A Theoretical Proof and Case Study of Ant Colony Optimization Improved Particle Filter Algorithm

J.P Zhong
Department of Computer Science, University of Hamburg
Y.F. Fung
Department of Electrical Engineering, The Hong Kong Polytechnic University

Abstract: Particle filters, as a kind of non-linear/non-Gaussian estimation method, are suffering from two problems in large dimensional cases, namely particle impoverishment and sample size dependency. Previous papers from the authors have proposed a novel particle filtering algorithm that incorporates Ant Colony Optimization (PF_{ACO}), to alleviate these problems. In this paper, we will provide a theoretical foundation of this new algorithm; two theorems are introduced to validate that the PF_{ACO} introduces smaller Kullback-Leibler Divergence (K-L divergence) between the proposal distribution and the optimal one comparing to those produced by the generic PF. In addition, with the same threshold level, the PF_{ACO} has a higher probability than the generic PF to achieve a certain K-L Divergence. The mobile robot localization problem is applied to examine the performance between various PF schemes.

1. Introduction

Particle Filter (PF) is based on point mass particles that represent the probability densities of the solution space and it is widely used for solving non-linear and non-Gaussian state estimation problems [1]. As an alternative method of Kalman Filter [2, 3], it is widely used in applications under non-linear and non-Gaussian environments, such as control of an Unmanned Aerial Vehicle (UAV)[4] and Autonomous Underwater Vehicle (AUV) [5]. The advantage of PF is that it can approach any probability distribution estimate [6] with infinite samples. Although this optimal estimation is not available in real applications, it can still produce better

results in the non-linear/non-Gaussian environment when comparing to traditional methods. However, particle impoverishment is inevitably induced due to the random particles prediction and re-sampling applied in generic PF [7], especially with large number of state dimensions: after a number of iterations, if the generated particles are too far from the likelihood distribution, their particle weights will approach zero with only a few particles carrying significant weights, making other particles not efficient to produce accurate estimation results.

Therefore, there are other enhanced PF algorithms that employ different sampling strategies to minimize the impoverishment effect and these strategies include Binary Search [8], Systematic Resampling [9] and Residual Resampling [10]. The basic idea of these algorithms is to achieve their targets by copying the important samples and discarding insignificant ones by different calculation and selection methods mainly based on their weights. The idea of these methods has been proved in [11] that a better proposal distribution $q(x_k \mid x_{1:k-1}, y_{1:k}) = p(x_k \mid x_{1:k-1}, y_{1:k})$ can be achieved and minimize the variance of the importance weights conditional on state history $x_{1:k-1}$ and measurement $y_{1:k}$. However, at the mean time, the robustness of the filtering is lost, because the diversity of particles is reduced by a certain extent as discussed in[12].

In [13, 14], the authors introduced a metaheuristic method, the Ant Colony Optimization (ACO), to PF to optimize the particle distribution, however, theoretical evaluations were excluded therefore in this paper, a more comprehensive study is presented. In Section 2, we will first introduce the basic features of particle filter as well as the PF_{ACO} mechanism. In Section 3, two theorems concerning the Kullback–Leibler Divergence (K-L Divergence) between the proposal distribution and the optimal solution are adopted to prove that the PF_{ACO} would bring better performance than the generic PF. After that, in Section 4, the performance of the PF_{ACO}, the generic PF as well as other devised PF (i.e. extended Kalman Particle Filter and Unscented Particle Filter) will be compared based on the robot localization

problem. Finally, conclusion of the paper is given in Section 5.

2. Particle Filters

In this section we are going to derive an optimal Bayesian solution to the posterior distribution. Assume that the system (x_k) and measurement (y_k) equations for the Bayesian estimation 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 (Probability Density Function) 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 according to what we observe y_k recursively.

A general expression of the prior probability distribution of the first order Markov system can be derived by Chapman-Kolmogorov equation from Equation 1 and the result is given in Equation 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 historical observation sequence from time-step 1 to $k-1$ 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 can be derived from the likelihood function and statistical characteristic of the observation noise.

The recurrence relation of Equations 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 can never be obtained analytically because the integration in Equation 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 (weights) of particles observed in the state space. The posterior can be approximated by the weighted summation of every particle (Equation 5).

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

where the weighting value of particle i at time-step k, w_k^i is updated according to Equation 6.

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

It can be shown that as $N \to \infty$ the approximation (Equation 5) approaches the true posterior density $p(x_k \mid y_{1:k})$ [15].

Also the resampling method is needed after iterations to solve two problems: namely

particle impoverishment and sample size dependency, which will probably occur during the particle updating stage. The most popular resampling method is SIS [1].

The PF algorithm is illustrated by the following pseudo-codes, employing N particles.

Algorithm 1: The generic PF

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

(1) Initialization: Generate particle samples $\{x_0^i, w_0^i\}_{i=1}^N$

(2) Prediction:

For i =1:N

- Predict $x_k^i \sim q(x_k^i \mid x_{k-1}^i, y_k)$
- Assign the particle a weight (Eqn. 6)

End For

(3) Measurement update

- Calculate total weight: $t = sum[\{w_k^i\}_{i=1}^N]$
- For i=1:N

  ---Normalize: $w_k^i = t^{-1} w_k^i$
- End For

(4) Resampling

However, in problems that involve a large number of dimensions, such as the multi-robot SLAM problem, a huge number of particles have to be used in order to maintain an accurate estimation, the generic resampling method is not sufficient to avoid the impoverishment and size dependence problems. Consequently these problems will become more severe after iterations, rendering a large portion of the particles negligible and reducing the accuracy of the estimation results. In general, there are two families of improved resampling methods or improvement methods:

- Improving resampling methods and techniques such as Binary Search[8], Systematic Resampling[9] and Residual Resampling [10] have been proposed. However, as we mentioned before these methods are not ideal because the copied samples are no longer statistically independent after resampling therefore the convergence result obtained from previous step will be lost. It is called losing sampling diversity [12].
- Optimizing the proposal distribution with modified estimation algorithms such as Extended Kalman Particle Filter[16], or Unscented Particle Filter[17]. They apply extended Kalman Filter or Unscented Kalman Filter to construct the proposal distribution in order to minimize the variance of the importance weights. Though they approach the optimal proposal distribution better than the generic one, theoretically they are fixed models to determine the sub-optimal solution.

