

## FastSLAM Method Based on Gaussian Particle Swarm Optimization

Songzhou Wu<sup>1,a</sup>, Pengfei Li<sup>1,a</sup>, Fengshen Zhao<sup>2,b</sup>, Yuanpei Yang<sup>1,a</sup>

<sup>1</sup>School of Electrical Engineering, Nantong University, Jiangsu Nantong 226019, China;

<sup>2</sup>Nantong Shipping College, Jiangsu Nantong 226019, China

<sup>a</sup>dukouduren@163.com, <sup>b</sup>zfs@ntsc.edu.cn

**Keywords:** mobile robots; simultaneous localization and mapping; sampling particle; particle swarm optimization

**Abstract.** In simultaneous location and mapping of mobile robots, the traditional FastSLAM method has the problem of sampling particles degeneration and shortage and the problem of needing a large number of particles for positioning accuracy. To solve these problems, this paper studies a FastSLAM method based on gaussian particle swarm optimization(GPSO). The method uses the gaussian particle swarm optimization algorithm in the process of particle filter to update the sampling particles, and then increases the diversity of the sampling particle, relieves the poor sampling particle degradation problems. The other hand, the method makes the particles set distribution in the region of the true state of the high likelihood, and increases the convergence of particles as well as the accuracy of robot localization and map building, at the same time can reduce the particle filter for the required number of particles. The simulation result shows that the method is feasible and effective.

### Introduction

The emergence and development of mobile robots has brought a lot of convenience to human beings in fields of production, life, etc. A totally autonomous mobile robots will be an emerging need in the future. However, to realize it, the key technology will be the simultaneous localization and mapping(SLAM) without any doubt. SLAM was firstly proposed by Smith and Self in 1988, which refers to a mobile robot's special movement procedure, from an unknown place in a strange environment, during which the mobile robot can navigate itself through location estimation from its built sensor, based on which the incremental map was constructed, and the mobile robots' autonomous location and navigation has been realized.

Probability model will be a basic method for the wide application of the SLAM technology. Types of algorithms as follows: algorithms in framework of Kalman Filter(KF): Unscented Kalman Filter (UKF) of reference[1], Extended Kalman Filter (EKF) of reference[2], Cubature Kalman Filter of reference[3]. Algorithms in framework of Particle Filter (PF): DP-SLAM of reference[4], FastSLAM of reference[5], etc. Since the PF is free from Linear and Gaussian Hypothesis Constraints for System Models, etc., the SLAM method Based on PF is more widely applied than the SLAM method Based on UKF. Besides, FastSLAM is highly praised for its high efficiency and accuracy.

Particle Swarm Optimization (PSO) is an excellent algorithm in swarm intelligence optimization algorithm, and it has excellent searching ability. In addition, it has many similarities with particle filter and is easy to be merged. And with the addition of gaussian factor optimization, the parameter setting of particle group can be reduced. Therefore, the combination of PSO algorithm based on Gaussian optimization and FastSLAM method can improve the problem of poor degeneracy of

particle filter sample particles, and then improve the positioning accuracy of mobile robot.

**SLAM System Model.** Mobile robot SLAM system model can be shown in Figure 1[6].

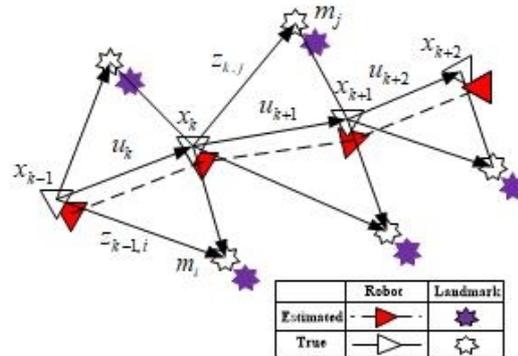


Fig. 1 Mobile robot SLAM system model

Where  $\mathbf{x}_k$  is the state vector of the mobile robot's position and state at  $k$ .  $\mathbf{m}_k$  is the state vector of the mobile robot's position and state at  $k$  in the No. $i$  landmark.  $\mathbf{u}_k$  is the controlled input vector from  $k-1$  to  $k$ .  $\mathbf{z}_k$  is the observed vector calculated from the acquiring information from the mobile robot's sensor at  $k$ .

