Research Article

A FastSLAM Algorithm Based on Nonlinear Adaptive Square Root Unscented Kalman Filter

Yu-feng Zhang,¹ Qi-xun Zhou,² Ju-zhong Zhang,³ Yi Jiang,³ and Kai Wang⁴

¹School of Electrical and Control Engineering, Xi'an University of Science and Technology, Shaanxi, Xi'an 710054, China

²Key Laboratory of Embedded System and Service Computing, Tongji University, Ministry of Education, Shanghai 201804, China

³713th Institute of China Shipbuilding Industry Corporation, Zhengzhou, China

⁴Naval Representative Office in Zhengzhou Region, Zhengzhou, China

Correspondence should be addressed to Yu-feng Zhang; xkdzhangyufeng@163.com

Received 26 August 2016; Revised 15 December 2016; Accepted 6 February 2017; Published 15 March 2017

Academic Editor: Francisco Valero

Copyright © 2017 Yu-feng Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For fast simultaneous localization and mapping (FastSLAM) problem, to solve the problems of particle degradation, the error introduced by linearization and inconsistency of traditional algorithm, an improved algorithm is described in the paper. In order to improve the accuracy and reliability of algorithm which is applied in the system with lower measurement frequency, a new decomposition strategy is adopted for a posteriori estimation. In proposed decomposition strategy, the problem of solving a 3-dimensional state vector and N 2-dimensional state vectors in traditional FastSLAM algorithm is transformed to the problem of solving N 5-dimensional state vectors. Furthermore, a nonlinear adaptive square root unscented Kalman filter (NASRUKF) is used to replace the particle filter and Kalman filter employed by traditional algorithm to reduce the model linearization error and avoid solving Jacobian matrices. Finally, the proposed algorithm is experimentally verified by vehicle in indoor environment. The results prove that the positioning accuracy of proposed FastSLAM algorithm is less than 1 cm and the azimuth angle error is 0.5 degrees.

1. Introduction

In simultaneous localization and mapping (SLAM), vehicle uses the carried sensors to sense surroundings and uses the sensed information to create environment map on one hand. On the other hand, vehicle uses created map to locate and guide.

There are several methods to deal with SLAM problem, in which EKF-SLAM and FastSLAM are the two most popular methods. EKF-SLAM has some obvious limitations: inconsistency due to errors accumulation introduced by linearization, complex computation to deal with high-dimensional joint covariance, lack of robustness to incorrect data association, and so on [1–3]. FastSLAM employs the Rao-Blackwellized particle filter (RBPF) to estimate position and extended Kalman filter (EKF) to estimate map features. Compared with EKF-SLAM, FastSLAM has a lower complexity and

is more robust regarding the data association problem. Nevertheless, the FastSLAM based on RBPF also suffers from some drawbacks such as particle degeneration, Jacobian Matrix solving and linear processing of nonlinear function [4–6]. To deal with these problems, the SLAM based on square root unscented Kalman filter (SRUKF) is proposed. Instead of approximating the state and measurement transition functions by Taylor series expansion, the unscented transformation employed by SLAM is used to update nonlinear state and measurement functions and the accuracy of the state estimation has been significantly improved [7–9].

The filtering approaches used in all aforementioned SLAM methods can achieve good performance under certain assumptions. However, the assumptions are typically not entirely satisfied in practical applications. Thus, the performance of SLAM algorithm may be downgraded from the theoretical performance, which can potentially lead to

FIGURE 1: Schematic diagram of SLAM problem.

divergence. To prevent divergence and to improve the practicability of SLAM algorithm, the so-called adaptive filtering approach has been used in SLAM algorithm to dynamically adjust the parameters of the supposedly optimum filter based on estimating the unknown parameters for online estimation of motion and by estimating the signal and noise statistics from the available data. The adaptive filter can be realized to adjust important coefficients through fuzzy logic algorithm or neural network algorithm [10, 11]. Another popular method to realize the adaptive filtering is to use estimators [12–14].

