



Research article

Research on obstacle avoidance algorithm for unmanned ground vehicle based on multi-sensor information fusion

Jiliang Lv^{1,2}, **Chenxi Qu**³, **Shaofeng Du**², **Xinyu Zhao**¹, **Peng Yin**^{1,2}, **Ning Zhao**² and **Shengguan Qu**^{1,2,*}

¹ School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China

² State Key Laboratory of Smart Manufacturing for Special Vehicles and Transmission System, Inner Mongolia First Machinery Group Co., Ltd., Baotou 014032, China

³ School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Manchester, M13 9PL, UK

* **Correspondence:** Email: qusg@scut.edu.cn; Tel: +86-131-6834-3353.

Abstract: With the wide application of unmanned ground vehicles (UGV) in a complex environment, the research on the obstacle avoidance system has gradually become an important research part in the field of the UGV system. Aiming at the complex working environment, a sensor detection system mounted on UGV is designed and the kinematic estimation model of UGV is studied. In order to meet the obstacle avoidance requirements of UGVs in a complex environment, a fuzzy neural network obstacle avoidance algorithm based on multi-sensor information fusion is designed in this paper. MATLAB is used to simulate the obstacle avoidance algorithm. By comparing and analyzing the simulation path of UGV's obstacle avoidance motion under the navigation control of fuzzy controller and fuzzy neural network algorithm, the superiority of the proposed fuzzy neural network algorithm was verified. Finally, the superiority and reliability of the obstacle avoidance algorithm are verified through the obstacle avoidance experiment on the UGV experimental platform.

Keywords: multi-sensor information fusion; unmanned ground vehicle; robotics control; obstacle avoidance; fuzzy neural network

1. Introduction

Unmanned ground vehicles (UGVs) have the characteristics of high movement stability and

carrying capacity, simple mechanical structure, fast movement speed, high movement flexibility and high work efficiency [1]. UGVs can perform tasks that difficult to be completed by human beings in a complex and harsh environment, which has attracted attention in more and more fields [2,3]. In the military field, UGVs can replace humans in high-temperature, high-radiation, or other high-risk environments to perform tasks such as reconnaissance, search and rescue, and explosion elimination [4]. In the civilian field, UGVs can replace security personnel to complete frequently repeated patrol tasks [5]; cleaning robots can enter the small gaps in the home environment to complete cleaning tasks [6].

The obstacle avoidance function is one of the important criteria for measuring the intelligence of UGVs [7], and it is also a research hot spot in the field of UGVs [8,9]. During the past few years, the studied of obstacle avoidance algorithm has produced outstanding results. For example, Jesus Savage et al. [10] combined genetic algorithm with periodic neural network to overcome the local minimum problem in the process of robot avoidance. Anish Pandey et al. [11] combined fuzzy reasoning, neural network algorithm and adaptive algorithm to independently design a reasoning system used in a static environment. The input of the reasoning system is the distance detected by the sensor system, and the output is the steering angle of the next movement of the robot, thus realizing the obstacle avoidance function during the autonomous movement of the robot. Kim and Chwa [12] proposed a fuzzy neural network(FNN) control algorithm using interval two fuzzy membership function, which increased the degree of freedom for uncertainty, and proved that the algorithm has a smaller linear velocity and angular velocity, making the obstacle avoidance path smoother. Shidrokh Goudarzi et al. [13] planned the shortest path based on the travel sales problem, aiming at the problem that UAV need smooth path. Bessel curve is used to transform the flyable path to realize the path smoothing. This method can efficiently and effectively collect data with high packet transmission rate and low energy consumption.

The working environment of UGVs is changeable and uncontrollable [14], which makes robots have stricter requirements on the accuracy of obtaining information from the surrounding environment [15]. The multi-sensor detection system contains a variety of different sensors, which can use a variety of different methods to detect the environment from multiple angles [16,17]. In recent years, multi-sensor information fusion technology has been widely studied and applied. For example, Sang Won Yoon et al. [18] used Kalman filter to fuse the information of speed sensor and positioning sensor in order to realize the coordinate calibration of UGV in the case of sliding. Risang Gatot Yudanto et al. [19] designed an extended Kalman filter information fusion algorithm based on velocity sensor and angle sensor, and realized the position tracking of robot with high dynamic motion. Alastise et al. [20] made use of extended Kalman filter to integrate the information of inertial measurement unit and visual sensor, to achieve accurate positioning of UGVs. Baofeng Ji et al. [21] proposed a two-hop cognitive network with the transmitter as a special radio frequency source to collect radio frequency energy from secondary user source nodes and secondary user relay nodes. Tao Tang et al. [22] discussed the important role of UAV in the future 5G Internet of Things, and proposed a UAV-PHD filter. They used KNN and K-means algorithms to improve GM-PHD algorithm, and applied it to target detection and tracking of UAV, and realized the trajectory tracking of multiple UAV targets.

The existing information fusion technology has poor processing capability for the information obtained by the sensor and long calculation time, which cannot meet the accuracy and real-time requirements of the UGV's obstacle avoidance function [23,24]. Therefore, the study of multi-sensor information fusion algorithms with high information fusion accuracy and strong real-time performance

is one of the important components to improve the obstacle avoidance level of UGVs, and has very important research value [25,26].

This paper analyzes and designs a FNN algorithm to avoid obstacles for the UGV. We designed the adaptive weighted fusion algorithm to process the information obtained by the sensors, and the result of the multi-sensor information fusion algorithm is the input of the FNN. The rest of the paper is organized as follows: Section 2 introduces the UGV and establishes the kinematics model. Section 3 analysis the multi-sensor information fusion algorithm and the FNN algorithm. Section 4 shows the simulation experiment and discusses the result. Section 5 shows the real experiment and discusses the result. Section 6 concludes the paper.

2. Unmanned ground vehicle experiment platform

The kinematics model of UGV is shown in Figure 1 [27], in the coordinate system (x_0y_0), let (x_0, y_0) represent the center of mass of the UGV, let θ_e represent azimuth angle of UGV, which is the angle between the forward direction of the robot and the x-axis, $\theta_e \in [-\pi, \pi)$; ω_e and v_e respectively represent the angular velocity and linear velocity of the UGV.

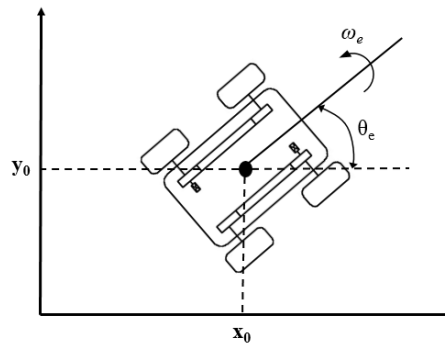


Figure 1. Kinematics model of UGV.

The matrix representation of the kinematics model of UGV is presented in Equation (1) [28].

$$\begin{bmatrix} x_e(k+1) \\ y_e(k+1) \\ \theta_e(k+1) \end{bmatrix} = \begin{bmatrix} x_e(k) \\ y_e(k) \\ \theta_e(k) \end{bmatrix} + \begin{bmatrix} \cos\theta_e & 0 \\ \sin\theta_e & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_e \\ \omega_e \end{bmatrix} \Delta t, \quad (1)$$

where Δt represents the sampling time during the movement.

