An Improved Algorithm of UKF-SLAM Based on RBF Neural Network Adaptive Robot

X Y Lee¹, G W Gao*¹

¹ Key Laboratory of Sensors, Beijing Information Science & Technology University
Beijing, China
*Corresponding author’s e-mail: xyleejoey@foxmail.com

Abstract—For Extend Kalman Filtering (EKF) algorithm in the robot simultaneous localization and mapping (SLAM), the linearization of the nonlinear system leads to the error of the system state equation, the large amount of computation caused by the Jacob matrix calculation, and the noise model uncertainty. Filtering instability and other issues. A modified UKF-SLAM algorithm based on RBF network adaptive is proposed. In the absence of prior noise information, the algorithm identifies the process noise and observation noise through RBF network adaptive identification unit, and iteratively corrects the noise covariance and the mean filter new covariance, so that the real-time positioning accuracy of the robot is improved. The experimental results show that this method has higher positioning precision and adaptive ability compared with the EKF algorithm and UKF algorithm.

1. INTRODUCTION
The simultaneous localization and mapping (SLAM) is the key technology to solve the real autonomous navigation of the mobile robot. It is mainly used to solve the problem of the position of the robot in an unknown environment. The mobile robot is equipped with a specific sensor. In the absence of environment prior information, the environment model is established in the moving process and its own motion trajectory is considered.

The most classical algorithm is to Extend Kalman Filtering (EKF) [2], which is to transform the nonlinear system into a linear system by carrying out the Taylor expansion of the model function of the robot and ignoring the high-order terms. In the actual process, the higher order term of the system is sometimes not negligible, the system will have a large error, and the filter is difficult to stabilize, resulting in the filter being unable to converge, that is, it is difficult to improve the accuracy of the environmental model [3-5]. At the same time, it is difficult to obtain the prior estimation characteristics of observation noise and system noise by using EKF algorithm, and it is difficult to improve the estimation accuracy of the filter when the prior data is not accurate enough or the amount of data is insufficient. Julier [6] proposed an untraceable Kalman filter SLAM algorithm (UKF-SLAM), which is based on untraceable transformation (UT) and realizes random linearization by using weighted statistical linear regression process. The error caused by the high-order term of the Taylor expansion in the linearization process of the EKF-SLAM algorithm is avoided, and the error and the divergence probability of the linearization of the SLAM problem are effectively reduced.

However, in the practical application, the UKF algorithm has the problem that the stability and precision of the filter are not accurately caused by the noise model, and on the other hand, the on-line
adjustment cannot be made because the filter gain does not have the on-line self-adaptive capacity. In view of the above disadvantages, the literature [7] proposes an improved algorithm for adjusting the UKF parameter, and the paper [8] proposes an algorithm to combine the UKF with the particle filter. In this paper, an adaptive UKF-SLAM algorithm based on RBF network is proposed to identify noise online, and at the same time, real-time positioning and map construction is carried out. In the case where the noise covariance is unknown, a larger noise covariance value is preset, the theoretical covariance and the actual covariance are input into the self-adaptive controller of RBF network, so that the noise covariance is continuously corrected, and the positioning error is reduced.

2. PROBLEM ELABORATION

2.1 Mathematical Model of SLAM problem

The basic meaning of SLAM is to carry out robot positioning and environment modeling at the same time. The robot corrects its position state by using the established environment model, so as to improve its positioning accuracy; At the same time, the robot relies on a reliable position state to establish a more accurate environment model. The SLAM problem may be represented by the formula (1):

$$p(x_i, m | Z_i, u_i)$$

(1)

Where, $x$ is the pose of the k-time robot, $m$ is the map, $Z(k)$ and $u(k)$ are the measured value and the control value.

