Journal of *Marine Science and Engineering*

MDPI

Article

A 2D Optimal Path Planning Algorithm for Autonomous Underwater Vehicle Driving in Unknown Underwater Canyons

Yushan Sun, Xiaokun Luo, Xiangrui Ran * and Guocheng Zhang

School of Naval Engineering, Harbin Engineering University, Harbin 150001, China; sunyushan@hrbeu.edu.cn (Y.S.); luoxiaokun@hrbeu.edu.cn (X.L.); zhangguocheng@hrbeu.edu.cn (G.Z.)
* Correspondence: ranxiangrui@hrbeu.edu.cn

Abstract: This research aims to solve the safe navigation problem of autonomous underwater vehicles (AUVs) in deep ocean, which is a complex and changeable environment with various mountains. When an AUV reaches the deep sea navigation, it encounters many underwater canyons, and the hard valley walls threaten its safety seriously. To solve the problem on the safe driving of AUV in underwater canyons and address the potential of AUV autonomous obstacle avoidance in uncertain environments, an improved AUV path planning algorithm based on the deep deterministic policy gradient (DDPG) algorithm is proposed in this work. This method refers to an end-to-end path planning algorithm that optimizes the strategy directly. It takes sensor information as input and driving speed and yaw angle as outputs. The path planning algorithm can reach the predetermined target point while avoiding large-scale static obstacles, such as valley walls in the simulated underwater canyon environment, as well as sudden small-scale dynamic obstacles, such as marine life and other vehicles. In addition, this research aims at the multi-objective structure of the obstacle avoidance of path planning, modularized reward function design, and combined artificial potential field method to set continuous rewards. This research also proposes a new algorithm called deep SumTree-deterministic policy gradient algorithm (SumTree-DDPG), which improves the random storage and extraction strategy of DDPG algorithm experience samples. According to the importance of the experience samples, the samples are classified and stored in combination with the SumTree structure, high-quality samples are extracted continuously, and SumTree-DDPG algorithm finally improves the speed of the convergence model. Finally, this research uses Python language to write an underwater canyon simulation environment and builds a deep reinforcement learning simulation platform on a high-performance computer to conduct simulation learning training for AUV. Data simulation verified that the proposed path planning method can guide the under-actuated underwater robot to navigate to the target without colliding with any obstacles. In comparison with the DDPG algorithm, the stability, training's total reward, and robustness of the improved Sumtree-DDPG algorithm planner in this study are better.

Keywords: autonomous underwater vehicle; 2D optimal path planning; deep reinforcement learning; unknown underwater canyons environment

Citation: Sun, Y.; Luo, X.; Ran, X.; Zhang, G. A 2D Optimal Path Planning Algorithm for Autonomous Underwater Vehicle Driving in Unknown Underwater Canyons. *J. Mar. Sci. Eng.* **2021**, *9*, 252. https://doi.org/10.3390/jmse9030252

Academic Editor: Philippe Blondel

Received: 16 January 2021
Accepted: 22 February 2021
Published: 27 February 2021

1. Introduction

In recent years, autonomous underwater vehicles (AUVs) have elicited wide attention because of revolutionizing the oceanic research with applications on numerous scientific fields, such as marine geoscience, submarine oil exploration, submarine salvage, submarine pipeline repair, and archeology [1–3]. Among all of the functions of AUV, autonomous obstacle-avoidance capability is the most important one because the obstacles are usually unknown for AUVs in underwater environment; thus, AUV can easily run into obstacles, thereby causing them to malfunction or even damage the robot [4].

Several autonomous navigation methods for obstacle-avoidance have been reported in the literature. Lozano et al. [5,6] proposed visibility graph algorithm. In this algorithm,

AUV is regarded as a bit, and obstacles are considered plane polygons. Subsequently, the starting point, goal point, and polygon obstacle of each vertex are connected. Moreover, all attachment fellowships without path obstructions are considered collision-free path. Finally, a safety view is formed, and some algorithms are used to search the optimal path. The principle of this method is simple and easy to realize. Takahashi et al. [7] proposed the Voronoi diagram method. In 1983, the existence of a certain distance between the planned path and obstacles can be satisfied, and factors, such as safety, can be considered. The planning time increases and decreases with the density of obstacles. Although the shortest path can be determined by this method, it lacks flexibility. In Takahashi et al. [8,9], precise raster algorithm divided free space into no-overlap grid units. The grid is dominated by obstacles for grid assignment and makes a series of parallel lines to each obstacle vertices. Edges and obstacles in the planning environment are stopped. Eventually, the space is decomposed into a series of trapezoidal area to realize obstacle avoidance. In literature [10,11], quad-tree and octree decomposition methods were used to establish the plane sea area obstacle model and the submarine terrain model, respectively. In addition, the current velocity and direction could be used as grid attribute information to establish the current model. A* and D* algorithms are widely used path search algorithms. A* algorithm selects the optimal path node by calculating the evaluation function of all candidate nodes to the target point, which is suitable for static path planning [12]. In the bionic fish path planning problem, Qiang et al. [13] adopted the deployable point method to reduce the search nodes and improve the search efficiency; however, the environmental factors were not considered. D* algorithm [14] is the dynamic A* algorithm that is suitable for solving dynamic path planning problems by detecting the changes in the previous or nearby nodes of the shortest path. Artificial potential field method is a virtual method proposed by Khatib et al. [15]. This method is widely used in the path planning field, and its concept is to construct various virtual potential fields for the path planning of AUV [16]. Warren [17] used the artificial potential field method to carry out path planning for underwater robots and realized the global path planning of AUV in two-dimensional (2D) and three-dimensional (3D) spaces by reducing the local minima through heuristic knowledge. Chao [18] adopted optimization theory, combined artificial potential field with obstacle constraint, and transformed path planning problem into solving constraint and semi-constraint problems. Cheng et al. [19] used velocity vector synthesis algorithm to enable the combined velocity of ocean current velocity and AUV velocity point to the target, thereby minimizing resource consumption. In Ferrari et al. [20], aiming at the problem of collaborative planning of multi-AUV to avoid multi-detection platform network, the detection platform was considered a virtual obstacle, and the planning result could determine the minimum exposure probability and the non-collision path by modifying the fitness function.

With the progress of computer technology, artificial intelligence has received extensive attention in various fields. The artificial intelligence-based path planning technology aims to transform the behavior and thoughts of some natural animals into algorithms that will be used in the path planning of mobile devices. Currently, artificial intelligence algorithms, such as particle swarm optimization algorithm, ant colony algorithm, evolutionary computing [21], genetic algorithm, and self-organizing neural network [22], have emerged and are widely applied. Xu et al. [23] used the genetic algorithm and particle swarm optimization (GA-PSO) hybrid planning algorithm to realize the AUV global path planning under current conditions. Wang et al. [24] designed cutting and handicap operators to solve the problem of ant colony path planning and realized the AUV global path planning in a 2D grid environment model. In paper [25], particle swarm optimization (PSO) was used to solve the path planning problem of dynamic environment, and the speed and heading information of the robot were introduced into the objective function. The results verify that PSO has good real-time performance in solving the path planning problems. Xin et al. [26] improved the ant colony path planning, designed the cutting and handicap operators, and realized the AUV global path planning in the 2D grid environment model.

These methods need to know the global environment and does not have the ability to learn and explore the unknown environment path planning.

AUV is one of the most important means to explore the deep sea world. The deep ocean is a complex and changeable environment which is distributed with various mountains. When the underwater autonomous vehicle reaches the deep sea, it will face many large and small underwater canyons, and hard valley walls and other serious threats to the safety of the underwater autonomous vehicle [27]. So, path planning and obstacle avoidance are important components of autonomous navigation for AUV. The goal is to find a collision-free path from the start to the end in a complex underwater environment. Algorithm design is the core of path planning. The learning algorithms of artificial intelligence are regarded by the majority of researchers as the future of artificial intelligence. Neural network is an important content of machine learning. In the recent path planning of underwater robots, a large number of scholars used sensor data as network input and behavior and actions as network outputs; moreover, network models were obtained through training [28,29]. In paper [30], a 2D environment traversal path planning method based on biologically inspired neural networks was proposed. A recurrent neural network with convolution was developed [31] to improve the autonomous ability and intelligence of obstacle avoidance planning. Zhu et al. [32] focused on the study of sudden obstacles and used environmental changes to cause variations in neuron excitation and activity output values, thereby outputting collision-free path points. Reinforcement learning (RL) is an artificial intelligence algorithm that does not require prior knowledge and directly performs trial-and-error iterations with the environment to obtain feedback information to optimize strategies and is therefore widely used in mobile robot path planning in complex environments [33,34]. In paper [35], an adaptive neural network obstacle control method of AUVs with control input nonlinearities using RL was considered. In addition to improving accuracy, you can also learn control strategies from data to avoid cumbersome manual tuning parameters [36,37]. In 1989, Watkins [38] proposed a typical model-free RL algorithm called Q-learning algorithm. which is one of the most widely used algorithm in RL solutions [39]. Considering that the Q-learning algorithm [40] can guarantee convergence without knowing the model and can obtain good path planning in the case of a small state space, some scholars [41,42] have also applied it to the path planning of robots. However, for the research on underwater robot path planning and obstacle avoidance, such as large-dimensional and large state space, solving the optimal policy using Q-learning algorithm is difficult. Mnih et al. [43] proposed a deep reinforcement learning (DRL) algorithm based on Deep Q Network (DQN). The performance of this algorithm in many Atari games has reached the same level as that of humans; however, it cannot be applied directly to high-level dimensional continuous motion space control problem. Cheng et al. [44] proposed a DRL-based obstacle avoidance planning algorithm for underwater robots. Two convolutional layers in the algorithm structure extract the input state quantity features. The focus is on the distance to the target point, the distance to the obstacle, and the endpoint nearby speed and drift four-term return function (e.g., R distance, R collisions, R end, and R drift). However, no obvious advantage over traditional path planning algorithm was noted. Moreover, most researchers only considered the obstacle avoidance of static obstacles and some other works [45,46] presented applications where depth exploration in semi-static conditions could be improved; they seldom carried out real-time obstacle avoidance research on dynamic obstacles.

Lillicrap et al. [47] proposed the deep deterministic policy gradient (DDPG) algorithm based on DQN and DPG. This algorithm shows strong robustness and stability and performs well when processing high-dimensional continuous motion space control tasks. More than 20 complex control tasks have been implemented, but they have not been applied to the control of AUV path planning and obstacle avoidance. Therefore, the research on the path planning and obstacle avoidance in unknown underwater canyons for AUV based on the DDPG algorithm is carried out.

The remainder of this paper is organized as follows: In Section 2, four mathematical models required for AUV navigation in unknown underwater canyons are established. In Section 3, the path planning and obstacle avoidance algorithm are designed. In Section 4, the path planning and obstacle avoidance in unknown canyon simulation tests are discussed. In Section 5, the study is concluded.

