2018 International Conference on Electronics Technology

Mobile Robot SLAM Algorithm for Transformer Internal Detection and Location

Xun Li, Ronghui Huang
Shenzhen Power Supply Co., Ltd.
Shenzhen, China
epcman@vip.qq.com
e-mail: 13631561618@139.com

Yuming Zhao
Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang, China
e-mail: zhaoyuming@sia.cn

Xin Zhang
School of Mechanical Engineering, Shenyang University of Technology
Shenyang, China
e-mail: 478483407@qq.com

Tianze Guo
Beijing University of Posts and Telecommunications
Beijing, China
e-mail: 2015212974@bupt.edu.cn

Abstract—**Power transformer is one of the most important equipments for the power distribution system, and it is of crucial significance to guard electricity safety. A small wireless controlled robot was developed for the internal detection of oil-immersed transformer, which improves the automation and intelligent level of oil-immersed transformer fault detection and maintenance. Conventional FastSLAM is known to degenerate over time in terms of accuracy due to the particle depletion in resampling phase. In order to achieve a precise and efficient SLAM for mobile robots in high similarity environments, a Particle Swarm Optimization (PSO) SLAM algorithm is applied to the robot. Simulations verify its effectiveness and feasibility.**

Keywords-simultaneous localization and mapping (SLAM); Particle Swarm Optimization (PSO); mobile robot; transformer internal fault detection

I. INTRODUCTION

Large scale oil-immersed transformers are one of the most valuable and most important equipment in power systems. The reliability and operating life of transformers have direct impact on power grid operating safety and asset performance.

Currently transformer internal detection needs personnel access inside the transformer, which means the transformer oil has to be completely emptied. The maintenance work is heavy and easy to cause pollution. Sometimes the protective coil or insulating sleeve will be damaged, even the life of detection workers is at risk. A robot for oil-immersed transformer internal fault detection was developed to improve the automation and intelligence of transformer maintenance, utilized technologies such as intelligent robots, multi-sensors, motion control, and wireless transmission.

SLAM is the key technology for autonomous navigation of mobile robots. The information mobile robot extracts of specific target is used to construct the environmental map, which provides the position of the characteristic object by observing models, and stores the position of specific target and poses of robot in a variable state vector[1]. With the movement of robot, the sensor extract the environmental information constantly, and updating the robot's own posture state through the movement model[2].

In recent years, many algorithms have been proposed to overcome the particle depletion problem. Kwak et al.[3] analyzed several particle filters used in FastSLAM and proposed a compensation technique to deal with the depletion problem. Recently, Zhang et al.[4] proposed PSO-PF which combines generic particle filtering algorithm with standard PSO. In their work, after resampling in the generic particle filter, PSO is preformed using the weight of a particle as a fitness value. The most weighted particle, therefore, is used as a global attractor in every iteration step. Applying PSO in the particle filter allows the particles to maintain diversity and makes them to move toward the most weighted particle. Kim et al.[5] proposed unscented FastSLAM which unscented transform is used for particle filter, feature initialization, and feature estimator.

But the particle impoverishment is another major issues that need to be addressed prior to resample[6-8]. The particle impoverishment arises when particles converge to the wrong solution[9]. In that case, as particles drift around according to the motion of robot, there may be a lot of particles of improbable solutions, and may only one or a few particles exist within the area where probability of obtaining the correct solution is high[10]. An optimized algorithm for SLAM problem called PSO-FastSLAM proposed by Lee and Park[11] was applied in this paper, to solve the above problem.

An improved FastSLAM framework, named PSO-FastSLAM, using PSO as the particle cooperation technique for estimating the robot position with FastSLAM. First, as a target for PSO, the most probable location of a robot is obtained using the positions of particles. Then as the particle cooperation method, PSO is used for particles to be gathered around the target. As a result, the number of particles estimating the robot position becomes larger accurately, which means the degeneracy of FastSLAM can be alleviated.

