J. Shanghai Jiao Tong Univ. (Sci.), 2020, 25(1): 76-87
https://doi.org/10.1007/s12204-019-2146-7

Robust Variational Bayesian Adaptive Cubature Kalman Filtering Algorithm for Simultaneous Localization and Mapping with Heavy-Tailed Noise

ZHANG Zhuqing[1] (张铸青), *DONG Peng*[1*] (董 鹏), *TUO Hongya*[1] (庹红娅),
LIU Guangjun[2] (刘光军), *JIA He*[1] (贾 鹤)
(1. School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China;
2. Department of Aerospace Engineering, Ryerson University, Toronto, Canada)

Abstract: Simultaneous localization and mapping (SLAM) has been applied across a wide range of areas from robotics to automatic pilot. Most of the SLAM algorithms are based on the assumption that the noise is time-invariant Gaussian distribution. In some cases, this assumption no longer holds and the performance of the traditional SLAM algorithms declines. In this paper, we present a robust SLAM algorithm based on variational Bayes method by modelling the observation noise as inverse-Wishart distribution with "harmonic mean". Besides, cubature integration is utilized to solve the problem of nonlinear system. The proposed algorithm can effectively solve the problem of filtering divergence for traditional filtering algorithm when suffering the time-variant observation noise, especially for heavy-tailed noise. To validate the algorithm, we compare it with other traditional filtering algorithms. The results show the effectiveness of the algorithm.

Key words: simultaneous localization and mapping (SLAM), variational Bayesian (VB), heavy-tailed noise, robust estimation
CLC number: TP 242 **Document code:** A

0 Introduction

Simultaneous localization and mapping (SLAM) comprises the simultaneous estimation of the state of a robot equipped with on-board sensors and the construction of a model (map) of the environment that the sensors are perceiving[1-3]. In a simple instance, the robot state is described by its pose (position and orientation), although other quantities may be included in the state, such as robot velocity, sensor biases, and calibration parameters. On the other hand, the map, a representation of aspects of interest (e.g., position of landmarks, and obstacles), describes the environment in which the robot operates. Moreover, SLAM is also fundamental building blocks for many emerging technologies from autonomous cars and UAVs to virtual and augmented reality.

Over the past decades, many researchers have devoted to solving this problem and many achievements have been made. From the theoretical level, SLAM could be regarded as a solved problem[1]. However, there remain various issues when it comes to practical realization. According to different model formulations, there are two major techniques to give the "solution" of the SLAM problem, which are filter-based[4-10] and nonlinear-optimization-based solutions[11-13].

Filter-based solution has the advantages of high computational efficiency, compared with nonlinear-optimization-based solution. For most cases, landmarks are also considered as the components of state vector[1]. The dimension of the state vector can be very large with the increase of the running time, which leads to high computation. For this reason, elaborate skills are necessary to filtrate and manage landmarks[2].

The first applicable SLAM system was developed by Smith et al.[14], where they employed an extended Kalman filter (EKF) framework. Williams et al.[15] modelled robotic states and landmarks by a vector which is multi-dimensional normal distribution. Because the formulation of EKF is a first-order approximation for nonlinear function, it induces large errors when system is strongly nonlinear. Moreover, as mentioned before, in complicated environment there is too much information of landmarks added into state vector and covariance matrix, which can lead to high computation. In order to address the drawbacks of EKF,

Received date: 2018-12-06
Foundation item: the National Natural Science Foundation of China (No. 61803260)
*****E-mail:** dongpengkty@sjtu.edu.cn

particle filter (PF) SLAM system was proposed to offset large error caused by linearization[16]. However, PF requires a huge amount of computation, which is hard to meet on-line requirement. Inspired by PF, unscented Kalman filter (UKF) SLAM was developed by Shojaie and Shahri[6] and was improved by Wang et al[7]. UKF as an unscented-transform-based nonlinear filter method exploits the deterministic sampling strategy to approximate nonlinear distribution. Compared with EKF, UKF can realize 3-order approximation with competitive computation. However, with the increase of the system dimensions, the distance between the sampling point and the central point becomes increasingly distant, and the unscented transformation (UT) generates non-local sampling effect, which may lead to the instability and divergence of the filtering data[17]. Following the similar spirit of UKF, cubature Kalman filter (CKF) was proposed and applied to SLAM[18]. According to the volume criterion, CKF selects a group of samples with the same weight and approximates the Gaussian distribution by weighting after the nonlinear changes. Compared with UKF, CKF has fewer sampling points and can solve the issue of numerical instability and accuracy reduction presented in UKF when the systems are high-dimensional.

The methods mentioned above are all based on the assumption that the distribution of observation noise is Gaussian and the related parameters are invariant. However, when the Gaussian assumption does not hold, the estimates are very likely to be misleading and even induce wrong results about the system. Especially for the systems (e.g. SLAM) that highly rely on sensor data, they are sensitive to outliers, and EKF-based solution can be vulnerable to false data associations that can lead to the estimates to diverge[19]. The lightweight tails of Gaussian distribution, which can eliminate the possibility of incorrect measurement, are regarded as the root of this problem, because in many cases the noise distributions are heavy-tailed. In order to solve this issue, various methods aiming to improve the robustness of the Kalman filter in the case of non-Gaussian noise and many outliers were proposed[20-21]. Lately, the methods based on variational Bayesian (VB) approximation, which possess the advantages of low computation and good robustness, were applied to filter problems[22-27]. Agamennoni et al.[28] proposed an approach to extending the applicability of state-space models to a wider range of noise distributions without losing the computational advantages of the associated algorithms. Using linear VB, Xu[29] iteratively updated the map features and noise variance in SLAM. Zhang et al.[30] proposed an SLAM algorithm based on VB adaptive cubature Kalman filter (VB-ACKF) to solve the issue that the observed noise parameters are changed with time.

In this paper, a robust VB adaptive cubature Kalman filter (RVB-ACKF) algorithm is proposed to deal with the time-variant noise which is heavy-tailed. To handle the problem caused by heavy-tailed noise, we model the measurement noise covariance as inverse-Wishart distribution. Unlike standard inverse-Wishart distribution[30], here we make some adaptation with "harmonic mean". As a result, the measurement likelihood is a Student-t distribution with the heavy-tailed character. Thus, we must estimate both the coupled state and measurement noise covariance. We decouple the posterior of state and noise covariance by VB method. Since the system of SLAM is nonlinear, the cubature integration is applied to solve this problem. Through the algorithm we propose, SLAM system can deal with non-Gaussian-heavy-tailed noise with great accuracy, whereas in simulation part, many algorithms perform badly in this kind of case.

1 Robust State-Space SLAM Model

1.1 Description of Probabilistic SLAM

The classic SLAM model can be formulated by

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k, \tag{1}$$

$$\boldsymbol{z}_{k,j} = h(\boldsymbol{m}_j, \boldsymbol{x}_k) + \boldsymbol{v}_{k,j}, \tag{2}$$

where $f(\cdot,\cdot)$ represents kinematic equation, $h(\cdot,\cdot)$ represents observation equation, $\boldsymbol{x}_k$ is the robotic state at the time k, $\boldsymbol{u}_k$ is the control variable at the time k, $\boldsymbol{w}_k$ is the control noise at the time k, $\boldsymbol{z}_{k,j}$ is the measurement of the jth landmark at the time k, $\boldsymbol{m}_j = [x_j \;\; y_j]^{\mathrm{T}}$ is the jth landmark location, and $\boldsymbol{v}_{k,j}$ is the observation noise of the jth landmark at the time k.

From the view of probability, SLAM model with the control and measurement variables $\boldsymbol{u}$ and $\boldsymbol{z}$ can be adapted to the following state distribution:

$$P(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{u}), \tag{3}$$

where $\boldsymbol{X}$ is state variable, among which $\boldsymbol{X}_k = [\boldsymbol{x}_k \;\; \boldsymbol{m}_1 \;\; \boldsymbol{m}_2 \;\; \cdots \;\; \boldsymbol{m}_n]^{\mathrm{T}}$ is a stacking vector of robotic state and landmark location. Therefore, the SLAM problem can be regarded as the problem that computes the conditional probability distribution of state $\boldsymbol{X}$ with given control data and measurements.

Equation (3) is a posterior probability, which, through the Bayes' theorem, can be transformed into

$$\begin{aligned} &P(\boldsymbol{X}_k|\boldsymbol{X}_0, \boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = \\ &\qquad \frac{P(\boldsymbol{z}_k|\boldsymbol{X}_k)P(\boldsymbol{X}_k|\boldsymbol{X}_0, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})}{P(\boldsymbol{z}_k|\boldsymbol{X}_0, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k})} \propto \\ &\qquad P(\boldsymbol{z}_k|\boldsymbol{X}_k)P(\boldsymbol{X}_k|\boldsymbol{X}_0, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k}), \end{aligned} \tag{4}$$

where $\boldsymbol{z}_k$ is the observation at the time k, $\boldsymbol{z}_{1:k-1}$ denotes the observation from the beginning to the time $k-1$, and $\boldsymbol{u}_{1:k}$ denotes the input of the system from the beginning to the time k. Then from total probability formula and Markov chain, we can obtain

$$P(\boldsymbol{X}_k|\boldsymbol{X}_0, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k}) = \int P(\boldsymbol{X}_k|\boldsymbol{X}_{k-1}, \boldsymbol{X}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k-1})\times P(\boldsymbol{X}_{k-1}|\boldsymbol{X}_0, \boldsymbol{u}_{1:k-1}, \boldsymbol{z}_{1:k-1})\mathrm{d}\boldsymbol{X}_{k-1}. \tag{5}$$

In Eq. (4), $P(\boldsymbol{z}_k|\boldsymbol{X}_k)$ is likelihood; in Eq. (5), $P(\boldsymbol{X}_k|\boldsymbol{X}_{k-1}, \boldsymbol{X}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k-1})$ is called state-transaction density.

