TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 22/38 pp1678−1690
DOI: 10.26599/TST.2024.9010265
Volume 31, Number 3, June 2026

Research Article

Research on Three-Dimensional Global Dynamic Path Planning Algorithm for Mobile Intelligent Agent

Kene Li*, Wanning Lu, Chunyi Tang, and Chengzhi Yuan

Abstract: This paper intends to address the challenges of mobile intelligent agents in complex three-dimensional environments, specifically the difficulties in obtaining a globally feasible path and avoiding dynamic obstacles. For this purpose, a hybrid global dynamic path planning approach is proposed, which incorporates a neural network based six-direction search (SDS) algorithm and an improved dynamic window approach (DWA). Initially, the SDS algorithm is utilized to rapidly plan an initial feasible path. Subsequently, key nodes from this feasible path are extracted as sub-goal points for the DWA algorithm, ensuring that the path achieves global feasibility. When encountering dynamic obstacles, the mobile agent employs an improved dynamic window approach for local path planning. However, to enhance the environmental adaptability of the mobile intelligent, fuzzy logic is incorporated to dynamically adjust the weight of each parameter in the evaluation function. This combined approach enables the mobile agent to plan a globally feasible path in real-time within complex three-dimensional environments. Finally, both simulations and robotic experiments demonstrate that the proposed algorithm consistently enables mobile end-effector of the robotic arm to find a globally feasible obstacle-avoiding paths in complex three-dimensional spaces.

Key words: six-direction search (SDS); improved dynamic window approach (DWA); fuzzy logic; path planning

1 Introduction

With the rapid development of artificial intelligence and robotics technology, mobile intelligent agents such as spaceflight robots, underwater robots, and autonomous vehicles have found increasingly widespread applications across various fields[1−4]. The capability of these agents to plan efficient and safe navigation autonomously when executing tasks in complex and ever-changing environments is a core manifestation of their intelligence[5]. Particularly in intricate and dynamic three-dimensional (3D) spaces, achieving global dynamic path planning has emerged as a highly challenging research topic[6].

Extensive research has been conducted on path planning problems in three-dimensional spaces. For instance, Nasir et al.[7] proposed an extension of rapidly exploring random tree star algorithm, which accelerated convergence through intelligent sampling and path optimization. However, as the algorithm continuously iterated, the number of nodes would increase indefinitely, which could be a significant challenge to the computer memory. Cao et al.[8] employed particle swarm optimization (PSO) to optimize the trajectory search function for robot path planning. However, the convergence speed of the PSO algorithm in the initial period of the search process was fast, while it in the late period became slow. To handle this problem, Jin et al.[9] proposed a coevolutionary neural solution in finding the exact solution to nonconvex optimizations with convergence speed

• Kene Li, Wanning Lu, and Chunyi Tang are with School of Automation, Guangxi University of Science and Technology, Liuzhou 545006, China. E-mail: likene@163.com; lwn01010@163.com; tangchunyi@gxust.edu.cn.
• Chengzhi Yuan is with Department of Mechanical, Industrial and Systems Engineering, The University of Rhode Island, Kingston, RI 02881, USA. E-mail: cyuan@uri.edu.
* To whom correspondence should be addressed.
Manuscript received: 2024-09-20; revised: 2024-11-18; accepted: 2024-12-30

accelerated. Boulares et al.[10] applied deep Q-learning to optimize the path of unmanned aerial vehicles (UAVs) in marine environments, significantly improving search and rescue mission efficiency through autonomous adjustment. Although the method was very effective, it lied on a substantial amount of training data. Zhang et al.[11] introduced a novel fruit fly optimization algorithm with a mutation adaptation mechanism based on phase angle encoding, providing enhanced search capabilities and faster convergence to optimal paths. However, the convergence of the algorithm and optimality of the better solution were not all satisfactory in practical applications. Liu et al.[12] proposed a competitive-cooperative scheme addressed by a distributed coordination method for multi-robot manipulation scenarios, enabling robots to collaborate effectively and ensure safety. Adhikari et al.[13] proposed a fuzzy adaptive differential evolution for solving 3D UAV path planning problems. However, despite their widespread application, these algorithms still had some drawbacks such as decreased accuracy and insufficient stability. Chai et al.[14] developed a multi-strategy fusion differential evolution algorithm that integrated multiple strategies to optimize path planning in complex scenarios. However, this algorithm could not be used for real-time path planning. In general, these intelligent optimization methods can effectively address the path planning problem for mobile intelligent agents, but they also face some challenges such as computational complexity, low robustness, the presence of local optimal solutions, and ineffective dynamic obstacle avoidance.

This paper proposes an upgrade strategy that extends the two-dimensional dynamic window approach (DWA) to a three-dimensional DWA for local path planning in three-dimensional environments. This 3D DWA is guided by a global path planned using the six-direction search (SDS) method based on neural networks[15]. The algorithm features fewer iterations and shorter path planning time. Furthermore, considering that the fixed weights in the evaluation function of DWA may not be universally applicable in complex and variable environments, using the same weight coefficients in different environments can lead to failure in obstacle avoidance[16]. Zhang et al.[17] proposed an improved DWA integrated with fuzzy logic, which utilized fuzzy logic to analyze information about targets and obstacles to determine suitable weights in real-time. However, this method lacks global information guidance, the mobile intelligent agent is prone to getting stuck in local minima. Therefore, this paper adds a local target point term to establish a new evaluation function to address the issue of local optimality. Adjusting weight parameters through fuzzy logic enables mobile intelligent agent to adapt to environmental changes and improve their real-time obstacle avoidance capabilities. Through extensive simulation and robotic experiments, the performance of the algorithm in terms of path length, minimum safety distance, and obstacle avoidance capabilities has been carefully evaluated. The main contributions of this paper are summarized as follows:

(1) Aiming at the obstacle avoidance problem for mobile agents in three-dimensional space, this paper proposes a six-direction search algorithm based on neural networks, which is combined with an improved dynamic window approach.

(2) This paper extends the dynamic window approach to three-dimensional space and designs a new evaluation function. Additionally, fuzzy logic is introduced to enhance the ability of the mobile agent to adapt to the environment.

(3) The effectiveness of the proposed algorithm is verified through simulations and robotic experiments. The results demonstrate that, in complex three-dimensional environments, this algorithm outperforms other existing algorithms in path planning for mobile intelligent agents.

The rest of this paper is organized as follows. Section 2 introduces the principle of the six-direction search method based on neural networks. Section 3 presents the 3D DWA algorithm and its improvements. Section 4 details the simulation experiments and results of the fusion algorithm in different scenarios. Section 5 verifies the proposed algorithm on a robotic arm through physical experiments. Finally, Section 6 concludes the paper with a brief summary and outlook.

2 Neural Network Based Six-Direction Search Method

In a three-dimensional space with known obstacle information, a neural network is utilized to model the potential field energy of all obstacles in the space, and the distance between the start point and the target point is calculated to obtain the path energy, thereby deriving the weighted energy. Based on the obtained total weighted energy, a six-direction search algorithm is designed to perform path searching, resulting in a collision-free optimal path.

2.1 Obstacle energy modeling

Firstly, obstacles in the three-dimensional space are redefined as spheres that include the size of the agent. Then, a neural network model is employed to obtain the energy distribution of obstacles in the environment.

The input layer of the neural network is set as coordinate points, with the nodes of the first hidden layer representing spatial constraints of obstacles. The nodes of the second hidden layer's functions represent the collision energy function corresponding to each obstacle, and the nodes of the top output layer represent the total collision energy sum of all obstacles.

