Proceedings of the 2018 13th World
Congress on Intelligent Control and Automation
July 4-8, 2018, Changsha, China

Adaptive Iterated Cubature Particle Filter for Mobile Robot Monte Carlo Localization*

Yi Zhang, DaoFang Chen, HaiBo Lin, LiMing Zhao

Abstract— **In order to solve the problem of computational burden and poor performance in real time in cubature Monte Carlo localization (CMCL), a novel algorithm is presented in this paper. Firstly, a Cubature Particle Filter (CPF) for generating the importance proposal distribution by Gauss-Newton iterative Cubature Kalman Filter (ICKF) is designed. This algorithm is not limited by the high-order truncation error of ordinary cubature particle filters. Subsequently, enhance CPF by automatically adjusting the particle set size using the Kullback-Leibler Distance (KLD) standard, thereby increasing the speed of the newly proposed Adaptive Iterative CPF (AICPF). Simulation result is compared with standard cubature particle filter which demonstrates that the proposed AICPF is superior to the previous method in estimating the mean square and computational cost of the error. In addition, this study also applies AICPF to robot positioning on robotic operating systems (ROS). An analysis is conducted to confirm feasibility and efficiency of the adaptive iterated cubature MCL (AICMCL), which improves the accuracy of robot localization, and recovers more quickly from interference. It adjusts the number of particles needed for localization in real time, reduces computational burden, and improves the real-time processing capability.**

I. Introduction

Mobile robot localization uses priori information of the environment map, its pose estimation, current measurements of sensors and other input information, producing more accurate current pose estimate of robot through a series of processing and conversion [1]. Monte Carlo localization (MCL) based on particle filter [2] uses the posterior distribution to replace prior distribution for sampling. The observation likelihood function is combined with to evaluate the importance weight of each particle, and the correction affection of state estimation based on the current measurements in robot environment is ignored, thus resulting in particle set degeneracy problem. To solve this problem, scholars have made a great deal of research work. Unscented particle filter algorithm was put forward by R. Merwe [3], the proposal distribution of particle filter with unscented Kalman filter (UKF) was designed, making particles more focused on the likelihood of a high observation area. E. Y. Wu [4] used the extended Kalman filter to design the proposal distribution of particle filter, and Markov chain Monte Carlo (MCMC) re-sampling method with variable boundary of proposal density was put forward to increase the particles' refining capacity. Extending Monte Carlo localization method was proposed by J. Pérez [5], which detected and reset its positions with visual place recognition information when the robot was lost. A. Alhashimi [6] effectively reduced computation burden in MCL algorithm by improving the measurement model. Monte Carlo localization based on cubature particle filter (CPF) was advanced by Q. L. LI [7], overcoming particle set degeneracy problem. Cubature Kalman filter (CKF) was utilized to generate more accuracy proposal distribution, and the results verified the efficiency superior to general and unscented MCL algorithm. However, linearization process of the nonlinear system in cubature Kalman filter leads to the higher-order truncation error [8]. For each iterated in cubature MCL algorithm, it has to obtain particle set through sequential importance sampling and calculate corresponding importance weights, resulting in a large amount of computation, and the negative impact of the algorithm in real time.

Based on above, this paper proposes a new algorithm for robot localization. Initially, a new CPF using Gauss-Newton iterated cubature Kalman filter (ICKF) to generate the importance proposal distribution is designed, which avoids the higher-order truncation error. Subsequently, the Kullback–Leibler distance (KLD) standard is utilized to automatically adjusting the particle set size thus enhancing the efficiency and the real-time processing capability of the newly proposed adaptive iterated CPF (AICPF). Superiority of the proposed AICPF in terms of mean square error (MSE) and computational cost have been demonstrated utilizing an analysis on simulation and experiment through a comparative study with the standard CPF.

