Particle Filter with Hybrid Proposal Distribution for Nonlinear State Estimation

Fasheng Wang, Yuejin Lin, Tao Zhang, Jingbo Liu
Department of Computer Science and Technology, Dalian Neusoft Institute of Information, Dalian, P.R. China
Email: fashengw@acm.org

***Abstract* – Particle filters have been widely used in solving nonlinear filtering problems. Proposal Distribution design is a key issue for these methods and has vital effects on simulation results. Various proposal distributions have been proposed to improve the performance of particle filters, but practical situations have encouraged the researchers to design better candidate for proposal distributions in order to gain better performance. This paper proposes a hybrid proposal distribution designed for particle filtering framework. The algorithm uses hybrid Kalman filter to generate the proposal distribution, which make efficient use of the latest observations and generates more close approximation of the posterior probability density. The yielded algorithm is named as hybrid Kalman particle filter. In the experiments, a scalar estimation model and a real world application problem are used to evaluate the new algorithm. The experimental results show that the new algorithm outperforms any other algorithm with different proposal distributions. In order to reduce the time consumption of the new algorithm, an improvement strategy, namely partition-conquer strategy, is proposed, in which the particles are partitioned into two parts, with one part drawn from the hybrid Kalman filter, the other part from the transition prior. The validity of the strategy is verified through an experiment.**

***Index Terms*— nonlinear filter, particle filter, proposal distribution, hybrid Kalman particle filter, partition-conquer**

I. INTRODUCTION

Many signal processing problems involve the estimation of the unobserved state of a random dynamical system from a sequence of related observations. When the dynamical system is linear and Gaussian, the optimal solution is given by the well-known Kalman filter. But, many real world applications such as target tracking, positioning, and financial options pricing etc., can not be well represented with linear Gaussian state space models. The nonlinearity of these nonlinear state estimation problems has invalidated the Kalman filter in practice. There are a variety of solutions to these problems, among which the extended Kalman filter (EKF) is well known [1]. The EKF is based on the Taylor series expansion of the nonlinear functions around the estimates of the state. However, the EKF is likely to diverge when the linearizing approximation methods gives poor representations of the nonlinear functions.

Manuscript received December 10, 2010; revised May 22, 2011; accepted June 1, 2011.

The unscented Kalman filter (UKF), also known as sigma point Kalman filter, is another kind of interesting solution, which is founded on the fact that it is easier to approximate a Gaussian distribution than an arbitrary nonlinear function [2,3]. Unlike the EKF, the UKF does not use the approximated models of the nonlinear dynamic system, but approximates the distribution of the state random variables by using a set of carefully chosen sigma points. It is shown that the UKF can acquire more accurate estimation results than the EKF, but might lead to notable errors for non-Gaussian distributions.

Recently, many researchers have focused on another solution strategy – sequential Monte Carlo methods, also known as particle filters. The aim of these methods is to approximate the filtering probability density function of the system state using a collection of weighted samples (also call particles) in the state-space. These methods can generate a good representation of the posterior distribution of the system states, so that any statistics such as the mean, the modality and the variance can be easily calculated [4]. An important issue in these methods is the selection of the proposal distributions, which can give vital effects on simulation results and forms the main focus of this paper. In this paper, a new proposal distribution generating scheme for the particle filtering framework is proposed, and the algorithm obtained is named as the hybrid Kalman particle filter (HKPF). This scheme uses hybrid Klman filter (HKF) to generate proposal distribution. In this algorithm, each particle is updated by the UKF and the EKF sequentially. Through this procedure, efficient use of the latest observations is made, which consequently improves the performance of particle filters. For comparison purposes, in the experiments, we present a synthetic, scalar estimation problem to show that the proposed algorithm outperforms the existing particle filters with different proposal distributions. We also address the application of the new algorithm on a real world problem concerning the pricing of financial options. In order to reduce the time consumption of the proposed algorithm, an improving strategy is propounded in which the particles used in the algorithm are partitioned into two parts and are drawn from different proposal distributions. This strategy makes a trade-off between accuracy and time-consumption.

The paper is organized as follows. The related work is introduced in Section 2. In section 3, a brief introduction of the standard particle filter will be given. In section 4 firstly, a brief introduction of the EKF and the UKF is

given, and then the hybrid Kalman filter will be introduced, followed by the hybrid Kalman particle filter algorithm. In the last part of this section, some experimental results are drawn. The performance of the new particle filtering algorithm is compared with that of several other particle filters. Section 5 gives an improving strategy to the hybrid Kalman particle filter aiming at circumventing the high time cost of the HKPF algorithm. Finally, Section 6 concludes this paper.

II. RELATED WORK

Particle filters were originally introduced in the early 1950s by physicists. But its actual development and implementation began in the 1990s, because of the powerful ability of computation provided by computers which made this method a reality. It is commonly considered by the researchers that particle filters were first proposed in [5]. This algorithm is known variously as bootstrap filtering, the condensation algorithm, interacting particle approximation and survival of the fittest.

Since particle filters were proposed, many researchers have been working in this area and have acquired several significant results. Liu and Chen [6] proposed to draw particles based on a tail importance density which was also used similarly by Guo et al. [7]. N. Bergeman [8] introduced the effective sample size to measure the degeneracy problem, which was the main drawback of particle filtering algorithms. In order to solve the degeneracy problem, researchers have proposed several solutions. Resampling is an effective solution to the degeneracy problem [1,4]. Liu and Chen [6] proposed a residual sampling method. Kitagawa [9] proposed a systematic resampling method which was preferred by other researchers. Joaquín Míguez [10] investigated distributed resampling algorithm in detail, and applied it to vehicle navigation and tracking the dynamic variables of the chaotic Lorenz system. Jeroen D. Hol et al. [11] gave a detailed introduction of several resampling methods. Although resampling can improve the performance of particle filters, some other practical problems appear, which are particle impoverishment problems. Gilks and Berzuini [12] proposed to rejuvenate particles based on Markov chain Monte Carlo (MCMC), and there were some other similar contributions like Doucet [13] and Chopin [14]. Benlian Xu et al. [15] combined the particle filter and ant colony optimization (ACO) algorithm and devised an ant estimator which was applied to target tracking. In their work, the ACO algorithm is extended to the parameter estimation field. They proposed to replace the resmapling process in particle filter with the stochastic searching behaviour.

The second method to reduce degeneracy problem is to choose a good proposal distribution which is used to generate the importance sampling density, and this forms the main topic of this paper. Doucet et al. [16] has proved that the optimal proposal distribution that can minimize the variance of the particle weights is $q(x_k \mid x_{0:k-1}, y_{1:k}) = p(x_k \mid x_{0:k-1}, y_{1:k})$. The choice of proposal distribution has also been advocated by other researchers, e.g., Liu and Chen [17], Kong, Liu and Wong [18], etc.

