department of Electrical Engineering, Columbia University, New York, NY 10027.*
[‡]*Department of Information and Decision Science, University of Illinois at Chicago, Chicago, IL 60607*
[§]*Department of Business Statistics and Econometrics, Peking University, Beijing, China*
rongchen@uic.edu

Received December 2002 and accepted December 2004

In this paper we present several new sequential Monte Carlo (SMC) algorithms for online estimation (filtering) of nonlinear dynamic systems. SMC has been shown to be a powerful tool for dealing with complex dynamic systems. It sequentially generates Monte Carlo samples from a *proposal distribution*, adjusted by a set of importance weight with respect to a target distribution, to facilitate statistical inferences on the characteristic (state) of the system. The key to a successful implementation of SMC in complex problems is the design of an efficient proposal distribution from which the Monte Carlo samples are generated. We propose several such proposal distributions that are efficient yet easy to generate samples from. They are efficient because they tend to utilize both the information in the state process and the observations. They are all Gaussian distributions hence are easy to sample from. The central ideas of the conventional nonlinear filters, such as extended Kalman filter, unscented Kalman filter and the Gaussian quadrature filter, are used to construct these proposal distributions. The effectiveness of the proposed algorithms are demonstrated through two applications—real time target tracking and the multiuser parameter tracking in CDMA communication systems.

Keywords: bayesian inference, sequential Monte Carlo, kernel representation, nonlinear dynamic system

1. Introduction

Consider the following nonlinear dynamic system described by a state-space model

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}) + \boldsymbol{u}_t, \tag{1}$$

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{v}_t, \tag{2}$$

$$t = 0, 1, \ldots,$$

where (1) is the state equation, with $\boldsymbol{f}(\cdot)$ being the nonlinear state transition function, and the state noise $\boldsymbol{u}_t \sim \mathcal{N}(0, \boldsymbol{Q})$ being white and stationary; and where (2) is the measurement equation, with $\boldsymbol{h}(\cdot)$ being the nonlinear measurement function, and the measurement noise $\boldsymbol{v}_t \sim \mathcal{N}(0, R)$ being white and stationary, and independent of the state noise $\boldsymbol{u}_t$. Such nonlinear dynamic systems arise frequently from many areas in science and engineering, such as communications, radar tracking, sonar ranging, and satellite or airplane orbit tracking.

In general, a statistical inference problem for the above nonlinear dynamic system involves computing the expected value of some function $\psi(\cdot)$ of the state variable $\boldsymbol{x}_t$ conditioned on the observations $\boldsymbol{y}_{1:t} \triangleq (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t)$ up to time t, i.e., $E\{\psi(\boldsymbol{x}_t) \mid \boldsymbol{y}_{1:t}\}$. Starting with $p(\boldsymbol{x}_0) \triangleq p(\boldsymbol{x}_0 \mid \boldsymbol{y}_0)$ as the prior for $\boldsymbol{x}_0$, the posterior density $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t})$ can be obtained for $t = 1, 2, \ldots$ by using the Bayes rule as follows

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\, p(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{1:t-1})\, \mathrm{d}\boldsymbol{x}_{t-1},$$

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}) = \frac{1}{C}\, p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{y}_{1:t-1})\, p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1}),$$

$$\text{with} \quad C \triangleq \int p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{y}_{1:t-1})\, p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1})\, \mathrm{d}\boldsymbol{x}_t.$$

However, this solution is often analytically intractable due to nonlinearity.

*This work was supported in part by the U.S. National Science Foundation (NSF) under grants CCR-9875314, CCR-9980599, DMS-9982846, DMS-0073651 and DMS-0073601.

The classical inference methods for nonlinear dynamic systems are the extended Kalman filter (EKF) and its variants, which are based on linearization along the trajectories. The EKF has been successfully applied to numerous nonlinear problems. However, if the system exhibits severe nonlinearity, the performance of the EKF may not be satisfactory. Thus many algorithms have been developed to improve upon the EKF performance. Most of these algorithms fall into two categories, model-based algorithms and nonlinear Kalman filters. In model-based algorithms, the posterior density is approximated by an expansion using certain basis functions. For example, the Gaussian sum filter (Alspace and Sorenson 1972) approximates the posterior density by a mixture of Gaussians; whereas the reduced sufficient statistics algorithm (Kulhavý 1990, Agate and Iltis 1999) approximates the posterior density by the well-known Haar basis function under the wavelet framework. On the other hand, nonlinear Kalman filter algorithms attempt to choose a set of deterministic points to capture the posterior mean and the covariance accurately. Two representative algorithms in this class are the unscented Kalman filter (UKF) (Julier and Durrant-Whyte 2000, Merwe *et al.*, 2000) and the Gaussian quadrature Kalman filter (QKF) (Ito and Xiong 2000). The UKF is based on the so-called "sigma points", and the QKF is based on the Gauss-Hermite quadrature integration rule. One of the significant advantages of these algorithms is that they don't need the calculation of the Jacobian matrix, often a computational intensive component in the EKF.

Recently, the sequential Monte Carlo (SMC) methodology (Arulampalam *et al.* 2002, Chen and Liu 2000, 2001, Doucet 1999, Doucet *et al.* 2001, Liu 2001, Liu and Chen 1998, Pitt and Shephard 1999, Shephard and Pitt 1997) emerged in the fields of statistics and engineering, has shown a great promise in solving a wide class of nonlinear filtering problems. The SMC method is also named the condensation algorithm in computer vision (Dellaert *et al.* 1999; Isard and Blake 1996, 1997, 1998, Thrum *et al.* 2001). The SMC method uses Monte Carlo simulation to solve on-line estimation problems in dynamic systems. By recursively generating Monte Carlo samples of the state variables or some other latent variables, these methods can easily adapt to the dynamics of the underlying stochastic systems. Although it can be shown that the resulting sample density approaches the true posterior density as the Monte Carlo sample size tends to infinity, the performance/efficiency of the SMC methods with finite Monte Carlo sample size depends largely on the proposal distribution. The proposal distribution in Gordon, Salmon and Ewing (1995) makes use of only the state equation, without exploiting the new measurement. In Liu and Chen (1998), a better proposal distribution is recommended. It fully utilizes the information in both the state process and the observation. However, generating samples from it is often difficult due to the nonlinear nature of the system.

In this paper we develop several new sequential Monte Carlo (SMC) algorithms. Specifically we propose several proposal distributions within the SMC framework. They are efficient because they tend to utilize both the information in the state process and the observations. They are also easy to sample from since all of them are Gaussian distributions. The central ideas of the conventional nonlinear filters such as EKF, UKF and QKF are used to construct these proposal distributions.

The remainder of the paper is organized as follows. Section 2 briefly reviews the SMC algorithm and the role of its proposal distribution. In Section 3, we present several new proposal distributions to be used in the SMC framework. In Section 4, we demonstrate the effectiveness of the proposed algorithms through two applications—real time target tracking and the multiuser parameter tracking in CDMA communication systems.

2. Background

2.1. *The general SMC algorithm*

Consider the dynamic system given by (1) and (2). Denote $\boldsymbol{x}_{1:t} \triangleq (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ and $\boldsymbol{y}_{1:t} \triangleq (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$. Suppose an on-line inference of $\boldsymbol{x}_{1:t}$ is of interest; that is, at current time t we wish to make a timely estimate of a function of the state variable $\boldsymbol{x}_{1:t}$, say $\psi(\boldsymbol{x}_{1:t})$, based on the currently available observation, $\boldsymbol{y}_{1:t}$. The optimal solution (in the minimum mean-square error sense) to this problem is $E\{\psi(\boldsymbol{x}_{1:t}) \mid \boldsymbol{y}_{1:t}\} = \int \psi(\boldsymbol{x}_{1:t}) p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}) d\boldsymbol{x}_{1:t}$. In most cases, an exact evaluation of this expectation is often analytically intractable because of the high complexity of such a dynamic system. Sequential Monte Carlo methods, which are based on importance sampling, provide us with viable approaches to the required estimation.

The basic idea of importance sampling is to draw m random samples $\{\boldsymbol{x}_{1:t}^{(j)}\}_{j=1}^{m}$ from some *proposal* distribution $q(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t})$ with the same support as $p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t})$. By associating a weight $w_t^{(j)} = p(\boldsymbol{x}_{1:t}^{(j)} \mid \boldsymbol{y}_{1:t}) / q(\boldsymbol{x}_{1:t}^{(j)} \mid \boldsymbol{y}_{1:t})$ to the sample $\boldsymbol{x}_{1:t}^{(j)}$ in j-th Markov stream ($j = 1, \ldots, m$), we can approximate the quantity of interest, $E\{\psi(\boldsymbol{x}_{1:t}) \mid \boldsymbol{y}_{1:t}\}$, by

$$E_p\{\psi(\boldsymbol{x}_{1:t}) \mid \boldsymbol{y}_{1:t}\} \cong \frac{1}{W_t} \sum_{j=1}^{m} w_t^{(j)} \psi\left(\boldsymbol{x}_{1:t}^{(j)}\right),$$

with $W_t \triangleq \sum_{j=1}^{m} w_t^{(j)}$. The pair $(\boldsymbol{x}_{1:t}^{(j)}, w_t^{(j)})$, $j = 1, \ldots, m$, is called a *properly weighted sample* with respect to the target distribution $p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t})$.