2.2 Ant Colony Optimization (ACO) Improved Particle Filter

The ACO is derived from the natural optimization ability demonstrated by ants. When searching for food, ants tend to take the best route (or path) between their nest and some external landmarks 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. With higher 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 applies 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 which is being traversed by an ant in practice. In our scheme, 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 a 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 higher level of pheromone and shorter distance will have a higher probability being selected;

4) the amount of pheromone associated with a trail is evaporating with time. This characteristic makes the path with most ants traversed will always carry the maximum level of pheromone.

The direction of movement for an ant is based upon the following probability function defined in Equation 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.

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

where the parameter α stands for the amount of pheromone, β usually represents the heuristic value, and their definitions are depended on the application. The definitions of these two parameters in our application will be introduced in next section. This probability distribution is biased by these two parameters α and β that determine the relative influence of the pheromone trails and the heuristic information, respectively..

2.3 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 [13, 14]. An ant will replace one particle and they will move based on the route that leads towards the local peak of the optimal proposal distribution function, which is depicted in the following.

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

$$\begin{cases} \tau_{i*}(t+1) = (1-\rho)\tau_{i*}(t) + \Delta\tau_{i*}(t) & \text{set of particles lie in the movement path} \\ \tau_{i*}(t+1) = (1-\rho)\tau_{i*}(t) & \text{set of other particles} \end{cases} \quad (9)$$

where $0 < \rho \leq 1$ 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 deplete with time. The longer 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 majority of the ants will eventually choose this path. For example, let us define *ant i* to be the one intending to move during the iteration and *ants m,n,…,z* lie in the path between *ant i* and the potential destination *ant j*, as shown in Equation 9, the pheromone τ_{i*} 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) is defined as the reciprocal of the distance between two particles (end points):

$$\eta_{i*}(t) = \frac{1}{d_{i*}} \quad (10)$$

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 Equation 11. It represents the probability of a particle *i* selecting particle *j* among *N-1* particles as the moving direction.

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

α and β are parameters with the same definition as in the original ACO. The initial value of parameter α (τ_{i*}) equals to the particle weight, as stated in Equation 12.

$$\tau_{i*}(0) = w_{*} \quad (12)$$

During iterations, τ_{i*} either increases or decreases by Equation 6, depending on whether the particle j lies in the movement path; the parameter β is determined by Equation 10.

The ACO algorithm converges when P_{ij} approaches 1 [18], so it implies that the particle i will definitely re-locate at a closer proximity of particle j. When this convergence is satisfied during each iteration then most particles will converge to this particle j, which is considered as the neighborhood of a higher likelihood (higher weights) based on Equation 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 $p(x_k \mid y_k)$.

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 as defined in Equation 13.

$$Threshold^{j} = (1 - w^{j}) \times (randn) \times constant\ value \quad (13)$$

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

A pseudo-program describing the PF_{ACO} algorithm is given below.

Algorithm 2: The PF-ACO Algorithm

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

(1) The initialization and prediction steps [same as steps (1~2) in Algorithm 1]

(2) ACO improved PF

While the distance between particles' measurement and the true measurement are not within a certain threshold (Eqn. 13) and the iteration number 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. η, τ), and particle weights

(3) End While

(4) Update Step [same as steps (3~4) in Algorithm 1]

(5) Resampling

Optimized by the ACO, the effect of particle impoverishment problem existing in generic PF is alleviated, because the particle samples tend to be around neighborhood regions with high likelihood. 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 Figure 1. Therefore, when configured with suitable parameters (ρ, τ_0, η_0, $\Delta\tau$, threshold value, etc.), ACO is able to balance between the diversity and the impoverishment of particle filters.

In addition, since the impoverished particles concentrate in the higher likelihood region, the overlapping area between likelihood and the particles make the particles produce better contribution in the Monte Carlo simulation, so that the sample size dependency problem of the generic particle filter is also minimized.

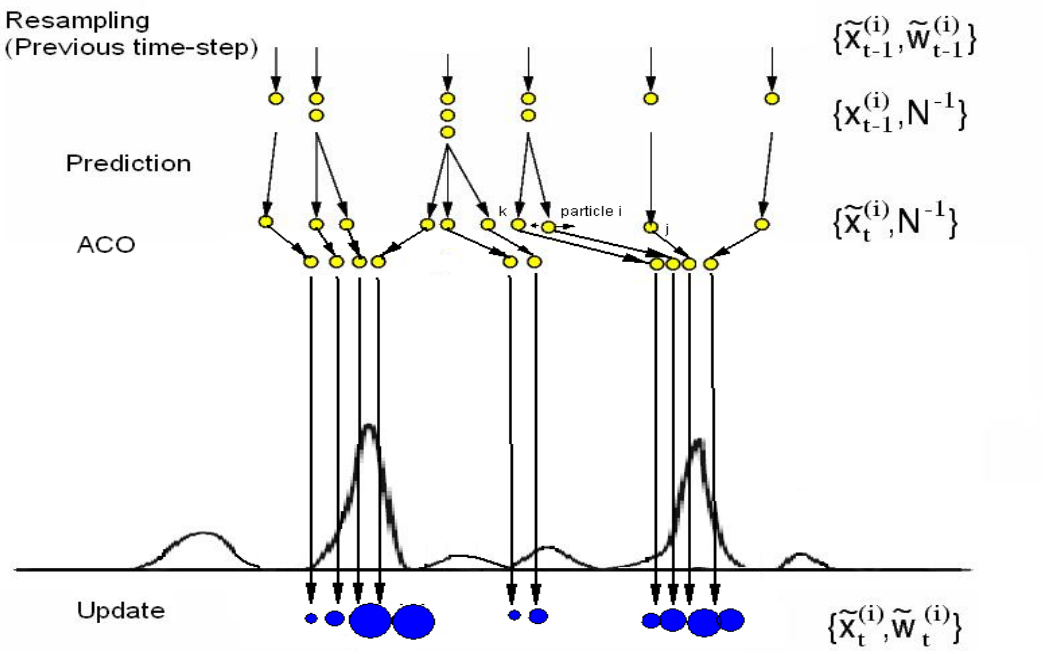


Figure 1 The PF_{ACO} Demonstration

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

3 Proofs of Optimization

In this section, a theorem will be proposed together with its proof in order to elaborate how the PF_{ACO} can produce better solution when compared to the generic PF, which employs a transition function as the proposal distribution.