According to the Bays theorem, we can see that:

$$p(x_i | Z_i, u_i) = \eta p(Z_i | x_i),$$
$$\int p(x_i | u_i, x(k-l)) p(x(k-l))dx(k-l)$$

(2)

Where, $\eta$ is the normalization constant; $p(Z_i | x_i)$ is the observation model of the robot; $p(x(k-l) | Z(k-l), u(k-l))$ is the motion model of the robot. The position and the map model of the robot are obtained from the probability of the observed model and the probability distribution of the motion model.

Assumed that a certain non-linear system model:

$$x_{i+1} = f(x_i, \mu_i, \omega_i)$$
$$z_{i+1} = h(x_{i+1}, u_{i+1})$$

(3)

Where, $x_i$ is the state vector of the system, $z_i$ is the observation vector of the system, $\mu_i$ is the control vector of input, $f(.)$ is the state transition function of the system, $h(.)$ is the observation function of the system, $\omega_i$ is the state noise of the system, $v_i$ is the observation noise of the system, $\omega_i$ and $v_i$ are zero-mean Gaussian white noise and are independent of each other, as equation (4).

$$\omega_i \sim N(0, Q_i)$$
$$v_i \sim N(0, R_i)$$

(4)

2.2 EKF-SLAM algorithm

The forecast and observation updates in EKF- SLAM is as follows:

1) Prediction process: acquiring weighted average value and covariance matrix of system state:

$$\hat{x}_{i+1} = f(\hat{x}_i, \mu_i)$$
$$P_{x_{i+1}} = FP_{x_{i}}F^T + W_iQ_iW_i^T$$

(5)

2) Observation process:

a) Observation measurement: according to the system model, at the $k+1$ time, the environmental characteristic measurement aggregate $Z_{k+1}$ (including azimuth and distance) is obtained by the sensors:

$$z_{k+1} = \{z_i | i \in M\}$$

(6)

Where, $M$ is the total number of environmental characteristics, $z_i$ is the value of the observation of the i-th feature in the environment.
b) Observation prediction: the map feature is extended to the system state vector, and the state quantity of the dismantled robot is predicted according to the state, including the prediction of the map feature position.

c) Data correlation: the observed eigenvalues are obtained from the observed data at the k+1 time and correlated with the estimated observations at k time. When the prediction is consistent with the observation, the system updates the state, and vice versa updates the map.

3) Innovations process:

\[
K_{k+1} = P_{k+1|k} H^T_{k+1} (H_{k+1} P_{k+1|k} H^T_{k+1} + R_{k+1})^{-1}
\]

\[
\hat{x}_{k+1} = \hat{x}_{k+1|k} + K_{k+1} [z_{k+1} - h(\hat{x}_{k+1|k}, 0)]
\]

\[
P_{k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}
\]

The Jacob matrix of the state transition function and observation function of the system with regard to the state vector is represented by \(F\) and \(H_{k+1}\).

2.3 UKF-SLAM algorithm

In order to find the mean and variance of the state vector in UKF-SLAM prediction, set \(x_t\) to \(n\)-dimension, sampling \((2n+1)\) sigma points, the sampling rule for sigma points is [9]:

\[
Z_{r,k-lk-1} = \begin{cases} 
\tilde{Z}_{k-1, i} & i = 0 \\
\tilde{Z}_{k-1, i} - \sqrt{(n+k)P_{n-k-1}} & i = 1, 2, \ldots, n \\
\tilde{Z}_{k-1, i} + \sqrt{(n+k)P_{n-k-1}} & i = n+1, \ldots, 2n 
\end{cases}
\]

Where, \((\sqrt{(n+k)P_{n-k-1}})\) is the i-sequence of square root matrices of \((n+k)P_{n-k-1}\), available by Cholesky decomposition, generally \(n+k=3\) for Gaussian distribution. The sigma point is brought into a non-linear motion model, and a sigma point set \(Z_{r,k-lk-1}\) after the transformation can be obtained, therefore, the mean and variance of the prediction are:

\[
Z_{r,k-lk-1} = \sum_{i=0}^{2n} w_{i} Z_{r,k-lk-1}
\]

\[
P_{r,k-lk-1} = \sum_{i=0}^{2n} w_{i} [Z_{r,k-lk-1} - \tilde{X}_{k-lk-1}] [Z_{r,k-lk-1} - \tilde{X}_{k-lk-1}]^T
\]

