International Journal of Control, Automation, and Systems 22(4) (2024) 1324-1336
http://dx.doi.org/10.1007/s12555-022-1172-7
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

A Path Planning Method for Unmanned Surface Vessels in Dynamic Environment

Jiabin Yu*, Zhihao Chen, Zhiyao Zhao, Jiping Xu, and Yang Lu

Abstract: A path planning method for unmanned surface vessels (USV) in dynamic environment is proposed to address the impact of dynamic environments on path planning results and the lack of dynamic obstacle avoidance capabilities. First, the considering ocean current rapidly exploring random tree (RRT*) (COC-RRT*) algorithm was proposed for global path planning. The RRT* algorithm has been enhanced with the integration of the virtual field sampling algorithm and ocean current constraint algorithm. The COC-RRT* algorithm optimizes the global planning path by adjusting the path between the parent nodes and child nodes. Second, according to the limitations of the International Regulations for Preventing Collisions at Sea (COLREGs), the improved dynamic window approach (DWA) is applied for local path planning. To enhance the ability of avoid dynamic obstacles, the dist function in the DWA algorithm has been improved. Simulation experiments were conducted in three scenarios to validate the proposed algorithm. The experimental results demonstrate that, in comparison with other algorithms, the proposed algorithm effectively avoids dynamic obstacles and mitigates the influence of the space-varying ocean current environment on the path-planning outcome. Additionally, the proposed algorithm exhibits high efficiency and robustness. The results verified the effectiveness of the proposed algorithm in dynamic environments.

Keywords: COLREGs, DWA algorithm, path planning, RRT algorithm, space-varying ocean current.

1. INTRODUCTION

In recent years, unmanned surface vessels (USVs) have seen widespread application in conducting water quality sampling in public water bodies. The presence of ocean currents and dynamic obstacles in the dynamic environment has a negative impact on the safety of USV navigation. Therefore, scholars have done much research on USV navigation technology in the dynamic environment, and path planning is an active research area of USV navigation technology [1].

Currently, there are diverse path planning methods, broadly categorized into global path planning methods and local path planning methods [2]. The global path planning method involves searching for the optimal path within a pre-constructed environmental model. Therefore, the global path planning method is commonly employed for solving path planning problems in spatially changing ocean current environments. The global path planning methods primarily include A* algorithm [3], Dijkstra algorithm [4], rapidly-exploring random trees (RRT) [5], and deep reinforcement learning (DRL)-based algorithms [6]. The local path planning method can use sensors to obtain information about the surrounding environment and autonomously plan collision free paths in partially unknown environments, making it ideal for dynamic unknown environments. The commonly used local path planning methods are divided into two categories. The first type refers to path planning methods based on virtual potential fields, such as the artificial potential field (APF) method [7]. This method features a simple structure and high efficiency. The second category encompasses sampling-based path planning methods, such as the dynamic window approach (DWA) [8]. These methods are characterized by their fast sampling speed, and these methods are often used to solve local path planning problems [9].

In global path planning algorithms, RRT algorithm has the dominance of fast sampling speed, and is widely utilized for solving multi-objective path planning problems in spatially changing ocean current environments. However, the path generated by this algorithm has longer

Manuscript received December 7, 2022; revised March 17, 2023; accepted April 20, 2023. Recommended by Associate Editor HyungGi Jo under the direction of Senior Editor Hyun Myung. This work was supported in part by the National Key Research and Development Program of China(2022YFF1101103), the Project of Cultivation for young top-notch Talents of Beijing Municipal Institutions (BPHR202203043).

Jiabin Yu, Zhihao Chen, Zhiyao Zhao, Jiping Xu, and Yang Lu are with the School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China; Beijing Laboratory for Intelligent Environmental Protection, Beijing Technology and Business University, Beijing 100048, China; Key Laboratory of Industrial Internet and Big Data, China National Light Industry, Beijing Technology and Business University, Beijing 100048, China (e-mails: yujiabin@th.btbu.edu.cn, 2130062048@st.btbu.edu.cn, zhaozy@btbu.edu.cn, xujp@th.btbu.edu.cn, 2230602063@st.btbu.edu.cn).
* Corresponding author.

Table 1. Global performance of path planning algorithm.

Algorithm	Reduced the cost	Ocean current	Obstacles and current combination
RRT [5]	/	/	/
A boundary-RRT* [10]	Yes	/	/
A dynamic-RRT* [11]	Yes	/	/
Energy optimized D* [12]	Yes	Yes	/
FLFF-controller [13]	Yes	Yes	/
OCI-RRT [14]	Yes	Yes	/

length. To address this issue, Park *et al*. [10] proposed a boundary RRT * algorithm that improves real-time performance by simply calculating and configuring spatial boundaries. The cost of path planning is reduced using this algorithm. In addition, a decentralized iterative algorithm has been proposed by Verbari *et al*. [11] based on a single agent dynamic RRT star, which introduces a decentralized strategy that compares the allocation process based on iterative plans to reduce the length of planned paths. While the algorithm mentioned above does decrease the path length of the RRT algorithm, it fails to account for the ocean current issue in real water environments and is also not practical in complex environments. To address this issue, Sun *et al*. [12] believe that the energy consumption of AUVs is significantly influenced by ocean currents. Then, they developed a cost model to quantify the influence of ocean currents on AUV energy consumption and integrated this model into the D* path planning algorithm, enabling AUVs to utilize ocean currents to achieve energy consumption reduction. Peng *et al*. [13] considered ocean current disturbances and used neural networks combined with adaptive filtering methods to extract model uncertainty and low-frequency components of ocean disturbances, achieving collaborative control of multiple USVs in ocean current environments. An enhanced ocean current based RRT (IOC-RRT) algorithm was proposed by Lan *et al*. [14] to sort out path planning problems encountered in practical applications. Although the above algorithm takes into account the path planning and collaborative control of multiple unmanned submersibles in the current environment, it does not take into account the multi-objective task allocation problem of multiple unmanned submersibles. Furthermore, although it considers the impact of ocean currents, it does not consider obstacles and currents together. Therefore, this algorithm is ineffective in complex marine environments. Table 1 presents the performance evaluation of the aforementioned algorithms.

In the local path planning algorithm, the DWA algorithm sets a speed preselection window based on the current speed and acceleration of the USV, and an evaluation function was used to select the optimal speed on the basis of heading angle, obstacle avoidance, and speed. It takes into account the physical and environmental constraints of the USV, ensuring good global and real-time path planning performance. Therefore, the DWA algorithm has a wide application in local path planning. In order to improve the path planning capabilities of the DWA algorithm, Yang *et al*. [15] introduced fuzzy control. Bai *et al*. [16] defined the safety threshold of the objective function to use the DWA algorithm to reduce path length. The above methods can improve the path planning capability of the DWA algorithm. However, they lack standards for obstacle avoidance in the presence of ships. The Convention on International Regulations for Preventing Collisions at Sea (COLREGs) clarifies the general rules for navigation decision-making in the event of a ship encountering a collision, constrains the navigation decision-making of ships, and improves maritime traffic safety [17]. Some studies have combined the DWA algorithm with COLREGs to enable USVs to avoid dynamic obstacles based on COLREGs constraints [18,19]. The speed of obstacles should also be considered when avoiding obstacles. Chen *et al*. [20] combined the mutual speed between USV and dynamic obstacles, the sampling ability and the obstacle avoidance ability was improved. Liang *et al*. [21] combined these samples with the width of the robot and integrated the combined results into the DWA algorithm to enhance its dynamic obstacle avoidance ability in narrow spaces. However, it's worth considering that this algorithm may exhibit complexity and inefficiency. The performance comparison of the mentioned algorithms is detailed in Table 2.