We can get that when the rotation speed of the left wheel of the UGV is equal to the rotation speed of the right wheel, and the direction is opposite, the turning radius of the UGV is 0, and the UGV rotates in situ around its particle. The kinematics model of the UGV is the basis of the robot motion control strategy. It provides the theoretical basis of kinematics for the MATLAB simulation of the subsequent obstacle avoidance control algorithm, and provides theoretical support for the motion simulation of the robot.

3. Proposed methodology

3.1. The adaptive weighted multi-sensor information fusion algorithm

According to the basic principles of the adaptive weighting algorithm and the characteristics of each sensor, distinguish the degree of influence of the data obtained by each sensor on the fusion result [29]. Then assign a certain weight to the output data of each sensor in the sensing detection system. The total variance of the weighted fusion result of all data output by the system at one time is used to evaluate the reliability of the algorithm [30]. The model structure of the algorithm is shown in Figure 2.

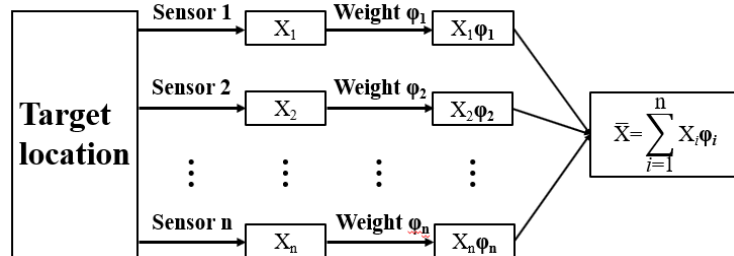


Figure 2. The model structure of the adaptive weighted multi-sensor information fusion algorithm.

As shown in Figure 2, the actual value of the target feature quantity of the sensor detection system is X . In actual measurement work, the output values of each sensor are X_1, X_2, \dots, X_n , the measurement variances of each sensor are $\sigma_1, \sigma_2, \dots, \sigma_n$, the weights assigned to each sensor are $\phi_1, \phi_2, \dots, \phi_n$. According to the model of information fusion algorithm, we can get

$$\begin{cases} \bar{X} = \sum_{i=1}^n \phi_i X_i \\ \sum_{i=1}^n \phi_i = 1 \end{cases} \quad (2)$$

The total variance is

$$\sigma^2 = E \left[\sum_{i=1}^n \phi_i^2 (X - X_i)^2 \right] = \sum_{i=1}^n \phi_i^2 \sigma_i^2 \quad (3)$$

According to Equation (3), the fusion variance σ^2 obtained by the system once detected is a quadratic function relationship with the weight ϕ_i . Therefore, when the fusion variance is the smallest, the accuracy of the fusion result is the highest, and the sensor has been given the most appropriate weight ϕ [31]. Using Lagrange product method to solve the conditional extreme value problem to obtain the weight when the total variance is minimum as

$$\phi_i = \frac{1}{\sigma_i^2 \sum_{i=1}^n \frac{1}{\sigma_i^2}} \quad (4)$$

The ϕ_i obtained by Equation (4) is the most suitable weight that minimizes the data fusion error under the condition that the sum of all the values is 1. Substituting Equation (4) into Equation (3) can get the data distribution after fusion as

$$\sigma^2 = \frac{1}{\sum_{i=1}^n \frac{1}{\sigma_i^2}} \quad (5)$$

After obtaining the fusion weight of the sensors in the system from Equation (4), substituting into Equation (2) to obtain the data fusion result after one measurement. However, from Equation (4), it can be known that if the value of each sensor is required for each detection, the variance of each sensor needs to be known [32].

If $X_i(r)$ represents the output result of the i^{th} sensor when the sensing detection system works for the k^{th} time, the average value of the k^{th} output of all n sensors in the system is calculated as

$$\bar{X}(k) = \frac{1}{n} \sum_{i=1}^n X_i(k) \quad (6)$$

The average value of the variance of each output result of the corresponding sensor included in the information fusion range is obtained, and then used as the final variance value of the i^{th} sensor at the k^{th} measurement sampling.

$$\sigma_i^2(k) = \frac{1}{k} \sum_{s=1}^k E[(X_i(s) - \bar{X}(s))^2] \quad (7)$$

Substitute $\sigma_i^2(r)$ obtained in Equation (7) into Equation (4) to obtain the weight $\varphi_i(r)$ of the i^{th} sensor at the r^{th} measurement sampling. Substitute the weight $\varphi_i(r)$ into Equation (2) to obtain the data fusion result \bar{X} of the i^{th} sensor at the r^{th} measurement sampling.

The multi-sensor information fusion algorithm can process the environmental information obtained by the sensors installed on the UGV, reduce the impact of environmental noise and sensor measurement errors on the data, and input it into the trained FNN. During the obstacle avoidance movement of the unmanned vehicle, the distance information transmitted to the fuzzy neural network is the information after multiple sampling and fusion. Therefore, this algorithm is suitable for the fusion processing of the data collected by the sensor several times in a certain time.

3.2. Design of fuzzy logic controller

Because there are many uncertain factors in the complex working environment of UGVs, it is impossible to establish accurate mathematical models, resulting in many commonly used obstacle avoidance algorithms that are not suitable for UGVs [33]. The fuzzy control algorithm can use the form of fuzzification to transform the information that cannot be accurately described in the environment into the expression form acceptable to human beings, and use professional experience in related fields to reason, and obtain control instructions that meet the system requirements [34].

3.2.1. Determine input and output variables and fuzzification

The distance between the UGV and the obstacle and the azimuth angle of the target point relative to the robot are used as input parameters. The distance \mathbf{d} between the UGV and the obstacle is divided into near and far, denoted by N and F respectively, which is $\{\text{near, far}\} = \{\text{N, F}\}$. The domain is $[0, 100 \text{ mm}]$. The UGV has five sets of sensors for distance measurement, so \mathbf{d} includes $\{d_1, d_2, d_3, d_4, d_5\}$. When the measuring distance is greater than 100mm, it will be treated as 100mm. The membership function with fuzzy parameter N is defined as Equation (8), and the membership function with fuzzy parameter F is defined as Equation (9).

