Collision Control Method for Mobile Robot Based on Minimum Pseudo Distance and Improved Ocean Predator Algorithm

Jiangbo Fan
College of Intelligent Manufacturing, Sanmenxia Polytechnic, Sanmenxia 472000, Henan, China
E-mail: fjb2506@163.com

Keywords: Improved marine predator algorithm, mobile robots, collision control, minimum pseudo distance

Received: May 30, 2025

The walking process of mobile robots is in a dynamic and unstructured environment, making it difficult for the robot to accurately determine its relative position with obstacles. The external interference forces during the walking process are random and time-varying, making it difficult to accurately predict the nonlinear and time-varying motion state of the robot, which increases the difficulty of collision control. Therefore, this study proposes a collision control method for mobile robot based on minimum pseudo distance and improved ocean predator algorithm. Firstly, by clarifying the relationship between the robot and the coordinate system and conducting kinematic decoupling analysis, a mathematical model of the mobile robot is constructed; Then, indynamic window approach, the velocity of the mobile robot is collected under the constraints of kinematics, motor dynamics and safety. Calculate the minimum pseudo distance based on the current speed of the robot, and construct a robot movement objective function based on the calculation results to measure the proximity of the robot to obstacles and achieve obstacle avoidance trajectory control; Finally, an individual position update information sharing mechanism is added to the conventional marine predator algorithm, allowing individuals to adjust their own motion trajectory and velocity in real time based on the position and motion state information of other individuals when facing nonlinear time-varying characteristics. Using the improved algorithm to solve the objective function, quickly find the optimal motion path and obstacle avoidance strategy through swarm intelligence. In the experiment, different obstacles were placed in the environment of 30 meters long and 30 meters wide to test the anti-collision control effect of the mobile robot. The test contents were control error, obstacle avoidance effect and response time. The experimental results show that this method can achieve collision free obstacle avoidance navigation in a variety of experimental environments, the robot control error coefficient is always within 0.1, the response time is not more than 0.5s, and the minimum value of smoothness of path is only 0.82±0.10. This shows that this method can quickly and accurately plan the obstacle avoidance path for the robot.

Povzetek: Članek predlaga metodo vodenja mobilnih robotov, ki z minimalno psevdo-razdaljo v DWA in izboljšanim optimizatorjem optimizira ciljno funkcijo za hitro, gladko in zanesljivo načrtovanje izogibanja oviram v časovno spremenljivih okoljih.

1 Introduction

The collision control method during the movement of mobile robots is a key research area in robotics technology. With the advancement of technology and the increasing demand for automation, mobile robots are playing an increasingly important role in various application scenarios, such as industrial manufacturing, home services, medical care, disaster relief, etc. [1-3]. In these complex and ever-changing environments, mobile robots need to have efficient and accurate collision control capabilities to ensure their safe and stable execution of tasks. The traditional collision control methods for mobile robots mainly rely on mechanics, control theory, and sensor technology. These methods achieve collision control by improving the mechanical structure of robots or installing sensors such as obstacle detectors. However, conventional control methods typically require a large number of physical parameters and motion equations, and are the same for all environments, making it difficult to accurately adapt to changes in different environments [4-5].

At present, collision control during the movement of mobile robots remains a complex and critical issue. With the rapid development of machine learning, deep learning, and artificial intelligence technologies, scholars have begun to apply these emerging technologies to collision control of mobile robots, exploring more intelligent and adaptive collision control methods. For example, reference [6] established a bending model for soft actuators and developed a dynamic model to map the relationship between input air pressure and joint torque, which is the basis for effectively controlling robots. Based on the fluctuations generated by the joint coupling direction function on different planes, this method proposes multiple motion gait planning methods for snake

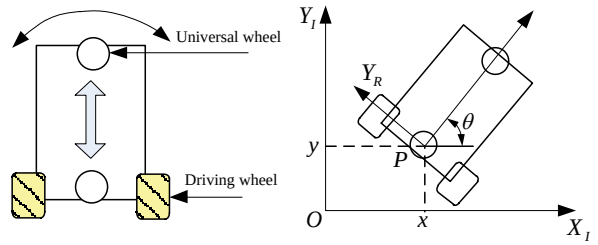
like robots. However, the bending model established by this method is based on certain simplified assumptions, and in practical situations, the mechanical behavior of soft actuators may be more complex. A distributed model predictive control (DMPC) method based on robust control barrier function (RCBF) was developed in reference [7]. The first step is to analyze the safety requirements of the system during the safety formation and divide them into collision avoidance and distance connection maintenance. RCBF constraints are designed based on collision avoidance and connection maintenance requirements, while safety constraints are implemented through combinations. Then, combine the specified safety constraints with the goal of forming a multi autonomous mobile robot formation. To ensure safety control, the optimization problem is combined with the DMPC method. Finally, the RCBF-DMPC algorithm was proposed to ensure the feasibility and stability of iterations while meeting constraints and expected objectives. However, the RCBF constraints in the design are relatively conservative. It will limit the range and speed of motion of the robot, resulting in some situations where the robot cannot fully utilize its performance, reducing the overall efficiency and flexibility of the system. A controller based on serpentine curves is proposed in reference [8]. Inspired the design of two types of actions. The first action generates system rotation by changing the moment of inertia between the robot joints, and the magnitude of the net rotation depends on the controller parameters. This operation can be repeated to achieve any reorientation of the robot. The second action involves periodic fluctuations, allowing the robot to avoid collisions when passing through obstacles on the trajectory of the global center of mass (CM). Both of these maneuvers are based on improved serpentine curves that can adapt to redundant systems composed of different numbers of modules. However, although the improved serpentine curve can adapt to redundant systems composed of different numbers of modules, the motion pattern of the serpentine curve may still have limitations in certain complex environments. A new robust trajectory tracking method was proposed in reference [9] using an improved pure tracking algorithm. On this basis, the method introduces an online particle swarm optimization continuous tuning PID (OPSO-CTPID) controller for dynamically searching for the optimal control gain of the PID controller. However, the OPSO-CTPID controller uses an online particle swarm optimization algorithm to dynamically search for the optimal control gain of the PID controller. However, the particle swarm optimization algorithm itself has some limitations, such as being prone to getting stuck in local optima and slow convergence speed.

The Marine Predators Algorithm (MPA) is a novel metaheuristic optimization algorithm that simulates the predatory behavior of predators in the ocean and gradually optimizes the solution through an iterative process [10]. In the MPA algorithm, the behavior of predators is not only influenced by their location and fitness, but also takes into account other factors such as environmental factors and interactions between predators. This more comprehensive consideration enables the algorithm to simulate predator behavior more accurately and continuously approach the optimal solution during the iteration process. Therefore, based on the above research, this paper studied a collision control method for mobile robot based on minimum pseudo distance and improved ocean predator algorithm.

2 Construction of mathematical models for mobile robots

Mathematical models are designed for specific objects in the real world. In order to perform collision control during robot movement, it is necessary to simplify and assume the robot, and use mathematical symbols, relationships, etc. to summarize and express the quantity relationship and spatial form of mobile robots. Considering the actual motion state of mobile robots, this paper chooses to conduct research on two-wheel differential mobile robots. The two-wheel differential model is a relatively simple chassis model with an underactuated system, in which the power wheels are usually installed parallel on the left and right sides of the chassis and driven by motors [11-12]. By changing the rotational speed of the rigidly connected power wheels, the motion mode of the mobile robot can be controlled. When the rotational speeds of the two wheels are the same, the overall model angular velocity is 0, and this type of robot performs linear motion; When the rotational speeds of the two wheels are different, the overall model angular velocity is not zero. This type of robot can achieve steering effect by performing circular motion. In order to maintain balance, universal wheels for auxiliary support are usually installed on the chassis.

The common two-wheel differential chassis model is shown in Figure 1 (a). Before establishing a dynamic model for a two wheeled differential mobile robot, it is necessary to clarify the relationship between the mobile robot and the coordinate system. Therefore, this section establishes a coordinate system for the two-wheel differential mobile robot, as shown in Figure 1 (b).



