Journal of
Marine Science and Engineering

MDPI

Article

AUV 3D Path Planning Based on the Improved Hierarchical Deep Q Network

Yushan Sun, Xiangrui Ran, Guocheng Zhang *, Hao Xu and Xiangbin Wang

Science and Technology on Underwater Vehicle, Harbin Engineering University, Harbin, 150001, China; sunyushan@hrbeu.edu.cn (Y.S.); ranxiangrui@hrbeu.edu.cn (X.R.); xuhao0619@hrbeu.edu.cn (H.X.); 2010011501@hrbeu.edu.cn (X.W.)

* Correspondence: zhang_china2018@163.com

Received: 17 January 2020; Accepted: 21 February 2020; Published: 24 February 2020

Abstract: This study proposed the 3D path planning of an autonomous underwater vehicle (AUV) by using the hierarchical deep Q network (HDQN) combined with the prioritized experience replay. The path planning task was divided into three layers, which realized the dimensionality reduction of state space and solved the problem of dimension disaster. An artificial potential field was used to design the positive rewards of the algorithm to shorten the training time. According to the different requirements of the task, this study modified the rewards in the training process to obtain different paths. The path planning simulation and field tests were carried out. The results of the tests corroborated that the training time of the proposed method was shorter than that of the traditional method. The path obtained by simulation training was proved to be safe and effective.

Keywords: AUV; path plan; deep Q network; hierarchical structure; prioritized experience replay

1. Introduction

As a key technology in the marine industry, autonomous underwater vehicles (AUVs) have been given considerable attention and application [1]. They play an important role in the development of marine resources and environmental protection. In civil applications, AUVs are mostly used in underwater topography exploration, hydrological information and water quality detection, underwater wreck detection, and other aspects [2–4].

Path planning is one of the keys to the AUV system, which has been extensively studied. Ben Li [5] proposed an improved genetic algorithm for the global three-dimensional path planning of an under-actuated AUV. The shortest path was obtained by hierarchical path planning. However, the genetic algorithm is prone to premature convergence, and its local optimization ability is poor. J. D. Hernández [6] presented a framework for planning collision-free paths online for AUVs in unknown environments. It was composed of three main modules that incrementally explored the environment while solving start-to-goal queries. They planned paths for the SPARUSII AUV performing autonomous missions in a 2-dimensional workspace. However, the obstacles in the simulation process were too simple. Petres [7] designed a continuous state, which used an anisotropic fast matching algorithm to complete the AUV path planning task. However, this method only used linear evaluation function, which had certain limitations. Sun B [8] proposed an optimal fuzzy control algorithm for 3D path planning. Based on the environment information, the virtual acceleration and velocity of AUV could be obtained through the fuzzy system, so that AUV could avoid dynamic obstacles automatically. However, because of the subjectivity of fuzzy boundary selection, the generated path could not be guaranteed to be optimal.

There are some problems with the above path planning methods. Currently, artificial intelligence technology is developing rapidly [9], which can greatly improve the intelligence level and autonomy of

AUVs [10]. Reinforcement learning has been studied for AUV path planning. Hiroshi et al. [11] proposed a multi-layer training structure based on Q-Learning. They carried out a planning simulation experiment on the R-ONE vehicle. However, Q-learning is difficult to apply in a continuous environment. Yang and Zhang [12] integrated reinforcement learning with the fuzzy logic method for AUV local planning under the sea flow field. Q-learning was used to adjust the peak point of the fuzzy membership function. The recommendations of behaviors were integrated through adjustable weighting factors to generate the final motion command for AUVs. However, the environment model was simple and unrealistic. Liu [13] used the Q-learning to make local path planning for AUVs. The simulation was carried out in the electronic chart, but he did not set the rewards effectively. Cheng et al. [14] proposed a motion planning method based on DRL. They used CNN to extract the characteristics of sensor information in order to make decisions on the motion. However, this type of training requires a long period.

In view of the existing problems in the present studies, this study proposed an improved hierarchical deep Q network (HDQN) method with the prioritized experience replay to realize the three-dimensional path planning of AUV. The AUV path planning task was divided into three layers, and the planning strategy of AUV was trained layer by layer to reduce the learning time and improve the learning efficiency. Compared with the standard practice in AUVs, HDQN had four benefits: 1. HDQN did not require researchers to pre-program tasks. 2. HDQN method ensured the security of AUV by training in simulation. 3. The method solved the problem of dimension disaster based on the idea of stratification. 4. The method solved the problem of the local optimal solution. This study used a triangular prism mesh to discrete the environmental state model. It could increase the choice of horizontal motions to optimize the path and simplify the vertical motions to reduce the environmental state. Based on the water flow and terrain obstacles, the rewards of the AUV training process were set in detail. Combining the prioritized experience replay [15] with the HDQN (HDP) improved the learning rate of AUVs and shortened the learning time. The idea of the artificial potential field was added to the HDP (HDPA) to improve the problem of the sparse rewards and to shorten the training time. According to the different requirements of tasks, the rewards were modified to obtain the paths of different selection strategies.

The remainder of this paper is organized as follows: In Chapter 2, the path planning algorithm is designed. In Chapter 3, the path planning simulation tests are discussed. In Chapter 4, a field experiment is completed to prove that the path obtained by training is reliable. Chapter 5 concludes the study.