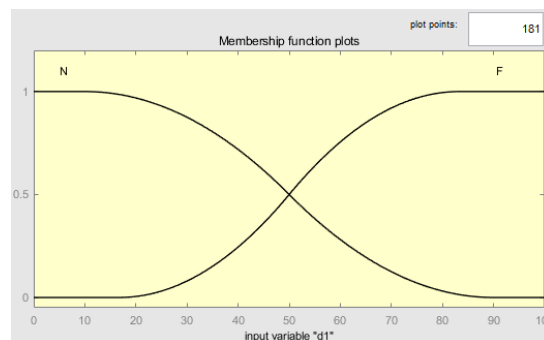
$$u_{ij} = \begin{cases} 1, & x < c_{ij} \\ 1 - \frac{|x - c_{ij}|}{\sigma_{ij}}, & c_{ij} < x < c_{ij} + \sigma_{ij} \\ 0, & x > c_{ij} + \sigma_{ij} \end{cases} \quad (8)$$

$$u_{ij} = \begin{cases} 1, & x > c_{ij} \\ 1 - \frac{|x - c_{ij}|}{\sigma_{ij}}, & c_{ij} - \sigma_{ij} < x < c_{ij} \\ 0, & x < c_{ij} - \sigma_{ij} \end{cases} \quad (9)$$

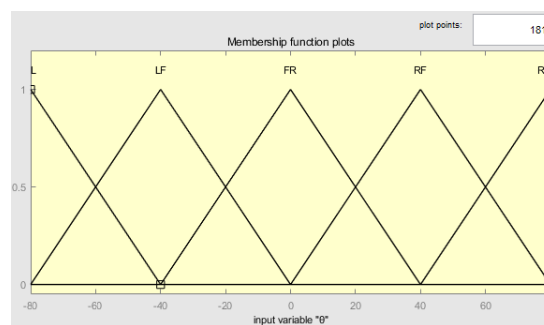
$$u_{ij} = \begin{cases} 1 - \frac{|x - c_{ij}|}{\sigma_{ij}}, & c_{ij} - \frac{\sigma_{ij}}{2} < x < c_{ij} + \frac{\sigma_{ij}}{2} \\ 0, & \text{the others} \end{cases} \quad (10)$$

Where c_{ij} is the central value and σ_{ij} is the width value.

Set the azimuth angle θ of the target point relative to the UGV as the left side, left front side, front right side, right front side and right side, denoted by L, LF, FR, LR, and R respectively, which is {left side, left front side, front right side, right front side, right side} = {L, LF, FR, RF, R}. The domain is $[-80^\circ, 80^\circ]$. When the angle is negative, the target point is on the left side of the UGV; when the angle is zero, the target point is in front of the UGV; when the angle is positive, the target point is on the right side of the UGV. The membership function is defined as Equation (10). The membership function graph of the input quantity is shown in the Figure 3.



(a)



(b)

Figure 3. The membership function graph of the input quantity.

The output parameter inferred by the fuzzy logic controller is set as the deflection angle of the robot, denoted by TG. The fuzzy logic controller designed in this paper divides the output deflection angle TG into five parts: left side, left front side, front right side, right front side and right side, denoted by TL, TLF, ST, TRF and TR respectively, which is {left side, left front side, front right side, right front side, right side}={TL, TLF, ST, TRF, TR}. The membership function is defined as Equation (10). The membership function graph of the input quantity is shown in the Figure 4. When the angle is less than 0, the robot is controlled to turn left; when the angle is equal to 0, the robot travels in a straight line; when the angle is greater than 0, the robot turns to the right.

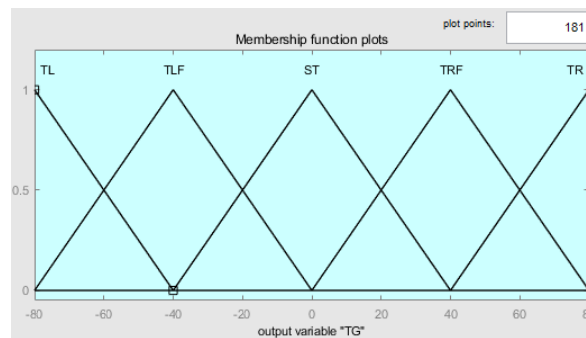


Figure 4. The membership function graph of the output quantity.

3.2.2. Designing Fuzzy Rules

Based on the membership functions of the fuzzy set and inputs and outputs, rules are defined. There are 160 rules for obstacle avoidance of the UGV. We can use the form of ‘IF... THEN’ to express the rules. For example, the number 6 rule is ‘IF ($d_1 = F$, $d_2 = N$, $d_3 = F$, $d_4 = F$, $d_5 = F$, $\theta = L$), THEN ($TG = TL$)’, which means that there is an obstacle in the front left of the UGV and it is closed to the UGV, the target point is on the left side of the forward direction, so the UGV needs to turn left. In addition, Table 1 lists some fuzzy rules.

3.2.3. Defuzzification

The last step of designing the fuzzy logic fusion system is the defuzzification process where outputs are generated based on fuzzy rules, membership values, and a set of inputs. The method used for defuzzification is the Centroid method.

3.3. Design of fuzzy neural network

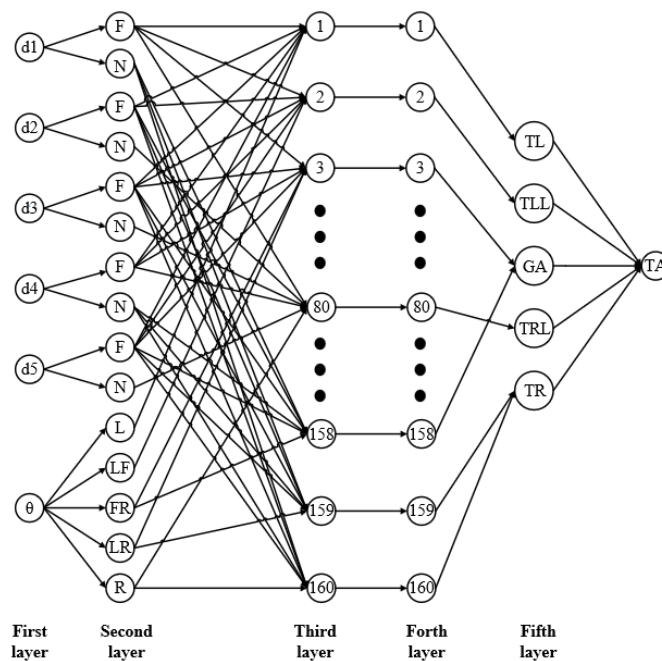
However, the fuzzy control algorithm has the shortcomings of lack of independent learning ability, complex knowledge base construction, and the control rules can’t be adjusted effectively according to the changes of the environment, and it cannot fully meet the needs of UGVs to avoid obstacles [35,36]. The neural network algorithm can optimize the parameters in time according to the actual input data through the learning mechanism, and has a strong ability to adapt to the environment [37]. But the neural network also has the problem of high computational complexity when the input data is non-linear [38,39]. We combined the fuzzy control algorithm with the neural network algorithm, and designed the FNN algorithm to realize the obstacle avoidance function of the UGV.

Table 1. Fuzzy logic rules.

No.	d_1	d_2	d_3	d_4	d_5	θ	TG
1	F	N	F	F	F	L	TL
2	F	N	F	F	F	LF	TL
3	F	F	N	F	F	FR	TRF
4	F	F	N	F	F	LR	TRF
5	F	N	N	F	F	LR	TRF
6	F	N	N	N	F	LF	TL
7	F	N	N	N	F	FR	TR
8	F	F	N	N	N	L	TL
9	F	F	N	N	N	LF	TLF
10	F	N	N	N	N	FR	TL
11	F	N	N	N	N	LR	TL
12	N	F	F	N	N	L	TLF
13	N	F	F	N	N	LF	TLF
14	N	N	N	N	N	L	TL
15	N	N	N	N	N	LF	TL

3.3.1. Structure of fuzzy neural network

The structure of FNN is shown as Figure 5. The first layer is the input layer, whose function is to directly transmit the input data to the next layer. The number of neurons in this layer is $N_1 = 6$, that is, there are six input variables. d_1 , d_2 , d_3 , d_4 , and d_5 represent the distance vectors detected by the sensors in the five directions, and θ is the azimuth angle of the target relative to the robot.