The rest of the paper is organized as follows. The basic Monte Carlo localization is described in Section 2. Section 3 is focuses on the development of the AICPF. Section 4 provides the results of the numerical simulation and experiment, presenting the comparison of AICPF Section 4 provides the results of simulation calculations and real-world experiments, with standard cubature particle filter. In addition, this section contains a comprehensive analysis for robot localization on ROS. Section 5 summarizes the results.

II. Basic Monte Carlo Localization

The motion state equation and the measurement mode of robot are expressed as

*Resrach supported by Chongqing Education Commission Natural Science Foundation(Grant No. KJ1500418).

Yi Zhang, School of Advanced Manufacturing Engineering, Chongqing University of Posts & Telecommunications, Chongwen Ave. #2, Chongqing, 400065, China. (e-mail: zhangyi@cqupt.edu.cn).

DaoFang Chen, School of Advanced Manufacturing Engineering, Chongqing University of Posts & Telecommunications, Chongwen Ave. #2, Chongqing, 400065, China. (e-mail: s151331001@stu.cqupt.edu.cn).

HaiBo Lin, School of Automation, Chongqing University of Posts & Telecommunications, Chongwen Ave. #2, Chongqing, 400065, China.

LiMing Zhao, School of Advanced Manufacturing Engineering, Chongqing University of Posts & Telecommunications, Chongwen Ave. #2, Chongqing, 400065, China. (e-mail: zhaolm@cqupt.edu.cn).

978-1-5386-7346-1/18/$31.00 © 2018 IEEE

$$x_t = f_t(x_{t-1}, u_{t-1}) \tag{1}$$

$$z_t = h_t(x_t, v_t) \tag{2}$$

Where $x_t \in R^{n_x}$ represents the motion state of the system at t. $z_t \in R^{n_z}$ represents measurement information of the system at t. f_t, h_t represent the motion state transition probability function and measurement function of the system, respectively. u_t, v_t represent process noise and measurement noise of the system, respectively.

Monte Carlo localization (MCL) is also known as particle filter, it is one of probability localization methods. When the priori map m and the system's initial state x_0 are given, discrete weight particle set S_t is used to estimate posterior probability of the robot's pose state x_t. Based on the sequential importance sampling (SIS) [9], firstly draw the weight particle set S_t from the importance function $q(x_t \mid x_{t-1}, z_t, u_{t-1})$; in order to obtain posterior belief discrete estimation $Bel(x_t)$ of the current system status, re-sampling is applied through the particle set; finally, the mean of particle set represents minimum variance estimation in the system state.

In particle filter, $\{x_t^{[i]}, \omega_t^{[i]}\}_{i=1}^{n}$ represents the weight particle set S_t, $x_t^{[i]}$ is the current particle state whose index is i acquired from the importance function. $\omega_t^{[i]}$ are nonnegative numerical factors called importance weights, which sum up to one. n is the total number of particles in weight particle set. Importance weights $\omega_t^{[i]}$ are calculated as

$$\omega_t^{[i]} = \omega_{t-1}^{[i]} \frac{p(z_t \mid x_t^{[i]}) p(x_t^{[i]} \mid x_{t-1}^{[i]}, u_{t-1})}{q(x_t^{[i]} \mid x_{t-1}^{[i]}, z_t, u_{t-1})} \tag{3}$$

The importance function represents a robot system dynamics as

$$q(x_t \mid x_{t-1}, z_t, u_{t-1}) \omega_t^{[i]} = p(x_t^{[i]} \mid x_t^{[i-1]}, u_{t-1}) \tag{4}$$

And it is expressed as Gaussian distribution

$$q(x_t \mid x_{t-1}, z_t, u_{t-1}) = N(x_t, P_t) \tag{5}$$

The posterior belief estimation of pose state x_t is to be

$$Bel(x_t) \approx \sum_{i=1}^{n} \omega_t^{[i]} \delta(x_t - x_t^{[i]}) \tag{6}$$

Re-sampling is a commonly used and effective method to eliminate particle degradation. Re-sampling balances the importance of particles. However, re-sampling can also remove good particles from the particle set to increase poor particles, resulting in loss of particle diversity. Hence, it's important to monitor the sampling efficiency in each step to enhance its robustness. The effective sample size is taken as the average of the measured particle diffusion [10] as described below

$$N_{eff} = \frac{1}{\sum_{i=1}^{n} (\omega_t^{[i]})^2} \tag{7}$$

Where $N_{eff} \in [1, n]$, $N_{thr} = 0.75n$.

