Simultaneous Localization and Mapping of Mobile Robot Based on Improved RBPF

Jinging Zhang[1,2], Xiaogang Ruan[1,2], Pengfei Dong[1,2] and Jing Zhou[1,2]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124 , China*
[2]*Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China*

Abstract. The traditional SLAM based on RBPF has the problem of constructing high-precision map which requires large amounts of particles to make the calculation complexity and the phenomenon of particle depletion caused by particle degradation. Aiming at these problems, an improved RBPF particle filter based on adaptive bacterial foraging optimization algorithm and adaptive resampling is proposed for mobile robot SLAM problem. Firstly, the introduction of adaptive bacterial foraging algorithm to RBPF making the distribution of particles before resampling closer to the real situation. Then use the adaptive resampling method makes the newly generated particles closer to the real movement, thereby increasing the robot position estimation accuracy and map creation accuracy. The experimental results show that this method can improve the practicability of the system, reduce the computational complexity, improve the operation speed and get more effective particles while guaranteeing the accuracy of the grid map.

1 Introduction

Robot positioning is to determine the location of the robot in the environment, and in order to build an unknown environment map and must understand the location of the robots' position.Early studies have independently studied the two models, ignoring their relevance and making the study fall into a temporary pause. In 1986, Smith and Cheesemanet proposed the SLAM problem at first [1].The mobile robot starting from an unknown location, uses the sensor to observe the environment incrementally during the movement while synchronizing the established map based on the established map. The problem has always been a hotspot and difficult point in the field of mobile robot research [2], which is considered to be the key to the real autonomy of robot navigation.

At present, the solution of SLAM problem is mainly based on the Kalman filter (KF) algorithm and the algorithm based on particle filter (PF). Kalman filter based on the various methods often require a relatively high degree of linear system, and requires a larger amount of computing and storage space. The particle filter is a basic statistical tool, the core is based on the Bayesian sampling estimation of the order of important sampling (sequential importance sampling, SIS). It becomes the more commonly used SLAM solution because it has relatively short storage space, the system of linear and high Gaussian requirements are not high, when the number of particles enough time to approximate any probability distribution and so on.

The Rao-Blackwellized Particle filters proposed by Doucet et al. are effective methods to solve SLAM problems [3]. Each particle represents a possible motion trajectory of the robot and has its own global map [4].

They correspond to the trajectories of the particles. The algorithm can well approximate the joint probability density of the robot pose and the environment map. However, when the number of particles and the size of the environment map increase, the algorithm will take up a lot of memory space, and the global map copy need to take up a lot of memory space in resampling process [5].While the resampling process ignores a large number of useful particles, resulting in reduced particle diversity. Therefore, while reducing the computational complexity and reducing the number of particles required to build an accurate map, it is necessary to ensure that the particle diversity becomes a problem that needs to be addressed. The application of the algorithm has been improved in [6-9], but there are still some shortcomings.

This paper applies the bacteria foraging optimization algorithm and system resampling to the mobile robot SLAM problem based on the existing research. Compared with the traditional method, this method introduces the bacterial foraging algorithm into RBPF and the latest robot observation information into the sampling distribution, and makes the distribution of the particles close to the real motion before resampling, and accelerates the convergence of the particles. So the number of particles used in the traditional SLAM method can be reduced while the accuracy of the state prediction is guaranteed, and the calculation amount of the

algorithm can be effectively reduced. In addition, considering that the resampling process is susceptible to sample depletion, the reuse of the system can be used to improve the particle degradation phenomenon while maintaining the diversity of particles and makes the newly generated particles closer to the real movement.

