The Journal of Supercomputing (2022) 78:8678–8707
https://doi.org/10.1007/s11227-021-04160-1

Kinematic and dynamic control model of wheeled mobile robot under internet of things and neural network

Qiang Liu[1] · Qun Cong[1]

Accepted: 19 October 2021 / Published online: 12 January 2022

Abstract

This study aims to solve the issues of nonlinearity, non-integrity constraints, under-actuated systems in mobile robots. The wheeled robot is selected as the research object, and a kinematic and dynamic control model based on Internet of Things (IoT) and neural network is proposed. With the help of IoT sensors, the proposed model can realize effective control of the mobile robot under the premise of ensuring safety using the model tracking scheme and the radial basis function adaptive control algorithm. The results show that the robot can be controlled effectively to break the speed and acceleration constraints using the strategy based on the model predictive control, thus realizing smooth movement under the premise of safety. The self-adapting algorithm based on the IoT and neural network shows notable advantages in parameter uncertainty and roller skidding well. The proposed model algorithm shows a fast convergence rate of about 2 s, which has effectively improved performances in trajectory tracking and robustness of the wheeled mobile robot, and can solve the difficulties of wheeled mobile robots in practical applications, showing reliable reference value for algorithm research in this field.

Keywords Internet of Things · Neural network · Wheeled mobile robots · Kinematics · Dynamics model · Trajectory tracking

1 Introduction

With the continuous in-depth research of the Internet of Things (IoT), artificial intelligence (AI), and big data algorithms, more and more emerging technology products appear and become an indispensable part of people's life and production [1]. As a

✉ Qun Cong
congqun@sspu.edu.cn

Qiang Liu
liuqiang@sspu.edu.cn

[1] Shanghai Polytechnic University, Shanghai 201209, China

product of the industrialization and informatization of IoT, mobile robots integrate the contents and technologies of multiple disciplines, such as biology, informatics, electric power, and structural science, and are an important manifestation of the development of human wisdom [2]. Mobile robots have been widely used in various fields of medicine, space exploration, design and construction, industrial production, and teaching practice, bringing great convenience to people's daily lives [3]. In the Chernobyl incident, massive nuclear radiations threatened people's lives, and the cleaning of radiation materials could only be achieved by mobile robots [4]. In the 2019 Corona Virus Disease (COVID-19) epidemic, mobile robots are adopted to perform blood collection and testing, which can prevent spread of the virus [5]. In the Mars exploration mission of China, mobile robots are adopted for scientific experiments and samples collection [6]. Obviously, mobile robots are playing an increasingly important role in our lives. Wheeled mobile robots are typical of mobile robots. Because of their simple structure and flexible control, they are also used in various fields of civil and military affairs. Although currently wheeled mobile robots are widely used, it also has some drawbacks in its application due to specific restrictions [7]. Path planning and trajectory tracking control are basic difficulties. Reasonable path planning can effectively save costs, improve production efficiency, and optimize resource allocation. A better trajectory tracking method can ensure the stability of the system and ensure the efficiency of task completion [8]. Therefore, studying the kinematics and dynamics control of wheeled mobile robots has important research value for the practical application of mobile robots.

IoT is an intelligent network that realizes the interconnection of everything. It connects all items to the Internet through radio frequency identification and other information sensing equipment, which can realize intelligent identification and provide safe, controllable, and even personalized real-time online management and service functions [9]. The main feature of the IoT is that it can obtain various information of the physical world through radio frequency identification, sensors, etc., and combine the Internet, mobile communication network, and other networks for information transmission and interaction. Moreover, it can adopt intelligent computing technology to analyze and process information, thereby improving the perception to the material world and achieving intelligent decision-making and control [10]. The emergence of the IoT has broken the barriers between the physical world and the information world and successfully integrated reinforced concrete, machines, and the Internet to make them a new whole and better serve human production and life. Research on mobile robots focuses on the improvement of its own performance, but it is difficult to continue to improve the perception and intelligence of mobile robots when they reach a certain height limited by electronic technology such as chips and other factors [10]. The research on the IoT focuses on the intelligent recognition, object positioning, target monitoring and tracking, and management of sensor networks. However, there is a lack of sufficient means for the development of intelligent mobility and autonomous control, which severely restricts the application expansion of IoT. Therefore, studying the IoT technology and applying it to all aspects of the field have become the trend of the IoT development.

Neural network is a mathematical model that imitates the movement mode and control system of living organisms, transforms its actual problems into corresponding mathematical models, and uses corresponding functions to realize the transmission and control of information [11]. Many scholars have reported on the application of neural networks in mobile robots. Singh and Thongam [13] used artificial neural networks (ANN) to reasonably control the mobile robots and generate collision-free, close-to-optimal paths and speed; and the effectiveness of the algorithm was proved by the simulation experiments [12]. Huang et al. [14] proposed a control scheme based on the robust neural network and proved that the algorithm can handle the tracking and stabilization of wheeled mobile robots affected by external interference and input saturation at the same time [13]. Hu et al. [15] proposed an optimized nonlinear model predictive control (MPC) algorithm based on the neurodynamic to track the movement trajectory of the mobile robot and proved that the method can accurately track the trajectory based on the solver and can maintain high robustness [14]. Lobo et al. (2020) proposed a simple spiking neural networks (SNN) algorithm, which has the Hebbian rules in the form of spike-time-dependent plasticity. Moreover, it used the spatial characteristics of spike-timing-dependent plasticity (STDP) to achieve associative learning; and it was proved that when the environment changes, the robot has the ability to learn again [15]. Therefore, the neural network can effectively solve the difficulties in mobile robot path planning and trajectory tracking.

The innovations of this research lie in solving the predictive tracking control of wheeled mobile robots under the constraint of speed saturation, proposing an adaptive tracking control algorithm for wheeled mobile robots based on radial basis function (RBF) neural network, and realizing the automatic disturbance rejection tracking control of the wheeled mobile robot against the conditions of sliding and skidding. Therefore, it provides important reference value for the practical application of wheeled mobile robots. The objective of this study is discovering the current difficulties in the practical application of wheeled mobile robots and to propose scientific experimental methods to solve them. For speed saturation constraints, roller skidding, and parameter uncertainty, a kinematics and dynamics model of wheeled mobile robots based on the neural network is proposed, and the performance of which is further verified through actual simulation experiments. This research is of imperative value in promoting the practical application of wheeled mobile robots.

A dynamic model of a mobile robot is established using RBF neural network, which doesn't require to know all the dynamic parameters of the mobile robot compared with the conventional torque control method. When the parameters of the controlled object change suddenly, the control scheme can adjust its own grid structure and parameters to adapt to the change, thereby improving the trajectory tracking control performance of the mobile robot and overcoming the model uncertainty and external interference. However, there are some limitations. In many special occasions, the accuracy of the neural network model will decrease after the system parameters are changed. When the grid and parameters are adjusted online, whether the grid structure is streamlined is also the actual situation that need to be considered.

2 Theoretical basis

2.1 Control of mobile robots

The mobile robots adopt the nonlinear time-varying systems and have complex characteristics such as internal parameter perturbation, model uncertainty, and external time-varying interference. Therefore, the early classical control theory and linear system theory are quite mature and applied widely. However, it is difficult to meet the control requirements of mobile robots, so it is necessary to combine other control methods to cope with the complexity of mobile robots, so as to achieve precise control of mobile robots. The common control methods are as follows. I. The central idea of self-adaptive control is that some important parameters of the controller can be adjusted by itself during the dynamic change of the system. If a specific method is adopted to enable some unknown or real-time changing parameters of the system to be estimated during system dynamic changes, then the unknown parameters in the adaptive controller are directly replaced by estimators, so that the system can maintain good performance even when the model has unknown structural parameters [16]. II. Robust control is usually used for control systems with unstable models that frequently suffer from parameter perturbations. Its unique feature is that a pre-designed controller with constant structure and parameters makes it feasible for the system to respond to parameter changes and disturbances. It is as insensitive as possible, that is, the system has considerable immunity and performance [17]. III. ANNs have excellent recognition and self-learning capabilities for functions and unknown features of the outside world. In recent years, they have frequently appeared in the fields of control and signal processing. The adaptive neural network control method formed by fusing the advantages of adaptive control and neural network can effectively learn and approximate the unknown part of the system model. The output of the neural network is added to the controller to form feedback, thereby reducing the adverse effects of system model uncertainty on the transient or steady-state performance of the control system and improving the robustness of the system [18]. IV. A transmission strategy without a fixed period instead of the traditional fixed period transmission is proposed to balance control accuracy and network burden simultaneously, which is called event-triggered control. In event trigger control, whether the signal at a certain sampling moment is transmitted or not is determined by an event trigger condition, so the transmission cycle is not fixed [19].

2.2 Movement control of IoT

To achieve the organic integration and complementary advantages of mobile robots and the IoT, the execution capabilities of mobile robot on the IoT require to be improved, and the IoT should provide necessary information support for mobile robots. Of which, the most imperative point is providing positioning services. In this study, interaction with the surrounding environment is realized through flexible installation of load sensors. AVR meg16 single-chip microcomputer is used as the main control chip to reduce the degree of coupling with the robot control subsystem in the sensor subsystem. The sensor can collect environmental information including distance and temperature and then transmit them to the robot control subsystem through the RS232 serial port. The specific module is shown in Fig. 1, including the following three parts.

Ultrasonic ranging module is responsible for avoiding the robot hitting the wall when the robot encounters an obstacle after it is too late to control it. It aimed to prevent the hardware from being damaged by collision and to delay the service life of the robot. The user can change the detection distance by modifying its frequency after the ultrasonic ranging is turned on. The single-chip microcomputer immediately sends the information to the upper system when an obstacle is found within the specified distance, which will prompt the user to turn, or otherwise the robot will stop.

The smoke detection module is featured with relay control output, level detection signal output for controlling the alarm effectively, direct current (DC) 9 V battery as power supply, high induction sensitivity, and extremely low false alarm rate. When the indicator light of the smoke detection sensor starts to flash, the relay will be closed, which indicates that the smoke concentration has exceeded the normal value. The indicator light starts to flash in about three seconds and the relay is closed. When the AVR Mega16 microcontroller reads a low level from the I/O port, an abnormal interrupt will be triggered, a buzzer alarm is issued, and the dangerous information is transmitted to the ARM6410 main control chip through the RS232 interface.

