

Article

A Predictive Guidance Obstacle Avoidance Algorithm for AUV in Unknown Environments

Juan Li ^{1,2} , Jianxin Zhang ², Honghan Zhang ^{2,*} and Zheping Yan ²

¹ Science and Technology on Underwater Vehicle Technology, Harbin Engineering University, Harbin 150001, China

² College of Automation, Harbin Engineering University, Harbin 150001, China

* Correspondence: oceanzh@hrbeu.edu.cn; Tel.: +86-0451-8252-8261

Received: 18 May 2019; Accepted: 25 June 2019; Published: 27 June 2019



Abstract: A predictive guidance obstacle avoidance algorithm (PGOA) in unknown environments is proposed for autonomous underwater vehicle (AUV) that must adapt to multiple complex obstacle environments. Using the environmental information collected by the Forward-looking Sonar (FLS), the obstacle boundary is simplified by the convex algorithm and Bessel interpolation. Combining the predictive control secondary optimization function and the obstacle avoidance weight function, the predicting obstacle avoidance trajectory parameters are obtained. According to different types of obstacle environments, the corresponding obstacle avoidance rules are formulated. Lastly, combining with the obstacle avoidance parameters and rules, the AUV's predicting obstacle avoidance trajectory point is obtained. Then AUV can successfully achieve obstacle avoidance using the guidance algorithm. The simulation results show that the PGOA algorithm can better predict the trajectory point of the obstacle avoidance path of AUV, and the secondary optimization function can successfully achieve collision avoidance for different complex obstacle environments. Lastly, comparing the execution efficiency and cost of different algorithms, which deal with various complex obstacle environments, simulation experiment results indicate the high efficiency and great adaptability of the proposed algorithm.

Keywords: autonomous underwater vehicle; forward-looking sonar; predictive control; line-of-sight guidance; obstacle avoidance algorithm

1. Introduction

Autonomous underwater vehicle (AUV) [1] is an important tool for marine resource exploitation and marine scientific research [2–4]. As more research interests turn to the cooperative target search, many factors should be considered, such as the environment information, target states, etc. In reality, the working environment for AUV is often unknown. In different types of obstacles, static or dynamic environments are encountered. Therefore, an effective obstacle avoidance algorithm is needed.

In recent years, significant contributions have been made by many researchers in developing obstacle avoidance methods and applying them to various obstacle avoidance environments for AUV. These problems can be classified into the global obstacle avoidance path planning problems and local obstacle avoidance methods. When the global environmental information including various obstacles are known, the global obstacle avoidance path planning problem becomes a nonlinear optimal programming problem to find global optimal solutions on the pre-requisite that global variables are known. For example, two missile guidance algorithms are proposed for the intercept and for the rendezvous of a maneuvering target while avoiding a static obstacle by a specified avoidance distance [5]. However, the method simply solves the secondary optimization solution for static obstacles without considering the obstacle type and the impact of dynamic obstacles on the obstacle avoidance

process [6,7]. According to the grid-based artificial potential field method, to solve the multi-mobile vehicle cooperative obstacle avoidance problem, it requires an effective method for managing general convex obstacles. However, it is easy to fall into local minimum value points when dealing with concave obstacles and failing to escape from concave obstacles. At the same time, the obstacle avoidance problem of the path planning is different from the calculation of obstacle avoidance trajectory points based on the real-time detection of obstacles. Control optimization problems for obstacle avoidance include non-linear programming [8,9], heuristic algorithm [10,11], and the graph search method including the A* algorithm [12–14] and D* algorithm [15]. In addition, if an AUV works in a locally known but globally unknown environment with various types of obstacles, nonlinear methods are needed to plan out the AUV trajectory points to ensure the safety of AUV in missions. To solve this problem, there is an artificial potential field method (APF) [16–18], as well as evolutionary algorithms such as the genetic algorithm (GA) [19–21] and the particle swarm optimization algorithm (PSO) [22,23]. Compared with traditional optimization methods, these algorithms usually lead to global optimal solutions, or approaches that are close to global optimal solutions. However, these evolutionary algorithms may cause poor numerical precision and difficult execution, when they solve the nonlinear optimal problem. At the same time, the iterative period of these algorithms is long. If not optimized, they will easily fall into the local minimum value.

Some significant achievements have been obtained in the obstacle avoidance problem. Masoud, Dadgar, et al. proposed an A-RPSO (Adaptive Robot PSO) algorithm [24], which considered the obstacle avoidance problem of the robot performing tasks, and, in the obstacle avoidance, there was also a control mechanism that escapes from the local optimum. In Reference [25], research studies are conducted on analyzing different trajectories presented by dynamic obstacles in the environment to predict their future positions and to realize obstacle avoidance. By estimating future areas where collisions between robots and obstacles may occur, mobile robots can take corrective actions before collisions. The Montegrey, Calif. AUV Research Naval Postgraduate Institute conducted an experimental study on underwater reactive obstacle avoidance (OA) for AUVs, by mainly focusing on using the ARIES AUV and Blueview Blazed Array FLS for obstacle detection and avoidance [26]. In References [27,28], an improved line-of-sight (LOS) guidance algorithm is used for obstacle avoidance. At the same time, in the process of underwater obstacle avoidance, to achieve the optimal path obstacle avoidance [29,30], the path planning method was used. For example, to achieve path smoothness, Joono, Sur. et al. [31] used a streamline-based autonomous underwater vehicle obstacle avoidance path planning method. Yufei, Zhuang, et al. [32] combined the particle swarm optimization (PSO) algorithm with the Legendre pseudo-spectral method (LPM), which achieved real-time collision avoidance of static obstacles and moving obstacles with different levels of positional uncertainty. Zheping, Y. et al. only classified the obstacles in the obstacle-avoiding environment, and did not consider the influence of the unevenness of the obstacles on obstacle avoidance. At the same time, the real-time obstacle avoidance method mentioned for the obstacle with uneven surface needs to calculate the turning radius when a large number of obstacle avoidance positions are calculated. In the case of multi-obstacle distribution, the obstacle data of the overlapping portion is not processed, but it is unreasonable to directly propose an obstacle avoidance gap for the overlapping obstacles as the direction of the obstacle avoidance. There is no comprehensive consideration of multiple factors affecting AUV obstacle avoidance. However, the previously mentioned methods are developed either for specific cases where different numbers of obstacles are distributed in the environment, or aimed at the obstacle and the AUV relative model to obtain the optimal solution in order to achieve AUV obstacle avoidance. No solution regarding complex irregular obstacles or obstacles of different types is proposed, and no optimization obstacle avoidance strategy is suggested for completely unknown environments.

Based on the previously mentioned obstacle avoidance problems, and combining target search and tracking in unknown underwater environments with complex irregular obstacles, an obstacle avoidance method for AUV based on PGOA is proposed. The main idea of this algorithm is: The FLS (Forward-looking sonar) is equipped in the AUV front to obtain the obstacle information. Obstacles

are classified according to the detected character information. Then, the contour convex algorithm and Bezier interpolation are used to change the irregular contour into a convex polygon, which simplifies the boundaries of the irregular complex obstacle, so that the AUV can respond quickly to various obstacle environments. In the second stage, by using the maximum turning radius calculation method based on the obstacle type and boundary data, different turning radii are obtained. Then, by using obstacle avoidance parameters and obstacle avoidance rules, the obstacle avoidance weight function is established. Lastly, the prediction parameters are obtained based on the predictive control second optimization function. In the final stage, the corresponding predictive guidance track points are calculated for the parameters, which are obtained by the obstacle avoidance rules and weight functions of different obstacle environment types. Combining with the guidance [33,34] method, the AUV successfully avoids obstacles and gets to the position of an intended target by traveling over the predictive track. By using the obstacle avoidance method proposed, the AUV obstacle avoidance in an unknown underwater environment is successfully achieved. Experiments compared the work efficiency and task completion of AUV using APF and PSO algorithms, respectively. Lastly, it is proven via experimental data that, compared with the traditional search methods and conventional optimization algorithms mentioned above, the PGOA proposed has the clear advantage. Moreover, the simulation experiment fully demonstrates the adaptive characteristics of AUV in different environments with complex obstacles.

The rest of this paper is organized as follows. Section 2 introduces problem description and mathematical modeling. Section 3 presents environments where obstacles distribute differently. Section 4 develops AUV PGOA design. Section 5 gives a discussion of the results of experimental data in various situations. Lastly, conclusions are given in Section 6.

2. Problem Statement and Model Description

2.1. Problem Description

In unknown underwater environments, AUV may encounter complex terrain such as reefs, islands, trenches, and valleys, and the obstacles they meet are different in shape, complexity, and quantity. While ensuring the completion of underwater missions such as target search and intelligence reconnaissance, guaranteeing its own safety, it is necessary that AUV has the ability to avoid these complex obstacles in each situation. Therefore, it is important to analyze the factors that may affect cooperative searching for multi-AUVs.

1. Visual noise and threshold: Through the AUV-configured forward-looking sonar, the observation data is affected by Gaussian noise during the measurement of the observation target. In addition, AUV observation of the unknown environment through the forward-looking sonar is also limited by the sensor's detection distance. As such, the sonar cannot observe and extract environmental features that are outside the view.

2. Movement limitation: The AUV's own motion state will be affected by equipment such as thrusters and rudders. In an unknown underwater environment, the AUV will also be affected by unknown factors such as ocean currents and submarine topography. Therefore, the influence of the AUV motion restriction features on the track planning during the AUV navigation process need to be considered.

3. Obstacle types: When AUV performs tasks in an unknown underwater environment, it encounters a variety of obstacle types, such as simple convex obstacles, complex convex obstacles, and complex vortex obstacles. Therefore, a single obstacle avoidance algorithm cannot solve sudden problems. Therefore, different obstacle avoidance strategies and algorithms are developed in this paper for different obstacle types appearing in the AUV field of view, so as to achieve the optimal obstacle avoidance effect.

4. Obstacle avoidance: AUVs may operate in an unknown underwater environment, and it is inevitable for AUVs to encounter obstacles, which may threaten the AUV's normal trajectory in the

process of performing target searching tasks. Therefore, the AUV is expected to have the ability to avoid obstacles in a timely manner, to ensure that AUV can travel safely and reliably. Ultimately, the economic cost caused by equipment damage can be avoided and the searching task can be reliably completed.

2.2. AUV Movement Model

The AUV four-degrees-of-freedom constant-speed motion model $x_{t+1} = f(x_t)$ was established to describe the form of AUV motion under water. In this paper, the updating of velocity and position follows the following formula.

In this paper, according to the standard AUV equation of motion [35], the updating of velocity and position follows the formula below.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R(\Theta) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (1)$$

$$R(\Theta) = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2)$$

Because the roll movement is uncontrollable for AUV and the simulation environment of the obstacle avoidance algorithm is built on the 2-D environment, defining $\phi, \theta = 0$. Considering that the AUV's additional hydrodynamic resistances in the horizontal and vertical direction are greater than those in the longitudinal one, when the speed over ground (SOG) exceeds 1 knot, the propulsive efficiency of the auxiliary thrusters is very low. Therefore, when the AUV is navigating at a normal speed, its auxiliary thrusters are idle. Usually, we take $w = 0, v = 0$, and then Equations (1) and (2) are simplified as [36]:

$$\begin{cases} u = v_c \\ \dot{x} = u \cos(\psi) \\ \dot{y} = u \sin(\psi) \\ \dot{\psi} = r \\ x(t+1) = x(t) + \dot{x} \\ y(t+1) = y(t) + \dot{y} \end{cases} \quad (3)$$

where $(x(t), y(t))$ represents the positioning information that is related to the time variable t , (\dot{x}, \dot{y}) represents the velocity vector of the AUV in the global coordinate system, v_c is a normal constant, and ψ represents the angle between the AUV direction and the direction of the global coordinate system axis, and r is a heading velocity variable in this case. AUV is affected by its own equipment, and its speed v_c and corner ψ are limited.

2.3. Forward-Looking Sonar Model

Based on the unknown underwater environment, the real multi-beam active forward-looking sonar data is simulated through a mathematical model of forward-looking sonar [37] in this paper. According to the common multi-beam sonar background, the Seabeat6012 sonar is selected as the forward-looking sonar of the AUV. The Seabeat6012 sonar has a visible range R of 150 m, a horizontal opening angle α of 120° , a vertical opening angle β of 15° , and an operating frequency of 2 kHz. To obtain the target data, the sonar model performs the array statistics $d \in \mathbf{R}^{80 \times 3}$ of the range of the sonar opening angle. Then based on the elements filled in the matrix, it judges whether there is a target in a certain position in the visible range. The FLS can be roughly represented, as shown in Figure 1.

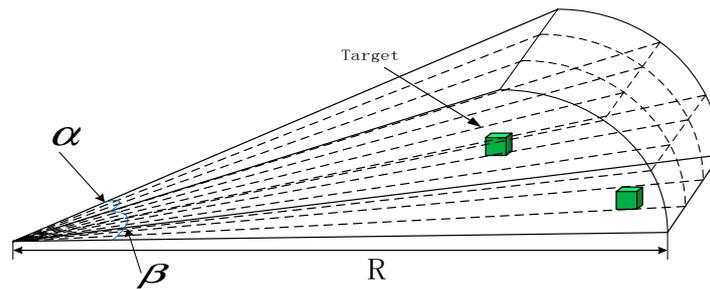


Figure 1. FLS model.

