

# The Square Root UKF-SLAM Algorithm Based on the Smallest Proportion of Skewness in Single Line Sampling

MengYuan Chen<sup>1,2</sup>, GuoWei Qin<sup>1</sup> and Tong Xu<sup>1</sup>

<sup>1</sup>Key Lab of Electric Drive and Control of Anhui Province, Anhui Polytechnic University, Wuhu 241000, China

<sup>2</sup>Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei 230027, China

**Abstract.** In view of the distortion in the filter gain matrix calculation as well as the high computational complexity and the nonlocal effect of symmetric sampling that exists in the UKF-SLAM algorithm, the square root UKF-SLAM algorithm based on the smallest proportion of skewness in single line sampling was proposed. According to the mended algorithm, the square root of covariance matrix is brought into iteration operation instead of covariance matrix, moreover, the smallest proportion of skewness in single line sampling is utilized for the optimization of sampling strategy. The results of simulation show that the algorithm can effectively improve the position accuracy in robot as well as the estimation accuracy of feature map. Furthermore, the computational complexity is reduced and the algorithm stability is improved.

## 1 Introduction

In the unknown environment, the robot takes advantage of its own sensors to create feature maps for the environment incrementally. At the same time, the machine corrects its own position that has been created in the map, namely SLAM, a problem about simultaneous localization and mapping in mobile robot. Moreover, a classical algorithm that had been applied to SLAM is the method that dynamic equations and observation equations of robot are expanded by first order Taylor series, and then the Extended Kalman Filter algorithm was used for the optimal estimation of the robot's position and the feature map. This model was first proposed by Smith and Cheeseman [1]. Many scholars optimized and improved it based on EKF-SLAM framework. For example, Lu presented that polar coordinates can be used to compare two adjacent observed values to reduce external interference and improve the algorithm's estimation accuracy as well as robustness [2]; Zhanghai Jiang put forward an improved compression-type EKF-SLAM algorithm to reduce the computational complexity [3]. Zhou Wu proposed a global observation map model to improve the presentation methods [4] of the map. However, the inherent defects, such as the truncation error generated from the linearization procedure and the large amount of calculation from Jacobi matrix, make the EKF-SLAM algorithm unsuited to the large-scale environmental map.

Therefore, Julier et al. pointed out Unscented Kalman Filter(UKF), a new filtering algorithm that can be directly applied to nonlinear systems when they studied how to improve EKF-SLAM. UKF is a nonlinear Gaussian state estimator based on the minimum variance estimation criteria [5]. Similar to the Extended Kalman

Filter algorithm, Gaussian random variables are used to approximate the state distribution. But the difference is that UKF approximate the state distribution through deterministic sampled (Sigma points) obtained from some certainty sampling. These sampling points can better describe the true mean and variance of the Gauss random variable. In addition, its precision approximates to the second-order of Taylor while the EKF can only approximates to the first-order accuracy under the same conditions. Besides, in terms of computational complexity, UKF and EKF belong to the same order and do not need to calculate the Jacobian matrix, so UKF is easier to meet our objectives than the EKF [6]. Furthermore, the follow-up work found that the state covariance matrix of UKF algorithm is easy to lose non-negative qualitative and symmetry due to the rounding errors from the computer in the process of data transmission, which will lead to filter divergence [7]. On the contrary, the square root filter and the strategy that use the square root of the covariance matrix instead of covariance matrix itself can solve this problem perfectly, which has been successfully applied to GPS / DR combination [8] and indoor geolocation [9].

In the first place, this article focuses on MATLAB comparative simulation experiments that simulate the process of SLAM which UKF algorithm and the square root UKF algorithm (SR-UKF) apply to. The result shows that the SR-UKF algorithm can significantly improve the accuracy of the robot's pose estimation. However, in terms of SLAM, a kind of system which need high requirements of real-time, it is particularly important to reduce the computational complexity and improve the computational efficiency [10]. To solve the above problems, the SR-UKF algorithm was given a depth study by changing the sampling strategy and using

the smallest proportion of skewness in single line sampling instead of symmetric sampling, as well as reducing the computational complexity through reducing the number of points of Sigma, which enables the systems towards the direction of better real-time. Therefore, a square root UKF-SLAM algorithm based on the smallest proportion of skewness in single line sampling has been proposed in this paper.

