

Research Article

Obstacle Avoidance Control for Multisteering Mode of Multi-axle Wheeled Robot Based on Trajectory Prediction Strategy

Yongqiang Zhu ¹, Junru Zhu ², and Pingxia Zhang ¹

¹School of Mechanical and Automotive Engineering, Qingdao University of Technology, Qingdao 266520, China

²Pegasus California School, Qingdao 266105, China

Correspondence should be addressed to Yongqiang Zhu; yongqiangzhu@163.com

Received 13 July 2021; Accepted 15 October 2021; Published 30 October 2021

Academic Editor: Radek Matušů

Copyright © 2021 Yongqiang Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A multi-axle wheeled robot is difficult to be controlled due to its long body and a large number of axles, especially for obstacle avoidance and steering in narrow space. To solve this problem, a multisteering mode control strategy based on front and rear virtual wheels is proposed, and the driving trajectory prediction of the multi-axle wheeled robot is analyzed. On this basis, an obstacle avoidance control strategy based on trajectory prediction is proposed. By calculating the relationship between the lidar points of the obstacle and the trajectory coverage area, the iterative calculation of the obstacle avoidance scheme for the proposed steering is carried out, and the feasible obstacle avoidance scheme is obtained. The mechanical structure, hardware, and software control system of a five-axle wheeled robot are designed. Finally, to verify the effectiveness of the obstacle avoidance strategy, a Z-shaped obstacle avoidance experiment was carried out. The results confirm the effectiveness of the proposed control strategy.

1. Introduction

A multi-axle wheeled robot has the advantages of many wheels and large carrying capacity, so it has been widely used in the fields of bulk cargo transportation in factories and field rescue. However, at the same time, more wheels and longer body also lead to complex steering control, more steering space occupied, and other problems. Especially now with the development of intelligent robot control technology, it is very essential to carry out research on automatic obstacle avoidance and other aspects in view of the above problems, to realize flexible steering obstacle avoidance control, especially the research on autonomous driving with a low space occupancy rate.

At present, obstacle avoidance technology is a hotspot in the field of intelligent mobile robot. Many studies about autonomous obstacle avoidance of robots are mainly focusing on two aspects: multi-robot swarm control [1–4] and a single robot control [5]. Research on the single robot is further divided into global path planning [6, 7] and local obstacle avoidance path planning [8, 9]. Robots of global

path search need to obtain the global distribution of obstacles from the starting point to the destination in advance [10]. However, the robot of local path planning does not know the distribution of obstacles from the starting point to the destination in advance, so it can only detect the situation of obstacles around the robot and conduct autonomous obstacle avoidance while driving [11]. In the local navigation, the robot can decide or control its motion and orientation autonomously using sensors. Many researchers presented various algorithms, such as genetic algorithm [12], neural network [13–15], ant colony optimization algorithm [16], fuzzy logic [17], neuro-fuzzy [18], simulated annealing algorithm [19], and particle swarm optimization algorithm [20]. But these studies do not consider obstacle avoidance in narrow areas.

After the robot detects the obstacle, it is essential to select strategies about how to avoid obstacles. Aimed at the complex working environment of the ward inspection robot and the nonstructured and diverse obstacles, Yu et al. [21] contrive obstacle detection and avoidance method based on the laser sensor camera. Xie et al. [22] proposed an improved

obstacle avoidance method based on the constraint of obstacle boundary condition by detecting the obstacle boundaries to obtain the reference direction of the robot and avoiding the loss of obstacle details; it eliminates the blindness of finding the feasible direction of the robot and improves the efficiency of finding the reference direction. Chen et al. [23] constructed a supervisory control algorithm based on a barrier function method, which works in a plug-and-play fashion with any lower-level navigation algorithm. Rostami et al. [24] proposed a modified artificial potential field algorithm. By using this method, the robot can pass obstacles around the target without collision and reach the target.

To navigate autonomously, it is important for mobile robots to sense the outer environment efficiently and reliably. Robots can approximate their environment relying on the combination of one or more sensors with proper methods to process the information produced by sensors and then make decisions [25–28]. The main sensors to detect the obstacles are infrared radar, ultrasonic radar, vision modules, and laser radar. Because of the small detection range, infrared radars have little use to detect obstacles. Because the ultrasonic sensor can only reflect the distance of the obstacles in a certain area and cannot accurately reflect the specific angle of the obstacles, it cannot be used for accurate obstacle avoidance control of the robot. Laser radars can be divided into 2D- and 3D-laser radars [29, 30]. 2D-laser radar can accurately reflect the distance and angle position of obstacles, and it is relatively cheap, so it is widely used at present. Lidar has the advantages of high precision, large detection range, and fast sweep frequency. It is widely used in the obstacle detection field of the ground mobile robot [31]. Madhavan and Adharsh [32] proposed a method about how to get the data from the lidar sensor and an algorithm is designed to get the information of obstacles by filtering clustering technique. Peng et al. [33] proposed efficient obstacle detection and obstacle avoidance algorithm based on 2D lidar. In this way, it can get the information of obstacles by filtering and clustering the laser-point cloud data. Takahashi et al. [34] developed a new emergency obstacle avoidance module for moving robots that use Light Detection and Ranging (LIDAR) to detect static and moving obstacles. 3D lidar can get the height information of obstacles, so it is suitable for the bumpy environment, such as the sea and field. However, its data is large which costs more processing time, more expensive than 2D lidar [35, 36]. Therefore, we use 2D lidar as the detection sensor in this paper.

When avoiding obstacles, the robot needs to make fast steering decisions according to the distribution of obstacles to meet the fast-moving speed. To accelerate the obstacle avoidance speed and realize real-time obstacle avoidance, Wu et al. [37] proposed a deep reinforcement learning method ANOA (Autonomous Navigation and Obstacle Avoidance) to enhance the intelligence of USVs in the sophisticated mane environment, the ANOA algorithm is proposed for real-time path planning with obstacle avoidance. Borenstein and Koren [38] developed a new real-time obstacle avoidance approach for mobile robots. By

permitting the detection of unknown obstacles simultaneously with the steering of the mobile robot to avoid collisions and advancing toward the target, this method can save processing time. Xu et al. [39] proposed a new maximum-speed aware velocity obstacle (MVO) algorithm. It can control a mobile robot to avoid one or multiple high-speed obstacles. Zaheer et al. [40] proposed a new real-time “Free-configuration Eigenspace” (FCE) technique for obstacle avoidance and navigation. The FCE enables an autonomous robot to detect unknown obstacles and avoid collisions while guiding the robot toward the target. Hu et al. [41] introduced an experiential aggregative reinforcement learning method based on multiattribute decision-making. To build a virtual-force field between the obstacles and the robot, Zheng et al. [42] proposed a fast hybrid position/virtual force controller. However, these studies do not consider the problem of fast obstacle avoidance for multi-axle robots.

Other researchers have studied the trajectory curves of obstacle avoidance to get good performance. Akka and Khaber [43] proposed a trajectory tracking control method for a mobile robot when there are static obstacles on the reference trajectory. The tracking control is based on the fuzzy parallel distribution compensation scheme and the linear quadratic controller is used as the state feedback controller of each subsystem. By fully considering the nonholonomic constraints of mobile robot systems, Yuan et al. [44] proposed a new quadrupole potential field method to plan collision-free trajectories. Kuo et al. [45] implemented a kind of obstacle avoidance method focusing on curvature constraint for a nonholonomic mobile robot. It treated the robot as a particle and adopted the curvature constraint streamline and pure tracking streamline change method. However, these studies do not consider the problem of robot trajectory control in multimode steering.

Most of the robots used in the above research have round bodies and a few of them are multi-axle wheeled robots with long bodies. These obstacle avoidance strategies are very suitable when the length and width of the robot are almost equal or the shape of the robot is close to a circle. However, for the multi-axle robot, its body length is much larger than its width. Therefore, the obstacle avoidance control becomes very complex compared with the wheeled robot with a general circular body. The multi-axle robot will occupy more space and have a larger turning radius while turning. Therefore, it is urgent to propose a new obstacle avoidance method combining the characteristics of the long body, to realize obstacle avoidance in a narrow space. Then, it can reflect the mobility and flexibility of the long body robot.