During the observation update process, to calculate a new sigma point set by referring to the sigma point sampling rule of (10), \(\tilde{X}_{k-lk-1}\) is mean value, \(P_{r,k-lk-1}\) is covariance matrix. Bring it into the observation model, can get \(Z_{r,k-lk-1} = h(\tilde{X}_{r,k-lk-1})\), the corresponding mean and variance are:

\[
Z_{r,k-lk-1} = \sum_{i=0}^{2n} w_{i} Z_{r,k-lk-1}
\]

\[
S_{k} = \sum_{i=0}^{2n} w_{i} [Z_{r,k-lk-1} - \tilde{Z}_{k-lk-1}] [Z_{r,k-lk-1} - \tilde{Z}_{k-lk-1}]^T + R_{k}
\]

\(V_{k}\) is innovation, that is \(V_{k} = Z_{k} - \tilde{Z}_{k-lk-1}\), the mean and variance of \(k\) time can be calculated as follows:

\[
\tilde{X}_{k} = \tilde{X}_{k-lk-1} + W_{k} V_{k}
\]

\[
P_{k-lk-1} = P_{k-lk-1} - W_{k} V_{k} W_{k}^T
\]

Where, \(W_{k} = P_{r,k-lk-1} S_{k}^{-1}\) is Kalman gain, \(P_{r,k-lk-1}\) is cross covariance matrix.

Finally, update the state and covariance matrix of the map.
2.4 State Enhancement and Map Update

If a new set of feature points is generated, the new feature needs to be extended to the state vector to update the global map. Let the $i$-th new feature be $Z_i = (r_i, \Phi_i)$, and the feature position form converted to the system coordinate system is:

$$
\begin{bmatrix}
  z_i \\
y_i
\end{bmatrix} = \begin{bmatrix}
x + r_i \cos(\theta + \Phi_i) \\
y + r_i \sin(\theta + \Phi_i)
\end{bmatrix}
$$

Extending $(x_i, y_i)$ to the state vector of the system and find its covariance matrix:

$$
P_{k+1} = \begin{bmatrix}
P_{v,v} & P_{v,p} & P_{v,pi} \\
P_{v,p} & P_{p,p} & P_{p,pi} \\
P_{v,pi} & P_{p,pi} & P_{pi,pi}
\end{bmatrix}
$$

Calculation of covariance matrix $P_{v,p}$ between new feature points based on error propagation law, the covariance matrix $P_{v,v}$ between posture and feature, the covariance matrix $P_{p,v}$ between features and posture, and the covariance matrix $P_{pi,p}$ and $P_{p,pi}$ between features and features.

2.5 RBF Network PID Control

The traditional PID controller formula is as follows:

$$
u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

The discreteness is as follows:

$$
u(k) = K_p e(k) + K_i \sum_{j=1}^k e(j) + K_d [e(k) - e(k-1)]$$

Where, $e(k)$ is the input deviation value of the $k$-sampling, $u(k)$ is the output value of the $k$-sampling.

RBF neural network is a forward network that can approximate the nonlinear continuous function with arbitrary precision. Its input-output mapping is non-linear, only the hidden layer to the output layer is linear, it avoids the local minimal problem of the BP neural network, and accelerated the speed of learning [10]. The structure of the RBF neural network is divided into three layers, namely an input layer, a hidden layer, and an output layer. The structure diagram of the RBF neural network is shown in Figure 1.

![RBF Neural Network Structure](image)

The output of the RBF neural network is:

$$
Y_i = \sum_{i=0}^n w_{ia} f(c_i - x_i)
$$

Where, $w_{ia}$ is node weight, $Y_i$ is the output of the $i$-th node, $n_i$ is the number of implied-layer neurons, $f(.)$ is the radial basis function.

