Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

Dynamic Path-Planning Approach of Garbage Cleanup Oriented Unmanned Ship Based on Simplified Flow Velocity Prediction

Ziyi Huang[1], **Nana Chen**[1*], **Yangtian Yan**[1], **Kai Yan**[1], **Xin Chen**[1], **Xianhua Zheng**[1], **Hankun Yin**[2], **Kaixin Xie**[3], **Lin Zhang**[1]

[1] School of Robot Engineering, Yangtze Normal University, Chongqing 408100, China
[2] School of Intelligent Manufacturing Engineering, Chongqing University of Arts and Sciences, Chongqing 402160, China
[3] School of Artificial Intelligence, Anhui University of Science and Technology, Huainan 232001, Anhui, China
E-mail: nanachen@yznu.edu.cn

Received: 10 January 2024; **Revised:** 28 February 2024; **Accepted:** 6 March 2023

Abstract: Research on artificial intelligence and robotics for the ecological protection and restoration of waters has increased in importance with the promotion of national green sustainable development strategies. Autonomous Surface Vessel for Water Cleanup (ASV) research has not yet considered the uncertainty of water surface float positioning under complex current fluctuation factors. The result is that ASV has low accuracy in planning water surface float cleaning paths and its efficiency cannot meet the practical application requirements. To solve the above problems, this paper proposes a dynamic path-planning method architecture based on UAV flow velocity prediction for narrow and shallow water environments such as urban rivers and lakes. We have developed a fusion algorithm based on the progressive optimal rapid random search tree (RRT*) algorithm, DWA dynamic window method, and artificial potential field, and we have realized multi-objective water surface floating under uncertain water flow fluctuations. The location prediction of objects and optimum cleaning paths are carried out, and the method verification is carried out through numerical simulations and prototype experiments. Based on simulation and experimental results, the unmanned surface trash cleaning boat is capable of dynamically predicting trash floaters on the water surface under different water flow environments and generating an optimal cleaning path in real-time. It has certain advantages over existing planning methods in terms of accuracy and timeliness and holds significant practical significance.

***Keywords*:** autonomous surface vehicle, path-planning, obstacle avoidance, garbage cleanup

MSC: 01A01, 22B22, 31K13

1. Introduction

With the continuous advancement of the green sustainable development strategy, and the rapid development of artificial intelligence and robotics today, water surface cleaning vessels have become one of the key equipment for water ecosystem protection and restoration. However, due to the low level of traditional water surface cleaning technology, poor equipment reliability, limited intelligence, and the complexity of water environments, traditional large and heavy water surface cleaning equipment struggle to perform efficiently in specific, narrow, shallow water environments such as

Copyright ©2024 Nana Chen, et al.
DOI: https://doi.org/10.37256/cm.5220244247
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

urban rivers and lakes. This leads to high operational costs, low efficiency, and high labor intensity for workers. With the continuous advancement of Autonomous Surface Vessel (ASV) technology for water surface cleaning, it is bound to become the development trend for water surface cleaning in urban rivers and lakes. However, the current water surface cleaning operation path-planning for ASVs has not considered the complex water flow dynamics in water bodies like urban rivers and lakes, leading to inefficient traditional water surface cleaning operation path-planning. To address this, this paper focuses on the operational spatial characteristics of specific, narrow, shallow water environments like urban rivers and lakes. It considers different water flow velocity scenarios and combines drone-based flow velocity prediction to research a dynamic path-planning method for ASVs based on flow velocity prediction. This aims to enhance the efficiency of water surface cleaning operations for ASVs.

1.1 *Research status of path planning algorithm*

Currently, research on water surface cleaning and recovery using ASVs is relatively widespread in small to medium-sized complex water environments [1–3]. However, there are still limitations, especially concerning narrow, shallow water environments like urban rivers and lakes. There is a need for more in-depth research in the area of path-planning for ASVs. In response to this, international scholars have conducted extensive research and discussion, including algorithms such as the Dijkstra algorithm [4], A* algorithm [5], RRT algorithm [6], and path-planning algorithms based on swarm intelligence optimization [7–9]. Zhang et al. [10] proposed an improved optimization RRT path-planning algorithm based on target attraction and adaptive mixed dynamic step size, which enhances maneuverability and traversal speed in narrow areas. Enevoldsen et al. [11] introduced a closed-water avoidance RRT path-planning algorithm based on ground perception. It utilizes water depth information retrieved through electronic navigation charts (ENC) to calculate optimal path deviations, thereby mitigating the risks of collisions and grounding scenarios. Lin et al. [12] employed a grid-based RRT* algorithm and optimized the node selection method within the penalty function, reducing the computational complexity of optimal path-planning. The APF-Bi-RRT algorithm proposed by Zhang et al. [13] not only enhances search efficiency but also significantly reduces path length and the number of nodes. It improves path smoothness which makes it more suitable for ASV tracking and control.

However, the path-planning scenarios in the above research rarely focus on the impact of external environmental factors on the path-planning algorithms themselves. Ma et al. [14] studied the multi-objective path-planning problem for ASVs in the presence of ocean current effects. In this context, they considered the influence of water flow fluctuations and obstacles, obtaining optimal paths for ASVs under multiple constraints such as collision avoidance, motion boundaries, and speed, in a multi-objective path-planning scenario. Singh et al. [15] proposed a constrained A* algorithm designed for complex ocean environments with dynamic ocean current characteristics. Their research focused on optimal path-planning methods in complex scenarios with both dynamic and static obstacles. Zhen et al. [16] investigated path-planning in the presence of dynamic obstacles and varying sea conditions. They introduced a hybrid path-planning algorithm that combines global paths and intersections based on local map edges. This allows ASVs to find a globally optimal path in a marine environment with both dynamic and static obstacles, as well as complex ocean currents. Zhang et al. [17] introduced an improved VF-RRT* algorithm based on the stream function to find a safe and energy-efficient path in spatially varying ocean currents, offering ASVs a smoother path in ocean currents.

1.2 *Research status of water surface cleaning path planning*

Water surface cleaning solutions can be challenging to apply in complex water environments., many researchers have proposed corresponding path-planning solutions for water surface cleaning. Huo [18] proposed an optimized algorithm based on the internal spiral method for achieving complete coverage in small water environments. This approach guarantees that ASVs can conduct a thorough search without repeating areas or leaving any regions uncleaned. Huang [19] introduced an open water detour traversal algorithm, which accomplishes full path traversal of the target water area and path-planning for water surface cleaning. Xu [20] introduced a Convolutional Coverage Neural Network (CCNN) algorithm for ASV path-planning, which significantly reduces cumulative turning angles, deadlock situations, and computation time. It also maintains directional coverage and exhibits adaptive characteristics in complex environments. Guo et al. [21] proposed a full coverage path-planning algorithm based on an improved A* algorithm. This algorithm utilizes a unit division method and a long-edge reciprocal traversal method. The full coverage path-planning solution based on this algorithm can autonomously plan collision-free paths, enhancing the efficiency of path-planning for ASVs during coverage tasks. Deng et al. [22] discussed the use of unmanned devices for water detection and cleaning, and they proposed an automatic water-cleaning strategy based on the collaboration of Unmanned Aerial Vehicles (UAVs) and Unmanned Surface Vehicles (USVs). This algorithm has significant advantages in terms of computational efficiency and cleaning effectiveness.

1.3 *Existing problems*

Based on the above analysis, there are still some shortcomings in path-planning for narrow and shallow water environments like urban rivers and lakes.

(1) Traditional path-planning methods often neglect the effects of complex environments and flow velocity fluctuations in urban rivers and lakes. What’ s more, path-planning methods traditionally assume static target points.

(2) in the case of water surface cleaning, the target points for floating objects are uncertain due to water flow fluctuations. This makes path-planning for Autonomous Surface Vehicles (ASVs) during water surface debris cleaning operations more complex.

Bearing the above considerations in mind, this paper introduces a dynamic path-planning method for ASV based on drone flow velocity prediction. It considers the water flow fluctuations in specific narrow and shallow water environments such as urban rivers and lakes and designs an improved RRT* fusion path-planning algorithm suitable for water surface debris cleaning. By improving RRT* to guide DWA, this approach prevents DWA from getting stuck in local optima while simultaneously addressing dynamic obstacle avoidance objectives. By integrating the concept of Artificial Potential Fields (APF), the ASV maintains a safe distance from stationary obstacles, thus preventing collisions. Moreover, since floating objects on the water’s surface are prone to changing positions due to variations in water flow, the ASV calculates the fluctuating positions of these objects as they move with the water current. It uses an improved RRT* fusion path-planning algorithm to predict floating object trajectories and find the optimal cleaning path, enhancing cleaning efficiency.

The paper is organized as follows. In Section 2, we describe the architecture of the Autonomous Surface Vehicle (ASV) path-planning system based on flow velocity prediction. Section 3 presents an improved RRT* fusion path-planning algorithm. Section 4 provides simulation results of water surface cleaning path-planning for ASVs under various flow velocity fluctuations. In Section 5, we showcase the application of the ASV prototype in real-world water surface cleaning. Finally, we summarize our findings and outline future research directions in Section 6.

2. System architecture based on flow velocity prediction

To effectively clean floating objects on water surfaces in dynamic water conditions, it’s crucial to detect the water environment conditions based on the cleaning task’s requirements for the Autonomous Surface Vehicles (ASV) working environment. However, ASVs have limited detection ranges, which results in low search efficiency within a given time frame. To overcome this problem, Unmanned Aerial Vehicles (UAVs) can be used for rapid detection of the water environment. The UAVs can help create water maps, locate floating objects, identify obstacles, and predict water flow velocity and direction.

The UAVs utilize cameras and radar vision to gather global waterway information, while also recording water surface flow velocity data with the help of radar speed measurements. The flow velocity sensor is installed beneath the UAV, with a vertical distance of greater than 0.5 m from the sensor to the water surface. It is oriented at an angle of approximately 45-60 degrees concerning the water surface. The sensor emits radar signals towards the water surface, and the reflected signals are received by the sensor. Through analysis and calculation, the signals are then converted into surface-averaged flow velocity and flow direction data. Please refer to Figure 1 for a visual representation.

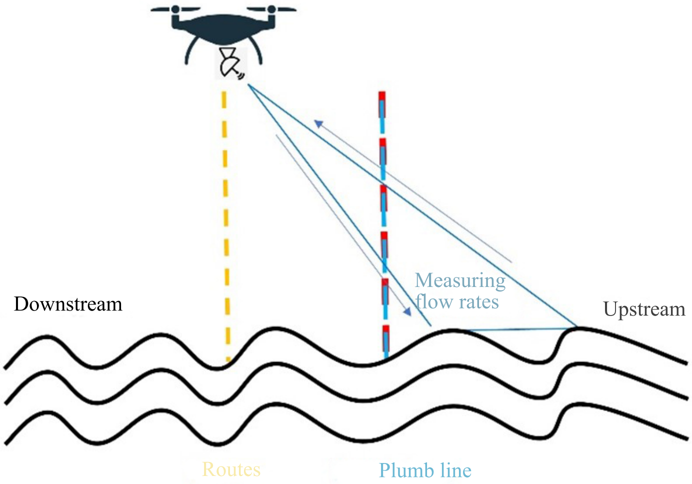


Figure 1. The drone measures the flow rate of the water body

Predicting the trajectory of objects floating on water surfaces is primarily achieved by forecasting their movement based on the direction and velocity of water flow. In this paper, we collect data on the velocity and direction of water surface flow using UAVs and use this information to calculate the original cleaning route for water surface floating objects. We then constructed a predictive trajectory path based on this data. To determine the flow direction trajectory of water surface floating objects, we followed the methodology outlined below:

(1) Given the flow direction and velocity, with the moving velocity being v_m, The coordinates of water surface floating objects are represented as (x_{li}, y_{lj}), Then, after a time interval of Δt, the position is updated as:

$$S(x_{lj}, y_{lj}) = [(x_{lj} + V_m cos\alpha t, y_{lj} + V_m sin\alpha)t] \tag{1}$$

(2) The UAV transmits all the collected data (waterway map information, floating object coordinates, water flow velocity, and direction, etc.) in real-time to the ASV through a communication system the ASV, using the velocity information of the water surface floating objects at various points along the route, can deduce the predicted coordinates range for each floating object to be cleaned. This allows for the connection of the necessary cleaning points to form a cleaning route.