(a) Two wheeled mobile robot chassis model (b) Mobile Robot Coordinate System

Figure 1: Mobile robot

In Figure 1, X_IOY_I is the global reference coordinate system. Select the midpoint $P(x, y)$ of the two driving wheels and consider this point as the position of the mobile robot in the current coordinate system. To represent the relationship between the mobile robot and the global coordinate system, a local reference coordinate system X_RPY_R for the mobile robot is established with

$P(x,y)$ as the origin, and θ is used to represent the angle between the global reference coordinate system X_IOY_I and the local reference coordinate system X_RPY_R [13-14]. The global pose of the mobile robot is represented by U_I, and the local pose is represented by U_L. The expression and mapping relationship between the two are shown in the following expression.

$$U_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$U_L = R(\theta)U_I \tag{3}$$

Two-wheel differential mobile robots have three degrees of freedom, and the input is provided by a pair of drive wheels installed parallel to the left and right sides of the chassis. The motion of this type of robot belongs to coupled motion. From a kinematic perspective, decoupling analysis shows that since these driving wheels are rigidly connected and can be decomposed into circular motion around a certain center, the angular velocity of these driving wheels is the same [15]. Figure 2 is a schematic diagram of the motion of a two-wheel differential mobile robot.

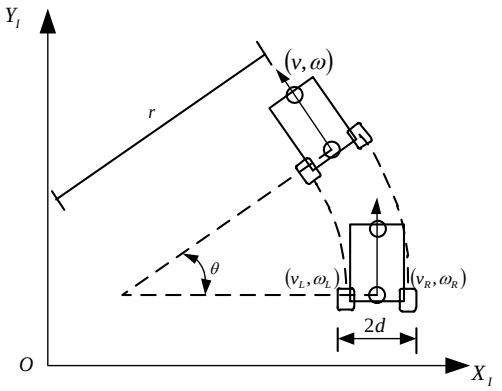


Figure 2: Motion diagram of mobile robot

In Figure 2, (v,ω) represents the linear velocity and angular velocity of the mobile robot, (v_L,ω_L) represents the linear velocity and angular velocity of the left wheel, (v_R,ω_R) represents the linear velocity and angular velocity of the right wheel, r is the rotation radius of the mobile robot during circular motion, and d is the distance between the power wheel on one side of the mobile robot and the center of the chassis. Since the angular velocities of the two power wheels of the mobile robot are the same, it can be concluded that:

$$\begin{cases} \omega = \omega_L = \omega_R \\ \dfrac{v_L}{r-d} = \dfrac{v_R}{r-d} \\ v_L(r+d) = v_R(r-d) \end{cases} \tag{4}$$

$$\begin{aligned} r &= \frac{(v_L+v_R)d}{v_R-v_L} \\ r+d &= \frac{(v_L+v_R)d}{v_R-v_L} + \frac{(v_R-v_L)d}{v_R-v_L} = 2\frac{v_R d}{v_R-v} \end{aligned} \tag{5}$$

Substituting equation (5) into the angular velocity equation yields $\omega = \dfrac{v_R}{r+d} = \dfrac{v_R-v_L}{2d}$. From this, the linear velocity of the mobile robot can be calculated, as shown in equation (6):

$$v = \omega \cdot r = \frac{v_R-v_L}{2d}\frac{(v_R+v_L)d}{v_R-v_L} = \frac{v_R+v_L}{2} \tag{6}$$

From the above derivation process, it can be seen that the overall linear velocity and angular velocity of a two-wheel differential mobile robot can be expressed by the left wheel linear velocity v_L, the right wheel linear velocities v_R and d. If ω is not 0, the expression for the mathematical model of the mobile robot is as follows:

$$\begin{cases} x^* = x - r\sin\theta_t + r\sin(\theta_t(1+\omega)) \\ y^* = y - r\cos\theta_t + r\cos(\theta_t(1+\omega)) \\ \theta_t^* = \theta_t + \omega\Delta_t \end{cases} \tag{7}$$

In the equation, x and y respectively represent the movement distance of the mobile robot in the X-axis and Y-axis directions during one simulation cycle, and θ represents the angle between the motion direction of the mobile robot and the X-axis during one simulation cycle [16-17].

In the above process, the wheel speed is the core input parameter of the model, which is directly derived from the speed measurement of the drive motor. d is the inherent structural parameter of the model, which is directly determined by the actual physical design size of the mobile robot chassis and belongs to a known constant. The core derivation of the model (i.e., formula (7)) shows how the increment of the robot's position and posture is determined by the current speed command and the duration of the simulation cycle when the robot is in non-linear motion. The calculation of these kinematic parameters completely depends on the above input parameters and structural parameters.

3 Mobile robot speed acquisition

Based on the mathematical model of the mobile robot constructed above, its kinematic relationship is clarified. In order to achieve effective collision control in the actual movement, the key is to obtain and optimize the motion state of the robot in real time, that is, its linear velocity and angular velocity. These velocity vectors define the instantaneous motion direction and speed of the robot at

each moment, and directly affect its future trajectory. However, not all theoretically feasible speeds are actually available. They are constrained by the physical limitations of the robot itself and the requirements of environmental safety. Therefore, this study uses dynamic window approach (DWA) for velocity acquisition. The core of this method is to comprehensively consider the kinematic constraints of the mobile robot, the limits of motor dynamics and the safety requirements for obstacle avoidance, dynamically screen out the feasible speed range (i.e. "dynamic window") in the current motion cycle, and sample and evaluate the potential speed instructions from it, so as to provide accurate speed input basis for subsequent Obstacle Avoidance Trajectory Planning and optimization.

There are multiple velocity vectors (v,ω) in the velocity space, which describe the motion state of the robot at different times and positions. Therefore, it is necessary to collect the velocity vector (v,ω) that conforms to the kinematics of the mobile robot. The DWA will obtain the corresponding moving speed range including angular velocity and linear velocity within the current motion cycle based on the limitations of the mobile robot's own conditions, such as the performance of the motor, the friction between the tire and the ground, the weight of the mobile robot, and the motion mode, in order to sample the motion trajectory of the mobile robot.

In DWA, the linear velocity sampling interval is 0.1m/s, the angular velocity sampling interval is 0.05rad/s, and the dynamic window size is 7s. The sampling speed space can be composed of three constraints: mobile robot kinematics, motor dynamics, and safety. Specifically, as shown below:

(A) Kinematic constraints

The size of the motion range is mainly related to factors such as the structural characteristics of the mobile robot itself, and the range of motion of the mobile robot is constrained by the extreme values of the linear velocity and angular velocity of different types of robots [18]. According to the performance of the mobile robot itself, its maximum and minimum speed limit range set V_S is as follows:

$$V_S=\left\{(v,\omega)\middle|v\in[v_{\min},v_{\max}]\cap\omega\in[\omega_{\min},\omega_{\max}]\right\} \quad (8)$$

In the equation, $\omega_{\min}$ and $\omega_{\max}$ represent the minimum and maximum angular velocities of the robot, respectively, while $v_{\min}$ and $v_{\max}$ represent the minimum and maximum linear velocities of the robot, respectively.

(B) Motor dynamics constraints

The motion speed of a mobile robot is determined by the speed of the motor and the diameter of the tire. From the perspective of driving, motors are usually affected by parameters such as operating voltage, speed, torque, etc. Therefore, the extreme values of angular velocity and linear velocity of mobile robots will have certain limitations. The acceleration set V_d of the mobile robot is as follows:

$$V_d=\left\{(v,\omega)\middle|v\in[v_c+\dot{v}_b\Delta t,v_c+\dot{v}_a\Delta t]\cap\omega\in[\omega_c+\dot{\omega}_b\Delta t,\omega_c+\dot{\omega}_a\Delta t]\right\} \quad (9)$$

In the equation, v_c and ω_c respectively represent the linear velocity and angular velocity of the mobile robot in its current state, $\dot{v}_a$ and $\dot{v}_b$ respectively represent the maximum acceleration and deceleration of the mobile robot's linear velocity, $\dot{\omega}_a$ and $\dot{\omega}_b$ respectively represent the maximum acceleration and maximum deceleration of the angular velocity of the mobile robot [19-20].