## 2 The probability description about SLAM

It assumes that the state vector of a robot is  $X_k^v = [x_k^v, y_k^v, \phi_k^v]^T$ ; the feature map is  $X_k^m = [m_1, m_2, \dots, m_n]^T$ ; the observation sequence is  $Z_{1:k} = [Z_1, Z_2, \dots, Z_k]^T$ ; the control command is  $u_{1:k} = [u_1, u_2, \dots, u_k]^T$ . Dynamic and observation equation are respectively supposed as  $X_k = f(X_{k-1}, u_k) + w_k$  and  $Z_k = h(X_k) + v_k$ , moreover,  $w_k$  and  $v_k$  are gaussian distribution followed by  $N(0, Q_k)$  and  $N(0, R_k)$ . The mathematical description of SLAM problem is:

$$p(X_k^v, X_k^m | Z_{1:k}, u_{1:k}) \quad (1)$$

$Z_{1:k}$  and  $u_{1:k}$  are known conditions. We calculate posterior probability distribution in formula (1) by Baye filter principle to obtain the optimal state estimation. The whole process is divided into two steps, the first step is prediction, the prior probability distribution at k time is obtained by computational model of robot pose  $p(X_k^v | X_{k-1}^v, u_k)$  and posterior probability distribution at k+1 time:

$$p(X_k^v, X_k^m | Z_{1:k}, u_{1:k}) = \int p(X_k^v | X_{k-1}^v, u_k) p(X_{k-1}^v, X_{k-1}^m | Z_{1:k-1}, u_{1:k-1}) dX_{k-1}^v \quad (2)$$

The second step is observation update,  $Z_k$  is gained

from the observation model of system at k time, and the posterior probability distribution is solved by the prior probability distribution.

$$p(X_k^v, X_k^m | Z_{1:k}, u_{1:k}) = \eta p(Z_k | X_k^v, X_k^m) p(X_k^v, X_k^m | Z_{1:k-1}, u_{1:k}) \quad (3)$$

$\eta$  is the normalization coefficient.

## 3 SR-UKF-SLAM algorithm and the simulation analysis

### 3.1 UKF-SLAM algorithm

UT Transformation as the core, kalman filter as the basic framework, UKF take advantage of certainty sampling to get the posterior state distribution from state estimation. Before filtering, the original state vector is augmented to gauss noise variable:

$$X_k^a = [X_{k-1}^{aT}, w_k^T, v_k^T]^T \quad (4)$$

$$P_k^a = \begin{bmatrix} P_k & 0 & 0 \\ 0 & Q_k & 0 \\ 0 & 0 & R_k \end{bmatrix} \quad (5)$$

It assumes that the mean is  $\hat{X}_{k-1}^a$  and the variance is  $P_{k-1}^a$  at k-1 time. The UKF-SLAM algorithm block diagram was shown in Figure 1. The whole algorithm is divided into three parts, including sigma point, forecast period and update phase calculations. Besides, the calculation of sigma point adopts the symmetrical sampling method.  $2n+1$  Sigma points,  $\xi_i, i=0,1,\dots,2n$  ( $n$  is the system status dimension of  $X_k^a$ ), whose corresponding first and second order metric are  $W_i^m$  and  $W_i^c$  respectively. These sampling points are taken into the model of nonlinear system for transition, which helps to get the statistical properties of the state vector.

