ISSN (Online): 1694-0784
ISSN (Print): 1694-0814

A Multi Swarm Particle Filter for Mobile Robot Localization

Ramazan Havangi [1], Mohammad Ali Nekoui [2] and Mohammad Teshnehlab [3]

**[1] Faculty of Electrical Engineering, K.N. Toosi University of Technology
Tehran, Iran**

**[2] Faculty of Electrical Engineering, K.N. Toosi University of Technology
Tehran, Iran**

**[3] Faculty of Electrical Engineering, K.N. Toosi University of Technology
Tehran, Iran**

Abstract

Particle filter (PF) is widely used in mobile robot localization, since it is suitable for the nonlinear non-Gaussian system. Localization based on PF, However, degenerates over time. This degeneracy is due to the fact that a particle set estimating the pose of the robot looses its diversity. One of the main reasons for loosing particle diversity is sample impoverishment. It occurs when likelihood lies in the tail of the proposed distribution. In this case, most of particle weights are insignificant. To solve those problems, a novel multi swarm particle filter is presented. The multi swarm particle filter moves the samples towards region of the state space where the likelihood is significant, without allowing them to go far away from the region of significant values for the proposed distribution. The simulation results show the effectiveness of the proposed algorithm.

Keywords: Localization, Particle Filter, Particle Swarm Optimization (PSO)

1. Introduction

Mobile localization is the problem of estimating a robot's pose (location, orientation) relative to its environment. It represents an important role in the autonomy of a mobile robot. From the viewpoint of probability, the localization problem is a state estimation process of a mobile robot. Many existing approaches rely on the kalman filter (KF) for robot state estimation. But it is very difficult to be used in practice since KF can only be used in Gaussian noise and linear systems. To solve the problem of nonlinear filtering, the extended kalman filter (EKF) was proposed. The localization based on EKF was proposed in [1], [2], [3], [4], [5], [6] for the estimation of robot's pose. However, the localization based on EKF has the limitation that it doses not apply to the general non-Gaussian distribution. In order to represent non-linearity and non-Gaussian characteristics better, particle filter was proposed in [19], [20]. Particle filter outperforms the EKF for nonlinear systems and has been successfully used in robotics. In recent years, the particle filter (PF) is widely used in localization [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. The central idea of particle filters is to represent the posterior probability density distribution of the robot by a set of particles with associated weights. Therefore, the particle filters do not involve linearzing the models of the system and are able to cope with noises of any distribution. However, localization based on particle filter also has some drawbacks. In [19], [20], [21], [22], [23], [24], it has been noted that it degenerates over time. This degeneracy is due to the fact that particle set estimating the pose of the robot looses its diversity. One of main reasons for loosing particle diversity is sample impoverishment. It occurs when likelihood is highly peaked compared to the proposed distribution, or lies in the tail of the proposed distribution. On the other hand, PF highly relies on the number of particles to approximate the distribution density [19], [20], [21], [22], [23], [24]. Researchers have been trying to solve those problems in [21], [22], [23], and [24]. In all the aforementioned studies, the reliability of measurement plays a crucial role in the performance of the algorithm and additive noise was considered only. In this paper to solve those problems, a novel multi swarm particle filter is purposed. The multi swarm particle filter move samples towards the region of the state space where the likelihood is significant, without allowing them to go far away from the region of significant values of the proposed distribution. For this purpose, the multi swarm particle filter employs a conventional multi objective optimization approach to weight the likelihood and prior of the filter in order to alleviate the particle impoverishment problem. The minimization of the corresponding objective function is performed using the Gaussian PSO algorithm,

2. Kinematics Modeling Robot and its Odometery

The state of robot can be modeled as (x, y, θ) where (x, y) are the Cartesian coordinates and θ is the orientation respective to the global environment. The kinematics equations for the mobile robot are in the following form [1-2] and [4]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = f(X) = \begin{bmatrix} (V+v_v)\cos(\phi+[\gamma+v_\gamma]) \\ (V+v_v)\sin(\phi+[\gamma+v_\gamma]) \\ \frac{(V+v_v)}{B}\sin(\gamma+v_\gamma) \end{bmatrix} \quad (1)$$

Where B is the base line of the vehicle and $u = [V \quad \gamma]^T$ is the control input consisting of a velocity input V and a steer input γ, as shown in Fig.1.

The process noise $v = [v_v \quad v_\gamma]^T$ is assumed to be applied to the control input, v_v to velocity input, and v_γ to the steer angle input. The vehicle is assumed to be equipped with a sensor (range-laser finder) that provides a measurement of range r_i and bearing θ_i to an observed feature ρ_i relative to the vehicle as follows:

$$\begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = h(X) = \begin{bmatrix} \sqrt{(x-x_i)^2+(y-y_i)^2} + \omega_r \\ \tan^{-1}\frac{y-y_i}{x-x_i} - \phi + \omega_\theta \end{bmatrix} \quad (2)$$

where (x_i, y_i) is the position landmark in the map and $W = [\omega_r \quad \omega_\theta]^T$ relates to the observation noise.