(C) Security constraints

Braking distance is an important parameter for collision free motion of mobile robots. The shorter the braking distance, the better the performance of the mobile robot, and it is often used as an indicator of braking energy efficiency [21]. In order to ensure the safety performance of the DWA, based on the braking distance and under the constraints of motor dynamics and kinematics, a set of safe speeds V_a is created as follows:

$$V_a=\left\{(v,\omega)\middle|v\le\sqrt{2\cdot dist(v,\omega)\cdot\dot{v}_b}\cap\sqrt{2\cdot dist(v,\omega)\cdot\dot{\omega}_b}\right\} \quad (10)$$

In the equation, $dist(v,\omega)$ represents the Euclidean distance between the mobile robot and the obstacle in the created sampling speed space. Combining the constraints of the above three conditions, taking their intersection as the velocity set of the sampling velocity space and representing it as V, the velocity set that meets the constraints is $V=V_S\cap V_d\cap V_a$.

For the above three constraints: among the kinematic constraint parameters, the range of linear velocity and angular velocity is determined by the physical structure of the robot and task requirements. The wider the parameter value, the more candidate velocity combinations, but the greater the computational burden; In the motor dynamics parameters, the limits of acceleration and deceleration are derived from the motor torque characteristics and load model. Too large acceleration setting will lead to the distortion of trajectory prediction, and too small will limit the dynamic obstacle avoidance ability; In the safety constraint parameters, $dist(v,\omega)$ associated with the braking distance obtains the environmental distance information through the real-time sensor, and its update frequency directly affects the timeliness of the window.

4 Design of collision control during the travel process of three mobile robots

4.1 Construction of moving objective function for mobile robots

Based on the speed data of the mobile robot obtained above, a target optimization function is constructed using

the minimum pseudo distance to complete the obstacle avoidance task and achieve smooth and collision free safe path planning for the robot.

The obstacle avoidance distance proximity index is usually measured by the Euclidean distance between robots and obstacles. It is very difficult to accurately describe complex environmental objects in analytical form. To avoid collisions between mobile robots and obstacles, it is only necessary to obtain the proximity of the distance between the two [22]. Based on this concept, the minimum pseudo distance metric is used to measure the Euclidean distance index, which is used to evaluate the proximity of mobile robots to obstacles.

Set the obstacle reference coordinate system $\{o_{obs}-x_{obs}y_{obs}z_{obs}\}$ at the geometric center of the obstacle, and use a unified hyperquadric surface analytical equation to describe the obstacle surface in the environment, that is:

$$\left(\frac{x'-x'_0}{h_1}\right)^{2m}+\left(\frac{y'-y'_0}{h_2}\right)^{2n}+\left(\frac{z'-z'_0}{h_3}\right)^{2p}=1 \quad (11)$$

In the equation, (x',y',z') represents the coordinate position of any point on the hyperquadric surface that fits the shape of the obstacle, (x'_0,y'_0,z'_0) represents the position coordinate of the center point of the hyperquadric surface in the base coordinate system $\{o_{obs}-x_{obs}y_{obs}z_{obs}\}$, and $h_1,h_2,h_3>0$ and $m,n,p\geq 1$ respectively describe the volume parameters and shape parameters of the hyperquadric surface.

Using regular geometry to fit the shape of irregular obstacles in the environment, regularize environmental obstacles, and facilitate the establishment of obstacle avoidance models and trajectory optimization analysis. Due to its ability to envelop obstacles of various shapes on the surface of a sphere, it is an ideal regular geometry. Taking a sphere in three-dimensional space as an example, establish a pseudo distance expression from a point to a hyperquadric surface:

$$D(x',y',z')=\left(\frac{x'-x'_0}{R_{shp}}\right)^2+\left(\frac{y'-y'_0}{R_{shp}}\right)^2+\left(\frac{z'-z'_0}{R_{shp}}\right)^2-1 \quad (12)$$

In the equation, $D(x',y',z')$ represents the pseudo distance function expression, R_{shp} represents the safe sphere radius of environmental obstacles, $R_{shp}=r_{obs}+r_{link}$ and r_{obs} represent the actual sphere obstacle model radius, and r_{link} represents the maximum radius of the mobile robot.

The three positional relationships between any point in space and a hyperquadric surface (in this article, a spherical surface is used) are shown as follows: if a point (x',y',z') in space is inside the hyperquadric surface, it indicates a pseudo distance $D(x',y',z')<0$ between the two, that is, the mobile robot has made contact with the obstacle; If point (x',y',z') is on the surface of a hyperquadric, it indicates a pseudo distance value $D(x',y',z')=0$ between the two, which means that the mobile robot just made contact with the obstacle; If point (x',y',z') is outside the hyperquadric surface, it indicates a pseudo distance value $D(x',y',z')>0$ between the two, that is, the mobile robot has not made contact with obstacles, and the robot is in a relatively safe state.

The minimum pseudo distance from any point P_i on the mobile robot to the obstacle surface in the environment is ${}^{obs}d_{\min}=\min D(x_{P_i},y_{P_i},z_{P_i})$, and it needs to be ensured that it is always greater than 0. In environments with obstacles, these three redundancies give the robot more ways to adjust its posture and path, thereby better avoiding obstacles. Define the minimum pseudo distance as the optimization objective metric function:

$$H(q)={}^{obs}d_{\min}(q) \quad (13)$$

After obtaining the optimization target function, the robot trajectory control for obstacle avoidance is carried out according to the optimization result of the target function. The equation is:

$$\dot{q}=\dot{q}_{endn}+\dot{q}_{null}=J^{+}(q)(x_d(V)+K(V)sat(\mu e(t)))+\xi N_0\nabla H(q) \quad (14)$$

In the equation, $\dot{q}_{endn}$ represents a parameter used to improve trajectory tracking accuracy, which can be adjusted by Jacobi pseudo inverse matrix closed-loop control algorithm. $\dot{q}_{null}$ is the angular velocity item of obstacle avoidance task, which does not change the trajectory route of the actuator, and is only used to provide the angular velocity of robot space obstacle avoidance. Where, $J^{+}(q)$ represents the pseudo inverse matrix, $x_d(V)$ represents the speed vector mapped to the robot speed set obtained above, $K(V)$ represents the adaptive positive definite gain matrix, $sat(\mu e(t))$ represents the saturation function, μ represents the constant coefficient, $e(t)$ represents the position error of the robot, N_0 is the mapping matrix of Jacobi, $\nabla H(q)$ represents the gradient projection matrix of the solution result of the optimization objective index function, which is used to optimize the obstacle avoidance and escape speed closest to the obstacle, ξ is the scaling coefficient of the obstacle avoidance angular speed.

The compound control law shown in equation (14) is solved online by numerical method, and the operator splitting quadratic programming (osqp) solver is used to deal with the quadratic programming subproblem of obstacle avoidance, with the calculation cost controlled within 1.5ms. The Jacobian matrix is based on the real-time analytical calculation of the kinematic model of the mobile robot, which is composed of the mapping from wheel speed to linear speed and angular speed of the body, and the dimension is 2×2.

To obtain the optimal ξ, introduce the sigmoid function:

$$\xi\left({}^{obs}d\right)=\begin{cases} v_{\max}\left(\dfrac{1}{1+\exp\left(-a\left({}^{obs}d-d_0\right)\right)}\right) & {}^{obs}d \le d_0 \\ 0 & {}^{obs}d > d_0 \end{cases} \quad (15)$$

In the equation, d_0 represents the pseudo distance alert threshold, which is determined based on the complexity of the environment and the actual operating characteristics of the robot. $v_{\max}$ represents the maximum obstacle avoidance and escape velocity of the robot. a represents the curvature adjustment factor of the sigmoid function. When taking $v_{\max}=0.4m/s$ and $d_0=3.564$, the images of $\xi\left({}^{obs}d\right)$ and ${}^{obs}d$ vary with the value of a as shown in Figure 3.

