International Journal of Innovative
Computing, Information and Control
Volume 15, Number 5, October 2019

ICIC International ©2019 ISSN 1349-4198
pp. 1779–1792

RESEARCH ON MULTI-ROBOT COOPERATIVE LOCATION ALGORITHM BASED ON WIRELESS SENSOR NETWORKS

Yang Xu*, Hanyu Lu and Zhiming Xie

College of Big Data and Information Engineering
Guizhou University
No. 2708, South Section of Huaxi Avenue, Huaxi District, Guiyang 550025, P. R. China
*Corresponding author: xuy@gzu.edu.cn; luhanyu@163.com; 494657906@qq.com

Received August 2018; revised January 2019

Abstract. *Aiming at the shortcomings of the existing multi-robot positioning system, such as large complexity, low real-time validity, positioning error and relative observation increase, which is not proportional to the positioning accuracy, a PSO-FastSLAM (Particle Swarm Optimization – Fast Simultaneous Localization and Mapping) Multi-Robot cooperative location algorithm based on ZigBee wireless network is proposed. The algorithm introduces PSO (Particle Swarm Optimization) in the stage of FastSLAM, drives all particles to move to the high likelihood probability region, updates the particle state multiple times, reduces the influence of redundant nodes. The information fusion between nodes solves the problem of particle exhaustion and expands the scope of understanding space. The experimental results show that the PSO-FastSLAM algorithm, integrated wireless sensor for assisted navigation, and the fusion of the observation information of each robot can greatly reduce the spatial dimension of the solution, so that the observation information can be reasonably applied in the filtering stage. In this paper, the principle and design process of PSO-FastSLAM algorithm under wireless sensor networks are introduced in detail, and the effectiveness and applicability of the algorithm are proved by simulation experiments.*

Keywords: Multi-robot, PSO-FastSLAM, Wireless sensor networks

1. Introduction.

Among the many research problems of multi-robot cooperative location systems, the most basic problem is the positioning problem. The positioning ability of the robot in a complex and uncertain environment is the basis for accomplishing various advanced tasks. If a robot can detect other robots and can communicate with them, the system can improve the positioning accuracy of each robot in the group by merging the relative observations between the robots and the motion information [1-3]. Especially, when the robot cooperation system is a heterogeneous robot group, the positioning of the multi-robot cooperation system will have greater advantages.

Autonomous positioning first faces the problem of perception of the working environment, that is, the multi-robot must be able to interpret its own state data and working environment data collected by its own internal and external sensors, and extract cognitively meaningful information from it. Secondly, the data association problem, the multi-robot must be able to correlate the current perceptual information with the data and the historical cumulative perceptual information and the same part of the data, so that the robot maintains the consistency of the observation of the working environment information after the movement. The third is the positioning problem. The multi-robot must be able to accurately determine its position in the environment in real time during the movement, and not be lost in the environment. Finally, the map model creation

DOI: 10.24507/ijicic.15.05.1779

problem of the unknown environment, in the unknown environment, because there is no a priori map, the multi-robot must first traverse its working environment and complete the map creation of the working environment.

Lots of research focused on the multi-robot cooperative system methods. For example, Shao et al. [2] studied the distributed cooperative control problem for multi-robot systems with dynamic leaders in the presence of parametric uncertainties. In [4], a particle swarm localization algorithm with penalty function is designed, which not only improves the positioning accuracy, but also improves the convergence speed of positioning. [5] uses a nonlinear function with parameter estimation, which makes the positioning algorithm more intelligent and realizes the node-free positioning method without search. Extended Kalman Filter (EKF) was first used for combined perceptual positioning of mobile robots. The EKF algorithm uses the first-order Taylor expansion near the current point estimation point, linearizes the mobile robot's system model, and then performs Kalman filtering. The EKF algorithm is simple and fast, and can solve the pose tracking problem with the initial pose in the positioning well when the current local state satisfies the linearization assumption. In [6], for the complexity of the unscented Kalman filter algorithm, a positioning method based on volume Kalman filter algorithm is proposed, which improves the positioning accuracy and simplifies the complexity of the algorithm. He et al. [7] proposed PSO-UFastSLAM based on the Unscented-FastSLAM (UFastSLAM) and the particle swarm optimization to estimate the robot poses and features. Li et al. [8] discussed a new Jacobian free Neural Network (NN) based FastSLAM algorithm.

In the field of wireless networks, the research on the location problem of multi-robot cooperative system is also carried out, and the application of this technology in this field is analyzed. Network sensors have the ability to collaborate and collect a large amount of context-aware data in continuous time and continuous space to monitor and rebuild the operating environment of the robot. In a large-scale sensor network, each node must have the ability to autonomously locate in various environments, and each node can perform autonomous positioning through relative observation information. Each network node needs self-positioning and co-location to obtain a topology map of the entire network, so as to improve information transmission efficiency and information reception intensity.

The main contributions of this paper are listed as follows.

① Applying the wireless sensor network node as the specific identifier of the robot positioning, using the particle swarm optimization multi-robot cooperative location algorithm in the WSN environment, the weighted particles are used to estimate the position of the robot motion multiple iterations.

② A multi-mobile robot co-localization algorithm based on PSO-FastSLAM in ZigBee wireless network environment is proposed. This algorithm introduces PSO in the stage of FastSLAM, drives all particles to move to the high likelihood probability region, and updates the particle state multiple iterations, reducing the impact of redundant nodes, achieving information fusion between nodes, solving the problem of particle exhaustion, and expanding the scope of understanding space.

The rest of this paper is organized as follows. In Section 2, the relevant researches of particle swarm optimization algorithm are presented in brief. A multi-robot cooperative location algorithm based on wireless sensor networks is proposed in Section 3. Section 4 experimentally compares the proposed algorithm with "EKF-SLAM" by simulation environment. Finally, we conclude our algorithm in Section 5.

