MOBILE ROBOT LOCALIZATION USING FUZZY NEURAL NETWORK BASED EXTENDED KALMAN FILTER

NGUYEN THI THANH VAN, PHUNG MANH DUONG, TRAN THUAN HOANG
TRAN QUANG VINH

University of Engineering and Technology, Vietnam National University, Hanoi 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam; Email: vanntt@vnu.edu.vn

Abstract. This paper proposes an approach to improve the performance of the extended Kalman filter (EKF) for the problem of mobile robot localization. A fuzzy logic system is employed to continuously adjust the noise covariance matrices of the filter. A neural network is implemented to regulate the membership functions of the antecedent and consequent parts of the fuzzy rules. The aim is to gain the accuracy and avoid the divergence of the EKF when the noise covariance matrices are fixed or incorrectly determined. Simulations and experiments are given. The results show that the proposed filter is better than the EKF in localizing the mobile robot.

Key words. Fuzzy logic, neural network, extended kalman filter, localization, mobile robot.

1. INTRODUCTION

Localization, that is to determine the robot’s position and orientation from sensor data, is a fundamental problem in mobile robotics. In order to complete given tasks, the robot need to know its own pose at each sampling time to make the path planning and motion control. The difficulty is that there always exist two kinds of error in the robot system: the systematic and non systematic errors [1]. The systematic error is caused by the imperfectness of robot mechanics such as the limitation of encoder resolution, the ansymmetry of robot chassis, and the inequality and misalignment of driven wheels. The non systematic error is often random and unknown. It is caused by uncertainty elements such as the slippage of wheels or the unevenness of floor during the robot operation. Filtering these errors to extract the actual
pose of the robot is the final goal of localization algorithms. Various approaches have been proposed with their strengths and weaknesses.

In [2], one linear and two rotary encoders are used to measure the relative distance and bearing between two wheels. The result is then injected to the conventional dead reckoning method to improve the accuracy. Another approach is the Monte Carlo method, which is able to localize the mobile robot without knowledge of its starting location. This method is more accurate, faster, and less memory intensive than grid based methods [3]. A survey of Bayesian filter applied to real world estimation was done in [4], where the authors show that the Bayesian filter technique is a powerful statistical tool to perform the multi-sensor fusion, estimate the system state and manage the measurement uncertainties. Its drawback is the expensive computation.

A less computation but effective method is the extended Kalman filter (EKF) [5]. It estimates the robot position under the scenario that both sensor and system data are subjected to zero-mean Gaussian white noises [5, 6]. In this method, the choice of noise covariance matrices greatly affects the estimate accuracy. Due to random essence of errors, these matrices change according to the time of operation and therefore are difficult to determine. In practice, these matrices are often assumed to be fixed and chosen through off-line processes. However, this simplification may cause the EKF to diverge in some cases. In this paper, a fuzzy system is implemented to online adjust the noise covariance matrices at each Kalman step. The membership functions of the fuzzy system are regulated by a neural network. The incorporation of fuzzy logic and neural network into the EKF creates a new optimal filter called the fuzzy neural network based EKF (FNN-EKF). This filter enhances the accuracy and convergence of the EKF for the problem of localization in unknown indoor environment. This combines with previous results in path planning and obstacle avoidance of the author’s group constitutes a quite complete set of mechanisms for the autonomous operation of a mobile robot [7–9].

2. MOBILE ROBOT LOCALIZATION USING EXTENDED KALMAN FILTER

As mentioned in previous section, the EKF is considered as one of the most effective method for mobile robot localization. This section first presents the kinematic model of the robot. The implementation of the EKF is then described and discussed.

A. Robot model

The mobile robot considered in this paper is two-wheeled, differential-drive mobile robot with non-slipping and pure rolling. Figure 1 shows the coordinate system of the robot, where \((X_G, Y_G)\) is the global coordinate system and \((X_R, Y_R)\) is the local coordinate system relative to the robot chassis. With this type of mobile robot, the dead reckoning is often used to determine the relative position of the robot in the work space. It estimates the robot’s position and orientation based on encoder data.

From Figure 1, the conversion factor that translates encoder pulses into linear wheel displacement is given by
MOBILE ROBOT LOCALIZATION USING FUZZY NEURAL NETWORK

Figure 1. Pose and parameters of mobile robot

\[ C_m = \frac{\pi R}{n C_e} \]  \hspace{1cm} (1)

where \( R \) is the wheel diameter, \( n \) is the gear ratio of the reduction gear between the motor and the drive wheel, and \( C_e \) is the encoder resolution (in pulses per revolution).