The optimal proposal distribution can not be used efficiently in practice because it suffers some drawbacks [1]. And researchers have worked on this topic and have proposed several suboptimal candidates. The most popular choice is the transition prior, that is, $q(x_k \mid x_{0:k-1}, y_{1:k}) = p(x_k \mid x_{k-1})$, which is proposed by Gordon et al. [5]. This proposal function is easy to implement. Particle filter using this proposal distribution with resampling step is usually named as the generic particle filter [4]. The generic particle filter has been used successfully in visual tracking [19]. However, as a result of not incorporating the most recent observations, the transition prior is not always effective. Doucet et al. [20] analysed this problem in detail and proposed another approximation method, that is, local linearization technique. This is a popular method for devising proposal distributions that approximate the optimal importance distribution, by incorporating the most current observation with the optimal Gaussian approximation of the state. It relies on the first order Taylor expansions of the likelihood and transition prior. Within the particle filtering framework, a separate EKF is used to generate a Gaussian proposal distribution for each particle, i.e., $q(x_k \mid x_{0:k-1}, y_{1:k}) = \mathcal{N}(\bar{x}_k, \hat{P}_k)$. The algorithm is called extended Kalman particle filter (EKPF) [4]. However, this choice of proposal distribution can be ineffective due to inaccuracies introduced by linearization. Merwe et al. [4] has pointed out the drawbacks of the EKF proposal distribution and proposed a better candidate, the UKF. Because UKF can calculate the posterior covariance accurately to the 3rd order, this merit makes the UKF a better choice for more accurate proposal distribution generation within the particle filtering framework. The resulting algorithm is named as unscented particle filter (UPF) [4].

Li Liang-qun et al. [21] gave another choice of proposal distributions. In their work, the iterated extended Kalman filter (IEKF) is used in the particle filtering framework to generate the proposal distribution. IEKF can reduce the linearization errors introduced by the EKF through iteratively updating the estimations $\bar{x}_k$ and $\bar{x}_{k|k-1}$, because the accuracy of the EKF is determined by $\left\|x_{k-1} - \bar{x}_{k-1}\right\|^2$ and $\left\|x_k - \bar{x}_{k|k-1}\right\|^2$. This makes the IEKF a better candidate for proposal distribution generation. The yielded particle filter is named as iterated extended Klman particle filter (IEKPF) [21]. For some nonlinear systems, IEKPF outperforms the UPF and the EKPF.

Kiyoshi Nishiyama [22] proposed to generate proposal distribution using the extended H_∞ filter (EHF), and the resulting filter is called the extended H_∞ particle filter. This particle filter had better estimation accuracy than the EKPF and less computational cost than the UPF. It obviously made a trade-off between accuracy and

computational cost. In addition, WANG [23] proposed a mixed particle filter which combines two filters for proposal distribution and performs better than the IEKPF. But the time consumption of the new algorithm is very high.

Over the past several years, particle filters have been applied in a variety of fields. M. Isard [19] firstly applied particle filtering algorithm (condensation) to solving visual tracking problems. Since then, other researchers have applied some improved particle filters in vision tracking [24,25,26]. Dr. Rickard Karlsson made particle filter more attractive by applying this method in navigation under the sea, which could make the global satellite system missing their places for under-water positioning [27]. On the other hand, the particle filters have also achieved great success in robot area, especially in robot localization problems [28,29].

Yihua Yu et al. [30] investigated maneuvering target tracking problem using particle filters. In their work, a suitable model was proposed to characterize the maneuvering acceleration and a state space model was developed to describe the maneuvering target tracking problem. Some tracking algorithms based on particle filters were developed and applied to single and multi target tracking problems. Branko Ristic et al. [31] addressed the problem of bearing-only maneuvering target tracking using particle filters.

Recently, M.H. Jaward et al. [32] propounded to apply particle filters to contour tracking for airborne emission of contaminant clouds. They proposed to estimate the contour boundary positions using a set of particle filters, and obtained importance result through experiments. In addition, particle filters are also successfully applied to positioning and tracking [33,34], communication [35], signal processing [36,37], financial and economics [38,39,40], etc.

Arulampalam et al. [1] gave a tutorial of particle filters for online nonlinear non-Gaussian Bayesian tracking, in which they summarized several popular types of particle filters with their properties and adaptation. Doucet et al. [41] discussed in detail about the particle filters, improved particle filters and corresponding applications in various fields. O. Cappé et al. [42] summarized recent advances of particle filters and reviewed a range of existing core topics in particle filters. C. Andrieu et al. [43] gave a tutorial of particle filters for solving problems like change detection, parameter estimation, and control.

III. STANDARD PARTICLE FILTER

Considering the following nonlinear system,

$$x_k = f_k(x_{k-1}, v_{k-1}) \tag{1}$$

$$y_k = h_k(x_k, u_k) \tag{2}$$

Where x_k denotes the system state, and y_k denotes the observation at time k. The functions $f(.)$ and $h(.)$ represent the system transition model and the measurement model respectively. The process noise v_k and the measurement noise u_k are assumed independent with known distributions. The prior knowledge of the initial state is given by the probability distribution $p(x_0)$. Our aim is to learn more about the unknown state variables with the historical observations.

According to the basic idea of particle filters, we use $\{x_{0:k}^i, w_k^i\}_{i=1}^{N_p}$ to denote a random measure used to represent the posterior density function $p(x_{0:k} \mid y_{1:k})$. $\{x_{0:k}^i, i = 0,..., N_p\}$ is a set of support particles with associated weights $\{w_k^i, i = 0,..., N_p\}$, and $x_{0:k}$ shows the states up to time k. When the number of particles N_p tends to infinity, the posterior density function can be approximated by,

$$p(x_{0:k} \mid y_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i) \tag{3}$$

where $\delta(\cdot)$ is the Dirac Delta function.

Many particle filters rely on the principle of importance sampling. Because it is usually difficult to draw particles directly from the posterior density $p(x_{0:k} \mid y_{1:k})$ to represent the density function, particles can be alternatively generated from a proposal distribution density function $q(x_{0:k} \mid y_{1:k})$, which is also known as an importance function, and can be decomposed as

$$q(x_{0:k} \mid y_{1:k}) = q(x_k \mid x_{0:k-1}, y_{1:k}) q(x_{0:k-1} \mid y_{1:k-1}) \tag{4}$$

The weights are computed recursively according to

$$w_k^i = w_{k-1}^i \frac{p(y_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, y_k)} \tag{5}$$

Detailed derivation of the formulations can be found in [1,3,37].

Suppose we have gained the approximation of the posterior distribution at time $k-1$, that is, $p(x_{0:k-1} \mid y_{1:k-1})$ can be approximated by the discrete random measure $\{x_{0:k-1}^i, w_{k-1}^i\}_{i=1}^{N_p}$, which is drawn from $q(x_{0:k-1} \mid y_{1:k-1})$. Next we aim to obtain $\{x_{0:k}^i, w_k^i\}_{i=1}^{N_p}$ using the new observation y_k. So long as we get the particles x_k^i and augment them onto the old trajectory $\{x_{0:k-1}^i, w_{k-1}^i\}_{i=1}^{N_p}$, the new trajectory can be acquired. The new particles are generated as follows:

$$x_k^i \sim q(x_k \mid x_{0:k-1}^i, y_{0:k}) \tag{6}$$

The weights w_k^i can be obtained by using Equ.(5).