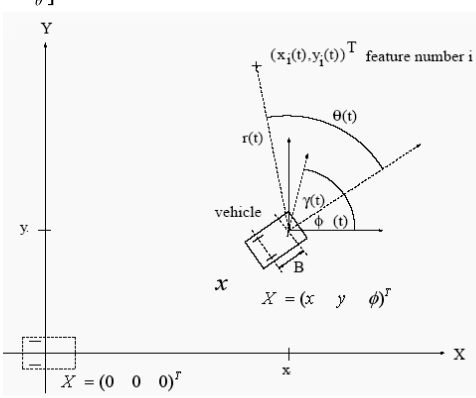

Fig.1 The Robot and Feature

3. Particle filter Principle

The particle filter is a special version of the Bayes filter, and is based on sequential Monte Carlo (SMC) sampling. A dynamic system represented by

$$x_k = f(x_{k-1}, \omega_k) \quad (3)$$

$$y_k = h(x_k, v_k) \quad (4)$$

is considered, where $x_k \in R^n$ is the state vector and $y_k \in R^m$ is an output vector. $f(.)$ and $g(.)$ denote the system and measurement equations, respectively. ω_k and v_k are independent white-noise variables. Particle filter represents the posterior probability density function $p(x_k \mid y_{1:k})$ by a set of random samples with associated weight as follows [19], [20]:

$$S_k = \{(x_k^i, w_k^i) \mid i = 1, \ldots, N\} \quad (5)$$

where x_k^i denotes the i th particle of S_k, w_k^i is the associated importance weight and $y_{1:k}$ denotes the measurements accumulated up to k . Then, the posterior density $p(x_k \mid y_{1:k})$ can be approximated as follows [19], [20]:

$$p(x_k \mid y_{1:k}) \approx \sum_{i=1}^{n} w_k^i \delta(x_k - x_k^i) \quad (6)$$

Where $\delta(x)$ is Dirac's delta function ($\delta(x) = 1$ for $x = 0$ and $\delta(x) = 0$ otherwise), and $w_k^{(i)}$ is associated weight x_k^i with $w_k^i > 0$, $\sum_{i=1}^{n} w_k^i = 1$. In general, it is not possible to draw samples directly from posterior $p(x_k \mid y_{1:k})$. Instead, the samples are drawn from a simpler distribution called the proposed distribution $q(x_k \mid y_{1:k})$ .The mismatch between the posterior and proposed distributions is corrected using a technique called importance sampling. Therefore, in regions where the target distribution is larger than the proposed distribution, the samples are assigned a larger weigh. Also, in regions where the target distribution is smaller than the proposed distribution the samples will be given lower weights. An example of importance sampling is shown in Fig. 2

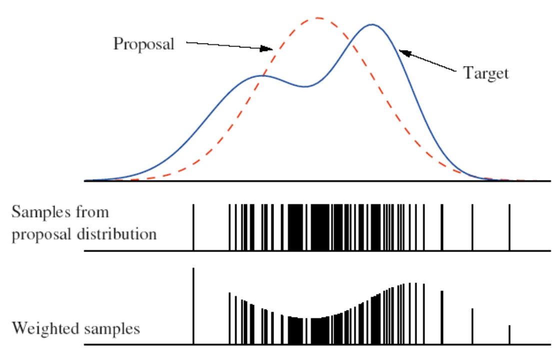

Fig2.Important Sampling

As a result, the important weight of each particle is equal to the ratio of the target posterior and the distribution proposal as follows:

$$w_k^i = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(x_k^i \mid y_{1:k})}{q(x_k^i \mid y_{1:k})} \qquad (7)$$

The proposed $q(x_k^i \mid y_{1:k})$ can be represented by a recursive form as:

$$q(x_k^i \mid y_{1:k}) = q(x_k^i \mid x_{k-1}^i, y_{1:k}) q(x_{k-1}^i \mid y_{1:k}) \overset{Markov}{=} q(x_k^i \mid x_{k-1}^i, y_{1:k}) q(x_{k-1}^i \mid y_{1:k-1}) \qquad (8)$$