2. **PSO and FastSLAM.** Particle Swarm Optimization (PSO) [9] is an evolutionary computation that was developed by Dr. Eberhart and Dr. Kennedy from a behavioral study of predation of birds. The algorithm was originally inspired by the regularity of

the bird cluster activity, and then a simplified model is built using group intelligence. Based on the observation of the activity behavior of animal clusters, the particle swarm optimization algorithm uses the individual's sharing of information in the group to make the movement of the whole group in the problem solving space from the disordered to the orderly evolution process, so as to obtain the optimal solution. In PSO, the potential solution to each optimization problem is a bird in the search space, called a particle. All particles have a fitness value determined by the function being optimized, and each particle has a velocity that determines the direction and distance from which they fly. The particles then follow the current optimal particle search in the solution space. The PSO is initialized as a group of random particles (random solutions) and then finds the optimal solution. In each iteration, the particle updates itself by tracking two extremes; the first is the optimal solution found by the particle itself, and the solution is called the individual extreme value; the other extreme is the optimal solution currently found for the entire population. This extreme is the global extremum. In addition, it is also possible to use a part of the neighbors of the particles instead of the entire population, and the extreme value in all neighbors is the local extremum.

Suppose that in a D-dimensional target search space, there are N particles forming a community, where the i particle is represented as a dimension vector.

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{id}), \quad i = 1, 2, \ldots, N \tag{1}$$

The "flying" speed of the i particle is also a D-dimensional vector, denoted as

$$V_i = (v_{i1}, v_{i2}, \ldots, v_{id}), \quad i = 1, 2, \ldots, N \tag{2}$$

The optimal position that the i particle has searched so far is called the individual extreme, which is recorded as

$$p_{best} = (p_{i1}, p_{i2}, \ldots, p_{id}), \quad i = 1, 2, \ldots, N \tag{3}$$

The optimal position that the entire particle swarm has searched so far is the global extreme, which is recorded as

$$g_{best} = (p_{g1}, p_{g2}, \ldots, p_{gd})$$

When these two optimal values are found, the particles update their speed and position according to the following Formulas (4) and (5) [10]:

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \tag{4}$$

$$x_{id} = x_{id} + v_{id} \tag{5}$$

Among them: w is the inertia factor, which reflects the ability of particles to inherit previous speeds; c_1 and c_2 are learning factors, also called acceleration constants; and r_1 and r_2 are uniform random numbers in the range $[0, 1]$. The right side of Equation (4) consists of three parts. The first part is the "inertia" or "momentum" part, which reflects the movement "habit" of the particles, which means that the particles have a tendency to maintain their previous speed; the second part is the "cognition" part, which reflects the memory or remembrance of the particle's own historical experience, which represents the tendency of the particle to approach its best position in history; the third part is "social" part. The (social) part reflects the group historical experience of synergy and knowledge sharing between particles, and represents the trend of the optimal position of the particle directed group or neighborhood history.

According to experience, usually $c_1 = c_2 = 2$, $i = 1, 2, 3, \ldots, N$, v_{id} is the speed of the particle, $v_{id} \in [-v_{\max}, v_{\max}]$, $v_{\max}$ is a constant that is set by the user to limit the speed of the particles. r_1 and r_2 are random numbers between $[0, 1]$ [11,12].

2.1. **Basic particle swarm algorithm flow.** The flow of the algorithm is as follows:

① Initialize the particle swarm, including the population size N, the position of each particle X_i and the speed V_i;

② Calculate the fitness value of each particle $Fit[i]$;

③ For each particle, compare its fitness value $Fit[i]$ with the individual extreme value $p_{best}(i)$. If $Fit[i] > p_{best}(i)$, replace $p_{best}(i)$ with $Fit[i]$;

④ For each particle, compare its fitness value $Fit[i]$ with the global extreme value $g_{best}(i)$. If $Fit[i] > g_{best}(i)$, use $Fit[i]$ for $g_{best}(i)$;

⑤ According to Formulas (4) and (5) update the velocity V_i and position X_i of the particle;

⑥ Exit if the end condition is met (the error is good enough or the maximum number of cycles is reached); otherwise it returns.

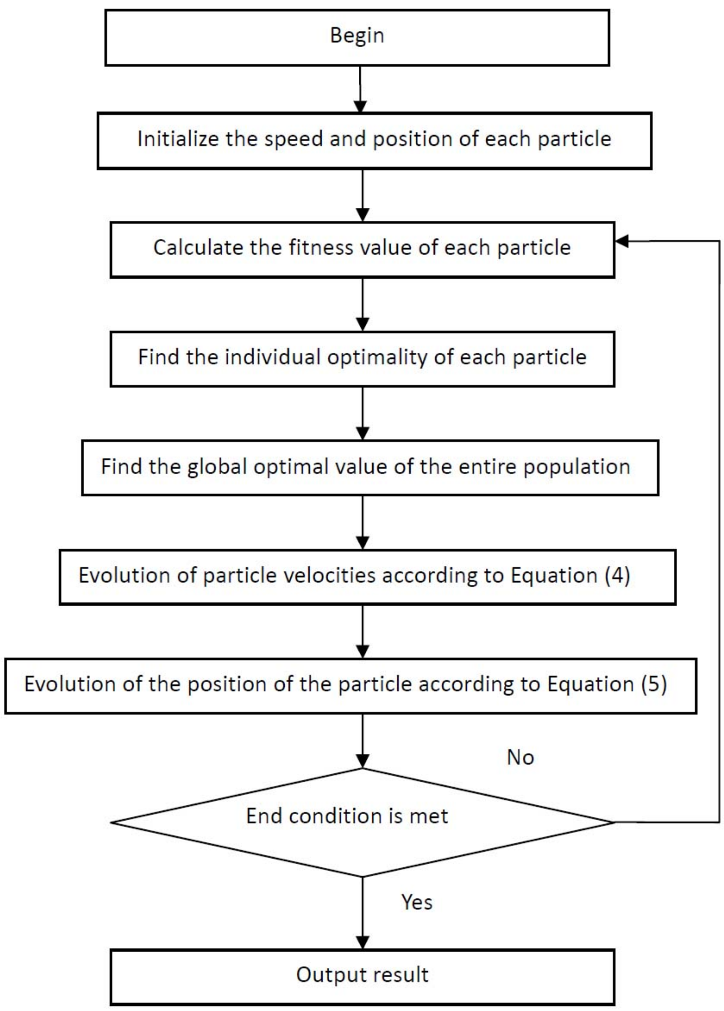

FIGURE 1. PSO algorithm flow

2.2. **Particle representation of FastSLAM.** In FastSLAM, road sign estimates are implemented using independent EKF, one road sign corresponds to an EKF filter, and the map is a set of N 2-dimensional Gaussian models instead of a single joint $2 * N$-dimensional Gaussian model. There are a total of M particle filters for estimating the path. Each particle contains N independent EKF filters for estimating N road signs. The

i particle represents:

$$s_t^i = \left\{ s^{i,[m]}, m_{1,t}^i, s_{1,t}^{[i]}, \ldots, m_{N,t}^i, s_{N,t}^{[m]} \right\} \tag{6}$$