1.2 Robust State-Space Model

The SLAM probabilistic model mentioned above is in a nonlinear form. In many cases, nonlinear systems are formulated as nonlinear-Gaussian system, which means the noises of states and measurements are Gaussian distribution, and the following equations hold:

$$P(\boldsymbol{X}_k|\boldsymbol{X}_{k-1}) \sim \mathcal{N}(f(\boldsymbol{X}_{k-1}), \boldsymbol{Q}), \tag{6}$$

$$P(\boldsymbol{z}_k|\boldsymbol{X}_k) \sim \mathcal{N}(h(\boldsymbol{X}_k), \boldsymbol{R}), \tag{7}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a (multivariate) Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, $f(\cdot)$ denotes state-transaction function, and $h(\cdot)$ denotes observation function. Here, we assume that the noises $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$. Unlike the representation of Eq. (4), control variable is omitted because control variable is always given at each time k, and what we care about is the state.

However, Gaussian assumption cannot always hold. We define a robust state-space model with an uncertain measurement noise covariance matrix. Specifically, the robust state-space model can be obtained by changing Eq. (7) into

$$P(\boldsymbol{z}_k|\boldsymbol{X}_k, \boldsymbol{R}_k) \sim \mathcal{N}(h(\boldsymbol{X}_k), \boldsymbol{R}_k), \tag{8}$$

where $\boldsymbol{R}_k$ is a stochastic process over the set of positive-definite matrices at the time k.

To derive the specific formulation of robust state-space model, we need to confirm a prior distribution of $\boldsymbol{R}_k$. Because inverse-Wishart distribution is the conjugate prior distribution for the covariance of a Gaussian distribution, it is commonly used for covariance of Gaussian models in Bayesian analysis[31-32], and inverse-Wishart distribution is employed as the natural approximating distribution in the VB approach considered here. We follow the track of Agamennoni et al.[28], and define that $\boldsymbol{R}$ with the dimension of d is inverse-Wishart distribution:

$$\boldsymbol{R} \sim W^{-1}(\nu\boldsymbol{V}, \nu), \tag{9}$$

where $\boldsymbol{V}$ is harmonic mean, and ν is degree of freedom (DoF). Note that some adaptations are made to standard inverse-Wishart distribution. Here, $\nu\boldsymbol{V}$ forms corresponding parameter called inverse scale matrix of standard inverse-Wishart distribution, which is an aspect of difference from VB method[30]. The probability density function of $\boldsymbol{R}$ is given by

$$P(\boldsymbol{R}) = \frac{|\nu\boldsymbol{V}|^{\frac{\nu}{2}}}{2^{\frac{\nu d}{2}}\Gamma_d\left(\frac{\nu}{2}\right)}|\boldsymbol{R}|^{-\frac{\nu+d+1}{2}}\exp\left(-\frac{\nu}{2}\mathrm{tr}(\boldsymbol{V}\boldsymbol{R}^{-1})\right), \tag{10}$$

where $\mathrm{tr}(\cdot)$ and $|\cdot|$ denote the trace and determinant of the matrix, respectively.

With this kind of observation noise covariance model, we can derive that the likelihood of the measurement is t-distributed[28], and it has much heavier tails than the Gaussian distribution. Therefore, this model can assign a non-negligible probability to outliers. In this way, we can handle the situation when the system has heavy-tailed noise or outliers.

2 RVB-ACKF

CKF was proposed by Arasaratnam and Haykin[18] and applied to SLAM by Chandra et al[10]. According to the cubature integration principle, CKF utilizes a set of sampling points with the same weight to perform nonlinear transformation, after which a new corresponding set of sampling points which can be used as the approximation of probability distribution are obtained.

Compared with EKF, CKF can approximate nonlinear or even strong-nonlinear functions with much smaller error. CKF is able to handle the system with non-differentiable equations while EKF may fail in this case. Besides, UKF has the limitation like numerical inaccuracy, unavailability of a square-root solution, and filter instability[18], which is not presented in CKF. Therefore, CKF is a good choice to be employed in nonlinear system to solve the approximate problems of nonlinear system model with high accuracy. In this paper, we propose a new RVB-ACKF to give a solution to the SLAM system mentioned above.

2.1 System Model

Before deriving the algorithm, we firstly give the exact SLAM model we employ in this paper. Suppose that there is a mobile robot with the following state and observation functions:

$$\boldsymbol{X}_k = F(\boldsymbol{X}_{k-1}, \boldsymbol{u}_k) = \begin{bmatrix} f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k \\ \boldsymbol{M}_{k-1} \end{bmatrix}, \tag{11}$$

$$\boldsymbol{z}_k = h(\boldsymbol{X}_k) + \boldsymbol{\gamma}_k, \tag{12}$$

where $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ and $\gamma_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ denote control noises, and $F(\cdot,\cdot)$ represents the state function. Note that $F(\cdot,\cdot)$ is related to the state $\boldsymbol{X} = [\boldsymbol{x} \;\; \boldsymbol{m}_1 \;\; \boldsymbol{m}_2 \;\; \cdots \;\; \boldsymbol{m}_n]^{\mathrm{T}}$ and the input control variable $\boldsymbol{u}$. The robot motion equation $f(\cdot,\cdot)$ is related to the robot state $\boldsymbol{x}$ (not including the landmarks) and the input control variable $\boldsymbol{u}$. For simplicity of writing, we denote $\boldsymbol{M}_k \triangleq \{\boldsymbol{m}_1,\ \ \boldsymbol{m}_2,\ \ \cdots\ \ ,\boldsymbol{m}_n\}_k$ so that $\boldsymbol{M}_k$ means all the landmarks observed from the beginning to the time k. In this way, the state at the time k can be represented as $\boldsymbol{X}_k = [\boldsymbol{x}_k\ \ \boldsymbol{M}_k]^{\mathrm{T}}$. In Eq. (11), we assume that the landmark set $\boldsymbol{M}_{k-1}$ is the same as $\boldsymbol{M}_k$ (before the state augment). Unlike the standard Gaussian noise, the noise covariance $\boldsymbol{R}_k$ is time-variant, which obeys inverse-Wishart distribution, i.e., $\boldsymbol{R}_k \sim W^{-1}(\nu_k \boldsymbol{V}_k, \nu_k)$.

Specifically, Eqs. (11) and (12) can be written as

$$\boldsymbol{X}_k = F(\boldsymbol{X}_{k-1}, \boldsymbol{u}_k) = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \boldsymbol{M}_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta t v_k \cos(\theta_{k-1} + \alpha_k) \\ \Delta t v_k \sin(\theta_{k-1} + \alpha_k) \\ \Delta t v_k B^{-1} \sin\alpha_k \\ \mathbf{0} \end{bmatrix}, \tag{13}$$

$$\boldsymbol{z}_k = \begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \\ \arctan\left(\dfrac{y_i - y_k}{x_i - x_k} - \theta_k\right) \end{bmatrix}, \tag{14}$$

where (x_k, y_k, θ_k) denotes the vehicle's position and direction in 2D orthogonal world coordinate system (i.e., the robot state $\boldsymbol{x}_k$), Δt is the sampling interval of the sensor, B is the wheel base, v_k is the speed of the vehicle, and α_k is the veer angle of the front wheel relative to the central axis of the vehicle. As mentioned before, $\boldsymbol{M}_k \triangleq \{\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_n\}_k$ is a set of location of landmarks, where $\boldsymbol{m}_i = [x_i\ \ y_i]^{\mathrm{T}}$; $\boldsymbol{u}_k = [v_k\ \ \alpha_k]^{\mathrm{T}}$ is control variable. The output of the system is the range r and the bearing angle ϕ between the vehicle and observed landmarks. In Eqs. (13) and (14), mobile robot is modelled as a three-wheel vehicle driven by the front wheel with laser sensor providing range and bearing information[1].

2.2 Approximate Estimation by VB

To estimate the state of nonlinear system, we need to approximate the joint posterior distribution over the sequence of state and noise covariance as it evolves through time (i.e., $P(\boldsymbol{X}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k})$). However, this is a tough task as the state and noise are coupled. From Eq. (8), we can find that $\boldsymbol{X}_k$ and $\boldsymbol{R}_k$ are conditionally dependent as soon as the measurement $\boldsymbol{z}_k$ is obtained, even though their priories are independent. Fortunately, VB inference provides an approach to solve this kind of problem.