Then one can obtain samples $x_k^i \sim q(x_k \mid y_{1:k})$ by augmenting each of the exiting samples $x_{k-1}^i \sim q(x_{k-1} \mid y_{1:k-1})$ with the new state $x_k^i \sim q(x_k \mid X_{k-1}, y_{1:k})$. Similarity, the posterior can also be given by a recursive form using Bayes rule as follows:

$$\begin{aligned} p(x_k^i \mid y_{1:k}) &= \frac{p(y_k \mid x_k^i, y_{1:k-1}) p(x_k^i \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})} \\ &= \frac{p(y_k \mid x_k^i, y_{1:k-1}) p(x_k^i \mid x_{k-1}^i, y_{1:k-1})}{p(y_k \mid y_{1:k-1})} p(x_{k-1}^i \mid y_{1:k-1}) \\ &= \frac{p(y_k \mid x_k^i, y_{1:k-1}) p(x_k^i \mid x_{k-1}^i)}{p(y_k \mid y_{1:k-1})} p(x_{k-1}^i \mid y_{1:k-1}) \\ &\propto p(y_k \mid x_k^i, y_{1:k-1}) p(x_k^i \mid x_{k-1}^i) p(x_{k-1}^i \mid y_{1:k-1}) \end{aligned} \qquad (9)$$

Therefore, a sequential importance weight of the m th particle can be obtained as follows:

$$w_k^i \propto \frac{p(x_k^i \mid y_{1:k})}{q(x_k \mid x_{k-1}, y_{1:k-1}) q(x_{k-1} \mid y_{1:k-1})} \qquad (10)$$

$$\begin{aligned} w_k^i &\propto \frac{p(y_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i) p(x_{k-1}^i \mid y_{1:k-1})}{q(x_k^i \mid x_{k-1}^i, y_{1:k}) q(x_{k-1}^i \mid y_{1:k-1})} \\ &= w_{k-1}^i \frac{p(y_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, y_{1:k})} \end{aligned} \qquad (11)$$

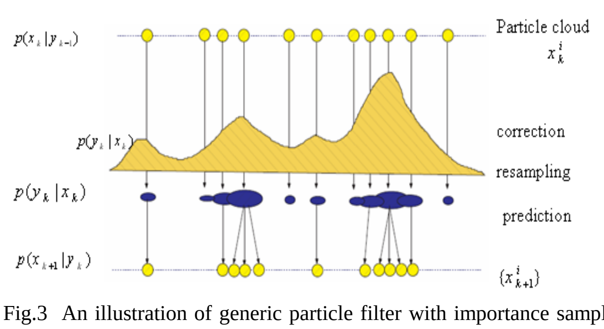

Fig.3 An illustration of generic particle filter with importance sampling and resampling.

A particle filter described above is called the sequential Importance Sampling (SIS). The SIS algorithm has a problem that it degenerates quickly over time. In practical terms this means that after a certain number of recursive steps, most particles will have negligible. Degeneracy can be reduced by using a resampling step [19], [20]. Resampling is a scheme to eliminate particles small weights and to concentrate and replace on particles with large weights. Fig.3 shows the generic particle filter with importance sampling and resampling.

4. Localization Based on Particle Filter

From the viewpoint of Bayesian, Mobile robot localization is basically a probability density estimation problem. In fact, Localization is estimating the posterior probability density of the robot's pose relative to a map of its environment. Assuming that the robot's pose at time k is denoted by x_k and measurements up to time k is denoted by Y_k, the posterior probability distribution is as follow:

$$p(x_k \mid Y_k, m) \qquad (12)$$

Where m is the map of the environment which is known. The measurement data Y_t comes from two different sources: motion sensors which provide data relating to the change of situation (e.g., odometer readings) and perception sensors which provide data relating to the environment (e.g., laser range scans). In other words, measurement data can be divided in two groups of data as $Y_k = \{Z_k, U_{k-1}\}$ where $Z_k = \{y_0, \ldots, y_k\}$ contains the range laser finder measurements and $U_{k-1} = \{u_0, .. u_{k-1}\}$ contains the odometric data. The Bayesian recursive determination of the posterior density can be computed in two steps:

1) Measurement update

$$\begin{aligned} p(x_k \mid Y_k, m) &= \frac{p(y_k \mid x_k, Y_{k-1}, m) p(x_k \mid Y_{k-1}, m)}{p(y_k \mid Y_{k-1}, m)} \\ &= \frac{p(y_k \mid x_k, m) p(x_k \mid Y_{k-1}, m)}{p(y_k \mid Y_{k-1}, m)} \end{aligned}$$

(13)

where

$$p(y_k \mid Y_{k-1}, m) = \int p(y_k \mid x_k . m) p(x_k \mid Y_{k-1}, m) dx_k \qquad (14)$$