Theorem 1: With the convergence nature of ACO, the PF_{ACO} can always achieve the optimal proposal distribution when the ACO converges to an optimal solution.

Proof: In the generic form, a transition model is often employed as the predicted proposal distribution:

$$q(x_k \mid x_{k-1}, y_k) = p(x_k \mid x_{k-1})_{tran} \tag{14}$$

while the optimal distribution is defined by Equation 15.

$$q(x_k \mid x_{k-1}^i, y_k)_{opt} = p(x_k \mid x_{k-1}^i, y_k) \tag{15}$$

where the $q(x_k \mid x_{k-1}, y_k)$ in Equation 14 represents the true distribution for the likelihood of state *x* when all previous states and observations are available. Since the probability is difficult to be determined by integration, so we usually employ the transition function $p(x_k \mid x_{k-1})_{tran}$ to approximate the true distribution.

The second term $p(x_k \mid x_{k-1}^i, y_k)$ in real world represents the probability that moving to state x_k in time *k*, given the samples in previous time step x_{k-1} and the measurement y_k.

In ideal cases, the proposal distribution should consider two kinds of noises: noises from the odometer and noise from the sensor. However, the generic transition model only incorporates the probability from motion detector noise. Consequently, the generic transition model can approximately equivalent to the optimal model only if either of following two conditions is satisfied.

1) The odometer has no error in measurement, or,
2) Variance from the odometer's noise and observation sensor noise are the same.

Nevertheless, the above two conditions are difficult to achieve in most of our experiments due to the different variances contributed by various sensors' measurement errors. The observation sensors, such as laser and vision sensors, are becoming more accurate, but this is not the same case for motion detectors. With different magnitudes of variance level, traditional transition model based on the motion sensors is not as suitable as it used to be, especially in experiments that

include observation sensors and motion sensors. Figure 2 shows a comparison of two different normal distributions and their combined distribution, which shows a non-Gaussian property illustrating the possible error included from the transition model (blue) from the true one (green).

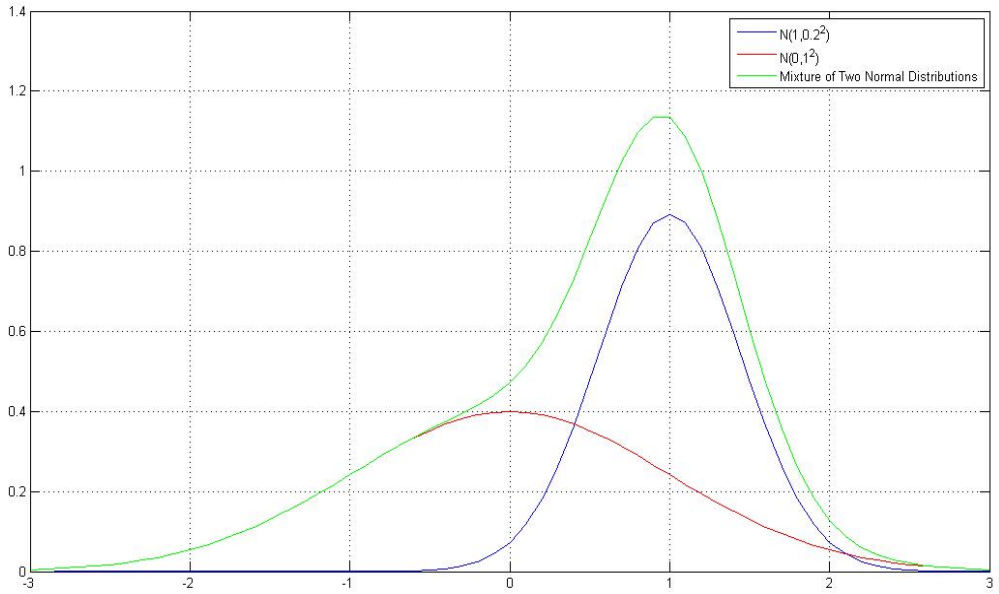


Figure 2 A mixture density of two different normal distributions, which having different variance values. We can tell that the mixture distribution is mainly dominated by the smaller variance normal distribution rather than the larger one.

In order to prove that ACO is able to solve this problem, Kullback–Leibler divergence (K-L divergence) is introduced. K-L divergence is a non-symmetric measure of the difference between two probability distributions. The approximation of K-L divergence [19] is generated by a set of sample data set: $s_1, s_2, \ldots s_N$, based on the model density *p(x)*, so

$$D(p \,\|\, q) \approx \frac{1}{N}\sum_{i=1}^{N}[\log p(x(n)) - \log q(x(n))] \tag{16}$$

For the generic PF, the above K-L Divergence equals to

$$D(p \,\|\, q) \approx \frac{1}{N}\sum_{i=1}^{N}[\log p(x_k(n) \,|\, x_{k-1}(i), y_k) - \log q(x_k(n) \,|\, x_{k-1}(i))] \tag{17}$$

To evaluate the K-L Divergence, we take *N* Monte Carlo samples in state space for

x_k, calculate their probability density given the condition of particle $x_{k-1}(i)$ and y_k. Note that two Monte Carlo methods, for the independent calculations of K-L Divergence and posterior, are separated, in which *N* samples (denoted as index *n*) are used to calculate K-L Divergence and drawn from *N* bins, and *M* samples (denoted as index *i*) are used to calculate the posterior (also known as PF). This definition is also used in the proof of theorem 2.