2 General description of SLAM based on RBPF

2.1 Rao-Blackwellized transform

The key idea of SLAM based on RBPF is to estimate its possible pose information $x_{1:t} = x_1, x_2, \cdots, x_t$ that is, posterior probability $p(x_{1:t}|z_{1:t}, u_{1:t-1})$, through the robot's observation information $z_{1:t} = z_1, z_2, \cdots, z_t$ and the odometer control information $u_{1:t} = u_1, u_2, \cdots, u_t$, and use it to compute the union based on the joint posterior probability distribution map m and pose $x_{1:t}$,

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) = p(m \mid x_{1:t}, z_{1:t}) \bullet p(x_{1:t} \mid z_{1:t}, u_{1:t-1}) \quad (1)$$

From the formula (1) we can see that the posterior probability $p(m|x_{1:t}, z_{1:t})$ of the map is calculated based on the known $x_{1:t}$and $z_{1:t}$. The algorithm uses the particle filter to estimate the posterior probability $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ of the robot pose, and each particle represents a possible path. For $p(m|x_{1:t}, z_{1:t})$, we can use kalman filter algorithm for analytical calculation after obtaining $p(x_{1:t}|z_{1:t}, u_{1:t-1})$.

2.2 Use RBPF for map building

The general procedure for RBPF-SLAM is as follows:

Step1.Initialization:When t = 0s, according to the robot motion model prior probability $p(x_0)$ selected N particles, recorded as $x_0^{(i)} (i = 1,2, \cdots N)$, each particle corresponding to the weight $\tilde{\omega}_0^{(i)} = 1/N$, according to the target state a priori probability$(\tilde{x}_0^{(i)}, p_0^{(i)})$.

Step2.Sampling: Generates the next generation particle set $\{x_t^{(i)}\}_{i=1,2,\cdots,N}$ from the particle set $\{x_{t-1}^{(i)}\}_{i=1,2,\cdots,N}$ according to the proposed distribution of π samples, usually using the odometer motion model$p(x_t|x_{t-1}^{(i)}, u_{t-1})$ as the proposed distribution π .

Step3.Importance weight calculation: According to the principle of importance sampling, the importance weight of each particle is calculated by:

$$w_t^{(i)} = \frac{p(x_{1:t}^{(i)}|z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(i)}|z_{1:t}, u_{1:t-1})} \propto w_{t-1}^{(i)} \cdot \frac{p(z_t|m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)}, u_{t-1})}{\pi(x_t|x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1})} \propto w_{t-1}^{(i)} p(z_t|m_{t-1}^{(i)}, x_t^{(i)}) \quad (2)$$

Step4.Resampling: resampling when the number of valid particles is less than the preset threshold value Nth. After resampling, all particles have the same weight.

Step5.Update map: update the corresponding map $p(m^{(i)}|x_{1:t}^{(i)}, z_{1:t})$ according to the pose $x_{1:t}^{(i)}$ and historical observation information $z_{1:t}$ of the particle.

It is easy to calculate when using the odometer motion model as a proposed distribution, but when the accuracy of the sensor information is significantly higher than that of the odometer movement model, only a small fraction of the particles is located in the observed high likelihood region so that they increase in the updated weight. In addition, resampling leads to particle degradation, loss of important particles, and therefore requires a large number of particles to fully cover the high observed likelihood of the region, thereby increasing the computational complexity.

3 Improved RBPF-SLAM algorithm

3.1 Brief introduction of bacteria foraging optimization algorithm(BFOA)

Passino was inspired by the foraging behavior of escherichia coli in 2002 and proposed a BFOA (bacteria foraging optimization algorithm) [10]. This is a simple and effective random global optimization algorithm, which points out a new direction for practical engineering problems that many traditional optimization methods can not satisfy [11]. BFO algorithm is not yet perfect because the proposed time later, and the country is also from the beginning of 2007 to study it[12].

The BFO algorithm implements global search through collaboration and competition among individuals. The system is initialized into a set of random solutions. Each bacterium corresponds to a random solution, which is updated by chemotaxis [13]. The replication process is used to optimize the survival process of the fittest and the local search stagnation problem is improved by migration. These three steps are nested to allow the bacteria to move toward nutrient-rich areas to achieve group intelligence [14]. The position of the bacteria after each chemotaxis is updated as follows:

$$P^i(i, j+1, k, l) = P^i(i, j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (3)$$

$P^i(i, j+1, k, l)$ is the location of the i-th trend, the k-th copy operation and the z-th migration; C (i) is the chemotaxis of the bacteria step ;$\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ is the random vector of the direction at which the bacterial i is turned over; Δ(i) is the random vector of arbitrary generation of the bacterial i, which is in the range of [-1,1].