Let \( N_{L,i} \) and \( N_{R,i} \) be the numbers of pulses counted by encoders of the left and right wheels at time \( i \), respectively. The displacement of each wheel is then given by

\[ \Delta S_{L,i} = C_m N_{L,i}; \quad \Delta S_{R,i} = C_m N_{R,i}. \]  \hspace{1cm} (2)

These can be translated to the linear incremental displacement of the robot’s center \( \Delta S \) and orientation angle \( \Delta \theta \) as

\[ \Delta S_i = \frac{\Delta S_{L,i} + \Delta S_{R,i}}{2}; \quad \Delta \theta_i = \frac{\Delta S_{R,i} + \Delta S_{L,i}}{L} \]  \hspace{1cm} (3)

where \( L \) is the distance between two wheels. The coordinates of the robot in the global coordinate frame then can be given by

\[ x_i = x_{i-1} + \Delta S_i \cos(\theta_{i-1} + \Delta \theta_{i-1}), \]
\[ y_i = y_{i-1} + \Delta S_i \sin(\theta_{i-1} + \Delta \theta_{i-1}), \]  \hspace{1cm} (4)
\[ \theta_i = \theta_{i-1} + \Delta \theta_i. \]

From equation (4), it is able to determine the robot’s position and orientation if we know the number of encoder pulses in each sampling period. In another word, equation (4) is the localization equation of the dead reckoning method. Nevertheless, equation (4) does not include errors appeared in the system. As analyzed in Section 1, the errors are unavoidable and may downgrade the system performance if appropriate compensation is not investigated. For this reason, the robot model is rewritten in state-space representation with the appearance of disturbances as

\[ x_i = f(x_{i-1}, u_{i-1}, w_{i-1}), \]  \hspace{1cm} (5)
where \( \mathbf{z}_i = h(\mathbf{x}_i, \mathbf{v}_i) \) is the state vector described the instantaneous position and orientation of the robot, \( \mathbf{u} = [\Delta S_{L,i} \Delta S_{R,i}]^T \) is the input, \( \mathbf{z}_i \) is the measurements, \( f \) and \( h \) are the system functions, \( \mathbf{w}_i \) and \( \mathbf{v}_i \) are random variables described the process and measurement noises. These noises are assumed to be zero-mean, independent, white, and Gaussian with the process noise covariance matrix \( \mathbf{Q} \) and the measurement noise covariance matrix \( \mathbf{R} : \mathbf{w}_i \sim \mathcal{N}(0, \mathbf{Q}_i), \mathbf{v}_i \sim \mathcal{N}(0, \mathbf{R}_i) \). The state-space presentation (5)-(6) is basis for the implementation of the EKF.

B. Implementation of the EKF for mobile robot localization

As \( \mathbf{x} \) is the state vector described the robot position and orientation, the problem of localization becomes the problem of state estimation. An optimal solution to this problem is the Kalman filter [12]. Its implementation is performed through two steps: prediction and correction as follows:

1. Prediction step with time update equations:

\[
\hat{\mathbf{x}}^-_i = f(\hat{\mathbf{x}}_{i-1}, \mathbf{u}_{i-1}, 0),
\]

\[
P^-_i = A_i P_{i-1} A_i^T + W_i Q_{i-1} W_i^T
\]

where \( \hat{\mathbf{x}}^-_i \) is the prior state estimate at step i given knowledge of the process prior to step i-1, \( P^-_i \) is the covariance matrix of the state prediction error, \( A_i \) is the Jacobian matrix of partial derivatives \( f \) to \( \mathbf{x} \), \( W_i \), the Jacobian matrix of partial derivatives \( f \) to \( \mathbf{w} \), \( Q_i \) the input noise covariance matrix.

2. Correction step with measurement update equations:

\[
K_i = P^-_i H_i^T (H_i P^-_i H_i^T + V_i R_i V_i^T)^{-1},
\]

\[
\hat{\mathbf{x}}_i = \hat{\mathbf{x}}^-_i + K_i (\mathbf{z}_i - h(\hat{\mathbf{x}}^-_i, 0)),
\]

\[
P_i = (I - K_i H_i) P^-_i,
\]

where \( \hat{\mathbf{x}}_i \) is the posterior state estimate at step i given measurement \( \mathbf{z}_i \), \( K_i \) is the Kalman gain, \( R_i \) is the covariance matrix of measurement noise, \( H_i \) is the Jacobian matrix of partial derivatives \( h \) to \( \mathbf{x} \), \( V_i \) is the Jacobian matrix of partial derivatives \( h \) to \( \mathbf{v} \).