From the design of the importance proposal distribution function can be seen, it ignores the recent evidence z_t, resulting in particle degradation and the failure of robot localization. An alternative remedy for the degeneracy problem and computational burden are that choosing a good proposal distribution and sample size adaptation.

III. Improved Cubature MCL For Mobile Robot Localization

Cubature particle filter utilizes cubature Kalman filter [11] to generate more accuracy proposal distribution, which introduces most recent measurements into sequential importance sampling routine of the particle filter. Gauss-Newton iterated updating is used to reduce higher-order truncation error in cubature particle filter, improving its accuracy. In addition, the KLD algorithm is implemented to reduce the computational burden and ensure the accuracy of the point estimate or probability density function. According to the distribution of prediction particles in free space, adjusting the number of particles online, as it is more stable and of a faster processing speed.

In order to describe and illustrate the AICPF, the ICKF and the KLD-based sampling are described in the current section.

A. *Iterated Cubature Kalman Filter as Proposal Distribution*

Cubature Particle Prediction

a) Calculate cubature particle set according to the state of last time of augmented particle $x_{t-1}^{a[i]}$ and augmented particle covariance matrix $P_{t-1}^{a[i]}$.

$$\{C_{j,t-1}^{[i]}\}_{j=1...2(n_x+n_u)}^{i=1...n_{ckf}} = \{\sqrt{P_{t-1}^{a[i]}}\xi_j + x_{t-1}^{a[i]}\}_{j=1...2(n_x+n_u)}^{i=1...n_{ckf}} \tag{8}$$

Where augmented particle state meets Gaussian distribution, $x^a \sim N(x_{t-1}^{a[i]}, P_{t-1}^{a[i]})$.

b) Propagate cubature particle set using the system state transition function f to get transformed cubature particle set.

$$\{X_{j,t|t-1}^{[i]}\}_{j=1...2(n_x+n_u)}^{i=1...n_{ckf}} = \{f(C_{j,t-1}^{x[i]})C_{j,t-1}^{u[i]}\}_{j=1...2(n_x+n_u)}^{i=1...n_{ckf}} \tag{9}$$

c) Calculate prediction particle state and covariance matrix by cubature particle set, combining Spherical-radial rules.

$$x_{t|t-1_{ckf}}^{[i]} = \frac{1}{2(n_x+n_u)} \sum_{j=1}^{2(n_x+n_u)} X_{j,t|t-1}^{[i]} \tag{10}$$

$$P^{i}_{t|t-1_{ckf}} =$$

$$\frac{1}{2(n_x+n_u)}\sum_{j=1}^{2(n_x+n_u)}(X^{[i]}_{j,t|t-1}-x^{[i]}_{t|t-1})(X^{[i]}_{j,t|t-1}-x^{[i]}_{t|t-1})^T \tag{11}$$

Iterated Cubature Particle State Updation

Iterated updating can effectively reduce higher-order truncation error caused by the nonlinear system linearization in cubature Kalman filter, so that Gauss-Newton iterated updating is used to reduce this error, improving the accuracy of filtering. The iterated updating equation [12] as

$$x_{t,it} = x_{t|t-1} + K_{t,it-1}(z_t - z_{t|t-1} - H_{t,it-1}(x_{t|t-1} - x_{t|it})) \tag{12}$$