At present, most robots can only turn their front wheels or part of the wheels, which results in low flexibility and requires a large turning space. Therefore, some studies have been carried out from the perspective of wheel steering control, mainly including four-wheel steering [46], multi-wheel steering [47], and trailer following control driving [48]. However, these studies do not consider the multimode steering problem of the multi-axle robot. The multi-axle robot has a longer body and more wheels. In this case, if we can control the steering of each wheel individually, especially for different steering modes used in the different distribution of

obstacles, it will significantly reduce the difficulty of obstacle avoidance. Moreover, the space occupation will be reduced and the flexibility will be improved. In addition, in the actual obstacle avoidance, the robot's obstacle avoidance algorithm cannot occupy too much time; otherwise, it will affect the speed of obstacle avoidance.

Therefore, it is necessary to study the fast and autonomous obstacle avoidance in a narrow space in combination with the long body of the multi-axle wheeled robot.

The contribution of this article is to propose a new obstacle avoidance control strategy based on multisteering mode and trajectory prediction. Then, we designed a five-axle wheeled robot with a long body and all wheels can steer independently for the experiment.

The rest of this article is organized as follows: firstly, the mechanical structure and hardware circuit of a lidar robot used in the obstacle avoidance experiment are analyzed in Section 2. It has five axles, and all wheels can steer and drive. Secondly, we propose three strategies in Section 3. They are the multisteering mode strategy based on the front and rear virtual wheels, moving trajectory prediction strategy, and obstacle avoidance strategy based on trajectory prediction. Thirdly, the on-board control system and the upper computer control system of the robot are introduced in Section 4. On this basis, to verify the proposed method, the experiments of obstacle avoidance with Z-shaped obstacle distribution are carried out to verify the effect of the control algorithm in Section 5. Conclusions are given in Section 6.

2. Structure of the Five-Axle Lidar Wheeled Robot

2.1. Whole Structure. The mechanical dimension of the robot is 250 mm × 600 mm. The mechanical overview of the robot is in Figure 1. The robot has 10 wheels, and each wheel has two motors. One is the steering servo motor for driving wheel to steer, and another is the driving motor for driving wheel to roll [49]. The wheel system structure is shown in Figure 2. The steering servo is fixed to the robot body through the steering gear plate. The wheel driving motor is fixed on the steering gear plate. In this way, the steering servo can drive the wheel driving motor and wheel to realize the robot steering through the steering gear plate. Hence, every wheel has two degrees of freedom.

2.2. Hardware Components. The hardware circuit of the wheeled robot is composed of a steering servo, lidar, wheel driving motor, motor driving module, PCA9685, ESP32S, USART GPU serial communication touch screen, and 7.4V3000mAh30C Li Battery, as shown in Figure 3.

The core control unit is ESP32S, which is responsible for transmitting lidar signals to the laptop through WiFi and receiving autonomous obstacle avoidance instructions from the laptop. The ESP32S communicates with the lidar through a serial port to obtain the distance and angle information of the surrounding obstacles. ESP32S and the serial touch screen communicate through the serial communication port

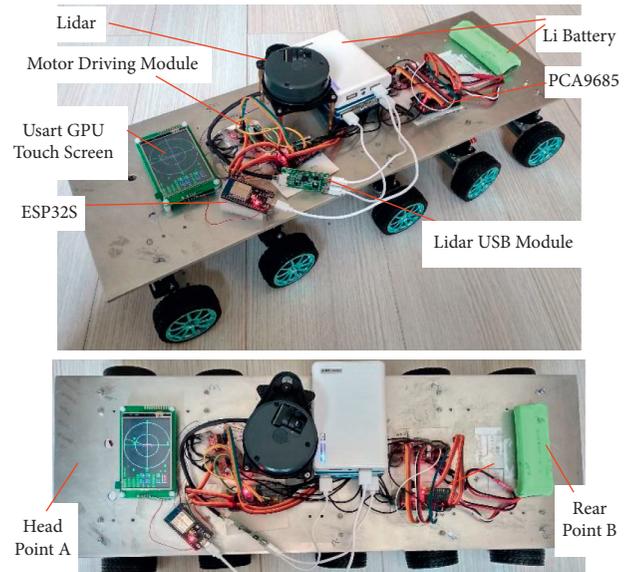


FIGURE 1: The architecture of a five-axle wheeled robot for research.

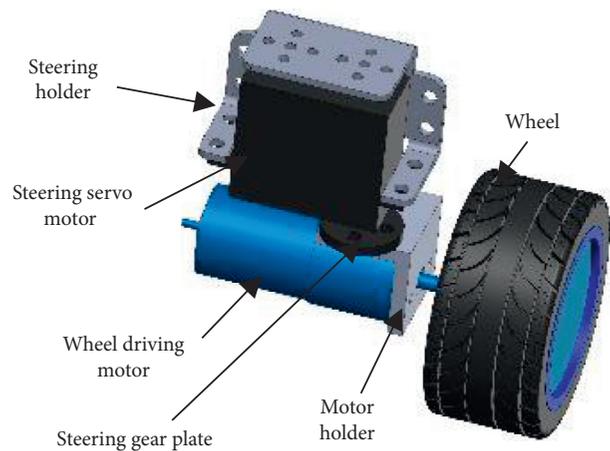


FIGURE 2: Wheel system structure.

to display the information of the surrounding obstacles in real time. We can set the displayed information through the screen. ESP32S communicates with the PCA9685 module through the I²C bus. PCA9685 module controls the steering servo of each wheel in PWM mode to realize the control of wheel steering angle. At the same time, the speed of each wheel driving motor is controlled to realize the control of the wheel driving speed. Because each wheel is driven by two actuators, we can independently control the steering angle, driving speed, and direction of each wheel. Then, we can achieve very flexible multisteering modes. For cost-efficient purposes, this article uses a Delta-2B laser radar. This is a 2D lidar. Its maximum scanning range is 8 meters. Its minimum scanning range is 0.2 meters. The scanning angle is 360°, and the angle resolution is 0.592°. This means that we can get 608 pairs of lidar data points including angles and distance at a time. The scanning frequency is 5~10 Hz, so the interval time is about 100~200 ms. Therefore, the control unit is needed to calculate the next movement direction of the robot

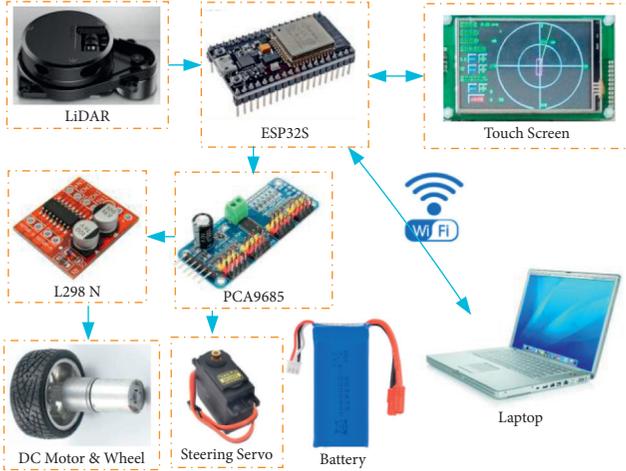


FIGURE 3: Hardware structure of the five-axle all-wheel-steering wheeled robot.

according to these lidar points of obstacles within 100 ms. The Lidar uses a 3.3V-TTL serial port (UART) as the communication interface. To get live data scanned by the lidar, we connect the lidar with ESP32S. The communication between the lidar and ESP32S is UART communication. The data obtained with the ESP32S from the lidar will contain the distance of the obstacle, at what angle the lidar has rotated to scan the point. This data can then be used to employ the algorithms used for obstacle detection and obstacle avoidance. The mounting position of the lidar was on top of the robot body to achieve a proper and unobstructed plane of measurement.

3. Control Strategy

3.1. Multimode Steering Control Strategy Based on Front and Rear Virtual Wheels. When the robot turns to the left or right, to ensure the symmetry of the trajectory and easy control, we propose to set the virtual steering wheels at the midpoint of the first and last axles of the robot. It is shown from Figures 4–8. In this way, according to Ackerman's theorem, we can calculate the steering angles of all actual wheels of the robot according to the steering angles of the two virtual wheels: AF and AR . According to the need for obstacle-avoiding steering of the robot, the following steering modes are proposed based on the front and rear virtual wheel steering angles AF and AR and moving speed V .