It contains a path $s^{t,[m]}$ and based on the N landmark Gaussian estimates on the path. The mean and variance of the Gaussian representation of the Nth landmark are $\mu_{N,t}^{[m]}$ and $\sum_{N,t}^{[m]}$, respectively.

FastSLAM organically integrates the particle filter with the Kalman filter to robustly solve the data association and multi-target tracking problems. The FastSLAM algorithm decomposes the state estimation into a sampling part and a parsing part. The FastSLAM algorithm is based on particle filtering. It samples the robot pose and data fusion, calculates the position of the road sign on the basis of each particle, and uses some balanced binary trees to represent the road sign, so that updating the road sign only requires the number of road signs. The uncertainty of each particle's pose is from the data fusion, which can distinguish the road sign in the noise environment more clearly, and even update the road sign under very simple data fusion conditions. Road sign updates as shown in Figure 2.

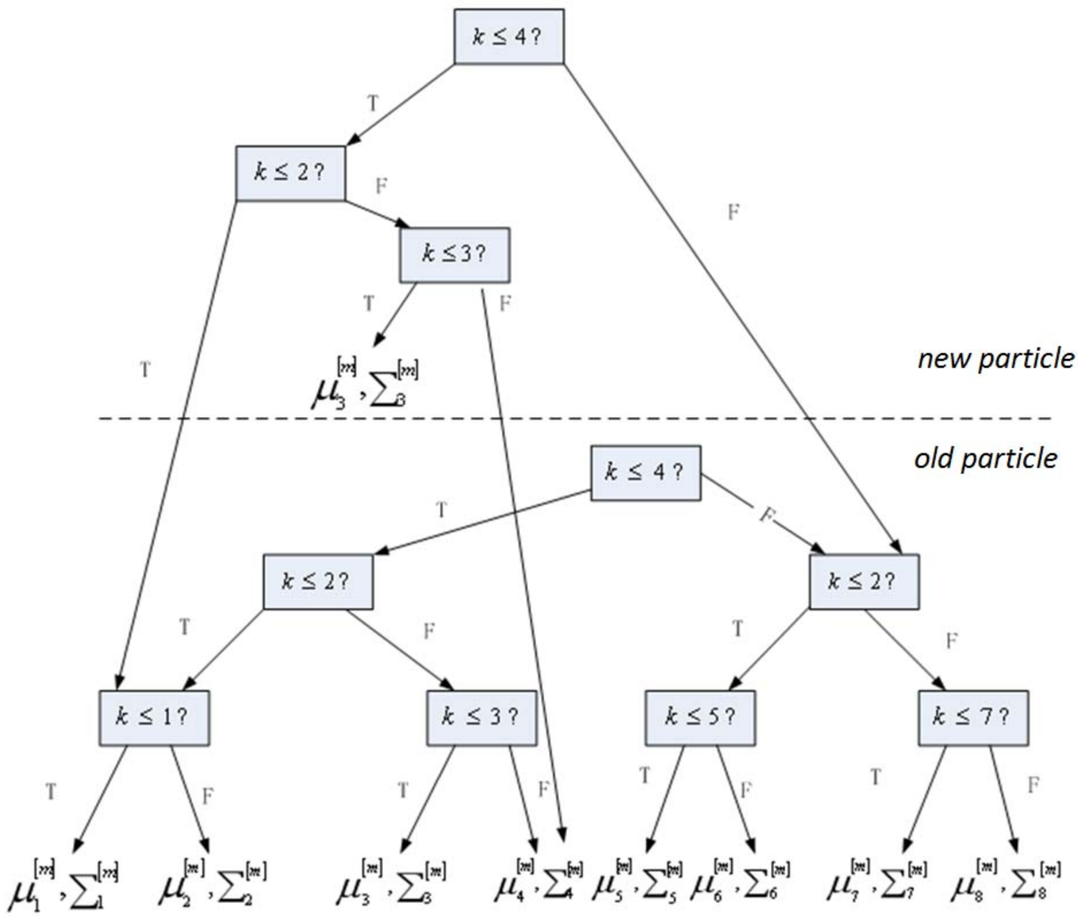

Figure 2. Updating the landmark tree

We assume that $k = 3$, that is, only the landmark Gaussian parameters $\mu_3^{[m]}$, $\sum_3^{[m]}$ are updated. Instead of duplicating the entire tree, a single path is duplicated, from the robot to the third Gaussian. This path is an incomplete tree. The tree is completed by copying the missing pointers from the tree of the generation particle. Thus, branches that leave the modified path will point to the unmodified subtrees of the generating particle.

Clearly, generating this modified tree takes time logarithmic in N. Moreover, accessing a Gaussian also takes time logarithmic in N, since the number of steps required to navigate to a leaf of the tree is equivalent to the length of the path.

3. Model Establishment.

3.1. System structure.

According to the multi-robot cooperation relationship, the robot system is mainly divided into: centralized multi-robot system, distributed multi-robot system and hybrid multi-robot master robot system. In a centralized multi-robot system, the central module is also called the master robot. The function of the central module is mainly to realize the decomposition and distribution of system tasks and to complete the exchange of information with all other robots in the system. Considering the characteristics of wireless sensor networks, each wireless sensor network has a coordinator responsible for the organization and management of the network. All data is concentrated on the coordinator, so the multi-machine system involved in this paper is centralized multi-robot system.