A thorny problem of particle filters is the degeneracy phenomenon, that is, after several iterations all but one particle would probably have negligible weights. This phenomenon will deteriorate the performance of particle

filters. In order to solve the problem, a resampling scheme is introduced [20]. During the resampling process, the particles with low importance weights are eliminated, and those particles with high importance weights are multiplied. Choosing a good proposal distribution can also weaken the degeneracy phenomenon as we have discussed in section 2. For more details, please refer to [1,20,41].

The standard particle filter with resampling step can be shown as Algorithm 1.

Algorithm 1. Standard Particle Filter

Step 1. Initialization. k =0

FOR $i = 1,..., N_p$

Draw the states x_0^i from the prior $p(x_0)$;

END FOR

Step 2. FOR $k = 1,2,...$

(1) FOR $i = 1,..., N_p$

Draw particle from the proposal distribution $x_k^i \sim q(x_k^i \mid x_{k-1}^i, y_k)$;

Assign the particle a weight according to Equation (5);

END FOR

(2) FOR $i = 1,..., N_p$

Normalize the weights $w_k^i = w_k^i / \sum_{j=1}^{N_p} w_k^j$;

END FOR

(3) Resample

- Eliminate the samples with low importance weights and multiply the samples with high importance weights, to obtain N_p random sample $x_{0:k}^i$ which are approximately distributed according to $p(x_{0:k} \mid y_{1:k})$;
- FOR $i = 1,..., N_p$, let w_k^i =1/ N_p, END FOR

(4) Output: calculated the required estimations using the particle set.

END FOR

Step 3. $k = k + 1$, go to Step 2 or end the algorithm.

IV. HYBRID KALMAN PARTICLE FILTER

Before the hybrid Kalman particle filter is introduced, firstly the extended Kalman filter and the unscented Kalman filter will be briefly introduced.

A. Extended Kalman Filter

The extended Kalman filter (EKF) is a minimum mean-square-error (MMSE) estimator based on the Taylor series expansions of the nonlinear functions $f(.)$ and $h(.)$ around the current estimates. In the EKF, the state distribution is represented by using a Gaussian random variable. It only uses the linear expansion terms to compute the estimations. For more details of EKF, please refer to [2,44].

For systems with strong nonlinearity, the EKF tends to yield large errors in the estimated data of the states, because it only uses the first order terms of the Taylor series expansion of the nonlinear functions. The method can only achieve first-order accuracy in estimations.

B. Unscented Kalman Filter

The unscented Kalman filter (UKF) is also a MMSE estimator. In the UKF, the state distribution is still represented by a Gaussian random variable, but it is now specified by a set of carefully chosen sample points (sigma points), which can completely capture the true mean and the true covariance of the state variable. When these sample points are propagated through the true nonlinear models, the third order accuracy of the Taylor series expansion can be achieved.

The UKF is a straightforward extension of the unscented transformation [2,45,46]. The unscented transformation is used for calculating the statistics of a random variable which undergoes a nonlinear transformation. Suppose the dimension of a state random variable x is L, and it has the mean $\bar{x}$ and the covariance P_x. The variable needs to be propagated through a nonlinear function $y = g(x)$. Firstly, $2L+1$ weighted sigma points are carefully chosen, after which the nonlinear function is applied to each point to yield a cloud of transformed trajectory. The sigma points $\chi^{(i)}$ with the weights $W^{(i)}$ are selected according to the following formulas.

$$\chi_0 = \bar{x} \tag{7}$$

$$\chi^{(i)} = \bar{x} + (\sqrt{(L+\lambda)P_x})^{(i)}, \quad i = 1,...,L \tag{8}$$

$$\chi^{(i)} = \bar{x} - (\sqrt{(L+\lambda)P_x})^{(i)}, \quad i = L+1,...,2L \tag{9}$$

$$W_m^{(0)} = \lambda/(L+\lambda) \tag{10}$$

$$W_c^{(0)} = \lambda/(L+\lambda) + (1-\alpha^2+\beta) \tag{11}$$

$$W_m^{(i)} = W_c^{(i)} = 1/(2(L+\lambda)), \quad i = 1,...,2L \tag{12}$$

The meanings of some notations in the above are given as follows.

$\lambda = \alpha^2(L+\kappa) - L$. λ is a scaling parameter.

$(\sqrt{(L+\lambda)P_x})^{(i)}$ denotes the i^{th} row of the matrix square root.

The parameter α decides the diffusion of the sigma points around $\bar{x}$, usually set to be a small positive value.

The parameter κ is a secondary scaling parameter, usually set to be zero.

The parameter β is used for incorporating the prior knowledge of the state distribution.

The notation $W_m^{(i)}$ is the weight for calculating the mean, while $W_c^{(i)}$ is the weight for calculating the covariance.

The sigma points are propagated through the nonlinear function $Y^{(i)} = g(\chi^{(i)})$, for $i = 0,...,2L$. As a result,

the mean and covariance of y are approximated by using a set of weighted sample mean and covariance of the transformed trajectory.

$$\bar{y} = \sum_{i=0}^{2L} W_m^{(i)} Y^{(i)} \quad (13)$$

$$P_y = \sum_{i=0}^{2L} W_c^{(i)} \{Y^{(i)} - \bar{y}\} \{Y^{(i)} - \bar{y}\}^T \quad (14)$$

In the UKF, the state random variable is redefined as the combination of the original state and noise variables $x_{k,a} = [x_k^T \quad v_k^T \quad u_k^T]^T$, where the subscript a indicates that $x_{k,a}$ is the augmented state at time *k*. In the unscented transformation, the selection method (Equ.(7)-Equ.(12)) of those sigma points is applied to this new augmented state random variable to calculate the corresponding sigma matrix $\chi_{k,a}$. The UKF algorithm can be summarized as in Algorithm 2. For more details of the UKF, please refer to [2,45,46].

Algorithm 2. The UKF algorithm

Step 1. Initialization

$\bar{x}_0 = E(x_0)$

$P_0 = E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T]$

$\bar{x}_{0,a} = E[x_{0,a}] = [\bar{x}_0^T, 0, 0]^T$

$P_{0,a} = E[(x_{0,a} - \bar{x}_{0,a})(x_{0,a} - \bar{x}_{0,a})^T] = diag(P_0, Q, R)$

Step 2. FOR $k = 1,2,...$

(1) Select Sigma points

$\chi_{k-1,a} = [\bar{x}_{k-1,a} \quad \bar{x}_{k-1,a} \pm \sqrt{(n_a + \lambda) P_{k-1,a}}]$

(2) Predict

$\chi_{k|k-1,x} = f(\chi_{k-1,x}, \chi_{k-1,v})$.

