TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.6, June 2014, pp. 4206 ~ 4214
DOI: 10.11591/telkomnika.v12i6.5059

An SLAM Algorithm Based on Square-root Cubature Particle Filter

Xuefeng Dai, Zuguo Chen*, Chao Yang, Laihao Jiang, Biao Cai
College of Computer and Control Engineering, Qiqihar University, No.42 of Wenhua Street, Jianhan District, Qiqihar, Heilongjiang, 161006, China, Tel (+86)0452-2738173
*Corresponding author, e-mail: boss888.cool@163.com

Abstract

The lack of the latest measurement information and the Particle serious degradation cause low estimation precision in the tradition particle filter SLAM (simultaneous localization and mapping). For solve this problem, a SRCPF-SLAM (square cubature particle filter simultaneous localization and mapping) is proposed in this paper. The algorithm fuses the latest measurement information in the stage of the prior distribution updated of the particle filter SLAM. It designs importance density function by SRCKF (Square-root Cubature kalman filter) that is more close to the posterior density, and it spreads the square root of state covariance. So, the algorithm ensures the symmetry and the positive semi-definiteness of the covariance matrix and improves numerical estimation precision and stability. The simulation results show that the proposed algorithm has higher accuracy of the state estimation when compared with the the PF-SLAM (particle filter simultaneous localization and mapping) algorithm, EPF-SLAM (extend particle filter simultaneous localization and mapping) algorithm and the UPF-SLAM (unscented particle filter simultaneous localization and mapping) algorithm.

Keywords: *particle filter, square-root cubature kalman filter, simultaneous localization and mapping, mobile robot*

1. Introduction

The mobile robot SLAM is that the robot does localization and builds incremental map on the basis of the position estimation and the observation data of the senor in the unknown environment. It is the basis that the robot finishes Environment detection, navigation and target tracking. The SLAM is seen as the key that the robot realize completely autonomy. So, the mobile robot has been the most popular project, the most abundant project and the most pioneering project in the robot research field [1].

The extend kalman filter was first proposed by smith and cheeseman [2]. This algorithm is widely used in the SLAM. It estimates the posterior probability density of the pose and the environment feature of robot. The major disadvantage is that this algorithm has more amounts of computations. And the algorithm must assure that the input noise and the observation noise of system should obey a gaussian distribution [3]. In recent years, the particle filter was used as a new method to deal with the SLAM problem by Murphy and doucent [4]. The algorithm can build feature map and grid map according to the needs. And this algorithm can effectively solve the problem of data correlation. But the choice of importance density function affects the particle filter performance because the particle need be extracted from the importance density function [4]. However, the state transition prior distribution which does not contain the latest measurement data is as the importance density function in the traditional particle filter. So, the algorithm introduces larger weight variance and it can't approximate posterior probability very well [5-6]. Especially, when the measurement data appear in the tail end of the transition probability distribution or the likelihood function excessive concentration, compared with transition probability distribution, the particle filter may failure. For solving the above problems in the traditional particle filter very well, a lot of research scholars have done large amount of work in terms of the choice of importance sampling function. The UKF has been used to design the importance density function of the particle by [7], and they have proposed the UPF-SLAM algorithm. When the robot builds map, the algorithm reduces the needed particle number. But, it needs to ascertain the three unknown parameters in scaled unscented transformation (SUT)

Received November 4, 2013; Revised December 30, 2013; Accepted January 19, 2014

according to experience. The choice of parameter directly affects the accuracy of the SLAM [8-9].

In order to make up for the defects of the SLAM algorithm and to improve the precision of the mobile robot SLAM algorithm, the SRPF-SLAM algorithm is proposed in this paper. The algorithm basic idea is that the particle filtering framework introduces the latest SRCKF nonlinear filtering algorithm and fuses the latest observation data. It generates the importance density function of particle filter by using the SRCKF algorithm. The importance density function is more close to the system posterior probability distribution according to giving consideration to nonlinearity and non-gaussian characteristics. And, the algorithm has spread the square root of state covariance. So, it ascertains the symmetry and the half positive definitiveness of covariance matrix. It greatly improves the performance of the standard particle filter, and improves the precision and the stability of the mobile robot SLAM algorithm.