(3) Starting from the ASV's coordinates as the initial point and designating the completion of the cleaning point as the target point, an improved RRT* algorithm is employed to sample various water surface floating object points within the waterway. The algorithm then assesses whether the water surface floating objects are located along the planned path. If they are, a path is generated; otherwise, the search for an appropriate path continues. This approach ensures efficient and adaptive path-planning for the ASV while avoiding collisions with floating objects.

(4) The predicted route trajectory is segmented, and within each segment of the cleaning route, dynamic searching and dynamic obstacle avoidance are conducted using the Dynamic Window Approach (DWA). Additionally, an artificial potential field obstacle avoidance algorithm is incorporated to provide real-time obstacle avoidance for underwater obstacles like reefs and submerged rocks. This combination of techniques enhances the ASV's ability to navigate through complex environments, ensuring efficient and safe cleaning operations.

(5) The ASV follows the optimal trajectory to clean water surface floating objects and checks whether the cleaning is complete. Once the cleaning is finished, it reaches the designated target planning point. The deatilaed flowchart of this algorithm is shown in Figure 2

The image analysis and processing modules of UAV and ASV can use existing SIFT (Scale-Invariant Feature Transform) algorithms to extract feature points from the images collected in the data acquisition layer. They generate feature description vectors for all the feature points and then transfer the visual images and their corresponding feature points to a deep neural network training model, which is saved in the UAV and ASV floating object image recognition libraries. Equipped with cameras, the system can then recognize the floating objects based on their images. The ASV utilizes a combination of different sensors to achieve environmental perception, enabling dynamic obstacle avoidance and the tracking of floating objects.

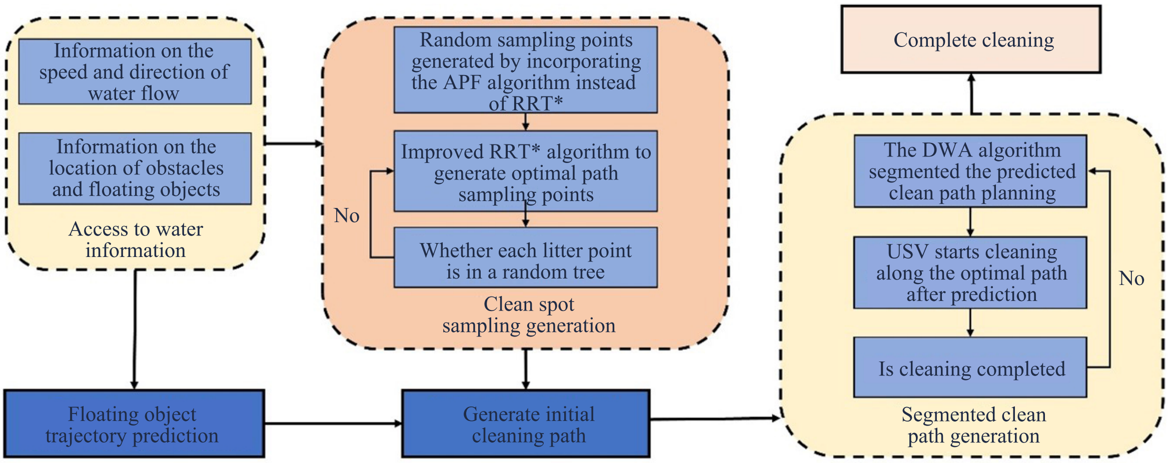

Information on the speed and direction of water flow
Information on the location of obstacles and floating objects
Access to water information
Random sampling points generated by incorporating the APF algorithm instead of RRT*
Improved RRT* algorithm to generate optimal path sampling points
No
Whether each litter point is in a random tree
Clean spot sampling generation
Complete cleaning
The DWA algorithm segmented the predicted clean path planning
USV starts cleaning along the optimal path after prediction
No
Is cleaning completed
Segmented clean path generation
Floating object trajectory prediction
Generate initial cleaning path

Figure 2. Clean path algorithm flow

3. Improved RRT* converged path-planning algorithms

3.1 *RRT* tree search algorithm*

The process of the RRT* tree search algorithm for rapidly finding the optimal path to reach the target point is mainly divided into three stages: quickly finding the initial path, reselecting the parent node, and reassigning random numbers.

(1) Sampling in space from the starting point towards the target point direction to determine the direction from the growing tree to the generated random point target. By continuously traversing the existing nodes on the growing tree, the initial path is determined to be 0-4-6-10.

(2) In the process of tree backpropagation, the parent node 4 of node 6 is found, and by growing with random step size, a new node 9 is created. Within the given radius range near the new node, search for neighboring nodes as alternative parent nodes. Calculate the path cost from each node to the starting point and the new node. Finally, connect to the neighboring node with the lowest path cost.

(3) By using random rewiring numbers, the goal is to minimize the connection cost between random tree nodes as much as possible, which leads to some nodes having smaller path costs. For the specific process, refer to Figure 3.

Continuously updating parent nodes is a core process of the RRT* algorithm. As shown in Figure 3, where node 9 is the newly generated node, the original path 0-4-6-10 had a cost of 10 + 5 + 1 + 5 = 21, The three alternative nodes 6, 5, 8 combined with P_n form the paths 0-4-9-6-10, 0-1-5-9-6-10 and 0-1-5-8-9-6-10 with costs of 10 + 4 + 1 + 5 = 20, 3 + 6 + 3 + 1 + 5 = 18 and 3 + 6 +1 + 3 + 1 + 5 = 19. If changing the parent node of the new node 9 to 4 results in a minimum path cost of 0-1-5-9-6-10 = 18, which is less than the original path cost of 21, then the parent node of 9 is changed to node 5, generating a new random tree. Finally, the distance from the previous node to the final target point is evaluated concerning the step size. If it is equal to the minimum step size and allows for a direct path to the target point, the process is concluded.

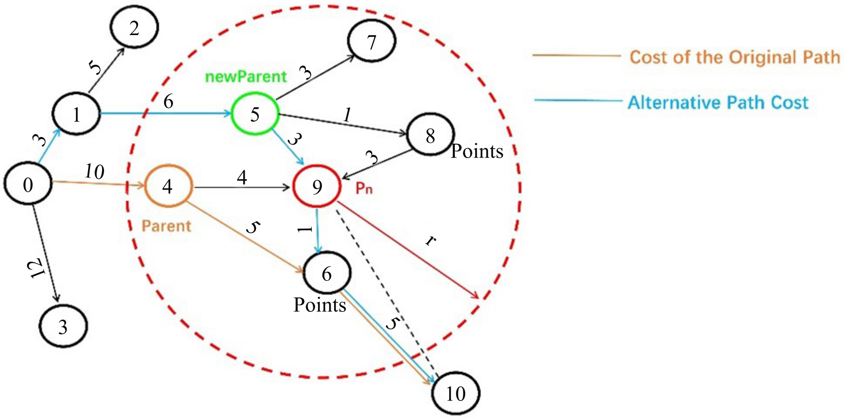


Figure 3. RRT* reselects the parent node and reroutes the process

3.2 *Improved RRT* algorithm*

3.2.1 *Coordinate assessment*

In the process of searching for water surface floating objects during ASV's cleaning operation, it checks whether the path search process passes through the coordinates of the water surface floating object (x_n, y_n) that have been obtained. If the required cleaning point has not been cleaned, the path search continues. If a cleaning point is found during the search, it is used to replace the original key node in the segmented DWA search. This allows the ASV to adapt its path to the location of the cleaning point and perform the cleaning task. As shown in Figure 4, the path after the original redistribution and selection of new nodes in RRT* is 0-1-2-3, and a water surface floating object cleaning point 4 has been found. The key node 1 is replaced with node 4. It is determined that the optimal path from node 4 to node 3 is 0-4-3.

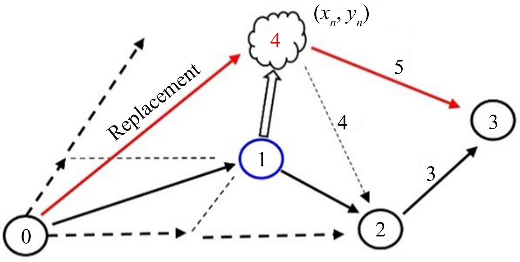


Figure 4. Garbage cleaning points replace key nodes

3.2.2 *Constraint limitations*

The RRT* algorithm has issues such as a high number of iterations and large space complexity (As shown in Figure 5a. In more complex scenarios, the expansion of the random tree in space can be quite complex, and it may not achieve the desired results as expected. Contrary to the known optimization algorithms for RRT*, which primarily perform a route search first and then search for a better path through constraints, the reverse approach involves imposing constraints first on the original algorithm and then searching for a better path.

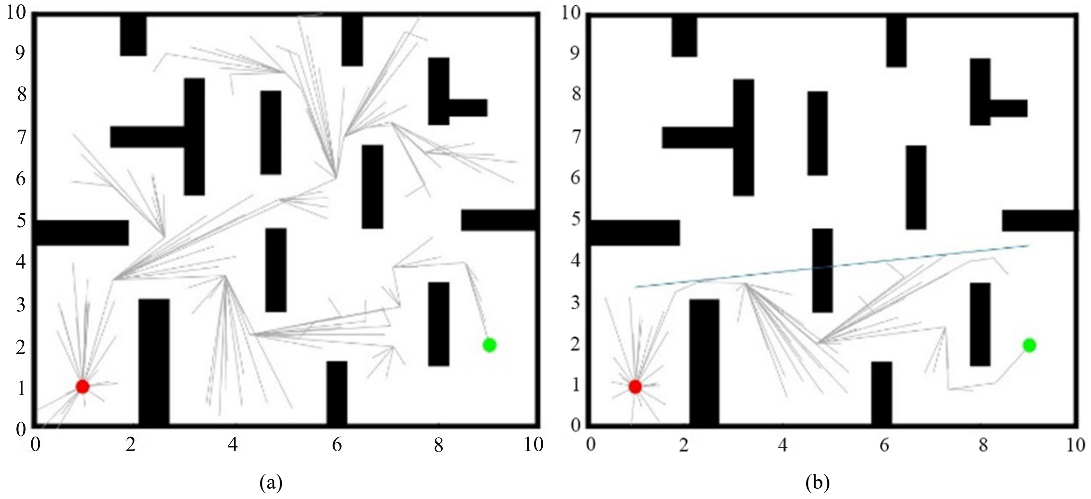


Figure 5. Improved RRT* algorithm compared to traditional RRT* algorithm

As shown in Figure 5b, a straight line is drawn between the initial point (x_0, y_0) of the small boat and the final target point (x_m, y_m). Initially, this line intersects static obstacles, which are denoted as initial obstacles $x_i(x_1, x_2, x_3, \cdots)$, The line is then extended in the vertical direction to create two lines: one is shifted upward by Δs, and the other is shifted downward by Δs, when one of these lines no longer intersects the initial obstacles $x_i(x_1, x_2, x_3, \cdots)$, the initial point and the final point are connected. At this point, RRT* is performed between the two lines created by the extension, forming constraints for RRT*. The pseudocode formula for this is as follows:

$$\begin{cases} y_i^+ > (y_m - y_0)/(x_m - x_0)(x_i - x_0) + y_0 + s \\ \\ y_i^- < (y_m - y_0)/(x_m - x_0)(x_i - x_0) + y_0 - s \end{cases} \tag{2}$$

Comparing the traditional RRT* algorithm with the optimized and improved RRT* algorithm, the experimental data for 10 sets are presented in Figure 6. From the data, it can be observed that the optimized and improved RRT* algorithm performs better in terms of path-planning efficiency. It has significantly fewer iterations, a shorter path length, and fewer turning points compared to the traditional RRT* algorithm. The optimized and improved RRT* algorithm reduces the number of iterations by approximately 66.6%, shortens the path length by around 9.1% and decreases the number of nodes by approximately 33.3% when compared to the traditional RRT* algorithm. The pseudocode for improving and optimizing the random number algorithm is shown in Table 1.