The radial basis function is [10]:
\[ R_i = \exp\left(-\frac{||x - c_i||^2}{2\sigma_i^2}\right) \]  

(18)

Where, \( c_i \) is the network node center vector, \( \sigma_i \) is the base width parameter of node \( j \).

3. Improved Algorithm of Robot SLAM Based on Neural Network PID

3.1 Neural network PID adaptive online identification

The neural network PID controller and median filter constitute the on-line identification of process noise and observation noise of unknown system [10]. The method of on-line identification of noise is shown in figure 2.

In UKF-SLAM, the innovation \( I_{mk} \) at \( k \) time is the difference between the actual observation \( Z(k+1) \) and the observation prediction \( h(x_k, v_k) \), the innovation \( I_{nk} \) is the difference between actual observation \( x(k+1) \) and observation prediction \( f(x_k, u_k) \):

\[
I_{mk} = Z(k+1) - h(x_k, v_k) \\
I_{nk} = x(k+1) - f(x_k, u_k)
\]

(19)

![Figure 2. Principle Diagram of Neural Network PID Adaptive Online Identification](image)

Under steady state filtering, innovation has the characteristic of stationary ergodic, the covariance of \( I_{mk} \) and \( I_{nk} \) samples can be approximated by the average value of covariance \( \hat{C}_{ik} \) in \( n \) moving windows:

\[
\hat{C}_{ik} = \frac{1}{2n} \sum_{i=0}^{n-1} (I_{mk} f_{mk}^T + I_{nk} f_{nk}^T)
\]

(20)

The covariance correction value \( \Delta Q_k \) (system process noise covariance correction value) and \( \Delta R_k \) (observation noise covariance correction value) obtained by neural network PID controller, after noise correction, the asymptotic stability of the noise covariance \( Q_k \) and \( R_k \) can be obtained by median filtering.

3.2 Improved SLAM algorithm based on PID Adaptive Control

The input of PID adaptive controller is the theoretical covariance and the actual covariance of innovation. The SLAM system presupposes a large noise covariance and uses online adjustment to approximate sufficient noise information. The self-adaptive identification of the observed noise is realized by the reference [9], in this paper, the system observation noise and the process noise are identified jointly, and the covariance correction values of the system observation noise and the process noise are obtained. After correcting the noise covariance, the noise covariance is applied to the UKF-SLAM update step, and the iterative positioning results are obtained. Figure 3 shows the structure of UKF-SLAM improved algorithm proposed in this paper.

The improved UKF-SLAM algorithm based on PID adaptive control mainly includes the following 6 steps:

1) Prediction of pose and covariance matrix of the dismantled robot at \( k \) time according to (12).
2) The observation of environmental characteristic $i$ is obtained by the sensor, and the observed value $Z_{ik}$ is obtained.

3) According to the environmental characteristic point estimation of pose prediction and observation of the robot, the predicted observation value $\hat{Z}_{i+1}$ is calculated.

![Figure 3. Improved Algorithmic of SLAM Structure](image)

4) According to (19), the innovation $I_{mk}, I_{nk}$ is calculated and the innovation covariance matrix $R_k, Q_k$ is obtained by the neural network PID controller.

5) Calculating $\Delta Q_k$ and $\Delta R_k$ by neural network PID controller, and modified the noise covariance.

6) Finally, according to (13) ~ (14), the results of the robot at the $k+1$ time are obtained $\hat{X}_{k+1}$ and the corresponding $P_{k+1}$.