2. Square-root Cubature Particle Filter

2.1. Cubature Rules

To calculate the nonlinear transfer probability density of the Gaussian distribution is the most important step that it Implement Bayesian filtering under gaussian field. That is to say, it will calculate the gaussian weighted integral of the nonlinear function [10]. The gaussian weighted integral of the nonlinear function can be given by:

$$I = \int_{R} f(x) N(x; \mu, \Im) dx \tag{1}$$

Here, x specifies n_x dimensional vector. $f(\bullet)$ specifies the nonlinear function. N（:）specifies the Gaussian distribution. R specifies the domain of integration. The analytic value of integral function is obtained difficultly above the integral function. Using equal weights $2n^x$ point numerical integral results are close to the gaussian weighted integral in cubature.

$$I_N(f) \approx \frac{1}{2n_x} \sum_{i-1}^{n_x} f(\sqrt{\frac{2n_x}{2}}[1]_i) = \sum_{i=1}^{2n_{xx}} w_i f(\eta_i) \tag{2}$$

There, n is the state dimension. η_i and w_i can be obtained by:

$$\begin{cases} \eta_i = \sqrt{\dfrac{2n_x}{2}}[1]_i \\ w_i = \dfrac{1}{2n_x} \end{cases} \quad i=1, \ \ldots, \ 2n^x \tag{3}$$

The square-root cubature kalman filter does not need to calculate the Jacobian matrix. It calculates the conditional transition probability density by using the cubature. So, this algorithm can be close to the third order accuracy. The accuracy of the algorithm is higher than the extend kalman filter. The paper [11] gives detailed the calculation steps of the square-root cubature kalman filter. The main calculation formula is as follows.

2.2. Square-root Cubature Particle Filter

The particle filter is that the integral operation translates into the summation of sample points by using the Monte Carlo method. Thereby, it obtains the recursive Bayesian estimation which has the State minimum variance. The algorithm needs obtain directly sample point from the posterior probability density of state. But, the analytic form of the posterior probability density can not be obtained. So, it is difficult to sample for particle. It requires usually an importance density function $\sigma(x_k^i / x_{(0,k-1)}^i, y_{(1,k)}^i)$ which is similar to the posterior probability distribution and it is sampled. The importance density function is easy to sample. But it does not use the latest measurements data. Only a few minorities of particles have larger weights after several times iterations, and most of the particles die rapidly. Thereby, this exist great differences between the collection of the particles and the real posterior probability function generated samples. In order to make the importance density function more close to the posterior probability density, the importance density function needs to consider the influence of the latest measurement value. It makes more particle moves into the higher likelihood function value area [12]. So, the state estimation incorporates the latest measurement value to design the importance density function by using the SRCKF algorithm. A SRCPF algorithm is proposed in this paper.

The SRCPF algorithm calculates the mean and the variance of the importance probability density function by using the SRCPF algorithm, and produces a new posterior probability density distribution. Meanwhile the algorithm generates new particles by the new posterior probability density distribution and completes the state estimation. Each particle sampling points have different Gaussian distribution. However, any non-gaussian distribution can be composed of a series of different Gaussian distribution combinations. It is reasonable to assume the importance density function is Gaussian distribution [13]. The new importance probability density function can be defined as:

$$\sigma(x_k^i / x_{0,k-1}^i, y_{1.k}^i) = \Re(\overline{x}_{k/k}^i, s_{k/k}^i) \tag{4}$$

The detail step of the SRCPF algorithm is as follows:

Step 1:

When k=0, the algorithm Choose the initial particle set X_0^i from the prior probability distribution P(X^0). And compute their expectations and covariance.

$$\overline{X}_0^{(i)} = E[X_0^i] \tag{5}$$

$$P_0^i = E[(X_0^i - \overline{X}_0^i)(X_0^i - \overline{X}_0^i)^T] \tag{6}$$

Here, i=1, 2,…, m

Step 2:

When K>0, the algorithm designs the sampling density function of the particle filter by using the SRCPF algorithm. Each particle is updated by using the SRCPF algorithm in the particle set at the importance particle sampling.