3.1.1. $AF \cdot AR < 0$. This is the adverse-phase steering mode, as shown in Figure 4. At this point, the steering direction of the front and rear virtual wheels is opposite, and the actual steering direction of the front and rear wheels of the robot is also opposite. The robot can have a relatively small turning radius. Space used for steering is also small. Therefore, this mode is suitable for a relatively narrow steering space.

The distance between the turning center point O and the first axle of the robot is projected on the longitudinal centerline of the robot as

$$L_{11} = \sum_{j=1}^{n-1} L_j \frac{\tan(AF/180\pi)}{\tan(AF/180\pi) + \tan(AR/180\pi)}, \quad (1)$$

where L_i is the distance between the i th and $(i+1)$ th axle. n is the number of robot axles, in this paper, $n=5$. AF and AR are the steering angles of the front and rear virtual wheels on the center of the first axle and last axle, respectively. The unit is degree. When the wheels are steering clockwise, the angle is positive; otherwise, it is negative.

The distance between the turning center point O and the longitudinal center axis of the robot body is

$$R = \frac{L_{11}}{\tan(AF/180\pi)}. \quad (2)$$

The steering angle of the j th wheel in the i th axle is

$$\alpha_{ij} = \tan^{-1} \left(\frac{(L_{11} - \sum_{k=1}^{i-1} L_k)}{(R - (-1)^j B/2)} \right), \quad (3)$$

where α_{ij} is the steering angle of the j th wheel in the i th axle. When the wheels are steering clockwise, the angle is positive; otherwise, it is negative. $i=1, 2, \dots, n$, where n is the number of robot axles, and $j=1, 2$ is the left wheel and 2 is the right wheel. L_k is the distance between the k th and $(k+1)$ th axle. B is the wheelbase of the left and right wheel.

The distance from the turning center point O to the j th wheel in the i th axle is

$$R_{ij} = \frac{(R - (-1)^j B/2)}{\cos(\alpha_{ij})}. \quad (4)$$

The maximum radius is

$$R_{\max} = \max(|R_{11}|, |R_{12}|, |R_{51}|, |R_{52}|). \quad (5)$$

According to the size of the moving radius, the program carries out the differential speed (the degree to which the speed V_{ij} deviates from the middle point 90), and the speed of the j th wheel in the i th axle is

$$V_{ij} = 90 + (V - 90) \left(\frac{|R_{ij}|}{R_{\max}} \right), \quad (6)$$

where V_{ij} is the speed of the j th wheel in the i th axle and V is the robot moving control speed. V is defined as

$$\begin{cases} 90 < V \leq 180, & \text{robot moves forward,} \\ V = 90, & \text{robot stops,} \\ 0 \leq V < 90, & \text{robot moves backward,} \end{cases} \quad (7)$$

where 180 and 0 correspond to the maximum speed forward and the maximum speed backward, respectively, and there is a linear correspondence between the threshold and the speed.

3.1.2. $AF = 0$ and $AR \neq 0$. This is nonsteering first-axle wheel mode, as shown in Figure 5.

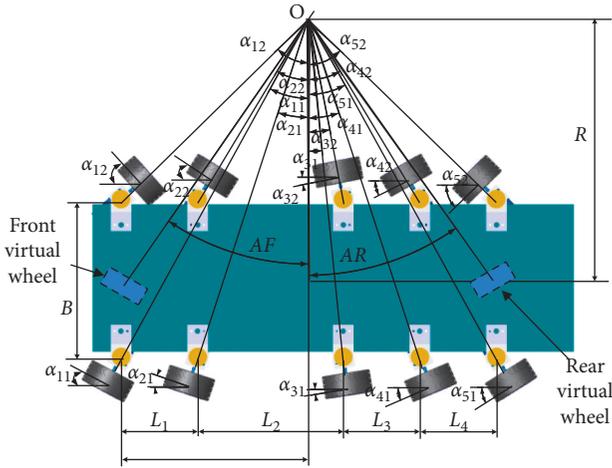


FIGURE 4: Geometric relation diagram of wheel steering angles in other steering modes of the robot.

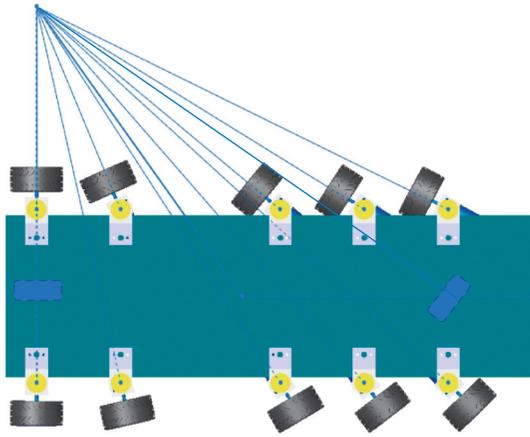


FIGURE 5: Nonsteering first-axle wheel mode.

The distance between the turning center point O and the longitudinal center axis of the robot body is

$$R = \frac{\sum_{i=1}^{n-1} L_i}{\tan((AR/180)\pi)} \quad (8)$$

The steering angle of the j th wheel in the i th axle is

$$\alpha_{ij} = -\tan^{-1}\left(\frac{\sum_{k=1}^{i-1} L_k}{(R - (-1)^j B/2)}\right), \quad i > 1, j = 1, 2. \quad (9)$$

The distance from the turning center point O to the j th wheel in the i th axle is

$$R_{ij} = \begin{cases} \frac{R - (-1)^j B}{2}, & i = 1, \\ \frac{\sum_{k=1}^{i-1} L_k}{\sin \alpha_{ij}}, & i = 2, 3, 4, 5, \end{cases} \quad (10)$$

where $j = 1, 2$. At this point, the turning radius of the wheel of the fifth axle is the largest, so the maximum radius is

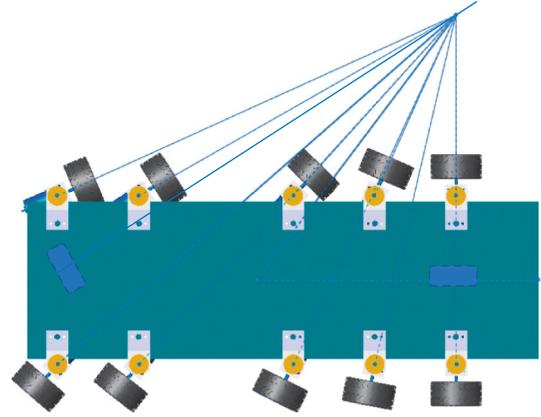


FIGURE 6: Nonsteering last-axle wheel mode.

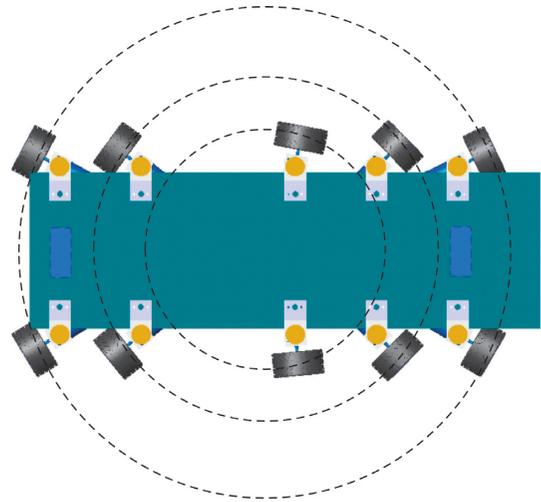


FIGURE 7: In situ rotation mode.

$$R_{\max} = \max(|R_{51}|, |R_{52}|). \quad (11)$$

According to the size of the moving radius, the program carries out the differential speed to get the speed of each wheel as

$$V_{ij} = 90 + (V - 90) \left(\frac{|R_{ij}|}{R_{\max}} \right). \quad (12)$$

3.1.3. $AF \neq 0$ and $AR = 0$. This is the nonsteering last-axle wheel mode, as shown in Figure 6.