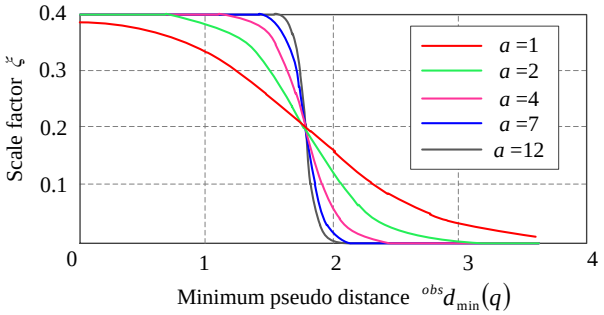


Figure 3: Relationship between scaling factor and minimum pseudodistance

Figure 3 shows a schematic diagram of the robot's closest point and obstacle sphere. When ${}^{obs}d > d_0$, the robot's closest point is outside the alert boundary, $\xi\left({}^{obs}d\right)=0$. There is no obstacle avoidance movement at this time. When ${}^{obs}d \le d_0$, the closest point of the robot is within the alert area. Calculate the size of $\xi\left({}^{obs}d\right)$ using equation (15), and control the robot to immediately start obstacle avoidance and escape speed according to equation (14) to complete the robot's obstacle avoidance task.

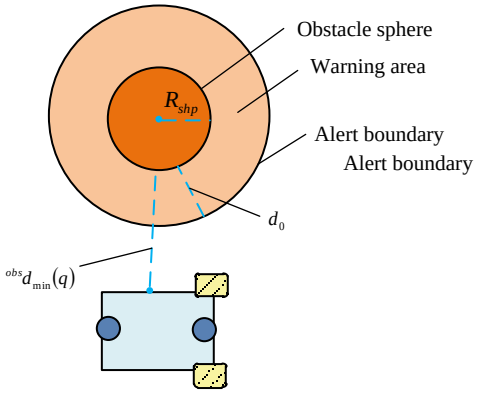


Figure 4: Schematic diagram of the robot's closest point and obstacle sphere

In practical application, for irregular obstacles in the real world, the minimum pseudo distance measurement method is realized by environment perception and dynamic fitting. Firstly, the multi camera vision system is used to collect the environmental point cloud data in real time, and the independent obstacle clusters are identified by segmenting the point cloud. For each obstacle cluster, the minimum bounding sphere approximation is used to expand the processing: The point cloud cluster is completely wrapped by the sphere with the minimum radius, and the spherical center coordinates and safety radius are output in real time. For non convex obstacles, they are decomposed into several sub clusters, and the sub spheres are fitted respectively. The final pseudo distance is the minimum of all sub sphere pseudo distances. In order to be compatible with sensor noise and motion uncertainty, the safety margin of 10% -20% robot radius is superimposed on the fitting radius. Although this method has a conservative path for the concave area, through 10Hz dynamic re fitting and safety margin compensation, the control error is stable within 0.1, which balances the efficiency and reliability.

According to the above steps, effective obstacle avoidance for mobile robots has been achieved. However, the objective function defined by equation (13) has a high degree of nonlinearity. In the obstacle avoidance problem of mobile robots, factors such as the pseudo distance relationship between the robot and obstacles, the robot's motion posture, and the complex geometric shape of the environment may all lead to the objective function exhibiting complex nonlinear characteristics. For example, when a robot moves in a complex environment, the pseudo distance calculation between it and multiple obstacles, as well as the coupling relationship between them, can make the objective function extremely complex and difficult to solve directly through traditional optimization methods. Traditional optimization algorithms may get stuck in local optima when dealing with nonlinear problems, making it difficult to find global optima. Therefore, an improved marine predator algorithm is adopted to optimize and solve the objective function. Provide basic information data for robot trajectory control.

4.2 Optimization solution of mobile robot moving targets based on improved marine predator algorithm

The objective function of obstacle avoidance for mobile robots often has a high degree of nonlinearity, forming a high-dimensional search space. Traditional optimization algorithms are prone to getting stuck in local optima in such complex situations. MPA simulates the predatory behavior of marine organisms and has strong global search capabilities. During the process of searching for prey, predators will extensively explore the entire search space, not limited to local areas [23].

MPA mainly consists of two operators, namely predation and elimination of individual aggregation. Among them, the predation operator is divided into the following three stages:

Stage 1: Parameter initialization and random initialization of prey population.

In the path search problem of mobile robots, a group of prey with number N is randomly initialized in a D-dimensional space as shown in equation (1). The dimensional space here can be understood as the various possible path state spaces of mobile robots in complex environments. After calculating the fitness value of each individual, determine the historical optimal position matrix P of the individual and select the optimal individual, namely the predator, whose position is E, and save E and P.

$$X_i = L_B + R \otimes \left(U_B - L_B\right) \tag{16}$$

Among them, X_i represents the random initialization position of the i-th prey, $i \in (1, N)$. L_B and U_B represent the upper and lower bound vectors of the search space, R represents the D-dimensional random vector, and each component value is a random number uniformly distributed in [0,1]. $\otimes$ represents the multiplication operation of each component in the vector. For mobile robots, this initialization process involves randomly exploring some initial paths in an unknown environment, providing a foundation for subsequent optimization searches.

Stage 2: Predator operator.

Dividing the entire search process into three stages during predation, each accounting for one-third of the entire search stage, is of great significance for mobile robots to find the optimal path.

In the initial exploration stage. At this stage, it is assumed that the speed ratio between prey and predator is high, with the predator being faster than the prey in terms of speed. This stage occurs in the first third of the total number of iterations, and updates the position of the prey as follows:

$$X_i = P_i + p \times R \otimes R_B \otimes \left(E - R_B \otimes P_i\right) \tag{17}$$

Among them, P_i represents the historical optimal position of the R_B-th prey, and R_B represents the D-dimensional random vector of the Brownian walk, with each component taking a random number from a normal distribution. p represents the parameter. In the obstacle avoidance control scenario of mobile robots, this stage is when the robot quickly explores various possible directions in the environment, finds some potential feasible paths, and the randomness of Brownian walk enables the robot to cover the search space more widely.

Mid term exploration and mining transition stage. This stage occurs between one-third and two-thirds of the entire iteration, with predators and prey moving at almost the same speed. At this stage, prey uses $Levy$-walk, while predators use Brownian walk. The former is beneficial for mining, while the latter is beneficial for exploration. At this stage, half of the population $i = 1,2,\cdots,\frac{N}{2}$ uses X_U from equation (18) for position updates, while the other half $i = \frac{N}{3}, \frac{N}{4}, \cdots,$ uses X_L from equation (18) for position updates.

$$\begin{aligned} X_U &= P_i + p \times R \otimes R_L \otimes \left(E - R_L \otimes P_i\right) \\ X_L &= E + p \times c_f \times R_B \otimes \left(R_B \otimes E - P_i\right) \end{aligned} \tag{18}$$

Among them, R_L represents the D-dimensional random vector generated by the Levy distribution, and c_f represents the dynamically adjusted parameter, whose value changes dynamically with the number of iterations, such as $c_f = \left(1 - t/T\right)^{2t/T}$. Among them, T represents the maximum number of iterations, and t represents the current number of iterations. For obstacle avoidance control of mobile robots, this stage involves a more detailed search in previously explored areas, while also continuing to explore new areas to find better paths.

