doi:10.3772/j.issn.1006-6748.2016.01.006

Robust SLAM using square-root cubature Kalman filter and Huber's GM-estimator①

Xu Weijun(徐巍军)[*,**], Jiang Rongxin②[*,**], Xie Li[*,**], Tian Xiang[*,**], Chen Yaowu[*,**]
([*] Institute of Advanced Digital Technology and Instrumentation, Zhejiang University, Hangzhou 310027, P. R. China)
([**] Zhejiang Provincial Key Laboratory for Network Multimedia Technologies, Zhejiang University, Hangzhou 310027, P. R. China)

Abstract

Mobile robot systems performing simultaneous localization and mapping (SLAM) are generally plagued by non-Gaussian noise. To improve both accuracy and robustness under non-Gaussian measurement noise, a robust SLAM algorithm is proposed. It is based on the square-root cubature Kalman filter equipped with a Huber's generalized maximum likelihood estimator (GM-estimator). In particular, the square-root cubature rule is applied to propagate the robot state vector and covariance matrix in the time update, the measurement update and the new landmark initialization stages of the SLAM. Moreover, gain weight matrices with respect to the measurement residuals are calculated by utilizing Huber's technique in the measurement update step. The measurement outliers are suppressed by lower Kalman gains as merging into the system. The proposed algorithm can achieve better performance under the condition of non-Gaussian measurement noise in comparison with benchmark algorithms. The simulation results demonstrate the advantages of the proposed SLAM algorithm.

Key words: square-root cubature Kalman filter, simultaneous localization and mapping (SLAM), Huber's GM-estimator, robustness

0 Introduction

Simultaneous localization and mapping (SLAM) is a fundamental issue in the autonomous robot systems designed to realize more complex and advanced tasks, such as underground mining, planetary exploration, and disaster rescue. The objective of SLAM is to incrementally build a map of the unknown environment while concurrently using this map to localize the robot[1].

The nonlinear discrete-time state-space model was typically formulated in the SLAM problem with Gaussian noise. The most popular filter implemented for SLAM is extended Kalman filter (EKF)[2]. However, EKF approach for SLAM tends to be inconsistent due to the accumulation of linearization error. The sigma-point Kalman filters (SPKF) which achieve second-order or higher accuracy have been proven to be far superior to standard EKF. Among the family of SPKF-class estimators, the unscented Kalman filter (UKF) and the cubature Kalman filter (CKF) have shown the capability to reduce linearization error effectively, and therefore are used in SLAM algorithms[3,4]. Especially, the third-order cubature rule of the CKF is claimed to be more theoretically justified and more accurate in mathematical terms than the unscented transformation of the UKF[5].

However, the distribution of measurement noise in practical systems may deviate from the commonly assumed Gaussian model[6]. This non-Gaussian noise model is usually characterized by thick-tailed probability distributions and randomly appearing outliers, which may originate from glint noise of reflection[7] or are induced by unanticipated environment turbulence, temporary sensor failure, and incorrect modelling[8]. To deal with non-Gaussian noise model in the SLAM implementations, several methods have been proposed. For example, the Rao-Blackwellized particle filter based Fast SLAM[9] estimated the state posterior of arbitrary probability distribution by a finite number of particle samples. However, the algorithm will be computationally intensive in situations where the state vectors are high-dimensional. The H_∞-filter based SLAM algorithms[4,10] are also able to estimate the state perturbed with non-Gaussian noise by treating the noise as unknown-but-bounded quantities. However the algo-

① Supported by the National High Technology Research and Development Program of China (2010AA09Z104), and the Fundamental Research Funds of the Zhejiang University (2014FZA5020).
② To whom correspondence should be addressed. E-mail: rongxinj@zju.edu.cn
Received on Dec. 3, 2014

rithms are prone to be diverged from in the presence of random outliers.

In essence, the conventional Kalman type filters belong to the recursive minimum l_2-norm or least mean-square technique, and the performance of the filters quickly degrades in the presence of outliers and thick-tailed noise. In contrast, Huber's GM-estimator[11] is an estimation technique that gains robustness by optimizing a cost function represented in a combined minimum l_1-and l_2-norm. Using this estimator, the effect of the outliers is suppressed by down-weighting the normalized measurement residuals that are larger than a given threshold. Huber's method has also shown its high robustness by integrating it into various Kalman type filters[12-14].

Taking the advantage of square-root cubature rule's numerical stability, a robust SLAM algorithm based on SCKF is proposed. Moreover, to accommodate the non-Gaussian measurement noise model, Huber's GM-estimator is further introduced to improve the measurement update for each revisited landmark. Simulation results are provided to illustrate the effectiveness of the proposed algorithm in the complex scenarios with non-Gaussian measurement noise models.

1 Problem formulation

Consider the general discrete-time nonlinear SLAM system with the process model and measurement model

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{v}_{k-1} \\ \boldsymbol{z}_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{w}_k \end{aligned} \tag{1}$$

where $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are specific known nonlinear functions; $\boldsymbol{x}_k = [\boldsymbol{x}_{v,k}, \boldsymbol{m}_k]$ is the state vector consisting of the robot pose $\boldsymbol{x}_{v,k}$ and varying-size map of landmark $\boldsymbol{m}_k$ at time step k; $\boldsymbol{u}_{k-1}$ is the control input of the proprioceptive sensors; z_k is the measurement obtained from the on-board sensors; $\boldsymbol{v}_{k-1}$ and $\boldsymbol{w}_k$ are additive process and measurement noise, respectively. The noises are assumed to be mutually independent Gaussian random variables with zero mean and covariances $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$, respectively.