To overcome the shortcomings of traditional adaptive filter, we developed a new nonlinear adaptive square root unscented Kalman filter (NASRUKF) which can be used in nonlinear or linear system for multisensory data fusion with uncertain process noise [15, 16]. In [17], the prototype of FastSLAM algorithm based on NASRUKF is described, and the validity is demonstrated by simple preliminary experiment.

In this paper, the improved FastSLAM algorithm based on NASRUKF is described in detail. To verify the accuracy of FastSLAM under the condition of low measurement frequency, the algorithms based on SRUKF and ASRUKF are compared and analyzed. And a complete experiment is designed for further validation.

2. FastSLAM Framework

The idea of FastSLAM algorithm comes from an analysis result of Dynamic Bayesian Network (DBN): if the vehicle pose is fully determined, then the map features are mutually independent. As shown in Figure 1, if the vehicle pose estimation $(\hat{\mathbf{x}}_R(0), \hat{\mathbf{x}}_R(1), \dots, \hat{\mathbf{x}}_R(k))$ is known, the estimations of map features F_1 and F_2 are independent from each other. The input information of FastSLAM includes the control information $\mathbf{u}(0 : k - 1)$ and the measurement information $\mathbf{z}(1 : k)$.

From a probabilistic point of view, SLAM problem is to solve the posterior probability distribution of system state vectors composed of vehicle pose vector $\mathbf{x}_R$ and map feature vector $\mathbf{x}_F$, which can be expressed as follows [18]:

$$p(\mathbf{x}(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \mathbf{x}_R(0)). \quad (1)$$

From the aforementioned DBA analysis result and Bayesian formula, formula (1) can be expressed as follows:

$$\begin{aligned} & p(\mathbf{x}(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \mathbf{x}_R(0)) \\ &= p(\mathbf{x}_R(1 : k), \mathbf{x}_F(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \\ & \mathbf{x}_R(0)) = p(\mathbf{x}_R(1 : k) | \mathbf{z}(1 : k), \mathbf{u}(0 : k - 1), \mathbf{x}_R(0)) \quad (2) \\ & \cdot \prod_{i=1}^N p(\mathbf{x}_{Fi}(1 : k) | \mathbf{x}_R(1 : k), \mathbf{z}(1 : k)). \end{aligned}$$

Formula (2) shows the idea of FastSLAM algorithm that the joint SLAM state estimation can be factored into two independent estimations: vehicle pose component and map feature component. Thus, the problem of solving $(2N + 3)$ dimension state vectors is converted into two parts: a 3-dimension state vector solving of vehicle pose and N 2-dimension state vectors solving of map features.

For traditional FastSLAM algorithm, estimation of vehicle pose is achieved by particle filtering. Each map feature, in each particle, can be estimated using Extended Kalman Filters conditioned on the robot pose of the particle. And the factor weight of particles is calculated to determine the probability that a certain particle enters the final set.

The implementation process of FastSLAM algorithm is as follows. (1) The posterior probability distribution of vehicle pose is estimated using the control information and the motion model. (2) Map features of particles are associated with measurement information via maximum likelihood estimation. (3) According to correlated measurement information of each particle, update the estimations of each map feature and vehicle pose.

3. Optimization of FastSLAM

As the performance of FastSLAM depends on measurement information, the accuracy and reliability of traditional algorithm are downgraded when measurement data are limited by detecting frequency of sensors. In addition, the linearization method increases the estimation error for the reason that the actual system has strong nonlinearity and uncertain noise.

In [17], the basic idea of an modified FastSLAM was introduced simply. In this section, the modified FastSLAM introduced in [17] will be described in detail which is especially appropriate for low speed vehicle system (such as the wheelchair used for the aged). Considering low scanning speed sensors, a new decomposition strategy is designed for the posterior probability distribution. Moreover, an improved a posteriori estimation method is proposed which simultaneously estimates the joint state vectors comprised of vehicle pose and map features.

