APEM journal

Advances in Production Engineering & Management
Volume 15 | Number 1 | March 2020 | pp 31–43
https://doi.org/10.14743/apem2020.1.347

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Estimating the position and orientation of a mobile robot using neural network framework based on combined square-root cubature Kalman filter and simultaneous localization and mapping

Wang, D.[a],*, Tan, K.[a], Dong, Y.[a], Yuan, G.[b], Du, X.[b]

[a]College of Electronic Information and Electrical Engineering, Anyang Institute Of Technology, Anyang Henan, P.R. China
[b]College of Automation, Harbin Engineering University, Harbin Heilongjiang, P.R. China

A B S T R A C T

The real-time performance of target tracking, detection, and positioning behaves not well for non-Gaussian and nonlinear model with circumstance uncertainty. The weak observability of the system under large noise causes the algorithm unstable and slow to converge. A new estimation algorithm combining square-root cubature Kalman filter (SRCKF) with simultaneous localization and mapping (SLAM) is proposed. By connecting neural network weights, network input, functional types and ideal output network, the algorithm firstly update iteratively the SRCKF-SLAM state model and observation model, then conduct the cubature point estimate (weights) neural network framework. Thus, a point set better representing the target state and a more accurate state estimation are achieved, which can improve the filtering accuracy. This paper also estimates robot and characteristic states by filtering in groups. The simulation results showed that the proposed algorithm is feasible and effective. Compared with other filtering algorithms such as SRUKF and SRCDKF, it improves the estimation accuracy. Applying the new algorithm to the position filtering estimation of mobile robot can effectively reduce the positioning error, achieve high-precision tracking detection, and improve the accuracy of robot target detection.

A R T I C L E I N F O

Keywords:
Robot;
Mobile robot;
Square-root cubature Kalman filter;
Simultaneous localization and mapping;
Sensors;
Artificial neural networks;
Iteration update;
Filter estimate

**Corresponding author:*
lansejingling1988@126.com
(Wang, D.)

Article history:
Received 9 January 2020
Revised 15 March 2020
Accepted 17 March 2020

1. Introduction

During the navigation process, robot carries various navigation sensors to track and detect the target, and there is a certain error between true value and estimated position of target observed by the sensors. How to improve navigation and positioning accuracy, use appropriate system state models and observation models, and use more appropriate positioning estimation algorithms to reduce navigation errors and other issues have become the core issues in robot navigation and positioning. In the 1988s, foreign scholars proposed simultaneous localization and mapping technology [1], which uses a carrier to carry sensors to complete navigation and localization. This technology has successfully broken the traditional constraints of a priori maps and has been widely concerned and studied. The models in practical engineering applications are mostly non-linear models. The discrete observations of sensors are often used to estimate the continuous state of the target, random noise is filtered out, then state parameters of robot and position information of the target are solved. The method used is the extended Kalman filter algorithm(EKF)[2-3], which expands the Taylor series of non-linear function, and then ignores the higher-order part, and successfully performs the non-linear problem. Linearization, but at the same time introduces a cut-off error, which increases the estimation error. Juliear, Umhlman, Durrant-Whyte, etc. proposed an Unscented Kalman Filtering (UKF) algorithm[4-5]. Based on UT transformation, state quantity and covariance matrix of the system were estimated by weighted average. Then the time update and measurement update obtain the filtered system state quantity equation and its covariance matrix. However, due to the cumbersome matrix decomposition and inversion of the UKF algorithm, the positive definiteness of the state estimation covariance is difficult to guarantee. The cubature Kalman filter(CKF) [6] algorithm generates a new set of $2N$ point sets on the unit sphere after a set of N sampling points with weights undergoes a non-linear cubature spherical radial transformation. The set of points on the heavy unit sphere is then iteratively updated by the state equation to capture the relevant parameters such as the mean and variance of the state quantities. The cubature radial criterion is used to approximate the nonlinear model to integrate the probability density and finally obtain a higher-precision nonlinearity filter effect.

When the standard CKF algorithm solves the SLAM problem, the numerical value is unstable during the recursion process due to the large amount of calculation. Reference [7] proposed a Strong Tracking Filter (STF) cubature Kalman filter algorithm. The introduction of a fading factor realized the online adjustment of the gain matrix, thereby maintaining the high tracking and positioning ability when the system dimension suddenly changed, but also increased computational complexity. In this regard, covariance matrices of a series of adaptive high-order cubature Kalman filtering (HCKF), CDKF and other algorithms also introduce a fading factor to improve the strong tracking ability, but all have their own numerical calculation and filtering accuracy problems. On the other hand, with the growing of intelligent algorithms, in solving the problem of measurement noise uncertainty in navigation, reference [8] proposed a fuzzy iterative cubature Kalman filter algorithm (FISCKF), which uses fuzzy control to adjust model parameters to improve to a certain degree of accuracy. In view of the large calculation amount and numerical problems of the above algorithms, and under large initial error or large observation noise, the observability of the system is weak, leading to problems such as unstable filtering algorithms and slow convergence speed [9]. This paper follows the filter of square-root form. By recursively updating square-root form of covariance matrix, it can not only reduce the computational complexity, but also ensure the symmetry and semipositive definiteness of covariance matrix and improve accuracy and stability of the filtering . The learning and training of neural networks can be regarded as seeking the best weight parameter through optimal estimation [10-11]. The weight matrix and network output values can be regarded as state quantities and measured values, respectively, and described by state space training. Since neural networks do not need accurate mathematical models, and have good convergence and robustness, so a SRCKF-SLAM with neural network framework algorithm is proposed. The weights of neural network are associated with network input, function type and ideal network output respectively, and neural network framework is performed on the weights of SRCKF to obtain a cubature point set that can better characterize the state of the target. In order to more clearly analyze the superiority of SRCKF neural network framework for non-linear filtering estimation problem of target tracking, this paper carried out comparative experiments of SRCKF, SRUCF, and SRCDKF, and carried out corresponding experimental analysis.

