

Mobile Robot Local Path Planning Based on Improved T-S Fuzzy Neural Network

Suying Zhang, Yankai Shen, Zexu Sun

College of Electrical Engineering, Hebei University of Science and Technology,

Shi Jiazhuang, China

1164492565@qq.com

Keywords: T-S fuzzy neural network; local path planning; transformative Gaussian function; modified BP algorithm

Abstract. Aiming at the problems that BP algorithm has slow convergence rate and is likely to fall into local minimum point. This paper presents a modified fuzzy neural network which takes the distances between the robot and obstacles as input variables, and the velocity and the steering angle of the robot as output variables. Besides, the accurate environmental information can be obtained by fusing the information of ultrasonic sensors, and the network parameters can be adjusted by training the network. With the application of Matlab simulation software, the result shows that the improved fuzzy neural network has good performances in optimizing the convergence speed, learning speed and stability of the network.

Introduction

At present, the mobile robot path planning is one of the hot points in the field of intelligent robot research. The path planning is that how a robot searches out an optimal or sub-optimal path from the starting position to the target position according to some certain performance indexes [1]. Depending on how much the environmental information the robot acquires, the path planning strategies are divided into two kinds: one is global path planning, the other one is local path planning.

Up till now, some global path planning approaches, that the environment information has already known, have good performances. However, some means dealing with those situations where the environment is complex and time-varying are still insufficient. Especially the aspects of the local path planning and information acquisition, which make them become the future development direction of the path planning [2].

This article will combine the fuzzy logic algorithm with neural network making the network has the advantages of both being easy to express human knowledge and learning ability. What's more, an improved algorithm to solve the problem existing in traditional BP algorithm is proposed, and a optimal path is generated.

The environmental information acquisition

The distance information from the direction of left, right and front can be obtained by using ultrasonic sensors uniformly distributed around the robot (See Fig.1(a)). Assuming that the robot is viewed as a particle, the simplified diagram is shown in Fig.1(b). The angle between two adjacent sensors is 30° , and the effective measurement distance ranges from 2cm to 100 cm [3].

Presuming that $d_i (i = 1, 2, \dots, 6)$ is the i th sensor value, then the distance values fused are expressed as:

$$D_L = \min(d_4, d_5, d_6), D_F = \min(d_3, d_4), D_R = \min(d_1, d_2, d_3). \quad (1)$$

Where D_L means *left-distance*, D_F means *front-distance*, D_R means *right-distance*.

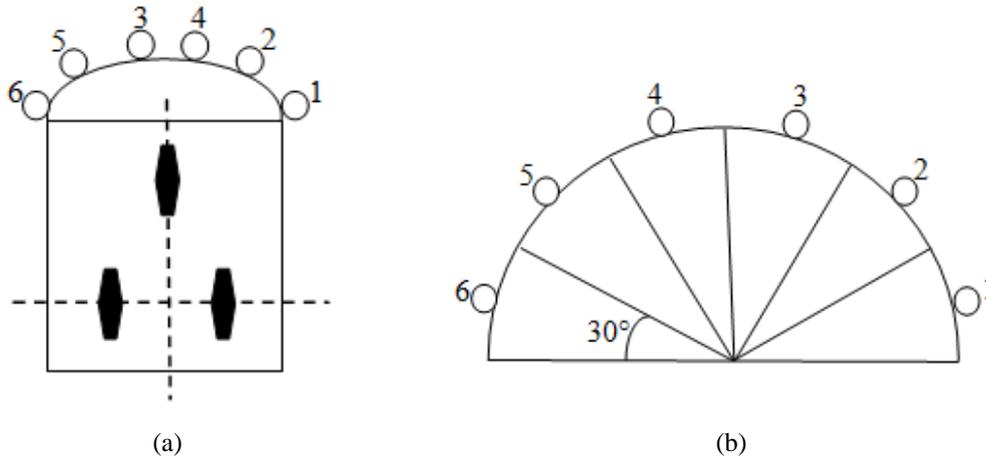


Fig.1 Ultrasonic Sensors Layout

Design of fuzzy-neural network

3.1 Fuzzification

The linguistic values of input variables are:

$$D = \{VN, RN, MD, RF, VF\}. \quad (2)$$

Where D includes D_L , D_F and D_R ; VN stands for *very-near*, RN for *relatively-near*, MD for *moderate-distance*, RF for *relatively-near*, VF for *very-far*.