Where it represents the number of iterated updating. $K_{t,it-1}$ is the Kalman filter gain after $it-1$ iterations. $H_{t,it-1}$ is the Jacobian of measurement model after $it-1$ iterations. The equation above is expressed as the form of a linear statistical error propagation [13]. The Kalman filter gain and the measurement Jacobian in iterated updating equation are rewritten as

$$K_{t,it-1} = \mathrm{cov}(x,z)(\mathrm{cov}(z,z)+R)^{-1} \tag{13}$$

$$H_{t,it-1} = \mathrm{cov}(x,z)^T P^{-1}_{t|t-1} \tag{14}$$

We can calculate the state iterated updating of the $i-t\mathrm{h}$ cubature particle by cubature conversion using statistical linearization in place of the first-order linearization.

a) Calculate cubature particle set according to prediction particle state $x^{[i]}_{t|t-1,it-1}$ and covariance matrix $P^{[i]}_{t|t-1,it-1}$.

$$\begin{aligned}\{X^{[i]}_{t|t-1,it-1}\}^{i=1\dots n_{ckf}}_{j=1\dots 2(n_x+n_u)} \\ =\{\sqrt{P^{[i]}_{t|t-1,it-1}}\xi_j + x^{[i]}_{t|t-1,it-1}\}^{i=1\dots n_{ckf}}_{j=1\dots 2(n_x+n_u)}\end{aligned} \tag{15}$$

Where the equation meets the initialization condition, $X^{[i]}_{t,0} = X^{[i]}_{t|t-1}$.

b) Calculate measurement prediction using the measurement cubature particle set propagated by measurement function h.

$$\begin{aligned}\{Z^{[i]}_{j,t|t-1,it-1}\}^{i=1\dots n_{ckf}}_{j=1\dots 2(n_x+n_u)} \\ = \{h(X^{x[i]}_{j,t|t-1,it-1}, X^{v[i]}_{j,t|t-1,it-1})\}^{i=1\dots n_{ckf}}_{j=1\dots 2(n_x+n_u)}\end{aligned} \tag{16}$$

$$z^{[i]}_{t|t-1,it-1} = \frac{1}{2n_x}\sum_{j=1}^{2n_x} Z^{[i]}_{j,t|t-1,it-1} \tag{17}$$

c) Calculate covariance matrix by cubature particle set to obtain particles status of iterated updating.

$$\mathrm{cov}(\boldsymbol{x},\boldsymbol{z})^{[i]}_{t,it-1} = \frac{1}{2n_x}\sum_{j=1}^{2n_x}(\boldsymbol{X}^{[i]}_{j,t|t-1,it-1} - \boldsymbol{x}^{[i]}_{t,it-1})(\boldsymbol{Z}^{[i]}_{j,t|t-1,it-1} - \boldsymbol{z}^{[i]}_{t|t-1,it-1})^T \tag{18}$$

$$\begin{aligned}&\mathrm{cov}(\boldsymbol{z},\boldsymbol{z})^{[i]}_{t,it-1} \\ &= \frac{1}{2n_x}\sum_{j=1}^{2n_x}(\boldsymbol{Z}^{[i]}_{j,t|t-1,it-1} - \boldsymbol{z}^{[i]}_{t|t-1,it-1})(\boldsymbol{Z}^{[i]}_{j,t|t-1,it-1} - \boldsymbol{z}^{[i]}_{t|t-1,it-1})^T\end{aligned} \tag{19}$$

The Kalman filter gain of the *i-th* particle after the *it-th* iteration is

$$K^{[i]}_{t,it-1} = \mathrm{cov}(x,z)^{[i]}_{t,it-1}(\mathrm{cov}(z,z)^{[i]}_{t,it-1})^{-1} \tag{20}$$