3.1. Posterior Probability Distribution Decomposition Strategy. The proposed posterior probability distribution decomposition strategy of state vectors is shown as

$$\begin{aligned}
p(\mathbf{x}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)) &= p(\mathbf{x}_R(1:k), \mathbf{x}_F(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)) \\
&= \begin{cases} \prod_{i=1}^N p(\mathbf{x}_{Ri}(k), \mathbf{x}_{Fi}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & (k=1) \\ \prod_{i=1}^N p((\mathbf{x}_R(1:k-1), \mathbf{x}_{Ri}(k)), \mathbf{x}_{Fi}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & (k>1). \end{cases} \quad (3)
\end{aligned}$$

Assuming that there are N particle state estimations $\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \dots, \hat{\mathbf{X}}_N$ and relevant covariance matrices of estimation error $\mathbf{P}_{11}, \mathbf{P}_{22}, \dots, \mathbf{P}_{nn}$, the particle state estimations are irrelevant; that is, $\mathbf{P}_{ij} = 0$ ($i \neq j$). Then optimal estimation of the joint state vectors can be expressed as

$$\hat{\mathbf{X}}_g = \frac{\sum_{i=1}^N \mathbf{P}_{ii}^{-1} \hat{\mathbf{X}}_i}{\sum_{i=1}^N \mathbf{P}_{ii}^{-1}}. \quad (4)$$

The physical meaning of formula (4) is that if the estimation accuracy of $\hat{\mathbf{X}}_i$ is low, its global contribution $\mathbf{P}_{ii}^{-1} \hat{\mathbf{X}}_i$ will be low.

It can be drawn from (3) and (4) that

$$\hat{\mathbf{x}}_R(k) = \frac{\sum_{i=1}^N \mathbf{P}_{Ri}^{-1}(k) \hat{\mathbf{x}}_{Ri}(k)}{\sum_{i=1}^N \mathbf{P}_{Ri}^{-1}(k)}. \quad (5)$$

As shown in (3) and (5), the proposed FastSLAM algorithm converts the high-dimensional joint state estimation into several independent low dimensional joint state estimations. That is, the state vectors solving of $(2N+3)$ dimensions is converted into N 5-dimension state vectors solving.

Compared with the traditional FastSLAM algorithms, the correlation between vehicle pose and map features in low dimension state vectors is considered which is helpful to improve the accuracy and reliability of FastSLAM, especially when one frame data of the sensor can be used to observe at most one map feature which cannot create a new full map feature. This often appears in a vehicle system with lower measuring frequency sensors, such as sonar or infrared sensor, which can only obtain one map feature in a sampling cycle.

Equations (3) and (5) can be simplified as

$$\begin{aligned}
p(\mathbf{x}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)) &= \begin{cases} p(\mathbf{x}_R(1:k), \mathbf{x}_{Foi}(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & \mathbf{z} \text{ from Fo} \\ p(\mathbf{x}_R(1:k) | \mathbf{z}(1:k), \mathbf{u}(0:k-1), \mathbf{x}_R(0)), & \text{others} \end{cases} \quad (6) \\
&\quad (i = 1, 2, \dots, N),
\end{aligned}$$

where Fo represents the created map feature; $\mathbf{x}_{Foi}$ represents the i th state vector with created feature. The meaning of formula (6) is that if the k th measurement data is from created feature, then joint estimation of vehicle pose and the created feature is carried out. If not, the vehicle pose is estimated only and the k th measurement data is saved to create a new map feature.

3.2. NASRUKF. For a nonlinear and discrete system, stochastic sequence is described by

$$\begin{aligned}
x(k) &= f(x(k-1), u(k-1)) + w(k-1), \\
z(k) &= h(x(k)) + v(k), \quad (7)
\end{aligned}$$

where $f(x)$ and $h(x)$ are nonlinear functions of the system; $x(k)$ is the state vector of system; $u(k-1)$ is the control vector applied at time $k-1$; $w(k-1)$ is process noise; $z(k)$ is observation vector; $v(k)$ is the measurement noise.