The linguistic values of output variables are:

$$V = \{S, M, F\}. \quad (3)$$

$$\theta = \{TL, TLL, F, TRL, TR\}. \quad (4)$$

For the term set V , S stands for *slow*, M for *moderate*, F for *fast*; For the term set θ , TL stands for *turn-left*, TLL for *turn-left-little*, F for *forward*, TRL for *turn-right-little*, TR for *turn-right*.

3.2 Structure of fuzzy-neural network

Adaptive network-based fuzzy inference system is one of hybrid intelligent fuzzy-neural structure and it functions under T-S-type fuzzy interference system, which was designed by Jang in 1993 [4]. The inference system if-then rules based on experts' experience and knowledge are defined as:

Rule : IF x_1 is A_i and x_2 is B_i and x_3 is C_i ,

$$\text{THEN } f_j = p_j x_1 + q_j x_2 + r_j x_3 + u_j. \quad (5)$$

Where A_i , B_i and C_i are the fuzzy membership sets for the input variables $D_L(x_1)$, $D_R(x_2)$, $D_F(x_3)$ respectively. $i = 1, 2, \dots, 5$, $j = 1, 2, \dots, 125$; p_j , q_j , r_j and u_j are the linear parameters of function f_j and changing these parameters we can modify the output of the controller [5].

Assuming that Y_i is the output vector of the i th layer, y_{ij} is the output between the i th neuron from last layer and the j th neuron of the next layer. Then, the function of each layer in fuzzy-neural network structure (See Fig.2) is discussed as follows:

Input layer: In this layer, nodes receive signals from array of sensors which specify the position of the obstacles.

First layer: This layer is the adaptive fuzzy layer. Neurons in this layer complete the

fuzzification process. Every node in this stage is an adaptive node and calculating the membership function value in fuzzy set. For three inputs the outputs from nodes in this layer are presented as:

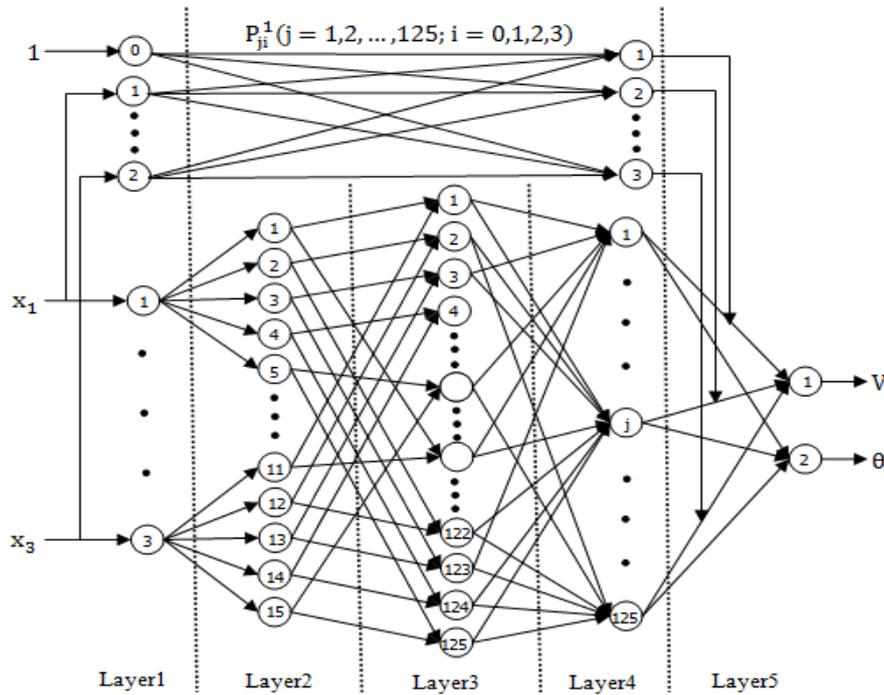


Fig.2 The Structure of Fuzzy-neural Network

$$Y_2 = [\mu_{ij}]_{15}^T \quad (6)$$

The input membership functions are the transformative Gaussian function and defined as follows:

$$\mu_{ij} = \exp(-b_{ij}(x_i - c_{ij})^2), b_{ij} > 0, b_{ij} = \frac{1}{2\sigma_{ij}^2} \quad (7)$$

Where c_{ij} , σ_{ij} are parameters that control the centre, width respectively. Changing these parameters will give the various contour of the function as required in accordance with the data set for the problem defined. Not only does this novel form shorten the training time, but also accelerate the running speed of the network significantly [6].