Based on Equation 11, it is trivial to derive that the ACO algorithm converges if and only if $p_{ij}(k)=1$, which indicates the necessary and sufficient conditions of ACO convergence is $d_{ij}=0$ or $\lim_{t\to\infty}\eta_{ij}(k)\to\sum_{s\in\text{all particles}}\eta_{is}(k)$. These two conditions are corresponding to the high probability density $p(x_k \mid y_k)$ and $p(x_k \mid x_{k-1})$, which are presented by the blue curve and red curve shown in Figure 2, respectively. Thus, **majority of particles will be located around the peak of the combined likelihood density function.**

Secondly, assuming that samples $\hat{x}_1,\hat{x}_2,...,\hat{x}_n$ in the optimal proposal distribution are taken, in order to approach the optimal proposal distribution according to the definition of K-L Divergence and our Theorem 1, we will derive the relationship between the number of samples and the optimal distribution. If it is necessary to have *M* samples ($\tilde{x}_k,\tilde{x}_{k+1},...,\tilde{x}_{k+M}$) in order to generate *N* samples ($\tilde{x}_k,\tilde{x}_{k+1},...,\tilde{x}_{k+N}$) in the continuous optimal proposal distribution, the number of samples needed to be considered is proportional to the second derivative of the optimal distribution according to the interpolation error [20], which can be illustrated by Figure 3 and Equation 18.

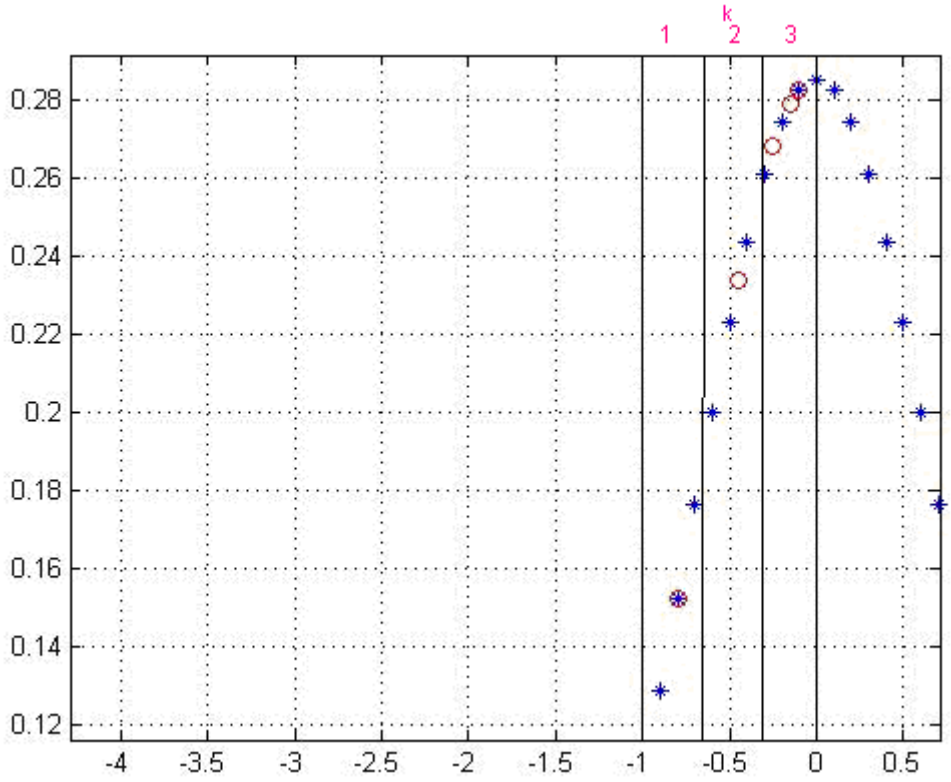


Figure 3 Demonstration of the samples size M and N.

Within [-1, 0], there are k=3 intervals. In the 1st and 2nd interval, M_1=1 and M_2=1 samples may be sufficient to represent the distribution because all the second derivatives in this interval are nearly equal to zero ,and M_3=3 sample are needed to re-construct the distribution. So the red samples are the most efficient and accurate representation samples. Comparatively, the blue ones are less efficient and accurate.

$$\begin{aligned} M &= \lambda N[f''(s_k) + f''(s_{k+1}) + \ldots + f''(s_{k+N})]/N \\ &= \lambda \bullet [f''(s_k) + f''(s_{k+1}) + \ldots + f''(s_{k+N})] \end{aligned} \tag{18}$$

In the above equation, λ is a constant, indicating that **number of *M* is proportional to the summation of the second derivatives of all samples in this interval.**

As shown in Figure 3, *k* samples in the optimal Gaussian distribution are taken in uniform intervals within which *M* samples are included in the original discrete distribution, denoted as $M_{i_k} \in \{M_1, M_2, \ldots, M_k\}$. Similarly, samples in the proposal

distribution are also separated into k intervals, that is $N_{i_k} \in \{N_1, N_2, \ldots, N_k\}$.

Because of the convergence of Ant Colony Optimization algorithm[18], given a certain continuous optimal proposal distribution with *M* samples, the sample s^+_t are moved from s_t after the ACO improvement.

Therefore, we can compare two K-L Divergence before the ACO improvement, then Equation 17 becomes

$$D(p \| q) \quad \frac{1}{M} \sum_{i_k=1}^{k} \sum_{n \in \text{interval k}} [\log \hat{p}(x_k(n) \mid x_{k-1}(i), y_k) - \log \tilde{q}(x_k(n) \mid x_{k-1}(i))] \quad (19)$$

and

$$D(p \| q^+) \approx \frac{1}{M} \sum_{i_k=1}^{k} \sum_{n \in \text{interval k}} [\log \hat{p}(x_k(n) \mid x_{k-1}(i), y_k) - \log \tilde{q}^+(x_k(n) \mid x_{k-1}(i))]. \quad (20)$$