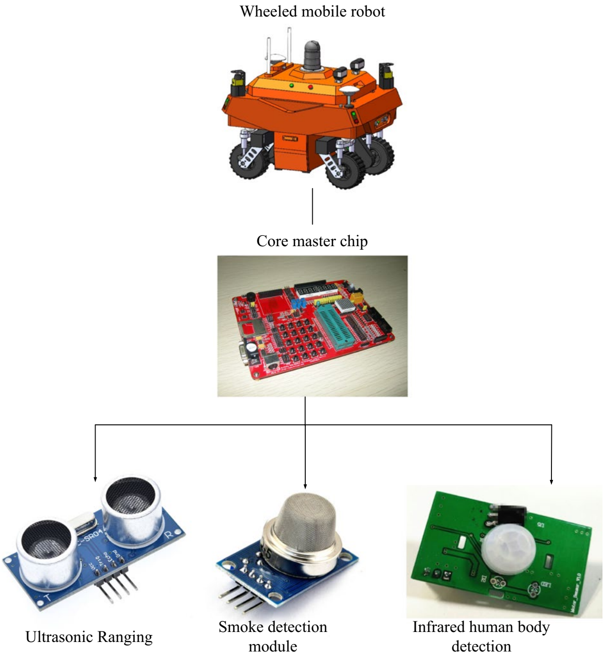


Fig. 1 Mobile robot control system based on IoT

There is another model, infrared human body induction detection module. Human body sensing is a new technology that uses infrared and pyroelectric principles to sense temperature change information. It can automatically and accurately identify, detect, and sense information about thermal changes including the human body. It does not rely on light, so it can work stably and reliably even in dark. The infrared human body sensing detection module is used to detect the function of human activities, and such function is adopted in disaster relief scenarios. For example, the robot can provide reference information for searching and rescuing trapped persons in some dangerous environments.

2.3 Kinematics and dynamics model

For wheeled mobile robots, the most important thing is establishing kinematics and dynamics models. In Fig. 2(A), the center line of the left and right wheels of the robot is taken as the center point *M*, which is deemed as the reference point [20], and then, the kinematics model established can be expressed as Eq. (1).

A two-wheel differential robot consists of two independently driven wheels of radius *r*, rotating on the same axis, with one or more casters, spherical casters, or low-friction sliders to keep the robot level.

According to Fig. 2, the motion equation of the trolley is Eq. (1–3).

$$x = v \cdot \cos\theta \tag{1}$$

$$y = v \cdot \sin\theta \tag{2}$$

In the above equations, *v* is the linear velocity of the trolley in all directions. Therefore, there is the following functions.

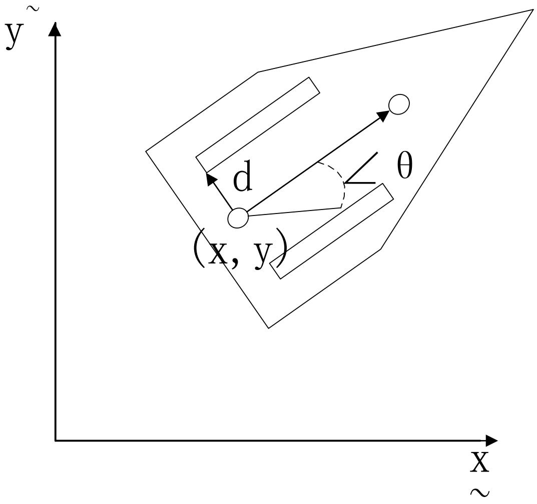


Fig. 2 Motion vector diagram of a wheeled robot

$$\begin{cases} x\prime = v\cos\xi\cos\theta \\ y\prime = v\cos\xi\sin\theta \\ \theta = \omega \end{cases} \tag{3}$$

$$\omega = \frac{v\sin\xi}{d} \tag{4}$$

In the above equations, ξ refers to the front wheel deflection angle of the robot, which can be positive (left deflection) or negative (right deflection) based on the reference point, θ is the heading angle of the robot, ω is the heading angular velocity of the robot, d refers to the distance from the center of the front wheel to the reference point M, and v represents the speed of the front wheel drive of the robot. The above equation can be discretized as the following equation.

$$\begin{cases} x(k+1) = x(k) + v(k)T_s\cos\xi(k)\cos\theta(k) \\ y(k+1) = y(k) + v(k)T_s\cos\xi(k)\sin\theta(k) \\ \theta(k+1) = \theta(k) + \omega(k)T_s \end{cases} \tag{5}$$

In Eq. (3), T_s represents the sampling period.

$$\omega(k) = v(k)\sin\xi(k)/d \tag{6}$$

The driving force of the front wheels is set to f_1, then Eq. (5) is acquired.

$$f_1\cos\xi = M\frac{\mathrm{d}}{\mathrm{d}t}(v\cos\xi) = M(v\cos\xi - v\xi\sin\xi) \tag{7}$$

$$f_1\sin\xi = 1\cdot\omega \tag{8}$$

$$v = e\omega_1 \tag{9}$$

$$\xi{=}\omega_1 \tag{10}$$

The dynamic equation of the direct current motor for front wheel drive is given as follows.

$$(k_2j_2)\omega_1 + \left(k_2B_1 + B_2 + \frac{k_m^2}{R_a}\right)\omega_1 = \frac{k_mk}{R_a}V_{a1} - rf_1 \tag{11}$$

Eq. (10) is acquired by combining above equations.

$$\left[k^2j_1 + j_2Mr^2\right]v + \left(k^2B_1 + B_2 + \frac{k_m^2}{R_a} - Mr^2\omega_2\tan\xi\right)v = \frac{k_mkr}{R_a}V_{a1} \tag{12}$$

The dynamic equation of the direct current motor in the forward direction is expressed as follows.

$$(k^2 j_1 + j_3)\omega_2 + \left(k^2 B_1 + B_2 + \frac{k_m^2}{R_a} \right)\omega_2 = \frac{k_m k}{R_a} V_{a2} \tag{13}$$

In Eq. (13), ξ is the front wheel deflection angle of the robot, M is the mass of the robot, v is the driving speed of the front wheels of the robot, d is the distance between the center of the front wheel of the robot and the reference point C of the robot, j_1 is the moment of inertia of the motor, and j_2 refers to the moment of inertia of the front wheel to the rolling axis. j_3 is the moment of inertia of the front wheel about the steering axis. In addition, B_1 is the viscous friction coefficient of the motor shaft, B_2 is the viscous friction coefficient of the front wheel, k_m is the electromagnetic torque constant of motor, k is the gear ratio, V_{a1} and V_{a2} are the output voltages of the direct current motor. r is the radius of the front wheels, ω_1 is the angular velocity of the front wheel drive, ω_2 is the steering angular velocity of the front wheels, and R_a represents the armature resistance.

In Fig. 3B, the center line of the left and right wheels of the robot is taken as the center point M, which is determined as the reference point. It is assumed that the angular velocities of the left and right driving wheels are ω_1 and ω_2, respectively, and then, the established kinematics model [21] can be expressed as Eq. (12).

$$\begin{cases} x = v\cos\theta \\ y = v\sin\theta \\ \theta = \omega \end{cases} \tag{14}$$

$$v{=}\frac{r}{2}(\omega_1 + \omega_2) \tag{15}$$

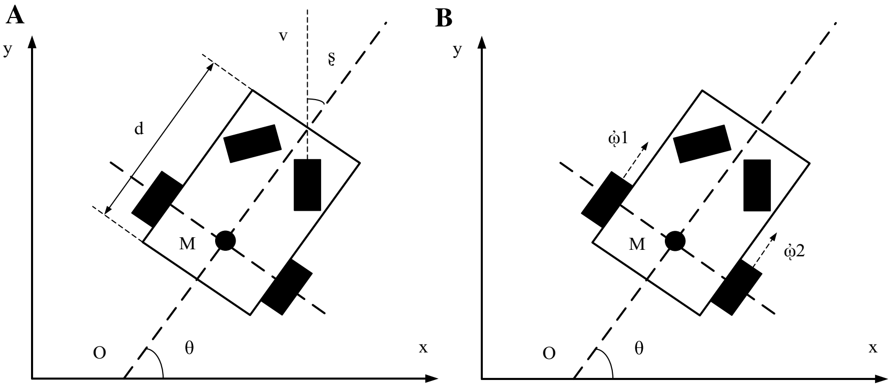


Fig. 3 The kinematics and dynamics model of wheeled mobile robots

$$\omega = \frac{r}{W}(\omega_2 - \omega_l) \tag{16}$$

After discretizing the above equations, Eq. (16) is acquired.

$$\begin{cases} x(k+1) = x(k) + v(k)T_s \cos\theta(k) \\ y(k+1) = y(k) + v(k)T_s \sin\theta(k) \\ \theta(k+1) = \theta(k) + \omega(k)T_s \end{cases} \tag{17}$$

In Eq. (15), T_s represents the sampling period.

$$v(k) = \frac{r}{2}[\omega_1(k) + \omega_2(k)] \tag{18}$$

$$\omega(k) = \frac{r}{W}[\omega_2(k) - \omega_l(k)] \tag{19}$$

In the above two equations, W is the distance between the two wheels, and ω refers to the heading angular velocity of the robot. The driving forces of the two wheels are assumed as f_1 and f_2, the dynamics model can be expressed as follows:

$$f_1 + f_2 = Mv \tag{20}$$

$$\frac{W}{2}(f_2 - f_l) = Iw \tag{21}$$

In Eq. (19) above, I refers to the moment of inertia of the robot, and the dynamic equations on the left and right axles are given as follows:

$$(k^2 j_1 + j_2)\omega_1 + \left(k^2 B_1 + B_2 + \frac{k_m^2}{R_a}\right)\omega_1 = \frac{k_m k}{R_a}V_{a1} - rf_1 \tag{22}$$

$$(k^2 j_1 + j_2)\omega_2 + \left(k^2 B_1 + B_2 + \frac{k_m^2}{R_a}\right)\omega_2 = \frac{k_m k}{R_a}V_{a2} - rf_2 \tag{23}$$