A natural idea is to find a simple and tractable formula Q to approximate the posterior distribution $P(\boldsymbol{X}, \boldsymbol{R}|\boldsymbol{z})$, where $\boldsymbol{z}$, $\boldsymbol{X}$ and $\boldsymbol{R}$ represent the sequences $\{\boldsymbol{z}_k\}$, $\{\boldsymbol{X}_k\}$ and $\{\boldsymbol{R}_k\}$ respectively.

For writing convenience, we use P to denote $P(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{R})$ as follows. The similarity between Q and P can be measured by Kullback-Leibler (KL) divergence[33]. From the definition of KL divergence, the following equation holds:

$$\begin{aligned} &\boldsymbol{D}_{\mathrm{KL}}(Q\|P) = \\ &\iint Q(\boldsymbol{X}, \boldsymbol{R}) \ln \frac{Q(\boldsymbol{X}, \boldsymbol{R})}{P(\boldsymbol{X}, \boldsymbol{R}|\boldsymbol{z})} \mathrm{d}\boldsymbol{X}\mathrm{d}\boldsymbol{R} = \\ &\iint Q(\boldsymbol{X}, \boldsymbol{R}) \ln \frac{Q(\boldsymbol{X}, \boldsymbol{R})}{P(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{z})} \mathrm{d}\boldsymbol{X}\mathrm{d}\boldsymbol{R} + \ln P(\boldsymbol{z}). \end{aligned} \tag{15}$$

Then, the marginal log-likelihood of the data $\ln P(\boldsymbol{z})$ can be expressed as

$$\begin{aligned} &\ln P(\boldsymbol{z}) = \\ &\boldsymbol{D}_{\mathrm{KL}}(Q\|P) - \iint Q(\boldsymbol{X}, \boldsymbol{R}) \ln \frac{Q(\boldsymbol{X}, \boldsymbol{R})}{P(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{z})} \mathrm{d}\boldsymbol{X}\mathrm{d}\boldsymbol{R} = \\ &\boldsymbol{D}_{\mathrm{KL}}(Q\|P) + \boldsymbol{L}(Q), \end{aligned} \tag{16}$$

where $\boldsymbol{L}(Q)$ is a lower bound on the marginal likelihood of the data.

Since the relative entropy (KL divergence) is non-negative and the marginal log-likelihood is independent of Q, minimizing the relative entropy is equivalent to maximizing the lower bound $\boldsymbol{L}(Q)$. Besides, as mentioned before, the root of intractability lies in the coupling of $\boldsymbol{X}_k$ and $\boldsymbol{R}_k$, so we would like to formulate the approximated posterior distribution as the following form:

$$Q(\boldsymbol{X}, \boldsymbol{R}) = Q_{\boldsymbol{X}}(\boldsymbol{X}) Q_{\boldsymbol{R}}(\boldsymbol{R}). \tag{17}$$

Using the methods from calculus of variations for maximizing the lower bound with respect to $Q_{\boldsymbol{X}}(\boldsymbol{X})$ and $Q_{\boldsymbol{R}}(\boldsymbol{R})$ in turn while keeping the other fixed derives[31]

$$\left.\begin{aligned} Q_{\boldsymbol{X}}(\boldsymbol{X}) \propto \exp(E_{Q_{\boldsymbol{R}}(\boldsymbol{R})}[\ln P(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{z})]) = \\ \exp\left(\int Q_{\boldsymbol{R}}(\boldsymbol{R}) \ln P(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{z}) \mathrm{d}\boldsymbol{R}\right) \\ Q_{\boldsymbol{R}}(\boldsymbol{R}) \propto \exp(E_{Q_{\boldsymbol{X}}(\boldsymbol{X})}[\ln P(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{z})]) = \\ \exp\left(\int Q_{\boldsymbol{X}}(\boldsymbol{X}) \ln P(\boldsymbol{X}, \boldsymbol{R}, \boldsymbol{z}) \mathrm{d}\boldsymbol{X}\right) \end{aligned}\right\}, \tag{18}$$

where $E_Q[\cdot]$ means the expectation with respect to Q. Equation (18) does not provide analytic solutions since $Q_{\boldsymbol{X}}(\boldsymbol{X})$ and $Q_{\boldsymbol{R}}(\boldsymbol{R})$ are coupled. To solve the Q posteriors, we must follow the approach of iteration by alternating between the two functions of Eq. (18) until $Q_{\boldsymbol{X}}(\boldsymbol{X})$ and $Q_{\boldsymbol{R}}(\boldsymbol{R})$ are convergent.

With the assumption that the prior $\boldsymbol{X}_{k-1}$ is Gaussian distribution with the form of $\boldsymbol{X}_{k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{k-1}, \boldsymbol{p}_{k-1})$ and $\boldsymbol{R}_{k-1}$ is inverse-Wishart distribution in the form of $\boldsymbol{R}_{k-1} \sim W^{-1}(\nu_{k-1}\boldsymbol{V}_{k-1}, \nu_{k-1})$, where $\boldsymbol{V}_{k-1}$ is a positive matrix, the predictions of state and observation noise covariance can be obtained from the following equations:

$$P(\boldsymbol{X}_k|\boldsymbol{z}_{1:k-1}) = \int \mathcal{N}(\boldsymbol{X}_k|f(\boldsymbol{X}_{k-1}))\mathcal{N}(\boldsymbol{X}_{k-1}|\boldsymbol{\mu}_{k-1},\boldsymbol{p}_{k-1})\mathrm{d}\boldsymbol{X}_{k-1} = \mathcal{N}(\boldsymbol{X}_k|\boldsymbol{\mu}_{k|k-1},\boldsymbol{p}_{k|k-1}), \tag{19}$$

$$P(\boldsymbol{R}_k|\boldsymbol{z}_{1:k-1}) = \int P(\boldsymbol{R}_k|\boldsymbol{R}_{k-1})W^{-1}(\boldsymbol{R}_{k-1}|\boldsymbol{\mu}_{k-1},\boldsymbol{p}_{k-1})\mathrm{d}\boldsymbol{X}_{k-1} = W^{-1}(\boldsymbol{R}_k|\nu_{k|k-1}\boldsymbol{V}_{k|k-1},\nu_{k|k-1}). \tag{20}$$

Here we denote Eq. (19) as $\boldsymbol{X}_{k|k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{k|k-1},\boldsymbol{p}_{k|k-1})$ and Eq. (20) as $\boldsymbol{R}_{k|k-1} \sim W^{-1}(\nu_{k|k-1}\boldsymbol{V}_{k|k-1},\nu_{k|k-1})$. By solving Eq. (18), we can obtain that $Q_{\boldsymbol{X}}(\boldsymbol{X})$ is Gaussian distribution and $Q_{\boldsymbol{R}}(\boldsymbol{R})$ is inverse-Wishart distribution with the forms:

$$P(\boldsymbol{X}_k|\boldsymbol{z}_{1:k}) \sim \mathcal{N}(\boldsymbol{\mu}_k,\boldsymbol{p}_k), \tag{21}$$

$$P(\boldsymbol{R}_k|\boldsymbol{z}_{1:k}) \sim W^{-1}(\nu_k\boldsymbol{V}_k,\nu_k). \tag{22}$$

From Eq. (18), we can derive

$$\int Q_{\boldsymbol{R}}(\boldsymbol{R}_k)\ln P(\boldsymbol{X}_k,\boldsymbol{R}_k,\boldsymbol{z}_k|\boldsymbol{z}_{1:k})\mathrm{d}\boldsymbol{R}_k = C_1 - \frac{1}{2}\Big[(\boldsymbol{z}_k - h(\boldsymbol{X}_k))^{\mathrm{T}}(\nu_k - d - 1)(\nu_k\boldsymbol{V}_k)^{-1}(\boldsymbol{z}_k - h(\boldsymbol{X}_k)) + (\boldsymbol{X}_k - \boldsymbol{\mu}_{k|k-1})^{\mathrm{T}}(\boldsymbol{p}_{k|k-1})^{-1}(\boldsymbol{X}_k - \boldsymbol{\mu}_{k|k-1})\Big], \tag{23}$$

$$\int Q_{\boldsymbol{X}}(\boldsymbol{X}_k)\ln P(\boldsymbol{X}_k,\boldsymbol{R}_k,\boldsymbol{z}_k|\boldsymbol{z}_{1:k})\mathrm{d}\boldsymbol{X}_k = C_2 - \frac{1}{2}\Big\{(\nu_{k|k-1} + d + 2)\ln|\boldsymbol{R}_k| + \mathrm{tr}(\nu_{k|k-1}\boldsymbol{V}_{k|k-1}\boldsymbol{R}_k^{-1}) + \mathrm{tr}(E_{\boldsymbol{X}}[(\boldsymbol{z}_k - h(\boldsymbol{X}_k))(\boldsymbol{z}_k - h(\boldsymbol{X}_k)^{\mathrm{T}}]\boldsymbol{R}_k^{-1})\Big\}. \tag{24}$$

