



# The Analysis of Trajectory Control of Non-holonomic Mobile Robots Based on Internet of Things Target Image Enhancement Technology and Backpropagation Neural Network

Lanfei Zhao<sup>1</sup>, Ganlin Wang<sup>2</sup>, Xiaosong Fan<sup>2</sup> and Yufei Li<sup>3,4\*</sup>

<sup>1</sup> The Higher Educational Key Laboratory for Measuring and Control Technology and Instrumentations of Heilongjiang Province, Harbin University of Science and Technology, Harbin, China, <sup>2</sup> Huabei Oil Communication Co., Ltd., Cangzhou, China, <sup>3</sup> Medical Cosmetic Center, Department of Dermatology, Tongji Hospital, Tongji University School of Medicine, Shanghai, China, <sup>4</sup> Department of Plastic Surgery, Huashan Hospital, Fudan University, Shanghai, China

## OPEN ACCESS

### Edited by:

Mu-Yen Chen,  
National Taichung University of  
Science and Technology, Taiwan

### Reviewed by:

Hsin-Te Wu,  
National Ilan University, Taiwan  
Gwo-Jiun Horng,  
Southern Taiwan University of Science  
and Technology, Taiwan

### \*Correspondence:

Yufei Li  
lufylee@hotmail.com

Received: 27 November 2020

Accepted: 22 February 2021

Published: 22 March 2021

### Citation:

Zhao L, Wang G, Fan X and Li Y  
(2021) The Analysis of Trajectory  
Control of Non-holonomic Mobile  
Robots Based on Internet of Things  
Target Image Enhancement  
Technology and Backpropagation  
Neural Network.  
*Front. Neurorobot.* 15:634340.  
doi: 10.3389/fnbot.2021.634340

The trajectory tracking and control of incomplete mobile robots are explored to improve the accuracy of the trajectory tracking of the robot controller. First, the mathematical kinematics model of the non-holonomic mobile robot is studied. Then, the improved Backpropagation Neural Network (BPNN) is applied to the robot controller. On this basis, a mobile robot trajectory tracking controller combining the fuzzy algorithm and the neural network is designed to control the linear velocity and angular velocity of the mobile robot. Finally, the robot target image can be analyzed effectively based on the Internet of Things (IoT) image enhancement technology. In the MATLAB environment, the performances of traditional BPNN and improved BPNN in mobile robots' trajectory tracking are compared. The tracking accuracy before and after the improvement shows no apparent differences; however, the training speed of improved BPNN is significantly accelerated. The fuzzy-BPNN controller presents significant improvements in tracking speed and tracking accuracy compared with the improved BPNN. The trajectory tracking controller of the mobile robot is designed and improved based on the fuzzy BPNN. The designed controller combining the fuzzy algorithm and the improved BPNN can provide higher accuracy and tracking efficiency for the trajectory tracking and control of the non-holonomic mobile robots.

**Keywords:** backpropagation neural network, Internet of Things, image enhancement, non-holonomic mobile robot, trajectory tracking and control

## INTRODUCTION

As human society enters the era of science and technology, computers, and artificial intelligence have developed rapidly; machines to replace human labor to improve production efficiency have become a reality (Ma et al., 2020). However, Lv et al. (2019) also proposed that the existing network structure migrated computing tasks to the cloud, while the increase in cloud data transmission put huge pressure on the core network and affected the quality of service (Lv and Xiu, 2019). Internet of Things (IoT) is a famous object vision.

Information-sensing equipment, such as sensors and electronic tags installed on the object, transmits information collected back and forth through the internet connection according to the agreed protocol. Connecting a simple robot to the internet will become valuable because it can obtain updated information about its environment from sensors or understand the user's whereabouts and the status of nearby devices (Marques et al., 2019). In short, robots integrated with IoT can use IoT data to help machines interact with each other and take necessary actions, enabling robots to communicate effectively and make appropriate decisions by themselves. The core of "IoT+Robot" is the ubiquitous sensors, cameras, and actuators embedded in the environment, as well as autonomous robots that collect data in real-time (Rehman et al., 2019). Sensors provide not only raw data but also interpretation and abstraction to some degree, which can be utilized for decision-making or high-level automation. Lv et al. (2019) applied the sensor technology of the ZigBee wireless network organization, which could provide people with a smarter and more comfortable living environment (Lv et al., 2019). Connecting machine vision systems to IoT can create powerful network functions that can recognize objects from cameras. Such functions can enhance the local nodes' intelligence and autonomy, reducing the processing load on the central server and achieving a better-distributed control architecture.

The IoT-based multi-robot collaborative operation utilizes the intelligent perception inside and outside the robot, makes timely judgments and control decisions according to the signals collected by the network, and timely issues control instructions to the robot to ensure that multiple robots complete tasks safely and efficiently (Özdemir, 2019; Michie et al., 2020). Mobile robots are highly intelligent systems that can continuously obtain information of the surrounding environment and themselves through sensors in real-time, make decisions, analyze, plan for different environments, and control the drive motors to move autonomously toward the targets, thereby completing specific tasks.

Tracking control in mobile robots' motion control has always been a sophisticated problem, which has received extensive attention from researchers in this field. Yang and Pan proposed a sliding mode control method for wheeled mobile robots, established a motion control model for mobile robots, and designed a sliding mode trajectory tracking controller, which effectively reduced the jitter of sliding mode control's input, accelerated the convergence speed, and improved the tracking accuracy (Yang and Pan, 2018). Tinh and Linh improved the online weight adjustment algorithm based on backpropagation and proposed an adaptive tracking controller based on a three-layer neural network, which could ensure the stability of the entire closed-loop system and realize the desired mobile robots' trajectory tracking performance (Tinh and Linh, 2018).

In practical applications, trajectory tracking of mobile robots should ensure versatility while ensuring stability and robustness optimization. Therefore, the focus is on the trajectory control of mobile robots. Here, a trajectory tracking method combining improved Backpropagation Neural Network (BPNN) and fuzzy neural network is proposed, denoted as fuzzy-BPNN. Then

this method's feasibility and accuracy in robots' trajectory tracking and control are verified through simulation. On this basis, the IoT multi-sensor data fusion and target infrared image enhancement technology are researched. The multi-robot system can fuse the sensor data, enhance the target images, and provide new ideas for robot obstacle avoidance through fuzzy control.