The FastSLAM based on multi-robot cooperative location algorithm in WSN environment is the advantage of integrating PSO and FastSLAM algorithms and real-time precise positioning combined with external wireless sensors [13]. Using wireless sensors for auxiliary positioning, data association is easy, no mis-correlation occurs during observation, and seed nodes are no longer needed for positioning during observation, which improves the robustness and accuracy of the network. In addition, particle filter and particle swarm optimization have their own advantages. Particle filter approximates posterior probability density distribution by searching a series of random samples in state space, replaces integral operation with sample mean, and finally realizes state estimation. Particle swarm optimization is an optimization algorithm based on group intelligence. It finds the optimal value by constantly updating the speed and position of particles in the search space. In order to improve the performance of FastSLAM, PSO will be introduced in the prediction stage of FastSLAM to drive all particles to move to the high likelihood probability region, and the particle state is updated several times. In the particle estimation, the individual particles and the group particles are considered together. The influence is obtained to obtain more realistic prediction of system state distribution. Combined with external wireless sensors, a multi-robot cooperative location algorithm based on PSO-FastSLAM in wireless sensor environment is proposed.

3.2. Motion model.

There are two system models in SLAM: a motion model and an observation motion model. The motion model describes the probability distribution of the robot pose at time t. The robot motion model is shown in Figure 3 [14].

In Figure 3, γ is the front wheel deflection angle of the robot, θ is the direction angle of the robot, B is interval of wheel.

The motion model of the robot is shown in Equation (7) [15]:

$$\begin{bmatrix} x_t^{(i)} \\ y_t^{(i)} \\ \theta_t^{(i)} \end{bmatrix} = \begin{bmatrix} x_{t-1}^{(i)} + \Delta t \cdot v_{t-1} \cos\left(\theta_{t-1}^{(i)} + w_{t-1}\Delta t\right) + w_x \\ y_{t-1}^{(i)} + \Delta t \cdot v_{t-1} \sin\left(\theta_{t-1}^{(i)} + w_{t-1}\Delta t\right) + w_y \\ \theta_{t-1}^{(i)} + \Delta t \cdot w_{t-1} + w_\theta \end{bmatrix} \tag{7}$$

Among them: $x_{t-1}^{(i)}$, $y_{t-1}^{(i)}$, $\theta_{t-1}^{(i)}$ are the centroid coordinates and direction angles of the robot characterized by the i particle at $t-1$; Δt is the time interval from $t-1$ to t; v and w are the speed and angular velocity of the robot respectively; w_x, w_y, and w_θ are the corresponding noise terms.

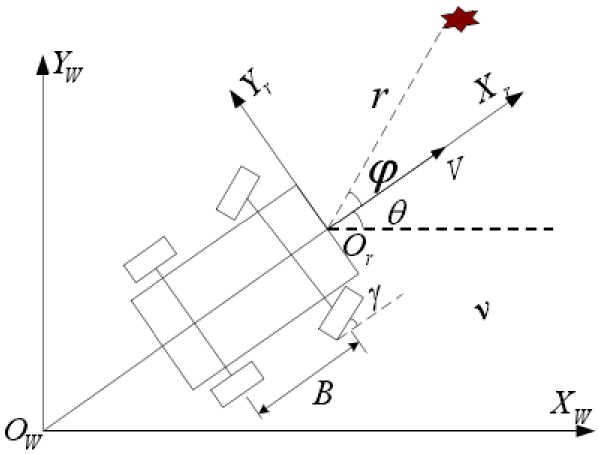

FIGURE 3. Robot motion model

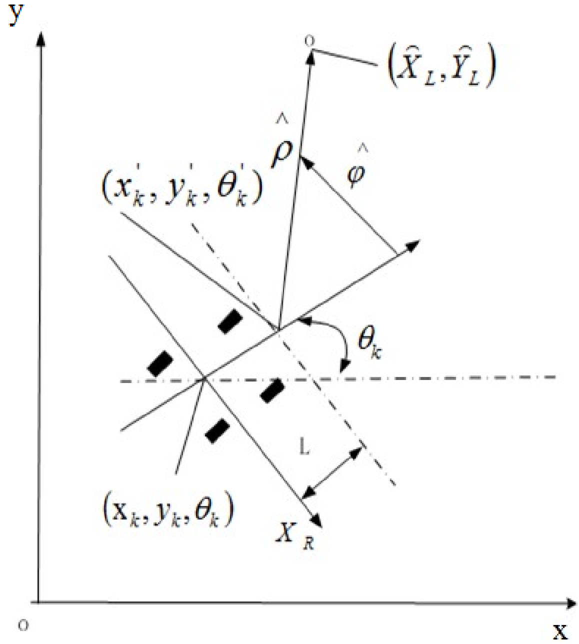

FIGURE 4. Robot observation model

3.3. **Observation model.** The observation model $p\left(z_t|x_t\right)$ describes the probability that the observed information is z_t when the robot pose is x_t. $u_{t-1} = (v_{t-1}, w_{t-1})$ is the control information given by the odometer at time $t-1$. v_{t-1} and w_{t-1} respectively represent the speed and angular velocity of the robot at time $t-1$.

The robot uses its own distance/direction sensor to detect the landmark and obtain the distance and direction angle of the landmark. Its observation model is shown in Equation (8) [16]:

$$\begin{bmatrix} z_\gamma^i \\ z_\beta^i \end{bmatrix} = \begin{bmatrix} R \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{\left(x_c - x_i\right)^2 + \left(y_c - y_i\right)^2} + w_r \\ \arctan \dfrac{(y_c - y_i)}{(x_c - x_i)} - \theta + w_\beta \end{bmatrix} \tag{8}$$

where (x_c, y_c) is the position of the landmark, (x_i, y_i) is the position of the robot; θ is the direction angle of the robot; w_r and w_β are the noise corresponding to the measured distance and direction, respectively. The schematic diagram of robot observation model is shown in Figure 4. In Figure 4, o is landmark, $\left(\hat{X}_L, \hat{Y}_L\right)$ is positional coordinates of landmark, θ_k is the direction angle of the robot, $\hat{\varphi}$ is direction angle of the landmark, L is distance between front and rear axles.

