Humanoid Robot Vehicle Driving Navigation Based on Monocular Vision

Dongdong Chen, Yuan Wei, Zhigang Zhou, Zongwen Li, Peiyuan Chen

***Abstract*—This paper presents a novel approach aimed at enabling humanoid robots to navigate by utilizing vision for obstacle avoidance. A distance prediction model is proposed that incorporates visual recognition and range algorithms, as well as an optimized BP neural network model refined through an improved Whale Algorithm (CIWOA-BP). The primary objective is to accurately estimate obstacle distances to establish appropriate steering angles, thereby preventing collisions. The obstacles are identified by visual recognition, and the distance between the robot and obstacles are estimated, and then the distance is mapped to the distance between the car and the obstacle. The neural network prediction model takes distance inputs and wheel angles as outputs. Through extensive training and validation using humanoid driving data, a robust prediction model is successfully developed. The predicted wheel angles are translated into steering wheel and robot joint angles, enabling obstacle avoidance through joint angle adjustments. Experimental validation is conducted using the NAO humanoid robot and a micro electric vehicle within a Linux environment. Real-time vehicle control is achieved by interfacing the NAO's Qi system with the main controller, Jetson Nano. Results demonstrate the effectiveness of the proposed approach, with the distance prediction model exhibiting an error rate of approximately 6%, well within acceptable margins. Comparative studies show that the algorithm proposed in this paper has greater prediction accuracy and generalization capabilities.**

***Index Terms*—Humanoid robot, BP neural network, Vision-based navigation, Whale optimization algorithm, Obstacle avoidance**

I. INTRODUCTION

ROBOT driving involves the execution of specific driving tasks by a robot, encompassing vehicle navigation, obstacle avoidance, path planning, and control operations. This specialized branch of robotics is designed to enable autonomous vehicle operation in hazardous and challenging environments, thereby reducing the risk of human casualties. The driving system can be installed in the vehicle's cab without any modifications, ensuring a seamless integration that does not alter the vehicle itself [1, 2]. With the rapid advancement of autonomous robots, they are at the forefront of modern technology, and the research on the efficient robot operations and interoperability of robots is becoming increasingly crucial. The study of robotic operations is continually evolving, and effective interaction and performance of robots are vital across various fields, including safety, environmental monitoring, and disaster response [3].

Currently, research on robot driving is focused on several key areas. Firstly, in the domain of machine vision, Wang [4] *et al* have proposed the RL-CWtran Net, which employs a swing-transformer as its computer vision model. This system utilizes robot vision to capture and analyze the real-time movements and postural characteristics of athletes, providing immediate feedback. Additionally, Lukman Y [5] *et al* have developed a robotic system for vehicle navigation that detects road markings through visual image acquisition. This system processes distance information and feeds it back to the vehicle's front-wheel controller, enabling the car to drive straight along a designated route and to distinguish between straight paths and left or right turns by recognizing the trajectory's shape. Antonio Paolillo [6] *et al* have introduced a sensor-based response framework designed to address the central aspects of driving tasks. Their visual servoing scheme leverages features from road images to provide a reference angle for the steering wheel, ensuring that the vehicle remains centered in its lane. By sending references for the steering wheel and throttle to the robot control system, they successfully executed driving tasks with an anthropomorphic approach. Furthermore, they showcased a driving experience involving a real car and the humanoid robot HRP-2Kai, which has been partially employed in driving tasks during the DARPA Robotics Challenge.

The second area of focus is vehicle driving control. Naoto Mizutani [7] *et al.* proposed a speed control method for a continuously variable transmission (CVT) robot driver, taking vehicle dynamics into account. They designed a control system aimed at improving speed control performance, which was validated through vehicle operation tests conducted with the robot driver. Additionally, Wang [8] *et al.* introduced a speed tracking approach for driving robots based on dynamic fuzzy neural network (DFNN) direct inverse control. This method incorporates self-learning of the vehicle's longitudinal performance, enabling accurate speed tracking under various operating conditions. In addition, Shenyang Siasun Robot Automation

Manuscript received November 6, 2024; revised March 1, 2025. This work was supported in part by National Natural Science Foundation of China under Grant 51805149, and in part by the Master of Key Research and Development and Promotion of Henan Province under Grant 202102210317.

Dongdong Chen is a Postgraduate student of the College of Vehicle and Traffic Engineering, Henan University of Science and Technology. Luoyang 471000, China (e-mail: chendongdong30996@163.com).

Yuan Wei is an Associate Professor of the College of Vehicle and Traffic Engineering, Henan University of Science and Technology. Luoyang 471000, China (Corresponding author, phone: +86 18538809696 e-mail: 9905675@haust.edu.cn)

Zhigang Zhou is a Professor of the College of Vehicle and Traffic Engineering, Henan University of Science and Technology. Luoyang 471000, China (e-mail: hnmczzg@163.com)

Zongwen Li is a Postgraduate student of the College of Vehicle and Traffic Engineering, Henan University of Science and Technology. Luoyang 471000, China (e-mail: 1720090776@qq.com).

Peiyuan Chen is a Postgraduate student of the College of Vehicle and Traffic Engineering, Henan University of Science and Technology. Luoyang 471000, China (e-mail: 1755005661@qq.com).

Co., Ltd. has successfully developed a set of self-driving robot products, which have humanized design characteristics and autonomous learning ability, and can drive and control vehicles to perform various vehicle tests. This capability not only improves the accuracy of standard experiments, but also greatly reduces the test time while providing accurate and reliable data.

The third application area is vehicle performance testing. Compared to human drivers, robotic vehicle drivers offer several advantages, including shorter testing times, reduced costs, and enhanced accuracy. They alleviate the burden on human drivers, who often face uncomfortable and monotonous working conditions, resulting in more reliable and repeatable test outcomes [9]. With advancements in automobile design and manufacturing, a wide array of tests is required, such as emissions durability testing, fuel economy assessments, and high- and low-temperature environmental evaluations. The driving behavior of human drivers can introduce statistical and systematic errors in many automotive tests [10]. In contrast, driving robots can be installed directly in the cabs of various vehicles without modification, making them suitable for a range of tests, including fuel consumption and exhaust emissions. Utilizing driving robots in place of human drivers for emissions durability testing can significantly enhance test accuracy [11]. The fourth area of research involves rescue operations. Unmanned robots are capable of operating engineering vehicles for disaster relief tasks, including rescue missions during earthquakes and floods. Kim H [12] *et al.* developed an autonomous driving robot based on a robotic operating system that estimates the robot's location in underground mines, generates global maps using SLAM (Simultaneous Localization and Mapping), and identifies and avoids obstacles through sensors, enabling stable navigation. Guo [13] *et al.* created a firefighting robot equipped with an infrared vision fire identification system, designed for reconnaissance and rescue in complex environments and at fire scenes filled with toxic smoke. Shin S [14] *et al.* proposed a disaster response vehicle drive system based on humanoid robots. To enhance the robot's ability to substitute for human activities during natural and man-made disasters, safe vehicle operation at disaster sites is a critical skill for rescue robots. Consequently, they developed a humanoid robot driving system capable of guiding vehicles through unknown obstacles, even under challenging communication conditions such as latency and power outages.

Currently, most driving robots are designed to perform specific, repetitive tasks or are integrated with the vehicle, relying on vehicle calibration. Other systems depend on multiple sensors—including cameras, various radars, ultrasonic sensors, GPS/GNSS receivers, and inertial measurement units—to perceive their surroundings through sensor fusion and make driving decisions. Thus, we propose a method that relies solely on camera-based perception to gather visual information and perform visual ranging, enabling the robot to navigate the vehicle and avoid obstacles effectively. Meanwhile, the main contributions of this paper are as follows:

1. The robot's camera is used as a sensing tool to perceive and obtain the surrounding environmental information, and the precise visual distance measurement of a single obstacle is realized by using the monocular distance measurement technology of the robot.

2. An efficient distance prediction model is proposed. When the reference distance is known, the distance between the robot and other obstacles is estimated by an image processing algorithm.

3. A BP neural network prediction model optimized based on the improved whale algorithm was successfully constructed, and the prediction from distance to turning angle is realized, which greatly improves the prediction accuracy.

The structure of this paper is as follows: In section II, the geometric ranging model of NAO robot is deeply discussed. In section III, the construction process of BP neural network prediction model based on improved whale algorithm is described in detail. Section IV focuses on the establishment of robot driving model. In section V, detailed simulation and experimental verification of the proposed method are carried out. In section VI, summarizes the content of this study, points out the shortcomings of the study, and puts forward the prospect of the future research direction.

II. GEOMETRIC RANGING MODEL OF NAO ROBOT

A. Humanoid robot NAO vision system

The NAO robot is an artificial intelligence humanoid robot developed by Aldebaran Robotics. Standing at a height of 58 cm, it is one of the most widely used humanoid robots globally. The NAO robot has 25 degrees of freedom, over 100 sensors, and is equipped with two cameras and two speakers located in its head. Its appealing design allows for a wide range of flexible movements. The specific body structure of the NAO robot is illustrated in Fig. 1.

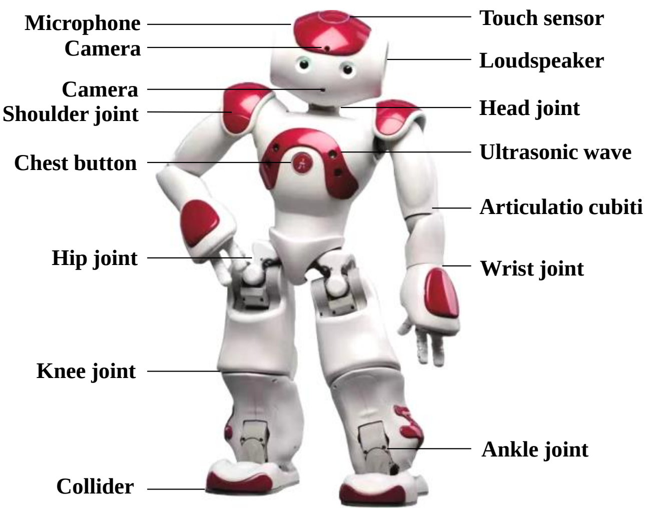


Fig. 1. NAO robot structure

The NAO robot gathers environmental information using two high-definition cameras mounted on its head, with the camera angles adjusting according to the robot's movement and head orientation. The NAO robot uses a monocular vision system [15]. During operation, the NAO vision system utilizes the two cameras sequentially rather than simultaneously. The lower camera is designed for close-range visual recognition, while the upper camera is intended for longer-distance scanning. Each camera has a field of view of 40 degrees. The system calculates the relative coordinates of a target in relation to the NAO robot based on the field of view from both cameras and the visual environment.