Combining the above equations can get the following two equations:

$$\left[k^2 j_1 + j_2 + \frac{M}{2}r^2\right]v + \left(k^2 B_1 + B_2 + \frac{k_m^2 k^2}{R_a}\right)v = \frac{k_m kr}{R_a}\frac{V_{a1} + V_{a2}}{2} \tag{24}$$

$$\left[k^2 j_1 + j_2 + \frac{2I}{W^2}r^2\right]\omega + \left(k^2 B_1 + B_2 + \frac{k_m^2 k^2}{R_a}\right)\omega = \frac{k_m kr}{R_a}\frac{V_{a2} - V_{a1}}{W} \tag{25}$$

3 3. Research methods

3.1 3.1 Model under speed saturation constraints

In actual work, wheeled mobile robots are subject to many constraints. For example, the starting acceleration should not be too large, the angular velocity during turning should not be too large, and the linear velocity should be limited. In terms of performance, the controller is required to meet the real-time and accuracy, that is, the robot can accurately reach the desired position in a short period of time and consider the complex and changeable environment where the robot is located [22]. The proposed MPC method uses a single-layer recurrent neural network in a general projection network to iteratively solve the formulaic quadratic programming problem when external interference is not explicitly considered. The two-layer recurrent neural network is adopted to iteratively solve the reconstructed minimum–maximum optimization if disturbances are considered [23]. MPC solves a finite-time open-loop optimization online at each time of adoption based on the current measurement information obtained, and the first element of the obtained control sequence is applied to the controlled object. The above process should be repeated at the next sampling moment: The new measured value is undertaken as the initial condition for predicting the future dynamics of the system at this time, so as to loop and solve the optimization.

Actuator saturation is a common phenomenon in actual engineering systems. Most actuators will experience saturation, which is not easy to avoid. If the amount of input information of the control system actuator reaches a certain limit, it will enter a saturated state. If the input amount is increased, it won't have effect on the output of the actuator, and it will also cause the dynamic performance of the control system to decrease, or make the closed loop system as a whole Deterioration of dynamic performance [24]. From the perspective of theoretical research and practical application, saturation phenomenon makes the system control more complicated. Therefore, it is of great significance to study the stability of the constraint system. If it is undertaken as a predictive model, then the optimal performance of the wheeled mobile robots at all times can be defined as the following equation.

$$\vartheta(k) = \sum\nolimits_{j=1}^{N} \left[q^T(k+j)k + Qq(k+j)k\right] \tag{26}$$

In the equation above, N is the prediction time domain, j refers to the inertia moment, q represents the error, and Q and R are the weight matrixes. The optimal control sequence that can minimize the performance index Φ(k) is obtained through each sampling moment k. The trajectory tracking control structure of wheeled mobile robots based on model prediction is shown in Fig. 4 [25]. The tracking error and control variable error of wheeled mobile robots in the prediction time domain N are defined as follows.

$$\vartheta(k) = \left[q^T(k+1)k + Qq(k+1)k\right] \tag{27}$$

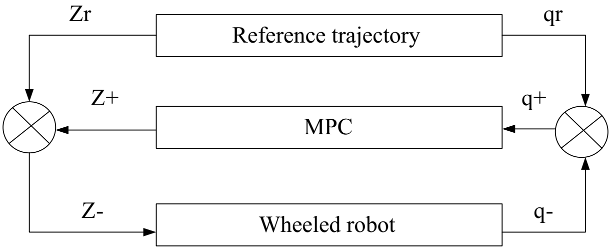


Fig. 4 Structure diagram of trajectory tracking control based on model prediction

3.2 3.2 Neural network adaptability model

ANN, also known as the connected machine model, is produced based on modern neurology, biology, psychology, and other disciplines. It reflects the basic process of the biological nervous system processing external things and is simulating the human brain nerve tissue. The computing system developed on that is a network system composed of massive processing units through extensive interconnection, and the multi-layer forward BP network is currently the most widely used form of neural network.

RBF neural network is a novel and effective feedforward neural network, which owes the best approximation and global optimal performance. In addition, the training method is fast and easy, and there is no local optimal problem. It has the general advantages of neural networks, but it is not perfect either.

Compared with the algorithm used in this study, the advantages of the multi-layer antecedent BP grid algorithm include nonlinear mapping capability, self-learning and adaptive capability, generalization capability, and fault tolerance. However, many shortcomings are also exposed, such as local minimization, slow convergence speed of BP neural network algorithm, different choice of BP neural network structure, contradiction between application examples and network scale, contradiction between BP neural network prediction ability and training capacity, and BP neural network sample dependence.

The RBF network used for system modeling is generally divided into the following steps of selecting the appropriate learning sample, processing the learning sample data, determining the preposition and delay of the model, using the learning algorithm in front to complete offline learning of RBF network, verifying and testing the model, and realizing online correction of RBF network using online learning method.

RBF networks are used in the design of control systems, generally in the form of adaptive control, nonlinear compensation control, internal model control, and predictive control.

Wheeled mobile robots not only adopt the highly complex nonlinear system with multiple inputs and multiple outputs, under-drive, and strong coupling, but also are typical incomplete systems. Therefore, the wheeled mobile robots have uncertainties such as parameter perturbation and external interference. The uncertainty of wheeled mobile robots mainly includes parameter uncertainty and non-parameter uncertainty [26]. The parameter uncertainty is usually related to the model itself, such as the radius of the wheel, the distance between the wheels, the moment of inertia, and the distance from the center of mass to the geometric center. The non-parameter uncertainty is mainly caused by measurement noise, external interference, sampling time lag, rounding errors in calculations, other non-controlled object factors, and unmodeled dynamics (high-frequency unmodeled dynamics) [27]. Uncertainty affects the performance of the control tracker, and part of the control algorithm only satisfies the deterministic condition of the controlled object. Therefore, the trajectory tracking of wheeled mobile robots under uncertain conditions has always been a hot topic for scholars [28].

Neural networks can approximate arbitrary nonlinear functions and have strong self-learning capabilities. The RBF kernel is adopted in this study to approach the uncertainty in the dynamics model of wheeled mobile robots and realizes the indirect compensation for the uncertainty of wheeled mobile robots, thus realizing the accurate trajectory tracking of wheeled mobile robots [29]. In trajectory tracking, it only needs to know the mid-coordinate and azimuth of the pose q, which is re-defined as $q = \left[X \cdot Y^{\theta}\right]^{T}$, then the error equation for pose tracking of wheeled mobile robots in the local coordinate system is acquired as follows.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta\ \sin\theta 0 \\ \sin\theta\ \cos\theta 0 \\ 001 \end{bmatrix} \begin{bmatrix} x - x_1 \\ y - y_1 \\ \theta - \theta_1 \end{bmatrix} \tag{28}$$

After its motion equation is incorporated, the differential equation of the pose tracking error can be expressed as Eq. (27).

$$\begin{cases} x = yw + v_r \cos\theta - v \\ y = -xw + v_r \sin\theta \\ \theta = w_r - w \end{cases} \tag{29}$$

The principle of Eq. (29) is the same as Eq. (1), and the same principle is summarized in Eq. (1).

The kinematics virtual control law of the mobile robot is written as Eq. (28) below.

$$z = \begin{bmatrix} v_c \\ w_c \end{bmatrix} = \begin{bmatrix} k_1 x + vr\cos\theta \\ w_r + k_2 y v_r + k_3 v_r \sin\theta \end{bmatrix} \tag{30}$$

In the above Eq. (28), k_1, k_2, and k_3 are all design parameters and meet > 0, V_r is the auxiliary speed, and ω_r is the tracking speed.

The Lyapunov function is selected for design. Due to the model parameter uncertainty and unknown disturbance in the dynamic system modeling of wheeled mobile robots, there is an uncertainty f in the dynamic controller. Besides, the neural network has a simple structure, can approximate any nonlinear function, and has a strong self-learning ability. Therefore, the neural network is adopted in this study to compensate for the uncertainty *f*. The structure of neural network adaptive trajectory tracking control is shown in Fig. 5 [30]. Since the dynamic system of wheeled mobile robots will be affected by uncertainty, controllers based on kinematics model are often not the true linear and angular speeds of wheeled mobile robots. The *v*c and ωc are assumed as virtual speeds, then a dynamic controller can be designed based on the dynamics model by combining with the neural network and adaptive control, so as to achieve the gradual convergence of the error between the virtual speed and the actual speed of the robot [31].

3.3 Model under active disturbance rejection tracking Control Strategy

Wheeled mobile robot is a highly complex nonlinear system. In order to facilitate the study of wheeled mobile robot, scholars generally assume that it satisfies the condition of "pure rolling without sliding." However, the working environment of wheeled mobile robots is very complicated. Due to icy roads in winter, wet roads in rainy days, or rapid turns, wheeled mobile robots may slide laterally and longitudinally, which destroys the ideal imperfect constraints of the robot [32]. Therefore, it is meaningful to study the trajectory tracking control of wheeled mobile robots under sliding. For wheeled mobile robots in practical applications, wheels often slip, which has been reported in many articles [33, 34]. Therefore, an active disturbance rejection tracking control strategy is designed to solve this problem. The dynamic controller based on active disturbance rejection includes a lumped disturbance compensation part and a virtual control speed tracking part. The lumped disturbance compensation part is designed as follows.