2. Path Planning Algorithm

As shown in Figure 1, in the path planning and control process of AUVs, the global path planning of AUVs is realized in the upper computer. After launching an AUV, the path nodes obtained by global path planning are transmitted to the lower computer by radio [16]. The AUV navigates to the path nodes according to the path following strategy. Furthermore, the AUV detects the surrounding environment and completes the task by using the detection equipment. The target heading, target velocity, and target depth are calculated in the planning system, which sends them to the control system. The control system controls the AUV in navigating according to the target command [17]. In this study, the improved HDQN algorithm was proposed to realize global path planning.

2.1. HDQN and Prioritized Experience Replay

HDQN is the improved reinforcement learning algorithm based on hierarchical thinking. Reinforcement learning refers to the learning process of humans, which sets the reward artificially and enables agents to search for the optimal strategy through constant trials and errors [18]. Figure 2 depicts that reinforcement learning is a process of trial and error [19].

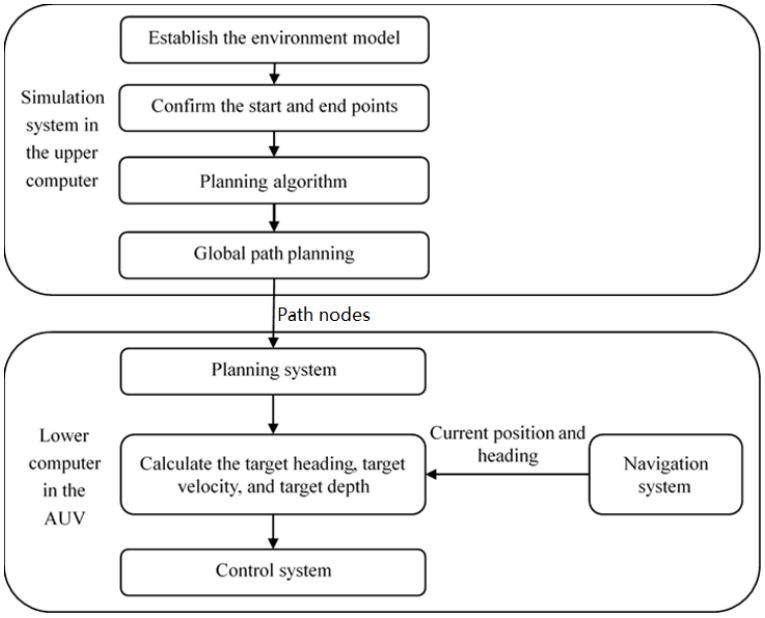


Figure 1. Flowchart of the autonomous underwater vehicle (AUV) path planning.

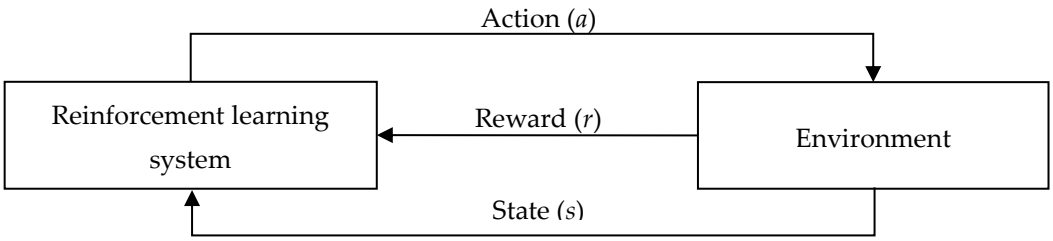


Figure 2. Process of reinforcement learning.

2.1.1. Hierarchy of the Path Planning Task

The path planning task is seen as a three-layer model, as shown in Figure 3. The algorithm framework is a bottom-up hierarchical structure consisting of the environment interaction layer, sub-task selection layer, and root task collaboration layer. The environment interaction layer acquires the environment information and interacts with the accumulated experience of the AUV. Experience accumulation of environmental information in the learning experience database is then obtained. The data is compared with the current state of the marine environment, and the result of the comparison is passed to the root task collaboration layer. The root task cooperation layer transmits the action sequence to the sub-task selection layer based on the current state information and generates the sub-task decision based on the environment state. The subtask selection layer receives the output from the root task layer and selects subtasks or actions based on the policy. The AUV selects actions to act on the environment and updates the learning experience based on feedback from the environment. In conclusion, the process of hierarchical reinforcement learning method is the top-down decision-making process and the bottom-up learning process.

2.1.2. HDQN

The Q-learning method was proposed by Watkins in 1989 [20]. It is a temporal difference method with model-free based on off-policy. Similar to that of reinforcement learning, the idea of Q-learning is to construct a control strategy to maximize agents' behavioral performance [21]. Agents process the information that is perceived from the complex environment. In this process, four parameters are

required, namely, action set, rewards, environment state set, and action-utility function. In the training process, the Bellman equation [22] is used to update the q-table:

$$Q(s,a) = r + \gamma \max Q(s',a') \tag{1}$$

In Equation (1), Q is the action-utility function, which means the immediate reward for the action taken in the current state. r is the reward. s is the environment state. a is the action. γ is the discount parameters.