B. The Monocular ranging model of NAO Robot.

Since only one camera of the NAO robot can be used at a time, it employs a monocular ranging method to determine the position information of a target. First, the monocular ranging model of the NAO robot is established [16], as illustrated in Fig. 2.

In the above figure, the camera is located at point O', and its projection on the ground is located at point I. The camera's optical axis OO" passes through point O'. The projection of any point P(x, y) in space onto the image coordinate system is P'(x', y'). As the NAO robot moves, the

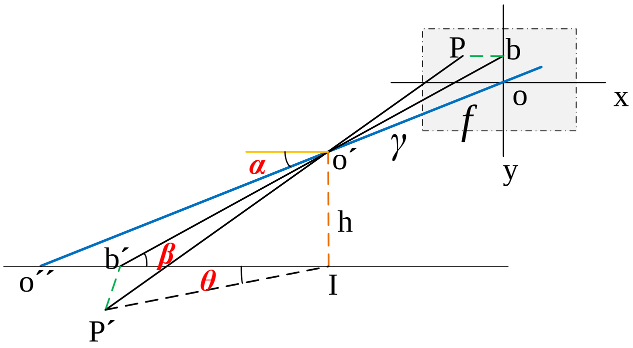


Fig. 2. NAO robot monocular ranging model

height h of the camera changes continuously with the joint movement. The camera's pitch angle is α. The perpendicular line of OO", which passes through point P' as the main optical axis, intersects with point b' and then passes through point b' as a straight line through point O' to intersect with the coordinate axis of the image coordinate system at point b. At this point, the distance d between point P' and point O' in the world coordinate system, as well as the angle θ between point P' and the projection OI of the main optical axis OO", can be obtained. The calculation formula of each parameter is shown as follows:

$$d = \frac{h}{\tan\beta} \tag{1}$$

$$\beta = \alpha + \gamma \tag{2}$$

$$\tan\gamma = \frac{bo}{f} \tag{3}$$

Thus, the direction angle and distance can be calculated as follows:

$$\theta = \frac{y \times h}{f \times d \times \cos\gamma \times \sin\beta} \tag{4}$$

$$d = \frac{h}{\tan(\alpha + \arctan\frac{y}{f})} \tag{5}$$

As can be seen from the above equation, to construct a monocular ranging model for the NAO robot and obtain depth information of the images collected by the robot, it is necessary to know the pitch angle α of the robot camera, the height h of the camera, and the focal length f [15]. The pitch angle of the camera can be obtained through a fixed API function of the robot. The height of the robot in an upright state is 58.3cm, and the focal length of the robot is f=372.10501. It is well-known that the humanoid robot NAO's head joints can perform pitch and yaw movements. The pitching angle of the NAO robot can be obtained through a Python program by importing the ALProxy module and creating an ALMotion proxy object. The getAngles method is used to obtain the head pitch Angle α of NAO robot.

C. Visual analysis and the establishment of distance prediction model

Robots are increasingly utilized to assist with daily activities in both home and work environments. Vision-based robotic systems can analyze their surroundings, enabling them to independently detect, recognize, and locate objects [17]. In this paper, we focus primarily on identifying objects through their contour features in images, allowing us to obtain information such as the position and outer boundary of these objects. To enhance our description of these features, we introduce a pixel coordinate system, as illustrated in the Fig. 3. In this system, (u, v) represents the position information of the object contour in the image, W is the pixel width, and H is the pixel height.

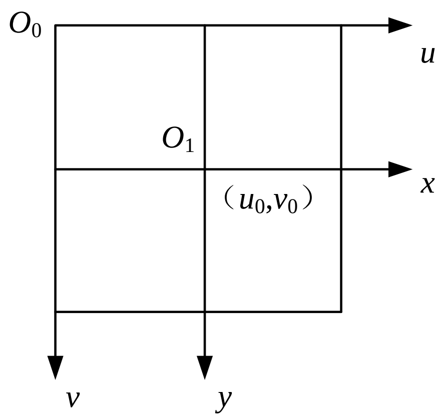


Fig. 3. Pixel coordinate diagram

In the image processing of the acquired picture, the Canny edge detection algorithm [18] is employed to highlight and extract the outlines of objects within the image. This technique enables the calculation of the pixel coordinates and pixel size of each object's outline. The results before and after image processing are presented in Fig. 4.

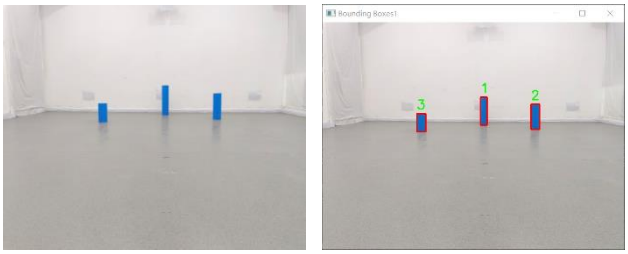


Fig. 4. Image recognition before and after comparison

First of all, an obstacle placement in a three-dimensional spatial environment is proposed in this paper, in which five obstacles have different sizes and positions, but similar shapes as shown in the following Fig. 5.

As shown in Fig. 6, we want to derive from the distance d_i of the reference object to the distance d_{j+n} (n=0, 1, 2, 3⋯) of the other non-reference objects. Area S_i of the reference

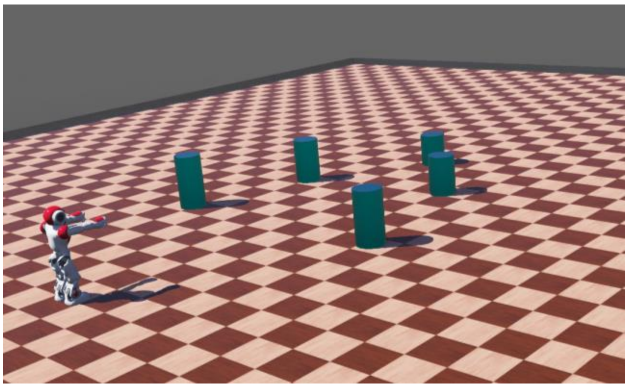


Fig. 5. Spatial position diagram of robot and obstacles

object N_i, which we choose as the reference object, area S_j of the object N_j other than the reference object, pixel area px_i of the reference object N_i, pixel area px_j of the object N_j other than the reference, pixel area $px_{i'}$ of the offset object $N_{i'}$ of the reference object N_i, pixel area $px_{j'}$ of the offset object $N_{j'}$ of the object N_j. Fig. 7 is the distance prediction model analysis diagram.

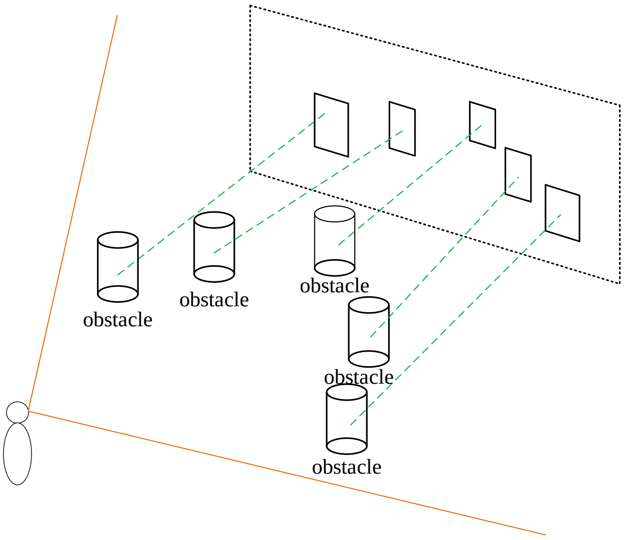


Fig. 6. Three-dimensional side view and projection of NAO robot and obstacle in space

From N_i to N_j:
The ratio of the areas of objects N_j and N_i is:

$$k_0 = \frac{S_j}{S_i} \tag{6}$$

The equivalent pixel area of object N_i is:

$$px_i' = px_i \bullet k_0 \tag{7}$$

The ratio of the pixel areas of objects N_j and N_i is:

$$k_0' = \frac{px_j}{px_i'} \tag{8}$$

The pixel ratio of object N_i is:

$$k_i = \frac{S_i}{px_i} \tag{9}$$

The pixel ratio of object N_j is:

$$k_j = \frac{S_j}{px_j} \tag{10}$$

Therefore, the horizontal pixel distance between object N_i and object N_j is:

$$p_{ii'} = x_i - x_j - w_j \tag{11}$$

The horizontal distance between object N_i and object N_j is:

$$l_{ii'} = k_i \bullet p_{ii'} \tag{12}$$

If the diameter of the object is taken into account and the central position of the object is taken as the reference point, the actual horizontal distance between the object N_i and the object N_j is:

$$l_{ii'}' = l_{ii'} + 2r \tag{13}$$

$$l_{ij}' = l_{ii'}' \tag{14}$$

Where, r is the radius of the object.

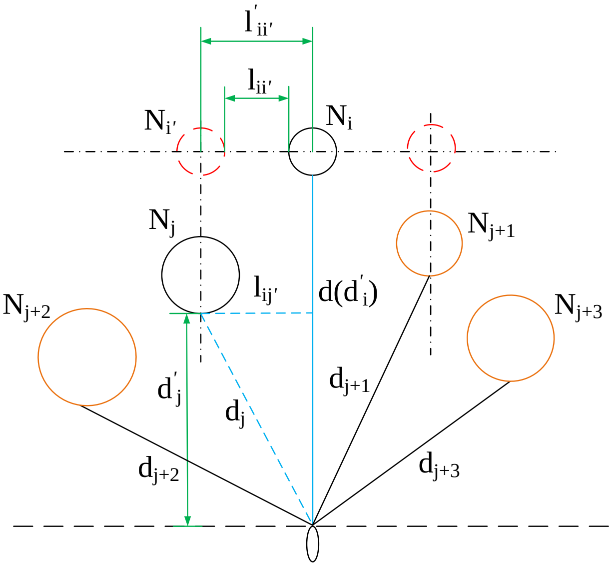


Fig. 7. Distance prediction model analysis diagram

Due to the characteristics of the human visual system and psychological perception, people in the observation of two or more of the same objects in the distance will have a sense of near big and far small, that is, in the picture, closer to the observer of the object looks bigger, and far from the observer of the object looks smaller visual phenomenon. As shown in Fig. 8. O_1 represents the observation position, and there are two identical objects M_1 and M_2 directly in front of the observer. Due to the characteristics of the human visual system and psychological perception, the observed objects M_1 and M_2 will appear as in Fig. 8 in the case of AB and CD, (this figure is the top view). in the actual observation of the situation, the object M_1 will be obscured by M_2.