## MATERIALS AND METHODS

### Mathematical Kinematics Models of Non-holonomic Mobile Robots

In mechanics, mathematical equations containing coordinate parameters can express constraints. The motion constraints that mobile robots are subjected to include holonomic constraints and non-holonomic constraints. Holonomic constraints are restrictions on the configuration space, and non-holonomic constraints are restrictions on system motion (Chu et al., 2018). The holonomic constraint reduces the dimensionality of the configuration space. The system can transform the holonomic constraints it receives into the constraints on the position through integration during the motion. The non-holonomic constraints reduce the dimensionality of the velocity. The equations of holonomic constraints and non-holonomic constraints can be expressed as Equations (1) and (2):

$$h(q, t) = 0 \quad (1)$$

$$h(q, \dot{q}, t) = 0 \quad (2)$$

In Equations (1) and (2),  $q$  represents the coordinate vector of the system,  $\dot{q}$  represents the velocity vector of the system, and  $t$  is the time parameter.

For practical problems, the motion constraint can be transformed into a linear relationship with the system velocity  $\dot{q}$ ; that is, the Pfaffian constraint:

$$h(q, \dot{q}, t) = \sum_{i=1}^n A_i(q) \cdot (\dot{q}) = A(q) \cdot (\dot{q}) = 0 \quad (3)$$

In Equation (3),  $A(q) \in R^{m \times n}$  represents a set of  $m$  velocity constraints, and  $A_i(q) \in R^{1 \times n}$  is the row vector of  $A(q)$ , which is a constraint on the direction of the generalized velocity  $q$  of the system.

A wheel moves on the ground, as shown in **Figure 1**. Four parameters can describe the wheel's configuration: the contact points  $x$  and  $y$  with the ground, the current rotation angle  $\theta$ , and the forward direction  $\Phi$ . If the wheel makes a non-slip motion, the direction of the wheel will always be  $(\cos\phi, \sin\phi)$ . The non-slip constraint does not reduce the dimension of the wheel configuration space; that is, the wheel can reach any position on the plane. However, this constraint reduces the dimensionality

of the wheel's velocity space so that the wheel can only move in direction  $(\cos\phi, \sin\phi)$  at a particular time.

Assuming that the wheel's radius is  $r$ , the distance between the wheel's center  $(x_c, y_c)$  and the ground track is always the same during the non-slip rolling motion of the linear track. The non-slip rolling between it constrains the wheel's motion and the ground, expressed as the instantaneous velocity  $v$  of the wheel's contact point and the ground track is zero. Therefore, the wheel's motion can be regarded as a rapid rotation around the contact point, and the constraint equation can be expressed as:

$$y_c = r \tag{4}$$

$$A(q) \cdot (\dot{q}) = [1 \ 0 \ -r] \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\phi} \end{bmatrix} = \dot{x}_c - r \dot{\phi} = v = 0 \tag{5}$$

Equation (4) is a geometric constraint on the wheel, and Equation (5) represents a linear motion constraint, which can be further expressed by integral:

$$x_c - r\phi = C \tag{6}$$

In Equation (6),  $C$  is the integral constant.

The non-holonomic constraint equation does not have a corresponding geometric constraint. Hence, it puts no restriction on the position vector but only restrictions on the particle's velocity at each position. Therefore, the non-holonomic constraint does not reduce the number of independent generalized coordinates; instead, it only reduces the number of independent generalized velocities (Gutiérrez-Giles et al., 2018).

A non-holonomic mobile robot is also a non-holonomic system, and the non-holonomic constraint equation can reflect

its motion characteristics. While the robots are moving, two physical phenomena, tire rolling and sliding, will occur when the wheels contact the ground. The structure of a non-holonomic mobile robot is shown in **Figure 2**. One coordinate system is the global coordinate system  $XOY$ ; the other coordinate system is the local coordinate system  $xoy$  (the mass center of the robot is the origin). In **Figure 2**,  $R$  is the radius of the robot's driving wheel,  $\theta$  is the robot's forward direction angle,  $q$  is the robot's pose,  $[v \ \omega]^T$  is the robot's control quantity (linear velocity and angular velocity),  $v_L$  and  $v_R$  are the linear velocity of the robot's left and right wheels, and  $L$  is the distance between the centers of the robot's two driving wheels.

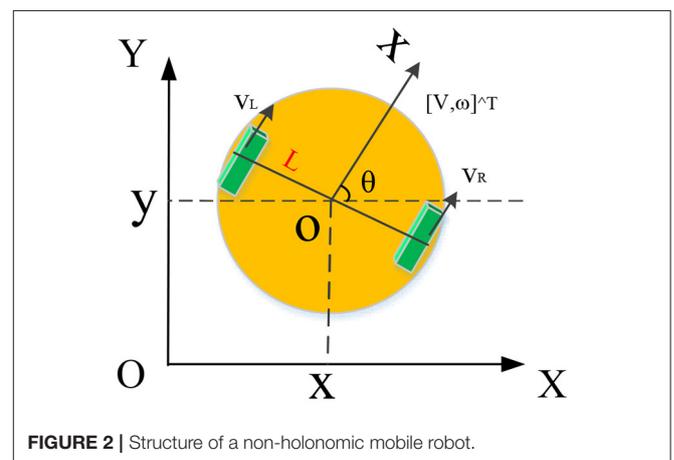
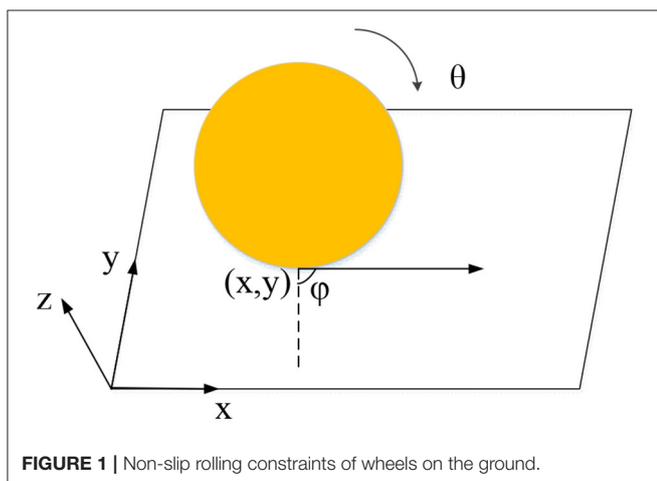
For error reduction, assuming that the mobile robot moves along a linear track gradually, the wheels do not slide left and right. Then the non-holonomic constraints and the kinematics model can be expressed as in Equations (7) and (8):

$$\dot{y} \cos\phi - \dot{x} \sin\theta = 0 \tag{7}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8}$$

