A Biologically Inspired Improvement Strategy for Particle Filter: Ant Colony Optimization Assisted Particle Filter

Article *in* International Journal of Control Automation and Systems · June 2010

Impact Factor: 0.95 · DOI: 10.1007/s12555-010-0304-7

Junpei Zhong
Waseda University
10 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Yu-Fai Fung
The Hong Kong Polytechnic University
59 PUBLICATIONS **275** CITATIONS
SEE PROFILE

Mingjun Dai
Shenzhen University
28 PUBLICATIONS **103** CITATIONS
SEE PROFILE

International Journal of Control, Automation, and Systems (2010) 8(3):1-8 http://www.springer.com/12555
DOI 10.1007/s12555-008-0000000

A Biologically Inspired Improvement Strategy for Particle Filter: Ant Colony Optimization Assisted Particle Filter

Junpei Zhong, Yu-fai Fung, and Mingjun Dai

Abstract: Particle Filter (PF) is a sophisticated model estimation technique based on simulation. Due to the natural limitations of PF, two problems, namely particle impoverishment and sample size dependency, frequently occur during the particles updating stage and these problems will limit the accuracy of the estimation results. In order to alleviate these problems, Ant Colony Optimization is incorporated into the generic PF before the updating stage. After executing the Ant Colony optimization, impoverished particle samples will be re-positioned and closer to their locally highest likelihood distribution function. Our experimental results show that the proposed algorithm can realize better tracking performance when comparing to the generic PF, the Extended Kalman Filter and other enhanced versions of PF.

Keywords: Ant colony optimization, filtering theory, model estimation, particle filters.

1. INTRODUCTION

Particle Filter (PF), which is widely used for solving non-linear and non-Gaussian state estimation problems [1], is based on point mass particles that represent the probability densities. Therefore, in state estimation problems, PF is often used as an alternative to the Extended Kalman Filter (EKF) [2] or the Unscented Kalman Filter (UKF) [3]. With infinite samples, PF can approach the Bayesian optimal estimate [4], so it is more accurate than the EKF or UKF. Although this optimal situation is not available in real applications, the advantage of Particle Filters makes them useful in non-Gaussian and non-linear environments, such as in the control of Unmanned Aerial Vehicle (UAV) [5] and Autonomous Underwater Vehicle (AUV) [6]. However, particle impoverishment is inevitably induced due to the random particles prediction and re-sampling applied in generic PF [7]. After several iterations, if the generated particles are too far from the likelihood distribution, their particle weights will approach zero while only a few particles will be carrying a much higher weight, making these particles not efficient to produce accurate estimate results. Some algorithms employ different sampling strategies to reduce the impoverishment, such as Binary Search [8], Systematic Resampling [9] and Residual Resampling [10]. These algorithms achieve their targets by copying the important samples and discarding insignificant ones based on their weighting. However, in the mean while, the robustness of the filtering is lost, because the diversity of particles is reduced in a certain extent as discussed in [11].

Advocated by Doucet [12], a more refined strategy is to implement an optimal proposal distribution functions which minimize the variance of the importance weights,. This result has been proved in [13] that the proposal distribution $q(s_t \mid s^{t-1}, z^t) = p(s_t \mid s^{t-1}, z^t)$ minimizes the variance of the importance weights conditional on s^{t-1} and z^t. Other than the generic PF utilizes the transition model as the proposal distribution, extended Kalman Filter [14] and Unscented Kalman Filter [15] are employed as the model estimation methods to derive the optimal proposal distribution.

Ant Colony Optimization (ACO) is an iterative optimization method similar to the Genetic Algorithm. It has proved to be able to produce good optimization performance in many fields and this inspires us to apply ACO to optimize the particles distribution before the updating step and consequently minimizing the particle impoverishment problem. In this paper, we will first introduce the fundamental particle filter mechanism. The ACO improved Particle Filter (PF$_{\text{ACO}}$) together with an introduction of the ACO algorithm will be discussed in Section 3. In Section 4, the performance of the PF$_{\text{ACO}}$, the generic PF as well as other devised PF (i.e. extended Kalman Particle Filter and Unscented Particle Filter) will be compared and finally, conclusion is given in Section 5.

2. PARTICLE FILTERS

Firstly we derive the optimal Bayesian solution of the posterior distribution. Assuming that the system (x_k) and measurement (y_k) equations for Bayesian estima-tion

Manuscript received August 14, 2008; revised September 27, 2009; accepted December 5, 2009. Recommended by Editorial Board member Young Soo Suh under the direction of Editor Young Il Lee. This work was supported by the Department of Electrical Engineering of the Hong Kong Polytechnic University.