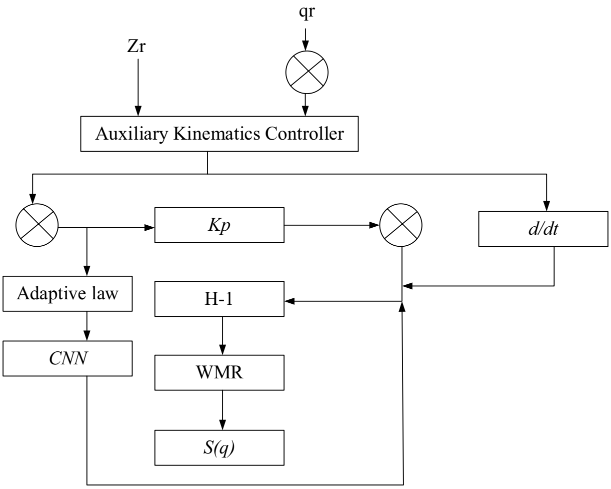


Fig. 5 The structure of neural network adaptive trajectory tracking control

$$\tau_1 = -H^{-1}x_2 \tag{31}$$

The virtual control speed tracking part can be designed as follows.

$$\tau_2 = H^{-1}(z_c + k_p(z_c - x_1)) \tag{32}$$

Based on relevant literature, the dynamic control law is acquired as Eq. (31).

$$\tau_2 = H^{-1}(z_c + k_p(z_c - x_1) - x_2) \tag{33}$$

Based on the previous analysis, the automatic disturbance rejection trajectory tracking control system of wheeled mobile robots is shown in Fig. 5. For the dynamic system equation of wheeled mobile robots, the speed tracking error is set as $e_c = z—z_c$, the linear extended state observer is selected, which is the active disturbance rejection dynamics controller type. If d(t) is bounded and derivable, there is a constant vector $p = [p_1, p_2]$T ($p_i > 0$) for any $\omega_0 > 0$ and $\omega_c > 0$, so that every element in e_c in a limited time meets $|(e_c)i| \leq p_i$ ($i = 1, 2$). In practical engineering applications, state variables are sometimes difficult to directly measure with sensors. In this case, it is very common to use a state observer to estimate the actual state of the system. If the value of each state is required for the linear expansion observer, the core is obtaining the system matrix of the system to be observed. This system goes beyond the limitation of the linear model required by the Lomberg observer and can reconstruct and estimate the state of an unknown nonlinear system. State variables are sometimes difficult to directly measure with sensors in practical engineering applications. In this case, it is very common to use a state observer to estimate the actual state of the system. If the value of each state is required for the linear expansion observer, the core is obtaining the system matrix of the system to be observed. This system goes beyond the limitation of the linear model required by the Lomberg observer and can reconstruct and estimate the state of an unknown nonlinear system.

$$e_c = K_p(-e_c + x)e_c + x_2 \tag{34}$$

It can be transferred into Eq. (33).

$$e_c = A_e e_c + A_x x \tag{35}$$

In the above equation,

$$A_e = -K_p \cdot A_x = \left[K_p, I_2\right] \tag{36}$$

The speed tracking error e_c is bounded and decreases with the increase in the controller bandwidth, that is, the active disturbance rejection dynamic control system is stable (Fig. 6).

3.4 Performance verification and parameters setting of the model

I. To verify the feasibility of the model predictive controller proposed under speed saturation constraints, two reference trajectories (straight line and circle) are

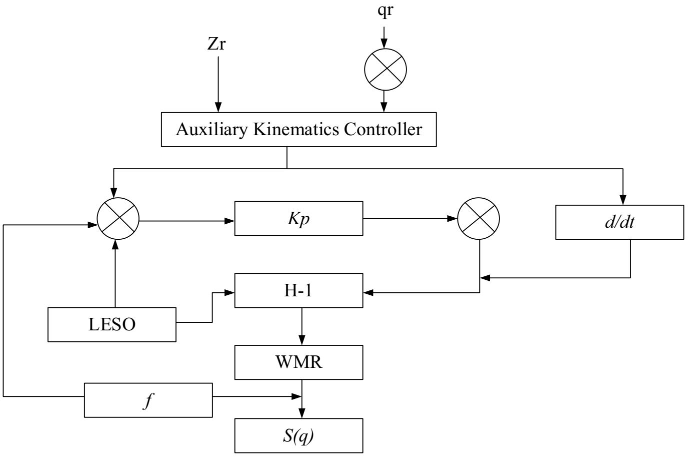


Fig. 6 The structure of automatic disturbance rejection trajectory tracking control system. WMR refers to wheeled mobile robot; and LESO refers to linear extended state observer

selected for MATLAB simulation experiments. The sampling time of MPC is set to $\Delta T=0.05$, the prediction time domain $N_p=40$, the weight matrix $Q=d_m\ (1,1,1)$, and $R=d_a\ (0.5,0.5)$. The constraints that the wheeled mobile robots control input satisfies are supposed as $0(\text{m/s}) \le v \le 1.5(\text{m/s})$ and $-1(\text{rad/s}) \le r \le 1(\text{rad/s})$.

II. To verify the feasibility of the control scheme proposed under the neural network adaptability, two reference trajectories (cosine and circle-straight line) are selected for MATLAB simulation experiments. The model parameters are set as follows: RBF neural network adaptive dynamics controller parameters: $K_p=diag(6,6)$, $\beta_1=\beta_2=0.1$, the number of neurons in the middle layer of the neural network N is 7, the center cm of Gaussian function hm is determined as the even distribution of $\|cm\|$ with an interval of 5 (the initial value of which is 0), the base width σm is set to 8, and the initial value of the weight matrix W is 0.

III. To verify the feasibility of the control scheme proposed under active disturbance rejection tracking control strategy, two reference trajectories (double-twisted line and circular) are selected for MATLAB simulation experiments. The model parameters are set as follows: $r=0.05$ m, $b=0.15$ m, $m_c=3$ kg, $m_w=0.5$ kg, $I_m=m_w$, $r_2=0.00125$ kg · m^2, $I_c=m_c$, $b_2=0.0675$ kg • m^2, $I_w=m_w$, *and* $b_2=0.01125$ kg • m^2. The kinematics controller parameters are defined as follows: $k_1=11.7$, $k_2=11.1$, *and* $k_3=8$; the active disturbance rejection dynamics controller parameters are set to $K_p=diag(8,8)$, and the linear observer parameter $\omega_0=40$.

3.5 Radial basis function neural network model parameter setting and structure

The RBF neural network receives training samples $x_i \in R^d$ from the input layer, and the hidden layer maps the input samples to a new space through the radial basis function. The number of hidden layer nodes is M. If the radial basis function is a Gaussian function, then $ci \in Rd$ represents the center vector of the Gaussian function, and δ_i represents the kernel width of the Gaussian function. The mapping from the input space to the new space is realized by formula (1).

$$\varphi\left(\left\|x-c_i\right\|\right)=exp\left\{-\frac{\left\|x-c_i\right\|^2}{\delta i}\right\} \tag{37}$$

The output layer implements linear weighted summation in the new space, and it is set that w_i is the connection weight between the hidden layer and the output layer. φ(·) is the radial basis function, $y \in R$ is the output result, and the mapping function of $Rd \rightarrow R$ is the following equation.

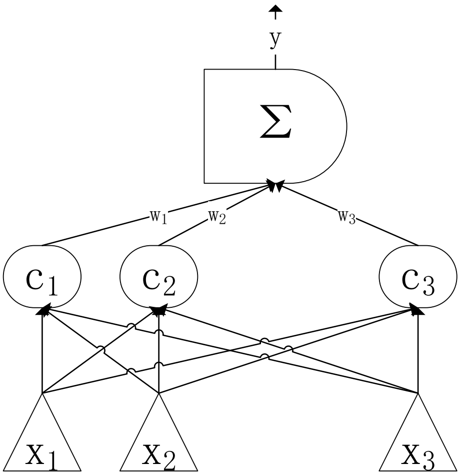
$$y=f(x)=\sum_{i=1}^{M} w_i\varphi\left(\left\|x-c_i\right\|\right), i=1,2,\ldots,M \tag{38}$$

As a result, the RBF neural network completes the nonlinear mapping of f: $Rd \rightarrow R$, as illustrated in Fig. 7.

As mentioned above, the central vector $c_i=[c_{i1},c_{i2},\ldots,c_{id}]^T$ of the radial basis function, the kernel width δ_i and the connection weight w_i between the hidden layer and the output layer are the parameters of the RBF neural network. Among them, c_i and c_i can be determined by formulas (39) and (40) through the FCM clustering algorithm, and the parameter w_i uses the gradient descent learning algorithm to obtain the following equations.

$$c_{ik}=\sum_{j=1}^{n}\mu_{ji}x_{jk}/\sum_{j=1}^{n}\mu_{ji} \tag{39}$$

Fig. 7 RBF network architecture model



$$\delta i = \sum_{j=1}^{n} \mu_{ji} \left\| x_j - c_i \right\|^2 / \sum_{j=1}^{n} \mu_{ji} \tag{40}$$

μ_{ji} represents the fuzzy membership of the sample x_j obtained by the FCM clustering algorithm for the i-th category, and n represents the size of the training sample.

$x^{\sim i} = \varphi(\|x\text{-}c_i\|)$, i = 1, 2, …, M, and there is the following equation.

$$x^{\sim} = [x^{\sim 1}, x^{\sim 2}, \ldots, x^{\sim M}]^T \tag{41}$$

Substituting the center c_i and kernel width δ_i of the radial basis function obtained from Eqs. (39) and (40) into Eq. (35) realizes f: $Rd \rightarrow R$, that is, the nonlinear mapping from the input layer to the hidden layer.

It is assumed that $P = [w_1, w_2, \ldots, w_M]^T$, and Eq. (39) is substituted into Eq. (36) to obtain the following equation.

$$y = p^T x^{\sim} \tag{42}$$

It is obtained from Eq. (42) that the hidden layer to the output layer is a $f{:}RM \rightarrow R$ linear mapping.

When the mapping is established, it first introduces the insensitive loss function corresponding to the RBF linear model. To minimize the value of the insensitive loss function of ε, ε is solved as the constraint term of the optimization problem, and then, it introduces the structural risk term and Gaussian kernel to construct the RBF neural network optimization model with large sample processing ability. The steps are as follows.

Step 1: the values of c_i and δ_i are obtained from Eqs. (39) and (40), and the model input term z is obtained from Eq. (41).
Step 2: insensitive loss function is introduced.

The ε-insensitive loss function $L^E(x, y, f)$ is defined as the following equation.

$$L^E(x,y,f) = |y - f(x)|_E = \max(0.|y - (x)|_E) \tag{43}$$

$x \in R^d$, $y \in R$.