Computing cubature point:

$$\hat{X}_{k-1/k-1}^{(i)} = \overline{X}_{k-1/k-1}^{(i)} \tag{7}$$

$$X_{n,k-1/k-1}^{(i)} = S_{k-1/k-1}^{(i)}\eta_n + \hat{X}_{k-1/k-1}^{(i)} \tag{8}$$

Where n is the state dimension

Time updating:

$$X_{n,k/k-1}^{(i)} = f(X_{n,k-1/k-1}^{(i)}) \tag{9}$$

$$\hat{X}_{k/k-1}^{(i)} = \frac{1}{m}\sum_{i=1}^{m} X_{n,k/k-1}^{(i)*} \tag{10}$$

$$S_{k/k-1}^{(i)} = Tria([X_{n,k/k-1}^{(i)*} \sqrt{Q_k}]) \tag{11}$$

$$X_{k/k-1}^{(i)*} = \frac{1}{\sqrt{m}}[X_{1,k/k-1}^{(i)*} - \hat{X}_{k/k-1}^{(i)}\ X_{2,k/k-1}^{(i)*} - \hat{X}_{k/k-1}^{(i)} \ldots X_{m,k/k-1}^{(i)*} - \hat{X}_{k/k-1}^{(i)}] \tag{12}$$

Measure updating:
The latest observation information is blended into the algorithm

$$X_{n,k/k-1}^{(i)} = S_{k/k-1}^{(i)}\eta_i + \hat{X}_{k/k-1}^{(i)} \tag{13}$$

$$Z_{n,k/k-1}^{(i)} = h(X_{n,k/k-1}^{(i)}) \tag{14}$$

$$\hat{Z}_{k/k-1}^{(i)} = \frac{1}{m}\sum_{i=1}^{m} Z_{n,k/k-1}^{(i)} \tag{15}$$

$$\zeta_{k/k-1}^{(i)} = \frac{1}{m}[Z_{1,k/k-1}^{(i)} - \hat{Z}_{k/k-1} Z_{2,k/k-1}^{(i)} - \hat{Z}_{k/k-1} \ldots Z_{m,k/k-1}^{(i)} - \hat{Z}_{k/k-1}] \tag{16}$$

$$S_{zz,k/k-1}^{(i)} = qr\{[\zeta_{k/k-1}^{(i)}\sqrt{R_k}\,]\} \tag{17}$$

$$X_{k/k-1}^{(i)} = \frac{1}{\sqrt{m}}[X_{1,k/k-1}^{(i)} - \hat{X}_{k/k-1}^{(i)}\ X_{2,k/k-1}^{(i)} - \hat{X}_{k/k-1}^{(i)} \ldots X_{m,k/k-1}^{(i)} - \hat{X}_{k/k-1}^{(i)}] \tag{18}$$

2.1.3. State Updating

$$P_{xz,k/k-1}^{(i)} = X_{k/k-1}^{(i)}\zeta_{k/k-1}^{T(i)} \tag{19}$$

$$K_{k/k-1}^{(i)} = (P_{xz,k/k-1}^{(i)} s_{zz,k/k-1}^{T(i)}) / s_{zz,k/k-1}^{(i)} \tag{20}$$

$$X_{k/k}^{(i)} = \hat{X}_{k/k-1}^{(i)} + K_k^{(i)}(Z_k^{(i)} - \hat{Z}_{k/k-1}^{(i)}) \tag{21}$$

$$U^{(i)} = K_k^{(i)} S_{zz,k/k-1}^{(i)} \tag{22}$$

$$S_{k/k}^{(i)} = cholupdate\{S_{k/k-1}^{(i)}, U^{(i)}, -1\} \tag{23}$$

The algorithm regenerates the particle set based on the state estimation and the covariance matrix of the SRCKF algorithm, and calculates the importance weights of each particle in the particle set.
The importance weights are normalized.

$$X_k^{(i)} \sim \gamma(\hat{X}_k^{(i)} / X_{k-1}^{(i)}, Y_k) = N(\hat{X}_k^{(i)}, S_{k/k}^{(i)}) \tag{24}$$

$$\alpha_k^i = \frac{p(Y_k / \hat{X}_k^{(i)}) P(\hat{X}_k^{(i)} / X_{k-1}^{(i)})}{\gamma(\hat{X}_k^{(i)} / X_{k-1}^{(i)}, Y_k)} \tag{25}$$

$$\alpha_k^i = \alpha_k^{(i)} / \sum_{j=1}^{m} \alpha_k^{(i)} \tag{26}$$

Step 3: Particle re-sampling
Firstly, determine whether it needs samples according to the effective particle number. If it needs samples, the algorithm re-sampled particles by using the particle combination method [14]. The re-sampled particles are $\left\{x_s^j, n_j\right\}_{j=1}^{m}$. Otherwise, the algorithm do state estimation.