**Figure 5.** Fuzzy neural network structure diagram.

The second layer is the fuzzification layer, the parameters of each node are fuzzy variables and the corresponding membership results. According to the previous design, one input parameter of distance corresponds to two fuzzy language variables, and one input parameter of angle corresponds to five fuzzy language variables. Therefore, the first five neurons in the first layer correspond to 2 neurons in the second layer, which is $m_1 = m_2 = m_3 = m_4 = m_5 = 2$. The sixth neuron in the first layer corresponds to the 5 second-layer neurons. The total number of neurons in this layer is $N_2 = 15$.

The third layer is the reasoning layer, which corresponds to the fuzzy reasoning process, and each fuzzy rule corresponds to a node. $N_3 = \prod_{i=1}^n m_i = 160$. The function of the third layer is to obtain the matching degree of the corresponding fuzzy rules by using the membership degree of each fuzzy variable in this cycle, which is

$$\alpha_j = \mu_1^{i_1} \mu_2^{i_2} \mu_3^{i_3} \mu_4^{i_4} \mu_5^{i_5} \mu_6^{i_6}, \quad (11)$$

where are $i_1 \in \{1,2\}$, $i_2 \in \{1,2\}$, $i_3 \in \{1,2\}$, $i_4 \in \{1,2\}$, $i_5 \in \{1,2\}$, $i_6 \in \{1,2,3,4,5\}$, $j=1, 2, \dots, N_3$.

The fourth layer is the normalization layer to prepare for the clarity of the fuzzy results, which is

$$\bar{\alpha}_j = \alpha_j / \sum_i^{N_3} \alpha_i, \quad (12)$$

where are $j = 1, 2, \dots, N_3$. Number of neurons in this layer is $N_4 = N_3 = 160$.

The fifth layer is the defuzzification layer, which realizes the defuzzification calculation, which is

$$y_i = \sum_{j=1}^{N_4} \bar{\alpha}_j \omega_j, \quad (13)$$

The output is the steering angle, and the weight ω_j is the central value in the membership function of the linguistic variable corresponding to the inference result of the fuzzy rule.

The complexity of neural network can be divided into space complexity and time complexity. The number of layers and the number of parameters that need to be optimized are used to represent the space complexity. Time complexity is usually represented by multiplication and addition operations in neural networks. According to the structure of fuzzy neural network, we can get that the number of layers with computing power is 5, the number of parameters that need to be optimized is 26.

The time complexity of fuzzy neural network can be calculated layer by layer. If we only have one training sample, the time complexity is $15 + 160 + 160 + 160 + 1 = 496$. Assuming that the actual number of training samples is m , the time complexity of the whole algorithm is $O(496m) = O(m)$.

3.3.2. Parameters Optimization Process

Using neural network training to optimize the central value c_{ij} and width value σ_{ij} of the membership function in the second layer of the neural network, as well as the weight w_j between each node. There are 15 central values c_{ij} , which correspond to the neurons of the second layer one-to-one; a total of 6 width values σ_{ij} , which are the same as the number of input parameters; and a total of 5 weights w_j , which correspond to the number of elements in the fuzzy language set of the output parameters. The training of the neural network adopts the feedforward adjustment method and the gradient descent method of error back propagation. The weight adjustment formula is

$$c_{ij}(t+1) = c_{ij}(t) - \eta \frac{\partial E}{\partial c_{ij}}, \quad (14)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta \frac{\partial E}{\partial \sigma_{ij}}, \quad (15)$$

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial E}{\partial \omega_{ij}}, \quad (16)$$

where η is the learning rate. The flow of FNN algorithm is shown in Figure 6.

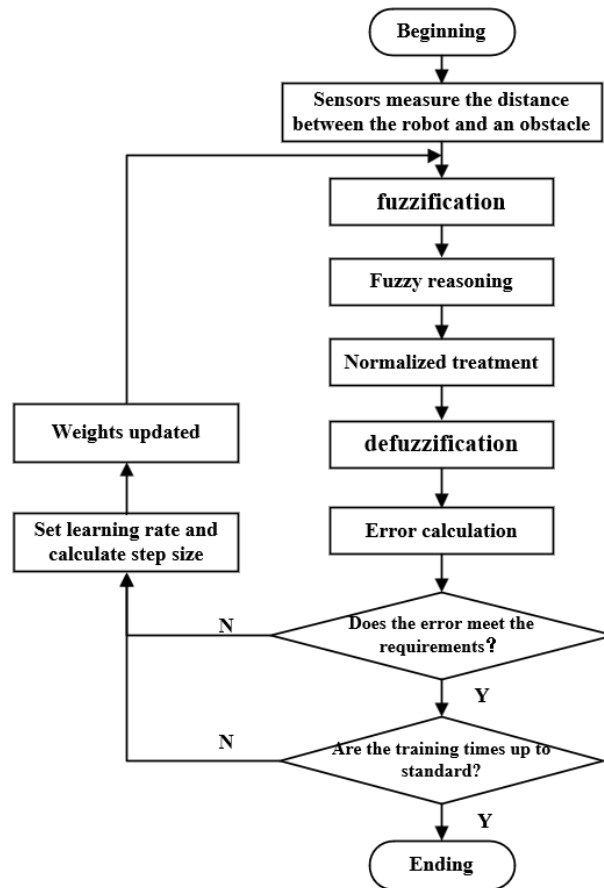


Figure 6. The flow of FNN algorithm.

4. Simulation of fuzzy neural network algorithm and discussion

Formulating fuzzy inference rules based on the input parameters, output parameters and their corresponding fuzzy language variables as mentioned above. Selecting the initial value of the weight parameter and establish a fuzzy controller. The initial value of the weight parameter is shown in Table 2.

Using MATLAB software to establish a static multi-obstacle simulation environment. Before the neural network is used to optimize the weight parameters, a fuzzy controller is used to conduct the simulation experiment of obstacle avoidance motion of the UGV. When the rotation speed of the left wheel of the UGV is equal to the rotation speed of the right wheel, and the direction is opposite, the turning radius of the UGV is 0. Therefore, the position of the robot does not change when turning. We have established a robot sports field in MATLAB. There are nine obstacles of different shapes randomly distributed in the field. The robot needs to move from the lower left corner to the upper right corner of the field. Since the UGV designed in this article can turn in a fixe place, the influence of the rectangular structure on the obstacle avoidance function can be ignored in the simulation experiment, and the UGV was assumed to be a circle. In the simulation process, the robot's working sequence is to first measure the distance between the robot and the obstacle, the fuzzy controller calculates the steering angle, and the robot advances one step according to the steering angle. According to the actual distribution of sensors, the distance in five directions needs to be calculated for each advance.