Thus, its equation of motion and observation equation can be represented by an abstract mathematical model:

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{z}_{k,j} = h(\mathbf{m}_j, \mathbf{x}_k, \mathbf{v}_{k,j}) \end{cases} \quad (1)$$

where  $f$  represents state transfer function,  $\mathbf{w}_k$  represents noise,  $h$  represents observation function,  $\mathbf{v}_{k,j}$  represents Observed Noise.

The state space model of SLAM system based on probabilistic estimation can be represented by the following two formulas:

$$p(s_k | s_{0:k-1}, \mathbf{z}_{0:k}, \mathbf{u}_{1:k}) = p(s_k | s_{k-1}, \mathbf{u}_k) \quad (2)$$

$$p(\mathbf{z}_k | s_{0:k-1}, \mathbf{z}_{0:k-1}, \mathbf{u}_{1:k}) = p(\mathbf{z}_k | s_k) \quad (3)$$

Formulas (2) represents the probability estimation of the current system state, where the SLAM system state is represented at  $k$  by  $s_k$ , which contains a posture of the mobile robot and landmark position information at  $k$ , that is  $s_k = \{\mathbf{x}_k, \mathbf{m}_k\}$ ; Formulas (3) represents the observation information of the system.

**Analysis of the Conventional FastSLAM Algorithm.** SLAM estimation is mainly based on Bayesian filtering theory, but as the number of landmarks increases, the dimension of landmarks state vectors becomes larger and larger, and the calculation becomes more and more complicated, which makes the SLAM problem difficult to solve. The FastSLAM method proposed by Montemerlo et al. is based on particle filter[6]. The essence of this method is to decompose the state space into the independent estimation of robot trajectory and environmental landmarks by Rao-Blackwellised decomposition. Specifically, if the robot state  $\mathbf{x}_{0:k}$  and landmarks map  $\mathbf{m}_k$  are known, where  $\mathbf{x}_{0:k}$  represents the trajectory of the robot, the probability description of SLAM can be expressed as[7]:

$$p(\mathbf{x}_{0:k}, \mathbf{m}_k | \mathbf{z}_{0:k}, \mathbf{u}_{1:k}) \quad (4)$$

decomposed by Rao-Blackwellised:

$$p(\mathbf{x}_{0:k}, \mathbf{m}_k | \mathbf{z}_{0:k}, \mathbf{u}_{1:k}) = p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}, \mathbf{u}_{1:k}) p(\mathbf{m}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k}, \mathbf{u}_{1:k}) = p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}, \mathbf{u}_{1:k-1}) \prod_{k=1}^N p(\mathbf{m}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k}) \quad (5)$$

$p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}, \mathbf{u}_{1:k-1})$  is the posterior probability distribution of the trajectory of the robot's moving path;  $p(\mathbf{m}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k})$  is the posterior probability distribution of the map, thus the state probability distribution of the system is given.

The conventional FastSLAM algorithm includes four steps[8]:

(1): Sampling. Sampling is based on the proposed distribution  $q$  and the next generation particle set  $\{s_k^i\}_{i=1}^N$  is generated by the particle set  $\{s_{k-1}^i\}_{i=1}^N$ , which  $N$  represents the number of particles. It is recommended to distribute the robot's motion model:

$$q \sim p(s_k | \mathbf{u}_k, s_{k-1}) \quad (6)$$

(2) : Particle weight calculation;

(3) : Resampling which is based on the weights of the particles, and selecting the particles with larger weights;

(4) : Map updating, for each particle, the Kalman filter is used to update the environment landmarks position.

The FastSLAM algorithm uses a large number of particles to represent the probability distribution of robots on trajectories of moving paths. The particles themselves have weights, and the weights represent the probabilities of the moving robots moving to the position. With the sequential sampling, the difference between the weights of the particles will become larger and larger. Only a very small part of the particles with relatively high weights will be retained, and most of the particles will be washed out. Problems such as particle degeneration ensue. Although the optimization of the importance function and resampling as well as other methods can ease the problem, it makes the particle diversity reduce and the problem of particle depletion ensues. If the remaining particles are distributed in a lower likelihood region, that is to say deviation from the true position, the error gradually accumulates as the robot moves continuously, which eventually leads to instant positioning and composition failure[9]. The simpler solution to this problem is to increase the number of particles, but this inevitably complicates the calculation of the system.