In this context, the first hidden layer depends on the position and shape of obstacles. Assuming that the i-th obstacle is a spherical obstacle, the neuron function of the first hidden layer can be defined as

$$u_j(x_i, y_i, z_i) = R_c^3 - (x_i - C_{jx})^2 - (y_i - C_{jy})^2 - (z_i - C_{jz})^2 \quad (1)$$

where u_j represents the output of the j-th neuron in the first hidden layer, R_c represents the radius, (C_{jx}, C_{jy}, C_{jz}) is the center of the j-th spherical obstacle, and x_i, y_i, and z_i denote the coordinates of the i-th path node.

The neuron function for the second hidden layer is

$$o_j(u_{j1}, u_{j2}, \ldots, u_{jk}) = \frac{1}{1 + e^{-\left(\sum_{i=1}^{k} u_{ji} - \sigma_{oj}\right)/T}} \quad (2)$$

where o_j represents the output of the j-th neuron in the second hidden layer, u_{jk} represents the jk-th input of the j-th neuron in the second hidden layer, k represents the j-th neuron with k input in the second hidden layer, σ_{oj} represents the tuning parameter of the j-th neuron, oj represents the output of the j-th neuron in the second hidden layer, $\sigma_{oj} = k - 0.5$ (for spherical obstacles, $\sigma_{oj} = 0.5$), and T represents the forward design parameter.

In a three-dimensional space with n obstacles, the energy of each point can be obtained, which is the output layer function

$$q_i(o_1, o_2, \ldots, o_n) = \sum_{j=1}^{n} (o_1, o_2, \ldots, o_n) \quad (3)$$

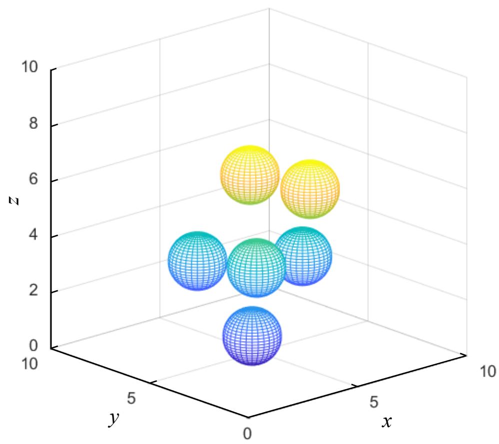
where q_i represents the total output layer energy of the neural network corresponding to the i-th input pair (x_i, y_i), o_n represents the n-th input of the i-th neuron in the output layer, and n represents the i-th neuron with n input in the output layer. Assuming that there are some obstacles present in the three-dimensional space, we can obtain the energy distribution as depicted in Fig. 1.

2.2 Six-direction method

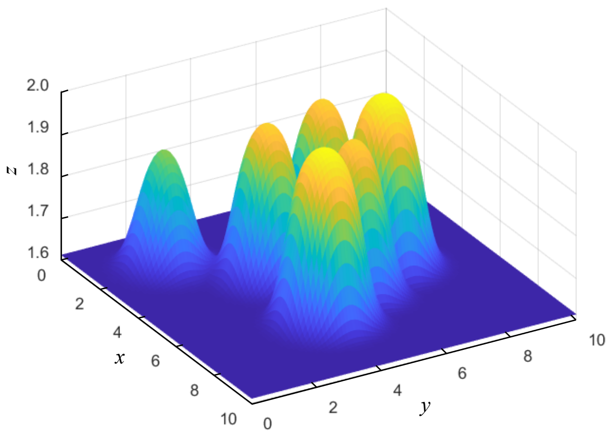
In a three-dimensional space, the movement of any node has six directions. When an agent moves in the space and encounters obstacles, it can perform a search in six directions based on its current position and the distribution of obstacles in the space to obtain a safe path.

Taking the line connecting the start point and the target point as the initial path, it can be equally divided into s segments, resulting in $s + 1$ nodes. For the i-th node, six search directions can be obtained in the space

$$x_i^{(k+1)} = x_i^{(k)} + \delta \cdot \cos\left(j \cdot \frac{\pi}{2}\right) \cdot \sin\left(l \cdot \frac{\pi}{2}\right),$$
$$y_i^{(k+1)} = y_i^{(k)} + \delta \cdot \sin\left(j \cdot \frac{\pi}{2}\right) \cdot \sin\left(l \cdot \frac{\pi}{2}\right),$$



(a) Obstacle distribution in space



(b) Energy distribution in space

Fig. 1 Obstacle distribution and energy distribution in space.

$$z_i^{(k+1)} = z_i^{(k)} + \delta \cdot \cos\left(l \cdot \frac{\pi}{2}\right) \tag{4}$$

where $k+1$ represents the $(k+1)$-th iteration, π denots 180°, $j = 1, 2, 3, 4$, $l = 2, 3, 4$, and δ is the search step size. The six new nodes are connected with the $(i-1)$-th and the $(i+1)$-th nodes to form six optional paths, as shown in Fig. 2.

2.3 Constructing the path energy function

If a mobile agent only considers collision energy during the search process, it may take a detour, resulting in a longer path. To prevent generating overly long paths, path energy is introduced in addition to collision energy. The squared sum of adjacent nodes in the path is taken as the path energy for that segment, and the total path energy is the sum of the path energies of all s line segments. The expression for path segment energy is as follows:

$$P = \sum_{i=0}^{s-1}\left[(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2\right] \tag{5}$$

where P represents the total path energy.

Thus, calculate the collision energy of obstacles and the path energy for each of the six optional paths searched above, and then derive the total energy Q of the path through weighted averaging.

$$Q = \zeta\sum_{j=1}^{n} o_j + \eta\sum_{i=1}^{n}[(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2] \tag{6}$$

where ζ and η are the weighting coefficients. The minimum energy path is obtained by comparing the results of the k-th iteration with the current path.

3 Dynamic Window Approach

The traditional dynamic window approach is mainly designed for path planning in two-dimensional planes, but in three-dimensional spaces, it requires an extension of dimensions to the traditional dynamic window approach, which means considering the velocity components in three directions as well as rotations in multiple angles.

3.1 Kinematic modeling of mobile intelligent agent

As the dynamic window approach requires considering the velocity constraints of the agent and calculating its positional states at different time instances, it is necessary to conduct an analysis of its kinematic state. In a three-dimensional environment, the attitude of an agent is represented by pitch angle θ, yaw angle φ, and roll angle ϕ, as shown in Fig. 3. In the reference coordinate system $(x_\mathrm{f}, y_\mathrm{f}, z_\mathrm{f})$, the kinematic equation of motion for a mobile intelligent agent can be expressed as

$$\begin{bmatrix} x_\mathrm{f}(t+1) \\ y_\mathrm{f}(t+1) \\ z_\mathrm{f}(t+1) \end{bmatrix} = \begin{bmatrix} x_\mathrm{f}(t) \\ y_\mathrm{f}(t) \\ z_\mathrm{f}(t) \end{bmatrix} + \Delta t \cdot R_\mathrm{b}^\mathrm{f} \begin{bmatrix} v_\mathrm{x}^\mathrm{b} \\ v_\mathrm{y}^\mathrm{b} \\ v_\mathrm{z}^\mathrm{b} \end{bmatrix} \tag{7}$$

where $x_\mathrm{f}(t)$, $y_\mathrm{f}(t)$, and $z_\mathrm{f}(t)$ represent the places of the mobile intelligent agent at the time t, respectively. Δt is the sampling time. v_x^b, v_y^b, and v_z^b represent the velocity components on the x, y, and z axes. The superscript b denotes that these three variables are defined in the body coordinate system b. The transformation order between the reference coordinate system $(x_\mathrm{f}, y_\mathrm{f}, z_\mathrm{f})$ and the body coordinate system $(x_\mathrm{b}, y_\mathrm{b}, z_\mathrm{b})$ is in the order of 321 (i.e., first rotation around the z-axis, then around the y-axis, and finally around the x-axis). Therefore, the transformation matrix from the body coordinate system to the reference coordinate system is

$$R_\mathrm{b}^\mathrm{f} = \begin{bmatrix} \cos\varphi\cos\theta & -\sin\varphi\cos\phi + \cos\varphi\sin\theta\sin\phi & \sin\varphi\sin\phi + \cos\varphi\sin\theta\cos\phi \\ \sin\varphi\cos\theta & \cos\varphi\cos\phi + \sin\varphi\sin\theta\sin\phi & -\cos\varphi\sin\phi + \sin\varphi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{8}$$