Sampling based motion planning methods perform sampling to cover map space, and path planning algorithms such as RRT* and DWA require a large amount of sampling to obtain the optimal solution in the current situation. Due to the complex water environment and collision constraints, the sampling based motion planning methods will increase the computational burden during the path sampling process. To reduce computational burden, we improved the RRT * algorithm by establishing a virtual field. The main contribution of this article is not only to find low-cost paths that meet constraints, but also to improve the efficiency of sampling.

To overcome the abovementioned drawbacks, first, the considering ocean current RRT* (COC-RRT*) algorithm

Table 2. Local performance of path planning algorithm.

Algorithm	Improve avoidance ability	COLREGs	Considering relative velocity
DWA [8]	/	/	/
Fuzzy-based DWA [15]	Yes	/	/
A* and DWA [16]	Yes	/	/
Algorithms based on COLREGs [18]	/	Yes	/
Path planning algorithm for MSV [19]	/	Yes	/
DWA in complex environment [20]	/	/	Yes
DWA considering constraint [21]	/	/	Yes

is proposed. This method introduces the virtual field sampling function and ocean current constraint function into the RRT* algorithm so that the algorithm can sample and optimize according to the direction and intensity of the current. At the same time, the global planning path is optimized by adjusting the path between the parent and child nodes. Second, the COLREGs constraint is introduced into the DWA algorithm, and the dist function in the DWA algorithm is improved to optimize its obstacle avoidance capability.

The main contributions of this work can be summed up in the following:

1) In this paper, the COC-RRT* algorithm is proposed, the efficiency of the RRT algorithm in the space-varying ocean environment is improved, shortens the time of USVs sailing, and reduces the length of the planning path.
2) An improved DWA algorithm was proposed in this paper, which can optimize the rationality of the path planning and the dynamic obstacle avoidance ability of the DWA algorithm.

The subsequent sections of this article are structured as follows: Section 2 outlines the modeling process, while Section 3 presents the proposed algorithm. Section 4 discusses the experimental results obtained through implementing the algorithm. Finally, Section 5 presents the conclusion.

2. PRELIMINARIES AND PROBLEM FORMULATION

2.1. Ocean current modeling

Multiple single-point eddies can form the dynamic model of the ocean, and the expression of single-point eddies is as follows:

$$eddy\{p,a\}:\begin{cases} f(x,y)=(x-p_x)^2+(y-p_y)^2, \\ c_x=\left[-\left(|a_x|\dfrac{\partial f}{\partial x}\right)-\left(|a_y|\dfrac{\partial f}{\partial y}\right)\right]\cdot\dfrac{1}{2f}, \\ c_y=\left[sgn(a_y)|a_x|\dfrac{\partial f}{\partial x}-|a_y|\dfrac{\partial f}{\partial y}\right]\cdot\dfrac{1}{2f}, \end{cases} \tag{1}$$

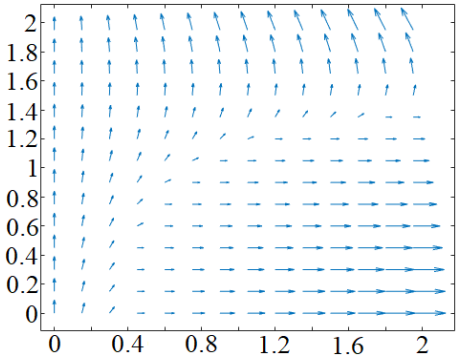


Fig. 1. The model of ocean current.

where c_x is the strength of the seawater flow in the x-axis direction; c_y denotes the strength of the seawater flow in the y-axis direction; a_x denotes the strength coefficient of the seawater flow on the x-axis; a_y denotes the strength coefficient of the seawater flow on the y-axis; p_x denotes the coordinate of the eddy p on the x-axis; the coordinate of eddy p on the y-axis is p_y; and sgn() is a symbolic function [22]. Fig. 1 illustrates the ocean current, with the arrow indicating the direction of the seawater flow.

2.2. Obstacle modeling

The Unity3D platform is applied to model obstacles in maps. To ensure the safety of USV sailing, extended modeling of obstacles was carried out. In Fig. 2, the obstacle coordinate system $\{o\}$ is established with the center of the obstacle serving as the origin.

The outline of obstacles is denoted as $(x_{o.i}, y_{o.i})$, $i = 1, 2, ..., n$. The expression of expanded outline is as follows:

$$\begin{cases} x^*_{o.i}=(1+E)\cdot x_{o.i}, \\ y^*_{o.i}=(1+E)\cdot y_{o.i}, \end{cases} \tag{2}$$

where E is the obstacle expansion coefficient, $x^*_{o.i}$ and $y^*_{o.i}$ are the expanded outline coordinate. In Figs. 2(a) and 2(c) display the obstacle prior to expansion, Figs. 2(b) and 2(d) depict the obstacle after expansion.

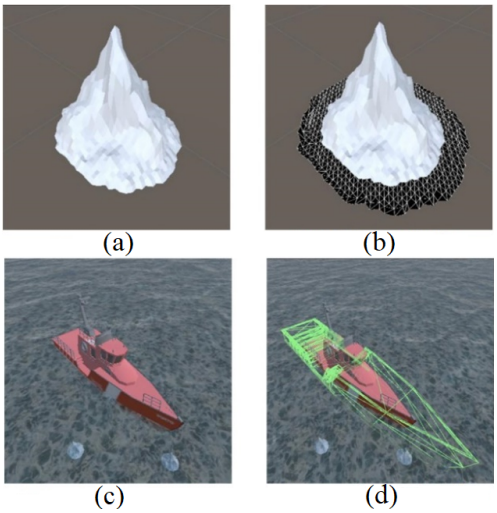


Fig. 2. Obstacle modeling: (a) Static obstacle. (b) Static obstacle after expansion. (c) Dynamic obstacle. (d) Dynamic obstacle after expansion.

2.3. USV modeling

The USV coordinate system [23] is specified, as shown in Fig. 3. A coordinate system is established on the ocean's surface, where the origin O_e can be anywhere on the ocean's surface. The Y_e- and X_e-axes point to the north and east respectively. This system is transformable to an $\{e\}$ coordinate system. This paper exclusively focuses on the surge, sway, and yaw motions of a USV. A three-degree-of-freedom mathematical model for the USV is established as follows:

$$\begin{cases} \dot{\eta} = J(\eta) \cdot \upsilon, \\ M \cdot \dot{\upsilon} + C \cdot \upsilon + D \cdot \upsilon = \tau + \omega. \end{cases} \tag{3}$$

Modeling real USVs through the Unity 3D platform to construct 3D USV models. The actual USV and its 3D model are depicted in Figs. 4(a) and 4(b). To facilitate the observation of 3D USV collisions, the collision and collision detection module of the Unity3D platform has been

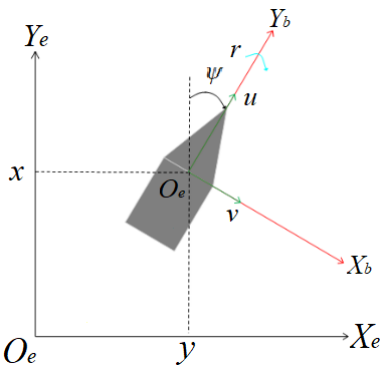


Fig. 3. The USV movement coordinate system.