To implement an online estimation, a set of random samples properly weighted with respect to $p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t})$ are needed at every time t. The Markovian structure of the system allows us to implement a recursive importance sampling strategy. Suppose a set of properly weighted samples $\{(\boldsymbol{x}_{1:t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^{m}$ with respect to $p(\boldsymbol{x}_{1:t-1} \mid \boldsymbol{y}_{1:t-1})$ are available at time $(t-1)$. Although $p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t})$ is often a probability density from which it is difficult to draw samples $\{(\boldsymbol{x}_{1:t}^{(j)})\}_{j=1}^{m}$, they can be drawn from an factorized importance density $q(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t})$ given by

$$q(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}) = q(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t}) q(\boldsymbol{x}_{1:t-1} \mid \boldsymbol{y}_{1:t-1}), \qquad (3)$$

with associated weights

$$w_t^{(j)} \propto \frac{p\big(\boldsymbol{x}_{1:t}^{(j)} \,\big|\, \boldsymbol{y}_{1:t}\big)}{q\big(\boldsymbol{x}_{1:t}^{(j)} \,\big|\, \boldsymbol{y}_{1:t}\big)} \propto w_{t-1}^{j} \frac{p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_{1:t}) p\big(\boldsymbol{x}_t^{(j)} \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}\big)}{q\big(\boldsymbol{x}_t^{(j)} \,\big|\, \boldsymbol{y}_t\big)}. \qquad (4)$$

Then the sequential importance sampling procedure can be summarized as the following procedure (Chen and Liu 2000, Kong, Liu and Wang 1994, Liu and Chen 1995, 1998).

For $j = 1, \dots, m$,

- Draw a sample $\boldsymbol{x}_t^{(j)}$ from a proposal distribution $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t})$ and let $\boldsymbol{x}_{1:t}^{(j)} = (\boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{x}_t^{(j)})$;
- Compute the importance weight $w_t^{(j)}$ by (4).
- Do resampling if the effective sample size $\bar{m}_t$ is below a specified threshold:
  - Sample a new set of streams $\{\hat{\boldsymbol{x}}_t^{(j)}\}_{j=1}^m$ from $\{\boldsymbol{x}_t^{(j)}\}_{j=1}^m$ with probability proportional to the importance weights $\{w_t^{(j)}\}_{j=1}^m$;
  - Assign equal weight to each stream in $\{\hat{x}_t^{(j)}\}$, i.e., $\hat{w}_t^{(j)} = \frac{1}{m}, j = 1, \dots, m$.

The algorithm is initialized by drawing a set of i.i.d. samples $\boldsymbol{x}_0^{(1)}, \dots, \boldsymbol{x}_0^{(m)}$ from $p(\boldsymbol{x}_0)$, which corresponds to the prior distribution of $\boldsymbol{x}_0$. In resampling, the effective sample size $\bar{m}_t$ is defined as Liu and Chen (1998)

$$\bar{m}_t \triangleq \frac{m}{1 + v_t^2},$$

where v_t is called the coefficient of variation given by

$$v_t^2 = \frac{1}{m} \sum_{j=1}^{m} \left(\frac{w_t^{(j)}}{\bar{w}_t} - 1 \right)^2,$$

with $\bar{w}_t = \frac{1}{m} \sum_{j=1}^m w_t^{(j)}$.

The resampling step is an important component of the SMC algorithm. Heuristically, resampling can provide chances for good sample streams to amplify themselves and hence "rejuvenate" the sampler to produce a better result for future states as system evolves. It can be shown that the samples drawn by the above resampling procedure are also indeed properly weighted with respect to $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t})$, provided that m is sufficiently large (Doucet 1999, Chen and Liu 2000, Doucet *et al.* 2000, Liu and Chen 1998). In practice, when small to modest m is used (we use $m = 100$ in this paper), the resampling procedure can be seen as a trade-off between the bias and the variance. That is, the new samples with their weights resulting from the resampling procedure are only approximately proper, which introduces small bias in the Monte Carlo estimation. On the other hand, however, resampling significantly reduces the Monte Carlo variance for future samples.

2.2. *The proposal distribution*

In the SMC algorithm, the choice of the proposal distribution $q(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t})$ is directly related to the efficiency of the algorithm. A straightforward choice of proposal distribution $q(\cdot)$ is based on the state equation (1)

$$q\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}\big) = p\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t-1}\big) = p\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{t-1}^{(j)}\big), \quad (5)$$

with weight

$$w_t^{(j)} = w_{t-1}^{(j)} p\big(\boldsymbol{y}_t \,\big|\, \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{1:t-1}\big) = w_{t-1}^{(j)} p\big(\boldsymbol{y}_t \,\big|\, \boldsymbol{x}_t^{(j)}\big),$$

where the last equality follows from the observation equation (2). This class of SMC algorithms is also called bootstrap filter or SIR particle filter (Doucet *et al.* 2001; Gordon, Salmon and Ewing 1995). Such an approach is easy to implement, since it is easy to generate samples from the Gaussian state transition density $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)})$ and to compute the weights based on the Gaussian observation density $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$. However, often times such a proposal distribution is not efficient since it does not exploit the information in the most recent observation $\boldsymbol{y}_t$. Note that the information on the current state $\boldsymbol{x}_t$ comes from two sources: $\boldsymbol{x}_{t-1}$ through the state equation (1) and $\boldsymbol{y}_t$ through the observation equation (2). When the information in $\boldsymbol{y}_t$ is strong, it is important to exploit it in generating the importance samples to gain efficiency.

To fully exploit the information in both $\boldsymbol{X}_{t-1}$ and $\boldsymbol{Y}_t$, it is suggested in Chen and Liu (2000), Doucet (1999), Doucet *et al.* (2001), Kong, Liu and Wong (1994), and Liu and Chen (1998) to use the following proposal distribution

$$q\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}\big) = p\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}\big) = p\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t\big), \quad (6)$$

with the weight update given by

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} p\big(\boldsymbol{y}_t \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t-1}\big) = w_{t-1}^{(j)} p\big(\boldsymbol{y}_t \,\big|\, \boldsymbol{x}_{t-1}^{(j)}\big) \\ &= w_{t-1}^{(j)} \int p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t) p\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{t-1}^{(j)}\big) \, \mathrm{d}\boldsymbol{x}_t. \end{aligned}$$

The advantage of the above proposal distribution can be seen by rewriting the weight expression (4) as

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{p\big(\boldsymbol{x}_{1:t-1}^{(j)} \,\big|\, \boldsymbol{y}_{1:t}\big)}{p\big(\boldsymbol{x}_{1:t-1}^{(j)} \,\big|\, \boldsymbol{y}_{1:t-1}\big)} \frac{p\big(\boldsymbol{x}_t^{(j)} \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}\big)}{q\big(\boldsymbol{x}_t^{(j)} \,\big|\, \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}\big)}.$$

Intuitively, the second ratio is needed to correct the discrepancy between $p(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t})$ and $q(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t})$ when they are different. This second ratio tends to increase the variation in the weight distribution, hence results in less efficiency (Liu and Chen 1998).

3. New SMC algorithms for nonlinear dynamic systems

It is generally quite difficult to directly generate samples from $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$ in (6) due to its complexity in nonlinear systems. Thus we are interested in finding some proposal distributions that are "close" to $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$ and yet easy to sample from. In this paper, we propose to use Gaussian proposal distributions

$$q\big(\boldsymbol{x}_t \,\big|\, \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t\big) = \mathcal{N}\big(\boldsymbol{x}_{t|t}^{(j)}, \boldsymbol{B}_t^{(j)}\big).$$

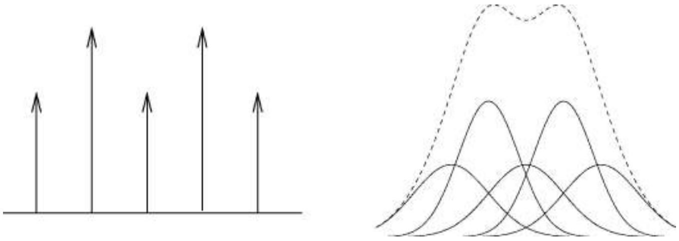

Fig. 1. *The posterior density $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t})$ is approximated by the discrete point mass of $\boldsymbol{x}_t^{(j)}$ (left) or the kernel based density representation at discrete points $\boldsymbol{x}_t^{(j)}$ with dynamic bandwidth (right)*

We study several approaches to obtain the proper $\boldsymbol{x}_{t|t}^{(j)}$ and $\boldsymbol{B}_t^{(j)}$ so that the proposal distribution is close to the "optimal" distribution $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$.

In standard SMC algorithms, the properly weighted samples are viewed as a discrete approximation (with point mass at the discrete points $\boldsymbol{x}_t^{(j)}$) to the target distribution $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t})$. That is,

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}) \cong \sum_{j=1}^{m} w_j \delta\big(\boldsymbol{x}_t - \boldsymbol{x}_t^{(j)}\big),$$

where $\delta(\cdot)$ is the Dirac delta function. On the other hand, using these samples, we can also approximate the target distribution by a kernel density estimate

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}) \cong \sum_{j=1}^{m} w_j K_j\big(\boldsymbol{x}_t - \boldsymbol{x}_t^{(j)}\big),$$

where $K_j(\cdot)$ is a certain kernel function. For example, if we use a Gaussian kernel $K_j(\cdot) = \phi(\cdot; \boldsymbol{0}, \boldsymbol{P}_t^{(j)})$ (where $\phi(\cdot; \boldsymbol{0}, \boldsymbol{P})$ denote a Gaussian $\mathcal{N}(\boldsymbol{0}, \boldsymbol{P})$ density function), then effectively the target distribution can be viewed as an approximation by a Gaussian mixture. Figure 1 illustrates the discrete density approximation and the kernel density approximation. The kernel method in SMC was firstly proposed in Liu and West (2001) to add the Gaussian disturbance to each discrete sample to increase sample diversity. Note that, the kernel bandwidth $\boldsymbol{P}_t^{(j)}$ reflects the tradeoff between the bias and variance in the density estimate. When $\boldsymbol{P}_t^{(j)} = 0$, the kernel density estimate reduces to the discrete representations.

We are interested in propagating the posterior density of the state from $(t-1)$ to t. As shown in the left part of Fig. 2, in the discrete representation case, we propagate each individual sample $x_{t-1}^{(j)}$ using a Gaussian proposal distribution that is close to $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$.