Computing State estimation:

$$X_k = \frac{1}{N}\sum_{j=1}^{m} n_j X_k^{(j)} = \sum_{j=1}^{m} \frac{n_j}{N} X_s^j \tag{27}$$

$$P_{k/k} = \frac{1}{N}\sum_{j=1}^{N} (X_k^j - X_{k/k})(X_k^j - X_{k/k})^T \tag{28}$$

According to the new particle state estimation, variance matrix and the above probability model, repeat the steps above for the next filter.

3. An SLAM Algorithm Based on Square-root Cubature Particle Filter

In order to overcome problems which the SLAM algorithm based on conventional particle filter have larger calculated amount and particle degradation, the thought of the SRCKF algorithm lead in the particle filter to improve the sampling process. It enriches the particle samples in the process of particle filter resampling. The algorithm integrates the latest observation data at the prior distribution update stage and designs the importance density function by the SRCKF algorithm. It makes state estimation closer to the posterior probability density of the system state. The algorithm also has spread the square root of the state covariance. It can insure the symmetry and the positive semi-definiteness of the covariance matrix. Thus, it can improve the numerical accuracy and the stability of the SLAM algorithm.

The state vector X_k should include the position information G_s and the environmental information P_N of the mobile robot in the mobile robot SLAM algorithm.

$$X_k = \{G_s, P_N\}^T \tag{29}$$

Here, $G_s = \{x_k, y_k, \theta_k\}$ specifies the position coordinate and the angle coordinate of the mobile under global coordinate. P_N is N environment coordinate set.

The basic thought of the SLAM algorithm based on square-root cubature particle filter is to estimate the mobile robot trajectory and the posterior probability density of the environment map on the basis of the perceptual information of the robot in the environment and the update information of the robot pose. The specific steps of the algorithms are as follows:

1) The algorithm produce randomly the particle set which is constituted by the N particle and the particle weight is W_0. Then, this particle is initialized by using (5)-(6) equations.

2) Particles Predicted: The algorithm calculates the information vector and matrix of the particles according to control input u_k and the robot position distribution at K time.

3) Observation and Data Correlation: The algorithm calculates the particle corresponding feature point coordinates and makes it to association with a landmark in the environment landmark set. Then, it makes the observation new information to associate with the estimation map at the K-1 time by using the data association method of the minimization observation probability function.

4) Particles Updated: When the robot obtains the new observation feature points or the robot predicted pose at K-1 time comparing with one at K time have great changes, the SRCPF algorithm updates particle by using (7)-(13) equations. It will calculate the information vector and matrix and the importance weight of the each particle at K time and the particle importance weights will be normalized.

5) The algorithm resample particle in the particle set according to the particle importance weights α_k^j and removes the small weights particle in the particle set and reserves the big weights particle in the particle set.

6) The algorithm updates feature information in the map by using (27)-(28) equations and makes the correlation failure observation information as the new feature information added to the map.

4. The Experimental Conclusion and Analysis

4.1. Experiment Modeling

Before the SLAM simulation experiment, we need to build a system model for the mobile robot. The established models mainly include system model, robot location model, control command model, environment map model, robot motion model, sensor measurement model and system noise model. In this paper, the Bailey SLAM model is used [15].

(1) The motion model can be obtained by:

$$x_{V,k+1} = \begin{bmatrix} x_{Vx,k+1} \\ x_{Vy,k+1} \\ x_{V\theta,k+1} \end{bmatrix} = \begin{bmatrix} x_{Vx,k} + \Delta T V_{k+1} \cos(x_{V\theta,k} + \varphi_{k+1}) \\ x_{Vy,k} + \Delta T V_{k+1} \cos(x_{V\theta,k} + \varphi_{k+1}) \\ x_{V\theta,k} + \dfrac{\Delta T V_{k+1} \sin \varphi_{k+1}}{B} \end{bmatrix} \tag{30}$$

Input: $x_{V,k}$ specifies the pose of the robot at time k. ΔT specifies the sampling time of the dead reckoning sensors. V_k specifies the velocity of the robot. φ_k is rudder angle. B is two interaxial wheelbases. Output: $x_{V,k+1}$ specifies the pose of the robot at time k+1.