Second Layer: It is known as the rule layer. Every node in this stage corresponds to a single T-S fuzzy rule. Each rule point receives inputs from the respective points of layer-2 and calculates the firing strength of the each fuzzy rule. Output from each node is the product of all incoming signals.

$$Y_3 = [y_{3j}]_{125 \times 1} \quad (8)$$

$$y_{3j} = \mu_{i_1 1} \cdot \mu_{i_2 2} \cdot \mu_{i_3 3} \cdot \mu_{i_4 4} \cdot \mu_{i_5 5}; i_1, i_2, i_3, i_4, i_5 \in \{1, 2, 3, 4, 5\} \quad (9)$$

Third Layer: It is the normalization layer. Each point in this layer receives inputs from all points in the fuzzy rule layer and calculates the normalized firing strength of a given rule. The normalized firing strength of the j th point of the j th rules firing strength to sum of all rules firing strength.

$$Y_4 = [y_{4j}]_{125 \times 1} \quad (10)$$

$$y_{4j} = \mu_{i_1 1} \cdot \mu_{i_2 2} \cdot \mu_{i_3 3} \cdot \mu_{i_4 4} \cdot \mu_{i_5 5} / \sum_{j=1}^{125} y_{3j} \quad (11)$$

Output layer: Each node in this layer is connected to the corresponding normalization node and

also receives initial inputs x_1, x_2, x_3 . A defuzzification node determines the weighted consequent value of a given rule presented as:

$$Y_5 = Y * Y_4 = [V, \theta]^T. \quad (12)$$

$$Y = [y_{5j}]_{2 \times 125}. \quad (13)$$

$$y_{5j} = \sum_{i=0}^3 p_{ji} * x_i = p_{j0} * 1 + p_{j1} * x_1 + p_{j2} * x_2 + p_{j3} * x_3. \quad (14)$$

Modified learning method

The cost function and the parameter adjust algorithm are defined as:

$$E = \frac{1}{2} \sum_{k=1}^5 (O_k - Y_k)^2. \quad (15)$$

$$p_{ji}(k+1) = p_{ji}(k) - \eta \frac{\partial E}{\partial p_{ij}}(k). \quad (16)$$

$$c_{ij}(k+1) = c_{ij}(k) - \eta \frac{\partial E}{\partial c_{ij}}(k). \quad (17)$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta \frac{\partial E}{\partial \sigma_{ij}}(k). \quad (18)$$

Where O_k represents the desired output, Y_k is on the behalf of the actual output, and η ($0 < \eta < 1$) is on the behalf of the learning efficiency.

In order to solve the local minimum problem existing in the traditional BP algorithm, a correction method based on Romberg is proposed, taking the p_{ji} for example.

$$S^{k-2} = \frac{4P^{k-2} - P^{k-3}}{3}, S^{k-1} = \frac{4P^{k-1} - P^{k-2}}{3}, S^k = \frac{4P^k - P^{k-1}}{3}. \quad (19)$$

$$C^{k-1} = \frac{16S^{k-1} - S^{k-2}}{15}, C^k = \frac{16S^k - S^{k-1}}{15}. \quad (20)$$

$$P^k = \frac{64C^k - C^{k-1}}{63}. \quad (21)$$

Where $P^{k-3}, P^{k-2}, P^{k-1}, P^k$ represent the training vectors of every four times. ε is an arbitrarily small positive number. If $E > \varepsilon$, then make $E = 0$ and judge whether $P^{k-3}, P^{k-2}, P^{k-1}, P^k$ have saved? If so, use the Romberg method amend the vector values, and keep the correction value P^k as the next training data. Otherwise, stop training, and output the training results.

Simulation

This part mainly describes the simulation results of the improved T-S-type neural network. The initial membership functions is shown in Fig.4. After adjusting the parameters of the membership function, the centre and width positions of the membership functions become more fit to the actual control requirement (See Fig.5). From Fig.6, we can see, when the robot is in the situation where obstacles are concentrated, it has the possibility to hit obstacles. After using the modified learning

method to amend the training vectors, the robot can bypass all obstacles, and the generated path is much more smooth then before as it is shown in Fig.7.