On the other hand, if we adopt the Gaussian kernel approximation case, as shown in the right part of Fig. 2, we propagate each Gaussian component $\mathcal{N}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{P}_{t-1}^{(j)})$ still using a Gaussian proposal distribution that is close to $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$. Hence, we will use a Gaussian proposal distribution that is close to the distribution $\int p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_t) p^{(j)}(\boldsymbol{x}_{t-1}) \mathrm{d}\boldsymbol{x}_{t-1}$, where $p^{(j)}(\boldsymbol{x}_{t-1}) \sim \mathcal{N}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{P}_{t-1}^{(j)})$. This distribution takes in some additional variation in the j-th sample.

Let $\pi_{t-1}^{(j)}$ be the joint distribution of $(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t, \boldsymbol{v}_t)$ where $\boldsymbol{x}_{t-1} \sim \delta(\boldsymbol{x}_{t-1})$ or $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{P}_{t-1}^{(j)})$. And denote $E_{\pi_{t-1}^{(j)}}$, $\mathrm{Var}_{\pi_{t-1}^{(j)}}$ and $\mathrm{Cov}_{\pi_{t-1}^{(j)}}$ as respectively the mean, variance and covariance operator with respect to $\pi_{t-1}^{(j)}$.

To obtain the mean and covariance matrix for the Gaussian proposal distribution for the j-th stream, we start by assuming $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are jointly Gaussian, given $\boldsymbol{x}_{t-1}$ either being fixed at $\boldsymbol{x}_{t-1}^{(j)}$ or following a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{P}_{t-1}^{(j)})$. It then follows that $p(\boldsymbol{x}_t \mid \boldsymbol{y}_t)$ is Gaussian with mean and covariance

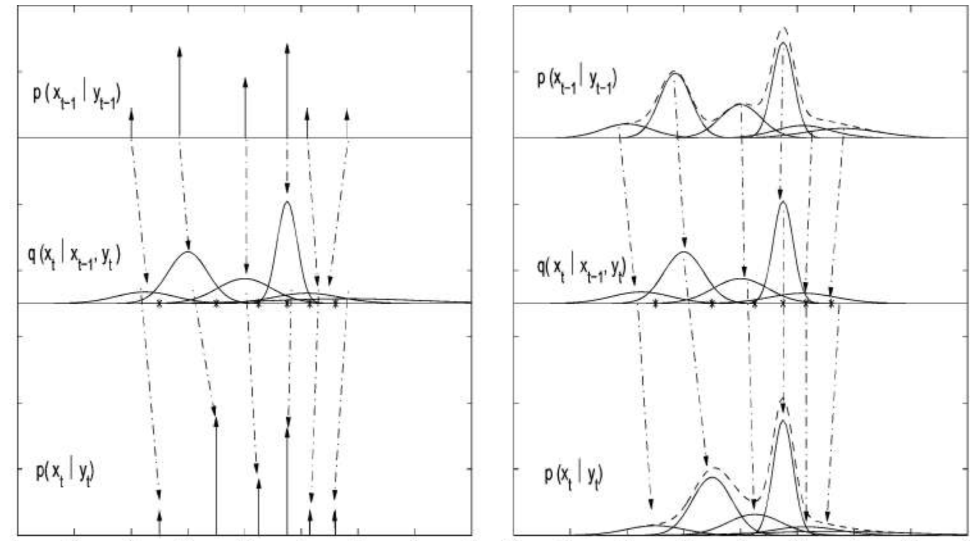

Fig. 2. *The propagation of the posterior density $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t})$ of the state from $(t-1)$ to t based on the discrete point mass of $\boldsymbol{x}_t^{(j)}$ (left) or the kernel based density representation at discrete points $\boldsymbol{x}_t^{(j)}$ with dynamic bandwidth (right)*

given respectively by

$$\begin{aligned} \boldsymbol{x}_{t|t}^{(j)} &\triangleq E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t \mid \boldsymbol{Y}_t\} \\ &= E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} + \boldsymbol{L}_t^{(j)}\big(\boldsymbol{y}_t - E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\}\big), \qquad (7) \\ \boldsymbol{\Sigma}_{t|t}^{(j)} &\triangleq \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t \mid \boldsymbol{Y}_t\} \\ &= \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} - \boldsymbol{L}_t^{(j)}\mathrm{Cov}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t, \boldsymbol{X}_t\}, \qquad (8) \\ \text{with} \quad \boldsymbol{L}_t^{(j)} &\triangleq \mathrm{Cov}_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t, \boldsymbol{Y}_t\}\big(\mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\}\big)^{-1}, \qquad (9) \end{aligned}$$

The proposal distribution is then chosen to be

$$q\big(\boldsymbol{X}_t \,|\, \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t\big) = \mathcal{N}\big(\boldsymbol{x}_{t|t}^{(j)}, \boldsymbol{B}_t^{(j)}\big),$$

where $\boldsymbol{B}_t^{(j)} = h_q^2 \boldsymbol{\Sigma}_{t|t}^{(j)}$. The bandwidth matrix for the Gaussian kernel representation at time t (for the next iteration) can be chosen as

$$\boldsymbol{P}_t^{(j)} = h_k^2 \Sigma_{t|t}^{(j)}.$$

The parameters h_q determines how much variation we put in the proposal distribution. For discrete representation, one usually choose $h_q = 1$ since in this case the proposal distribution should be as close to $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{y}_t)$ as possible. On the other hand, with the Gaussian mixture representation, certain shrinkage ($h_q < 1$) has been found to be very helpful. It tries to compensate the extra variation introduced in the kernel estimation.

The parameter h_k plays the role of adjusting the influence range of the kernel. Intuitively, an increase of h_k will increase the diversity of the particle, and hence will help to find the global optimum solution. However, the efficiency of the SMC may be sacrificed because of the increasing of variation and possibility of introducing many inefficient particles. Note that if $h_k = 0$, then the kernel density approximation reduces to the discrete representation. If $h_k = 1$ is used, the bandwidth may approach to a constant in some cases when the variance $\Sigma_{t|t}^{(j)}$ is stable.

In what follows we discuss three approaches to computing the first-order moments

$$E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \quad \text{and} \quad E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\}, \qquad (10)$$

and the second-order moments

$$\mathrm{Var}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\big\}, \quad \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\}, \quad \text{and} \quad \mathrm{Cov}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \boldsymbol{Y}_t\big\} \qquad (11)$$

appeared in (7)–(9), under both the discrete density approximation and the Gaussian kernel density approximation. They are the key components of EKF, UKF and QKF. Note that the conventional filters EKF, UKF and QKF are obtained with the newly estimated value and variance at time t as the starting point input into the Gaussian kernel density approximation at next time $t+1$ and repeating this procedure at next time, and so on.

3.1. *Extended Kalman filter*

In EKF, the state equation (1) is first linearized by its first-order Taylor series expansion at $E_{\pi_{t-1}^{(j)}}\{\boldsymbol{x}_{t-1}\} = \boldsymbol{x}_{t-1}^{(j)}$:

$$E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} \cong \boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \boldsymbol{F}_t^{(j)}\big(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{(j)}\big) + \boldsymbol{u}_t, \qquad (12)$$

with $\boldsymbol{F}_t^{(j)} \triangleq \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_{t-1}^{(j)}}$. Then the measurement equation (2) is linearized at $E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} = \boldsymbol{f}(\boldsymbol{x}_{t-1}^{(j)})$:

$$\boldsymbol{y}_t \cong \boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big)\big) + \boldsymbol{H}_t^{(j)}\big(\boldsymbol{x}_t - \boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big)\big) + v_t, \qquad (13)$$

with $\boldsymbol{H}_t^{(j)} \triangleq \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{h}(\boldsymbol{x})|_{\boldsymbol{x}=f\boldsymbol{x}_{t-1}^{(j)}}$. Based on the above linearization, the moments in (10) and (11) can be approximated as follows (the discrete case uses $\boldsymbol{P}_{t-1}^{(j)} = 0$).

$$\begin{aligned} E_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\big\} &\cong \boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big), \\ E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big)\big), \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\big\} &\cong \boldsymbol{F}_t^{(j)}\boldsymbol{P}_{t-1}^{(j)}\big(\boldsymbol{F}_t^{(j)}\big)^T + \boldsymbol{Q}, \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \boldsymbol{H}_t^{(j)}\big[\boldsymbol{F}_t^{(j)}\boldsymbol{P}_{t-1}^{(j)}\big(\boldsymbol{F}_t^{(j)}\big)^T + \boldsymbol{Q}\big] \\ &\quad \times \big(\boldsymbol{H}_t^{(j)}\big)^T + \boldsymbol{R}, \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \boldsymbol{Y}_t\big\} &\cong \boldsymbol{F}_t^{(j)}\boldsymbol{P}_{t-1}^{(j)}\big(\boldsymbol{F}_t^{(j)}\big)^T\big(\boldsymbol{H}_t^{(j)}\big)^T. \end{aligned}$$

3.2. *Unscented Kalman filter*

The unscented transformation (UT) is a method of approximating the mean and variance of a nonlinearly transformed random vector. Let $\boldsymbol{z}$ be an n_z-dimensional random vector with mean $\bar{\boldsymbol{z}}$ and variance $\boldsymbol{\Sigma}_z$. Suppose that we want to compute the mean and variance of $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{z})$. The UT chooses a set of $(2n_z + 1)$ deterministic weighted points (called sigma points) $\{\boldsymbol{\xi}_\ell, \omega_\ell\}$ as follows (Julier and Durrant-Whyte 2000, Merwe *et al.* 2000):

$$\boldsymbol{\xi}_0 = \bar{\boldsymbol{z}}, \quad \omega_0 = \kappa/(n_z + \kappa), \quad \ell = 0, \qquad (14)$$

$$\boldsymbol{\xi}_\ell = \bar{\boldsymbol{z}} + \sqrt{n_z + \kappa}[\sqrt{\boldsymbol{\Sigma}_z}]_\ell, \quad \omega_\ell = \kappa/(2n_z + \kappa), \ell = 1, 2, \ldots, n_z, \qquad (15)$$

$$\boldsymbol{\xi}_\ell = \bar{\boldsymbol{z}} - \sqrt{n_z + \kappa}[\sqrt{\boldsymbol{\Sigma}_z}]_{\ell - n_z}, \quad \omega_\ell = \kappa/(2n_z + \kappa), \ell = n_z + 1, n_z + 2, \ldots, 2n_z, \qquad (16)$$

where $\kappa > 0$ is a parameter; $[\sqrt{\boldsymbol{\Sigma}_z}]_\ell$ denotes the ℓ-th column of the square root of $\boldsymbol{\Sigma}_z$. Note that the weights satisfy $\sum_{\ell=0}^{2n_x} \omega_\ell = 1$.