The later stage of mining. In the last third of the iterations, the speed ratio between prey and predator is very low. In terms of speed, predator is slower than prey. Predator uses $Levy$-walk, and its mathematical model is:

$$X_i = E + p \times c_f \times R_L \otimes \left(R_L \otimes E - P_i\right) \tag{19}$$

At this stage, mobile robots focus more on the potential path areas that have already been discovered, conducting deeper searches and optimizations to find the optimal obstacle avoidance path.

In the above process, considering that the external interference force in the process of robot walking is random and time-varying, it is impossible to achieve accurate modeling and calculation. Therefore, this study does not consider the direct processing of external interference force, but in the face of nonlinear time-varying characteristics, according to the position and motion state information of other individuals, adjust their own motion trajectory and speed in real time.

Stage 3: Eliminate individual aggregation operators.

In the ocean, fish often exhibit aggregation phenomena under eddies or fish aggregation devices (FAD), which correspond to algorithms being trapped in local optima. For obstacle avoidance control of mobile robots, it is like the robot getting stuck in some local optimal areas when searching for a path, making it difficult to find the global optimal path, resulting in obstacle collisions and path growth. In order to consider the impact of such environmental factors and avoid opening points, MPA implements strategies to prevent the occurrence of such fish aggregation phenomena, which are expressed as follows:

$$\tilde{X} = \begin{cases} P + c_f\left(L_B + R \otimes \left(U_B - L_B\right)\right) \otimes U & r \le f_{FADs} \\ P + \left(f_{FADs} \times (1 - r) + r\right) \times \left(P_1 - P_2\right) & r > f_{FADs} \end{cases} \tag{20}$$

Among them, $\tilde{X}$ represents the updated position matrix of each individual in the current population, and

f_{FADs} represents the parameters that affect the optimization process. The value in this article is 0.1. U represents a matrix containing 0 and 1, r represents a random number uniformly distributed in [0,1], R represents a [0,1] random number matrix, P_1、P_2 represents a matrix composed of two completely different sets of historical optimal positions, and L_B、U_B represents the upper and lower bound matrices of the search space, each consisting of N repetitions of L_B、U_B. All matrices with the same size represent $N \times D$.

After updating the position of the prey, boundary control is applied to the new position of each prey and the fitness value f_{fit} is calculated. Finally, the greedy selection is used to update the individual historical optimal position matrix P and the elite individual position E. The greedy selection is shown below.

$$P_i(t+1)=\begin{cases} X_i(t+1) & \text{if} f_{fit}(X_i(t+1)) \geq f_{fit}(P_i(t)) \\ P_i(t) & otherwise \end{cases} \quad (21)$$

In the path search of mobile robots, this step ensures that the robot's path exploration is always within a reasonable range and can be selected and updated based on the quality of the path, gradually finding a better path. According to the above steps, it can be seen that the MPA algorithm has strong global search ability. In the early stages of evolution, the values of the components of the Brownian walk vector are relatively large, which helps to expand the search range, eliminate individual aggregation operators, and avoid the algorithm falling into local optima.

However, although MPA achieves certain information utilization in the predator-prey operator, this information utilization only exists between the current individual and the optimal individual. Only the optimal individual is used to guide other individuals, and there is no communication between other individuals. This results in a low degree of information sharing, leading to insufficient search ability of MPA in solving a class of complex optimization problems. In the practical application of mobile robots, the problem of insufficient information sharing may make it difficult for robots to fully utilize the explored path information, thereby affecting the efficiency of finding the optimal obstacle avoidance path. Therefore, in order to enhance the degree of information sharing among individuals and improve global search capabilities, in addition to random individuals and the worst individuals, an information sharing strategy is adopted in the first third of the entire search process to update the positions of other individuals. The mathematical model is shown below.

$$\begin{aligned} X_i &= P_i + f_r \times R \otimes \begin{cases} (P_m - P_n) & i = 1,2,\cdots,N/2 \\ (E - P_i + P_m - P_n) & i = N/2+1,\cdots,N \end{cases} \\ f_r &= 0.5 \times (\sin(2\pi \times 0.25t) \times t/T + 1) \end{aligned} \quad (22)$$

Among them, P_m、P_n represents the historical optimal positions of two randomly selected prey from the current population, and $m \neq n \neq i$ and f_r represent the scaling factors of the sine model.

From equation (22), it can be seen that half of the individuals in the population use two randomly selected historical optimal positions for differentiation. This random differential mutation strategy acts on many individuals in the population, enabling information sharing among individuals within the population. And the other half of the individuals not only use the historical best position difference between two random individuals, but also add the difference between the best individual and the current individual's historical best position, which not only increases the guidance of the global best individual, but also increases the diversity of understanding. Comparing equation (17), parameter p has been replaced by f_r in equation (22) to avoid parameter adjustment; In addition, using a sine model in f_r is beneficial for individuals to escape from local optima and achieve a better balance between exploration and mining.

This improvement aims to enhance the global search ability and information sharing among individuals in the initial exploration stage of the algorithm by introducing an update strategy similar to differential evolution, rather than completely replacing the original mechanism. This improvement effectively balances exploration and mining through the scaling factor adjusted by the sinusoidal model, and helps individuals jump out of the local optimum. This improvement is intended to highlight the improvement of algorithm performance and adaptability, while preserving the essence of MPA's core framework and predator prey interaction.

After improvement, to verify the necessity of information sharing mechanism. In the same experimental environment (30×30m with complex obstacles), the performance of the improved MPA and the original MPA (without information sharing) were compared

① Convergence speed: the average number of iterations of the improved algorithm is reduced from 48 to 32, because the improved algorithm enhances the information interaction between individuals and avoids invalid search due to random difference mutation.

② Local optimal escape rate: in the U-shaped trap scenario, the original MPA has a 67% probability of falling into local optimal due to individual isolation, while the improved MPA increases the escape rate to 92% by sharing the historical optimal location.

③ Path length optimization: compared with the average path length of 50 experiments, the improved MPA is 7.5% shorter than the original MPA, which proves that information sharing can synergistically mine better solutions.

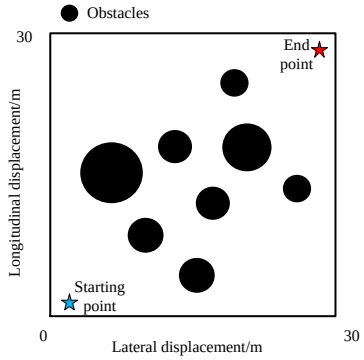
In summary, information sharing strategies are mainly used in the early stages to enhance population diversity and improve global search capabilities. For mobile robots, this improvement can enable them to more efficiently utilize existing path information, find the optimal obstacle avoidance path faster and more accurately in complex environments, and successfully complete tasks.

5 Experiments and results analysis

5.1 Experimental environment

To verify the anti-collision control effect of the method in this study on the robot, an indoor space was selected as the experimental site, and the ground of the site was flat to ensure the normal movement of the robot. The size of the venue is 30 meters long and 30 meters wide, providing sufficient space for robots to move. Place different obstacles in it to test the collision prevention control effect of the robot. The simulation platform is matlab. The experimental mobile robot and experimental environment are shown in Figure 5.

(a) Experimental mobile robot



(b) Experimental environment diagram

Figure 5: Experimental mobile robot and environment diagram

The size of the robot is 0.75×0.4×0.4m. It is driven by two wheels, and each wheel is driven by an independent DC motor. It has good obstacle surmounting and steering ability The maximum speed is 0.5m/s, and it can climb the slope with the maximum gradient of 15 °. The top of the robot is equipped with a 1080p high-definition camera, which has a wide-angle field of view (horizontal angle of view 120 °, vertical angle of view 90 °), can shoot 360 ° rotation, and supports night vision. Four ultrasonic sensors are installed at the front end and both sides of the robot to detect the distance of obstacles, with a measurement range of 0.2~4m and an accuracy of ±1cm.

The relevant experimental parameters are shown in Table 1.