The obstacle avoidance motion path of the UGV and the angle at which the UGV turns each time is shown in Figure 7. The blue circle in the lower left corner of the figure represents the starting position of the UGV, and the red cross in the upper right corner represents the target position of the UGV.

Table 2. The initial value of the weight parameter.

weight	value	weight	value	weight	value	weight	value	weight	value
c_{11}	10	c_{12}	90	c_{21}	10	c_{22}	90	c_{31}	10
c_{32}	90	c_{41}	10	c_{42}	90	c_{51}	10	c_{52}	90
c_{61}	-80	c_{62}	-40	c_{63}	0	c_{64}	40	c_{65}	80
σ_1	80	σ_2	80	σ_3	80	σ_4	80	σ_5	80
σ_6	40	w_1	-80	w_2	-40	w_3	0	w_4	40
w_5	80								

The simulation result of the fuzzy control algorithm navigating the obstacle avoidance movement of the UGV is shown in Figure 7. The simulation results show that there is an intersection point between the robot's motion path and the obstacle. Although there is a certain distance between the particle representing the robot and the obstacle, there is an intersection point between the figure representing the contour of the robot and the obstacle. The location of the collision is indicated by the red circle. The simulation results show that the obstacle avoidance motion of the robot under the navigation of the fuzzy control algorithm cannot completely avoid all obstacles and reach the target position safely. Therefore, using fuzzy control algorithm alone cannot meet the obstacle avoidance requirement of UGV.

Using fuzzy controllers to navigate UGVs in different simulation environments, planning optimal robot motion paths artificially in the simulation environments, the distances between the robot and the obstacle in the path and the relative angle between the robot and the target point are collected. The deflection angle of each advance in the ideal trajectory is calculated as the expected output value of the FNN. Using FNN to optimize the weight parameters, setting the learning rate to 0.001, and the allowable error to 0.001. When the error is less than 0.001 or the number of training times is greater than 3000 times, the training stops. The network weights after training adjustment (retaining two decimal places) are shown in Table 3.

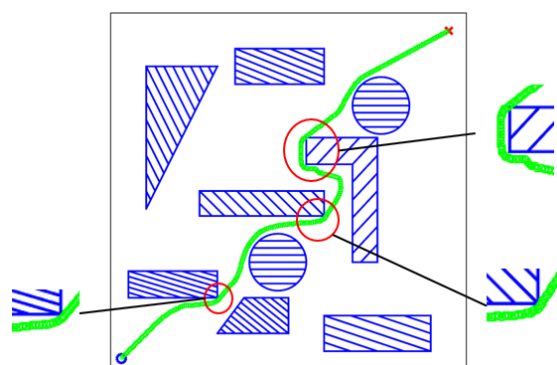


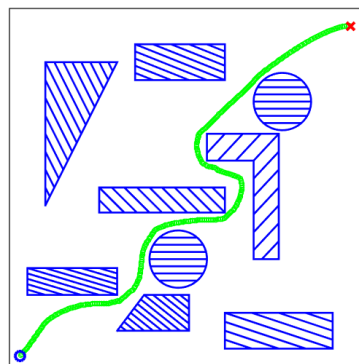
Figure 7. Simulation results of the simulation experiment without neural network.

According to the optimization result, the value of the weight parameter corresponding to the fuzzy controller is adjusted and substituted into the obstacle avoidance algorithm program to obtain the simulation result of the obstacle avoidance motion path as shown in Figure 8.

Table 3. The value of the weight parameter after training adjustment.

weight	value	wight	value	weight	value	weight	value	weight	value
c ₁₁	9.95	c ₁₂	90.04	c ₂₁	9.91	c ₂₂	90.10	c ₃₁	9.90
c ₃₂	90.10	c ₄₁	9.91	c ₄₂	90.10	c ₅₁	9.91	c ₅₂	90.10
c ₆₁	-80.60	c ₆₂	-40.60	c ₆₃	-0.50	c ₆₄	40.50	c ₆₅	80.60
σ_1	80.9	σ_2	80.19	σ_3	80.20	σ_4	80.19	σ_5	80.5
σ_6	40	w ₁	-79.50	w ₂	-39.50	w ₃	-0.50	w ₄	39.50
w ₅	80.00								

As shown in Figure 8, the robot can maintain a certain distance from the surrounding obstacles and reach the target point smoothly during the movement. Comparing Figure 7 and Figure 8 we can find that the obstacle avoidance movement path of the robot in Figure 8 is smoother, and the distance between the robot and the obstacle is more obvious. Especially at the apex of the rectangular obstacle, the movement path is significantly farther away from the obstacle. The simulation results show that the designed FNN obstacle avoidance algorithm can realize the obstacle avoidance function and has a better obstacle avoidance effect.

**Figure 8.** Simulation results of the simulation experiment.

We collected the distance between the robot and the obstacle calculated every time the robot advanced, and selected the smallest value among the five distances. The distance between the robot and the obstacle in the simulation experiment under the fuzzy controller and FNN algorithm navigation is shown in Figure 9. By comparing the two curves, the overall trend of the blue curve is above the yellow curve, indicating that the distance between the robot and the obstacle under the control of the FNN algorithm is larger than that of the fuzzy controller. The count of different distance ranges is shown in Figure 10, we can get that the distances between the robot and the obstacle under the FNN navigation are all greater than 10cm. These conformed to the boundary setting of the robot in the simulation experiment, indicating that the robot will not collide with obstacles. In addition, the distance controlled by the FNN algorithm is more distributed in a larger interval. Therefore, we proved that the FNN algorithm has certain advantages compared with the fuzzy control algorithm.

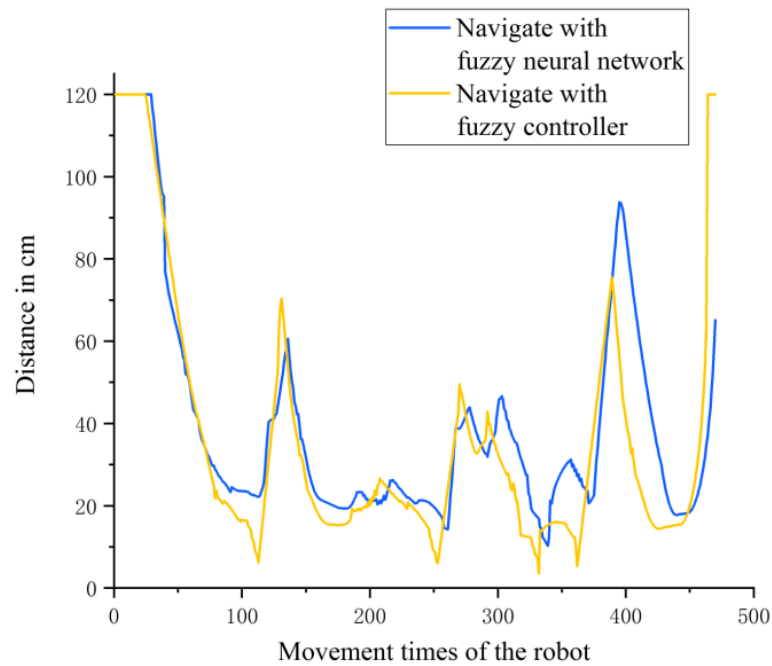


Figure 9. The distance between the robot and the obstacle in the simulation movement.