Iterated updating of status and state covariance matrix are

$$\begin{aligned}x^{[i]}_{t,it} &= x^{[i]}_{t|t-1} + K^{[i]}_{t|t-1}(z_t - z^{[i]}_{t|t-1,it-1}) \\ &\quad -(\mathrm{cov}(x,z)^{[i]}_{t,it-1})^T (P^{[i]}_{t|t-1})^{-1}(x^{[i]}_{t|t-1} - x^{[i]}_{t,it-1})\end{aligned} \tag{21}$$

$$P^{[i]}_{t,it} = P^{[i]}_{t|t-1} - K^{[i]}_{t,it-1}\,\mathrm{cov}(z,z)^{[i]}_{t,it-1}(K^{[i]}_{t,it-1})^T \tag{22}$$

Execute sequential importance sampling after cubature Kalman filter iterated updating.

$$x^{[i]}_t \sim N(x^{[i]}_{t,it}, P^{[i]}_{t,it}) \tag{23}$$

If the robot simultaneously obtains multiple environmental characteristics, status iterated updating of each particle in cubature particle set is implemented, then execute sequential importance sampling and calculate the importance weights of the particle according to the (3). Particularly, when the number of iterations take $it=1$, the iterated updating of cubature particle set will degenerate to CPF updating process.

B. *KLD-based Sampling*

Kullback-Lerbler distance (KLD) sampling [14] has been successfully applied in areas such as nonlinear state estimation. The key idea is to determine the number of particles in each iteration of the particle filter so that the error between the true posterior and the particle-based approximation is less than ε in the case of a probability $1-\delta$ .The error is determined by calculating Kullback-Lerbler distance. KLD represents the distance between two probability distributions p and q .

$$K(p,q) = \sum_x p(x)\log\frac{p(x)}{q(x)} \tag{24}$$

KLD is never negative and it is zero if and only if the two distributions are identical. It is a measure of the difference between the probability distribution. Analysis by the probability theory, when n satisfies a certain amount, we can guarantee that the KLD between the true posterior and the particles-based approximation is less than ε , so as to ensure a minimum approximation error between the two, then the value of n is

$$n = \frac{1}{2\varepsilon}\chi^2_{k-1,1-\delta} \tag{25}$$

Where the quantiles of the chi-square distribution is given by

$$P(\chi^2_{k-1} \le \chi^2_{k-1,1-\delta}) = 1-\delta \quad (26)$$

A good approximation is given by the Wilson-Hilferty transformation, which yields

$$n = \frac{k-1}{2\varepsilon}\left\{1-\frac{2}{9(k-1)}+\sqrt{\frac{2}{9(k-1)}}z_{1-\delta}\right\}^3 \quad (27)$$

Where $z_{1-\delta}$ is the upper $1-\delta$ quantile of the standard normal distribution.

IV. Experimental Results and Analysis

A. Simulation Research

We evaluate the efficiency of adaptive iterated cubature particle filter by simulation, and make a comparison with basic cubature particle filter using MATLAB 7.0 on a standard PC with 2.19 GHz processor, 1.96 GB RAM. The parameters of adaptive iterated cubature particle filter are set as $\varepsilon = 0.1$, $\delta = 0.01$, $\Delta = 0.5cm \times 10^{\circ}$, $n_{\min} = 10$. We use the following typical prediction model

$$x_t = x_{t-1}/2 + 2.5x_{t-1}/(1+x_{t-1}^2) + 8\cos(1.2t) + u_t \quad (28)$$

and updating model

$$z_t = x_t^2/20 + v_t \quad (29)$$

Where u_t and v_t are zero mean Gaussian random variables with variance 10 and 1. There are 100 initial particles at first. The filter performance is demonstrated via a 100-independent- run Monte Carlo simulation. The state mean square error (MSE) and number effect particles percentage (NEFF) measuring particles degeneracy degree are as evaluation index, MSE and NEFF calculated by the following formula

$$MSE(\hat{x}_t) = \frac{1}{N_T}\sum_{t=1}^{N_T}(x_t - \hat{x}_t)^2 \quad (30)$$

$$NEFF = \frac{100\%}{n(\sum_{i=1}^{n}(\omega_t^{[i]})^2)} \quad (31)$$

Where x_t is true state; $\hat{x}_t$ is estimated state; N_T is the number of iterations (time steps) and the value of 100; n is the number of particles; $\omega_t^{[i]}$ is the importance weights.