The non-holonomic mobile robot's left and right wheels' linear velocities  $v_L$  and  $v_R$  share the following relationship with its linear velocity  $v$  and angular velocity  $\omega$  :

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1/R & L/2R \\ 1/R & L/2R \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{9}$$



In summary, the kinematics model of the discussed non-holonomic mobile robot is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \quad (10)$$

### Robot Trajectory Tracking Based on Improved BPNN

Mobile robots are widely used in unmodeled spaces and environments; thus, accurate trajectory tracking is the basis of practical applications (Alshakarchi and Al-Maliky, 2018; Singh and Thongam, 2018; Tu et al., 2019; Wang et al., 2019). The traditional control method depends too much on the dynamic model, resulting in low robustness. The intelligent neural networks have strong robustness and adaptability, presenting significant advantages in trajectory tracking and control of mobile robots. As the core of the feedforward network in artificial neural networks, BPNN is widely applied to solve problems such as function approximation and pattern recognition (Yi et al., 2019). The most commonly used transfer functions of backpropagation neurons are log function and tan function, and the output can be expressed as  $y = \log \text{sig}(Wp + b)$ . Generally, BPNN presents a multi-layer structure. The model of backpropagation neuron and the two-layer structure diagram are shown in Figure 3.

BPNN needs to adjust the weights according to each training sample, which requires a massive amount of training data in practical applications, resulting in reduced efficiency of weight adjustment and failure to meet the real-time requirements (Singh and Thongam, 2019). Therefore, the traditional BPNN is divided into several smaller sub-networks, which are trained separately to improve computational efficiency. While dividing the BPNN into  $n$  sub-networks, it is also necessary to divide the training samples into  $n$  groups of sub-samples. It is also necessary to compare the current training sample with the previous training sample to calculate the corresponding group's mean square error sum. Finally, according to the comparison result, whether to input the data into BPNN for operation is decided. The improved BPNN eliminates the need for repeated calculations, which significantly

shortens the calculation time for large-scale neural networks and improves weight adjustment efficiency.

The improved BPNN is adopted to model the unknown parameters of the robot, and a dynamic controller that meets the real-time requirements of mobile robots is designed. First, the kinematic equation of the mobile robot is expressed as:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (11)$$

In Equation (11),  $(x, y)$  is the actual position of the mobile robot,  $\theta$  is the azimuth angle, and both the linear velocity  $v$  and the angular velocity  $\omega$  are control inputs in the kinematic model. Robot trajectory tracking is to track the target robot with pose  $q_r = [x_r, y_r, \theta_r]^T$  and velocity  $\dot{q}_r = [v_r, \omega_r]^T$ . The tracking error of the mobile robot is expressed as:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = T_e (q_r - q) \quad (12)$$

The error change rate of the mobile robot can be expressed as:

$$\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - v + v_r \cos e_3 \\ -\omega e_2 + v_r \sin e_3 \\ \omega_r - \omega \end{bmatrix} \quad (13)$$

The improved BPNN is applied to robot dynamics control. If a vector  $P(\cdot)$  is a variable function, a static neural network is utilized for modeling, and the following equation will be obtained:

$$P(\cdot) = [\{W_P\}^T \cdot \{\xi_P(\cdot)\}] + E_P(\cdot) \quad (14)$$

In Equation (14),  $\{W_P\}^T$  and  $\{\xi_P(\cdot)\}$  are OpenGL vectors, each element is a model error vector, and  $(\cdot)$  represents a general vector or matrix.

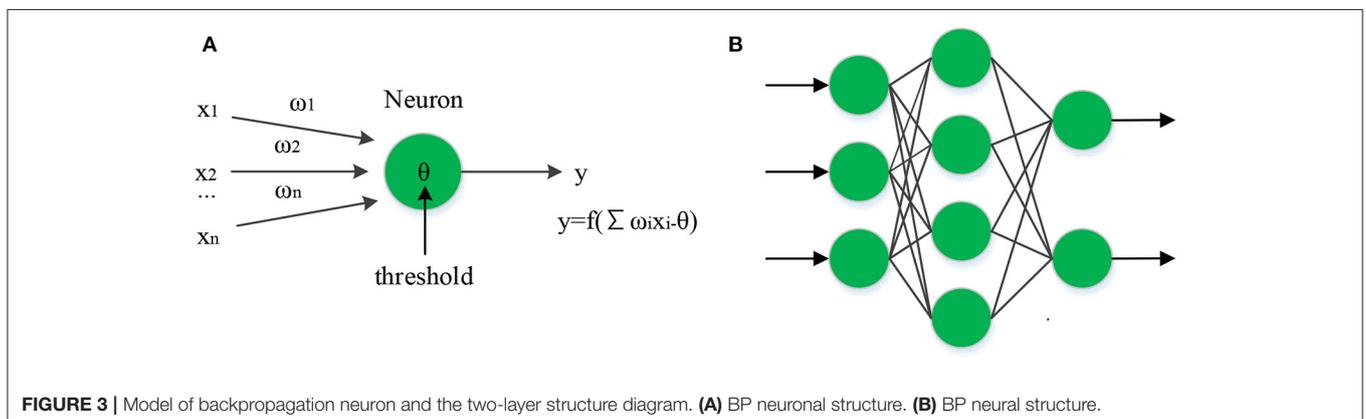


FIGURE 3 | Model of backpropagation neuron and the two-layer structure diagram. (A) BP neuronal structure. (B) BP neural structure.