Let the sequence $\hat{M}_{i_k} \in \{\hat{M}_1, \hat{M}_2, \ldots, \hat{M}_k\}$ denote the required particle number in each interval based on Equation 18. After sufficient iterations to achieve the optimal solution, if in an interval that the required particle number $\hat{M}_{i_k} \le N_{i_k}$, such as k = 1,2 as shown in Figure 3, it is trivial that

$$D(p \| q) = D(p \| q^+). \quad (21)$$

If within the intervals that the required particle number $\hat{M}_{i_k} > N_{i_k}$, such as k = 3 as illustrated in Figure 3, then

$$
\begin{aligned}
D(p\,\|\,q) - D(p\,\|\,q^{+}) &\approx \frac{1}{M}\sum_{n=1}^{M}[\log\frac{p(\hat{x}_k(n)\,|\,x_{k-1}(i), y_k)}{q(\tilde{x}_k(n)\,|\,x_{k-1}(i))} - \log\frac{p(\hat{x}_k(n)\,|\,x_{k-1}(i), y_k)}{q(\tilde{x}_k^{+}(n)\,|\,x_{k-1}(i))})] \\
&= \frac{1}{M}\sum_{n=1}^{M}[\log\frac{p(\hat{x}_k(n)\,|\,x_{k-1}(i), y_k)}{q(\tilde{x}_k(n)\,|\,x_{k-1}(i))} - \log\frac{p(\hat{x}_k(n)\,|\,x_{k-1}(i), y_k)}{p(\hat{x}_k(n)+\varepsilon\,|\,x_{k-1}(i), y_k)}] \\
&\to \frac{1}{M}\sum_{n=1}^{M}[\log\frac{p(\hat{x}_k(n)\,|\,x_{k-1}(i), y_k)}{q(\tilde{x}_k(n)\,|\,x_{k-1}(i))} - 0] \qquad (22.1) \\
&> 0
\end{aligned}
$$

(22)

The step (22.1) comes from the convergence of Ant Colony Optimization. So within the intervals that the required particle number $\hat{M}_{i_k} > N_{i_k}$, we get $D(p\,\|\,q) > D(p\,\|\,q^{+})$ Given a small number ε, with sufficient iterations, we can always achieve arbitrarily small K-L Divergence. When we take summation in all intervals for K-L Divergence calculation, we can conclude that $D(p\,\|\,q) \geq D(p\,\|\,q^{+})$.

□

The above theorem qualitatively shows that the proposal distribution can ultimately achieve the optimal with Ant Colony Optimization. A quantitative study to the performance of PF_{ACO} is included in the following theorem and proof.

The main purpose of the following proof is to validate the magnitude of error introduced by the sample-based representation of PF. To determine this bounded value, we assume that the optimal distribution is given by a discrete, piecewise constant distribution [21] such as a discrete density or a multi-dimensional histogram. For such a representation, we can determine the number of samples so that the K-L Divergence between the estimation based on the samples and the optimal distribution does not exceed a pre-defined probability, for proofing Theorem 1: the optimization of PF can be achieved if its probability is larger than that of generic PF with the same number of particle samples.

Theorem 2: If we need the efficient particle size n to approach the true distribution with an upper bound ε on the K-L Divergence with probability $1-\delta$ in generic PF, then in $PF_{ACO,}$ the probability is always 1.

Proof :

If we draw n samples $(X_1, X_2, ..., X_n)$ to represent the estimated distribution to estimate a multinomial true distribution. Assuming the true distribution is a multinomial distribution with N bins, we piecewise calculate its K-L Divergence and denote it as $\sum_{n=1}^{N} \hat{p}_n \log(\frac{\hat{p}_n}{p_n})$. Then we multiply this K-L divergence with the number of samples M, so it becomes a statistics ratio of testing $\hat{p}$ given a set of samples $X_j = (X_1, X_2, ..., X_n)$ as follows:

$$M \sum_{n=1}^{N} \hat{p}_n \log(\frac{\hat{p}_n}{p_n}) = \sum_{n=1}^{N} X \log(\frac{\hat{p}_n}{p_n}) = \log \varsigma_M \tag{23}$$

where ς_M denotes the likelihood ratio statistic. From [22], we know that $M \to \infty$, it converges to a chi-square distribution

$$2 \log \varsigma_M \to_d \chi^2_{K-1} \tag{24}$$

Therefore, we define that $P_p(D(\hat{p} \,||\, p) \le \varepsilon)$ is the probability of K-L Divergence between the true bin's distribution and the proposal distribution is less than or equal to ε. The number of samples can be calculated under chi-square distribution as follows:

$$\begin{aligned} P_p(D(\hat{p} \,||\, p) \le \varepsilon) &= P_p(2MD(\underline{\hat{p}} \,||\, \underline{p}) \le 2M\varepsilon) \\ &= P_p(2 \log \varsigma_M \le 2M\varepsilon) \\ &= P(\chi^2_{K-1} \le 2M\varepsilon) \end{aligned} \tag{25}$$

That is, if we choose the number of samples as

$$M = \frac{1}{2\varepsilon} \chi^2_{k-1,1-\delta} \tag{26}$$

then we can guarantee that with the probability *1-δ*, the K-L Divergence between the proposal distribution in one bin and the true distribution is less than ε with a finite particle size n.

According to Theorem 1, with each sample to represent a normal distribution, the K-L divergence between the proposal distribution and optimal distribution not bigger than ε, so that the quantile δ^+ should not be larger than the original δ, so

$$\begin{aligned} & D(\hat{p}_n^+ \| p_n) \le D(\hat{p}_n \| p_n) \\ \Rightarrow & \sum D(\hat{p}_n^+ \| p_n) \le \sum D(\hat{p}_n \| p_n) \\ \Rightarrow & M \sum D(\hat{p}_n^+ \| p_n) \le M \sum D(\hat{p}_n \| p_n) \\ \Rightarrow & \log \varsigma_M{}^+ \le \log \varsigma_M \end{aligned} \tag{27}$$

Therefore,

$$P(\chi^2_{k-1}{}^+ \le \varepsilon) \ge P(\chi^2_{k-1} \le \varepsilon) \tag{28}$$

Equation 28 implies that with any ε, after the algorithm of PF_{ACO}, we can achieve the K-L Divergence with higher probability to make the K-L divergence smaller or equal to ε with the same number of samples.