2. Materials and methods

2.1 Proposed SRCKF-SLAM algorithm

In the cubature transformation, the numerical integration of a set of cubature points with equal weight is used to approximate the Gaussian integral of the above formula, that is,

$$I = \int_{U_n} f(y)p(y;\mu,\Sigma)dy \approx \frac{1}{2n_y}\sum f\left(\sqrt{\Sigma}\xi_i + \mu\right) \tag{1}$$

Where, y is n_y dimension vector; The non-linear function is $f(\cdot)$; p is the probability density under a Gaussian distribution. cubature points that are orthogonal to each other at the coordinates $\xi_i = \sqrt{n}\{[1]_i\}, i = 1,2,\cdots,2n$ in the i dimensional coordinate system. When $n = 2$, the cubature point base in a 2-dimensional coordinate system is $\sqrt{2}\left\{\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}-1\\0\end{bmatrix},\begin{bmatrix}0\\-1\end{bmatrix}\right\}$.

The SLAM problem is decomattituded into robot and feature map state quantities x_v and x_m.

$$x_m = \left[\mu_{k,1}, \mu_{k,2}, \cdots, \mu_{k,m}\right] \quad (2)$$

Feature map state quantity, including the specific state of each feature in the map, where $\mu_{k,i}, i = 1,2,\cdots,m$ represents the state of the the i-th feature point at k moment and is added to the map database.

Synthesis of two parts of state quantities, namely:

$$s_k = \left[x_{k,v}, x_{k,m}\right]^T = \left[x_{k,v}, \mu_{k,1}, \mu_{k,2}, \cdots, \mu_{k,i}, \cdots, \mu_{k,m}\right]^T, i = 1,2,\cdots,m \quad (3)$$

Where, $\mu_{k,i}$ representing state value in the i-th feature point at k moment.

Then describe the SLAM problem as a Bayesian model as

$$p\left(s_k|z^{1:k}, u^{1:k}\right) \sim N(s_k; \hat{s}_k, P_k) \quad (4)$$

Where, $z^{1:k}$ is observed by the sensor from 1 to k; $u^{1:k}$ is robot motion control information from time 1 to k. Then the SLAM filtering problem is the process of solving the optimal estimate $\hat{s}_k$ of the system state quantity s_k and its covariance P_k and then updating it. Assuming that robot state model and the map feature model obey the Gaussian normal distribution, the SLAM Bayesian model of the above formula is expressed as a prior estimation form:

$$p\left(s_k|z^{1:k-1}, u^{1:k}\right) \sim N\left(s_k; \hat{s}_{k|k-1}, P_{k|k-1}\right) \quad (5)$$

From Markov chain hypothesis, the formula (5) is simplified to the following form:

$$p\left(s_k|z^{1:k-1}, u^{1:k}\right) = \int p\left(\hat{x}_k|\hat{x}_{k-1}, u^{1:k}\right) \cdot p\left(s_k|z^{1:k-1}, u^{:k-1}\right) ds_{k-1} \quad (6)$$

The prior estimation model is expressed as the system joint posterior probability density form:

$$p\left(x_{v,k}, x_{m,k}|z^k, u^k\right) = p\left(z_k|x_{v,k}, x_{m,k}\right) \int p\left(x_{v,k}|x_{v,k-1}, u^k\right) \cdot p\left(x_{v,k}, x_{m,k}|u^k, z_{k-1}\right) dx_{v,k-1} \quad (7)$$

Where, u_k is control input, drive $x_{v,k-1}$ to $x_{v,k}$; z_k is characteristic measurement of robot at position $x_{v,k}$ at time k. $p\left(x_{v,k}|x_{v,k-1}, u_k\right)$ is motion model which is the probability of downloading robot posture $x_{v,k}$ at time k, on the premise that control input u_k at time k and robot posture $x_{v,k-1}$ at time $k-1$ are known; $p\left(z_k|x_{v,k}, x_{m,k}\right)$ is observation model which is the conditional probability of obtaining observation z_k under premise of $x_{v,k}$ and $x_{m,k}$.

For nonlinear motion models:

$$\begin{cases} x_{v,k} = f\left(x_{v,k-1}, u_k + \delta u\right) \\ z_k = h\left(x_{v,k}\right) + \delta z \end{cases} \quad (8)$$

Where, $f(\cdot)$ and $h(\cdot)$ are system nonlinear function and observation function, respectively. System noise and observation noise are represented here by $\delta u \sim N(0, Q)$ and $\delta z \sim N(0, R)$. u_k represents robot motion control information; let system dimension be n_s; the feature dimension be n_m; then system joint state quantity is:

$$s_k = \left[x_{k,v}, x_{k,m}\right]^T = \left[x_{k,v}, \mu_{k,1}, \mu_{k,2}, \cdots, \mu_{k,i}, \cdots, \mu_{k,m}\right]^T, i = 1,2,\cdots,m \quad (9)$$

Then robot state vector at two neighboring moments can be expressed as the attitude information of robot itself and extended state vector of system's input control information, that is, $\chi_k = [s_k u_k]^T$ decomposing the covariance matrix into $P_k = \zeta_k \zeta_k^T$, then the extended square-root covariance matrix is $S_k = diag\left[\zeta_k \sqrt{Q}\right]$. When the algorithm is updated, the object of action is square-root covariance subformula, so here it is square-root covariance subformula ζ_k and its extended subformula S_k.