2) Prediction

$$\begin{aligned} p(x_{k+1} \mid Y_k, m) &= \int p(x_{k+1} \mid x_k, u_k, Y, m_k) \\ p(x_k \mid Y_{k-1}, m) dx_k &= \int p(x_{k+1} \mid x_k, u_k, m) \\ p(x_k \mid Y_{k-1}, m) dx_k & \end{aligned} \qquad (15)$$

The localization based on PF represents the posterior probability density function $p(x_k \mid Y_k, m)$ with N weighted samples

$$p(x_k \mid Y_k, m) = \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i) \qquad (16)$$

The localization algorithm of the mobile robot is realized using particle filter as following:

1. Sampling a new robot pose.

3. Calculate importance weight and normalization.
4. Normalized Wight
The normalized weights are given by:

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^{N} w_k^i} \quad (17)$$

4. Resampling
In the following subsections we give details of the main steps. To alleviate the notation, the term m is not included in the following expressions, $p(x_k | Y_k)$.

4.1 Sampling a New of Pose

The choice of importance density $q(x_k^i | x_{k-1}^i, Y_k)$ is one of the most critical issues in the design of a particle filter. Two of those critical reasons are as follows: samples are drawn from the proposed distribution, and the proposed distribution is used to evaluate important weights. The optimal importance density function minimizes the variance of the importance weights through the following equation [19], [20].

$$q_{opt}(x_k^i | x_{k-1}^i, Y_k) = p(x_k^i | x_{k-1}^i, Y_k) \quad (18)$$

However, there are some special cases where the use of the optimal importance density is possible. The most popular suboptimal choice is the transitional prior

$$q_{opt}(x_k^i | x_{k-1}^i, Y_k) = p(x_k^i | x_{k-1}^i) \quad (19)$$

In this paper, the proposed distribution in equation (19) is used due to its easy calculation. Hence, by the substitution of (19) into (11), the weight's update equation is:

$$w_k^i \propto w_{k-1}^i p(y_k | x_k^i) \quad (20)$$

4.2 Resampling

Sine the variance of the importance weights increases over time [21], [23], [25], resampling plays a vital role in the particle filter. In the resampling process, particles with low importance weight are eliminated and particles with high weights are multiplied. After, the resampling, all particle weights are then reset to

$$w_t^i = \frac{1}{N} \quad (21)$$

This enables the particle filter to estimate the robot's pose defiantly without growing a number of particles. However, resampling can delete good samples from the sample set, and in the worst case, the filter diverges. The decision on how to determine the point of time of the resampling is a fundamental issue. Liu introduced the so-called effective number of particles N_{eff} to estimate how well the current particle set represents the true posterior. This quality is computed as

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} w_k^i} \quad (22)$$

Where w^i refers to the normalized weight of particle i .The resampling process is operated whenever N_{eff} is bellow a pre-defined threshold, N_{tf} . Here N_{tf} is usually a constant value as following

$$N_{tf} = \frac{3}{4} M \quad (23)$$

Where M is number of particles.

5. A Modified Localization Based on Particle Filter

Particle Filter relies on importance sampling, i.e., it uses proposed distributions to approximate the posterior distribution. The most common choice of the proposed distribution that is used also in this paper is the probabilistic model of the states evolution, i.e., the transition prior $p(x_k^i | x_{k-1}^i)$. Because the proposed distribution is suboptimal, there are two serious problems in particle filter. One problem is sample impoverishment, which occurs when the likelihood $p(z_k | x_k^i)$ is very narrow or likelihood lies in the tail of the proposed distribution $q(x_k^i | x_{k-1}^i, y_{1:k})$. The prior distribution is effective when the observation accuracy is low. But it is not effective when prior distribution is a much broader distribution than the likelihood (such as Fig.4.). Hence, in the updating step, only a few particles will have significant importance weights.

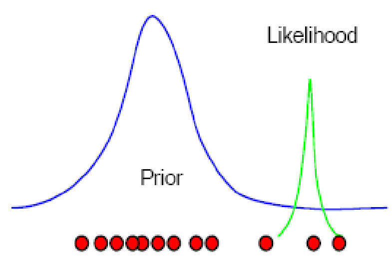

Fig.4 Prior and Likelihood

This problem implies that a large computational effort is devoted to update the particles with negligible weight. Thus, the sample set only contains few dissimilar particles and sometimes they will drop to a single sample after several iterations. As a result, important samples may be lost. Another problem of particle filter is the number of particles dependency that estimates the pose of the robot. If the number of particles is small, then there might not have been particles distributed around the true pose of the robot. So after several iterations, it is very difficult for particles to converge to the true pose of the robot. For standard particle filter, there is one method to solve the problem. This is to augment the number of the particles. But this would make the computational complexity unacceptable. To solve these problems of particle filter, particle swarm optimization is considered to optimize the sampling process of the particle filter.