The mathematical model for FLS is established, which can describe the constraint relationship between the target and the FLS. Then the data information of the detected object is obtained. The model can be given by the equation below.

$$\begin{cases} \frac{|y_t|}{\sqrt{x_t^2 + y_t^2}} \leq \sin \frac{\alpha}{2} \\ \sqrt{x_t^2 + y_t^2 + z_t^2} \leq R \\ \frac{|z_t|}{\sqrt{x_t^2 + y_t^2}} \leq \sin \frac{\beta}{2} \end{cases} \quad (4)$$

where (x_t, y_t, z_t) can be expressed by the equation below.

$$\begin{cases} x_t = x - x_0 \\ y_t = y - y_0 \\ z_t = z - z_0 \end{cases} \quad (5)$$

where (x, y, z) is the coordinate of the target in the hull coordinate system $(Ox_0y_0z_0)$. (x_0, y_0, z_0) is the body coordinate of the AUV sonar. (x_t, y_t, z_t) denotes the relative positional relationship between the target and the AUV. By judging the position and angle relationship of (x_t, y_t, z_t) , this model is determined whether the target is in the sonar field.

Because the forward-looking sonar equipped on the AUV can be easily affected by the medium of water or other external factors during the data collection process, such as data interference, the measurement of the environmental characteristics is likely to be affected. Therefore, the description of the sonar is given in Equation (6) [38].

$$y_{x-q} = \begin{cases} none & |x - q| > L \\ none & |x - q| \text{ in Obstacles} \\ h(x, q) + d(x, q)\zeta & |x - q| < L \end{cases} \quad (6)$$

where y_{x-q} denotes the FLS measurement from an AUV at position x to a sensing point at position q . *none* indicates that the environmental feature data does not exist. L is the visual threshold. $h(x, q)$ is the sensor model in the noise free case, $d(x, q)$ is the distance between x and q , and ζ is standard Gaussian noise.

The above description indicates that, when the relative position of the FLS and the detected object is outside the sight range, or there is an obstacle between the sonar and the detected object, obtaining feedback of characteristic information is unavailable. If there is characteristic information in the sight range, the observed data indicates the distance increases, and the measured disturbance increases, accordingly.

3. Classification of Obstacles and Solutions

3.1. Type of Obstacles

For a wide expanse of unknown waters, when AUVs are dispatched to perform underwater operations, they often encounter unpredictable harsh environments with obstacles everywhere, which affects the normal movement of AUVs. For example, Figure 2 lists several typical environments of obstacle distributions: environments of convex obstacles, vortex obstacles, and dense convex obstacles.

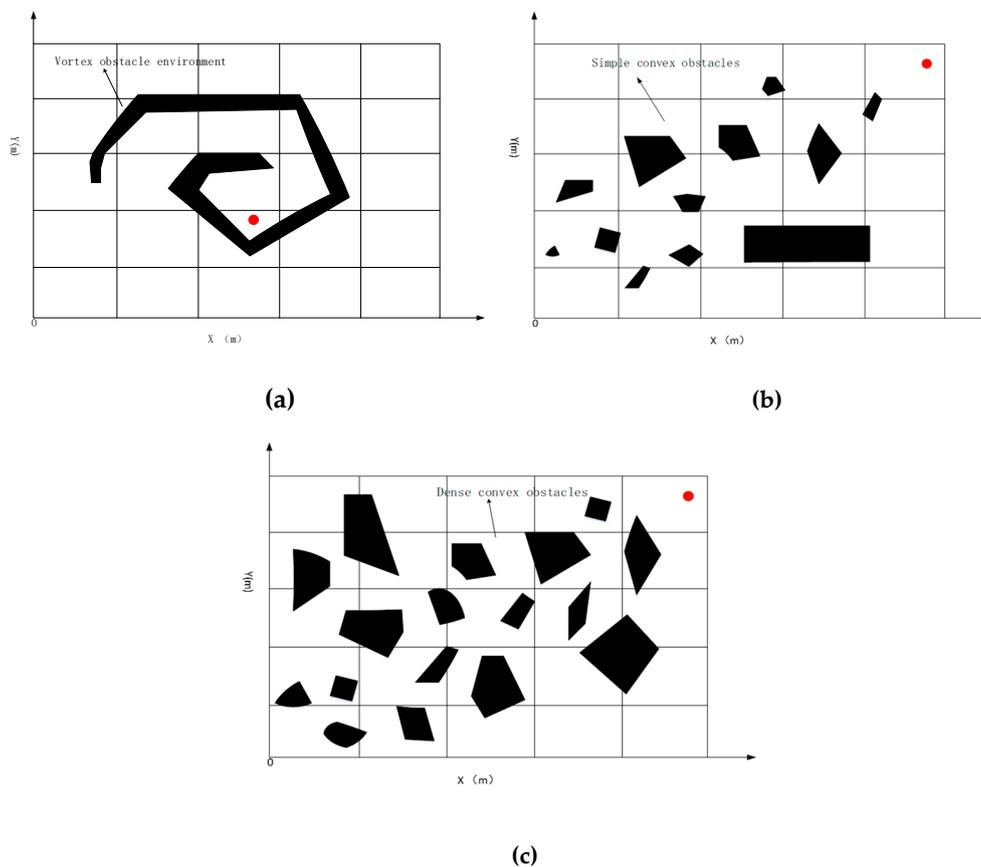


Figure 2. Different types of obstacles. (a) No-Convex Obstacles. (b) Simple convex obstacle environment. (c) Dense Convex Obstacles.

(1) Convex Obstacles

In this paper, the sporadic distribution of a small number of convex obstacles is defined as a simple convex obstacle environment type in an unknown environment in Figure 2b. In this situation, AUV could easily avoid collisions and plan a safe and reasonable trajectory to complete the mission successfully.

(2) No-Convex Obstacles

In some cases, AUV can encounter some non-convex obstacles, such as concave structures or helical shape. In such a case, a simple obstacle avoidance algorithm is not enough for AUVs escaping from obstacles in Figure 2a. When an AUV is stuck in such an obstacle, a corresponding escape algorithm needs to be developed to escape the vortex obstacles.

(3) Dense Convex Obstacles

Part 2.1(3) describes the types of obstacle structures. In reality, there is also an environment of densely distributed convex obstacles in Figure 2c, which demands higher obstacle avoidance abilities for AUV. Therefore, in intensive obstacle environments, a safe trajectory needs to be planned for AUV to safely navigate through the densely distributed obstacles region.

Figure 2 shows specific characteristics and distributes for each type obstacle in unknown environments. The red spot is a virtual target point, which indicates that AUV has completed its task. To verify the effectiveness of the algorithm, an unknown environment with an area of $2000\text{ m} \times 2000\text{ m}$ is designed, which is divided into $400\text{ m} \times 400\text{ m}$ task sub-areas, which supposes that the obstacle information is unknown for the AUV. For the three types of obstacles mentioned above, different obstacle avoidance methods are designed for AUV to complete its task and reach the desired target point safely. The following solutions are developed to three situations.

3.2. Obstacle Detection Principles

AUV uses FLS to realize the collision avoidance function. A real-time obstacle avoidance strategy is proposed in this paper. All obstacles in the environment are unknown, and their shapes and positions are randomly generated. The obstacle boundary is generated based on the detected information of the FLS. Considering that the AUV pitch angle rarely changes, the multi-beams sector on the horizon in the body coordinates is used. The purple lines are the sonar beams on the horizontal plane in the AUV body coordinates. The gray part is the obstacle, and the blue boundary is the obstacle contour curve (Figure 3).

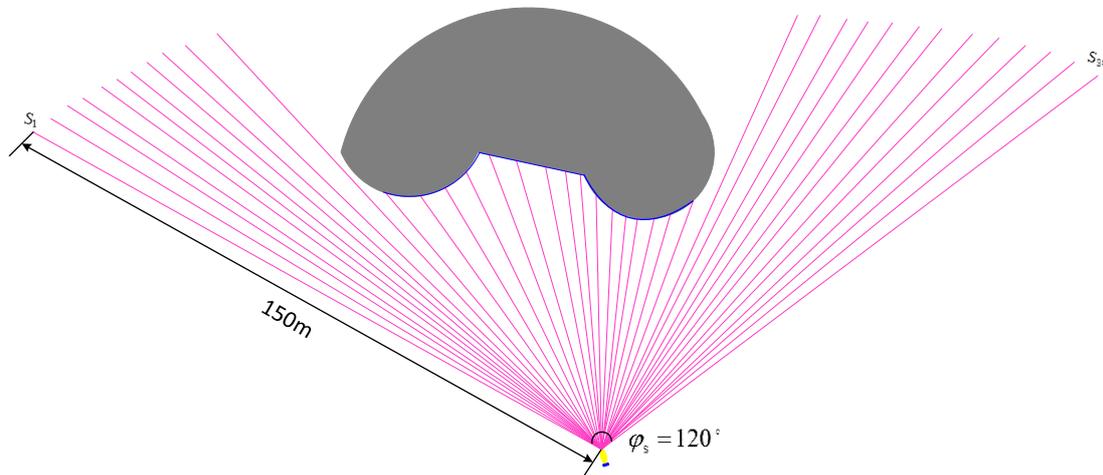


Figure 3. Obstacle detected by a Forward-looking sonar.

3.3. Obstacle Condition Classification

To improve the effectiveness and safety of the AUV in obstacle avoidance when performing underwater tasks, the obstacles are divided into four conditions according to the positions of obstacles relative to AUV: bounded obstacles, left bounded obstacles, right bounded obstacles, and left and right edge unbounded obstacles. When the obstacle enters the FLS detection range, the sonar will classify the obstacle based on detected data by emitting 80 beams. The detection zone is a fan-shaped range of 80 m, as shown in Figure 4. In addition, k and l is the left boundary and right boundary detected by FLS, respectively, i, j are the serial numbers of beams, δ, ζ is the arbitrary nature number [39].

(1) If the boundary of the obstacle is in the beam range of the FLS, the current obstacle is considered as the bounded obstacle (BO).

$$\exists k, l, \delta, \zeta \in \mathbb{Z}, k < l, i, j \in \mathbb{N}^+, i \in [k, l], j \in [k - \zeta, k - 1] \cup [l + 1, l + \delta], st. 0 < S_i \leq L_e, st. 0 < S_j \leq L_e$$

(2) If the right boundary of the obstacle is outside the sonar beam range, and the left boundary is in the range, the detected obstacle is called the left bounded obstacle (LB).

$$\exists k, \delta \in \mathbb{Z}, i \in [1, k], j \in [k, k + \delta], st. 0 < S_j \leq L_e, S_i = 0$$

(3) If the left boundary of the obstacle is outside the detection range of the sonar, and the right boundary is in the range, the obstacle is defined as the right bounded obstacle (RBO).

$$\exists l, \zeta \in Z, i \in [l - \zeta, l], j \in [l, 80], st.0 < S_i \leq L_e, S_j = 0$$

(4) If both sides of the obstacle are outside the detection range of the sonar, the obstacle is defined as an unbounded obstacle (UBO).

$$\exists i, j \in [1, 80], st.0 < S_i \leq L_e, st.0 < S_j \leq L_e$$

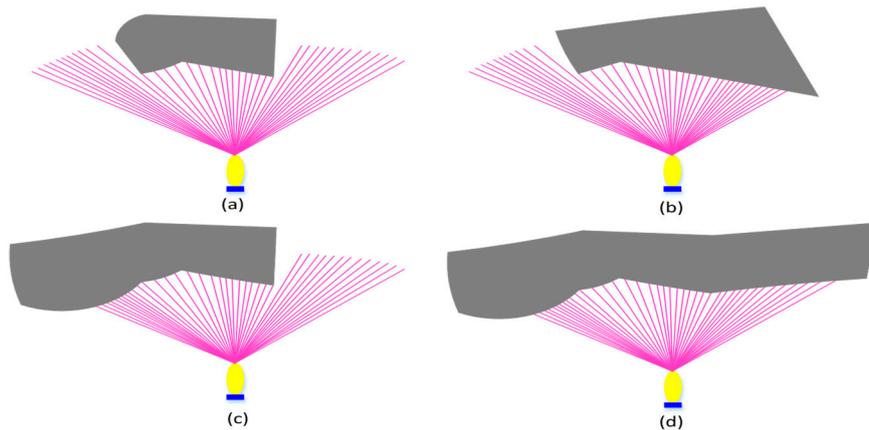


Figure 4. Obstacle classification. (a) Bounded obstacle. (b) Left bounded obstacle. (c) Right bounded obstacle. (d) Unbounded obstacle.

3.4. Obstacle Avoidance Boundary Data Processing

First, the obstacle information in the sight range is obtained using FLS of AUV. The detected data is stored in the matrix $\beta \in R^{80 \times 3}$, and only the horizontal plane of the sonar opening angle is used in this paper, so just the second column of data of $\gamma \in \beta$ is needed, with each representing the object distance and angle information detected by the sonar beam. If there are elements in γ equal to zero, it means that the sonar beams do not detect any objects. Group data is considered an obstacle. For example, the detected data can be divided into two groups in Figure 5.