Junpei Zhong and Yu-fai Fung are with the Department of Electrical Engineering of the Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mails: {joni.zhong, eeyffung}@polyu.edu.hk).

Mingjun Dai is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: daimingjun1982@hotmail.com).

are governed by the following equations.

$$x_{k+1} = f_k(x_k, w_k), \tag{1}$$

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

where $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is a nonlinear function of the previous state x_k and process noise w_k, $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_z}$ is a nonlinear function of state x_k and measurement noise v_k. $\{w_k\}$ and $\{v_k\}$ are assumed to be independent noises. Assuming that the *pdf* of the initial state $p(x_0)$ is known, our problem is to compute the posterior density $p(x_k|y_{1:k})$ of each state x_k recursively.

A general expression of the prior probability distribution of the first order Markov system can be derived by Chapman-Kolmogorov equation with (1) and result is given in (3).

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{1:k-1}) dx, \tag{3}$$

where $y_{1:k-1}$ is defined as the history observation sequence with random variables.

With the observation y_k in each time step, the posterior probability distribution is calculated by

$$p(x_k \mid y_{1:k}) = \frac{p(y_k \mid x_k) p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})}. \tag{4}$$

In the above equation, the denominator $p(y_k \mid y_{1:k-1}) = \int p(y_k \mid x_k) p(x_k \mid y_{1:k-1}) dx_k$ is a constant value, which is available from the likelihood function and statistical characteristic of the observation noise.

The recurrence relations of (3) and (4) form the basis for the optimal Bayesian solution. However, this recursive propagation of the posterior density is just an optimal solution in theory, and it never can be obtained analytically because the integration in (3) is usually intractable. In the next section, the Particle Filters (also known as Sequential Monte Carlo methods) are introduced and its capability to approximate the optimal Bayesian solution will be discussed.

2.1. Generic particle filters

Particle filters are algorithms to perform recursive Bayesian estimation using Monte Carlo simulation and importance sampling, in which the posterior density is approximated by the relative density of particles in a neighborhood of state space:

$$p(x_{0:k} \mid z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i), \tag{5}$$

where the weighting value w_k^i is updated according to

$$w_k^i \propto w_{k-1}^i \frac{p(y_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, y_k)}. \tag{6}$$

It can be shown that as $N \to \infty$ the approximation (5) approaches the true posterior density $p(x_k \mid y_{1:k})$ [16].

The PF algorithm is illustrated in the following pseudo-codes.

Algorithm 1: The generic PF

$$\left[\left\{ x_k^i, w_k^i \right\}_{i=1}^{N} \right] = PF\left[\left\{ x_{k-1}^i, w_{k-1}^i \right\}_{i=1}^{N}, y_k \right]$$

```
1) Initialization: Generate particle samples $\left\{ x_0^i, w_0^i \right\}_{i=1}^{N}$
2) Prediction:
   For i =1:N
     • Predict $x_k^i \sim q\left( x_k^i \mid x_{k-1}^i, y_k \right)$
     • Assign the particle a weight (Eqn. 6)
   End For
3) Measurement update
     • Calculate total weight: $t = sum\left[ \left\{ w_k^i \right\}_{i=1}^{N} \right]$
     • For i=1:N
       ---Normalize: $w_k^i = t^{-1} w_k^i$
     • End For
4) Resampling
```

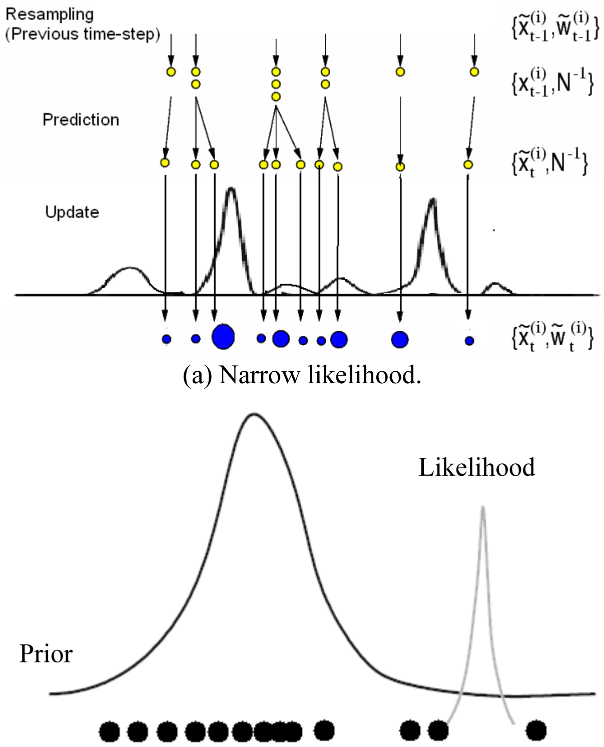
2.2. Particle impoverishment

During the iteration, the variance of the particle weights will probably increase very quickly [16], especially in applications with large variance noise, which is named Particle impoverishment.