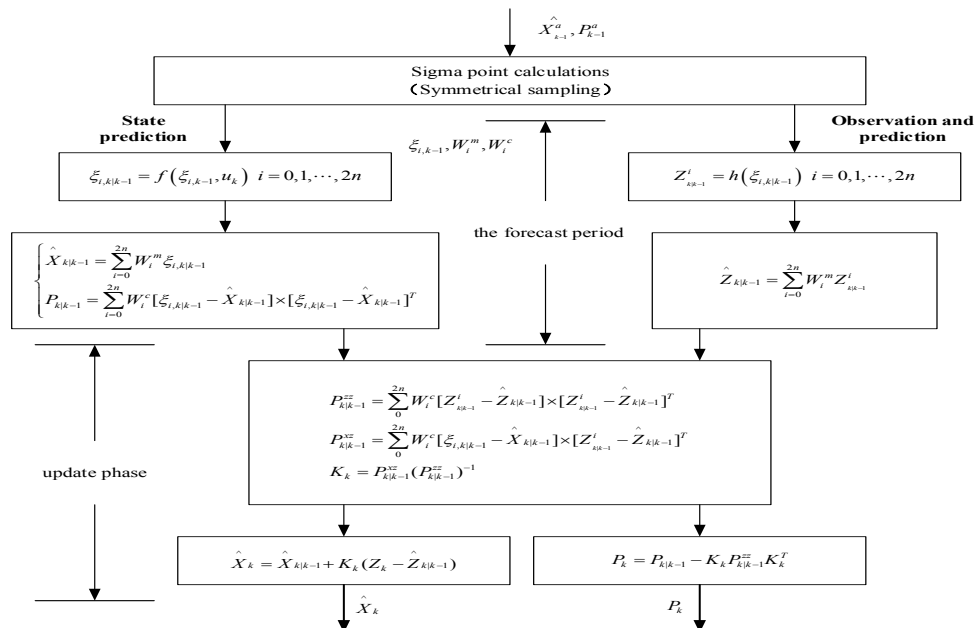


Figure 1. EKF-SLAM algorithm block diagram.

### 3.2 SR-UKF-SLAM algorithm

In the aforesaid filtering model, the rounding errors of calculation are gradually accumulated with the depth of filtering. In addition, the possibility that the filtering error covariance matrix  $P_k$  and the predicting error covariance matrix  $P_{k|k-1}$  lose non-negative qualitative and symmetry will increase, which makes the calculation of filtering gain distorted as well as the filter diverging. Therefore, on the basic of UKF-SLAM algorithm, with the update of covariance square root  $S_k$  instead of covariance  $P_k$  ( $P_k = S_k S_k^T$ ), a UKF-SLAM algorithm based on the square root [11] (SR-UKF-SLAM) is proposed. Furthermore, in the specific process of algorithm, the two calculation methods including QR decomposition and Cholesky factor update are utilized to reduce the computational complexity. As distinguished from the EKF-SLAM algorithm, its optimization process is as follows:

Predict phase:

$$S_{k|k-1} = QR \left\{ \left[ \sqrt{W_{1:2n}^c} \begin{pmatrix} \xi^a \\ \xi_{1:2n, k|k-1} \end{pmatrix} - \hat{X}_{k|k-1} \right] \sqrt{Q_{k-1}} \right\} \quad (6)$$

$$S_{k|k-1}^z = QR \left\{ \left[ \sqrt{W_{1:2n}^c} \begin{pmatrix} Z^i \\ \xi_{1:2n, k|k-1} \end{pmatrix} - \hat{Z}_{k|k-1} \right] \sqrt{R_{k-1}} \right\} \quad (7)$$

Update phase:

$$S_{k|k-1} = cholupdate \left\{ S_{k|k-1}, \xi_{0, k|k-1}^a - \hat{X}_{k|k-1}, W_0^c \right\} \quad (8)$$

$$S_{k|k-1}^z = cholupdate \left\{ S_{k|k-1}^z, Z_{0, k|k-1}^i - \hat{Z}_{k|k-1}, W_0^c \right\} \quad (9)$$

In order to avoid the matrix inverse operation, back substitution was used to deal with filtering gain  $K_k$ . As shown in the formula (10), formula (11)、formula (13) are applied to the iterative update of state vector.

$$K_k = P_{k|k-1}^{xz} / (S_{k|k-1}^z)^T / S_{k|k-1}^z \quad (10)$$

$$\hat{X}_k = \hat{X}_{k|k-1} + K_k (Z_k - \hat{Z}_{k|k-1}) \quad (11)$$

$$U = K_k S_{k|k-1}^z \quad (12)$$

$$S_k = cholupdate \left\{ S_{k|k-1}, U, -1 \right\} \quad (13)$$

### 3.3 Simulation and Analysis

In order to study the effect of the estimation accuracy from square root transition, the SR-UKF-SLAM and UKF-SLAM algorithm were respectively simulated through open source simulation platform developed by Tim Bailey in MATLAB. The computer in the experiment employs Intel Core-i5 dual-core processor with 2.6 GHz basic frequency and 4 GB internal storage. In the experiment, with 250\*200m analog outside circumstance, the robot paraded a circle along 17 path points (“●”) from coordinates (0,0), the running speed of the robot is up to 2 m/s, 35 static feature points (“\*”) were randomly distributed in the environment. The simulation results of two algorithms are shown in Figures 2 and 3, the filament represents the ideal trajectory of the robot, the “dotted line” indicates the