For the linear model of Eq. (42), the corresponding ε-insensitive loss function can be expressed as the following equation.

$$\sum_{i=1}^{n} |y_i^0 - y_i|_i = \sum_{i=1}^{n} \max(0, |y_i^0 - y_i| - \varepsilon) \tag{44}$$

$$\sum_{i=1}^{n} \max(0, \left| p^T x_i^{\sim} - y_i \right| - \varepsilon) \tag{45}$$

Among them, y_i^0 represents the output of the neural network, and y_i represents the real output.

In formulas (44) and (45), the two constraint strips $p^T x_i^{\sim} - y_i < \varepsilon$ and $y_i - p^T x_i^{\sim} < \varepsilon$ may not meet the requirements, so the introduction of relaxation factors ξ_i and $\xi*_i$ can obtain Eqs. (44) and (45).

$$\{y_i - p^T x_i^{\sim} < \varepsilon + \xi i, \xi i \geq 0 \tag{46}$$

$$\{p^T x_i^{\sim} - y_i < \varepsilon + \xi^* i, \xi^* i \geq 0 \tag{47}$$

This algorithm minimizes the ε-insensitive loss function value represented by Eqs. (44) and (45). The value of the insensitive parameter ε will directly affect the accuracy of modeling. Therefore, the parameter λ is introduced, and ε is used as the constraint term of the optimization problem to solve. Combining Eqs. (44) and (45), the optimization problem can be equivalently expressed as the following equation.

$$\min 2\lambda\varepsilon + \frac{\lambda}{\mu n} \sum_{i=1}^{n} \left(\xi_i^2 + \xi_i^{*2}\right) \tag{48}$$

$$s.t.(y_i - p^T \varphi\left(x_i^{\sim}\right) < \varepsilon + \xi i, i = 1, 2, \ldots, n \tag{49}$$

$$s.t.(p^T \varphi\left(x_i^{\sim}\right) - y_i < \varepsilon + \xi * i, i = 1, 2, \ldots, n \tag{50}$$

Among them, the parameter μ is the balance factor, where ξ_i is automatically satisfied, and $\xi*_i \geq 0$.

Step 3: the structural risk terms and kernel functions are introduced.

The principle of minimizing the structural risk of support vector machines is learned, and the regularization term $P^T P$ is introduced to ensure that the structural risk of the algorithm is minimized. The kernel algorithm is an important component of the support vector machine, which is used to improve the computing power of the linear learning machine. After the introduction of the regular term and the kernel function, the optimization problem can be expressed by Eqs. (51) and (52).

$$\min_{p,\varepsilon,\xi_i,\xi_i^*} ||P||^2 + 2\lambda\varepsilon + \frac{\lambda}{\mu n} \sum_{i=1}^{n} \left(\xi_i^2 + \xi_i^{*2}\right) \tag{51}$$

$$s.t. \begin{cases} y_i - p^T \varphi\left(x_i^{\sim}\right) < \varepsilon + \xi_i \\ p^T \varphi\left(x_i^{\sim}\right) - y_i < \varepsilon + \xi_i \end{cases}, i = 1, 2, \ldots, n \tag{52}$$

The Lagrange multiplier is introduced, and the Lagrange function of formula (51–52) can be expressed as the following equation.

$$
\begin{aligned}
L =&||p||^2 + 2\lambda\varepsilon + \frac{\lambda}{\mu n}\sum_{i=1}^{n}\left(\xi_i^2 + \xi_i^{*2}\right) \\
&+ \sum_{i=1}^{n} a_i(y_i - p^T\varphi\left(x_i^{\sim}\right) - \varepsilon - \xi i) \\
&+ \sum_{i=1}^{n} a_i^*(p^T\varphi\left(x_i^{\sim}\right) - y_i - \varepsilon - \xi * i)
\end{aligned}
\tag{53}
$$

The matrix form of the dual problem corresponding to Eq. (53) is the following equation.

$$
\begin{cases}
\max\left[a^T a^{*T}\right]\left[\begin{smallmatrix}\frac{2}{y}y \\ -\frac{2}{y}y\end{smallmatrix}\right] - \left[a^T a^{*T}\right]K^{\sim}\left[\begin{smallmatrix}a \\ a^*\end{smallmatrix}\right] \\
s.t.\left[a^T a^{*T}\right]1 = 1, a, a^* \geq 0
\end{cases}
\tag{54}
$$

$$
y = \begin{bmatrix} y_1 \\ \ldots \\ y_n \end{bmatrix}, a = \begin{bmatrix} a_1 \\ \ldots \\ a_n \end{bmatrix}, a^* = \begin{bmatrix} a_1^* \\ \ldots \\ a_n^* \end{bmatrix}
\tag{55}
$$

$$
K^{\sim} = \left[k^{\sim}\left(x_i^{\sim}, x_j^{\sim}\right)\right] = \begin{bmatrix} K + \frac{\mu n}{\lambda}I & -K \\ -K & K\frac{\mu n}{\lambda}I \end{bmatrix}
\tag{56}
$$

The value of each variable obtained by the solution is the following equation.

$$
\begin{cases}
p = \lambda\sum_{i=1}^{m}\left(\alpha_i - \alpha_i^*\right)\varphi\left(x_i^{\sim}\right) \\
\xi_i = \alpha_i\mu n \\
\xi_i^* = \alpha_i^*\mu n
\end{cases}
\tag{57}
$$

In addition, since $\sum_{i=1}^{m}\left(\alpha_i + \alpha_i^*\right) = 1$, $\mu = \sum_{i=1}^{m}\left(\xi_i + \xi_i^*\right)/n$. Therefore, the parameter μ is similar to v in v-SVR, which can be interpreted as an expected error.

Step 4: forecast.

The prediction function is shown in Eq. (58–59).

$$
y = p^T\varphi\left(x_{test}^{\sim}\right) = \lambda\sum_{i=1}^{n}\left(a_i - a_i^*\right)\varphi^T\left(x_i^{\sim}\right)\varphi\left(x_{test}^{\sim}\right)
\tag{58}
$$

$$
p^T\varphi\left(x_{test}^{\sim}\right) = \lambda\sum_{i=1}^{n}\left(a_i - a_i^*\right)K^{\sim}\left(x_i^{\sim}, x_{test}^{\sim}\right)
\tag{59}
$$

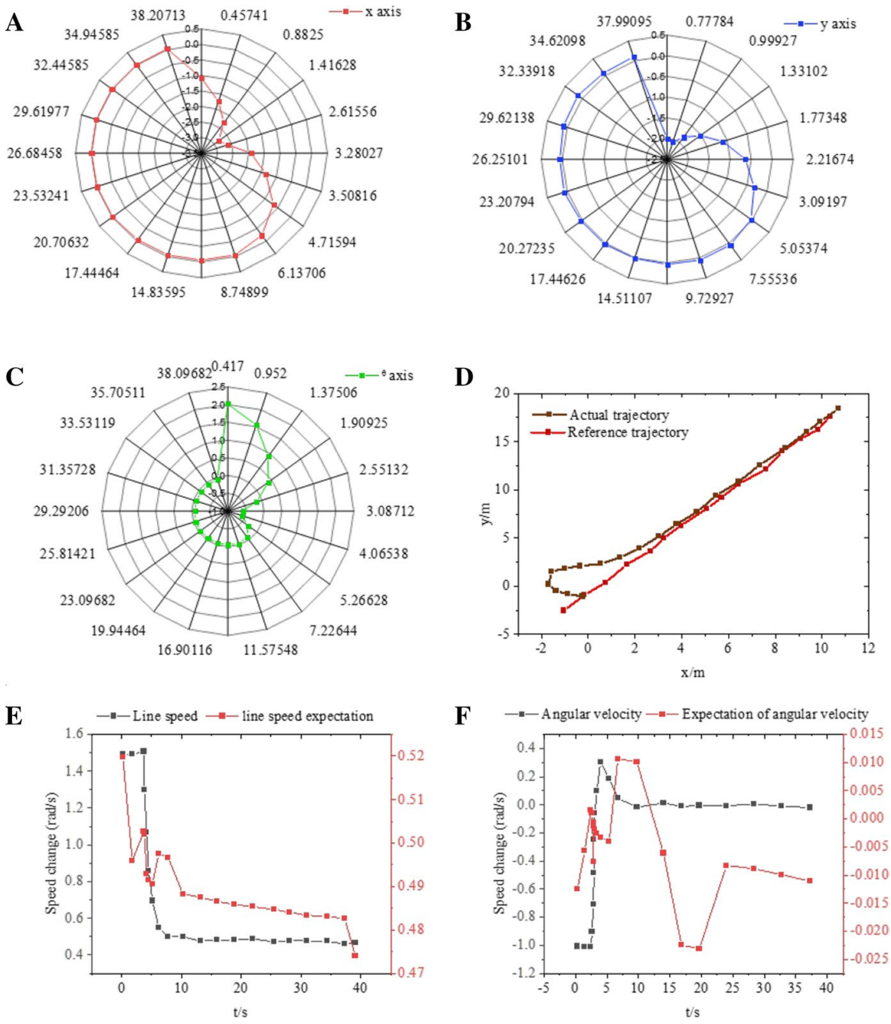


Fig. 8 The movement trajectory of the mobile robot under a straight path

Equations (58) and (59) are equivalent in theory. If used for classification, y = sign $\left(p^T\varphi\left(x_{test}^{\sim}\right)\right)$. If $y > 0$, it belongs to the positive class, and if $y < 0$, it belongs to the negative category.