### 4. Simulation Results and Analysis

Simulation through Matlab, the EKF-SLAM algorithm, the UKF-SLAM algorithm and the improved algorithm proposed in this paper are tested. The outdoor environment of $250m \times 200m$ is simulated, including 17 artificial navigation points and 35 road signs. The experimental parameters and experimental environment are shown in Table 1 and Figure 4. The robot moves counterclockwise from the origin of the given map, and the experimental model is as follows:

$$
\begin{bmatrix}
x(k)
y(k)
\theta(k)
\end{bmatrix} =
\begin{bmatrix}
x(k - 1) + V \Delta t \cos(G + \theta(k - 1)) \\
y(k - 1) + V \Delta t \sin(G + \theta(k - 1)) \\
\theta(k - 1) + V \Delta t
\end{bmatrix}
$$

(21)

| Table 1: Experimental Parameters |
|----------------------------------|
| velocity                        | 3m/s |
| steering locking angle          | 30°  |
| Steering angle change rate      | 20°/s|
| System sampling interval        | 0.025s|
| Axle spacing                    | 4m   |
| velocity error                  | 0.3m/s|
| maximum detectable range        | 30m  |
| range error                     | 0.3m |
| angle error                     | 1°   |
Figure 4. Experimental environment

Figure 5. Trajectory diagram of EKF_SLAM algorithm

Figure 6. Trajectory diagram of UKF_SLAM algorithm

Figure 7. Trajectory diagram of improved UKF_SLAM algorithm
5. CONCLUSION
In this paper, an improved UKF-SLAM algorithm for demolition robot based on the RBF network is proposed. This algorithm uses the RBF network adaptive identification unit to identify the process noise and observation noise of the unknown system, and the iterative correction of noise covariance and average filtering innovation covariance is also given. The simulation results show that the improved adaptive identification SLAM algorithm realizes the synchronous positioning and map construction of the demolition robot under the condition of unknown noise. The experimental results show that this method has higher positioning accuracy and adaptive ability than the EKF algorithm and the UKF algorithm.

Acknowledgement
Projects supported by the National Natural Science Foundation of China (61571053)

REFERENCES
[1] M. Emharraf, M. Bourhaleb, M. Saber, M. Rahmoun, "Mobile Robot: SLAM Implementation for Unknown Indoor Environment Exploration"[J]. Journal of Computer Science, vol. 12, no. 2, pp. 106-112, 2016.
[2] Durrant Whyte H, Bailey T. Simultaneous localization and mapping: part I [J]. Robotics & Automation Magazine, IEEE, 2006, 13 (2): 99 ≤ 110
[3] Cadena C, Carlone L, Carrillo H. Past, present, and future of simultaneous localization and Mapping : Toward the Robust Perception Age[J]. IEEE Transactions on Robotics, 2016, 32(6): 1309
[4] Huletski A, Kartashov D, Krinkin K. TinySLAM improvements for indoor navigation[C]// IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. IEEE, 2017: 493-498
[5] Jiang X, Li T, Yu Y. A novel SLAM algorithm with Adaptive Kalman filter[C] // International Conference on Advanced Robotics and Mechatronics. IEEE, 2016: 107-111.
[6] Minh-Duc Tran, Hee-Jun Kang. A novel adaptive finite time tracking control for robotic manipulators using nonsingular terminal sliding mode and RBF neural
networks[J]. International Journal of Precision Engineering and Manufacturing, 2016, 17 (7): 864-869

[7] Shoichi Maeyama, Yuta Takahashi, Keigo Watanabe. A solution to SLAM problems by simultaneous estimation of kinematic parameters including sensor mounting offset with an augmented UKF[J]. Advanced Robotics, 2015, 29(17).

[8] Kim A, Eustice R M. Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency[J]. IEEE Transactions on Robotics, 2013, 29 (3): 719-733

[9] Astrom K J, Hang C C, Hagglund T, et al. Automatic tuning and adaptation for PID controllers-survey[J]. Control Engineering Practice, 1993, 28 (1): 699-714

[10] Zeng Jun, Fang Houhui. Neural Network PID Control and its Matlab Simulation[J]. Modern Electronics Technique, 2004, 27(2): 51-53