The relationship between the angular velocity components and the Euler angles is

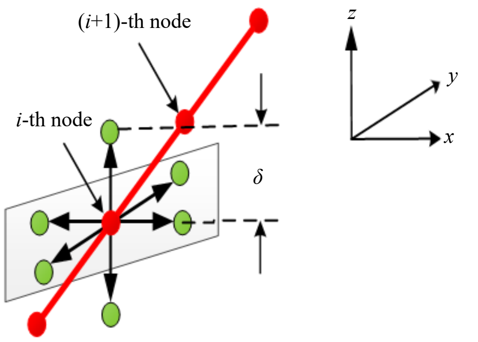


Fig. 2 Six-direction search.

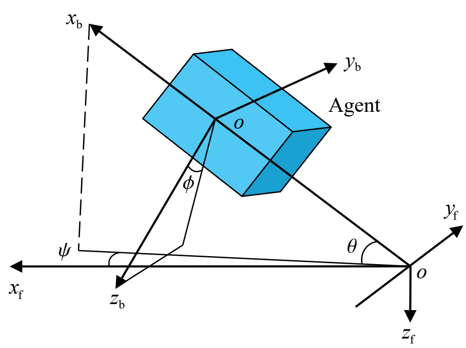


Fig. 3 Agent coordinate system.

$$R(\xi)=\begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \sin\varphi\cos\theta \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix}^{-1} \tag{9}$$

$$\dot{\xi}=R(\xi)\omega \tag{10}$$

where $\xi=[\phi,\ \theta,\ \varphi]^{\mathrm{T}}$, $\omega=[\omega_{\mathrm{x}},\ \omega_{\mathrm{y}},\ \omega_{\mathrm{z}}]^{\mathrm{T}}$, and ω_{x}, ω_{y}, and ω_{z} are the components of the angular velocity ω in the axis of the body coordinate. The equation of the attitude of the agent is written as

$$\begin{bmatrix} \phi_{(t+1)} \\ \theta_{(t+1)} \\ \varphi_{(t+1)} \end{bmatrix}=\begin{bmatrix} \phi_{(t)} \\ \theta_{(t)} \\ \varphi_{(t)} \end{bmatrix}+\dot{\xi}\cdot\Delta t \tag{11}$$

3.2 Three-dimensional dynamic window

3.2.1 Sampling in the velocity space

Compared to mobile intelligent agents in a two-dimensional plane, in three-dimensional space, in addition to the heading angle, motion in the pitch angle direction also needs to be considered (yaw motion is not considered in this paper). Therefore, the velocity constraint for the mobile intelligent agent is

$$V_{\mathrm{S}}=\left\{(v,\ r,\ q)\middle|\ \begin{matrix} v_{\min}\leqslant v\leqslant v_{\max} \\ r_{\min}\leqslant r\leqslant r_{\max} \\ q_{\min}\leqslant q\leqslant q_{\max} \end{matrix}\right\} \tag{12}$$

where v is the resultant velocity of the velocity components along the x_{b}, y_{b}, and z_{b} axes, and q and r are the pitch and yaw angular velocities, respectively.

Additionally, the speed of the mobile agent is also limited by acceleration constraints. Assuming that the current velocity speed of the mobile intelligent agent is v_{c}, the pitch angular velocity is q_{c}, and the yaw angular velocity is r_{c}, the achievable velocity space at the next time step is

$$V_{\mathrm{d}}=\left\{(v,\ r,\ q)\middle|\ \begin{matrix} v_{\mathrm{c}}-\dot{v}\cdot\mathrm{d}t\leqslant v\leqslant v_{\mathrm{c}}+\dot{v}\cdot\mathrm{d}t \\ r_{\mathrm{c}}-\dot{r}\cdot\mathrm{d}t\leqslant r\leqslant r_{\mathrm{c}}+\dot{r}\cdot\mathrm{d}t \\ q_{\mathrm{c}}-\dot{q}\cdot\mathrm{d}t\leqslant q\leqslant q_{\mathrm{c}}+\dot{q}\cdot\mathrm{d}t \end{matrix}\right\} \tag{13}$$

Lastly, to prevent the mobile intelligent agent from colliding with obstacles, it is necessary to consider the braking distance of the mobile agent, which ensures that at any given velocity, the mobile intelligent agent can decelerate to zero before reaching the obstacle. Assuming that it takes m time intervals of $\mathrm{d}t$ to come to a complete stop, the braking distance is denoted as L.

$$\begin{cases} v(t+1)=v(t)-\dot{v}\cdot\mathrm{d}t, \\ L=\displaystyle\sum_{t=1}^{m}(v(t)\cdot\mathrm{d}t) \end{cases} \tag{14}$$

Then the condition that the speed in the collision feasible region needs to satisfy is

$$V_{\mathrm{a}}=\{(v,\ r,\ q)\,|\,D(v,\ r,\ q)>L\} \tag{15}$$

where $D(v,r,q)$ is the shortest distance from the mobile intelligent agent to the obstacle at the current moment.

The intersection of these three constraints defines the final range of operable velocities for the mobile intelligent agent $V=V_{\mathrm{s}}\cap V_{\mathrm{d}}\cap V_{\mathrm{a}}$.

3.2.2 Evaluation function

Among all the possible velocity combinations, the optimal one is selected through an evaluation function G, which is as follows:

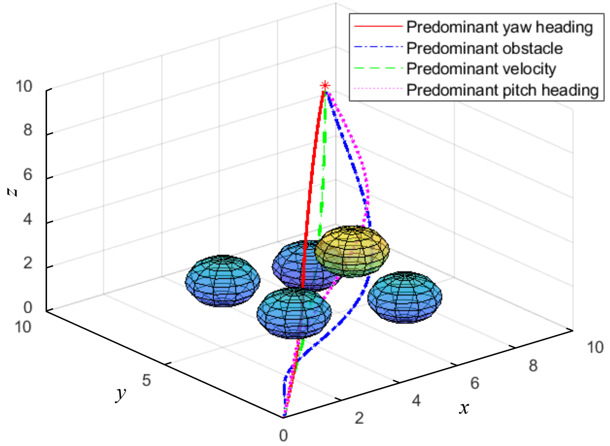
$$\begin{aligned} G(v,\ r,\ q)=\ &\alpha\cdot\mathrm{heading}_{\mathrm{yaw}}(v,\ r,\ q)+ \\ &\beta\cdot\mathrm{heading}_{\mathrm{pitch}}(v,\ r,\ q)+ \\ &\gamma\cdot\mathrm{dist}(v,\ r,\ q)+\mu\cdot\mathrm{speed}(v,\ r,\ q) \end{aligned} \tag{16}$$

where $\mathrm{heading}_{\mathrm{yaw}}(v,\ r,\ q)$ represents the difference in heading angle between the mobile intelligent agent and the target direction, $\mathrm{heading}_{\mathrm{pitch}}(v,\ r,\ q)$ represents the difference in pitch angle between the mobile intelligent agent and the target direction, $\mathrm{dist}(v,\ r,\ q)$ represents the distance between the mobile intelligent agent and the obstacle, and $\mathrm{speed}(v,\ r,\ q)$ is an evaluation term for the magnitude of the velocity. α, β, γ, and μ are the weighting factors, $\alpha,\ \beta,\ \gamma,\ \mu\in[0,\ 1]$.