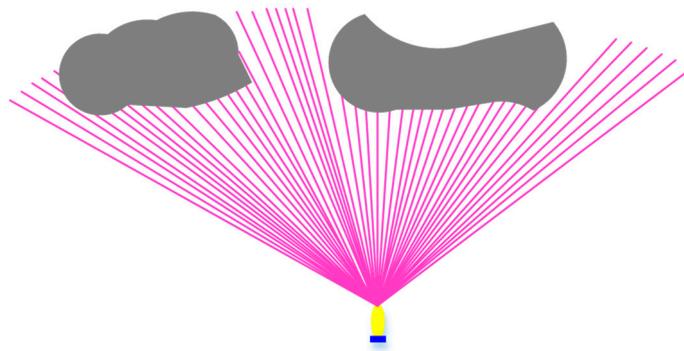


Figure 5. Example of FLS detecting data grouped.

Data in γ is grouped based on Formulas (10) and (11). First, the proper sonar beam spacing is selected based on the FLS type. Then based on Equation (10), the obstacle information is judged

whether it is a continuous sonar beam data, and whether the 2-norm of the obstacle boundary point satisfies the beam spacing condition. The obstacle data can be grouped by iterating boundary data.

$$\|\overline{S_i S_{i-1}}\| < d_b, S_i \cdot S_{i-1} \neq 0, i \in [2, 80] \quad (7)$$

$$d_b = \frac{\lambda_t (l_e \cdot \varphi_s)}{n-1}, \quad (n = 80, \lambda_t \in [1, 4]) \quad (8)$$

where d_b is the beam spacing. S_i is the obstacle point detected. λ_t is the screening factor, and l_e is the detection range of the sonar.

FLS can output a data structure for the detected obstacles, where the obstacle boundary data are stored in the same array. In reality, the obstacle shapes are always irregular, so the output data from the sonar cannot be directly used to avoid obstacles. In this paper, the convex hull algorithm combined with the simplified Bezier interpolation algorithm is used to transform the obstacle outline into a regular shape, and the data is smoothed. It not only improves the accuracy of obstacle avoidance, but also avoids the adverse effects of complex obstacles. In Figure 6, when the sonar detects a bounded obstacle, to improve the accuracy of the obstacle avoidance efficiency, it is necessary to simplify and smooth the boundary of the obstacle. By using the horizontal plane layer data $\beta \in R_2^{80 \times 2}$, coordinate transformation is performed on the distance and angle data from the sonar beam point, since the input data streams into the convex algorithm. When the data is simplified, it can be smoothed using Bezier interpolation. To improve the algorithm speed, this paper selects 80 variables to segment the smooth data. The specific implementation process is as follows.

Step 1: The data output from the sonar beam line is solved as an obstacle boundary point.

$$\begin{cases} O_{ix} = a_{uvx} + d_i \cos(\psi - \theta_i) \\ O_{iy} = a_{uvy} + d_i \sin(\psi - \theta_i) \end{cases}, \quad (\theta_i \in [\psi, \psi + \pi/3], i \in [k, l], k, l \in N^+) \quad (9)$$

$$\begin{cases} O_{ix} = a_{uvx} + d_i \cos(\psi + \theta_i) \\ O_{iy} = a_{uvy} + d_i \sin(\psi + \theta_i) \end{cases}, \quad (\theta_i \in [\psi - \pi/3, \psi], i \in [k, l], k, l \in N^+) \quad (10)$$

where ψ is the current heading of the AUV. d_i denotes the distance of the i -th sonar beam output. θ_i is the angle of the i -th beam output, and a_{ix}, a_{iy} is the position of the current AUV.

Step 2: For the set X of the obstacle boundary points calculated by the solution, a convex hull set, which includes all the points in the X set, needs to be found to replace the X set. Then we select the leftmost and lowest point in the set X as the origin of the polar coordinates. Then we sequence all the points in the set, based on the principle that their distance from the polar origin is short to long and the polar angle ranges from small to large. Subsequently, the ray is selected, which is generated by the second point after sorting and the polar coordinates. Then the vector angle from small to large is found by using the vector cross method. The convex hull is found, according to the distance from small to large, if the angles are the same.

Step 3: The convex hull set, calculated in step 2, represents a simplified set of contour points of the obstacle that are used as boundary points of the Bessel interpolation to generate the smooth obstacle boundary points. The specific calculation is shown below.

$$B(t) = \sum_{i=0}^n C_i^n P_i (1-t)^{n-i} t^i \quad t \in [0, 1] \quad (11)$$

where P_i is the interpolation point $P_i = (x_i, y_i)$, and the cubic Bessel interpolation function is used in this paper.

$$\begin{aligned} B_3(t) &= \sum_{i=0}^3 C_i^3 P_i (1-t)^{3-i} t^i \\ &= (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3 \quad t \in [0, 1] \end{aligned} \quad (12)$$

Step 4: The last processed boundary points are sorted from left to right in the order of the sonar beam. The final result is shown in the following figure. The red line is the contour line processed by the convex algorithm, and the blue line is the final processing result.

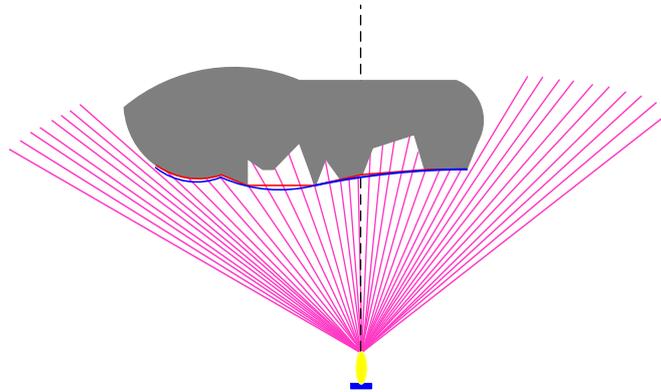


Figure 6. Obstacle surface effect after using the convex algorithm and Bessel interpolation.

4. Predictive Guidance Obstacle Avoidance Algorithm Design

4.1. AUV Maximum Obstacle Avoidance Turning Radius

When AUV navigates at 2 m/s in an underwater environment without ocean currents, and the rudder angle is set at a maximum steering angle of 35° , the minimum turning radius is about five times the length of the AUV. It takes approximately 3.5 seconds for the rudder angle of the AUV to vary from 0° to 35° . If the time delay of the steering angle transition is considered, the trajectory deviation distance is 1–1.5 m, which is small compared to the turning radius. To simplify the problem that the deviation distance is neglected. In other words, the trajectory deviation rotation is replaced by an arc with a certain radius.

In Figure 7, D_i is the best obstacle avoidance point detected by the FLS. Based on the current AUV state information and the obstacle data collected by the FLS, the angle and distance information (α_i, ρ_i) relative to the AUV of the current obstacle point can be known, where α_i is the angle between the AUV heading and the obstacle point detected by the sonar, and ρ_i is the distance between the obstacle point and the current AUV. R_i denotes the maximum turning radius relative to the current AUV. o is the center of the circle where the maximum conversion radius is located. ob is the mid-perpendicular line of Ac , and D_s is the safety distance. The specific definition is as follows:

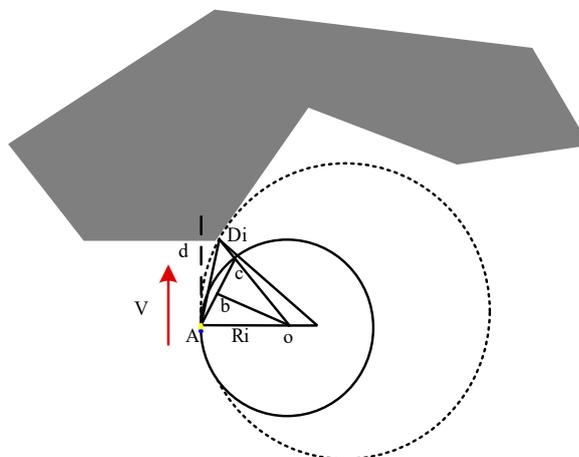


Figure 7. Expected maximum turning radius of the AUV obstacle avoidance point.

$$\angle dAD_i = i\alpha_i, \angle dAc = \beta_i, \overline{AD_i} = \rho_i, \overline{D_i c} = D_s, \overline{Ao} = R_i.$$

The maximum turning radius is expressed as follows [39].

$$\frac{\overline{AD_i}}{\sin(\angle D_i c A)} = \frac{\overline{D_i c}}{\sin(\angle D_i A c)} \quad (13)$$

$$\begin{cases} \angle D_i A c = \beta_i - \alpha_i \\ \angle D_i c A = \pi - (\pi/2 - \beta_i) = \pi/2 + \beta_i \end{cases} \quad (14)$$

Based on Equations (16) and (17):

$$\begin{cases} \alpha_i = \varphi_s(i - 0.5n + 0.5)/n \\ \beta_i = \frac{D_s}{\rho_i} \sec \alpha_i + \tan \alpha_i \end{cases} \quad (15)$$

$$\frac{\overline{AD_i}}{\sin(\angle D_i o A)} = \frac{\overline{oA}}{\sin(\angle o D_i A)} \quad (16)$$

The maximum turning radius obtained by Equations (18) and (19) is as follows:

$$R_i = \rho_i [\cos \alpha_i - \sin \alpha_i / \tan(2\beta_i)] \quad (17)$$

If AUV avoids obstacles by rounding the right edge of the obstacle, then R_i denotes the maximum turning radius of the obstacle point detected by the i -th sonar beam. The expected maximum turning radius is expressed as follows.

$$R_{\max} = \min\{R_i | i = 41, 42, \dots, 80\} \quad (18)$$

Otherwise, when the AUV avoids the obstacle around the left edge of the obstacle, the expected maximum turning radius is expressed as follows.

$$R_{\max} = \min\{R_i | i = 1, 2, \dots, 40\} \quad (19)$$

4.2. AUV Obstacle Avoidance Rules

Safe obstacle avoidance distance and the emergency obstacle avoidance distance is designed in the AUV obstacle avoidance process. Therefore, the following rules are formulated to deal with the two existing situations.

Rule 1: When there is a safe obstacle avoidance distance

a. If it is a left bounded obstacle that meets the conditions for safe obstacle avoidance, the AUV turns to the left side of the obstacle to avoid it.

b. If it is a right bounded obstacle that meets the conditions for safe obstacle avoidance, the AUV turns to the right side of the obstacle to avoid it.

c. If it is a bounded obstacle or an unbounded obstacle that meets the safe obstacle avoidance distance, the AUV turns in the direction close to the virtual target to avoid the obstacle.

Rule 2: When an emergency obstacle avoidance situation occurs, a corresponding rule needs to be established to ensure the safety of the AUV. To ensure absolute safety, in practice, it is necessary to immediately turn off the propeller and initiate reverse propulsion in order to offset the forward speed caused by inertia. However, this extreme situation is not considered in this article.

a. If there is an emergency obstacle avoidance setting where a bounded obstacle or an unbounded obstacle exists, the current AUV heading is taken as the dividing line to estimate the boundary point of the obstacle that is closer to the virtual target. Then based on the boundary point data generated,

we use the method mentioned above to calculate the maximum turning radius. Lastly, the minimum turning radius that meets the safe obstacle avoidance distance is selected.

$$\begin{cases} \lambda_r > \lambda_l & \text{choose left} \\ \lambda_r \leq \lambda_l & \text{choose right} \end{cases} \quad (20)$$

$$\begin{cases} D_v \sin(\lambda_r) - 2\text{sgn}(R_{\min})R_{\min}^\delta > D_v \sin(\lambda_l), & \text{turn left} \\ \text{else,} & \text{turn right} \end{cases} \quad (21)$$

where D_v is the distance of the AUV from the virtual target. R_{\min} is the minimum radius of the radius, which meets a maximum turning radius. λ_r, λ_l is the right and left angle of the obstacle avoidance path away from the target, respectively. δ is the set constant influence factor.

b. If there is an emergency obstacle avoidance and the bounded obstacle is in the field of view of the sonar, in order to perform a safe and most energy-efficient obstacle avoidance, the AUV sails toward the bounded side while calculating the turning radius.

The obstacle avoidance rule flow chart is shown in Figure 8.

