ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

Research on Improved 2D SLAM Algorithm based on Gmapping

Xiaobo Li, Sheng Xu*, and Chengyue Su

School of Physics and Optoelectronic Engineering, Guangdong University of Technology, Guangzhou 510006, China

*Corresponding author: Sheng Xu

Abstract

In the traditional Gmapping algorithm, particle filtering is susceptible to motion noise, and particles may degrade or fall into local optima, leading to decreased mapping accuracy. This paper proposes an improved Gmapping method that integrates Extended Kalman Filter (EKF) prediction with Adaptive Particle Swarm Optimization (APSO) to enhance SLAM accuracy and robustness. First, EKF is used to fuse IMU and wheeled odometry information for pose prediction, ensuring more precise particle initialization and reducing the impact of motion noise on particle sampling. Second, APSO dynamically adjusts particle velocity and pose to improve global search capability, preventing premature convergence. Finally, APSO is introduced in the resampling process to optimize particle diversity and mitigate particle degradation. Experimental results demonstrate that the proposed algorithm achieves higher localization accuracy and mapping quality in long corridors and low-feature areas while reducing computational cost and enhancing SLAM system robustness.

Keywords

SLAM; Gmapping; EKF; APSO.

1. Introduction

With the continuous advancement of robotics technology, robots have been integrated into various industries, replacing humans in performing repetitive and hazardous tasks. Simultaneous Localization and Mapping (SLAM) [1] is a core technology in robotics and autonomous driving. SLAM can be categorized based on sensor types into visual SLAM and LiDAR-based SLAM, and based on dimensionality into 2D SLAM and 3D SLAM. This paper focuses on the study of 2D LiDAR-based SLAM algorithms. Visual sensors are highly susceptible to lighting conditions, leading to lower stability. Meanwhile, 3D LiDAR sensors are generally expensive and are primarily used in autonomous driving. In contrast, 2D LiDAR sensors are widely adopted in robotics due to their high ranging accuracy, lower susceptibility to environmental variations, and relatively low cost.

In 2002, Montemerlo et al. [2] proposed the Rao-Blackwellized Particle Filter (RBPF)-based SLAM algorithm, which decomposes SLAM into localization and mapping. It estimates the robot's pose using particle filtering and then constructs the map based on the estimated trajectory. In 2007, Grisetti et al. [3] identified the issue of particle degradation and improved the proposal distribution and selective resampling in RBPF to mitigate this problem effectively. Chatterjee et al. [4] utilized a neural-fuzzy system for offline learning of SLAM parameters. In 2011, Zhu Lei et al. [5] applied the artificial fish swarm algorithm to enhance the distribution of particles, improving SLAM robustness. In 2015, Li et al. [6] introduced a scan-matching-based method to directly sample from the observed distribution, achieving higher accuracy with fewer particles. In 2016, Zhu Qiguang [7] integrated the firefly algorithm into particle filtering to improve localization accuracy. In 2018, Vallicrosa et al. [8] proposed H-SLAM, which combined RBPF with Hilbert Maps to enable trajectory correction. In

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

2021, He Jiaze et al. [9] addressed issues of map overlapping and particle degradation in traditional RBPF-SLAM by designing a threshold-based resampling function for optimized sampling. Although these RBPF-based improvements have been extensively studied, localization accuracy remains significantly affected in degraded environments.

This paper proposes an improved Gmapping method that integrates Extended Kalman Filter (EKF) prediction with Adaptive Particle Swarm Optimization (APSO). EKF is employed to fuse IMU and wheeled odometry data for pose prediction, providing particles with more accurate initial values. Subsequently, APSO dynamically adjusts particle poses and velocities to enhance the global search capability of the algorithm, preventing it from falling into local optima.

2. Principle of the Gmapping Algorithm

2.1 RBPF-SLAM Algorithm

Gmapping is a 2D LiDAR-based SLAM algorithm based on the RBPF framework, with improvements in the proposal distribution and resampling process. The SLAM data model is typically represented as Equation 1:

$$p(x_{1:k}, m \mid u_{1:k-1}, z_{1:k}) \tag{1}$$

Here, $x_{1:k}$ represents the set of robot poses from the initial time to the current time k , m denotes the environmental map information, $u_{1:k-1}$ is the motion estimation data (such as odometry measurements),and $z_{1:k}$ represents the sensor observations (such as LiDAR range measurements). The core objective of this model is to perform a joint probabilistic estimation of the robot's pose and the map, given the motion estimation and environmental observations.