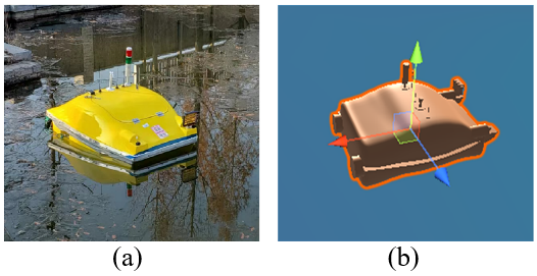


Fig. 4. The model of USV: (a) The real USV. (b) The 3D USV.

incorporated into the 3D USV model. Additionally, a rigid body and kinematic rigid body collider have been added to the 3D USV model to account for its inertia and gravity. Finally, a buoyancy module has been integrated into the 3D USV model. These modeling methods enable the 3D USV model to exhibit realistic physical characteristics in simulation experiments, thereby allowing for more effective validation of the proposed algorithm through simulation experiments.

2.4. Problem statement

Defined distance-based evaluation criteria. The planning path P is constructed by linking multiple path nodes or grids. The Euclidean distance of the planning path P is presented as

$$D(P) = \sum_{i=1}^{k-1} s_i, s_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \tag{4}$$

where s_i is the Euclidean distance between adjacent path nodes or grids, (x_i, y_i), $i = \{1, 2, ..., k-1\}$ is the coordinates of each path node or grid. There are various obstacles in the lake, which form a prohibited area O. The USV must avoid this area. Assuming that the planning path can avoid the prohibited area O, the optimal path is determined based on time $T(P_i)$. In this study, the USV has to travel to the target point in the shortest possible time. Therefore, the objective function is

$$X^* = \arg\min_X T(P_i), \ \forall i = \{1, 2, ..., K\}, \tag{5}$$

where $T(P_i)$ is the time cost function. In the simulation, water flow with variable space and constant time was considered; The direction and intensity of water flow vary between different positions, but remain unchanged in different time series. $T(P_i)$ is

$$T(P_i) = \begin{cases} \dfrac{|D(P_{i.0}) - D(P_{i.1})|}{|V_{a.1}|} + \displaystyle\sum_{m=2}^{M-1} \dfrac{|D(P_{i.m-1}) - D(P_{i.m})|}{|V_{a.m}|} \\ \quad + \dfrac{|D(P_{i.M-1}) - D(P_{i.M})|}{|V_{a.M}|}, \quad P_i \cap O = \emptyset, \\ +\infty, \qquad \text{otherwise}, \end{cases} \tag{6}$$

where P_i is the candidate for the ith path planing, that does not travel through the prohibited area O; $P_{i.m}$ is the local path on P_i; $V_{a.m}$ is sailing speed of USV sailing in the local path $P_{i.m}$; Meanwhile, $V_{a.m}$ represents the combined vector of the USV's forward speed and environmental disturbances.

3. PROPOSED ALGORITHMS

3.1. COC-RRT* algorithm

Ocean currents will affect the sailing of USVs. In this paper, ocean currents were transformed into a virtual field. At each point in the virtual field, the strength of $P(x, y)$ is U_c

$$U_c = F_{oc}\sqrt{c_x^2 + c_y^2}, \tag{7}$$

where F_{oc} is point P's current gain; c_x is the size of the ocean current along the x-axis; c_y is the size of the ocean current along the y-axis.

Fig. 5 displays the virtual obstacle field, where the sphere represents the boundary of the obstacle virtual field. The virtual field extends from the center of the obstacle to the USV.

The value of U_o represents the strength of any point $P(x, y)$ within the obstacle virtual field, and its expression is as follows:

$$U_o = \begin{cases} \frac{1}{2}F_{oo} \cdot \sum_{i=1}^{n} \frac{x_{o.i}^*}{x_{o.i}} \cdot \left[\frac{1}{\rho} - \frac{1}{\rho_0}\right], & \rho \le \rho_0, \\ 0, & \rho > \rho_0, \end{cases} \tag{8}$$

where F_{oo} represents the strength parameter of the virtual field surrounding the obstacle; $x_{o.i}^*$ is the boundary coordinate of the obstacle after expansion; $x_{o.i}$ is the boundary coordinate of the obstacle before expansion; ρ_0 is the range of virtual field influenced; x_o and y_o are the x- and y-axis coordinates of the obstacle; and ρ is the Euclidean distance of (x, y) and (x_o, y_o), and its expression is

$$\rho = \sqrt{(x - x_o)^2 + (y - y_o)^2}. \tag{9}$$

The sampling range of the COC-RRT* algorithm is M; p is an extension step; T is a planning path of random tree;

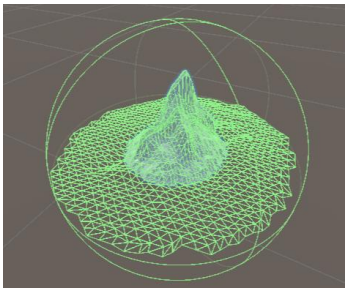


Fig. 5. Virtual field model of obstacles.

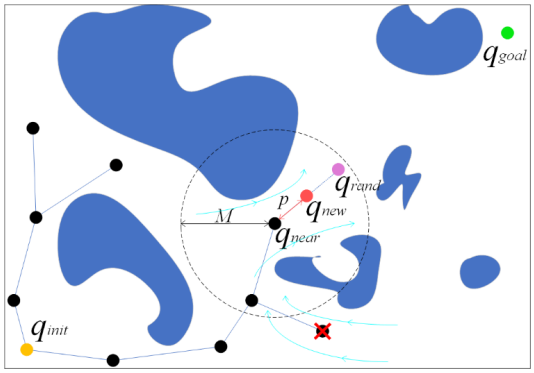


Fig. 6. The expansion process involves random sampling points in the COC-RRT* algorithm.

Algorithm 1: COC-RRT* (q_{init}, q_{goal}).

Step 1: $T \leftarrow InitializeTree(T, q_{init})$;
Step 2: **For** i = 1: n **do**
Step 3: $q_{rand} \leftarrow SOCMFSample(T, q_{goal}, M)$;
Step 4: $q_{near} \leftarrow Nearest(T, q_{rand})$;
Step 5: $q_{new} \leftarrow Steer(q_{near}, q_{rand}, p)$;
Step 6: $q_{neighbor} \leftarrow Findnearneighbor(T, q_{new}, M)$;
Step 7: **if** $CollisionFree(q_{new}, T, M)$ **then**
Step 8: **if** $OCConstraints(q_{new}, T, M)$ **then**
Step 9: $T \leftarrow Chooseparent(q_{new}, q_{neighbor}, T)$;
Step 10: $T \leftarrow Rewire(T, q_{new}, q_{neighbor})$;
Step 11: **return** T

q_{init} is the initial point; q_{goal} is the target; n is the maximum iterations. The COC-RRT* algorithm is based on RRT* to achieve path planning under spatially changing ocean currents. Fig. 6 shows the expansion process of the sampling points of the COC-RRT* algorithm. The path nodes are created to maximize their utilization, shorten the sailing time of USVs, and avoid excessive energy costs. The pseudocode of the COC-RRT* algorithm is represented in Algorithm 1.