(2) Observation model can be obtained by [16]:

$$Z_k = \begin{bmatrix} r_i \\ \theta i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_{Vx,k})^2 + (y_i - x_{Vy,k})^2} \\ \arctan \dfrac{y_i - x_{Vy,k}}{x_i - x_{Vx,k}} - x_{V\theta,k} \end{bmatrix} \tag{31}$$

Input: (x_i, y_i) specifies the position coordinates of detected the *i* th landmark features. Output: r_i and θ_i respectively specifies the range of the *i* th landmark feature related to the robot and angle of the *i* th landmark feature related to the robot direction.

4.2. Experimental Analyses

In this paper, it adopts respectively the PF algorithm, the EPF algorithm, the UPF algorithm and the SRCPF algorithm in the SLAM experiment, and compares the experimental results of four kinds of algorithm. It compares mainly the deviation of the state estimation with the real path value, the state estimation covariance, the posterior probability distribution and the running time and so on. In order to make the experimental results have more universality, it takes the average result of the 20 time repeated the experiments final result were comparing analyzed.

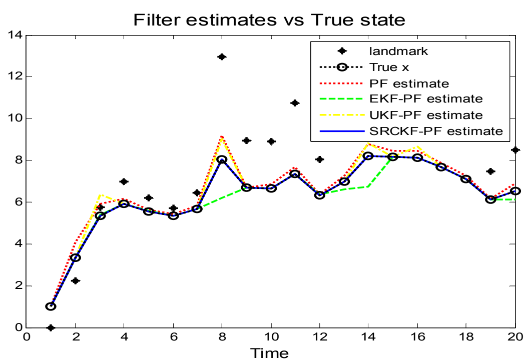

Figure 1. Results of the PF-SLAM Algorithm, the EPF-SLAM Algorithm, the UPF-SLAM Algorithm and the SRCPF-SLAM Algorithm under the Gaussian Noise

Figure 1 is the chart between state estimation and the real path value deviation that the PF algorithm, the EPF algorithm, the UPF algorithm and the SRCPF algorithm are used in the mobile robot SLAM. From Figure 1 shown, the deviation of the PF algorithm and the UPF algorithm between the state estimation and the real value is greater during the first half period of the mobile robot SLAM. The deviation of the PF algorithm between the state estimation and the real value is the greatest. The deviation of the EPF algorithm and the SRCPF algorithm between the state estimation and the real value is relatively less. Among then, the deviation of the SRCPF algorithm between the state estimation and the real value is the minimum. So, the estimation accuracy of the SRCPF algorithm is the highest. The estimation accuracy of the EPF algorithm is slightly poorer than that of the SRCPF algorithm, but the estimation accuracy of the EPF algorithm is higher than that of the UPF algorithm and the PF algorithm. The estimation accuracy of the UKF algorithm is near to that of the EPF algorithm, but the estimation accuracy of the UKF algorithm is higher than that of the PF algorithm. The estimation accuracy of the PF algorithm is the worst.

The deviation of the four kinds of algorithms between the state estimation and the real value shows a trend of decreasing during the second half period of the mobile robot SLAM. Among then, the decreasing rate of the deviation of the EPF algorithm between the state estimation and the real value is larger. It can be seen from Figure 1, the deviation of the PF algorithm between the state estimation and the real value is bigger than that of the other three kinds of algorithms. The deviation of the SRCPF algorithm between the state estimation and the real value is the minimum. The deviation of the UPF algorithm between the state estimation and the real value is relatively smaller than that of the EPF algorithms.

Hence, the estimation accuracy of the SRCPF algorithm is the highest. The estimation accuracy of the UPF algorithm is slightly poorer than that of the SRCPF algorithm, but the estimation accuracy of the UPF algorithm is higher than that of the EPF algorithm and the PF algorithm. The estimation accuracy of the EPF algorithm is poor, but the estimation accuracy of the EPF algorithm is higher than that of the PF algorithm. The estimation accuracy of the PF algorithm is the worst. The deviation of the SRCPF algorithm between the state estimation and the real value is always the minimum during all the processes of the mobile robot SLAM. According on this, It may obtains that the state estimation and the stability of the SRCPF algorithm is better than that of the other three kinds of algorithms.