The RBPF algorithm separates robot pose estimation from map construction, using a particle filter to sample and infer poses while maintaining an independent map for each particle. Equation 2can be expressed as [10]:

$$p(x_{1:k}, m \mid u_{1:k-1}, z_{1:k}) = p(m \mid x_{1:k}, z_{1:k}) \cdot p(x_{1:k} \mid z_{1:k}, u_{1:k-1}) \tag{2}$$

Equation 2 transforms the SLAM problem into the product of two posterior probabilities.First, $p(x_{1:k} \mid z_{1:k}, u_{1:k-1})$ is used to estimate the robot's trajectory $x_{1:k}$ .Then, given the estimated trajectory $x_{1:k}$ and the observation data $z_{1:k}$,the map m can be determined.

Thus, the key focus of the problem shifts to solving the localization probability $p(x_{1:k} \mid z_{1:k}, u_{1:k-1})$ .The algorithm consists of four main steps: initial sampling, weight calculation, resampling, and map estimation.

2.1.1 Initial Sampling

When the robot moves, the new set of particle points $\{x_t^{(i)}\}$ is obtained by sampling from the previous set of particle points $\{x_{t-1}^{(i)}\}$ using the proposal distribution π .Typically, the robot's probabilistic motion model is used as the proposal distribution π .Thus, the generation process of the new particle set $\{x_t^{(i)}\}$ can be expressed as:

$$x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}, u_{t-1}) \tag{3}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

2.1.2 Weight Calculation

During the robot's motion, the particle set $x_{1:t}^{(i)}$ represents the robot's pose. The weight of each particle, $\omega_t^{(i)}$ is calculated based on the degree of matching between the proposal distribution π and the target distribution ρ.

$$w_t^{(i)} = \frac{p(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1})} \tag{4}$$

In Equation 4, the numerator represents the target distribution, and the denominator represents the proposal distribution. The importance weight reflects the difference between the proposal distribution and the target distribution.

2.1.3 Resampling

The resampling process aims to address the particle degeneration issue in particle filtering. When the environmental similarity is high or measurement noise affects the particles, particles close to the correct state may have smaller weights, while particles in incorrect states may have larger weights. Resampling is performed based on particle weights, meaning that correct particles may be discarded. Frequent resampling increases the likelihood that correct particles with smaller weights will be discarded.

2.1.4 Map Estimation

For each trajectory corresponding to a particle $x_{1:t}^{(i)}$, a map $m^{(i)}$ can be computed using $p(m \mid x_{1:k}, z_{1:k})$.Then, the maps generated from each trajectory are integrated to obtain the final map m.

2.2 Improvement of the Proposal Distribution in RBPF

The traditional RBPF algorithm's proposal distribution is primarily based on the odometry motion model. Relying solely on the odometry motion model leads to increased pose estimation errors and greater particle depletion after resampling. Therefore, Gmapping introduces LiDAR data during the sampling process. By combining sensor observations with the motion model, a more optimal proposal distribution is constructed, which allows for more accurate sampling within the possible pose space, thereby reducing the number of particles required.

The proposal distribution at this point is shown in Equation 5 [11]:

$$p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}) = \frac{p(z_t \mid x_t^{(i)}, m_{t-1}^{(i)}) \cdot p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{p(z_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1})} \tag{5}$$

2.3 Improvement of Resampling in RBPF

In Gmapping, adaptive resampling is used to balance the issues of weight degeneration and reduced diversity. The purpose of resampling is to increase the number of high-weight particles and reduce or discard low-weight particles. Resampling is determined by Equation 6 [12].

$$Neff = \frac{1}{\sum_{i=1}^{N} (\overline{w}^{(i)})^2} \tag{6}$$

The larger the $Neff$,the smaller the weight difference between particles.When $Neff$ drops below a certain threshold, it indicates that the particle distribution deviates significantly from the true

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

distribution. On the particle level, this means most particles are far from the true value, and resampling is required at this point.