Particle impoverishment [7] occurs when the likelihood distribution is so narrow that the overlapping region of likelihood and the prior distribution is too small [17]. As a result, particle weights of most particles become relatively smaller; the reason is that only a few particles are dispersed in the region of likelihood, as shown in Fig. 1(a). Another reason of causing the problem is that the likelihood lies in the tail of the prior distribution as shown in Fig. 1(b). If such situation occurs repeatedly, all but one sample will have negligible weights.

Let us discuss the situation when particle impoverishment occurs. If samples from the posterior and proposal distribution are taken, it can be seen that samples which are far away from each other make corresponding weighting values decrease during the iterations from (6), while just a small number of particles can maintain their weight values. This iteration runs over time, so the variance of the whole set of particles increases and a detailed proof can be found in [4]. In such cases, we want the samples taken from the proposal distribution to be comparable to those from the posterior distribution.

If the likelihood is so dense (mostly happens when the measurement is too accurate) then probably a few particles from the proposal distribution will lie within the region of the likelihood. Thus their weights are so small and they are negligible in the posterior density estimation as shown in (6). The efficiency of the whole particle set


Resampling
(Previous time-step)
$\{\tilde{x}_{t-1}^{(i)},\tilde{w}_{t-1}^{(i)}\}$
$\{x_{t-1}^{(i)},N^{-1}\}$
Prediction
$\{\tilde{x}_{t}^{(i)},N^{-1}\}$
Update
$\{\tilde{x}_{t}^{(i)},\tilde{w}_{t}^{(i)}\}$
Likelihood
Prior

(a) Narrow likelihood.

(b) Likelihood lies in the tail of prior distribution.

Fig. 1. Limitations in generic particle filter.

decreases. There is another situation that particle impoverishment happens. The new measurements (i.e., the likelihood) appear in the tail of the prior, because the prior model (mostly the transition model used in general cases [16]) becomes not accurate enough. So particles predicted from the prior density will distribute far from the likelihood. Therefore most of the particles have small weights.

Resampling methods, such as Sequential Importance Sampling (SIS) [8,18], is employed to solve these problems. Besides of SIS, some improved re-sampling methods have been developed [9,10,19-21]. However, since these sampling techniques are merely producing a suboptimal solution, in some applications, only one sample is remained and copied to the whole particle set after several steps of resampling. This is called losing of particle diversity [11].

In addition to particle impoverishment, sample size dependency is another problem. If the size of the particle set is small, potentially there may not have sufficient particles that can approach the true state, which will limit the filter from converging. In generic particle filters, the only way to solve this problem is to increase the number of particle, but that will abbreviate the computational requirements.

2.3. Devised version of particle filters

In order to eliminate the particle impoverishment and dependency problems, researchers focused on the following methods.

- Improving resampling methods and techniques such as Binary Search [8], Systematic Resampling [9] and Residual Resampling [10] have been proposed. However, these methods are not ideal because the copied samples are no longer statistically independent after resampling therefore the previous convergence result will be lost. It is called losing sampling diversity [11].
- Optimizing the proposal distribution with modified PF algorithms such as Extended Kalman Particle Filter [22], or Unscented Particle Filter [23]. These methods are discussed in the next section.

2.3.1 Extended Kalman particle filter

Extended Kalman Particle Filter tried to construct the proposal distribution function by incorporating the current observation with the optimal Gaussian approximation of the state obtained from the Extended Kalman Filter (EKF) [24,25]. It relies on the first order Taylor series expansions of the likelihood and transition prior, as well as a Gaussian assumption on all random variables in question. In this framework, the EKF approximates the optimal MMSE estimator of the system state by calculating the conditional mean of the state, given all observations. In other words, the EKF implements the following recursive approximation (7) to the true posterior filtering density,

$$p(s_t \mid z^t) \approx p_N(s_t \mid z^t) = N(\overline{x}_t, \hat{P}_t). \tag{7}$$

However, although EKF possibly builds a better proposal distribution by making a Gaussian assumption on the form of the posteriors as well as introducing inaccuracies due to linearization. In fact, the current observation at time *t* in the proposal distribution generated by EKF will not be Gaussian. This can be easily shown by a Bayes' rule expansion of the proposal distribution.