In this study, the third order spherical-radial cubature rule is utilized to approximate the nonlinear Gaussian integral with a set of $2N$ (N is the dimensionality of the state vector to be estimated) equally weighted cubature points. The set of cubature points is determined by $\{\boldsymbol{\xi}_i, \boldsymbol{w}_i\}$, where $\boldsymbol{\xi}_i$ is the i-th element and w_i is the corresponding weight factor:

$$\boldsymbol{\xi}_i = \begin{cases} \sqrt{N}\boldsymbol{e}_i, & i = 1, \cdots, N \\ -\sqrt{N}\boldsymbol{e}_{i-N}, & i = N+1, \cdots, 2N \end{cases} \tag{2}$$

where $\boldsymbol{e}_i$ denotes a unit vector in the direction of the coordinate axis i; the weight factor is simply calculated as $w_i = \dfrac{1}{2N}$ for all points.

2 SLAM based on square-root cubature rule

The objective of the SLAM algorithm is to keep the system state estimate up to date by recursively evolving with the time update, measurement update and new landmark initialization steps. The square-root cubature rule is applied to all the SLAM steps to propagate the square-root factors of the predictive and posterior covariance directly. The complete procedure for the SCKF based SLAM (SCKF-SLAM) is derived in this section.

2.1 Time update step

When the robot moves according to the control signals from the proprioceptive sensors, the robot state has to be predicted based on its prior estimate of the state and the control inputs. As the process noise is non-additive, the robot state vector and its covariance squared root factor are required to be augmented as

$$\hat{\boldsymbol{x}}^a_{k-1} = \begin{bmatrix} \hat{\boldsymbol{x}}_{k-1} \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{S}^a_{k-1} = \begin{bmatrix} \boldsymbol{S}_{k-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_{Q,k-1} \end{bmatrix} \tag{3}$$

where $\boldsymbol{S}_{k-1}$ is the upper Cholesky factor of $\boldsymbol{P}_{k-1}$ such that $\boldsymbol{P}_{k-1} = \boldsymbol{S}^T_{k-1}\boldsymbol{S}_{k-1}$ and $\boldsymbol{S}_{Q,k-1}$ is the upper Cholesky factor of $\boldsymbol{Q}_{k-1}$ with $\boldsymbol{Q}_{k-1} = \boldsymbol{S}^T_{Q,k-1}\boldsymbol{S}_{Q,k-1}$. $\hat{\boldsymbol{x}}_{k-1}$ and $\boldsymbol{P}_{k-1}$ are the prior mean and covariance matrix of the robot state, respectively.

The augmented state vector and its covariance squared root factor are used to determine a set of $2N_1$ cubature points which are calculated by

$$\boldsymbol{X}_{i,k-1} = \boldsymbol{S}^a_{k-1}\boldsymbol{\xi}_i + \hat{\boldsymbol{x}}^a_{k-1}, \quad i = 1, 2, \cdots, N_1 \tag{4}$$

where N_1 is the size of the augmented state vector. Each cubature point is propagated through the process model:

$$\boldsymbol{X}^*_{i,k|k-1} = \boldsymbol{f}(\boldsymbol{X}_{i,k-1}, \boldsymbol{u}_{k-1}), \quad i = 1, 2, \cdots, N_1 \tag{5}$$

where $\boldsymbol{u}_{k-1}$ is the control input. Note that the dimensionality of each propagated cubature point is the same as the original robot state vector, rather than the augmented one. The predicted robot state mean is estimated by

$$\hat{\boldsymbol{x}}_{k|k-1} = \frac{1}{2N_1}\sum_{i=1}^{2N_1} \boldsymbol{X}^*_{i,k|k-1} \tag{6}$$

The square-root factor of the predicted robot covariance matrix is found by performing the QR decomposition

$$\boldsymbol{S}_{k|k-1} = \mathrm{qr}(\boldsymbol{\chi}^{*}_{k|k-1}) \tag{7}$$

where $\boldsymbol{\chi}^{*}_{k|k-1}$ is a column matrix and each column $\boldsymbol{\chi}^{*}_{i,k|k-1}$ is calculated as

$$[\boldsymbol{\chi}^{*}_{i,k|k-1}]_{i=1,2,\cdots,2N_1} = [\frac{1}{2n}(\hat{\boldsymbol{X}}^{*}_{i,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1})]_{i=1,2,\cdots,2N_1} \tag{8}$$

2.2 Measurement update step

Each time the robot revisits the landmarks that have already been mapped by means of its on-board sensors, the measurements are exploited to correct the estimate of both the robot state and the map of the landmarks. Landmark measurements are processed sequentially with a serial of update steps.

The cubature points with respect to the current state are evaluated by

$$\boldsymbol{X}_{i,k|k-1} = \boldsymbol{S}_{k|k-1}\boldsymbol{\xi}_i + \hat{\boldsymbol{x}}_{k|k-1}, \quad i = 1,2,\cdots,2N_2 \tag{9}$$

where N_2 is the size of the current state vector, $\hat{\boldsymbol{x}}_{k|k-1}$ is the predicted state vector, and $\boldsymbol{S}_{k|k-1}$ is the upper Cholesky factor of the corresponding predicted covariance matrix.