Table 1. Improved optimization of random number pseudocode

Index	Code
1	$function\ rrtstar = rrtstar([x_0, y_0], xn, yn, xm, ym, obs, step, countMax, TrashSum)$
2	$function\ getlinelimit$
3	$sum = getObsSum()$
4	$Obs[] = getObs()$
5	$while\ sum1 < sum\vert\vert sum2 < sum$
6	$for\ 1: sum$
7	$ifobs1(i) == 1line + escape(obs1(i), yi^{+})$
8	$sum1++; obs1(i) = 0$
9	$endif$
10	$endfor$
11	$for\ 1: sum$
12	$if\ obs2(i) == 1\&\&line - escape(obs2(i), yi^{-})$
13	$sum2++; obs2(i) = 0$
14	$endif$
15	$endfor$
16	$yi^{+} = yi^{+} + s; yi^{-} = yi^{-}s$
17	$endwhile$
18	$return\ yi^{+}, yi^{-}$
19	$points(1, :) = [x_0, y_0];$
20	$while\ count < countMax\ \&\&\ \ over$
21	$if\ norm([x_m, x_m] - points(currentIndex, :)) < growStep\&\&n == TrashSum$
22	$newPoint = target;$
23	$over = true;$
24	$endif$
25	$if\ \ norm([x_n, y_n] - points(currentIndex, :)) < growStep$
26	$n++;$
27	$endif$
28	$randPoint = NewRandPoint();$
29	$currentParent = FindNearestPoint(points, currentIndex, randPoint);$
30	$newPoint = Grow(points(currentParent, :), randPoint, growStep);$
31	$if(!Collisionless(obs, newPoint, points(currentParent, :), k, c_1, c_2))$
32	$continue;$
33	$endif$
34	$whilecurrentParent! = 1\&\&Collisionless(obs, newPoint, points(parent(currentParent), :), k, c_1, c_2)$
35	$currentParent = parent(currentParent);$
36	$endwhile$
37	$currentIndex = currentIndex + 1;$
38	$points(currentIndex, :) = newPoint;$
39	$parent(currentIndex) = currentParent;$
40	$endwhile$
41	$over = false$

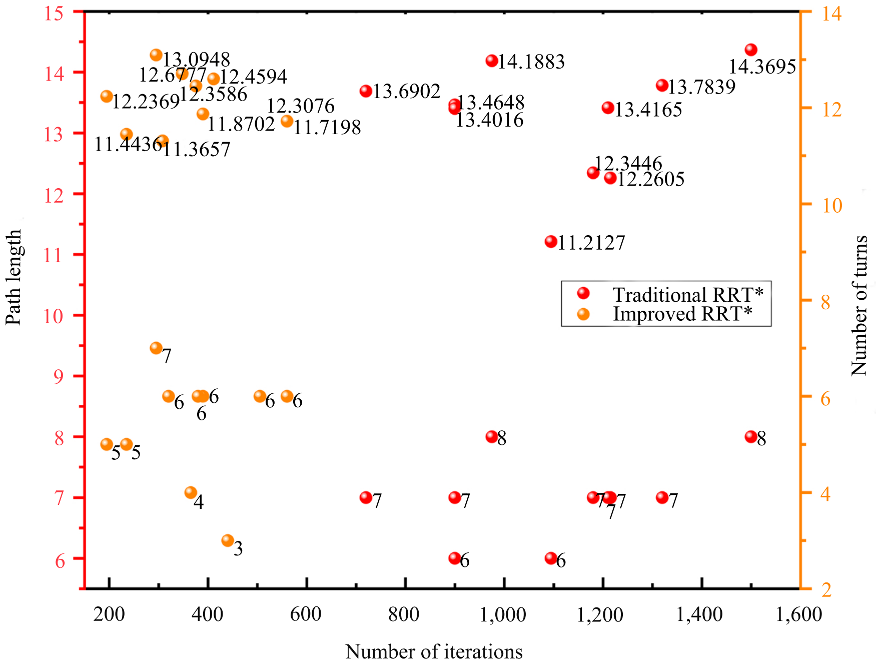


Figure 6. Performance comparison of traditional RRT* and improved optimization RRT* algorithms chart

3.3 *Artificial potential field*

The traditional RRT* algorithm expands in a relatively random direction in the search for a path, which leads to longer search times and lower efficiency. In the RRT* algorithm, adding APF [23] components when adding expansion points introduces randomness while also considering the impact of obstacles and the target point on the expansion of the current node. This helps improve the efficiency of the search process. This makes the expansion of the random tree more purposeful, improving the search efficiency. The attractive function of the artificial potential field is as follows:

$$U_{att}(q) = 1/2\ \zeta\rho^2\ (q,\ q_{goal}\ ) \tag{3}$$

where ε is the attractive force coefficient and $\rho(q,\ q_{goal})$ is the distance between the current point and the target point. The repulsive function of the artificial potential field is as follows:

$$\begin{cases} Urep(q) = \dfrac{1}{2}\eta\left(\dfrac{1}{\rho(q,\ q_{obs})} - \dfrac{1}{\rho_0}\right)^2,\ if\ \rho(q,\ q_{obs}) < \rho_0 \\ \\ 0,\ if\rho(q,\ q_{obs}) \geq \rho_0 \end{cases} \tag{4}$$

where η is the repulsive force coefficient, $\rho(q,\ q_{obs})$ is the distance between the current point and the obstacle point, and ρ_0 is the range within which obstacles exert a repulsive force on the current node. Beyond this range, obstacles do not exert a repulsive force on the current node.

In this method, to solve dynamic obstacle avoidance for ASV, the pseudocode formula is as follows:

$$R_{rad} \leq [(x_{red}, y_{red}) - (x_n, x_y)] + \tag{5}$$

R_{rad} represents the radius of the small red sphere in the diagram, (x_{rad}, y_{rad}) represents the coordinates of the small red sphere, (x_n, y_n) represents the coordinates of various obstacle points, and γ is the defined safety distance, which is 0.75 in this case. The trajectories of positive and negative charges in the electric field are shown in Figure 6：

The red arrow in Figure 7 represents the ASV's repulsive force against obstacles, which is defined as the repulsive coefficient between the black square obstacle and the ASV. By setting the safety distance parameter between the ASV and the obstacle, when the ASV approaches the obstacle during navigation and reaches this parameter threshold, the ASV will automatically correct its course and maintain a set safety distance from the obstacle.

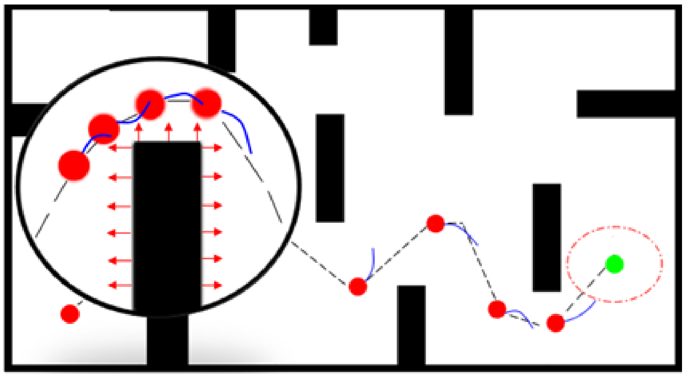


Figure 7. Simulate artificial potential fields

The pseudocode for the Artificial Potential Field method in the fusion algorithm is presented in Table 2.

Table 2. Pseudocode for artificial potential field method

Index	Code
1	$defAPF(points, goal, obs)$:
2	$attractiv_force = calculat_attractiv_force\ (points, goal)$
3	$repulsive_force = calculate_repulsive_force\ \ (points, obs)$
4	$update_node_position(points, total_force)$
5	$return\ update_node_position$

3.4 *Dynamic-window approach*

Incorporating the DWA [24] involves segmenting the path of water surface debris cleared by the ASV through the RRT* algorithm. During each segment of the water surface debris-clearing process, the ASV's heading, velocity, obstacle distances, and other factors are weighted and scored, and these scores are aggregated according to a certain ratio. This process leads to the identification of the path with the highest total score, which is considered the best path. Assuming that the ASV moves in a straight line during a very short period, with an angular velocity of ω, and a linear velocity of v, the ASV's motion within time Δt can be derived from the following formula:

$$\begin{cases} x = x + v_x \Delta t cos(\theta_t) - v_y \Delta t sin(\theta_t); \\ y = y + v_x \Delta t sin(\theta_t) - v_y \Delta t cos(\theta_t); \\ \theta_t^{'} = \theta_t + \omega \Delta t \end{cases} \tag{6}$$

By using the calculation formula and the collected velocity information, it is possible to estimate the trajectory of the ASV in the next time interval. The evaluation function used is:

$$G(v, \omega) = \sigma[\alpha \times H(v, \omega)] + \beta \times D(v, \omega) + \gamma \times V(v, \omega) \tag{7}$$

When the $G(v, \omega)$ value is minimized, the optimal path is obtained. Here: $H(v, \omega)$is the angle difference evaluation function between $Head(v, \omega)$, $D(v, \omega)$ is the path evaluation function between $Dist(v, \omega)$, $V(v, \omega)$ represents the velocity evaluation function for $Vel(v, \omega)$. σ is the smoothing factor, and α、β、γ is a coefficient. The specific implementation of the key evaluation functions is as shown in Figure 8.

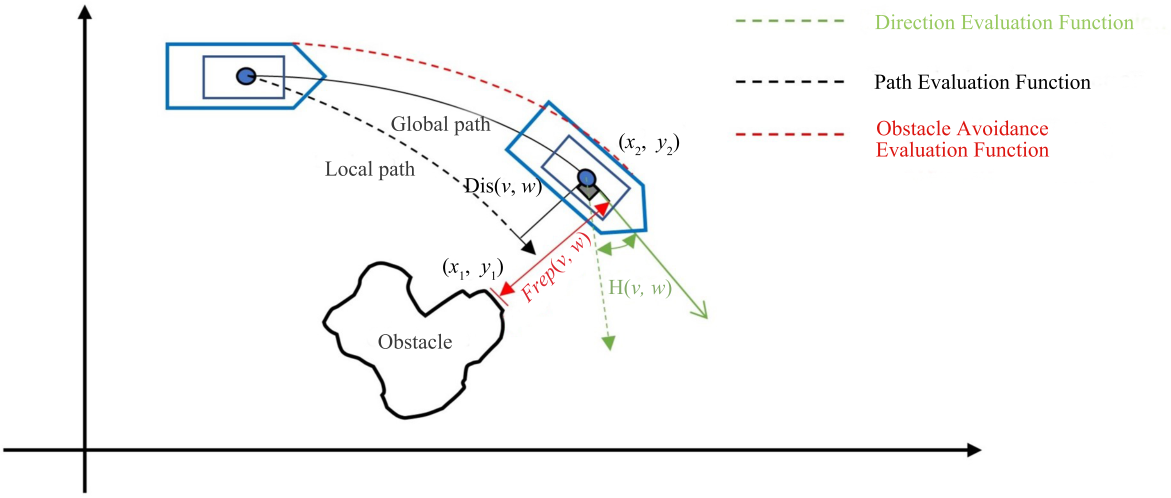


Figure 8. Kinematic model of an unmanned ship

$Head(v, \omega)$ represents the angle difference between ASV and the water surface floating object target when moving at the current velocity. The formula for the angle difference evaluation function $Head(v, \omega)$ is as follows:

$$Head(v, \omega) = 180° - \theta \tag{8}$$

$Dist(v, \omega)$ is used to evaluate the distance between the ASV's navigational trajectory and obstacles, reflecting the ASV's obstacle avoidance capability. If the distance between the ASV's navigational trajectory and an obstacle is greater than the safety radius formed by the ASV, there will be no danger of collision. Conversely, a smaller distance indicates a higher probability of collision, and the ASV will abandon that navigational trajectory. The formula for the path evaluation function D is as follows:

$$Dist\,(v,\,\omega) = min\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{9}$$

The variables$(x_1,\, y_1)$ represent the estimated coordinates of the nearest point to the obstacle based on the ASV's kinematic model, while$(x_2,\, y_2)$ represent the coordinates of nodes from the global path-planning. The function evaluates the path by considering the proximity of the ASV's estimated path to obstacles and the positions of nodes in the global path-planning.