The distance between the turning center point O and the longitudinal center axis of the robot body is

$$R = \frac{\sum_{i=1}^{n-1} L_i}{\tan((AF/180)\pi)} \quad (13)$$

The steering angle of the j th wheel in the i th axle is

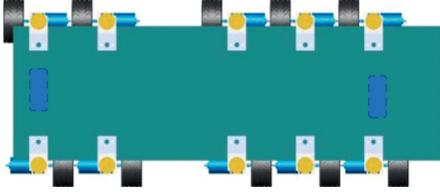


FIGURE 8: Lateral driving mode.

$$\alpha_{ij} = \tan^{-1} \left(\frac{\sum_{k=i}^{n-1} L_k}{R - (-1)^j B/2} \right), \quad i = 1, 2, 3, 4, j = 1, 2. \quad (14)$$

The distance from the turning center point O to the j th wheel in the i th axle is

$$R_{ij} = \begin{cases} \frac{\sum_{k=i}^{n-1} L_k}{\sin \alpha_{ij}}, & i = 1, 2, 3, 4, \\ R - \frac{(-1)^j B}{2}, & i = 5, \end{cases} \quad (15)$$

where $j = 1, 2$.

At this point, only the first axle wheel has the largest turning radius, so the maximum radius is

$$R_{\max} = \max(|R_{11}|, |R_{12}|). \quad (16)$$

The speed of the j th wheel in the i th axle is

$$V_{ij} = 90 + (V - 90) \left(\frac{|R_{ij}|}{R_{\max}} \right). \quad (17)$$

3.1.4. $AF = 90^\circ$ and $AR = -AF$. This is the in situ rotation state, as shown in Figure 7. The turning center is located between the 2nd and 3rd axles.

The steering angle of the j th wheel in the i th axle is

$$\begin{aligned} \alpha_{1j} &= (-1)^{j-1} \tan^{-1} \left(\frac{(\sum_{i=1}^{n-1} L_i/2)}{(B/2)} \right), \quad j = 1, 2, \\ \alpha_{2j} &= (-1)^{j-1} \tan^{-1} \left(\frac{(\sum_{i=1}^{n-1} L_i/2 - L_1)}{(B/2)} \right), \quad j = 1, 2, \\ \alpha_{ij} &= (-1)^j \tan^{-1} \left(\frac{(\sum_{j=1}^{n-1} L_j/2 - \sum_{k=i}^{n-1} L_k)}{(B/2)} \right), \quad i = 3, 4, j = 1, 2, \\ \alpha_{5j} &= -\alpha_{1j}, \quad j = 1, 2. \end{aligned} \quad (18)$$

The distance from the turning center point O to the j th wheel in the i th axle is

$$R_{ij} = \frac{(B/2)}{\cos \alpha_{ij}}. \quad (19)$$

In this case, only the wheel trajectories of the first and fifth axles will have the maximum radius, so the maximum radius is

$$R_{\max} = \max(|R_{11}|, |R_{12}|, |R_{51}|, |R_{52}|). \quad (20)$$

The speed of the wheel on the j th side of the i th axle is

$$V_{ij} = 90 - (-1)^j (V - 90) \left(\frac{|R_{ij}|}{R_{\max}} \right), \quad i = 1, 2, \dots, 5, j = 1, 2. \quad (21)$$

3.1.5. $AF = 90^\circ$ and $AR = AF$. This is the lateral driving state of 90° , as shown in Figure 8. The wheel steering angle of the j th side of the i th axle is

$$\alpha_{ij} = 90^\circ, \quad i = 1, 2, \dots, 5, j = 1, 2. \quad (22)$$

The wheel speed of the j th side of the i th axle is

$$V_{ij} = V, \quad i = 1, 2, \dots, 5, j = 1, 2. \quad (23)$$

3.2. Trajectory Prediction Strategy. This module mainly calculates the robot's next running trajectory according to the steering angle AF , AR of the front and rear virtual wheels, and the robot moving speed V . The specific geometric relationship is shown in Figure 9.

When the robot is steering, according to Ackerman's theorem, all wheels are controlled to turn around the same steering instantaneous center. Based on this principle, we can calculate the steering moving trajectory circle of each wheel and get the change direction of the trajectory according to the moving direction.

When the robot is running, the trajectories of the wheels of the 2nd, 3rd, and 4th axles are all enveloped between the trajectory of the first and fifth axles. Therefore, only the trajectories of the first and fifth axles are needed to be calculated for predicting the whole robot's moving trajectory. In this way, the calculation model of the whole multi-axle robot's driving trajectories is simplified to the model of the two-axle robot, which greatly reduces the calculation work. This is shown in Figure 9.

The trajectory calculation theory of the first three steering control strategies is the same. For example, when the steering angle of the front wheel or rear wheel is zero in the first steering control strategy, the trajectory of the second or third can be obtained respectively. For the fourth and fifth steering strategies, they do not need to calculate the complex trajectory. In the fourth strategy, it only needs to detect whether there are obstacles in the circle trajectory of Figure 7. In the fifth strategy, it only needs to detect whether there are obstacles on the left or right side of the robot to determine the direction of the next step. Therefore, the trajectory line of the most typical first steering mode is only analyzed.

According to equation (3), the steering angles of wheels α_{11} , α_{12} , α_{51} , and α_{52} can be obtained. The angle unit is rad.

The longitudinal distance between the robot turning center O and the front wheel is (in mm)

$$L_{11} = \frac{L \tan(\alpha_{12})}{(\tan(\alpha_{12}) - \tan(\alpha_{52}))}. \quad (24)$$

Longitudinal distance from center O to the rear wheels is (in mm)

$$L_{55} = \frac{L \tan(\alpha_{52})}{(\tan(\alpha_{52}) - \tan(\alpha_{12}))}. \quad (25)$$

Lateral distance from center O to the left wheel of the 1st axle is (in mm)

$$BB = \frac{L_{11}}{\tan(\alpha_{12})}. \quad (26)$$

Coordinates of the turning center point O are

$$\begin{aligned} X_C &= xw_{11} + BB, \\ Y_C &= yw_{11} - L_{11}, \end{aligned} \quad (27)$$

where xw_{11} , yw_{11} are the x -coordinate and the y -coordinate of the left wheel of the 1st axle, respectively.

According to equations (4), (24)–(27), the coordinates of the k th point in the body's corner point trajectory curves corresponding to the j th wheel in the i th axle in Figure 9 can be obtained as follows:

$$\begin{aligned} XCC_{1j}[k] &= X_C - \text{sign}(\alpha_{12})R_{1j}\cos(\text{abs}(\alpha_{1j}) + k\Delta\theta) + \Delta Lx_{1j}, \\ XCC_{5j}[k] &= X_C - \text{sign}(\alpha_{12})R_{5j}\cos(\text{abs}(\alpha_{5j}) - k\Delta\theta) + \Delta Lx_{5j}, \\ YCC_{1j}[k] &= Y_C + R_{1j}\sin(\text{abs}(\alpha_{1j}) - k\Delta\theta) + \Delta Ly_{1j}, \\ YCC_{5j}[k] &= Y_C - R_{5j}\sin(\text{abs}(\alpha_{5j}) + k\Delta\theta) + \Delta Ly_{5j}, \\ \Delta\theta &= \frac{\pi}{180}, \\ j &= 1, 2, \\ k &= 1, 2, \dots, 30, \end{aligned} \quad (28)$$

where ΔLx_{ij} and ΔLy_{ij} are the compensation distance between four body's corner points and the corresponding four wheels in the x -axis and y -axis, respectively. $\Delta\theta$ is the iterative step size of the steering angles α_{12} and α_{52} .

3.3. Obstacle Avoidance Control Strategy Based on Trajectory Prediction. If there are no lidar points in the area enveloped by the steering track lines of each wheel of the robot as shown in Figure 10, then the robot can pass the current obstacle with this steering scheme, and it is a feasible steering scheme. Therefore, after the trajectory lines corresponding to the steering angles AF and AR are iteratively calculated and the relationship between the lidar points and the trajectory envelope region is calculated, if the lidar points are all outside the envelope region, then the obstacle avoidance steering scheme is feasible at this time. At this time, AF , AR , and V (the speed of the robot) are the obstacle avoidance control scheme for the next running of the robot.

Since the steering angles of other wheels of the robot are all calculated according to the front and rear virtual wheels' steering angles AF and AR , it only needs to calculate AF , AR , and speed V in the analysis of obstacle avoidance control. The steering angle of the actual wheels can be calculated by the robot on-board control unit ESP32S according to the algorithm of the multimode steering control strategy.