A specific landmark measurement depends only on the predicted robot pose and the particular landmark's state, which are parts of the state vector. The propagated cubature points are evaluated with the measurement model by

$$\boldsymbol{Z}^{*}_{i,k|k-1} = \boldsymbol{h}(\boldsymbol{X}_{i,k|k-1}), \quad i = 1,2,\cdots,2N_2 \tag{10}$$

The predicted measurement is estimated as

$$\hat{\boldsymbol{z}}_{k|k-1} = \frac{1}{2N_2}\sum_{i=1}^{2N_2}\boldsymbol{Z}^{*}_{i,k|k-1} \tag{11}$$

The square-root factor of the innovation matrix is found by performing the QR decomposition:

$$\boldsymbol{S}_{zz,k|k-1} = \mathrm{qr}([\boldsymbol{Z}_{k|k-1} \quad \boldsymbol{S}_{R,k}]) \tag{12}$$

where $\boldsymbol{S}_{R,k}$ is the upper Cholesky factor of $\boldsymbol{R}_k$, and $\boldsymbol{Z}_{k|k-1}$ is a column matrix with each column calculated as

$$[\boldsymbol{Z}_{i,k|k-1}]_{i=1,2,\cdots,2N_2} = [\frac{1}{2N_2}(\boldsymbol{Z}^{*}_{i,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1})]_{i=1,2,\cdots,2N_2} \tag{13}$$

The cross-covariance matrix is obtained by matrix multiplying:

$$\boldsymbol{P}_{xz,k|k-1} = \boldsymbol{\chi}_{k|k-1}\boldsymbol{Z}^{\mathrm{T}}_{k|k-1} \tag{14}$$

where $\boldsymbol{\chi}_{k|k-1}$ is a column matrix with each column calculated as

$$[\boldsymbol{\chi}_{i,k|k-1}]_{i=1,2,\cdots,2N_2} = [\frac{1}{2N_2}(\boldsymbol{X}_{i,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1})]_{i=1,2,\cdots,2N_2} \tag{15}$$

The Kalman gain of the SCKF is calculated by

$$\boldsymbol{K} = (\boldsymbol{P}_{xz,k|k-1}/\boldsymbol{S}^{T}_{zz,k|k-1})/\boldsymbol{S}_{zz,k|k-1} \tag{16}$$

The corrected state vector and corresponding square-root factor of the covariance matrix are finally obtained by

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}(z_k - \hat{z}_{k|k-1}) \tag{17}$$

$$S_{k|k} = \mathrm{qr}([\boldsymbol{\chi}_{k|k-1} - \boldsymbol{K}\boldsymbol{Z}_{k|k-1} \quad \boldsymbol{K}\boldsymbol{S}_{R,k}]) \tag{18}$$

2.3 New landmark initialization step

Landmark initialization happens when the robot detects a number of landmarks for the first time and decides to incorporate them into the map. The robot state vector and its covariance square-root factor are augmented with each new landmark measurement:

$$\hat{\boldsymbol{x}}^{a}_{k} = \begin{bmatrix}\hat{\boldsymbol{x}}_k \\ \boldsymbol{z}_{j,k}\end{bmatrix}, \ \boldsymbol{S}^{a}_{k} = \begin{bmatrix}\boldsymbol{S}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_{R,k}\end{bmatrix} \tag{19}$$

where $\boldsymbol{z}_{j,k}$ is the measurement of the j-th newly detected landmarks. $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{S}_k$ are the robot state vector and the upper Cholesky factor of the robot covariance matrix, respectively.

The expected position of the new landmark is calculated analogously using the cubature rule. A set of $2N_3$ cubature points is calculated to represent the probability density of the augmented state:

$$\boldsymbol{X}_{i,k} = \boldsymbol{S}^{a}_{k}\boldsymbol{\xi}_i + \hat{\boldsymbol{x}}^{a}_{k}, \ i = 1,2,\cdots,2N_3 \tag{20}$$

where N_3 is the size of the augmented state vector. Each cubature point is propagated through the nonlinear inverse measurement model as

$$\boldsymbol{X}^{*}_{i,k} = \boldsymbol{h}^{-1}(\boldsymbol{X}_{i,k}), \ i = 1,2,\cdots,2N_3 \tag{21}$$

where $\boldsymbol{h}^{-1}$ denotes the nonlinear inverse observation function, which transforms the new landmark measurements into the landmark coordinates in the global frame.

The predicted mean of the augmented state is calculated by

$$\hat{\boldsymbol{x}}_k = \frac{1}{2N_3}\sum_{i=1}^{2N_3}\boldsymbol{X}^{*}_{i,k}, \ i = 1,2,\cdots,2N_3 \tag{22}$$

The square-root factor of the corresponding covariance matrix is also obtained with the QR decomposition

$$\boldsymbol{S}_k = \mathrm{qr}(\boldsymbol{\chi}^{*}_{k}) \tag{23}$$

where $\boldsymbol{\chi}^{*}_{k}$ is a column matrix and each column $\boldsymbol{\chi}^{*}_{i,k}$ is calculated as

$$[\boldsymbol{\chi}^{*}_{k}]_{i=1,2,\cdots,2N_3} = [\frac{1}{2N_3}(\hat{\boldsymbol{X}}^{*}_{i,k} - \hat{\boldsymbol{x}}_k)]_{i=1,2,\cdots,2N_3} \tag{24}$$

3 Huber-based SCKF SLAM

This section presents the application of Huber's GM-estimator to improve the robustness of the SCKF-SLAM. The measurement update has to be recast as a

linear regression formulation, where the measurement equation is combined with a virtual linear state transition equation. The virtual equation demonstrates the linear relationship between the true state and its predicted estimation $\hat{\boldsymbol{x}}_{k|k-1}$ obtained from the time update step:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{x}_k - \boldsymbol{\delta}_{k|k-1} \tag{25}$$

where $\boldsymbol{\delta}_{k|k-1}$ is an unknown error vector.