2. Materials and Methods

2.1. Preliminaries

AUV-Kinematic Model with 3 Degrees of Freedom

A differential AUV model is constructed. In Figure 1, a 2-dimensional (2D) AUV model is shown in the geodetic-fixed frame ($\xi - E - \eta$). The length of the AUV model is 1.46 m, its mass is 45 kg, and the center of gravity coordinates in the body-fixed frame is $(0, 0)$. The AUV has seven range-finding sonars (In Figure 1, S_i $(i = 1, 2, 3, \ldots 7)$). These sonars can obtain obstacle information around the AUV. In addition, to facilitate research, the red dotted line in Figure 1 is used to indicate the sonar detection beam. The arrangement is shown in Figure 1. The sampling frequency of the range-finding sonars is 2 Hz, and their detection distance is 150 m. The stern of the robot has three propellers, $P_i (i = 1, 2, 3)$ with 0.2 m from the y-axis in the body-fixed frame. The propeller can generate 15 kg of force. The AUV performs an inertial navigation method for measuring velocity, position, and attitude.

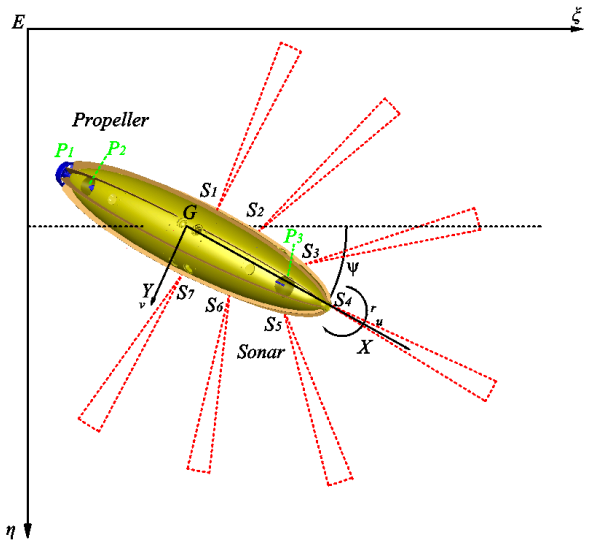


Figure 1. Major components of the autonomous underwater vehicles (AUV).

In this paper, the path planning and obstacle avoidance research of AUV are based on the kinematic model of AUV, because the route planned by combining the real movement process of AUV has the advantages of continuity and smoothness. An AUV usually moves in a 3-dimensional (3D) space with 6 degree of freedoms (DOFs), thereby resulting in the coupled dynamics in its planner and diving motions. To facilitate control design, the model is usually decoupled, whereas the designed control will be validated using the coupled nonlinear dynamics [44]. We consider the horizontal motion of the AUV with 3 degree of freedoms (DOFs) (Figure 1), which is described by the motion components as surge, sway, and yaw. On the basis of this consideration, $v = [u, v, r]^T \in R^3$ denotes the velocity vector, whereas $\eta = [x, y, \psi]^T \in R^3$ denotes the position vector. Let us denote the position coordinate of an AUV as (x, y) and the yaw as (ψ) in the earth-fixed inertial frame. The linear velocities in the body-fixed frame of the AUV $v = [u, v, r]^T \in R^3$ correspond

to surge, sway, and yaw. The horizontal maneuvering models [48] of the AUV can be expressed as:

$$\dot{\eta} = R(\psi)v \tag{1}$$

$$\dot{\psi} = r \tag{2}$$

where $R(\psi)$ is the rotation matrix for the horizontal motion of the AUV with three DOFs, which can be expressed as:

$$R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2.2. Obstacle Avoidance Strategy

This research aims to solve the problem on the safe navigation of AUVs in deep ocean, which is a complex and changeable environment with various mountains. When AUVs reach deep sea navigation, it faces many underwater canyons, and the hard valley walls threaten its safety seriously. In addition, other submersibles that navigate in the deep sea and moving marine life also threaten the safety of AUVs. The valley wall of the underwater canyon is a large-scale continuous obstacle relative to AUV, whereas other submersibles and moving marine organisms in deep sea navigation are dynamic obstacles of similar size to AUV. Thus, we should come up with different obstacle avoidance strategies. The following content involves the obstacle avoidance strategy proposed in this study for two different types of obstacle avoidance strategies.

2.2.1. Large-Scale Continuous Obstacle Avoidance Strategy of AUV

When an AUV goes to the deep sea, it faces many underwater canyons, and the hard valley wall threatens its safety seriously. In this study, a large area of the underwater canyon wall that AUV can only detect a very small part of this obstacle by sonars (less than 20 percent of its overall size) is regarded as a large-scale continuous obstacle, and the distance of the sensors from the center is negligible. The large-scale continuous obstacle strategy of AUV is presented as follows.

First, AUV is assumed to have the capability to measure the distance $D_i (i = 1, 2, 3, \ldots 7)$ and the azimuth angle ψ_w of the underwater canyon wall in the geodetic-fixed frame $(\xi - E - \eta)$ between the obstacle in the angle $\varsigma_i (i = 1, 2, 3, \ldots 7)$ of X axis direction by using the seven sonars installed on the left and right sides ahead of X axis.

Second, the AUV obstacle avoidance model in the face of a large-scale continuous obstacle is detailed as follows.

When the AUV drives into an unknown underwater canyon environment, in order to ensure the safety of the AUV itself, the AUV maintains the current dive depth and avoids the underwater canyon rock wall at the horizontal level. Therefore, in the current situation, the AUV kinematic model conforms to the Equation (1) (AUV 3-DOF kinematic model), and Equation (1) is rewritten to obtain the 3-DOF kinematic model of the AUV at time t to obtain Equation (4). Equation (4) into:

$$\begin{cases} \eta(t) = [x(t), y(t), \psi(t)]^T \\ \dot{\eta}(t) = R(\psi(t))V(t) \\ \dot{\psi}(t) = r(t) \\ V(t) = \sqrt{u(t)^2 + v(t)^2} \end{cases} \tag{4}$$

where $\eta(t)$ is the horizontal position vector in the coordinate method of the AUV at time t, including the horizontal position coordinates $x(t)$, $y(t)$, and the yaw angle $\psi(t)$; $\dot{\eta}(t)$ is the derivative of $\eta(t)$ with respect to time t; $\dot{\psi}(t)$ is the derivative of $\psi(t)$ with respect to time

t; $V(t)$ is the horizontal velocity vector of the AUV under the carrier at time t, including the horizontal velocity along the X axis longitudinal velocity $u(t)$ and the Y axis lateral velocity $v(t)$ and yaw angular velocity $r(t)$; $R(\psi(t))$ is the rotation matrix for the horizontal motion of the AUV with three DOFs at time t, which can be expressed as:

$$R(\psi(t)) = \begin{bmatrix} \cos(\psi(t)) & -\sin(\psi(t)) & 0 \\ \sin(\psi(t)) & \cos(\psi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Figure 2 shows the geometric relationship of AUV facing continuous obstacles in a wide range, such as walls and canyons.

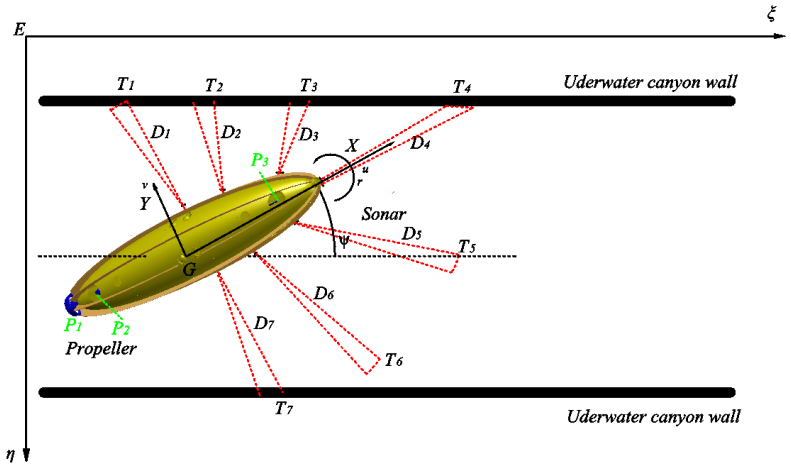


Figure 2. The geometric relationship of AUV facing continuous obstacles.

Continuous surface of obstacles, the bold part of the Figure 2 represents the triangle symbol of the AUV position, $T_i(i = 1, 2, 3, \ldots 7)$ represents the underwater sonar of autonomous navigation and the continuous obstacle detection on the surface of the intersection, $D_i(i = 1, 2, 3, \ldots 7)$ is AUV to $T_i(i = 1, 2, 3, \ldots 7)$ point distance, ψ is the AUV yaw angle, point 7 sonar and AUV body the angle between the axis X, which can be deduced from the geometric relationship in Figure 2.

$$\dot{D_i} = V_{T_i} \cos(\psi + \varsigma_i - \psi_w) - V \cos \varsigma_i \ (i = 1, 2, 3, \ldots 7) \tag{6}$$

$$\dot{\varsigma_i} = \mathrm{D_i}^{-1} \left[V_{T_i} \cos(\psi + \varsigma_i - \psi_w) - V \sin \varsigma_i \right] - \dot{\psi} \ (i = 1, 2, 3, \ldots 7) \tag{7}$$

where V_{T_i} $(i = 1, 2, 3, \cdots 7)$ is the moving speed of the intersection point T detected by the sensor along the continuous obstacle surface, V is the horizontal speed of the AUV, ψ_w is the azimuth angle of the underwater canyon wall in the geodetic-fixed frame $(\xi - E - \eta)$.

Because the angle $\varsigma_i(i = 1, 2, 3, \ldots 7)$ between the sonar pointing and the body axis X of the AUV is an inherent property of AUV, which is determined by the installation position and installation angle of the sonar, thus:

$$\dot{\varsigma_i} = 0 \ (i = 1, 2, 3, \ldots 7) \tag{8}$$

The combination of Equations (6) and (7) can solve the differential equation of the distance $D(1, 2, 3, \ldots 7)$ of the obstacle detected by sonar:

$$\dot{D_i} = \frac{D_i \dot{\psi}}{\tan(\psi + \varsigma_i - \psi_w)} + \frac{V \sin \varsigma_i}{\tan(\psi + \varsigma_i - \psi_w)} - V \cos \varsigma_i \ (i = 1, 2, 3, \ldots 7) \tag{9}$$

It can be seen from Equation (9) that the magnitude of $D(1, 2, 3, \ldots 7)$ is related to AUV's yaw angle ψ and the horizontal speed of the AUV V.