## Robot Trajectory Tracking Based on the Fuzzy Algorithm Combining Neural Networks

In fuzzy systems, the design of fuzzy sets, membership functions, and fuzzy rules are based on empirical knowledge. This analysis method has a lot of subjectivity (Lu et al., 2018). Hence, the learning mechanism is introduced into the fuzzy system to modify and improve the membership function and fuzzy rules through continuous learning. The connection between the fuzzy system and the fuzzy neural network shows that the fuzzy neural network is essentially the realization of the fuzzy system. The difference between the two reveals that the fuzzy neural network has the characteristics of the neural network. Introducing the learning ability of the neural network into the fuzzy system and representing the fuzzy processing, fuzzy reasoning, and precise calculation of the fuzzy system through a distributed neural network is an important way to realize the self-organization and self-learning of the fuzzy system (Amador-Angulo et al., 2016). In a fuzzy neural network, the input and output nodes are used to fuzzify the input and output signals of the system (Caraveo et al., 2017; Lagunes et al., 2019). The hidden nodes of this neural network express the membership function and fuzzy rules, and the parallel processing capability of the neural network makes the inference ability of the fuzzy system greatly improved. The fuzzy neural network combines fuzzy system and neural network. A fuzzy neural network is essentially a conventional neural network that assigns fuzzy input signals and fuzzy weights. Its learning algorithm is usually a typical neural network's learning algorithm or its extension.

According to the kinematic model of the non-holonomic mobile robot, the current pose of the robot can be obtained as long as  $u = [v \ \omega]^T$  is controlled. Assuming that the actual pose of the robot is  $p = [x \ y \ \theta]^T$ , the actual motion velocity is  $[v \ \omega]^T$ ; the reference pose is  $p_r = [x_r \ y_r \ \theta_r]^T$ , and the reference motion velocity is  $[v_r \ \omega_r]^T$ . Then the error vector between the actual pose and the reference pose is  $p_e = [x_e \ y_e \ \theta_e]^T$ . Essentially, trajectory tracking of a non-holonomic mobile robot is to find a bounded input for any initial pose and velocity error and make:

$$\lim_{t \rightarrow \infty} \left\| [x_e \ y_e \ \theta_e]^T \right\| = 0 \tag{15}$$

The error vector between the actual pose and the reference pose can be expressed as:

$$p_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}^T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{16}$$

Equation (16) is derived to obtain the differential equation of the mobile robot's tracking error, expressed as:

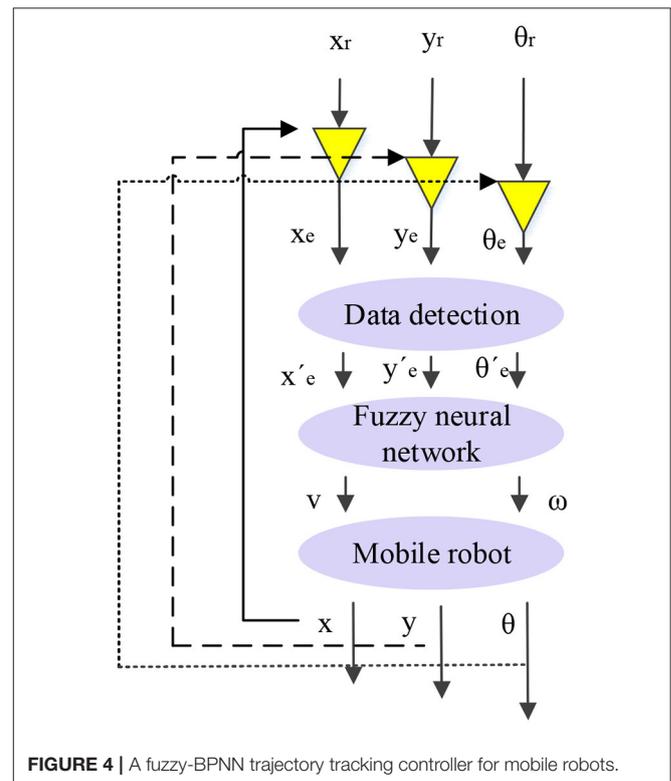
$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_r \cos \theta_e - v + y_e \omega \\ v_r \sin \theta_e - x_e \omega \\ \omega_r - \omega \end{bmatrix} \tag{17}$$

According to the above principles and derivations, a fuzzy-BPNN trajectory tracking controller for mobile robots is designed, as shown in **Figure 4**. It can control the linear velocity and angular velocity of mobile robots.

The actual trajectory tracking process will be disturbed by obstacles and other external environmental factors. Hence, the pose error will change significantly (Boujelben et al., 2017; Lu et al., 2017; Bencherif and Chouireb, 2019). Fuzzy logic can imitate the thinking way of the human brain and process systems with unknown models. Therefore, fuzzy logic can determine the location of obstacles while mobile robots are moving. Its core is to process the pose error data to avoid problems such as an increased number of fuzzy neural network's rules and the repeated change of weights. A data detection step is added, which uses the averaged error changes of the previous pose as a reference and compares it with the new pose error for decision-making. The data detection step can eliminate the error data with apparent mutations in the input. The fuzzy neural network's parameters do not need to be adjusted significantly. Therefore, the stability of the system and the overall calculation efficiency can be improved.

## IoT Multi-Sensor Data Fusion and Image Enhancement

A primary function of IoT technology is information perception. The object is connected to the network for information exchange through the installed sensors, electronic tags, and other sensing devices, realizing intelligent identification, positioning, and



**FIGURE 4** | A fuzzy-BPNN trajectory tracking controller for mobile robots.

supervision operations. In the process of information collection, the mobile robots' sensors summarize multiple sensors' data through data fusion, thereby reducing the transmission of redundant information and improving the stability and accuracy of the system (Jing et al., 2017).