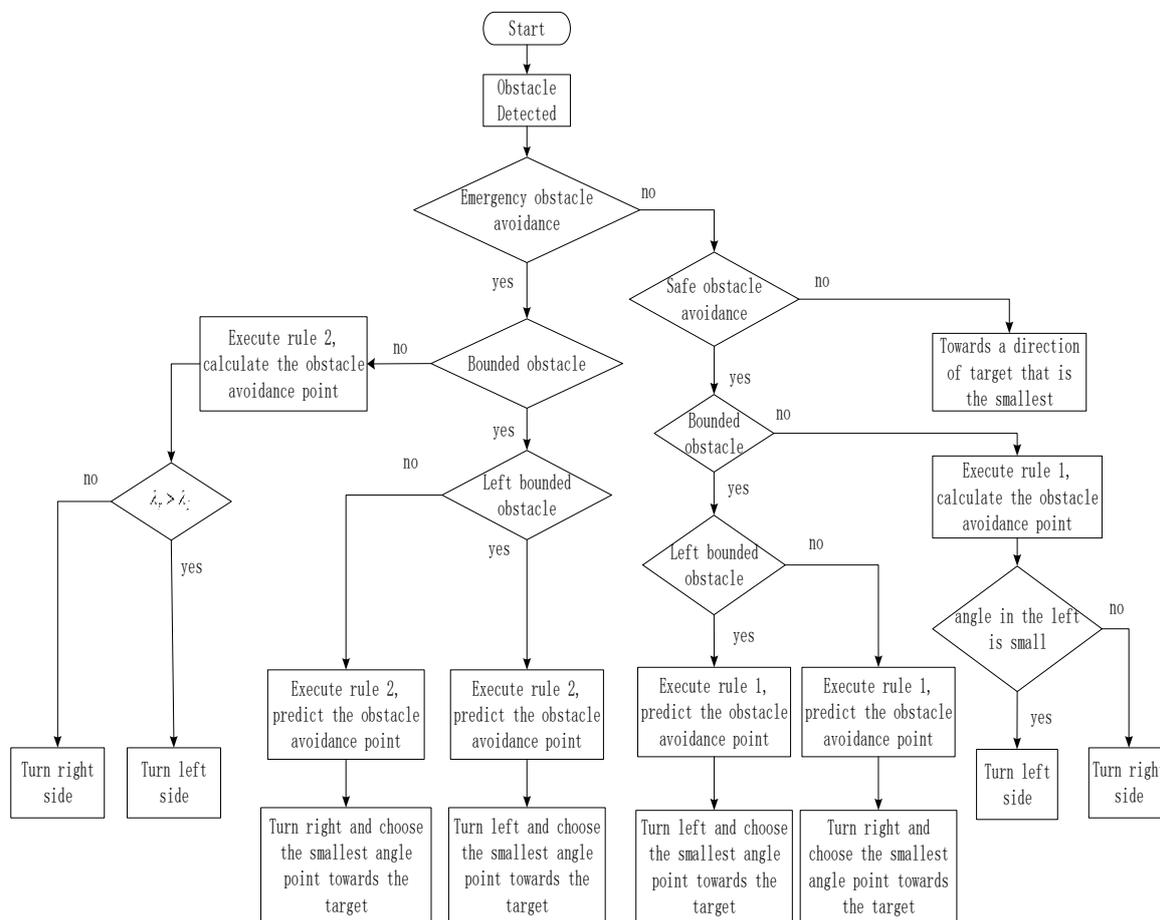


Figure 8. The obstacle avoidance rule shown by a flowchart.

4.3. Constructing the Weighting Function of the Obstacle Avoidance Algorithm

4.3.1. Weight Function for Avoiding Influencing Factors

According to the principle of the predictive control model [40], the proposed predictive step size is M . To ensure the safety of the AUV, the optimal obstacle avoidance trajectory points in the locally known area of the AUV are obtained. Five factors affecting obstacle avoidance are considered: safety,

rate of change of AUV yaw angle turning radius, obstacle avoidance path, and obstacle avoidance area mode.

- The Weight Function of Safe Distance

To improve the safety of AUV obstacle avoidance, the relationship between the reference obstacle point of the AUV selection and the current AUV distance as well as the safety distance must be considered. The relationships are shown below.

$$\begin{aligned} d_s(t+m) &= \|A_{uv}(x, y) - O_i(x, y)\|, \quad i = k, \dots, l \\ \gamma_s(t+m) &= \frac{|d_s(t+m) - D_s|}{D_s} \\ f_{s,value} &= k_1 \gamma_s(t+m) \end{aligned} \quad (22)$$

where d_s is the relative distance between the AUV and the boundary point of the obstacle is detected by the sonar. D_s is the safety distance. γ_s is the safety threshold, and k_1 is the weight coefficient.

Clearly, the weight function can reduce the impact of this problem on the obstacle avoidance algorithm. The risk of obstacle avoidance failure due to a single collision avoidance condition can be avoided.

- Weight Function for the Rate of Change of the AUV Yaw Angle

To prevent the shaking movement of the yaw angle during the control process, the influence of the rate of change of the AUV yaw angle is introduced to achieve the smooth control of the AUV turning heading. The weight function is defined below.

$$f_{z,value} = k_2(\psi_{t+m+1} - \psi_{t+m}) \quad (23)$$

where k_2 is the weight coefficient, and ψ_{t+m} is the AUV heading at a certain moment.

- Weight Function of the Turning Radius

The predictive track points are used for guidance control to achieve local obstacle avoidance. According to the method of calculating the turning radius proposed above, the maximum turning radius cannot be directly used as the guidance parameter, because the turning radius is only one factor that affects the obstacle avoidance performance. The specific weight function is shown below.

$$\begin{aligned} \alpha_{t+m} &= a \tan 2(A_{uv,t+m}(x, y) - O(x_o, y_o)) \\ \beta_{t+m} &= \frac{b^2 + a^2 - o^2}{2ab} \\ \gamma &= \begin{cases} \alpha_{t+m}, & A_{uv,t,x} - S_{x_o} \geq 0, A_{uv,t+m,y} \leq S_{y_o} \\ \beta_{t+m} + \pi/2, & A_{uv,t+m,x} - S_{x_o} < 0, A_{uv,t,y} \leq S_{y_o} \\ -\alpha_{t+m}, & A_{uv,t,x} - S_{x_o} \geq 0, A_{uv,t+n,y} \geq S_{y_o} \\ -\beta_{t+m} - \pi/2, & A_{uv,t+m,x} - S_{x_o} \leq 0, A_{uv,t+m,y} \geq S_{y_o} \end{cases} \\ A_{uv,t+m,x} &= O_{x_o} + R \cos(\gamma) \\ A_{uv,t+m,y} &= O_{y_o} + R \sin(\gamma) \end{aligned} \quad (24)$$

$$f_{l,value} = k_3 \|A_{uv}(x_{t+m}, y_{t+m}) - T_{\text{target}}(x, y)\|_2 \quad (25)$$

where γ is the angle between the position of the maximum turning radius where the AUV is located and the center of the circle is shown as the polar coordinate origin. R is the turning radius. S_{x_o}, S_{y_o} is the coordinate of the obstacle point where the maximum turning radius is located, and $A_{uv,t+m}(x, y)$ is the real time position of the AUV obstacle avoidance track. $T_{\text{target}}(x, y)$ is the virtual target point, and k_3 is the weight coefficient.

- Weight Function of the Obstacle Avoidance Path

To ensure that the AUV is able to reach the desired virtual target point by the shortest travel path within the predictable range, the weight function of the obstacle avoidance path can be defined by the equation below.

$$f_{d,value} = k_4 \int_{t=0.1}^{t=T} \|A_{uv,t+m}(x, y) + A_{uv,t+m+1}(x, y)\|_2 dt \quad (26)$$

where T denotes the time it takes to reach the desired target. $A_{uv,t+m}(x, y)$ is the AUV position information at time m , and k_4 is the weight coefficient.

- Weight Function of the Obstacle Avoidance Area

To improve the reliability of obstacle avoidance, the fan-shaped field of view detected by FLS is generally divided into three parts: non-avoidance areas, safe obstacle avoidance areas, and emergency obstacle avoidance areas. Therefore, it is necessary to judge which area the obstacle detected belongs to, and adopt different obstacle avoidance strategies. Therefore, the weight function of the obstacle avoidance area is defined below.

$$f_{a,value} = k_5 \cdot \begin{cases} 1, & d_i \in (l_s, l_e] \\ 0, & d_i \in [l_d, l_s) \\ -1, & d_i \in [D_s, l_d) \end{cases} \quad (27)$$

where d_i is the distance between the obstacle boundary point and the current position of the AUV. l_e, l_s, l_d, D_s is the longest distance that the sonar can detect, the maximum distance of the safe obstacle avoidance area in the field of view of sonar, the maximum range of emergency obstacle avoidance area, and safety distance, respectively. k_5 is the weight coefficient.

4.3.2. Conditional Constraints of Weight Function

Due to the impact of physical structure characteristics of the AUV itself, and the obstacle avoidance environment, several constraints must be met to achieve the purpose of collision avoidance successfully.

a. According to the structural characteristics of the physical design of the AUV, Maximum turning angular velocity meets: $-35^\circ \leq \omega_t \leq 35^\circ$;

b. To ensure the safety of the AUV, the safety distance is set to meet the conditions: $5l_o \leq D_s \leq 6l_o$, where l_o is the length of the AUV,

c. The maximum turning radius meets the conditions: $R \in [R_{\max}, R_{\min}]$.

4.3.3. Conditional Constraints of Weight Function

Combining weight functions, model parameters, and the obstacle avoidance principle, the secondary optimization function based on predictive control is obtained below.

$$\min \sum_{m=0.1}^{N-1} \|f_{s,value}\|^2 + \|f_{z,value}\|^2 + \|f_{l,value}\|^2 + \|f_{d,value}\|^2 + \|f_{a,value}\|^2 \quad (28)$$

Since Equation (27) introduces non-contiguous Boolean variables as the weight function, the objective function (28) is made non-convex. The corresponding problem becomes the mixed integer nonlinear programming (MINLP) problem, but, so far, this type of problem has not found a unanimous and mature solution [41]. However, it can be seen from Equation (27) that Boolean variables introduced are only used for the evaluation of the weight function, but not for state variables or control variables that need to be optimized. Therefore, it is still essentially nonlinear programs (NLP) problem with constraints. Furthermore, there are many excellent algorithms for solving NLP problems [42]. However, since the weight function (27) is non-convex, such NLP problems can only obtain a local optimal solution. To weaken the influence of Boolean variables, the Boolean variables of the weight function (27) are relaxed into a contiguous space, by using the hyperbolic tangent function in the Sigmoid function as follows.

$$f'_{a,value} = k_5 \tanh(\mu * d_i - 80) = k_5 \frac{1 - \exp(-\mu d_i - 80)}{1 + \exp(-\mu d_i - 80)} \quad (29)$$

where μ is the slope of the Sigmoid function. d_i is the continuous variable of the feasible domain.

The weight function before and after relaxation is shown in Figure 9. In addition, it has a good approximation and retains the property of the original weight function. The continuous convex problem is obtained after the relaxation of Boolean variables, and the predicted trajectory point of the AUV local obstacle avoidance can be obtained using the convex NLP problem. Thus, we can use the existing Sequential Quadratic Programming (SQP) method to solve the problem [43,44].

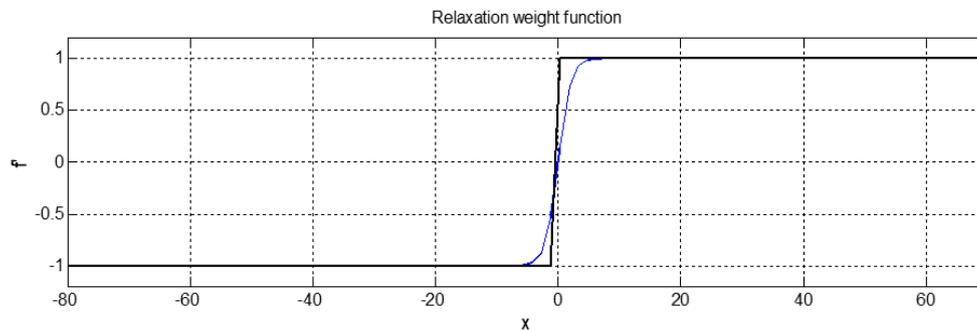


Figure 9. Comparison results of the weight functions before and after relaxation.

4.4. Overview of AUV Obstacle Avoidance Algorithms

The predictive guidance control obstacle avoidance algorithm is a real-time collision avoidance algorithm that ensures the safety of the AUV when the AUV performs tasks. The details are as follows:

(1) Always moving toward the virtual target point, which is the direction of the AUV's minimum cost. When no obstacles or many obstacles detected by the FLS configured on the AUV are outside the obstacle avoidance distance, the AUV needs to move in the direction of the target point where the greatest profit is obtained.

(2) Maintaining a safe distance: When the obstacle detected by the FLS meets the obstacle avoidance distance, the safety distance must be considered in the AUV collision avoidance. The length of the AUV is 5 m, and the safety distance is generally four times longer than the AUV's length. When selecting the obstacle avoidance guidance point, it is necessary to judge whether the safety distance meets the safe distance of the obstacle avoidance.

(3) Simple convex obstacles: When the AUV keeps the current heading in a certain area, and a single convex obstacle is detected by the FLS, the appropriate track points are chosen, according to the obstacle avoidance rule. This uses the dynamic guidance algorithm to adjust the heading in real time, and, at the same time, control the appropriate speed to achieve safe obstacle avoidance.

(4) Vortex obstacles: In the current heading path of the AUV, there may be a complex vortex obstacle. Since the obstacle appears only partly in the detectable area of the sonar and it is too hard to identify, the AUV may enter inside the concave obstacle. Therefore, it is necessary to establish obstacle avoidance rules for concave obstacles, and, based on the obstacle avoidance rules mentioned above, the effective real-time algorithm can be achieved between the two obstacle avoidance guidance algorithms along the vortex obstacle wall for the AUV to drive safely and, lastly, escape the vortex obstacles.