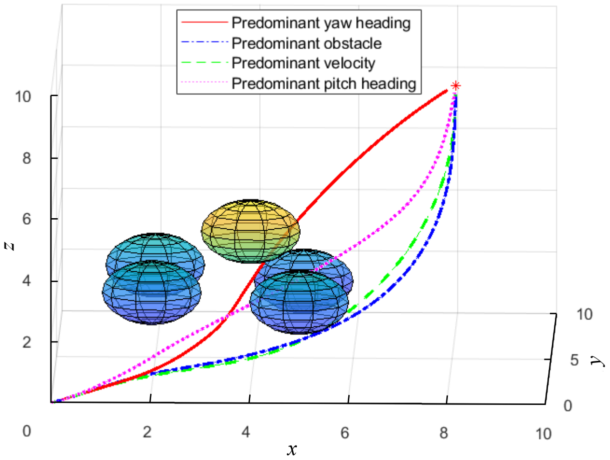
3.3 Improvement of 3D DWA

In the traditional dynamic window approach algorithm, the weighting coefficients of the trajectory evaluation function play a decisive role in the final trajectory selection. Typically, the weighting coefficients are fixed, making it difficult for the mobile intelligent agents to adapt to the needs of path planning in complex and dynamic environments. Therefore, when determining the values of the weighting coefficients, multiple factors need to be considered to ensure that the mobile intelligent agent can obtain the optimal path planning even in complex environments. To test the effect of each evaluation function's weight ratio, each evaluation term is set as the primary parameter one by one.

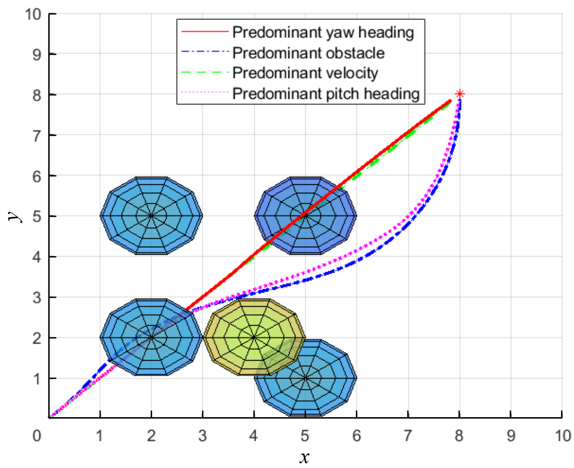
As can be seen from Fig. 4 and Table 1, when the yaw heading angle term is the primary weight, the mobile intelligent agent exhibits strong target directivity, and the algorithm will prioritize planning paths towards the target point, making it suitable for relatively open environments. When the distance to obstacle term is the primary weight, the mobile intelligent agent has stronger obstacle avoidance



(a) Front view



(b) Side view



(c) Top view

Fig. 4 Path planning result with different evaluation function parameters set as the primary weight.

capability, and the algorithm will prioritize planning paths further away, resulting in higher safety in avoiding single obstacles. When the speed term is the primary weight, the algorithm will prioritize planning the fastest path with the fewest iterations, but this may easily lead the mobile intelligent agents to bypass obstacles, resulting in longer planned paths, and faster speeds may compromise the safety of the mobile intelligent agents. When the pitch angle term is the primary weight, it causes the mobile intelligent agent to place more emphasis on pitch heading angle changes in path planning and action decision-making. Specifically, this may make the mobile intelligent agent focus more on adjustments and optimizations in the vertical direction, potentially sacrificing overall movement efficiency or flexibility in avoiding obstacles.

Table 1 Result of the parameter test.

Term	Planning result	Number of iterations	Path length (m)
Predominant yaw heading	Success	219	14.15
Predominant obstacle	Success	189	15.46
Predominant velocity	Success	164	15.99
Predominant pitch heading	Success	196	14.34

Based on the above analysis, fixed weighting coefficients for the evaluation function cannot meet the requirements for optimal trajectory planning of mobile intelligent agents. When the environment changes, it is difficult for mobile intelligent agents to adapt and make corresponding changes. Therefore, this paper introduces fuzzy rules based on the traditional dynamic window approach to dynamically adjust the weights of the evaluation function.

3.3.1 Design of fuzzy controller

The core of the fuzzy controller is to use fuzzy rules to describe the relationship between inputs and outputs. It achieves control of the system by fuzzifying the inputs and performing fuzzy reasoning through a series of fuzzy rules.

The input section of the fuzzy controller in this paper consists of the distance DistGoal between the mobile intelligent agent and the target point, and the closest distance DistObstacle between the mobile intelligent agent and the obstacle. The output section comprises the weights α, β, γ, and μ of the evaluation function. The specific approach is as follows:

(1) When both the distance between the mobile intelligent agent and the target and the distance to the obstacle are relatively large, the mobile intelligent agent should prioritize speed and be biased towards moving towards the target point, thereby increasing the weight of the term related to the direction towards the target.

(2) When both the distance between the mobile

intelligent agent and the target point and the distance to the obstacle are relatively small, the mobile intelligent agent prioritizes avoiding the obstacle and reduces its speed.

(3) When the distance between the mobile intelligent agent and the target point is relatively large, but the distance to the obstacle is small, the mobile intelligent agent prioritizes avoiding the obstacle and does not rush to get closer to the target.

(4) When the distance between the mobile intelligent agent and the target point is relatively small, but the distance to the obstacle is large, the mobile intelligent agent prioritizes moving closer to the target point while reducing its speed.

Based on the above ideas, the formulated fuzzy control rules are shown in Table 2.

Table 2 Fuzzy control rule.

Rule	Input		Output			
	DistGoal	DistObstacle	α	β	γ	μ
1	ZS	ZS	ZS	ZS	PB	ZS
2	ZS	PS	PS	PS	PB	ZS
3	ZS	PM	PS	PS	PM	PS
4	ZS	PB	PS	PS	PS	PM
5	ZS	PH	PS	PS	ZS	PB
6	PS	ZS	ZS	ZS	PH	ZS
7	PS	PS	PS	PS	PB	ZS
8	PS	PM	PS	PS	PM	PS
9	PS	PB	PS	PS	PS	PM
10	PS	PH	PS	PS	ZS	PB
11	PM	ZS	PS	PS	PH	ZS
12	PM	PS	PS	PS	PB	ZS
13	PM	PM	PS	PS	PM	PS
14	PM	PB	PS	PS	PS	PM
15	PM	PH	PS	PS	ZS	PB
16	PB	ZS	PS	PS	PH	ZS
17	PB	PS	PS	PS	PB	PS
18	PB	PM	PM	PM	PM	PM
19	PB	PB	PM	PM	PS	PM
20	PB	PH	PB	PB	PS	PB
21	PH	ZS	PS	PS	PH	ZS
22	PH	PS	PS	PS	PH	PS
23	PH	PM	PM	PM	PB	PM
24	PH	PB	PB	PB	PM	PB
25	PH	PH	PB	PB	PS	PH

Note: ZS, PS, PM, PB, and PH are defined on the fuzzy set, representing the distance between the mobile robot and the target point as a smaller value, small value, medium value, large value, and larger value respectively, with the universe of discourse being [0, 1].