The variable $Vel\,(v,\,\omega)$ is the speed evaluation function. Since ASV may not effectively balance the distance from obstacles and the distance from the target point during movement, this function is designed to allow DWA (Dynamic-Window Approach) to rapidly plan effective paths during ASV's operation. The specific calculation formula for the speed evaluation function is provided as follows.

$$Vel = \frac{V_{max} - V}{V} \tag{10}$$

In this approach, to implement the simulated cleaning vessel using the DWA algorithm (As shown in Table 3) in Matlab2022a, the pseudo-code, and formulas are as shown in Figure 8:

(1) Function $Head(v,\,\omega)$, Where head represents the percentage of the angle, theta is the angle, and tar represents the next surface floating object cleaning target point in the path-planning.

(2) Function $Frep(v,\,\omega)$, frep represents the minimum safe distance percentage, rho is the distance between two points, k_rep is the safety coefficient 1/2, k_rho defines the safe distance as 3.

(3) Function $Vel(v,\,\omega)$, v_max represents the maximum operating speed and represents the driving speed.

Table 3. DWA dynamic window evaluation function pseudocode

Angular difference from target position	Evaluating the distance from the robot trajectory to the obstacle	valuating the robot speed function
$Input:\, x,\, y,\, \theta,\, tar$ $Output:\, functionhead = Head\,(x,\, y,\, \theta,\, tar)$ $tar = tar\,(x1,\, y1)$ $Px = x + cos\,(\theta)\; tar = \frac{tar(x_1,\, y_1)}{\sum\lvert tar(x_1,\, y_1) - (x_i,\, y_i)\rvert}$ $head = (\sum((Px,\, Py) - tar\,(x_n,\, y_n)))^2$ end	$Input:\, x,\, y,\, obs,\, k_rep,\, k_rho$ $Output:\, functionfrep = Frep\,(x,\, y,\, obs,\, k_rep,\, k_rho)$ $if\;\; rho = \sum\lvert\,(x,\, y) - obs\,(x_{obs},\, y_{obs})\,\rvert - R_{obs}$ $frep = 0$ $return$ end $frep = \frac{1}{2} \times k_rep \times \left(\frac{1}{rho} - \frac{1}{k_rho} \times R_{obs}\right)^2$ end	$Input:\, v,\, v_max$ $Output:\, function\; Vel = Velocity\,(v,\, v_max)$ $Vel = \frac{(v_max - v)}{v}$ end

The pseudocode for the Dynamic Window method in the fusion algorithm is shown in Table 4.

Table 4. Dynamic window approach pseudocode

Index	Code
1	$def\ DWA(count_{m}ax,\ robot_{s}tate,\ kinematic_{l}imit,\ rrt_{r}esult,\ dobs,\ obs,\ obsv,\ )$
2	$while\ count < count_{m}ax \Vert robot_{s}tate(3) > 0.5 * kinematic_{l}imit(1)$
3	$if\ \ norm([robot_{s}tate(1),\ robot_{s}tate(2)] - rrt_{r}esult(rrt_{i},\ 1{:}\ 2)) < tar(3)$
4	$success = true;$
5	$return;$
6	$endif$
7	$dobs = getNewdobs(dobs)$
8	$for\ v = v_{min}{:}\ v_{step}{:}\ v_{max}$
9	$for\ w = w_{min}{:}\ w_{step}{:}\ w_{max}$
10	$if\ Collision(sta,\ obs,\ fortime,\ dt,\ dobs,\ obsv)$
11	$continue;$
12	$endif$
13	$for\ t = dt{:}\ dt{:}\ fortime$
14	$[state,\ \ ] = GetTrajectory(sta,\ t,\ dt);$
15	$fit = fit + (Head,\ Frep,\ Velocity);$
16	$endfor$
17	$if\ fit < fit_{m}in$
18	$[fit_{min},\ best_{v},\ best_{w},\ best_{t}raj] = [fit,\ v,\ w,\ traj];$
19	$endif$
20	$endfor$
21	$endfor$
22	$robot_{s}tate = NextState([best_{v},\ best_{w}],\ dt);$
23	$endwhile$

3.5 *Algorithm integration*

DWA algorithm is prone to getting stuck in local optima during path-planning. Therefore, it requires a global path-planning algorithm to provide guidance. While the improved and optimized RRT* algorithm can handle the cleaning of floating surface targets, it cannot facilitate dynamic obstacle avoidance between ASV and moving obstacles. It is not suitable for ASV movement. By incorporating the concept of an artificial potential field into the optimized RRT* algorithm, you can use the optimized and improved RRT* algorithm for global planning. During the expansion of each node, you can enhance the bias in the artificial potential field. At this point, the expansion of nodes is guided by the APF algorithm, which also checks whether each cleaning point is on the generated random points. Finally, the cleaning points are used as key nodes for global path-planning, and the path between adjacent cleaning points is planned using the DWA.

4. Simulation and discussion

4.1 *Uniform flow velocity*

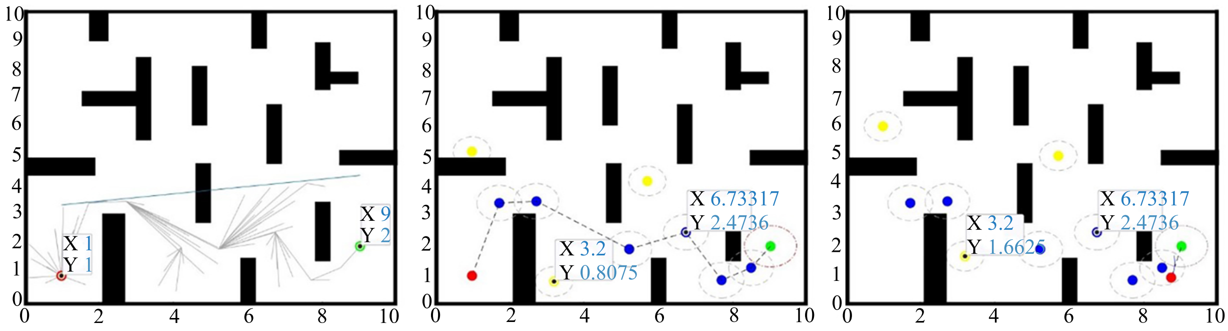
To validate the effectiveness of the improved RRT* algorithm, flow velocity prediction method, and fusion algorithm for multi-surface floating object path-planning, simulation experiments were conducted using MATLAB software. In the simulations In the following simulation experiment (As shown in Figure 9), The black area indicates water surface reef obstacles, yellow dots indicates dynamic points representing other moving surface obstacles, blue dots represents dynamic surface floating object points to be cleared, red dots represents the ASV, and green dots represents the RRT* planning target points. The experimental simulation area was a square with dimensions of 10 cm × 10 cm. The starting point had coordinates (1, 1), and the target point had coordinates (9, 2). The experiments were conducted in uniform flow velocity water environments, water environments with periodically changing flow velocity, and water environments with non-periodically changing flow velocity. In these simulations, it was stipulated that the maximum speed of the ASV (red dots) was 0.8 m/s, and its acceleration was 0.8 m/s². The yellow points representing moving obstacles had a stipulated speed of 0.2 m/s. Simulation experiments implementing the optimized and improved RRT* algorithm, the Dynamic Window Approach (DWA), and the Artificial Potential Field (APF) method were conducted using MATLAB R2022a. The hardware configuration and software parameters for the simulation are outlined in Table 5.

Table 5. Simulate the parameters of software and hardware equipment

Hardware	Configuration	Parameters	Values
Processor	AMD Ryzen 7 5800H 3.20 GHz	RRT* Maximum Iterations	50000
Memory	16.0GB	DWA Maximum Iterations	10000
Operating System	Win 10, x64	Variable Bounds	[10, 10]
Software Tools	MATLAB2022a	Data Maintenance	0.05

The figure illustrating the results of the ASV's surface debris cleaning in a uniform flow velocity water environment with multiple target points is shown in Figure 9.

As shown in Figures 9 and 10 above, in a uniform flow velocity water environment, compared to the traditional RRT algorithm, the optimized and improved RRT* algorithm significantly reduces the iteration count and search time of the RRT* algorithm when searching for surface debris coordinates. Moreover, it has fewer turning points, making it faster and achieving better path-planning results. By integrating the dynamic window approach for obstacle avoidance, this method is capable of both avoiding obstacles effectively and accurately locating surface debris to obtain a globally optimal path. During the segmented debris cleaning process, ASV maintains smooth velocity and achieves stable heading for debris cleaning. As a result, the path remains relatively smooth during the debris-cleaning process.