(5) Dense convex obstacles: When more than two obstacles were detected by the forward-looking sonar of the AUV, the above two methods of obstacle avoidance are not fully competent. For the AUV to sail safely in the multiple convex obstacle environments, we combed the predictive guidance obstacle avoidance algorithm and obstacle avoidance rules based on different classifications of multiple obstacle environments. The specific algorithm for such a setting is described in Section 4.5.3.

4.5. Different Obstacle Avoidance Algorithm Designing Various Types of Obstacles

To ensure the AUV complete its task safely, it is necessary to design reasonable obstacle avoidance algorithms for different types of obstacle environments so that the AUV can adapt to the harsh obstacle environments when sailing in an unknown underwater environment with obstacles. Therefore, by using the predictive guidance control obstacle avoidance algorithm, we can obtain the shortest and

smoothest obstacle avoidance path, so that the AUV can successfully achieve collision avoidance with less rotation and rudder angle correction. Obstacle avoidance track points are predicted based on the obstacles detected by sonar. Then, the AUV follows the obstacle avoidance track points by linear guidance and arc guidance (the radius is the turning radius) to achieve a safe obstacle avoidance path.

4.5.1. Obstacle Avoidance Algorithm Designing for Simple Convex Obstacles

The AUV uses the obstacle avoidance rule mentioned above, and the obstacle avoidance weight function is used to calculate the appropriate trajectory point to get the best track point and bypass the obstacle. Taking the right bounded obstacle in Figure 10 as an example, the predictive guidance avoidance algorithm is designed as follows.

1. By processing the boundary data of obstacles that has been detected by FLS, the number of obstacles and their relative positions to the AUV can be obtained.

2. The relation between the obstacle points detected by sonar beam and the current AUV is calculated by using the optimal parameters obtained by the weight function and the coordinate information of all points are calculated by using the method mentioned above.

$$P_{j,i,\min} = \min(D = [(O_{ix}, O_{iy}), i = 1, \dots, m]) , \quad m \leq n \quad (30)$$

$$P_{j,i,\max} = \max(D = [(O_{ix}, O_{iy}), i = 1, \dots, m]) , \quad m \leq n \quad (31)$$

$$Q_{\min j,i} = \{(P_{j,i,\min_x}, P_{j,i,\min_y}), d_{j,i,\min}, \theta_{j,i}\} , \quad \theta_{j,i} \in [\psi - \pi/3, \psi + \pi/3] \quad (32)$$

$$Q_{\max j,i} = \{(P_{j,i,\max_x}, P_{j,i,\max_y}), d_{j,i,\max}, \alpha_{j,i}\} , \quad \theta_{j,i} \in [\psi - \pi/3, \psi + \pi/3] \quad (33)$$

where $P_{j,i,\min}$ is the beam spot with the shortest distance from the AUV in all detected beam points. D is the set of all beam points. $Q_{\min j,i}$ is the set of all information of the shortest distance point. $Q_{\max j,i}$ is the set of all information of the longest distance point, and ψ is the current AUV heading.

3. Judging whether the current point is on the left or right side of the sonar center line is demarcated by the AUV heading. If it is on the left side and the result of the classification using step 1 is applied, for example, $j = 1$, then it indicates that it is a succession of single obstacles, and judges the relationship between $d_{j,i,\min}$ and the general obstacle avoidance distance l_s as well as the warning obstacle avoidance distance l_d .

$$\eta_{\text{obs}} = \frac{\text{abs}(\theta_{j,i} - \alpha_{j,i})}{\pi * 2/3} \quad (34)$$

$$\begin{cases} d_{j,i,\min} \in [l_d, l_s], & \text{SOA} \\ d_{j,i,\min} \in [D_s, l_d], & \text{EOA} \end{cases} \quad (35)$$

$$\begin{cases} \eta_{\text{obs}} \in [0, 0.5] , & \text{Hobs} \\ \eta_{\text{obs}} \in (0.5, 0.8] , & \text{OHobs} \\ \eta_{\text{obs}} \in (0.8, 1] , & \text{Eobs} \end{cases} \quad (36)$$

where η_{obs} is the proportion of the obstacle take in the sonar's field of view occupied by obstacles. SOA is the safe obstacle avoidance range. EOA is the emergency obstacle avoidance range. Hobs is the primary sonar sight range proportion. OHobs is the intermediate sonar sight range proportion. Eobs is the emergency sonar sight range proportion.

4. If $P_{j,i,\min} \in [S_i, i = 1, \dots, 40]$ and it meets the SOA and Hobs conditions, then the predictive obstacle avoidance guidance point is derived as follows and, after calculation, it goes to step 6 for execution.

$$\delta = \frac{\text{abs}(P_{j,i,\max}(x, y) - P_{\psi}(x, y))}{\text{abs}(P_{j,i,\max}(x, y) - S_{80}(x, y))} \quad (37)$$

$$P_{j,i,\text{guide}} = S_{80}(x, y) - \delta \cdot (1 - \text{abs}(S_{80}(x, y) - \frac{S_{80}(x, y)}{P_{\psi}(x, y)})) \cdot S_{80}(x, y) \quad (38)$$

$$P_{j,i,guide} = \varepsilon \cdot \text{abs}(q_{j,i}(x, y) - \frac{q_{j,i}(x, y)}{S_{80}(x, y)}) + q_{j,i}(x, y) \quad (39)$$

If $P_{j,i,\min} \in [S_i, i = 41, \dots, 80]$, and it meets the SOA and Hobs conditions, then the predictive obstacle avoidance guidance point is as follows and, after calculation, it goes to 6 for execution.

$$q_{j,i} = \frac{(P_{j,i,\max}(x, y) + S_{80}(x, y))}{2} \quad (40)$$

$$\varepsilon = \frac{\text{abs}(q_{j,i}(x, y) - P_{j,i,\min}(x, y))}{\text{abs}(P_{j,i,\min}(x, y) - S_{80}(x, y))} \quad (41)$$

$$P_{j,i,guide} = \varepsilon \cdot (1 + \text{abs}(q_{j,i}(x, y) - \frac{q_{j,i}(x, y)}{S_{80}(x, y)})) + q_{j,i}(x, y) \quad (42)$$

where δ, ε are the remaining proportion of obstacles occupying the sight range of the sonar.

5. If $P_{j,i,\min} \in [S_i, i = 1, \dots, 40]$ and it meets the SOA and Eobs conditions, then the predictive obstacle avoidance guidance point is $P_{j,i,guide} = S_{80}(x, y)$, go to 7 for execution.

If $P_{j,i,\min} \in [S_i, i = 1, \dots, 40]$ and it meets the SOA and OHobs conditions, then the predictive obstacle avoidance guidance point is $P_{j,i,guide} = S_{80}(x, y)$. Go to 7 for execution.

If $P_{j,i,\min} \in [S_i, i = 41, \dots, 80]$ and it meets the SOA and Eobs conditions, then the predictive obstacle avoidance guidance point is $P_{j,i,guide} = S_1(x, y)$, go to 7 for execution.

If $P_{j,i,\min} \in [S_i, i = 41, \dots, 80]$ and it meets the SOA and OHobs conditions, then the predictive obstacle avoidance guidance point is $P_{j,i,guide} = S_1(x, y)$, go to 7 for execution.

6. Based on the obstacle avoidance track point obtained above, we can perform the following obstacle avoidance guidance algorithm to correct the position and heading of the AUV to sail a safe obstacle avoidance path.

$$\begin{aligned} \beta_i &= a \tan 2(P_{j,i,guide}(x, y) - A_{uv}(x, y)) \\ \delta(t) &= \beta_i - a \tan 2(P_{j,i,guide}(x, y) - A_{uv}(x(t), y(t))) \\ d(t) &= \|P_{j,i,guide}(x, y) - A_{uv}(x(t), y(t))\|_2 \\ \varepsilon(t) &= d(t) \cdot \sin(\delta(t)) \\ \psi_d &= \beta_i - \alpha(t) \end{aligned} \quad (43)$$

where the selection of $\alpha(t)$ has certain rules as follows: When the current position of the AUV is far from the desired path and $\varepsilon(t) > \Delta$, the front-looking vector has no intersection with the path. Then $\alpha(t)$ is selected as an angle perpendicular to the path direction, and it is $\pi/2$. If the current position of the AUV is closer to the desired path, then $\alpha(t) = a \sin(\varepsilon(t)/\Delta)$.

$$\alpha(t) = \begin{cases} a \sin(\varepsilon(t)/\Delta), & |\varepsilon(t)| \leq \Delta \\ (\pi/2) * \text{sign}(\varepsilon(t)), & \text{else} \end{cases} \quad (44)$$

where $A_{uv}(x, y)$ is the current position of the AUV. $A_{uv}(x(t), y(t))$ is the real-time obstacle avoidance position of the AUV. $\delta(t)$ is the angle between the current AUV position and the obstacle avoidance path end connection and the path. ψ_d is the desired heading. β_i is the angle between the obstacle avoidance track point and the true north direction. $\varepsilon(t)$ is the distance between the center of the circle and the AUV. $\alpha(t)$ is the angle between the forward-looking vector and the obstacle avoidance path.

7. Based on the obstacle avoidance track points obtained above, we perform the following obstacle avoidance guidance algorithm to correct the position and heading of the AUV to form a safe obstacle avoidance path.

$$\begin{aligned} \beta_i &= a \tan 2(P_{j,i,guide}(x, y) - A_{uv}(x(t), y(t))) \\ \varepsilon(t) &= \|P(x_o, y_o) - A_{uv}(x(t), y(t))\|_2 - R_{i,\max} \\ \psi_d &= \beta_i - \alpha(t) \end{aligned} \quad (45)$$

where $\alpha(t)$ is selected as follows.

$$\alpha(t) = \begin{cases} a \cos(|\varepsilon(t)|/\Delta), & |\varepsilon(t)| \leq \Delta \\ 0, & \text{else} \end{cases} \quad (46)$$

where $P(x_o, y_o)$ is the center of the circle where the maximum turning radius $R_{i,\max}$ of the obstacle avoidance track point is located. ψ_d is the desired heading. β_i is the angle between the obstacle avoidance track point and the true north direction. $\varepsilon(t)$ is the distance between the center of the circle and the AUV, and $\alpha(t)$ is the angle between the forward-looking vector and the AUV to the center of the circle.

8. End.

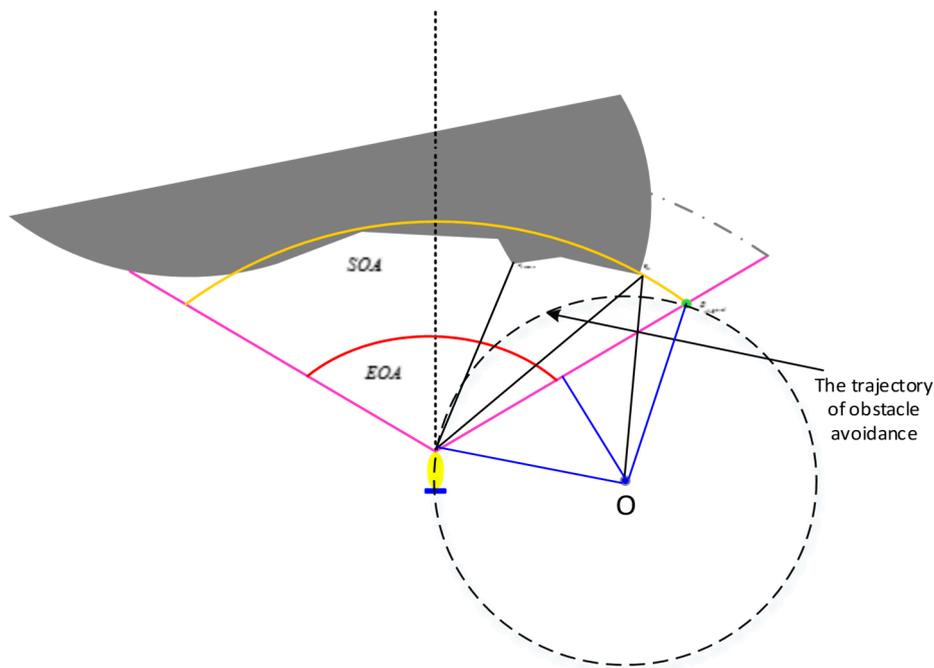


Figure 10. Path design for the right bounded obstacle.

4.5.2. Obstacle Avoidance Algorithm Design for Vortex Obstacles

To solve the trap problem caused by the vortex obstacles for the AUV in the actual underwater environment and improve the collision avoidance efficiency of the complex vortex obstacles, the line-of-sight guidance mechanism for predicting the update continuously of the trajectory segment is used, so that the trap problem caused by the obstacle can be overcome. In Figure 11, the AUV enters the trap of the vortex obstacle. Once the FLS detects the vortex obstacle and the obstacle avoidance distance requirement is satisfied, the vortex obstacle avoidance algorithm is activated. According to the data processing algorithm mentioned above, a black obstacle profile is formed, which can overcome the interference of the complex obstacle profile in obstacle avoidance.