IoT intelligent image enhancement establishes an image enhancement model through a wavelet conversion scale, which can adaptively adjust the window of different digital image frequencies to intelligently and adaptively enhance the IoT images. To improve the accuracy of environmental information collection, a mobile robot often uses an adaptive fusion algorithm to process the environmental information collected by its system (Yamashita et al., 2017; Ravi and Krishnan, 2018; Fan et al., 2019). Since the background of the original image collected by the infrared thermal imager and the monitored target robot are not notable, the details of the target image are difficult to identify, and the image feature cannot be clearly extracted. Hence, the image enhancement method is adopted to process the target image. The grayscale transformation method is adopted to enhance the image's contrast and improve its visual effects. Standard methods include (1) direct grayscale transformation; (2) transformation with the help of histogram; (3) transformation with a series of operations between images (such as addition and subtraction) (Chen et al., 2020). The direct grayscale transformation is the most commonly used and most convenient method. First, in the process of negating the gray, the image needs to be negated; that is, to reverse the gray value of the original image. Second, the image's contrast enhances the contrast of each part of the original image. Sometimes the dynamic range of the original image can exceed the allowable range of display devices. Therefore, if the original image is used directly, some details will be lost (Long et al., 2018; Singh et al., 2018). The solution is to compress the original image in grayscale. The principle of grayscale negation, contrast enhancement, and dynamic image compression is shown in **Figure 5**.

Moreover, image noise processing is vital in image enhancement, including impulse noise and Gaussian noise. Usually, image denoising separates the image into two kinds of noise; then, the median filter algorithm and the mean filter algorithm are used to eliminate these noises. The median filter algorithm arranges the to-be-processed pixels' gray values in the

neighborhood from large to small. Then it selects the median value to replace the to-be-processed pixel value in the template center. The mean filtering algorithm removes the sudden change by calculating the mean value of a central point and several surrounding points, thereby removing the noises.

### Simulations Experiments

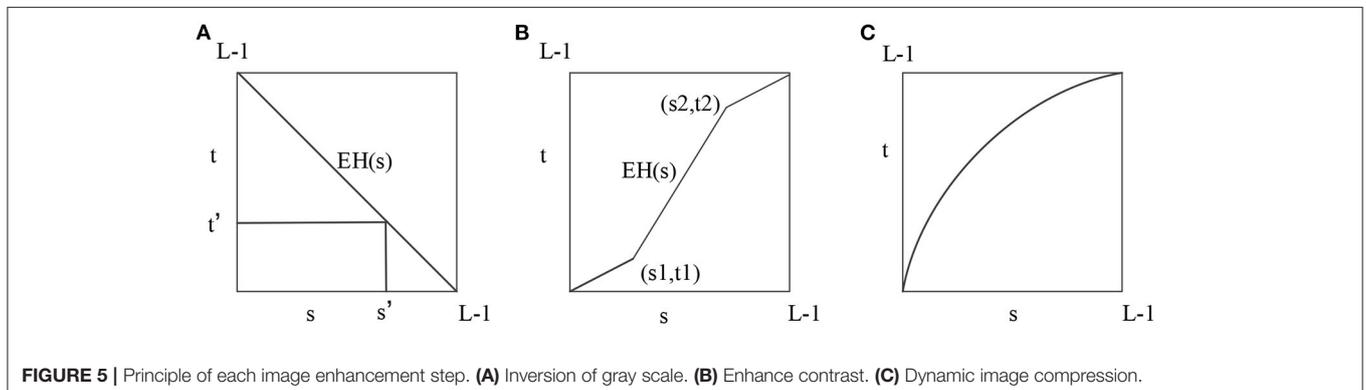
MATLAB is chosen as the simulation software to analyze the performances of the improved BPNN and the fuzzy-BPNN in robot trajectory tracking and control. MATLAB is one of the excellent science and technology application software. It has powerful calculation and visualization functions but is simple and easy to operate. In particular, the accompanying toolbox that supports more than 30 different fields has made it the basic tool and preferred platform for computer-aided design and analysis in various fields.

The performance differences between the two algorithms are compared. The reference linear velocity of the target robot is set to  $v_r = 3$  m/s, the reference angular velocity is set to  $\omega_r = 2$  rad/s, the initial pose is set to  $[x_r \ y_r \ \theta_r] = (-5, 0.25, 0)$ , and the initial pose error is set to  $[e_1 \ e_2 \ e_3]^T = [2.6 \ 2.4 \ \pi/2]^T$ . Ten groups of samples are taken for training. Each group of samples is divided into five groups of sub-samples equally. The mean square errors of the corresponding sub-samples of each group of samples (**Table 1**) are summed. The errors eventually converge to 0, the convergence speed is fast, and satisfactory results can be achieved.

## RESULTS AND DISCUSSION

### Trajectory Tracking Performance of Fuzzy-BPNN

The effects of traditional BPNN and improved BPNN in mobile robots' trajectory tracking are compared, and the results are shown in **Figures 6** and **7**. The improved BPNN algorithm has a slight improvement in tracking accuracy than the traditional BPNN; however, the difference between the two is not notable. On the contrary, the improved BPNN algorithm has a significant improvement in training speed, showing its performance advantages in processing huge samples in reality.



The fuzzy-BPNN algorithm is applied to track the mobile robot's trajectories, whose effects are compared with the improved BPNN. The simulation results are shown in **Figure 8**. The fuzzy-BPNN controller has significantly improved tracking speed and tracking accuracy compared with the improved BPNN, proving its effectiveness.

### Non-holonomic Mobile Robot's Trajectory Tracking and Control Results

The designed fuzzy-BPNN is integrated with the sliding mode trajectory tracking controller to solve the trajectory tracking and control problems of non-holonomic mobile robots. The tracking and control of typical circular and curved trajectories are simulated. **Figures 9** and **10** show the tracking effects and pose error changes in circular trajectory tracking, and **Figures 11** and **12** demonstrate the tracking effects and pose error changes in curve trajectory tracking. The designed algorithm integrating the sliding mode trajectory tracking controller shows

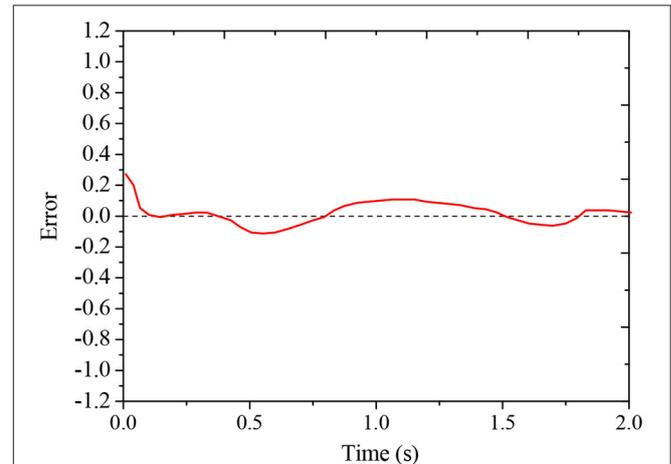
**TABLE 1** | The sum of the mean square errors of the sub-samples of each training sample.