In Algorithm 1, $InitializeTree(T, q_{init})$ is used to initialize the random tree T and take q_{init} as the initial node. $Nearest(T, q_{rand})$ is used to select the path node closest to the point q_{rand} in T as the point q_{near}. The point from the line between point q_{near} and point q_{rand} are took by $Steer(q_{near}, q_{rand}, p)$ as q_{new}, and the distance between point q_{near} and point q_{rand} is p. $Findnearneighbor(T, q_{new}, M)$ is used to find the point q_{new}. Set the sampling range as M, q_{new} as the center. The point $q_{neighbor}$ are select by above parameters, and it is the highest index in T. $CollisionFree(q_{new}, T, M)$ is used to detect whether the planning path collides. $Chooseparent(q_{new}, q_{neighbor}, T)$ connects the point $q_{neighbor}$ in T with the point q_{new}. $Rewire(T, q_{new}, q_{neighbor})$ is used to reselect the shortest planning path, calculate the distance between the point $q_{neighbor}$ and the point q_{new} in T, and return the path with the shortest distance to T [25].

This article improves the RRT* algorithm, and proposes $SOCMFSample(T, q_{goal}, M)$. $SOCMFSample(T, q_{goal}, M)$ is used to generate point q_{rand}. The length between the point q_{rand} and the point with the highest index in T is must less than M. $q_T(x_T, y_T)$'s index is T. U_{sum} is the sum of the virtual field at q_T, and its expression is

$$U_{sum} = \sqrt{U_c^2 + U_o^2 + 2U_c U_o \cos(\delta - \epsilon)}, \tag{10}$$

where U_c is at q_T, which refers the virtual field strength; U_o is at q_T, which refers the virtual field strength of obstacles; δ is current virtual field angle; ϵ is obstacle virtual field angle. δ_{sum} is at q_T, which refers the total virtual field angle, and its expression is

$$\delta_{sum} = \frac{\delta \cdot U_c + \epsilon \cdot U_o}{U_c + U_o}. \tag{11}$$

The coordinates of point q_{rand} are (x_{rand}, y_{rand}). The expression of x_{rand} is as follows:

$$x_{rand} = x_T + \frac{U_{sum}}{U_s} \cdot \sin[\delta + \text{rand}(\frac{|\psi - \delta|}{2})] \cdot \text{rand}(M), \tag{12}$$

where x_T is the x-axis coordinate; U_s is the range parameter, and q_{rand} will be affected by U_s; rand(n) takes a random number from 0 to n; and ψ is the planning path at q_T. The y_{rand} is

$$\begin{aligned} &y_{rand} \\ &= y_T + \frac{U_{sum}}{U_s} \sin[\delta + \text{rand}(\frac{|\psi - \delta|}{2})] \\ &\quad \times \text{rand}(\sqrt{M^2 - (x_{rand} - x_T)^2}), \end{aligned} \tag{13}$$

where y_T is the y-axis coordinate of the path node q_T. The pseudo-code of $SOCMFSample(T, qgoal, M)$ is represented in Algorithm 2.

In Algorithm 2, U_{sum} is the sum of U_c and U_o; x_{rand} is the x-axis coordinate of q_{rand}; and the y-axis coordinate of q_{rand} is y_{rand}.

$OCConstraints(q_{new}, T, M)$ is the ocean current constraint function in RRT*. Sampling at a distance of M can obtain the center q_{new}; $q_s(x_s, y_s)$ is the midpoint between

Algorithm 2: $SOCMFSample(T, qgoal, M)$.

Step 1: if rand(1) > 0.9 **then**
Step 2: **return** q_{goal}
Step 3: else $U_{sum} \leftarrow$ (10);
Step 4: $\delta_{sum} \leftarrow$ (11);
Step 5: $\delta \leftarrow \delta_{sum}$;
Step 6: $x_{rand} \leftarrow$ (12);
Step 7: $y_{rand} \leftarrow$ (13);
Step 8: $q_{rand} \leftarrow (x_{rand}, y_{rand})$;
Step 9: return q_{rand}

Algorithm 3: $OCConstraints(q_{new}, T, M)$.

Step 1: $q_s \leftarrow (q_{new} + q_T)$ / 2;
Step 2: For j = 1: e **do**
Step 3: $x_{dom} \leftarrow$ (14);
Step 4: $y_{dom} \leftarrow$ (15);
Step 5: **if** $U_{cdom} > 2B_s$
Step 6: **return** False
Step 7: return True

q_{new} and q_T; $q_{dom}(x_{dom}, y_{dom})$ is a random sampling point. The expression of x_{dom} is

$$x_{dom} = x_s + \text{rand}(M). \tag{14}$$

The expression of y_{dom} is as follows:

$$y_{dom} = y_s + \text{rand}(\sqrt{M^2 - (x_{dom} - x_s)^2}). \tag{15}$$

The pseudo-code of $OCConstraints(q_{new}, T, M)$ is represented in Algorithm 3.

In Algorithm 3, the maximum iterations is e; U_{cdom} is at q_{dom}, which refers to ocean current virtual field strength.

3.2. Improved DWA algorithm

The DWA [26] algorithm constrains sampling from three aspects: distance, speed, and security, and then uses the objective function to determine the optimal planning path. The expression of the DWA algorithm is

$$G(v, \omega) = K[\alpha H(v, \omega) + \beta D(v, \omega) + \gamma V(v, \omega)]. \tag{16}$$

3.2.1 COLREGs constraints

To guarantee navigational safety of unmanned underwater vehicles, International Regulations for Preventing Collisions at Sea (COLREGs) [27]. have been incorporated into the constraints of obstacle avoidance maneuvers, which can improve the safety and rationality of local path planning. When the USV encounters dynamic obstacles or in the $R-crossing$ area, the DWA algorithm will not plan the path from the back end of the dynamic obstacle to the L intersection area, as shown in Fig. 7. In Fig. 7, only green paths are planned. However, when the USV encounters dynamic obstacles in the $L-crossing$ area, any feasible path can be planned by the DWA algorithm. The local planned path P_i is

$$P_i = \begin{cases} (u_i \pm \dot{v}_{i.c}\Delta t, r_i - \dot{\omega}_{i.c}\Delta t), & A_R \cap O_d \neq \emptyset, \\ (u_i \pm \dot{v}_{i.c}\Delta t, r_i \pm \dot{\omega}_{i.c}\Delta t), & \text{otherwise}, \end{cases} \tag{17}$$

where $\mu < \dot{\omega}_{i.c} \leq \sqrt{2dist(u_i \pm \dot{v}_{i.c}\Delta t, r_i - \dot{\omega}_{i.c}\Delta t)\dot{\omega}_b}$. For USV, P_i is the planning path; u_i is velocity; r_i is the angular speed; $\dot{v}_{i.c}$ is the collision avoiding acceleration; $\dot{\omega}_{i.c}$ is the collision avoiding angular acceleration; μ is the angle between the connecting line and USV's geometric ray, and this connecting line is between the USV and the dynamic

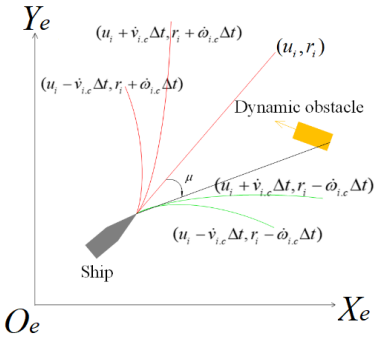


Fig. 7. The path planning results of the DWA algorithm with the COLREGs.