$\bar{x}_{k|k-1} = \sum_{j=0}^{2n_a} W_m^{(j)} \chi_{k|k-1,x}^{(j)}$

$P_{k|k-1} = \sum_{j=0}^{2n_a} W_c^{(j)} [\chi_{k|k-1,x}^{(j)} - \bar{x}_{k|k-1}][\chi_{k|k-1,x}^{(j)} - \bar{x}_{k|k-1}]^T$

$Y_{k|k-1} = h(\chi_{k|k-1,x}, \chi_{k-1,u})$

$\bar{y}_{k|k-1} = \sum_{j=0}^{2n_a} W_m^{(j)} Y_{k|k-1}^{(j)}$

(3) Update

$P_{\tilde{y}_k \tilde{y}_k} = \sum_{j=0}^{2n_a} W_c^{(j)} [y_{k|k-1}^{(j)} - \bar{y}_{k|k-1}][y_{k|k-1}^{(j)} - \bar{y}_{k|k-1}]^T$

$P_{x_k y_k} = \sum_{j=0}^{2n_a} W_c^{(j)} [\chi_{k|k-1}^{(j)} - \bar{x}_{k|k-1}][y_{k|k-1}^{(j)} - \bar{y}_{k|k-1}]^T$

$K_k = P_{x_k y_k} P_{\tilde{y}_k \tilde{y}_k}^{-1}$

$\bar{x}_k = \bar{x}_{k|k-1} + K_k (y_k - \bar{y}_{k|k-1})$

$P_k = P_{k|k-1} - K_k P_{\tilde{y}_k \tilde{y}_k} K_k^T$

END FOR

Step 3. $k = k + 1$, go to step 2 or end the algorithm.

Where $x_a = [x^T v^T u^T]^T$, $\chi_a = [\chi_x^T \chi_v^T \chi_u^T]$, and $n_a = n_x + n_v + n_u$ is the dimension of the augmented state. Q and R represent the process noise covariance and the measurement noise covariance respectively. The meanings of some notations in Algorithm 2 are as follows.

x_a is the augmented state variable.

$\chi_{k,a}$ is the corresponding sigma matrix at time k.

$\chi_{k,x}$ is the sigma vector with respect to the state at time k. $\chi_{k,v}$ and $\chi_{k,u}$ are the sigma vectors with respect to the noise at time k.

K is the Kalman gain.

$\chi_{k|k-1}^{(j)}$ is the j th element of $\chi_{k|k-1,x}$.

C. The Hybrid Kalman Filter

The hybrid Kalman filter (HKF) is a combination of the UKF and the EKF. Like the UKF and the EKF, the state distribution in the HKF is represented by a Gaussian random variable and is specified by a set of deterministically chosen sample points. At time k, the UKF is firstly used to obtain the estimates of the state mean $\bar{x}_{k,ukf}$ and the covariance $P_{k,ukf}$, then the EKF is used to update the estimations $\bar{x}_{k,ukf}$ and $P_{k,ukf}$ that are obtained by the UKF, and the final state mean $\bar{x}_{k,ekf}$ and the covariance $P_{k,ekf}$ are obtained. Using HKF, we can achieve more accurate posterior mean and covariance estimations than using the UKF. The HKF algorithm can be summarized as in Algorithm 3.

Algorithm 3. The HKF algorithm

Step 1. Initialization

The same as in Algorithm 2.

Step 2. FOR $k = 1,2,...$

(1) Using the UKF to obtain state mean estimate $\bar{x}_{k,ukf}$ and the covariance $P_{k,ukf}$.

(2) Using the EKF to obtain the updated estimates.

$\bar{x}_{k|k-1,ekf} = f(\bar{x}_{k-1}) = f(\bar{x}_{k,ukf})$

$P_{k|k-1,ekf} = F_k P_{k,ukf} F_k^T + G_k Q_k G_k^T$

$K_k = P_{k|k-1,ekf} H_k^T [U_k R_k U_k^T + H_k P_{k|k-1,ekf} H_k^T]^{-1}$

$P_{k,ekf} = P_{k|k-1,ekf} - K_k H_k P_{k|k-1,ekf}$

$\bar{x}_{k,ekf} = \bar{x}_{k|k-1,ekf} + K_k (y_k - h(\bar{x}_{k|k-1,ekf}))$

(3) Let $\bar{x}_k = \bar{x}_{k,ekf}$, and $P_k = P_{k,ekf}$

END FOR

Step 3. $k = k + 1$, go to step 2 or end the algorithm.

Where,

K is the Kalman gain

Q is the covariance of the process noise.

R is the covariance of the measurement noise.

F and G are the Jacobians of the process model.

H and U are the Jacobians of the measurement model.

D. The Hybrid Kalman Particle Filter

The HKF inherits the excellent properties of the UKF, and can make efficient use of the latest observations, which make it very attractive for the generation of proposal distribution within the particle filtering framework. The new particle filter results from a HKF for proposal distribution generation is called hybrid Kalman particle filter (HKPF).

At time k, the UKF is firstly used to update the particles, and to obtain the state estimate $\bar{x}^i_{k,ukf}$ and the corresponding covariance estimate $P^i_{k,ukf}$, then the particles are updated using the EKF with $\bar{x}^i_{k,ukf}$ and $P^i_{k,ukf}$. After the EKF-update, the final state and covariance estimates $\bar{x}^i_k$ and $\hat{P}^i_k$ of time step k are obtained. Using the estimates, the required proposal distribution $\mathcal{N}(\bar{x}^i_k, \hat{P}^i_k)$ is formed. Here, samples can be drawn from the approximated distribution $\mathcal{N}(\bar{x}^i_k, \hat{P}^i_k)$.

The HKPF particle filter algorithm is summarized as Algorithm 4.

Algorithm 4. The HKPF algorithm

Step 1. Initialization: $k = 0$

FOR $i = 1,..., N_p$

Draw the particles x^i_0 from the prior $p(x_0)$ and set:

$$\bar{x}^i_0 = E(x^i_0)$$

$$P^i_0 = E[(x^i_0 - \bar{x}^i_0)(x^i_0 - \bar{x}^i_0)^T]$$

$$\bar{x}^i_{0,a} = E[x^i_{0,a}] = [(x^i_0)^T, 0, 0]^T$$

$$P^i_{0,a} = E[(x^i_{0,a} - \bar{x}^i_{0,a})(x^i_{0,a} - \bar{x}^i_{0,a})^T] = diag(P^i_0\, Q\, R)$$

END FOR

Step 2. FOR $k = 1,2,...$

(1) FOR $i = 1,..., N_p$

(a) Update the particles using the UKF

▪ Calculate the sigma points

$$\chi^i_{k-1,a} = [\bar{x}^i_{k-1,a} \quad \bar{x}^i_{k-1,a} \pm \sqrt{(n_a + \lambda) P^i_{k-1,a}}\,]$$