Training samples	(1)	(2)	(3)	(4)	(5)
Group 1	0	0	0	0	0
Group 2	3.4	53.5	6.6	12.3	78.9
Group 3	11.3	5.6	5.3	4.3	53.5
Group 4	30.4	11.2	54.5	45.7	6.6
Group 5	6.9	22.3	34.1	23.6	5.9
Group 6	12.5	91.3	1.8	66.5	45.6
Group 7	34.5	11.7	7.2	4.8	12.3
Group 8	88.9	23.5	4.7	67.8	3.4
Group 9	2.2	23.4	44.6	32.4	11.1
Group 10	77.4	32.5	38.8	44.6	65.4

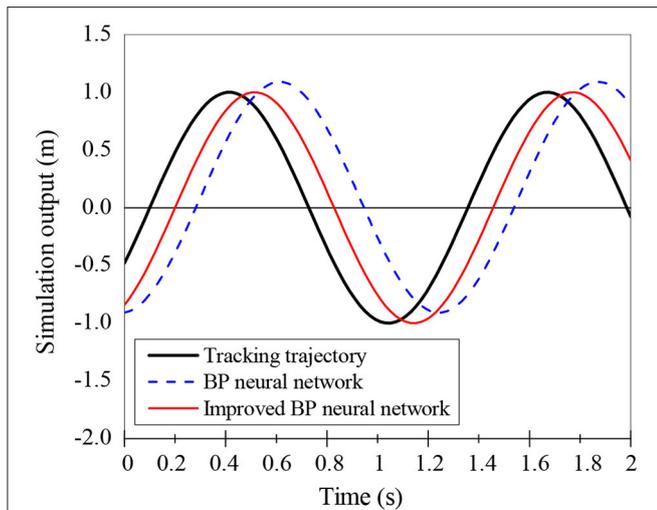
good trajectory tracking and control effects for these two different trajectories.

### CONCLUSION

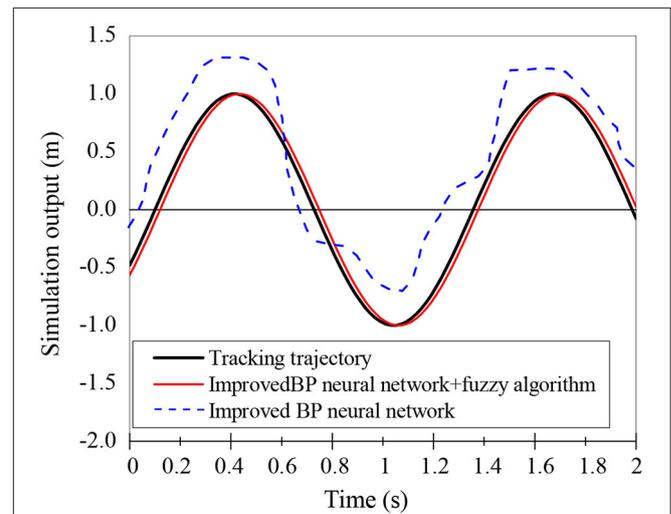
As an essential branch of robotics, mobile robots can continuously obtain the surrounding environment's status through sensors to make decisions and complete highly intelligent tasks. Robot intelligence is embodied in planning and executing an optimal path for mobile robots. Its core is trajectory tracking and control. Here, the mathematical kinematics models of non-holonomic mobile robots are analyzed first. Then the advantages of BPNN being widely used in pattern recognition and other problems are discussed. The traditional BPNN is improved, considering loads of data need to be trained in practical applications. The computational