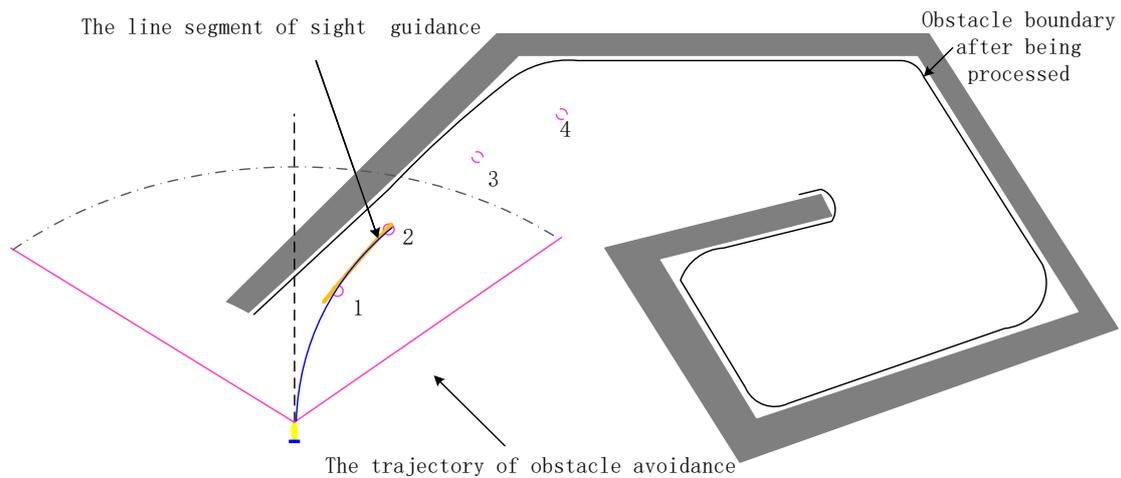


Figure 11. Path design for the obstacle of the vortex obstacle.

The predictive guidance trajectory segment calculation is performed with the partial vortex obstacle information detected by the sonar and the obstacle avoidance guidance trajectory segment obtained in the sonar's field of view in Figure 11. First, the obstacle data satisfying the obstacle avoidance distance requirements are selected. Then, the current heading angle and position information of the AUV are combined to calculate the predictive guidance track segment map vector. Lastly, all the obstacle data satisfying the obstacle avoidance distance are processed by formula (47) to obtain the predictive guidance obstacle avoidance trajectory segment.

$$P_{ov} = (p_o(x_o, y_o) - p_v(x_v, y_v)) \quad (47)$$

$$\{p_{gv} = (x_n - P_{ov}(x), y_n - P_{ov}(y)) \mid n = 1, \dots, m\}$$

where P_{ov} is the mapping vector. p_{gv}, m is the predictive guidance obstacle avoidance track point and effective range of all effective obstacles detected by sonar.

The orange guidance trajectory segment in Figure 11 can be generated by performing the line-of-sight guidance mechanism for predicting the update continuously of the trajectory segment. Then the historical tracking information is recorded by setting the memory unit $D \in R^{n \times 2}$ including the heading and path position information that the AUV has traveled, which is taken as comparative data for the AUV that enters the vortex obstacle, finds the target, and escapes from the trap. By comparing the current predicted data with the historical track information in D , the correct predicted trajectory segment of escaping the trap can be obtained. Lastly, by choosing the correct predictive guidance track, the AUV can escape from the obstacle trap. After the AUV executes the orange line segment guidance trajectory, the FLS will further detect the remaining information of the vortex obstacle. Then we repeat the above steps to continuously obtain the obstacle avoidance predictive guidance trajectory segment, and realize the purpose of collision avoidance and escape the vortex obstacle trap.

Regarding the termination condition of the vortex obstacle collision avoidance, a virtual target point is designed in this case, and the AUV moves toward the direction combing target's direction (Tolerance deviation from the target angle: $\Delta \in [-15^\circ, 15^\circ]$) and the relative distance reduced between the AUV and the target.

4.5.3. Design of the Obstacle Avoidance Algorithm for Dense Convex Obstacles

If the number of obstacles appearing in the field of view of the FLS are more than or equal to two and they suit the range requirements for obstacle avoidance, the environment is considered to be a dense complex obstacle environment.

It is necessary to design a special obstacle avoidance algorithm for the dense obstacle environment because the single obstacle avoidance algorithm and the vortex obstacle avoidance algorithm cannot

solve the problem caused by the dense obstacle environment in the current field of view. The multiple obstacle's data processing in the field of view of FLS is shown in Figure 12. The optimal predictive obstacle avoidance track point can be obtained by combing obstacle avoidance rules and weight functions with the following formula as follows.

$$P_{j,i,guide} = \frac{P_{j,S_i,\min}(x,y) + P_{j+1,S_k,\min}(x,y)}{2}, \quad (j = 1, \dots, m+1; i, k \in [1, 80]) \quad (48)$$

$$G_{uide} = \{P_{j,i,guide}(x,y) | j = 1, \dots, m+1; i \in [1, 80]\} \quad (49)$$

$$G_{ap} = \{l_i, i = 1, \dots, m+1\} \quad (50)$$

$$\lambda_{s,i} = \begin{cases} 0, & G_{ap,i} \leq 4l_o \\ \left(\frac{|G_{ap,i}-4l_o|}{6l_o}\right)^{1/2}, & 4l_o \leq G_{ap,i} \leq 6l_o \\ 1, & G_{ap,i} \geq 6l_o \end{cases} \quad (51)$$

$$\lambda_{c,i} = \rho_{d,i} \cdot \sin(\varphi_{S_i}) \cdot \gamma_c + \lambda_{s,i} \quad (52)$$

$$P_{guide} = f(G_{ap}, \lambda_{c,i}) \quad (53)$$

where G_{uide} is the possible existing guidance track points of all obstacle gaps. G_{ap} is the spacing of all obstacles. $\lambda_{s,i}, \lambda_{c,i}$ is the spacing width influence factor and track point selection factor, respectively. γ_c is the attenuation factor. $\rho_{d,i}, l_o$ are the distance between the track point and the AUV and the length of the AUV, respectively. f is the evaluation function of the optimal guidance trajectory point. m is the number of obstacles detected, and, in Figure 12, $m = 4$.

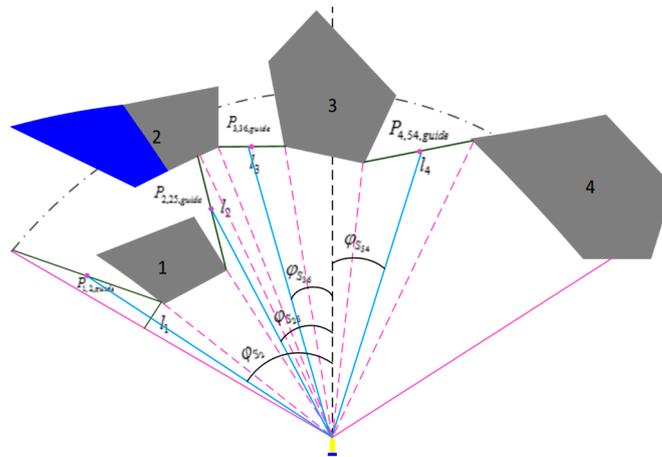


Figure 12. Dense obstacles environment avoidance.

Considering the influence of the length of the turning radius on the safety of the AUV avoidance obstacle, it is unreasonable to select one location in the current gap as the guidance track point when the largest obstacle gap is found. As a result, it will bring the cost of energy and cause the AUV to deviate from the target point. Therefore, a variety of factors should be taken into account when selecting the attenuation factor and evaluation function. It needs to meet the impact of weights in the selection of trajectory points that $\gamma_c = 0.5$ is selected, and f selects piecewise linear function. In Figure 12, the blue portion of the obstacle is an area that cannot be detected by the sonar in reality. It is impossible to distinguish between the obstacles No. 1 and No. 2 for the AUV because they belong to one area in the field of view of sonar. However, in a real situation, the obstacle is assembled by the number 1 and 2 obstacles. It is undoubtedly beneficial for obstacle avoidance, in reality, to use the corresponding segmentation algorithm to classify different obstacles. The data processing method in Section 3.4 can be used to realize the segmentation of obstacles No. 1 and No. 2 in Figure 12 and obtain the obstacle

spacing l_2 and the predictive guidance point $P_{2,25,guide}$. Therefore, the optional guidance trajectory point of AUV is added to improve the safety and selectivity of obstacle avoidance.

5. Simulation Results and Discussions

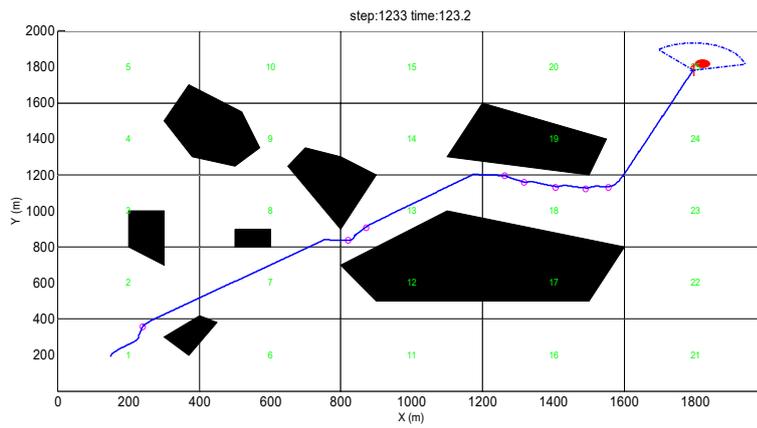
In the simulation environment of this paper, three kinds of obstacle environments were established to verify that the obstacle avoidance algorithm mentioned above could solve the obstacle avoidance problem in a complex environment and improve the obstacle avoidance efficiency. Obstacle information in the entire simulation environment, including the type, number, and location of obstacles, is generated randomly. The AUV does not know this information in advance. During the process of AUV sailing to the predetermined target point, the obstacle can be detected when the unknown obstacle enters the field of view of the sonar. Then the AUV avoids obstacles in real time, according to the obstacle avoidance algorithm proposed above. Software simulation experiments were carried out in the MATLAB 2014. A 2000 m × 2000 m two-dimensional area was established, which was divided into 400 m × 400 m task sub-areas. In each task sub-region, several grids of equal size were divided, according to unit length to evaluate the cost of AUV's trajectory when executing the obstacle avoidance algorithm. In three kinds of the obstacle environment, the speed of the AUV when it does not detect the obstacle is 3.5 m/s. The speed of executing the obstacle avoidance algorithm was 2 m/s, and the unit step time was 0.1 s. The proposed obstacle avoidance algorithm was tested and verified in three kinds of obstacle environments, respectively.

5.1. Simulation Verification in a Simple Convex Obstacle Environment

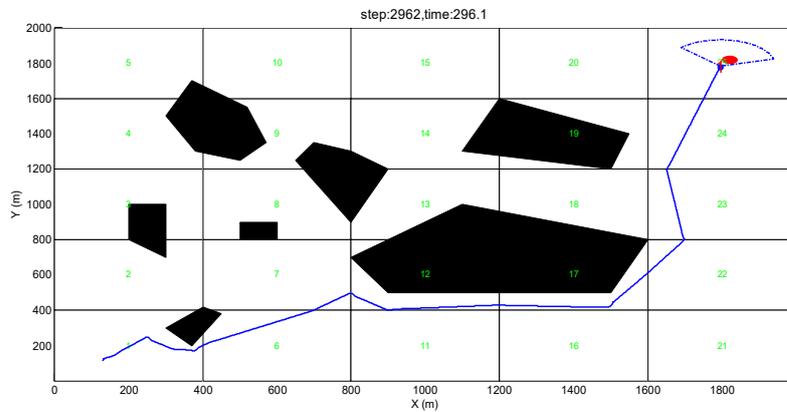
This design requires AUV to have a simple convex obstacle environment obstacle avoidance capability to arrive at virtual target points, to verify the effectiveness and efficiency of the obstacle avoidance algorithm in dealing with simple convex obstacle environments. As shown in Figure 13a, the starting position of the AUV is (180, 180), and the heading angle is set to 0° with the two-dimensional y axis as the true north direction. The virtual target point is a red circle, and the center of the circle is (1800, 1800) and the radius is 10. Meanwhile, seven convex obstacles are generated randomly in the two-dimensional environment to form a simple convex obstacle environment. It is stipulated that, when the distance between the AUV and the virtual target point is 10 m, the obstacle avoidance process is ended, and AUV has reached the target and successfully completed the target searching task.

The obstacle avoidance trajectory of APF in a simple, convex obstacle environment is shown in Figure 13b. Clearly, it can be seen that the obstacle avoidance trajectory is not very smooth. Since the selected obstacle avoidance path is not the best path for the current obstacle environment, the cost value of the obstacle avoidance is much higher than that in Figure 13a. The distance between the AUV and the obstacle is relatively close during the obstacle avoidance process, which is likely to increase the risk of the obstacle avoidance failure.