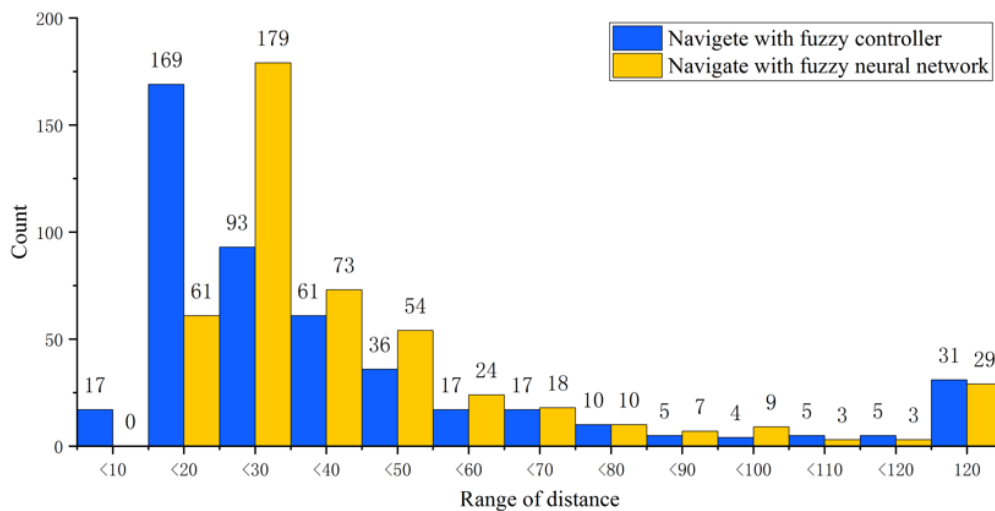


Figure 10. Counting distance distribution under navigation of two algorithms.

5. Real obstacle avoidance experiment and discussion

5.1. Unmanned ground vehicle

The UGV used in this paper is a wheeled unmanned vehicle designed and developed independently. The UGV compactly installs all the electronic components and protects them with mechanical structure to avoid unnecessary damage caused by the exposed electronic components, especially for the protection of sensors and control center. The overall appearance of the UGV (including all parts and components such as tires) is 260mm×280mm×110mm. The 3D model and the

physical picture of the UGV is shown in Figure 11.

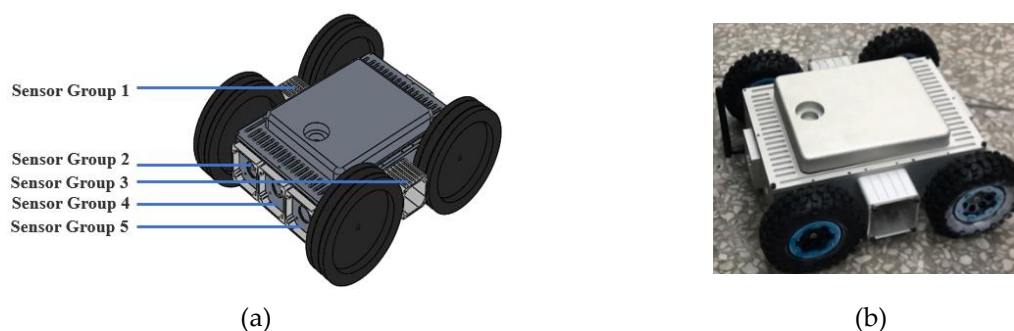


Figure 11. The 3D model and the physical picture of the UGV.

In order to meet the application requirements of UGVs for avoiding obstacles and moving in complex environments, the robot's environment detection system includes multiple sets of sensors to obtain distance information of obstacles. The obstacle avoidance system includes a total 10 sensors, including five ultrasonic sensors and five infrared sensors. The locations where the sensors are installed are depicted in Figure 1. The 10 sensors were divided into five groups and installed around the robot body. Five groups of sensors are distributed within a 180° range in front of the robot. Each group of sensors consists of an ultrasonic sensor and an infrared sensor, providing a hardware foundation for the application of multi-sensor information fusion technology. In addition, there is an electronic compass installed inside the UGV to measure the deflection angle of the UGV.

5.2. The experiment and discussion

In order to verify the designed obstacle avoidance technology based on multi-sensor information fusion technology and FNN algorithm, we designed an obstacle avoidance experiment based on the self-designed UGV platform mentioned above. The effect of obstacle avoidance experiment with the robot were analyzed and discussed to verify the correctness and feasibility of the algorithm.

Imitated the motion environment in the simulation experiment to build the motion environment of the robot obstacle avoidance experiment. The initial motion direction of the UGV was set as a constant 90° , that is, it moves towards the front. The steering angle obtained by the UGV in subsequent obstacle avoidance calculation needs to be calculated by adding or subtracting with this constant to get a new angle value, which is then stored in the register of the control core. After the first steering of the UGV, each obstacle avoidance calculation took the angle value as the relative direction of the target point as the input of the fuzzy controller. The purpose of this design is to minimize the impact of the obstacle avoidance process on the original driving direction and keep the original driving direction. We repeated the experiment under the same conditions. The result of the obstacle avoidance experiment is shown in Figure 12 and Figure 13. The black solid lines behind the UGV in the figures are the motion track of the UGV.

Figure 12 shows the obstacle avoidance experiment results of the UGV under the navigation of the fuzzy controller designed above. There is little difference between the results of repeated experiments, which means that the fuzzy controller has poor adaptability to the environment. As shown in Figure 12(c) and Figure 12(f), the robot collided with the obstacles twice. In the process of robot movement, the distance between the UGV and the obstacle is relatively close. The fuzzy control algorithm cannot be adjusted according to the actual situation of obstacle avoidance, and the weight

parameters used are not applicable to the actual environment. This indicates that the fuzzy control algorithm cannot meet the obstacle avoidance requirement of the robot.