The mean and variance of $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{z})$ are approximated by

$$\bar{\boldsymbol{y}} \cong \sum_{\ell=0}^{2n_z} \omega_\ell \boldsymbol{g}(\boldsymbol{\xi}_\ell),$$

$$\boldsymbol{\Sigma}_y \cong \sum_{\ell=0}^{2n_z} \omega_\ell[\boldsymbol{g}(\boldsymbol{\xi}_\ell) - \bar{\boldsymbol{y}}][\boldsymbol{g}(\boldsymbol{\xi}_\ell) - \bar{\boldsymbol{y}}]^T.$$

It is known that the sigma points capture the first-order and the second-order Tailor expansion of the nonlinear equation accurately, with errors only in the third- and higher-order expansion. Furthermore, the errors introduced in the third- and higher-order expansion can be scaled to some extent by the choice of the parameter κ. However, if the nonlinearities are very severe, the above approximation may be inaccurate. In that case the scaled unscented transformation (Julier and Durrant-Whyte 2000) can be employed by introducing another positive scaling parameter α, so that the sigma points are replaced by

$$\boldsymbol{\xi}'_\ell = \bar{\boldsymbol{z}} + \alpha(\boldsymbol{\xi}_\ell - \bar{\boldsymbol{z}}), \qquad \ell = 0, \dots, 2n_z, \tag{17}$$

$$\omega'_\ell = \begin{cases} \dfrac{\omega_0}{\alpha^2} + \left(1 - \dfrac{1}{\alpha^2}\right), & \ell = 0, \\ \dfrac{\omega_\ell}{\alpha^2}, & \ell = 1, \dots, 2n_z. \end{cases} \tag{18}$$

Using the above unscented transformation, we can approximate the first-order and the second-order moments in (10) and (11) as follows.

Discrete density case:

In this case the random vector $\boldsymbol{z}$ is set as $\boldsymbol{z} = [\boldsymbol{u}_t^T, \boldsymbol{v}_t^T]$ where u_t and v_t are respectively the state noise and the measurement noise defined in (1) and (2). We set $n_z = n_u + n_v$, there n_u and n_v are the corresponding dimensions of u_t and v_t respectively. By letting

$$\bar{\boldsymbol{z}} = \boldsymbol{0}, \quad \text{and} \quad \Sigma_z = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix},$$

we calculate the sigma points $\{(\boldsymbol{\xi}_\ell, \omega_\ell), \ell = 0, 1, \dots, 2n_z\}$ according to (14)–(16) or (17) and (18). Denote $\boldsymbol{\xi}_\ell^u$ as the components of $\boldsymbol{\xi}_\ell$ corresponding to the state noise sample $\boldsymbol{u}_t$. And similarly define $\boldsymbol{\xi}_\ell^v$. Then based on the state-space model (1) and (2), the moments in (10) and (11) are approximated by the following:

$$\begin{aligned} E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} &\cong \boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big), \\ E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \boldsymbol{\xi}_\ell^u\big) \triangleq \bar{\boldsymbol{y}}_t, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\big\} &\cong \boldsymbol{Q}, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \boldsymbol{\xi}_\ell^u\big) + \boldsymbol{\xi}_\ell^v - \bar{\boldsymbol{y}}_t\big] \\ &\quad \cdot \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \boldsymbol{\xi}_\ell^u\big) + \boldsymbol{\xi}_\ell^v - \bar{\boldsymbol{y}}_t\big]^T, \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \boldsymbol{Y}_t\big\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \boldsymbol{\xi}_\ell^u \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \boldsymbol{\xi}_\ell^u\big) \\ &\quad + \boldsymbol{\xi}_\ell^v - \bar{\boldsymbol{y}}_t\big]^T. \end{aligned}$$

Gaussian kernel density case:

In this case we set the random vector $\boldsymbol{z} = [\boldsymbol{x}_{t-1}^T \;\; \boldsymbol{u}_t^T \;\; \boldsymbol{v}_t^T]^T$ and $n_z = n_x + n_u + n_v$. By letting

$$\bar{\boldsymbol{z}} = \big[\big(\boldsymbol{x}_{t-1}^{(j)}\big)^T \;\; \boldsymbol{0}^T \;\; \boldsymbol{0}^T\big]^T \quad \text{and} \quad \Sigma_z = \begin{bmatrix} \boldsymbol{P}_{t-1}^{(j)} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R} \end{bmatrix},$$

we calculate the sigma points $\{(\boldsymbol{\xi}_\ell, \omega_\ell), \ell = 0, 1, \dots, 2n_z\}$. Then the moments in (10) and (11) are approximated by the following:

$$\begin{aligned} E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) = \bar{\boldsymbol{x}}_t, \\ E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u\big) \triangleq \bar{\boldsymbol{y}}_t, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\big\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \big[\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u - \bar{\boldsymbol{x}}_t\big] \\ &\quad \times \big[\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u - \bar{\boldsymbol{x}}_t\big]^T, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u\big) + \boldsymbol{\xi}_\ell^v - \bar{\boldsymbol{y}}_t\big] \\ &\quad \cdot \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u\big) + \boldsymbol{\xi}_\ell^v - \bar{\boldsymbol{y}}_t\big]^T, \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\big\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \boldsymbol{Y}_t\big\} &\cong \sum_{\ell=0}^{2n_z} \omega_\ell \big[\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u - \bar{\boldsymbol{x}}_t\big] \\ &\quad \cdot \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{\xi}_\ell^x\big) + \boldsymbol{\xi}_\ell^u\big) + \boldsymbol{\xi}_\ell^v - \bar{\boldsymbol{y}}_t\big]^T. \end{aligned}$$

3.3. *Quadrature Kalman filter*

The Gaussian quadrature methods are a family of deterministic numerical integration techniques which approximate the integral by a weighted sum of the its functional values at certain abscissas (Press *et al.* 1992). They provide the optimal weighting coefficients as well as the location of the abscissas at which the function is to be evaluated. In particular, the Gauss-Hermite quadrature rule can be used to calculate the expected value of a function with respect to a Gaussian density. For example, suppose $z \sim \mathcal{N}(0, 1)$, then

$$E\{g(z)\} = \int g(z)\pi^{-\frac{1}{2}} \exp\left(-\frac{1}{2}z^2\right) \mathrm{d}z \cong \sum_{\ell=1}^{k} \omega_\ell g(\xi_\ell),$$

where the weight w_ℓ and the abscissas xi_ℓ are given according to the Gauss-Hermite quadrature rule (Ito and Xiong 2000, Press *et al.* 1992) as follows. Let $\boldsymbol{J}$ be a symmetric tridiagonal matrix with zero diagonals and

$$[\boldsymbol{J}]_{i,i+1} = \sqrt{\frac{i}{2}}, \quad 1 \le i \le k - 1.$$

Then the abscissas $\xi_\ell \triangleq \sqrt{2} x_\ell$, where x_ℓ is the ℓ-th eigenvalue of $\boldsymbol{J}$; and the corresponding weight $\omega_\ell \triangleq \mid (\boldsymbol{v}_\ell)_1 \mid^2$, where $(\boldsymbol{v}_\ell)_1$ is the first element of the ℓ-th normalized eigenvector of $\boldsymbol{J}$. Note that $\Sigma_{\ell=1}^k \omega_\ell = 1$. The number k of Gauss-Hermite quadrature points make the integration accurate hold for all polynomials of degree up to $2k - 1$.

Now suppose $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_{n_z})$, then the Gauss-Hermite quadrature method can be used to approximate $E\{\boldsymbol{g}(\boldsymbol{z})\}$ as

$$E\{\boldsymbol{g}(\boldsymbol{Z})\} = \int \boldsymbol{g}(\boldsymbol{z}) \pi^{-\frac{n_z}{2}} \exp\left(-\frac{1}{2}||\boldsymbol{z}||^2\right) \mathrm{d}\boldsymbol{z} \cong \sum_{\ell=1}^{K} \omega_\ell \boldsymbol{g}(\boldsymbol{\xi}_\ell),$$

where $K \triangleq k^{n_z}$ and the weight and the abscissas are obtained by the tensor product, i.e., $\omega_\ell = \Pi_{j=1}^{n_z} \omega_{\ell_j}$, and $\boldsymbol{\xi}_\ell = [\xi_{\ell_1}, \ldots, \xi_{\ell_{n_z}}] \in \{\xi_1, \ldots, \xi_k\}^{n_z}$. On the other hand, suppose $\boldsymbol{z} \sim \mathcal{N}(\bar{\boldsymbol{z}}, \boldsymbol{\Sigma}_z)$, then we have

$$\begin{aligned} E\{\boldsymbol{Z}\} &= \int \boldsymbol{g}(\boldsymbol{z}) \frac{1}{\pi^{\frac{n_z}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{z} - \bar{\boldsymbol{z}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{z} - \bar{\boldsymbol{z}})\right] \mathrm{d}\boldsymbol{z} \\ &= \int \boldsymbol{g}(\sqrt{\boldsymbol{\Sigma}}\boldsymbol{r} + \bar{\boldsymbol{z}}) \pi^{-\frac{n_z}{2}} \exp\left(-\frac{1}{2}||\boldsymbol{r}||^2\right) \mathrm{d}\boldsymbol{r} \\ &\cong \sum_{\ell=1}^{K} \omega_\ell \boldsymbol{g}(\sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}_\ell + \bar{\boldsymbol{z}}). \end{aligned}$$

Using the above Gauss-Hermite quadrature rule, we can approximate the first-order and the second-order moments in (10) and (11) as follows.