978-1-5386-5752-2/18/$31.00 ©2018 IEEE

This paper is organized as follows: Section II introduced the robot designed for the transformer, and described the framework of conventional FastSLAM and the particle filter problem. In Section III, PSO-FastSLAM was stated with a brief explanation of PSO. Section IV showed the evaluation of the proposed framework using computer simulations. Finally, conclusions were gave in Section V.

II. RELATED WORK

A. Robot Overall Scheme

According to the structure inside the transformer, the robot structure should have the characteristics of small size, light weight and high stability. The shape of the robot is designed as spherical, for spherical robot has the advantages of flexible movement, high movement efficiency and strong environmental adaptability. The robot shape is a sphere with diameter less than 200mm, which can travel through narrow space inside the transformers. The overall mechanical structure[12] is shown in Figure 1.

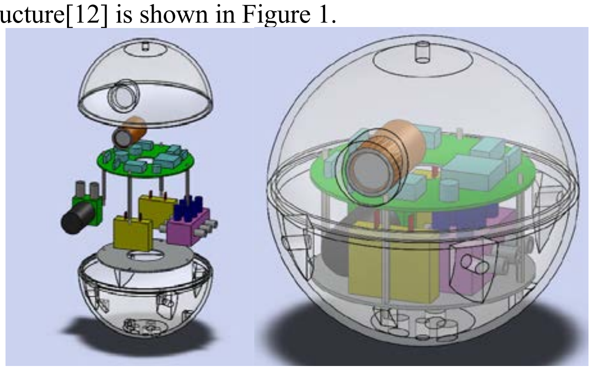

Figure 1. The robot structure

Control system of the transformer internal detection robot mainly consists of remote control and robot ontology control system, and communicate through WiFi technology, and was tested both theoretically and practically. The robot vision system combined with positioning technology and other related technologies, is used to visually detect transformer internal faults, to achieve the intelligence of power system.

B. The SLAM Method for Robot

The SLAM posterior can be written as:

$$p\left(X_k, M_k | Z_{1:k}, U_{1:k}\right) \tag{1}$$

where the complete trajectory of the robot is denoted by X_k, M_k is the entire map including feature locations. The posterior is conditioned on all observations $Z_{1:k} = [z_1, \ z_2, \ldots, \ z_k]^T$, the sequence of control inputs $U_{1:k} = [u_1, \ u_2, \ldots, \ u_k]^{\mathrm{T}}$. The k is a current time index, $1:k$ represents from the initial moment to the k.

The purpose of SLAM is to achieve the location and mapping of the robot simultaneously. To explain SLAM algorithm for robot, suppose the robot motion model is written as:

$$x_k = f\left(x_{k-1}, \ u_k\right) + \gamma_{k-1} \tag{2}$$

The robot observation model can be written as:

$$z_k = h\left(x_k\right) + \vartheta_k \tag{3}$$

γ_k and ϑ_k are the process and observation noises respectively.

C. The Paritile Filter

The purpose of the particle filter algorithm is to find random sample points and their weights in a set of state space, and to approximate the probability density function. The algorithm is applicable to various linear and non-linear Gaussian and non-Gaussian distribution models.

The posterior probability of the states of the system and the environmental feature map observation are expressed as $p\left(x_k \middle| x_{k-1}\right)$、$p\left(z_k \middle| x_{k-1}\right)$, where x_k and y_k represent the states and observations. Posterior probability density of system observation y_k is expressed as Bayesian probability model:

$$\begin{aligned} &p\left(x_k \middle| z_k, x_{v_k}\right) = \xi p\left(z_k \middle| x_k\right) \int p\left(x_k \middle| x_{v_k}, x_{k-1}\right) \cdot \\ &p\left(x_{k\text{-}1} \middle| z_{k\text{-}1}, x_{v_{k\text{-}1}}\right) dx_{k\text{-}1} \end{aligned} \tag{4}$$

wherein, ξ is a normalization coefficient, $p\left(z_k \middle| x_k\right)$ is the observation probability, $p\left(x_k \middle| x_{v_k}, x_{k-1}\right)$ is the probability of system states x_k under x_{v_k}.