□

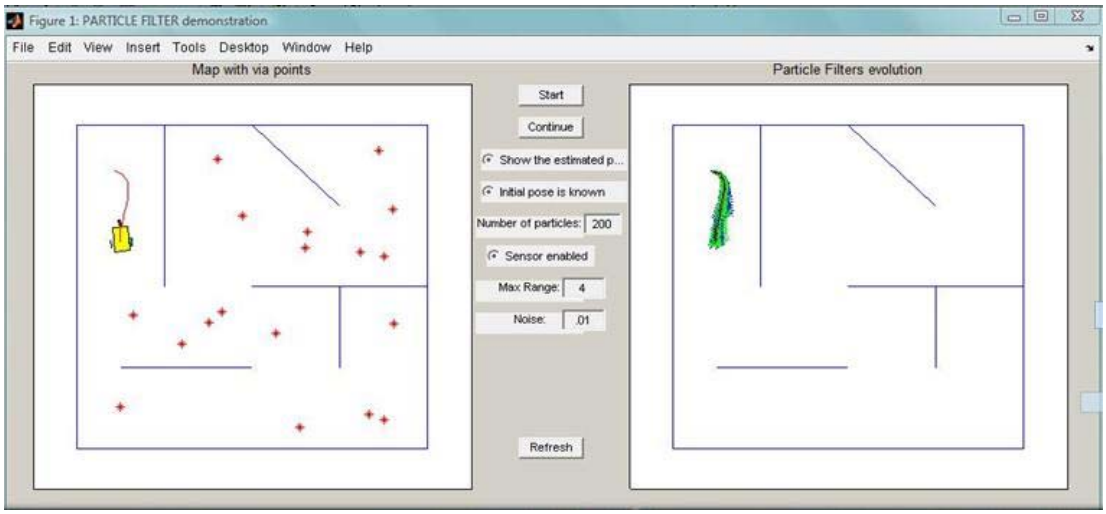
To conclude, the previous two theorems present the theoretical foundation of PF_{ACO}. First, it will self-converge to the optimal proposal distribution after infinite iterations. Second, with a pre-defined threshold, if the generic PF need an efficient particle size to approach the optimal distribution with a certain probability, then the PF_{ACO} has a greater probability to achieve the optimal value with the same threshold level and with the same particle number.

4 Case Study in Robot Localization

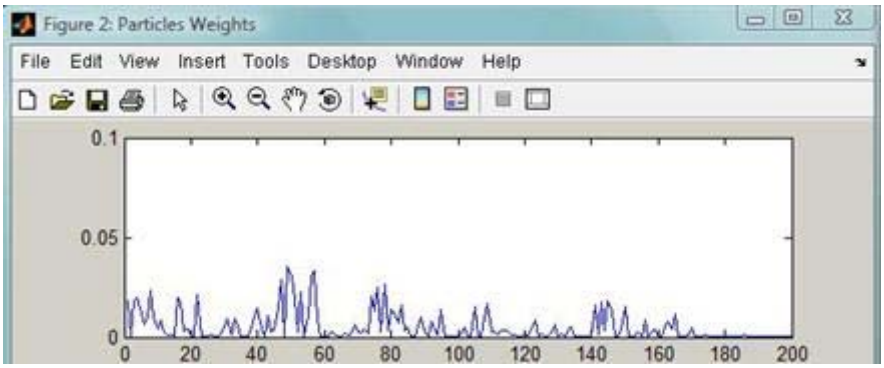
In order to evaluate the performance of the PF_{ACO} in engineering applications, the robot localization problem is chosen. The program was implemented and tested based

on a Matlab program originally developed by Vale [23] and modified by the first author.

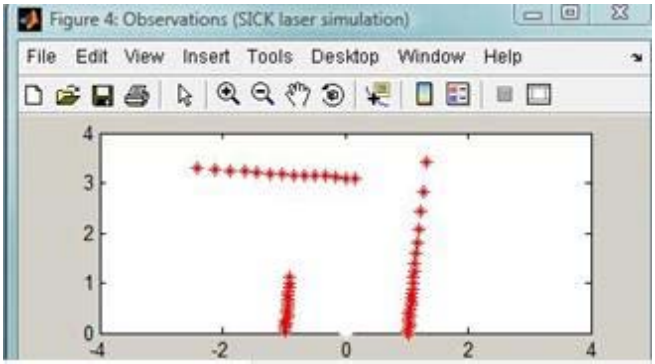
The program is capable to simulate the movement of a mobile robot as well as its noisy measurement data coming from a SICK laser sensor when maneuvering in an environment with a given map which is represented by a matrix stored in a file. As shown in Figure 4a, blue particles on the right chart represent the hypothesis of positions produced by generic PF, while the green particles are generated by the PF_{ACO}. When the estimation is in progress, the blue and green particles will be re-drawn continuously and weight of each particle is being plotted (Figure 4b). The noisy observation of sensors is illustrated in Figure 4c at the same time. In the following experiments, various PFs were applied to estimate the robot-pose (including the position and orientation) while the robot is moving along a series of asterisks. The sequence and positions of these asterisks are set and stored in a matrix and the robot will move towards these asterisks one after another as shown in Figure 5.



(a)



(b)



(c)

Figure 4 Interface of Particle Filters in Robotic Localization Demonstration (a)The left map shows the true map and the trajectory of the robots' navigation (red curve) along the asterisks set in advance, and the right map is the evolution demonstrations of particles representing the robot's pose while the robot is traversing in the area. (b) The graphs depict the pose particles' weight distribution of robot. (c)It shows the observation from the SICK laser sensor equipped in robot.