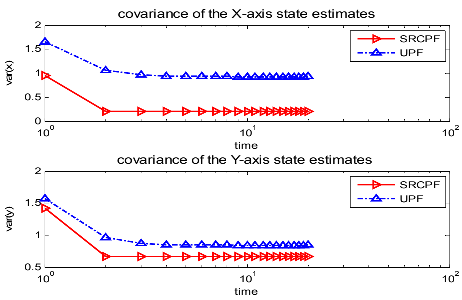

Figure 2. The State Estimates Covariance of the SRCPF Algorithm on the X, Y Coordinates of the Mobile Robot Position with the State Estimates Covariance of the Latest UPF Algorithm on the X, Y Coordinates of the Mobile Robot Position under the Gaussian Noise

It compares the state estimates covariance of the SRCPF algorithm on the X, Y coordinates of the mobile robot position with the state estimates covariance of the latest UPF algorithm on the X, Y coordinates of the mobile robot position in Figure 2. As Figure 2 experimental results, it shows the following analysis results. From the algorithm estimation precision analysis, the state estimates covariance of the SRCPF algorithm for the X, Y coordinates of the mobile robot position is relatively smaller than that of the UPF algorithm. The state estimates covariance of the SRCPF algorithm for the X coordinates of the mobile robot position is 0.5m around smaller than that of the UPF algorithm. The state estimates covariance of the SRCPF algorithm for the Y coordinates of the mobile robot position is 0.1m around smaller than that of the UPF algorithm. The estimation accuracy of the SRCPF algorithm for the coordinates of the mobile robot position is higher than that of the UPF algorithm. The estimation accuracy of the SRCPF algorithm for the X coordinates of the mobile robot position is higher than the estimation accuracy of the SRCPF algorithm for the Y coordinates of the mobile robot position.

From the algorithm estimation stability analysis, the stability of the SRCPF algorithm is better than that of the UPF algorithm. Figure 3 is the posterior probability distribution map of the SRCPF algorithm and the UPF algorithm. It can be seen from Figure3, the posterior probability distribution map of the SRCPF algorithm is gentler than that of the UPF algorithm. This is equivalent to increasing the particle filter distribution range in the sampling process. So, the generating particle sample by using the SRCPF algorithm is closer to the true posterior probability density distribution than that by using the UPF algorithm during the mobile robot SLAM.

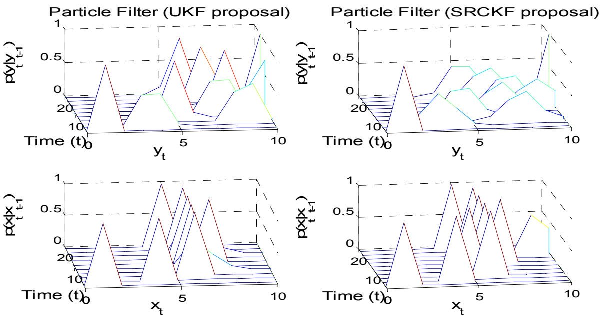

Figure 3. The Posterior Probability Distribution Map of the SRCPF Algorithm and the UPF Algorithm under the Gaussian Noise

The data contrasts of the system noise and the observation noise under the Gaussian white noise condition is shown in the Table 1. From the running time contrast, the running time of the UPF algorithm is the longest. The running time of the SRCPF algorithm is relatively shorter than that of the UPF algorithm. The running time of the EPF algorithm is relatively shorter than that of the SRCPF algorithm. The running time of the PF algorithm is the shortest. Because the SRCPF algorithm, the UPF algorithm and the EPF algorithm join respectively the SRCKF algorithm, the UKF algorithm and the EKF algorithm in the process of generating important density function. During the process of the estimation error contrast, the error of the PF algorithm is the biggest. The error of the SRCPF algorithm is smaller than that of the UPF algorithm. The error of the UPF algorithm is smaller than that of the EKF algorithm. So, the validity of the SRCPF-SLAM algorithm is verified.

Table 1. The Data Contrast of the System Noise and the Observation Noise under the Gaussian White Noise Condition

SLAM	Run time	precision	Gm/t
PF	1.3125s	First-Order Accurate	5.2356
EPF	2.0741s	First-Order Accurate	4.2651
UPF	4.8643s	Second-Order Accurate	3.6952
RCPF	3.5271s	Second-Order Accurate	3.1256

Here, Gm/t is the root mean square error of the map estimation respectively.