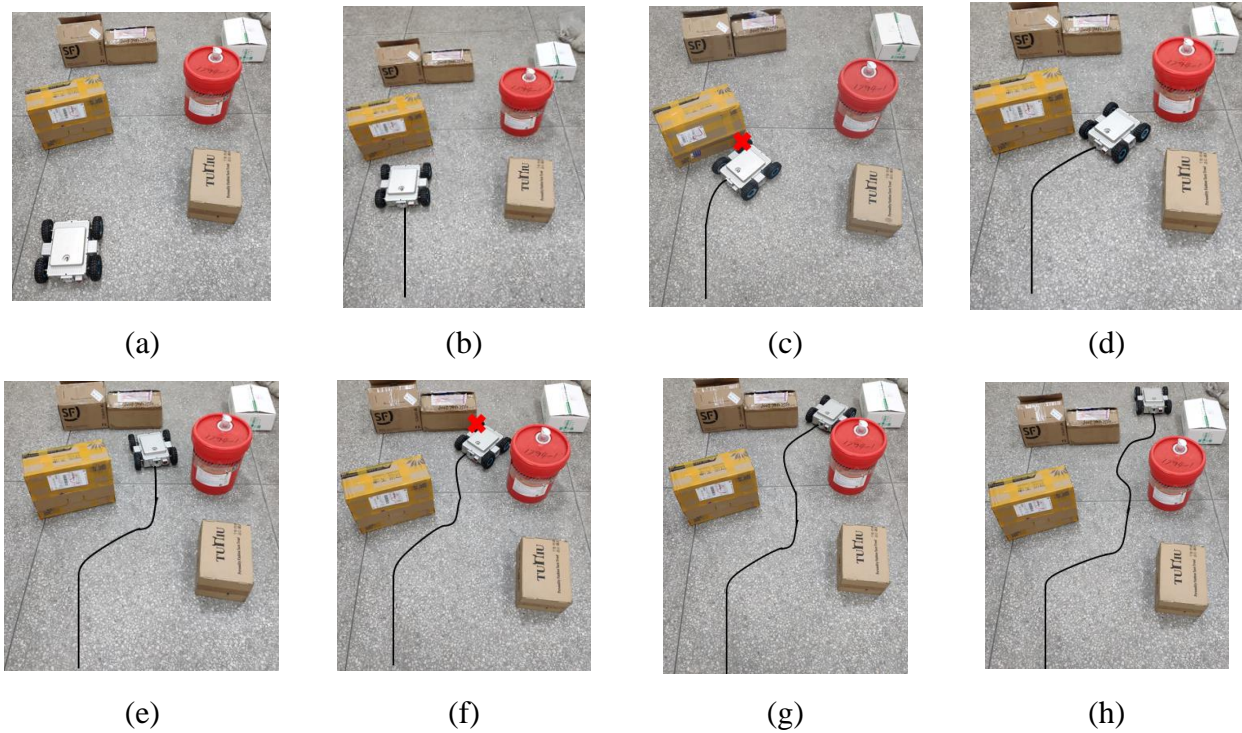


Figure 12. Experiment on robot obstacle avoidance under the control of fuzzy controller.

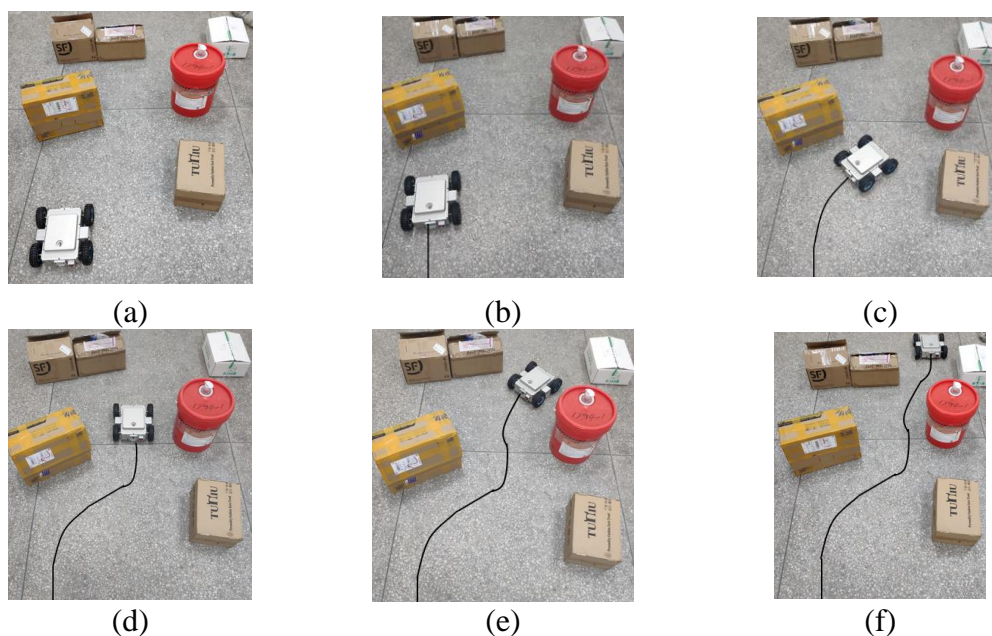


Figure 13. Experiment on robot obstacle avoidance under the control of FNN.

Figure 13 shows the obstacle avoidance experiment of the UGV under the control of FNN algorithm. The experimental results show that the UGV can smoothly avoid obstacles and keep a proper distance from the obstacle in this complex environment, which proves that the robot has a good

obstacle avoidance function. By comparing the entire path of motion showed in Figure 12(h) with Figure 13(f), the number of movement direction changes of UGV is significantly less in Figure 13(f), which means the UGV reacted more quickly to obstacles under the navigation control of FNN algorithm. By analyzing the results of obstacle avoidance experiments, the correctness and feasibility of the designed FNN algorithm are verified.

6. Conclusions

This article designed and verified an obstacle avoidance method based on multi-sensor information fusion technology and FNN algorithm. The motion model of the UGV was established combined the structure and motion form, and the motion control strategy of the UGV was determined. Ten distance sensors and an angle measuring sensor were used for the obstacle avoidance behavior. Then an adaptive weighted multi-sensor information fusion algorithm and a FNN obstacle avoidance algorithm were designed and applied to the UGV. The fusion model designed is based on the fuzzy logic inference system, which is composed of six inputs, one output and 160 fuzzy rules. Multiple membership function for inputs and outputs are chosen. The parameters of membership functions were optimized by the neural network. By analyzing the simulation experiment results and actual experiment results under the navigation control of fuzzy controller and FNN algorithms, the superiority and reliability of the FNN algorithm have been proved.

Acknowledgments

The work presented in this paper was supported by the Inner Mongolia First Machinery Group Co., Ltd. State Key Laboratory of Special Vehicles and Drive Systems Intelligent Manufacturing Project Open Project (GZ2019KF001). The work presented in this paper was also supported by the 2017 Baiyun District, Guangzhou Innovation and Entrepreneurship leading team. Research and industrialization project of key technologies of intelligent flexible drive assisted exoskeleton (BYCX-(Innovation)).

Conflict of interest

All authors declare that they have no conflict of interest.

References

1. M. Al-Sagban, R. Dhaouadi, Neural based autonomous navigation of wheeled mobile robots, *J. Autom. Mob. Robot. Intell. Syst.*, **10** (2016), 64–72.
2. K. H. Anabi, R. Nordin, N. F. Abdullah, Database-assisted television white space technology: challenges, trends and future research directions, *IEEE Access*, **4** (2016), 8162–8183.
3. D. Chwa, Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances, *IEEE Trans. Fuzzy Syst.*, **20** (2012), 587–593.
4. K. Goldberg, One Robot is Robotics, Ten Robots is Automation, *IEEE Trans. Autom. Sci. Eng.*, **13** (2016), 1418–1419.
5. C. M. Luo, J. Y. Gao, X. D. Li, H. W. Mo, Q. M. Jiang, Sensor-based autonomous robot navigation under unknown environments with grid map representation. 2014 IEEE Symposium on Swarm Intelligence, (2014), pp. 1–7.