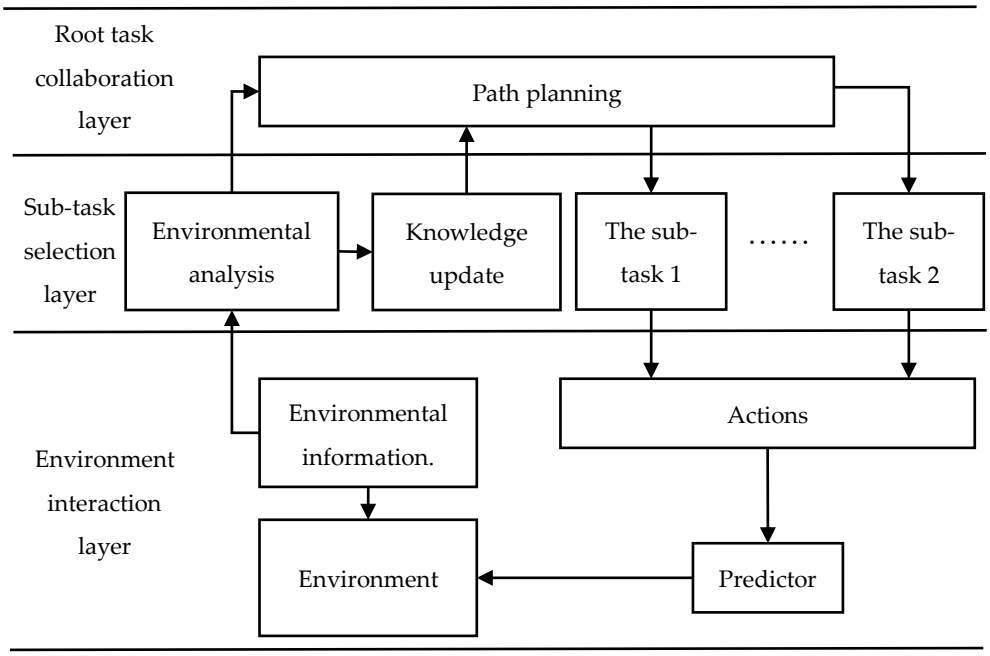


Figure 3. Hierarchical theoretical framework.

The purpose of dynamic programming is to optimize the evaluation function, that is, to maximize the expectation of infinite discount rewards. The optimal evaluation function is defined as:

$$V(s) = \max E\left(\sum_{t=0}^{\infty} \gamma R\right) \tag{2}$$

where $R = R(s,a)$ represents the reward obtained by the system in selecting an action according to the current environment state. E is the expectation of the cumulative rewards. According to formula (1), the sufficient and necessary condition for the optimal evaluation function is:

$$V(s) = \max_{a \in A}\left\{R(s,a) + \gamma \sum_{s' \in S} P[s'|s,a]V(s')\right\} \tag{3}$$

$P(s'|s,a)$ is a state transition probability function, which indicates the probability of the system choosing an action based on the current state to move to the next state.

The optimal strategy is searched by strategy iteration. The idea is to start with a random strategy and then evaluate and improve it until the optimal strategy is found. For any state, the evaluation function of the strategy named π is calculated as follows:

$$V^{\pi}(s) = R(s,a) + \gamma \sum_{s' \in S} P[s'|s,a]V^{\pi}(s') \tag{4}$$

In the path planning task, obstacle avoidance and approaching the target are defined as subtasks. The transition function is defined as $P\left(s_{j+1}\middle|s_j,a\right)$. $V^{\pi}(a,s)$ represents the evaluation of the system, performing an action to change the state from s_j to s_{j+1} according to strategy:

$$V^{\pi}(a,s) = \sum P\left(s_{j+1}\middle|s_j,a\right)\cdot r\left(s_{j+1}\middle|s_j,a\right) \tag{5}$$

Starting at the s, after N time steps, the system terminates at the s'. The state transition probability is $P^{\pi}(s',N|s,a)$. The evaluation function of sub-task is defined as:

$$V^{\pi}(i,s) = V^{\pi}(\pi(s),s) + \sum_{s',N} P^{\pi}\left(s',N\middle|s,\pi_i(s)\right)\cdot V^{\pi}(i,s'). \tag{6}$$

where i is the sub-task. $Q^{\pi}(i,s,as)$ is defined as the expectation of accumulated rewards:

$$Q^{\pi}(i,s,as) = V^{\pi}(as,s) + \sum_{s',N} P^{\pi}\left(s',N\middle|s,as\right)\gamma^{N}Q^{\pi}(i,s',\pi(s')) \tag{7}$$

where *as* are the actions performed in a sub-task. Formulas (6) and (7) are the evaluation functions of the algorithm structure.