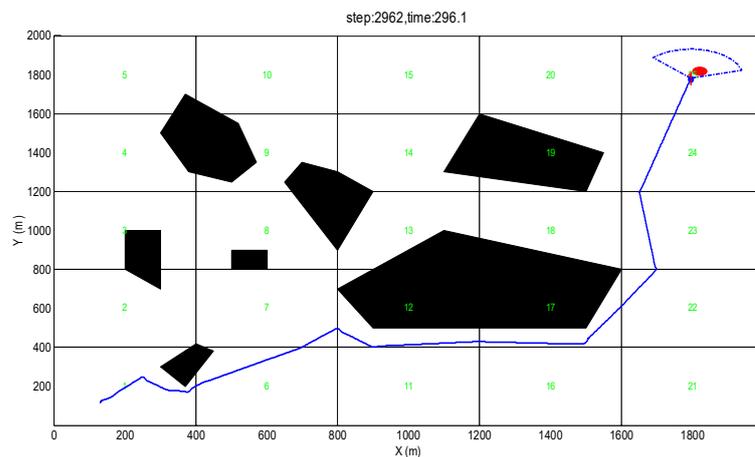
The trajectory of the PSO algorithm used for obstacle avoidance is shown in Figure 13c. It can be seen from Figure 13c that the AUV chooses the path to avoid obstacles as the obstacle avoidance trajectory, which is close to the virtual target point at the same time. However, this also increases the overall cost of the obstacle avoidance process. Compared with Figure 13b, its track is relatively smooth, but its cost increases. However, there is no probability of an increase in the risk of avoiding obstacles. In a word, the obstacle avoidance algorithm proposed is more suitable to the simple obstacle environment.



(a)



(b)



(c)

Figure 13. Simple obstacle environment avoidance trajectory. (a) PGOA (Predictive guidance obstacle avoidance). (b) APF (Artificial potential field). PSO (Particle swarm optimization).

5.2. Simulation Verification in the Vortex Obstacle Environment

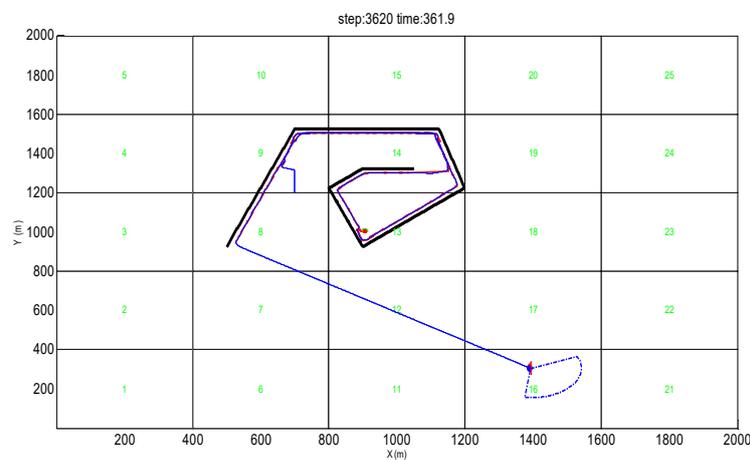
The AUV is designed to find the target point in the vortex obstacle and escape from the obstacle to verify the effect of the obstacle avoidance algorithm proposed for escaping from the vortex obstacle

environment. The starting position of the AUV is set to (700, 1200), and the heading angle is set to 0° with the two-dimensional y axis as the true north direction. A red circular target point is set in the vortex obstacle with its center being (910, 1010) and the radius is 10. The stop position is set to (1400, 300) after the AUV escapes from the vortex obstacle. Eventually, the AUV enters the vortex obstacle and escapes from the obstacle safely after detecting the target. Then AUV reaches the specified destination. As shown in Figure 14, a concave obstacle environment in a two-dimensional environment composed of complex vortex obstacles is set. It is stipulated that the target point can be marked when the target point appears within the range of 80 m of the AUV sonar's field of view. It indicates that the AUV has detected the target in the vortex obstacle and completed the obstacle avoidance task successfully and reaches the predetermined end position when the AUV is within the range of 10 m from the end position.

An obstacle avoidance algorithm is set for the vortex obstacle. Figure 14a shows that the AUV enters the obstacle and detects the target and does not fall into the vortex obstacle. The AUV can escape the obstacle at once and quickly reach the designated termination point. The orange trajectory represents the predicted trajectory formed from the detected obstacle data. Additionally, the blue trajectory represents the true obstacle avoidance trajectory of the AUV. The figure shows that the red target point has been marked in green, which indicates that the target point has been successfully detected.

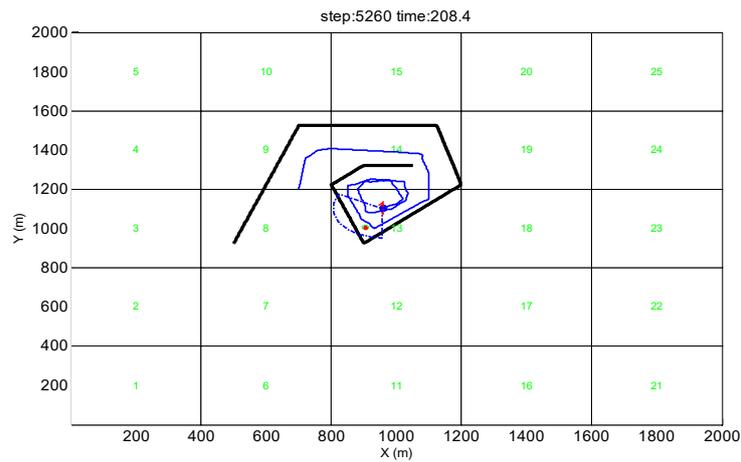
Figure 14b shows the AUV obstacle avoidance trajectory for the return obstacle realized by the APF method. AUV is stuck in a return obstacle and cannot escape from the obstacle because the repulsive force inside the vortex obstacle is large and the gravitational force at the end position is small. AUV only completes the target detection, but cannot escape the obstacle.

Figure 14c shows the AUV obstacle avoidance trajectory formed by the PSO algorithm. As the figure shows, the AUV can detect the target point and escape the obstacle and reach the specified end position. However, it can fall into the obstacle for a period of time before it escapes from the obstacle, which increases the value of the whole obstacle avoidance process.

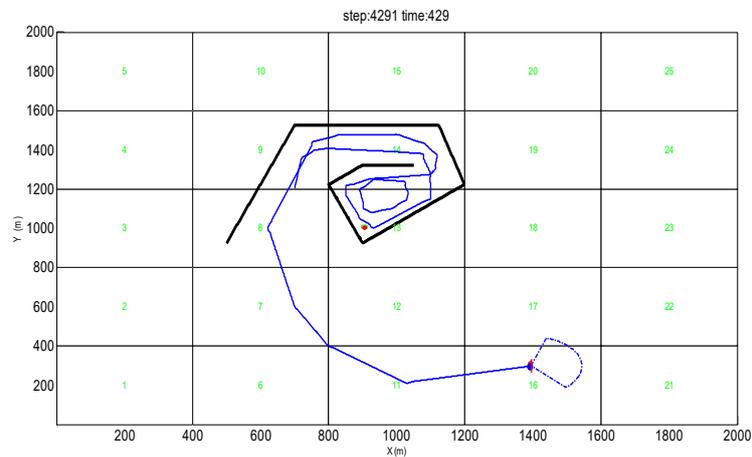


(a)

Figure 14. Cont.



(b)



(c)

Figure 14. Trap obstacles environment obstacle avoidance trajectory. (a) PGOA (Predictive guidance obstacle avoidance). (b) APF (Artificial potential field). (c) PSO (Particle swarm optimization).

5.3. Simulation Verification in a Dense Obstacle Environment

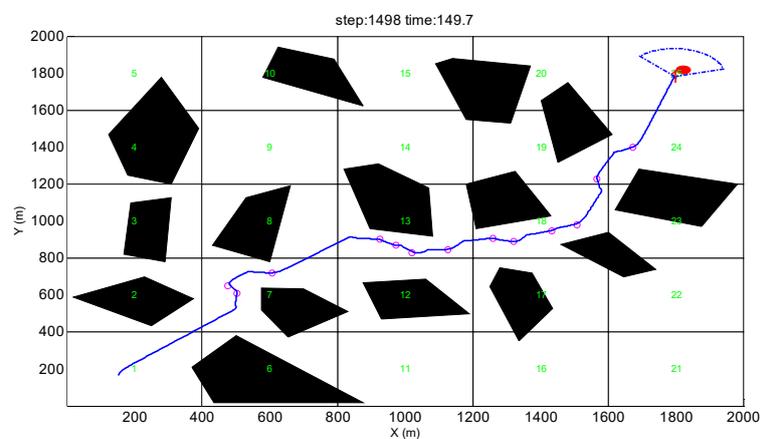
This design requires AUV to have a dense convex obstacle environment with an obstacle avoidance capability to arrive at virtual target points and to verify the effectiveness and efficiency of the obstacle avoidance algorithm in dealing with a dense convex obstacle environment. As shown in Figure 15, the starting position of the AUV is set to (180, 180), and the heading angle is set to 0° with the two-dimensional y axis as the true north direction. The virtual target point is the red circle with its center at (1800, 1800) and the radius is 10. Fifteen convex obstacles are generated randomly in the two-dimensional environment to form a dense convex obstacle environment. In this experiment, the situation of the obstacle avoidance process ended when the distance between the AUV and the virtual target point is 10 m, which indicates that the AUV has successfully completed the obstacle avoidance task and reached the target.

Figure 15a shows the obstacle avoidance track formed by the PGOA algorithm in a dense convex obstacle environment. The pink circle in the figure indicates the predictive track points. As the figure shows, the AUV chooses a reasonable obstacle avoidance path to reach the virtual target point without

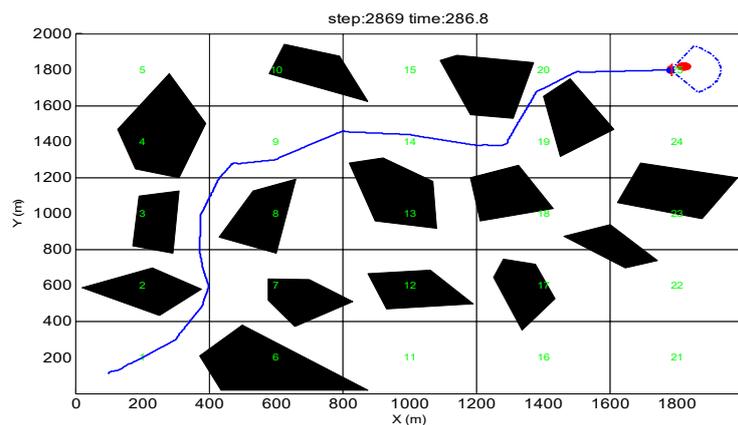
bringing an obstacle avoidance risk. Meanwhile, it can maintain the requirement of safe obstacle avoidance distance. In addition, the cost of the entire obstacle avoidance is controlled within 1500.

Figure 15b shows the obstacle avoidance trajectory formed by the APF method. As the figure shows, the AUV does not select an appropriate path as the track to reach the target. Meanwhile, it tends to increase the risk of obstacle avoidance when AUV crosses the obstacle group with relatively small obstacle spacing, which may result in the failure of obstacle avoidance. The path to avoid obstacles does not meet the safe obstacle avoidance distance at the beginning of obstacle avoidance, which does not meet the obstacle avoidance requirements of this paper. In addition, compared with the cost of Figure 15a, there are also many resources that are wasted.

Figure 15c shows the obstacle avoidance path formed by the PSO algorithm. As the figure shows, the path selected by AUV is very close to the obstacle avoidance trajectory of Figure 15a and the path is relatively smooth. It fully meets the safe obstacle avoidance distance. However, it increases the cost of the overall obstacle avoidance because the trajectory to the target is not the most suitable. Moreover, it is similar to the APF obstacle avoidance cost, which is twice the cost of PGOA.

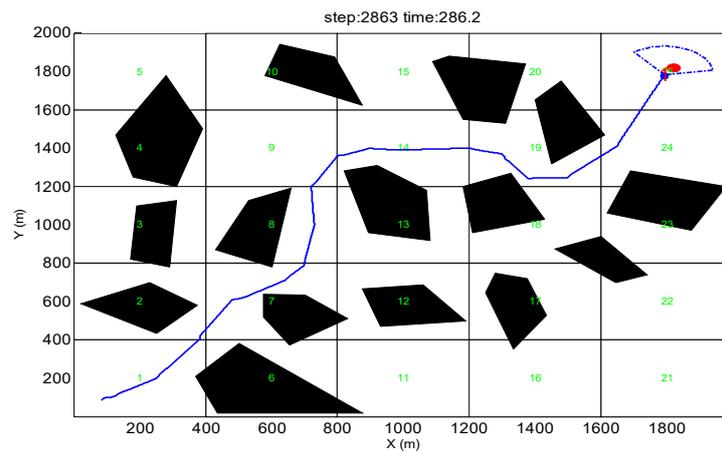


(a)



(b)

Figure 15. Cont.



(c)

Figure 15. Vortex obstacles environment obstacle avoidance trajectory. (a) PGOA (Predictive guidance obstacle avoidance). (b) APF (Artificial potential field). (c) PSO (Particle swarm optimization).