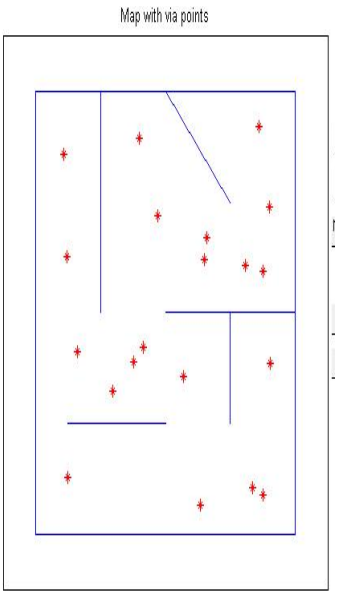
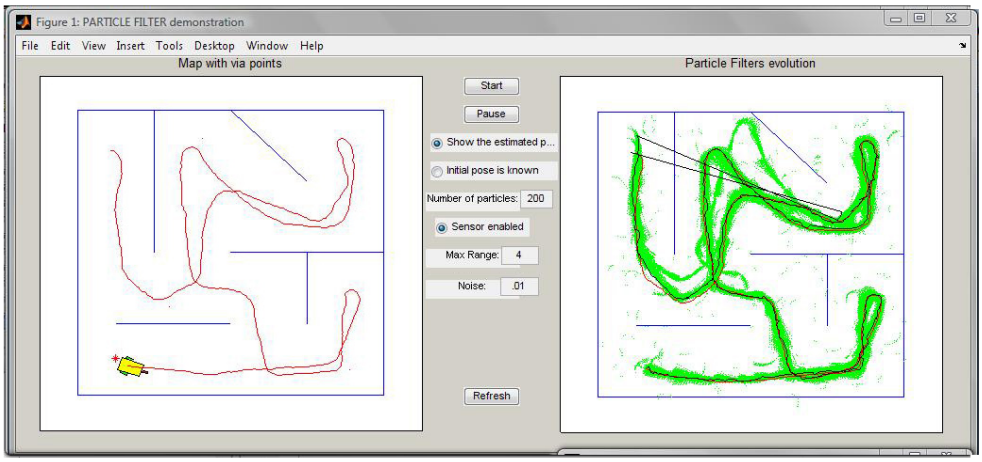


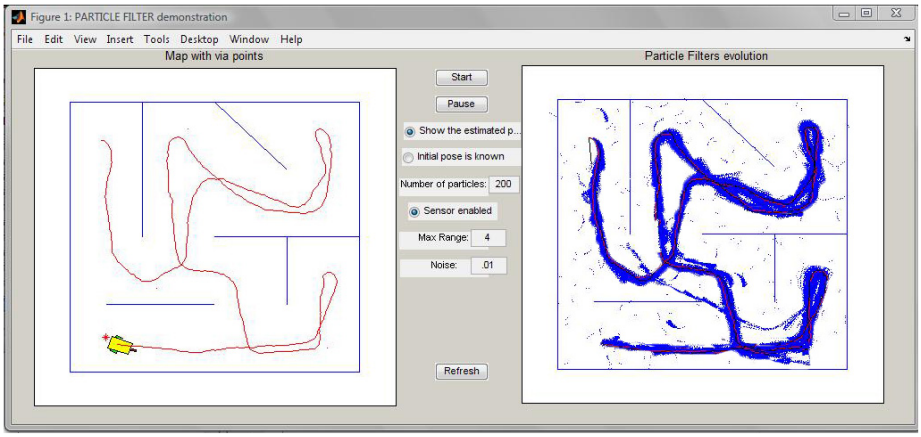
Figure 5 Asterisks as the navigation targets

The positions of asterisks are pre-defined and stored in a matrix. Robot will move along these asterisks during the experiments.

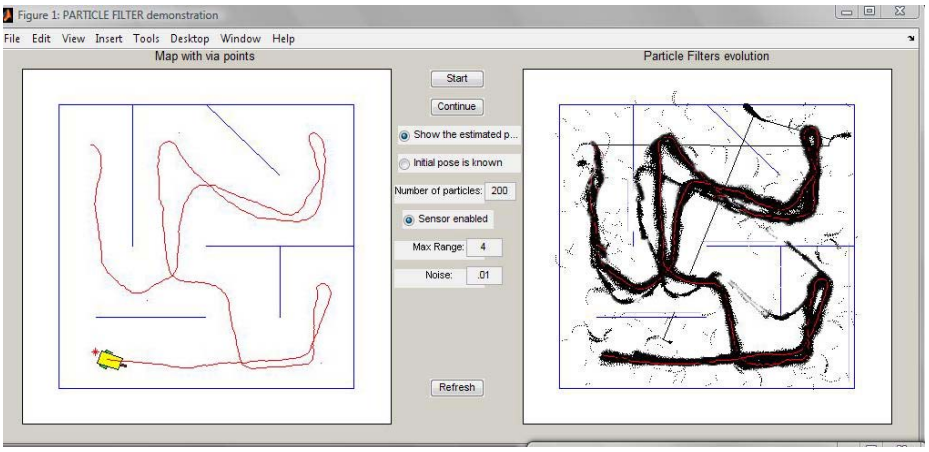
The generic PF, extended Kalman PF, Unscented PF, and PF_{ACO} were applied to the same localization problem. Without the prior information of initial poses, neither filter could track the poses accurately at the beginning so some traces, representing multiple hypotheses, appear in the particle evolution demonstration window (the right-hand-side of Fig. 6) at some initial steps. However, once sufficient information has been obtained, all filters could return to the correct trajectory.



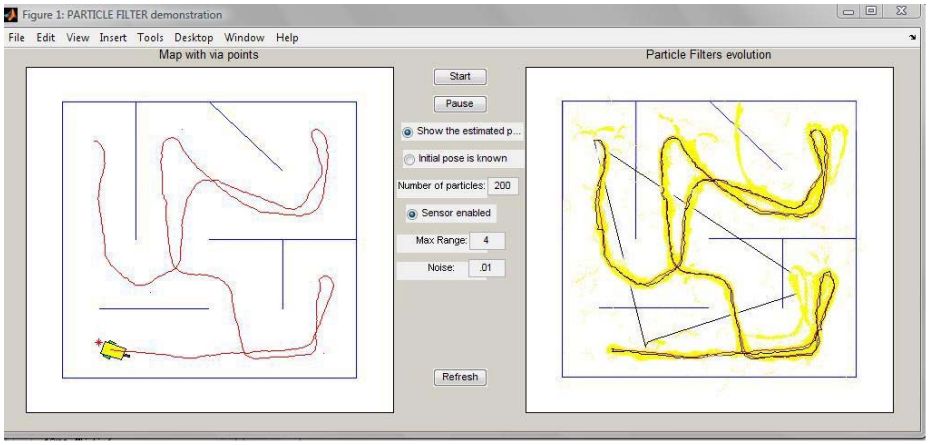
(a)



(b)



(c)



(d)

Figure 6 Navigation Progress

(α) The localization result from generic PF, in which the particle deviation happened in upper middle of the map, and lasted for a long running process. (b) The localization result from extended Kalman PF, in which the particle deviation happened in right lower corner of the map. (c) The localization result from Unscented PF, in which the particle deviation happened from the right top corner to the middle lower part.(d) The localization result from extended Kalman PF, in which there are also some deviations. But comparatively, they are shorter than the other three PF methods.