The neural network is used to train the Q function. Take the states as the inputs and the Q value of all actions as the outputs of the neural network. According to the Q-learning idea, the action with the maximum Q value is directly selected as the next action. Figure 4 illustrates that, when training the AUV with the neural network, the Q value of the action, which is calculated by Equation (1), is required first. The Q value is estimated by updating the neural network. Thereafter, the action corresponding to the maximum estimated value of Q is selected to exchange for rewards in the environment. The estimated Q value is subtracted from the maximum Q value to obtain the error loss. The loss function is as follows:

$$L = E_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q\left(s',a';\theta_i^{-}\right) - Q(s,a;\theta_i)\right)^2\right] \tag{8}$$

where θ is the parameter of the network. The target Q value is expressed by Target Q:

$$\text{Target}Q = r + \gamma \max_{a'} Q\left(s',a';\theta_i^{-}\right) \tag{9}$$

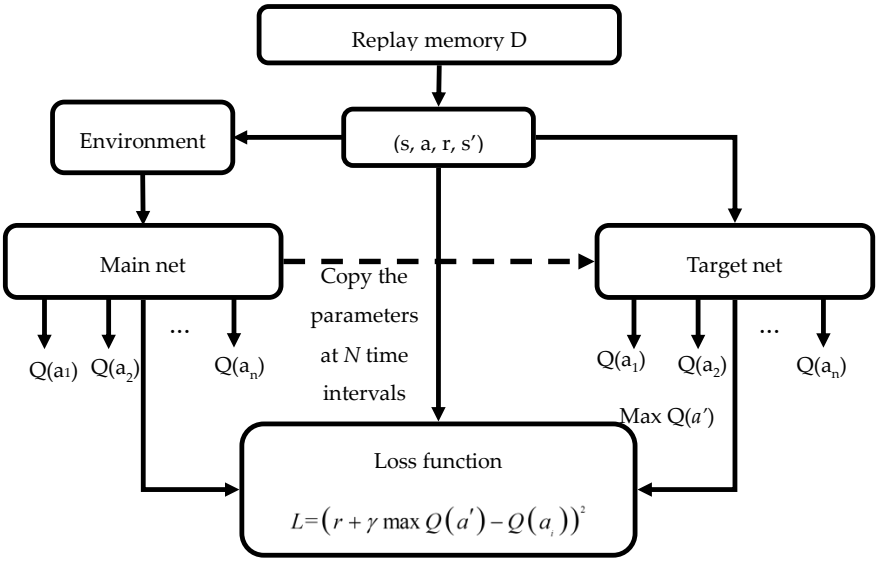


Figure 4. Schematic of the neural network.

The loss function is determined on the basis of the second item of the Q-learning. The gradient of $L(\theta)$ in relation to θ is then obtained, and the network parameter is updated according to the SGD [23] or other methods.

2.1.3. Prioritized Experience Replay

The experience replay is an important link to the HDQN. The HDQN uses a matrix named memory bank to store learning experiences. The structure of each memory is (s, a, r, s'). The q-target is determined on the basis of the reward, and the maximum Q value is obtained by the target-net network. The HDQN method stores the transfer samples into the memory bank. Some memories are extracted randomly when training. However, a problem arises with this experience replay method. The learning rate of agents will be slow down due to lacking successful experiences when positive rewards are few in the early stage. A strategy is needed for the learning system to prioritize the good experience in the memory bank to study the successful experience, that is, prioritized experience replay.

Prioritized experience replay is not random sampling but sampling according to the priority of samples in the memory bank to make learning more efficient. The priority of the sample is determined by td-error, in which q-eval is used to determine the order of learning. The larger the td-error, the lower the prediction accuracy. Therefore, the priority of the sample that must be learned is high. The priority is calculated on the basis of the following equation:

$$X \sim P(X) = p_X^\alpha / \sum_y p_y^\alpha \tag{10}$$

where p_y is the td-error.

The method named Sum Tree is used to sample memories from the bank after determining the priority of memories. Sum Tree is a tree structure. Each leaf stores the priority expressed by p of each sample. Each branch node only has two branches. The value of the node is the sum of the two branches' value. Thus, the top of Sum Tree is the sum of all the values of p. Figure 5 affirms that the lowest leaf layer stores the p of each sample.

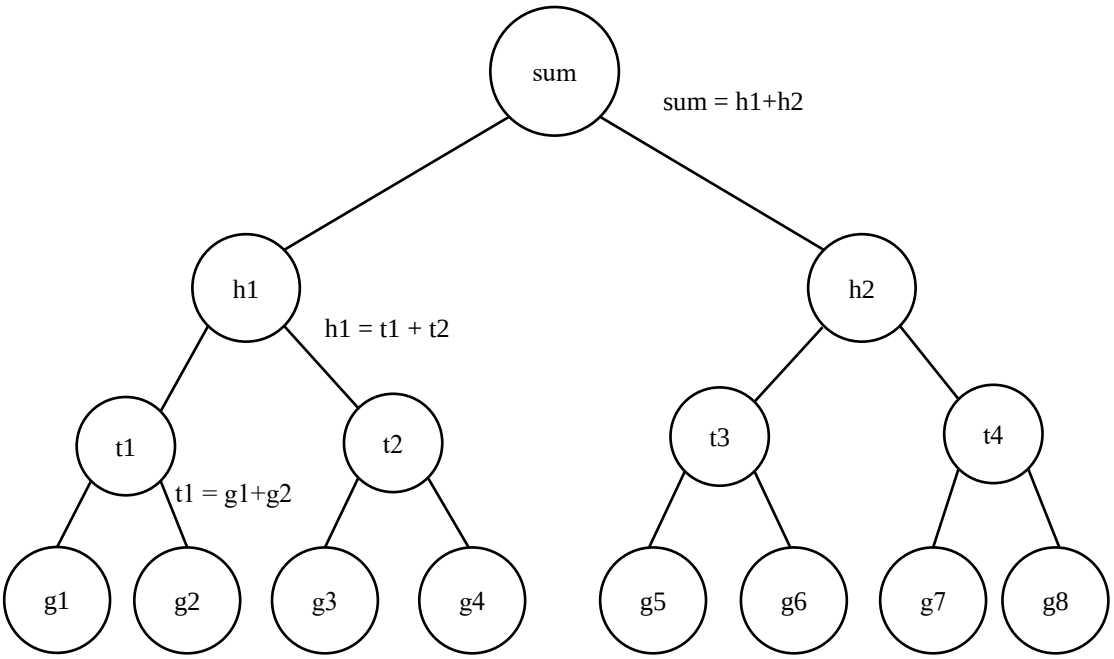


Figure 5. Schematic of Sum Tree.