In the NAO robot monocular ranging model, the reference distance d measured by the robot from the obstacle can be equated to O_1O_3, as well as the perpendicular distance d_i' between the object N_i and the camera. From the above (8), can be seen that the pixel area of the object is equivalent to the length of line segment AB and CD in Fig. 8, so we can get:

$$AB = \sqrt{px_i'} \tag{15}$$

$$CD = \sqrt{px_j} \tag{16}$$

$$\frac{AB}{CD} = 1/\sqrt{k_0'} \tag{17}$$

According to the similar triangle principle, so we can get:

$$O_1O_3 \bullet AB = O_1O_2 \bullet CD \tag{18}$$

therefore,

$$O_1O_2 = 1/\sqrt{k_0'} \bullet O_1O_3 \tag{19}$$

Then, we can find the vertical distance of object M from the position of the robot camera.

$$d_j' = \frac{1}{\sqrt{k_0'}} \bullet d_i' \tag{20}$$

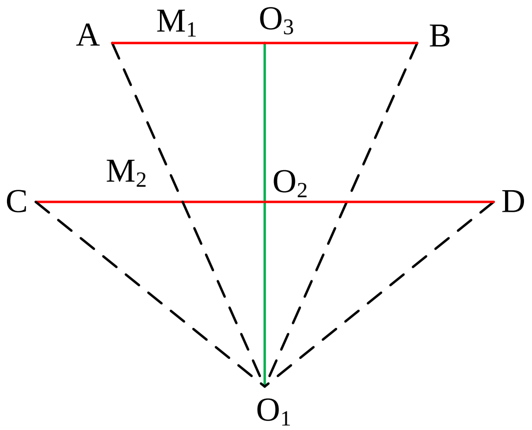


Fig. 8. Vertical distance prediction model analysis diagram

According to the distance prediction model, the distance between the NAO robot and the object N_j can be obtained.

$$d_j = \sqrt{(l_{ii'}')^2 + (d_j')^2} \tag{21}$$

As shown in Fig. 7 and Fig. 8, according to the distance prediction model, we can deduce the distance d_{j+n} (n=0, 1, 2, 3⋯) between the robot and the remaining non-reference object from the reference distance d_i.

D. The identification of critical obstacles

Through the distance estimation model, we can estimate the distance between the robot and multiple obstacles N_{j+n} (n=0, 1, 2, 3...). However, only the obstacles around the car will have a great impact on the path of the NAO robot driving car to avoid obstacles. Therefore, this article takes obstacle N_i as a reference point, selects the nearest obstacles on its left and right sides, and forms a group of key obstacles Q_i, Q_j, Q_{j+1}. Therefore, the distance d_i, d_j, d_{j+1} between the car and these key obstacles will be an important basis for determining the next obstacle avoidance action of the vehicle.

In the pixel coordinate system shown in Fig. 3, u represents the horizontal axis. By processing the image, we can obtain the lateral position information of each obstacle in the image. Assuming that the position information of the reference obstacle N_i is represented by u_i, the position information of the remaining obstacles N_{j+n} can be represented as u_{j+n}. Therefore, we can determine the key obstacles according to the following formula.

$$\Delta u_{j+n} = u_{j+n} - u_i \tag{22}$$

Where, $\triangle u_{j+n}$ represents the distance between the obstacle N_{j+n} and N_i. When the value of $\triangle u_{j+n}$ is closer to 0, it means that the obstacle N_{j+n} is closer to N_i. Then, it can be determined to find the two obstacles Q_j and Q_{j+1} that are closest to the reference base obstacle N_i in the middle distance on the left and right sides. The judgment method is as follows:

$$\begin{array}{ll} Q_j = N_{j+n} & \Delta u_{j+n} < 0 \wedge \left|\Delta u_{j+n}\right|_{min} \\ Q_{j+1} = N_{j+n} & \Delta u_{j+n} > 0 \wedge \left|\Delta u_{j+n}\right|_{min} \end{array} \tag{23}$$

Where, $|\Delta u_{j+n}|_{min}$ means that the absolute value of Δu_{j+n} is the least, that is, Δu_{j+n} is closest to 0.

At this point, we can determine a group of key obstacles Q_i, Q_j, Q_{j+1} and their distance from the robot d_i, d_j, d_{j+1}.

III. THE CONSTRUCTION OF BP NEURAL NETWORK PREDICTION MODEL OPTIMIZED BY CIWOA

A. BP neural network

After collecting the humanoid driving data, we need a neural network prediction model to connect the data information. The BP neural network, consisting of input layer, hidden layer and output layer, is one of the most widely used neural network models [19]. In traditional BP neural networks, the weights and biases are initialized randomly, requiring adjustment during training to minimize errors. Typically, the gradient descent method is employed to update these parameters; however, this approach is highly sensitive to the initial weights, which can adversely affect algorithm performance and convergence speed. To address this issue, this paper proposes the use of an improved Whale Optimization Algorithm (CIWOA) to optimize the initial weights and biases of BP neural networks. This optimization aims to reduce errors and enhance the stability and accuracy of the network. In addition, the introduction of optimization algorithm significantly helps the neural network to achieve better results in searching for the global optimal solution, thus enhancing the comprehensive performance and prediction ability of the neural network [20]. The structure of the BP neural network is illustrated in Fig. 9. In this paper, we use the distance between the car and the obstacles as the inputs and the wheel Angle as the output.

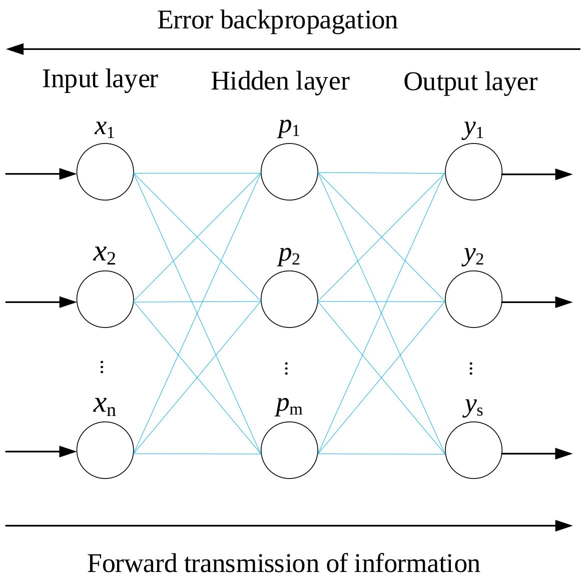


Fig. 9. BP neural network structure

B. Collection of humanoid driving data

To better analyze and simulate human driving behavior, and to gain a deeper insight into drivers' intentions and habits, we aimed to enhance the robot's ability to generalize from human examples and improve its adaptability in unfamiliar situations. We employed a remote-controlled vehicle-testing platform and selected 20 participants (SD=2, Avg=25) to conduct simulated driving experiments across various scenarios. A total of 246 sets of data were collected, with five-sixths of the data being used as training data and one-sixth as validation data. The driving behavior data collected from these experiments serves as training data of the neural network, which is helpful to train a more robust neural network model. The experimental setup is illustrated in Fig. 10. We measure the car's movement trajectory, the distance of obstacles to the car at different times, and the car's movement information, such as wheel Angle and steering wheel Angle.

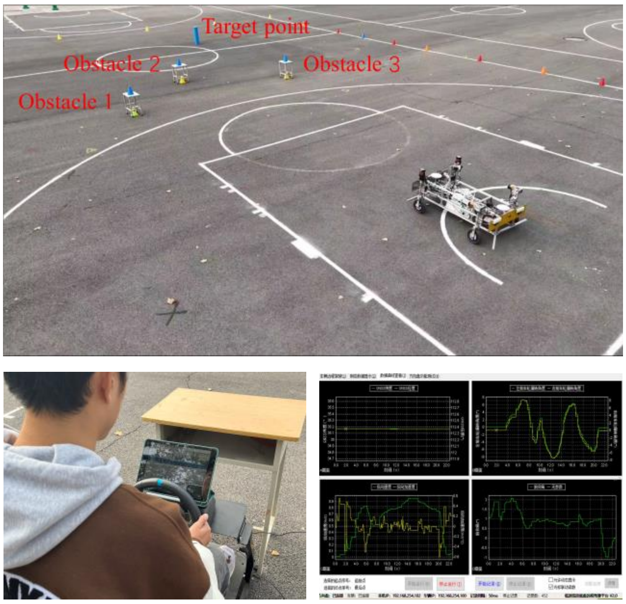


Fig. 10. Collection of humanoid driving data

C. *Improved Whale Optimization Algorithm*

The Whale Optimization Algorithm (WOA) is a novel group intelligence optimization method introduced by Mirjalili and Lewis [21] in 2016, inspired by the group hunting behavior of humpback whales. The algorithm encompasses three key phases: rounding up prey, bubble net feeding, and searching for prey. Its three population updating mechanisms operate independently, allowing for separate control of global exploration and local development during the optimization process, eliminating the need to manually set various parameter values. This enhances efficiency and simplifies application. Compared to other population-based optimization algorithms, WOA features a novel structure with fewer control parameters, demonstrating superior optimization performance across numerous numerical and engineering problems, outperforming algorithms like the ant colony and particle swarm optimization and other intelligent optimization algorithms [22]. Although WOA performs well in many areas, it suffers from the same problems as other population-based intelligent optimization algorithms, WOA also suffers from the problems of slow convergence speed and the tendency to fall into local optimum prematurely[23].

This paper proposes improvements in the following areas:

1) Cubic chaotic mapping initializes the population

In intelligent optimization algorithms, the diversity of the initial population plays a crucial role in determining the convergence rate and accuracy of the algorithm. The traditional WOA typically generates its initial population using random methods, which can lead to challenges in ensuring sufficient diversity. This approach often fails to guarantee a uniform distribution of search agents within the search space, resulting in decreased search efficiency during the iterative process. In the case of poor population diversity, the initial population directly affects the accuracy and speed of convergence and the overall performance of the algorithm. Compared to random searches, chaotic series are more likely to thoroughly explore the search space while maintaining population diversity [24]. Chaotic mapping, characterized by its randomness, ergodicity, and regularity, is frequently employed to generate the initial population for algorithms or as a disturbance during the optimization process [25].

Common chaotic mapping methods include Tent mapping, Logistic mapping, and Circular mapping. In this paper, we utilize Tent mapping to enhance the search process by leveraging the randomness, ergodicity, and regularity of chaotic variables. This approach helps the algorithm to bypass local optima, maintain population diversity, and improve global search capabilities [26]. The mathematical expression for Tent mapping is as follows:

$$x_{n+1} = f(x_n) = \begin{cases} x_n / \alpha & x_n \in [0, \alpha) \\ (1 - x_n)/(1 - \alpha) & x_n \in [\alpha, 1] \end{cases} \tag{24}$$

Where, α is a control parameter, the value range is $0< \alpha <1$.