estimated path of the robot, “+” is the position of feature point, “ellipse” is the error range of the feature point estimation. Figure 2 and figure 3 shows that, during the initial period, the calculation error between UKF-SLAM and SR-UKF-SLAM algorithm became relatively small. However, as time goes on, due to the constantly increased new features, there has been an increasingly deviation gap between estimated and ideal trajectory. and the error ellipse from the features map is also significantly higher than the SR-UKF-SLAM algorithm.

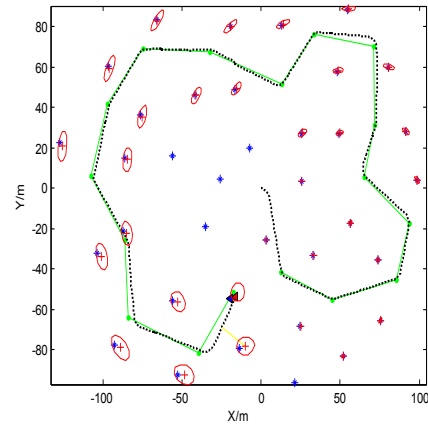


Figure 2. Simulation result of UKF-SLAM.

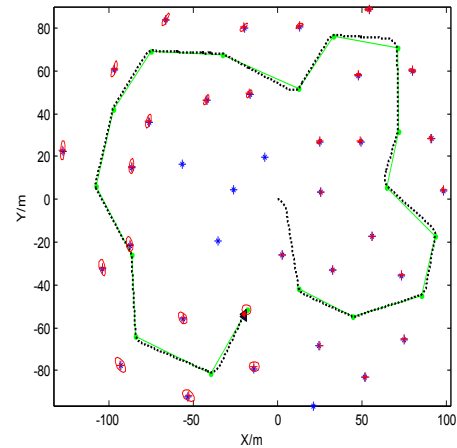
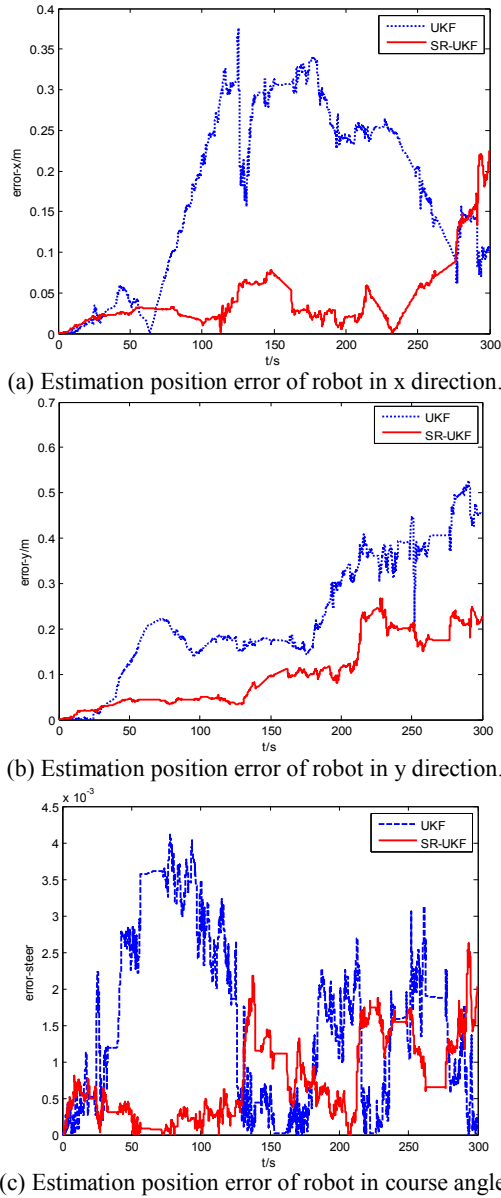


Figure 3. Simulation result of SR-UKF-SLAM.