Conclusion

This article puts forward two improved points based on T-S-type fuzzy neural network, namely: the membership function is the deformation form of Gaussian function and the learned parameters are corrected by the Romberg method. This solution not only improves the convergence speed and learning efficiency, but also effectively solves the local minima problem existing in BP algorithm, and the simulation results also verify the feasibility and effectiveness of the proposed solution.

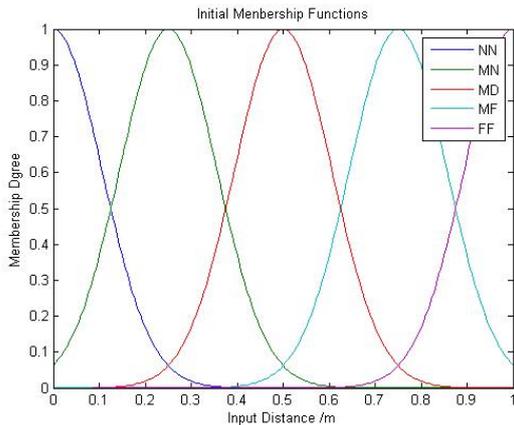


Fig.4 The Initial Membership Functions

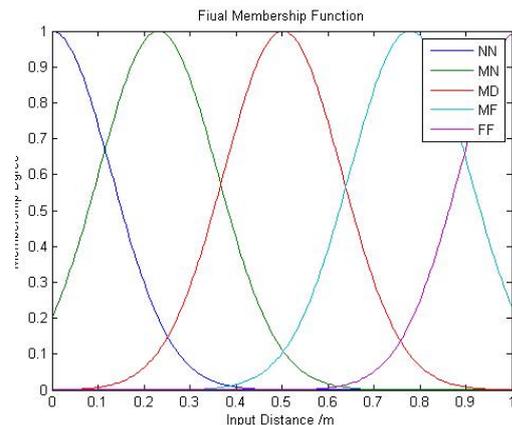


Fig.5 The Final Membership Functions

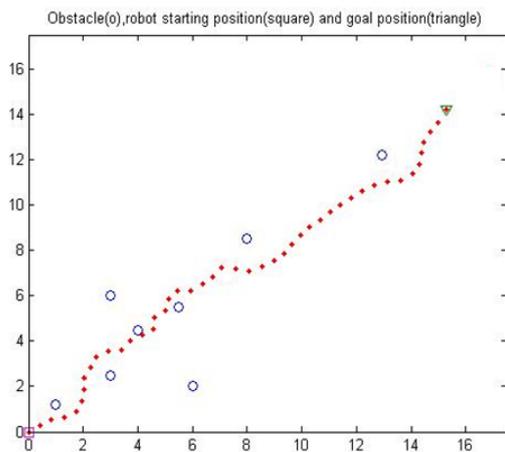


Fig.6 The Path Generated with The Traditional Method

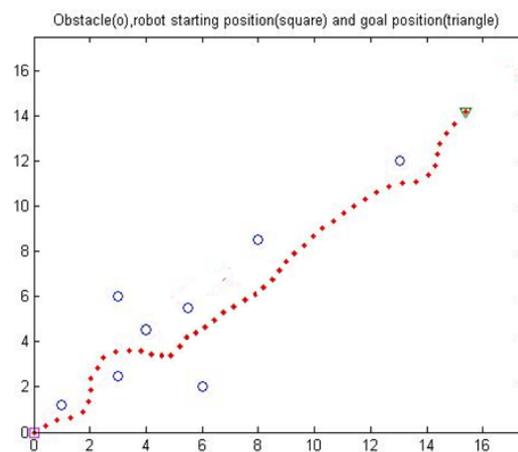


Fig.7 The Path Generated with The Modified Method

References

- [1] Liang Jin, Song Ke-pu: Journal of System Simulation Vol.269-272(2010).
- [2] Zhang Han-dong, Zhang Rui, Cen Yu-wan: Journal of System Simulation Vol.439-443(2005).
- [3] Xiong Kai-feng, Zhang Hua: Manufacturing Automation Vol.1-9(2013).
- [4] J.S.R. Jang: Adaptive network-based fuzzy inference system. IEEE Transaction on System, Man and Cybernetics -part B Vol. 665-685 (1993).
- [5] Prases K. Mohanty, Dayal R. Parhi: Applied Mathematics & Information Sciences Vol. 2527-2535(2014).
- [6] Wei Wu, Long Li, Jie Yang, Yan Liu: Information Sciences Vol. 1630-1642(2010).