Results and analysis

3.6 Performance under speed saturation constraints

Fig. 8A –C shows the tracking error curves of wheeled mobile robots with different trajectories (x-axis, y-axis, and $^{\theta}$-axis). In Fig. 8D, the predicted trajectory is

consistent with the actual trajectory, and the maximum difference is about 5 m. In online speed prediction, although the trend of the gap between the actual value and the predicted value is different, the maximum difference does not exceed 1 rad/s. In the angular velocity test, the maximum difference is about 0.98 rad/s, which is within the error range. The wheeled mobile robots can effectively converge the tracking error to zero in a relatively short time, and almost all directional trajectories reach convergence in about ten seconds. It indicates that the predictive controller designed has the advantage of fast convergence. Figure 8D illustrates the trajectory tracking pose curve. For the predictive controller of the model, wheeled mobile robots under speed saturation constraints also have a good

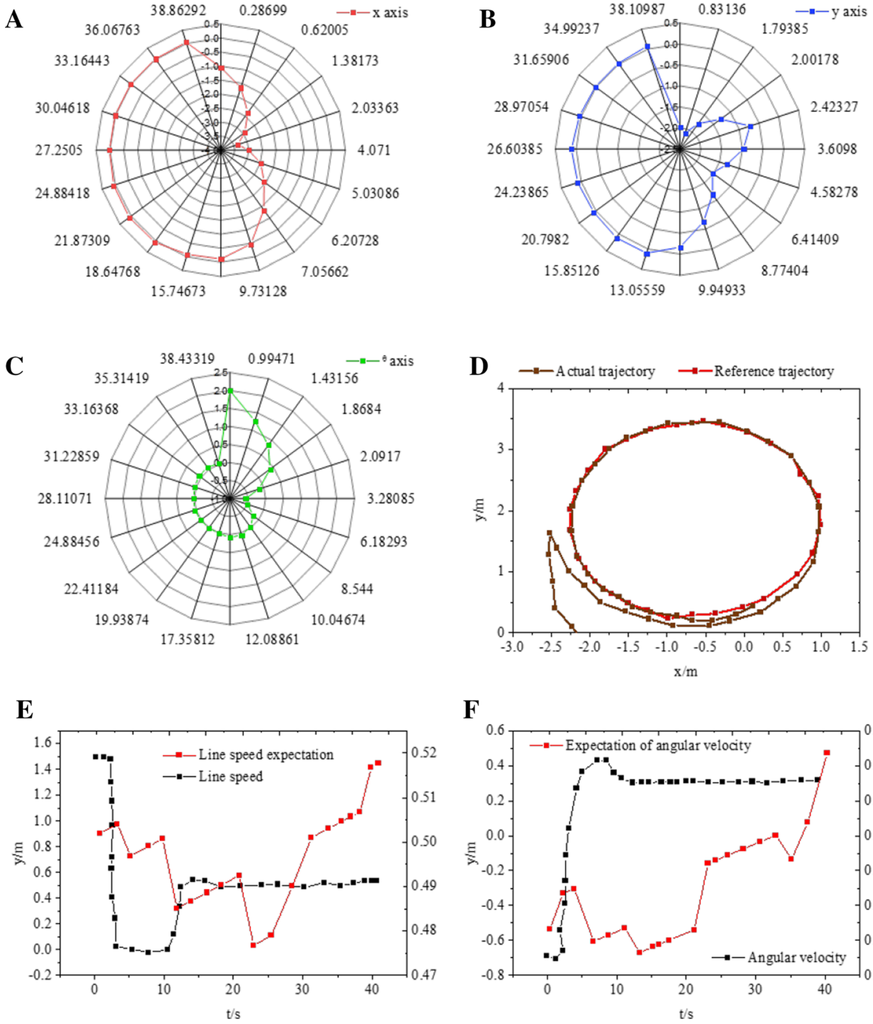


Fig. 9 The trajectory movement of the mobile robot under the circular path

control effect. The actual orbit is basically consistent with the reference orbit. Figure 8E–F are the trajectory control curves. It is found that the input of MPC is still within the specified range and does not deviate from the actual expectations under the constraints.

Figure 9A–C shows the tracking error curves of wheeled mobile robots' trajectories (x-axis, y-axis, and $^{\theta}$ -axis) under a circular trajectory. In Fig. 9D, the predicted trajectory is almost the same as the actual trajectory. In the linear velocity and angular velocity tests, the maximum difference in linear velocity is 1 rad/s. The maximum angular velocity difference is about 0.995 rad/s, which is within the acceptable

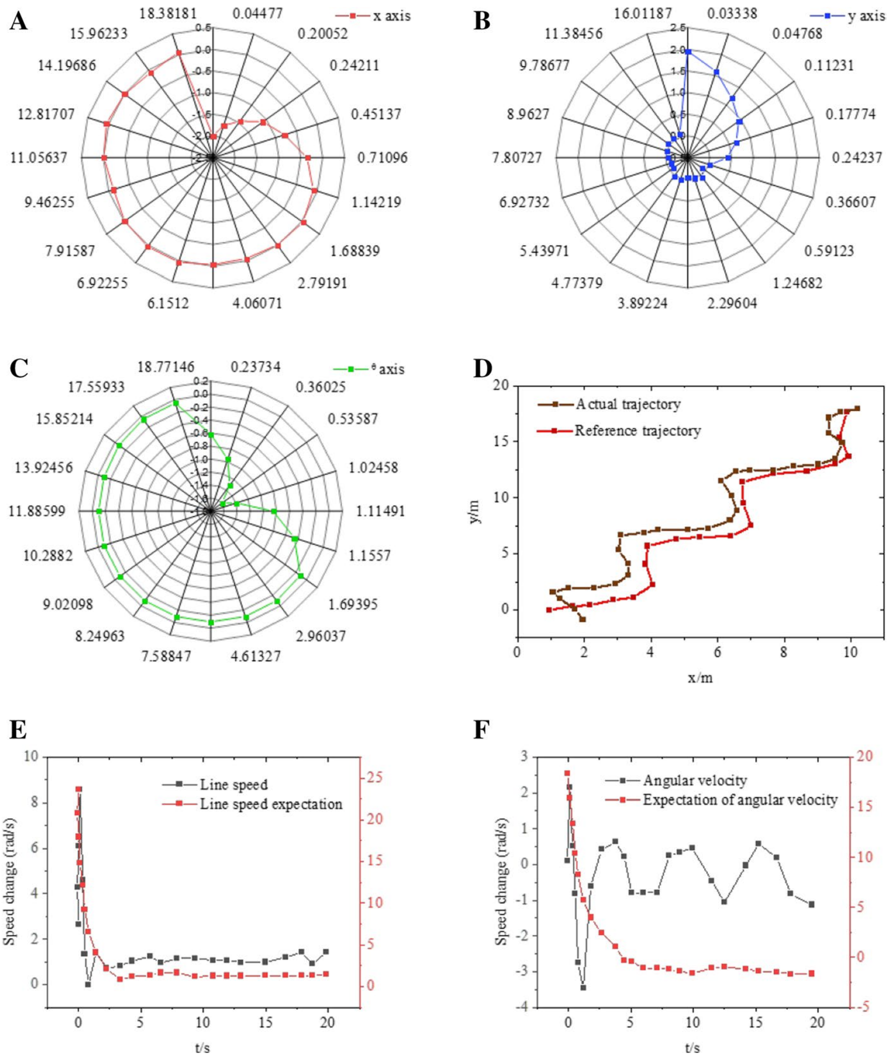


Fig. 10 The movement trajectory of the mobile robot under the control of neural network

error range. They reveal that for a circular trajectory, the wheeled mobile robots also can quickly converge. Figure 9D shows the posture curve following a circular trajectory, and the actual circular trajectory is basically consistent with the reference trajectory. Figure 9E–F shows the control curves under the circular trajectory. The input of MPC is still within the prescribed range under constraints. The above results show that the control algorithm under constraints proposed in this study can effectively break the speed constraints.

3.7 Performance under neural network adaptability

Figure 10A–C shows the tracking error curves under the cosine trajectory, Fig. 10D is the cosine trajectory tracking pose, and Figs. 10E–F shows the cosine trajectory tracking control curves. In Fig. 10D, the predicted trajectory is almost the same as the actual trajectory. In Fig. 10E, in the linear velocity and angular velocity test, the maximum difference in linear velocity is 20 rad/s. In Fig. 10F, the maximum angular velocity difference is about 17.5 rad/s, which is within the acceptable error range. The neural network adaptive control system can also have ideal convergence, and the convergence speed is fast. When the time is 2 s, the data in all directions are effectively converged. For the actual trajectory, the algorithm has good stability in the curved trajectory tracking; and the linear velocity, angular velocity, and the corresponding expected value are kept within a reasonable range. It suggests that the neural network can effectively guarantee the accuracy of movement trajectory.

Figure 11A–H shows the comparisons in tracking error, pose curve, trajectory tracking speed, and control torque curve under different axial trajectories. According to the analysis of the above eight result graphs, the performance results under the two optimization strategies of CNN and BKCPID are close. The predicted value and error value of the angular velocity in Fig. 11F and the torque of the right wheel in Fig. 11G have a significant gap after five seconds. The angular velocity difference is generally maintained within 1 rad/s, and the difference between the right wheel torque and BKCPID under the CNN optimization strategy is generally within 5 rad/s, which is within the acceptable error range. Notably, the proposed algorithm and the latest researched BKCPID algorithm have some fluctuations after the wheeled mobile robots are subject to unknown external interference or voltage changes, but the proposed algorithm is superior in data processing and solution of complex issues. Thus, the range of change is smaller, especially in the control torque. After a short period of fluctuation, the right torque can quickly recover to a certain level. This is also because the proposed algorithm has the ability of self-adaptation. Therefore, it is concluded that the neural network algorithm can adjust the parameters of the controller in real time, and the algorithm proposed is more robust than the latest research algorithm.