The population initialization method based on Tent chaotic sequences enhances the diversity of whale populations, which in turn improves the search capability and convergence speed of the algorithm. Fig. 11 illustrates the overall distribution of the initialization with α=0.499. Fig. 12 presents the histogram of the initialization distribution.

2) Improvement of the nonlinear convergence factor *a*

In group iterative intelligent optimization algorithms, balancing global exploration and local exploitation is crucial for ensuring the algorithm's global optimization. An imbalance between these two aspects can lead to slow convergence rates or premature convergence during iterations. Specifically, if the balance is not well maintained, the algorithm may experience inefficient performance throughout the iterative process [27].

Analysis of the WOA reveals that the convergence factor *a* plays a vital role in balancing global exploration and local exploitation capabilities. However, in WOA, the convergence factor *a* decreases linearly from 2 to 0 as the number of iterations increases, which can slow the algorithm's iteration speed.

To enhance the search accuracy and efficiency of the algorithm, this paper proposes a non-linear adjustment of the convergence factor with respect to the number of iterations [28]. In the early stages of iteration, *a* is set to a higher value to accelerate convergence and strengthen global exploration. As the iterations progress, *a* decrease non-linearly, becoming smaller in the later stages to improve local search performance. This approach aims to achieve an optimal balance between global exploration and local optimization.

The specific formula is as follows:

$$\begin{cases} a = 1 + \cos(\pi \bullet \dfrac{t}{T})^{u} & t \le 0.5T \\ a = 1 + \cos(\pi \bullet \dfrac{t}{T})^{u} & t > 0.5T \end{cases} \tag{25}$$

Where, a is the convergence factor, t is the current iteration number, and T is the maximum iteration number. As shown in Fig. 13, the variation trend of a with the number of iterations t when u=0.8 and u=1.0 are shown.

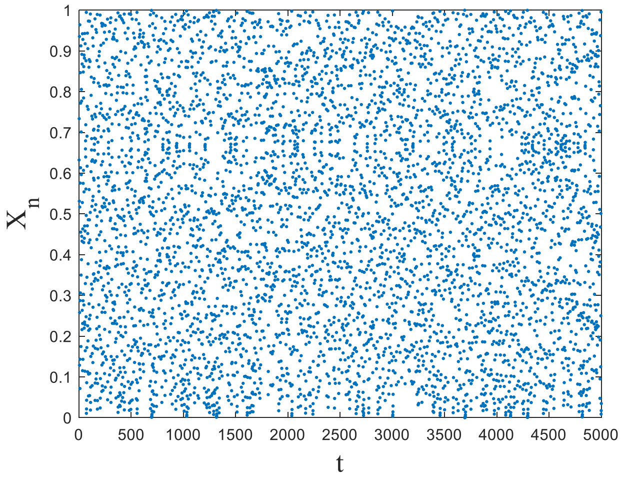


Fig. 11. Scatterplot

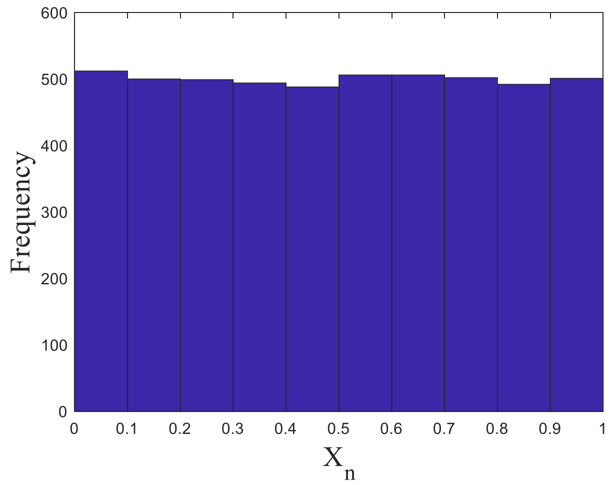


Fig. 12. Bar chart

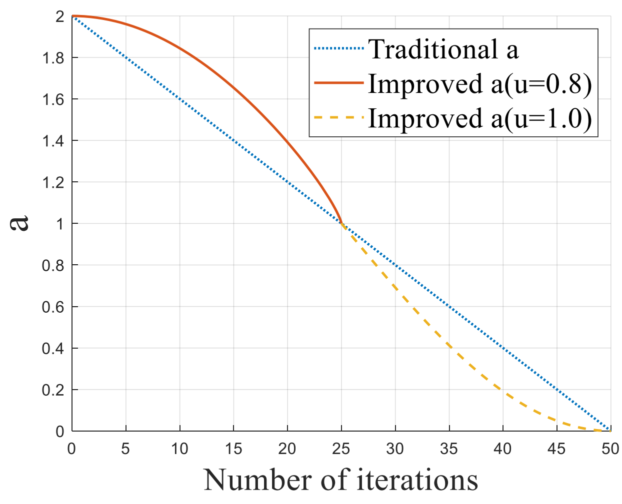


Fig. 13. Convergence factor comparison diagram

3) Spiral update position model

Whales hunt using a bubble spiral, the shape of which is determined by a specific parameter b. In the WOA, this parameter b is typically set to a constant value of 1. However, this fixed parameter b results in overly uniform movement during the optimization process, making it difficult for the algorithm to escape local optima in later stages. To address the limitations of traditional search algorithms' movement strategies, this paper proposes a variable logarithmic spiral position update strategy [29]. This strategy dynamically adjusts the shape of the logarithmic spiral, allowing the search range of individual whales to vary as the number of iterations increases. This approach effectively enhances the search precision of the whale population and prevents it from becoming trapped in local optima, leading to improved optimization results. Fig. 14 shows the variation trend of b with the number of iterations t. The improved method for calculating the updated spiral parameter is outlined as follows:

$$b = k * \cos(\pi * (1 - t / T)) \tag{26}$$

Where, k is the adjustment coefficient, which is a random number between 0 and 5, t represents the current population's iteration number iteration number of the population; T is the maximum number of iterations.

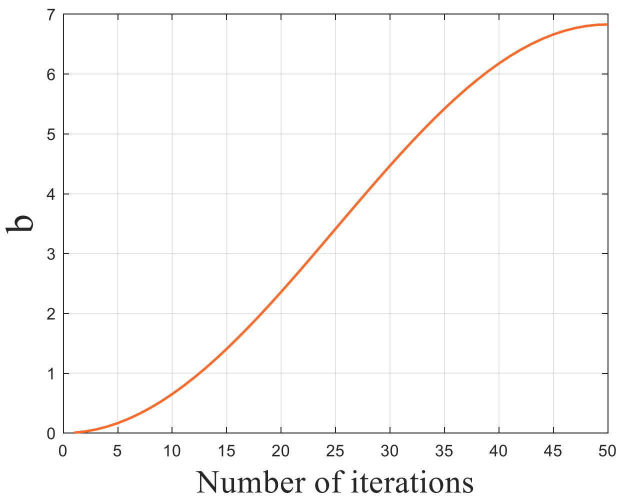


Fig. 14. Dynamic spiral renewal coefficient curve

4) Adaptive adjustment of weight ω

Inertia weight is a crucial parameter in the WOA, and using a constant inertia weight can diminish the algorithm's efficiency, hindering its global optimization capabilities. An appropriate weight value can significantly enhance the algorithm's optimization ability. To address the issues of slow convergence and low precision in the WOA, this paper introduces an adaptive weight factor to optimize the algorithm. In the early stages of iteration, a larger inertia weight is employed to ensure robust search performance, allowing for a broader search range and promoting global exploration. This slower convergence helps the algorithm escape local optima. Conversely, in the later stages, a smaller inertia weight is utilized, facilitating local exploitation and enabling slightly faster convergence, thereby enhancing local search capabilities [30]. Consequently, this paper proposes a new adaptive weight method. As the whale approaches the prey, a smaller adaptive weight is applied to adjust its position, improving local search efficacy. The adaptive weight formula is as follows:

$$\omega = \sin(\pi * (t / 2T)) \tag{27}$$

Where, t represents the number of iterations of the current population; T is the maximum number of iterations.

After the introduction of the adaptive adjustment strategy, the current formula of the whale individual adaptive adjustment weight shrinkage encircling update position and adaptive adjustment weight spiral update position is as follows:

$$X(t+1) = \omega X^*(t) - AD \tag{28}$$

$$X(t+1) = \omega X^*(t) + D_p e^{bl} \cos(2\pi l) \tag{29}$$

By incorporating an adaptive weight factor ω, the balance between exploration and exploitation can be dynamically adjusted at different stages of the algorithm. This approach reduces the risk of premature convergence to local optima, facilitating a quicker convergence to the global optimum or a near-optimal solution. Additionally, it enhances the local optimization capabilities of individual whales. As a result, the WOA can sustain its strong global search capabilities while simultaneously improving its convergence speed.

IV. ROBOT DRIVING MODEL

A. Mapping between steering wheel Angle and wheel angle

Since the NAO robot controls the steering wheel by adjusting the joint angles of its arms, it is essential to map the wheel angle α derived from the prediction model to the steering wheel Angle θ of the car and subsequently to the joint Angles of the robot's arms. Assuming the steering wheel angle ranges from -60° to 60° and the allowable turning range for the car's wheels is from -30° to 30°. Therefore, the mapping relationship between the wheel angle α and the steering wheel angle θ can be established as follows:

$$\theta = \frac{60}{120}\alpha = 0.5\alpha \tag{30}$$

B. The inverse solution of NAO robot

Aiming at the arm structure of NAO robot, D-H coordinate method is used to model the arm kinematics. The connecting rod coordinate system of each joint is established according to the relevant parameters of both arms of NAO, as shown in Fig. 15. The Inverse solution of NAO.

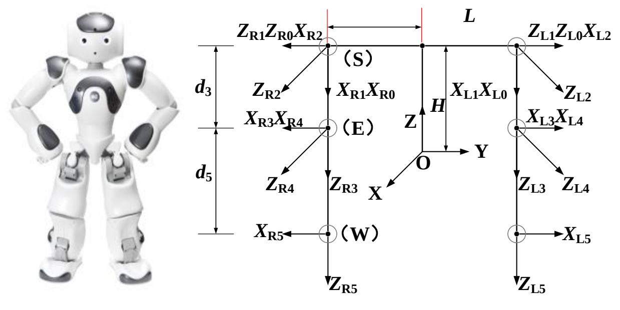


Fig. 15. D-H linkage coordinate system

In the Fig. 15, point O is the origin of the Cartesian coordinate system of NAO robot, d_3 is the length of the upper arm, d_5 is the sum of the length of the forearm and the length of the hand, L represents the offset of the origin of the relative coordinates of the shoulder joint in the Y-axis direction, H represents the offset of the origin of the relative coordinates of the shoulder joint in the Z-axis direction. According to the structural parameters of the NAO robot, d_3=90mm, d_5=108mm.