5.1 Particle Swarm Optimization

James Kennedy and Russell C.Eberhart [25] originally proposed the PSO algorithm for optimization. PSO is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. PSO is initialized with a group of random particles and then computes the fitness of each one. Finally, it can find the best solution in the problem space via many iterations. In each iteration, each particle keeps track of its coordinates which are associated with the best solution it has achieved so far (pbest) and the coordinates which are associated with the best solution achieved by any particle in the neighboring of the particle (gbest). Supposing that the search space dimension is D and number particles is N, the position and velocity of the i-th particle are represented as $x_i = (x_{i1},...,x_{iD})$ and $v_i = (v_{i1},...,v_{iD})$ respectively.

Let $P_{bi} = [p_{i1},...p_{iD}]$ denote the best position which the particle i has achieved so far, and P_g the best of P_{bi} for any $i = 1,...,N$. The PSO algorithm could be performed by the following equations:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (24)$$

$$\vec{v}_i(t) = w\vec{v}_i(t-1) + c_1 r_1(\vec{P}_{bi} - \vec{x}_i(t-1)) + c_2 r_2(\vec{P}_g - \vec{x}_i(t-1)) \quad (25)$$

Where t represents the iteration number and c_1, c_2 are the learning factors. Usually $c_1 = c_2 = 2$. r_1, r_2 are random numbers in the interval $(0,1)$. w is the inertial factor, and the bigger the value of w , the wider is the search range.

5.2 Localization based Multi Swarm particle filter

As discussed in the previous section, impoverishment occurs when the number of particles in the high likelihood area is low. We address this problem by intervening at Localization based on PF after the generation of the samples in prediction phase and before resampling. The aim is to move these samples towards the region of the state space where the likelihood is significant, without allowing them to go far away from the region of significant prior. For this purpose, we consider a multi objective function as follows:

$$F = F_1 + F_2 \quad (26)$$

The first objective consists of a function that is maximized at regions of high likelihood as follows:

$$F_1 = e^{-\frac{1}{2}(y_k - \hat{y}_k)^T [R]^{-1}(y_k - \hat{y}_k)} \quad (27)$$

Here, R is the measurement noise covariance matrix, $\hat{y}$ is the predicted measurement and y is the actual measurement. While the second objective, F2, is maximized at regions of high prior.

$$F_2 = e^{-\frac{1}{2}(x_k - \hat{x}_k)^T [Q]^{-1}(x_k - \hat{x}_k)} \quad (28)$$

where Q is the measurement noise covariance matrix. We use an easy idea to solve this problem. The basic idea is that particles are encouraged to be at the region of high likelihood by incorporating the current observation without allowing them to go far away from the region of significant prior before the sampling process. This implies that a simple and effective method for this purpose is the using of PSO. In fact, by using PSO, we can move all the particles towards the region that maximizes the objective function F before the sampling process. For this purpose, we consider a fitness function as follows:

$$\text{Fitness(k)} = \frac{1}{2}(y_k - \hat{y}_k)R^{-1}(y_k - \hat{y}_k)^{-1} + (x_k - \hat{x}_k^-)Q^{-1}(x_k - \hat{x}_k^-)^T \quad (29)$$

The particles should be moved such that the fitness function is optimal. This is done by tuning the position and velocity of the PSO algorithm. The standard PSO algorithm has some parameters that need to be specified before use. Most approaches use uniform probability distribution to generate random numbers. However it is difficult to obtain fine tuning of the solution and escape from the local minima using a uniform distribution. Hence, we use velocity updates based on the Gaussian distribution. In this situation, there is no more need to specify the parameter learning factors c_1 and c_2. Furthermore, using the Gaussian PSO the inertial factor ω was set to zero and an upper bound for the maximum velocity $v_{\max}$ is not necessary anymore [26]. So, the only parameter to be specified by the user is the number of particles. Initial values of particle filter are selected as the initial population of PSO. Initial velocities of PSO are set equal to zero. The PSO algorithm updates the velocity and position of each particle by following equations [26]:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (30)$$

$$\vec{v}_i(t) = |randn|(P_{pbest} - \vec{x}_i(t-1)) + |randn|(P_{gbest} - \vec{x}_i(t-1)) \quad (31)$$

PSO moves all particles towards particle with best fitness. When the best fitness value reaches a certain threshold, the optimized sampling process is stopped. With this set of particles the sampling process will be done on the basis of the proposed distribution. The Corresponding weights will be as follows:

$$w_k^i = w_{k-1}^i p(y_k \mid x_k^i) \quad (32)$$

Where

$$p(y_k \mid x_k^i) = \frac{1}{\sqrt{(2\pi)|R|}} \exp \{-\frac{1}{2}(y_k - \hat{y}_k)^T [R]^{-1} (y_k - \hat{y}_k)\} \quad (33)$$

Flowchart proposed algorithm is shown in Fig.5.