If the minimum safety distance of AUV is ρ_s and the sonar detection distances of the AUV is $D(1,2,3,\ldots 7)$, when the AUV faces a large range of continuous obstacles, such as walls of canyons, the condition for AUV not to collide is:

$$\min(D_i) \geq \rho_s \ (i = 1,2,3,\cdots 7) \tag{10}$$

where ρ_s is a positive constant.

Combining Equations (9) and (10), it can be known that AUV can ensure obstacle avoidance condition $\min(D_i) \geq \rho_s \ (i = 1,2,3,\ldots 7)$ by adjusting its own yaw angle ψ and the horizontal speed V, so that AUV can drive safely and autonomously in an unknown canyon environment.

2.2.2. Multi-Dynamic Obstacle Avoidance Strategy of AUV

AUVs will not only encounter static obstacles during the navigation in the deep sea but also often encounter dynamic obstacles, such as other underwater vehicles, marine life, and marine floating objects. In this paper, we only consider small-scale dynamic obstacles whose shape and size can be completely detected by the sonars of AUV. This research combines the idea of artificial potential field method to design a multi-dynamic obstacle avoidance strategy. Thus, AUVs need to reach the designated target area when navigating in the underwater environment to the target behavior. This research establishes the target behavior of AUV as a potential field function [15]:

$$U_1(x_t, y_t) = k_1 \sqrt{\left(x_t - x_{goal}\right)^2 + \left(y_t - y_{goal}\right)^2} \tag{11}$$

where k_1 is the negative constant, $\left(x_{goal}, y_{goal}\right)$ is the coordinate of the center position of the target area in the Cartesian coordinate method, and (x_t, y_t) is the coordinate of the center position of the AUV in the Cartesian coordinate method at time t.

Different from the scope of obstacle repulsion potential field established by artificial potential field method proposed by other researchers, the anticipating scope of AUV repulsion potential field is established in this study. When a dynamic obstacle enters into the repulsion field scope of AUV, the smaller the distance between them, the greater the repulsion force will be. On the contrary, the repulsion force of AUV is low. When the AUV changes course and speed to enable the obstacle to leave the scope of AUV repulsion potential field, the repulsion force received by AUV is 0. In this study, the behavior of AUV avoiding dynamic obstacles is established as the repulsion potential field function of AUV, as shown in Equation (12):

$$U_2(x_t, y_t) = \begin{cases} 0 & \text{if } (x'_t, y'_t) \notin \frac{4(x_t - x'_t)^2}{{L_1}^2} + \frac{4(y_t - y'_t)^2}{{L_2}^2} \leq 1 \\ k_2 \left(\frac{1}{\sqrt{\left(x_t - x'_t\right)^2 - \left(y_t - y'_t\right)^2 + 1}} \right) & \text{if } (x'_t, y'_t) \in \frac{4(x_t - x'_t)^2}{{L_1}^2} + \frac{4(y_t - y'_t)^2}{{L_2}^2} \leq 1 \end{cases} \tag{12}$$

where k_2 is the negative constant; (x_t, y_t) is the position coordinate of the AUV in the Cartesian coordinate method at time t, (x'_t, y'_t) is the position coordinate of the dynamic obstacle in the Cartesian coordinate method at time t; and L_1 and L_2 are the distance between the long axis and the short axis after expanding the AUV into an ellipsoid, respectively.

The shape of AUV is determined, and most of them are ellipsoidal, because the shape of the dynamic obstacles is uncertain. In this study, AUV is expanded to provide certain safety space. The specific treatment is shown in Equation (13):

$$\begin{cases} L_1 = \alpha \cdot L \\ L_2 = \alpha \cdot B \end{cases} \tag{13}$$

where α is the constant whose value is greater than 1; and L and B are the maximum length and width of the underwater vehicle, respectively.

In this study, $\alpha = 2.25$ is set. As shown in Figure 3, the blue ellipse represents security boundary of 2D in which the length of the major axis is $2.25L$ and the length of the minor axis is $2.25B$.

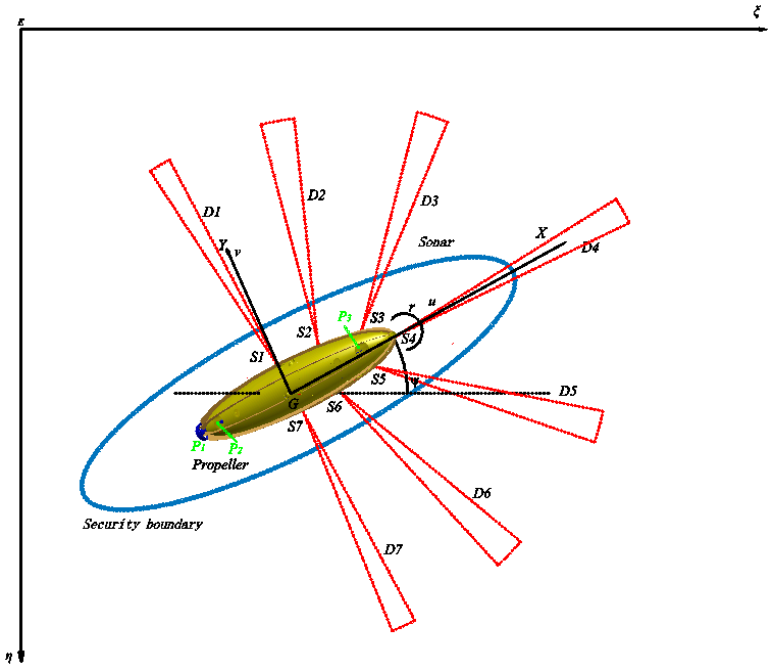


Figure 3. Scope of the repulsive force potential field of AUV.

In this study, the position information of dynamic obstacle and the relative distance information between dynamic obstacle distance and AUV are detected by the sonar of the AUV. When the sonar of the AUV detects dynamic obstacles, dynamic obstacle does not enter the scope of AUV's repulsive force field, and AUV is safe. Some processing, such as the kinematics model for predicting and estimating dynamic obstacles, is not the focus of this research and will not be explained. If dynamic obstacles enter the scope of the repulsion field of the AUV, then obstacles will be avoided by AUV and changes the heading angle and navigation speed continuously. If the dynamic obstacle is still within the scope of the repulsion potential field of the AUV and the distance between the dynamic obstacle and the AUV is less than the safe distance, AUV collides with obstacles.

2.3. MDP Model of AUV Path Planning

This section describes how to combine the contents of the first three sections of this chapter to generate a Markov Decision Process (MDP) model of AUV that can be trained in deep RL.

MDP is a classic form of sequential decision-making and is usually used to model RL problems [49]. MDP is composed of four tuples $MDP = (S, A, P, R)$, where A represents the action set, S represents the state set, $P : S \times A \times S \to (0, 1)$ represents the state transition probability, and R is the reward function. Moreover, AUV interacts with the environment (Markov Decision Process is shown in Figure 4). After receiving the status information at time t, the AUV outputs the action $\mu_t \in A$. The reward value generated at time t is $R_t = f(s_t, a_t, s_{t+1})$, and the state becomes s_{t+1}. The action μ_t outputted by the agent is determined by the policy π, which is the probability that the state s_t is mapped to each action: $S \to P(A)$.

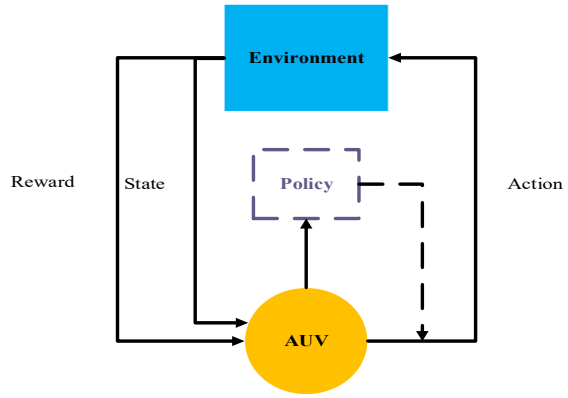


Figure 4. Reinforcement learning (RL) mechanism.

The MDP model of AUV path planning in this paper first follows two assumptions:

(1) First, the path planning task is horizontal with three degrees of freedom;
(2) Second, time is discretized; to meet the real-time requirements, the planning method outputs at regular intervals with a sampling rate of $T_S = 0.5$ s.

As introduced in Section 2.1, the AUV used in this research is equipped with 7 obstacle avoidance sonar sensors to detect the distance to obstacles or targets in real time. Sonars are the direct device for AUV to interact with the environment. Therefore, in this study, the 7 obstacle avoidance sonars detected by the AUV at time t are the value $D_i(t)$ $(i = 1, 2, 3, \ldots 7)$ of the distance to the obstacle or target, which is set as the state space S of the MDP model of AUV path planning, and the detection capability of the sonar sensor is limited as state constraints. Hence, the detection distance range is $[0, 150\text{ m}]$, the state constraint is set to $[0, 150\text{ m}]$, and the final state set expression at time t is shown in Equation (14):

$$s_t = \left\{ s_t^1, s_t^2, s_t^3, s_t^4, s_t^5, s_t^6, s_t^7 \right\} = \{D_1(t), D_2(t), D_3(t), D_4(t), D_5(t), D_6(t), D_7(t)\} \quad (14)$$

where s_t represents the state set at time t, including 7 values of s_t^i $(i = 1, 2, 3, \ldots 7)$, corresponding to the distance $D_i(t)$ $(i = 1, 2, 3, \ldots 7)$ between AUV and surrounding obstacles or target detected by 7 sonars at time t.

The state set S of the MDP model for AUV path planning is shown in Table 1.

Table 1. State sets.

State	Sonar Detection Results	Range of Values (m)
s_t^1	sonars 1 $D_1(t)$	$[0, 150]$
s_t^2	sonars 2 $D_2(t)$	$[0, 150]$
s_t^3	sonars 3 $D_3(t)$	$[0, 150]$
s_t^4	sonars 4 $D_4(t)$	$[0, 150]$
s_t^5	sonars 5 $D_5(t)$	$[0, 150]$
s_t^6	sonars 6 $D_6(t)$	$[0, 150]$
s_t^7	sonars 7 $D_7(t)$	$[0, 150]$

According to the 3-degree-of-freedom kinematics model of the AUV in Section 2.1, the action space A of the MDP model in this study is defined as the yaw angular velocity $\omega(t)$ and the horizontal velocity vector $V(t)$, and the action space constraints correspond to the limitations of its own maneuverability. The AUV used in this research has three thrusters, one tail thruster and two side thrusters, which can realize turning and forward and backward, respectively. Therefore, the heading angle range is $[-180^\circ, +180^\circ]$, and considering the limitation of its own maneuverability, the yaw rate range is $\left[-1.0\text{ rad}^{-1}, 1.5\text{ rad}^{-1}\right]$, and the sailing speed range is $[-1.0\text{ m/s}, 1.5\text{ m/s}]$. (x_t, y_t) is the position coordinate of the

AUV at time t in the geodetic-fixed frame ($\xi - E - \eta$). Thus, the action set a_t at time t is shown in Equation (15):

$$a_t = \left\{a_t^1, a_t^2, a_t^3, a_t^4\right\} = \{\omega(t), V(t), x(t), y(t)\} \tag{15}$$

The action set A of the MDP model for AUV path planning is shown in Table 2, $[x_{\min}, x_{\max}]$ and $[y_{\min}, y_{\max}]$ are the horizontal and vertical limits of the AUV driving range in the geodetic coordinate system, respectively.