3.3.2 DWA based on fuzzy control (FC-DWA)

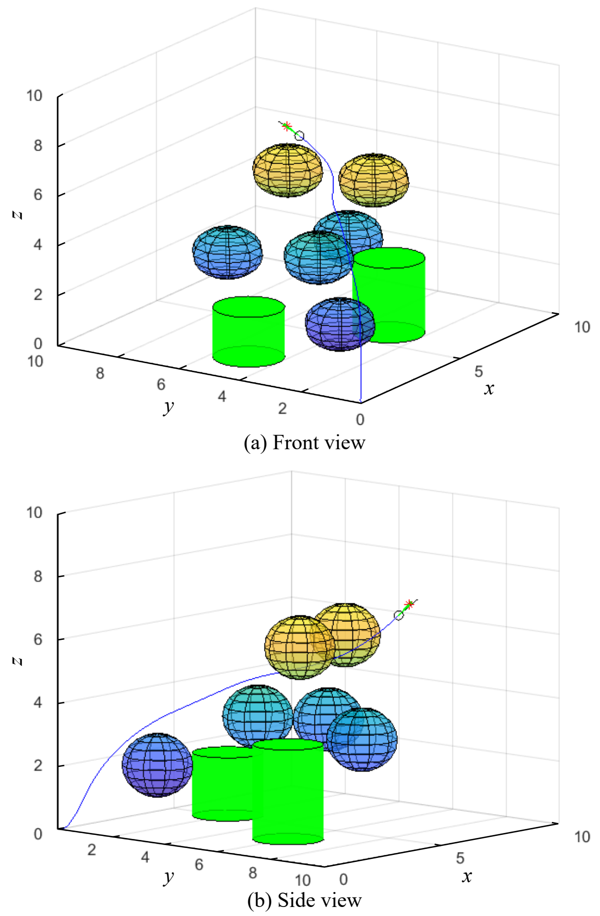
The fuzzy controller can adapt to changes in the environment by selecting different weights for the evaluation function based on the distances between the mobile intelligent agent, the target point, and obstacles. To verify the effectiveness of the proposed method, simulation experiments were conducted in MATLAB. The simulations were performed using MATLAB R2022a 64-bit on a 2.4 GHz Intel Core i5-9300H processor, and comparisons were made with the traditional dynamic window approach. Table 3 lists the parameters used in the experiments, and the simulation results are presented below.

Figure 5 shows the path planned using the FC-DWA algorithm, with 138 iterations and a path length of 14.11 m, significantly reduces the number of iterations and shortens the path length compared to the traditional fixed-weight approach, meeting the requirements of mobile intelligent agent path planning. Furthermore, as can be seen from Fig. 6, the weights of the evaluation function are constantly adjusted to adapt to changes in the environment, ensuring that the mobile intelligent agent reaches the target point as quickly as possible while maintaining safety.

It can be observed that when the mobile intelligent agent is initially far away from both the target point and the obstacle, the speed term is set as the primary weight. As the mobile intelligent agent moves within the obstacle avoidance range, the weight of the distance-to-obstacle term gradually increases, reaching nearly 0.9. After bypassing the obstacle, the speed term gradually increases again, allowing the mobile intelligent agent to advance towards the target point.

Table 3 Mobile intelligent agent and obstacle parameter.

Parameter	Value
Minimum linear velocity (m/s)	0
Maximum linear velocity (m/s)	2
Limited linear acceleration (m/s^2)	0.8
Limited yaw heading angular velocity (deg/s)	20
Limited yaw heading angular acceleration (deg/s^2)	20
Limited pitch heading angular velocity (deg/s)	20
Limited pitch heading angular acceleration (deg/s^2)	20
Linear velocity resolution (m/s)	0.05
Angular velocity resolution (deg/s)	2
Prediction time (s)	3
Time resolution (s)	0.1
Obstacle radius (m)	1



(a) Front view

(b) Side view

Fig. 5 FC-DWA algorithm planning result.

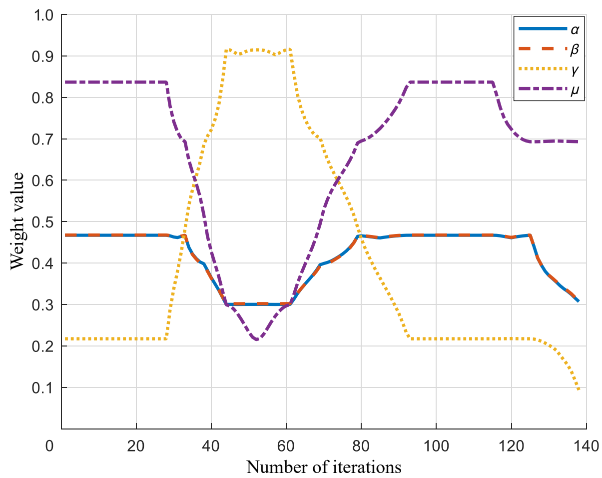


Fig. 6 Change in the weight of the evaluation function.

3.3.3 FC-DWA integrated with global algorithm

The FC-DWA algorithm proposed above can achieve obstacle avoidance for both static and dynamic obstacles. However, as seen from the path results of the simulation experiments, the path planned by the FC-DWA algorithm sometimes still cannot guarantee global optimality. Therefore, a single algorithm cannot perfectly meet the requirements of mobile intelligent agent path planning in complex environments. To address this issue, the combination of the six-direction search method and the FC-DWA algorithm is proposed to make up for this shortcoming. The optimal trajectory points obtained by the six-direction search algorithm are used as global target guidance points, combined with the FC-DWA algorithm for real-time dynamic local path planning. This drives the mobile intelligent agent to navigate and move based on the global target guidance points while avoiding dynamic obstacles on the route until reaching the destination.

To ensure that the mobile intelligent agent achieves optimal overall path planning while also taking into account its safety, the original variable $\mathrm{heading}_{\mathrm{yaw}}(v, r, q)$ is revised to $\mathrm{Gheading}_{\mathrm{yaw}}(v, r, q)$, representing the heading angle difference between the mobile intelligent agent and the local target direction; similarly, the original variable $\mathrm{heading}_{\mathrm{pitch}}(v, r, q)$ is revised to $\mathrm{Gheading}_{\mathrm{pitch}}(v, r, q)$, representing the pitch angle difference between the mobile intelligent agent and the local target direction. The other two variables remain unchanged. The revised dynamic window evaluation function, incorporating the six-direction method, can be translated as follows:

$$
\begin{aligned}
G(v, r, q) = {} & \alpha \cdot \mathrm{Gheading}_{\mathrm{yaw}}(v, r, q) + \\
& \beta \cdot \mathrm{Gheading}_{\mathrm{pitch}}(v, r, q) + \\
& \gamma \cdot \mathrm{dist}(v, r, q) + \mu \cdot \mathrm{speed}(v, r, q)
\end{aligned} \tag{17}
$$

The algorithm flowchart is shown in Fig. 7.

4 Simulation Experiment and Analysis

To evaluate the superiority and adaptability of the FC-DWA algorithm integrated with global algorithms compared to the traditional DWA algorithm in complex static environments and environments with dynamic obstacles, this paper designs two simulation experiments for verification and analysis. The experimental environments and parameters are consistent with those mentioned above.