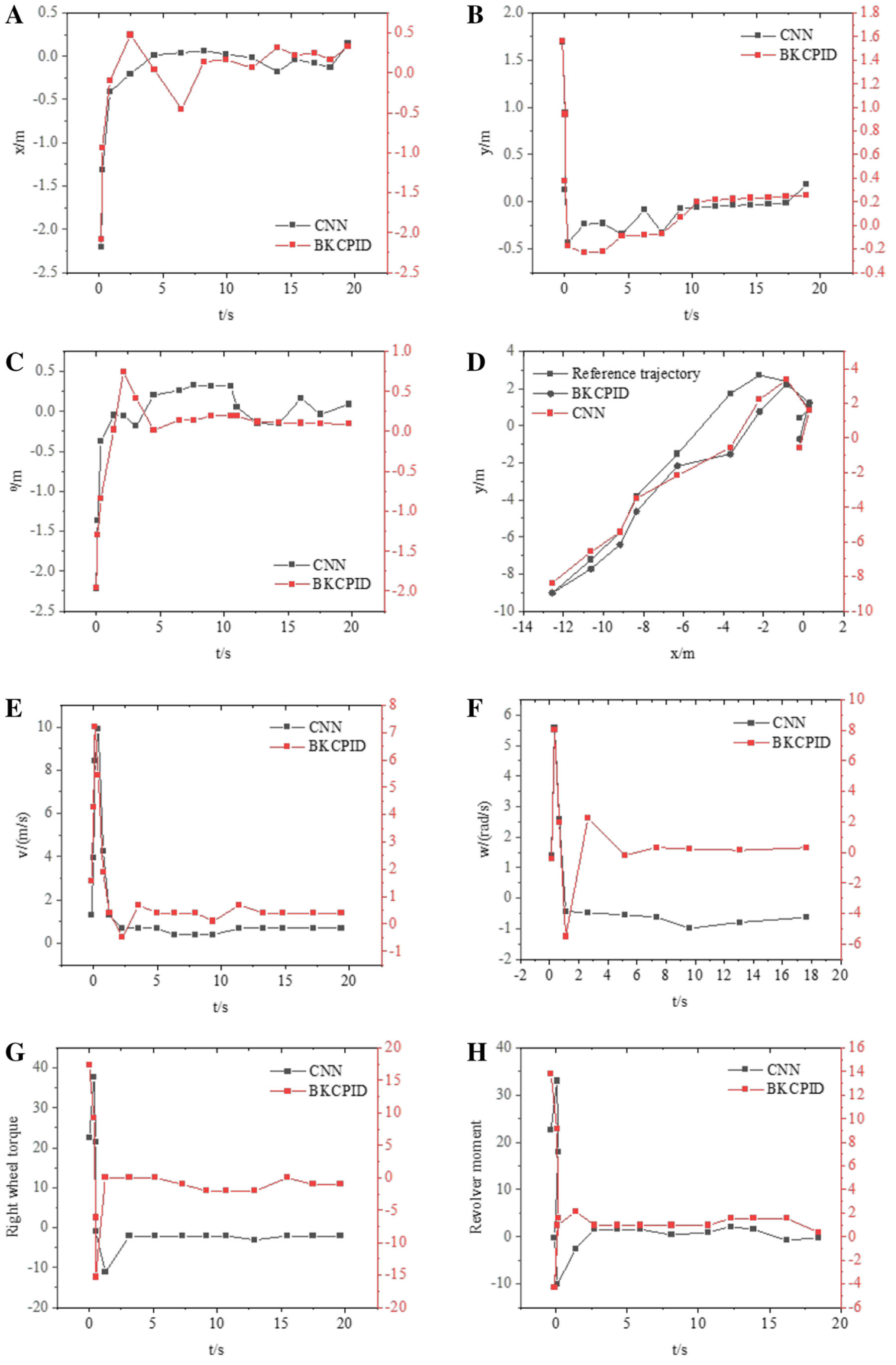


Fig. 11 Performance comparisons of algorithms under different optimization strategies

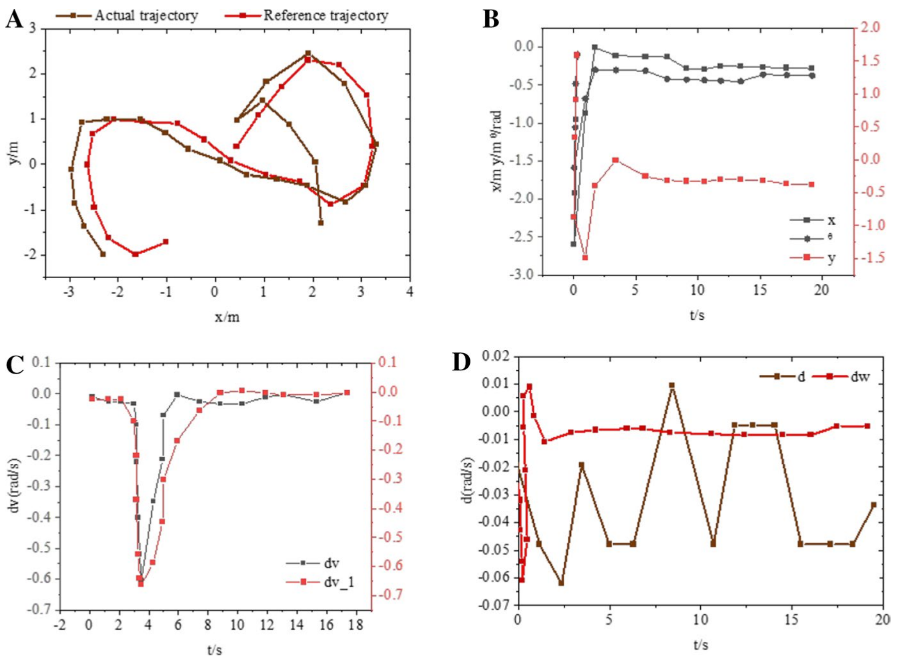
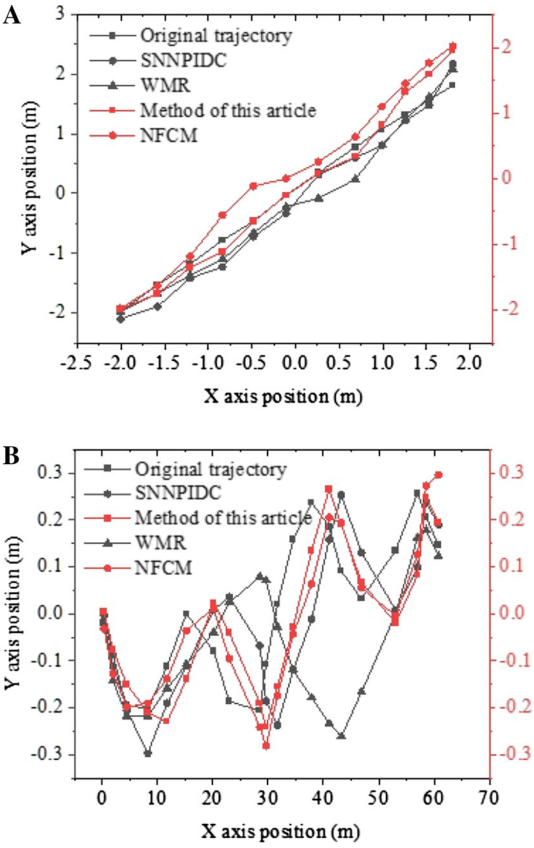


Fig. 12 Actual planning of movement path

3.8 Performance of active disturbance rejection tracking Control strategy

Figure 12A ~ 12D shows the tracking results for trajectory of wheeled mobile robots under lateral sliding, the change of pose error, and the estimated value of the lumped disturbance *d*. In Fig. 12B, the gap between the movement path in the *x* direction and the *y* direction starts from 2.5 s, and the gap is generally maintained within 0.5 rad/s. In Fig. 12D, the trajectory gap between d and *dw* is obvious, and *dw* generally remains unchanged after 2.5 s. However, the fluctuation range of d in the whole process is relatively large, the maximum value is 0.01 rad/s, and the minimum value is −0.06 rad/s. The lateral sliding algorithm proposed in this study can still effectively track the reference trajectory, and the pose error converges quickly and stabilizes at the beginning. Although there are some fluctuations at the beginning, the impact is small. The linear expansion observer can also accurately observe the lateral sliding time in real time. There is a d-value under the lumped disturbance with a finite number of non-derivable points. Such results indicate that the proposed active disturbance rejection tracking control strategy can also well solve the roller skidding of wheeled mobile robots.

Fig. 13 Comparison of the performances of different model algorithms



3.9 Comparisonon performances Of different model algorithms

Under different performance optimization strategies, the performance gap of the model is relatively small, indicating that the model has a stable advantage. In Fig. 13, the performances of different algorithm models are compared on straight trajectories and curved trajectories. The spiking neural network PID optimization (SNPIDC) algorithm is a robot navigation that can make autonomous decisions to overcome obstacles and/or stop the engine, achieving the protective effect. The actuator that drives the robot is not damaged and stops changing directions in the event of insurmountable interference. The neuro-fuzzy cognitive map (NFCM) algorithm uses the rules for updating the NFCM parameters and uses the Lyapunov method to converge the motion control of the mobile robot. The wheeled mobile robot (WMR) realizes a novel motion control structure based on visual serving and hybrid algorithms. The algorithm proposed can basically coincide with the original trajectory regardless of whether it is in a straight line or a curved trajectory. Compared with other algorithms, the algorithm has smaller errors and higher trajectory prediction accuracy, which further proves the effectiveness of the model proposed.

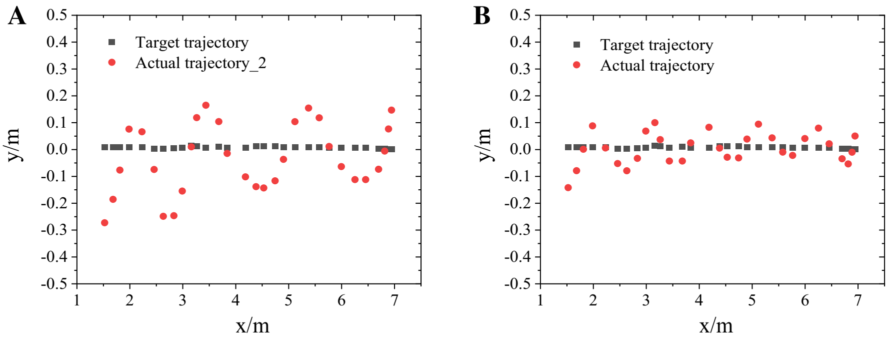


Fig. 14 Control performance analysis based on IoT without occlusion

3.10 Control performance analysis based on IoT

According to the analysis in Fig. 13, although the actual trajectory is different from the predicted trajectory, the actual trajectory changes tend to be a more regular trend, similar to the image trajectory of a trigonometric function. The experimental environment is the IoT and Intelligent Control Laboratory, with an area of 8 × 12 square meters. There are four anchor nodes installed in the Ultra-Wide Band (UWB) positioning system on the four corners of the wall with the coordinates of (4, -2), (− 4, 2), (− 4,10), and (4,10). A tag to be located is marked on the mobile robot. In the case of no occlusion, the mobile robot moves slowly along a straight line from coordinates (2,0) to (7,0) and sends the measured positioning data to the computer via ZigBee, based on which the positioning results can be known clearly. In order to verify the effectiveness of the designed software-based time synchronization method, a set of experiments without time synchronization is now added, and the positioning experiment results are shown in Fig. 14. The positioning accuracy of UWB after time synchronization is significantly improved compared to UWB positioning without time synchronization, which