The D-H parameters of the right arm of the NAO robot are shown in the table below. Since the left arm is symmetrical to the right arm, the D-H parameters of the left arm correspond to the D-H parameters of the right arm. In this section, we will use the right arm as an example for our discussion.

TABLE I
D-H PARAMETERS OF THE RIGHT ARM OF THE NAO

i	θ_{ri}	α_i/ (°)	a_i/mm	d_i/mm	range/ (°)
1	θ_{r1}	90	0	0	-119.5～119.5
2	θ_{r2}	-90	0	0	-76～18
3	θ_{r3}	-90	0	d_3	-119.5～119.5
4	θ_{r4}	90	0	0	2～88.5
5	θ_{r5}	-90	0	d_5	-104.5～104.5

Taking the right arm as an example, the forward kinematics of the NAO arm is solved according to the D-H parameters of the arms. Since all joints of the NAO arms are rotating joints, the influence of moving joints is not considered. The homogeneous coordinate transformation matrix T_i represents the pose relationship of the connecting rod relative to the previous connecting rod, and the expression is as follows:

$$\begin{aligned} T_i &= \mathrm{Rot}(Z,\theta_i)\mathrm{Trans}(0,0,d_i)\mathrm{Trans}(\alpha_i,0,0)\mathrm{Rot}(X,\alpha_i) \\ &= \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i C\alpha_i & \alpha_i C\theta_i \\ S\theta_i & C\theta_i C\theta_i & C\theta_i S\alpha_i & \alpha_i C\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{31}$$

Where, Rot is the rotation transformation matrix, and Trans is the translation transformation matrix. $C\theta_i$ is the $\cos(\theta_i)$, $S\theta_i$ is the $\sin(\theta_i)$, $C\alpha_i$ is the $\cos(\alpha_i)$, and $S\alpha_i$ is the $\sin(\alpha_i)$.

The transformation matrices from the right arm end-effectors to the center of the robot are denoted by T_{RN}, respectively. The specific expressions are as follows:

$$T_{RN} = T_{RSO} R_{RX}\, {}^{1}_{6}T_R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_{LX} \\ R_{21} & R_{22} & R_{23} & P_{LY} \\ R_{31} & R_{32} & R_{33} & P_{LZ} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32}$$

Where, T_{RSO} - right arm shoulder joint relative robot center matrix, R_X - rotation matrix around X axis, ${}^{1}_{6}T_R$ - right arm from wrist to shoulder coordinate transformation matrix, their expressions are as follows:

$$T_{RSO} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -L \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

$$R_{\mathrm{RX}}=\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&-1&0&0\\0&0&0&1\end{bmatrix} \tag{34}$$

$$_{6}^{1}T_{R}=T_1T_2Rot(Y,\pi/2)T_3T_4T_5Rot(Y,\pi/2)T_6 \tag{35}$$

The hands of the NAO are simple toggle grippers that do not affect the kinematics calculations, so the calculation of the hand is ignored. To solve the joint Angle of the NAO robot, the solution of the right arm joint Angle is [31]:

$$\begin{aligned}
\theta_{\mathrm{R1}}&=\arctan\left(\frac{(d_5+l_h)R_{32}-P_{\mathrm{RZ}}+H}{P_{\mathrm{RX}}-(d_5+l_h)R_{12}}\right)\\
\theta_{\mathrm{R2}}&=\arctan\left(\frac{(d_5+l_h)R_{22}-L-P_{\mathrm{RY}}}{(d_5+l_h)(R_{32}S_{\mathrm{R1}}-R_{12}C_{\mathrm{R1}})-S_{\mathrm{R1}}(P_{\mathrm{RZ}}-H)+C_{\mathrm{R1}}P_{\mathrm{RX}}}\right)\\
\theta_{\mathrm{R3}}&=\arctan\left(\frac{R_{12}S_{\mathrm{R1}}-R_{32}C_{\mathrm{R1}}}{R_{12}C_{\mathrm{R1}}S_{\mathrm{R2}}+R_{22}C_{\mathrm{R2}}-R_{32}S_{\mathrm{R1}}S_{\mathrm{R2}}}\right)\\
\theta_{\mathrm{R4}}&=\arccos(R_{12}C_{\mathrm{R1}}C_{\mathrm{R2}}-R_{22}S_{\mathrm{R2}}-R_{32}S_{\mathrm{R1}}C_{\mathrm{R2}})\\
\theta_{\mathrm{R5}}&=\arctan\left(\frac{-R_{11}C_{\mathrm{R1}}C_{\mathrm{R2}}+R_{21}S_{\mathrm{R2}}+R_{31}S_{\mathrm{R1}}C_{\mathrm{R2}}}{R_{13}C_{\mathrm{R1}}C_{\mathrm{R2}}-R_{23}S_{\mathrm{R2}}-R_{33}S_{\mathrm{R1}}C_{\mathrm{R2}}}\right)
\end{aligned} \tag{36}$$

This paper takes the Cartesian origin of NAO as the center to determine the relative position relationship between NAO and vehicle. The center of the steering wheel is placed on the XZ plane of the Cartesian coordinates, so when the steering wheel Angle is 0, the NAO arms are symmetrical. According to the structure and arrangement of the steering wheel, the position relationship between the center of the steering wheel and the center of the robot can be determined. Fig. 16 shows the position relationship between the center of the steering wheel and the center of the robot.

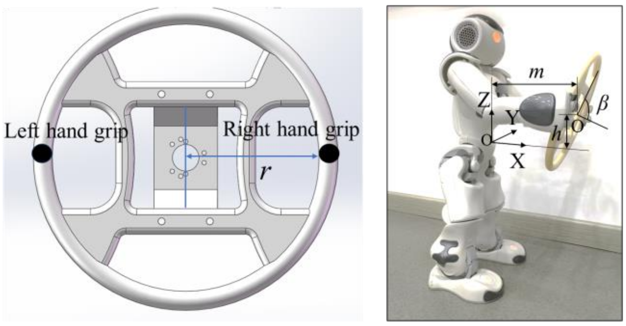


Fig. 16. Steering wheel relative to the robot center position

In the figure, m and h represent the deviation of the steering wheel center from the robot center coordinate in the X and Z axis directions respectively. r is the radius of the steering wheel; β represents the rotation Angle of the steering wheel axis relative to the central coordinate system Y-axis. According to the position relationship between the steering wheel and the robot center, the pose matrix T_0 of the steering wheel center relative to the robot center can be determined with the following expression:

$$T_0=\mathrm{Trans}(c,0,h)\mathrm{Rot}(Y,\beta)$$
$$=\begin{bmatrix}\cos\beta&0&-\sin\beta&c\\0&1&0&0\\\sin\beta&0&\cos\beta&h\\0&0&0&1\end{bmatrix} \tag{37}$$

According to relevant parameters, the pose matrix of the right hand grip on the steering wheel relative to the robot center is determined:

$$T_{\mathrm{RN}}=T_0\mathrm{Rot}(Z,\varphi)\mathrm{Trans}(Y,-r)\mathrm{Rot}(X,-\frac{\pi}{2})\mathrm{Rot}(Z,-\frac{\pi}{2})$$
$$=\begin{bmatrix}\sin\beta&\cos\beta\cos\varphi&-\cos\beta\sin\varphi&c+r\cos\beta\sin\varphi\\0&\sin\varphi&\cos\varphi&-r\cos\varphi\\\cos\beta&-\sin\varphi\cos\varphi&\sin\beta\sin\varphi&h-r\sin\beta\sin\varphi\\0&0&0&1\end{bmatrix} \tag{38}$$

Therefore, when we get the steering wheel Angle φ, we can get T_{RN} according to (38), and then according to (36), we can get the joint Angle θ_{Ri} of robot.

V. SIMULATION AND EXPERIMENT

A. *Verification and error analysis of distance prediction model*

To verify the validity of the distance prediction model proposed above, experiments were conducted in a controlled environment. The camera captured images of three obstacles positioned directly in front of it, at distances ranging from 3 to 10 meters. The reference distance between the camera and a baseline obstacle is known and serves as a basis for deriving the distances to the other two obstacles using the prediction model. With the camera held in a fixed position (ensuring the reference distance remains constant), the experiments accounted for the three obstacles in various positions and configurations, including differences in size. The validation results are summarized in the following table.

From the analysis of the table above, it is evident that at different reference distances, the distances derived from the prediction model exhibit a certain degree of error compared to the actual distances. The errors come from the different size and position of obstacles, camera shooting Angle and shooting accuracy. These factors can significantly affect the images captured, impacting subsequent image processing

TABLE II
PROGNOSTIC MODEL ERROR ANALYSIS

Reference distance(cm)	Error 1(%)	Error 2(%)	Aggregate error (%)
300	3.44	3.28	3.360
400	3.11	3.17	3.140
500	1.61	2.89	2.250
600	2.49	2.14	2.315
700	3.68	2.69	3.185
800	3.93	2.84	3.385
900	3.14	2.84	2.990
1000	2.42	2.68	2.550

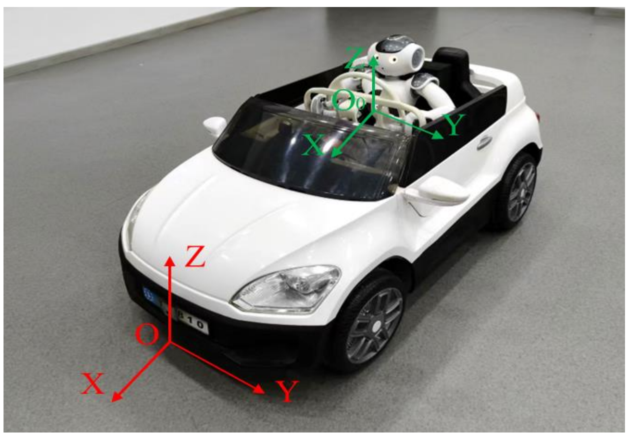


Fig. 17. Coordinate systems for vehicles and robots

TABLE III
BENCHMARK FUNCTION

Function	Dimensionality	Domain of definition	Theoretical optimum	Number of iterations Before improvement	Number of iterations After improvement	Fitness value Before improvement	Fitness value After improvement
$f_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100，100]	0	41	8	1.2545	0.2635
$f_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	30	[-1.28，1.28]	0	84	65	0.0035	8.8318e-05
$f_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	[-5.12，5.12]	0	58	36	0	0
$f_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_1x_2)}{b_i^2+b_1x_3+x_4}\right]^2$	4	[-5，5]	0.1484	47	33	7.2592e-04	5.0903e-04