The joint posterior probability density based on the particle filter SLAM algorithm is:

$$\begin{aligned} &p\left(x_{v_k}, x_m\right) = p\left(x_{v_k} \middle| z_k\right) p\left(x_{m_k} \middle| x_{v_k}\right) = \\ &p\left(x_{v_k} \middle| z_k\right) \prod_{i=1}^{n} p\left(x_{m_i} \middle| x_{v_k}\right) \end{aligned} \tag{5}$$

This formula is the basis of particle filtering, but it is difficult to sample, you can use approximate method, which is the Important Sampling Sequence.

Important Sampling Sequence adopts the Monte Carlo method to simulate recursive Bayesian filtering, and the probability density function is described by a random sample substitution function. The weighted sum of these samples is used to represent the density of posterior probability. It can be described as

$$p\left(x_{0:k} \middle| z_{1:k}\right) = \frac{1}{N} \sum_{i=1}^{N} \delta_{0:k}^{i} \tag{6}$$

Defining the sampling close and easy distribution function $q\left(x_k \middle| z_{1:k}\right)$ of probability distribution $p\left(x_{0:k} \middle| z_{1:k}\right)$.

Using random sampling method to get random sample points:

$$\left\{s_k^i\right\}, i=1,\cdots,n$$

The probability density function can be expressed as:

$$p\left(x_k \left| z_{1:k}\right.\right)=\sum_{i=1}^{N}\omega_k^i \delta\left(x_k - s_k^i\right) \tag{7}$$

The weight can be described as:

$$\omega_k^i \propto \frac{p\left(s_k^i \left| z_{1:k}\right.\right)}{q\left(s_k^i \left| z_{1:k}\right.\right)} \tag{8}$$

The probability density function $p\left(x_{k-1}\left|y_{1:k-1}\right.\right)$ at time k-1 is:

$$p\left(x_k \left| y_{1:k-1}\right.\right)=\int p\left(x_k \left| x_{k-1}\right.\right)p\left(x_{k-1}\left| y_{1:k-1}\right.\right)dx_{k-1} \tag{9}$$

In the update step, the prior probability density function is updated while the measurement y_k becomes available at time step k by:

$$p\left(x_k \left| y_{1:k}\right.\right)=\frac{p\left(y_k \left| x_k\right.\right)p\left(x_k \left| y_{1:k-1}\right.\right)}{p\left(y_k \left| y_{1:k-1}\right.\right)} \tag{10}$$

The formula constitute the optimal Bayesian estimation of the state x_k, then the importance distribution can be evaluated as:

$$q\left(x_k \left| x_{0:k-1}, y_{1:k}\right.\right)=q\left(x_k \left| x_{k-1}, y_k\right.\right) \tag{11}$$

The particle weights based on the likelihood is calculated as:

$$\omega_k^i \propto \omega_{k-1}^i \frac{p\left(y_k \left| x_k^i\right.\right)p\left(x_k^i \left| x_{k-1}^i\right.\right)}{q\left(x_k^i \left| x_{k-1}^i, y_k\right.\right)} \tag{12}$$

According to formula (7) and (8), the number of effective particles is calculated to measure the degradation degree of the particle weight, and the resampling is carried out as:

$$N_{eff}=1\bigg/\sum_{i=1}^{N}\left(\omega_k^{*(i)}\right)^2 \tag{13}$$

$$\omega_k^{*(i)}=\frac{p\left(x_k^i \left| y_{1:k}\right.\right)}{q\left(x_k^i \left| x_{k-1}^i, y_{1:k}\right.\right)} \tag{14}$$

The main drawback of particle filter, is that with the continuous iteration of the particle, the variance of the sample weight increases with time, the weight of the few particles is relatively large, the weight of the sample is degraded, and the resampling will lead to the lack of particles, reduce the diversity of sampled particles.