Comparing Eq. (23) and Eq. (24) with the log-likelihood of Gaussian and inverse-Wishart distribution, we can derive

$$\boldsymbol{z}_{k|k-1} = \int h(\boldsymbol{X}_{k|k-1})\mathcal{N}(\boldsymbol{X}_{k|k-1}|\boldsymbol{\mu}_{k|k-1},\boldsymbol{p}_{k|k-1})\mathrm{d}\boldsymbol{X}_{k|k-1}, \tag{25}$$

$$\boldsymbol{p}_{zz} = \int [h(\boldsymbol{X}_k) - h(\boldsymbol{X}_{k|k-1})][h(\boldsymbol{X}_k) - h(\boldsymbol{X}_{k|k-1})]^{\mathrm{T}} \times \mathcal{N}(\boldsymbol{X}_{k|k-1}|\boldsymbol{\mu}_{k|k-1},\boldsymbol{p}_{k|k-1})\mathrm{d}\boldsymbol{X}_{k|k-1} + \boldsymbol{V}_k = \bar{\boldsymbol{p}}_{zz} + \boldsymbol{V}_k, \tag{26}$$

$$\boldsymbol{p}_{xz} = \int (\boldsymbol{X}_{k|k-1} - \boldsymbol{\mu}_{k|k-1})(h(\boldsymbol{X}_{k|k-1}) - \boldsymbol{z}_{k|k-1})^{\mathrm{T}} \times \mathcal{N}(\boldsymbol{X}_{k|k-1}|\boldsymbol{\mu}_{k|k-1},\boldsymbol{p}_{k|k-1})\mathrm{d}\boldsymbol{X}_{k|k-1}, \tag{27}$$

$$\boldsymbol{K}_k = \boldsymbol{p}_{xz}\boldsymbol{p}_{zz}^{-1}, \tag{28}$$

$$\boldsymbol{\mu}_{k|k} = \boldsymbol{\mu}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{z}_{k|k-1}), \tag{29}$$

$$\boldsymbol{p}_{k|k} = \boldsymbol{p}_{k|k-1} + \boldsymbol{K}_k\boldsymbol{p}_{zz}\boldsymbol{K}_k^{\mathrm{T}} - \boldsymbol{K}_k\boldsymbol{p}_{xy}^{\mathrm{T}} - \boldsymbol{p}_{xy}\boldsymbol{K}_k^{\mathrm{T}}. \tag{30}$$

The relationships between the parameters of prior and approximated posterior with respect to $\boldsymbol{R}_k$ are given by

$$\boldsymbol{\Omega}_k = (1-a)\nu_{k-1}\boldsymbol{V}_{k-1} + \boldsymbol{S}_k, \tag{31}$$

$$\nu_k = (1-a)\nu_{k-1} + a(d-1), \tag{32}$$

$$\boldsymbol{S}_k = \int (\boldsymbol{z}_k - h(\boldsymbol{X}_{k|k}))(\boldsymbol{z}_k - h(\boldsymbol{X}_{k|k}))^{\mathrm{T}} \times \mathcal{N}(\boldsymbol{X}_{k|k}|\boldsymbol{\mu}_{k|k},\boldsymbol{p}_{k|k})\mathrm{d}\boldsymbol{X}_{k|k}, \tag{33}$$

where a is the constant discount factor, $0 < a \ll 1$, and $\boldsymbol{\Omega}_k = \nu_k\boldsymbol{V}_k$ is the parameter of $\boldsymbol{R}_k$. During the derivation, we adopt the Carvalho and West' model[34], introducing the discount factor that is originally from the field of econometrics for tracking the portfolio volatility and is another main difference from VB-ACKF[30]. As the discount factor a is small, the harmonic mean $\boldsymbol{V}_k = \boldsymbol{\Omega}_k/\nu_k$ changes little. However, thanks to the factor a, the DoF ν_k would be discounted at every time step, which leads to the increase of the spread of the distribution.

2.3 CKF Adaptation for the VB Estimation

Although we have already got the approximate estimation of VB, we cannot use the equations mentioned above directly. For Eqs. (25), (26), (27) and (33), some adaptions are needed. Here, we employ the structure of CKF. To use CKF, we need to construct cubature points. Suppose that the state covariance at the time $k-1$ is $\boldsymbol{p}_{k-1|k-1}$. Then the cubature points are calculated as

$$\boldsymbol{C}_{k-1|k-1} = \mathrm{cholesky}(\boldsymbol{p}_{k-1|k-1}), \tag{34}$$

$$\boldsymbol{p}_{k-1|k-1} = \boldsymbol{C}_{k-1|k-1}\boldsymbol{C}_{k-1|k-1}^{\mathrm{T}}, \tag{35}$$

$$\boldsymbol{X}_{i,k-1|k-1} = \boldsymbol{C}_{k-1|k-1}\boldsymbol{\xi}_i + \boldsymbol{X}_{k-1|k-1}, \quad i = 1,2,\cdots,2n, \tag{36}$$

where, $\boldsymbol{C}$ is the result of Cholesky decomposition over $\boldsymbol{p}$; $\boldsymbol{\xi}_i = \boldsymbol{e}_i$ for $i = 1,2,\cdots,n$, and $\boldsymbol{\xi}_i = -\boldsymbol{e}_{i-n}$ for $i = n+1,n+2,\cdots,2n$ ($\boldsymbol{e}_i$ is a vector, in which the ith element is 1 and the other elements are 0); $\boldsymbol{X}_{i,k-1|k-1}$ are so-called cubature points. Through the cubature points, we can make prediction over state vector and state covariance matrix, and the detailed information can be found in the algorithm (Lines 7—11) of the next section.

Then, we use the predicted results $\boldsymbol{\mu}_{k|k-1}$ and $\boldsymbol{p}_{k|k-1}$ to calculate the cubature points $\boldsymbol{\mu}_{i,k|k-1}$ again, so that we can compute observation:

$$\boldsymbol{z}_{i,k|k-1} = h(\boldsymbol{\mu}_{i,k|k-1}), \tag{37}$$

$$\boldsymbol{z}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\boldsymbol{z}_{i,k|k-1}. \tag{38}$$

With computed observation and predicted state, Eqs. (26), (27) and (33) can be derived as

$$\bar{\boldsymbol{p}}_{zz} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{z}_{i,k|k-1}\boldsymbol{z}_{i,k|k-1}^{\mathrm{T}} - \boldsymbol{z}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}}, \tag{39}$$

$$\boldsymbol{p}_{xz} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{\mu}_{i,k|k-1}\boldsymbol{z}_{i,k|k-1}^{\mathrm{T}} - \boldsymbol{\mu}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}}, \tag{40}$$

$$\boldsymbol{S}_k = \sum_{i=1}^{2n} \left(\boldsymbol{z}_{i,k|k-1}^{(\mathrm{iter})} - \boldsymbol{z}_{k|k-1}^{(\mathrm{iter})}\right)\left(\boldsymbol{z}_{i,k|k-1}^{(\mathrm{iter})} - \boldsymbol{z}_{k|k-1}^{(\mathrm{iter})}\right)^{\mathrm{T}}. \tag{41}$$

The superscript “iter” in Eq. (41) represents the number of iterations; as we can see in the following section, the algorithm is iterative.

2.4 Proposed SLAM Algorithm

The complete process of the SLAM algorithm is shown in Algorithm 1. First of all, initializing is necessary: state vector $\boldsymbol{X}_0$, state covariance $\boldsymbol{p}_0$, control noise $\boldsymbol{Q}_0$, parameters of inverse-Wishart distribution ν_0 and $\boldsymbol{V}_0$, and parameters of sensor and vehicle. Then, we perform the predicting step, in the form of CKF, which is followed by the process of data association. During the data association, we can get the location of an observed landmark through observation function, and then we match it with $\boldsymbol{m}_k$ in state vector $\boldsymbol{X}_k$ to judge whether it is a new landmark or already exists in $\boldsymbol{M}_k$. If the case is the latter, we perform the process of iteration until the results converge (Lines 14—37). However, if the observed landmark is a new landmark $\boldsymbol{m}_{\mathrm{new}}$, it is added to the state vector $\boldsymbol{X}_k^* = [\boldsymbol{X}_k \quad \boldsymbol{m}_{\mathrm{new}}]^{\mathrm{T}}$. As for the augment state covariance $\boldsymbol{p}_k^*$, it can be derived following the similar process of prediction by $\boldsymbol{X}_k^*$ and $\boldsymbol{V}_k$.