**SLAM Model and Analysis Based on Standard Particle Swarm Optimization Algorithm.** Particle Swarm Optimization (PSO) was first proposed by Eberhart and Kennedy in 1995. Swarm Intelligence (PSO) is a kind of swarm intelligence algorithm which is inspired by the study of behavior of biological communities.

The PSO algorithm is an algorithm that requires fewer parameters to be set and has excellent optimization effects for various functions. It treats all individuals as particles in a D-dimensional space and flies in a search space at an initial velocity and at a certain initial position.

Particle velocity in each iteration update formula is given by:

$$\mathbf{v}_{id}^{k+1} = \mathbf{v}_{id}^k + c_1 r_1 (\mathbf{p}_{id} - \mathbf{z}_{id}^k) + c_2 r_2 (\mathbf{p}_{gd} - \mathbf{z}_{id}^k) \quad (7)$$

$$\mathbf{z}_{id}^{k+1} = \mathbf{z}_{id}^k + \mathbf{v}_{id}^{k+1} \quad (8)$$

$\mathbf{z}_i=(z_{i1}..z_{id}..z_{iD})$  is the D-dimensional vector of the  $i$  particle( $i=1..n$ ),  $\mathbf{v}_i=(v_{i1}..v_{id}..v_{iD})$  is the flying velocity of the particle  $i$ ,  $\mathbf{p}_i=(p_{i1}..p_{id}..p_{iD})$  is the optimal position searched for by the corresponding particle,  $\mathbf{p}_g=(p_{g1}..p_{gd}..p_{gD})$  is the optimal position searched by particle swarm,  $k$  is the number of iteration,  $r_1$  and  $r_2$  are random numbers in  $[0,1]$ ,  $c_1$  and  $c_2$  are learning factors.

Standard PSO algorithm process:

(1): Initialization. Parameter Initialization: The particle's initial position  $x_i$  and initial velocity  $v_i$  are generally randomly generated within a certain range. Each particle's  $pbest_i$  is temporarily set to its current position and the current global best position is set to  $pbest_i$ ;

(2): Evaluating all the particles. The current fitness of each particle is calculated according to the fitness function;

(3) Compare the fitness value of each particle with the fitness value of the optimal position that it passes through, and replace the optimal value with the best optimal value.

(4): Optimal solution updating;

(5): Speed and position updating;

(6): Determine end condition, output the result while the iteration is over.

According to the analysis, there are some similarities between FastSLAM algorithm and PSO algorithm[10]. First of all, the PSO algorithm finds the global optimal solution by constantly updating the velocity and position of the particle, while the particle filter approximates to the high likelihood region of the real state by continuously updating the weight value of the particle and the position of the robot. Secondly, in the PSO algorithm, the optimal solution in the search space is determined by the fitness function of the particle, whereas in the particle filter, the actual state of the system is represented by particles with relatively large weights. Finally, the particles in both algorithms have their own motion mechanism. The former updates the current local optimal solution and the global optimal solution by constantly updating the individual particle velocity and position, and then passes the current local optimal solution and the global optimal solution updates the particles in the next iteration process. The latter starts by updating the initial state and generating new particles through the motion model, and then updating the particle weight by the measurement model, and then to estimate the new state.

Above all, the idea of PSO algorithm can be integrated into particle filter to improve the performance of particle filter SLAM algorithm.

**Optimal FastSLAM Algorithm of Gaussian Particle Swarm Model.** Particle Filter of Particle Swarm Optimization SLAM Method is: With the excellent ability of global optimization of PSO, particles in particle filter can be approximated to the high likelihood region by the evolution idea of PSO, that is to say, it is possible to approach the most likely state of the robot. In this way, particle depletion and other problems can be improved in particle filter resampling process and the algorithm performance is optimized.

Since the parameters such as the maximum speed in the standard PSO algorithm are difficult to determine, Gaussian can be added to the PSO algorithm, which is called Gaussian PSO[11][12](GPSO). In this way, the particle velocity update is based on the Gaussian distribution, whose convergence is better than the standard PSO algorithm. Its velocity and position update are determined by the following formula:

$$\mathbf{v}_k^i = |\text{rand } n| (\mathbf{x}_{pbest}^i - \mathbf{x}_k^i) + |\text{Rand } n| (\mathbf{x}_{gbest} - \mathbf{x}_k^i) \quad (9)$$

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_k^i \quad (10)$$

$|rand\ n|$  and  $|Rand\ n|$  are random numbers of positive Gaussian distribution,  $\mathbf{x}_{pbest}$  represents the local optimal solution,  $\mathbf{x}_{gbest}$  represents the global optimal solution.