The sigma points are calculated as

$$\begin{aligned}
\chi(k-1) &= [\hat{\mathbf{x}}(k-1) \quad \hat{\mathbf{x}}(k-1) + \gamma \mathbf{S}(k-1) \quad \hat{\mathbf{x}}(k-1) - \gamma \mathbf{S}(k-1)]. \quad (8)
\end{aligned}$$

The time update equations are the following:

$$\begin{aligned}
\hat{\chi}^*(k) &= \mathbf{f}(\chi(k-1), \mathbf{u}(k-1)), \\
\hat{\mathbf{x}}^-(k) &= \sum_{i=0}^{2L} W_i^{(m)} \hat{\chi}_i^*(k), \\
\mathbf{S}^-(k) &= \text{cholupdate} \left\{ [\mathbf{S}^-(k), \hat{\chi}_0^*(k) - \hat{\mathbf{x}}^-(k), W_0^{(c)}] \right\}. \quad (9)
\end{aligned}$$

$$= qr \left\{ \left[\sqrt{W_1^{(c)}} (\hat{\chi}_{1:2L}^*(k) - \hat{\mathbf{x}}^-(k)) \quad \sqrt{\hat{\mathbf{Q}}(k-1)} \right] \right\},$$

$$\mathbf{S}^-(k) = \text{cholupdate} \left\{ [\mathbf{S}^-(k), \hat{\chi}_0^*(k) - \hat{\mathbf{x}}^-(k), W_0^{(c)}] \right\}.$$

Augment sigma points are

$$\hat{\chi}^-(k) = [\hat{\mathbf{x}}^-(k) \quad \hat{\mathbf{x}}^-(k) + \gamma \mathbf{S}^-(k) \quad \hat{\mathbf{x}}^-(k) - \gamma \mathbf{S}^-(k)] \quad (10)$$

Estimate the square root of the measurement noise matrix:

$$\begin{aligned} \hat{\mathbf{Z}}^-(k) &= \mathbf{h}(\hat{\chi}^-(k)), \\ \hat{\mathbf{z}}^-(k) &= \sum_{i=0}^{2L} W_i^{(m)} \hat{\mathbf{Z}}_i^-(k), \\ \tilde{\mathbf{z}}(k) &= \mathbf{z}(k) - \hat{\mathbf{z}}^-(k), \\ \sqrt{\mathbf{R}^{**}} &= \text{cholupdate} \left\{ \sqrt{1-d(k)} \sqrt{\hat{\mathbf{R}}(k-1)}, |\tilde{\mathbf{z}}(k)|, \right. \\ &\quad \left. d(k) \right\}, \end{aligned} \quad (11)$$

$$\begin{aligned} \sqrt{\mathbf{R}^*} &= \text{cholupdate} \left\{ \sqrt{\mathbf{R}^{**}}, \hat{\mathbf{Z}}_{0:2L}^-(k) - \hat{\mathbf{z}}^-(k), -d(k) \right. \\ &\quad \left. \cdot W_i^{(c)} \right\}, \end{aligned}$$

$$\sqrt{\hat{\mathbf{R}}(k)} = \text{diag} \left\{ \sqrt{\text{diag}(\sqrt{\mathbf{R}^*} \sqrt{\mathbf{R}^{*T}})} \right\}.$$