From the procedure mentioned above, we can realize a real-time SLAM system based on RVB-ACKF.

```
Algorithm 1: RVB-ACKF
1: Set initial values $\boldsymbol{X}_0$, $\boldsymbol{p}_0$, $\boldsymbol{Q}_0$, $\nu_0$, $\boldsymbol{V}_0$
2: For $k = 1, 2, \cdots, t_n$ do
3: Predict step
4: $\boldsymbol{p}_{k-1|k-1} = \boldsymbol{C}_{k-1|k-1}\boldsymbol{C}_{k-1|k-1}^{\mathrm{T}}$
5: Construction of cubature points
6: $\boldsymbol{X}_{i,k-1|k-1} = \boldsymbol{C}_{k-1|k-1}\boldsymbol{\xi}_i + \boldsymbol{X}_{k-1|k-1}$,
   $i = 1, 2, \cdots, 2n$
7: Prediction of state vector
8: $\boldsymbol{Y}_{i,k|k-1} = F(\boldsymbol{X}_{i,k-1|k-1})$
9: $\boldsymbol{\mu}_{k|k-1} = (1/2n)\sum_{i}\boldsymbol{Y}_{i,k|k-1}$
10: Prediction of state covariance
11: $\boldsymbol{p}_{k|k-1} = (1/2n)\sum_{i=1}^{2n}\boldsymbol{Y}_{i,k|k-1}\boldsymbol{Y}_{i,k|k-1}^{\mathrm{T}} -$
    $\boldsymbol{\mu}_{k|k-1}\boldsymbol{\mu}_{k|k-1}^{\mathrm{T}} + \boldsymbol{Q}_k$
12: End predict step
13: Data association
14: Update step
15: Construct cubature points
16: $\boldsymbol{\mu}_{i,k|k-1} = \boldsymbol{C}_{k|k-1}\boldsymbol{\xi}_i + \boldsymbol{\mu}_{k|k-1}, i = 1, 2, \cdots, 2n$
17: Compute observation with cubature points
18: $\boldsymbol{z}_{i,k|k-1} = h(\boldsymbol{\mu}_{i,k|k-1})$
19: $\boldsymbol{z}_{k|k-1} = (1/2n)\sum_{i=1}^{2n}\boldsymbol{z}_{i,k|k-1}$
20: $\boldsymbol{p}_{xz} = (1/2n)\sum_{i=1}^{2n}\boldsymbol{\mu}_{i,k|k-1}\boldsymbol{z}_{i,k|k-1}^{\mathrm{T}} - \boldsymbol{\mu}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}}$
21: $\bar{\boldsymbol{p}}_{zz} = (1/2n)\sum_{i=1}^{2n}\boldsymbol{z}_{i,k|k-1}\boldsymbol{z}_{i,k|k-1}^{\mathrm{T}} - \boldsymbol{z}_{k|k-1}\boldsymbol{z}_{k|k-1}^{\mathrm{T}}$
22: $\boldsymbol{\mu}_{k|k}^{(0)} = \boldsymbol{\mu}_{k|k-1}, \boldsymbol{p}_{k|k}^{(0)} = \boldsymbol{p}_{k|k-1}$
23: $\boldsymbol{z}_{i,k|k-1}^{(0)} = \boldsymbol{z}_{i,k|k-1}, \boldsymbol{z}_{k|k-1}^{(0)} = \boldsymbol{z}_{k|k-1}$
24: $\nu_k = (1-a)\nu_{k-1} + a(d-1) + 1$
25: Repeat
26: Predict and update noise given state
27: $\boldsymbol{S}_k = (1/2n)\sum_{i=1}^{2n}\left(\boldsymbol{z}_{i,k|k-1}^{(\mathrm{iter})} - \boldsymbol{z}_{k|k-1}^{(\mathrm{iter})}\right)\left(\boldsymbol{z}_{i,k|k-1}^{(\mathrm{iter})} -\right.$
    $\left.\boldsymbol{z}_{k|k-1}^{(\mathrm{iter})}\right)^{\mathrm{T}}$
28: $\boldsymbol{\Omega}_k = (1-a)\nu_{k-1}\boldsymbol{V}_{k-1} + \boldsymbol{S}_k$,
29: Update state given noise
30: $\boldsymbol{K}_k = \boldsymbol{p}_{xz}\left(\bar{\boldsymbol{p}}_{zz} + \frac{\boldsymbol{\Omega}_k}{\nu_k}\right)^{-1}$
31: $\boldsymbol{\mu}_{k|k}^{(\mathrm{iter})} = \boldsymbol{\mu}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{z}_{k|k-1})$
32: $\boldsymbol{p}_{k|k}^{(\mathrm{iter})} = \boldsymbol{p}_{k|k-1} - \boldsymbol{K}_k\left(\bar{\boldsymbol{p}}_{zz} + \frac{\boldsymbol{\Omega}_k}{\nu_k}\right)\boldsymbol{K}_k^{\mathrm{T}} =$
    $\boldsymbol{p}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{p}_{zz}\boldsymbol{K}_k^{\mathrm{T}}$
33: Construct cubature points with updated state
34: $\boldsymbol{\mu}_{i,k|k}^{(\mathrm{iter})} = \boldsymbol{C}_{k|k}^{(\mathrm{iter})}\boldsymbol{\xi}_i + \boldsymbol{\mu}_{k|k}^{(\mathrm{iter})}, i = 1, 2, \cdots, 2n$
35: $\boldsymbol{z}_{i,k|k-1}^{(\mathrm{iter})} = h(\boldsymbol{\mu}_{i,k|k-1}^{(\mathrm{iter})})$
36: $\boldsymbol{z}_{k|k-1}^{(\mathrm{iter})} = (1/2n)d\sum_{i=1}^{2n}\boldsymbol{z}_{i,k|k-1}^{(\mathrm{iter})}$
37: Until converge
38: End update step
39: State augment
40: End for
```

3 Experimental Validation

To validate the effectiveness of the proposed algorithm, we do the experiments both in simulation environment and real dataset.

3.1 Simulation Validation

To do simulation, we construct an environment with 17 key points which determine the vehicle's path and 35 landmarks which denote the circumstance around the vehicle. The path is annular with a length of around 680 m, as shown in Fig. 1. At the beginning, the vehicle is located at (0, 0) and then it tracks the key points (landmarks) to explore the environment. This procedure is performed for

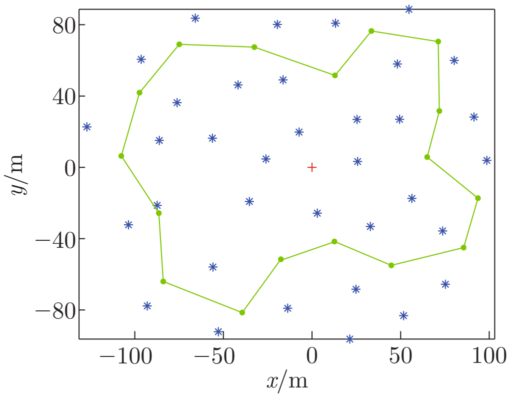


Fig. 1 Path and landmarks

two loops. The simulation experiments are carried on in two cases: time-invariant observation noise, and time-variant observation noise. We compare our algorithm of RVB-ACKF with EKF-SLAM[35], UKF-SLAM[36], CKF-SLAM[10], squared-root-CKF-SLAM (SRCKF-SLAM)[37] and VB-ACKF[30].

To perform these simulation experiments with different algorithms mentioned above, we adapt the source code (www-personal.acfr.usyd.edu.au/tbailey), which is EKF-SLAM, to the UKF, CKF, SRCKF, VB-ACKF and RVB-ACKF. Note that all the algorithms include four parts: prediction, getting observation, updating states with observation, and doing state augment, as mentioned in Subsection 2.4. The parameters used in the algorithms will be listed in the following part.

3.1.1 *Simulation with Gaussian Observation Noise*

Under this situation, we set the observation noise covariance as $\boldsymbol{R} = \boldsymbol{R}_0 = \mathrm{diag}(0.01, 0.000\,3)$ for all the algorithms. For VB-ACKF, we set $r = 1.0$, $\nu = 10$ and 5 iterations. For our method, we set $a = 0.1$, $\nu = 10$ and 5 iterations. For all the algorithms, 50 Monte Carlo simulation experiments are conducted.

Table 1 lists the root-mean-square error (RMSE) of x and y axes for different algorithms. Figure 2 shows the errors along x and y axes, which are denoted as ζ_x and ζ_y respectively, for five mentioned SLAM algorithms (x and y axes are corresponding to the world coordinate system). In the plots, the horizontal axis denotes the number of steps (t), which means the time of simulation duration.

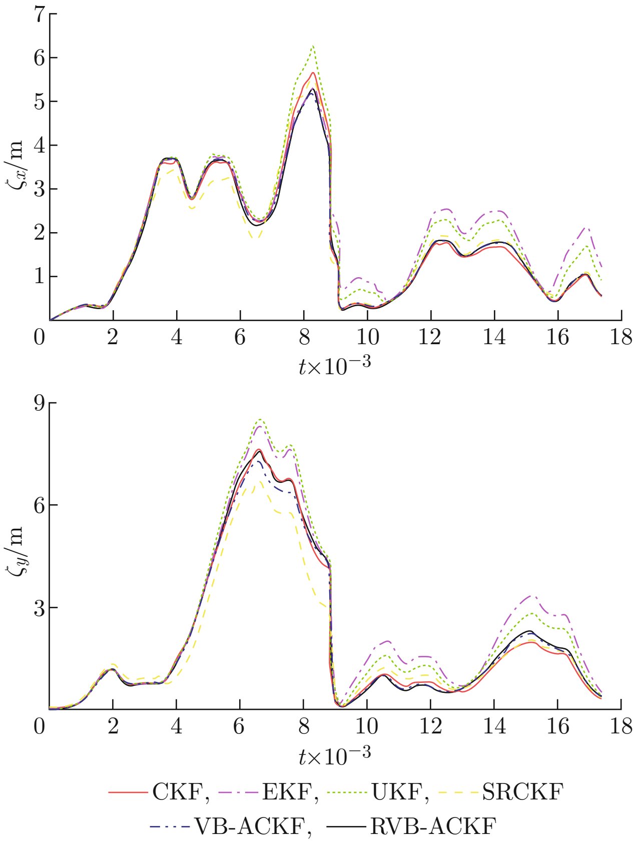


Fig. 2 Errors along x and y axes for different SLAM algorithms with Gaussian observation noise

From Fig. 2, we can find that at the steps around 9 000, the errors along x and y axes decrease significantly. That is because at that time the vehicle finishes the first loop and is revisiting old landmarks, which is called loop-closure. Loop-closure is a key process in SLAM algorithms to reduce cumulative error[2].