3.2Adaptive Bacterial Foraging Optimization Algorithm

The ABFO algorithm is an improvement of the BFO algorithm. In order to achieve the survival of the fittest

and the size of the group to maintain the same purpose, the classic BFO algorithm replication operations mainly through the ability to split the foraging bacteria and poor bacteria will be eliminated. The migration operation of the improved algorithm is combined with the random migration of the flora to produce the replication operation of the better flora. The classical algorithm framework is adjusted: the copy operation of the classical BFO algorithm is integrated into the migration operation of the ABFO algorithm and the replication loop is removed. This allows the algorithm to reduce the complexity of time at the same time more conducive to achieving rapid convergence. The accuracy of the optimal value will also be improved due to the introduction of dimension adaptive learning algorithms.

3.2.1 Improved chemotaxis steps

Bacteria swimming search directly affects the convergence rate of the algorithm, which is the key in the operation of chemotaxis. The improved algorithm adds the variation law of the cosine function to the swimming step, C_{max} and C_{min} are the maximum and minimum values of the swimming step, N_s is the total number of chemotaxis, and C (i, k) is the chemotaxis step of the bacteria i at the kth swimming. Its improvement is expressed as:

$$C(i,k) = \frac{C_{max} - C_{min}}{2}\cos\left(\frac{\pi}{N_s}k\right) + \frac{C_{max} + C_{min}}{2} \quad (4)$$

where$\omega = 0,1,\cdots,N_s$.

3.2.2 Dimensional adaptive learning algorithm

In some cases, the fitness of the bacteria and the overall quality of the solution are improved, but some dimensions are far from the optimal solution. The algorithm takes longer to search and increase the optimization time. In order to avoid the above problems, we introduce the dimension adaptive learning algorithm to improve the BFO algorithm.

Suppose there are d dimensions, The position of a bacterium in high dimensional space before the start of adaptive learning is expressed by $p_{k=0} = [x_0^1, x_0^2, \cdots, x_0^d]$, $y = [y^1, y^2, \cdots, y^d]$ for each dimension of the y^i value is used to adjust x^i, that is, y is the initial value of 0 for the measurement of learning changes.$\frac{\alpha}{\beta(k)}$is the function of the number of learning k, where α is the initial maximum learning step, β(k) variable k non-linear increasing function, where $\beta(k) = k^{1,2}$; c is an adaptive learning factor [15]. The basic flow of the algorithm is as follows:

Step1.Initialization. Initialize the dimension d, the number of iterations per dimension is T and the initial learning speed $V_{k=0} = [0,0,\cdots,0]$, $p_{k=0} = [x_0^1, x_0^2, \cdots, x_0^d]$, $y_{k=0} = [y_0^1, y_0^2, \cdots, y_0^d] = [0,0,\cdots,0]$.

Step2.Set the cycle parameters. Where the dimension factor $i = 1,2,\ldots,d$, the number of adaptive learning $k = 1,2,\ldots,T$.

Step3.The ithdimension of $p_{k-1} = [x_{k-1}^1, x_{k-1}^2, \cdots, x_{k-1}^d]$ is subjected to the kth learning according to the following formula:

$$V_k^i = \frac{\alpha}{\beta(k)} \cdot rand + c \cdot V_{k-1}^i \quad (5)$$

$$y_k^i = x_{k-1}^i + V_{k-1}^i \quad (6)$$

Get the new position vector $p_k=[x_k^1, \cdots, x_k^{i-1}, y_k^I, , x_k^{i+1}, \cdots, x_k^d]$.

Step4.After the i-dimensional update, get the new vector p_k, calculate the corresponding fitness function value $f(p_k)$.If $f(p_k) \geq f(p_{k-1})$, then the fitness function value is not improved after dimension learning, which is not conducive to the optimization of the whole solution and continue to follow Step3 to learn. Let $V_k^i = V_{k-1}^i$, c = 0.5, turn Step6.

Step5.Update the location. Increase the learning factor c and continue to learn in this dimension, that is let $V_k^i = y_k^i, c = 2$.

Step6: Judgment. Update k value and turn Step3 if k <T; otherwise, update the value of i. When i = d, the dimension adaptive learning process ends and outputs the learning result $p_k = x_k^1, x_k^2, \cdots, x_k^d$.