The nonlinear measurement function is also rewritten as a linear form by approximating:

$$\boldsymbol{z}_k \approx \hat{\boldsymbol{z}}_{k|k-1} + \boldsymbol{H}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) \tag{26}$$

where measurement matrix $\boldsymbol{H}_k$ is calculated by the predicted state covariance matrix and the cross-covariance matrix:

$$\boldsymbol{H}_k = (\boldsymbol{P}_{k|k-1}^{-1}\boldsymbol{P}_{xz,k|k-1})^{\mathrm{T}} \tag{27}$$

By combining Eqs(25) and (26) together, the linear regression equation can be obtained as

$$\begin{bmatrix} \boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1} + \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} \\ \hat{\boldsymbol{x}}_{k|k-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_k \\ \boldsymbol{I} \end{bmatrix}\boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{\delta}_{k|k-1} \end{bmatrix} \tag{28}$$

The error covariance matrix with respect to the far right component of the above equation is given by

$$E(\varepsilon\varepsilon^{\mathrm{T}}) = \begin{bmatrix} \boldsymbol{R}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{k|k-1} \end{bmatrix} = \boldsymbol{S}_k\boldsymbol{S}_k^{\mathrm{T}} \tag{29}$$

where $\boldsymbol{S}_k$ is the Cholesky factor of the error covariance.

Several nominal matrices are defined through left multiplying each component of Eq. (28) by the inverse Cholesky factor $\boldsymbol{S}_k^{-1}$:

$$\boldsymbol{y}_k = \boldsymbol{S}_k^{-1}\begin{bmatrix} \boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1} + \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} \\ \hat{\boldsymbol{x}}_{k|k-1} \end{bmatrix} \tag{30}$$

$$\boldsymbol{M}_k = \boldsymbol{S}_k^{-1}\begin{bmatrix} \boldsymbol{H}_k \\ \boldsymbol{I} \end{bmatrix} \tag{31}$$

$$\boldsymbol{\xi}_k = \boldsymbol{S}_k^{-1}\begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{\delta}_{k|k-1} \end{bmatrix} \tag{32}$$

The linear regression equation can be transformed to a compact form as

$$\boldsymbol{y}_k = \boldsymbol{M}_k\boldsymbol{x}_k + \boldsymbol{\xi}_k \tag{33}$$

Huber's GM-estimator is used to find the solution to this linear regression problem, by minimizing the cost function as

$$\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}_k} \boldsymbol{J}(\boldsymbol{x}_k) = \arg\min \sum_{i=1}^{N_2+\dim(\boldsymbol{z}_k)} \boldsymbol{\rho}(r_i) \tag{34}$$

where $\boldsymbol{\rho}(\cdot)$ is a symmetric, positive-define score function with a unique minimum at zero, $\dim(\boldsymbol{z}_k)$ is the size of a single landmark measurement, r_i is the i-th component of the residual between observation $\boldsymbol{y}_k$ and its fitted value $\boldsymbol{M}_k\boldsymbol{x}_k$, i.e., $r_i = [\boldsymbol{M}_k\boldsymbol{x}_k - \boldsymbol{y}_k]$. The solution of Eq. (34) is determined by the following implicit equation:

$$\sum_{i=1}^{N_2+\dim(\boldsymbol{z}_k)} \psi(r_i)\frac{\partial r_i}{\partial \boldsymbol{x}} = 0 \tag{35}$$

where the derivative $\psi(r_i) = \mathrm{d}\boldsymbol{\rho}(r_i)/\mathrm{d}\boldsymbol{x}$ is known as the influence function. By defining a weight function $\boldsymbol{w}(r_i) = \psi(r_i)/r_i$ and an associated diagonal weight matrix $\boldsymbol{W} = \mathrm{diag}[\boldsymbol{w}(r_i)]$, it can be written in a matrix form as

$$\boldsymbol{M}_k^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{M}_k\boldsymbol{x}_k - \boldsymbol{y}_k) = 0 \tag{36}$$

This equation can be solved by using the iterated reweighted least-square algorithm, where the weight matrix $\boldsymbol{W}$ is recalculated in each iteration and is used in the next iteration. This process is represented as

$$\hat{\boldsymbol{x}}_{k|k}^{(j+1)} = (\boldsymbol{M}_k^{\mathrm{T}}\boldsymbol{W}^{(j)}\boldsymbol{M}_k)^{-1}\boldsymbol{M}_k^{\mathrm{T}}\boldsymbol{W}^{(j)}\boldsymbol{y}_k \tag{37}$$

where the superscript j denotes the iteration index. The initial value of the weight matrix $\boldsymbol{W}$ is given as an identity matrix. And then the initial state estimation $\hat{\boldsymbol{x}}^{(0)}$ is the least-square solution:

$$\hat{\boldsymbol{x}}_{k|k}^{(0)} = (\boldsymbol{M}_k^{\mathrm{T}}\boldsymbol{M}_k)^{-1}\boldsymbol{M}_k^{\mathrm{T}}\boldsymbol{y}_k \tag{38}$$