It took about 330 seconds for robot to explore one lap through average value from multiple experiments. In this paper, the simulation results gained from the first 300s were analysed. (a)、(b)、(c) in figure 4 respectively indicate the calculation accuracy of robot in x and y direction as well as the error comparison of course angle that were obtained from UKF-SLAM UKF-SLAM (as UKF showed in the picture, dotted line) and SR-UKF-SLAM (as SR-UKF showed in the picture, solid line) algorithm. From the picture, the estimating errors of position in robot based under two algorithms became greater, however, in contrast to UKF-SLAM, the estimation error of SR-UKF-SLAM algorithm is relatively stable and always maintain a low level.



**Figure 4.** A comparison of estimation error of UKF-SLAM and SR-UKF-SLAM.

#### 4 Simulation experiment and analysis of SPSR-UKF-SLAM algorithm

Although the SR-UKF-SLAM algorithm improves the estimation accuracy of robot-position and features maps to a certain degree, however, in terms of SLAM which requires hard real-time performance, we hope that the computational complexity can be further reduced through reducing the number of sigma points. In the symmetric sampling, the distance from sigma point to  $\hat{X}_k^a$  ascends with the increasement of state dimension of  $X_k^a$ , which easily causes nonlocal effect and affects the accuracy and stability of the filtering.

The smallest proportion of skewness in single line is a kind of sampling mode that needs minimal points, due to the direct influence on the calculation amount of UKF from the number of sampling points, the strategy is more suitable for the system with higher real-time performance.

According to Julier's analysis in the literature [11], at least  $n+1$  sampling points are required to determine the posterior distribution of the system for the  $n$ -dimensional state vector distribution. Moreover, the study found that the proportion correction can be a good solution to the non-local effects. Therefore, combined with the advantages of the sampling strategy above, a square root UKF-SLAM algorithm based on the smallest proportion of skewness in single line sampling, namely the SPSR-UKF-SLAM algorithm, was proposed in the process of SLAM, Which helps modify the sampling strategy of SR-UKF-SLAM algorithm, its specific sampling process is as follows:

- (1) The weight coefficient of initialization  $0 \leq W_0^m = W_0^c < 1$ .
- (2) The first and second order weight coefficient of sigma sampling points are:

$$W_i^m = \begin{cases} \frac{W_0^m}{\alpha^2} + \left(1 - \frac{1}{\alpha^2}\right), & i=0 \\ \frac{1-W_0^m}{2^n \alpha^2}, & i=1,2 \\ \frac{2^{i-2} W_1^m}{\alpha^2}, & i=3,4,\dots,n+2 \end{cases} \quad (14)$$

$$W_i^c = \begin{cases} W_0^m + (1 + \beta - \alpha^2), & i=0 \\ W_i^m, & i \neq 0 \end{cases} \quad (15)$$

In the equation:  $\alpha$ , as a smaller positive number, is used to control the distance from sigma point to the center point  $\hat{X}_k^a$  with the range of  $[10^{-4}, 1]$ ;  $\beta$  is used to describe a priori information about the distribution of random variables., in the case of a Gaussian distribution, the best approach is taking  $\beta = 2$ .

- (3) The initial vector ( $L = 1$ ) is

$$\xi_0^1 = [0], \xi_1^1 = \left[-\frac{1}{\sqrt{2W_1^m}}\right], \xi_2^1 = \left[\frac{1}{\sqrt{2W_1^m}}\right] \quad (16)$$

According to the state dimension of input  $L = 2, 3, \dots, n$ , the iteration formula of vector is:

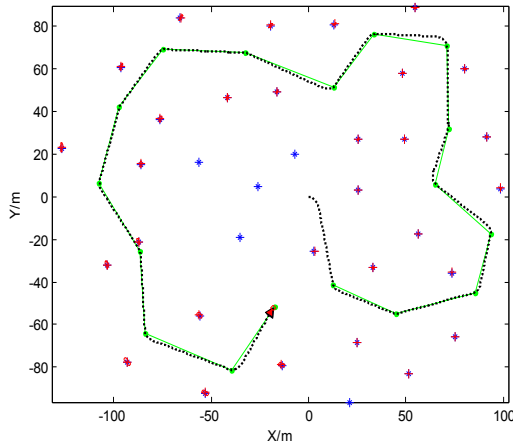
$$\xi_i^L = \begin{cases} \begin{bmatrix} \xi_0^{L-1} \\ 0 \end{bmatrix}, & i=0 \\ \begin{bmatrix} \xi_0^{L-1} \\ 1 \\ -\sqrt{2W_{L+1}^m} \end{bmatrix}, & i=1,2,\dots,L \\ \begin{bmatrix} 0 \\ 1 \\ -\sqrt{2W_{L+1}^m} \end{bmatrix}, & i=L+1 \end{cases} \quad (17)$$

- (4) The sigma points obtained from formula (24) is:

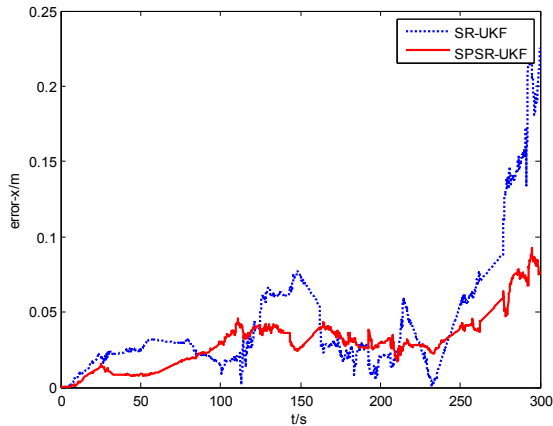
$$\xi_{i,k-1}^a = \hat{X}_{k-1}^a + \alpha S_{k-1} \xi_i^L, i=0,1,2,\dots,L+1 \quad (18)$$

The simulation experiments was carried out on the optimized algorithm in the same simulation environment, the simulation results shown in Figure 5. With respect to the results shown in Figure 2, the estimated trajectory from robot goes a step further approaching the ideal

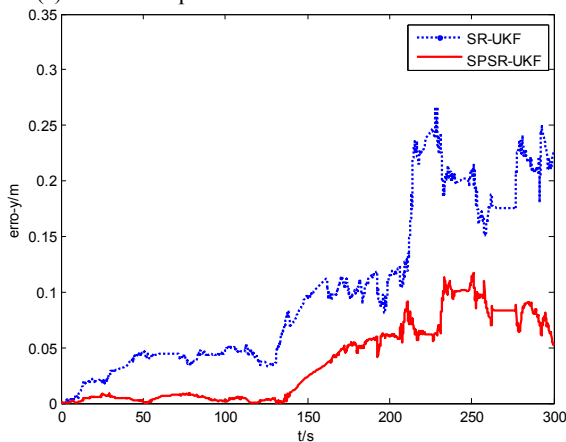
trajectory in the process of SPSR-UKF-SLAM algorithm, and the error ellipse is further reduced. (a), (b), (c) in figure 6 respectively indicate the calculation accuracy of robot in x and y direction as well as the error comparison of course angle that were obtained from SR-UKF-SLAM (as SR-UKF showed in the picture, dotted line) and SPSR-UKF-SLAM (as SPSR-UKF-SLAM showed in the picture, solid line) algorithm. From the picture, SPSR-UKF-SLAM improves the estimation precision of the robot pose state while reducing the computational complexity.



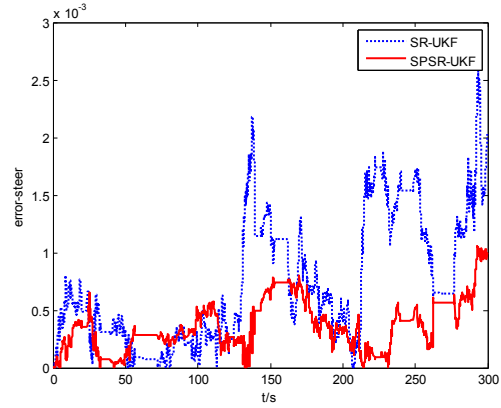
**Figure 5.** The simulation results of SPSR-UKF-SLAM.