3.4. **Location algorithm for PSO-FastSLAM in WSN environment.** A set of random samples $s_i = \{X_i, w_i\}$ with correlated weights and the estimation based on these samples are used to represent the posterior probability density of robot posture. $X_i = \{x_i, y_i, \theta_i\}^T$ and w_i repiesents weights.

1) Motion model sampling is used to construct the prediction sample set and obtain the measurement value.

$$z_k \sim \mathit{fitness} = \exp\left[-\frac{1}{2R_k}\left(z_k - \hat{z}^i\frac{k}{k-1}\right)^2\right] \tag{9}$$

In the formula, z_k is the latest measurement value and $\hat{z}^i\frac{k}{k-1}$ is the predicted measurement value.

2) Initialization. At the moment $k = 0$, N is that the number of sampled particles comes from the importance function. Sampled particles are represented by $\left\{x^i_{0:k}, \frac{1}{N}\right\}^N_{i=1}$. The importance density function is taken as a transfer prior probability.

$$\left\{x^i_{0:k}, \frac{1}{N}\right\}^N_{i=1} x^i_k \sim q\left(x^i_k \middle| x^i_{k-1}, z_k\right) = p\left(x^i_k \middle| x^i_{k-1}\right) \tag{10}$$

3) Particles are extracted from the predicted sample set, and PSO is used to drive the particle set to move to the region with high likelihood probability.

4) Using the latest relative observation information to update the weight of the particle set. Observation model of multi-robot is shown in Figure 5.

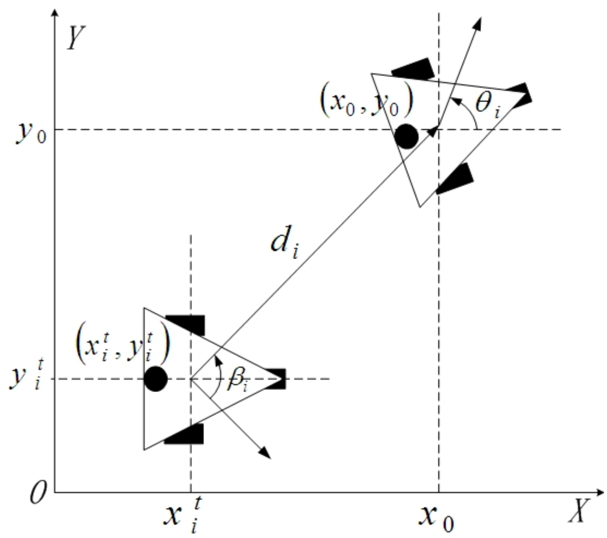

FIGURE 5. Observation model of multi-robot

The Gaussian noise with zero mean is set for the observation noise of relative distance and relative azimuth between robots. The variances are σ_p^2, σ_θ^2 respectively, and the observations are independent of each other.

$$w^t_i = w^{t-1}_i \frac{1}{\sqrt{2\pi}\sigma_\rho} e^{\frac{-(d-d_i)^2}{2\sigma_\rho^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_\theta} e^{\frac{-(\beta-\beta_i)^2}{2\sigma_\theta^2}} \tag{11}$$

Among them, d is the observation distance of t moments and d_i is the predicted distance of the robot whose X_i^t particles are observed.

$$d_i = \sqrt{\left(x_0 - x_i'\right)^2 + \left(y_0 - y_i'\right)^2} \tag{12}$$

(x_0, y_0) is the latest position estimation of the observe and (x_i^t, y_i^t) is the position estimation of particle X_i^t. β_i is the predicted orientation of particle X_i^t relative to the observed robot, and there are:

$$\beta_i = \tan^{-1}\left(\frac{y_0 - y_i'}{x_0 - x_i'}\right) - \theta_i \tag{13}$$

In this paper, for the convenience of sampling, the prior probability density $p\left(x_t|x_{t-1}^i\right)$ is used as the important sampling function, that is $q\left(x_t|x_{t-1}^i, z_t\right) = p\left(x_t|x_{t-1}^i\right)$, weights is $w_t^i \propto w_{t-1}^i p\left(z_t|x_t^i\right)$. If a robot observes a companion at time t, these observations are independent of each other and have the same variance, then the weight is w_t^i.

$$w_t^i = w_{t-1}^i \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_d} e^{\frac{-(d_j - d_i)^2}{2\sigma_d^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_\beta} e^{\frac{-\left(\beta_j - \beta_i\right)^2}{2\sigma_\beta^2}} \quad (i = 1, \ldots, N) \tag{14}$$

$d_j, \beta_j\ (j = 1, \ldots, M)$ are the relative distance and azimuth observed. According to the optimum value and using Formulas (15) and (16) to update the particle's speed and position, the particle keeps close to the real state.

$$v_{k-1}^i = |Rand\ n|\left(p_{best}(i) - x_{k-1}^i\right) + |Rand\ n|\left(g_{best}(i) - x_{k-1}^i\right) \tag{15}$$

$$x_k^i = x_{k-1}^i + v_{k-1}^i \tag{16}$$

5) Normalization of weights.

$$\widetilde{w}_t^i = \frac{w_t^i}{\sum_{i=1}^{N} w_t^i} \tag{17}$$

6) Adaptive resampling.

If $N_{\mathit{eff}} = \frac{1}{\sum_{i=1}^{N}\left(w_k^i\right)^2} < N_{th}$, the original sample $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ with weights is mapped to the sample with equal weights by resampling.

7) Output.

State estimation:

$$\hat{x}_k = \sum_{i=1}^{N} w_k^i x_k^i \tag{18}$$

Variance estimates:

$$P_k = \sum_{i=1}^{N_1} w_k^i (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \tag{19}$$

8) End judgement. If yes, quit the algorithm; if not return to the algorithm step 2).

4. Simulation Results and Analysis.

4.1. **Prerequisites of multi-robot cooperative location algorithm for PSO-Fast-SLAM in WSN environment.** In order to optimize the sampling process of particle filtering, the PSO algorithm is incorporated into particle filtering. The algorithm makes the following assumptions.