Fig. 1 shows state estimation error of basic cubature particle filter and the adaptive iterated cubature particle filter. We can see the convergence speed of AICPF is similar to CPF, and AICPF has smaller mean square error with higher accuracy under the same conditions. Fig. 2 indicates the volatility of sample size in filtering. From Fig. 2, the number of particles is fixed in basic CPF, but adaptively reduces according to the state in free space in AICPF, significantly less than CPF. Table 1 compares the performance of two particle filters. We see that less time is wasted on updating the weight of small particles, and the percentage of effective particle is relatively higher, effectively weakening particle degradation in AICPF. Therefore, the improved algorithm has the characteristics of higher estimation accuracy, reducing particle degradation. It is an effective remedy for the computational burden, and poor real-time processing capability of CPF.

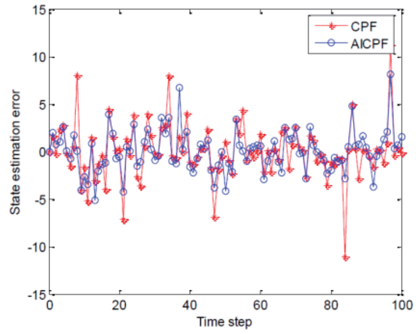


Figure 1. The state estimation error of two particle filters

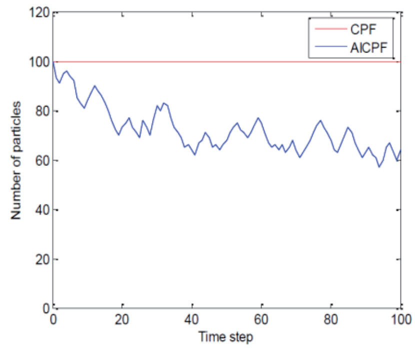


Figure 2. Number of particles in two particle filters

TABLE I. Performance Comparison of Two Particle Filters

Filters	Number of particles	MSE(cm)	NEFF(%)	Running time(s)
CPF	100	2.8476	79.6%	26.809
AICPF	72.287	2.1650	88.4%	15.547

B. Mobile Robot Localization Using AICMCL Algorithm on ROS

In the experiment, CMCL algorithm and AICMCL algorithm are verified in experiments, respectively. Pioneer3-DX robot equipped with a standard PC with 2.19 GHz processor, 1.96 GB RAM, installing robot operating system (ROS) [15], collects data using a URG-hokuyo laser ranger. An environment as shown in Fig. 3. The map of this environment is built using the slam gmapping node in ROS with resolution 0.050 m/pix. The measurement error and the movement error are mixture-Gauss set to be 40cm/3m with probability 0.95. Robot translation speed is set to 0.2m/s. Initial number of particles is fixed to 500 particles. Initial position of robot is unknown, using the odometer and laser data as observation information and realizing robot localization and navigation on a known environment map finally.

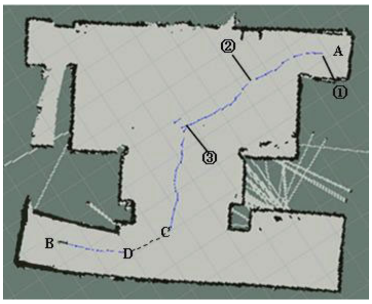
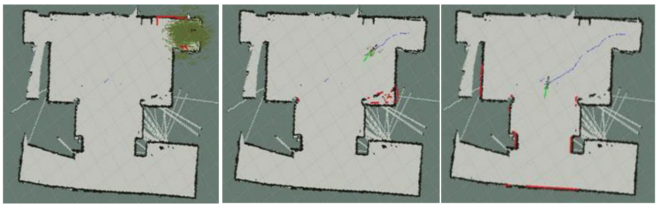