Equations (7) - (11) show that the noise covariance matrices \( \mathbf{Q} \) and \( \mathbf{R} \) need to be accurately determined to ensure the efficient operation of the EFK. In addition, these matrices should be changed at each step of the EKF. The dynamic determination of \( \mathbf{Q} \) and \( \mathbf{R} \) however is often not easy to perform, especially during the operating session of the mobile robot. In practice, \( \mathbf{Q} \) and \( \mathbf{R} \) are often assumed to be fixed. Their values are determined before the execution of the EKF. This approach reveals several drawbacks. Firstly, fixed matrices \( \mathbf{Q} \) and \( \mathbf{R} \) do not reflect the variation of noises. Secondly, the covariance error \( \mathbf{P} \) and the Kalman gain \( K \) will quickly
reach the steady state. This is equivalent to the convergence of the EKF to certain degree of accuracy. Finally, the fixation of $Q$ and $R$ may break the stability operation of the EKF and cause the system to be halted in some cases. To overcome these, fuzzy logic \cite{13, 14} can be employed to enable an online adjustment (not the determination) of $Q$ and $R$. In addition, the efficiency of fuzzy rules can be further improved by using neural network to regulate its membership functions.

3. IMPROVEMENT EKF BY FUZZY NEURAL NETWORK SYSTEM

This section first presents the basis idea to adjust the covariance matrices $Q$ and $R$. Details of implementation this idea using fuzzy logic and neural network are then described.

A. Concept of noise covariance matrices adjustment

From equation (10), let $r_i = z_i - h(\hat{x}_{i-1}, 0)$ be the residual between the actual and the predicted measurements. This residual, gained by $K$, is the correction factor to form the posterior estimate $\hat{x}_i$ from the prior estimate $\hat{x}_{i-1}$. It also reflects the accuracy of the estimation value. A small value of $r_i$ implies good estimation as the predicted measurement is closed to the actual measurement. As mentioned in Section 2.B, if the covariance matrices $Q$ and $R$ are fixed, the Kalman gain $K$ will quickly stabilize and remain constant. It means that the EKF will converge to certain accuracy degree. In order to improve this, we can adjust $Q$ and $R$ so that $r_i$ is reduced.

Assume that the system operates with a predetermined and fixed $Q$, the adjusting process of $R$ to reduce $r_i$ is performed as follows:

- Determining the residual $r_i$ and computing its present covariance:

$$S_i = H_i P_i H_i^T + R_i.$$  \hspace{1cm} (12)

- The average covariance of $r_i$ through a number of steps from the past to present is given by

$$\hat{C}_i = \frac{1}{N} \sum_{j=j_0}^{i} r_j r_j^T$$ \hspace{1cm} (13)

where $j_0 = i - N + 1$ and $N$ is the number of past periods.

- The difference between the present and average covariances of $r_i$ is given by

$$D_i = S_i - \hat{C}_i.$$ \hspace{1cm} (14)

This difference is considered as the reference to adjust $R_i$.

- Let $\Delta R_i$ be an adjustment factor, $\Delta R_i$ is changed through each step as follows:

1. If $D_i \geq 0$ then maintain $\Delta R_i$
2. If $D_i > 0$ then decrease $\Delta R_i$
3. If $D_i < 0$ then increase $\Delta R_i$

- $R$ is then adjusted by
\[ R_i = R_i + \Delta R_i. \] (15)

The adjustment of covariance matrix \( Q \) is similar to the steps for \( R \) except that the present covariance matrix of \( r_i \) is replaced by

\[ S_i = H_i(A_i P_i A_i^T + Q_i)H_i^T + R_i. \] (16)

B. Fuzzy system implementation

The fuzzy system executed by equation (15) is implemented through three steps as follows.

\textbf{Step 1}: Define input/output language variables.

The input/output language variables of the fuzzy system are chosen as follows:
- \( D_i(j,j) \): Negative (N), Zero (Z), Positive (P);
- \( \Delta R_i \): Decrease (D), Maintain (M), Increase (I).

\textbf{Step 2}: Define membership functions.