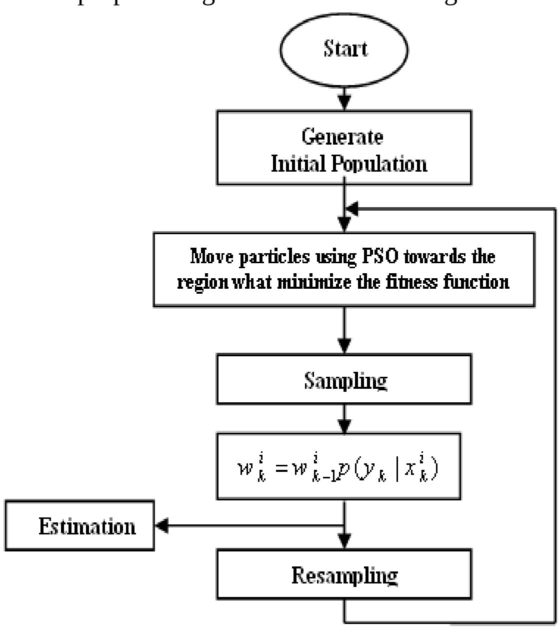

Fig.5 Modified particle algorithm

The pseudo code of Multi Swarm particle filter is as follows:

Step1. General initial population initialization

1) Initialize particle velocity
2) Initialize particle position
3) Initialize particle fitness value
4) Initialize pbest and gbest

Step2. Move particles using PSO towards the region what minimize the fitness function

Adjust the speed and location of particles

Step3. Sampling

Step4. Assign the particle a weight

$$w_k^i = w_{k-1}^i p(y_k \mid x_k^i)$$

Step5. The normalized weights

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^{N} w_k^i}$$

Step5. Resampling

The resampling is operated whenever N_{eff} is bellow a predefined threshold

Step6. Prediction

Each pose is passed through the system model

Setp7. Increase time k and return to step 2.

6. Implementation and Results

Simulation experiments have been carried out to evaluate the performance of the proposed approach in comparison with the classical method. The proposed solution for the Localization problem has been tested for the benchmark environment, with varied number and position of the landmarks. Fig.6 shows the robot trajectory and landmark location (Map of environment). The star points (*) depict the location of the landmarks that are known and stationary in the environment.

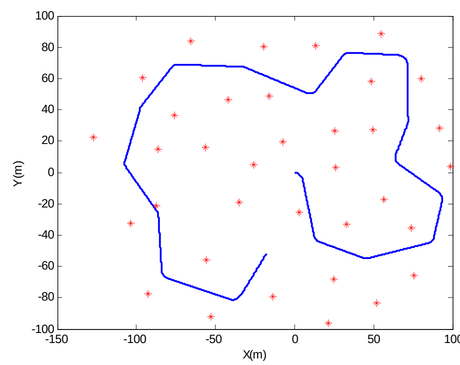

Fig.6 The experiment environment: The star point "*" denote the landmark positions (Map) and blue line is the path of robot.

The initial position of the robot is assumed to be $x_0 = 0$. The robot moves at a speed of 3m/s and with a maximum steering angle of 30 degrees. Also, the robot has 4 meters wheel base and is equipped with a range-bearing sensor with a maximum range of 20 meters and a 180 degrees frontal field-of-view. The control noise is $\sigma_v = 0.3$ m/s and $\sigma_\gamma = 3^o$. A control frequency is 40 HZ and observation scans are obtained at 5 HZ. The measurement noise is 0.2 m in range and 1^o in bearing. Data association is assumed known. The performance of the two algorithms can be compared by keeping the noises level (process noise and measurement noise) and varying the number of particles. Fig.7 to Fig.12 shows the performance of the two algorithms. The results are obtained over 50 Monte Carlo runs. As observed, localization based on multi swarm particle filter (PFPSO) is more accurate than the localization based on PF. Also, performance of the proposed method does not depend on the number of particles while the performance of localization based on PF highly depends on the number of particles. For very low numbers of particles, localization based on PF diverges while the proposed method is completely robust. This is because PSO in the proposed method places the particles in the high likelihood region. In addition, we observed that the proposed method requires fewer particles than localization based on PF in order to achieve a given level of accuracy for state estimates.