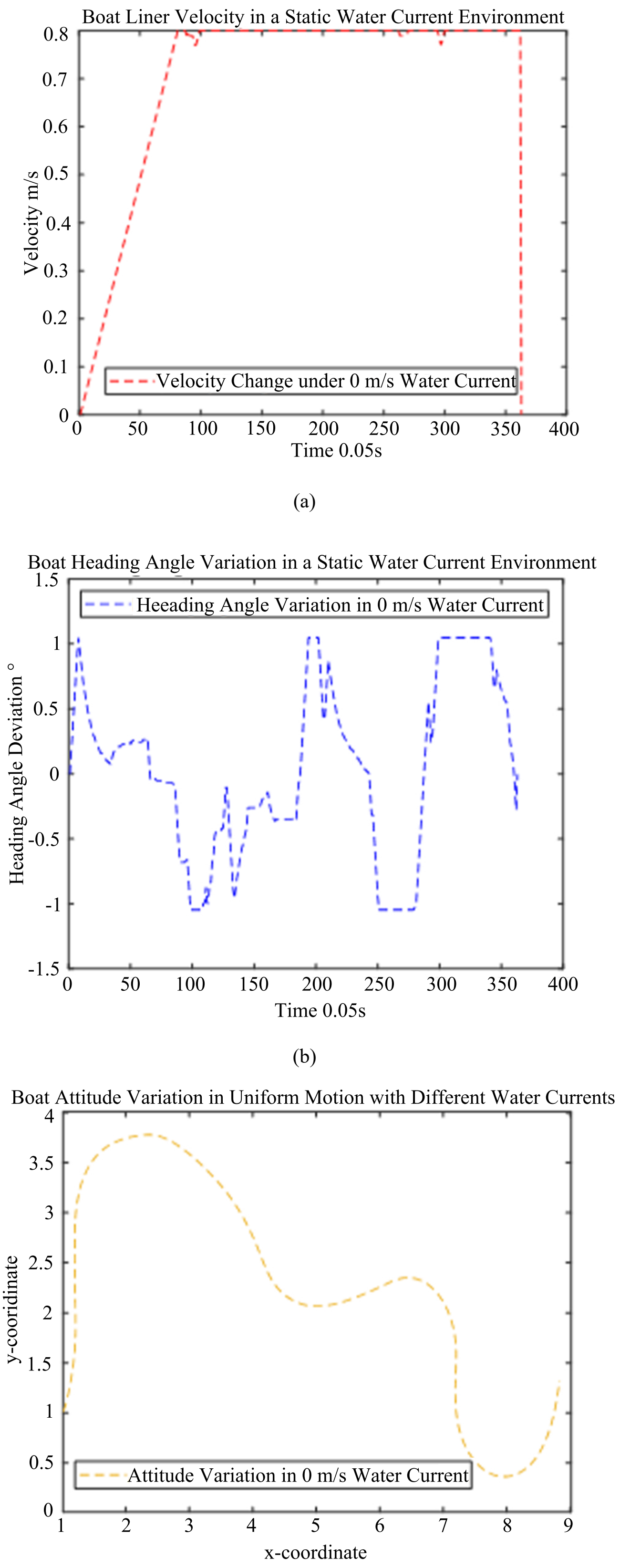


(a)

(b)

(c)

Figure 9. The velocity and attitude changes of ASV during the displacement process for cleaning in a static water environment



(a)

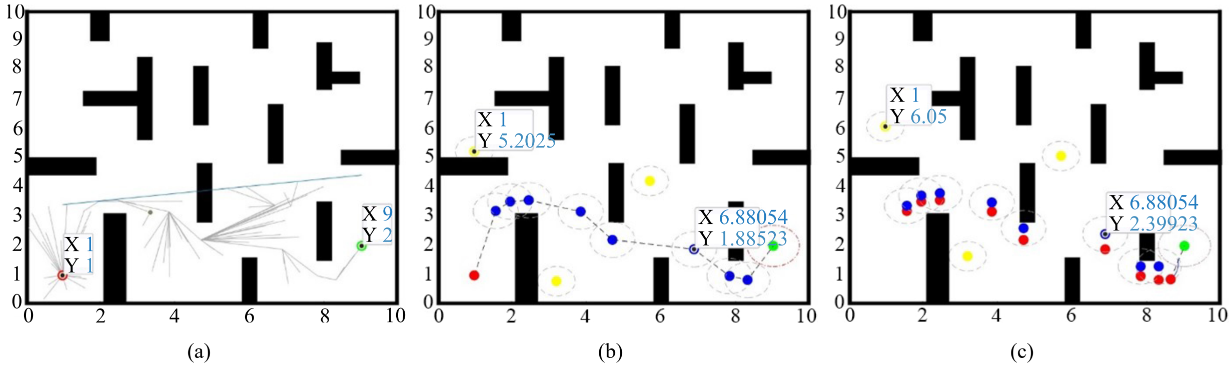
(b)

(c)

Figure 10. The velocity and attitude changes of ASV during the displacement process for cleaning in a static water environment

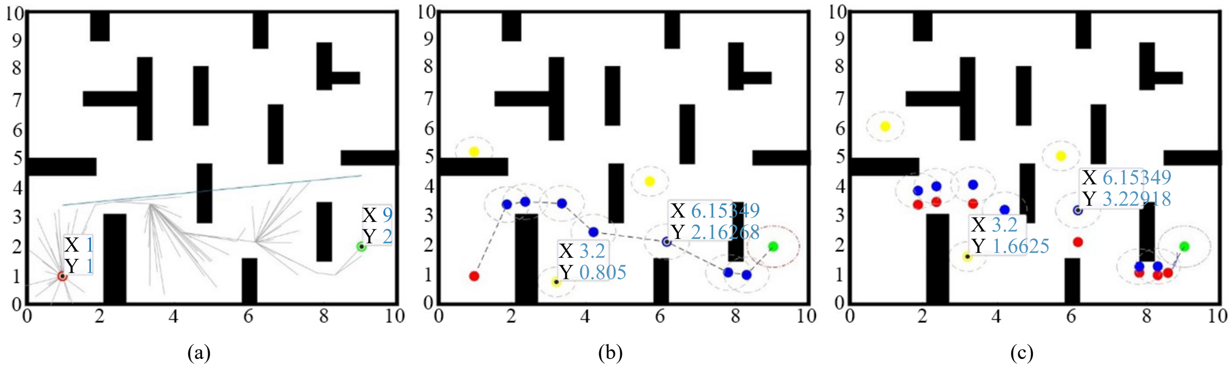
4.2 *Flow velocity with periodic fluctuations*

The figures, as shown in Figures 11-13, depict the surface debris cleaning performance of the ASV in environments with periodically fluctuated water flow, in contrast to the uniform water flow.



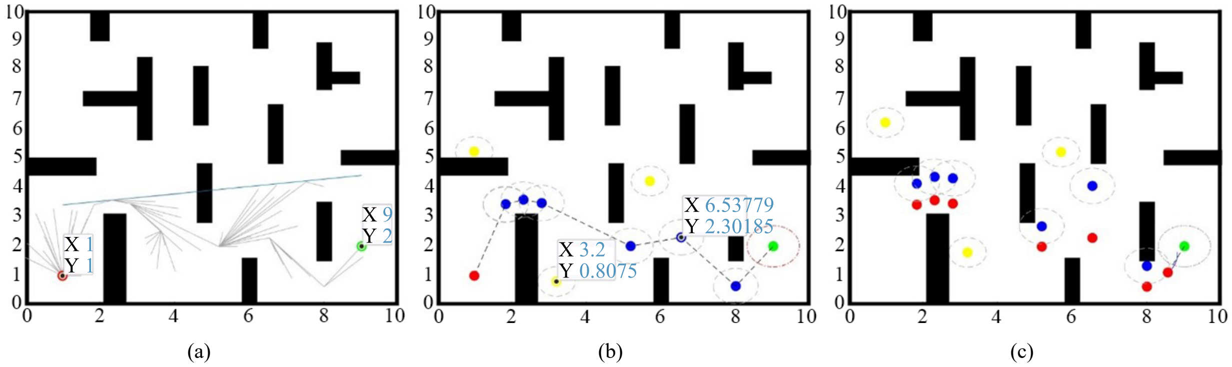
(a) (b) (c)

Figure 11. ASV cleaning trajectory route under a uniform velocity water flow of 0.05 m/s in periodically varying waters



(a) (b) (c)

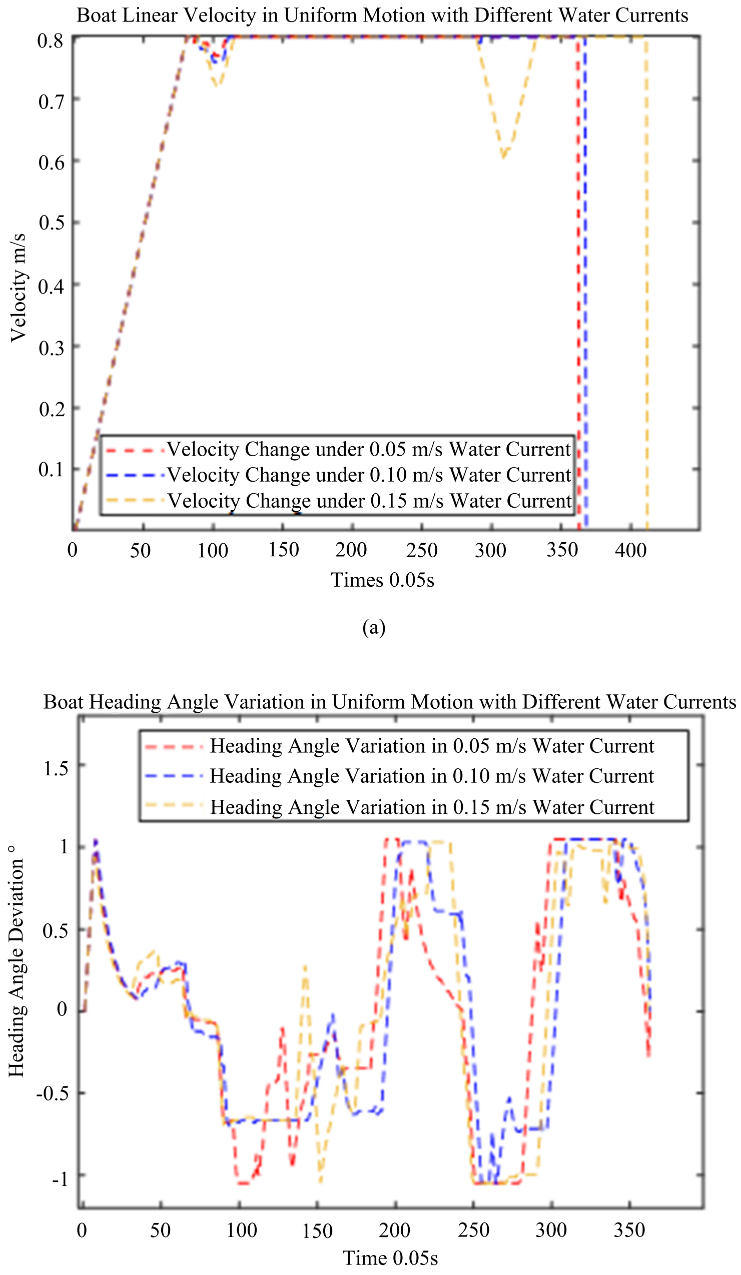
Figure 12. ASV cleaning trajectory route under a uniform velocity water flow of 0.10 m/s in periodically varying waters



(a) (b) (c)

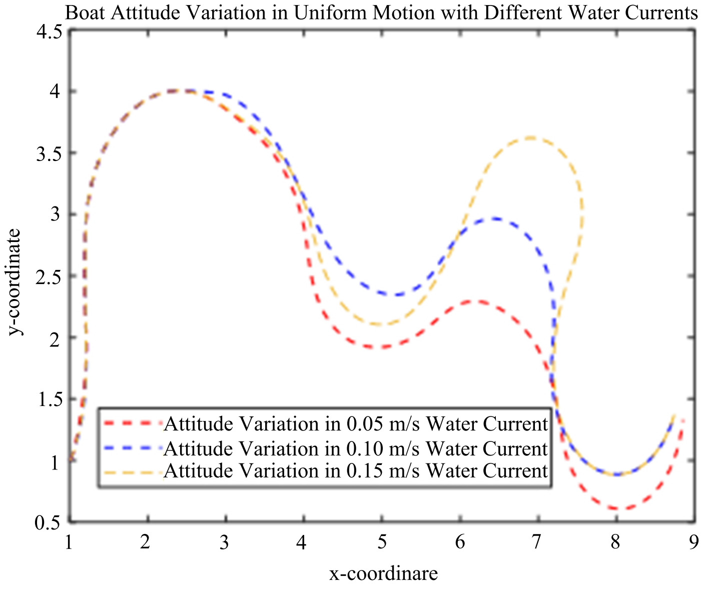
Figure 13. ASV cleaning trajectory route under a uniform velocity water flow of 0.15 m/s in periodically varying waters

This simulation assumes the presence of periodic water flow disturbances in the water environment and aims to verify the stability of the ASV's hull during its navigation in the presence of dynamic water flow effects. It is assumed that the ASV is affected by water flow disturbances from the south to the north, with three different water flow speeds (As shown in Figures 11-13): under the conditions of water velocity of 0.05 m/s (Figure 11), 0.10 m/s (Figure 12), and 0.15 m/s (Figure 13) respectively.In the absence of wind. The simulation results compare the ASV's navigation trajectories under these three different water flow disturbances, as shown in Figure 14. In a comprehensive analysis, it is observed that different water flow disturbances have a certain impact on the ASV's traveling speed under external interference. The simulation environment is complex, and the ASV needs to track surface floating objects while avoiding threats from terrain and moving obstacles. This leads to significant changes in heading angles during avoidance maneuvers, but after avoiding obstacles, the ship's heading angle quickly returns to a stable state. In scenarios without obstacle threats in the terrain, the heading angle remains stable. Although this may lead to longer times to reach the target direction, the overall performance of the ASV suggests good stability.