▪ Propagate samples into future and compute the one-step-ahead estimates:

$$\chi^i_{k|k-1,x} = f(\chi^i_{k-1,x}, \chi^i_{k-1,v})$$

$$Y^i_{k|k-1,ukf} = h(\chi^i_{k|k-1,x}, \chi^i_{k-1,u})$$

$$\bar{x}^i_{k|k-1,ukf} = \sum_{j=0}^{2n_a} W^{(j)}_m \chi^{(j)i}_{k|k-1,x}$$

$$P^i_{k|k-1,ukf} = \sum_{j=0}^{2n_a} W^{(j)}_c [\chi^{(j)i}_{k|k-1,x} - \bar{x}^i_{k|k-1,ukf}][\chi^{(j)i}_{k|k-1,x} - \bar{x}^i_{k|k-1,ukf}]^T$$

$$\bar{y}^i_{k|k-1,ukf} = \sum_{j=0}^{2n_a} W^{(j)}_m Y^{(j)i}_{k|k-1,ukf}$$

▪ Incorporate the new observation y_k, and update the on-step-ahead estimates to obtain $\bar{x}^i_{k,ukf}$

$$P_{y_k y_k} = \sum_{j=0}^{2n_a} W^{(j)}_c [Y^{(j)i}_{k|k-1,ukf} - \bar{y}^i_{k|k-1,ukf}][Y^{(j)i}_{k|k-1,ukf} - \bar{y}^i_{k|k-1,ukf}]^T$$

$$P_{x_k y_k} = \sum_{j=0}^{2n_a} W^{(j)}_c [\chi^{(j)i}_{k|k-1} - \bar{x}^i_{k|k-1,ukf}][Y^{(j)i}_{k|k-1,ukf} - \bar{y}^i_{k|k-1,ukf}]^T$$

$$K_{k,ukf} = P_{x_k y_k} P^{-1}_{y_k y_k}$$

$$\bar{x}^i_{k,ukf} = \bar{x}^i_{k|k-1,ukf} + K_{k,ukf}(y_k - \bar{y}^i_{k|k-1,ukf})$$

$$P^i_{k,ukf} = P^i_{k|k-1,ukf} - K_{k,ukf} P_{y_k y_k} K^T_{k,ukf}$$

(b) Use the EKF to update the estimations obtained through UKF update process

▪ Compute one-step-ahead estimates of the state and the covariance:

$$\bar{x}^i_{k|k-1,ekf} = f(\bar{x}^i_{k,ukf})$$

$$P^i_{k|k-1,ekf} = F^i_k P^i_{k,ukf} (F^i_k)^T + G^i_k Q_k (G^i_k)^T$$

$$K_{k,ekf} = P^i_{k|k-1,ekf} (H^i_k)^T [U^i_k R_k (U^i_k)^T + H^i_k P^i_{k|k-1,ekf} (H^i_k)^T]^{-1}$$

Compute the updated mean and the covariance estimates:

$$P^i_{k,ekf} = P^i_{k|k-1,ekf} - K_k H^i_k P^i_{k|k-1,ekf}$$

$$\bar{x}^i_{k,ekf} = \bar{x}^i_{k|k-1,ekf} + K_{k,ekf}(y_k - h(\bar{x}^i_{k|k-1,ekf}))$$

▪ Let $\bar{x}^i_k = \bar{x}^i_{k,ekf}$, $\hat{P}^i_k = P^i_{k,ekf}$

(c) Draw $x^i_k \sim q(x^i_k \mid x^i_{0:k-1}, z_{1:k}) = \mathcal{N}(\bar{x}^i_k, \hat{P}^i_k)$.

(d) Assign the particle a weight w^i_k according to Equ.(5).

END FOR

(2) Normalize the weights

FOR $i = 1,..., N_p$, $w^i_k = w^i_k / \sum_{j=1:N_p} w^j_k$, ENDFOR

(3) Resample.

(4) Output: calculate the required estimations using the particle set.

END FOR

Step 3. $k = k + 1$, go to Step 2 or end the algorithm.

E. Experimental Results

In this part, we present the simulation results of the HKPF and give a performance comparison between the HKPF and several other existing filters, including the generic particle filter (PF), EKPF, UPF, IEKPF. The first problem is a synthetic, scalar estimation problem and the second is a real world applications In the second experiment, we will apply the HKPF to pricing options, and compare the performance of different particle filters with different proposal distributions.

(1) Scalar estimation problem

Suppose the nonlinear process model and measurement model are:

$$x_k = 1 + \sin[0.04\pi(k-1)] + 0.5 x_{k-1} + v_{k-1} \quad (15)$$

$$z_k = \begin{cases} 0.2 x^2_k + u_k & k \le 30 \\ 0.5 x_k - 2 + u_k & k > 30 \end{cases} \quad (16)$$

Where v_k is a Gamma $\varsigma_a(3,2)$ random variable modelling the process noise, and the measurement noise u_k is drawn from a Gaussian distribution $\mathcal{N}(0,0.00001)$. In this experiment 200 particles are used and the program is repeated 100 times for time-steps k =1,..,60. The unscented transformation parameters are set to be α=1, β=0, and κ=2.

Fig.1 shows the estimates of the system state generated from a single run of different filters. It is shown that the estimates of the particle filter and the EKPF deviate from the true states very large at some time steps, but the UPF, the IEKPF and the HKPF can improve the performance. Table 1 gives states estimation results of different filters. From the means and the variances of the mean square error (MSE), we can clearly see that the HKPF gives the best performance among these filters, with the lowest mean 0.01131 and the lowest variance about 0.0004.

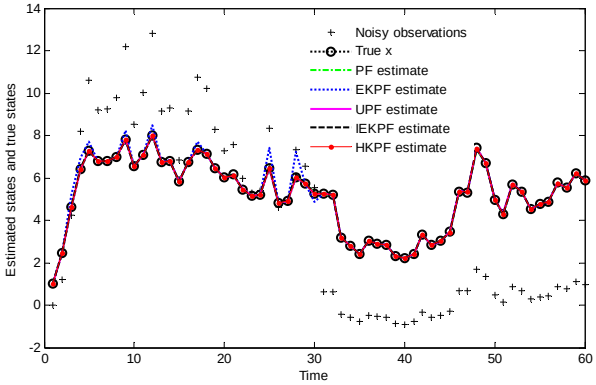

Figure 1. True state and estimated state (posterior mean) generated by different particle filters

Table 1. State estimation results. Means and variances of MSE over 100 Independent Runs

Algorithm	MSE	
	mean	variance
PF	0.43901	0.046876
EKPF	0.30537	0.0088812
UPF	0.068102	0.00663545
IEKPF	0.039418	0.00047719
HKPF	0.01131	0.00039514

Fig.2 shows the MSE in each independent runs for different filters. In this figure, the bottom real line with round shape is the HKPF performance line. It also obviously shows that the HKPF gives the best results almost at every independent run.

The time consumption of the particle filters will be discussed in section 5.

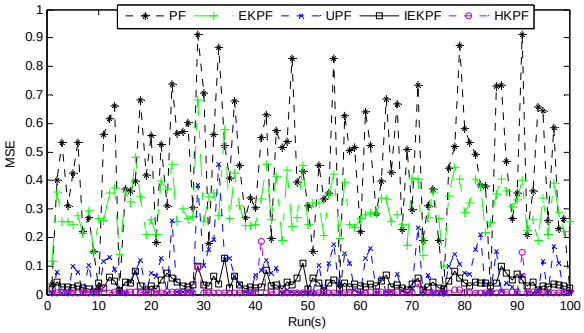

Figure 2. Estimation MSEs of different particle filters at each run

(2) Real world application

In this example, we will use the HKPF to solve the real world problem, pricing financial options, to predict the call and the put option prices, and compare its performance with the other four particle filters.