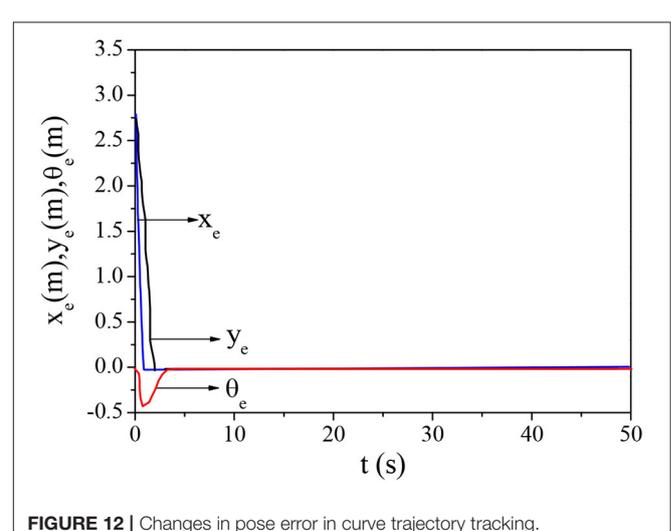
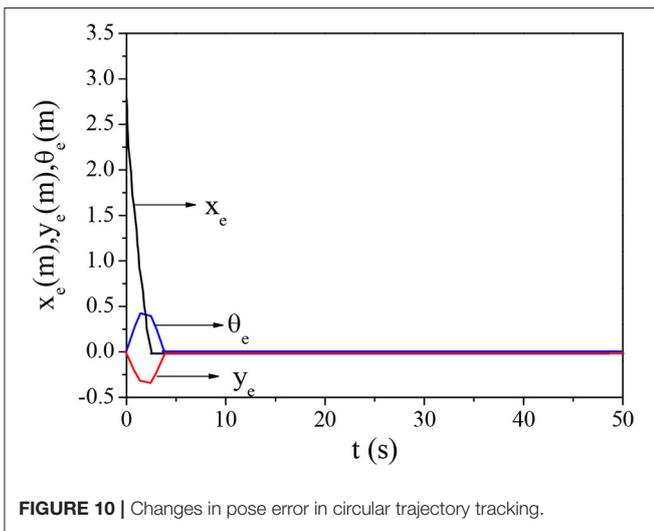
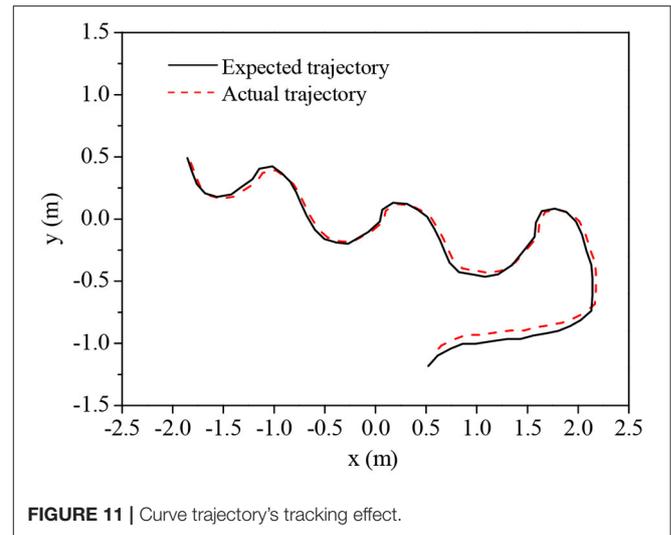
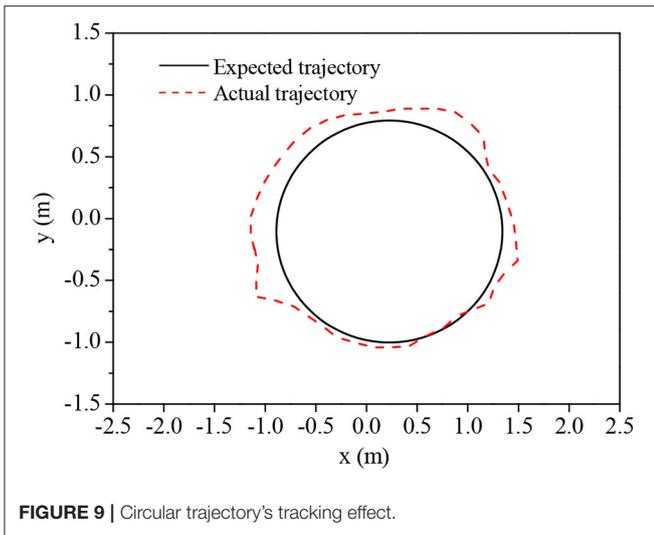
**FIGURE 7** | Trajectory tracking error curve of the improved BPNN algorithm.



**FIGURE 6** | Trajectory tracking effect of the improved BPNN algorithm.



**FIGURE 8** | Trajectory tracking effect of fuzzy-BPNN.



efficiency is improved by dividing BPNN into several smaller sub-networks for separate training. A fuzzy-BPNN tracking controller for mobile robot trajectories is designed to control the robot's linear velocity and angular velocity, in an effort to improve the accuracy of trajectory tracking. In addition, adaptive fusion algorithms and image enhancement techniques are used to process the environmental information collected by mobile robots to improve the accuracy of the environmental information collection.

A simulation experiment is run in the MATLAB environment to analyze the performances of the improved BPNN and the fuzzy-BPNN in robot trajectory tracking and control. Compared with traditional BPNN, the improved BPNN algorithm has a significant improvement in training speed, which has better application value for the large sample problems in reality. The fuzzy-BPNN controller has notably improved the tracking speed and accuracy compared with the improved BPNN algorithm.

The designed fuzzy-BPNN algorithm is integrated with the sliding mode trajectory tracking controller, whose performances in tracking and controlling circular and curve trajectories are simulated. The designed fuzzy BPNN algorithm is integrated with the sliding mode trajectory tracking controller. The circular and curved trajectories are simulated, and both show good trajectory tracking and control effects. However, the external environment is complicated and changeable; maintaining the stability of the mobile robot controller will be the principal direction of the following research.

### DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

## REFERENCES

- Alshakarchi, D. A. K., and Al-Maliky, F. T. (2018). Lower Limb Rehabilitation Robotics System Based on BP Neural Network. *Int. J. Appl. Eng. Res.* 13, 5889–5893.
- Amador-Angulo, L., Mendoza, O., Castro, J. R., Rodríguez-Díaz, A., Melin, P., and Castillo, O. (2016). Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *Sensors* 16:1458. doi: 10.3390/s16091458
- Bencherif, A., and Chouireb, F. (2019). A recurrent TSK interval type-2 fuzzy neural networks control with online structure and parameter learning for mobile robot trajectory tracking. *Appl. Intell.* 49, 3881–3893. doi: 10.1007/s10489-019-01439-y
- Boujelben, M. Á., Rekić, C., and Derbel, N. (2017). A reactive approach for mobile robot navigation in static and dynamic environment using fuzzy logic control. *Int. J. Model. Identif. Control* 27, 293–302. doi: 10.1504/IJMIC.2017.084722
- Caraveo, C., Valdez, F., and Castillo, O. (2017). A new meta-heuristics of optimization with dynamic adaptation of parameters using type-2 fuzzy logic for trajectory control of a mobile robot. *Algorithms* 10:85. doi: 10.3390/a10030085
- Chen, J., Yang, X., Lu, L., Li, Q., Li, Z., and Wu, W. (2020). A novel infrared image enhancement based on correlation measurement of visible image for urban traffic surveillance systems. *J. Intell. Transp. Syst.* 24, 290–303. doi: 10.1080/15472450.2019.1642753
- Chu, X., Peng, Z., Wen, G., and Rahmani, A. (2018). Distributed fixed-time formation tracking of multi-robot systems with nonholonomic constraints. *Neurocomputing* 313, 167–174. doi: 10.1016/j.neucom.2018.06.044
- Fan, Y. C., Liu, Y. C., and Chen, H. Y. (2019). Image Enhancement Using Back-propagation Neural Network Based Histogram Equalization for Image Sensor. *Sensor Lett.* 17, 124–131. doi: 10.1166/sl.2019.4055
- Gutiérrez-Giles, A., Ruggiero, F., Lippiello, V., and et al., (2018). Nonprehensile manipulation of an underactuated mechanical system with second-order nonholonomic constraints: the robotic hula-hoop. *IEEE Robot. Autom. Lett.* 3, 1136–1143. doi: 10.1109/LRA.2018.2792403
- Jing, L., Wang, T., Zhao, M., and Wang, P. (2017). An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox. *Sensors* 17:414. doi: 10.3390/s17020414
- Lagunes, M. L., Castillo, O., Soria, J., Garcia-Valdez, M., and Valdez, F. (2019). Optimization of granulation for fuzzy controllers of autonomous mobile robots using the Firefly Algorithm. *Granul. Comput.* 4, 185–195. doi: 10.1007/s41066-018-0121-6
- Long, M., Li, Z., Xie, X., Li, G., and Wang, Z. (2018). Adaptive image enhancement based on guide image and fraction-power transformation for wireless capsule endoscopy. *IEEE Trans. Biomed. Circuits Syst.* 12, 993–1003. doi: 10.1109/TBCAS.2018.2869530
- Lu, X., Zhao, Y., and Liu, M. (2018). Self-learning interval type-2 fuzzy neural network controllers for trajectory control of a Delta parallel robot. *Neurocomputing* 283, 107–119. doi: 10.1016/j.neucom.2017.12.043
- Lu, X. G., Liu, M., and Liu, J. X. (2017). Design and optimization of interval type-2 fuzzy logic controller for delta parallel robot trajectory control. *Int. J. Fuzzy Syst.* 19, 190–206. doi: 10.1007/s40815-015-0131-3
- Lv, Z., Hu, B., and Lv, H. (2019). Infrastructure monitoring and operation for smart cities based on IoT system. *IEEE Trans. Industr. Inform.* 16, 1957–1962. doi: 10.1109/TII.2019.2913535