In this section, PSO algorithm is integrated into particle filter to optimize the sampling process. For this purpose, new measurement information is added during the sampling process and the fitness function is defined as:

$$fitness = \exp\left[-\frac{1}{2\mathbf{R}_k}(\mathbf{z}_{New} - \mathbf{z}_{Pred})^2\right] \quad (11)$$

$\mathbf{z}_{Pred}$  can be predicted by the location and has been constructed map to determine,  $\mathbf{R}_k$  represents measurement noise variance,  $\mathbf{z}_{New}$  represents the latest measured value,  $\mathbf{z}_{Pred}$  represents the predicted measurement value.  $\mathbf{z}_{Pred}$  can be determined by the predicted location and constructed map to determine:

$$\mathbf{z}_{Pred} = g(\mathbf{x}_{k+1}^i, \mathbf{m}_{k-1}) \quad (12)$$

GPSOFastSLAM algorithm steps:

(1): predicting. According to the proposed distribution, the particle set of the current moment generate from the particle set of the previous moment.

(2): GPSO optimization

a) Set robot pose particle sets  $\mathbf{s}_k^*$  as  $\mathbf{x}_{pbest}$ ;

b) update the particle set to get the optimized particle set  $\mathbf{s}_k \sim \mathbf{s}_k^*$  according to formulas (9) and (10);

c) Calculate the predicated value of landmarks and the fitness function *fitness*;

d) While  $fitness < d$ , it will get the new landmarks measurement information  $\mathbf{z}_{New}$  then update the map and return to (2);

e) While  $fitness \geq d$ , consider particle set  $\mathbf{s}_k^*$  as  $\mathbf{x}_{gbest}$ , then the PSO optimization ends.

(3) : Importance weight calculation

$$w_k^i = w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) = w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} = w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i) = w_{k-1}^i \exp\left[-\frac{1}{2\mathbf{R}}(\mathbf{z}_k - \hat{\mathbf{z}}_{k/k-1}^i)^2\right] \quad (13)$$

Weight normalization:

$$w_k^i = w_k^i / \sum_{i=1}^N w_k^i \quad (14)$$

(4) : Resampling. The original weighted sample  $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^N$  is mapped to the equal weight sample  $\{\mathbf{x}_{0:k}^i, N^{-1}\}_{i=1}^N$ .

(5): Output:

Status estimation:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^N w_k^i \mathbf{x}_k^i \quad (15)$$

Variance estimation:

$$P_k = \sum_{i=1}^N w_k^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_k)(\mathbf{x}_k^i - \hat{\mathbf{x}}_k)^T \tag{16}$$

Adding GPSO to the particle filter optimizes the particle set so that the particle set approximates the higher likelihood region of the real state before calculating the weight of the particle set, and the value of each particle is greatly increased, alleviating the problem of poor particle degradation and accelerating the convergence speed of particle sets.

**Results and Analysis of the Experiment.** In order to verify the feasibility and effectiveness of the algorithm studied in this paper, this paper verifies the algorithm and compares with the conventional FastSLAM method. The experimental computer is configured as: CPU: Intel Core i3-2330 2.2 GHz, memory: 2.00 GB, operating system: Windows10 64-bit, simulation software: Matlab2016a.

In the simulation environment, this paper uses FastSLAM2.0 algorithm and the GPSOFastSLAM algorithm studied in this paper to simulate the simulation environment and the results are shown in Figure 2. Here, the set number of particles is 100, in which the green solid line represents the robot's actual path trajectory, the blue polyline represents a pre-set path trajectory, and the green asterisk indicates a random set of environmental landmarks, which is in a total of 37, and the red points represent the constructional environmental landmarks.