The Black-Scholes partial differential equation is the main industrial standard for option pricing [47]. The related parameters in the equation are: current value of an option f_o to the current value of underlying cash product S, the volatility of the cash product σ, and the risk-free interest rate r. More information about the model can be found in [48].

$$\frac{\partial f_o}{\partial t} + rS\frac{\partial f_o}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f_o}{\partial^2 S} = rf_o \tag{17}$$

The system model used is the same as in [40], where the prices of the call option and put option are calculated by,

$$C_p = S\mathcal{N}_c(d_1) - Xe^{-rt_m}\mathcal{N}_c(d_2) \tag{18}$$

$$P_p = -S\mathcal{N}_c(d_1) + Xe^{-rt_m}\mathcal{N}_c(-d_2) \tag{19}$$

Here, C_p is the call option price, P_p the put option price, X the strike price, t_m is time to maturity, $\mathcal{N}_c(.)$ is the cumulative normal distribution, and d_1 and d_2 are computed as follows,

$$d_1 = \frac{\ln(S/X) + (r+\sigma^2/2)t_m}{\sigma\sqrt{t_m}} \tag{20}$$

$$d_2 = d_1 - \sigma\sqrt{t_m} \tag{21}$$

We use the state-space representation to model the system given by the equation (18) and (19) as in [48]. r and σ are treated as the hidden states, C_p and P_p are the output observations, t_m and S are input observations. In this experiment, five pairs of call and put option contracts on the British FTSE 100 index (from Feb. 1994 to Dec. 1994) are used to evaluate the HKPF algorithm. The strike prices of the five contracts are: 2925, 3025, 3125, 3225, and 3325. One hundred particles are used, and the program is repeated 100 times. The one-step-ahead normalized square errors (NSE) obtained using different particle filters on a pair of options with strike price 3025 is compared.

$$NSE_{C_p} = \sqrt{\sum_k (C_p^{(k)} - \hat{C}_p^{(k)})^2} \tag{22}$$

$$NSE_{P_p} = \sqrt{\sum_k (P_p^{(k)} - \hat{P}_p^{(k)})^2} \tag{23}$$

Here, $\hat{C}_p^{(k)}$ and $\hat{P}_p^{(k)}$ denotes the one-step-ahead predictions of the call and put prices at the k th day.

Figure 3 shows the predictions of the call prices and the put prices of different particle filters in an independent run. We plot the predictions from the 20th day to 204th day. The red dashed line is the performance of the HKPF. Figure 4 shows a zoomed version of Figure

3 which clearly shows that the HKPF performs better than the other algorithms.

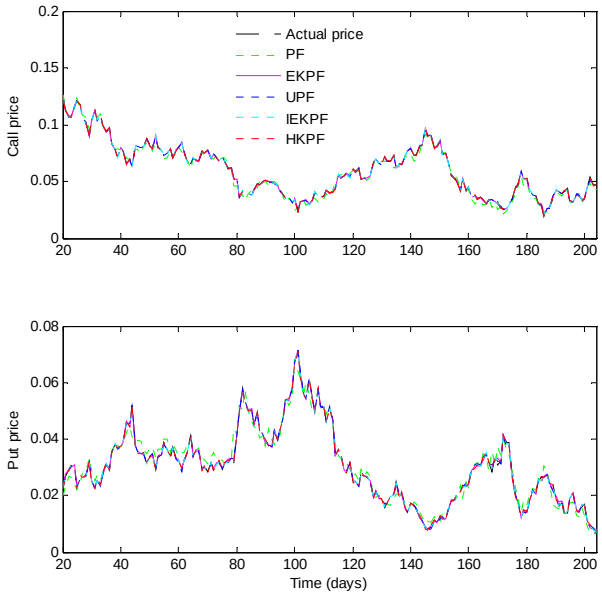

Figure 3. Predicted prices of the options obtained by different algorithms.

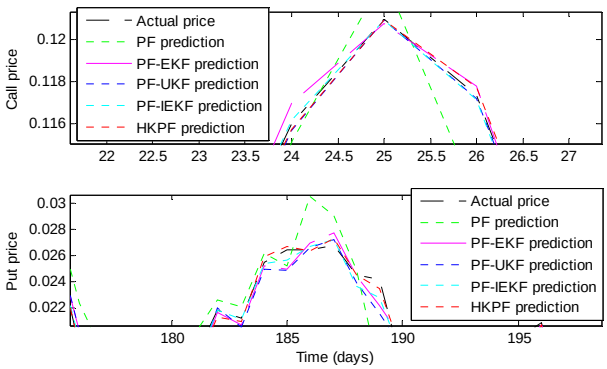

Figure 4. Zoomed version of figure 3.

In Table 2, the average one-step-ahead NSEs over 100 runs obtained with each algorithm are compared. The NSEs are only measured over the last 100 days of trading (from 104th day to 204th day), so as to allow the algorithms to converge. The results show the superiority of the HKPF to the other four algorithms. In this experiment, the variances of NSE are so small that they are all set to be zero. Figure 5 plots the NSEs of the algorithms acquired after 100 runs. In this figure, the bottom red line with round shape marker is the HKPF prediction NSEs curve in each run, and the top line with black point marker is trivial prediction NSE curve. The trivial prediction is obtained by assuming that the price on the following day corresponds to the current price. We can see that the IEKPF shows better performance than the UPF, the EKPF, and the PF. The HKPF gives the best performance.

Table 2. Normalized Square Errors of Different PFs over 100 Runs

Option type	Algorithm(s)	NSE	
		mean	var
call	PF	0.032634	0.000
	EKPF	0.0084664	0.000
	UPF	0.0083221	0.000
	IEKPF	0.0044333	0.000
	HKPF	0.0040431	0.000
put	PF	0.026474	0.000
	EKPF	0.0080944	0.000
	UPF	0.0081215	0.000
	IEKPF	0.0043744	0.000
	HKPF	0.0040068	0.000

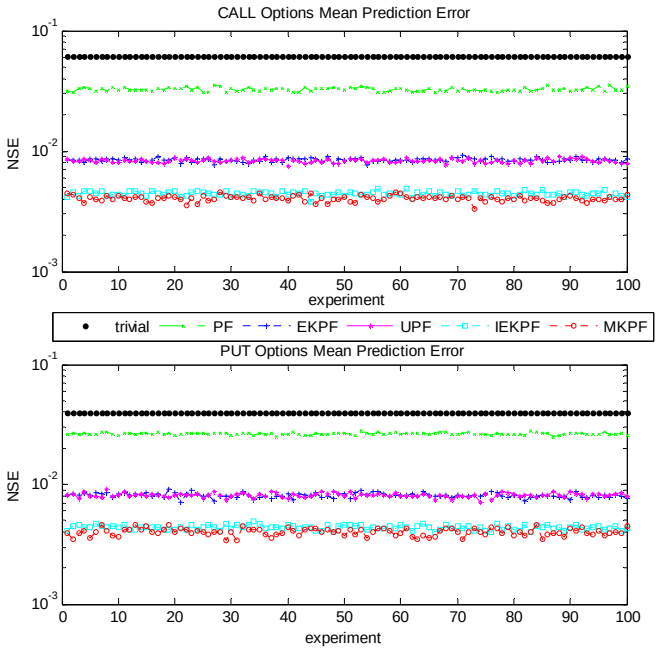

Figure 5. Plot of the prediction error over 100 runs.