The measurement update equations are as follows:

$$\begin{aligned} \mathbf{P}_{\mathbf{zx}}(k) &= \sum_{i=0}^{2L} W_i^{(c)} (\hat{\chi}_i^-(k) - \hat{\mathbf{x}}^-(k)) (\hat{\mathbf{Z}}_i^-(k) - \hat{\mathbf{z}}^-(k))^T, \\ \mathbf{S}_z(k) &= qr \left\{ \left[\sqrt{W_1^{(c)}} (\hat{\mathbf{Z}}_{1:2L}^-(k) - \hat{\mathbf{z}}^-(k)) \quad \sqrt{\hat{\mathbf{R}}(k)} \right] \right\}, \\ \mathbf{S}_z(k) &= \text{cholupdate} \left\{ [\mathbf{S}_z(k), \hat{\mathbf{Z}}_0^-(k) - \hat{\mathbf{z}}^-(k), W_0^{(c)}] \right\}, \quad (12) \\ \mathbf{K}(k) &= \frac{(\mathbf{P}_{\mathbf{zx}}(k) / \mathbf{S}_z(k)^T)}{\mathbf{S}_z(k)}, \\ \hat{\mathbf{x}}(k) &= \hat{\mathbf{x}}^-(k) + \mathbf{K}(k) \tilde{\mathbf{z}}(k), \\ \mathbf{U} &= \mathbf{K}(k) \mathbf{S}_z(k), \\ \mathbf{S}(k) &= \text{cholupdate} \{ \mathbf{S}^-(k), \mathbf{U}, -1 \}, \end{aligned}$$

where $\gamma = \sqrt{L+\lambda}$; $d(k) = (1-b)/(1-b^{k+1})$ and b is the forgetting factor, typically $0 < b < 1$; the weights $(W_i^{(m)})$ and $(W_i^{(c)})$ of the mean and covariance are given by

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{L+\lambda}, \\ W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta, \\ W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, \quad i = 1, \dots, 2L, \end{aligned} \quad (13)$$

where $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter. The constant α determines the spread of the sigma points around the mean,

FIGURE 2: Implementation process of FastSLAM algorithm.

which is typically set to a small, positive value. The constant $\kappa \geq 0$ is a secondary scaling parameter. $\beta \geq 0$ is used to incorporate the prior knowledge of the distribution (for Gaussian distributions, $\beta = 2$ is optimal).

4. Implementation of New FastSLAM Algorithm

The proposed FastSLAM algorithm implementation process is shown as in Figure 2.

Initialization. During initialization, the initial pose of vehicle is obtained by the static state estimation method based on the measurement data of sensors. In addition, the initial feature of the map is created using the feature extraction method.

After initialization, the vehicle location and mapping will be calculated and updated based on proposed FastSLAM algorithm in the following steps.

Vehicle Pose Estimation. The vehicle pose is estimated using ASRUKF.

Data Association. According to (14), (15), and (16), data association of the measurement data at time k is dealt with to determine whether it is a created map feature. If the measurement data is a created map feature, algorithm turns to vehicle pose and map feature updating; otherwise, the data shall be stored as new feature data, and algorithm turns to new map feature creating.

Vehicle Pose and Map Feature Updating. According to [17], this step consists of two cases. The first one is that the measurement data $(\rho(k), \varphi(k))$ matches the created linear feature. The second case is that the measurement data matches the created arc feature.

(1) The Measurement Data Matches the Created Linear Feature. If the line segment is not parallel to Y_I -axis, that is, $y = a_j x + b_j$, it can be defined that $\mathbf{x}_j(k) = [x(k), y(k), \theta(k), a_j(k), b_j(k)]^T$ is the state variable of the nonlinear discrete system described as

$$\begin{aligned}
\mathbf{x}_j(k) &= \mathbf{x}_j(k-1) + \mathbf{w}_j(k-1), \\
z_j(k) &= \rho(k) = \frac{d_s}{|\cos \theta_j|} = \frac{d_s}{|\cos(\alpha_j - \theta(k) - \varphi(k))|} \\
&= \left| \frac{a_j(k) \cdot (x(k) - Ls \cdot \cos \theta(k)) - (y(k) - Ls \cdot \sin \theta(k)) + b_j(k)}{-a_j(k) \cos(\theta(k) + \varphi(k)) + \sin(\theta(k) + \varphi(k))} \right| + v_j(k).
\end{aligned} \tag{14}$$