When Eq. (37) is converged, the final value of the corrected estimation of the state vector is achieved and the corresponding corrected covariance matrix is computed with the converged weight matrix:

$$\boldsymbol{P}_{k|k} = (\boldsymbol{M}_k^{\mathrm{T}}\boldsymbol{W}\boldsymbol{M}_k)^{-1} \tag{39}$$

Finally, the corrected square-root factor of the covariance matrix is achieved by performing the Cholesky factorization:

$$\boldsymbol{S}_{k|k} = \mathrm{CHOL}(\boldsymbol{P}_{k|k}) \tag{40}$$

The score function of the GM-estimator should be chosen to yield robustness against non-Gaussian measurement noise. Note that if the score function is selected as a l_2-norm one, i.e., $\boldsymbol{\rho}(r_i) = r_i^2/2$, the solution to the linear regression problem is equivalent to that of the standard Kalman type filters. A number of robust score functions have been proposed such as the Huber function, thelogistic function[15] and Hampel's redescending function[16].

The pseudo-code of the proposed Huber based SLAMalgorithm (HSCKF-SLAM) is summarized in "Algorithm 1".

Algorithm 1 HSCKF-SLAM Algorithm

Require: *Initial robot state mean* $\boldsymbol{x}_0$ *and covariance* $\boldsymbol{P}_0$

Main Loop:
for $k = 1, 2, \cdots, T$ **do**
  Time-update:
  Compute the predicted robot pose $\hat{\boldsymbol{x}}_{v,k|k-1}$ and its Cholesky factor of the covariance $\boldsymbol{S}_{v,k|k-1}$ via (4) – (8);
  if new measurements received **then**
    Perform data association algorithm;
  end if
  Measurement update:
  for measurements of revisited landmarks **do**
    Compute the cross-covariance matrix $\boldsymbol{P}_{xz,k|k-1}$ via (9)-(15);
    Compute the corrected system state $\hat{\boldsymbol{x}}_{k|k}$ and the corresponding Cholesky factor of the covariance $\boldsymbol{S}_{k|k}$ via (25)-(40);
  end for
  New Landmark initialization:
  for measurements of newly visited landmarks **do**
    Perform landmark initialization step via (20)-(24);
  end for
end for

4 Simulations and results

A series of simulations have been conducted to evaluate the performance of the proposed HSCKF-SLAM algorithm in comparison with the UKF-SLAM and the SCKF-SLAM. The publicly available UKF-SLAM simulator ① is modified as a benchmark platform. The other two algorithms have been implemented in Matlab R2012a on a 2.9GHz Intel Corei7-3520M Processor. As presented in Fig. 1(a), the robot is assumed to move along the predefined trajectory in a rectangular plane. The robot starts from the origin of the global frame and detectsnearby landmarks with a laser sensor. The additive measurement noise is assumed to follow a Gaussian mixture distribution of the form:

$$w_k \sim (1-\alpha)N(w_k;0,\sigma_1^2) + \alpha N(w_k;0,\sigma_2^2)$$
$$0 \leqslant \alpha \leqslant 1,\ \sigma_2 = \beta\sigma_1$$

where α represents the noise model contamination, σ_1 and σ_2 are the standard deviations of the Gaussian mixture components, and β denotes the ratio between them. The process noise is 0.2m/s in wheel velocity and 2° in steering angle. Other simulation parameters are listed in Table 1.

Table 1 Simulation parameters.

Parameters	value	Parameters	Value
Wheel base length L	4(m)	Process time interval $T_\triangle$	0.025(s)
Wheel velocity V_k	3(m/s)	Control frequency f_c	40(Hz)
Maximum steering angle $G_{\max}$	±30(°)	Observation frequency f_o	5(Hz)
Maximum observation range $r_{\max}$	30(m)	View field of laser sensor F_v	0(°) ~ 180(°)