1) In the simulation experiment, the observed noise and state noise are Gaussian white noise and uniformly distributed noise, respectively.

2) All mobile robots in this experiment move in a two-dimensional barrier-free pre-selected area.

3) Each robot is equipped with a wireless sensor called ZigBee2530, which can measure the change of its own pose and can measure the relative position (relative distance and relative orientation) with other robots.

4) The corresponding information fusion will only be carried out when the robot detects that there are other networks nearby.

4.2. **Simulation and analysis.** In order to verify the feasibility and effectiveness of the algorithm, the experiment was simulated using MATLAB2016. As shown in Figure 6 and Figure 7, 35 road signs are added to the user interface, and 17 poses are added to create environmental features and motion control information on the user interface.

The position of the mobile robot is estimated using the EKF-SLAM (Extended Kalman Filter) [17-20] and WSN-FastSLAM algorithms, respectively. Figure 6 is a simulation

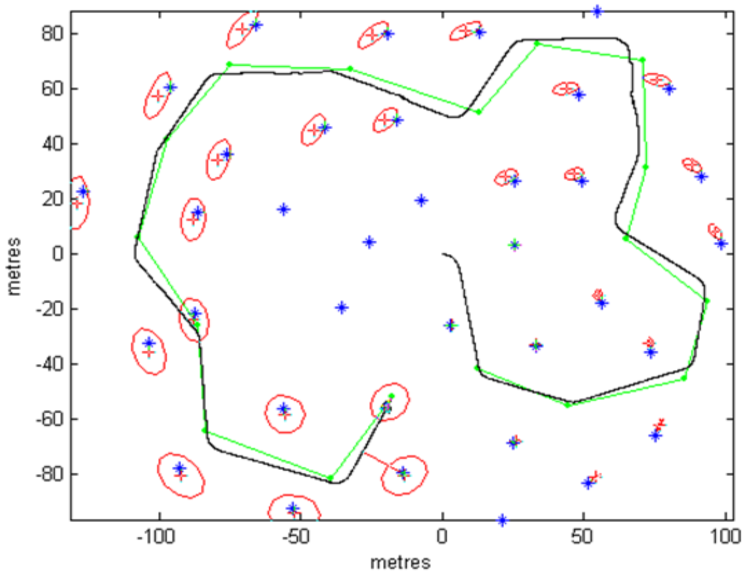

FIGURE 6. EKF-SLAM position simulation

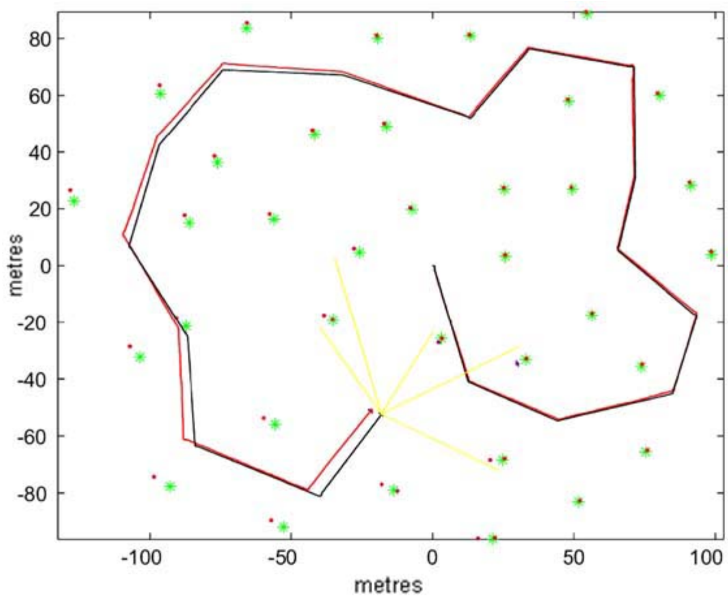

FIGURE 7. WSN-FastSLAM position simulation

effect of the EKF-SLAM algorithm, in which the star point represents the actual road sign position, the cross point represents the estimated road sign position, and the ellipse around it indicates the uncertainty of its observation. Figure 7 shows the simulation results of the WSN-FastSLAM simulation algorithm. The star points and black points represent the actual and estimated road sign positions, respectively.

When the control noise is fixed and relatively small, and the observation noise added at the same time is relatively small, EKF-SLAM and WSN-FastSLAM can better estimate the path and road sign. However, in terms of the X-axis and Y-axis errors, WSN-FastSLAM is still smaller than EKF-SLAM.

However, in the case that the control quantity noise must be relatively small, and the added observation measurement noise is relatively large, EKF-SLAM (Figure 8) and WSN-FastSLAM (Figure 9) can start to estimate the path and road sign, and get a better SLAM