Discrete density case:

In this case the expectation is with respective to the joint distribution of

$$\boldsymbol{z} \triangleq \boldsymbol{x}_t \sim \mathcal{N}(\bar{\boldsymbol{z}}, \boldsymbol{\Sigma}_z), \quad \text{with} \quad \bar{\boldsymbol{z}} = \boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big), \quad \boldsymbol{\Sigma}_z = \boldsymbol{Q}.$$

Then (10) and (11) are approximated by the following:

$$\begin{aligned} E_{\pi_{t-1}^{(j)}}\left\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\right\} &\cong \boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big), \\ E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=0}^{K} \omega_\ell \boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \sqrt{\boldsymbol{Q}}\boldsymbol{\xi}_\ell\big) \triangleq \bar{\boldsymbol{y}}_t, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\left\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\right\} &\cong \boldsymbol{Q}, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=1}^{K} \omega_\ell \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \sqrt{\boldsymbol{Q}}\boldsymbol{\xi}_\ell\big) - \bar{\boldsymbol{y}}_t\big] \\ &\quad \cdot \big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \sqrt{\boldsymbol{Q}}\boldsymbol{\xi}_\ell\big) - \bar{\boldsymbol{y}}_t\big]^T + \boldsymbol{R}, \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\left\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \boldsymbol{Y}_t\right\} &\cong \sum_{\ell=1}^{K} \omega_\ell (\sqrt{\boldsymbol{Q}}\boldsymbol{\xi}_\ell)\big[\boldsymbol{h}\big(\boldsymbol{f}\big(\boldsymbol{x}_{t-1}^{(j)}\big) + \sqrt{\boldsymbol{Q}}\boldsymbol{\xi}_\ell\big) \\ &\quad - \bar{\boldsymbol{y}}_t\big]^T. \end{aligned}$$

Gaussian kernel density case:

In this case we first set

$$\boldsymbol{Z} \triangleq \boldsymbol{Z}_{t-1} \sim \mathcal{N}(\bar{\boldsymbol{z}}, \boldsymbol{\Sigma}_z), \qquad \text{with} \quad \bar{\boldsymbol{z}} = \boldsymbol{x}_{t-1}^{(j)}, \quad \boldsymbol{\Sigma}_z = \boldsymbol{P}_{t-1}^{(j)};$$

and calculate the following:

$$\begin{aligned} E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\} &\cong \sum_{\ell=1}^{K} \omega_\ell \boldsymbol{f}\Big(\sqrt{\boldsymbol{P}_{t-1}^{(j)}}\boldsymbol{\xi}_\ell + \boldsymbol{x}_{t-1}^{(j)}\Big), \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\left\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}\right\} &\cong \sum_{\ell=1}^{K} \omega_\ell \Big[\boldsymbol{f}\Big(\sqrt{\boldsymbol{P}_{t-1}^{(j)}}\boldsymbol{\xi}_\ell + \boldsymbol{x}_{t-1}^{(j)}\Big) - \bar{\boldsymbol{x}}_t\Big] \\ &\quad \cdot \Big[\boldsymbol{f}\Big(\sqrt{\boldsymbol{P}_{t-1}^{(j)}}\boldsymbol{\xi}_\ell + \boldsymbol{x}_{t-1}^{(j)}\Big) - \bar{\boldsymbol{x}}_t\Big]^T + \boldsymbol{Q} \triangleq \boldsymbol{\Psi}_t. \end{aligned}$$

Then we set

$$\boldsymbol{Z} \triangleq \boldsymbol{X}_t \sim \mathcal{N}(\bar{z}, \boldsymbol{\Sigma}_z), \qquad \text{with} \quad \bar{z} = \boldsymbol{x}_t, \quad \boldsymbol{\Sigma}_z = \boldsymbol{\Psi}_t;$$

and calculate the following

$$\begin{aligned} E_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=1}^{K} \omega_\ell \boldsymbol{h}(\sqrt{\boldsymbol{\Psi}_t}\boldsymbol{\xi}_\ell + \bar{\boldsymbol{x}}_t) \triangleq \bar{\boldsymbol{y}}_t, \\ \mathrm{Var}_{\pi_{t-1}^{(j)}}\{\boldsymbol{Y}_t\} &\cong \sum_{\ell=1}^{K} \omega_\ell [\boldsymbol{h}(\sqrt{\boldsymbol{\Psi}_t}\boldsymbol{\xi}_\ell + \bar{\boldsymbol{x}}_t) - \bar{\boldsymbol{y}}_t] \\ &\quad \times [\boldsymbol{h}(\sqrt{\boldsymbol{\Psi}_t}\boldsymbol{\xi}_\ell + \bar{\boldsymbol{x}}_t) - \bar{\boldsymbol{y}}_t]^T + \boldsymbol{R}, \\ \mathrm{Cov}_{\pi_{t-1}^{(j)}}\left\{E_{\pi_{t-1}^{(j)}}\{\boldsymbol{X}_t\}, \boldsymbol{Y}_t\right\} &\cong \sum_{\ell=1}^{K} \omega_\ell (\sqrt{\boldsymbol{\Psi}_t}\boldsymbol{\xi}_\ell)[\boldsymbol{h}(\sqrt{\boldsymbol{\Psi}_t}\boldsymbol{\xi}_\ell + \bar{\boldsymbol{x}}_t) \\ &\quad - \bar{\boldsymbol{y}}_t]^T. \end{aligned}$$

3.4. *Summary of the new SMC algorithms*

Based on the different approaches of calculating the moments discussed in the previous subsections, we obtain a family of new SMC algorithms for the nonlinear dynamic system given by (1) and (2). These algorithms fit into the following general framework.

A0. Initialization: For $j = 1, 2, \ldots, m$, draw the initial state $\boldsymbol{x}_0^{(j)}$ from the prior distribution $p(\boldsymbol{x}_0) \sim \mathcal{N}(\boldsymbol{x}_{0|0}, \Sigma_{0|0})$, where $\boldsymbol{x}_{0|0}$ and $\Sigma_{0|0}$ are dependent upon different application. All importance weights are initialized as $w_0^{(j)} = 1$, $j = 1, \ldots, m$.

The following steps are implemented at the t-th recursion ($t = 1, 2, \ldots$).

For $j = 1, \ldots, m$:

A1. Compute the proposal density $q(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}) = \mathcal{N}(\boldsymbol{x}_{t|t}^{(j)}, h_q^2 \boldsymbol{\Sigma}_{t|t}^{(j)})$, where $\boldsymbol{x}_{t|t}^{(j)}$ and $\boldsymbol{\Sigma}_{t|t}^{(j)}$ are given by (7) and (8), where the moments are calculated based on either EKF, UKF, or QKF, as described in the previous subsections, under either the discrete density approximation or the Gaussian kernel density approximation (hence a total of six SMC algorithms). And for the kernel density case, update $\boldsymbol{P}_t^{(j)} = h_k^2 \boldsymbol{\Sigma}_{t|t}^{(j)}$.

A2. Draw a sample $\boldsymbol{x}_t^{(j)}$ from $\mathcal{N}(\boldsymbol{x}_{t|t}^{(j)}, h_q^2 \boldsymbol{\Sigma}_{t|t}^{(j)})$, and append $\boldsymbol{x}_t^{(j)}$ to $\boldsymbol{x}_{1:t-1}^{(j)}$ to from $\boldsymbol{x}_{1:t}^{(j)} = [\boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{x}_t^{(j)}]$.

A3. Compute the importance weight $w_t^{(j)}$ as follows:

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{p\left(\boldsymbol{y}_t \mid \boldsymbol{x}_t^{(j)}\right) p\left(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{x}_{t-1}^{(j)}\right)}{q\left(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{y}_{1:t}\right)}$$

$$= w_{t-1}^{(j)} \frac{\phi\left(\boldsymbol{y}_t; \boldsymbol{h}\left(\boldsymbol{x}_t^{(j)}\right), \boldsymbol{R}\right) \phi\left(\boldsymbol{x}_t^{(j)}; \boldsymbol{f}\left(\boldsymbol{x}_{t-1}^{(j)}\right), Q\right)}{\phi\left(\boldsymbol{x}_t^{(j)}; \boldsymbol{x}_{t|t}^{(j)}, \boldsymbol{h}_q^2 \boldsymbol{\Sigma}_{t|t}^{(j)}\right)},$$

where $\phi(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ density function.

A4. Resampling is done if the effective sample size $\bar{m}_t$ is less than a specified threshold.

4. Applications

In this section, we illustrate the performance of the proposed SMC algorithms via one illustrated example and two applications, namely, target tracking and multiuser parameter tracking in code-division multiple-access (CDMA) communication systems. We will compare the performance of these new SMC algorithms with three conventional filtering methods—EKF, UKF and QKF.

4.1. *Target tracking*

Target tracking is of interest to many engineers and computer scientists and has received significant recent attention. Many methods have been proposed, among which the SMC has become one of the very promising solutions (Doucet *et al.* 2001, Gordon *et al.* 1995, Liu and Chen 1998).