2.3.2 Unscented particle filter

Similar to Extended Kalman PF, the Unscented PF uses Unscented Kalman Filter (UKF) as a distribution generation within the PF framework. UKF is regarded as having a bigger support overlap with the true posterior distribution than the overlap achieved by the EKF estimates. This is in part related to the fact that the UKF calculates the posterior covariance accurately to the third order by a deterministic sampling technique known as the unscented transform to recursive minimum mean-square-error (RMMSE) estimation [26] to pick a minimal set of sample points (called sigma points) around the mean, while the EKF relies on a first order biased approximation. The sigma points in UKF are then propagated through the non-linear functions, from which the mean and covariance of the estimation are then recovered up to the second order of the Taylor expansion [23]. Consequently, the result is that Unscented Particle Filter has a proposal distribution which can capture the true mean and covariance more accurately to the optimal one than extended Kalman Particle Filter does. In addition, this technique removes the requirement to explicitly calculate the Jacobians, which for complex functions could be a difficult task by itself.

3. ANT COLONY OPTIMIZATION ASSISTED PARTICLE FILTER

Ant Colony Optimization (ACO) algorithm is essentially a biologically inspired system based on agents that simulate the natural behavior of ants [27]. The utilization of such a system as a new metaheuristic method was first proved to be successful in solving the traveling salesman problem (TSP) [28]. Furthermore, this new metaheuristic method has been shown to be robust and versatile and has been applied successfully in solving a range of combinatorial optimization problems [29], including the quadratic assignment problem [30,31],vehicle routing problem [32], fault detection [33] and rough data reduction [34].

The ACO is derived from the natural optimization ability demonstrated by ants. When finding food, they tend to take the best route (or path) between their nest and some external landmark because their particular pheromone trail becomes higher if more ants choose the same trail. The closer the landmark is to the nest, the higher the number of round-trips can be made by each ant. The higher the concentration of pheromones, more ants will choose this route over others that might be available. This iterative process achieves optimal trails between the endpoints. So the ACO algorithm uses the mathematical formulas to simulate this natural optimization process.

The ACO algorithm is based on the following mechanisms.

1) Given a problem, a candidate solution is associated with a path or a vertex that corresponds to a moving ant in reality. In our application, for example, each ant is associated with a particle being generated; the particles (ants) move along a path in the state space, a particle with new state is regarded as a vertex. So the new particle set forms a candidate solution;

2) when an ant follows a path, the amount of pheromone along that path is increased with a certain level;

3) when an ant has to choose between two or more paths, the path with a larger amount of pheromone and shorter distance has a greater probability of being chosen by it;

4) the amount of pheromone associated with a trail is under evaporation with time. This characteristic makes the path towards the optimized state will always correspond to the maximum pheromone.

The direction of movement for an ant is based upon the following probability function defined in (8). While an ant has not yet completed the movement, the following equation is used to identify the next path to be taken until they converge to an optimal solution.

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]_{\alpha}[\eta_{ij}(t)]_{\beta}}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]_{\alpha}[\eta_{is}(t)]_{\beta}}, \tag{8}$$

where η_{ij} is a heuristic value that is available as a priori. This probability distribution is biased by the parameters α and β that determine the relative influence of the pheromone trails and the heuristic information, respectively. These two parameter definitions vary in different problems. For example, in TSP, if $\alpha = 0,$ the closest cities are more likely to be selected, which runs like a classic stochastic greedy algorithm; if $\beta = 0,$ only pheromone update is dominated in the probability function, which probably leads to rather poor results and, in particular, for values of $\alpha > 1,$ it results in the rapid emergence of a stagnation situation [35], that is, a situation in which all ants will concentrate along the same path and construct a suboptimal solution [36].

3.1. Ant colony optimization in PF

To optimize the re-sampling step of the generic particle filter, we incorporate ACO into the PF and utilize the ACO before the updating step. As mentioned before, a single ant will replace the randomly-generated particle in the Sequential Monte Carlo concept and they will converge to the local peak of the optimal proposal distribution function in the following optimizing step.

The $\tau(t)$, as shown in (9), is affected by every movement of the particle by the following equation:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \\ \qquad\qquad j \in \text{set of particles lie in the movement path} \\ \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t)\; j \in \text{set of other particles,} \end{cases} \tag{9}$$

where $0 < \rho \le 1$ is the pheromone evaporation rate, $\Delta\tau$ is a constant enhanced value if particle j is located between the starting particle and the end point. The pheromone trail definition implies that with more ants going through the same path, it will enhance the amount of pheromone in this trail, while the pheromone of paths for which fewer ants have traversed will decline proportionally with time. The longer time it takes for an ant to travel along the path and back again, the pheromones will evaporate more. By comparison, the pheromone value remains high in the shortest path because most of the ants finally choose this path. For example, let us define *ant i* to be the one intending to move during the iteration and *ant j* lies in the path between *ant i* and the potential destination, as shown in (9), the pheromone τ_{ij} enhancement will be executed during this iteration based on the target of the stochastic move. In addition to the enhancement, we also know that all pheromone values τ associating with *ant i* (τ_{i*}), including τ_{ij} are reduced when the pheromone is evaporated at each iteration.