Table 1 RMSEs of different SLAM algorithms along x and y axes

Axis	RMSE/m					
	EKF	UKF	CKF	SRCKF	VB-ACKF	RVB-ACKF
x	2.571	2.514	2.288	2.245	2.228	2.254
y	3.473	3.455	3.055	2.771	2.979	3.105

The results (Fig. 2 and Table 1) indicate that five algorithms of EKF, UKF, CKF, SRCKF and RB-ACKF have similar accuracy while VB-ACKF has the best results. However, the proposed method also has competitive results.

3.1.2 *Simulation with Heavy-Tailed Observation Noise*

In this part, we test the effectiveness of the five algorithms under time-variant heavy-tailed noise. To simulate the heavy-tailed observation noise, we set the algorithm of the observation noise covariance as follows.

```
Algorithm 2: Heavy-tailed observation noise covariance
1: For each sampling interval
2: If rand(1)< $P_O$
3: $\boldsymbol{R} = n\boldsymbol{R}_0$
4: End if
```

In Algorithm 2, P_{O} denotes the probability of outliers, and n, which to some extent decides the distinctness of the heavy-tailed noise, is a positive scalar.

To test the influence of n, we divide our simulation into two parts: ① n is constant during the whole simulation process; ② n is time-variant during the whole simulation process.

For the first case, we set the parameters $\boldsymbol{R} = \boldsymbol{R}_0 = \mathrm{diag}(0.01, 0.000\,3)$ and $P_{\mathrm{O}} = 0.1$ for all the algorithms. For VB-ACKF, we set $r = 1.0$, $\nu = 10$ and 5 iterations. For our method, we set $a = 0.1$, $\nu = 10$ and 5 iterations. We test five situations: $n = 10, 30, 50, 70, 100$. For all the algorithms, 50 Monte Carlo simulation experiments are carried on, and the results are shown in Fig. 3 and Table 2.

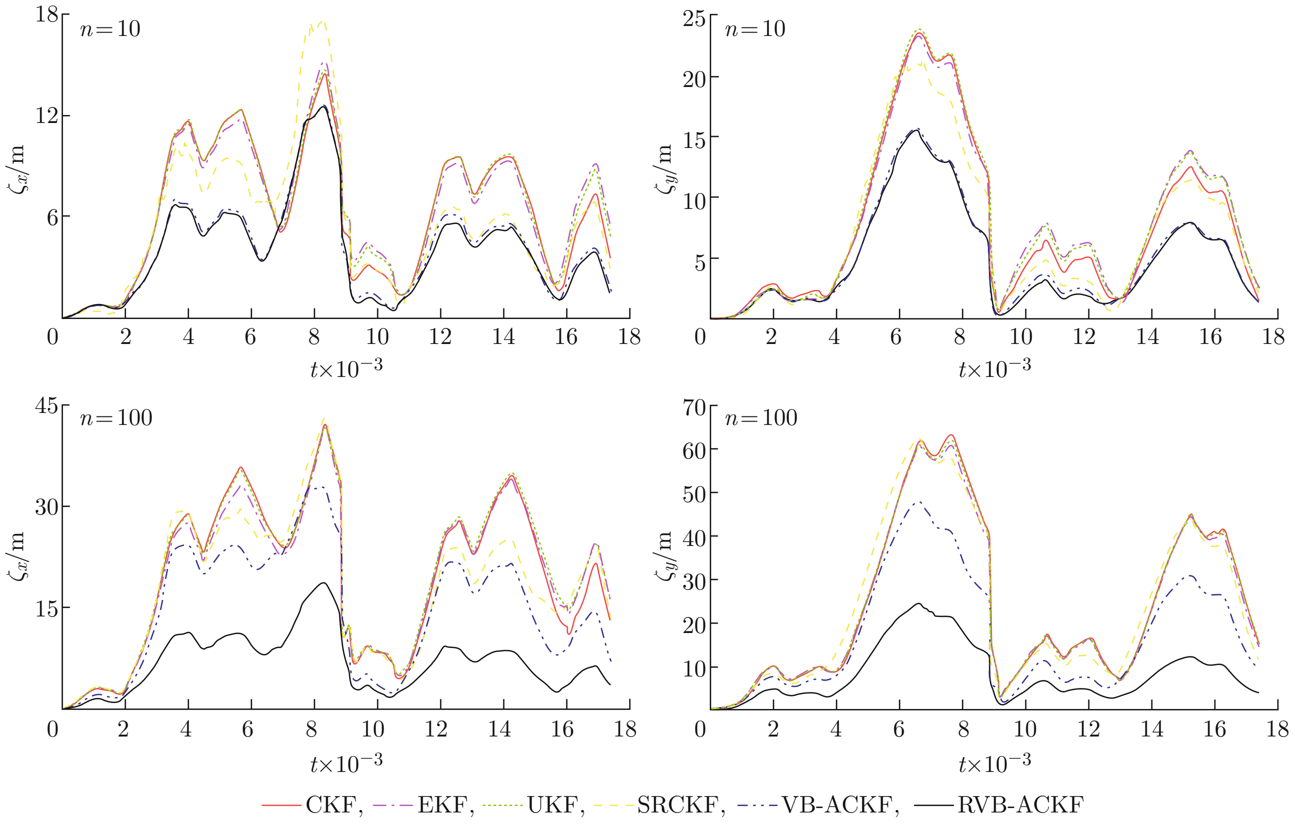


Fig. 3 Errors along x and y axes for different SLAM algorithms under time-variant heavy-tailed noise

Table 2 RMSEs of different SLAM algorithms with different n

n	Axis	RMSE/m					
		EKF	UKF	CKF	SRCKF	VB-ACKF	RVB-ACKF
10	x	7.547	7.592	7.437	7.044	5.155	5.007
	y	10.374	10.546	10.174	9.3463	6.570	6.439
30	x	13.521	13.581	13.493	12.564	9.242	7.523
	y	20.018	19.996	19.492	18.766	11.312	8.978
50	x	18.868	18.751	18.502	18.013	11.080	7.412
	y	23.409	21.999	22.358	21.472	14.204	8.770
70	x	22.195	22.526	22.251	21.064	16.545	7.818
	y	30.594	30.503	30.392	29.847	20.369	9.657
100	x	22.414	23.095	22.650	21.146	17.462	8.140
	y	29.814	30.309	30.558	30.511	22.021	10.638

For the second case, the parameters are the same as those of the first case except that we set n by the following principle:

$$n = \begin{cases} 10, & 0 < t \leqslant 3\,480 \\ 30, & 3\,480 < t \leqslant 6\,960 \\ 50, & 6\,960 < t \leqslant 10\,440 \\ 70, & 10\,440 < t \leqslant 13\,920 \\ 100, & 13\,920 < t \leqslant 17\,383 \end{cases}.$$

The results are shown in Fig. 4 and Table 3.

As can be seen from Figs. 3 and 4 and Tables 2 and 3, the EKF, UKF and CKF algorithms show poor performance when dealing with heavy-tailed noise; especially when the parameter n is big (i.e., heavy-tailed noise is more distinct), the errors of these three algorithm can be very large. On the contrary, the algorithms based on VB perform much better. As n is getting bigger, the performance of VB-ACKF method is also bad, although the results are better than the first three algorithms. However, the proposed algorithm shows the significant advantages when dealing with heavy-tailed noise.

To show the effectiveness of different SLAM algorithms towards different probabilities of outliers, we simulate the cases of varying P_{O} while $n = 100$. Note that the other parameters are the same as the above simulations. The results are shown in Table 4.