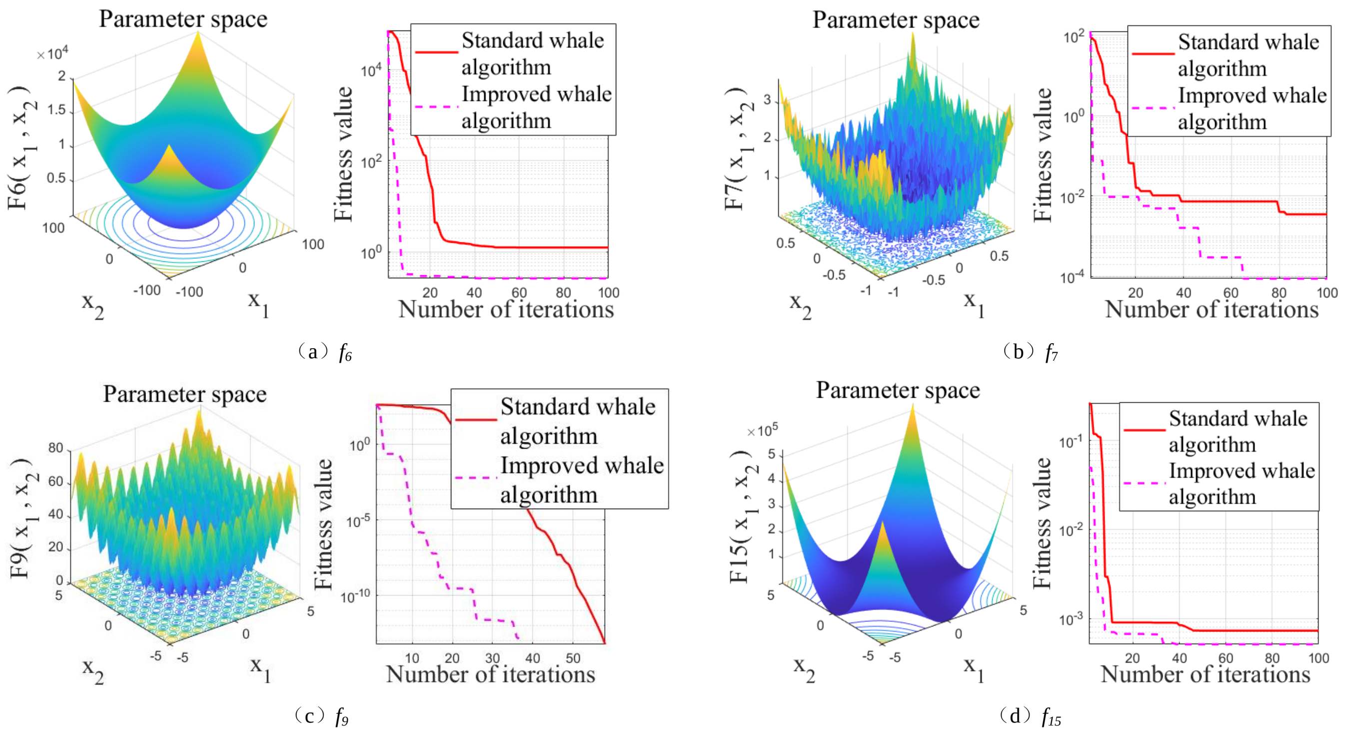


Fig. 18. Convergence curve comparison diagram

and resulting in variations and errors in the predictions. As indicated in the table, the overall integrated error is approximately 6% or less, which meets the experimental requirements.

Since the distances measured by the NAO robot are based on its Cartesian coordinate system, they must be converted into the coordinate system of the car when the robot is driving. Fig. 17 shows the relationship between the Cartesian coordinate system of the robot and the coordinate system of the car. Wen the NAO robot uses vision to obtain the reference distance d through the traditional distance measurement model, and subsequently maps the positional relationship between the robot and the vehicle to the reference distance $d'=d-O_0O$, then the distance d' can serve as the reference distance in the distance prediction model.

B. Improved validation of Whale Optimization Algorithm

To verify the effectiveness of the CIWOA, this paper randomly selects 4 out of 23 benchmark test functions for validation. By comparing it with the traditional WOA, the effectiveness of the improved Whale algorithm is verified. The 4 randomly selected benchmark test functions are shown in Table III.

To verify the optimization capabilities of the CIWOA and the traditional WOA, this paper conducted 50 simulation tests on both algorithms using Matlab. The convergence curves of the CIWOA and the traditional WOA on the benchmark test functions are shown in the Fig. 18.

The comparative results presented in Fig. 18 indicate that the CIWOA reaches the optimal solution in fewer iterations than the traditional WOA, demonstrating superior test results. This suggests that the CIWOA effectively avoids local optima while approaching the theoretical optimal solution with reduced error.

C. Validation and analysis of BP neural network model optimized by improved whale algorithm

The traditional Backpropagation (BP) neural network, as one of the most used shallow neural networks in machine learning, adjusts the network weights and thresholds through the error backpropagation algorithm and gradient descent method, thereby correcting the errors in the training and testing results of the network until the output values are consistent with the theoretical values.

Therefore, the distances between the vehicle and the three obstacles are used as input layer variables, and the steering angle of the wheels is used as the output layer variable. Thus, the number of input layer nodes c is 3, and the number of output layer nodes e is 1.

In this paper, we use an empirical formula to calculate the number of hidden layer neurons during the training process. We train and compare different numbers of neurons to derive the optimal number of neurons in the hidden layer. The specific expression of the empirical formula is as follows:

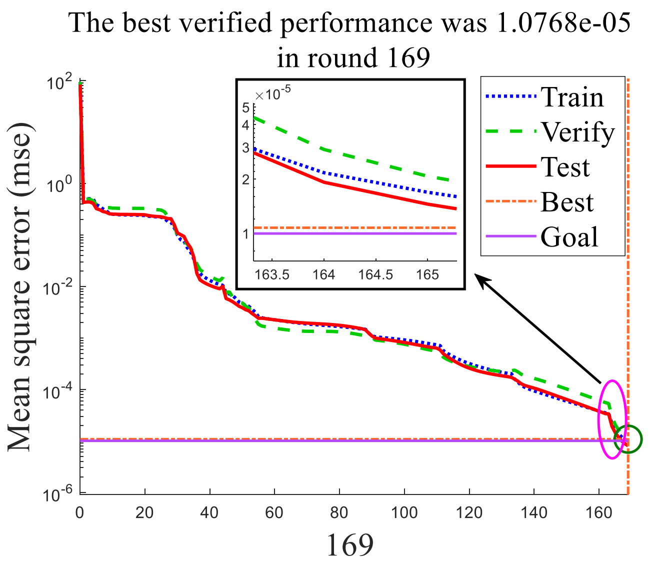
$$d = \sqrt{c+e} + \sigma \tag{39}$$

Where, σ is the constant between 1 and 10.

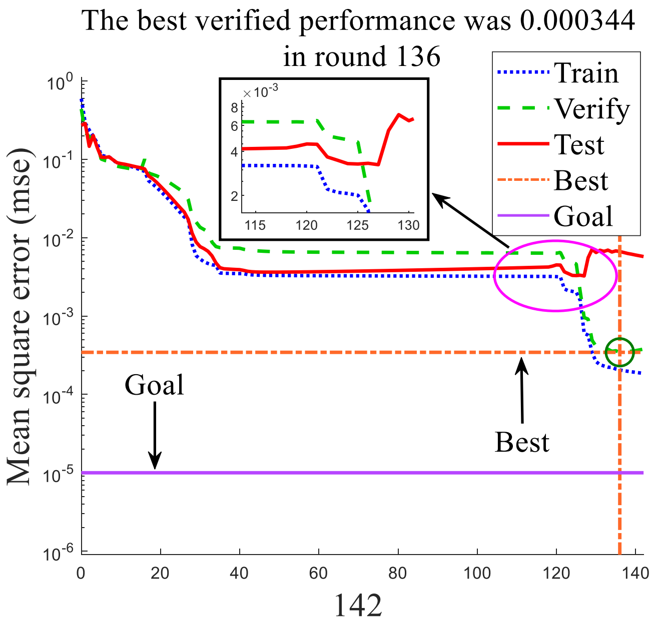
To avoid overfitting and getting stuck in local minima, and minimizing the training error is minimal and the network is more stable, according to the empirical formula, the network structure of the model is determined to be 3-7-1. In addition, we set the training precision, training times, learning rate, and iteration number for the BP neural network structure. The training precision is set to 10^{-5}, the training times is set to 1000, the learning rate is set to 0.01, and the maximum number of iterations is set to 200. The mean square error (MSE) is selected as the error function of the neural network. The mean square error function is expressed as:

$$MSE = \frac{1}{N}\left(\sum_{i=1}^{N} y_s - r_s\right)^2 \tag{40}$$

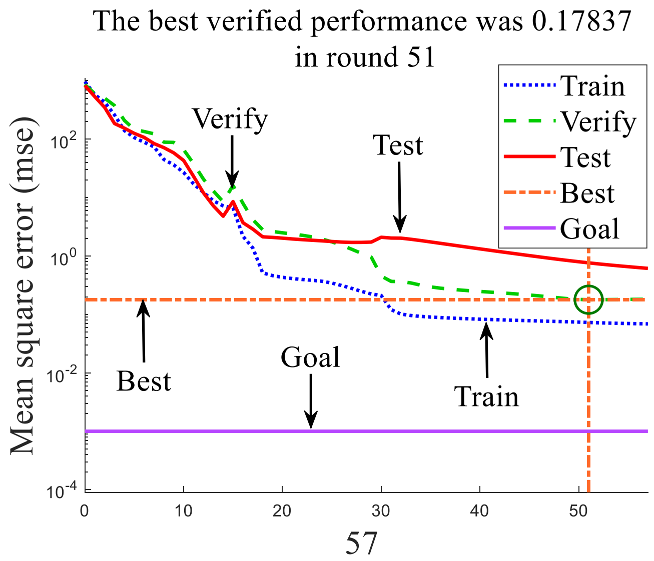
To verify the feasibility and the effectiveness of the prediction model proposed in this paper, the proposed method is compared with the traditional BP neural network, Grey Wolf Optimized BP neural network (GWO-BP) and Particle Swarm Optimized BP neural network (PSO-BP).