Consider the nonlinear dynamic system given by (1) and (2) in which the state vector $x_t = [x_t, \dot{x}_t, y_t, \dot{y}_t]^T$ consists of the position (x_t, y_t) and the velocity $(\dot{x}_t, \dot{y}_t)$ of a target at time t . The target dynamics evolve linearly from time t to time $(t + 1)$, with a sampling time interval Δ. The measurement vector $\boldsymbol{y}_t$ consists of the measured range and bearing angle. In particular, we have

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{u}_t = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \\ y_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} + \boldsymbol{u}_t, \quad (19)$$

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{v}_t = \begin{bmatrix} \sqrt{x_t^2 + y_t^2} \\ \tan^{-1} \frac{yt}{xt} \end{bmatrix} + \boldsymbol{v}_t, \quad (20)$$

where $\boldsymbol{u}_t$ and $\boldsymbol{v}_t$ are independent zero-mean Gaussian noise with covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively.

In our simulations, we set $\Delta = 1$, $\boldsymbol{Q} = \text{diag}(0.05, 5, 0.05, 5)$, $\boldsymbol{R} = \text{diag}(1,1)$. The initial state are $\boldsymbol{x}_{0|0} = [5, -20, 5, -20]^T$ and $\boldsymbol{\Sigma}_{0|0}^x = \text{diag}(0.5, 50, 0.5, 50)$. We have implemented both the conventional nonlinear filtering algorithms (including the EKF, the UKF and the QKF) and the new SMC algorithms developed in Section 3 (i.e., the SMC-EKF, the SMC-UKF, and the SMC-QKF, under either the discrete density approximation or the Gaussian kernel density approximation). Moreover, we have also implemented the SIR particle filter where the proposal distribution is based only on the state equation and given by (5). In the SIR particle filter, the number of the sample streams is set as $m = 300$ and the exact initial state $\boldsymbol{x}_0$ is used. In all the proposed SMC algorithms, the number of sample streams is set as $m = 50$, and the initial state is set as $p(\boldsymbol{x}_0 \mid \boldsymbol{y}_0) \sim \mathcal{N}(\boldsymbol{X}_{0|0}, \boldsymbol{\Sigma}_{0|0})$.

In the EKF and the SMC-EKF algorithms, the nonlinear measurement equation (20) is linearized as

$$\boldsymbol{y}_t = \boldsymbol{H}_t(\boldsymbol{x}_t - \boldsymbol{x}_{t|t-1}) + \boldsymbol{h}(\boldsymbol{x}_{t|t-1}) + \boldsymbol{v}_t,$$

$$\text{with} \quad \boldsymbol{H}_t = \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{h}|_{\boldsymbol{x}=\boldsymbol{x}_{t|t-1}}$$

$$= \left. \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} & 0 \\ \frac{-y}{\sqrt{x^2+y^2}} & 0 & \frac{x}{\sqrt{x^2+y^2}} & 0 \end{bmatrix} \right|_{x=x_{t|t-1}, y=y_{t|t-1}}.$$

In the UKF and the SMC-UKF algorithms, we set the parameters $\kappa = 2$ and $\alpha = 1.5$. In the QKF and SMC-QKF algorithms, the number of the abscissas is set to be $K = 4^5$, i.e., five abscissas in one dimension.

We run each algorithm over a simulated target trajectory of 200 time instants and calculate the corresponding root mean-square error for each component of the state vector $\boldsymbol{x}_t$. The results are listed in Table 1. The running time is in terms of seconds running in a Dell computer with a *Pentium*® 4 at 2.66 GHz. In this table, the algorithms considered are divided into three groups. The first group consists of the existing methods, including the extended Kalman filter, the unscented Kalman filter (UKF), the quadrature Kalman filter (QKF), and the SIR particle filter (PF). It is seen that in this group, the PF has the worst performance. This is because the initial variance $\boldsymbol{\Sigma}_{0|0}$ is so large that it takes long time to converge. With the increase number of Monte Carlo sample, the performance of the PF can

Table 1. *The root mean-square error (RMSE) performance of various nonlinear filtering algorithms for target tracking*

Algorithms	x_t	$\dot{x}_t$	y_t	$\dot{y}_t$	Time (s)
EKF	0.325	0.025	0.239	0.043	0.20
UKF	0.145	0.019	0.157	0.015	0.65
QKF	0.130	0.015	0.126	0.013	1.39
PF	0.611	0.051	0.701	0.017	51.96
D-SMC-EKF	0.104	0.016	0.110	0.015	89.17
D-SMC-UKF	0.084	0.018	0.070	0.012	192.39
D-SMC-QKF	0.073	0.055	0.081	0.054	334.86
K-SMC-EKF	0.092	0.018	0.121	0.014	91.17
K-SMC-UKF	0.008	0.011	0.005	0.012	192.39
K-SMC-QKF	0.003	0.015	0.006	0.014	337.86

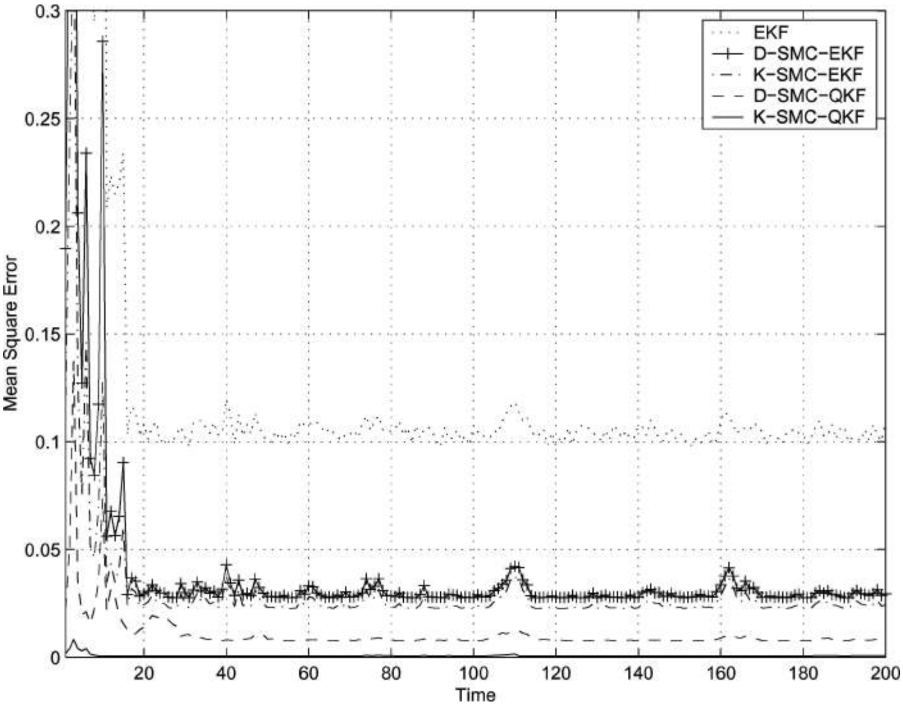

Fig. 3. *The mean-square error performance of various nonlinear tracking algorithm for the state variable x_t.*

be improved. Although the UKF and QKF can improve the performance over the EKF, the improvement is not very significant. The reason for this is that the state equation (19) is linear. Note that the complexity of the QKF is much higher than that of the UKF—the QKF needs 4^5 abscissas whereas the UKF only needs 24 deterministic sampling points. Hence for high-dimensional problem, we suggest to use the UKF instead of the QKF.

The second group in Table 1 consists of the three new SMC algorithms under the discrete density approximation (D-SMC-EKF, D-SMC-UKF, and D-SMC-QKF). It is clearly seen that these new SMC algorithms offer performance improvement of the existing nonlinear filtering algorithms in the first group. The third group consists of the three new SMC algorithms under the kernel density approximation (K-SMC-EKF, K-SMC-UKF, and K-SMC-QKF). It is seen that by replacing the discrete density approximation with the Gaussian kernel density approximation, the tracking accuracy can be significantly improved for the SMC-UKF and the SMC-QKF algorithms; whereas the effect on the SMC-EKF is quite small. The tracking performance of some of the above algorithms is also illustrated in Fig. 3.

4.2. *Multiuser parameter tracking in CDMA communications*

The area of direct-sequence code-division multiple-access (DS-CDMA) communications has attracted significant recent attention as it has become the major air interface technique for the next-generation cellular wireless communication systems. One of the main challenges in CDMA receiver design involves the estimation of some key system parameters, such as the received signal powers, the carrier phase offsets, and the propagation delays for all users of interest. Recently, several recent works have addressed the problem of joint estimation of the channel and the time delays (Iltis 2001, Caffery and Stuber 2000). In what follows, we treat the problem of multiuser parameter tracking under the nonlinear dynamic system framework.

In a CDMA system, the k-th user transmits the bandpass signal

$$x_k(t) = \sum_{m=0}^{M-1} \sqrt{2P_k} d_k(m) s_k(t - mT_b) \cos(\omega_c t + \theta_k(t)),$$

where P_k is the transmitted signal power of the k-th user; ω_c is the carrier frequency; $\theta_k(t)$ is the phase offset for the k-th user's carrier; $d_k(m)$ is the m-th data symbol of the k-th user; and $s_k(t)$ is the k-th user's spreading signature waveform of duration T_b, given by

$$s_k(t) = \sum_{n=0}^{N-1} a_{k,n} \psi_{T_c}(t - nT_c), \quad 0 \le t \le T_b,$$

where $a_{k,n} \in \{-1, 1\}$ is the n-th chip of the signature sequence of the k-th user, $N = \frac{T_b}{T_c}$ is called the spreading gain. The chip waveform $\psi_{T_c}(\cdot)$ is a rectangular pulse given by

$$\psi_{T_c}(t) = \begin{cases} 1, & 0 \le t \le T_c, \\ 0, & \text{otherwise.} \end{cases}$$

The received bandpass signal is the superposition of the delayed version of the multiuser signals plus the ambient noise,

$$\begin{aligned} r(t) &= \sum_{k=1}^{\kappa} x_k\left(t - \tau_k(t)\right) + n(t) \\ &= \sum_{k=1}^{\kappa}\sum_{m=0}^{M-1} \sqrt{2P_k} d_k(m) f_k(t) s_k\left(t - mT_b - \tau_k(t)\right) \\ &\quad \times \cos\left(\omega_c t + \phi_k(t)\right) + n(t), \end{aligned}$$

where $\phi_k(t) = \theta_k(t) - \omega_c \tau_k(t)$ is the received phase of the k-th user's carrier, $\tau_k(t)$ is the delay of the k-th user's signal; $f_k(\cdot)$ is channel attenuation experienced by the k-th user's signal; and $n(t)$ is additive white Gaussian noise.