The heuristic function (β) as stated in the above section, is now defined as the reciprocal of the distance between two particles (end points):

$$\eta_{ij}(t) = \frac{1}{d_{ij}}. \tag{10}$$

Finally, we need to evaluate the fitness function of each particle. It is proportional to the weight function, denoting the rate of a particle approaching the true state. The optimization step runs iteratively based on a probability function obtained from (11). It represents the probability of a particle i selecting particle j among N-1 particles as the moving direction.

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]_{\alpha}[\eta_{ij}(t)]_{\beta}}{\sum\limits_{s \in \text{all particles}} [\tau_{is}(t)]_{\alpha}[\eta_{is}(t)]_{\beta}}, \tag{11}$$

where α and β are parameters with the same definition as defined in the original ACO. The parameter α (τ_{ij}) has an initial value equal to the particle weight, as stated in

$$\tau_{ij}(0) = w_j. \tag{12}$$

During iterations, τ_{ij} is either enhanced or decreased by (6) depending on whether the particle j lies in the movement path and the parameter β is determined by (10).

The ACO algorithm converges when P_{ij} approaches 1 [37], so it implies that the particle i definitely re-locates to a closer proximity of particle j. When this convergence holds during each iteration, most particles converge to this particle j, which is represented as the neighborhood of higher likelihood (higher weights) based on (11). In this process, two parameters determine the relative influence. If $\alpha = 0$, all particles choose to remain in their original positions so the algorithm degenerates to a generic PF; if $\beta = 0$, particles tend to move towards neighborhood around higher likelihood, so the distribution is approaching $p(y_k \mid x_k)$.

Furthermore, the ACO terminates until all the moved particles' positions converge to the high likelihood region (the general or local optimal solution) within a certain threshold, defined in (13).

$$\mathit{Threshold}^{j} = (1 - w^{j}) \times (\mathit{randn}) \times \text{constant value}, \tag{13}$$

where w is the weight of the target particle, $randn$ is a normal distributed random number.

A pseudo-program describing the algorithm is given below.

Algorithm 2: The ACO

```
Algorithm 2: The ACO_PF Algorithm
    $[\{x_k^i, w_k^i\}_{i=1}^N] = ACOPF[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, y_k]$
1) The initialization and prediction steps [(1~2) in Algorithm 1]
2) ACO assisted PF
   While the distance between particles and their targets are not within a certain threshold (Eq. 13) and the iteration times does not exceed the maximum value
     • Choose particle $i$ whose distance is within the threshold
     • Select the moving target based on the probability (Eq. 11)
     • Move towards the target with a constant velocity
     • Update the parameters of the ACO (e.g. $\eta$, $\tau$), and particle weights
3) End While
4) Update Step [(3~4) in Algorithm 1]
5) Resampling
```

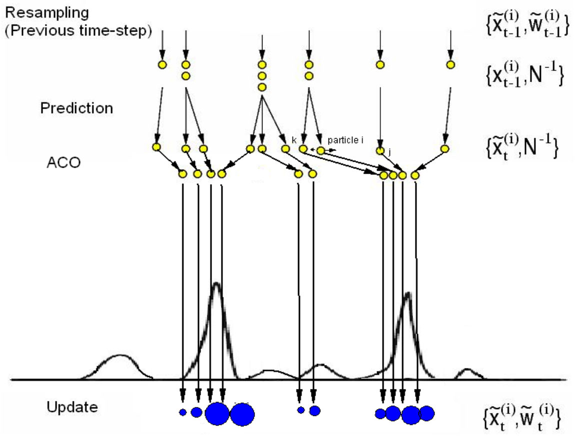

Resampling
(Previous time-step)
$\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}$
$\{x_{t-1}^{(i)}, N^{-1}\}$
Prediction
k
particle i
j
$\{\tilde{x}_t^{(i)}, N^{-1}\}$
ACO
Update
$\{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}$

Fig. 2. The PF$_{\text{ACO}}$ Demonstration.

Optimized by the ACO, the particle impoverishment problem is alleviated. The particle samples tend to be around high likelihood regions. As a result, most of the particles which are scattered far away from the true state will converge to states that represent high probability as shown in Fig. 2. Therefore, when configured with suitable parameters (ρ, τ_0, η_0, $\Delta\tau$, constant value of threshold, etc.), ACO is able to balance between the diversity and the impoverishment of particle filters.

In addition, since the impoverished particles are concentrated in the higher likelihood region, these particles will produce better contribution in the Monte Carlo simulation, so that sample size dependency problem of the generic particle filter is also minimized.