Table 2. Action sets.

Action	Type	Range of Values
a_t^1	Yaw angular velocity $\omega(t)$	$\left[-1.0\ \text{rad}^{-1}, 1.0\ \text{rad}^{-1}\right]$
a_t^2	Speed $V(t)$	$[-1.0\ \text{m/s}, 1.5\ \text{m/s}]$
a_t^3	$x(t)$	$[x_{\min}, x_{\max}]$
a_t^4	$y(t)$	$[y_{\min}, y_{\max}]$

In this study, large-scale continuous static obstacle avoidance strategies proposed in Section 2.2.1 of this chapter and multiple dynamic obstacle avoidance strategies proposed in Section 2.2.2 of this chapter are integrated into the specific setting of reward values of the MDP model of deep RL, which will be introduced in Section 3.3. Finally, $P : S \times A \times S \to (0, 1)$ of the MDP model represents the state transition probability, and it is updated through DRL algorithm.

3. SumTree-DDPG Algorithm

This study uses RL methods based on DDPG. Unlike traditional value-based RL, this method can search for strategies directly. Therefore, it can be applied to a continuous high-dimensional action space. DDPG is an actor-critic algorithm. This section introduces the critic, actor, reward function, and replay memory in four aspects and proposes an improved DDPG algorithm (SumTree-DDPG) for AUV path planning and obstacle avoidance.

3.1. Critic

The critic is used to fit the state action value function, including the target Q network and the online Q network, and the two networks are updated alternately. The initial parameters of the two networks, $\theta^{Q'}$ and θ^{Q}, are equal. After the random sampling of small batch data $N(s_i, a_i, r_i, s_{i+1})$ from the experience buffer pool, the online value Q network is updated by minimizing the loss value L. The calculation of L is shown in Equation (16).

$$\mathrm{L}\left(\theta^{Q}\right) = E_{s_t \sim \rho^{\beta}, a_t \sim \mu, r_t \sim E}[(Q_{T\text{arget}} - Q\left(s_t, a_t \middle| \theta^{Q}\right))^2] \tag{16}$$

In Equation (16), Target $Q_{T\text{arget}}$ refers to the target Q value, as shown in Equation (17).

$$\mathrm{Q}_{T\text{arget}} = \mathrm{r}(s_t, a_t) + \gamma Q'\left(s_{t+1}, \mu(s_{t+1}|\theta^{\mu}) \middle| \theta^{Q'}\right) \tag{17}$$

Different from the real-time update of online Q network, the target Q network is updated every other period of time, and its update method is shown in Equation (18).

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1 - \tau)\theta^{Q'} \tag{18}$$

where τ is a preset constant.

3.2. Actor

In the DDPG algorithm, a policy network with the parameter θ^{μ} is used to represent the deterministic policy $a = \mu(s|\theta^{\mu})$. The actor is used to fit the policy function. Its main

task is to output the deterministic action value t for the input state s_t. The update of online policy network parameters is shown in Equation (19).

$$\nabla_{\theta^\mu}\left(J_\beta(\mu)\right) \approx \frac{1}{N}\sum_i\left(\nabla_a Q\left(s,a\middle|\theta^Q\right)\Big|_{s=s_i,a=\mu(s_i)}\cdot\nabla_{\theta^\mu}\mu(s|\theta^\mu)\Big|_{s=s_i}\right) \tag{19}$$

In Equation (20), the state s follows the ρ^β distribution, and θ^μ is the online policy network parameter. The target policy network is updated in the same way as the target Q network, and it is updated every once in a while as Formula (20):

$$\theta^{\mu'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{\mu'} \tag{20}$$

In Equation (20), the parameter τ is a preset constant.

3.3. Reward Function Design

The reward function plays an important role in RL tasks, and it points to the direction of the network parameter update of actors and critics [50]. The reward function of this study is mainly designed according to the large-scale continuous static obstacle avoidance strategies in Section 2.2.1 and multiple dynamic obstacle avoidance strategies in Section 2.2.2. Aiming at the AUV obstacle avoidance problem, this paper designs a reward function algorithm that considers the three aspects of goal, safety, and stability.

AUV's tendency toward target behavior is reflected in the reward value of target module $r_1(s_t,a_t,s_{t+1})$. This study combines the gravitational potential field function in Section 2.2.2 to set the reward value of the target module of the first component of the reward value. The target module reward value function $r_1(s_t,a_t,s_{t+1})$ is designed as follows:

$$r_1(s_t,a_t,s_{t+1}) = -0.001\times\sqrt{\left(x_t-x_{goal}\right)^2+\left(y_t-y_{goal}\right)^2} \tag{21}$$

where $\left(x_{goal},y_{goal}\right)$ is the coordinate of the center position of the target area in the Cartesian coordinate method; and (x_t,y_t) is the coordinate of the center position of the AUV in the Cartesian coordinate method at time t.

When the AUV reaches the target area, the reward value of the target module will be updated:

$$r_1(s_t,a_t,s_{t+1}) \leftarrow r_1(s_t,a_t,s_{t+1}) + R_1 \tag{22}$$

where R_1 is a normal number.

The AUV's obstacle avoidance behavior is set as the safety module reward value $r_2(s_t,a_t,s_{t+1})$. The obstacles considered in this study include large-scale continuous static obstacle and multiple dynamic obstacles. According to Section 2.2.1, it is proposed that the distance between the seven sonars controlling AUV and the large-scale continuous static obstacles detected is always greater than or equal to the safe radius of AUV, so that the large-scale continuous static obstacles can be avoided. According to Section 2.2.2, the method of setting the scope of repulsion potential field of AUV is proposed to avoid collision with dynamic obstacles. Combined with the two obstacle avoidance strategies, the second component of the safety module reward value $r_2(s_t,a_t,s_{t+1})$ is shown in the Equation (23):

$$r_2(s_t,a_t,s_{t+1}) = r_2^1(s_t,a_t,s_{t+1}) + r_2^2(s_t,a_t,s_{t+1}) \tag{23}$$

where $r_2(s_t,a_t,s_{t+1})$ is the reward value of the safety module; and $r_2^1(s_t,a_t,s_{t+1})$ is the first component of the safety module $r_2(s_t,a_t,s_{t+1})$, which is used to avoid large-scale continuous static obstacles. $r_2^2(s_t,a_t,s_{t+1})$ is the second component of safety module $r_2(s_t,a_t,s_{t+1})$, which is used to avoid small dynamic obstacles.

The specific process set by $r_2^1(s_t,a_t,s_{t+1})$ is when the minimum detection distance $\min(D_i(t))$ $(i=1,2,3,\ldots 7)$ of the 7 sonar probes of AUV is twice longer than the safe distance r_s at time step t, indicating that AUV is safe and the reward value $r_2^1(s_t,a_t,s_{t+1})$

is 0. When $1.0r_s \leq \min(D_i(t)) \leq 2.0r_s$ $(i = 1, 2, 3, \ldots 7)$ is true, then the AUV is about to collide the large-scale continuous static obstacle and obtain the continuous negative reward $-(\min(D_i(t)) - r_s)^2$ $(i = 1, 2, 3, \ldots 7)$; when $\min(D_i(t)) \leq 1.0r_s$ $(i = 1, 2, 3, \ldots 7)$ is less than the safe distance, it means that AUV collides with the large-scale continuous static obstacle and obtains the negative reward $-R_2$. Therefore, the expression of $r_2^1(s_t, a_t, s_{t+1})$ is:

$$r_2^1(s_t, a_t, s_{t+1}) = \begin{cases} -\mathrm{R}_2 & \text{if } \min(D_i(t)) \leq 1.0r_s \ (i = 1, 2, 3, \ldots 7) \\ -0.01 \times (\min(D_i(t)) - r_s)^2 & \text{if } \min(D_i(t)) \leq 2.0r_s \ (i = 1, 2, 3, \ldots 7) \\ 0 & \text{if } \min(D_i(t)) > 2.0r_s \ (i = 1, 2, 3, \ldots 7) \end{cases} \tag{24}$$

where $\min(D_i(t))$ $(i = 1, 2, 3, \ldots 7)$ is the minimum detection distance of the 7 sonar probes of AUV between AUV and the large-scale continuous static obstacle at time step t; r_s is the set safety margin; and R_2 is a normal number.

The specific process set by $r_2^2(s_t, a_t, s_{t+1})$ is when the 7 sonars of AUV detect that the dynamic obstacle does not enter the repulsion area of AUV, indicating that AUV is safe and the reward value is 0. When the dynamic obstacle enters the repulsion area of AUV, then the AUV will get a continuous negative reward, and the closer the distance between the obstacle and AUV, the more negative reward it will get. If the dynamic obstacle finally reaches the safe radius of AUV, then the two collide, and the negative reward value $-R_2$ will be obtained. Therefore, the expression of $r_2^2(s_t, a_t, s_{t+1})$ is:

$$r_2^2(s_t, a_t, s_{t+1}) = \begin{cases} 0 & \text{if } (x_t', y_t') \notin \frac{4(x_t - x_t')^2}{{L_1}^2} + \frac{4(y_t - y_t')^2}{{L_2}^2} \leq 1 \\ -0.1 \times \left(\frac{1}{\sqrt{(x_t - x_t')^2 - (y_t - y_t')^2 + 1}} \right) & \text{if } (x_t', y_t') \in \frac{4(x_t - x_t')^2}{{L_1}^2} + \frac{4(y_t - y_t')^2}{{L_2}^2} \leq 1 \\ -R_1 & \text{if } (x_t', y_t') \in (x_t - x_t')^2 + (y_t - y_t')^2 \leq {r_s}^2 \end{cases} \tag{25}$$

where (x_t, y_t) is the position coordinate of the AUV in the Cartesian coordinate method at time t; (x_t', y_t') is the position coordinate of the dynamic obstacle in the Cartesian coordinate method at time t; L_1 and L_2 are the distance between the long axis and the short axis after expanding the AUV into an ellipsoid, respectively; r_s the set safety margin; and R_2 is a normal number.