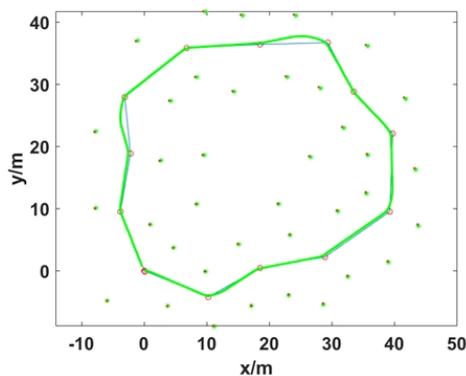


Fig 2a Conventional FastSLAM

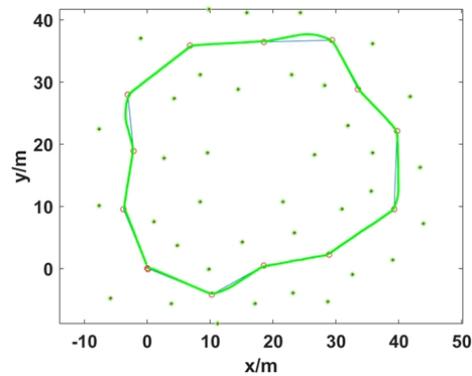


Fig 2b GPSOFastSLAM

Fig 2 Simulation effects of two methods

Figure 3 is a comparison of two algorithms for environmental landmarks estimation error. The abscissa indicates the index of the landmarks. It can be seen that the mean error of the landmarks estimation of FastSLAM is larger than that of the GPSOFastSLAM method, indicating that the latter has higher precision.

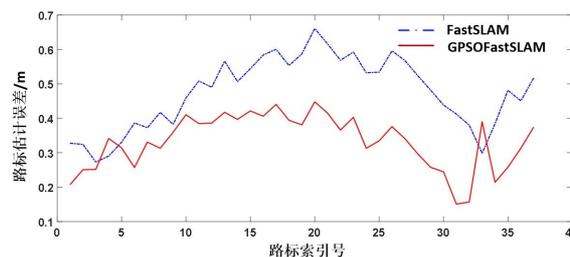


Figure 3 Landmarks estimation error comparison

Figure 4 and Figure 5 are comparison charts for the two algorithms path estimation of X-axis and Y-axis error, in which the error is based on the estimated path and the actual path. From the two maps, it can be seen that the error of FastSLAM method of path estimation is greater than the path estimation error of the GPSOFastSLAM method. By comparison, the GPSOFastSLAM method has higher accuracy than the conventional FastSLAM.

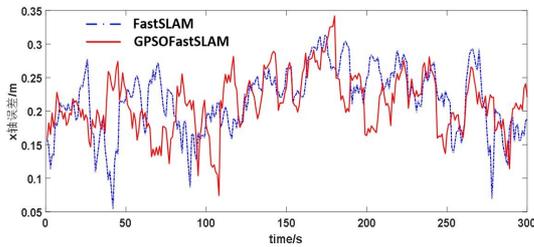


Fig 4 X-axis direction error comparison

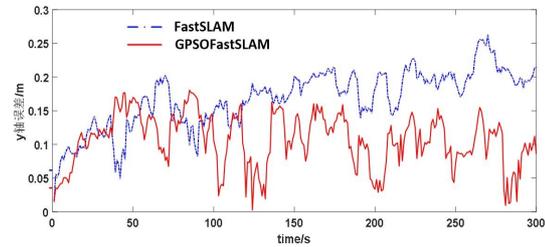


Fig 5 Y-axis direction error comparison

In addition, when the number of sampling particles are 50, 100 and 200 respectively, the simulation error of the two algorithms is compared and the results are shown in FIG. 6 and Table 1. As can be seen from Figure 6 and Table 1, the GPSOFastSLAM method is more accurate than the conventional FastSLAM method for the same number of sampling particles. And the more the number of particles, the higher the accuracy of the algorithm. At the same time, with the same accuracy, the GPSOFastSLAM method can reduce the number of sampling particles compared with the conventional FastSLAM method, thus reducing the computational complexity of the algorithm.

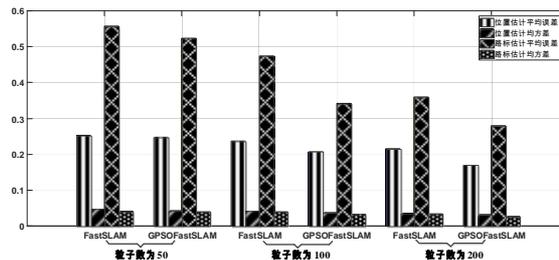


Fig 6 Error comparison of two methods in the different sampling particles

Table 1 Error comparison of two methods in the different sampling particles

Number of Particles	Method	Position Estimation Average Error / m	Mean Square Error of Location Estimation	the Average Error of Landmarks Estimation / m	Mean Square Error of Landmarks Estimation
50	FastSLAM	0.2534	0.0472	0.5567	0.0411
	GPSOFastSLAM	0.2473	0.0429	0.5229	0.0396
100	FastSLAM	0.2369	0.0425	0.4741	0.0398
	GPSOFastSLAM	0.2072	0.0376	0.3420	0.0335
200	FastSLAM	0.2158	0.0358	0.3598	0.0347
	GPSOFastSLAM	0.1702	0.0331	0.2795	0.0275