III. PSO-FASTSLAM FRAMEWORK

A. Particle Swarm Optimization

- Particle Swarm Optimization (PSO) is a kind of computational optimization algorithm based on swarm intelligence theory, inspired by the foraging behavior of birds. The algorithm was proposed by American psychologists Kennedy and Eberhart. PSO is an evolutionary technology that promotes groups to develop in a more optimal direction by simulating the behaviors of individuals and groups in biology. The PSO particle filter algorithm treats each sampled particle as a single individual. All particles are attracted by the optimal particle. The group optimization method is used to make the particle set closer to the optimal particle and improve the particle sampling quality.
- The PSO algorithm randomly initializes particles in the feasible solution space and velocity space, the initial position X_k of the particle and the initial velocity V_k, where position is used to represent the solution of the problem. By evaluating the objective function of individual particles, we determine the best location for each particle at k, namely the optimal solution g_{best} is found by the particle itself .

$$v_{k+1}^{(i)}=\omega\cdot v_k^{(i)}+c_1\cdot r_1\cdot\left(p_{best}-x_k^{(i)}\right)+c_2\cdot r_2\cdot\left(g_{best}-x_k^{(i)}\right) \tag{15}$$

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

where ω is the inertia weight coefficient, which balances the ability of global search and local search. c_1 and c_2 are learning factors, so that the particles can reach their own individual historical best position and group history best position, which allows particles to move closer to their own individual historical best position and group history best position. r_1, r_2 generates a random number between $[0,1]$, $x_k^{(i)}$ represents the position of the k-th iteration of the i-th particle. After iterations, the particle set moves to the high likelihood region as a whole.

The steps of the standard PSO algorithm are as follows:

Step 1 Define the initial position of the particle and speed.

Step 2 Calculate the fitness value of each particle using the evaluation function.

Step 3 Update the individual and group historical best position of the particle.

Step 4 Update the speed and the position of the particles.

Step 5 Iterates through steps 2 to 4 until it reaches the maximum number of iterations or the optimal fitness value reaches the set threshold.

Step 6 Set the particle's individual historical best position p_{best} to the particle optimized position.

B. The SLAM Method of Particle Swarm Optimization

The problem of particle degradation is that the variance of the weight of the particle gets bigger over time. This is a common defect in the traditional RBPF method, which affects the diversity of particle distribution and makes the estimation of robot pose less consistent. Usually, the solution to this problem is known as the resampling strategy for particles. However, the particle's resampling strategy will cause the particle to run out. For the sake of avoiding the phenomenon of exhaust particles, increase the diversity of the particle set, the PSO algorithm is introduced into the particle resampling strategy. Re-sampling particles were optimized by PSO iteration adjustment proposal distribution of the particle set.

Particle swarm optimization algorithm and particle filter method are used to describe the distribution of state by particle, and setting the mechanism of particle's own motion.

The SLAM method of mobile robot based on particle swarm optimization is a continuous iterative process, including prediction, PSO optimization and weight calculation.

- Prediction: according to the proposed distribution, the current particle set is predicted and sampled to obtain the particles set at the next moment, $s_k \sim p\left(s_k \middle| s_{k-1}, u_k\right)$.
- PSO algorithm optimization, taking p_{best} as the robot pose of the particle set. Calculate the particle's speed and update it, to obtain the optimized estimated particle set. Calculate the predicted values of the landmarks and the fitness function value of the landmarks, the pose of the particle set is g_{best}, and the PSO optimization ends.
- Calculate the weight of particles $\omega_k = \omega_{k\text{-}1} \cdot p\left(z_t \middle| s_t^*\right)$.
- Resampling, extracting the particles according to the weight and add the particles to the new particle set s_k from the temporary particle sets.