The speed learning factor c is changed according to the previous different speed update effect. When the new speed leads to the improvement of the fitness function value, it is shown that the speed is beneficial to the improvement of the overall quality of the solution. Then increase the intensity of learning, take c> 1 (such as Step5, so c = 2). On the contrary, it means that the speed is too large should reduce the next learning speed, take c<1 (such as Step4, so c = 0.5).

3.2.3 Improved migration operations

The BFO algorithm with stochastic migration according to the fixed probability does not preserve the favorable position information searched by the algorithm in advance, which leads to the large difference between the random new position and the optimal solution, and is not conducive to the convergence of the algorithm. In this paper, the migration operation is improved and the elite bacteria are initialized as the initial point of the Tent chaotic map. The mathematical expression of Tent mapping is:

$$x_{k+1} = \begin{cases} \frac{x_k}{a} & x \in [0,a] \\ \frac{(1-x_k)}{(1-a)} & x \in (a,1] \end{cases} \quad (7)$$

The system is in chaotic state and the mean value of chaotic sequence is 0.5 When $a \in (0,1), x_0 \in [0,1]$. Newly generated bacteria can jump out of local optimum and prevent search stagnation due to randomness of chaotic models. At the same time, the local optimal point is the initial point of the chaotic motion and the domain point that produces the local optimal point after chaotic search is beneficial to the convergence of the algorithm.

ABFO algorithm migration operation integrates the classic BFO algorithm replication operation and dimension adaptive learning algorithm, the specific steps are as follows.

3.2.4 ABFO algorithm operation steps

The operation of the ABFO algorithm is as follows:

Step1. initializes the parameters. $i = j = k = 0$ .The size of bacteria is S, the step size is C, the search space dimension is d, the number of chemotaxis cycles is N_c, the number of migration is N_{ed} ,the probability of migration is P, and the maximum number of chemotaxis in one direction is N_s.The number of iterations of each dimension in the dimension adaptive learning algorithm is t, the maximum swimming step is C_{max} ,and the minimum swimming step is C_{min} .

Step2. Initializes the flora position. Randomize the location of the bacteria and calculate the initial fitness value f for each bacterium.

Step3. Sets the loop variable. Where the number of migration cycles $l = 1,2,\dots,N_{ed}$, the number of chemotaxis cycles $j = 1,2,\dots,N_c$.

Step4. Entering chemotactic cycle and carrying out chemotaxis operation.

(1) Flip: Flip according to equation (3) and calculate the fitness function value f on the new position;

(2) Swimming: according to equation (4) to calculate the swimming step if the bacteria in the new position on the fitness function value has improved.The bacteria swim until the fitness function value no longer improves or reaches a predetermined maximum chemotaxis step.

Step5. Dimensional Adaptive Learning. The adaptive fitness of the bacteria with the best fitness function value (elite bacteria) was evaluated after the completion of the first chemotaxis and the position of the elite bacteria was updated.

Step6. Entering migration cycle and carrying out migration operation. Identify elite bacteria and adapt them to dimensional learning and update accuracy. For other bacteria, the random probability is generated and compared with the fixed migration probability P_{ed} .All the number of bacteria less than P_{ed} (recorded as m) is accumulated. S bacteria are initialized by Tent chaotic mapping. Replace the m bacteria that meet the migration conditions before.

Step7.Judgment.If the cycle is finished, the current optimal solution is dimensionally adaptive to obtain a higher precision elite bacteria, and the optimal solution is updated to the higher resolution and output it; Otherwise, go to Step4 and continue to search until the end of the algorithm to meet the conditions.

3.2.5 Improved resampling algorithm

The resampling method was first proposed by Gordon in [16] in 1993, and its basic idea is based on the weight of the sample. Small particles of particle size are removed and the weight of the particles is retained, so that the performance of the particle filter can be improved to a certain extent. There are many literatures on the resampling algorithm, such as polynomial resampling [16], hierarchical resampling [17] and system resampling [18] and so on. The resampling algorithm in the traditional RBPF-SLAM resets the importance of the sequence, and as the resampling continues to reduce the diversity of the particles, the filter diverges [19].So the key to improving filter performance is when and how to resample. The normalized weights are expressed by $\tilde{w}^{(i)}$.The number of effective particles N_{eff} is introduced to assess whether the current number of particles can present a true posteriori probability, defined as:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\tilde{w}^{(i)})^2} \tag{8}$$

Perform resampling when the number of valid particles is below the pre-set threshold N_{th} .