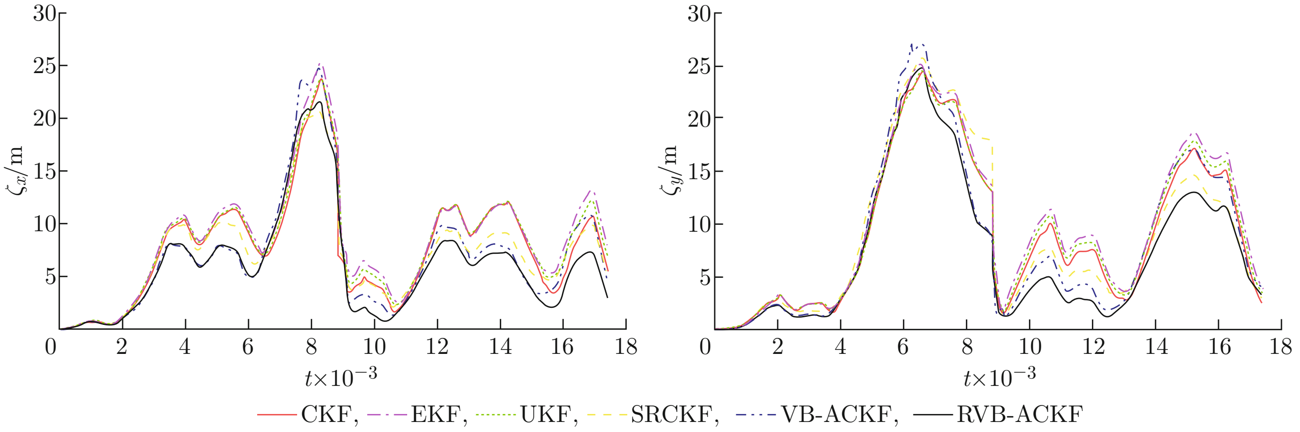


Fig. 4 Errors along x and y axes for different SLAM algorithms with time-variant n

Table 3 RMSEs of different SLAM algorithms with time-variant n

Axis	RMSE/m					
	EKF	UKF	CKF	SRCKF	VB-ACKF	RVB-ACKF
x	10.080	9.659	9.362	9.071	8.896	7.893
y	12.299	11.715	11.470	11.637	11.484	10.041

Table 4 RMSEs of different SLAM algorithms with different P_{O}

P_{O}	Axis	RMSE/m					
		EKF	UKF	CKF	SRCKF	VB-ACKF	RVB-ACKF
0.1	x	22.414	23.095	22.652	21.146	17.462	8.140
	y	29.814	30.309	30.558	30.511	22.021	10.638
0.2	x	24.901	24.891	24.763	22.446	15.203	7.205
	y	30.061	31.302	31.407	30.628	17.990	9.155
0.3	x	22.885	23.354	23.333	22.349	17.404	8.599
	y	30.343	30.625	30.737	30.654	20.829	10.891
0.4	x	23.857	24.461	24.315	24.018	16.330	7.995
	y	31.926	31.513	31.820	31.295	19.647	10.389
0.5	x	28.190	28.439	28.090	26.249	15.609	8.924
	y	34.248	34.511	34.352	33.416	16.249	9.475

From Table 4, we can find that as the probability of outliers increases, the RMSE has the trend of rising (not strictly). It is a natural consequence because with the increase of the probability of outliers, the heavy-tailed noise is more distinct and this leads to higher error. For the traditional methods (i.e., EKF, UKF, CKF and SRCKF), the errors are not only very large but also increase significantly as P_{O} becomes bigger. However, for the method we propose, as it can handle the heavy-tailed noise, the RMSE is much smaller and changes little with the increase of P_{O}.

3.2 Real Dataset Experiment

In this section, we test our algorithm on the real dataset. Here, we adapt the EKF-SLAM source code (github.com/ydsf16/aruco_ekf_slam) to CKF-SLAM, VB-ACKF and RVB-ACKF. The experiments are based on a four-wheel vehicle with a mono-camera. The camera is used to identify the landmarks which are ArUco Marker (see Fig. 5(a)). Note that although the poses of ArUco Markers observed by camera are 6 DoFs, we transform the observation from 3D to 2D using the related transform matrix $\boldsymbol{T}_{\mathrm{m}}^{\mathrm{c}}$ between the landmark and camera, and the transform matrix $\boldsymbol{T}_{\mathrm{c}}^{\mathrm{r}}$ between the camera and robot (Fig. 5(b)). In this way, we can get the range and bearing information from the camera. Figure 5(c) is a picture of the real experiment environment.

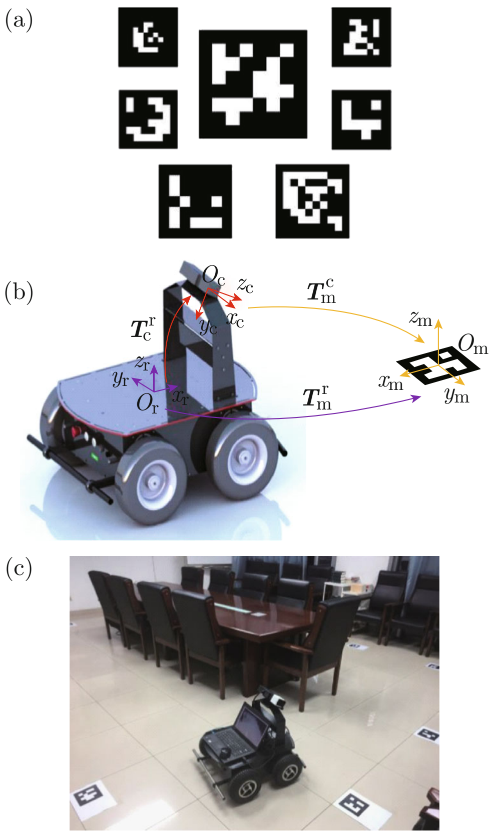


Fig. 5 The experimental set

Because the real dataset has no ground truth, it is not feasible to compare the algorithms' accuracy in the normal noise condition. Instead, we do the following experiments: we use the CKF-SLAM under the normal noise condition as a benchmark (see Fig. 6), and we compare the CKF-SLAM, VB-ACKF and RVB-ACKF with heavy-tailed noise with the benchmark to show the outperformance of the proposed algorithm. The robot runs three loops around the room, and the whole trajectory is about 50 m (see Fig. 6).

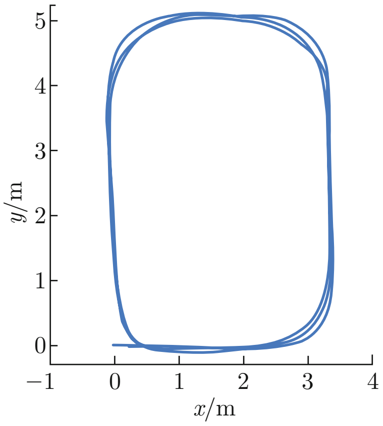
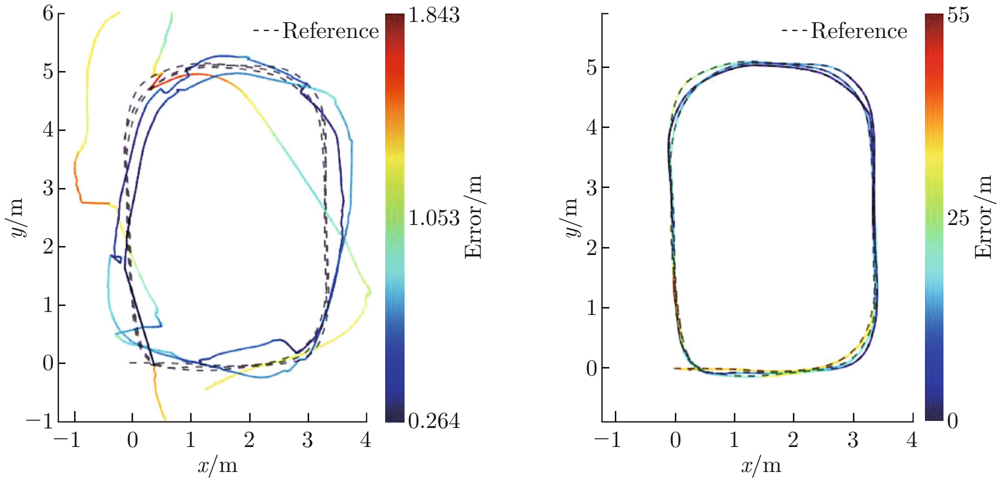


Fig. 6 The trajectory benchmark generated by CKF-SLAM with normal noise condition

For the observation noise, we set $\boldsymbol{R} = \mathrm{diag}(\|k_{\boldsymbol{r}}\boldsymbol{r}\|^2, \|k_{\boldsymbol{\phi}}\boldsymbol{\phi}\|)$, where $k_{\boldsymbol{r}}$ and $k_{\boldsymbol{\phi}}$ are constant coefficients of observation range matrix ($\boldsymbol{r}$) and bearing angle matrix ($\boldsymbol{\phi}$). Here we set $k_{\boldsymbol{r}} = k_{\boldsymbol{\phi}} = 0.1$. For VB-ACKF, the parameters are $r = 1.0$, $\nu = 10$ and 5 iterations. For our method, the parameters are $a = 0.1$, $\nu = 10$ and 5 iterations. We perform the experiments under three different sets of parameters: ① $n = 10$, $P_{\mathrm{O}} = 0.1$; ② $n = 10$, $P_{\mathrm{O}} = 0.3$; ③ $n = 50$, $P_{\mathrm{O}} = 0.3$. For each set of parameters, we perform 10 experiments, and the results (Table 5 and Fig. 7) shown below are the average of 10 experiments. In Fig. 7(a), reference is benchmark, and the colorful trajectory is CKF-SLAM with heavy-tailed noise ($n = 10$, $P_{\mathrm{O}} = 0.1$). In Fig. 7(b), reference is benchmark, and the colorful trajectory is RVB-ACKF with heavy-tailed noise ($n = 10$, $P_{\mathrm{O}} = 0.1$). The color bar besides the trajectory represents the error mapped onto trajectory.