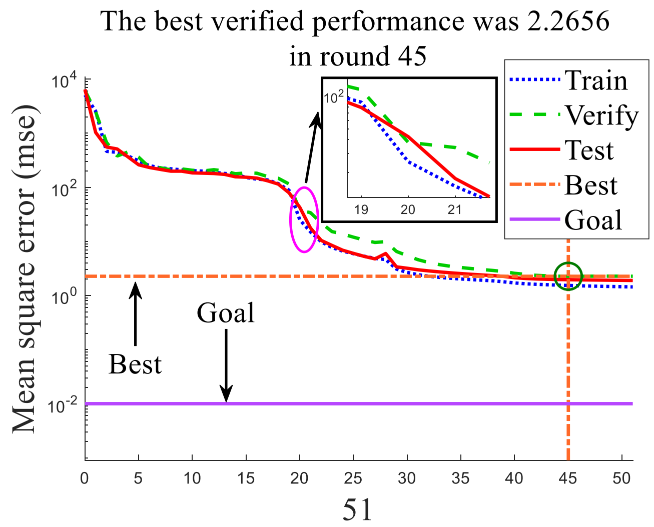
(a) The prediction model proposed in this paper



(b) Traditional BP neural network prediction model



(c) Grey wolf optimized BP neural network prediction model



(d) Particle swarm optimization BP neural network prediction model

Fig. 19. Training process of different prediction models

The training data is randomly shuffled to avoid concentration or regularity, which could interfere with the authenticity of the verification results.

In this paper, Matlab2022b is used to train the neural network. The training process of the proposed method and the traditional BP neural network, GWO-BP and PSO-BP are shown in Fig. 19.

The analysis of Fig. 19 shows that the proposed prediction model achieves the lowest MSE value, with the training, validation, and testing curves closely aligned. The MSE is a widely used metric for assessing the difference between predicted and actual values; a lower MSE indicates that the algorithm's predictions are closer to reality, reflecting higher model accuracy. Furthermore, the close proximity of the training, validation, and testing curves, combined with the low MSE, suggests that the algorithm exhibits strong generalization capabilities, enabling it to make accurate predictions on unseen data. This also indicates that the algorithm effectively avoids overfitting during the training process, which is essential for maintaining prediction accuracy in practical applications. Overall, these findings demonstrate that the model is well-suited for addressing real-world problems and can deliver reliable prediction results.

As shown in Fig. 20, the BP neural network optimized by the CIWOA achieves the best fitness value in fewer iterations. Moreover, its optimal fitness value is significantly lower than those of BP neural networks optimized by the GWO and PSO. This indicates that the improved optimization algorithm can converge quickly while effectively exploring the solution space. It also demonstrates adaptability by dynamically adjusting the search strategy, resulting in a lower fitness value while maintaining rapid iteration, showcasing strong global search capabilities and convergence performance.

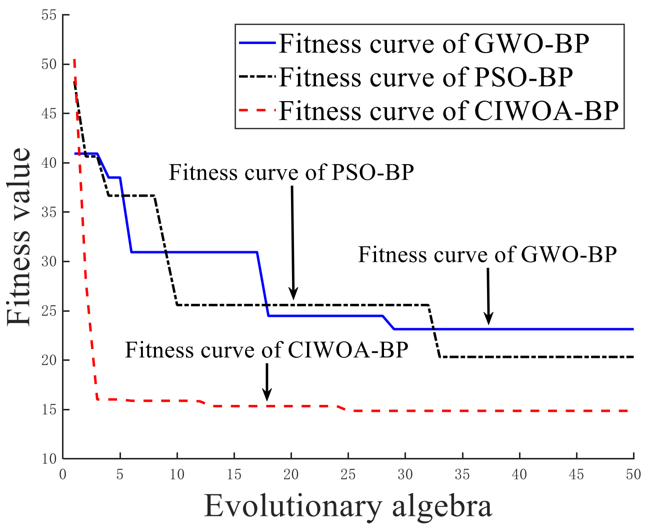
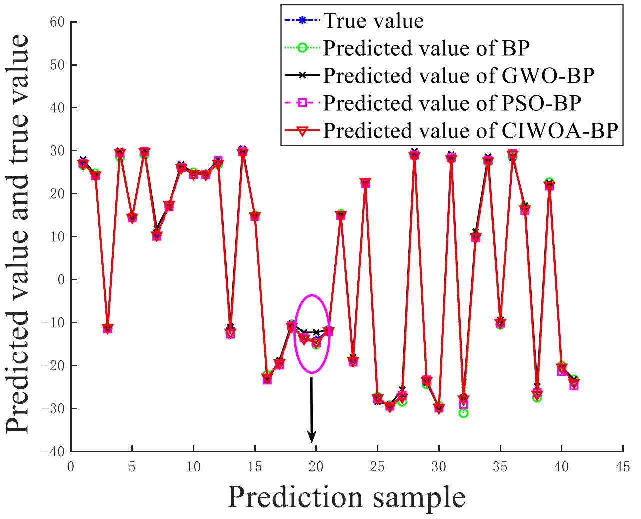


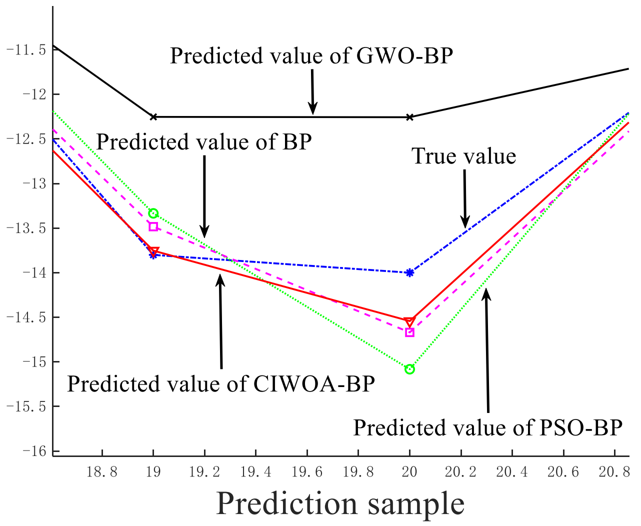
Fig. 20. Comparison of fitness curves of different prediction models

The prediction model proposed in this paper, along with the traditional BP neural network and the BP neural networks optimized by GWO and PSO, were utilized to predict the sample data. The predictions were then compared with the actual values. The comparison of predicted values and actual values, as well as the associated prediction errors, is illustrated in Fig. 21.

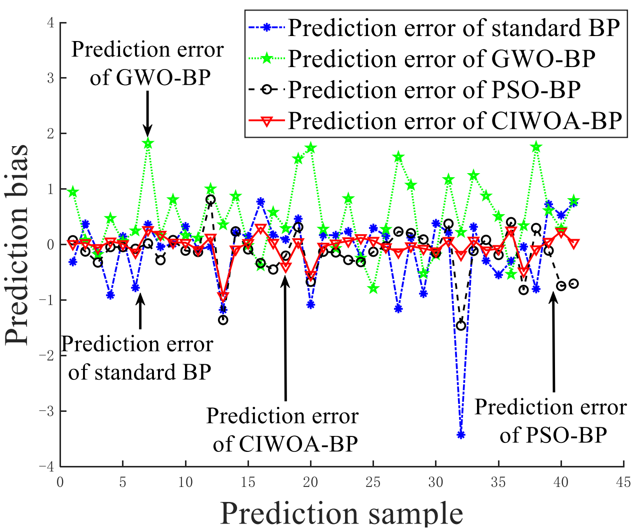
From the analysis of Fig. 21(a), it is evident that the prediction values generated by the proposed neural network



(a) The comparison between the predicted value and the true value of the BP neural network prediction model optimized by different algorithms



(b) The comparison between the predicted value and the true value of the BP neural network prediction model optimized by different algorithms (Locally enlarge)



(c) Comparison of prediction errors of BP neural network prediction models optimized by different algorithms

Fig. 21. Comparison of BP neural network prediction models optimized by different algorithms

prediction model are the closest to the actual values compared to those from the traditional BP neural network, the GWO-BP, and the PSO-BP. In order to more clearly observe the difference between the predicted value of different algorithms and the real value, as shown in Fig. 21 (b), we locally enlarged the Fig. 21 (a). Furthermore, as shown in Fig. 21(c), the proposed model exhibits the smallest prediction error, reinforcing this conclusion. In summary, the proposed neural network prediction model demonstrates significant feasibility and effectiveness.

D. Experimental verification and analysis of robot driving

The experimental platform for this project includes the NAO robot and a micro electric vehicle. The micro electric vehicle is shown in Fig. 22. The UI overview diagram of the vehicle's main controller in the Linux system is shown in Fig. 23. The relevant data of the car are shown in Table IV.

To verify the rationality and effectiveness of the prediction model proposed in this paper, we conducted simulation experiments and real vehicle experiments. To comprehensively verify the performance of the NAO robot in controlling the car to avoid obstacles, two scenarios with multiple obstacles were designed for the experiment. This paper takes the situation with five obstacles as an example. Through the identification of critical obstacles in section II. D, the robot selects three of these obstacles to form a group of key obstacles. A red target area is set in front of the obstacles. If the NAO robot can successfully control the car to avoid the key obstacles and reach the red target area, the experiment is considered successful. The simulation experiment was carried out using the robot simulation software Webots. As shown in Fig. 24, in the simulation experiment of the NAO robot driving the car, Fig. 24(a) shows scenario one, where the robot controls the vehicle to bypass the obstacles and successfully reaches the left side of the target area; Fig. 24(b) shows scenario two, where the robot controls the vehicle to bypass the obstacles and successfully reaches the right side of the target area. In addition, the experiment of the NAO robot driving a real vehicle is shown in Fig. 25, and its experimental scene is

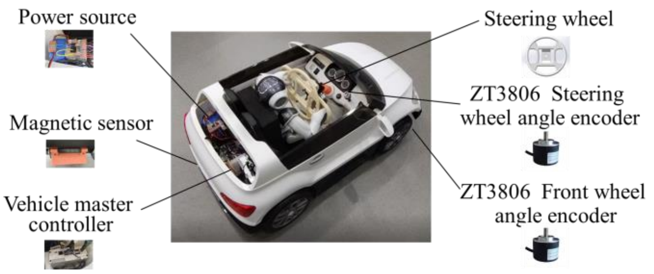
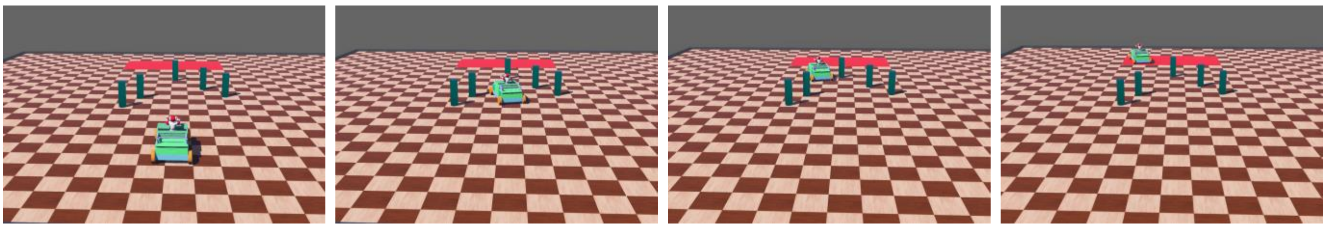