(a)

(b)

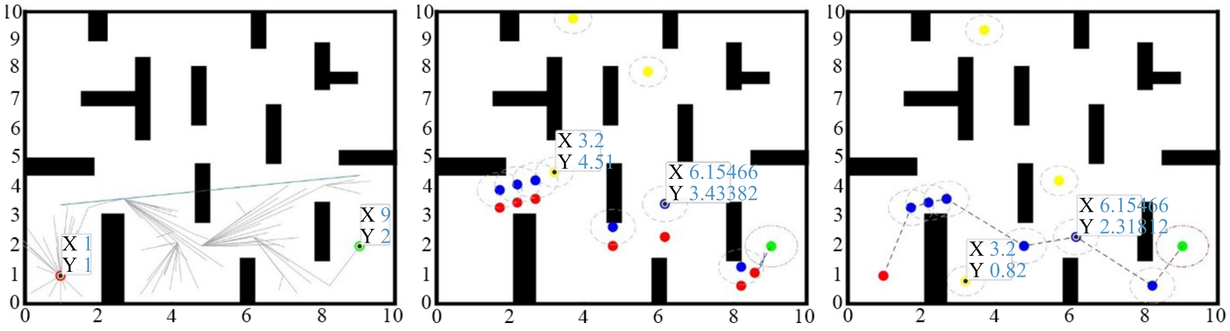


(c)

Figure 14. The speed and attitude change during the cleaning displacement of the unmanned cleaning ship under the uniform water flow of 0.05 m/s, 0.10 m/s, and 0.15 m/s, respectively

4.3 *Flow velocity with variable fluctuations*

The effectiveness of ASV in cleaning multiple surface floating objects in a non-static water environment with non-periodic variable-speed water flow is shown in Figure 15.



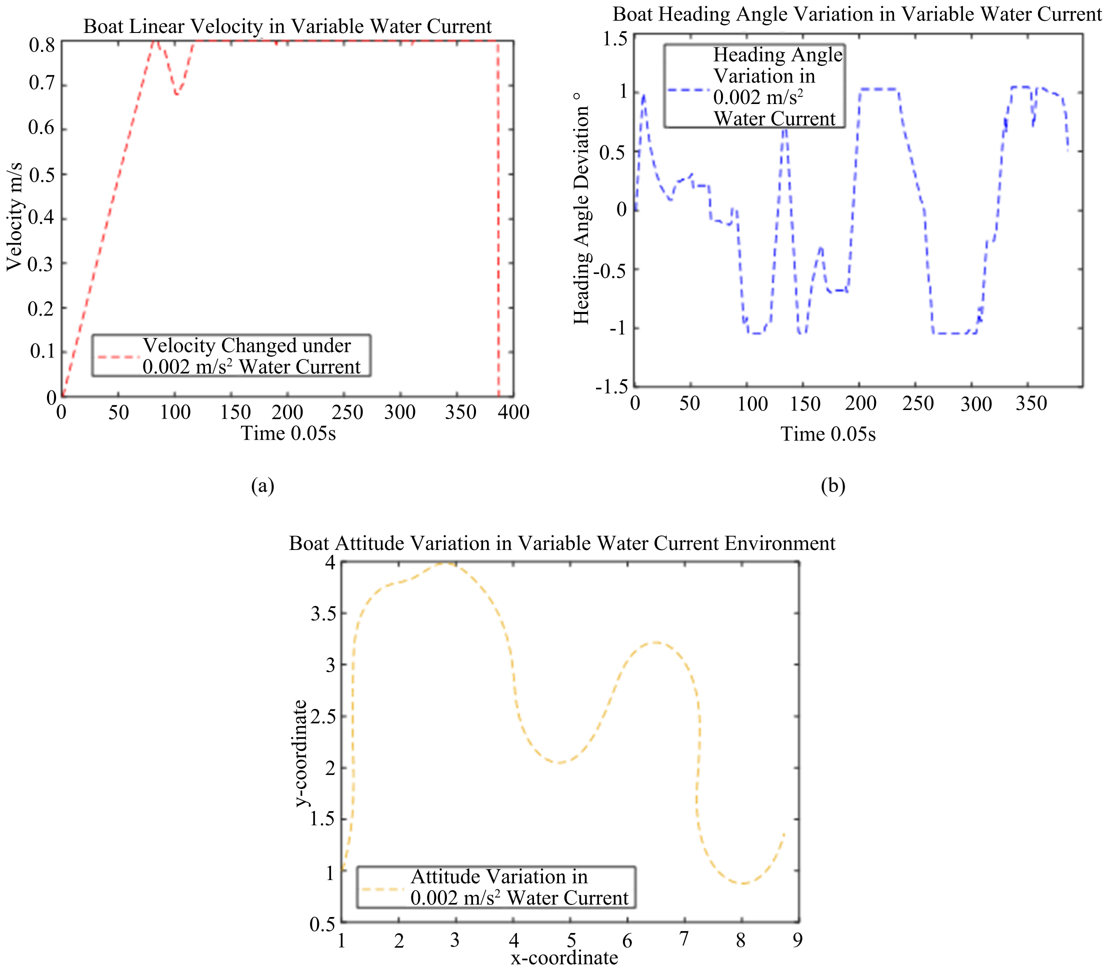
(a)

(b)

(c)

Figure 15. ASV cleaning trajectory route under 0.002 m/s² variable speed water flow in non-periodically varying waters

To verify whether the algorithm can achieve surface floating object prediction and cleaning in variable-speed water flow, as shown in Figures 15 and 16, a digital simulation experiment was conducted in the same environment with an added water flow acceleration of 2 mm/s^2. It can be observed that in comparison to cleaning under static or uniform water flow conditions, when water flow fluctuation is relatively small, specifically for t $\leq$ 15 s the impact on the boat's travel speed and heading angle is relatively minor. When t $\geq$ 15 s, the water flow speed increases significantly, and the boat's travel speed and heading angle are increasingly influenced by the water flow speed. This has a linear impact on the calculation of surface floating object poses and the ASV's movement. While this may result in a decrease in the efficiency of ASV's work completion, it is still able to complete the cleaning task.



(a)

(b)

(c)

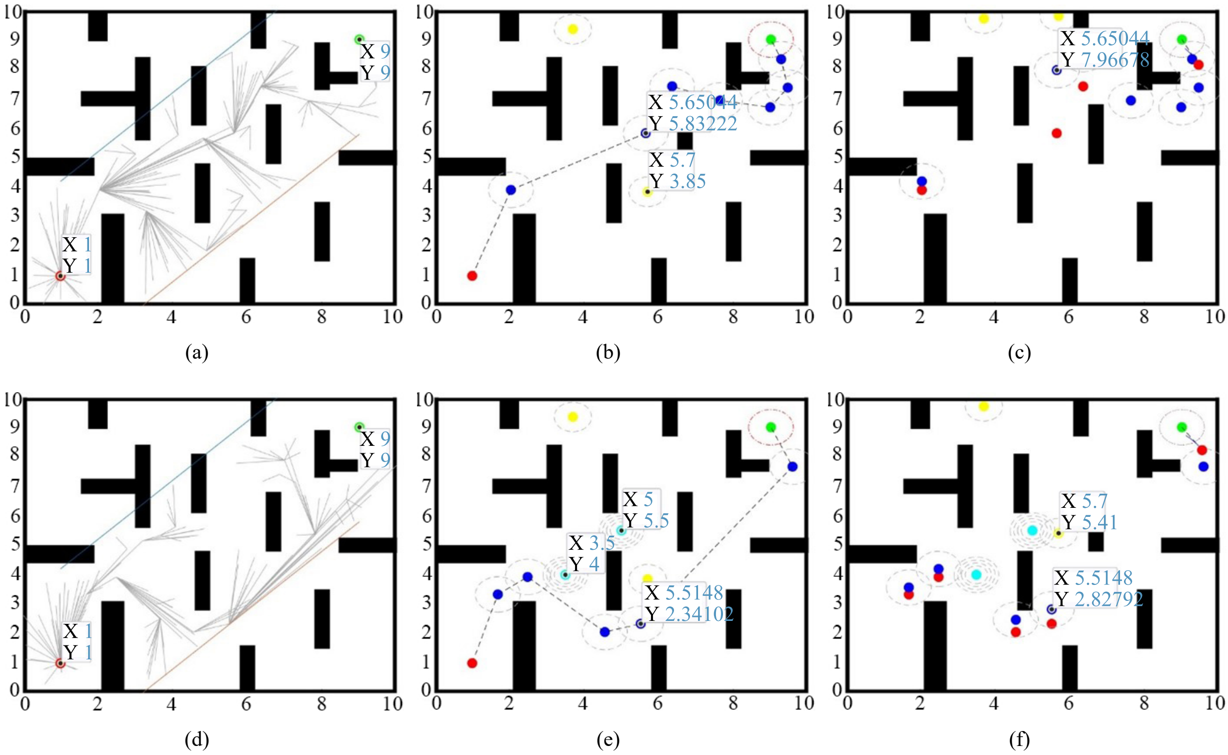
Figure 16. The speed and attitude changes of the ASV during the displacement process for cleaning under a variable water flow of 0.002 m/s²

4.4 *Special scenario simulations*

4.4.1 *Impact of eddies*

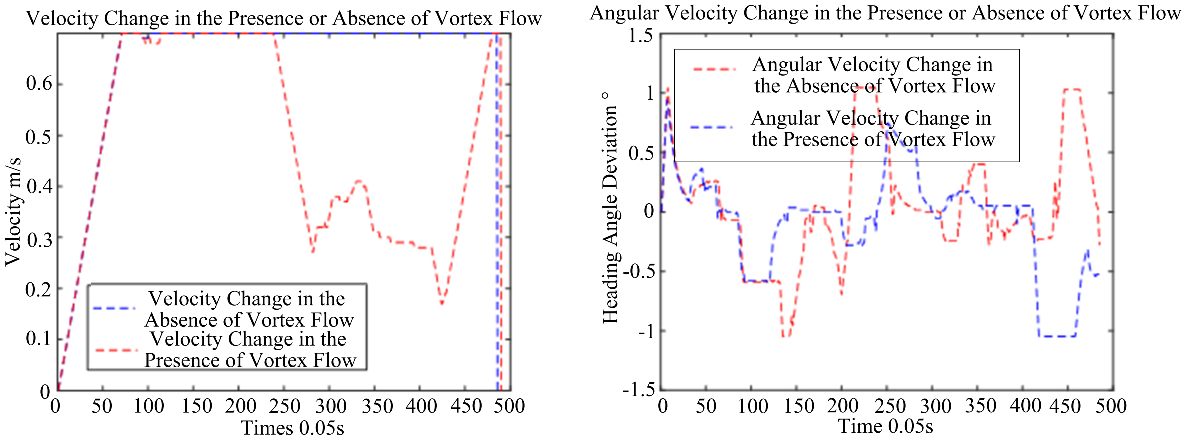
To address the presence of vortex phenomena in local river channels, ASV conducts a survey of the river channel based on its onboard sensors. The localized vortex phenomena are defined as obstacle targets. ASV uses obstacle avoidance techniques to complete the garbage cleaning task. The coordinates of the vortices are specified as (3.5, 4) and (5, 5.5). The ASV (red dots) has a maximum linear velocity of 0.7 m/s and an acceleration of 0.2 m/s², as illustrated in Figures 17 and 17.