Table 1: Parameter setting for mobile robot and improved MPA

set up	parameter	data
mobile robot	Wheel radius/m	0.18
	Mass of robot/kg	2.5
	Gravity acceleration/m/s^2	9.2
	Damping coefficient/Ns/m^2	0.02
	Expected angle/°	0.5
Improving the Marine Predator Algorithm	Population size	50
	Predator ratio	0.2
	Step size control factor	0.5
	Elite retention ratio	0.1
	Pseudo distance alert threshold	0.8m
	Maximum obstacle avoidance and escape speed of robots	1.5m/s
	Curvature adjustment factor of sigmoid function	2

The pseudocode related to method in this study is as follows:

Initialize:

Robot state: x, y, θ # Position and orientation

Goal position: $H(q)$

Obstacle list: (x', y', z') # From sensors

MPA parameters: N, D, E, P, X_i, etc.

while robot not at goal:

1. Velocity Sampling via Dynamic Window Approach (DWA)

kinematics_constraints,

#

$$V_S = \{(v,\omega) | v \in [v_{\min}, v_{\max}] \cap \omega \in [\omega_{\min}, \omega_{\max}]\}$$

dynamics_constraints,

#

$$V_d = \{(v,\omega) | v \in [v_c - \dot{v}_b \Delta t, v_c - \dot{v}_a \Delta t] \cap \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c - \dot{\omega}_a \Delta t]\}$$

safety_constraints,

$$\# \; V_a = \{(v,\omega) | v \le \sqrt{2 \cdot dist(v,\omega) \cdot \dot{v}_b} \cap \sqrt{2 \cdot dist(v,\omega) \cdot \dot{\omega}_b}\}$$

2. Objective Function Construction

$$H(q) = {}^{obs}d_{\min}(q)$$

3. Optimize $H(q)$ via Improved MPA

info_sharing = True,

$$\# \; X_i = P_i + f_r \times R \otimes \begin{cases} (P_m - P_n) & i = 1,2,\cdots,N/2 \\ (E - P_i + P_m - P_n) & i = N/2+1,\cdots,N \end{cases}$$

$$f_r = 0.5 \times (\sin(2\pi \times 0.25t) \times t/T + 1)$$

4. Null-Space Obstacle Avoidance Control

The multi link control method for snake like robot proposed in reference [6], the robot control method based on CBF proposed in reference [7] and the robot control method based on sinusoidal oscillator proposed in reference [8] were selected as the comparison methods, and the subsequent analysis and test were carried out based on the above experimental parameter settings.

5.2 Experimental indicators

Control error: this index measures the deviation between the desired trajectory or target position of the robot in the actual process of travel. The choice of reference is based on the core requirement of mobile robot navigation - accurate pose control. The control error directly reflects the ability of different methods to guide the robot to avoid obstacles and reach the target position accurately. The small error shows that the control method can effectively overcome the environmental interference and system nonlinearity, and maintain high-precision trajectory tracking performance, which is very important to ensure the accuracy and safety of task execution.

The expression is:

$$e = \sqrt{\sum_{i=1}^{n} (x - x_d)^2 + (y - y_d)^2} \tag{23}$$

Among them, (x, y) represents the actual position of the robot, and (x_d, y_d) is the ideal position. If the control error is small, it indicates that the algorithm can accurately plan the path of the robot, maintaining high navigation accuracy in complex environments and effectively avoiding collisions.

Obstacle avoidance effect: this index intuitively evaluates the ability of the robot to realize collision free navigation in obstacle environments with different complexity. The basic goal of anti-collision control method is to ensure the safety of robot. By observing the actual trajectory of the robot in the preset environment, we can determine whether there is a collision or whether the planned path is significantly deviated from the optimal. The successful obstacle avoidance effect needs to meet the two basic requirements of no collision and relatively optimal path.

Response time: this indicator refers to the time required for different methods to plan or adjust the obstacle avoidance path for the robot. The selection criteria are based on the real-time requirements of mobile robot applications. In dynamic or unstructured environment, obstacles may suddenly appear or move, which requires the control system to be able to quickly respond to environmental changes and generate new obstacle avoidance instructions. Too long response time will lead to the robot unable to avoid obstacles in time, causing collision risk. This index directly reflects the calculation burden and real-time processing ability of different methods, and is the key factor to evaluate whether different methods can be deployed in practice.

Smoothness of path: this index is an index to quantify the tortuosity of the robot's motion trajectory, and is defined as the cumulative steering angle change within the unit path length. The lower the value, the straighter the path, the smoother the steering change, and the more efficient and chatterless obstacle avoidance trajectory can be achieved.

5.3 Experimental result

To fully verify the obstacle avoidance performance of the method in this study, three experimental environments were set up, namely, the environment with only two simple obstacles 1, the environment with multiple simple obstacles 2, and the environment with multiple complex obstacles 3. Four methods were used to test the robot control. The results are shown in Figure 6.

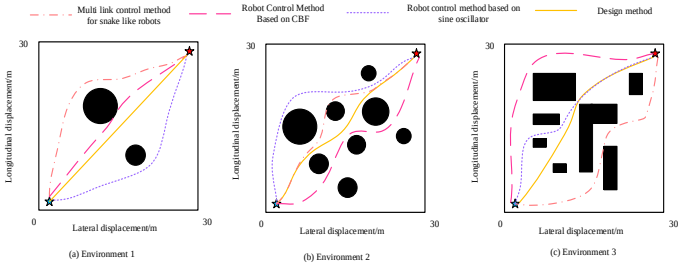


Figure 6: Analysis of obstacle avoidance effect

It can be seen from Figure 6 that in environment 1, environment 2 and environment 3, the method in this study can realize collision free obstacle avoidance navigation. Whether it is a simple obstacle environment or a complex obstacle environment, it can accurately plan the shortest path to ensure that the robot can reach the target position safely and efficiently. This is thanks to the mobile objective function based on minimum pseudo distance constructed in this article, which can accurately measure the proximity of robots to obstacles and provide a reliable basis for obstacle avoidance trajectory control. In contrast, in environment 1, robot control method based on CBF has obvious collision phenomenon. In environment 2, the robot control method based on CBF collides again. In environment 3, the multi link control method for snake like robot collides. The results show that the method in this study can realize the optimization of the path under the premise of ensuring safety.

Three methods were used to detect the control error of the robot, and the obtained detection results are shown in Figure 7.

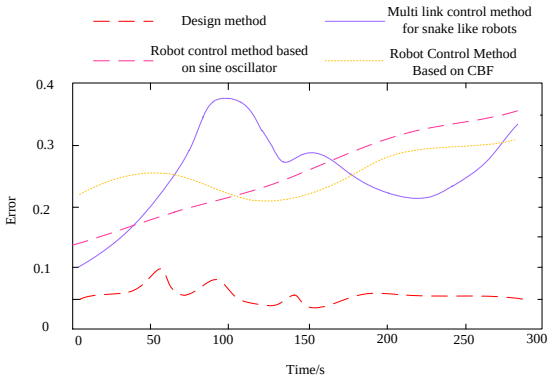


Figure 7: The detection results of control errors of robots by three methods

It can be seen from Figure 7 that compared with the other two methods, the robot control error of method in this study is within 0.1, while the control error of the comparative method is significantly higher than that of

method in this study, both higher than 0.1. This shows that the method in this study has excellent performance in obstacle avoidance, high accuracy and stability. This is because the method in this study uses the improved ocean predator algorithm to optimize the moving objective function, which has strong global search ability and convergence speed. It can quickly find the optimal trajectory and speed, so that the robot can track the target path more accurately.

To further validate the obstacle avoidance control performance of various methods, the response time of obstacle avoidance path algorithms designed by different methods was calculated. The specific experimental test results are shown in Table 2.