## FUNDING

This work was supported by the Fundamental Research Foundation for Universities of Heilongjiang Province (No. LGYC2018JC050).

- Lv, Z., and Xiu, W. (2019). Interaction of edge-cloud computing based on SDN and NFV for next generation IoT. *IEEE Internet Things J.* 7, 5706–5712. doi: 10.1109/JIOT.2019.2942719
- Ma, X., Wang, R., Zhang, Y., Jiang, C., and Abbas, H. (2020). A name disambiguation module for intelligent robotic consultant in industrial internet of things. *Mech. Syst. Signal Process.* 136:106413. doi: 10.1016/j.ymssp.2019.106413
- Marques, G., Pires, I. M., Miranda, N., and Pitarma, R. (2019). Air quality monitoring using assistive robots for ambient assisted living and enhanced living environments through internet of things. *Electronics* 8:1375. doi: 10.3390/electronics8121375
- Michie, C., Andonovic, I., Davison, C., Hamilton, A., Tachtatzis, C., Jonsson, N., et al. (2020). The internet of things enhancing animal welfare and farm operational efficiency. *J. Dairy Res.* 87, 20–27. doi: 10.1017/S0022029920000680
- Özdemir, V. (2019). The big picture on the “AI turn” for digital health: the internet of things and cyber-physical systems. *OMICS J. Integr. Biol.* 23, 308–311. doi: 10.1089/omi.2019.0069
- Ravi, P., and Krishnan, J. (2018). Image enhancement with medical image fusion using multiresolution discrete cosine transform. *Mater. Today Proc.* 5, 1936–1942. doi: 10.1016/j.matpr.2017.11.296
- Rehman, M. H., Yaqoob, I., Salah, K., Imran, M., Jayaraman, P. P., and Perera, C. (2019). The role of big data analytics in industrial Internet of Things. *Future Gener. Comput. Syst.* 99, 247–259. doi: 10.1016/j.future.2019.04.020
- Singh, H., Kumar, A., Balyan, L. K., and Singh, G. K. (2018). Swarm intelligence optimized piecewise gamma corrected histogram equalization for dark image enhancement. *Comput. Electr. Eng.* 70, 462–475. doi: 10.1016/j.compeleceng.2017.06.029
- Singh, N. H., and Thongam, K. (2018). Mobile robot navigation using MLP-BP approaches in dynamic environments. *Arab. J. Sci. Eng.* 43, 8013–8028. doi: 10.1007/s13369-018-3267-2
- Singh, N. H., and Thongam, K. (2019). Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intell. Serv. Robot.* 12, 55–67. doi: 10.1007/s11370-018-0260-2
- Tinh, N., and Linh, L. E. (2018). Neural network-based adaptive tracking control for a nonholonomic wheeled mobile robot with unknown wheel slips, model uncertainties, and unknown bounded disturbances. *Turk. J. Electr. Eng. Comput. Sci.* 26, 378–392. doi: 10.3906/elk-1705-167
- Tu, X., Zhou, Y. F., Zhao, P., and Cheng, X. (2019). Modeling the static friction in a robot joint by genetically optimized BP neural network. *J. Intell. Robot. Syst.* 94, 29–41. doi: 10.1007/s10846-018-0796-6
- Wang, F., Chao, Z., Huang, L., Li, H., and Zhang, C. (2019). Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode. *Clust. Comput.* 22, 5799–5809. doi: 10.1007/s10586-017-1538-4
- Yamashita, H., Sugimura, D., and Hamamoto, T. (2017). Low-light color image enhancement via iterative noise reduction using RGB/NIR sensor. *J. Electron. Imaging* 26:043017. doi: 10.1117/1.JEI.26.4.043017
- Yang, L., and Pan, S. (2018). A sliding mode control method for trajectory tracking control of wheeled mobile robot. *J. Phys. Conf.* 1074:012059. doi: 10.1088/1742-6596/1074/1/012059
- Yi, Z., Li, G., Chen, S., Xie, W., and Xu, B. (2019). A navigation method for mobile robots using interval type-2 fuzzy neural network fitting

Q-learning in unknown environments. *J. Intell. Fuzzy Syst.* 37, 1113–1121. doi: 10.3233/JIFS-182560

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

*Copyright © 2021 Zhao, Wang, Fan and Li. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.*