Figure 16 shows the comparison of the cost of the obstacle avoidance process brought by different algorithms in three kinds of obstacle environments. It can be seen from Figure 16 that the PGOA obstacle avoidance algorithm can successfully complete the obstacle avoidance process in different complex obstacle environments, and the cost of the entire obstacle avoidance process is also minimal. APF can adapt to a simple convex obstacle environment and a dense convex obstacle environment. However, its cost is twice as much as the PGOA algorithm, and it cannot handle the vortex obstacle environment. The PSO algorithm can complete obstacle avoidance for the three kinds of obstacle environments, but the cost is very high. In a word, the obstacle avoidance algorithm proposed in this paper has great advantages in dealing with various obstacle environments.

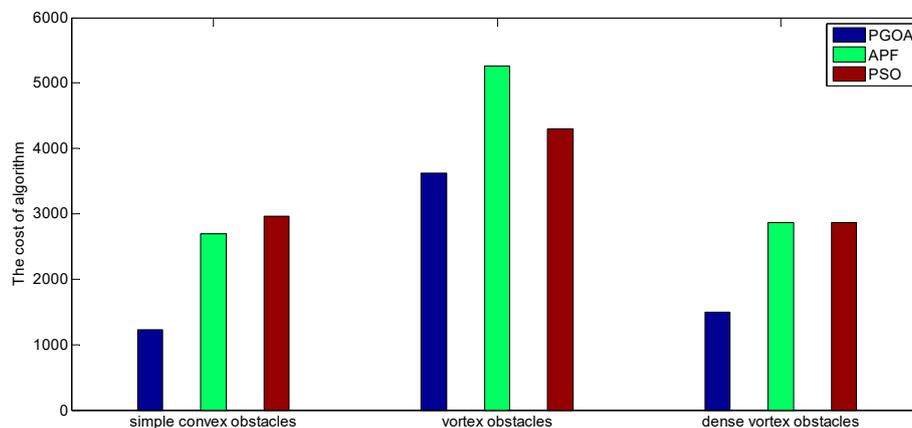


Figure 16. Algorithmic cost in three obstacle environments.

6. Conclusions

The PGOA proposed in this paper deals with various complex obstacle environments, which includes complex convex polygon obstacles and complex concave obstacles. This algorithm can handle simple convex obstacles, dense convex obstacles, and vortex obstacles. The AUV's obstacle avoidance trajectory is also close to smooth. The AUV through various obstacle environments easily reaches the predetermined virtual target point during the whole obstacle avoidance process. Furthermore, the AUV's obstacle avoidance process has always met the requirements of safe obstacle avoidance distance, and the cost of the entire obstacle avoidance process is less than other traditional

algorithms. During the process of facing the vortex obstacles, the AUV can escape the vortex obstacle environment once inside the vortex obstacles by remembering the direction of the path that it has traveled before, which avoids falling into the obstacles and causing an obstacle avoidance process failure. The next step in this paper is to optimize the proposed algorithm to adapt to the AUV pool experiment and extend the algorithm to the complex marine environment where ocean currents and dynamic obstacles exist. At the same time in the actual underwater environment, most obstacles have different shapes at different heights. The obstacle avoidance problem is much more complicated in the three-dimensional environment. Therefore, in the later research, the two-dimensional obstacle avoidance algorithm is used to penetrate into the three-dimensional environment.

Author Contributions: J.L. mainly provided design research ideas, data collection, comment writing and editing, and finalization. J.Z. wrote the manuscript, conducted the literature search, produced icons, and collected data and analysis. Z.Y. conducted the literature search, produced icons, and analyzed the data. H.Z. conducted data collection and data analysis.

Funding: The National Natural Science Foundation of China (Grant No. 51609046/E091002) and the Research Fund from Science and Technology on Underwater Vehicle Technology (Grant No. 614221502061701) supported this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yuh, J. Design and Control of Autonomous Underwater Robots: A Survey. *Auton. Robot.* **2000**, *8*, 7–24. [[CrossRef](#)]
2. Xin, J. The future underwater robot of the US Navy. *Robot. Appl.* **2000**, *35*, 9–11.
3. Incze, M.L. Light weight autonomous underwater vehicles (AUVs) performing coastal survey operations in REP10A. *Ocean Dyn.* **2011**, *61*, 1955–1965. [[CrossRef](#)]
4. Li, Z.; Yang, C.; Ding, N.; Bogdan, S.; Ge, T. Robust adaptive motion control for underwater remotely operated vehicles with velocity constraints. *Int. J. Control Autom. Syst.* **2012**, *10*, 421–429. [[CrossRef](#)]
5. Martin, W.; Tal, S. Linear Quadratic Optimal Control based Missile Guidance Law with Obstacle Avoidance. *Ieee Trans. Aerosp. Electron. Syst.* **2018**, *55*, 205–214.
6. Zi, B.; Lin, J.; Qian, S. Localization, obstacle avoidance planning and control of a cooperative cable parallel robot for multiple mobile cranes. *Robot. Comput. Integr. Manuf.* **2015**, *34*, 105–123. [[CrossRef](#)]
7. Wu, J.; Yue, X.; Li, W. Integration of Hardware and Software Designs for Object Grasping and Transportation by a Mobile Robot with Navigation Guidance Via a Unique Bearing-Alignment Mechanism. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 576–583.
8. Kumar, R.P.; Dasgupta, A.; Kumar, C.S. Real-time optimal motion planning for autonomous underwater vehicles. *Ocean Eng.* **2005**, *32*, 1431–1447. [[CrossRef](#)]
9. Fossen, T.I. *Guidance and Control of Ocean Vehicles*; John Wiley & Sons: Hoboken, NJ, USA, 1994.
10. Likhachev, M.; Ferguson, D.I.; Gordon, G.J.; Stentz, A.; Thrun, S. Anytimedynamic A *: An anytime, replanning algorithm. In Proceedings of the 15th International Conference on Automated Planning and Scheduling, Monterey, CA, USA, 5–10 June 2005; pp. 262–271.
11. Carsten, J.; Ferguson, D.; Stentz, A. 3D field D *: Improved path planning and replanning in three dimensions. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 3381–3386.
12. Carroll, K.P.; McClaran, S.R.; Nelson, E.L.; Barnett, D.M.; Friesen, D.K.; William, G. AUV path planning: An A * approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, USA, 2–3 June 1992.
13. Pereira, A.A.; Binney, J.; Jones, B.H.; Ragan, M.; Sukhatme, G.S. Toward risk aware mission planning for autonomous underwater vehicles. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011.
14. Pereira, A.A.; Binney, J.; Hollinger, G.A.; Sukhatme, G.S. Risk-aware pathplanning for autonomous underwater vehicles using predictive ocean models. *J. Field Robot* **2013**, *30*, 741–762. [[CrossRef](#)]

15. Ferguson, D.; Stentz, A. Using interpolation to improve path planning the field D* algorithm. *J. Field Robot* **2006**, *23*, 79–101. [\[CrossRef\]](#)
16. Daily, R.; Bevly, D.M. Harmonic potential field path planning for high speed vehicles. In Proceedings of the American Control Conference, Washington, DC, USA, 11–13 June 2008; pp. 4609–4614.
17. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [\[CrossRef\]](#)
18. Sullivan, J.; Waydo, S.; Campbell, M. Using stream functions for complex behavior and path generation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, USA, 11–14 August 2003.
19. Zeng, Z.; Sammut, K.; Lamma, A.; He, F.; Tang, Y. Efficient path re-planning for AUVs operating in spatiotemporal currents. *J. Intell. Robot. Syst.* **2015**, *79*, 135–153. [\[CrossRef\]](#)
20. Cao, J.; Li, Y.; Zhao, S.; Bi, X. Genetic-Algorithm-Based Global Path Planning for AUV. In Proceedings of the 2016 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 10–11 December 2016.
21. Aghababa, M.P. 3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles. *Appl. Ocean Res.* **2012**, *38*, 48–62. [\[CrossRef\]](#)
22. Harun, N.; Zain, Z.M.; Noh, M.M. PSO approach for a PID back-stepping control method in stabilizing an underactuated X4-AUV[C]// 2017 IEEE 7th International Conference on Underwater System Technology: Theory and Applications (USYS). 2018.
23. Hu, Q.Y.; Zhou, J.; Zha, Z. Application of PSO-BP Network Algorithm in AUV Depth Control. *Appl. Mech. Mater.* **2013**, *321–324*, 2025–2031. [\[CrossRef\]](#)
24. Dadgar, M.; Jafari, S.; Hamzeh, A. A PSO-based multi-robot cooperation method for target searching in unknown environments. *Neurocomputing* **2016**, *177*, 62–74. [\[CrossRef\]](#)
25. Zhuang, Y.; Sharma, S.; Subudhi, B.; Huang, H.; Wan, J. Efficient collision-free path planning for Autonomous Underwater Vehicles in dynamic environments with a hybrid optimization algorithm. *Ocean Eng.* **2016**, *127*, 190–199. [\[CrossRef\]](#)
26. Horner, D.P.; Healey, A.J.; Kragelund, S. AUV Experiments in Obstacle Avoidance. In Proceedings of the OCEANS 2005 MTS/IEEE, Washington, DC, USA, 17–23 September 2005.
27. Norgren, P.; Skjetne, R. Line-of-sight iceberg edge-following using an AUV equipped with multibeam sonar. *IFAC-Pap.* **2015**, *48*, 81–88. [\[CrossRef\]](#)
28. Moe, S.; Pettersen, K.Y. Set-based Line-of-Sight (LOS) path following with collision avoidance for underactuated unmanned surface vessel. In Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 21–24 June 2016.
29. Wiig, M.S.; Pettersen, K.Y.; Krogstad, T.R. Collision Avoidance for Underactuated Marine Vehicles Using the Constant Avoidance Angle Algorithm. *IEEE Trans. Control Syst. Technol.* **2019**, *99*, 1–16. [\[CrossRef\]](#)
30. Zhang, G.; Deng, Y.; Zhang, W. Robust neural path-following control for underactuated ships with the DVS obstacles avoidance guidance. *Ocean Eng.* **2017**, *143*, 198–208. [\[CrossRef\]](#)
31. Sur, J.; Lee, Y.; Choi, H.T. Obstacle Avoidance of Autonomous Underwater Vehicle by Using Streamline Function. *Ifac Proc. Vol.* **2012**, *45*, 236–241. [\[CrossRef\]](#)
32. Manup, B.; Raja, P. Collision-avoidance for mobile robots using region of certainty: A predictive approach. *J. Eng. Sci. Technol.* **2016**, *11*, 18–28.
33. Ataei, M.; Yousefi-Koma, A. Three-dimensional optimal path planning for waypoint guidance of an autonomous underwater vehicle. *Robot. Auton. Syst.* **2015**, *67*, 23–32. [\[CrossRef\]](#)
34. Wu, X.; Feng, Z.; Zhu, J.; Allen, R. Line of Sight Guidance with Intelligent Obstacle Avoidance for Autonomous Underwater Vehicles. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006.
35. Shojaei, K.; Arefi, M.M. On the neuro-adaptive feedback linearising control of underactuated autonomous underwater vehicles in three-dimensional space. *IET Control Theory Appl.* **2015**, *9*, 1264–1273. [\[CrossRef\]](#)
36. Fossen, T.I.; Sagatun, S.I. Adaptive Control of Nonlinear Systems: A Case Study of Underwater Robotic Systems. *J. Robot. Syst.* **1991**, *8*, 393–412. [\[CrossRef\]](#)
37. Zhao, S.; Lu, T.F.; Anvar, A. Automatic object detection for AUV navigation using imaging sonar within confined environments. In Proceedings of the 2009 4th IEEE Conference on Industrial Electronics and Applications, Xi'an, China, 25–27 May 2009; pp. 3648–3653.

38. Chatzichristofis, S.; Kapoutsis, A.; Kosmatopoulos, E.B.; Doitsidis, L.; Rovas, D.; de Sousa, J.B. The NOPTILUS project: Autonomous Multi-AUV Navigation for Exploration of Unknown Environments. *IFAC Proceedings Volumes* **2012**, *45*, 373–380. [[CrossRef](#)]
39. Yan, Z.; Li, J.; Zhang, G.; Wu, Y. A Real-Time Reaction Obstacle Avoidance Algorithm for Autonomous Underwater Vehicles in Unknown Environments. *Sensors* **2018**, *18*, 438. [[CrossRef](#)]
40. Xi, Y.-G. *Predictive Control*, 2nd ed.; National Defense Industry Press: Beijing, China, 2013.
41. Liu, M.-M.; Cui, C.-F.; Tong, X.-J.; Dai, Y.-H. Algorithm Software for Mixed Integer Nonlinear Programming and Recent Progress. *Sci. Sin.* **2016**, *46*, 1–20. [[CrossRef](#)]
42. Chen, B.-L. *Optimization Theory and Algorithm*; Tsinghua University Press: Beijing, China, 2005.
43. Powell, M.J.D. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In *Nonlinear Programming 3*; Mangasarian, O.L., Meyer, R.R., Robinson, M., Eds.; Academic Press: Cambridge, MA, USA, 1978.
44. Constrained Optimization, Sequential Quadratic Programming (SQP). *Optimization Toolbox User's Guide*; Math-Works Inc.: Natick, MA, USA, 2016.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).