V. IMPROVEMENT TO THE HKPF

As for computational complexity, the HKPF's time assumption is a bit higher, because it undergoes two particle-update steps. This is the major problem of this new particle filter. In section 4.5.1, the average time consumption of different particle filters are also counted, as shown in Table 3.

From Table 3, it is clear that the generic PF algorithm has the least time cost, about one second. The UPF has higher consumption than the EKPF and IEKPF. The HKPF has the highest run time consumption, about 16 seconds.

Table 3. Average time consumption of different particle filters obtained from experiment 1

Algorithm(s)	Time (seconds)
PF	0.91443
EKPF	8.1066
UPF	14.3398
IEKPF	9.1454
HKPF	15.9447

A. Partition- Conquer Strategy

In order to reduce the time consumption of HKPF, a partition-conquer strategy is adopted, that is, a part of the required particles (c percent) are drawn from the HKF proposal and the remaining part (1- c percent) are drawn from the transition prior. The particles are drawn from two distributions. Figure 6 gives a schematic description of the partition-conquer strategy. In this way, the time consumption of HKPF can be dropped, because the transition prior $p(x_k \mid x_{k-1})$ costs less time than the HKF generated proposal distribution.

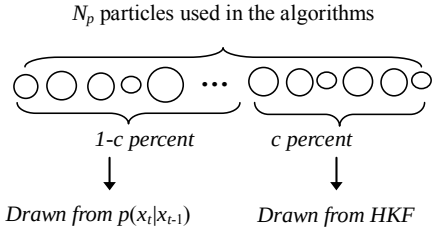

Figure 6. A schematic description of the partition-conquer strategy.

The improved HKPF algorithm is written as Algorithm 5.

Algorithm 5. The Improved HKPF algorithm

Step 1. Initialization,
The same as in Algorithm 4.
Step 2. FOR k=1,2,…
(1) FOR $i = 1,...,c * N_p$
The same as Step (a) -Step (b) in Algorithm 4;
c percent particles are drawn:
$\hat{x}_k^i \sim q(x_k^i \mid x_{0:k-1}^i, y_{1:k}) = \mathcal{N}(\bar{x}_k^i, \hat{P}_k^i)$;
Assign the particle a weight w_k^i according to Equ.(8);
END FOR
(2)FOR $i = c * N_p + 1,..., N_p$
Compute the state estimate using $\bar{x}_{k|k-1}^i = f(x_{k-1}^i)$;
(1- c) percent particles are drawn:
Draw $\hat{x}_k^i \sim p(x_{k|k-1}^i \mid x_{k-1}^i)$;
Assign the particle a weight;
(3) Normalize the weights,
The same as Step (2) in Algorithm 4;
(4) Resample,
The same as Step (3) in Algorithm 4;
(5) Output: calculate the required estimations using the particle set.
END FOR
Step 3. $k = k + 1$, go to Step 2 or end the algorithm.

B. Experimental Results

In order to evaluate the performance of the improved HKPF algorithm, the same models as in Section 4.5.1 are used in this experiment. The improved HKPF is only compared with the HKPF. All the parameters are the same as that in section 4.5.1. The parameter c is set to be 0.5 in this experiment. c can be set to be a random number in a closed interval [0,1]. When c =0, the improved HKPF algorithm turns to the generic particle filter, and when c =1.0, it becomes the HKPF algorithm.

From Table 4, it can be found that the improved HKPF needs less time consumption than the HKPF does, about half of the time of the HKPF. Meanwhile, the estimation accuracy does not lose much. By using the partition-conquer strategy, a trade-off can be drawn between accuracy and time consumption. For practical use, people can choose a proper c according to the real situations.

Table 4. Comparison between improved HKPF and the HKPF: estimation accuracy (MSE) and time consumption

Algorithm(s)	MSE	Time (seconds)
HKPF	0.01402	15.9524
Improved-HKPF	0.05529	8.3761

VI. CONCLUSION

In this paper, a new particle filtering algorithm is proposed, using the hybrid Kalman filter to generate the proposal distribution. Because the hybrid Kalman filter inherited the excellent attributes of the unscented Kalman filter and makes efficient use of the current observations, it could produce a closer approximation of the posterior distribution, which makes it a better choice for the proposal distribution. A simulation experiment and a real-world application experiment show that the HKPF can give better results than some other particle filters with different proposal distributions. In order to decrease the time consumption of the new algorithm, a new sampling strategy – partition-conquer strategy is given. This strategy can decreases the time consumption of the new particle filter algorithm with acceptable influence on the accuracy. Users can flexibly choose the parameter c to obtain desirable performance.

ACKNOWLEDGMENTS

This work was supported by Liaoning Education Ministry Foundation (NO.L2010043).

REFERENCES

[1] M. S. Arulampalam, M. Simon, N. Gordon and T. Clapp, "A tutorial on particle filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Trans. on Signal Process.*, vol.50, no.2, pp.174-188, February 2002.
[2] E. A. Wan and R. Merwe, "The Unscented Kalman Filter for Nonlinear Estimation", *Proceedings of the IEEE Symposium on Adaptive Systems for Signal Processing, Communication, and Control*, Lake Louise, Alberta, Canada, 1-4 October, 2000, pp. 153-158.
[3] S. Sadhu, S. Mondal, M. Srinivasan, and T.K. Ghoshal, "Sigma point Kalman filter for bearing only tracking", *Signal Processing*, vol.87, no.4, pp.3769-3777, December 2006.
[4] R. Merwe, A. Doucet, N. Freitas, et al., "The unscented particle filter", Technical report CUED/F-INFENG/TR380, Engineering Depart. Cambridge University, 2000, pp.1-46, unpublished.
[5] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state

estimation", *IEE Proc.-F*, vol. 140, no.2, pp.107-113, April 1993.

[6] J.S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems", *J. Amer. Statist. Assoc.*, vol.93, pp. 1032-1044, September 1998.

[7] D. Guo, X. Wang, and R. Chen, "New sequential Monte Carlo methods for nonlinear dynamic systems", *Statist. and Comput.*, vol.15, no.2, pp.35-147, April 2005.

[8] N. Bergman, "Recursive Bayesian estimation: Navigation and Tracking applications", Ph.D. Dissertation, Linköping University, Sweden, 1999.

[9] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models", *J. Comput. Graph. Statist.*, vol.15, no.1, pp.1-25, January 1996.

[10] J. Míguez, "Analysis of parallelizable resampling algorithms for particle filtering", *Signal Process.*, vol.87, no.12, pp.3155-3174, December 2007.

[11] J. D. Hol, T. B. Schon, and F. Gustafsson, "On Resampling Algorithms for Particle Filters", *Proc. of IEEE Workshop on Nonlinear Statistical Signal Processing*, Cambridge, UK, 13-15 September 2006, pp. 79 – 82.

[12] W. Gilks, and C. Berzuini, "Following a moving target: Monte Carlo inference for dynamic Bayesian models", *J. Roy. Stat. Soc. B*, vol.1, no.1, pp.127-146, January 2001.