(a) Estimation position error of robot in x direction.



(b) Estimation position error of robot in y direction.



(c) Estimation position error of robot in course angle.

**Figure 6.** A comparison of estimation error between SR-UKF-SLAM and SPSR-UKF-SLAM.

In order to compare the performance of the three SLAM algorithms, the average errors of the three algorithms are calculated as shown in table 1. From the data in table 1, The estimation accuracy of robot position from SPSR-UKF-SLAM is the highest in three algorithms. Compared with SR-UKF-SLAM algorithm, the errors of robot-position in X direction, Y direction and the course angle are respectively reduced 19.08%, 29.31% and 50.47% under SPSR-UKF-SLAM algorithm.

**Table 1.** Statistical comparison of algorithm errors.

algorithm	errors		
	x direction /m	y direction /m	course angle /rad
UKF-SLAM	0.1742	0.2284	0.1543
SR-UKF-SLAM	0.0435	0.1020	0.0741
SPSR-UKF-SLAM	0.0352	0.0721	0.0367

## 5 Conclusion

According to the filter divergence from the rounding errors of calculation that exists in EKF-SLAM algorithm, the covariance square root was utilized for update instead of covariance. Furthermore, in term of the problems such as a great quantity of sampling site, the poor real-time performance and the nonlocal effect that easily produced, the sampling schematism was transformed into the one based on smallest proportion of skewness in single line sampling, from which a more optimized algorithm, the square root UKF-SLAM algorithm based on the smallest proportion of skewness in single line sampling, was proposed. Moreover, the MATLAB simulation experiments show that, compared with the UKF-SLAM algorithm, the new algorithm can effectively improve the estimation accuracy of the robot's position and the feature maps with its relatively lower computational complexity and greater stability.



## Acknowledgments

This work was supported by 2018 Anhui Natural Science Foundation (Bionic SLAM study of rat mixed cells with DGSOM neural network) and 2018 Anhui Excellent top-notch talent cultivation project in Colleges and Universities.

## References

1. R Smith, M self, P Cheeseman. Estimating uncertain spatial relationships in robotics [A]. Proc of Conf Uncertainty in Artificial Intelligence[C]. Amsterdam: North -Holland, 1988, 435-461.
2. Taizhi Lv. A new anti-disturbance EKF-SLAM algorithm [J].computer engineering, 2012, 38(21):1-4.
3. Haiqiang Zhang, Lihua Dou, Hao Fang, etc. An improved SLAM algorithm based on compression type EKF [J]. System Simulation, 2009, 21(18):5668-5680.
4. Wu Zhou, Chunxia Zhao, Yaqiang Shen, etc. The study of SLAM based on global observation map model [J]. Robot, 2010, 32(5):647-654.
5. S J Julier, J K Uhlmann. H Durrant-Whyte. A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimation [J]. IEEE Transactions on Automatic Control (S0018-9286), 2000, 45(3):477-482.
6. Xinxi Shi, Chunxia Zhao, Jianhui Guo. SLAM framework algorithm of mobile robot based on PF / EKF UKF [J]. Digital Journal.2009, 8:1865-1868.
7. Merwe R V D, Wan E A, The Squire-Root Unscented Kalman Filter for State and Parameter-Estimation[C].International Conference on Acoustics, Speech, and Signal Processing, Utah,2001.
8. Jing Yang, Nanning Zheng. A GPS/DR Integrated Positioning Algorithm Based on SR-UKF. System Simulation [J].2009, 21(3):721-722,742.
9. Lei An, Guoliang Zhang, Wenjun Tan. The Indoor Kalman filter localization algorithm of Robot based on RSSI [J]. Computer Engineering and Application. 2010, 48(8):230-232.
10. Shurong Li, Pengfei Ni. SLAM algorithm based on the square root of UKF [J]. Computer Engineering and Application, 2011, 47(22):209-212.
11. S J Julier, J K Uhlmann. Reduced sigma point filters for propagation of means and covariances through nonlinear transformation: Proc. of American Control Conf[C]. Jefferson City, 2002: 887-892.