SRCKF algorithm includes the following steps:

(1) Time update phase

① Calculate the cubature point set from state system dimension $(n_s + n_m)$ to $2(n_s + n_m)$ dimension of the joint system:

$$\hat{x}_{k-1}^{i} = S_{k-1}\xi_i + \chi_{k-1}, i = 1,2,\cdots,2(n_s + n_m) \tag{10}$$

Where, i-th joint cubature point state quantity includes i-th robot attitude information cubature point and i-th map characteristic cubature point, that is,

$$\hat{x}_{k-1}^{i} = \left[x_{k-1,v}^{i}, x_{k-1,u}^{i}, x_{k-1,m}^{i}\right]^T = \left[x_{k-1,v}^{i}, x_{k-1,u}^{i}, \mu_{k-1,1}^{i}, \mu_{k-1,2}^{i}, \cdots, \mu_{k-1,j}^{i}, \cdots, \mu_{k-1,m}^{i}\right]^T, j = 1,2,\cdots,m \tag{11}$$

Where, $x_{k-1,v}^{i}$ and $x_{k-1,m}^{i}$ are robot and characteristic position information, respectively; $x_{k-1,u}^{i}$ is control input amount to robot.

② Cubature points are propagated through equation of state, and the prior estimates (weights) are calculated as:

$$\hat{x}_{k|k-1}^{*i} = f\left(x_{k-1,v}^{i}, x_{k,u}^{i}\right) \tag{12}$$

③ According to cubature seeking rule, square-root covariance of robot attitude estimation can be calculated:

$$S_{k|k-1} = qr\left(S_{k|k-1}^{*} S_{Q,k-1}\right), Q_{k-1} = S_{Q,k-1} S_{Q,k-1}^{T} \tag{13}$$

④ Apply cubature transformation to approximate the third-order state of robot.

$$\chi_{k|k-1,v}^{i} = \frac{1}{2(n_s+n_m)} \sum_{i=1}^{2(n_s+n_m)} \hat{x}_{k|k-1,v}^{*,i} \tag{14}$$

Where, qr means the matrix is decomposed into lower triangle and upper triangle forms; and covariance prediction value is:

$$P_{k|k-1} = \frac{1}{2(n_s+n_m)} \sum_{i=1}^{2(n_s+n_m)} \hat{x}_{k|k-1}^{*,i} \left(\hat{x}_{k|k-1}^{*,i}\right)^T - \hat{x}_{k|k-1} \hat{x}_{k|k-1}^{T} + Q_k \tag{15}$$

⑤ Estimation (weight) and state error innovation are represented by $S_{k|k-1}^{*}$.

$$S_{k|k-1}^{*} = \frac{1}{\sqrt{2(n_s+n_m)}} \left[\hat{x}_{k|k-1}^{*,1} - \chi_{k|k-1}^{1}, \hat{x}_{k|k-1}^{*,2} - \chi_{k|k-1}^{2}, \cdots, \hat{x}_{k|k-1}^{*,2(n_s+n_u)} - \chi_{k|k-1}^{2(n_s+n_u)}\right] \tag{16}$$

(2) Measurement update phase

The i-th feature point observation generalized from the nonlinear observation model is expressed as the posterior estimation formula:

$$z_{k|k-1}^{i} = h\left(x_{k|k-1,v}, x_{k|k-1,m}^{i}\right) + \delta z = h\left(x_{k|k-1,v}, \mu_{k|k-1}^{i}\right) + \delta z \tag{17}$$

Where, $x_{k|k-1,v}$ is attitude information of robot in the prior estimation, which does not change with observation of a certain feature point; $\mu_{k|k-1}^{i}$ is estimated value of i-th feature point at time $k-1$; Observation noise is $\delta z \sim N(0, R)$.

① First calculate the i-th cubature point.

$$\hat{x}_{k-1}^{i} = \zeta_{k|k-1}\xi_i + x_{k|k-1}, i = 1,2,\cdots,2(n_s + n_u) \tag{18}$$

$$\hat{x}_{k-1}^{i} = \left[x_{k|k-1,v}^{i}, v_{k|k-1,1}^{i}, v_{k|k-1,2}^{i}, \cdots, v_{k|k-1,m}^{i}\right] \tag{19}$$

is cubature point set of i-th cubature point $x_{k|k-1,v}^{i}$ of robot's own state and i-th cubature point $v_{k|k-1,j}^{i}$ of j-th characteristic state quantity, $j = 1,2,\cdots,m$.

② Propagation cubature point

$$z_{k|k-1}^{i,j} = h\left(x_{k|k-1}^{i}, v_{k|k-1,j}^{i}\right) \tag{20}$$

③ According to the rule for finding the cubature, calculate the observation and prediction values:

$$\hat{z}_{k|k-1}^{i} = \frac{1}{2n_s}\sum_{j=1}^{2n_s} z_{k|k-1}^{i,j} \tag{21}$$

④ Calculate observation and forecast estimate (weight) covariance error innovation square-root formula:

$$S_{zz,k|k-1} = qr\left\{\left[\varepsilon_{k|k-1}^{i}\sqrt{R}\right]^{T}\right\} \tag{22}$$

Where, $R_k = S_{R,k}S_{R,k}^{T}$. Innovation error is:

$$\varepsilon_{k|k-1}^{i} = \frac{1}{\sqrt{2n_s}}\left[z_{k|k-1}^{i} - \hat{z}_{k|k-1}^{i}\right],\ i = 1,2,\cdots,2n_s \tag{23}$$

$$\varepsilon_{k|k-1}^{i} = \frac{1}{\sqrt{m}}\left[z_{k|k-1}^{i} - \hat{z}_{k|k-1}^{i}\right],\ i = 1,2,\cdots,m \tag{24}$$