When the robot moves forward, as shown in Figure 9, the obstacle avoidance strategy based on trajectory prediction is shown in Figure 11.

3.3.1. Compare the Gap Width Dis[m] and the Size of the Body Width B and Preliminarily Select the Feasible Gap. First of all, from the obstacle lidar points at the front, left, and right sides, the gaps that meet the following driving conditions are found.

If the gap has two lidar points, as shown in Figure 9(a), the gap width Dis[m] between two adjacent lidar points P_m and P_{m+1} is compared with the size of the body width B . If $\text{Dis}[m] < B$, it indicates that the robot cannot go between two lidar points, so the gap is discarded. Then, the gap between the next set of adjacent lidar points is compared. Otherwise, it means that the robot may go between the two lidar points, which is the possible gap. Therefore, the program can continue to run Section 3.3.2.

If the gap has only one lidar point, as shown in Figure 9(b), it goes directly to Section 3.3.2.

The flowchart comparing the gap width strategy is shown in Figure 12.

3.3.2. Steering Angle Iteration. The steering angles of the front and rear virtual wheels are iterated in two layers in Figure 13. According to the position of the gap and the mechanical structure of the robot, if the gap is on the right front side, the range of the wheel steering angle is

$$\begin{aligned} AF &= 0 \sim 38^\circ, \\ AR &= -38 \sim 0^\circ. \end{aligned} \quad (29)$$

If the gap is on the left front side, then the range of the wheel steering angle is

$$\begin{aligned} AF &= -38 \sim 0^\circ, \\ AR &= 0 \sim 38^\circ. \end{aligned} \quad (30)$$

According to equations (3), (4), and (24)–(28), we can get the coordinates of the turning center points O, X_C , and Y_C .

For the lidar points at both sides of the gap, as shown in Figure 9, the one close to the longitudinal centerline of the robot body is the inside point, and the one far away from the longitudinal centerline of the robot body is the outside point. Then, the distance between the inside point of the gap and the turning center O is (in mm)

$$D_{in} = \sqrt{(X_C - P_m(x))^2 + (Y_C - P_m(y))^2} - \Delta L, \quad (31)$$

where $P_m(x)$ and $P_m(y)$ are the x and y coordinates of the inside point of the m th gap, respectively. ΔL is the

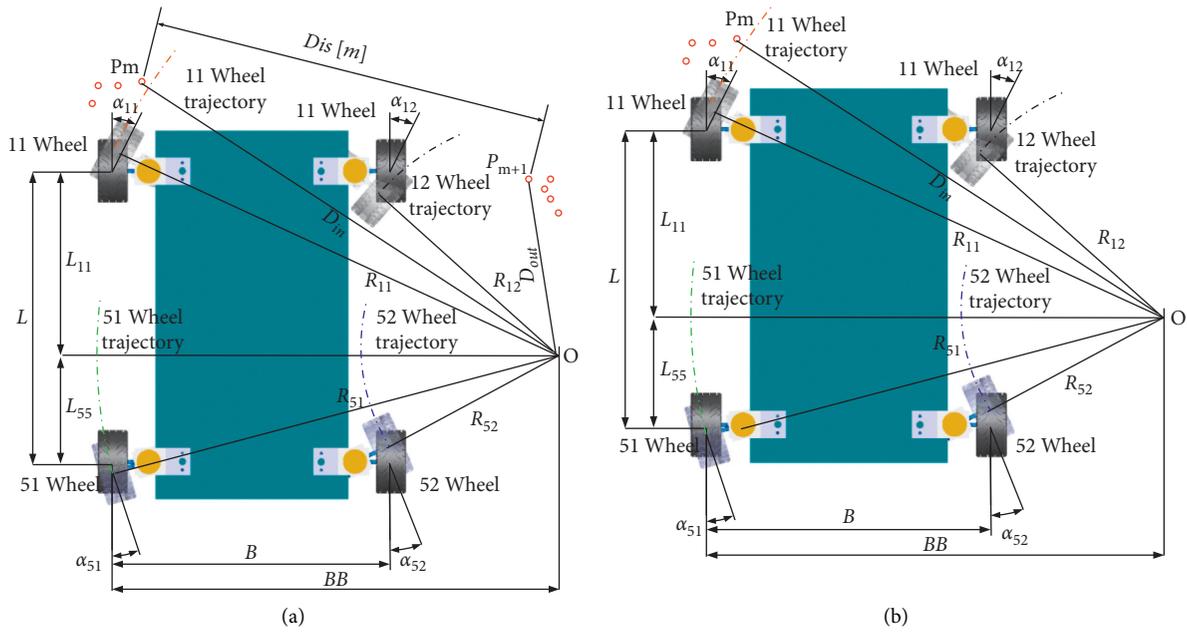


FIGURE 9: Obstacle avoidance geometric relation analysis diagram. (a) The gap has two end points, P_m and P_{m+1} . (b) The gap has only one point P_m .

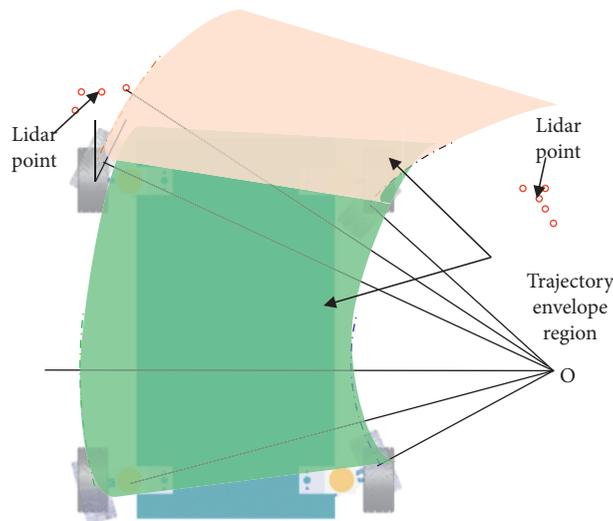


FIGURE 10: The envelope region formed by the trajectory line.

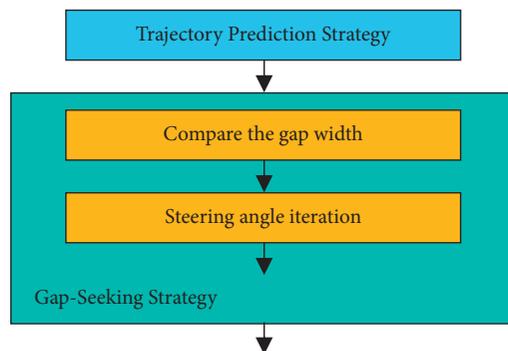


FIGURE 11: Obstacle avoidance strategy based on trajectory prediction.

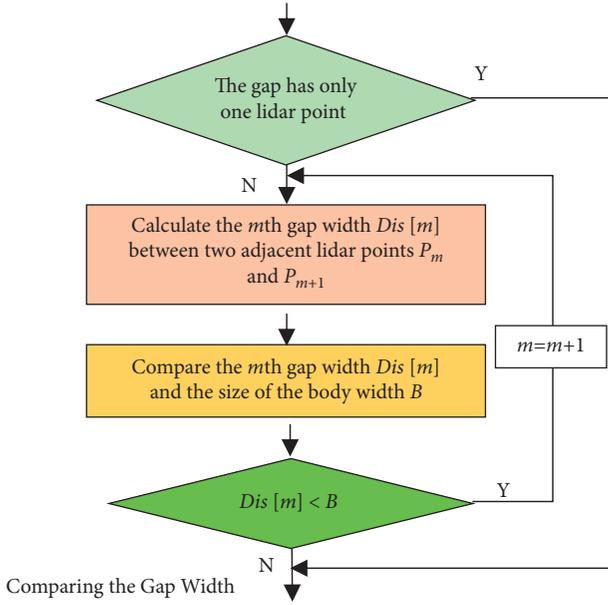


FIGURE 12: Flowchart of comparing the gap width strategy.

compensation value of the difference between the wheel and the body steering moving trajectory, and the unit is mm.