If the line segment is parallel to Y_I -axis, that is, $x = c_j$, $\mathbf{x}_j(k) = [x(k), y(k), \theta(k), c_j(k)]^T$ can be defined as the state variable of the nonlinear discrete system, which is described by

$$\begin{aligned}
\mathbf{x}_j(k) &= \mathbf{x}_j(k-1) + \mathbf{w}_j(k-1), \\
z_j(k) &= \rho(k) \\
&= \left| \frac{(x(k) - Ls \cdot \cos \theta(k)) - c_j(k)}{\cos(\theta(k) + \varphi(k))} \right| + v_j(k),
\end{aligned} \tag{15}$$

where $\mathbf{w}_j$ and v_j have the statistical property of zero mean, mutual independence, and Gaussian distribution. The covariance matrix of $\mathbf{w}_j$ and v_j is $\mathbf{Q}_j$ and R_j , respectively.

(2) *The Measurement Data Matches the Created Arc Feature.* It can be defined that $\mathbf{x}_{cj}(k) = [x(k), y(k), \theta(k), x_{cj}(k), y_{cj}(k), R_{cj}(k)]^T$ is the state variable of the nonlinear discrete system described as

$$\begin{aligned}
\mathbf{x}_{cj}(k) &= \mathbf{x}_{cj}(k-1) + \mathbf{w}_{cj}(k-1), \\
z_{cj}(k) &= \rho(k) \\
&= d_s |\cos \theta_j| \pm \sqrt{d_s^2 \cos^2 \theta_j - (d_s^2 - R_{cj}^2(k))} \\
&\quad + v_{cj}(k),
\end{aligned} \tag{16}$$

where

$$d_s = \sqrt{(x(k) - Ls \cos \theta(k) - x_{cj}(k))^2 + (y(k) - Ls \sin \theta(k) - y_{cj}(k))^2}. \tag{17}$$

New Map Feature Creating. If the number of new feature data exceeds threshold value N_{new} , a new map feature should be created.

5. Experimental Results

In order to evaluate the performance of proposed FastSLAM algorithm, experimental results are shown in this section. The test environment is built as Figure 3. The main parameters are depicted as follows: $Ls = 180$ mm, the length threshold of linear segment is $L_{\min} = 500$ mm, the distance threshold between point and line is $d_{\max} = 80$ mm, and the threshold of average vector distance is $md_{\max} = 30$ mm.

As shown in Figure 3, CPU Motion Controller implements the proposed FastSLAM algorithm and uploads the results to the computer via the receiving node R_1 and the transmitting node T_1 for real-time display.

The whole experiment process is divided into the static state estimation and the dynamic update.

5.1. The Static State Estimation. The surroundings are scanned by the sensors of vehicle under stationary state. The scanned data set is clustered by adaptive breakpoint detector. Then the scanned data set is segmented:

- (1) First, the point C is detected as the maximum distance from line AB which exceeds the threshold. Thus, the continuous data set AB is divided into two subsets of AC and CB. Then, the point D is detected as the maximum distance from line AC which exceeds the threshold and the data set AC is divided into two subsets of AD and DC. Similarly, the data set CB is divided into two subsets of CE and EB and the data set EB is divided into two subsets of EF and FB, as shown in Figure 4(a). At last, the distance between point C and line DE is detected to be less than the threshold. Thus, the subsets DC and CE are combined into one data set DE.
- (2) Second, the lengths of AD, DE, EF, and FB are calculated, which exceed the line threshold. Thus, the subsets AD, DE, EF, and FB are processed as line segment. Then the subsets FB are analyzed by most similar algorithm which belongs to the arc segment, as shown in Figure 4(b).

5.2. Dynamic Update

(1) *Evaluation of Vehicle Pose Estimation Accuracy.* The real-time pose is estimated using ASRKF and SRKF, respectively, when the vehicle travels along the desired trajectory. The

FIGURE 3: Schematic diagram of test data transmission.

