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

Abstract

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

\[
p(x(1:k) | z(1:k), u(0:k-1), x_R(0))
\]

\[
= p(x_R(1:k), x_F(1:k) | z(1:k), u(0:k-1),
\]

\[
x_R(0)) = p(x_R(1:k) | z(1:k), u(0:k-1), x_R(0))
\]

\[
\prod_{i=1}^{N} p(x_F(i:1:k) | x_R(1:k), z(1:k)).
\]

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 a 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.


The proposed posterior probability distribution decomposition strategy of state vectors is shown as
\[
p(x(1:k) | z(1:k), u(0:k-1), x_R(0)) = p(x_R(1:k), x_{idl}(1:k) | z(1:k), u(0:k-1), x_R(0)) \]
\[
= \begin{cases} 
\prod_{i=1}^{N} p(x_R(k), x_{idl}(1:k) | z(1:k), u(0:k-1), x_R(0)), & (k = 1) \\
\prod_{i=1}^{N} p((x_R(1:k-1), x_{Ri}(k)), x_{idl}(1:k) | z(1:k), u(0:k-1), x_R(0)), & (k > 1) 
\end{cases}
\]

Assuming that there are \(N\) particle state estimations \(\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n\) and relevant covariance matrices of estimation error \(P_{i1}, P_{22}, \ldots, P_{mm}\), the particle state estimations are irrelevant; that is, \(P_{ij} = 0 (i \neq j)\). Then optimal estimation of the joint state vectors can be expressed as

\[
\tilde{X}_j = \frac{\sum_{i=1}^{N} P_{ij}^{-1} \tilde{X}_i}{\sum_{i=1}^{N} P_{ij}^{-1}} \tag{4}
\]

The physical meaning of formula (4) is that if the estimation accuracy of \(\tilde{X}_i\) is low, its global contribution \(P_{ij}^{-1}\) will be low.

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

\[
\tilde{x}_j(k) = \frac{\sum_{i=1}^{N} P_{ji}^{-1}(k) \tilde{X}_i(k)}{\sum_{i=1}^{N} P_{ji}^{-1}(k)} \tag{5}
\]

where \(Fo\) represents the created map feature; \(x_{idl}\) 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

\[
x(k) = f(x(k-1), u(k-1)) + w(k-1),
\]
\[
z(k) = h(x(k)) + v(k),
\]

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.

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

\[
p(x(1:k) | z(1:k), u(0:k-1), x_R(0)) = \begin{cases} 
p(x_R(1:k), x_{idl}(1:k) | z(1:k), u(0:k-1), x_R(0)), & z from Fo 
p(x_R(1:k) | z(1:k), u(0:k-1), x_R(0)), & others \end{cases} \tag{6}
\]

The physical meaning of formula (7) is that if the creation map feature \((z(k) = \tilde{x}_j(k))\), 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.

The sigma points are calculated as

\[
\chi(k-1) = [\tilde{x}(k-1) \quad \tilde{x}(k-1) + \gamma S(k-1) \quad \tilde{x}(k-1) - \gamma S(k-1)] \tag{8}
\]

The time update equations are the following:

\[
\tilde{x}^* (k) = f(\chi(k-1), u(k-1)),
\]
\[
\tilde{x}^- (k) = \sum_{i=0}^{2L} W_i^{(m)} \tilde{x}^* (k),
\]
\[
S^- (k) = qr \left\{ \left[ \sqrt{W_i^{(c)}} (\tilde{x}^- (k) - \tilde{x}^- (k)) \right] \sqrt{Q(k-1)} \right\},
\]
\[
S^- (k) = \text{cholupdate} \left\{ \left[ S^- (k), \tilde{x}^- (k), S^- (k), W_0^{(e)} \right] \right\}. \tag{9}
\]
Augment sigma points are
\[
\hat{x}^- (k) = [\hat{x}^- (k), \bar{x}^- (k) + \gamma \bar{S}^- (k), \bar{x}^- (k) - \gamma \bar{S}^- (k)].
\] (10)

Estimate the square root of the measurement noise matrix:
\[
\hat{Z}^- (k) = h (\hat{x}^- (k)),
\]
\[
\bar{x}^- (k) = \sum_{i=0}^{2L} W^{(m)}_i Z^- (k),
\]
\[
\tilde{z} (k) = z (k) - \bar{x}^- (k),
\]
\[
\sqrt{R^{xx}} = \text{cholupdate} \left\{ \sqrt{1 - d (k)} \sqrt{R (k - 1)}, |\tilde{z} (k)|, d (k) \right\},
\]
\[
\sqrt{R^x} = \text{cholupdate} \left\{ \sqrt{R^{xx}}, \tilde{Z}_{0:2L} (k) - \bar{x}^- (k), -d (k), W^{(c)}_i \right\},
\]
\[
\sqrt{R} (k) = \text{diag} \left\{ \sqrt{R^x}, \sqrt{R^x}^T \right\}.
\] (11)