The calculation of the distance from the outer point of the gap to the turning center O can be divided into two cases:

- (1) Both points of the gap exist, as shown in Figure 9(a): Distance from the outside point of the gap to the turning center O is (in mm)

$$D_{\text{out}} = \sqrt{(X_C - P_{m+1}(x))^2 + (Y_C - P_{m+1}(y))^2} + \Delta L, \quad (32)$$

where $P_{m+1}(x)$ and $P_{m+1}(y)$ are the x and y coordinates of the outside point of the m th gap, respectively.

- (2) If the gap has only one point and no lidar point on the other side, as shown in Figure 9(b), we set

$$D_{\text{out}} = 0. \quad (33)$$

The front and rear wheel steering angles AF and AR are iteratively changed within the specified range of equation (29) or equation (30), until the gap width is larger than the robot body width, and the inside point of the gap is on the outside of the four track curves and the outside point is within the four track curves, as shown in Figure 9. Then, the following conditions are satisfied:

$$D_{\text{out}} < R_{ij} + \Delta L < D_{\text{in}}, \quad (i = 1, 5; j = 1, 2). \quad (34)$$

The angles of the front and rear virtual steering wheels meeting equation (34) are the steering angles for the robot through the gap.

If there are no front and rear virtual wheel steering angles AF and AR that meet equation (34) in the

whole iteration process, it means that the robot cannot go through the current gap, and it needs to continue to look for the gap that the robot can go through within the lidar point behind by carrying out Sections 3.3.1 and 3.3.2.

4. Control System

The whole control system of obstacle avoidance robot is divided into on-board control system program and an upper computer program.

4.1. On-Board Control System Program. The program is used to receive the data from the lidar module, calculate the information of obstacles, and display the relevant information through the touch screen. At the same time, the lidar information is sent to the upper computer through WiFi, and the robot obstacle avoidance steering and speed control instructions sent by the upper computer are received through WiFi. These instructions are decomposed into the steering angle and wheel speed of 10 wheels. Robot on-board control system procedures include WiFi initialization subroutine, lidar data reading and WiFi data sending subroutine, touch screen display subroutine, WiFi instruction reading subroutine, and steering and driving speed instruction execution subroutine.

4.1.1. WiFi Initialization Subroutine. It is used to set account, password, IP address, port number, and so on for WiFi communication, to realize wireless data transmission based on WiFi.

4.1.2. Lidar Data Reading and WiFi Data Sending Subroutine. It is used to read the information of obstacles produced by lidar through a serial communication port, get the angle and distance of obstacle points distribution, and transmit the information to the upper computer through WiFi.

4.1.3. On-Board Screen Display Subroutine. It is used to display lidar points data and robot running conditions in real time and set up the robot.

4.1.4. Instruction Reading Subroutine through WiFi. This program is used to read the information of steering and speed of the robot for obstacle avoidance sent by the upper computer, detect the status of WiFi communication, and control the robot to stop in time when the WiFi communication is interrupted.

4.1.5. Steering and Driving Speed Instruction Execution Subroutine. According to the information received about the speed and steering angles of the front and rear virtual wheels the steering angle of the ten wheels is calculated with equations (1)–(23).

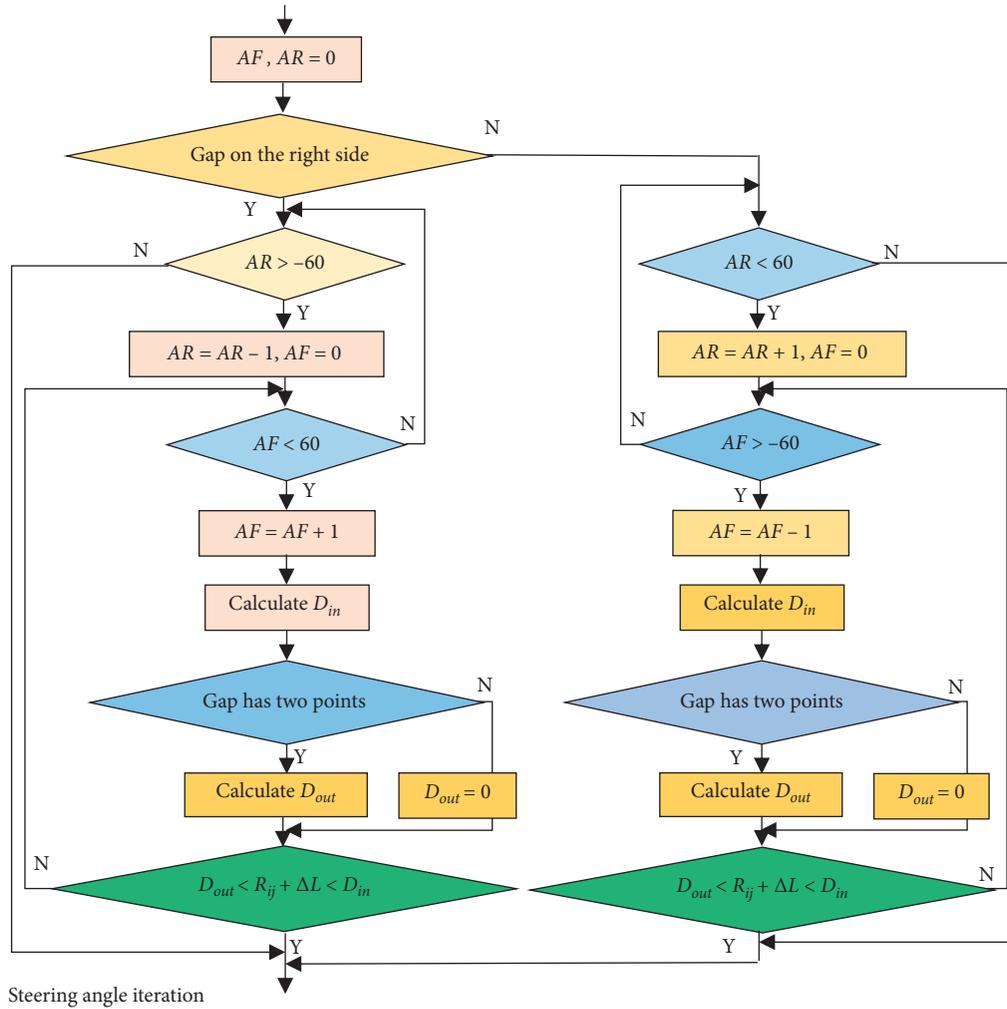


FIGURE 13: Two-level iterative flowchart.

4.2. *Upper Computer Program.* The upper computer program includes WiFi module, lidar data reading and decomposition module, travel path prediction-drawing module, and obstacle avoidance algorithm module.

4.2.1. *WiFi Module.* It includes WiFi initialization module, WiFi reading module, and WiFi instruction writing module, which is used to complete the wireless communication with the robot on-board control system based on UDP protocol.

4.2.2. *Lidar Data Reading and Decomposition Module.* According to the communication protocol of the lidar data, the data verification of the lidar points of obstacles and the calculation of the angle and distance of the lidar points are completed.

4.2.3. *Travel Path Prediction-Drawing Module.* This module mainly calculates the robot's next running trajectory according to the steering angle AF , AR of the front and rear wheels, and speed V with equation (28).

4.2.4. *Obstacle Avoidance Algorithm Module.* This module mainly uses equations (29)–(34) to iteratively calculate the moving trajectory corresponding to different steering angles of AF and AR , and then compares the relationship between lidar points and the trajectory envelope area. AF and AR corresponding to the absence of lidar points in the envelope area are selected as feasible obstacle avoidance steering schemes.

5. Experimental Verification

To verify the obstacle avoidance control strategy proposed above, experiments of Z-shaped obstacle distribution were carried out, as shown in Figure 14. All space among obstacles is smaller than the length of the robot body, it meets the layout requirements of the narrow space. During the whole autonomous driving process, the initial speed threshold of the robot is set to 108 (the threshold values of 0, 90, and 180 correspond to the robot's full speed backward, stop, and full speed forward, respectively; the relation between threshold and speed is linear), and the actual driving speed is about 256 mm/s.