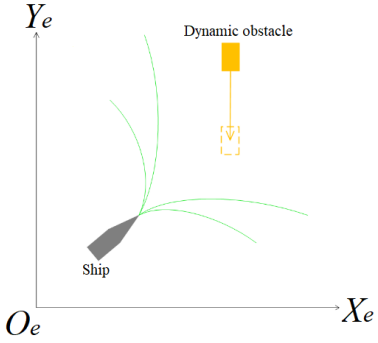


Fig. 8. The path planning results of the DWA algorithm with the COLREGs and $D(v,\omega)$.

obstacle; O_d is the area of dynamic obstacle; $dist(v,\omega)$ is the distance between the planning path of (v,ω) and an obstacle; A_R and A_L are the areas of $head-on$ between $R-crossing$ and the $L-crossing$; $\dot{\omega}_b$ is the obstacle's acceleration of angular velocity.

The DWA algorithm's planning path is red in Fig. 7. The planning path of the DWA algorithm with the COLREGs is green. The dynamic obstacle's trajectory is yellow.

3.2.2 Improved dist function

It's important to consider that the threat degree of a dynamic obstacle to a USV is significantly higher than that of a static obstacle. n is number of obstacles. Therefore, we improved the dist function $D(v,\omega)$ of the DWA algorithm as follows:

$$D^*(v,\omega)=\psi(V_{b.i})\cdot\frac{D(v_i,\omega_i)}{\sum_{i=1}^{n}D(v_i,\omega_i)}, \tag{18}$$

where $\psi(V_{b.i})$ is the obstacle speed factor

$$\psi(V_{b.i})=\frac{\sum_{i=1}^{n}V_{b.i}}{V_{b.i}+\sum_{i=1}^{n}V_{b.i}}, \tag{19}$$

where $l\in[1,n]$; $V_{b.l}$ is the velocity of the obstacle closest to the USV, and higher than $V_{b.l}$. $D^*(v,\omega)$ and the disence of the planning path to dynamic obstacle. Therefore, the DWA algorithm tends to choose a planning path that is further away from dynamic obstacles as the local optimal path as shown in Fig. 8. The green path denotes planned through the DWA algorithm with $D^*(v,\omega)$. $V_{b.i}$ and $\psi(V_{b.i})$ are negatively correlated, so $D^*(v,\omega)$ is high when a dynamic obstacle is far from the USV.

4. SIMULATION RESULTS AND DISCUSSION

We conducted the simulation experiments on a PC operating on Windows 11 using Unity3D. The hardware featured an Intel Core i7-10870H processor operating at a base frequency of 2.21 GHz and 8 GB of memory. To validate the effectiveness of the proposed algorithm, the positions of obstacles on the map were randomly selected, and We constructed a 3D square map of 1000 m × 1000 m that contained a number of areas of environmental disturbance, following the description provided in Section 3. The parameters of the proposed algorithm and simulation experiment in this paper are shown in Table 3.

Table 3. The parameters of the proposed algorithm and simulation experiment in this paper.

Parameters	Definition	Numerical value
u (m/s)	Forward speed of USV	10
r (rad/s)	Angular velocity of USV	0.8
E	Expansion coefficient of obstacle	1.3
α	Heading weight coefficient	0.08
β	Dist weight coefficient	0.3
γ	Velocity weight coefficient	0.2
F_{oc}	Ocean current gain	1.2
F_{oo}	Obstacle gain	1.1
U_s	Range coefficient	15
ρ_0 (m)	Influence range of obstacle	15

The data presented in Table 3 were obtained from [28]. The experiment in this paper was conducted in three steps. Experiment 1 primarily aimed to validate the effectiveness of the proposed algorithm in an environment with space-varying ocean currents. Experiment 2 mainly verified the effectiveness of the proposed algorithm in the dynamic obstacle environment. Experiment 3 mainly verified the effectiveness of the proposed algorithm in the dynamic environment.

4.1. Validation of the algorithm in the space-varying ocean current environment

In Scenario 1, the coordinates of the starting point and target point are set to (-430, -346) and (355, 400), respectively. Static obstacles are added to the map and their po-

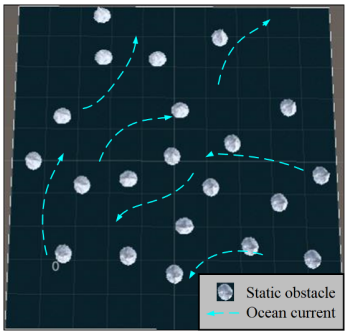


Fig. 9. The experiment map of Scene 1.

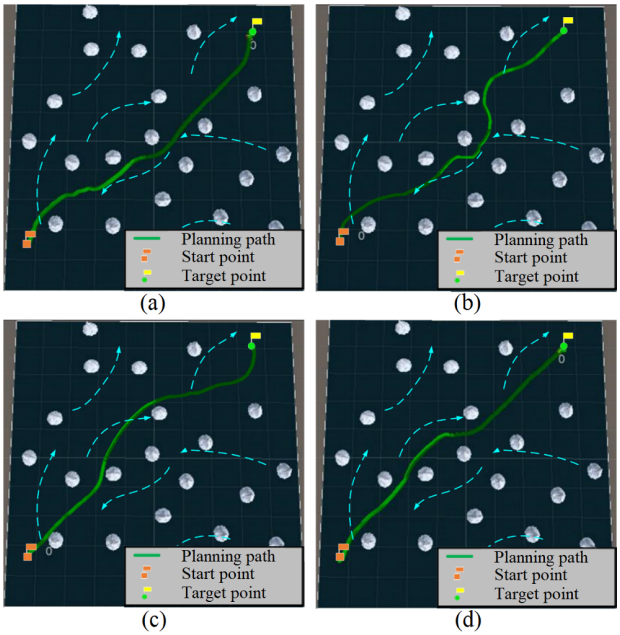


Fig. 10. The experiment results in Scene 1: (a) RRT*. (b) A*. (c) A* with B-spline. (d) COC-RRT*.

sitions are random. At the same time, there are also ocean currents on the map, as shown in Fig. 9, where the blue arrow indicates the direction of the current.

Select the RRT* algorithm, A* algorithm, and a hybrid algorithm of A* and B-spline curves as the comparison algorithms. The effectiveness of this algorithm in global path planning under spatially changing ocean currents was verified through comparative experiments. The experimental results are shown in Fig. 10 and Table 4.

From Figs. 10(a) and 10(b), it can be seen that the RRT* algorithm and A* algorithm did not plan the path in accordance with the direction of the ocean current. resulting in an extension of navigation time. From Fig. 10(c), it can be seen that although the first half of the planning path of A* with B-spline is planned in accordance with the ocean current direction, the second half of the planning path is not planned according to the ocean current direction. From Fig. 10 (d), it can be seen that the COC-RRT* algorithm

Table 4. The results of the four algorithms in Scene 1.

Algorithm	Path length (m)	Sailing time (s)
RRT* algorithm	1288.1553	118.5496
A* algorithm	1375.7496	101.9204
A* with B-spline curve	1310.2377	97.0671
COC-RRT* algorithm	1291.4585	**90.2185**

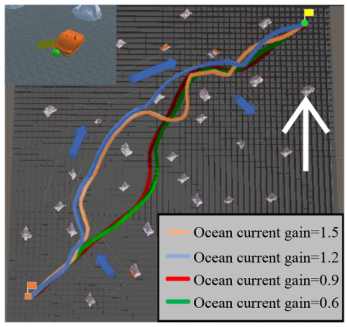


Fig. 11. The experiment results of four different ocean current gain.

plans the path based on the ocean current direction. From the results of the experiments in Table 4, it can be seen that although the planning path length of the COC-RRT* algorithm is longer than that of the RRT* method, the USV is affected by ocean currents during navigation. Therefore, the navigation time of USVs using the COC-RRT* algorithm is shorter than that of USVs using the RRT* method. Therefore, the performance of the COC-RRT* algorithm is superior to that of the comparison algorithm.