The particle set is closer to the accurate position of the robot before calculating the weight through PSO optimization, so that the weight calculation can better reflect the distribution of the particles, the resampling process is more effective, and the convergence of the particle set is accelerated, which is the position of the robot at the next moment. The forecast provides a better initial value.

IV. RESULTS AND ANALYSIS FOR PSO-FASTSLAM SIMULATION

The simulation environment is based on Tim Bailey's package from University of Sydney. SLAM algorithm precision research is carried out in Matlab software environment.

A. Simulation Environment and Parameters

The simulation experiments of PF SLAM, FastSLAM and PSO-FastSLAM were performed respectively to analyze the accuracy differences between algorithms. The asterisks and curves represent the map features (data points) and the true path of the robot[13].

TABLE I. PARAMETERS IN SIMULATION.

Parameter	Value	Parameter	Value
Wheelbase of vehicle	4 m	Control input noise	3 m/s
Maximum steering angle	30 Degrees	Speed	3 m/s
Maximum range	30m	Observation frequency	5Hz
The range sensor sampling interval	0.025s	External sensor scanning range	360 Degrees

The test is performed in the simulation environment. The solid line represents the real path of the robot, and the dotted line represents the estimated path of the robot in Figure 2-4. The diamond represents the actual sign position, plus indicates the estimated position of the environmental feature.

The performance of PF SLAM, FastSLAM and PSO-FastSLAM is compared, with the actual path and the actual positions landmarks through 50 Monte Carlo trials with 100 particles. The estimated robot paths and the maps obtained from the three algorithms are shown in Figure 2-4.

Figure 2-4 shows the comparison between algorithms. The results show clearly that the algorithm used in this paper is better than PF SLAM and FastSLAM. In other words, in the applied algorithm, estimated mobile robot path, estimated landmark with the actual path and the actual positions landmarks coincide as closely as possible.

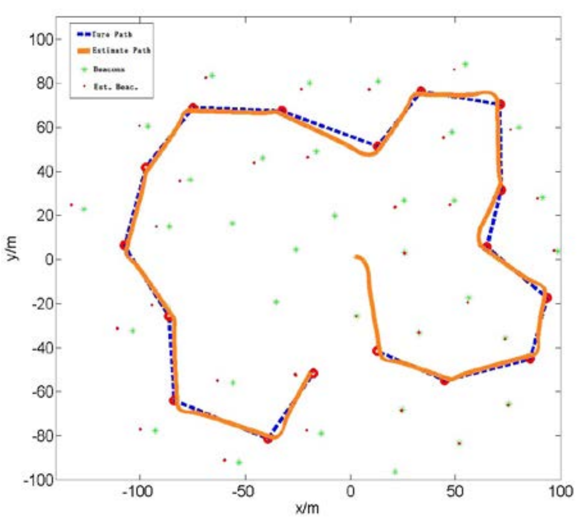

Figure 2. Estimated robot path and features using PF SLAM.