FIGURE 4: The static state estimation of map features.

statistic characteristic of process noise is unknown. The data measurement period is 25 ms.

The result of experiment is shown as in Figure 5. It is observed from Figure 5 that the errors of vehicle state (xy and θ) estimated by NASRUKF are significantly smaller than the errors estimated by SRUKF. After 4 seconds, as the identification of process noise statistical feature, the conclusions are more obvious. (1) From Figure 5(a), the estimated error of x - y plane calculated by ASRUKF is within 1 cm, while the estimated error calculated by SRUKF is within 5 cm. (2)

From Figure 5(b), the estimated error of azimuth calculated by ASRUKF is within 1.15° (0.02 rad), while the estimated error calculated by SRUKF is almost 5.16° (0.09 rad).

(2) *Evaluation of Map Creating Accuracy.* When vehicle slowly moves along the line, the vehicle track and environmental map feature are dynamically updated. The test results are shown in Figure 6 and Table 1.

Note. Full lines in the figure (real map, real track, and real theta) denote the true value of actual state. Short dash

FIGURE 5: Results of vehicle pose estimation.

TABLE 1: Error in created environmental features.

Feature	Line L_{AD}		Line L_{DE}		Line L_{EF}		Arc FB	
	Slope	Intercept	Slope	Intercept	Slope	Intercept	Center	Radius
Error	0.0616	-10.6518 mm	-0.0292	7.8651 mm	0.0247	-15.5775 mm	15.5822 mm	-13.3833 mm

Feature	Break point A	Corner point D	Corner point E	Corner point F
	21.7072 mm	20.3451 mm	18.5317 mm	31.0601 mm

lines (estimate map) denote the map feature estimated with the improved FastSLAM algorithm. Long dash lines (estimate track and estimate theta) denote the state estimation or estimation error using the improved FastSLAM algorithm.

As shown in Figure 6 and Table 1, the localization accuracy based on the improved FastSLAM algorithm is within 1 cm and the azimuthal error is approximately 0.5° . The position error of the corner point in the map feature is approximately 3 cm. The errors in both the center position

FIGURE 6: Results of SLAM dynamic update.

and radius of the arc are within 2 cm. The slope error of line segment is within 0.07. The intercept error is within 2 cm. It is demonstrated by the test results that the proposed FastSLAM algorithm is effective.

6. Conclusions

An improved FastSLAM algorithm based on NASRUKF is proposed in paper. To improve the accuracy and reliability

of the FastSLAM limited by measuring frequency, a new posterior probability distribution decomposition strategy is proposed. And the vehicle pose and map features are estimated using NASRUKF algorithm, which overcomes the limitations of the traditional FastSLAM algorithm (e.g., solutions to Jacobian matrix are needed; failure to meet consistency conditions and occurrence of “degeneration of particle set”). The adaptive ability of filters is improved while SLAM accuracy is maintained, which provides an

effective way to solve SLAM problem. The effectiveness of the proposed FastSLAM algorithm is verified by indoor experiment results.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors would like to express their great appreciation to the National Natural Science Foundation of China (no. 51307137), the doctoral scientific research foundation of XUST (no. 2016018), the Fundamental Research Funds of Xi'an University of Science & Technology (no. 201317), and the open fund project of the key laboratory for embedded system and service computing (Tongji University) of the education ministry of China (no. ESSCKF 2016-05).