**Conclusion.** This paper studies the Particle Swarm Optimization SLAM method based on Gaussian Particle Swarm Optimization to alleviate the problem of poor degradation of the sampled particles in the conventional FastSLAM method. This method introduces the latest measurement values into the sampling distribution, and uses Gaussian Particle Swarm Optimization algorithm to optimize the sampling process. Based on a Gaussian distribution, the sampling particle is updated in speed, which not only solves uncertain problems, such as parameters of the maximum speed of the PSO algorithm itself, but also achieves the effect of moving the sampling particles to a real high

likelihood region and increases the particle diversity, thereby alleviating the problem of poor particle degradation to a certain extent and reducing the number of particles required for particle filtering. At the same time, the accuracy of state estimation and the convergence speed of particle sets are also improved. Finally, the experimental simulation shows that the algorithm which is studied in this paper can obtain higher SLAM accuracy under the same number of particles than the conventional FastSLAM algorithm.

### **Acknowledgements**

This work was supported by the Industry-University-Research Prospective Research Project in Jiangsu Province of China (BY2016053-11).

### **References**

- [1] Yong Wu, Shengxiao Guan: Improve Solution Based on Unscented Kalman Filter in the SLAM. submitted to *Computer Systems & Applications*, vol. 26(2017), p.30-36, in Chinese.
- [2] Feng Zhang, Yang Sun, Shuai Yuan, et al: EKF-SLAM algorithm for robot based on observability analysis. submitted to *Journal of Shenyang University of Technology*, vol. 38(2016), p.319-325, in Chinese.
- [3] Chunhong Wang: The SLAM Algorithm of Mobile Robots Based on Gaussian Mixture Cubature Kalman Filter. submitted to *Journal of Inner Mongolia Normal University(Natural Science Edition)*, vol. 46(2017), p.560-565, in Chinese.
- [4] A. I. Eliazar, R. Parr: DP-SLAM 2.0. *IEEE. Forum Vol. 2(2004)*, p.1314-1320.
- [5] Hanxin Chen, Zhiyu Li, Congqing Wang: FastSLAM Algorithm for SUAV Based on Particle Swarm Optimization Algorithm. submitted to *Electronics Optics & Control*, vol. 23(2016), p.24-28+54, in Chinese.
- [6] M. Montemerlo, S. Thrun: Simultaneous localization and mapping with unknown data association using FastSLAM. *IEEE. Forum Vol. 2(2003)*, p.1985-1991.
- [7] Erliang Yao, Guoliang Zhang, Wenjun Tang, et al: Rao-Blackwellized particle filter simultaneous localization and mapping algorithm based on particle swarm optimization. submitted to *Journal of Computer Applications*, vol. 34(2014), p.37-40, in Chinese.
- [8] Jiachen Ma, Qi Zhang, Wei Xie, et al: SLAM Method for Mobile Robot Based on Particle Swarm Optimization. submitted to *Transaction of Beijing Institute of Technology*, vol. (2013), p.1151-1154+1202, in Chinese.
- [9] Chen Baifan, Cai Zixing, Yuan Cheng. Mobile robot SLAM Method Based on Particle Swarm Optimization. submitted to *Robot*, vol. 06 (2009), p.513-517, in Chinese.
- [10] Fang Zheng, Tong Guofeng, Xu Xinhe. Particle swarm optimized particle filter. submitted to *Control and Decision*, vol. 03(2007), p.273-277, in Chinese.
- [11] Zhao Zengshun, Lin Yanyan, Feng Xiang, et al. Optimized Rao-Blackwellized particle filter by Gaussian particle swarm. submitted to *Application Research Of Computers*, vol. 02 (2015), p.423-426, in Chinese.

- [12]Zhang Yi, Zheng Xiaofeng, Luo Yuan, et al. SLAM algorithm with Gaussian distribution resampling Rao-Blackwellized particle filter. submitted to Control and Decision, vol. 31(2016), p. 2299-2304, in Chinese.