The traditional resampling method of particle degradation is almost powerless because it is a random way. The weights of the particles will be replicated many times, and the low-weight particles will be removed. So that the path and map information represented by these low-weight particles will be deleted, resulting in some knowledge of past information. On the other hand, the weight particles are replicated many times, and many of the offspring particles inherit the same path and map estimates, which greatly reduces the particle diversity. Which may lead to "particle depletion" problem, so that the algorithm can only meet the consistency requirements in a short time.

The migration operation of ABFO algorithm integrates the copy operation and dimension adaptive learning algorithm of classical BFO algorithm. A new resampling method is proposed based on the idea of migration in the ABFO algorithm, that is, the particles in the RBPF are simulated in the ABFO algorithm. The particles in RBPF were simulated in ABFO algorithm, and the specific steps were as follows:

Step1. Selecting the current optimal particle as the elite particle and the accuracy of the particle is improved by adaptively learning it.

Step2. The probability of migration is generated for each non-elite particle and is compared with the fixed migration probability. If the probability of migration is less than the fixed migration probability, the particle is eliminated and the number of bacteria less than the fixed migration probability is accumulated.

Step3. The chaotic maps are generated by using the elite particles as the initial solutions of the Tent mappings, resulting in S feasible solutions and extracting the m-bacteria from the S feasible solutions to replace the depleted particles.

3.3 Improved RBPF-SLAM algorithm

The improved RBPF-SLAM algorithm steps are as follows:

Step1. Particle initialization. Generating the pose vector $\left\{x_0^{(i)}\right\}$ of the initial particle at k=0 based on the environmental information, and the average distribution

of the particle weights $w_0^i = 1/N, i = 1,2,\cdots,N$. N is the number of particles

Step2.First step prediction of pose. Based on the motion model of the robot, the initial pose $x_t^{(i)}$ of the robot is estimated according to the pose $x_{t-1}^{(i)}$ and the odometer control information u_{t-1}. And the estimated pose $x_{t|t-1}^{(i)}$ of each particle at time t is obtained.

Step3.Perform a scan match based on the map $m_{t-1}^{(i)}$.First get the real observation at time t through the sensor;Then we use the nearest neighbor method to correlate the data, and use the formula $w_t^i = w_{t-1}^i p\left(z_t^i | x_{t|t-1}^{(i)}\right)$ to update the particle weights.Down the weight of the normalization of the use of the formula:

$$w_t^i = w_{t-1}^i / \sum_{i=1}^{N} w_t^i \tag{9}$$

Step4.Second step prediction of the pose of the robot. Based on ABFO algorithm: the particle set $\left\{x_{t|t-1}^{(i)}\right\}$ is taken as the initial bacterial population using the fitness function of formula (1).The individuals in the flora were chemically transformed, replicated and migrated respectively, and then the second updated particle set $\left\{x_t^{(i)}\right\}$ was obtained.

Step5.The map update is based on the posture information of the updated particle set and the weights are updated and normalized according to the formula in Step 3;

Step6.Calculating the number of effective particles N_{eff} and settting a threshold N_{th}. The particles are resampled using the improved algorithm if $N_{eff} < N_{th}$.

Step7.Using the weighting formula to calculate m^i according to the robot pose x_t^i and the observation information z_t ,and update the map. Go to Step2 if you still need to calculate the next moment of the posture, otherwise the end.

4 Experimental results and analysis

Simulation of the laboratory environment to build the initial environment map, where the white part that can pass the part of the black part of the non-accessible part (obstacle part) as shown in Figure 1:

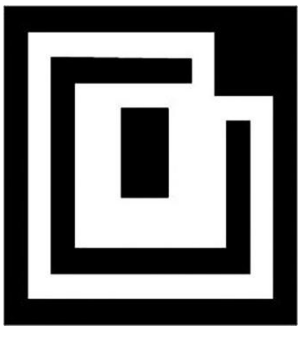

Figure1. Initial environment map.