6. A. Mukhtar, L. Xia, T. B. Tang, Vehicle detection techniques for collision avoidance systems: A review, *IEEE Trans. Intell. Transp. Syst.*, **16** (2015), 2318–2338.
7. P. Subbash, K. T. Chong, Adaptive network fuzzy inference system based navigation controller for mobile robot, *Front. Inform. Technol. Elect. Eng.*, **20** (2019), 141–151.
8. C. Treesatayapun, Discrete-time direct adaptive control for robotic systems based on model-free and if–then rules operation, *Int. J. Adv. Manuf. Technol.*, **68** (2013), 575–590.
9. C.-C. Tsai, H.-L. Wu, F.-C. Tai, Y.-S. Chen, Distributed consensus formation control with collision and obstacle avoidance for uncertain networked omnidirectional multi-robot systems using fuzzy wavelet neural networks, *Int. J. Fuzzy Syst.*, **19** (2016), 1375–1391.
10. J. Savage, S. Muñoz, M. Matamoros, R. Osorio, Obstacle avoidance behaviors for mobile robots using genetic algorithms and recurrent neural networks, *IFAC Proceed. Vol.*, **46** (2013), 141–146.
11. A. Pandey, S. Kumar, K. K. Pandey, D. R. Parhi, Mobile robot navigation in unknown static environments using ANFIS controller, *Perspect. Sci.*, **8** (2016), 421–423.
12. C.-J. Kim, D. Chwa, Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network, *IEEE Trans. Fuzzy Syst.*, **23** (2015), 677–687.
13. S. Goudarzi, N. Kama, M. H. Anisi, S. Zeadally, S. Mumtaz, Data collection using unmanned aerial vehicles for Internet of Things platforms, *Comput. Electr. Eng.*, **75** (2019), 1–15.
14. D. Z. Wan, C. S. Chin, Simulation and prototype testing of a low-cost ultrasonic distance measurement device in underwater, *J. Mar. Sci. Technol.*, **20** (2014), 142–154.
15. H. Yang, X. Fan, P. Shi, C. Hua, Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint, *IEEE Trans. Control Syst. Technol.*, (2015), 1.
16. A. M. Alajlan, M. M. Almasri, K. M. Elleithy, Multi-sensor based collision avoidance algorithm for mobile robot. 2015 Long Island Systems, Applications and Technology, (2015), pp. 1–6.
17. Z. H. Duan, T. H. Wu, S. W. Guo, T. Shao, R. Malekian, Z. Li, Development and trend of condition monitoring and fault diagnosis of multi-sensors information fusion for rolling bearings: A review, *Int. J. Adv. Manuf. Technol.*, **96** (2018), 803–819.
18. S. W. Yoon, S.-B. Park, J. S. Kim, Kalman filter sensor fusion for mecanum wheeled automated guided vehicle localization, *J. Sens.*, **2015** (2015), 1–7.
19. B. Khaleghi, A. Khamis, F. O. Karray, S. N. Razavi, Multisensor data fusion: A review of the state-of-the-art, *Inf. Fusion*, **14** (2013), 28–44.
20. M. Alataise, G. Hancke, Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended kalman filter, *Sensors*, **17** (2017).
21. B. F. Ji, Y. Q. Li, D. Cao, C. G. Li, S. Mumtaz, D. Wang, Secrecy performance analysis of UAV assisted relay transmission for cognitive network with energy harvesting, *IEEE Trans. Veh. Technol.*, **69** (2020), 7404–7415.
22. T. Tang, T. Hong, H. H. Hong, S. Y. Ji, S. Mumtaz, M. Cheriet, An improved UAV-PHD filter-based trajectory tracking algorithm for multi-UAVs in future 5G IoT scenarios, *Electronics*, **8** (2019).
23. Z. Y. Lin, L. L. Wang, Z. M. Han, M. Y. Fu, Distributed formation control of multi-agent systems using complex laplacian, *IEEE Trans. Autom. Control*, **59** (2014), 1765–1777.
24. C. G. Zong, Z. J. Ji, Y. Yu, H. Shi, Research on obstacle avoidance method for mobile robot based on multisensor information fusion, *Sens. Mater.*, **32** (2020).
25. T. Tian, S. L. Sun, N. Li, Multi-sensor information fusion estimators for stochastic uncertain systems with correlated noises, *Inf. Fusion*, **27** (2016), 126–137.

26. M. Almasri, K. Elleithy, A. Alajlan, Sensor fusion based model for collision free mobile robot navigation, *Sensors (Basel)*, **16** (2015).
27. A. Al-Mayyahi, W. Wang, P. Birch, Adaptive neuro-fuzzy technique for autonomous ground vehicle navigation, *Robotics*, **3** (2014), 349–370.
28. D. Y. Qu, Y. H. Hu, Y. T. Zhang, The investigation of the obstacle avoidance for mobile robot based on the multi sensor information fusion technology, *Int. J. Mater., Mecha. Manuf.* (2013), 366–370.
29. I. Aydin, S. B. Celebi, S. Barmada, M. Tucci, Fuzzy integral-based multi-sensor fusion for arc detection in the pantograph-catenary system, *Proc. Inst. Mech. Eng. Part F-J. Rail Rapid Transit*, **232** (2016), 159–170.
30. H. L. Xiong, Z. Z. Mai, J. Tang, F. Hen, Robust GPS/INS/DVL navigation and positioning method using adaptive federated strong tracking filter based on weighted least square principle, *IEEE Access*, **7** (2019), 26168–26178.
31. F. Xiao, B. Qin, A weighted combination method for conflicting evidence in multi-sensor data fusion, *Sensors (Basel)*, **18** (2018).
32. D. H. Li, C. Shen, X. P. Dai, X. Zhu, Z. Liang, Research on data fusion of adaptive weighted multi-source sensor, *CMC-Comput. Mat. Contin.*, **61** (2019), 1217–1231.
33. C.-H. Hsu, C.-F. Juang, Evolutionary robot wall-following control using type-2 fuzzy controller with Species-DE-Activated continuous ACO, *IEEE Trans. Fuzzy Syst.*, **21** (2013), 100–112.
34. G.-D. Wu, P.-H. Huang, A vectorization-optimization-method-based type-2 fuzzy neural network for noisy data classification, *IEEE Trans. Fuzzy Syst.*, **21** (2013), 1–15.
35. M. Faisal, R. Hedjar, M. Al Sulaiman, K. Al-Mutib, Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment, *Int. J. Adv. Robot. Syst.*, **10** (2013).
36. D. Wu, Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons, *IEEE Trans. Fuzzy Syst.*, **21** (2013), 80–99.
37. H. Boubertakh, M. Tadjine, P. Y. Glorennec, A new mobile robot navigation method using fuzzy logic and a modified Q-learning algorithm, *J. Intell. Fuzzy Syst.*, **21** (2010), 113–119.
38. P. Melin, L. Astudillo, O. Castillo, F. Valdez, Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm, *Expert Syst. Appl.*, **40** (2013), 3185–3195.
39. J. R. Castro, O. Castillo, P. Melin, A. Rodríguez-Díaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, *Inf. Sci.*, **179** (2009), 2175–2193.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)