Because particle j as a moving target has a higher weight and shorter distance than the other particles, p_{ij} (denoted by the length of arrow) is larger than other probabilities. Therefore particle i moves towards particle j. Comparing with Fig 1(a), particles are closer to the local maximum *pdf*, so that the particle impoverishment is avoided.

4. CASE STUDY

A nonlinear single variable economic model [23], defined by (14) and (15), is employed to test the performances of various PFs, including Extended Kalman Filter [38], Unscented Kalman Filter [22,39,40], generic PF (PF with sequential importance re-sampling) [16], extended Kalman PF [16], Unscented PF [23] and PF$_{\text{ACO}}$. To fully compare the results, the generic PF, extended Kalman PF and Unscented PF are all implemented with and without MCMC [16].

$$x(t+1) = 1 + \sin(4 \times 10^{-2}\pi t) + 0.5x(t) + w(t), \tag{14}$$

$$y(t) = \begin{cases} \dfrac{x(t)^2}{5} + u(t), & \text{t} \le 30 \\[2ex] -2 + \dfrac{x(t)}{2} + u(t), & \text{t} > 30, \end{cases} \tag{15}$$

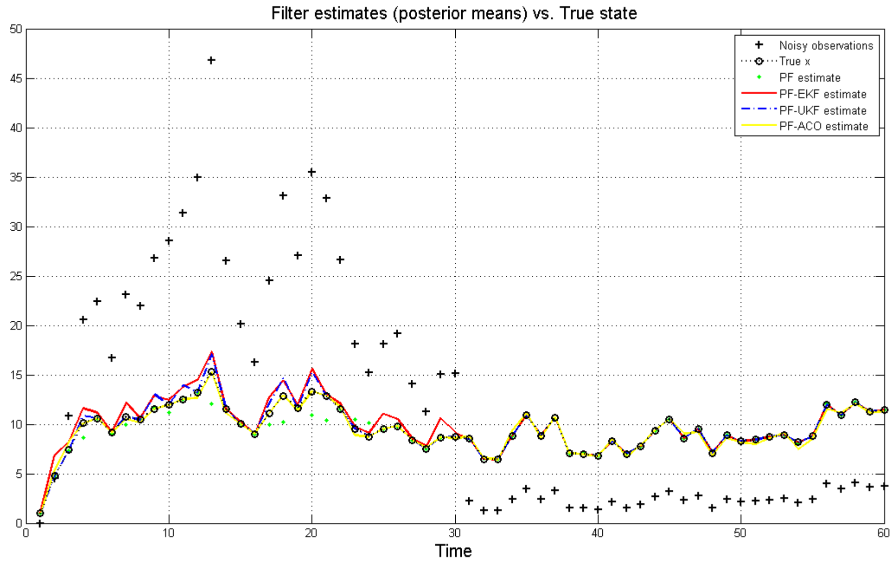


Fig. 3 The diagram of different PF tracking result.

where $w(t)$ stands for the zero-mean white noise, and $u(t)$ stands for noise with Gamma distribution [40]. The variance of $w(t)$ is 1×10^{-5}, and the two parameters of the Gamma distribution, k and θ, equal to 7 and 2, respectively.

Table 1. RMS value of error.

Filters	RMS Error	RMSE Percentage (EKF=100%)
EKF	0.98087	100
UKF	0.68237	69.57
Generic PF	0.77918	79.44
PF+MCMC	0.79492	81.04
PF+EKF	0.95391	97.25
PF+UKF	0.3792	38.66
PF+EKF+MCMC	0.95354	97.21
PF+UKF+MCMC	0.39387	40.16
PF+ACO	0.28153	28.70

Table 2. Variance of RMS error.

Filters	Variance
EKF	0.059334
UKF	0.029767
Generic PF	0.054233
PF+MCMC	0.041409
PF+EKF	0.044244
PF+UKF	0.021977
PF+EKF+MCMC	0.049126
PF+UKF+MCMC	0.01669
PF+ACO	0.001619

Table 3. Execution time of filters.

Filters	Time (Sec)
EKF	0.53321
UKF	0.92803
Generic PF	0.89604
PF+MCMC	1.958
PF+EKF	5.935
PF+UKF	11.6809
PF+EKF+MCMC	10.0614
PF+UKF+MCMC	22.8866
PF_{ACO}	3.1854

From $t = 1$ to 60 in a single test run, given the noise measurement, the state sequence x_t was estimated by all filtering methods. In order to minimize the effect of randomness, all experiments included 30 runs. For all PF algorithms, the number of particles used was 200. Tables 1 and 2 show the mean and variance of the Root Mean Square Error (RMSE) obtained from different PF algorithms.