We define four sigmoid-shape membership functions: \( P(a_1, b_1), N(a_2, b_2), D(a_3, b_3); I(a_4, b_4) \); two gauss-shape membership functions \( M(c_1, \sigma_1) \) and \( Z(c_2, \sigma_2) \) as follows

\[
sigmoid(x) = \frac{1}{1 + e^{-a_i(x-b_i)}}; \quad gauss(x) = e^{-\frac{(x-c_i)^2}{2\sigma_i^2}}
\] (17)

where \( a_i, b_i, c_i, \) and \( \sigma_i \) are the distinct values for each membership function.

\textbf{Step 3}: Design fuzzy rules.

Rules for the fuzzy system are defined so that the difference between the present and average covariance of the residual is reduced as

1. If \( D_i \) is Z then \( \Delta R_i \) is M.
2. If \( D_i \) is P then \( \Delta R_i \) is D.
3. If \( D_i \) is N then \( \Delta R_i \) is I.

\textbf{Step 4}: Defuzzification.

The defuzzification is accomplished by centroid method

\[
\phi = \frac{\sum x_i \mu(x_i)}{\sum \mu(x_i)}
\] (18)

where \( x_i \) is the i’th domain value and \( \mu(x_i) \) is the truth membership value for that domain point.

Among steps, the definition of membership functions requires manual customization. It is therefore time consuming and often do not adapt to the change of system parameters. To overcome this limitation, a neural network is implemented to regulate membership functions.

C. The fuzzy neural network system

The objective of the neural network is to tune parameters \( a_i, b_i, c_i, \) and \( \sigma_i \) so that the membership functions are fitter with the system. We design a neural network with five layers.
as shown in the Figure 2. Details of each layer are given as

Layer 1: The output of the node is the degree of the variable with respect to fuzzy set N, Z and P.

Layer 2: The firing level of each rule is computed by

$$\alpha_1 = N(x), \quad \alpha_2 = Z(x), \quad \alpha_3 = P(x).$$ (19)

Layer 3: The normalization of the firing levels is indicated. The normalized firing level of the corresponding rule is

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \quad \beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}, \quad \beta_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}. \quad (20)$$

Layer 4: The product of the normalized firing level and the individual rule output of the corresponding rule is the output of each neuron. They are determined respectively as

$$\beta_1 z_1 = \beta_1 I^{-1}(\alpha_1), \quad \beta_2 z_2 = \beta_2 I^{-1}(\alpha_2), \quad \beta_3 z_3 = \beta_3 I^{-1}(\alpha_3). \quad (21)$$

Layer 5: The overall output system is the sum of all incoming signals

$$z = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3. \quad (22)$$

Given the crisp training set \{ (D_1, \Delta R_1), (D_2, \Delta R_2), ..., (D_K, \Delta R_K), \}, we define the measure of error for the k-th training pattern as

$$E_k = \frac{1}{2} (z_k - \Delta R_k)^2, \quad k = 1...K \quad (23)$$

where \(z_k\) is the computed output form of the fuzzy system corresponding to the input pattern \(D_k\) and the desired output \(\Delta R_k\). The hybrid neural network learns the shape parameters \(a_i, \ b_i, \ c_i, \ \text{and} \ \sigma_i\) of each membership function by using the steepest descent method. They are described as

$$a_i(t+1) = a_i(t) - \eta \frac{\partial E_k}{\partial a_i}; \quad b_i(t+1) = b_i(t) - \eta \frac{\partial E_k}{\partial b_i}; \quad c_i(t+1) = c_i(t) - \eta \frac{\partial E_k}{\partial c_i}; \quad \sigma_i(t+1) = \sigma_i(t) - \eta \frac{\partial E_k}{\partial \sigma_i}. \quad (24)$$
where $\eta > 0$ is the learning constant and $t$ is the number of the adjustments.

Parameters extracted from equation (21) are used to construct new membership functions or learned membership functions for the fuzzy inference system.

4. SIMULATIONS

A. Simulation setup

Simulations are conducted to evaluate the efficiency of the proposed algorithm. In each evaluation, a Monte Carlo simulation with 100 times is executed to compare the performance of the EKF and the FNN-EKF. Parameters for simulations are extracted from a real mobile robot built by the author’s research group at the university’s laboratory [15].