To improve the robustness of the AUV obstacle avoidance method and enhance the ability of the AUV to maintain the heading and speed when it is in a safe local area and approaching the target point, this paper designs the stability reward value function as follows.

$$r_3(s_t, a_t, s_{t+1}) = -0.01 \times (|\omega_{t+1} - \omega_t| + |v_{t+1} - v_t|) \tag{26}$$

where $r_3(s_t, a_t, s_{t+1})$ represents the first component of current interference stability module value reward of the total reward $r(s_t, a_t, s_{t+1})$ for time step t time t; ω_t and ω_{t+1} of the Formula (26) represent respectively the current moment and the next moment of AUV's yaw angular velocity; v_t and v_{t+1} of the Formula (26) represent respectively the current moment and the next moment of AUV speed.

In this paper, the reward value function used for AUV path planning and obstacle avoidance is shown in Equation (27).

$$r(s_t, a_t, s_{t+1}) = \tau_1 r_1(s_t, a_t, s_{t+1}) + \tau_2 r_2(s_t, a_t, s_{t+1}) + \tau_3 r_3(s_t, a_t, s_{t+1}) \tag{27}$$

where τ_1, τ_2, and τ_3 are the weights of various factors.

The larger the value, the more the trained model focuses on this factor. The specific value needs to be set according to the specific environment and requirements. The algorithm pseudo code of reward function for AUV obstacle avoidance is shown in Algorithm 1.

Algorithm 1. Reward Algorithm for AUV Obstacle Avoidance

1: Initialize reward value $r(s_t, a_t, s_{t+1}) = 0$
2: Take action a_t and observe s_{t+1}
4: Get the stability reward value function $r_3(s_t, a_t, s_{t+1})$:
$r_3(s_t, a_t, s_{t+1}) \leftarrow r_3(s_t, a_t, s_{t+1}) - 0.01 \times (|\omega_{t+1} - \omega_t| + |v_{t+1} - v_t|)$
3: **if** transition from safe region to safe region
4: then the reward value of the target module $r_1(s_t, a_t, s_{t+1})$:
$r_1(s_t, a_t, s_{t+1}) \leftarrow r_1(s_t, a_t, s_{t+1}) - 0.001 \times \sqrt{\left(x_t - x_{goal}\right)^2 + \left(y_t - y_{goal}\right)^2}$
where $\left(x_{goal}, y_{goal}\right)$ is the coordinate of the center position of the target area;
(x_t, y_t) is the coordinate of the center position of the AUV at time t
5: **else** transition from safe region to unsafe region
6: **if** AUV encounters large-scale continuous obstacle
7: **then** the safety module reward value $r_2(s_t, a_t, s_{t+1})$: $r_2(s_t, a_t, s_{t+1}) \leftarrow r_2^1(s_t, a_t, s_{t+1})$
8: **else if** AUV encounters multi-dynamic obstacle
9: **then** $r_2(s_t, a_t, s_{t+1}) \leftarrow r_2^2(s_t, a_t, s_{t+1})$
10: **else if** transition from unsafe region to obstacle region
11: **then** $r_2(s_t, a_t, s_{t+1}) \leftarrow r_2(s_t, a_t, s_{t+1}) - R_2$ and restart the exploration
12: **else** transition from unsafe region to safe region
13: **then** $r_2(s_t, a_t, s_{t+1}) = 0$
14: **if** transition from safe region to goal region
15: **then** $r_1(s_t, a_t, s_{t+1}) \leftarrow r_1(s_t, a_t, s_{t+1}) + R_2$
16: $r(s_t, a_t, s_{t+1}) \leftarrow r(s_t, a_t, s_{t+1}) + \tau_1 r_1(s_t, a_t, s_{t+1}) + \tau_2 r_2(s_t, a_t, s_{t+1}) + \tau_3 r_3(s_t, a_t, s_{t+1})$.
where $0 \leq \tau_1 \leq 1$, $0 \leq \tau_2 \leq 1$, and $0 \leq \tau_3 \leq 1$ are the weights of various factors
17: **end**

3.4. Replay Memory

The DDPG algorithm uses the experience replay method to store the experience samples generated by the agent's interaction with the environment in the experience buffer pool and randomly sample samples from it to train the network. This method of randomly sampling samples neither considers the different importance of different data, nor does fully consider the diversity of the samples to be drawn, resulting in slower model convergence. To solve this problem, the sample storage and extraction strategy in this paper are to take the method of priority extraction according to the importance of the data, which effectively improves the convergence speed of the model.

In this article, the small batch sample sampling is not random sampling, but according to the sample priority in the memory bank. So this can more effectively find the samples we need to learn. In the DDPG algorithm, the parameters of the strategy network depend on the selection of the value network, and the parameters in the value network are determined by the loss function of the value network. So the sample priority P can be defined by the expectation of the difference between the target Q value and the actual Q value. The greater the difference between the target Q value of the value network and the actual Q value, the greater the prediction accuracy of the network parameters, that is, the more the sample needs to be learned, that is, the higher the priority P. With priority P, this article uses the SumTree method to effectively sample based on P. The SumTree method does not sort the obtained samples, which reduces the computing power compared to the sorting algorithm.

SumTree is a tree structure (Figure 5), the priority P of the sample is stored in the leaf node, and each leaf node corresponds to an index value. Using the index value, the corresponding sample can be accessed. Every two leaf nodes correspond to a parent node of an upper level. The priority of the parent node is equal to the sum of the priorities of the left and right child nodes. Thus, the top of the SumTree is $sum(P)$.

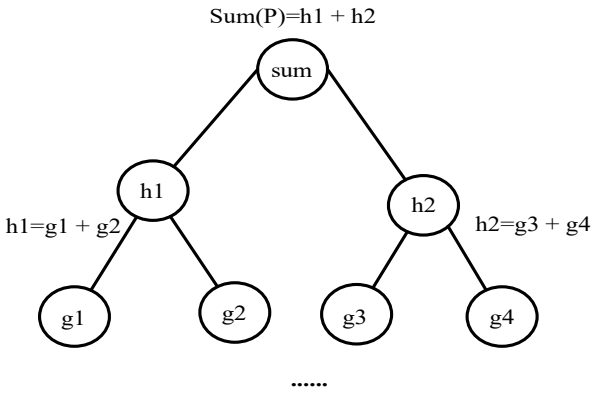


Figure 5. SumTree structure diagram.

When sampling, this study first divides the priority of the root node (the sum of the priority of all leaf nodes) by the number of samples N and divides the priority from 0 to the sum of priority into N intervals. Then, a number is randomly selected in each interval. Because nodes with higher priority will also occupy a longer interval, the probability of being drawn will also be higher, thus achieving the purpose of priority sampling. Each time a leaf node is drawn, its priority and corresponding sample pool data are returned. N samples $\left(s_i^k, a_i^k, r_i^k, s_{i+1}^k\right)$, $k = 1, 2, \ldots N$ are collected from SumTree, and the sampling probability and weight of each sample are shown in the following Equations (28) and (29), respectively.

$$P(k) = \frac{p_k}{\sum\limits_m p_m} \tag{28}$$

$$\omega^k = \left(\frac{P(k)}{\min_j(P(j))}\right)^{-\beta} \tag{29}$$

By improving DDPG experience replay and combining with algorithm 1, the algorithm for AUV path planning and obstacle avoidance is obtained, which we call SumTree-DDPG (Algorithm 2).

Algorithm 2. SumTree-DDPG Algorithm

Randomly initialize critic network $Q\left(s, a\middle|\theta^Q\right)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize the SumTree and define the capacity size $H = \phi$
for episode $= 1, M$ **do**
  Initialize a random process N for action exploration
  Receive initial observation state s_1
  for $step = 1, T$ **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + N_t$ according to the current policy and exploration noise
    Take action a_t and observe s_{t+1}
  Decide reward $r_t(s_t, a_t, s_{t+1})$ using Algorithm 1
  Store transition (s_t, a_t, r_t, s_{t+1}) in SumTree $H = \phi$
  do
Sample a minibatch of N transitions $\left(s_t^k, a_t^k, r_t^k, s_{t+1}^k\right)$, $k = 1, 2, \ldots N$ from SumTree $H = \phi$
with probability-sampling: $P(k) = p_k / \sum\limits_m P(m)$
with importance-sampling weight: $\omega_k = \left(P(k)/\min_j P(j)\right)^{-\beta}$
while $len(Data) > N$

Algorithm 2. *Cont.*

13: Set $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}\middle|\theta^{\mu'}\right)\middle|\theta^{Q'}\right)$
14: Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i|\theta^Q\right)\right)^2$
15: Update the actor policy using the sampled policy gradient:
$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q\left(s, a|\theta^Q\right)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{si}$
16: Update the target networks:
$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$. $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
17: Update transition priority: $p_k \leftarrow \left|\delta_j\right|$
19: **end for**
20: **end for**

In addition, the structure diagram of the SumTree-DDPG algorithm applied to AUV online path planning is shown in Figure 6.

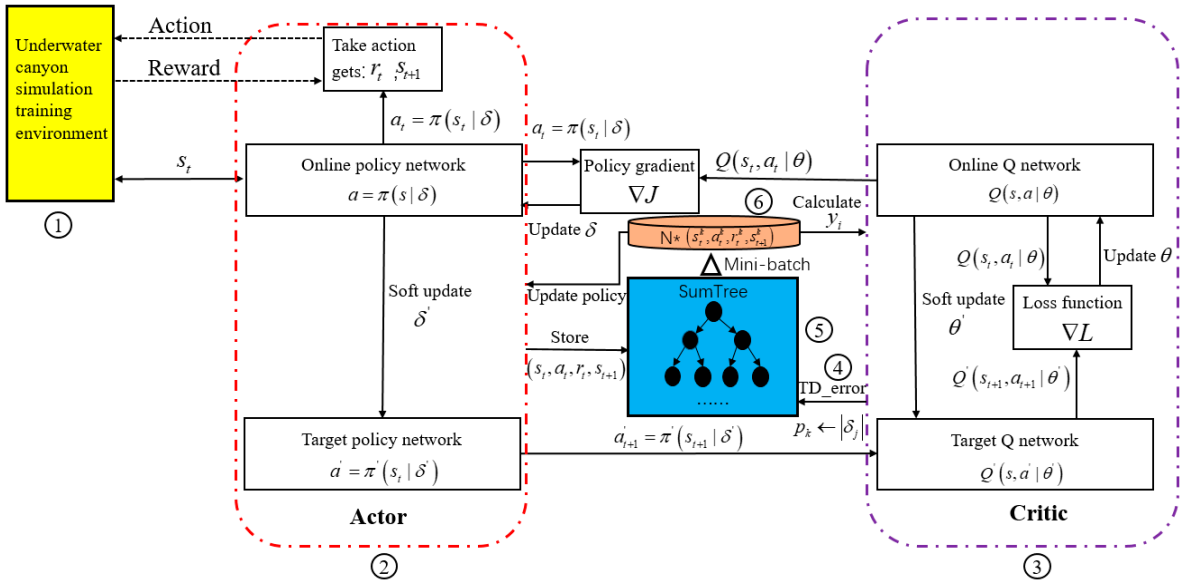


Figure 6. SumTree-deep deterministic policy gradient (SumTree-DDPG) algorithm structure diagram. (1): Under canyon simulation training environment is established by the information detected by the sonars of AUV. (2): Actor is a double-layer fully connected neural network structure composed of online policy network and Target policy network, which is used to obtain the optimal action output by interacting directly with under canyon simulation training environment. (3): Critic is a double-layer neural network structure composed of online Q network and target Q network, which is used to evaluate the action of Actor's selection. (4): Temporal-difference calculated by Critic. (5): Experience replay memory with SumTree-structure. (6): n samples with proportional prioritization.