⑤ Square-root of estimated (weight) cross-covariance is:

$$P_{xz,k|k-1} = \zeta_{k|k-1}\left(\varepsilon_{k|k-1}^{i}\right)^{T} \tag{25}$$

$$\zeta_{k|k-1} = \frac{1}{\sqrt{2n_s}}\left[\hat{x}_{k|k-1}^{*,1} - \hat{x}_{k|k-1,v}, \hat{x}_{k|k-1}^{*,2} - \hat{x}_{k|k-1,v}, \cdots, \hat{x}_{k|k-1}^{*,2n_s} - \hat{x}_{k|k-1,v}\right]^{T} \tag{26}$$

Kalman gain is:

$$W_k = \left(P_{xz,k|k-1}/S_{zz,k|k-1}^{T}\right)S_{zz,k|k-1} \tag{27}$$

According to Kalman gain expression in CKF algorithm, square-root of correlation error covariance can be obtained:

$$S_{k|k} = qr\left[\zeta_{k|k-1} - W_k\varepsilon_{k|k-1}, W_kS_{R,k}\right] \tag{28}$$

Under the given initial conditions, after the state prediction update iteration of SRCKF-SLAM algorithm, system's state estimation is obtained, and robot's navigation in the environment and positioning and observation of target are achieved. However, with the creation of SLAM map database, the cubature point set and its weights have been continuously updated and expanded, which has led to a reduction in the algorithm's filtering estimation accuracy. A neural network framework based on SRCKF-SLAM algorithm is proposed to reduce square-root cubature points. The weight matrix of set and the output value of network are regarded as state quantities and measured values, respectively, and state space is used to describe training.

2.2 Neural network structure model

Let the number of nodes in input layer of neural network be n, the number of nodes in hidden layer be m, and output layer be a non-time-varying neuron node. Then network model is shown in the Fig. 1 [12]:

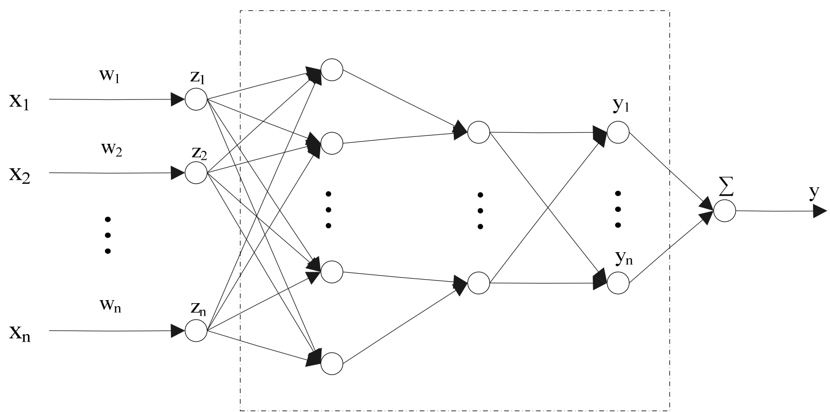

Fig. 1 The model of neural network

Let dimension of input signal be n, so the number of samples is the same as the dimension of input signal. In Fig. 1, $x_1, x_2, \cdots, x_n$ is input signal to be trained, and its corresponding weight is: $w_1, w_2, \cdots, w_n$. The function at each input node of hidden layer is $z_i = x_i w_i$.

Input signal of output layer is:

$$z = \sum_{i=1}^{n} x_i w_i = x^T w \tag{29}$$

The general formula for nonlinear relationship between output of neural network and input is:

$$y = f\left(\sum_{j=1}^{m} v_j \psi\left(\frac{\int_0^T \sum_{i=1}^{n} x_i(t) w_{ij}(t) dt - b_j}{a_j}\right)\right) \tag{30}$$

Where, x_i and y are network input and output. $\psi(\cdot)$ is hidden layer excitation function. $a_j$is the scaling function factor. b_j is translation function factor. $f(\cdot)$ is activation function of output layer. $w_{ij}(t)$ is the weight between j-th neuron in hidden layer and the i-th neuron in input layer, and v_j is the weight between output layer and the j-th neuron in hidden layer.

It is generally believed that increasing the number of hidden layers can reduce network errors, but also complicate network. Generally, the design of neural networks should give priority to 3-layer networks. When accuracy allows, reduce the number of nodes in hidden layer as much as possible [13-15]. State space of network consists of a weight matrix, network input and output, weight update functions, and parameterized non-linear functions. Assuming that there are I and J neurons in each of the two adjacent network layers, the weight matrix between the two layers is W_{IJ} and the weight space of neural network is $W = \{\sum W_{IJ}\}$.

Let W_{IJ} be the weight matrix between layer I and layer J consisting of $w_{ij}(i = 1...I; j = 1...J)$. Unitized weight matrix is W_{IJ}:

$$W_{IJ} = F_1(w_{ij}) = \frac{w_{IJ}}{\sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij}^2}} \tag{31}$$

The optimal estimate of the cubature point weights (state quantities) of SRCKF is obtained through neural network framework. State space model of neural network with weight as state quantity can be expressed as:

$$\begin{cases} w_{k+1} = F_k w_k + q_k \\ z_k = \psi_k(w_k, \tilde{x}_k) + r_k \end{cases} \tag{32}$$

Where, two kinds of Gaussian noise obey $q_k \sim N(0, \tilde{Q})$ and $r_k \sim N(0, \tilde{R})$, respectively. Output of neural network at time $k + 1$ is the iterative training result of the observation equation of network, which is determined by network weights and network observation noise at time k; F_k and ψ_k represent the linear state mapping and observation nonlinear mapping of network. Article selection is $F_k = I_k$.