The Z-shaped obstacle avoidance experiments were carried out 4 times in total, all of which were able to pass

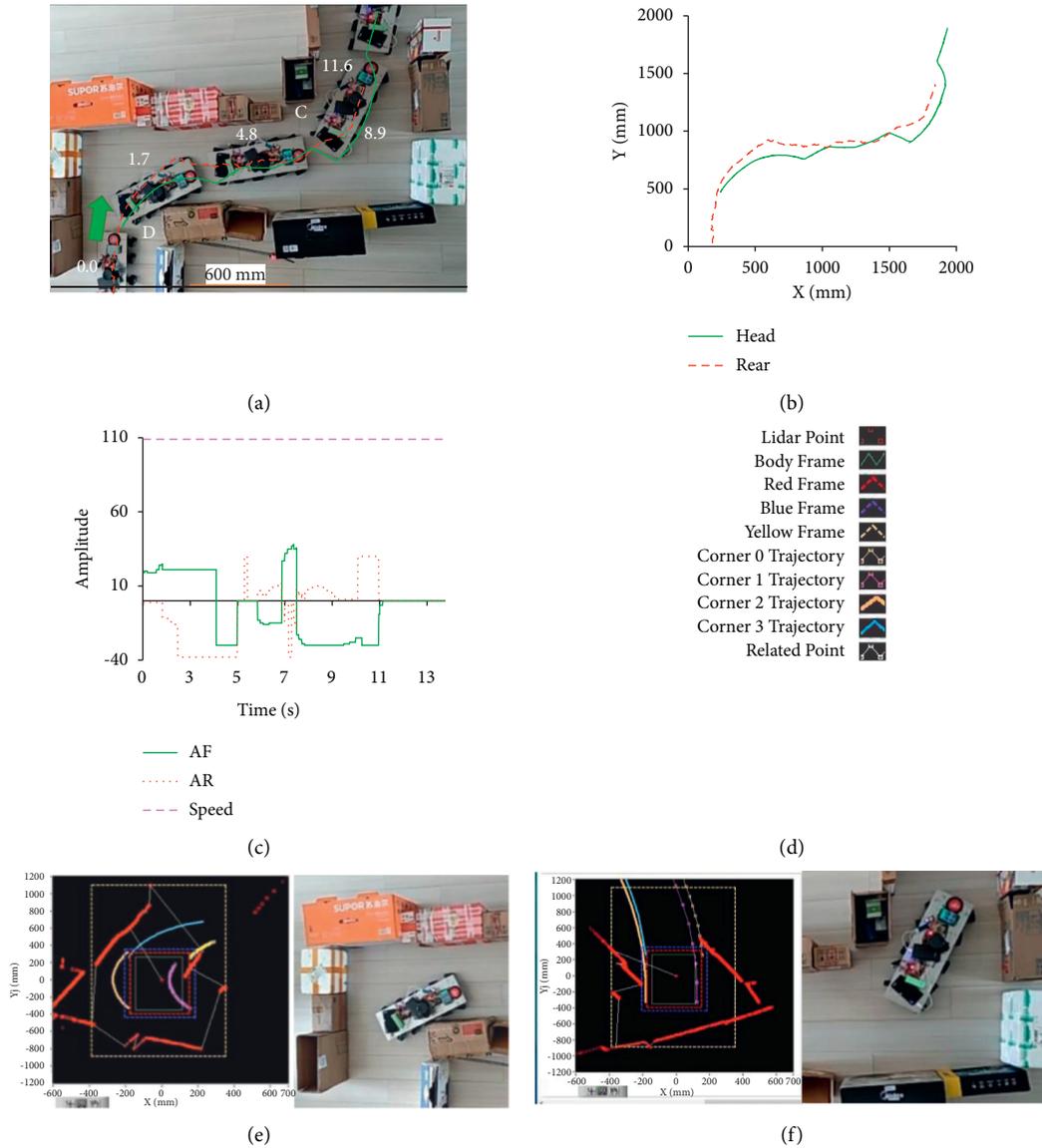


FIGURE 14: Z-shaped obstacle distribution experiment. (a) Robot trajectory diagram of Z-shaped obstacle avoidance. (b) Trajectory curves of head point *A* and rear point *B* of the body, and the position of *A* and *B* is shown in Figure 1. (c) Steering angle of *AF* and *AR* (the unit is degree), and speed control value *V* (zero speed and full speed are 90 and 180, respectively). (d) When the robot moves to the first corner *D*, the lidar map (the head of the robot is always upward in the figure) and the corresponding actual obstacle distribution map (the head of the robot is facing up to the right in the figure). (e) When the robot moves to the second corner *C*, the lidar map (the head of the robot is always upward in the figure) and the corresponding actual obstacle distribution map (the head of the robot is facing up to the right in the figure).

autonomously without hitting the obstacle. One representative experiment was selected for analysis. The area size is 2160 mm × 1580 mm, and the width of the driving channel is 560~680 mm. The distribution of obstacle passage and experimental results are shown in Figure 14.

The robot enters from the bottom-left corner of the figure. At the first corner *D*, the adverse-phase steering control strategy with the front wheel control variable *AF* turning 21° to the right and the rear wheel control variable *AR* turning 38° to the left was adopted to minimize the occupancy of the steering space, as shown in Figure 14(d). At the second corner *C*, the adverse-phase steering control strategy with the front wheel control variable *AF* turning 25°

to the left and the rear wheel control variable *AR* turning 30° to the right was adopted to realize the flexible steering mode with a small radius, as shown in Figure 14(e). During the whole driving process, the motion trajectory of body points *A* and *B* is shown in Figures 14(a) and 14(b), and the control output of steering variables *AF* and *AR* of the front and rear virtual wheel and speed *V* is shown in Figure 14(c). The robot can navigate out of the passage without colliding with any obstacles.

Through experiments, it is proved that the proposed control strategy can successfully achieve multimode autonomous obstacle avoidance in a narrow space of Z-shaped obstacles.

6. Conclusions

To improve the steering flexibility of multi-axle wheeled robot, a multimode steering control strategy based on front and rear virtual wheels is proposed in this paper, and the corresponding steering trajectory is analyzed. On this basis, by analyzing the relationship between the envelope area of the travel trajectory and the lidar points, the obstacle avoidance control strategy based on the travel trajectory prediction is proposed. Then, a five-axle all-wheel steering all-wheel driving wheeled robot was designed, the corresponding hardware and software control system was developed, and the obstacle avoidance experiment of Z-shaped obstacle distribution was carried out. The effectiveness of the proposed control strategy is verified.

In the next step, we will further optimize the obstacle avoidance control strategy and increase the speed of obstacle avoidance.

Data Availability

The data involved in the results of this study are accessible from the corresponding author based on the readers' request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (51005128), 2019 Teaching Research and Teaching Reform Project of Qingdao University of Technology (no. F2019-056), 2020 Teaching Research and Teaching Reform Project of Qingdao University of Technology (F2020-12), and 2020 Teaching Research and Teaching Reform Project of School of Mechanical and Automotive Engineering (2020-1).