From Tables 1and 2, we can conclude that our PF_{ACO} can produce the best result with the smallest RMS error. In addition, the variance of our method is also the smallest comparing to other methods, implying that the PF_{ACO} can give more stable performance according to [41].

The average execution time in each run is measured and given in Table 3. As listed in Table 3, the PF_{ACO} takes a longer computational time when comparing to the Kalman Filters and generic PF, which generally lead to larger estimation error. But comparing to similar PFs with improved proposal distribution including the extended Kalman Particle Filter and Unscented Kalman Particle Filter (both with and without MCMC), the PF_{ACO} takes shorter execution time since we pre-define a maximum iteration times.

Fig. 3 represented the estimation results from all filters of one run, which shows that the PF_{ACO} can track the estimation accurately throughout the whole experiment. The PF_{ACO} performs better than other PF in the single variable estimation test. Besides, the extended Kalman Particle Filter performs worse than generic PF, especially from time interval 1 to 30, it may be caused by the transition function with white noise, for which the EKF probability cannot track very well, even compared to the generic PF. After the time 30, with another measurement function, almost all PF perform quite well.

5. CONCLUSIONS

In a noisy tracking problem, generic Particle filter, as a method derived from sequential Monte Carlo Method, is easy to cause particle impoverishment and sampling dependency. The major cause of these problems is due to the randomly generated particles that usually cannot fully cover the distribution density function. In this paper, we proposed a new algorithm that utilizes the Ant Colony Optimization to re-arrange the particles before the updating step of particle filters. After the ACO optimization, particles will approach the higher likelihood density function and therefore it will minimize the problems of particle impoverishment and sample dependency. Our experimental results illustrated that the ACO assisted PF produces better tracking performance in solving the single variable nonlinear estimation problem. In addition, its computation time is shorter than other improved versions of PF.

REFERENCES

[1] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." *IEE Proceedings F Rader and Signal Processing*, vol. 140, no. 2, pp. 107-113, 1993.

[2] S. F. Schmidt, "The Kalman filter - its recognition and development for aerospace applications," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 4-7, 1981.

[3] E. A. Wan and R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation." *Proc. of IEEE Symposium of Adaptive Systems for Signal Processing, Communication, and Control*, pp. 153-158, 2000.

[4] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002.

[5] A. Ryan, et al., "Decentralized control of unmanned aerial vehicle collaborative sensing missions." *American Control Conference*, pp. 4672-4677, 2007.

[6] I. Nygren and M. Jansson, "Terrain navigation for underwater vehicles using the correlator method," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp. 906-915, 2004.

[7] M. G. S. Bruno and A. Pavlov, "Improved particle filters for ballistic target tracking." *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 705-708, 2004.

[8] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, 2000.

[9] M. Isard and A. Blake, "Condensation: conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[10] J. U. Cho, S. H. Jin, X. D. Pham, J. W. Jeon, J. E. Byun, and H. Kang, "A real-time object tracking system using a particle filter," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2822-2827, 2006.

[11] C. Stachniss, G. Grisetti, and W. Burgard, "Recovering particle diversity in a rao-blackwellized particle filter for slam after actively closing loops." *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 18-22, 2005.

[12] J. Carpenter, P. Clifford, and P. Fearnhead, "Building robust simulation-based filters for evolving data sets." Department of Statistics, Oxford University, 1999.

[13] A. Doucet and N. Gordon, "Simulation-based optimal filter for maneuvering target tracking," *SPIE Signal and Data Processing of Small Tragets*, vol. 3809, no. 1, pp. 241-255, 1999.

[14] L. Liang-Qun, J. Hong-Bing, and L. Jun-Hui, "The iterated extended Kalman particle filter," *Proc. of IEEE International Symposium on Communications and Information Technology*. vol. 2, pp. 1213-1216, 2005.

[15] R. Van der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," *Advances in Neural Information Processing Systems*, 2000.

[16] M. S. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[17] W. Zang, Z. G. Shi, S. C. Du, and K. S. Chen, "Novel roughening method for reentry vehicle tracking using particle filter," *Journal of Electromagnetic Waves and Applications*, vol. 21, no. 14, pp. 1969-1981, 2007.

[18] S. N. MacEachern, M. Clyde, and J. S. Liu, "Sequential importance sampling for nonparametric bayes models: the next generation," *The Canadian Journal of Statistics*, vol. 27, no. 2, pp. 251-267, 1999.

[19] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering." *Proc. of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64-69, 2009.

[20] R. Luo and B. Hong, "coevolution based adaptive monte carlo localization (CEAMCL)," *International Journal of Advanced Robotic Systems*, vol. 1, no. 3, pp. 183-190, 2004.