3.Results and discussion

3.1 Neural network framework based on proposed SRCKF-SLAM algorithm

Establish a neural network framework and tracking algorithm based on SRCKF-SLAM. The process includes:

(1) Update iterative formula according to the state of SRCKF-SLAM, generate and propagate cubature points, solve the prior estimates of the attitude of robot and calculate robot's attitude information and square-root covariance.

(2) Set the initial weight for the innovation error (weight) between the observation and estimation of the cubature point state with the new square-root sub-form in SRCKF in step (1) as the input of neural network. Hermit function is selected as excitation function, a network state space

model is established to train the SRCKF state quantity, and the filtered new optimal estimate and gain are obtained as output of neural network, which together serve as the next state parameter update.

The result of the expected output vector is that the position error of the feature point is required to be minimal, so network output layer node is set to 3 (Observe coordinates in the x-direction and y-direction, and observation angle of robot heading angle φ), so network input layer vector is constrained, and the number of nodes is set to 5 (Coordinates in the x-direction, coordinates in the y-direction, the size of robot's heading angle φ, and controls input noise and observation noise), hidden layer is first set to 10 nodes and is automatically adjusted by Matlab software after network training programming.

(3) Bring the trained estimates (weights) into the SRCKF algorithm to iterate, propagate cubature points and calculate state update parameters.

Square-root of the estimated cross-covariance in step (1) is improved to a neural network framework state model as:

$$\tilde{P}_{xz,k|k-1} = \sum_{i=1}^{m} w_i \zeta_{k|k-1} \left(\varepsilon_{k|k-1}^{i}\right)^T \tag{33}$$

Kalman gain is:

$$\widetilde{W}_k = \left(\tilde{P}_{xz,k|k-1} / S_{zz,k|k-1}^T\right) S_{zz,k|k-1} \tag{34}$$

According to Kalman gain expression in CKF algorithm, square-root of correlation error covariance can be obtained:

$$S_{k|k} = qr\left[\zeta_{k|k-1} - \widetilde{W} \varepsilon_{k|k-1}, \widetilde{W} S_{k|k}\right] \tag{35}$$

3.2 Verification and analysis of feature map simulation experiments based on SLAM

Let initial state of robot be zero. Initial values of robot and feature covariance are $P_{v0} = \mathrm{diag}[330]^T$ and $P_{m0} = \mathrm{diag}[0.20.20]^T$, so sensor sampling frequency is $T = 0.02s$. System noise and observation noise are $Q = \begin{bmatrix} 0.2^2 & 0 \\ \left(\frac{2\pi}{180}\right)^2 & \left(\frac{3\pi}{180}\right)^2 \end{bmatrix}$, $R = \begin{bmatrix} 0.1^2 0 \\ 0 \left(\frac{\pi}{180}\right)^2 \end{bmatrix}$. Robot motion parameters are as follows: speed $v = 2$ m/s, speed error $\sigma_v = 0.2$ m/s, angular velocity $\alpha = 15\ °/s$, ranging error 0.25 m, angle error $\sigma_\alpha = 1°/s$. In the experimental environment (Fig. 2), SRCKF-SLAM, SRUKF-SLAM, SCDKF-SLAM were subjected to 50 independent repeated simulation experiments, so the estimation errors in the x-direction and y-direction of robot path were analyzed and compared.