The measurement update equations are as follows:
\[
P_{zz} (k) = \sum_{i=0}^{2L} W^{(c)}_i \left[ \tilde{x}^- (k) - \hat{x}^- (k) \right] \left[ \tilde{Z}^- (k) - \bar{x}^- (k) \right]^T,
\]
\[
\begin{align*}
S_{z} (k) &= q r \left\{ \left[ \sqrt{W^{(c)}_i} \left( \tilde{Z}_{0:2L} (k) - \bar{x}^- (k) \right) \right] \sqrt{R} (k) \right\}, \\
S_{z} (k) &= \text{cholupdate} \left\{ [S_{z} (k), \tilde{Z}_{0} (k) - \bar{x}^- (k), W^{(c)}_0] \right\}, \\
K (k) &= \left( P_{zz} (k) / S_{z} (k)^T \right) S_{z} (k), \\
\hat{x} (k) &= \hat{x}^- (k) + K (k) \tilde{z} (k), \\
U &= K (k) S_{z} (k), \\
S (k) &= \text{cholupdate} \left\{ S^- (k), U, -1 \right\},
\end{align*}
\] (12)

where \( d (k) = (1 - b)/(1 - b^{L+1}) \) and \( b \) is the forgetting factor, typically \( 0 < b < 1 \); the weights \( W^{(m)}_i \) and \( W^{(c)}_i \) of the mean and covariance are given by
\[
\begin{align*}
W^{(m)}_0 &= \frac{\lambda}{L + \lambda}, \\
W^{(c)}_0 &= \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta, \\
W^{(m)}_i &= W^{(c)}_i = \frac{1}{2 (L + \lambda)}, \quad i = 1, \ldots, 2L,
\end{align*}
\] (13)

which is typically set to a small, positive value. The constant \( \kappa \) is a secondary scaling parameter. \( \beta \) 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 \( Y_1 \)-axis, that is, \( y = a_j x + b_j \), it can be defined that \( x_1 (k) = [x (k), y (k), \theta (k), a_j (k), b_j (k)]^T \) is the state variable of the nonlinear discrete system described as

![Figure 2: Implementation process of FastSLAM algorithm.](image-url)
\[ x_j(k) = x_j(k-1) + w_j(k-1), \]

\[ z_j(k) = \rho(k) = \frac{d_S}{\cos \theta_j} = \frac{d_S}{\cos (\alpha_j - \theta_j(k) - \varphi(k))} \]

\[ = \frac{a_j(k) \cdot (x(k) - L_s \cdot \cos \theta(k)) - (y(k) - L_s \cdot \sin \theta(k) + b_j(k))}{-a_j(k) \cos (\theta(k) + \varphi(k)) + \sin (\theta(k) + \varphi(k))} + v_j(k). \] (14)

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

\[ x_j(k) = x_j(k-1) + w_j(k-1), \]

\[ z_j(k) = \rho(k) = \frac{d_S}{\cos \theta_j} = \frac{d_S}{\cos (\alpha_j - \theta_j(k) - \varphi(k))} \]

\[ = \frac{(x(k) - L_s \cdot \cos \theta(k)) - c_j(k)}{\cos (\theta(k) + \varphi(k))} + v_j(k), \] (15)

where \( w_j \) and \( v_j \) have the statistical property of zero mean, mutual independence, and Gaussian distribution. The covariance matrix of \( w_j \) and \( v_j \) is \( Q_j \) and \( R_j \), respectively.

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: \( L_s = 180 \text{ mm} \), the length threshold of linear segment is \( L_{\text{min}} = 500 \text{ mm} \), the distance threshold between point and line is \( d_{\text{max}} = 80 \text{ mm} \), and the threshold of average vector distance is \( m_{d_{\text{max}}} = 30 \text{ mm} \).

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:

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

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

\[ x_j(k) = x_j(k-1) + w_{cj}(k-1), \]

\[ z_{cj}(k) = \rho(k) = \frac{d_S}{\cos \theta_j} \pm \sqrt{d_S^2 \cos^2 \theta_j - \left( \frac{d_S^2 - R_{cj}(k)}{2} \right)^2} + v_{cj}(k), \] (16)

where

\[ d_S = \sqrt{(x(k) - L_s \cos \theta(k) - x_{cj}(k))^2 + (y(k) - L_s \sin \theta(k) - y_{cj}(k))^2}. \] (17)

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
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 \((x, y)\) 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
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
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 for the National Natural Science Foundation of China (no. 51307137), the doctoral scientific research foundation of Xi’an University of Science & Technology (no. 2016018), the Fundamental Research Funds of the education ministry of China (no. ESSCKF 2016-05).

**References**