Particle deviation refers to the fact that the particle set deviates from the true trajectory with a significant distance. In mathematics, the deviation makes these particles negligible in Monte Carlo estimation because these particles will have a relatively small weights, therefore, this can be considered as an explanation of particle impoverishment problem.

To evaluate the results quantitatively, we compare the RMS error (Table 1) of the position and orientation of the robot.

Filters	RMS Error in position	RMS Error in orientation
Generic PF	4.5045	3.7746
Extended Kalman PF	3.2918	2.3207
Unscented PF	2.8605	2.2439
PF+ACO	1.9027	0.8529

Table 1 RMS Error of Four PF in single robot localization (particle number is 200)

Referring to Table 1, it is obviously that PF_{ACO} produces smaller errors in robot position and orientation estimation than the other three methods with the same number of particles. It is because our ACO plays an important role in the optimization of particle distribution in PF, therefore, it reduces the particle impoverishment problem.

Table 2 shows the execution time of the four PF algorithms, using 200 particles. The generic PF took the shortest computation time, and PF_{ACO} took shorter time than the extended Kalman PF as well as the Unscented PF.

Filters	Execution Time
Generic PF	42.17231
Extended Kalman PF	60.23145
Unscented PF	91.93079
PF+ACO	53.19893

Table 2 Execution Time for Four PF (in Sec)

From the above results, the RMS error from PF_{ACO} was less than that obtained from other PF algorithms. However, similar to the devised PF methods, PF_{ACO} needs longer execution time than generic PF. In order to reduce execution time, fewer particles can be utilized with the error level kept below a threshold. PF_{ACO} with fewer particles can even produce better results in terms of execution time and tracking accuracy as stated in [13, 14].

5 Conclusions

As a continuous study of the Ant Colony Improved Particle Filter (PF_{ACO}), a simulation based on an engineering application namely the robotic localization

problem and theorem proofs are presented in this paper. Our theorems show that the PF_{ACO} manipulates the proposal distribution to generate a smaller Kullback–Leibler divergence value than that obtained from generic PF; so PF_{ACO} will have a higher probability to maintain the pre-set error level. The case study shows that the PF_{ACO} provides better results than the generic PF as well as other devised PF algorithms. To conclude, the PF_{ACO} optimizes the particle distributions and approaches the optimal proposal distribution in PF estimation, and therefore, it introduces better estimation results in non-linear as well as non-Gaussian engineering estimation problems.

Acknowledgement

The work presented in this paper is supported by the Department of Electrical Engineering of the Hong Kong Polytechnic University.

Reference List

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." *Proceeding of IEEE Symposium of Adaptive Systems for Signal Processing, Communication, and Control* , pp. 153-158. 2000.
4. A.Ryan, J.Tisdale, M.Godwin et al., "Decentralized Control of Unmanned Aerial Vehicle Collaborative Sensing Missions." *American Control Conference, 2007.* pp. 4672-4677. 2007.
5. 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.
6. D.Crisan and A.Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing,* vol. 50, no. 3. pp.736-746, 2002.
7. M.G.S.Bruno and A.Pavlov, "Improved particle filters for ballistic target tracking." *IEEE International Conference on Acoustics, Speech, and Signal Processing* vol. 2, pp. 705-708. 2004.

Ref Type: Conference Proceeding

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 et al., "A Real-Time Object Tracking System Using a Particle Filter." *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 2822-2827. 2006.
11. A. Doucet and N. J. Gordon, "Simulation-based optimal filter for maneuvering target tracking." *Proceedings of SPIE* vol. 3809, p.241. 1999.
12. C.Stachniss, G.Grisetti, and W.Burgard, "Recovering particle diversity in a Rao-Blackwellized Particle Filter for SLAM after actively closing loops." *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* , pp. 18-22. 2005.
13. J. P. Zhong and Y. F. Fung, "A biological inspired improvement strategy for Particle Filters." *Industrial Technology, 2009.ICIT 2009.IEEE International Conference on* , pp. 1-6. 2009. IEEE.
14. J. Zhong, Y. Fung, and M. Dai, "A biologically inspired improvement strategy for particle filter: Ant colony optimization assisted particle filter," *International Journal of Control, Automation and Systems,* vol. 8, no. 3. pp.519-526, 2010.

15. M.S.Arulampalam, S.Maskell, N.Gordon et al., "A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking," *IEEE Transactions on Signal Processing,* vol. 50, no. 2. pp.174-188, 2002.

16. L.Li, H.Ji, and J.Luo, "The iterated extended Kalman particle filter." *IEEE International Symposium on Communications and Information Technology 2005* vol. 2, pp. 1213-1216. 2006.
17. R. Van Der Merwe, A. Doucet, N. De Freitas et al., "The unscented particle filter," *Advances in Neural Information Processing Systems*. pp.584-590, 2001.

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

19. J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models." *Acoustics, Speech and Signal Processing, 2007.ICASSP 2007.IEEE International Conference on* vol. 4, p.IV-317. 2007. Ieee.
20. P. J. Davis, *Interpolation and approximation*, 1975.

21. D. Koller and R. Fratkina, "Using learning for approximation in stochastic processes." *Proc.of the International Conference on Machine Learning (ICML)* , pp. 287-295. 1998. Citeseer.
22. J. A. Rice and J. A. Rice, *Mathematical statistics and data analysis,* 2: Duxbury press Belmont CA; 1995.

23. A. Vale and M. I. Ribeiro, "A probabilistic approach for the localization of mobile robots in topological maps." *Proc.of the 10th IEEE Mediterranean Conf.on Control and Automation, Lisboa, Portugal* . 2002.