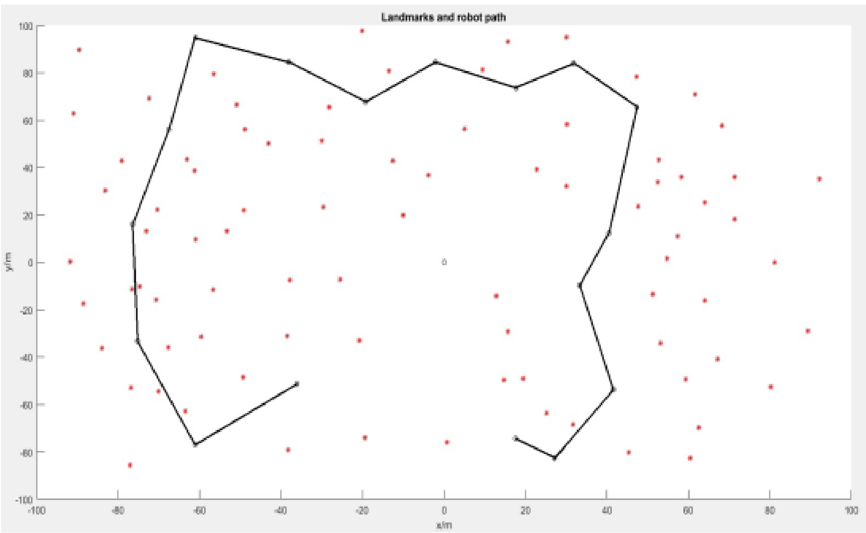

Fig. 2 Simulation area

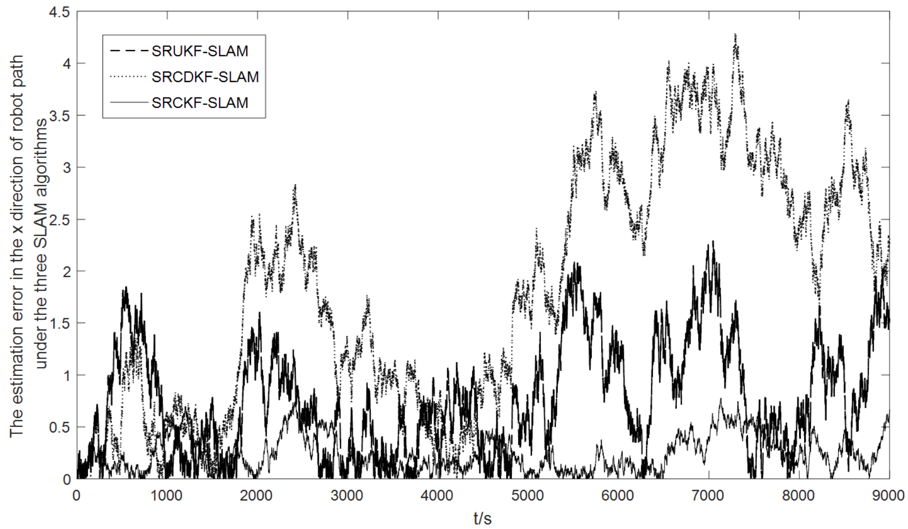

Fig. 3 The estimation error in the x-direction of robot path under the three SLAM algorithms

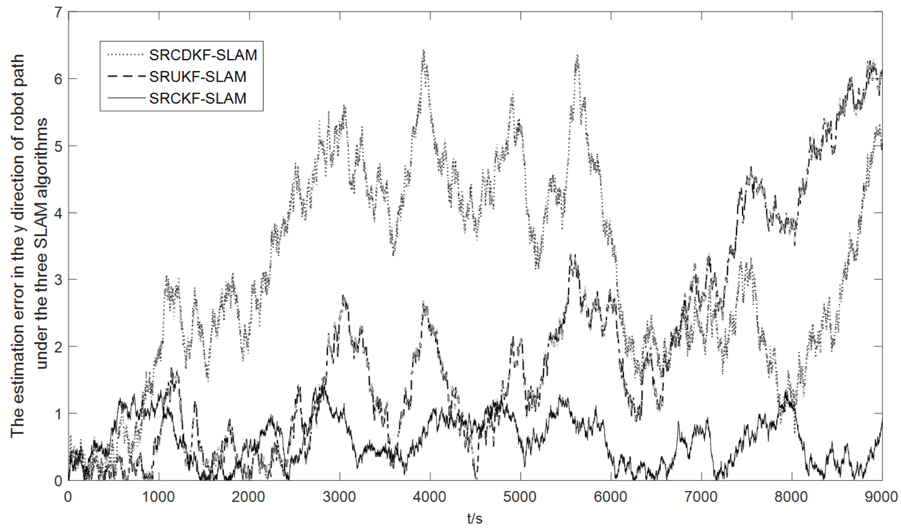

Fig. 4 The estimation error in the y-direction of robot path under the three SLAM algorithms

It can be seen from Figs. 3 and 4 that the position estimation error in the x-direction of robot path obtained using SRCKF-SLAM algorithm is [0, 0.8] m, and position estimation error in the y-direction is [0, 1.2] m. Compared with SRUCF-SLAM and SRCDKF-SLAM filter estimation errors, the filter estimation error is the smallest, and the filter estimation accuracy is superior to them. The experimental data fully proves that using SRCKF-SLAM algorithm for mobile robot navigation and positioning estimation can achieve high accuracy and good stability. Then compare the superiority of the target feature position of cubature point estimation (weight) under neural network framework based on the new and traditional SRCKF-SLAM algorithm.

3.3 Experimental verification and analysis of neural network frameworks based on SRCKF-SLAM

- *Simulation experiment and analysis of feature position estimation*

Hermit function is selected as excitation function, and the expression is:

$$f(x) = \frac{1.1(1-x+2x^2)\exp\left(-\frac{x^2}{2}\right)}{2} \tag{36}$$

Initialization algorithm can effectively reduce overshoot and shorten the convergence time of algorithm. Network itself has a certain good weight, so this difference can effectively reduce the possibility of divergence during network learning process. In this paper, initial network weight w is a random number between $[-1,1]$, and the noise follows normal distribution[0,0.1], the learning rate is 0.4, the momentum term coefficient is 0.6, and the number of learning times is 800. According to the function expression, P0 is 0.55, a = 0.2, b = 0.01.

In order to better compare the filtering estimation accuracy of SRCKF-SLAM filtering estimation algorithm of the cubature point weights and the traditional SRCKF-SLAM algorithm after neural network, increase the system noise and observation noise, set

$$Q = \begin{bmatrix} 0.2^2 & 0 \\ \left(\frac{2\pi}{180}\right)^2 & \left(\frac{3\pi}{180}\right)^2 \end{bmatrix}, R = \begin{bmatrix} 0.1^2 \left(\frac{\pi}{180}\right)^2 \\ 0 \quad \left(\frac{\pi}{180}\right)^2 \end{bmatrix}.$$

Other initial conditions remain unchanged, and 50 independent repeated simulation experiments are performed, and the estimated errors in the x-direction and y-direction of the target feature position are analyzed and compared.

From Figs. 5 and 6, it can be seen that under the conditions of increasing system noise and observation noise, as the simulation progresses, due to the setting of the reference path, the walking direction of robot is constantly changing, so the position estimation error has fluctuations during the simulation. The traditional SRCKF-SLAM algorithm's position estimation errors in the x-direction and y-direction are respectively [0, 1.8] m and [0, 2.6] m. The filtering algorithm based on SRCKF-SLAM's cubature point estimation (weight) after neural network framework, the position estimation errors in the feature x-direction and y-direction are respectively [0, 1.2] m, [0, 1.7] m. The filtering estimation accuracy is higher than the former, which is consistent with the theoretical analysis.