3. Gmapping Algorithm Improved with EKF and APSO

3.1 EKF Fusion of IMU and Odometry

The traditional Gmapping algorithm uses an odometry-based motion model to predict the robot's state. However, in low-texture environments or situations with significant wheel slip errors, this model is prone to accumulating errors, which can affect the accuracy and stability of the SLAM estimation. To improve the accuracy of pose prediction and reduce the sampling variance in particle filtering, this paper introduces Extended Kalman Filter (EKF) [13], which fuses IMU and odometry data to construct a more accurate motion model and enhance the robustness of SLAM estimation.

Extended Kalman Filter (EKF) is a state estimation algorithm designed for nonlinear systems, with the core idea of locally linearizing nonlinear systems. EKF uses a Taylor series expansion, retaining only the first-order terms and ignoring higher-order terms, thus obtaining an approximate linear model. Based on this, it combines traditional Kalman filtering for state estimation and updates, making state estimation more accurate and reliable in real-time dynamic systems.

In this paper, EKF is used to fuse IMU and odometry data to optimize robot pose prediction, enhancing localization robustness in degraded environments, such as long corridors. The IMU consists of an accelerometer and a gyroscope, where the accelerometer measures the robot's linear acceleration along each axis in the body coordinate system, and the gyroscope measures the angular velocity relative to the world coordinate system. The odometer records wheel rotation information using encoders and estimates the robot's pose and linear velocity based on pulse counting and wheel diameter parameters.

For the EKF state vector selection, this paper uses odometry-estimated position data, IMU-measured heading and angular velocity, and odometry-measured linear velocity to achieve more accurate motion estimation, thereby improving the robustness and stability of the SLAM system in complex environments.

The state vector is:

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \\ w_k \end{bmatrix} \tag{7}$$

The state transition equation is: $x_k = f(x_{k-1}, u_k) + w_k$,where $f(*)$ is the state transition function, u_k is the control input, including linear velocity and angular velocity, and w_k is the process noise. By performing a Taylor expansion to linearize $f(*)$, the state transition equation can be linearized as:

$$X_k = \begin{bmatrix} x_{k-1} + v_{k-1}\cos(\theta_{k-1})\Delta t \\ y_{k-1} + v_{k-1}\sin(\theta_{k-1})\Delta t \\ \theta_{k-1} + w_{k-1}\Delta t \\ v_{k-1} \\ w_{k-1} \end{bmatrix} + w_k \tag{8}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

The observation equation is:

$$Z_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \\ w_k \end{bmatrix} + v_k \tag{9}$$

Thus, the equations obtained by fusing IMU and odometry using EKF are as follows:
Prediction Step:
State Prediction:

$$X_k = \begin{bmatrix} x_{k-1} + v_{k-1}\cos(\theta_{k-1})\Delta t \\ y_{k-1} + v_{k-1}\sin(\theta_{k-1})\Delta t \\ \theta_{k-1} + w_{k-1}\Delta t \\ v_{k-1} \\ w_{k-1} \end{bmatrix} + w_k \tag{10}$$

Prior Covariance Matrix:

$$P_k^- = F_k P_{k-1} F_k^T + Q \tag{11}$$

Update Step:
Kalman Gain:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \tag{12}$$

Posterior State Equation:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-)) \tag{13}$$

Posterior Covariance Matrix:

$$P_k = (I - K_k H_k) P_k^- \tag{14}$$

In the traditional Gmapping algorithm, the particle filter relies on the motion model (Odometry Motion Model) for state prediction. However, in feature-sparse degraded environments, odometry errors may cause an increase in the dispersion of particle distribution, which in turn affects the map construction quality. The above algorithm flow describes the optimization strategy based on EKF fusion of IMU and odometry data. By utilizing the heading angle and angular velocity provided by

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

the IMU, combined with odometry position information and linear velocity, a more accurate and robust pose estimation is obtained. This method fully exploits the complementary advantages of different sensors, improving the robot's localization accuracy and stability in complex environments, such as long corridors and low-texture areas.

3.2 APSO Adaptive Weight Particle Swarm Optimization

In the Gmapping algorithm, particle filtering is used to track the robot's trajectory and compute environmental information through an occupancy grid map. However, in practical applications, particle filtering faces two main issues: particle degeneration and particle depletion.