When sampling, the memories must be divided by the batch size. Thereafter, a random number expressed by g is chosen in each interval. g is compared with $h1$. If $g > h1$, then g must be compared with $t1$. If $g < h1$, then g must be compared with $t2$. If $g > h1$ and $g > t1$, then g must be compared with $g1$. If $g > h1$ and $g < t1$, then $g = g - t1$, and g must be compared with $t2$. By analogy, down to the bottom of the tree, if $g > g1$, then the left sample is selected. Otherwise, the right sample is selected.

2.2. Set the Rewards and Actions

The artificial potential field method is introduced to design positive rewards to make the AUV find the target position more quickly. When the distance between the AUV's current position and the target position is smaller than that between the AUV's previous position and the target position, the AUV is rewarded positively by the following potential function:

$$R = \frac{l_{\max}}{l} \cdot 0.01 \tag{11}$$

where $l_{\max}$ is the distance between the AUV's start position and the target position, and l is the distance between the AUV's current position and the target position.

To be affected by the flow of water, for AUV navigation, is inevitable. AUVs are most affected by side flow and least affected by downstream or headstream. Therefore, when an AUV is set to navigate in a current, the reward is calculated as follows:

$$R = -0.01 \cdot \left|\sin(\phi - \phi')\right| \tag{12}$$

where ϕ represents the navigation direction of the AUV. ϕ' represents the water flow direction. ϕ and ϕ' are in the same coordinate system. Their value ranges are $[-\pi, \pi]$.

The rewards are constantly adjusted during the simulation training, which are set as follows:

$$R = \begin{cases} 10 & \text{arrive at the target position} \\ \frac{l_{\max}}{l} \cdot 0.01 & \text{close to the target position} \\ -0.01 & \text{navigate at the specified depth} \\ -0.05 & \text{navigate at an unspecified depth} \\ -1.5 & \text{collide with an obstacle} \\ R = -0.01 \cdot \left|\sin(\phi - \phi')\right| & \text{navigate with the influence of current} \end{cases} \tag{13}$$

In the process of AUV 3D path planning training, the AUV will receive a maximum reward of 10 when it reaches the target position. When the AUV collides with an obstacle, it will receive a minimum reward of −1.5. A negative reward of −0.01 is set for each movement of the AUV to avoid reciprocating movement. The AUV usually navigates at the specified depth to complete tasks. Therefore, the negative reward obtained by sailing the AUV away from the specified depth is set to be larger; that is, the reward of avoiding obstacles in the vertical plane of the AUV is −0.05.

The AUV's actions are divided into 14 parts. The AUV moves vertically first and then starting the horizontal navigation. Therefore, the vertical motion of the AUV can be divided into two movements: up and down. The horizontal motion of the AUV is divided into 12 actions with 30° of separation between the bow angles to ensure that the AUV has more direction choices in optimizing paths (Figure 6).

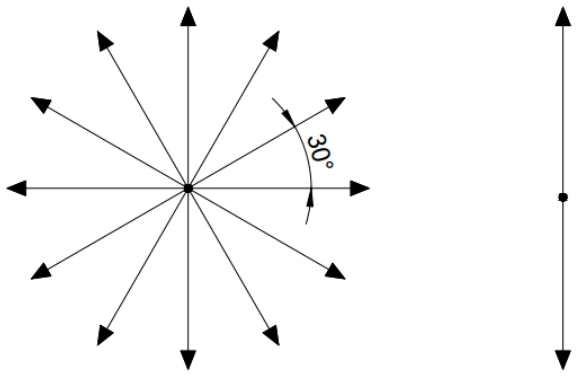


Figure 6. AUV actions: (**a**) The horizontal actions; (**b**) The vertical actions

2.3. Algorithm Process

After confirming the parameters of the HDQN, the 3D path planning for AUVs based on the improved HDQN is prepared. The process is as follows:

Algorithm 1: the path plan training process.