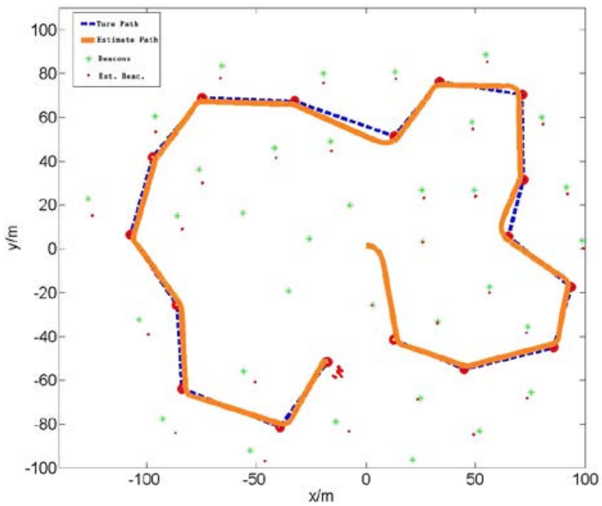

Figure 3. Estimated robot path and features using FastSLAM

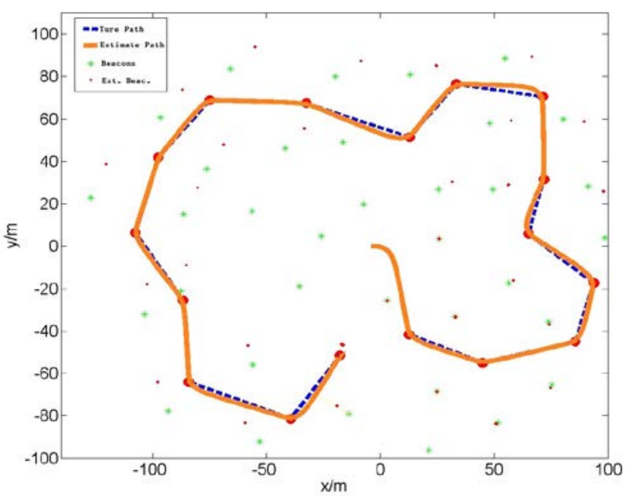

Figure 4. Estimated robot path and features using PSO-FastSLAM

The trajectories of the three algorithms are different from those of the ideal trajectories. Otherwise, PSO-FastSLAM algorithm is more accurate than PF SLAM and FastSLAM, and the pose estimation is more accurate. Whether in the X direction, Y direction, or the posture angle, PSO-FastSLAM has been kept within a stable range, and has a high accuracy compared to the PF SLAM and FastSLAM. As can be seen, the simulation results of mobile robot pose tracking are converged and landmarks are accurately positioned. As a result of measurement updates, resampling of the cubature point will reduce the linearity of the system model error, improve the accuracy of map feature point position estimation, and thus promote location accuracy of mobile robots.

B. Simulation Results and Analysis

In the simulation experiments, compared with PF SLAM and FastSLAM, PSO-FastSLAM validity verified from the aspects of robot positioning error, map feature estimation and calculation time.

- Simulation results

In addition, we use root mean squared errors (RMSE) of robot pose and feature position to measure the performances of the three algorithms, shown in Figure 5. RMSE are calculated as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{k=1}^{T}\left[\left(x_k - \hat{x}_k\right)^2 + \left(y_k - \hat{y}_k\right)^2\right]} \tag{27}$$

Figure 5 shows that PSO-FastSLAM has the smallest errors in both of estimated path and features.

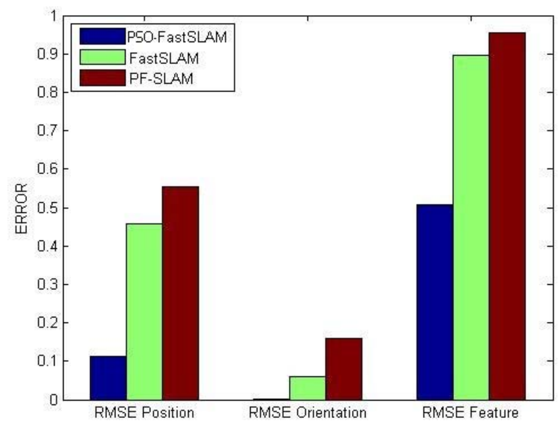

Figure 5. RMSE comparison.

- Evaluation of robot location efficiency

TABLE II. MEAN RUNNING TIME OF THE THREE METHODS

Algorithms	Average Time
PF	0.16
FastSLAM	0.31
PSO-FastSLAM	0.25

It is not difficult to see from Table II that PF has the smallest computational complexity, and the calculated amount of PSO-FastSLAM is also significantly reduced compared to FastSLAM.