In real environments, varying water flow speeds can lead to localized vortex phenomena. Therefore, in the simulation experiments, two vortices are placed along the ASV cleaning path to test the effect of vortices on the ASV's movement. The data graphs indicate that when the ASV is not affected by vortices, its position and speed remain stable as it moves uniformly. However, when the ASV encounters a local vortex during its journey, the vortex does have some influence on the ASV's position. Nonetheless, as the ASV approaches the vortex, it gradually reduces its movement speed, and through the adjustment of its navigation angle using the ASV's propellers, the vessel's position deviates from the vortex attraction. The ASV then selects a new route to continue its cleaning task.

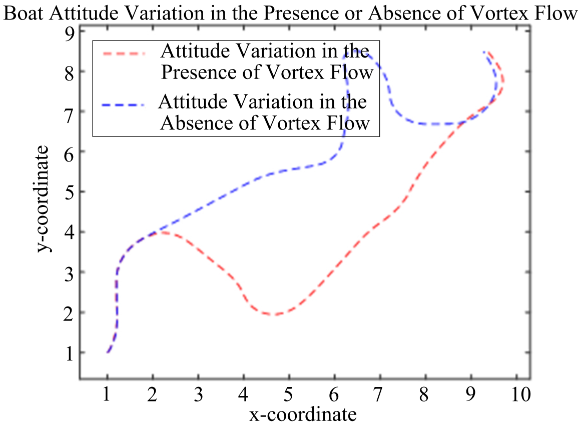


(a) (b) (c) (d) (e) (f)

Figure 17. Illustrates the trajectory route of the ASV for cleaning in dynamic water bodies with or without the presence of vortices



(a) (b)

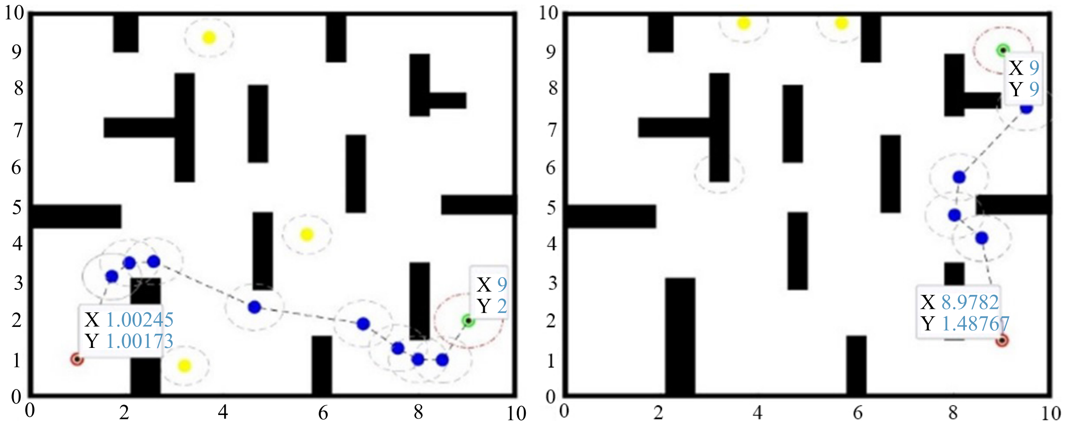


(c)

Figure 17. Illustrates the displacement velocity and attitude changes of the ASV under the influence of vortices, both with and without interference

4.4.2 *Segment-based fixed-point cleaning*

In response to the special characteristics of river and water flow fluctuations, as well as the challenge of large water areas with dispersed debris in urban rivers or lakes, an iterative and stepwise approach will be employed for thorough cleaning. After the UAV conducts an initial survey of floating debris coordinates and the ASV completes the cleaning, the cleaned areas will be marked with green nodes on the map. Subsequently, another UAV mission will be launched to detect new coordinates of floating debris. The ASV will receive this new information and proceed to re-plan the cleaning path. Care will be taken to avoid selecting previously cleaned paths to prevent redundant cleaning efforts. This process will be repeated until the entire water area is thoroughly cleaned, as illustrated in Figure 18.



(d)

(e)

Figure 18. Depicts the segmented cleaning trajectory route of the ASV

As illustrated in Figure 18. The left image shows the first cleaning segment, which starts at the red coordinate point (1, 1) and ends at the green coordinate point (9, 2). The right image shows the second cleaning segment, which starts at the red coordinate point (8.9, 1.5) and ends at the green coordinate point (9.9). The cleaning path is divided into several segments that increase sequentially until the entire water area is cleaned.

5. Prototype experiments

5.1 *Experimental setup*

The current prototype experiment aims to validate the feasibility of the proposed algorithms in a real water environment. An Autonomous Surface Vehicle (ASV) prototype, equipped with a homemade surface cleaning and monitoring robot, was utilized for testing. The ASV has a monolithic shell made of 1mm thick stainless steel, weighing approximately 18 kg. The buoyant volume of the float is 0.123 cubic meters, with a weight of around 3.7 kg. According to the formula $F = \rho gV$ a single float can provide approximately 120 kg of buoyancy. With floats installed on both sides, the total buoyancy is 240 kg. The entire boat system weighs about 165 kg and can collect around 60 kg of floating debris on the water surface. Table 6 provides the structural parameters of the unmanned cleaning boat prototype.

Table 6. ASV prototype structure parameter table

Hardware	Configuration	Parameters	Values
Single propeller	65 lbs thrust	11.26 kg self-weight	480 W power
surface floating debris storage box	0.8 dm^3 volume	6.513 kg self-weight	-
float	120 kg buoyancy	3.708 kg self-weight	-
collection tread motor	160 rpm speed	5.26 kg self-weight	500 W power
lithium iron phosphate battery	24 V rated voltage	210 Ah rated capacity	2500 W-4000 W power

The experimental site for this study was the Jinghu Lake at Yangtze Normal University. To ensure the smooth conduct of the experiments, the following experimental plan was established. The experimental plan is shown in Figure 19.

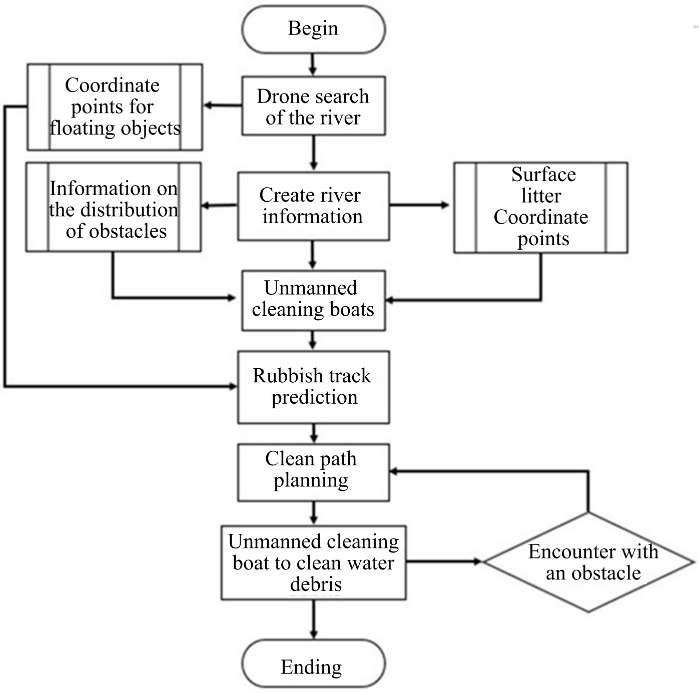

Begin
Drone search of the river
Coordinate points for floating objects
Create river information
Information on the distribution of obstacles
Surface litter Coordinate points
Unmanned cleaning boats
Rubbish track prediction
Clean path planning
Unmanned cleaning boat to clean water debris
Encounter with an obstacle
Ending

Figure 19. Simulate artificial potential fields

5.2 *Real machine demonstration*

This experiment tests the ASV's movement without considering the friction with water, and the water flow is measured by the UAV's speedometer to be in a non-uniform state, approximately in the range of 0.016 m/s to 0.031 m/s. There is a slight wave influence on the water, and the experiment is conducted under calm wind conditions. The coordinates (107.2625E, 29.7495N) are chosen as the starting point for both the UAV and ASV.

The UAV flies in the airspace above the artificial lake with an initial heading of 270°. It uses radar and a camera to build a map and determine the target positions of the water surface floating objects. Once this information is obtained, it is transmitted to the ASV, and an upper-level computer system monitors the real-time status of the UAV.

The ASV starts with an initial heading of 270°. After receiving the information transmitted by the UAV, the ASV forms a cleaning route and begins its operation. The upper-level computer system collects the feedback information from the UAV, generates a fitting curve using MATLAB, and continuously monitors the real-time navigation trajectory of the ASV. Figure 20 shows the predicted trajectory and the actual navigation trajectory obtained from the ASV prototype experiment.