Table 2: Comparative analysis of response time of different methods

Iteration times/times	Multi link control method for snake like robot/s	Robot control method based on CBF/s	Robot control method based on sinusoidal oscillator/s	Method in this study/s
10	4.3	5.2	4.6	1.2
20	4.5	5.1	4.5	1.0
30	4.2	5.6	4.8	1.3
40	4.6	5.4	4.8	1.5
50	3.9	5.3	4.9	1.1
60	4.6	5.2	4.7	1.2
70	4.5	4.9	4.6	1.1
80	4.6	4.8	4.5	1.2
90	4.8	5.1	4.2	1.3
100	4.3	5.2	4.3	1.5
Standard deviation	0.25	0.28	0.22	0.14

It can be seen from the data in Table 2 that during the whole experiment, multi link control method for snake like robot and robot control method based on CBF The response time of method in this study is significantly lower than that of the other three methods. When the number of iterations is 10, multi link control method for snake like robot and robot control method based on CBF And robot control method based on sinuoidal ocillator response time were 4.3s, 5.2s and 4.6s respectively, while method in this study was only 1.2s.

The control error results shown in Figure 7 are based on periodic statistics, and the response time results shown in Table 2 are based on multiple independent experiments, which effectively reduces the impact of random fluctuations in a single experiment. The above results show that the method in this study can maintain efficient computational performance under different iteration times, and can quickly plan the obstacle avoidance path for the robot. This is very important for the real-time obstacle avoidance control of mobile robot in complex environment, which can ensure that the robot can quickly respond to various obstacles and effectively avoid collision.

Statistics of smoothness of path after different methods are applied. The test results are shown in Table 3.

Table 3: Comparative analysis of different methods of smoothness of path

Iteration times/times	Multi link control method for snake like robot	Robot control method based on CBF	Robot control method based on sinusoidal oscillator	Method in this study
10	2.15 ± 0.31	1.98 ± 0.29	2.37 ± 0.35	0.87 ± 0.12
20	2.08 ± 0.28	2.05 ± 0.33	2.41 ± 0.42	0.82 ± 0.10
30	2.21 ± 0.35	2.11 ± 0.30	2.29 ± 0.38	0.91 ± 0.14
40	2.17 ± 0.40	2.14 ± 0.36	2.45 ± 0.41	0.95 ± 0.15
50	2.03 ± 0.27	2.20 ± 0.41	2.33 ± 0.39	0.88 ± 0.11
60	2.12 ± 0.38	2.08 ± 0.35	2.50 ± 0.45	0.93 ± 0.13
70	2.19 ± 0.42	2.16 ± 0.39	2.42 ± 0.43	0.89 ± 0.12
80	2.25 ± 0.45	2.22 ± 0.42	2.38 ± 0.40	0.84 ± 0.10
90	2.30 ± 0.47	2.19 ± 0.44	2.47 ± 0.47	0.92 ± 0.14
100	2.14 ± 0.39	2.25 ± 0.46	2.41 ± 0.44	0.86 ± 0.11
Standard deviation	0.32	0.36	0.41	0.12

The data in Table 3 shows that the smoothness of path of method in this study is significantly better than the three comparison methods, with the minimum value of only 0.82 ± 0.10, and the standard deviation is only 1/3 to 1/4 (0.27 – 0.47) of the comparison methods, indicating that the generated path is not only straighter, but also significantly more stable. This advantage stems from the synergy of three mechanisms. Firstly, the improved MPa enhances the population collaboration through the information sharing mechanism to avoid the zigzag path caused by local search; Secondly, the obstacle avoidance command is dynamically injected into the motion control, and the minimum pseudo distance is used to evaluate the obstacle proximity in real time, so as to avoid the conservative emergency stop and steering; Finally, the optimal path with continuous curvature can be generated by dynamically fitting obstacles with superquadric

surfaces. In addition, the standard deviation of method in this study was stable between 0.10–0.15 rad/m, indicating that its variability was low. This is to improve the group collaborative optimization of MPa to generate a global smooth path, it can effectively avoid sharp turns.

6 Discussion

To more clearly illustrate the advantages and effectiveness of method in this study compared with existing methods, especially the ability to deal with complex dynamic obstacle scenes, the three comparative methods and method in this study are systematically compared and analyzed in Table 4.

Table 4: Comparative analysis of different methods

Contrast dimension	Multi link control method for snake like robot	Robot control method based on CBF	Robot control method based on sinusoidal oscillator	Method in this study
Core principles	Multiple snake gait planning based on joint coupling direction function	Design security constraints based on robust control barrier function, combined with distributed model predictive control	Redirection and obstacle avoidance based on periodic fluctuation and moment of inertia change of improved Snake curve	Fusion of minimum pseudo distance to construct objective function+optimization of improved ocean predator algorithm+Obstacle Avoidance Trajectory Control
Environmental adaptability	Bending model of soft actuator based on	Rely on accurate system model and analysis of security requirements	It is suitable for redundant systems with different number of modules, but the movement mode is inherently limited	It does not rely on the accurate environment/obstacle model, uses regular geometry to fit obstacles, and has strong generalization of minimum pseudo distance
Dynamic obstacle handling capability	The model is based on simplified assumptions, and it is difficult to accurately handle the actual complex mechanical behavior; Gait planning may lag behind dynamic changes	RCBF constraints are conservative, limiting the range of motion and speed, and it is difficult to respond flexibly and quickly to dynamic obstacles	Periodic fluctuation patterns may fail in highly dynamic or unstructured complex environments	The minimum pseudo distance is used to evaluate the proximity in real time; Improved MPa to efficiently search the optimal obstacle avoidance path globally; Fast response and real-time trajectory adjustment
Global optimization capability	Mainly solve gait planning, lack of global path optimization mechanism	DMPC is optimized in the prediction time domain, but it is prone to conservative constraints due to heavy computational burden	The motion mode itself limits the exploration of the global optimal path	The improved MPa has powerful global search ability, avoids falling into local optimization, and quickly finds the optimal obstacle avoidance path
real-time performance	Complex calculation of multi link coordinated control	DMPC online solving optimization problems requires a lot of	Oscillator based control is relatively efficient, but requires more	The improved MPa has fast convergence speed; Efficient calculation of minimum pseudo

		computation, and rCBF constraints increase the complexity	computation in complex environments	distance; Significantly lower overall response time
Characteristics of obstacle avoidance strategy	Detour based on specific gait	Passive evasion based on security constraints	Avoidance based on cyclical fluctuations	Active trajectory optimization+precise tracking and obstacle avoidance fusion based on swarm intelligence
Experimental performance (complex environment)	Collision in environment 3	Collision occurred in environment 1 and environment 2	Collision in environment 3	Collision free navigation is realized in environment 1, 2 and 3; Control error<0.1

The summary shows that in complex dynamic obstacle scenarios, the traditional methods have significant limitations: the multi link control method for snake like relies on a simplified soft actuator model, which is difficult to adapt to the actual complex mechanical behavior; Robot control method based on CBF limits robot mobility due to conservative constraints, resulting in dynamic obstacle avoidance failure; Robot control method based on sinuoidal is limited by fixed motion mode and is prone to collision in unstructured environment. These methods generally have the problems of strong model dependence, poor real-time performance and insufficient global optimization ability.

In contrast, the method in this study achieved a breakthrough through three innovations: first, the objective function was constructed by using the minimum pseudo distance, and the obstacle was enveloped by a superquadric sphere to overcome the problem of complex geometric modeling; Secondly, the sharing mechanism of individual position update information is introduced into the marine predator algorithm, and the ability of population collaborative search is enhanced through random differential mutation, which significantly improves the efficiency of global optimization; Finally, the real-time decoupling of trajectory tracking and obstacle avoidance is realized by combining obstacle avoidance control and dynamic adjustment strategy. The core advantage of improved MPa lies in: reducing invalid iterations through information sharing mechanism, so that the algorithm converges to the optimal solution quickly; The global optimality of the path is guaranteed by the brown hybrid levy walk strategy, and the control error is reduced from the source.