[21] P. Torma and C. Szepesvari, "Enhancing particle filters using local likelihood sampling." *Lecture Notes in Computer Science*. pp. 16-27, 2009.

[22] L. Li, H. Ji, and J. Luo, "The iterated extended Kalman particle filter." *Proc. of IEEE International Symposium on Communications and Information Technology*, vol. 2, pp. 1213-1216, 2006.

[23] R. Merwe and A. Doucet, "The unscented particle filter." vol. CUED/F-INFENG/TR 380, Engineering Department, Cambridge University, 2000.

[24] C. Andrieu, J. de Freitas, and A. Doucet,

"Sequential MCMC for bayesian model selection." *IEEE Higher Order Statistics Workshop*. pp. 130-134, 1999.

[25] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges." *Proc. of International Joint Conference on Artificial Intelligence*. pp. 1151-1156, 2003.

[26] A. Doucet, "Monte carlo methods for bayesian estimation of hidden markov models: applications to radiation signals," University Paris-Sud, 1997.

[27] O. Cordon, F. Herrera, and T. Stutzle, "A review on the ant colony optimization metaheuristic: basis, models and new trends," *Mathware and Soft Computing*, vol. 9, pp. 141-175, 2002.

[28] A. Colorni, M. Dorigo, and Maniezzo, "Distributed optimization by ant colonies." *Proc. of the 1st European Conference on Artificial Life*, pp. 134-142, 1991.

[29] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization." *IEEE Computational Intelligence Magazine*, vol. 1 no. 4, pp. 28-39, 2006.

[30] V. Maniezzo and A. Colorni, "The ant system applied to the quadratic assignment problem," *IEEE Trans. on Knowledge and Data Engineering*, vol. 11, no. 5, pp. 769-778, 1999.

[31] T. Stutzle and H. Hoos, "The MAX-MIN ANT system and local search for combinatorial optimization problems," FG Intellektik, TH Darmstadt, vol. AIDA-97-4, 1997.

[32] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," *Annals of Operations Research*, vol. 89, pp. 319-328, 1999.

[33] C. S. Chang, L. Tian, and F. S. Wen, "A new approach to fault section estimation in power systems using Ant system," *Electric Power Systems Research*, vol. 49, no. 1, pp. 63-70, 2009.

[34] R. Jensen and Q. Shen, "Fuzzy-rough data reduction with ant colony optimization," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 5-20, 2005.

[35] Marco Dorigo and Thomas Stutzle, *Ant Colony Optimization*, The MIT Press, Cambridge, 2004.

[36] M. Dorigo and T. Stutzle, "The ant colony optimization metaheuristic: algorithms, applications, and advances," *Handbook of Metaheuristics*, vol. 57, pp. 250-285, 2006.

[37] T. Stutzle and M. Dorigo, "A short convergence proof for a class of ant colony optimization algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 4, pp. 358-365, 2002.

[38] G. Welch and G. Bishop, "An introduction to the Kalman filte," Univ. North Carolina at Chapel Hill, vol. 95-041, 2001.

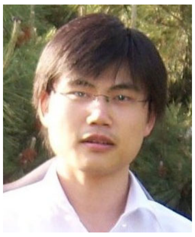
[39] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation." *Proc. of the IEEE* vol. 92, no. 3, pp. 401-422, 2004.

[40] E. Lukacs, "A characterization of the gamma distribution," *The Annals of Mathematical Statistics*, vol. 26, no. 2, pp. 319-324, 1955.

[41] R. Roll, "A mean/variance analysis of tracking error," *The Journal of Portfolio Management*, vol. 18, no. 4, pp. 13-22, 1992.

Junpei Zhong received his B.Eng. degree in Automation from South China University of Technology in 2006. He is currently pursuing M.Phil degree in Department of Electrical Engineering, the Hong Kong Polytechnic University. His research interests include Robotics SLAM, Artificial Intelligence, and Biologically Inspired Robotics.

Yu-fai Fung received his BSc(Eng) degree in 1986 and his Ph.D. in 1991, both from University College, London. He is currently a Senior Teaching Fellow in the Department of Electrical Engineering of The Hong Kong Polytechnic University. He conducts research and teaching in areas of computer engineering and robotics.



Mingjun Dai received his B.Eng, M.Phil, degrees from the Harbin Institute of Technology, in 2004 and 2006, respectively, all in Control Theory and Engineering. In 2008, he joined the Department of Electrical Engineering, the Hong Kong Polytechnic University as a research assistant. In the same year, he joined the Electronic Engineering department at City University of Hong Kong, and is currently a PhD candidate. His research interests include control theory, multiuser information theory, and control and optimization of wireless networks.