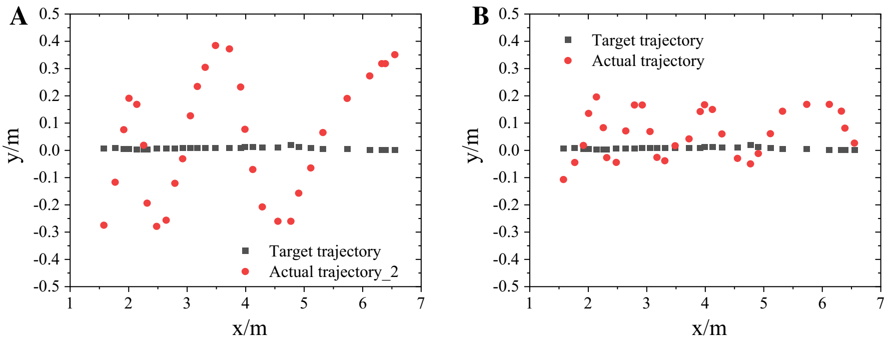


Fig. 15 Control performance analysis based on IoT with occlusion

can reach ± 0.15 m, so the positioning requirements of mobile robots can be basically met.

Similar to Fig. 14, the actual motion trajectory is close to a trigonometric function image trajectory, which is far from the predicted trajectory. The above experiment is repeated in the case that there are two channels blocked. Experiment under the situation without IoT recognition and processing is carried out to compare the superiority of the applied UWB positioning method. The results of the two sets of experiments are shown in Fig. 15. Without the identification and processing of the IoT, the UWB positioning result has a large positioning error, and the randomness of the data is large. After the IoT recognition and positioning through IoT algorithms, the positioning results under relatively no IoT conditions have deteriorated, but it is still better than the situation without IoT recognition and processing. In summary, the UWB wireless positioning system designed shows good accuracy and can meet the general needs of mobile robot positioning.

4 Conclusion

Based on the analysis of the kinematics and dynamics models of the existing wheeled mobile robots, a corresponding algorithm model is proposed in this study via IoT and neural network algorithms. The IoT sensor effectively guarantees the positioning accuracy of the mobile robot, and the algorithm model under the speed saturation constraint is within the specified range and does not deviate from the actual expectations. In addition, the model can reach fast convergence, the neural adaptive algorithm model can effectively ensure the accuracy of the trajectory movement, and the self-anti-interference tracking control strategy can better deal with the wheeled mobile robot's slip. The algorithm model proposed in this paper shows obvious advantages compared with the latest research algorithm. The neural network is adopted to optimize the existing wheeled mobile robots control system algorithm in this study, but there are still many shortcomings. Firstly, a large amount of calculation is often required in MPC, which will lead to poor real-time performance of the controller, causing slower convergence speed. Secondly, the accuracy of the model prediction is improved, but the tracking accuracy of the system is still very poor. Third, the roller skidding of wheeled mobile robots has to be studied further by adding more actual situations. In the follow-up, we will conduct in-depth research on these aspects and continuously improve the performance of the proposed model algorithm.

The online modeling of RBF neural grid has gained a certain degree of attention all over the world in recent years. In addition to the research on trajectory tracking of mobile robots, there are several issues that need to be further studied in the future.

How to optimize and adjust the grid structure to make it more streamlined has to be considered after the grid structure online is adjusted. A reasonable merging or deleting mechanism is required after the hidden units of the grid are merged and deleted.

In the research and design of the trajectory tracking of mobile robots, many parameters have to be set in advance for each controller, and each of these parameters will affect the accuracy of the control system. When the parameters are selected, certain experience is required. Therefore, it needs to consider using a reasonable method to select these parameters, which is to further improve the performance of the control system.

References

1. Tschider CA (2018) Regulating the internet of things: discrimination, privacy, and cybersecurity in the artificial intelligence age. Denv L Rev 96:87–96
2. Irawan Y (2021) Moving load robot using Wifi network and android based. J Robot Control JRC 2(3):217–220
3. Bader K, Lussier B, Schön W (2017) A fault tolerant architecture for data fusion: a real application of kalman filters for mobile robot localization. Robot Auton Syst 88:11–23
4. Tsitsimpelis I, Taylor CJ, Lennox B, Joyce MJ (2019) A review of ground-based robotic systems for the characterization of nuclear environments. Prog Nucl Energy 111:109–124
5. Zaman MHM, Ibrahim MF, Zainal N, Nasir MF (2020) Dual-arm robot with mobile robot platform with master-slave configuration for teleoperation application. Int J 9:1–4
6. Wang L, Shu C, Jin J, Zhang J (2017) A novel traveling wave piezoelectric actuated tracked mobile robot utilizing friction effect. Smart Mater Struct 26(3):035003–035011
7. Li S, Ding L, Gao H, Chen C, Liu Z, Deng Z (2018) Adaptive neural network tracking control-based reinforcement learning for wheeled mobile robots with skidding and slipping. Neurocomputing 283:20–30
8. Saleh AL, Hussain MA, Klim SM (2018) Optimal trajectory tracking control for a wheeled mobile robot using fractional order PID controller. J Univ Babylon Eng Sci 26(4):292–306
9. Lv Z, Li X, Wang W et al (2018) Government affairs service platform for smart city. Futur Gener Comput Syst 81:443–451
10. Chen Y, Hu S, Mao H et al (2020) Application of the best evacuation model of deep learning in the design of public structures. Image Vis Comput 102:103975
11. Van Gerven M, Bohte S (2017) Artificial neural networks as models of neural information processing. Front Comput Neurosci 11:114–121
12. Singh NH, Thongam K (2019) Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. Intel Serv Robot 12(1):55–67
13. Huang H, Zhou J, Di Q, Zhou J, Li J (2019) Robust neural network–based tracking control and stabilization of a wheeled mobile robot with input saturation. Int J Robust Nonlinear Control 29(2):375–392
14. Hu Y, Su H, Zhang L, Miao S, Chen G, Knoll A (2019) Nonlinear model predictive control for mobile robot using varying-parameter convergent differential neural network. Robotics 8(3):64–73
15. Lobov SA, Mikhaylov AN, Shamshin M, Makarov VA, Kazantsev VB (2020) Spatial properties of STDP in a self-learning spiking neural network enable controlling a mobile robot. Front Neurosci 14:88–96
16. Binh NT, Tung NA, Nam DP, Quang NH (2019) An adaptive backstepping trajectory tracking control of a tractor trailer wheeled mobile robot. Int J Control Autom Syst 17(2):465–473
17. Alakshendra V, Chiddarwar SS (2017) Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties. Nonlinear Dyn 87(4):2147–2169
18. Onate JMB, Chipantasi DJM, Erazo NdRV (2017) Tracking objects using Artificial Neural Networks and wireless connection for robotics. J Telecommun Electr Comput Eng JTEC. 9(1–3):161–164
19. Villarreal-Cervantes MG, Sánchez-Santana JP, Guerrero-Castellanos JF (2020) Periodic Event-Triggered Control strategy for a (3, 0) mobile robot network. ISA Trans 96:490–500
20. Chen D, Zhang Y (2017) Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators. IEEE Trans Neural Netw Learn Syst 29(9):4385–4397

21. Liao J, Chen Z, Yao B (2018) Model-based coordinated control of four-wheel independently driven skid steer mobile robot with wheel–ground interaction and wheel dynamics. IEEE Trans Industr Inf 15(3):1742–1752
22. Li X, Wang M (2020) Consensus control for wheeled mobile robots under input saturation constraint. IEEE Access 8:177125–1771230
23. Zhang Y, Li S, Guo H (2017) A type of biased consensus-based distributed neural network for path planning. Nonlinear Dyn 89(3):1803–1815
24. Wu Y, Wang Y (2020) Asymptotic tracking control of uncertain nonholonomic wheeled mobile robot with actuator saturation and external disturbances. Neural Comput Appl 32(12):8735–8745
25. Kassaeiyan P, Tarvirdizadeh B, Alipour K (2019) Control of tractor-trailer wheeled robots considering self-collision effect and actuator saturation limitations. Mech Syst Signal Process 127:388–411
26. Peng S, Shi W (2018) Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot. IEEE Access 6:43414–434124
27. Gopalakrishnan B, Singh AK, Krishna, KM, Manocha D (2018) Solving chance constrained optimization under non-parametric uncertainty through hilbert space embedding. arXiv preprint arXiv:181109311. 125–136
28. Zuo Y, Wang Y, Liu X (2018) Adaptive robust control strategy for rhombus-type lunar exploration wheeled mobile robot using wavelet transform and probabilistic neural network. Comput Appl Math 37(1):314–337
29. Mevo BB, Saad MR, Fareh R (2018) Adaptive sliding mode control of wheeled mobile robot with nonlinear model and uncertainties. 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE): IEEE 1–5
30. Shafei HR, Bahrami M (2020) Trajectory tracking control of a wheeled mobile robot in the presence of matched uncertainties via a composite control approach. Asian J Control. 245–263
31. Goswami NK, Padhy PK (2018) Sliding mode controller design for trajectory tracking of a nonholonomic mobile robot with disturbance. Comput Electr Eng 72:307–323
32. Terakawa T, Komori M, Matsuda K (2019) Motion analysis of an omnidirectional mobile robot with wheels connected by passive sliding joints. Springer, IFToMM World Congress on Mechanism and Machine Science, pp 2279–2288
33. Terakawa T, Komori M (2020) Static force analysis of an omnidirectional mobile robot with wheels connected by passive sliding joints. Springer, Symposium on Robot Design, Dynamics and Control, pp 347–354
34. Komori M,Terakawa T (2020) Omnidirectional mobile robot and vehicle, uninterrupted transmission system, intuitive operating method, and riding robotics. The International Conference of IFToMM ITALY: Springer, 591–598

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.