Figure 3. Path of mobile robot

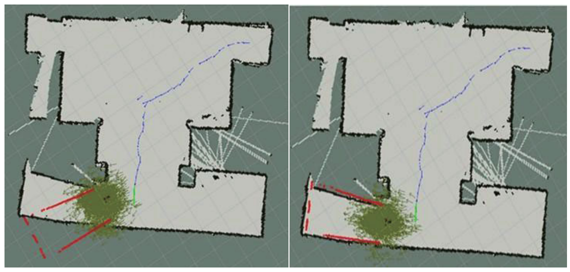
(a) CMCL in position ① (b) CMCL in position ② (c) CMCL in position ③



(d) AICMCL in position ① (e) AICMCL in position ② (f) AICMCL in position ③

Figure 4. Number of particles distribution in both algorithms

Fig. 3 shows the path from A to B of Pionner3-DX. The interference occurred between C and D. The number of particles needed in robot's localization according to its motion using both algorithms as Fig. 4, which are distribution of the particles in position ①,②,③. The small arrow around robot represents the particle. From Fig. 4(a)-(c), CMCL algorithm is used, particle distribution in space remains unchanged. With the movement of the robot, robot's position becomes deviated, resulting in the failure of localization; in reverse, the robot's pose has high uncertainty and larger number of particles dispersing around robot in initial position in AICMCL algorithm, as shown in Fig. 4(d). The robot combining with measurements and control information correct the pose, and number of particles needed in its localization decline with the movement of the robot. Measurement information of laser sensor collection coincides with the current robot's location information, and the localization accuracy is relatively higher.



(a) robot recovering in CMCL (a) robot recovering in AICMCL

Figure 5. Robot recovering from interference

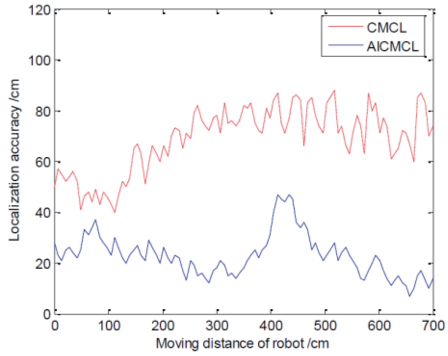


Figure 6. Accuracy in robot's localization according to moving distance change

Further on, the robot interference testing in experiment introduces large errors. The results show AICMCL algorithm recovers more quickly from interference with more particles, as shown in Fig. 5. Fig. 6 depicts localization accuracy in both algorithms with robot's moving distance change, which evaluates localization accuracy declining with robot's moving distance increasing, and the robot cannot recover from interference in CMCL algorithm because of particle degradation, leading to the failure of localization; in comparison the robot corrects current position based on the measurements of laser sensor in AICMCL algorithm, and weakening particle degeneracy problem, improving localization accuracy. The average localization error is 22.970cm.

Fig. 7 is number of particles in both algorithms with robot's moving distance change, which also illustrates that the number of particles is fixed in CMCL, however, robot can adjust the number of particles needed in localization in real-time according to the free space state in the process of motion using AICMCL. Particles needed in robot's localization is stable at 60 when the surrounding environment with less noise. While larger interference is in environment, the state uncertainty of robot is higher and larger number of particles is needed.

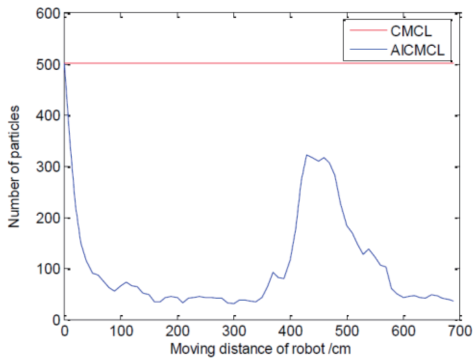


Figure 7. Number of particles needed in robot's localization according to moving distance change

TABLE II. PERFORMANCE COMPARISON OF TWO LOCALIZATION ALGORITHM

Localization algorithm	Averagenumber of particles	Localization accuracy(cm)	Localization time(s)	Recover time(s)
CMCL	500	68.653	114.317	∞
AICMCL	105.35	22.970	35.543	45.126

Table 2 compares the localization performance of both algorithms. It shows improved algorithm with higher localization accuracy can adjust the number of particles in real time according to the state of the robot in space, reducing computational cost, recovering more quickly from interference and improving efficiency of robot localization.