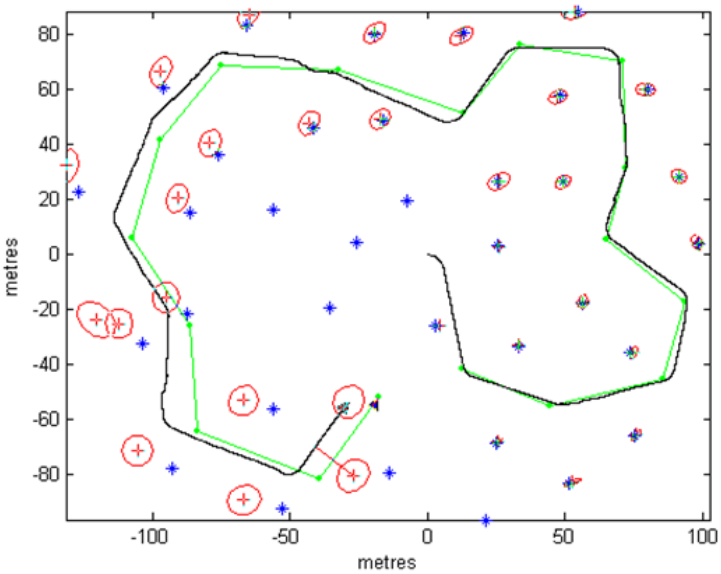

Figure 8. EKF-SLAM noise simulation

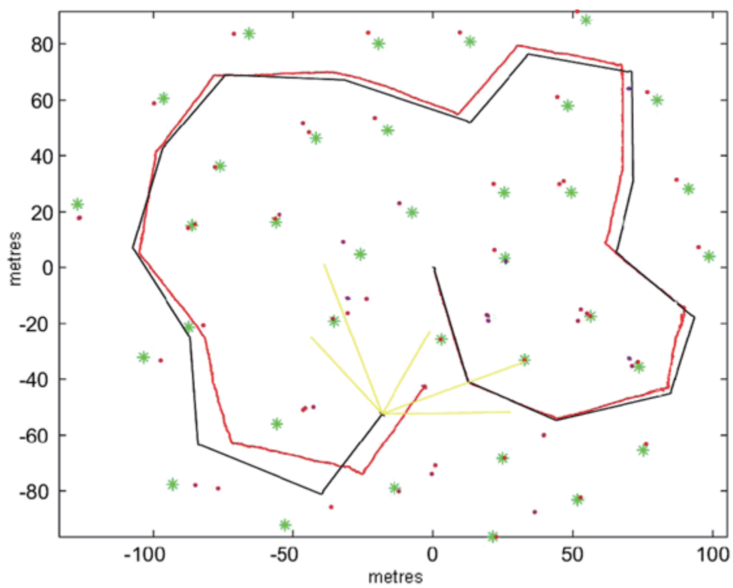

Figure 9. WSN-FastSLAM noise simulation

effect. With the accumulation of errors, EKF-SLAM is significantly larger in terms of X-axis and Y-axis errors, while WSN-FastSLAM still maintains better path and landmark estimation performance, and overall performance is better than EKF-SLAM.

In this simulation experiment, the population size is set to pop size = 30; the distribution area is the two-dimensional region of $(-2, -2)$ to $(2, 2)$, and the global optimal value is the first particle position pop of the population $(1, 1)$. As shown in the middle of Figure 10, set the number of iterations gen = 100, the learning factor $c_1 = 2$, and $c_2 = 2$. As shown in Figure 10, the right part in the figure is the state of the particle pose when the number of iterations reaches 50. The abscissa of the left graph is the number of iterations, and the ordinate is the fitness value.

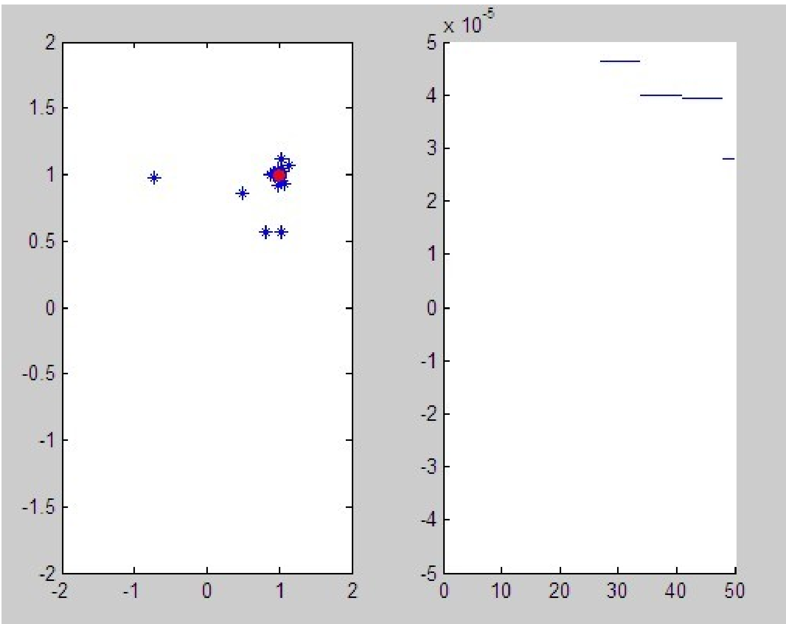

Figure 10. Iterative 50 particle distributions

When the number of iterations reaches 100, the particles each find their own optimal position and are concentrated in one position. At this time, the group optimal value is determined. As shown in Figure 11, this figure shows that the convergence of each particle is better.

In order to further verify the real-time performance and positioning accuracy of the algorithm, when the robot position is fixed, the experiment is carried out with different particle numbers and iteration times, and the algorithm running time (s) and positioning accuracy (m) are obtained as shown in Table 1. It can be seen from Table 1 that the larger the number of iterations and the more the number of particles, the more accurate the corresponding positioning accuracy is, but the corresponding time is also increased, and the timeliness of the algorithm is not enough. To meet different positioning requirements, the corresponding parameters can be adjusted appropriately.

5. **Conclusions.** In this paper, a multi-robot cooperative location algorithm based on PSO-FastSLAM in WSN environment is proposed. The wireless sensor network and particle swarm optimization algorithm are combined to achieve positioning. The wireless sensor network assists the robot positioning. The algorithm introduces PSO in the stage of FastSLAM, drives all the particles to move to the high likelihood probability region, updates the particle state multiple times, reduces the influence of redundant nodes, and