Fig. 22. Humanoid robot driving experiment platform

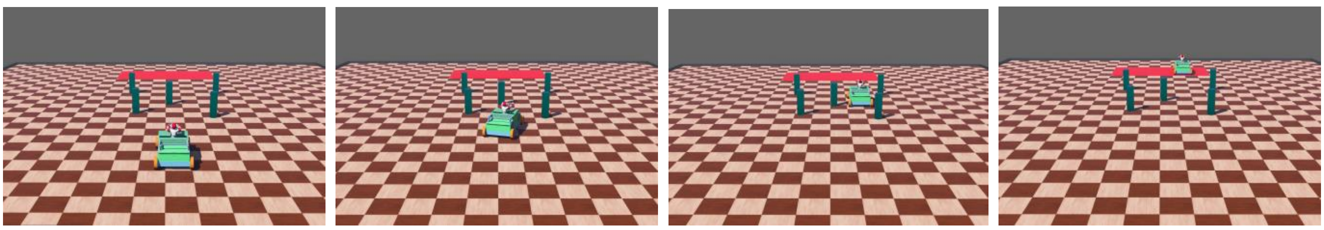
TABLE IV
CAR DATA

Argument	Data	Unit
wheelbase	70	cm
Length, width and height	100*62*54	cm
Maximum front wheel Angle	30	°
Maximum steering Angle	60	°
Speed of vehicle	0.2-1.1	m/s
Steering wheel radius	0.1	m
O_0O	75	cm

Fig. 23. Master controller sensor data overview



(a) Simulation experiment a



(b) Simulation experiment b
Fig. 24. Webots-Robot driving simulation experiment diagram

(a) Robot driving experiment a

(b) Robot driving experiment b

Fig. 25. Robot driving real experiment diagram

consistent with the simulation environment.

By comparing the simulation experiment shown in Fig. 24 with the real-vehicle experiment depicted in Fig. 25, it is evident that the NAO robot successfully steers the vehicle to navigate around obstacles and reach the red target area smoothly. Furthermore, the trajectories from both the real-vehicle and simulation experiments are largely similar, further confirming the effectiveness of the method proposed in this paper

VI. CONCLUSION

This research proposes a distance prediction model based on visual ranging and visual recognition, as well as a BP neural network prediction model based on the improved Whale Algorithm, to realize the obstacle avoidance of driving a vehicle by a humanoid robot. Not only can the robot vision be utilized to obtain information about the surrounding environment, but the proposed prediction model can also use the distance between the car and the obstacle to determine the appropriate wheel angle.

A vision-based distance prediction model has been developed to determine the distances between the robot-driven car and surrounding obstacles. After the NAO robot acquires a reference distance using a monocular ranging model, it captures images of the environment through its vision system. These images are then processed and analyzed, allowing the distance estimation model to estimate the distances between the robot and other obstacles. These estimated distances are subsequently mapped to the distances between the car and the obstacles. The paper includes an error analysis to validate the distance estimation model, demonstrating its feasibility and effectiveness.

To derive the appropriate wheel Angle of the car based on the distance information between the car and obstacles, this paper proposes a prediction model that optimizes a BP neural network using the CIWOA algorithm. The humanoid driving data collected serves as training data for the model. The proposed prediction model is compared with traditional BP neural network prediction models, as well as GWO-BP models and PSO-BP models. The results demonstrate that the prediction model presented in this paper outperforms the others in terms of the similarity between predicted and actual values, prediction error, and iterative accuracy of the algorithm. These findings validate the feasibility and effectiveness of the proposed prediction model.

In the future, we will consider more complex robot driving scenarios. Because in this paper, we only discuss the effects of static obstacles on robot-driven cars. However, in the actual driving environment, dynamic obstacles can also pose challenges for robot driving. Thus, we will to introduce the treatment mechanism of dynamic obstacles in the follow-up research. Through the vision technology, we will analyze the moving direction and speed of dynamic obstacles, combined with the existing static obstacle recognition ability, to achieve the robot driving vehicle in the complex environment of obstacle avoidance, and ensure that it can safely and accurately reach the predetermined target point.

REFERENCES

[1] CHEN G, ZHANG W. Control system design for electromagnetic driving robot used for vehicle test; *proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, F, 2019. IEEE.

[2] CHEN G, JIANG J, WANG L, et al. Clutch mechanical leg neural network adaptive robust control of shift process for driving robot with clutch transmission torque compensation. *IEEE Transactions on Industrial Electronics*, 2021, 69(10): 10343-53.

[3] MOON J, HONG D, KIM J, et al. Enhancing Autonomous Driving Robot Systems with Edge Computing and LDM Platforms. *Electronics*, 2024, 13(14): 2740-.

[4] WANG G. RL-CWtrans Net: multimodal swimming coaching driven via robot vision. *Frontiers in Neurorobotics*, 2024.

[5] LUKMAN Y, PRASETYA D W I, CAHYANTARI E. Design and implementation program identification of traffic form in self driving car robot. *Journal of Physics: Conference Series*, 2019, 1367: 012067-.

[6] PAOLILLO A, GERGONDET P, CHERUBINI A, et al. Autonomous car driving by a humanoid robot. *Journal of Field Robotics*, 2018, 35(2): 169-86.

[7] MIZUTANI N, MATSUI H, YANO K I, et al. Vehicle Speed Control by a Robotic Driver Considering Vehicle Dynamics for Continuously Variable Transmissions. *Journal of Robotics and Mechatronics*, 2018, 30(2): 300-10.

[8] WANG H, CHEN G, ZHANG W. A method for vehicle speed tracking by controlling driving robot. *Transactions of the Institute of Measurement and Control*, 2020, 42(8): 1521-36.

[9] CHEN G, ZHANG W-G, ZHANG X-N. Speed tracking control of a vehicle robot driver system using multiple sliding surface control schemes. *International Journal of Advanced Robotic Systems*, 2013, 10(2): 90.

[10] CHEN G, ZHANG W. Control method for electromagnetic

unmanned robot applied to automotive test based on improved smith predictor compensator. *International Journal of Advanced Robotic Systems*, 2015, 12(7): 104.

[11] CHEN G, ZHANG W. Comprehensive evaluation method for performance of unmanned robot applied to automotive test using fuzzy logic and evidence theory and FNN. *Computers in Industry*, 2018, 98: 48-55.

[12] KIM H, SOON C Y. Development of a ROS-Based Autonomous Driving Robot for Underground Mines and Its Waypoint Navigation Experiments. *Tunnel and Underground Space*, 2022, 32(3): 231-42.

[13] ANFU G, TAO J, JUNJIE L, et al. Design of a small wheel-foot hybrid firefighting robot for infrared visual fire recognition. *Mechanics Based Design of Structures and Machines*, 2023, 51(8): 4432-51.

[14] SHIN S, KIM M, AHN J, et al. Development of Tele-operation Interface and Stable Navigation Strategy for Humanoid Robot Driving. *Journal of Institute of Control, Robotics and Systems*, 2016, 22(11): 904-11.

[15] JIN Y, SHI Z, XU X, et al. Target Localization and Grasping of NAO Robot Based on YOLOv8 Network and Monocular Ranging. *Electronics*, 2023, 12(18).

[16] GAO W, CHEN Y, LIU Y, et al. Distance Measurement Method for Obstacles in front of Vehicles Based on Monocular Vision; *proceedings of the 2020 2nd International Conference on Computer,Communications and Mechatronics Engineering* (CCME2020), F, 2020.

[17] SHEN J, GANS N. Robot-to-human feedback and automatic object grasping using an RGB-D camera–projector system. *Robotica*, 2017, 36(2): 241-60.

[18] GAO J, GUI Y, JI W, et al. EU-Net: a segmentation network based on semantic fusion and edge guidance for road crack images. *Applied Intelligence*, 2024: 1-15.

[19] SHEN H, JI X. Optimization of garment sewing operation standard minute value prediction using an IPSO-BP neural network. *AUTEX Research Journal*, 2024, 24(1): 20230034.

[20] Lingzhi Wang, Cheng Li, and Chenyang Li, "A Short-term Wind Speed Prediction Method Based on the DGA-BP Neural Network," *IAENG International Journal of Computer Science*, vol. 51, no. 5, pp496-505, 2024

[21] MIRJALILI S, LEWIS A. The Whale Optimization Algorithm. *Advances in Engineering Software*, 2016, 95: 51-67.

[22] TIEN BUI D, ABDULLAHI M A M, GHAREH S, et al. Fine-tuning of neural computing using whale optimization algorithm for predicting compressive strength of concrete. *Engineering with Computers*, 2019, 37(1): 701-12.

[23] Ronghe Zhou, Yong Zhang, Xiaodong Sun, Haining Liu, and Yingying Cai, "MSWOA: Multi-strategy Whale Optimization Algorithm for Engineering Applications," *Engineering Letters*, vol. 32, no. 8, pp1603-1615, 2024

[24] Yutong Li, Yu Liu, Chunli Lin, Jiayao Wen, Pengguo Yan, and Yukun Wang, "An Improved Pelican Optimization Algorithm Based on Chaos Mapping Factor," *Engineering Letters*, vol. 31, no.4, pp1605-1634, 2023

[25] CHEN L, SONG N, MA Y. Harris hawks optimization based on global cross-variation and tent mapping. *The Journal of Supercomputing*, 2023, 79(5): 5576-614.

[26] LI Y, HAN M, GUO Q. Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization. *KSCE Journal of Civil Engineering*, 2020, 24(12): 3703-13.

[27] ZHONG M, LONG W. Whale optimization algorithm with nonlinear control parameter. *MATEC Web of Conferences*, 2017, 139: 00157.

[28] ZHANG H, PAN Y, ZHANG J, et al. Tent chaos and nonlinear convergence factor whale optimization algorithm. *International Journal of Innovative Computing, Information and Control*, 2021, 17(2): 687-700.

[29] LI M, XU G, FU Y, et al. Improved whale optimization algorithm based on variable spiral position update strategy and adaptive inertia weight. *Journal of Intelligent & Fuzzy Systems*, 2022, 42(3): 1501-17.

[30] LIU L, ZHANG R. Multistrategy improved whale optimization algorithm and its application. *Computational Intelligence and Neuroscience*, 2022, 2022(1): 3418269.

[31] Yuan Wei, Haoran Li, Zhigang Zhou, and Dongdong Chen, "Driving Path Tracking Strategy of Humanoid Robot Based on Improved Pure Tracking Algorithm," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 2, pp286-297, 2024