Initialize parameters.
Initialize memory.
for episode **in** range (10000):
  Initialize states.
  for step **in** 1000:
    Select a random action; Otherwise select the action according to the observation value of the state:;
    The system performs the selected action to move to the next state s';
    Judge the current state and calculate the reward according to formula (13);
    Save s, a, r, and s' to the memory bank.
    if (step > 200) **and** (step % 2 = 0):
      Update the parameters of target_net every 500 times;
      Calculate the td-error and to extract memories from the memory store according to the priority;Calculate q_target: Target$Q = r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right)$;
      Select the action corresponding to the maximum Q value. To set the Q value of other actions to 0, the error value of the action selected is transferred back in the neural network as the update credential;
      Perform a gradient descent step on $L = E_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right) - Q(s, a; \theta_i)\right)^2\right]$ with respect to the network parameters θ;
      Increase the size of epsilon parameter to reduce the randomness of the action;
    Use the next state as the state of the next loop.
    if done:
      Break;
      step = step + 1;
  end for;
end for;

3. Simulation Experiment

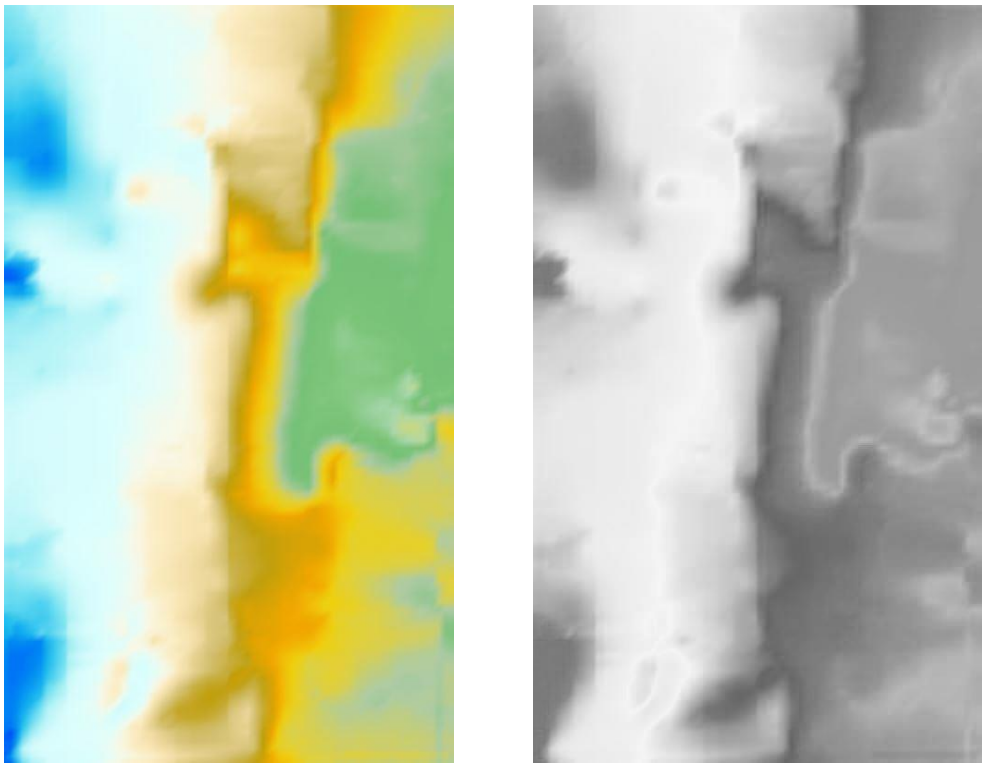
3.1. Experimental Definition

The autonomous underwater vehicle named AUV-R, as shown in Figure 7, developed by the Science and Technology on Underwater Vehicle Laboratory of Harbin Engineering University, was

used to perform comb scanning in the river of Qinghai Province, China. The altitude of the test water area was measured by the altimeter. The ADCP was used to measure the flow of the experimental water area.

Figure 7. Autonomous underwater vehicle – river (AUV-R).

Figure 8a shows the topographic information map. Figure 8b shows the height map containing only black and white. Figure 8b exhibits that the lighter the color, the higher the terrain. The terrain was higher near the river bank and became lower away from the shore.



a. b.

Figure 8. Topographic map of test waters: (**a**) Topographic information map; (**b**) Height map.

The gray value in Figure 8b was identified by the simulation system to determine the height of each coordinate to establish a three-dimensional environment model. By summarizing the existing modeling ideas [24,25], the three-dimensional environment model was discretized, and the triangular prism mesh model was established. Figure 9a shows the horizontal modeling. The environment was modeled in the form of an isosceles triangle. The black points in the figure are grid nodes. Twelve actions in the horizontal plane are shown in Figure 9a by the red points and red lines. Figure 9b shows the three-dimensional state model. Some lines were used to connect nodes to make the model clear.

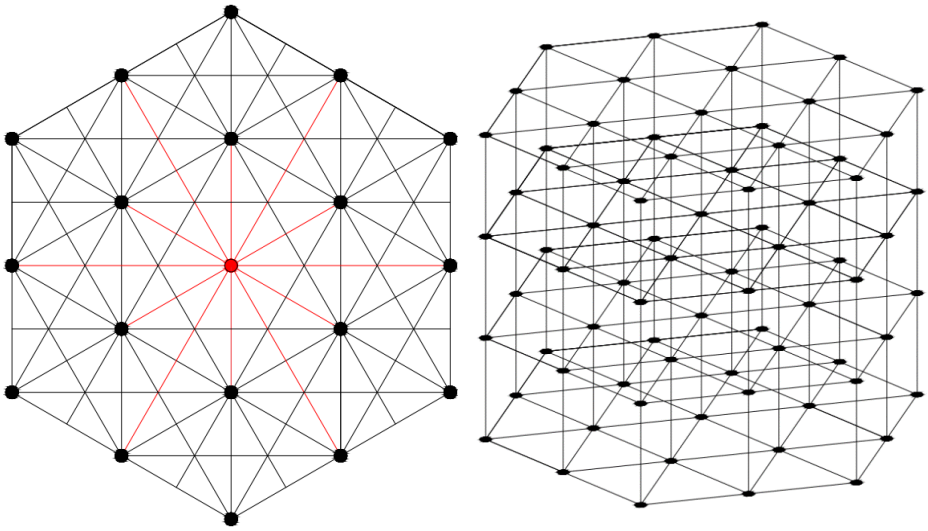


Figure 9. The discrete model. (**a**) Horizontal model; (**b**) 3D model.