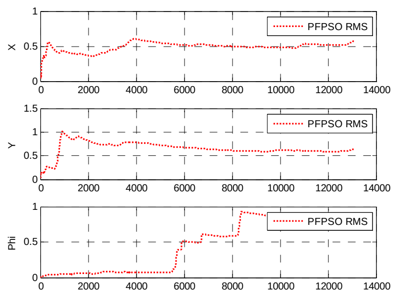

Fig.7 RMS error of localization based on PFPSO and number of particles is 5

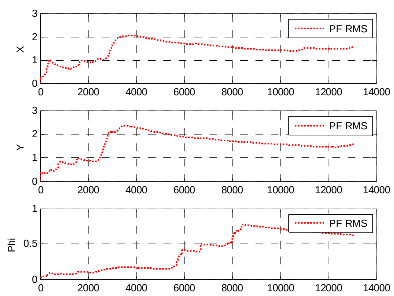

Fig.8 RMS error of localization based on PF and number of particles is 5

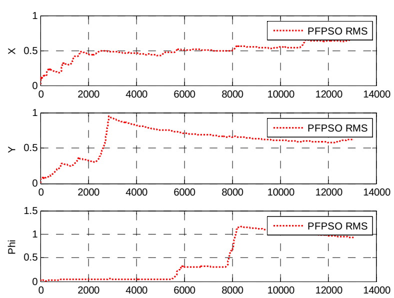

Fig.9 RMS error of localization based on PFPSO and number of particles is 10

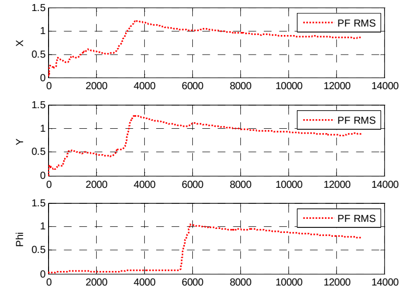

Fig.10 RMS error of localization based on PF and number of particles is 10

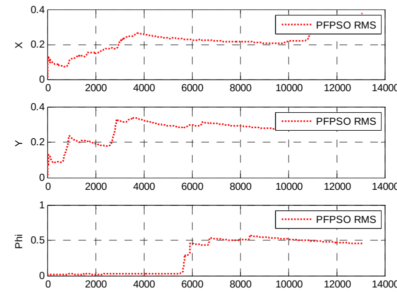

Fig.11 RMS error of localization based on PFPSO and number of particles is 20

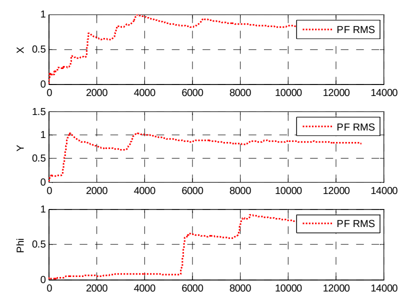

Fig.12 RMS error of localization based on PF and number of particles is 20

Conclusion

This paper proposed a new method for the accurate localization of a mobile robot. The approach is based on the use of PSO for improving the performance of the particle filter. The problem of localization based on PF is that it degenerates over time due to the loss of particle diversity. One of the main reasons for loosing particle diversity is sample impoverishment. It occurs when likelihood lies in the tail of the proposed distribution. In this case, most of participles weights are insignificant. This paper presents a modified localization based on PF by soft computing. In the proposed method, a particle filter based on particle swarm optimization is presented to overcome the impoverishment of localization based on particle filter. Finally, Experimental results confirm the effectiveness of the proposed algorithm. The main advantage of our proposed method is its more consistency than the classical method. This is because in our proposed method, when motion model is noisier than measurement, the performance of the proposed method outperforms the standard method. The simulation results show that state estimates from the multi swarm particle filter are more accurate than the Particle filter.

References

[1] F.Kong, Y.Chen, J.Xie, Gang, "Mobile robot localization based on extended kalman filter", Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21-23, 2006.

[2] J.Kim, Y.Kim, and S.Kim, "An accurate localization for mobile robot using extended kalman filter and sensor fusion", Proceeding of the 2008 International Joint Conference on Neural Networks, 2008.

[3] Tran Huu Cong, Young Joong Kim and Myo-Taeg Lim, "Hybrid Extended Kalman Filter-based Localization with a Highly Accurate Odometry Model of a Mobile Robot", International Conference on Control, Automation and Systems, 2008.

[4] Sangjoo Kwon, Kwang Woong Yang and Sangdeok Park, "An Effective Kalman filter Localization Method for mobile Robots", Proceeding of the IEEE/RSJ, International Conference on Intelligent Robots and Systems, 2006.

[5] W.Jin, X.Zhan, "A modified kalman filtering via fuzzy logic system for ARVs Localization", Proceeding of the IEEE, International Conference on Mechatronics and Automation, 2007.

[6] G.Reina, A.Vargas, KNagatani and K.Yoshida "Adaptive Kalman Filtering for GPS-based Mobile Robot Localization", in Proceedings of the IEEE, International Workshop on Safety, Security and Rescue Robotics, 2007.