4.1 Path planning for complex static obstacle environment

From Fig. 8, it can be observed that the SDS method enables the path to transition from the initial position (i.e., the line connecting the starting point and the goal point) to the final position, while successfully avoiding the obstacles. Furthermore, Fig. 8 demonstrates that the

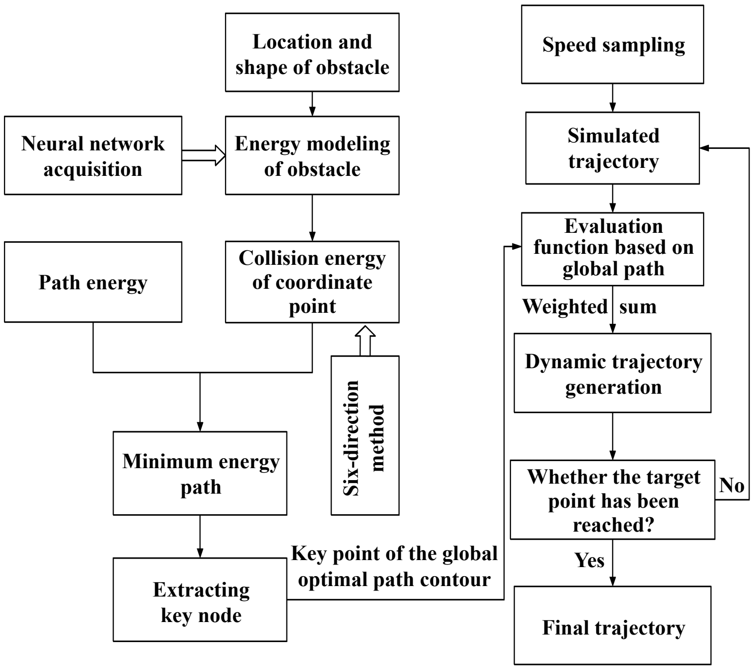


Fig. 7 Algorithm flowchart.

total energy decreases continuously from the initial state to the final state, with an iteration count of 168. These simulation results indicate that the SDS scheme is capable of finding an optimal path in the given environment.

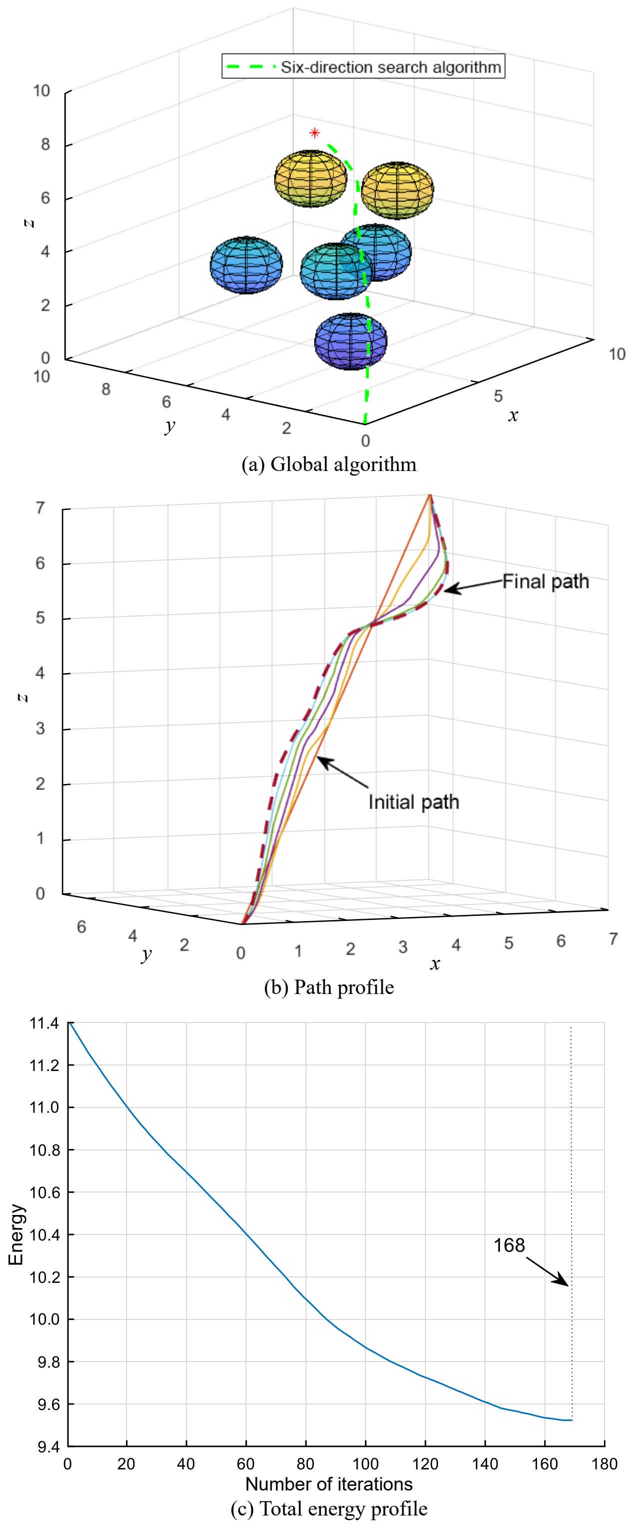
As shown in Fig. 9, all algorithms are capable of planning a final path. However, due to the lack of global guidance, the traditional DWA and FC-DWA algorithms do not produce the optimal trajectory. After avoiding the first obstacle at (2, 2, 2), they fail to follow the direction of the global path, opting to bypass the dense obstacle area to reach the target point, significantly increasing the path length. In contrast, the FC-DWA combined with the global algorithm, guided by the globally optimal path, successfully navigates through the dense obstacle area, demonstrating superior global tracking capability.

According to Table 4 and Fig. 10, the FC-DWA combined with the global algorithm outperforms the traditional DWA algorithm by reducing the path length by 3.6% and increasing the minimum safety distance by 31.9%. Compared to the FC-DWA algorithm, it shortens the path length by 1.1% and improves the minimum safety distance by 6.5%. This indicates that while the traditional DWA algorithm can reach the target point in a short time, it has lower safety margins. In addition, three commonly used path planning algorithms, namely the artificial potential field algorithm, rapidly-exploring random tree algorithm, and the optimized version of RRT known as RRT* algorithm, are employed for comparison with the algorithm presented in this paper. Based on the comparison results, the algorithm proposed in this paper generates shorter and safer paths. Therefore, considering both computational efficiency and safety, the FC-DWA combined with the global algorithm is superior to the standalone FC-DWA, as well as the traditional DWA, APF, RRT, and RRT* algorithms.

4.2 Path planning with dynamic obstacle

To verify the path planning capability of the proposed algorithm in dynamic environments, this paper sets up a 10 m × 10 m map with a starting point at (1, 1, 1) and an endpoint at (6, 6, 6). Subsequently, two dynamic obstacles are added to this map, with coordinates at (2, 4, 3) and (4, 2, 5), respectively. These obstacles move at the same speed as the mobile intelligent agent, traveling back and forth vertically. Static obstacles are positioned at (7, 4, 2), (7, 1, 3), (5, 8, 5), and (4, 4, 4), each with a radius of 1 m.

Figure 11 depicts several moments of the mobile intelligent agent avoiding obstacles. The mobile intelligent agent detects dynamic obstacles around it and begins to move downwards. The red dashed line represents the global path, while the blue solid line shows the mobile intelligent agent’s actual movement



(a) Global algorithm

(b) Path profile

(c) Total energy profile

Fig. 8 Synthetic path analysis.

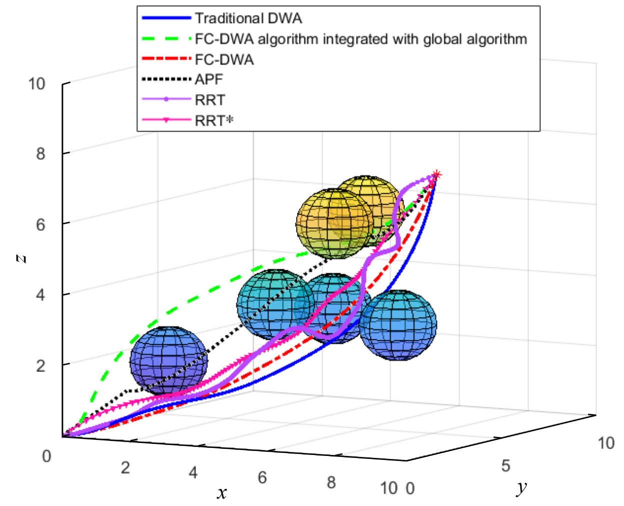


Fig. 9 Result of path planning using different algorithms. APF is artificial potential field, RRT is rapidly-exploring random tree.