The underwater 3D environment model was established on the basis of pyopen-gl by using Figure 8b. The water flow was simplified to uniform flow, and the direction was $\phi' = -\pi \cdot 2/3$. As shown in Figure 10, the yellow cylinder was the AUV, and the red sphere was the target. The AUV model was appropriately enlarged for visual clarity.

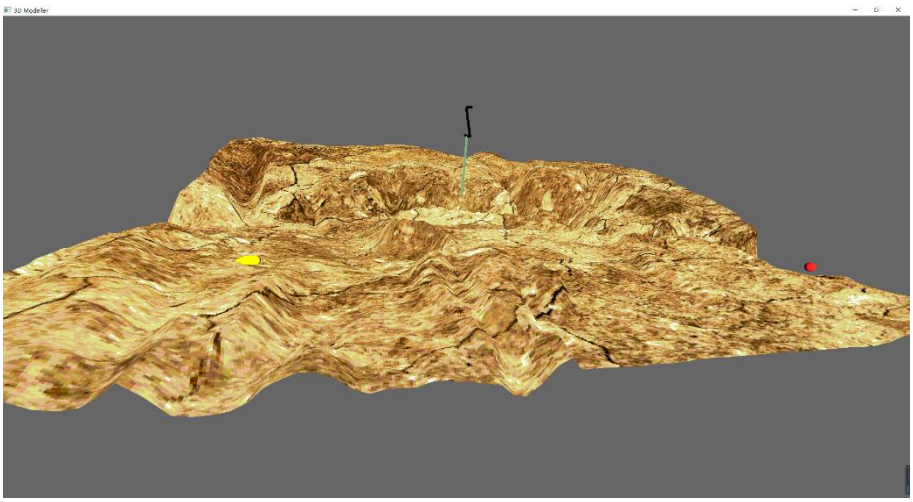
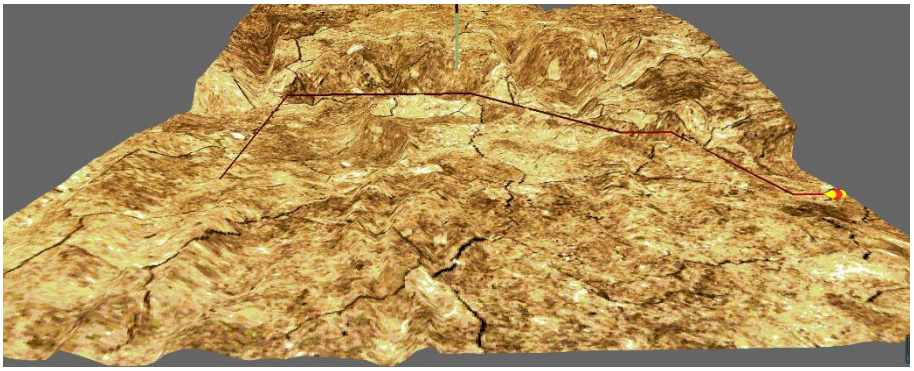


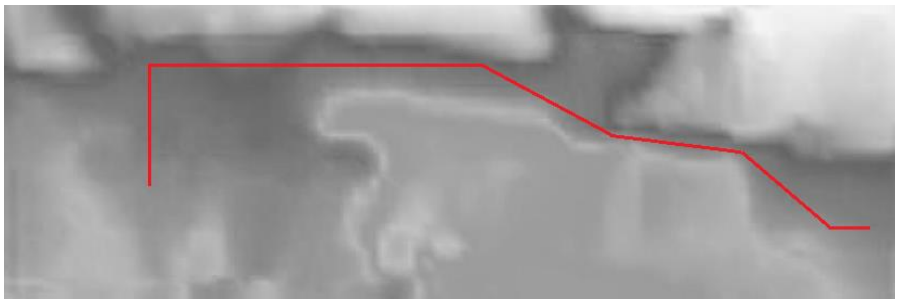
Figure 10. Environment model.

3.2. Path Planning

The AUV was simulated and tested in the simulation system. The task was to allow the AUV to navigate safely to the target position at a depth of 10 m. The 3D path planning simulation training was carried out for the AUV using the improved HDQN. The rewards and actions were set in accordance with the content of Section 2.2. The initial position of the AUV was set as the starting point. The task node of the AUV was set as the target point. The training step was set at 100,000 episodes. During the simulation, the AUV colliding with obstacles or arriving at the target point represented the completion of one episode. Figures 11 and 12 show the simulation results.



a.



b.

Figure 11. Simulation test results of path planning: path: (**a**) 3D view; (**b**) Top view based on a grayscale.