[7]Y.Xia , Y.Yang," Mobile Robot Localization Method Based on Adaptive Particle Filter", C. Xiong et al. (Eds.): ICIRA 2008, Part I, LNAI 5314, pp. 963–972, Springer-Verlag Berlin Heidelberg, 2008.

[8]J.Zheng-Wei,G.Yuan-Tao, "Novel Adaptive Particle Filters In Robot Localization", Journal of Acta Automatica Sinica,Vol.31,No.6,2005.

[9]S.Thrun," Particle Filters in Robotics", In Proceedings of Uncertainty in AI (UAI), 2002.

[10]Jeong Woo, Young-Joong Kim, Jeong-on Lee and Myo-Taeg Lim," Localization of Mobile Robot using Particle Filter", ICE-ICASE International Joint Conference ,2006.

[11]D. Zhuo-hua, F. Ming,, C. Zi-xing,, YU Jin-xia, " An adaptive particle filter for mobile robot fault diagnosis", Journal of Journal of Central South University, 2006.

[12]F. Chausse, S.Baek, S.Bonnet, R.Chapuis, J.Derutin," Experimental comparison of EKF and Constraint Manifold Particle Filter for robot localization", Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems Seoul, Korea, August, 2008.

[13]J.Woo, Y.Kim, J.Lee ,M.Lim, " Localization of Mobile Robot using Particle Filter", SICE-ICASE International Joint Conference , 2006.

[14]G.Cen, N.Matsuhira, J.Hirokawa, H.Ogawa, I.Hagiwara, " Mobile Robot Global Localization Using Particle Filters", International Conference on Control, Automation and Systems, 2008.

[15]S.Thrun, D.Fox, W.Burgard, F.Dellaert," Robust Monte Carlo localization for mobile robots", Journal of Artificial Intelligence, 2001.

[16]Thrun, S., Fox, D., Burgard, W., Dellaert, F., " Robust monte carlo localization for mobile robots", Artificial Intelligence, 2001.

[17]D. Fox, "Adapting the sample size in particle filters through KLD-sampling", The International Journal of Robotics Research, 2003.

[18] D. Fox., "KLD-sampling: Adaptive particle filters and mobile robot localization", In Advances in Neural Information Processing Systems , 2001.

[19] D.Simon, " Optimal State Estimation Kalman, H_∞ and Nonlinear Approaches “, John Wiley and Sons, Inc, 2006

[20] M.Sanjeev Arulampalam, S.Maskell, N.Gordon, and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing (S1053-587X) 50(2), 174–188, 2002.

[21] Liang Xiaolong,Feng Jinfu and Li Qian Lu Taorong, Li Bingjie, " A Swarm Intelligence Optimization for Particle Filter" ,Proceedings of the 7th World Congress on Intelligent Control and Automation June 25 - 27, Chongqing, China, 2008.

[22] Guofeng Tong, Zheng Fang, Xinhe Xu," A Particle Swarm Optimized Particle Filter for Nonlinear System State Estimation", IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada ,July 16-21, 2006.

[23] Jian Zhou, Fujun Pei, Lifang Zheng and Pingyuan Cui," Nonlinear State Estimating Using Adaptive Particle Filter", Proceedings of the 7th World Congress on Intelligent Control and Automation June 25 - 27, 2008.

[24]Gongyuan Zhang, Yongmei Cheng, Feng Yang, Quan Pan, " Particle Filter Based on PSO", International Conference on Intelligent Computation Technology and Automation, 2008.

[25] R.C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", in: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43, 1995.

[26] R. A. Krohling," Gaussian swarm: a novel particle swarm optimization algorithm", In Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems (CIS), Singapore, pp.372-376, 2004.

Ramazan Havangi received the M.S. degree in Electrical Engineering from K.N.T.U University, Tehran, Iran, in 2004; He is currently working toward the Ph.D. degree in K.N.T.U University. His current research interests include Inertial Navigation, Integrated Navigation, Estimation and Filtering, Evolutionary Filtering, Simultaneous Localization and Mapping, Fuzzy, Neural Network, and Soft Computing.

Mohammad Ali Nekoui is assistant professor at Department of Control, Faculty of Electrical Engineering K.N.T.U University. His current research interests include Optimal Control Theory, Convex Optimization, Estimation and Filtering, Evolutionary Filtering, Simultaneous Localization and Mapping.

Mohammad Teshnehlab is professor at Department of Control, Faculty of Electrical Engineering, K.N.T.U University. His current research interests include Fuzzy, Neural Network, Soft Computing, Evolutionary Filtering, and Simultaneous Localization and Mapping.