Table 4 Algorithm data comparison.

Method	Number of iterations	Path length (m)	Minimum distance from obstacle (m)
Traditional DWA	188	12.88	0.210
FC-DWA	162	12.55	0.260
FC-DWA algorithm integrated with global algorithm	504	12.41	0.277
APF	388	13.50	0.302
RRT	429	14.88	0.168
RRT*	300	13.25	0.182

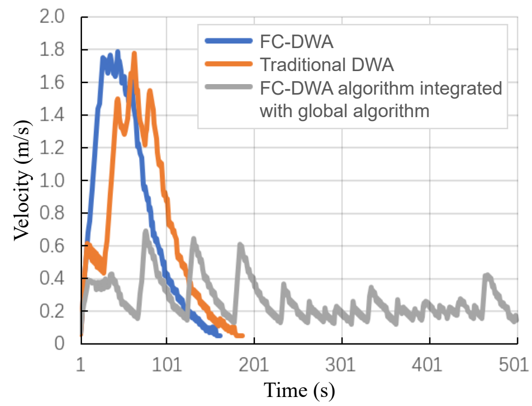
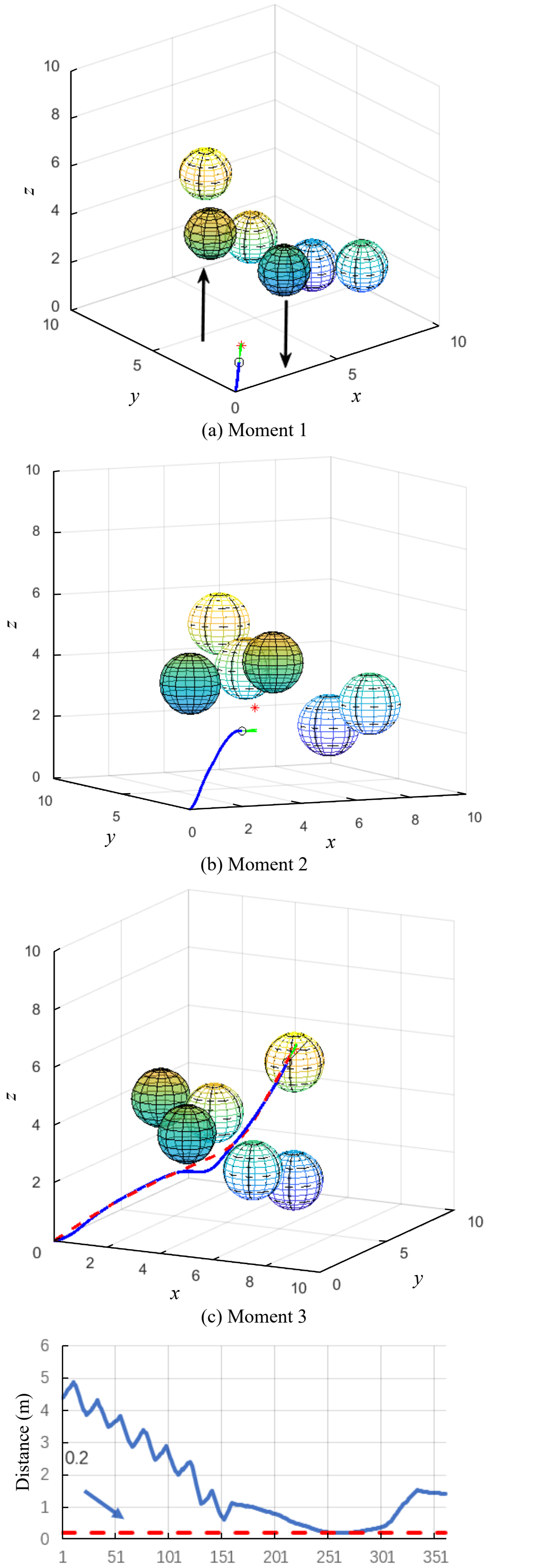


Fig. 10 Speed comparison of different algorithms.

trajectory. It can be observed that after avoiding the dynamic obstacles, the mobile intelligent agent promptly tracks the globally optimal path. The overall planned path is smooth, and the minimum safety distance from obstacles is maintained at 0.2 m, as shown in Fig. 11. This demonstrates the superiority of the algorithm proposed in this paper.

5 Physical Verification

Finally, to verify the effectiveness of the proposed



(a) Moment 1

(b) Moment 2

(c) Moment 3

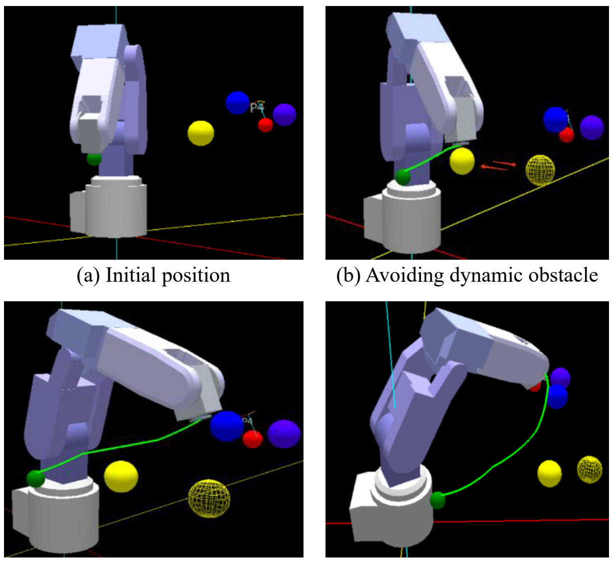
(d) Minimum distance

Fig. 11 Experimental result in 3D dynamic environment.

algorithm in this paper when applied to mobile intelligent agents, the experimental simulation path information was input into the Mitsubishi robotic arm simulation simulator to drive the joints of the robotic arm, enabling the mobile end-effector of the robotic arm to reach the target position while avoiding all obstacles. The experimental parameters are as follows. In Fig. 12, the starting point (110, −155, 230) of the end effector of the robotic arm is indicated by the green ball, the target point (405, 185, 325) is indicated by the red ball, Obstacle 1 (435, 42, 375) and Obstacle 2 (475, 175, 350) are indicated by the blue balls, and the moving obstacle (530, −135, 295) is indicated by the yellow ball, moving back and forth along the x-axis. The radius of each obstacle is 30 mm, with all units in millimeters. Figure 12 depicts the movement process of the robotic arm in the simulator. Due to the influence of the moving obstacle, the path is replanned, and the robotic arm moves upwards. After adjustment, it successfully avoids the moving obstacle. Figure 13 shows the movement process of the robotic arm at the destination in the actual environment. The experimental results indicate that the robotic arm can bypass the moving obstacle and reach the destination, verifying the effectiveness of the algorithm and its applicability in practical applications.

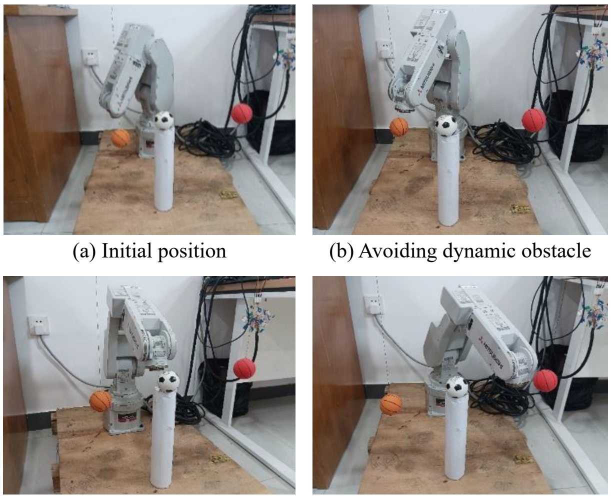
6 Conclusion and Future Prospect

This paper introduces the FC-DWA algorithm integrated with a global algorithm to address the path


(a) Initial position (b) Avoiding dynamic obstacle

(c) Avoiding static obstacle (d) Reaching the target

Fig. 12 Dynamic obstacle avoidance planning diagram of robot arm.