① https://svn.openslam.org/data/svn/bailey-slam/

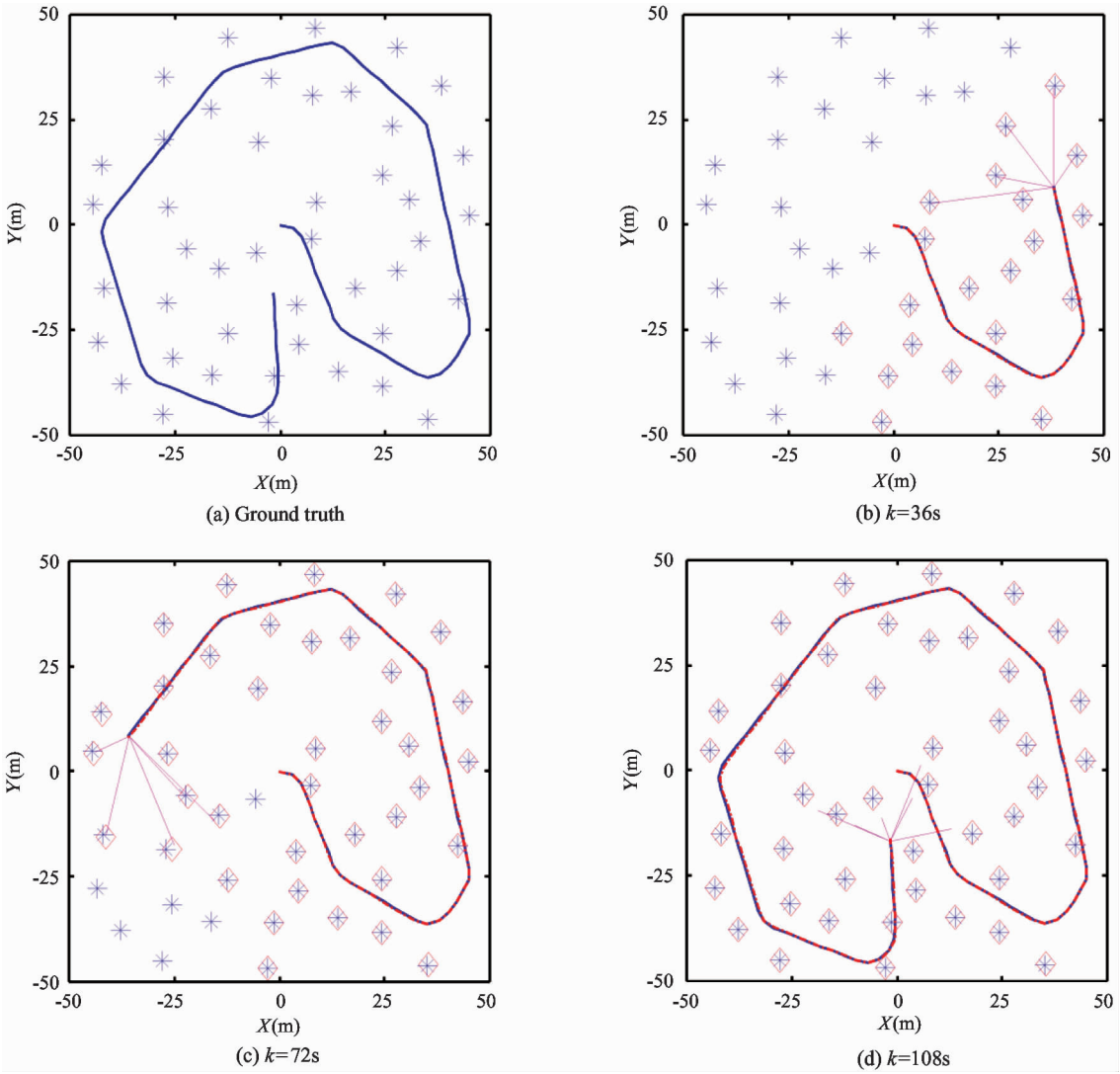

Fig. 1 The simulation results. True landmark (*) and robot path (thick solid lines), estimated landmark (diamonds) and robot path (dotted lines), laser measurement (thin solid lines).

In order to evaluate the performance of the proposed algorithm under various measurement noise conditions, two simulation scenarios with different measurement noise models are considered: a Gaussian mixture contaminated noise model and an outlier contaminated noise model. For a fair comparison, all the SLAM algorithms are carried out with the same simulation parameters except that of the measurement noise. 200 independent Monte Carlo simulation runs are conducted for each simulation scenario.

4.1 Results in Gaussian mixture contaminated noise case

In this simulation scenario, the measurement noise is assumed as a contaminated Gaussian model with two independent Gaussian mixtures. The standard deviation of the main mixture component σ_1 is set to 0.2m in range and 2° in bearing. Fig. 1(b) ~ (d) show the results of a typical simulation run where the contamination and ratio parameters are set to 0.3 and 10, respectively. These plots indicate that both the robot trajectories and landmarks are estimated accurately at different time steps (k = 36s, 72s, and 108s) by the proposed algorithm. The Kalman gain weights under different measurement residuals are also presented in Fig. 2. It can be seen that the weights reach local minimums in the most time steps when either the range measurement residual or the bearing measurement residual is a high peak of the curve. Moreover, larger measurement residuals can bring about smaller weights. As a result, the measurement outliers are supressed to a great extent in the Kalman update process.

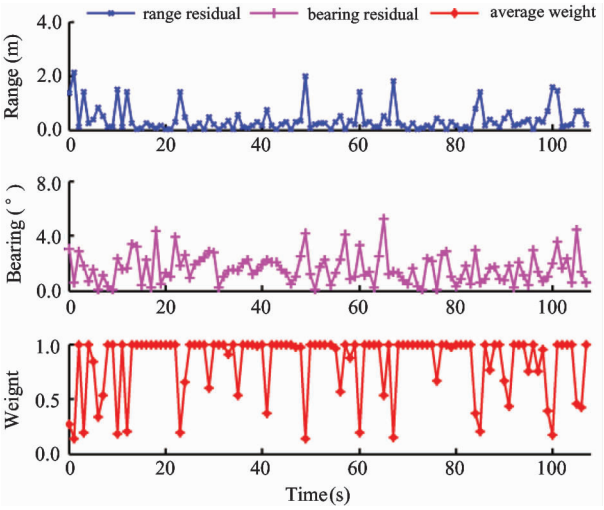

Fig. 2 Average Kalman gain weights under different measurement residuals

The effects of different combinations of parameters α and β on accuracy are illustrated in Fig. 3 and Fig. 4. Fig. 3 shows the relations between the average RMSE of the algorithms and contamination parameter α when ratio parameter β is fixed to 5. Similarly, the relations between average RMSE and ratio parameter β are shown in Fig. 4, where α is set to 0.4. As observed in the figures, the HSCKF-SLAM outperforms the other algorithms in all cases. Besides, the superiority of the HSCKF-SLAM is more obvious as the parameters increase. The results indicate that the Huber based update plays a more important role when the distribution of the non-Gaussian noise has thicker tails. The SCKF-SLAM exhibits slightly better performance than the UKF-SLAM due to the reason that SCKF can approximate the nonlinear functions in higher order than UKF. The unscented transformation is no longer numerically