Particle Degeneration Problem: During the resampling process, low-weight particles are gradually eliminated, while high-weight particles may suffer from a lack of diversity, leading to a decline in global convergence capability. This issue is particularly significant in degraded environments such as long corridors and open spaces, where SLAM primarily relies on motion model predictions. As errors accumulate, the disparity in particle weights increases, reducing the number of high-weight particles and ultimately affecting localization accuracy.

Particle Depletion Problem: As the number of iterations increases, some particles are gradually eliminated due to decreasing weights, leading to an insufficient number of particles. This reduces Gmapping's ability to explore unknown areas, thereby affecting the quality of map construction.

To address these issues, this paper proposes a Gmapping particle optimization method based on APSO (Adaptive Particle Swarm Optimization) to enhance the robustness and global convergence capability of the particle filter.

APSO [14] is an improved particle swarm optimization algorithm that dynamically adjusts the search direction and inertia weight of particles. By balancing global and local searches during the SLAM process, APSO enhances the robustness and computational efficiency of the particle filter.

In the APSO process, the search direction of particles is not only influenced by their local optimal pose but also adjusted based on the global optimal pose. This enables particles to perform global searches even in feature-sparse environments. The formulation is as follows:

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot r_1 (p_{i,best}^t - x_i^t) + c_2 \cdot r_2 (g_{i,best}^t - x_i^t) \tag{15}$$

$$x_i^{t+1} = v_i^{t+1} + x_i^t \tag{16}$$

Where v_i^{t+1} and x_i^{t+1} represent the velocity vector and position vector of the ith particle at the (t+1)th iteration, respectively. $p_{i,best}^t$ and $g_{i,best}^t$ denote the personal best solution of the ith particle and the global best solution of the entire particle swarm, respectively. c_1 and c_2 are learning factors that control the balance between local and global searches, while r_1 and r_2 are random numbers in the range [0,1], enhancing the randomness of the search process.

In the traditional PSO [15-16] algorithm, the inertia factor ω is usually a fixed value. However, in Gmapping, the inertia factor needs to be dynamically adjusted to adapt to different environmental features. This paper employs an adaptive inertia factor, which increases the search range when particle diversity is high and reduces the search range when particles converge, thereby balancing exploration and exploitation. The calculation method is shown in Equation 17.

$$\omega_t = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})}{T} \cdot t \tag{17}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

Where ω_{max} and ω_{min} are the maximum and minimum values of the inertia factor, T is the maximum number of iterations, and t is the current iteration step.

4. Experiments and Analysis

To comprehensively compare the performance of the improved Gmapping algorithm with the original Gmapping, we designed two datasets and conducted experiments under the same hardware and software conditions to ensure fairness and comparability. Each dataset includes three types of sensor data: 40Hz single-line LiDAR data, 20Hz wheeled odometry (Odometry) data, and 100Hz nine-axis IMU data (accelerometer, gyroscope, magnetometer).

The experimental environment consists of a laptop with an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz running the Ubuntu 18.04 operating system. By comparing the performance of different algorithms under the same computational resources and sensor input conditions, we can objectively evaluate the improvements in SLAM quality, localization accuracy, and computational efficiency brought by the modified Gmapping algorithm.

Experiment 1

This experiment was conducted in an environment with simple geometric structures, a relatively open space, and a straight corridor of a certain length. Figs 1 and 2 show the 2D occupancy grid maps generated by the original Gmapping algorithm and the improved Gmapping algorithm, respectively. Table 1 presents the Absolute Pose Error (APE) data obtained by both algorithms in the first experimental environment.