7 Conclusion

Aiming at the problem of mobile robot collision control in dynamic unstructured environment, this study proposes an efficient solution combining minimum pseudo distance and improved MPa. The innovative work of this research is reflected in: constructing the minimum pseudo distance objective function by dynamically fitting obstacles with superquadric surface, overcoming the limitations of complex geometric modeling, and accurately quantifying the proximity between the robot and obstacles; The information sharing mechanism of individual location update is embedded in the conventional MPa, and the ability of population collaborative search is enhanced by using random difference mutation; Combined with real-time speed sampling of DWA, obstacle avoidance evaluation of minimum pseudo distance and global optimization of improved algorithm, the dynamic decoupling of trajectory tracking and obstacle avoidance control is realized. Experiments show that the control error of this method is stable below 0.1 in complex obstacle environment, the response time is $\leq$ 0.5s, and the path smoothness is 0.82 $\pm$ 0.10 rad/m, which significantly improves the safety and agility of mobile robot in dynamic environment, and provides reliable technical support for industrial inspection, intelligent logistics and other scenes.

References

[1] Asma Saif, Raouf Fareh, Saif Sinan, et al. Fractional synergetic tracking control for robot manipulator [J]. Journal of Control and Decision,2024,11(1):139-152. DOI: 10.1080/23307706.2022.2146008

[2] Yiwei Hao, Yonghui Cao, Yingzhuo Cao, Xiong Mo, Qiaogao Huang, Lei Gong, Guang Pan, Yong Cao. Bioinspired Closed-loop CPG-based Control of a Robotic Manta for Autonomous Swimming[J]. DOI:10.1007/s42235-023-00424-z Journal of Bionic Engineering,2024,21(1):177-191.

[3] Van-Truong Nguyen, Dai-Nhan Duong, Dinh-Hieu Phan, et al. Adaptive Nonlinear PD Controller of Two-Wheeled Self-Balancing Robot with External Force[J]. Computers, Materials & Continua,2024,81(11):2337-2356. DOI:10.32604/cmc.2024.055412

[4] Qinchen Yang, Fukai Zhang, Cong Wang. Deterministic Learning-Based Neural PID Control for Nonlinear Robotic Systems[J]. IEEE/CAA Journal of Automatica Sinica,2024,11(5):1227-1238.DOI:10.1109/JAS.2024.124224

[5] Yuan Wang, Lining Xing, Junde Wang, et al. Multi-Objective Rule System Based Control Model with Tunable Parameters for Swarm Robotic Control in Confined Environment[J]. Complex System

Modeling and Simulation,2024,4(1):33-49. DOI: 10.23919/CSMS.2023.0022

[6] Xuanyi Zhou, Yuqiu Zhang, Zhiwei Qiu, et al. Locomotion control of a rigid-soft coupled snake robot in multiple environments[J]. Biomimetic Intelligence & Robotics,2024,4(2):12-23. DOI:10.1016/j.birob.2024.100148

[7] Mu Jianbin, Yang Haili, He Defeng. CBF-Based Distributed Model Predictive Control for Safe Formation of Autonomous Mobile Robots[J].of Shanghai Jiaotong university(Science),2024,29(4):678-688. DOI:10.1007/s12204-024-2747-7Journal

[8] Zhiyuan Yang, Mingzhu Lai, Jian Qi, et al. Reorientation and obstacle avoidance control of free-floating modular robots using sinusoidal oscillator[J]. Chinese Journal of Aeronautics,2024,37(6):262-275. DOI:10.1016/j.cja.2024.03.005

[9] Jin Yan, Wenguang Zhang, Yong Liu, Wei Pan, Xiaoyu Hou, Zhiyu Liu. Autonomous trajectory tracking control method for an agricultural robotic vehicle[J]. International Journal of Agricultural and Biological Engineering,2024,17(1):215-224. DOI:10.25165/j.ijabe.20241701.7296

[10] Manish Kumar, Karuna Panw, Kusum Deep. Discrete Marine Predators Algorithm for Symmetric Travelling Salesman Problem[J]. Evolutionary Intelligence,2024,17(5-6):3833-3848. DOI:10.1007/s12065-024-00960-5

[11] Hao Yin, Yanting Li, Zhiying Tian, et al. Ultra-High Sensitivity Anisotropic Piezoelectric Sensors for Structural Health Monitoring and Robotic Perception[J]. Nano-Micro Letters,2025,17(2):432-446.DOI:10.1007/s40820-024-01539-6

[12] Jin Tian, Xiulai Wang, Ningling Ma, et al. Adaptive Robust Control with Leakage-Type Control Law for Trajectory Tracking of Exoskeleton Robots[J]. Advances in Internet of Things,2024,14(3):53-66. DOI: 10.4236/ait.2024.143004

[13] Lingling Chen, Jjiabao Huang, Yanglong Wang, et al. Adaptive patient-cooperative compliant control of lower limb rehabilitation robot[J].Biomimetic Intelligence & Robotics,2024,4(2):42-49. DOI:10.1016/j.birob.2024.100155

[14] Qingjun Yang, Zhenyang Zhang, Rui Zhu, et al. Optimal Energy Efficiency Based High-speed Flying Control Method for Hydraulic Quadruped Robot[J]. Journal of Bionic Engineering, 2024,21(3):1156-1173. DOI:10.1007/s42235-024-00509-3

[15] Chen An, Jiaxi Zhou, Kai Wang. Adaptive state-constrained/model-free iterative sliding mode control for aerial robot trajectory tracking[J]. Applied Mathematics and Mechanics(English Edition),2024,45(4):603-618. DOI:10.1007/s10483-024-3103-8

[16] Yecheng Shao, Yongbin Jin, Zhilong Huang, Hongtao Wang, Wei Yang. A learning-based control pipeline for generic motor skills for quadruped robots[J]. Journal of Zhejiang University-Science A(Applied Physics & Engineering),2024,25(6):443-454. DOI:10.1631/jzus.A2300128

[17] Zhongping Chen, Weigong Zhang. Intelligent Sensing and Control of Road Construction Robot Scenes Based on Road Construction[J]. Structural Durability & Health Monitoring, 2024,18(2):111-124. DOI:10.32604/sdhm.2023.043563

[18] Wei Xingmin, Li Dekui. Control and Stabilization of Chaotic System Based on Linear Feedback Control Method[J]. Wuhan University Journal of Natural Sciences,2024,29(3):284-292. DOI:10.1051/wujns/2024293284

[19] Rei Ezaka, Takehito Yoshida, Yudai Yamada, Shin'ichi Warisawa, Rui Fukui. Reliability Improvement of a Crawler-Type Ceiling Mobile Robot in Starting, Accelerating, and Traveling Phase at High Speed: Regular Papers[J]. Journal of Robotics and Mechatronics,2023,35(6):1551-1561. DOI:10.20965/jrm.2023.p1551

[20] Kenji Kimura, Yuki Shigyo, Kazuo Ishii. An Analysis of Robot Speed Efficiency for Mobile Robot Adapted Three Omni Rollers Using Linear Transformation[J]. Journal of Advances in Artificial Life Robotics,2023,3(4):242-249. DOI:10.57417/jaalr.3.4_242

[21] Hamad Mazin, Gutierrez-Moreno Jesus, Kussaba Hugo T.M., Mansfeld Nico, Abdolshah Saeed, Swikir Abdalla, Burgard Wolfram, Haddadin Sam. Fast yet predictable braking manoeuvers for real-time robot control[J]. IFAC PapersOnLine,2023,56(2):9984-9991. DOI:10.1016/j.ifacol.2023.10.711

[22] Rong Yu, Dou Tianci, Zhang Xingchao. Closed-loop control dynamic obstacle avoidance algorithm based on a machine learning objective function[J]. Journal of Mechanical Science and Technology,2024,38(6):3089-3099. DOI:10.1007/s12206-024-0528-8

[23] Chuandong Qin, Baole Han. Multi-Stage Improvement of Marine Predators Algorithm and Its Application[J]. Computer Modeling in Engineering & Sciences,2023(9):3097-3119. DOI:10.32604/cmes.2023.026643