[13] A. Doucet, M. Briers, and S. Sennecal, "Efficient block sampling strategies for sequential Monte Carlo methods", *J. of Comput. Graph. Stat.*, vol.15, no.3, pp.693-711, September 2006.

[14] N. Chopin, "A sequential particle filter method for static models", *Biometrika*, vol.89, no.3, pp.539-552, August 2002.

[15] B. Xu, Q. Chen, J. Zhu and Z. Wang, "Ant estimator with application to target tracking", *Signal Processing*, vol.90, no.5, pp.1496-1509, May 2010.

[16] A. Doucet, and N.J. Gordon, "Simulation-based optimal filter for manoeuvring target tracking", *Proc. of SPIE Conference on Signal and Data Processing of Small Targets*, Denver, CO, 20-22 July 1999, pp. 241-255.

[17] J.S. Liu and R. Chen, "Blind deconvolution via sequential imputations", *J. Amer. Statist. Assoc.*, vol.90, no.430, pp.567-576, 1995.

[18] A. Kong, J.S. Liu, and W.H. Wong, "Sequential imputations and Bayesian missing data problems", *J. Amer. Statist. Assoc.*, vol.89, no.425, pp.278-288, 1994.

[19] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking", *Internat. J. Comp. Vision*, vol.29, no.1, pp.5-28, 1998.

[20] A. Doucet, S.J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Statist. Comp.*, no.10, pp.197-208, 2000.

[21] Li Liang-qun, Ji Hong-bing, and Luo Jun-hui, "The iterated extended Kalman particle filter", *Proc. of International Symposium on Communication and Information Technology*, vol. 2, Beijing, China, 12-14 October 2005, pp. 1172~1175.

[22] K. Nishiyama, "Fast and effective generation of the proposal distribution for particle filter"s, *Signal Process.*, vol.85, no.12, pp.2412-2417, December 2005.

[23] WANG Fa-sheng and ZHAO Qing-jie, "A new particle filter for nonlinear filtering problems", *Chinese Journal of Computers*, vol.31, no.2, pp.346-352, February 2008.

[24] Y. Rui and Y. Chen, "Better Proposal Distributions: Object Tracking Using Unscented Particle Filter", *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 8-14 December 2001, pp.786-794.

[25] Dan Mikami, Kazuhiro Otsuka, Junji Yamato. "Memory-based Particle Filter for Tracking Objects with Large Variation in Pose and Appearance", *Lecture Notes in Computer Science 6313*, Springer-Verlag, pp. 215–228, 2010.

[26] Min Li, Wei Chen, Kaiqi Huang, Tieniu Tan. "Visual Tracking via Incremental Self-tuning Particle Filtering on the Affine Group", *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1315-1322. San Francisco, CA, June 2010.

[27] R. Kalsson, "Particle Filtering for Positioning and Tracking Applications", Ph. D Dissertation, Linköpings University, 2005.

[28] C. Kwok, D. Fox, and M. Meila, "Real-Time Particle Filters", *Proc. IEEE*, vol.92, no.3, pp.469- 484, March 2004.

[29] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots", *Artificial Intelligence*, vol.128, no.1, pp.99-141, May 2001.

[30] Y. Yu and Q. Cheng, "Particle filters for maneuvering target tracking problem", *Signal Process.*, vol.86, no.1, pp.195-203, January 2006.

[31] Branko Ristic, M. Sanjeev Arulampalam, "Tracking a manoeuvring target using angle-only measurements: algorithms and performance", *Signal Process.*, vol.83, no.6, pp.1223-1238, June 2003.

[32] M.H. Jaward, D. Bull, N. Canagarajah, "Sequential Monte Carlo methods for contour tracking of contaminant clouds", *Signal Process.*, vol.90, no.1, pp.249-260, January 2010.

[33] T.B. Schön, F. Gustafsson, and P. Nordlund, "Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models", *IEEE Trans. on Signal Process.*, vol. 53, no.7, pp.2279-2289, July 2005.

[34] F. Gustafsson, F. Gunnarsson, et al., "Particle Filters for Positioning, Navigation and Tracking", *IEEE Trans. on Signal Process.*, vol.50, no.2, pp.425-437, February 2002.

[35] Manuel A. Vázquez, Mónica F. Bugallo and Joaquín Míguez, "Sequential Monte Carlo methods for complexity-constrained MAP equalization of dispersive MIMO channels", *Signal Process.*, vol. 88, no.4, pp.1017-1034, April 2008.

[36] P. M. Djuric, "Monte Carlo methods for signal processing: recent advances", *Proc. of the 12th European Signal Processing Conference*, Vienna, Austria, 6-10 September 2004, pp. 853-860.

[37] A. Doucet and X. Wang, "Monte Carlo methods for signal processing: a review in the statistical signal processing context", *IEEE Signal Process. Magazine*, vol. 22, no.6 pp.152–170, November 2005.

[38] N. Freitas, M. Niranjan, A.H. Gee, and A. Doucet, "Sequential Monte Carlo methods to train neural work models", *Neural Computation*, vol. 12, no. 4, pp.955–993, April 2000.

[39] F.A.L. Aiube, T.K.N. Baidya and E.A.H. Tito, "Analysis of commodity prices with the particle filter", *Energy Economics*, vol. 30, no.2, pp.597–605, February 2008.

[40] A. Jasra, "Sequential Monte Carlo methods for option pricing", Technical Report, London, UK, Department of Mathematics, Imperial College London, 2008, unpublished.

[41] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.

[42] O. Cappé, S. J. Godsill, and E. Moulines, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo", *Proc. IEEE*, vol. 95, no. 5, pp.899-924, May 2007.

[43] C. Andrieu, A. Doucet, S.S. Singh, and V.B. Tadic, "Particle methods for change detection, system identification, and control", *Proc. IEEE*, vol. 92, no.3, pp.423-438, March 2004.
[44] G. Welch and G. Bishop, "An Introduction to the Kalman Filer", Technical Report, University of North Carolina at Chapel Hill, April 2004, unpublished.
[45] S. Särkkä, "On Unscented Kalman Filtering for State Estimation of Continuous-Time Nonlinear Systems", *IEEE Trans. on Automatic Control*, vol. 52, no.9, pp.1631-1641, September 2007.
[46] S.J. Julier and J.K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *Proc. IEEE*, vol. 92, no. 3, March 2004, pp.401-422.
[47] J.C. Hull, *Options, Futures, and Other Derivatives*, third edition, Prentice Hall, 1997.
[48] M. Niranjan, "Sequential tracking in pricing financial options using model based and neural network approaches", *Advances in Neural Information Processing System*, vol. 8, pp.960-966, 1996.

Fasheng Wang Master, Lecturer in Dalian Neusoft Institute of Information. His research interests mainly focus on nonlinear filtering algorithms and corresponding applications in intelligent information processing area.

Yuejin Lin Master, Associate Professor in Dalian Neusoft Institute of Information. Received his master's degree in Northeastern University in 1988. His research interests include software architecture and applied mathematics.

Tao Zhang He is currently a postgraduate student of Northeastern University. His research interests include software architecture.

Jingbo Liu He is currently a postgraduate student of Northeastern University. His research interests include software architecture.