In the case of the same number of particles, the two algorithms are applied as follows:

By contrast, it is found that the RBPF-SLAM algorithm has a serious inconsistency when the number of particles is the same in the same environment and the accuracy of the improved algorithm is obviously higher than that of the traditional RBPF-SLAM algorithm.

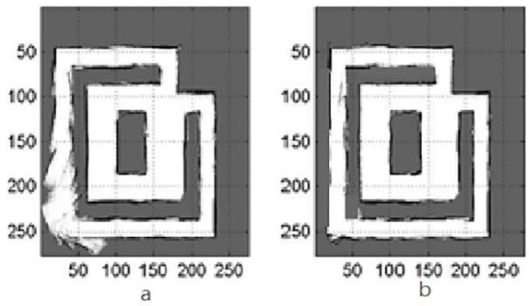

Figure 2. Construction of maps with the same number of particles(a.Grid map for traditional algorithms;b.Grid map for improved algorithms).

In order to verify the accuracy of the algorithm, the simulation experiment is carried out in MATLAB environment. The algorithm is analyzed by comparing the matching degree of the traditional RBPF-SLAM algorithm with the improved RBPF-SLAM algorithm to the particle trajectory. The results are shown in Figure3.

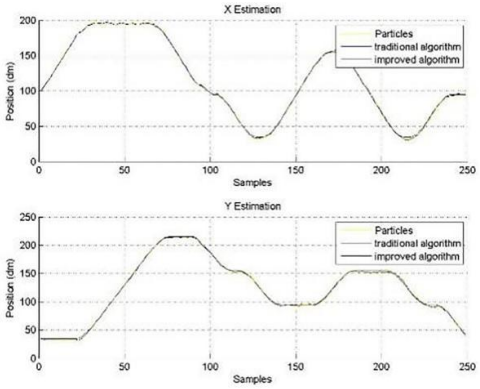

Figure 3. The two Algorithms Track the Particle in the x, y Coordinate System.

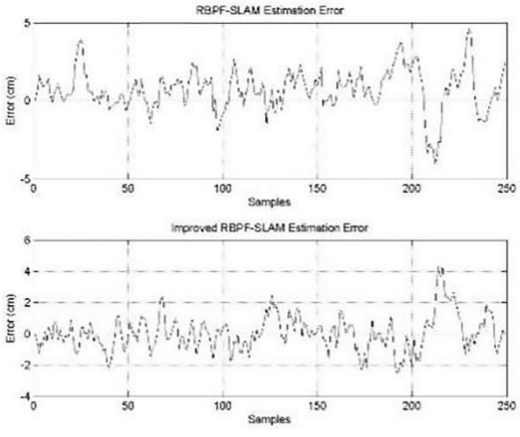

Figure 4. Estimationerror.

The estimation error of the improved RBPF-SLAM algorithm is obviously improved. The estimated error in the composition process is shown in the following Figure4:

In order to investigate the improvement of the phenomenon of particle resynchronization by the improved resampling method, the traditional resampling method and the method studied in this paper are run 9 times respectively, and the change of the mean value of the effective particles of the two algorithms is calculated and the simulation.

In order to investigate the improvement of the particle degradation phenomenon by the resampling method based on the adaptive bacterial foraging algorithm migration, the particles diversity of the two methods are represented by the traditional resampling method and the average Hamming distance of the particles produced by the improved method studied in this paper.The simulation results are shown in Table 1. The resampling method proposed in this paper can greatly improve the diversity of particles, and the average value is increased from 0.348 to 0.992.

Table 1. Diversity comparison results.

	1	2	3	4	5	6
Tradition	0.39	0.30	0.32	0.35	0.39	0.34
Improve	1.03	0.98	0.99	0.95	0.98	1.02

5Conclusion

In this paper, an adaptive bacterial foraging algorithm is proposed to introduce RBPF-based mobile robot synchronization location and map construction method.Firstly, the adaptive bacterial foraging algorithm is introduced into RBPF, which makes the particles close to the real motion before resampling.Then the improved migration operation in the ABFO algorithm is applied to the particle resampling part, which improves the diversity of the particles and increases the number of effective particles.The experimental results show that the improved RBPF algorithm and the traditional RBPF algorithm are used in the SLAM to improve the tracking accuracy and improve the accuracy of the estimation error, and more effectively improve the particle degradation phenomenon.In future work, the map building algorithm will be further optimized to achieve a precise map of a slightly larger environment.