Table 5 RMSEs of different SLAM algorithms with different sets of parameters

Parameter	RMSE/m		
	CKF-SLAM	VB-ACKF	RVB-ACKF
$n = 10$, $P_{\mathrm{O}} = 0.1$	0.924	0.030	0.025
$n = 10$, $P_{\mathrm{O}} = 0.3$	2.025	0.143	0.049
$n = 50$, $P_{\mathrm{O}} = 0.3$	0.539	0.072	0.059



(a) CKF-SLAM (b) RVB-ACKF

Fig. 7 The trajectory comparison

From Fig. 7 and Table 5, we can find that RVB-ACKF method has better performance than the other two algorithms and the results of RVB-ACKF method are much more robust than those of CKF-SLAM and VB-ACKF algorithms. We can find that the RMSEs of CKF-SLAM and VB-ACKF algorithms tend to change dramatically as n and P_{O} change, while the results of RVB-ACKF method change just a little as the heavy-tails noise becomes more and more evident. Actually, the VB-ACKF method is suitable for the condition where the observation noise is unknown, while the RVB-ACKF method is suitable for the more general condition where the observation noise is not only unknown but also changed along the process with heavy-tailed character, so the RVB-ACKF method performs much better than the VB-ACKF method. The simulation and real dataset experiment results demonstrate that the proposed algorithm has great robustness to the heavy-tails noise and can be safely used in state estimation.

4 Conclusion

In this paper, we propose a robust SLAM algorithm based on VB-ACKF. In the algorithm, we model the noise covariance as inverse-Wishart distribution with the parameter “harmonic mean”, which can handle heavy-tailed noise and outliers. To decouple and approximate the posterior of state and noise covariance, we utilize VB method. Besides, the cubature integration is introduced to handle the nonlinear system. According to simulation and real dataset experiments, we validate that RVB-ACKF method can deal with the time-variant heavy-tailed noise very well. For our future work, we will utilize this VB noise adaptive algorithm for visual inertial odometry where filter-based algorithms are used to solve the problem of data fusion between camera and inertial measurement unit.

References

[1] DURRANT-WHYTE H, BAILEY T. Simultaneous localization and mapping: Part I [J]. *IEEE Robotics & Automation Magazine*, 2006, **13**(2): 99-107.

[2] BAILEY T, DURRANT-WHYTE H. Simultaneous localization and mapping (SLAM): Part II [J]. *IEEE Robotics & Automation Magazine*, 2006, **13**(3): 108-117.

[3] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age [J]. *IEEE Transactions on Robotics*, 2016, **32**(6): 1309-1332.

[4] MOURIKIS A I, ROUMELIOTIS S I. A multi-state constraint Kalman filter for vision-aided inertial navigation [C]//*IEEE International Conference on Robotics and Automation*. Rome, Italy: IEEE, 2007: 3565-3572.

[5] BLOESCHM, BURRI M, OMARI S, et al. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback [J]. *The International Journal of Robotics Research*, 2017, **36**(10): 1053-1072.

[6] SHOJAIE K, SHAHRI A M. Iterated unscented SLAM algorithm for navigation of an autonomous mobile robot [C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nice, France: IEEE, 2008: 1582-1587.

[7] WANG H J, FU G X, LI J, et al. An adaptive UKF based SLAM method for unmanned underwater vehicle [J]. *Mathematical Problems in Engineering*, 2013, **2013**: 605981.

[8] BLOESCH M, OMARI S, HUTTER M, et al. Robust visual inertial odometry using a direct EKF-based approach [C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*. Hamburg, Germany: IEEE, 2015: 298-304.

[9] XIA J Z, IQBAL U, NOURELDIN A, et al. Adaptive square-root CKF based SLAM algorithm for indoor

UGVs [C]//*IEEE International Conference on Mechatronics and Automation*. Takamatsu, Japan: IEEE, 2017: 1942-1946.

[10] CHANDRA K P B, GU D W, POSTLETHWAITE I. Cubature Kalman filter based localization and mapping [J]. *IFAC Proceedings Volumes*, 2011, **44**(1): 2121-2125.

[11] ENGEL J, KOLTUN V, CREMERS D. Direct sparse odometry [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, **40**(3): 611-625.

[12] MUR-ARTAL R, MONTIEL J M M, TARDÓS J D. ORB-SLAM: A versatile and accurate monocular SLAM system [J]. *IEEE Transactions on Robotics*, 2015, **31**(5): 1147-1163.

[13] FORSTER C, PIZZOLI M, SCARAMUZZA D. SVO: Fast semi-direct monocular visual odometry [C]//*IEEE International Conference on Robotics and Automation*. Hong Kong, China: IEEE, 2014: 15-22.

[14] SMITH R, SELF M, CHEESEMAN P. Estimating uncertain spatial relationships in robotics [C]//*IEEE International Conference on Robotics and Automation*. Raleigh, NC, USA: IEEE, 1987: 850-850.

[15] WILLIAMS B, CUMMINS M, NEIRA J, et al. A comparison of loop closing techniques in monocular SLAM [J]. *Robotics and Autonomous Systems*, 2009, **57**(12): 1188-1197.

[16] MONTERNERLO M, THRUN S, KOLLER D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem [C]//*The Eighteenth National Conference on Artificial Intelligence and the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, California, USA: AAAI, 2002: 593-598.

[17] JULIER S, UHLMANN J, DURRANT-WHYTE H F. A new method for the nonlinear transformation of means and covariances in filters and estimators [J]. *IEEE Transactions on Automatic Control*, 2000, **45**(3): 477-482.

[18] ARASARATNAM I, HAYKIN S. Cubature Kalman filters [J]. *IEEE Transactions on Automatic Control*, 2009, **54**(6): 1254-1269.

[19] GIL A, REINOSO Ó, MOZOS Ó M, et al. Improving data association in vision-based SLAM [C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE, 2006: 2076-2081.

[20] NAKABAYASHI A, UENO G. An extension of the ensemble Kalman filter for estimating the observation error covariance matrix based on the variational Bayes's method [J]. *Monthly Weather Review*, 2017, **145**(1): 199-213.

[21] JANCZAK D. Estimation method for measurements with heavy-tailed noise variance [J]. *Przegld Elektrotechniczny*, 2017, **93**(12): 24-27.

[22] DONG P, JING Z L, LEUNG H, et al. Variational Bayesian adaptive cubature information filter based on Wishart distribution [J]. *IEEE Transactions on Automatic Control*, 2017, **62**(11): 6051-6057.

[23] DONG P, JING Z L, LEUNG H, et al. Robust consensus nonlinear information filter for distributed sensor networks with measurement outliers [J]. *IEEE Transactions on Cybernetics*, 2019, **49**(10): 3731-3743.

[24] DONG P, JING Z L, SHEN K, et al. A distributed consensus filter for sensor networks with heavy-tailed measurement noise [J]. *Science China Information Sciences*, 2018, **61**(11): 119201.

[25] SHEN K, JING Z L, DONG P. A consensus nonlinear filter with measurement uncertainty in distributed sensor networks [J]. *IEEE Signal Processing Letters*, 2017, **24**(11): 1631-1635.

[26] SÄRKKÄ S, HARTIKAINEN J. Non-linear noise adaptive Kalman filtering via variational Bayes [C]//*IEEE International Workshop on Machine Learning for Signal Processing*. Southampton, UK: IEEE, 2013: 1-6.

[27] SÄRKKÄ S, NUMMENMAA A. Recursive noise adaptive Kalman filtering by variational Bayesian approximations [J]. *IEEE Transactions on Automatic Control*, 2009, **54**(3): 596-600.

[28] AGAMENNONI G, NIETO J I, NEBOT E M. Approximate inference in state-space models with heavy-tailed noise [J]. *IEEE Transactions on Signal Processing*, 2012, **60**(10): 5024-5037.

[29] XU W J. Research on Bayesian filters based simultaneous localization and mapping algorithms for mobile robots [D]. Hangzhou, China: Zhejiang University, 2016 (in Chinese).

[30] ZHANG S Y, DONG P, JING Z L. Adaptive cubature Kalman filtering SLAM algorithm based on variational Bayes [J]. *Journal of Harbin Institute of Technology*, 2019, **51**(4): 12-18 (in Chinese).

[31] WEINSTOCK R. Calculus of variations: With applications to physics and engineering [M]. New York, USA: Dover Publications, Inc, 1974.

[32] GELMAN A, CARLIN J B, STERN H S, et al. Bayesian data analysis [M]. London, UK: Chapman & Hall, 1995.

[33] KULLBACK S, LEIBLER R A. On information and sufficiency [J]. *Annals of Mathematical Statistics*, 1951, **22**(1): 79-86.

[34] CARVALHO C, WEST M. Dynamic matrix-variate graphical models [J]. *Bayesian Analysis*, 2007, **2**(1): 69-98.

[35] BAILEY T, NIETO J, GUIVANT J, et al. Consistency of the EKF-SLAM algorithm [C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE, 2006: 3562-3568.

[36] MARTINEZ-CANTIN R, CASTELLANOS J A. Unscented SLAM for largescale outdoor environments [C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*. Edmonton, Alberta, Canada: IEEE, 2005: 3427-3432.

[37] LI X, FENG Y B, HUANG R H, et al. The application of square-root cubature Kalman filter in SLAM for underwater robot [C]//*2017 Chinese Automation Congress* (*CAC*). Jinan, china: IEEE, 2017: 2183-2187.