5. Conclusion

In this paper, the SRCPF-SLAM algorithm is proposed. This algorithm which is estimating the robot position and updating the landmark information has the more effects than the PF algorithm, the EPF algorithm and the UPF algorithm. The importance density function is the important factor of the particle filters resample. Whether can design reasonably the important density function, it impacts directly on the performance of particle filter. The algorithm fuse the latest measurement information in the stage of the prior distribution updated of the particle filter SLAM. It designs importance density function by SRCKF (Square-root Cubature Kalman Filter) that is more close to the posterior density, and it spreads the square root of state covariance. So, the algorithm ensures the symmetry and the positive semi-definiteness of the covariance matrix and improves numerical estimation precision and stability. And that the SRCKF algorithm decreases the mobile robot SLAM algorithm running time when compared with the UPF algorithm. So, the algorithm has more real-time comparing with the UPF algorithm. The paper considers the operational efficiency, the accuracy and the stability of these algorithms. The SRCPF algorithm is a better way that can improve particle filter algorithm related to the PF algorithm, the EPF algorithm and the UPF algorithm.

Acknowledgements

This work is supported by the Postgraduate technology innovation project Funds of the Qiqihar University, Heilongjiang Province Government, China, under Grant YJSCX2013-030X. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Bailey T, Durrant-Whyte HF. Simultaneous localization and mapping (SLAM): Part II, State of the art. *IEEE Robotics and Automation Magazine.* 2006; 13(3): 108-117.
[2] Thrun S, Liu YF, Koller D, et al. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research.* 2004; 23(7/8): 693-716.
[3] Dissanayake MWMG, Newman P, Clark S, et al. *A solution to the simultaneous location and map building (SLAM) problem*. The University of Sydney. 2006.
[4] Doucet A de Freitas N, Murphy KP, Russell SJ. *Rao-Black wellised particle filtering for dynamic Bayesian net-works*. Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. Stanford, USA: Morgan Kaufmann Publishers. 2000: 176−183.
[5] Montemerlo M, Thrun S, Koller D, Wegbreit B. *Fast SLAM: a factored solution to the simultaneous localization and mapping problem*. Proceedings of the AAAI National Conference on Artificial Intelligence. Edmonton, Canada: Springer. 2002: 1−6.
[6] Montemerlo M, Thrun S, Koller D, Wegbreit B. *Fast SLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges*. Proceedings of the 16th International Joint Conference on Artificial Intelligence. Acapulco, Mexico: Springer. 2003: 1151−1156.
[7] Vander Merwe R, Wan EA. *The sonare-root unscented Kalman filter for state and parameter-estimation*. Proceedings of the IEEE International Conference on Acoustics. Speech and Signal Processing, Piscat away, NJ, USA: IEEE. 2001: 3461-3464.
[8] Ito K, Xiong K. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control.* 2000; 45(5): 910−927.
[9] Hongjian Wang, Guixia Fu, Xinqian Bian, Juan Li. SRCKF Based Simultaneous Localization and Mapping of Mobile Robots. *Robot*. 2013; 35(2): 200-207.
[10] Arasaratnam I, Haykin S. Cubature Kalman filter. *IEEE Trans, on Automatic control.* 2009; 54(6): 1254-1269.
[11] Jin Mu, Yuanli Cai. *Square Root Cubature Kalman Filter Algorithm and Application*. Ordnance Industry Automation. 2011; 30(6): 11-14.
[12] Kang L, Xie WX, Huang JX. Tracking of infrared small target based on unscented particle filtering. *Systems Engineering and Electonics.* 2007; 29(1): 1-4.
[13] Feng Sun, Lijun Tang. Cubature particle filter. *Systems Engineering and Electronics*. 2011; 33(11): 2554-2557.
[14] ZOU Guo hui, JING Zhong liang, HU Hong-tao. A Particle Filter Algorithm Based on Optimizing Combination Resampling. *Journal of Shanghai Jiao tong University.* 2006; 50(7): 1135-1139.
[15] Zuguo Chen, Xuefeng Dai, Laihao Jiang, Chao Yang, Biao Cai. Adaptive Iterated Square-Root Cubature Kalman Filter and Its Application to SLAM of a Mobile Robot. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(12): 7213 – 7221.
[16] Mei Wu, Fujun Pei. Improved Distributed Particle Filter for Simultaneous Localization and Mapping. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(12): 7617 – 7626.