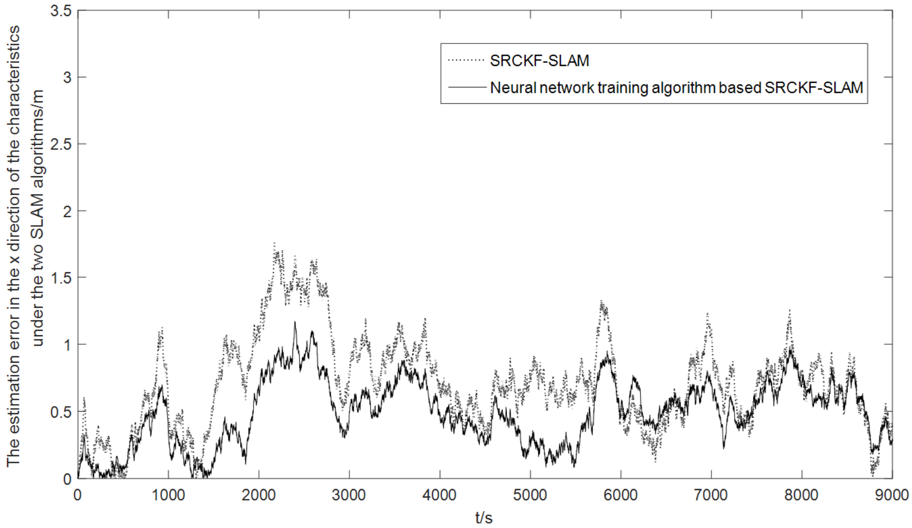

Fig. 5 The estimation error in the x-direction of the characteristics under the two SLAM algorithms

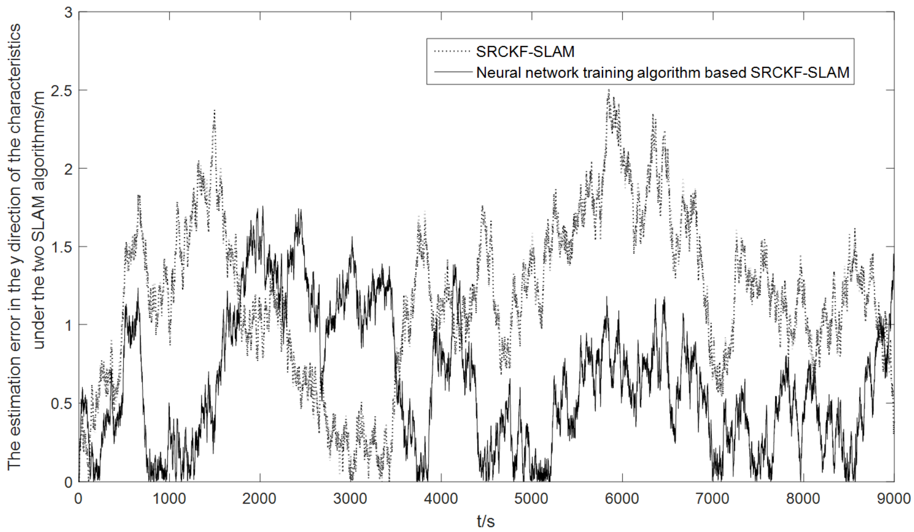

Fig. 6 The estimation error in the y-direction of the characteristics under the two SLAM algorithms

- *Observation of features under different SLAM algorithms*

Based on SLAM filtering algorithm, the position error observations in the x-direction and y-direction of features are collected.

Observe the characteristics in Fig. 2 of the simulation area. The three filtering algorithms using the SRUCF-SLAM, SRCKF-SLAM and SRCKF-SLAM cubature point weight training algorithms all observe 82 feature points, indicating that the new algorithm is feasible for SLAM feature target observation. It can be seen from Figs. 7 and 8 that the above-mentioned three algorithms reduce the position error in the x-direction and the y-direction of the features in turn, indicating that the SRCKF-SLAM algorithm after the cubature point weights are trained by network has the highest accuracy. And the algorithm error is convergent and stable, which is consistent with the theoretical analysis.