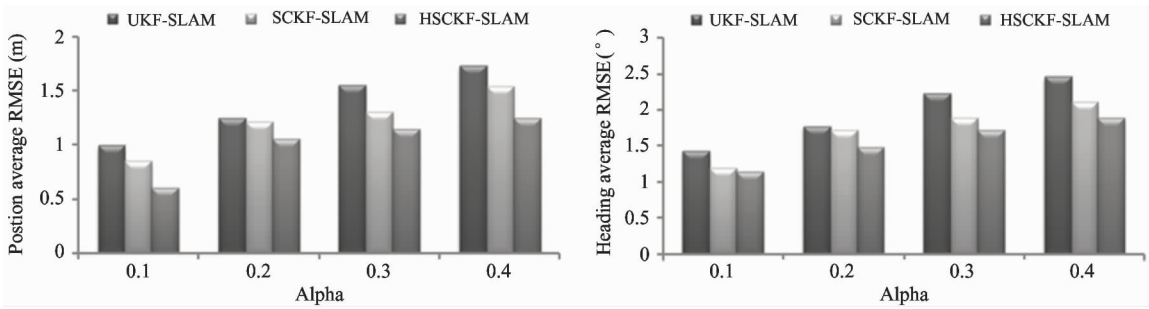

Fig. 3 Average RMSE under different contamination parameters

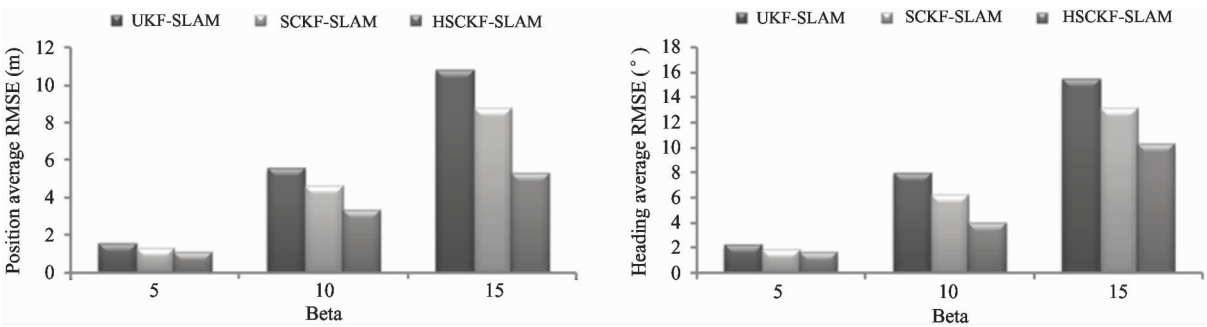

Fig. 4 Average RMSE under different ratio parameters

stable and the Cholesky decomposition of the state covariance encounters troubles in cases of extremely large noise.

4.2 Results in outlier contaminated noise case

In this simulation scenario, the basic measurement noise follows a Gaussian distribution by setting contamination parameter to 0. This Gaussian noise model is then contaminated by a number of random measurement outliers which are induced periodically. Totally 21 measurements are selected and biased by an offset [5m, 5°]. Fig. 5 depicts RMSE of the robot position and heading of the algorithms. It can be seen from the results that both the UKF-SLAM and the SCKF-SLAM suffer from estimation errors larger than the HSCKF-SLAM apparently. The position and heading RMSE are below 1m and 1.2° for HSCKF-SLAM. This proves that HSCKF-SLAM can detect all the measurement outliers and reduce their influence effectively. The average NEES of the outlier scenario are shown in Fig. 6, where the two horizontal dashed lines are plotted to mark the 95% two-side confidence re-

gion. It can be seen that both SCKF-SLAM and UKF-SLAM become inconsistent for all the time steps, while HSCKF-SLAM retains consistent for more than 75 time steps. These results demonstrate that by making use of Huber's update method, the conventional Kalman type filter is insensitive to the measurement outliers.

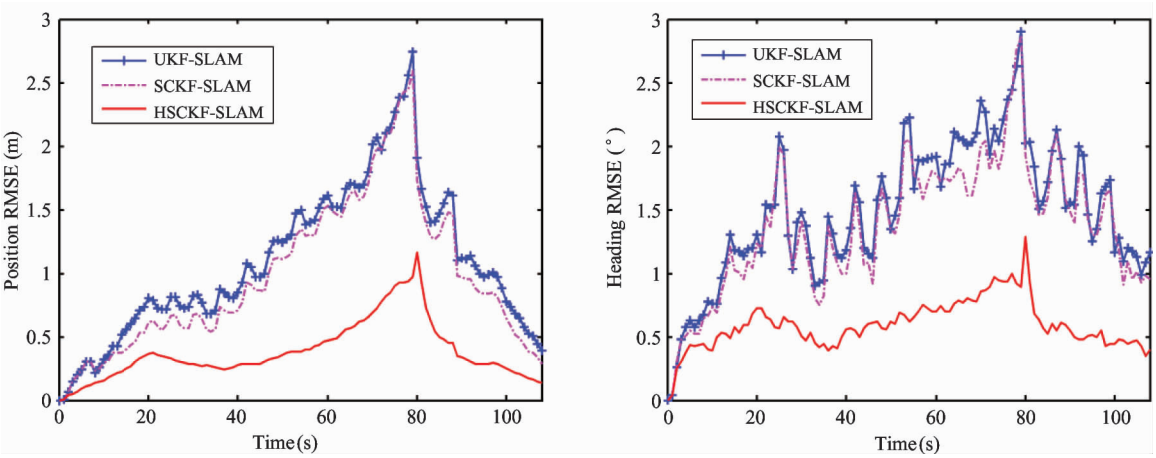

Fig. 5 Comparison of RMSE in outlier contaminated Gaussian measurement noise case