The mobile robot has following parameters: the wheel’s diameter $R$ is 0.05m, the distance between the wheels $L$ is 0.6m, the gear ratio of the reduction gear between the motor and the drive wheel $n$ equals to 1, and the encoder resolution $C_e$ is 500. The input noise covariance matrix $Q$ is defined as proportional to the deviations of the displacement, $\sigma_{\Delta S}$, and the orientation, $\sigma_{\Delta \theta}$, of the robot. The covariance of measurement noise is chosen based on the value of the real robot system described in next section. Formula (25) shows the values of $Q$ and $R$ as

$$
Q = \begin{bmatrix}
\sigma_{\Delta S}^2 & 0 \\
0 & \sigma_{\Delta \theta}
\end{bmatrix};
R = \begin{bmatrix}
0.01 & 0 & 0 \\
0 & 0.01 & 0 \\
0 & 0 & 0.018
\end{bmatrix}.
$$

Initial membership functions $P(a_1, b_1), N(a_2, b_2), D(a_3, b_3), I(a_4, b_4), M(c_1, \sigma_1)$ and $Z(c_2, \sigma_2)$ are chosen with parameters as follows:

$a_1 = -0.16, b_1 = -17, a_2 = 0.1, b_2 = 13, a_3 = 0, b_3 = -100, a_4 = 0.1, b_4 = 23, \sigma_1 = 0, \sigma_2 = 0.03, c_1 = 0.03, c_2 = 0.03$.

After learning process of neural network, these parameters are chosen as:

$a_1 = -0.1073, b_1 = -17.0995, a_2 = 0.1515, b_2 = 13.8108, a_3 = -0.06552, b_3 = -143, a_4 = 0.1030, b_4 = 24.2000, \sigma_1 = 0.1779, c_1 = 0.0571, \sigma_2 = 0.0620, c_2 = 0.0467$.

Figure 3 and Figure 4 show the initial and learned shapes of the membership functions $D, M$ and $I$ of $\Delta R_i$, respectively.

B. Simulation results

In the first comparison, the initial deviations of the displacement and the orientation of the robot are chosen to be small: $\sigma_{\Delta S} = 2\text{cm}, \sigma_{\Delta \theta} = 1^\circ$. The trajectory of the robot in this simulation is shown in Figure 5.

Figure 7 shows the comparative result of paths estimated by the EKF and FNN-EKF. We note that only 100 sample points are shown for the convenience of view and comparison. Figure 8-10 show the root mean square error (RMSE) of estimations in the $X, Y$, and $\theta$ direction. It indicates that the FNN-EKF estimation introduces smaller error than the EKF. In the second Monte Carlo simulation, the initial deviations of the displacement and the orientation of the robot are chosen to be 1: $\sigma_{\Delta S} = 5\text{cm}, \sigma_{\Delta \theta} = 3^\circ$. The trajectory of the robot in this simulation
is shown in Figure 6. Figure 11-14 show the results. They also indicate that the FNN-EKF estimation is better than than the EKF.

5. EXPERIMENTS

A. Experiment setup

Experiments were conducted in a real mobile robot developed by the author’s research group (Figure 15). The robot is a two wheeled, differential-drive mobile robot. Its wheel diameter is 0.05m and the distance between two drive wheels is 0.6m. The speed stability of motors controlled by the PID algorithm is ±5%. The sampling time $\Delta t$ is 375ms. The operation environment is flat-floor constructed by cement with several wooden plates surrounded (Figure
Details of the robot system can be referred from our previous work [15]. The initial value of matrices $Q$ and $R$ is similar to the second simulation case.

**B. Experiment results**

In experiments, the robot is controlled to follow a round path. Figure 17 shows the true path and the paths estimated by the EKF and the FNN-EKF. Some parts of trajectory are extracted for the convenience of view as shown in Figures 18-20. Due to the asynchronization between experimental values measured on the floor and the values estimated by the filters, it is not possible to compute the RMSE in experiments. Nevertheless, it is able to conclude from
6. CONCLUSIONS

This paper has proposed a filter called FNN-EKF for the problem of localization. This filter employs a fuzzy system and a neural network to regulate the noise covariance matrices so that the estimate is converged and more accurate. A number of simulations and experiments have been conducted and the results demonstrate the improvement of the FNN-EKF compared to the EKF. The good localization result in this research can be used as observation data for the controller to control the robot motion in various navigation tasks.
Figure 15
A real mobile robot at laboratory

Figure 16
A operation environment of robot

Figure 17. The trajectory of robot

Figure 18. The first corner on the trajectory

Figure 19. A line on the trajectory

Figure 20. The third corner on the trajectory

REFERENCES


Received on August 28, 2012

Revised on June 16, 2013