(a) Initial position (b) Avoiding dynamic obstacle

(c) Avoiding static obstacle (d) Reaching the target

Fig. 13 Motion planning diagram of robot arm in actual scene.

planning problem for mobile intelligent agents in complex three-dimensional spaces. Firstly, a six-direction search algorithm is utilized to generate a globally feasible path, from which key points are extracted to serve as local target points for global guidance. Subsequently, by dynamically adjusting the weights of the DWA algorithm, the FC-DWA algorithm effectively solves the obstacle avoidance problem for mobile intelligent agents in complex and dynamic environments. Consequently, the mobile intelligent agents are able to achieve global dynamic path planning in complex three-dimensional spaces. Future research directions include the coordination of multiple mobile intelligent agents working together in complex three-dimensional spaces.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61663003) and the PhD Start-up Foundation of Guangxi University of Science and Technology (No. 12Z05).

References

[1] X. Gu, L. Liu, L. Wang, Z. Yu, and Y. Guo, Energy-optimal adaptive artificial potential field method for path planning of free-flying space robots, *J. Franklin Inst.*, vol. 361, no. 2, pp. 978–993, 2024.

[2] H. Li, L. Zhang, T. Xiao, and J. Dong, Synergic motion trajectory planning for airplane docking based on 6PURU parallel mechanism, *Tsinghua Science and Technology*, vol. 20, no. 2, pp. 188–199, 2015.

[3] R. An, S. Guo, L. Zheng, H. Hirata, and S. Gu, Uncertain moving obstacles avoiding method in 3D arbitrary path planning for a spherical underwater robot, *Rob. Auton. Syst.*, vol. 151, p. 104011, 2022.

[4] X. Fu, Y. Jiang, D. Huang, K. Huang, and J. Wang, Trajectory planning for automated driving based on ordinal optimization, *Tsinghua Science and Technology*, vol. 22, no. 1, pp. 62–72, 2017.

[5] K. Zhu and T. Zhang, Deep reinforcement learning based mobile robot navigation: A review, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021.

[6] J. Ni, L. Wu, P. Shi, and S. Yang, A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles, *Comput. Intell. Neurosci.*, vol. 2017, p. 9269742, 2017.

[7] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, and M. S. Muhammad, RRT*-SMART: A rapid convergence implementation of RRT, *Int. J. Adv. Robot. Syst*., vol. 10, no. 7, p. 299, 2013.

[8] X. Cao, D. Zhu, and S. Yang, Multi-AUV target search based on bioinspired neurodynamics model in 3D underwater environments, *IEEE Trans. Neural Network. Learn. Syst.*, vol. 27, no. 11, pp. 2364–2374, 2016.

[9] L. Jin, Z. Su, D. Fu, and X. Xiao, Coevolutionary neural solution for nonconvex optimization with noise tolerance, *IEEE Trans. Neural Network. Learn. Syst.*, vol. 35, no. 12, pp. 17571–17581, 2024.

[10] M. Boulares, A. Fehri, and M. Jemni, UAV path planning algorithm based on deep Q-learning to search for a floating lost target in the ocean, *Rob. Auton. Syst*., vol. 179, p. 104730, 2024.

[11] X. Zhang, X. Lu, S. Jia, and X. Li, A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning, *Appl. Soft Comput.*, vol. 70, pp. 371–388, 2018.

[12] M. Liu, Y. Li, Y. Chen, Y. Qi, and L. Jin, A distributed competitive and collaborative coordination for multirobot systems, *IEEE Trans. Mob. Comput.*, vol. 23, no. 12, pp. 11436–11448, 2024.

[13] D. Adhikari, E. Kim, and H. Reza, A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization, in *Proc. 2017 IEEE Congress on Evolutionary Computation*, Donostia, Spain, 2017, pp. 2258–2265.

[14] X. Chai, Z. Zheng, J. Xiao, L. Yan, B. Qu, P. Wen, H. Wang, Y. Zhou, and H. Sun, Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment, *Aerosp. Sci. Technol.*, vol. 121, p. 107287, 2022.

[15] K. Li, L. Li, C. Tang, W. Lu, and X. Fan, Three-dimensional path planning based on six-direction search scheme, *Sensors*, vol. 24, no. 4, p. 1193, 2024.

[16] E. J. Molinos, Á. Llamazares, and M. Ocaña, Dynamic window based approaches for avoiding obstacles in moving, *Rob. Auton. Syst.*, vol. 118, pp. 112–130, 2019.

[17] H. Zhang, C. Sun, Z. Zheng, W. An, D. Zhou, and J. Wu, A modified dynamic window approach to obstacle avoidance combined with fuzzy logic, in *Proc. 2015 14th Int. Symp. Distributed Computing and Applications for Business Engineering and Science*, Guiyang, China, 2015, pp. 523–526.

Kene Li received the BS degree in high polymer materials from South China University of Technology, China, in 2002, the MS degree in signal and information processing from Guangdong University of Technology, China, in 2007, and the PhD degree from Sun Yat-sen University, Guangzhou, China, in 2011, respectively. He is currently an associate professor at School of Automation, Guangxi University of Science and Technology, China. From May 2017 to May 2018, he worked as a visiting scholar at Department of Mechanical, Industrial and Systems Engineering, The University of Rhode Island, Kingston, RI, USA. His research interests include robotics and intelligent control.

Wanning Lu received the BS degree in electrical engineering and automation from Wanjiang University of Technology, China, in 2022. He is currently pursuing the MS, degree in automation at Guangxi University of Science and Technology, China. His research interests include path planning and robotics.

Chunyi Tang received the PhD degree from Guangzhou Institute of Chemistry, Chinese Academy of Sciences, Guangzhou, China, in 2011. She is currently an associate professor at Guangxi University of Science and Technology, China. From May 2017 to May 2018, she worked as a visiting scholar at College of Arts and Sciences, The University of Rhode Island, Kingston, RI, USA. Her research interests include functional materials and robotics technology.

Chengzhi Yuan received the BS and MS degrees in control theory and applications from South China University of Technology, Guangzhou, China, in 2009 and 2012, respectively, and the PhD degree in mechanical engineering from North Carolina State University, Raleigh, NC, USA, in 2016. He is currently an associate professor at Department of Mechanical, Industrial and Systems Engineering The University of Rhode Island, Kingston, RI, USA, and the director of the Intelligent Control & Robotics (IcRobots) Laboratory. His research interests include dynamic systems and control theory, provable adaptive learning and control, hybrid systems, and multi-robot distributed control. He has authored and co-authored over 140 journal articles and conference papers. He has served extensively as an associate editor, chair and co-chair, and program committee member in numerous international conferences. In particular, he served in the technical program committee for the American Control Conference 2018, and the scientific committee for the International Conference on Mechanical System Dynamics in 2021. He served as a guest editor for special issue in the journal of *Frontiers in Robotics and AI* in 2023, the journal of *IET Control Theory & Applications* in 2018 and 2022, and the journal of *Advances in Mechanical Engineering* in 2017. He has been serving as an associate editor for the editorial board of the IEEE Control System Society and the editorial board of the American Society of Mechanical Engineering (ASME) since 2018. He has been an editorial board member of the *International Journal of Mechanical System Dynamics* since 2021.