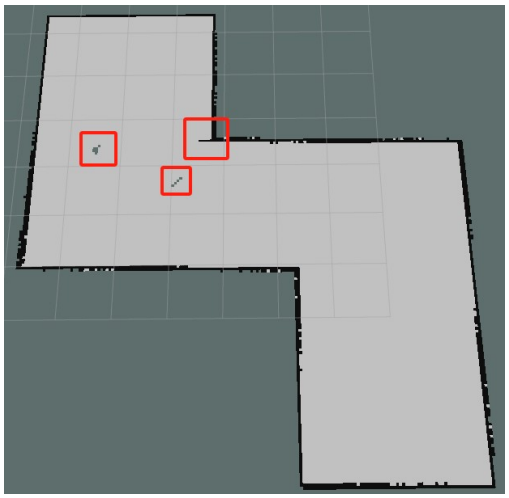

Fig. 1 Original Algorithm

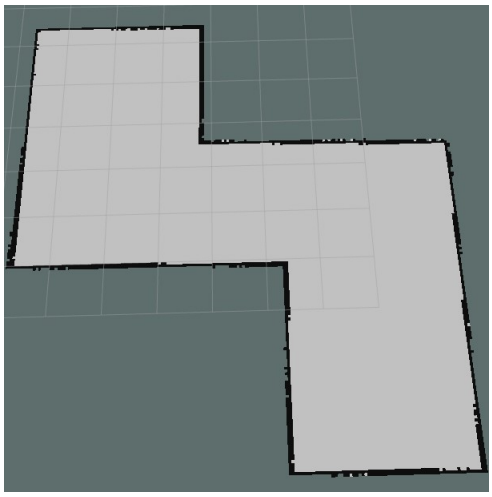

Fig. 2 Improved Algorithm

Table 1. shows the APE obtained by the two algorithms in the first environment

algroithm	max	mean	median	min	rmse	sse	std
Gmapping	0.259184	0.147923	0.147906	0.054920	0.154167	80.120559	0.043432
Improved Gmapping	0.201283	0.104108	0.128162	0.050027	0.120211	66.271100	0.020917

From the comparison between Fig. 1 and Fig. 2, it can be observed that the grid map generated by the original Gmapping algorithm contains some local detail errors. For example, in the area marked by the red box, there are discontinuities or offsets in the grid. In contrast, the map generated by the improved Gmapping algorithm in the same environment is more regular, with clearer boundaries and better overall consistency.Absolute Pose Error (APE) is an important metric for evaluating the accuracy of SLAM algorithms. It measures the deviation between the estimated trajectory and the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

true trajectory, reflecting the global consistency of the trajectory. From the data in Table 1, it is evident that the improved Gmapping algorithm outperforms the original algorithm in key metrics such as maximum error, mean error, root mean square error (RMSE), and total square error (SSE). Among these, RMSE, as the core evaluation metric of APE, effectively measures the overall accuracy of the generated trajectory and reflects the global trend of errors. The improved Gmapping algorithm has reduced the RMSE by approximately 22% compared to the original version, indicating improvements in both accuracy and robustness.

Experiment 2

This environment features a long, symmetrical layout, mainly consisting of a narrow central corridor that connects two larger open areas on either side. Due to the narrowness of the central area, the robustness of the laser SLAM algorithm in long corridor environments is highly challenged, often resulting in difficulties with loop closure detection or insufficient feature points. Figs 3 and 4 show the 2D grid maps generated by the original and improved Gmapping algorithms, respectively. Table 2 presents the absolute pose error (APE) data for both algorithms in this second environment.

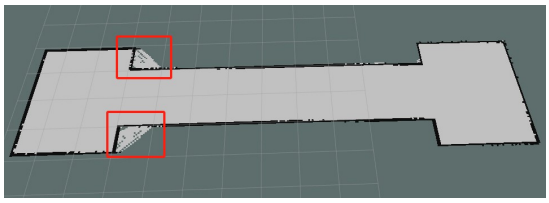

Fig. 3 Original Algorithm

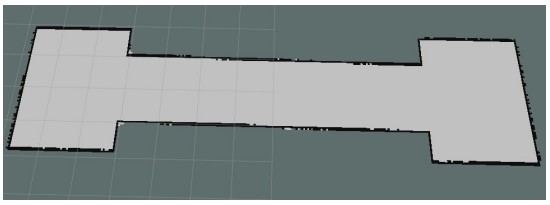

Fig. 4 Improved Algorithm

Table 2. shows the APE obtained by the two algorithms in the first environment

algroithm	max	mean	median	min	rmse	sse	std
Gmapping	1.951841	1.202220	1.254696	0.009007	1.275848	6148.1596	0.427150
Improved Gmapping	1.375165	0.649716	0.659282	0.199197	0.697859	2021.0819	0.254709

The environment in this experiment contains a long corridor with fewer features. Compared to the more structured environment in Experiment 1, the mapping and localization performance faces greater challenges. From the comparison between Figs 3 and 4, it can be observed that the grid map generated by the original Gmapping algorithm exhibits two obvious map overlap phenomena in the areas marked by the red boxes, reflecting its lack of robustness in feature-sparse environments. In contrast, the map generated by the improved Gmapping algorithm in the same environment is more regular, with clear boundaries and better overall consistency.