To avoid the randomness of the experimental results, three different sets of starting and target points for simulation experiments were selected in Scenario 1, and the results obtained are shown in Table 5.

To test the impact of current gain on the planned path, we selected four different current gains to plan the planned path based on the same starting and target points. The experimental results are shown in Fig. 11.

As can be seen from Fig. 11, with the reduction of the ocean current gain, the planning path is not planned according to the ocean current direction. When the ocean current gain increases, the planning path is planned along the ocean current direction, but when the ocean current gain is greater than 1.5, the planning path is excessively affected by the ocean current. Therefore, when the ocean current gain is 1.2, the planning path is most reasonable.

According to the experimental analysis, the planning path becomes farther away from obstacles as the obstacle gain increases. When the obstacle gain is 1.1, the algorithm has the best performance. The relationship between obstacle gain and algorithm performance is shown in Fig. 12.

Table 5. The simulation results of the four algorithms for different combinations of the starting and target points in Scene 1.

Start and target	Algorithm	Path length (m)	Sailing time (s)
(-94, -27), (347, -451)	RRT* algorithm	692.4512	78.8482
	A* algorithm	750.8269	88.4003
	A* with B-spline curve	715.0733	84.1908
	Proposed algorithm	701.1358	**61.2596**
(181, -419), (-84, 103)	RRT* algorithm	672.1135	72.2151
	A* algorithm	724.7819	71.4705
	A* with B-spline curve	690.2685	68.0671
	Proposed algorithm	683.6091	58.5226
(400, -400), (-400, 400)	RRT* algorithm	1298.2208	120.2126
	A* algorithm	1395.8702	117.9479
	A* with B-spline curve	1329.4002	112.3314
	Proposed algorithm	1302.1548	91.2158

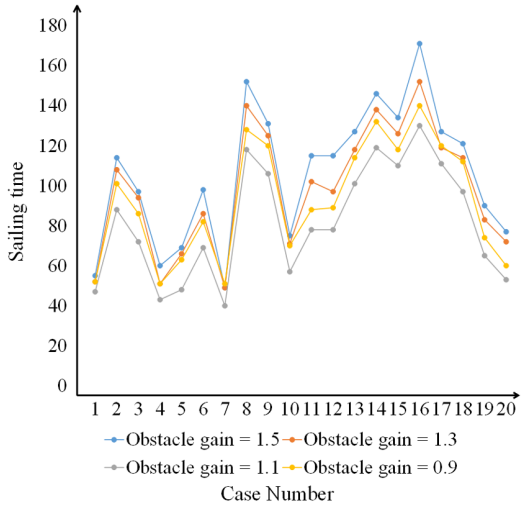


Fig. 12. The experiment results of four different obstacle gain.

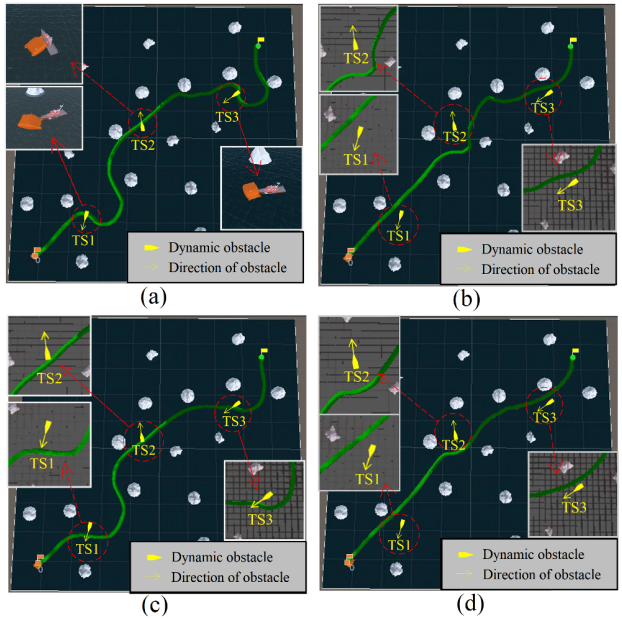


Fig. 13. The experimental results of the four algorithms in Scene 2: (a) DWA. (b) DWA with COLREGs. (c) DWA with $D^*(v,\omega)$. (d) The proposed algorithm.

4.2. Validation of the algorithm in the dynamic obstacle environment

The start point is (-410, -410), and target point is (370, 370). The DWA algorithm, the DWA algorithm with COLREGs constraints, and the DWA algorithm with the improved distance function $D^*(v,\omega)$ was used as comparison algorithm. Validate the dynamic obstacle avoidance ability of proposed algorithm in the experiment of Scenario 2. The yellow ship represented a threat ship. The experimental results are depicted in Fig. 13 and Table 6.

The USV collided with TS1, as shown in Table 6 and Fig. 13(a). In addition, due to the fact that the DWA algorithm does not include COLREGs constraints, USV collides with TS2. As shown in Fig. 13(b), when USV encounters TS1 and TS3, it complies with COLREG and does not perform unnecessary obstacle avoidance actions. When USV encounters TS2, it will perform the correct obstacle avoidance action, but the planned path is too long.

Table 6. The results of the four algorithms in Scene 2.

Algorithm	Planning time (s)	Path length (m)	Number of collisions
DWA	3.8498	1391.1518	3
DWA with COLREGs	3.1821	1273.2949	0
DWA with $D^*(v,\omega)$	3.2157	1324.8153	0
Proposed algorithm	**2.6543**	**1188.4486**	**0**

In addition, as shown in Fig. 13(c), when the USV encounters TS1 and TS3, even though the DWA algorithm con-

siders the relative speeds of the threat ship and the USV and can successfully avoid obstacles, the algorithm does not include the COLREGs constraint, resulting in an extension of the USV's navigation distance and an increase in navigation time costs. When USV encounters TS2, it is scratched by TS2. From the figure, it can be seen that in Fig. 13(d), due to the addition of COLREGs constraints in the algorithm, the proposed algorithm follows the left crossing rule and does not perform any unnecessary obstacle avoidance actions. In addition, when USV encounters TS1, this reduces the time spent on navigation. When USV encounters TS2, the algorithm follows the right side traversal rule and performs correct obstacle avoidance actions, improving safety. Therefore, the proposed algorithm is superior to comparative algorithms in avoiding dynamic obstacles.

4.3. Validation of the algorithm in the dynamic environment

In the experiment of Scenario 3, ocean currents and threat ships were added in order to verify the effectiveness of the proposed algorithm. The speed of the threat ship is 5 meters per second, as shown in Fig. 14. In order to simulate the ocean current in real sea areas, in this experiment, we generated the position of the ocean current based on the coordinates provided in the simulation; This position randomly moves within a circle with a radius of 10 meters, and the current intensity varies randomly between 0 meters per second and 7 meters per second. This algorithm adjusts the planned path based on real-time data of ocean currents.

In Scenario 3, the coordinates of the starting point and target point are set to (-410, -410) and (410, 410), respectively. Choose a hybrid algorithm that combines RRT* and DWA algorithms as the comparison algorithm. In the comparison algorithm, the parameters of RRT* algorithm and DWA algorithm are consistent with those of the proposed algorithm. The effectiveness of this algorithm in path planning in dynamic environments was verified through comparative experiments. The experimental results are shown in Fig. 15 and Table 7.