Suppose that the received continuous-time signal $r(t)$ is first down-converted to baseband and then sampled at q times the chip rate, giving a time interval between samples of $T_s = T_c/q$. Then the received baseband signal sampled at $t = lT_s$ is

$$\begin{aligned} r(l) &= \sum_{k=1}^{\kappa} \beta_k(l) d_k\left(m_k(l)\right) s_k(lT_s - m_k(l)T_s \\ &\quad - \tau_k(lT_s)) + n(lT_s), \qquad (21) \end{aligned}$$

$$\begin{aligned} \text{with} \quad \beta_k(l) &\triangleq \sqrt{P_k} f_k(lT_s) \exp\left(j\phi_k(lT_s)\right), \\ m_k(l) &= \left\lfloor \frac{lT_s - \tau_k(lT_s)}{T_b} \right\rfloor. \end{aligned}$$

Now assume that the complex-valued channel amplitudes and the real-valued propagation delays of the κ users obey a Gaussian-Markov dynamic model, i.e.,

$$\begin{aligned} \boldsymbol{\beta}(l) &= \boldsymbol{\Phi}_\beta \boldsymbol{\beta}(l-1) + \boldsymbol{u}_\beta(l), \\ \boldsymbol{\tau}(l) &= \boldsymbol{\Phi}_\tau \boldsymbol{\tau}(l-1) + \boldsymbol{u}_\tau(l), \end{aligned}$$

where $\boldsymbol{\Phi}_\beta$ and $\boldsymbol{\Phi}_\tau$ are the $K \times K$ state transition matrices for the amplitudes and delays, respectively. The noise $\boldsymbol{u}_\beta$ and $\boldsymbol{u}_\tau$ are assumed to be independent with zero-mean and covariance matrices given respectively by $\boldsymbol{Q}_\beta \triangleq \sigma_\beta^2 \boldsymbol{I}_\kappa$ and $\boldsymbol{Q}_\tau \triangleq \sigma_\tau^2 \boldsymbol{I}_\kappa$. Note that since the measurement equation (21) is not continuous in the delay variables, the EKF can not be directly applied to this problem.

We have implemented the conventional as well as the SMC-based UKF and QKF for parameter tracking in a two-user system. The user's spreading sequence are chosen from the set of Gold codes of length $N = 31$ generated by the polynomials $x^5 + x^2 + 1$ and $x^5 + x^4 + x^3 + x^2 + 1$. The signal-to-noise ratio (SNR) at the receiver for each user is 10 dB. The state transition matrices are chosen to be $\boldsymbol{\Phi}_\beta = 0.999\, \boldsymbol{I}_\kappa$ and $\boldsymbol{\Phi}_\tau = 0.999\, \boldsymbol{I}_\kappa$. The state noise variances are set as $\sigma_\beta^2 = \sigma_\tau^2 = 0.001$. The signal is sampled at the chip rate, i.e., $q = 1$. It is assumed that during the parameter tracking stage, the data symbols $\{d_k(m)\}$ are known to the receiver. The parameters in various algorithms are similarly set as in the previous subsection. The RMSE performance of various algorithms are shown in Table 2. The running is in terms of seconds running in a Dell computer with a *Pentium*® 4 at 2.66 GHz.

Table 2. *The RMSE performance of various nonlinear filtering algorithms for channel parameter tracking in a two-user CDMA system*

Algorithms	Amplitude User 1	Amplitude User 2	Delay User 1	Delay User 2	Time (s)
UKF	0.15	0.19	0.13	0.11	227.58
QKF	0.14	0.18	0.16	0.16	972.32
PF	0.26	0.24	0.12	0.11	1213.2
D-SMC-UKF	0.11	0.12	0.09	0.11	4517.6
D-SMC-QKF	0.12	0.13	0.11	0.13	19041.2
K-SMC-UKF	0.05	0.06	0.09	0.08	4608.6
K-SMC-QKF	0.04	0.03	0.06	0.06	19051.1

As in the previous example, it is seen that the K-SMC-UKF and the K-SMC-QKF algorithms exhibit better performance than the other methods in tracking the multiuser amplitudes and delays, especially compared with the existing nonlinear filtering schemes such as the UKF, the QKF, and the PF. Note that the simulations revealed that the performance of the conventional UKF and the QKF is very sensitive to the initial conditions, whereas the SMC-based methods is robust to the uncertainties in the initial conditions. This is due to the fact that the SMC starts with many streams of samples each with different starting point and tries to find the optimum streams using the importance weights. Figure 3 illustrates the amplitude tracking performance by various algorithms.

4.3. *An illustrated high-dimensional example*

Consider a nonlinear dynamic system given by (1) and (2) in which the state vectors $\boldsymbol{x}_t$ and the observations $\boldsymbol{y}_t$ are 10-dimensional vectors. In particular, we have

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{u}_t \qquad (22)$$

$$\boldsymbol{y}_t = \boldsymbol{F}\boldsymbol{x}_t^2 + \boldsymbol{v}_t \qquad (23)$$

where $\boldsymbol{u}_t$ and $\boldsymbol{v}_t$ are independent zero-mean Gaussian noise with covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively.

In our simulations, we set $\boldsymbol{Q} = 0.05\, \boldsymbol{I}_{10,10}$, $\boldsymbol{R} = 0.01\, \boldsymbol{I}_{10,10}$, where $\boldsymbol{I}$ is the identity matrix. The initial states are $\boldsymbol{x}_{0|0} = \boldsymbol{I}_{10,1}$ and $\Sigma_{0|0}^x = \boldsymbol{Q}$. Here, we have implemented both the conventional nonlinear filtering algorithms (including the EKF and the UKF) and the new SMC algorithms developed in Section 3 (i.e., the SMC-EKF and the SMC-UKF, under either the discrete density approximation or the Gaussian kernel density approximation). In this case QKF needs 4^{10} abscissas and is impractical for this problem. We have also implemented the SIR particle filter where the proposal distribution is based only on the state equation and given by (5). In the SIR particle filter, the number of the sample streams is set as $m = 300$ and the exact initial state $\boldsymbol{x}_o$ is used. In all the proposed SMC algorithms, the number

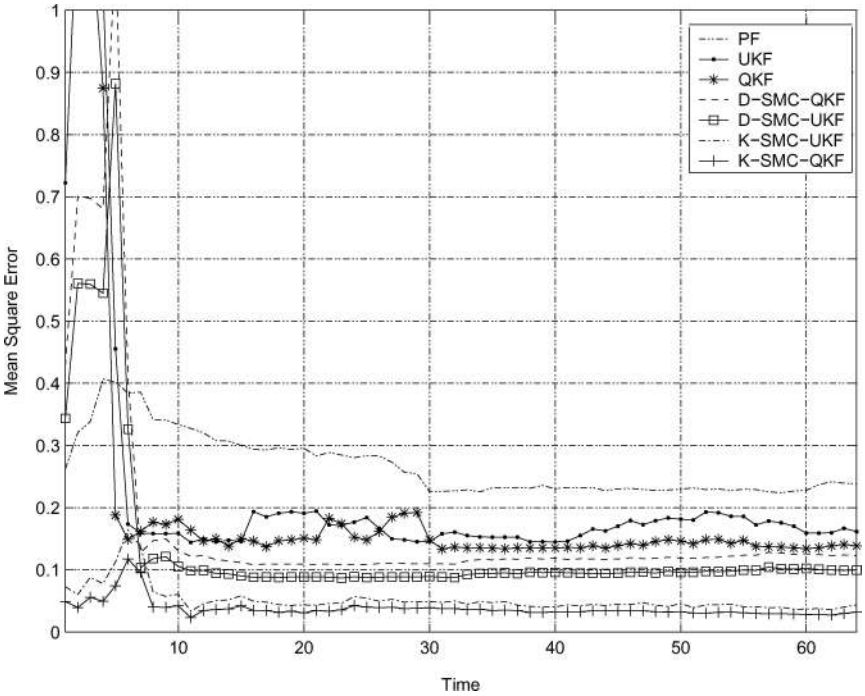

Fig. 4. *The mean square error performance of various nonlinear filtering algorithms for the tracking of User 1's amplitude*