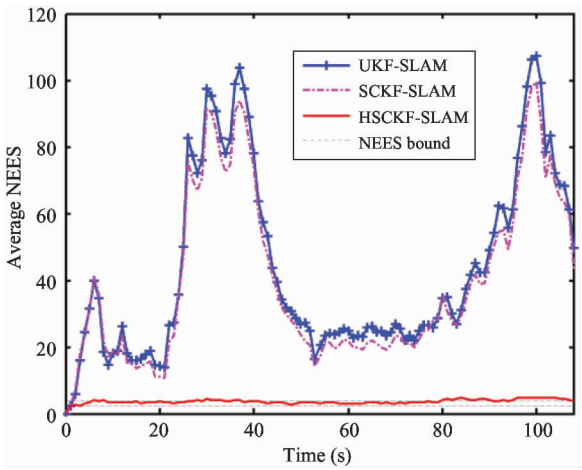

Fig. 6 Average NEES of the robot position in outlier case

4.3 Computational cost

The computational costs of these algorithms are also compared. As illustrated in Table 2, UKF-SLAM requires the minimum computational cost. SCKF-SLAM demands more running time because QR decompositions are employed to ensure numerical stability. HSCKF-SLAM takes the most computational effort due to the extra realization of robust linear regression in the measurement update stage. However, the increased average running time for one single update step with Huber's method is of the order of a 3ms. This increase is a worthwhile price to be paid for robustness and consistency. Besides, such a level of increase is often acceptable in real-time SLAM applications.

Table **2** Computational cost of algorithms

Algorithms	Average running time
UKF-SLAM	5.3(s)
SCKF-SLAM	5.7(s)
HSCKF-SLAM	7.1(s)

5 Conclusions

A robust SLAM algorithm based on SCKF and Huber's GM-estimator is proposed for robot systems with non-Gaussian measurement noises. The integration of a GM-estimator doesn't only retain the accurate merit of SCKF, but also provides an efficient way to work in non-Gaussian cases, with performance surpassing the benchmark algorithms in robustness and consistency. The influence of Huber's GM-estimator on the convergence rate and efficiency properties with different score functions will be further studied and optimized.

Reference

[1] Dissanayake M W M G, Newman P, Clark S, et al. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 2001, 17(3): 229-241

[2] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: Part I. *IEEE Robotics and Automation Magazine*, 2006, 13(2): 99-108

[3] Martinez-Cantin R, Castellanos J A. Unscented SLAM for large-scale outdoor environments. In: Proceedings of the International Conference on Intelligent Robots and Systems, Edmonton, Canada, 2005. 328-333

[4] Chandra K P B, Gu D, Postlethwaite I. A cubature H∞ filter and its square-root version. *International Journal of Control*, 2014, 87(4): 764-776

[5] Arasaratnam I, Haykin S. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 2009, 54(6): 1254-1269

[6] Gandhi M A, Mili L. Robust Kalman filter based on a generalized maximum-likelihood-type estimator. *IEEE Transactions on Signal Processing*, 2010, 58(5): 2509-2520

[7] Li X R, Jilkov V P. Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 2005, 41(4): 1255-1321

[8] Zoubir A M, Koivunen V, Chakhchoukh Y, et al. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Processing Magazine*, 2012, 29(4): 61-80

[9] Montemerlo M, Thrun S, Koller D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: Proceedings of the National Conference on Artificial Intelligence, Edmonton, Canada, 2002. 593-598

[10] Ahmad H, Namerikawa T. Feasibility study of partial observability in H_∞ filtering for robot localization and mapping problem. In: Proceedings of the 2010 American Control Conference, Baltimore, USA, 2010. 3980-3985

[11] Huber P J. Robust Statistics. New Jersey: John Wiley & Sons, 2004. 43-48

[12] Karlgaard C D, Schaub H. Huber-based divided difference filtering. *Journal of Guidance, Control, and Dynamics*, 2007, 30(3): 885-891

[13] Chang L, Hu B, Chang G, et al. Huber-based novel robust unscented Kalman filter. *IET Science, Measurement & Technology*, 2012, 6(6): 502-509

[14] Wang X, Cui N, Guo J. Huber-based unscented filtering and its application to vision-based relative navigation. *IET Radar, Sonar and Navigation*, 2010, 4(1): 134-141

[15] Coleman D, Holland P, Kaden N, et al. System of subroutines for iteratively reweighted least squares computations. *ACM Transactions on Mathematical Software*, 1980, 6(3): 327-336

[16] Hampel F R, Ronchetti E M, Rousseeuw P J, et al. Robust Statistics: the Approach Based on Influence Functions. Published Online, John Wiley & Sons, 2011. 307-341

Xu Weijun, born in 1985. He is a Ph. D. candidate in the College of Biomedical Engineering and Instrument Science of Zhejiang University. His main research fields arerobot simultaneous localization and mapping, multiple target tracking and multi-agent navigation.