From the results, we learn that the RMSE of position, orientation and feature estimation error of the PSO-FastSLAM is much smaller than that of the PF SLAM and FastSLAM, which means the PSO-FastSLAM algorithm can provide more accuracy estimation results. That is because the system noise and measurement noise can be estimated, which the filter precision can be improved. PF is the smallest in computational complexity, and the calculated amount of PSO-FastSLAM is also significantly reduced compared to FastSLAM.

V. CONCLUSION

FastSLAM has been shown to degenerate over time in terms of accuracy due to the particle depletion in resampling process, the particle swarm optimization algorithm is used to concentrate and extend the particles to obtain a more reasonable particle distribution and improve the estimation performance of the conventional particle filter, which adopts the PSO to the particle depletion problem was introduced to improve the performance of FastSLAM. We integrate PSO algorithm into the conventional FastSLAM. The performance of PSO-FastSLAM is verified by reduced RMSE in robot pose and features of map in computer simulation. The simulation and experimental results show the effectiveness of the method. The main advantage of the approach is that the consistency and accuracy of the state estimation is improved in compare with the previous algorithms.

ACKNOWLEDGMENT

This work is supported by the Research on Key Technologies and Systems of Robots for Grid Operation and Maintenance, 090000KK52150073. We thank Dr. Xun Li and Senior Engineer Ronghui Huang of Shenzhen Power Supply Co., Ltd., for provide relevant information and experimental conditions.

REFERENCES

[1] Bosse M., Newman P., Leonard J., and Teller S.: Simultaneous localization and map building in large scale cyclic environments using the Atlas framework. Int. J. Robotics Research, 23(12): 1113-1140, 2004.

[2] Castellanos J.A., Martnez J.M., Neira J., and Tardos, J.D. :Experiments in multisensory mobile robot localization and map building. In Third IFAC Symposium on Intelligent Autonomous Vehicles, pp 173-178, 1998.

[3] N. Kwak, G. W. Kim, and B. H. Lee, "A new compensation technique based on analysis of resampling process in FastSLAM," Robotica, Mar. 2008.

[4] G. Zhang, Y. Cheng, F. Yang, and Q. Pan, "Particle filter based onPSO," in Proc. Int. Conf. Intelligent Computation Technology and Automation, 2008, pp. 121-124.

[5] C. Kim, R. Sakthivel, and W. K. Chung, "Unscented FastSLAM: Arobust and efficient solution to the SLAM problem," IEEE Trans.Robotics, vol. 24, Aug. 2008.

[6] Eustice R., Singh H., Leonard J., Walter M. and Ballard R. : Visuallynavigating the RMS Titanic with SLAM information filters. In Robotics:Scinece and Systems, (2005).

[7] Newman P.M. and Leonard J.J. : Pure range only subsea SLAM. In Proc.IEEE Int. Conf. Robotics and Automation, (2003).

[8] Bailey T., Nieto J. and Nebot E.: Consistency of the FastSLAMalgorithm. IEEE International Conference on Robotics and Automation,pp 424-429, (2006).

[9] Bailey T., http://www-personal.acfr.usyd.edu.au/tbailey/software/, 05.31.2013.

[10] Montemerlo, M. and Thrun, S. and Koller, D. and Wegbreit, B.,"FastSLAM: A Factored Solution to the Simultaneous Localization andMapping Problem", Proceedings of the AAAI National Conference onArtificial Intelligence, (2002).

[11] Lee H C, Park S K, Choi J S, et al. "PSO-FastSLAM: An improved FastSLAM framework using particle swarm optimization" IEEE International Conference on Systems, Man and Cybernetics. IEEE Press, 2009:2763-2768.

[12] R. Huang, Y. Zhao, X. Li, Y. Feng, G. Wu, Z. Li, "Study on fault detection robot for oil-immersed transformer based on WiFi control", 2017 Chinese Automation Congress (CAC), 20-22 Oct. 2017, pp. 50-55.

[13] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in Proc. IEEE Int. Conf. Robotics and Automation, 2006,pp.424-429.