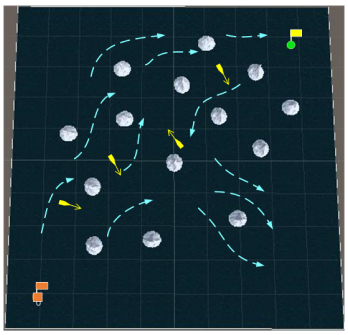
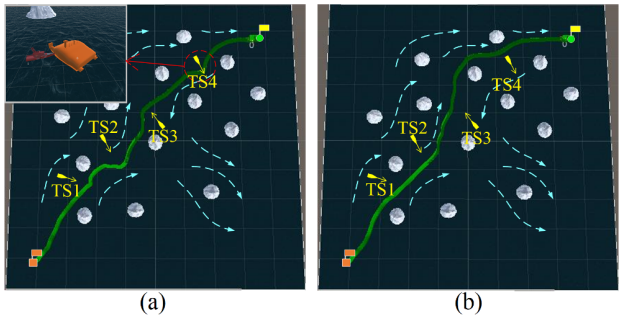


Fig. 14. The experiment map of Scene 3.



(a) (b)

Fig. 15. The experimental results of the two algorithms in Scene 3: (a) RRT* with DWA. (b) The proposed algorithm.

Table 7 shows that the proposed algorithm has lower path length, navigation time, and collision time compared to the RRT* of the DWA algorithm. Due to its more complex structure and longer computational time, the proposed algorithm is still within an acceptable range, making it effective. According to the experimental results in Table 7, it can be seen from the graph that when USV encounters TS1 and TS2 in Fig. 15(a), it does not comply with the left crossing rule, which prolongs the planning path length and increases the navigation time cost. When USV encounters TS3, it does not comply with the correct traversal rules, which prolongs the length of the planned path and increases the cost of navigation time. When USV encounters TS4, it does not comply with the left crossing rule and collides with TS4. From the figure, it can be seen that in Fig. 15(b), the proposed algorithm adds a COLREG constraint. When encountering TS1 and TS2, the USV follows the left crossing rule and does not engage in large-scale obstacle avoidance movements, reducing the planned path length. At the same time, the algorithm plans the path based on the direction of the ocean current, reducing the navigation time of the USV. Therefore, the performance of this algorithm in dynamic environments is superior to that of comparative algorithms, verifying its effectiveness.

To avoid the randomness of the experimental results, we selected three different sets of starting and target points for simulation experiments in Scenario 3, and the results obtained are shown in Table 8.

Table 7. The results of the two algorithms in Scene 3.

Algorithm	Path length (m)	Sailing time (s)	Number of collisions	Computational time (s)
RRT* with DWA	1319.1895	121.0453	1	0.75112
Proposed algorithm	**1301.4933**	**107.6377**	**0**	0.75896

Table 8. The simulation results of the two algorithms for different combinations of the starting and target points in Scene 3.

Start and target	Algorithm	Path length (m)	Sailing time (s)	Number of collisions	Computational time (s)
(-400, 400), (400, -400)	RRT* with DWA	1277.9765	67.1244	1	0.61845
	Proposed algorithm	**1262.1578**	**57.0785**	**0**	0.62048
(450, 350), (-400, -200)	RRT* with DWA	1075.3998	72.0058	2	0.66402
	Proposed Algorithm	**1062.3233**	**65.3064**	**0**	0.66506
(-50, 400), (50, -400)	RRT* with DWA	847.2039	52.3108	1	0.74848
	Proposed algorithm	**830.8113**	**42.9518**	**0**	0.75201

5. CONCLUSION

This paper proposed a path-planning method for USVs in the dynamic environment. In the experiment of Scene 1, the planning path of the proposed algorithm could effectively address the impact of ocean currents. The experimental results indicate that the proposed algorithm resulted in a shorter sailing time compared to the algorithm used for comparison, which verifies the effectiveness of the proposed algorithm in the ocean current environment. In the experiment of Scene 2, the proposed algorithm could effectively avoid dynamic obstacles. The experimental results indicate that the planning time, path length, and collision times of the proposed algorithm were better than those of the comparison algorithm, which verifies the effectiveness of the dynamic obstacle avoidance ability of the proposed algorithm. In the experiment of Scene 3, the proposed algorithm could effectively avoid dynamic obstacles in the ocean current environment and successfully reach the target point. Although the computational time of the proposed algorithm was longer, it was within the acceptable range. The experimental results show that the path length, sailing time, and collision times of the proposed algorithm were superior compared to the comparison algorithm, thereby confirming the effectiveness of the proposed algorithm in the dynamic environment.

In future research, we can improve this study in two aspects. First, the majority of the parameters in this paper were derived from accumulated experience gained from previous research, consequently, the deep learning algorithm can be employed to optimize these parameters. Second, the proposed algorithm does not consider the problem of multi-targets, so optimization algorithms can be introduced to specify the traversal order of multi-targets for improvement.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] A. Felski and K. Zwolak, "The ocean-going autonomous ship-challenges and threats," *Journal of Marine Science and Engineering*, vol. 4, no. 41, pp. 1-16, 2020.

[2] Z. Zhu, J. Xiao, and J. Q. Li, "Global path planning of wheeled robots using multi-objective memetic algorithms," *Integrated Computer Aided Engineering*, vol. 22, no. 4, pp. 387-404, 2015.

[3] C. Altunbas, T. Alexeev, and M. Miften, "Effect of grid geometry on the transmission properties of 2D grids for flat detectors in CBCT," *Physics in Medicine and Biology*, vol. 64, no. 22, pp. 225006-225006, 2019.

[4] Q. S. Zhang, X. Song, and Y. Yang, "Visual graph mining for graph matching," *Computer Vision and Image Understanding*, vol. 178, no. 1, pp. 16-29, 2019.

[5] Y. Wu and K. Low, "An adaptive path replanning method for coordinated operations of drone in dynamic urban environments," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4600-4611, 2021.

[6] J. Ni, L. Wu, and P. Shi, "A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 9269742, pp. 1-16, 2017.

[7] P. Wang, S. Gao, and L. Li, "Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm," *Energies*, vol. 12, no. 12, pp. 1-14, 2019.

[8] H. Kim, S. Yun, Y. Choi, J. Ryu, B. Won, and J. Suh, "Improved dynamic window approach with ellipse equations for autonomous navigation of unmanned surface vehicle," *Journal of Institute of Control*, vol. 26, no. 8, pp. 624-629, 2021.

[9] P. Wanna and S. Wongthanavasu, "An improved cellular automata-based classifier with soft decision," *Journal of Internet Technology*, vol. 21, no. 6, pp. 1701-1715, 2020.

[10] J. Park and T. Chung, "Boundary-RRT* algorithm for drone collision avoidance and interleaved path replanning," *Journal of Information Processing Systems*, vol. 16, no. 6, pp. 1324-1342, 2020.

[11] P. Verbari, L. Bascetta, and M. Prandini, "Multi-agent trajectory planning: A decentralized iterative algorithm based on single-agent dynamic RRT star," *Proc. of the 2019 American Control Conference*, pp. 1977-1982, 2019.

[12] B. Sun, W. Zhang, S. Li, and X. Zhu, "Energy optimised D* AUV path planning with obstacle avoidance and ocean current environment," *Journal of Navigation*, vol. 75, no. 3, pp. 685-703, 2022.