References

- [1] M. Cugliari and F. Martinelli, "A FastSLAM algorithm based on the unscented filtering with adaptive selective resampling," in *Field and Service Robotics*, vol. 42 of *Springer Tracts in Advanced Robotics*, pp. 359–368, Springer, Berlin, Germany, 2008.
- [2] D. Scaramuzza, R. Siegwart, and A. Martinelli, "A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics," *International Journal of Robotics Research*, vol. 28, no. 2, pp. 149–171, 2009.
- [3] Y. Song, Y. Song, and Q. Li, "Robust iterated sigma point FastSLAM algorithm for mobile robot simultaneous localization and mapping," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 24, no. 4, pp. 693–700, 2011.
- [4] Z. Kurt-Yavuz and S. Yavuz, "A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms," in *Proceedings of the IEEE 16th International Conference on Intelligent Engineering Systems (INES '12)*, pp. 37–43, June 2012.
- [5] A. Monjazeb, J. Z. Sasiadek, and D. Neculescu, "Autonomous navigation among large number of nearby landmarks using FastSLAM and EKF-SLAM—a comparative study," in *Proceedings of the 16th International Conference on Methods and Models in Automation and Robotics (MMAR '11)*, pp. 369–374, IEEE, Międzyzdroje, Poland, August 2011.
- [6] D. Trivun, E. Salaka, D. Osmankovic, J. Velagic, and N. Osmic, "Active SLAM-based algorithm for autonomous exploration with mobile robot," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '15)*, pp. 74–79, March 2015.
- [7] Y. Song, Q. Li, Y. Kang, and D. Yan, "Effective cubature FastSLAM: SLAM with Rao-Blackwellized particle filter and cubature rule for Gaussian weighted integral," *Advanced Robotics*, vol. 27, no. 17, pp. 1301–1312, 2013.
- [8] H. Ankishan, E. O. Tartan, and F. Ari, "Square root unscented based FastSlam optimized by particle swarm optimization passive congregation," in *Proceedings of the 10th IEEE International Conference on Mechatronics and Automation (ICMA '13)*, pp. 469–475, August 2013.
- [9] R. Havangi, H. D. Taghirad, M. A. Nekoui, and M. Teshnehlab, "A square root unscented fastSLAM with improved proposal distribution and resampling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2334–2345, 2014.
- [10] H. Du, Y. Hao, and Y. Zhao, "SLAM algorithm based on fuzzy adaptive Kalman filter," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 40, no. 1, pp. 58–62, 2012.
- [11] Q.-L. Li, Y. Song, and Z.-G. Hou, "Neural network based FastSLAM for autonomous robots in unknown environments," *Neurocomputing*, vol. 165, pp. 99–110, 2015.
- [12] H. Wang, G. Fu, J. Li, Z. Yan, and X. Bian, "An adaptive UKF based SLAM method for unmanned underwater vehicle," *Mathematical Problems in Engineering*, vol. 2013, Article ID 605981, 12 pages, 2013.
- [13] A. Torres-González, J. M. Ramiro-de Dios, and A. Ollero, "An adaptive scheme for robot localization and mapping with dynamically configurable inter-beacon range measurements," *Sensors (Switzerland)*, vol. 14, no. 5, pp. 7684–7710, 2014.
- [14] D. Liu, J.-M. Duan, and H.-X. Yu, "FastSLAM algorithm based on adaptive fading extended Kalman filter," *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, vol. 38, no. 3, pp. 644–651, 2016.
- [15] Z. Yong, Z. Yufeng, and Z. Juzhong, "A new adaptive square-root unscented kalman filter for nonlinear systems," *Applied Mechanics and Materials*, vol. 300–301, pp. 623–626, 2013.
- [16] Y. Zhou, C. Zhang, Y. Zhang, and J. Zhang, "A new adaptive square-root unscented kalman filter for nonlinear systems with additive noise," *International Journal of Aerospace Engineering*, vol. 2015, Article ID 381478, 9 pages, 2015.
- [17] J. Zhang, Y. Jiang, and K. Wang, "A modified FastSLAM for an autonomous mobile robot," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 1755–1759, Heilongjiang, China, August 2016.
- [18] J. C. Prieto, A. R. Jimenez, J. Guevara et al., "Performance evaluation of 3D-LOCUS advanced acoustic LPS," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2385–2395, 2009.