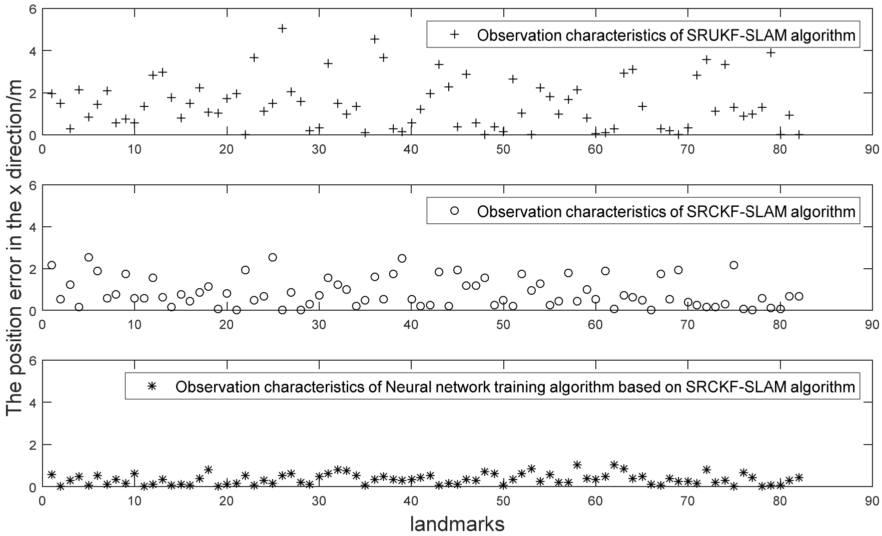

Fig. 7 The position error in the x-direction of landmarks

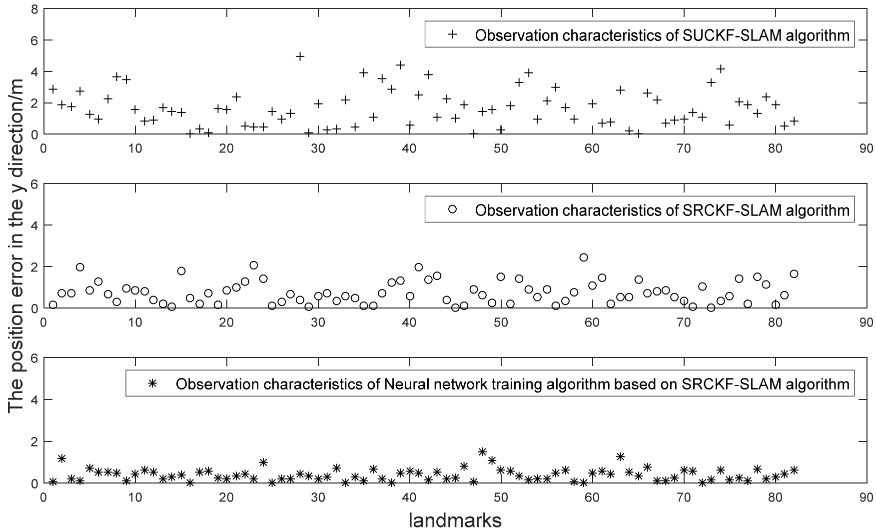

Fig. 8 The position error in the y-direction of landmarks

3.4 Possible practical implementation of the proposed SRCKF-SLAM algorithm

At present, single non-linear filtering algorithms used to deal with SLAM problems mainly includes EKF, UKF, CDKF, CKF, etc. Later, scholars further improved above several algorithms and proposed their derivative algorithms such as SRCDKF, SRUKF and SRCKF, then their accuracy are mainly discussed below.

In reference [16], accuracy of EKF, UKF and CDKF are compared, and it is concluded that when EKF algorithm was used to deal with nonlinear model, Jacobian matrix was needed to be solved, and Taylor expansion and high-order terms were needed to be eliminated, and larger truncation error was introduced, which led to larger algorithm error. UKF is a method that uses weighted statistical linear regression to achieve random linearization, which can approximate the Gaussian distribution. In theory, UT transformation could approach the posterior mean and covariance of the nonlinear Gaussian system with three times of Taylor precision. However, when UKF algorithm generated a set of random Sigma points with a large number, it would lead to more complicated calculation of state quantity parameters.

The Central Difference Kalman filter (CDKF) belongs to a suboptimal Gaussian filter. This algorithm uses polynomial interpolation to calculate multidimensional integrals. It uses a function sequence to approximate the integrand, which is similar to UT transform in UKF. It is just that the sampling point weights and calculation of prediction covariance expressions are different from the UKF algorithm in the form. The calculation of this algorithm is simple and easy to implement, but it is different from the selection of sigma points of UKF. In theory, the accuracy of CDKF and UKF algorithms is equivalent or slightly higher. According to references [17-18], it can be known that when using the CDKF algorithm to process non-linear models, the solution of the complex Jacobian matrix is avoided, the shortcomings of EKF algorithm are overcome, the linearization error is smaller than of EKF. The experiments have shown that it is less sensitive to state covariance and has a faster approximation speed than UKF.

Later scholars collectively referred to the UKF / CDKF as the sigma point Kalman filter (SPKF).

Then we discuss the accuracy problems of UKF, CDKF and CKF algorithms: the basic idea of the three filtering algorithms is to generate several groups of weight points through different methods, then calculate the parameters such as the mean value and covariance of propagation, and update state and measurement of algorithm. A large number of references, such as [19-20], have proved that the accuracy of filter estimation of the three algorithms is equivalent, and can reach the second order. For similar algorithms, another performance indicator that needs to be considered is computational complexity. According to the literature [20], it is verified that the computational complexity of the three algorithms is reduced in order, so that their filtering performance is sequentially enhanced.

However, for the SPKF filter and ordinary CKF filter, there is still a problem of computational divergence. Later, students introduced SR ideas into these methods. According to reference [9], Square-root can not only propagate square-root of state covariance, but also ensure the symmetry and semi positive setting of covariance matrix, and improve the accuracy, robustness and stability of filtering algorithm. So combining with SR idea, this paper applied three groups of filtering algorithms SRUKF, SRCDKF and SRCKF to SLAM nonlinear model, compared the filtering performance, and got the conclusion in this section. That is, SRCKF-SLAM algorithm had the highest filtering accuracy and the best stability. However, in order to get higher accuracy, this paper established a more suitable state model for engineering application after training the state quantity of SRCKF algorithm with neural network, and compared it with SRUKF and SRCKF algorithms without neural network framework, and obtained another group of experimental conclusions in this section.

It can be seen from the introduction and the theory of filtering in this paper, square-root cubature Kalman filter algorithm uses numerical integration to calculate the mean and covariance of the nonlinear random model, which avoids the derivative operation and reduces the computational complexity. Moreover, the algorithm propagates square-root of the state covariance, ensures the symmetry and semi positive finalization of the covariance matrix, and improves the accuracy, robustness and stability of the filtering. If the new algorithm is applied to the mobile robot, for example, in the practical application system of the aviation strategic missile in military field, when the tracked target encounters air resistance in the course of navigation, its state equation and observation equation become highly nonlinear. While in the estimation of the tracking state of the reentry ballistic target with unknown ballistic coefficient, the new algorithm can greatly reduce the tracked target state estimation error and improve the estimation accuracy. Therefore, the time to solve the its position and speed information is short, and the navigation operation speed is fast.