4. Simulations

To verify the feasibility of the method proposed in this paper, first, this study used the python programming language to build two underwater canyon simulation test environments based on the pyglet module. Then, the DDPG algorithm and SumTree-DDPG algorithm are applied to the path planning and obstacle avoidance of AUV for comparative analysis, respectively. According to the principle of control variable method, the same simulation environment is used in both cases.

4.1. Simulation of AUV Crossing Unkonwn Underwater Canyon

First of all, the simulation training process of the AUV in this research is based on four hypotheses:

(1) Assumption 1: When the AUV is trained in the underwater canyon simulation environment, the details of the AUV model have negligible influence on the generation of obstacle avoidance paths;
(2) Assumption 2: The effects of environmental disturbances, such as deep ocean currents, are ignored;
(3) Assumption 3: AUV avoid obstacles on the horizontal obstacle avoidance;
(4) Assumption 4: The next state of AUV is only related to the current state, and the condition distribution of the next state does not change with time based on the current state. The obstacle avoidance process of AUV is established as the MDP model in Section 2.3.

This study constructed two unknown 2D underwater simulation environments to simulate AUV traveling through unknown underwater canyon at a fixed depth by using the Python language compiler in a high-performance computer, as shown in Figure 7, namely, Environment 1: an unknown underwater environment with irregular narrow terrain, to simulate AUV driving in an unknown underwater canyon; Environment 2, as shown in Figure 8, is the addition of some small-scale dynamic obstacles represented by blue squares in the Figure 8 on the basis of Environment 1 to simulate other vehicles or Marine life traveling in the underwater canyon. The simulation environment size is 1000×300 m. In Figures 7 and 8, the black irregular blocks represent the walls of unknown underwater canyon, the green square represents the target, and the red rectangle represents the AUV. The solid black line around the AUV simulates the sound detection beam of obstacle avoidance. The initial position of AUV is at the map coordinate point $(980\text{ m}, 125\text{ m})$, and the target center position is $(30\text{ m}, 100\text{ m})$. Environment 1 is mainly used to verify the planning ability of AUV obstacle avoidance algorithm to avoid unknown large-scale continuous static obstacles (e.g., the walls of underwater canyon) and reach distant target points. Dynamic obstacles added in Environment 2 are uniform linear motion and uniform acceleration linear motion. Moreover, the red dotted line in Figure 8 represents the motion trajectories of dynamic obstacles. This environment is further used to verify the planning ability of AUV obstacle avoidance algorithm in the case of abrupt dynamic obstacles.

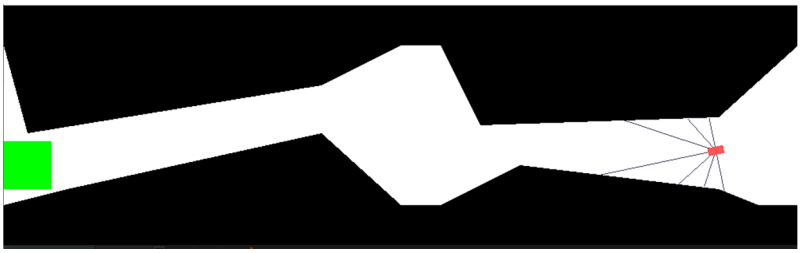


Figure 7. Environment 1. ■: The walls of unknown underwater canyon; ■: Target area; ■: AUV.

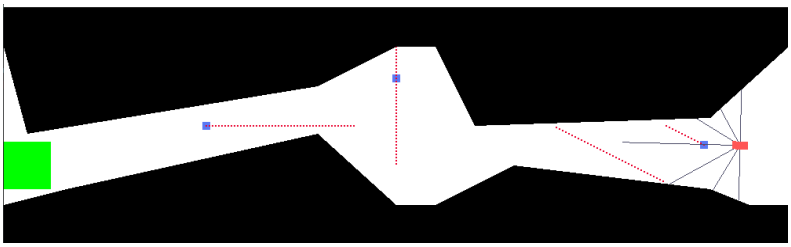


Figure 8. Environment 2. ■: The walls of unknown underwater canyon; ■: Target area; ■: Dynamic obstacle; ■: AUV.

After setting up AUV simulation environments 1 and 2, this study combines the DDPG algorithm and the SumTree-DDOG algorithm with the AUV model in Chapter 2 to establish two AUV path planning algorithm: DDPG algorithm AUV path planning method

and SumTree-DDPG algorithm AUV path planning method. Finally, the pros and cons of the two planning methods are analyzed by simulation.

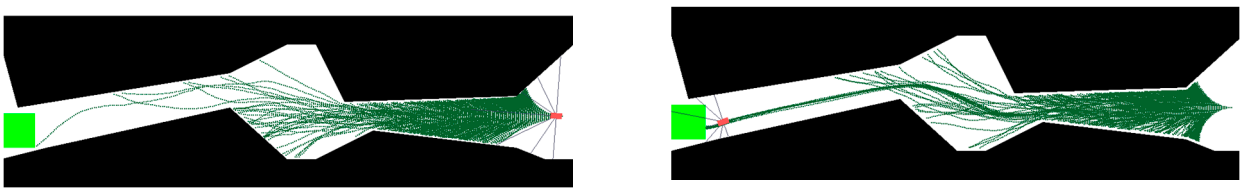
4.2. Large-Scale Continuous Obstacle Avoidance Simulations Results

First of all, in Environment 1: Single-target, the walls of unknown underwater canyon are regarded as large-scale continuous static obstacles to AUV, two planning methods were tested. During the test, the principle of controlling variables is always maintained, that is, the basic parameters of the two planning methods are consistent (Table 3).

Table 3. DDPG or SumTree-DDPG's hyperparameter in environment 1.

Hyperparameter	Value
Learning rate for Actor α_1	0.0001
Learning rate for Critic α_2	0.0001
Discount factor γ	0.99
Initial exploration ε	1
Replay memory size ϕ	100,000
Minibatch size m	32
Soft-update frequency f	0.01
Final exploration frame T	2000
Max episodes M	1000

Figure 9 shows the historical trajectory of the AUV online pathfinding process of the two planners (The green line is the AUV motion trajectory). Both methods can plan AUV to avoid large-scale continuous static obstacles and reach the target area. Qualitatively, SumTree-DDPG has a better planning effect than DDPG algorithm for the movement path of the target.



(**a**) (**b**)

Figure 9. Simulation results of AUV in Environment 1 (**a**) DDPG algorithm AUV path planning method. (**b**) SumTree-DDPG algorithm AUV path planning method. ■: The walls of unknown underwater canyon; ■: Target area; ■: AUV.

Table 4 shows that for 1000 rounds (2000 steps per round), the number of times the AUV successfully reached the target area during the DDPG planner training process is 1, and the DDPG planner does not converge to the optimal path in 1000 episodes. As shown in Figure 10, the optimal path planned by the DDPG planner in environment 1 is 1263.50 m. The SumTree-DDPG planner also runs 1000 rounds in the same environment (2000 steps per round). During online path planning process, the number of times the AUV successfully reaches the target area is 218, and the program runs to 796 episodes of convergence and this method converges to the optimal path. The optimal path planned is 1128.50 m. Compared with the DDPG planner, when the SumTree-DDPG planner AUV is trained in environment 1, the success rate is increased from 0.10% to 21.80%, the number of collisions with obstacles is reduced, the training efficiency is improved, and the planned collision-free optimal path is shorter.

Table 4. Performance evaluation in the training environment 1 over 1000 episodes.

Algorithm	Numbers of Successful Hits to Target	Min-Episode of Convergence	Optimal Path (m)
DDPG	1	791	1263.5
SumTree-DDPG	218	796	1128.5

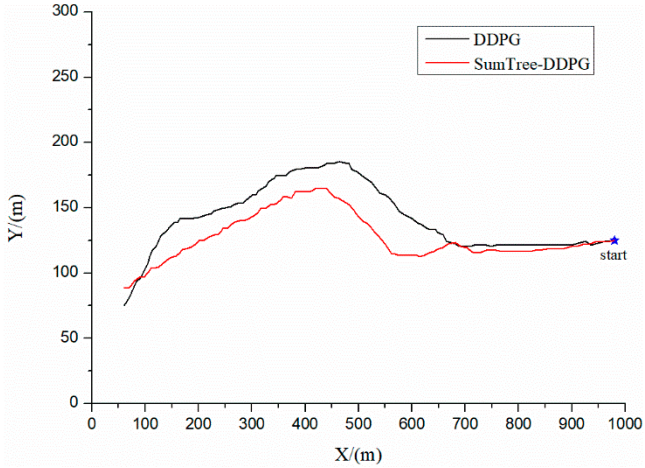


Figure 10. The optimized path of AUV in Environment 1.

Figure 11 shows the average cumulative reward curves obtained in 1000 rounds of the two algorithms. Figure 11 depicts that the stability, training's total reward, and robustness of the SumTree-DDPG planner are better than those of the DDPG planner.

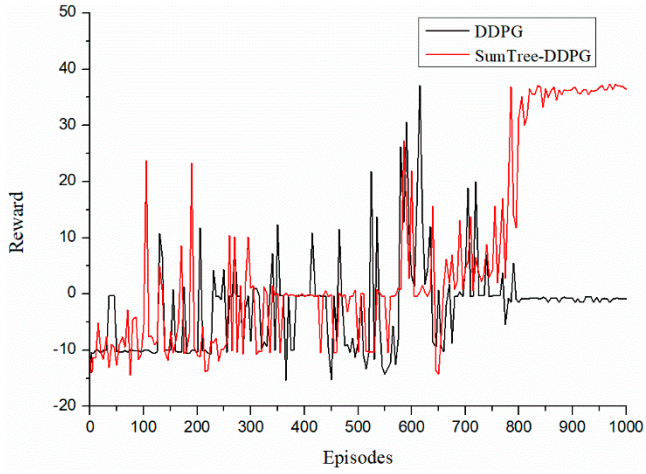


Figure 11. Total reward per episode of AUV training in Environment 1.