References

- [1] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment," *International Journal of Advanced Robotic Systems*, vol. 17, pp. 1–15, 2020.
- [2] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, IEEE, Taipei, Taiwan, September 2003.
- [3] G. Lee and D. Chwa, "Decentralized behavior-based formation control of multiple robots considering obstacle avoidance," *Intelligent Service Robotics*, vol. 11, no. 1, pp. 127–138, 2018.
- [4] H. Liang, G. Liu, H. Zhang, and T. Huang, "Neural-network-based event-triggered adaptive control of nonaffine nonlinear multiagent systems with dynamic uncertainties," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 99, pp. 1–12, 2020.
- [5] K. H. Chiang, S. H. Tseng, Y. H. Wu, G. H. Li, and L. C. Fu, "Multisensor-based outdoor tour guide robot NTU-I," in *Proceedings of the SICE Conference*, pp. 1425–1430, IEEE, Chofu, Japan, August 2008.
- [6] D. Li, Z. Pan, and H. Deng, "Two-dimensional obstacle avoidance control algorithm for snake-like robot in water based on immersed boundary-lattice Boltzmann method and improved artificial potential field method," *Transactions of the Institute of Measurement and Control*, vol. 42, no. 10, pp. 1840–1857, 2020.
- [7] D. Chen, "Fuzzy obstacle avoidance optimization of soccer robot based on an improved genetic algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 6187–6198, 2019.
- [8] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and H. Kim, "Real-time obstacle avoidance of mobile robots using state-dependent Riccati equation approach," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1–13, 2018.
- [9] K. Zheng, F. Wu, and X. Chen, "Laser-based people detection and obstacle avoidance for a hospital transport robot," *Sensors*, vol. 21, no. 3, pp. 961–984, 2021.
- [10] Y. Tao, C. Chen, T. Wang et al., "A re-entry path planning method for service robots based on dynamic inver-over evolutionary algorithm," *Applied Sciences*, vol. 10, no. 1, pp. 1–15, 2019.
- [11] A. A. Aldair, M. T. Rashid, and A. T. Rashid, "Navigation of mobile robot with polygon obstacles avoidance based on quadratic bezier curves," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 43, no. 4, pp. 757–771, 2019.
- [12] A. Ghorbani, S. Shiry, and A. Nodehi, "Using genetic algorithm for a mobile robot path planning," in *Proceedings of the IEEE International Conference on Future Computer and Communication*, pp. 164–166, Kuala Lumpur, Malaysia, April 2009.
- [13] I. Engedy and G. Horvath, "Artificial neural network based local motion planning of a wheeled mobile robot," in *Proceedings of the IEEE International Symposium On Computational Intelligence And Informatics (CINTI)*, pp. 213–218, Budapest, Hungary, November 2010.
- [14] H. Liang, X. Guo, Y. Pan, and T. Huang, "Event-triggered fuzzy bipartite tracking control for network systems based on distributed reduced-order observers," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 6, pp. 1601–1614, 2021.
- [15] L. Liu, Y.-J. Liu, A. Chen, S. Tong, and C. L. P. Chen, "Integral Barrier Lyapunov function-based adaptive control for switched nonlinear systems," *Science China Information Sciences*, vol. 63, no. 3, Article ID 132203, 2020.
- [16] M. A. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing*, vol. 9, no. 3, pp. 1102–1110, 2009.
- [17] M. B. Montaner and A. Ramirez-Serrano, "Fuzzy knowledge-based controller design for autonomous robot navigation," *Expert Systems with Applications*, vol. 14, no. 1-2, pp. 179–186, 1998.
- [18] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 4, pp. 610–621, 2007.
- [19] H. Miao and Y.-C. Tian, "Dynamic robot path planning using an enhanced simulated annealing approach," *Applied Mathematics and Computation*, vol. 222, pp. 420–437, 2013.

- [20] S. Ahmadzadeh and M. Ghanavati, "Navigation of mobile robot using the pso particle swarm optimization," *Journal of Academic and Applied Studies*, vol. 2, pp. 32–38, 2012.
- [21] J. Yu, W. Hao, and W. Xin, "The research on obstacle avoidance of hospital ward inspection robots in complex environment," in *Proceedings of the 2016 Chinese Control and Decision Conference (CCDC)*, IEEE, Yinchuan, China, May 2016.
- [22] H. Xie, J. Gao, Z. Lin, C. Zhang, and C. Luan, "An improved obstacle avoidance method for robot based on constraint of obstacle boundary condition," in *Proceedings of the International Conference on Cybernetics*, IEEE, Chengdu, China, July 2018.
- [23] Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 194–206, 2017.
- [24] S. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, 2019.
- [25] C. Manuela, M. Saverio, N. Luca, P. Sabatini Silvio, and S. Fabio, "Bio-inspired active vision for obstacle avoidance," in *Proceedings of the 9th International Conference On Computer Graphics Theory And Applications*, pp. 505–512, Lisbon, Portugal, January 2014.
- [26] M. Yi-Jen, "Machine vision based sliding fuzzy-PDC control for obstacle avoidance and object recognition of service robot platform," *Intelligent and Fuzzy Systems*, vol. 28, pp. 1119–1126, 2015.
- [27] B. Hong, S. Xia, Y. Wang, and Y. Zhu, "Based on visual information for mobile robot dynamic obstacle avoidance," *Journal of Harbin Institute of Technology*, vol. 36, pp. 854–856, 2004.
- [28] D. S. Marco, R. Zandra, and G. Domenico, "Obstacle avoidance system for unmanned ground vehicles by using ultrasonic sensors," *Machines*, vol. 6, p. 18, 2018.
- [29] F. Wang, H. Liu, and F. Yu, "Strategy based on binocular stereo vision mobile robot obstacle avoidance," in *Proceedings of the 27 Chinese Control Conference*, pp. 457–461, Kunming, China, July 2008.
- [30] L. Chia-How and S. Kai-Tai, "Robust ground plane region detection using multiple visual cues for obstacle avoidance of a mobile robot," *Robotica*, vol. 33, pp. 436–450, 2015.
- [31] C. Ye and J. Borenstein, "Characterization of a 2-D laser scanner for mobile robot obstacle negotiation," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 2512–2518, Washington, DC, USA, May 2002.
- [32] T. R. Madhavan and M. Adharsh, "Obstacle detection and obstacle avoidance algorithm based on 2-D RPLiDAR," in *Proceedings of the 2019 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, January 2019.
- [33] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-D lidar," in *Proceedings of the 2015 IEEE International Conference on Information and Automation*, IEEE, Lijiang, China, August 2015.
- [34] M. Takahashi, K. Kobayashi, K. Watanabe, and T. Kinoshita, "Development of prediction based emergency obstacle avoidance module by using LIDAR for mobile robot," in *Proceedings of the 2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, pp. 561–564, IEEE, Kitakyushu, Japan, December 2015.
- [35] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," in *Proceedings of the Intelligent Vehicles Symposium*, pp. 44–49, IEEE, Istanbul, Turkey, June 2007.
- [36] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 971–985, 2012.
- [37] X. Wu, H. Chen, C. Chen et al., "The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method," *Knowledge-Based Systems*, vol. 196, Article ID 105201, 2020.
- [38] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems Man & Cybernetics*, vol. 19, pp. 1179–1187, 2002.
- [39] T. Xu, S. Zhang, Z. Jiang, Z. Liu, and H. Cheng, "Collision avoidance of high-speed obstacles for mobile robots via maximum-speed aware velocity obstacle method," *IEEE Access*, vol. 8, pp. 138493–138507, 2020.
- [40] S. Zaheer, T. Gulrez, and I. A. Thythodath Paramabath, "From sensor-space to eigenspace—a novel real-time obstacle avoidance method for mobile robots," *IETE Journal of Research*, vol. 2019, pp. 1–13, 2019.
- [41] C. Hu, B. Ning, M. Xu, and Q. Gu, "An experience aggregative reinforcement learning with multi-attribute decision-making for obstacle avoidance of wheeled mobile robot," *IEEE Access*, vol. 8, pp. 108179–108190, 2020.
- [42] W. Zheng, H. Wang, Z. Zhang, and X. Lu, "Hybrid position/virtual-force control for obstacle avoidance of wheeled robots using Elman neural network training technique," *International Journal of Advanced Robotic Systems*, vol. 14, pp. 1–14, 2017.
- [43] K. Akka and F. Khaber, "Optimal fuzzy tracking control with obstacles avoidance for a mobile robot based on Takagi-Sugeno fuzzy model," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 10, pp. 2772–2781, 2018.
- [44] W. Yuan, Z. Li, and C. Y. Su, "Multisensor-based navigation and control of a mobile service robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, pp. 2624–2634, 2019.
- [45] P. L. Kuo, C. H. Wang, H. J. Chou, and J. S. Liu, "A real-time hydrodynamic-based obstacle avoidance system for non-holonomic mobile robots with curvature constraints," *Applied Sciences*, vol. 8, pp. 1–25, 2018.
- [46] Y. Xie, X. Zhang, W. Meng et al., "Coupled fractional-order sliding mode control and obstacle avoidance of a four-wheeled steerable mobile robot," *ISA Transactions*, vol. 108, pp. 282–294, 2021.
- [47] R. Lin, H. Huang, and M. Li, "An automated guided logistics robot for pallet transportation," *Assembly Automation*, vol. 41, no. 1, pp. 45–54, 2020.
- [48] A. Prasad, B. Sharma, J. Vanualailai, and S. A. Kumar, "A geometric approach to target convergence and obstacle avoidance of a nonstandard tractor-trailer robot," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 13, pp. 4924–4943, 2020.
- [49] P. Zhang, L. Gao, and Y. Zhu, "Study on control schemes of flexible steering system of a multi-axle all-wheel-steering robot," *Advances in Mechanical Engineering*, vol. 8, pp. 1–13, 2016.