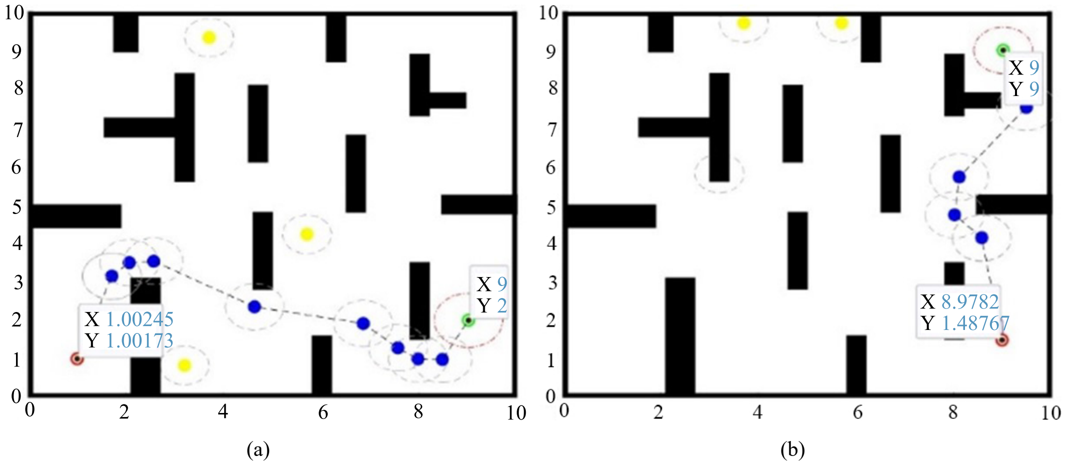
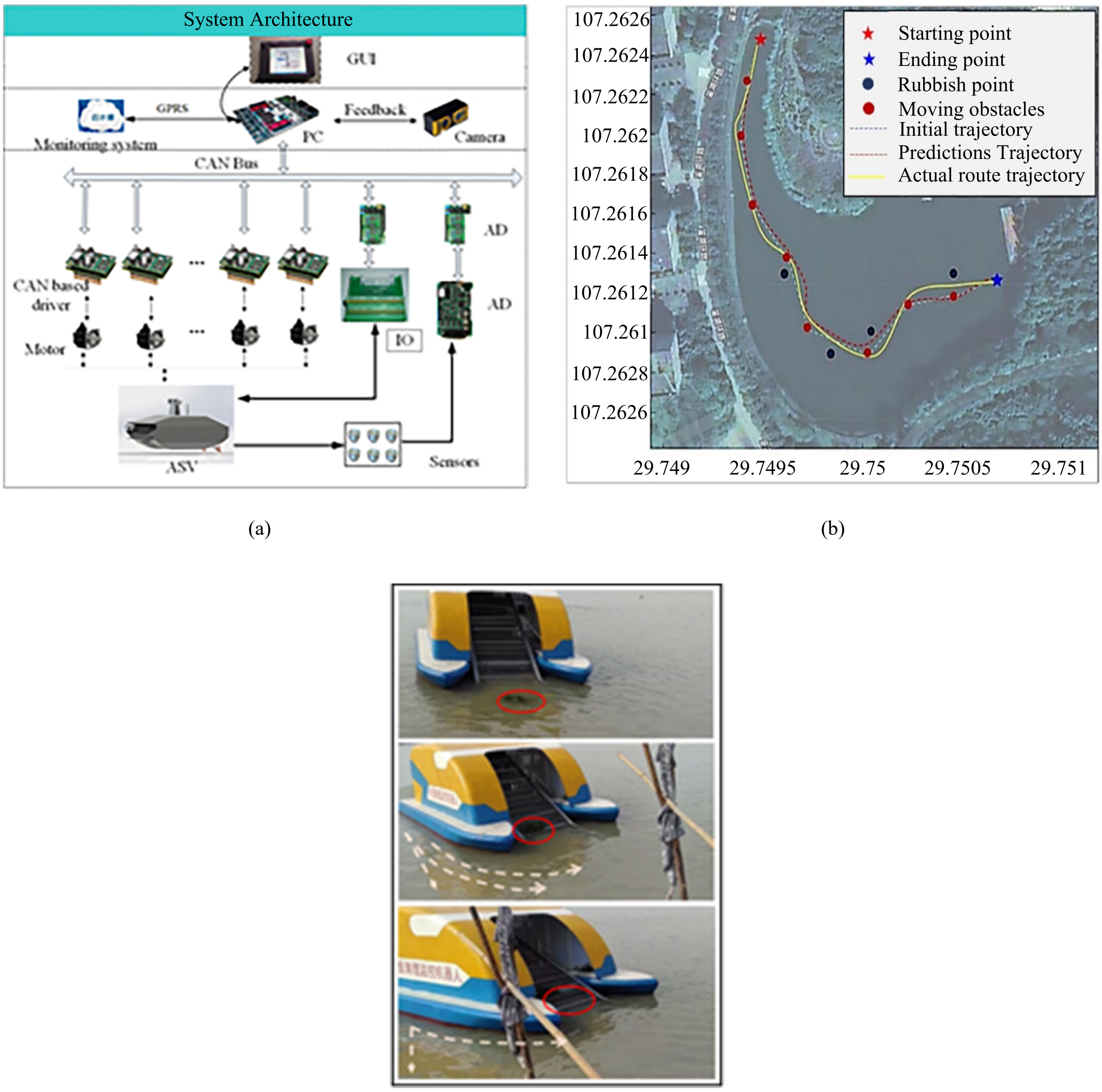


Figure 20. Depicts the segmented cleaning trajectory route of the ASV

When the Autonomous Surface Vehicle (ASV) approaches the location of a waterborne object based on its planned path, it captures the object using the target recognition module. Then, the ASV moves toward the object at a slightly reduced speed while using its visual system to capture the object and correct its path. Finally, it gets close to the target point for cleaning. This cleaning method helps to avoid external factors such as water surface disturbances caused by the high speed of the ASV, as well as errors in GPS positioning accuracy. As shown in Figure 21, the ASV successfully identifies and navigates around dynamic obstacles during the cleaning process.



(a)

(b)

(c)

Figure 21. Simulation experiment system, real simulated paths, and physical cleaning

To make the experimental water environment more realistic, movable obstacles were placed on the test site. When the ASV encountered a movable obstacle during its cleaning task, it used a combination of laser radar and cameras to identify the obstacle. It then employed the dynamic window algorithm to issue commands to the ASV, enabling it to navigate around the obstacle.

The experiment confirms the feasibility and potential for the development of the proposed method. However, it is important to note that the experiment only considered cleaning water surface debris under conditions of unstable water flow fluctuations, without considering other factors such as water resistance and wind speed. Further research and testing may be necessary to address these additional factors and real-world complexities.

6. Conclusions

The target environment and target points of traditional path planning methods for floating debris removal are mostly static, while the target points of floating debris removal operations in water surface will be uncertain due to the fluctuation of water flow. The proposed method in this paper can still complete the removal of floating debris in a dynamic water area and under a moving state, even when the floating debris is in a dynamic water area. This paper discusses the challenge of removing debris from water surfaces in dynamic water environments. It investigates experimental and optimization techniques using an improved fusion algorithm for path-planning. The study conducts simulation experiments under three different water flow fluctuation scenarios: static water environments, uniform water flow fluctuations, and non-periodic water flow fluctuations. Additionally, the paper carries out real-world Autonomous Surface Vehicle (ASV) prototype experiments in an actual aquatic environment. The results reveal that the algorithm is effective in non-periodic water flow fluctuation environments. It is also suitable for path-planning and cleaning multiple target water surface debris in complex settings.

Author contributions

Conceptualization, Ziyi Huang, Nana Chen and Yangtian Yan; methodology, Lin Zhang and Kai Yan; software, Kai Yan and Xianhua Zheng, Hankun Yin; validation, Lin Zhang and Xianhua Zheng; investigation, Ziyi Huang and Kaixin Xie; resources, Xin Chen; data curation, Ziyi Huang and Muhammad Saqib; writing-original draft preparation, Ziyi Huang, Nana Chen; writing-review and editing, Nana Chen and Xin Chen; supervision, Hankun Yin and Lin Zhang; project administration, Lin Zhang. All authors have read and agreed to the published version of the manuscript.

Data availability

The data used to support the findings of this study are included within the article.

Acknowledgement

This work was supported by China Postdoctoral Science Foundation (Grant No. 2023M742265). This work was also supported by National Natural Science Foundation of Chongqing (Grant No. 2023NSCQ-MSX0758), and Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJZD-K202301403).

Conflict of interest

The authors declare no conflict of interest.

References

[1] Satheesh J, Nair AP, Mahesh G, Jayasree PR. Wireless communication based water surface cleaning boat. In: *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*. IEEE; 2020. p.716-720. Available from: https://ieeexplore.ieee.org/document/9142960/.

[2] Phirke S, Patel A, Jani J. Design of an autonomous water cleaning bot. *Materials Today Proceedings*. 2021; 46: 8742-8747. Available from: https://linkinghub.elsevier.com/retrieve/pii/S2214785321028972.

[3] Procop I, Pacuraru F, Pacuaru S, Solea R, Cotoc G, Caramatescu A. Semi-autonomous system for lakes and rivers depollution. In: *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE; 2022. p.188-194. Available from: https://ieeexplore.ieee.org/document/9931869/.

[4] Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959; 1(1): 269-271. Available from: http://link.springer.com/10.1007/BF01386390.

[5] Hart P, Nilsson N, Raphael B. A formal basis for the heuristic determination of minimum cost paths. In: *IEEE Transactions on Systems Science and Cybernetics*. 1968; 4(2): 100-107. Available from: http://ieeexplore.ieee.org/document/4082128/.

[6] LaValle SM. Rapidly-exploring random trees: A new tool for path planning. *Computer Science Engineering*. 1998. Available from: https://api.semanticscholar.org/CorpusID:14744621.

[7] Kang JG, Jung JW. Post triangular rewiring method for shorter RRT robot path planning. *International Journal of Fuzzy Logic and Intelligent Systems*. 2021; 21(3): 213-221. Available from: https://www.ijfis.org/journal/view.html?uid=960&vmd=Full.

[8] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: An overview. *Soft Computing*. 2018; 22(2): 387-408. Available from: http://link.springer.com/10.1007/s00500-016-2474-6.

[9] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 1996; 26(1): 29-41. Available from: https://ieeexplore.ieee.org/document/484436/.

[10] Zhang Z, Wu D, Gu J, Li F. A path-planning strategy for unmanned surface vehicles based on an adaptive hybrid dynamic stepsize and target attractive force-RRT algorithm. *Journal of Marine Science and Engineering*. 2019; 7(5): 132. Available from: https://www.mdpi.com/2077-1312/7/5/132.

[11] Enevoldsen TT, Galeazzi R. Grounding-aware RRT∗ for path planning and safe navigation of marine crafts in confined waters. *IFAC-PapersOnLine*. 2021; 54(16): 195-201. Available from: https://linkinghub.elsevier.com/retrieve/pii/S2405896321014956.

[12] Lin Y, Zhang W, Mu C, Wang J. Application of improved RRT algorithm in unmanned surface vehicle path planning. In: *2022 34th Chinese Control and Decision Conference (CCDC)*. IEEE; 2022. p.4861-4865. Available from: https://ieeexplore.ieee.org/document/10034282/.

[13] Yifan1 Z, Guoyou1 S, Jiachen X. Path planning algorithm of unmanned surface vehicles based on Bi-RRT guided by artificial potential field. *Journal of Shanghai Maritime University*. 2022; 43(4): 16-22.

[14] Ma Y, Hu M, Yan X. Multi-objective path planning for unmanned surface vehicle with currents effects. *ISA Transactions*. 2018; 75: 137-156. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0019057818300478.

[15] Singh Y, Sharma S, Sutton R, Hatton D, Khan A. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. *Ocean Engineering*. 2018; 169: 187-201. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0029801818311193.

[16] Chen Z, Zhang Y, Zhang Y, Nie Y, Tang J, Zhu S. A hybrid path planning algorithm for unmanned surface vehicles in complex environment with dynamic obstacles. *IEEE Access*. 2019; 7: 126439-126449. Available from: https: //ieeexplore.ieee.org/document/8808914/.

[17] Zhang W, Shan L, Chang L, Dai Y. SVF-RRT*: A stream-based VF-RRT* for USVs path planning considering ocean currents. *IEEE Robotics and Automation Letters*. 2023; 8(4): 2413-2420. Available from: https://ieeexplore.ieee.org/document/10044915/.

[18] Xiu-xing H, Lu-qing S, Chuan F, Lin F, Zi-fan Z, Fan Q. Research on the model of complete coverage path planning for unmanned cleaning ship in small water area. *The Journal of New Industrialization*. 2019; 9(8): 92-106.

[19] Zhenkui H, Wenzhu S, Qiaoling DU, Tingting Y, Xinyu YAN, Dongrui WU. Studies on control system of small-scale float-garbage automatic cruise ship based on open-water traversal algorithm. *Journal of Jilin University (Information Science Edition)*. 2019; 37(2): 208-215.

[20] Xu PF, Ding YX, Luo JC. Complete coverage path planning of an unmanned surface vehicle based on a complete coverage neural network algorithm. *Journal of Marine Science and Engineering*. 2021; 9(11): 1163. Available from: https://www.mdpi.com/2077-1312/9/11/1163.

[21] Guo B, Kuang Z, Guan J, Hu M, Rao L, Sun X. An improved a-star algorithm for complete coverage path planning of unmanned ships. *International Journal of Pattern Recognition and Artificial Intelligence*. 2022; 36(03): 2259009. Available from: https://www.worldscientific.com/doi/abs/10.1142/S0218001422590091.

[22] Deng T, Xu X, Ding Z, Xiao X, Zhu M, Peng K. Automatic collaborative water surface coverage and cleaning strategy of UAV and USVs. *Digital Communications and Networks*. 2022. Available from: https://linkinghub.elsevier.com/retrieve/pii/S2352864822002826.

[23] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers; 1985. p.500-505. Available from: http://ieeexplore.ieee.org/document/1087247/.

[24] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*. 1997; 4(1): 23-33. Available from: http://ieeexplore.ieee.org/document/580977/.