4.3. Dynamic Obstacle Avoidance Control Simulations Results

To consider the influence of dynamic obstacles, such as other ships or underwater floats, on AUV's travel in underwater canyons, we designed Environment 2 based on Environment 1. Environment 2 refers to the addition of some small-scale dynamic obstacles on the basis of Environment 1, whose shape and size can be detected by sonars to simulate other underwater vehicles or marine life that travels in the underwater canyon. In Environment 2, the single-target underwater unknown environment with large-scale continuous static obstacle, some small-scale dynamic obstacles, and two planning methods were tested. In addition, during each episodes of online training, dynamic obstacles will move along the four red dotted lines in Figure 10 and specific parameter information of dynamic obstacles in the geodetic-fixed frame is shown in Table 5.

Table 5. Dynamic obstacles parameter information.

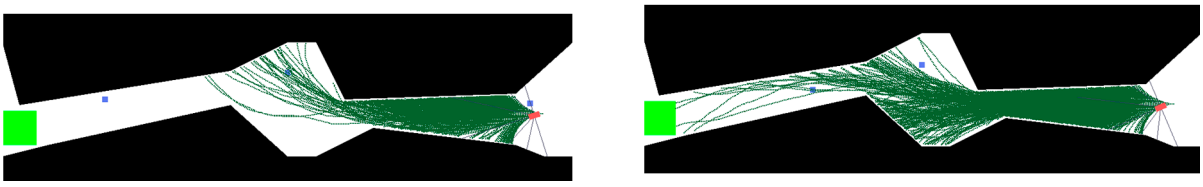
Obstacle	Start Point $\vec{x_0}$ (m)	End Point $\vec{x_{end}}$ (m)	Initial Velocity $\vec{V_0}$ (m/s)	Acceleration $\vec{a}$ (m^2/s)
1	(450,150)	(150,150)	(1.0,0)	(0.05,0)
2	(500,250)	(500,100)	(0,−2.0)	(0,−0.01)
3	(700,150)	(840,80)	(2.0,−1.0)	(0,0)
4	(840,150)	(980,80)	(2.0,−1.0)	(0,0)

During the test, the principle of controlling variables is always maintained, that is, the basic parameters of the two planning methods are consistent (Table 6).

Table 6. DDPG or SumTree-DDPG's per parameter in environment 2.

Hyperparameter	Value
Learning rate for Actor α_1	0.0001
Learning rate for Critic α_2	0.0001
Discount factor γ	0.99
Initial exploration ε	1
Replay memory size ϕ	200,000
Minibatch size m	32
Soft-update frequency f	0.01
Final exploration frame T	2000
Max episodes M	1000

Figure 12 shows the historical trajectory of the AUV online pathfinding process of the two planners (the green line is the AUV motion trajectory). It can be seen from the figure that the DDPG planner method failed to find a collision-free path to the target area. Only the SumTree-DDPG planner method successfully finds multiple safe and collision-free paths in an unknown underwater canyon with multiple dynamic obstacles, and the AUV can safely reach the target area by driving along these paths. Qualitatively, SumTree-DDPG planner has a better planning effect than DDPG algorithm for the movement path of the target in environment 2.



(**a**) (**b**)

Figure 12. Simulation results of AUV in Environment 2 with CV Model. (**a**) DDPG algorithm AUV path planning method. (**b**) SumTree-DDPG algorithm AUV path planning method. ■: The walls of unknown underwater canyon; ■: Target area; ■: Dynamic obstacle; ■: AUV.

Table 7 shows that for 1000 rounds (2000 steps per round), the number of times the AUV successfully reached the target area during the DDPG planner training process is 0, and the program does not converge in 1000 episodes. As shown in Figure 13, the DDPG planner did not find a safe and collision-free path to the target area. The SumTree-DDPG planner also runs 1000 rounds in the same environment (2000 steps per round). During the online path planning process of SumTree-DDPG planner, the number of times the AUV successfully reaches the target area is 39, and the optimal path planned is 1253 m among these safe driving paths. In addition, both algorithms converged within 1000 rounds, but neither planner converged to the optimal path, and the DDPG algorithm fell into the local optimal value earlier than the SumTree-DDPG algorithm. When testing in the complex

environment 2, the SumTree planner has a higher success rate than the DDPG planner, the success rate is increased from 0 to 3.90%.

Table 7. Performance evaluation in the training environment 2 over 1000 episodes.

Algorithm	Numbers of Successful Hits to Target	Min-Episode of Convergence	Optimal Path (m)
DDPG	0	896	None
SumTree-DDPG	39	760	1253

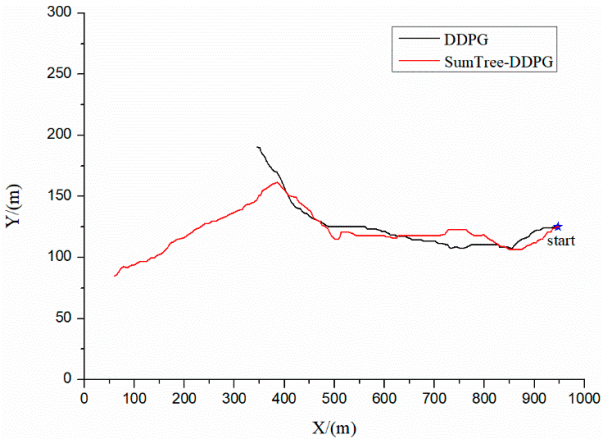


Figure 13. The optimized path of AUV in Environment 2.

Figure 14 shows the average cumulative reward curves obtained in 1000 rounds of DDPG and SumTree-DDPG. Figure 14 shows that when facing the complex environment 2 with dynamic obstacles, the reward value obtained by the DDPG algorithm in each round is mostly the negative reward value generated by the unsuccessful completion of the task. Less AUVs are guided to avoid obstacles to reach the target position during 1000 episodes of online path planning, which shows that the learning effect of the path planner based on the DDPG algorithm is poor. However, because of the SumTree structure selected by the memory pool, the SumTree-DDPG algorithm continuously accumulates good learning samples to eliminate bad memories, and the learning effect improves until it converges to an optimal path that successfully avoids all obstacles to reach the goal. Figure 14 depicts that stability, training effect, and robustness of the SumTree-DDPG planner are better than the DDPG planner.

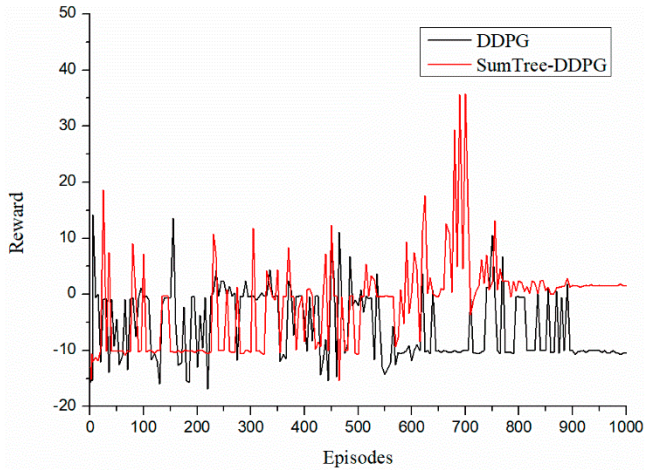


Figure 14. Total reward per episode of AUV training in Environment 2.

4.4. Analysis of Simulations

The simulation results show that the AUV path planning obstacle avoidance method based on DDPG algorithm and the SumTree-DDPG algorithm are effective and can solve the problem on the safe driving of AUV in underwater canyons. Moreover, the proposed SumTree-DDPG algorithm in this work, regardless whether it is the unknown underwater canyon environment 1 or the simulated underwater canyon environment 2 established in this paper, the learning effect is better than the DDPG algorithm, and the stability of the algorithm is better. In Section 4.2, the AUV path planning and obstacle avoidance method based on SumTree-DDPG algorithm is proven effective for path planning and obstacle avoidance in unknown underwater canyon environment. This method can face the AUV autonomous obstacle avoidance in an uncertain environment. However, the AUV motion planning obstacle avoidance method in this paper is a 2D plane space, whereas the actual environment is a 3D space, the energy consumption optimization and the influence of ocean waves in underwater canyons are not considered. We set this revision as a future work.

5. Conclusions

To solve the problem on the safe driving of AUV in underwater canyons and tap the potential of AUV autonomous obstacle avoidance in uncertain environments, this paper proposes an improved AUV based on DDPG path planning method. The method is an end-to-end path planning optimization strategy. Sensor information are considered input, and driving speed and yaw angle are outputs. The path planning method can reach the predetermined target point while avoiding large-scale static obstacles that AUV can only detect a very small part of this obstacle by sonars (less than 20 percent of its overall size), such as valley walls in the simulated underwater canyon environment, as well as sudden small-scale dynamic obstacles whose shape and size can be completely detected by the sonars of AUV, such as marine life and other underwater vehicles. In addition, this research aims at the multi-objective structure of the obstacle avoidance process of path planning, modularized reward function design, and combined artificial potential field method to set continuous rewards. This method solves the sparse reward problem in complex environments. This research also proposes the SumTree-DDPG algorithm, which improves the random storage and extraction strategy of the experience samples of the DDPG algorithm. Aiming at the model convergence rate, this algorithm is combined with the SumTree structure to classify and store the samples and extract high-quality samples continuously according to the different importance of the experience samples. Finally, the effectiveness of the method is verified by simulation.

The main contributions of this paper can be summarized as follows:

(1) To solve the problem on the safe driving of under-driven AUVs in underwater canyons, this research proposes a large-scale continuous obstacle avoidance model, a uniform straight line, and a uniform acceleration straight line state obstacle avoidance model to simulate large-scale static obstacles, such as valley walls in the underwater canyon environment and sudden small-scale dynamic obstacles, such as marine life and other vehicles.
(2) On the basis of the AUV dynamic model, this paper transforms the traditional AUV path planning process into a Markov decision process (MDP) model, which can be used for AUV DRL.
(3) According to the multi-objective structure of the obstacle avoidance process of motion planning, this research carried out a modular design of the reward function and combined the artificial potential field method to set continuous reward.
(4) This research also proposes the SumTree-DDPG algorithm, which improves the random storage and extraction strategy of the experience samples of the DDPG algorithm. According to the importance of the experience samples, the samples are classified and stored in combination with the SumTree structure, and high-quality

samples are continuously extracted, thereby ultimately improving the convergence speed of the model.