V. CONCLUSION

A novel AICPF is developed for estimation of robot localization. The proposed AICMCL algorithm combines the ordinary particle filter with the ICKF. ICKF alleviates the deficiency of ignoring the current state and eliminates higher-order truncation error. Additionally, the number of particles is also adjusted adaptively by the AICMCL based on KLD standard to enhance the real-time processing capability of the proposed method. Finally, simulation and experiment show the superiority of the proposed AICMCL over the CMCL algorithm in terms of accuracy and computational cost.

REFERENCES

[1] S. Thrun, D. Fox, W. Burgard, "Robust Monte Carlo Localization for Mobile Robots", Artificial Intelligence, vol. 128, no. 2, (2001), pp. 99-141.

[2] M. Arulampalam, S. Maskell S, N. Gordon, "A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking", IEEE Transactions on signal processing, vol. 50, no. 2, (2002), pp. 174-188.

[3] R. V. D. Merwe, A. Doucet, N. D. Freitas, "The Unscented Particle Filter", Advances in Neural Information Processing Systems, vol. 13, (2001), pp. 584-590.

[4] E. Y. Wu, Z. Y. Xiang, J. L. Liu, "Robust Robot Monte Carlo Localization", Acta Automatica Sinica, vol. 34, no. 8, (2008), pp. 907-911.

[5] J. Pérez, F. Caballero, L. Merino, "Enhanced Monte Carlo Localization with Visual Place Recognition for Robust Robot Localization", Journal of Intelligent and Robotic Systems, vol. 80, no.3, (2015), pp. 1-16.

[6] A. Alhashimi, R. Hostettler, T. Gustafsson, "An Improvement in the Observation Model for Monte Carlo Localization", Proceeding of 11th International Conference on Informatics in Control, Automation and Robotics, Austria, (2014), pp. 498-505.

[7] Q. LI, Y. SONG, "Cubature MCL: Mobile Robot Monte Carlo Localization Based on Cubature Particle Filter", Proceeding of 31st Chinese Control Conference, Herfei, China, (2012), pp. 5141-5145.

[8] B. M. Bell, F. W. Cathey, "The Iterated Kalman Filter Update as a Gauss-Newton Method", IEEE Transactions on Automatic Control, vol. 38, no. 2, (1993), pp. 294-297.

[9] S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics", Communications of the Acm, vol. 45, no. 3, (2005), pp. 52-57.

[10] J. S. Liu, "Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling", Statistics and Computing, vol. 6, no. 2, (2000), pp. 113-119.

[11] I. Arasaratnam, S. Haykin, "Cubature Kalman Filters", IEEE Transactions on Automatic Control, vol. 54, no. 6, (2009), pp. 1254-1269.

[12] G. Sibley, G. Sukhatme, L. Matthies, "The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo", Robotics Science and Systems, (2006), pp. 263−270.

[13] K. Shojaie, K. Ahmadi, A. M. Shahri, "Effects of Iteration in Kalman Filters Family for Improvement of Estimation Accuracy in Simultaneous Localization and Mapping", Proceeding of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Zurich, Switzerland, (2007), pp: 1−6.

[14] D. Fox, "Adapting the Sample Size in Particle Filters through KLD - sampling", International Journal of Robotics Research, vol.22, no.12, (2003), pp: 985-1103.

[15] About ROS [EB/OL]. URL: [2015-02-13] http://www.ros.org/about-ros/.