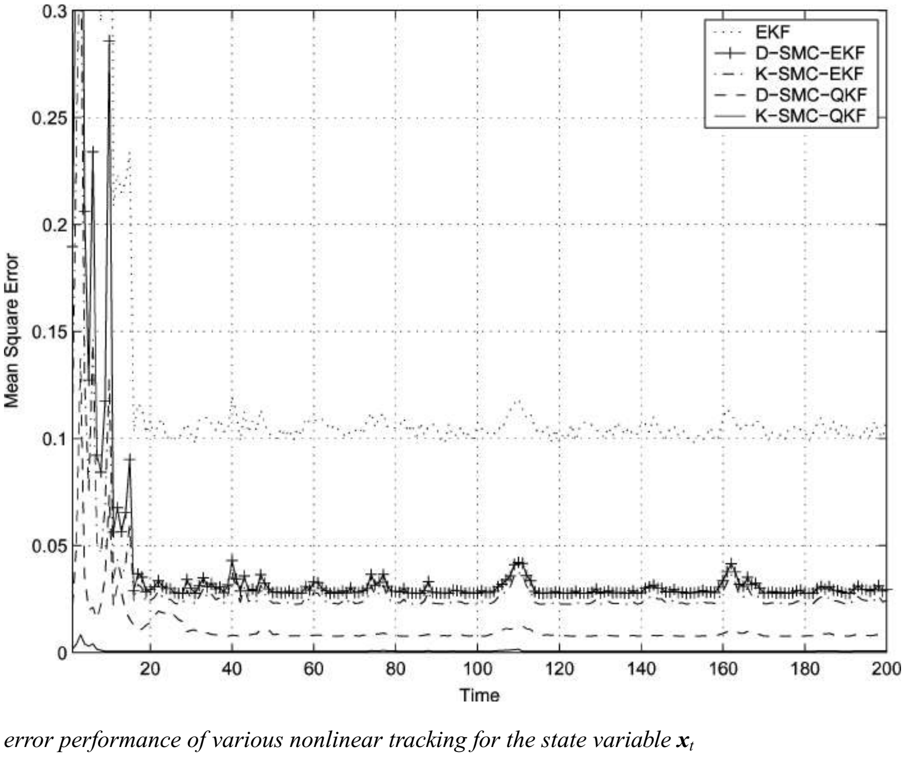

Fig. 5. *The mean-square error performance of various nonlinear tracking for the state variable* $\boldsymbol{x}_t$

of sample streams is set as $m = 50$, and the initial state is set as $p(\boldsymbol{x}_0 \mid \boldsymbol{y}_0) \sim \mathcal{N}(\boldsymbol{x}_{0|0}, \boldsymbol{\Sigma}_{0|0})$.

We run each algorithm over a simulated target trajectory of 200 time instants and calculate the corresponding root mean-square error for each component of the state vector $\boldsymbol{x}_t$. The results are listed in Table 3. Here we only list the estimated first component of the state vector since all components in the vector have nearly the same performance. The running time is also in terms of seconds. It is seen that, the UKF, the EKF and the PF have the similar performance. The second group in Table 3 consists of the two new SMC algorithms under the discrete density approximation (D-SMC-EKF and D-SMC-UKF). It is seen that

Table 3. *The root mean-square error (RMSE) performance of various nonlinear filtering algorithms for the high dimensional problem*

Algorithms	x_1	Time (s)
EKF	6.799	0.198
UKF	6.692	0.552
PF	6.694	43.92
D-SMC-EKF	6.472	73.841
D-SMC-UKF	6.561	141.19
K-SMC-EKF	5.694	74.02
K-SMC-UKF	5.419	145.49

these new SMC algorithms offer slight performance improvement over the existing nonlinear filtering algorithms in the first group. The third group consists of the two new SMC algorithms under the kernel density approximation (K-SMC-EKF and K-SMC-UKF). It is seen that by replacing the discrete density approximation with the Gaussian kernel density approximation, the estimating accuracy can be significantly improved.

5. Discussions and conclusions

In this paper, we proposed several efficient sequential Monte Carlo methods for nonlinear dynamic systems. Although the sequential Monte Carlo methods have the nice property that, in the limit of an infinite number of Markov streams, the approximate representation of the belief state approaches the exact belief state, the efficiency of the SMC methods are closely related to the proposal distribution. Hence, the proposal distributions in this paper are calculated using three efficient nonlinear Kalman filters, namely the extended Kalman filter, unscented Kalman filter and quadrature Kalman filter. The efficiency of these proposed algorithms is shown in the simulations on real time target tracking and multiuser parameter tracking problems.

Furthermore, we discussed a more complex and challenging problem—the representation of the posterior distribution $p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t)$ based on the properly weighted samples. In standard SMC algorithms, the properly weighted samples are viewed as a discrete approximation to the target distribution; each individual samples are propagated based on the Markov chain. On the other hand, we can also construct Kernel representation of the target distribution by using these samples. In particular, these Kernels have adaptive bandwidths, which are inherited from the nonlinear Kalman filters; thus a more efficient and accurate representation of the target distribution is obtained. In other words, the efficiency of the SMC is improved with Kernel representations with adaptive bandwidths.

However, in kernel-based SMC methods, the parameters h_q and h_k in Section 3 are still needed to be further investigated. The first parameter h_q determines how much variation we put in the proposal distribution; whereas the parameter h_k plays the role of adjusting the influence range of the kernel. Therefore, these two parameters are substantial to improve the performance of the SMC methods. As shown in our simulations, certain shrinkage on first parameter ($h_q < 1$) and certain spreading on the second parameter ($h_k > 1$) have been found to be helpful in improving the performance of SMC methods. However, how to adaptively choose these two parameters is still an open problem.

As shown in Section 3.2, the QKF suffered from the curse of dimension; consequently, the SMC employing the QKF is not suitable to solve high dimensional problems although the spare girds built on the Smolyak algorithm (Gerstner and Griebel 1998; Petras 2001) can be employed to mitigate the dimension curse. However, in low dimensional problem, it is often found more accurate than the unscented Kalman filter. Therefore, different algorithms can be applied in different application scenarios. On the other hand, the SMC methods based on the QKF and UKF provide excellent performance and are insensitive to the initial starting points, while the sensitive to the initial points is a critical problem for the EKF and SIR Particle filter. However, the drawback of the proposed novel algorithms is that it is computationally intensive, yet it can be straightforwardly implemented on parallel computation.

References

Agate C.S. and Iltis R. 1999. Statistics of the RSS estiamtion algorithm for gaussian measurement noise. IEEE Trans. on Sign. Proc. 47(1): 22–32.

Alspace D.L. and Sorenson H.W. 1972. Nonlinear bayesian estimation using the gaussian sum approximation. IEEE Transaction on Automatic Control 17: 439–448.

Arulampalam S., Maskell S., Gordan N., and Clapp T. 2002. A tutorial on particle filter for on-line non-linear/non-gaussian bayesian tracking. IEEE Trans. Sig. Proc. 50(2): 174–188.

Avitzour D. 1995. A stochastic simulation Bayesian approach to multitarget tracking. lEE Proceedings on Radar, Sonar and Navigation 142: 41–44.

Caffery J. and Stuber G.L. 2000. Nonlinear multiuser parameter estimation and tracking CDMA system. IEEE Trans. on Commun. 48(12): 2053–2063.

Chen R. and Liu J.S. 2000. Mixture Kalman filters. Journal of the Royal Statistical Society, Series B 62: 493–509.

Chen R., Wang X., and Liu J.S. 2000. Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. IEEE Trans. Inform. Theory 46(6): 2079–2094.

Dellaert F., Fox D., Burgard W., and Thrun S. 1999. Monte Carlo localization for mobile robots. In ICRA.

Dellaert F., Fox D., Burgard W., and Thrun S. 1999. Using the condensation algorithm for robust, vision-based mobile robot localization. In CVPR.

Doucet A., Dreitas N.D., Murphy K., and Russell S. 2000. Rao-blackwellised particle filtering for dynamic bayesian networks. In Sixteenth Conference on Uncertainty in Artificial intelligence, Stanford, pp. 176–183.

Doucet A., de Freitas J.F.G., and Gordon N. 2001. Sequential Monte Carlo in Practice. Cambridge University Press.

Doucet A., Godsill S.J., and Andrieu C. 2000. On sequential simulation-based methods for Bayesian filtering. Statisti. Comput. 10(3): 197–208.

Gordon N.J., Salmon D.J., and Ewing C.M. 1995. Bayesian state estimation for tracking and guidance using the bootstrap filter. AIAA Journal of Guidance, Control and Dynamics 18: 1434–1443.

Gerstner T. and Griebel M. 1998. Numerical integration using sparse grids. Numerical Algorithms 18(4): 209–232.

Handschin J.E. 2003. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. Atomatica, 1970(6): 555–563.

Isard M. and Blake A. 1997. A mixed-state Condensation tracker with automatic model switching. In ECCV.

Isard M. and Blake A. 1998. Condensation—conditional density propogation for visual tracking. In IJCV, vol. 29, pp. 5–28.

Ito K. and Xiong K. 2000. Gaussian filters for nonlinear filtering problems. IEEE Transaction on Automatic Control 45(5): 910–827.

Kong A., Liu J.S., and Wong W.H. 1994. Sequential imputations and Bayesian missing data problems. J. Amer. Statist. Assoc 89: 278–288.

Liu J.S. 2001. Monte Carlo Strategies for Scientific Computing. Springer-Verlag, New York.

Liu J.S. and Chen R. 1995. Blind deconvolution via sequential imputations. Journal of the American Statistical Association 90: 567–576.

Liu J.S. and Chen R. 1998. Sequential Monte Carlo methods for dynamic systems. Journal of the American Statistical Association 93: 1032–1044.

Minka T.P. 2001. Expectation propagation for approximate bayesian inference. In Uncertainty in AP01.

Petras K. 2001. Fast calculation of coefficients in the Smolyak algorithm. Numerical Algorithms 26(2): 93–109.

Pitt M.K. and Shephard N. 1999. Filtering via simulation: auxiliary particle filters. J. Amer. Statist. Assoc. 94(446): 590–601.

Press W., Teukolsky S., Vetterling W., and Flannery B.P. 1992. Numerical Recipes in C: The Art of Scientific Computing, 2nd edition. Cambridge Press, London.

Shephard N. and Pitt M.K. 1997.Likelihood analysis of non-gaussian measurement time series. Biometrika 84: 653–667.

Silverman B.W. 1986. Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York.

Thrun S., Dellaert F., Fox D., and Burgard W. 2001. Robust Monte Carlo localization for mobile robots. In Artifical Intelligence Journel.

Vlasis N., Terwijn B., and Krose B. 2002. Auxiliary partifcle filter robot localization from high-dimensional sensor observations. In IEEE Int. Conf. on Robots and Automation, Wahsington D.C., pp. 7–12.