Author Contributions: Conceptualization, X.R.; data curation, X.L.; funding acquisition, Y.S. and G.Z.; methodology, X.L. and X.R.; software, X.L.; validation, Y.S.; writing—original draft, X.L.; writing—review and editing, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Heilongjiang Province, grant number ZD2020E005, Financial support for Shaanxi Provincial Water Conservancy Science and technology program, grant number 2020slkj-5, and the China National Natural Science Foundation, grant number 51779057 and 51709061.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Wynn, R.B.; Huvenne, V.A.I.; Le Bas, T.P.; Murton, B.J.; Connelly, D.P.; Bett, B.J.; Ruhl, H.A.; Morris, K.J.; Peakall, J.; Parsons, D.R.; et al. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *J. Mar. Geol.* **2014**, *352*, 451–468. [CrossRef]
2. Guan-Yan, K.E.; Tao, W.U.; Ming, L.; Xiao, D.-B.; College of Mechatronic Engineering and Automation, National University of Defense Technology. The Improvements and Trends of the Unmanned Underwater Vehicles. *J. Natl. Def. Sci. Technol.* **2013**, *34*, 44–47.
3. Londhe, P.S.; Santhakumar, M.; Patre, B.M.; Waghmare, L.M. Task space control of an autonomous underwater vehicle manipulator method by robust single-input fuzzy logic control scheme. *IEEE J. Ocean. Eng.* **2017**, *42*, 13–28.
4. Antonelli, G.; Chiaverini, S.; Finotello, R.; Schiavon, R. Real-time path planning and obstacle avoidance for RAIS: An autonomous underwater vehicle. *IEEE J. Ocean. Eng.* **2001**, *26*, 216–227. [CrossRef]
5. Lozano-Pérez, T.; Wesley, M.A. An algorithm for planning collision-free paths among polyhedral obstacles. *J. Commun. ACM* **1979**, *22*, 560–570. [CrossRef]
6. Lozano-Pérez, T. Spatial planning: A configuration space approach. In *Autonomous Robot Vehicles*; Springer: New York, NY, USA, 1990; pp. 108–120.
7. Takahashi, O.; Schilling, R.J. Motion planning in a plane using generalized Voronoi diagrams. *J. IEEE Trans. Robot. Autom.* **1989**, *5*, 143–150. [CrossRef]
8. Canny, J.F. A Voronoi Method for the Piano Movers Problem. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; IEEE: Piscataway, NJ, USA, 2003; pp. 530–535.
9. Garrido, S.; Moreno, L.; Lima, P.U. Robot formation path planning using Fast Marching. *J. Robot. Auton. Methods* **2011**, *59*, 675–683. [CrossRef]
10. Tanakitkorn, K.; Wilson, P.A.; Turnock, S.R.; Phillips, A.B. Grid-based GA path planning with improved cost function for an over-actuated hover-capable AUV. In Proceedings of the 2014 IEEE/OES Autonomous Underwater Vehicles (AUV), Oxford, MS, USA, 6–9 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–8.
11. Yao, K.; Li, J.; Sun, B.; Zhang, J. An adaptive grid model based on mobility constraints for UAV path planning. In Proceedings of the 2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE), Singapore, 27–29 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 207–211.
12. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *J. IEEE Trans. Methods Sci. Cybern.* **2007**, *4*, 100–107. [CrossRef]
13. Qiang, G. Improved A* Algorithm Based Global Path Planning for Bio-mimetic Robotic Fish. *J. Xihua Univ.* **2011**, *30*, 34–37.
14. Carsten, J.; Ferguson, D.; Stentz, A. 3D Field D: Improved Path Planning and Replanning in Three Dimensions. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 3381–3386.
15. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; IEEE: Piscataway, NJ, USA, 2003; pp. 90–98.
16. Volpe, R.; Khosla, P. Manipulator control with super quadric artificial potential functions: Theory and experiments. *J. Methods Man Cybern. IEEE Trans.* **1990**, *20*, 1423–1436. [CrossRef]
17. Warren, C.W. A technique for autonomous underwater vehicle route planning. In *Symposium on Autonomous Underwater Vehicle Technology*; IEEE: Piscataway, NJ, USA, 1990; pp. 201–205.

18. Chao, W.; Zhu, D.Q. Path Planning for Autonomous Underwater Vehicle Based on Artificial Potential Field and Velocity Synthesis. *J. Control Eng. China* **2015**, *2015*, 717–721.
19. Cheng, C.; Zhu, D.; Sun, B.; Chu, Z.; Nie, J.; Zhang, S. Path planning for autonomous underwater vehicle based on artificial potential field and velocity synthesis. In Proceedings of the 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 3–6 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 717–721.
20. Ferrari, S.; Foderaro, G. A potential field approach to finding minimum-exposure paths in wireless sensor networks. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 335–341.
21. Qi, X.; Palmieri, F. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part I: Basic properties of selection and mutation. *J. IEEE Trans. Neural Netw.* **1994**, *5*, 102–119.
22. Luo, C.; Yang, S.X. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *J. IEEE Trans. Neural Netw.* **2008**, *19*, 1279–1298. [CrossRef]
23. Ru, Y.X.; Yao, Y.Z. Research on AUV Global Path Planning Considering Ocean Current. *J. Ship Build. China* **2008**, *49*, 109–114.
24. Wang, H.J.; Wu, H.X.; Shi, X.C. AUV global path planning method based on ant colony algorithm. *J. Ship Build. China* **2008**, *49*, 88–93.
25. Qiang, Z.; Shao, Z.Y. Collision-Free Path for Mobile Robots Using Chaotic Particle Swarm Optimization. In *Advances in Natural Computation*; IEEE: Piscataway, NJ, USA, 2005; pp. 632–635.
26. Pan, X.; Wu, X.; Hou, X.; Feng, Y. Global path planning based on genetic-ant hybrid algorithm for AUV. *J. Huazhong Univ. Sci. Technol. China* **2017**, *45*, 45–49.
27. Zapata-Ramírez, P.A.; Huete-Stauffer, C.; Scaradozzi, D.; Marconi, M.; Cerrano, C. Testing methods to support management decisions in coralligenous and cave environments. A case study at Portofino MPA. *Mar. Environ. Res.* **2016**, *118*, 45–56. [CrossRef]
28. Ghatee, M.; Mohades, A. Path planning in order to optimize the length and clearance applying a Hopfield neural network. *Expert Methods Appl.* **2009**, *36*, 4688–4695. [CrossRef]
29. Tavares, J. Bio-inspired Algorithms for the Vehicle Routing Problem. In *Studies in Computational Intelligence*; RLO, GER; Springer: Berlin/Heidelberg, Germany, 2009; Volume 161.
30. Yan, M.Z.; Zhu, D.Q. An Algorithm of Complete Coverage Path Planning for Autonomous Underwater Vehicles. *J. Key Eng. Mater.* **2011**, *467–469*, 1377–1385. [CrossRef]
31. Lin, J.C.; Wang, J.H.; Yuan, Y.J. An improved recurrent neural network for unmanned underwater vehicle online obstacle avoidance. *J. Ocean Eng.* **2019**, *189*, 56–72. [CrossRef]
32. Zhu, D.Q.; Bing, S.; Li, L.I. Algorithm for AUV's 3-D path planning and safe obstacle avoidance based on biological inspired model. *J. Control Decis.* **2015**, *30*, 798–806.
33. Duguleana, M.; Mogan, G. Neural networks based reinforcement learning for mobile robots obstacle avoidance. *J. Expert Methods Appl.* **2016**, *62*, 104–115. [CrossRef]
34. Polyoros, A.S.; Nalpantidis, L. Survey of model-based reinforcement learning: Applications on robotics. *J. Intell. Robot. Methods* **2017**, *86*, 1–21.
35. Cui, R.; Yang, C.; Li, Y.; Sharma, S. Adaptive neural network control of AUVS with control input nonlinearities using reinforcement learning. *IEEE Trans. Methods Man Cybern.* **2017**, *47*, 1019–1029. [CrossRef]
36. El-Fakdi, A.; Carreras, M.; Palomeras, N.; Ridao, P. Autonomous underwater vehicle control using reinforcement learning policy search methods. In Proceedings of the 2005 Ocean-Europe, Brest, France, 20–23 June 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 793–798.
37. Cui, R.; Yang, C.; Li, Y.; Sharma, S. Neural network based reinforcement learning control of autonomous underwater vehicles with control input saturation. In *Proceedings of 2014 UKACC International Conference on Control, Loughborough, UK, 9–11 July 2014*; IEEE: Piscataway, NJ, USA, 2014; pp. 50–55.
38. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
39. Li, S.; Xu, X.; Zuo, L. Dynamic path planning of a mobile robot with improved Q-learning algorithm. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015; IEEE: Piscataway, NJ, USA, 2015; p. 409.
40. Zhang, J.; Zhang, J.; Ma, Z.; He, Z. Using partial-policy Q-learning to plan path for robot navigation in unknown environment. In Proceedings of the 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; Volume 1, pp. 192–196.
41. Babu, V.M.; Krishna, U.V.; Shahensha, S.K. An autonomous path finding robot using Q-learning. In Proceedings of the 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 7–8 January 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
42. Wu, F.Z. Application of optimized q learning algorithm in reinforcement learning. *J. Bull. Sci. Technol.* **2018**, *36*, 74–76.
43. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; Petersen, S.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *7540*, 518–529. [CrossRef]
44. Cheng, Y.; Zhang, W. Concise deep reinforcement learning obstacle avoidance for under actuated unmanned marine vessels. *J. Neurocomput.* **2017**, *272*, 63–73. [CrossRef]

45. Omerdic, E.; Toal, D.; Nolan, S.; Ahmad, H.; Duffy, G. Design & development of assistive tools for future applications in the field of renewable ocean energy. In Proceedings of the OCEANS 2011 IEEE-Spain, Santander, Spain, 6–9 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1–6.
46. Kaminski, C.; Crees, T.; Ferguson, J.; Forrest, A.; Williams, J.; Hopkin, D.; Heard, G. 12 days under ice–an historic AUV deployment in the Canadian High Arctic. In Proceedings of the 2010 IEEE/OES Autonomous Underwater Vehicles, Monterey, CA, USA, 1–3 September 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–11.
47. Lilicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *J. Comput. Sci.* **2015**, *8*, 1–14.
48. Mccue, L. Handbook of Marine Craft Hydrodynamics and Motion Control. *J. IEEE Control Methods* **2016**, *36*, 78–79.
49. Wu, B.; Feng, Y. Policy Reuse for Learning and Planning in Partially Observable Markov Decision Processes. In Proceedings of the 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23 July 2017; IEEE Computer Society: Washington, DC, USA, 2017.
50. Lowe, R.; Ziemke, T. Exploring the relationship of reward and punishment in reinforcement learning. In Proceedings of the IEEE Symposium on Adaptive Dynamic Programming & Reinforcement Learning, Singapore, 16–19 April 2013; IEEE: Piscataway, NJ, USA, 2013.