Additionally, based on the root mean square error (RMSE) metric, the improved Gmapping algorithm shows a reduction of about 45% in error compared to the original version, significantly improving the accuracy and robustness in degraded scenarios. This demonstrates that the improved algorithm has better adaptability to feature-sparse environments.

5. Summary

This paper proposes an improved approach to address the challenges of local errors, map overlay issues, and decreased localization accuracy in traditional Gmapping algorithms, particularly in feature-sparse environments such as long corridors. By introducing an Extended Kalman Filter (EKF) to fuse IMU and wheeled odometry data, a more precise motion model is constructed, providing particles with more accurate initial poses. Furthermore, an Adaptive Particle Swarm Optimization

ISSN: 2414-1895 DOI: 10.6919/ICJE.202505_11(5).0035

(APSO) algorithm is incorporated to dynamically adjust the pose and velocity of particles, enhancing the algorithm's global search capability and preventing it from getting trapped in local optima.

Experimental results demonstrate that the improved Gmapping algorithm generates more structured occupancy grid maps with clearer boundaries and significantly outperforms the original Gmapping algorithm in terms of Absolute Pose Error (APE). Notably, the Root Mean Square Error (RMSE) is reduced by approximately 45%. This improvement effectively enhances the adaptability of the algorithm in degraded environments. The study provides new insights for optimizing 2D LiDAR SLAM algorithms in complex environments and offers more reliable technical support for mobile robots in autonomous navigation and localization within feature-sparse scenarios.

References

[1] Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and map**: part I. IEEE robotics & automation magazine, 13(2), 99-110.

[2] Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and map** problem. Aaai/iaai, 593598, 593-598.

[3] Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid map** with rao-blackwellized particle filters. IEEE transactions on Robotics, 23(1), 34-46.

[4] Chatterjee, A., & Matsuno, F. (2006, September). Improving EKF-based solutions for SLAM problems in Mobile Robots employing Neuro-Fuzzy Supervision. In 2006 3rd International IEEE Conference Intelligent Systems (pp. 683-689). IEEE.

[5] Zhu, L., Fan, J., Zhao, J., & Wu, X. (2011). SLAM Method for Mobile Robots in Unknown Environments. Journal of Huazhong University of Science and Technology: Natural Science Edition, 39(7), 9-13.

[6] Li, T. C., Villarrubia, G., Sun, S. D., Corchado, J. M., & Bajo, J. (2015). Resampling methods for particle filtering: identical distribution, a new method, and comparable study. Frontiers of Information Technology & Electronic Engineering, 16(11), 969-984.

[7] Zhu, Q., Xiao, Y., Chen, W., Ni, C., & Chen, Y. (2016). Research on Mobile Robot Localization Method Improved by Firefly Algorithm. Journal of Instruments and Instrumentation, 37(2), 323-329.

[8] Vallicrosa Massaguer, G., & Ridao Rodríguez, P. (2018). H-SLAM: Rao-Blackwellized particle filter SLAM using Hilbert Maps.

[9] He, J., & Zhang, S. (2021). Research on 2D LiDAR Mobile Robot SLAM System. Electronic Measurement Technology, 360(4), 35-39.

[10] Grisetti, G., Tipaldi, G. D., Stachniss, C., Burgard, W., & Nardi, D. (2007). Fast and accurate SLAM with Rao–Blackwellized particle filters. Robotics and Autonomous Systems, 55(1), 30-38.

[11] Lee, J., & Bang, H. (2018). A robust terrain aided navigation using the Rao-Blackwellized particle filter trained by long short-term memory networks. Sensors, 18(9), 2886.

[12] Grisetti, G., Stachniss, C., & Burgard, W. (2005, April). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 2432-2437). IEEE.

[13] Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In Autonomous robot vehicles (pp. 167-193). New York, NY: Springer New York.

[14] Ozcan, E., & Mohan, C. K. (1999, July). Particle swarm optimization: surfing the waves. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406) (Vol. 3, pp. 1939-1944). IEEE.

[15] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). ieee.

[16] Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. Communications in nonlinear science and numerical simulation, 17(12), 4831-4845.