[13] Z. Peng, D. Wang, H. Wang, and W. Wang, "Coordinated formation pattern control of multiple marine surface vehicles with model uncertainty and time-varying ocean currents," *Neural Computing and Applications*, vol. 25, no. 7, pp. 1771-1783, 2014.

[14] W. Lan, X. Jin, T. Wang, and H. Zhou, "Improved RRT algorithms to solve path planning of multi-glider in time-varying ocean currents," *IEEE Access*, vol. 9, no. 1, pp. 158098-158115 2021.

[15] Y. Yang, "Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 1, pp. 121-132, 2022.

[16] X. Bai, H. Jiang, J. Cui, K. Lu, P. Chen, and M. Zhang, "UAV path planning based on improved A and DWA algorithms," *International Journal of Aerospace Engineering*, vol. 2021, ArticleID 4511252, pp. 1-12, 2022.

[17] W. Zhang, L. Shan, and L. Chang, "Distributed collision avoidance algorithm for multiple unmanned surface vessels based on improved DWA," *Control and Decision*, 2022.

[18] Loe OAG, "Collision avoidance for unmanned surface vehicles," *Proc. of the 2008 Institutt for teknisk kybernetikk*, 2008.

[19] C. K. Tam and R. Bucknall, "Cooperative path planning algorithm for marine surface vessels," *Ocean Engineering*, vol. 57, pp. 25-33, 2013.

[20] Y. Chen, X. Kang, and P. Xu, "On algorithms for control of multiple mobile robots in complex environment," *Electronics Optics and Control*, vol. 28, no. 4, pp. 48–52, 2021.

[21] S. Liang, J. Liu, and X. D. Xian, "A dynamic window approach to collision avoidance considering robot size constraint," *Control Engineering of China*, vol. 18, no. 6, pp. 872-876, 2011.

[22] X. Yao, F. Wang, J. Wang, and X. Wang, "Energy-optimal path planning for AUV with time-variable ocean currents," *Control and Decision*, vol. 35, no. 10, pp. 2424-2432, 2020.

[23] D. B. Bisandu, I. Moulitsas, and S. Filippone, "Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction," *Journal of Navigation*, pp. 1-26, 2022.

[24] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT*: An improved path planning algorithm for mobile robots," *Expert Systems with Applications*, vol. 152, pp. 113425-113425, 2020.

[25] P. Rousseas, C. Bechlioulis, and K. Kyriakopoulos, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2005-2011, 2021.

[26] C. Jabbour, H. Fakhoury, V. Nguyen, and P. Loumeau, "Delay-reduction technique for DWA algorithms," *IEEE Transactions on Circuits and Systems II Express Briefs*, vol. 61, no. 10, pp. 733-737, 2014.

[27] X. Xu, Y. Lu, X. Liu, and W. Zhang, "Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs," *Ocean Engineering*, vol. 217, no. 1, pp. 107704-107704, 2020.

[28] Y. Zhang, Z. Ju, H. Zhang, and Z. Qi, "3-D path planning using improved RRT* algorithm for robot-assisted flexible needle insertion in multilayer tissues," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 45, pp. 50-62, 2022.

[29] X. Yao, F. Wang, J. Wang, and X. Wang, "Energy-optimal path planning for AUV with time-variable ocean currents," *Control and Decision*, vol. 35, no. 10, pp. 2424-2432, 2020.

[30] K. H. Olszynski, R. Polowy, A. Wardak, A. W. Grymanowska, J. Zielinski, and R. Filipkowski, "Spontaneously hypertensive rats manifest deficits in emotional response to 22-kHz and 50-kHz ultrasonic playback," *Progress In Neuro-Psychopharmacology and Biological Psychiatry*, vol. 120, 110615, 2022.

[31] P. Yao, Z. Zhao, and Q. Zhu, "Path planning for autonomous underwater vehicles with simultaneous arrival in ocean environment," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3185-3193, 2020.

[32] H. Ma, X. Wei, P. Wang, Y. Zhang, X. Cao, and W. Zhou, "Multi-arm global cooperative coal gangue sorting method based on improved Hungary algorithm," *Sensors*, vol. 22, no. 20, pp. 7987-8008, 2022.

[33] M. Soulignac, "Feasible and optimal path planning in strong current fields," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 89–98, 2011.

[34] W. Lan, X. Jin, X. Chang, T. Wang, H. Zhou, W. Tian, and L. Zhou, "Path planning for underwater gliders in time-varying ocean current using deep reinforcement learning," *Ocean Engineering*, vol. 262, 112226, 2022.

[35] Y. Sun, R. Gu, X. Chen, R. Sun, L. Xin, and L. Bai, "Efficient time-optimal path planning of AUV under the ocean currents based on graph and clustering strategy," *Ocean Engineering*, vol. 259, 111907, 2022.

[36] A. Ammar, H. Bennaceur, I. Chari, A. Koubaa, and M. Alajlan, "Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments," *Soft Computing*, vol. 20, no. 10, pp. 4149-4171, 2016.

[37] D. B. Bisandu, I. Moulitsas, and S. Filippone, "Social ski driver conditional autoregressive-based deep learning classifier for flight delay prediction," *Neural Computing and Applications*, vol. 34, pp. 8777-8802, 2022.

[38] L. Liu, G. Han, Z. Xu, J. Jiang, and M. Martinez-Garcia, "Boundary tracking of continuous objects based on binary tree structured SVM for industrial wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 849-861, 2020.

[39] J. Yu, Z. Chen, Z. Zhao, X. Wang, Y. Bai, J. Wu, and J. Xu, "Smooth path planning method for unmanned surface vessels considering environmental disturbance," *International Journal of Control, Automation, and Systems*, vol. 21, pp. 3285-3298, 2023.

[40] J. Yu, Z. Chen, Z. Zhao, P. Yao, and J. Xu, "A traversal multi-target path planning method for multi-unmanned surface vessels in space-varying ocean current," *Ocean Engineering*, vol. 278, no. 15, 114423, 2023.

Jiabin Yu received his B.S. degree from the Beijing Technology and Business University, Beijing, China, in 2007, an M.S. degree in automation from the Beijing Institute of Technology, in 2009, and a Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, in 2012. He has been an Associate Professor with the Beijing Technology and Business University, since 2017. His current research interests cover water environment evaluation and prediction, motor control, and complex system design.

Zhihao Chen received his B.S. degree in electrical engineering and automation from the Beijing Technology and Business University, Beijing, China, in 2021, where he is currently pursuing a master's degree. His current research interests include path planning of mobile equipment and research on path planning algorithm.

Zhiyao Zhao received his B.S. degree in automation from the Beijing Technology and Business University, Beijing, China, in 2011, and a Ph.D. degree in guidance, navigation, and control from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, in 2017. He has been a Lecturer with the Beijing Technology and Business University, since 2017. His current research interests include water environment evaluation and prediction, system health management, and stochastic hybrid systems.

Jiping Xu received his B.S. and M.S. degrees in automation from the Beijing Technology and Business University, Beijing, China, in 2002 and 2005, respectively, and a Ph.D. degree in control theory and control engineering from the School of Automation, Beijing Institute of Technology, Beijing, in 2010. He has been an Associate Professor with the Beijing Technology and Business University, since 2010. His current research interests include water environment evaluation and prediction, and big data analysis.

Yang Lu received his B.S. degree in electrical engineering and automation from the Beijing Technology and Business University, Beijing. China, in 2022, where he is currently pursuing a master's degree. His current research interests include paths planning of mobile equipment and task assigning.