References

1. Smith R, Self M, Cheeseman P. Estimating uncertainspatial relationships in robotics[M]//COX I J, WILFONG G T.*Autonomous Robot Vehicles*. New York: Springer, 167-193 (1990)
2. Dissanayake G, Huang S, Wang Z,et al. A review of recent developments in simultaneous localization and mapping[C].6th International Conference on Industrial and Information Systems(ICIIS), Kandy, Sri Lanka,477-482 (2011)
3. Murphy K P.Bayesian map learning in dynamic environments [C] //Proc of Neural Info Proc Systems (NIPS). Denver: MIT Press ,1015-1021 (2000)
4. Doucet A, De Freitas N, Murphy K, et al. Rao-Blackwellised particle filtering for dynamic Bayesian networks[C]//Proceedingsof the Sixteenth Conference on Uncertainty in Artificial Intelligence San Francisco，USA,176-183 (2000)
5. LipingQu,Hongjian Wang. An overview of robot SLAM problem[C]// International Conference on ConsumerElectronics,Communicationsand Net-works (CECNet),Xian-ning,China,1953-1956 (2011)
6. Grisetti G，Stachniss C，Burgardw W. Improved techniques for grid mapping with Rao-Blackwellized particle filters[J]．Robotics,**23**: 34-46 (2007)
7. Dong J F, Wijesoma W S, Shacklock A P. An efficient rao-blackwellized genetic algorithmic filter for SLAM [C]//Proceedings of 2007 IEEE International Conference on Robotics and Automation,April10-14,2007,Roma,Italy.Piscataway:IEEEPress,2427-2432 (2007)
8. Won D, Chun S, Sung S, et al. INS/vSLAM system using distributed particle filter [J]. International Journal of Control, Automation, and Systems,1232-1240 (2010)
9. Mengyin Fu, Hao Zhu, Yi Yang, Meiling Wang, Zhihong Deng. A navigation map building algorithm using refined RBPF-SLAM. 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), 2483 – 2487 (2016)
10. Jin Li. Research and Application of Hybrid Frog Leap Algorithm in Small Habitats[D].Xian: Journal of Xidian University, (2012)
11. Passino K M. Biomimicryof Bacterial Foraging for Distributed Optimization and Control [J]. Control System Magazine, 22(3):52-67 (2002)
12. Xiaolong Liu. Improvement and Application of Bacteria Feeding Optimization Algorithm [D]. Guangzhou: South China University of Technology, (2011)
13. Nandita S, Amitava C, SugataM.Fuzzy VQ based imagecompression by bacterial foraging optimization algorithm with varying population. Proceedings of the 2015 Third International Conference on Computer, Communication,Control and Information Technology (C3IT),1-6 (2015)
14. Li M S,Ji T Y,Tang WJ, et a1. Bacterial Foraging Algorithmwith Varying Population[J]. BioSystems, 100(3):185-197 (2010)
15. Gordon N J, Salmond D J, Smith A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation [J].IEE Proceedings F Reader and Signal Processing,140(2):107-113 (1993)
16. FengC, Wang M, Qing-Bo J I. Analysis and comparison of resampling algorithms in particle filter [J]. Journal of System Simulation ,21(4):1101-1105 (2009)

17. Jinxai Yu, Cai Z X, Duan Z H. Survey on some key technologies of mobile robot localization based on particle filter [J]. Application Research of Computers,24(11):9-14 (2007)
18. MerweR V D, Doucet, Freitas N D, et al. The unscented particle filter [J]. Adavances in Neural Information Processing Systems, 13:584-590 (2001)
19. Havangi R, Taghirad H D, Nekoui M A, et al. Asquare root unscented FastSLAM with improved proposal distribution and resampling [J]. IEEE Transactions on Industrial Electonics,61(5):2334-2345 (2014).