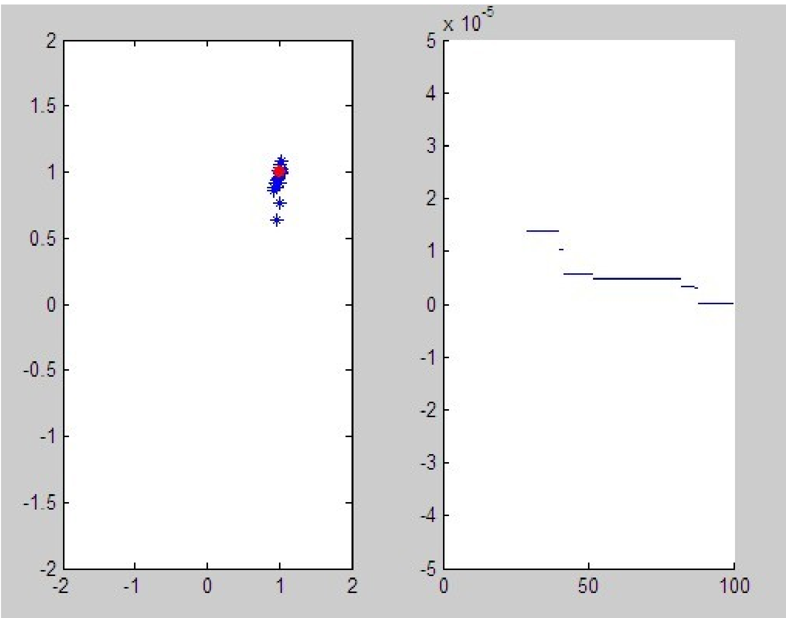

FIGURE 11. Iterative 100 particle distributions

TABLE 1. Experimental analysis on multi-robot cooperative location algorithm

Number of particles	Iteration 50 times	Iteration 60 times	Iteration 70 times	Iteration 80 times
20	0.380m (9.55s)	0.355m (10.45s)	0.349m (11.34s)	0.328m (12.38s)
30	0.364m (10.23s)	0.282m (12.65s)	0.266m (13.49s)	0.256m (14.55s)
40	0.327m (12.50s)	0.243m (14.20s)	0.220m (15.16s)	0.210m (17.48s)
50	0.337m (14.58s)	0.255m (16.15s)	0.246m (17.35s)	0.235m (19.21s)

realizes information fusion between nodes. Experimental simulations show that the proposed method effectively suppresses the cumulative error of the system and basically solves the particle degradation and depletion phenomenon in particle resampling.

As future work, one can study optimal control problem. A control policy that generates smaller loops will clearly perform better than a policy that generates large loops. In an environment with a large number of landmarks, determining the best action a robot takes in a given map is a challenging issue.

Acknowledgment. This work is partially supported by the Program for the Science and Technology Plan Project of Guizhou Province under Grant LH[2016]7429, in part by the Introducing Talent Research Projects of Guizhou University under Grant 2015-12. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Z. Mo, J. Ni, P. Shi and X. Fan, An improved tracking method based on kernelized correlation filter with a union feature, *International Journal of Innovative Computing, Information and Control*, vol.14, no.4, pp.1239-1252, 2018.

[2] N. Shao, H. Li, L. Liu and G. Li, Distributed cooperative control for multi-robot systems based on binocular visual, *ICIC Express Letters*, vol.9, no.7, pp.2051-2058, 2015.
[3] I. M. Rekleitis, G. Dudek and E. E. Milios, Multi-robot cooperative localization: A study of trade-offs between efficiency and accuracy, *International Conference on Intelligent Robots and Systems*, vol.10, no.3, pp.2690-2695, 2002.
[4] S. Cai, Z. Gao, H. Pan and Y. Shi, Localization based on particle swarm optimization with penalty function for wireless sensor network, *Journal of Computer Research and Development*, vol.49, no.6, pp.1228-1234, 2012.
[5] H. Zhou, D. Li, S. Shao et al., Free search algorithm based estimation in WSN location, *Journal of Donghua University*, vol.33, no.1, pp.52-55, 2009.
[6] Z. Zhao and W. Dai, Wireless indoor location estimation algorithm for robot based on unscented Kalman filter, *Mechanical & Electrical Technology*, vol.36, no.5, pp.2-5, 2015.
[7] B. He, L. Ying, S. Zhang, X. Feng, T. Yan, R. Nian and Y. Shen, Autonomous navigation based on unscented-FastSLAM using particle swarm optimization for autonomous underwater vehicles, *Measurement*, vol.71, pp.89-101, 2015.
[8] Q. Li, Y. Song and Z. Hou, Neural network based FastSLAM for autonomous robots in unknown environments, *Neurocomputing*, vol.165, pp.99-110, 2015.
[9] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.
[10] Y. Jiang, G. R. Liu and L. Zhou, A review on localization and mapping algorithm based on extended Kalman filtering, *International Forum on Information Technology and Applications*, Chengdu, China, pp.435-440, 2009.
[11] P. F. Alcantarilla, M. O. Sang et al., Learning visibility of landmarks for vision-based localization, *IEEE International Conference on Robotics and Automation*, Alaska, USA, pp.4881-4888, 2010.
[12] S. A. Holmes, G. Klein and D. W. Murray, An $O(N^2)$ square root unscented Kalman filter for visual simultaneous localization and mapping, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.31, no.7, pp.1251-1263, 2009.
[13] A. Frommknecht, J. Kuehnle, I. Effenberger and S. Pidan, Multi-sensor measurement system for robotic drilling, *Robotics and Computer – Integrated Manufacturing*, vol.47, pp.4-10, 2017.
[14] F. Martinelli, Robot localization: Comparable performance of EKF and UKF in some interesting indoor settings, *The 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, pp.499-504, 2008.
[15] Y. M. Xia and Y. M. Yang, A new particle filter and its application in mobile robot localization, *International Conference on Fuzzy Systems and Knowledge Discovery*, Jinan, China, pp.522-525, 2008.
[16] D.-H. Lee, Resource-based task allocation for multi-robot systems, *Robotics and Autonomous Systems*, vol.103, pp.151-161, 2018.
[17] A. Rosenfeld, N. Agmon, O. Maksimov and S. Kraus, Intelligent agent supporting human-multi-robot team collaboration, *Artificial Intelligence*, vol.252, pp.211-231, 2017.
[18] X. Chen, P. Zhang, G. Du and F. Li, Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems, *IEEE Access*, vol.6, pp.21745-21757, 2018.
[19] Y. Wu, B. Zhang, X. Yi and Y. Tang, Communication-motion planning for wireless relay-assisted multi-robot system, *IEEE Wireless Communications Letters*, vol.5, no.6, pp.568-571, 2016.
[20] Y. Kantaros and M. M. Zavlanos, Global planning for multi-robot communication networks in complex environments, *IEEE Trans. Robotics*, vol.32, no.5, pp.1045-1061, 2016.