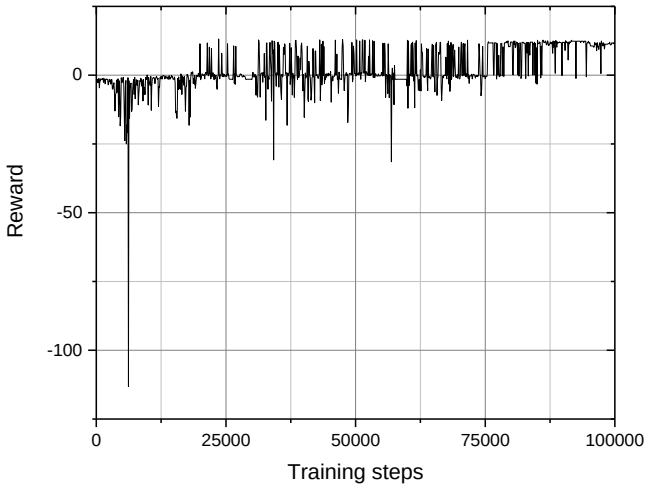


Figure 12. Simulation test results of path planning: cumulative rewards.

Figure 11 shows the path from the starting point to the endpoint of the AUV obtained by the simulation system. The improved HDQN algorithm could be used to obtain a suitable path at a fixed depth with avoiding topographic obstacles and little influence of water flow. Figure 12 shows the accumulated rewards of the AUV in each episode of the simulation system. It took approximately 19,000 episodes before the AUV reached the target position for the first time. After 75,000 times of training, better results could be obtained. The system randomly selected the action with a 10% probability. Therefore, the AUV could not reach the target position every time, even in the later stage of training.

The HDQN algorithm, HDQN algorithm combined with prioritized experience replay (HDP), and HDP combined with the artificial potential field method (HDPA) were used to train the path planning

of the AUV. The number of times that the AUV arrived at the target position in the training process was recorded. Figure 13 shows the results.

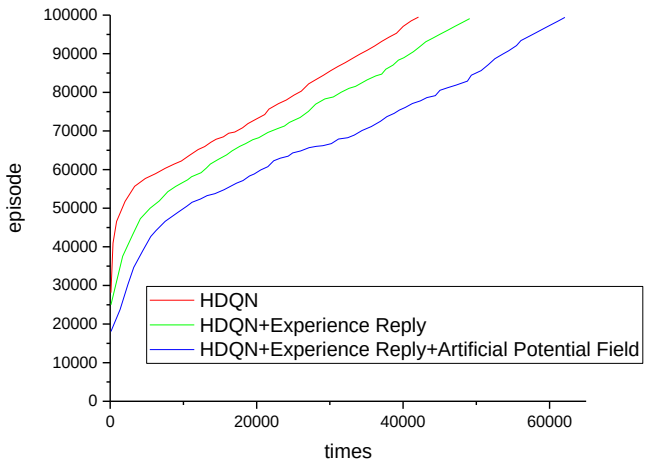


Figure 13. Comparison of the three methods.

In Figure 13, the x-coordinate represents the number of times that the AUV arrived at the target point, and the y-coordinate represents the episodes of the training process. Results showed that the HDP algorithm had a lower slope than the HDQN algorithm in the early training period, meaning that the time interval of the AUV reaching the target position with the HDP algorithm was shorter. The system made effective use of its successful training experience. In addition, the blue curve in Figure 13 shows that the artificial potential field module enabled the AUV to reach the target position faster and gain a successful experience earlier, causing the system to achieve a suitable path faster. Through experiments, it was found that the HDPA algorithm could make AUV reach the target point earlier and get more useful experience. It could be proved that the learning rate of the HDPA method was higher than that of the other two methods.

To remove the navigation limit of the AUV at a fixed depth of 10 m, the rewards obtained when the AUV navigated at an unspecified depth in Equation (13) must be modified. The negative reward of the AUV leaving the target depth must be reduced, as shown in Equation (14), to realize vertical obstacle avoidance. The parameters in Equation (14) were obtained through multiple adjustments during simulation training. A total of 100,000 steps of training episodes were set, and the simulation results are shown in Figures 14 and 15.

$$R = \begin{cases} 10 & \text{arrive at the target position} \\ \frac{l_{\max}}{l} \cdot 0.01 & \text{close to the target position} \\ -0.01 & \text{navigate at the specified depth} \\ -0.01 & \text{navigate at an unspecified depth} \\ -1.5 & \text{collide with an obstacle} \\ R = -0.01 \cdot \left|\sin(\phi - \phi')\right| & \text{navigate with the influence of current} \end{cases} \quad (14)$$

In Figure 14, the red line is the path. In Figure 14c, the red and yellow lines represent paths at different depths. Results showed that the AUV chose the vertical obstacle avoidance strategy when meeting the terrain obstacles. Figure 15 shows that the AUV reached the target position for the first time at nearly the 18,000th episode. After approximately 50,000 times of training, the simulation system of AUV path planning could obtain good results.

Figures 11 and 14 illustrate that the path of the AUV with a horizontal obstacle avoidance strategy was longer than that of the AUV with vertical obstacle avoidance. However, given the limitations of some tasks, the AUV must keep in constant depth as much as possible. By setting the different rewards of the improved HDQN, the AUV could be trained to choose the appropriate path. All in all,

the improved HDQN omitted the process of establishing the optimal model of the task so that the optimal solution could be found conveniently and quickly.

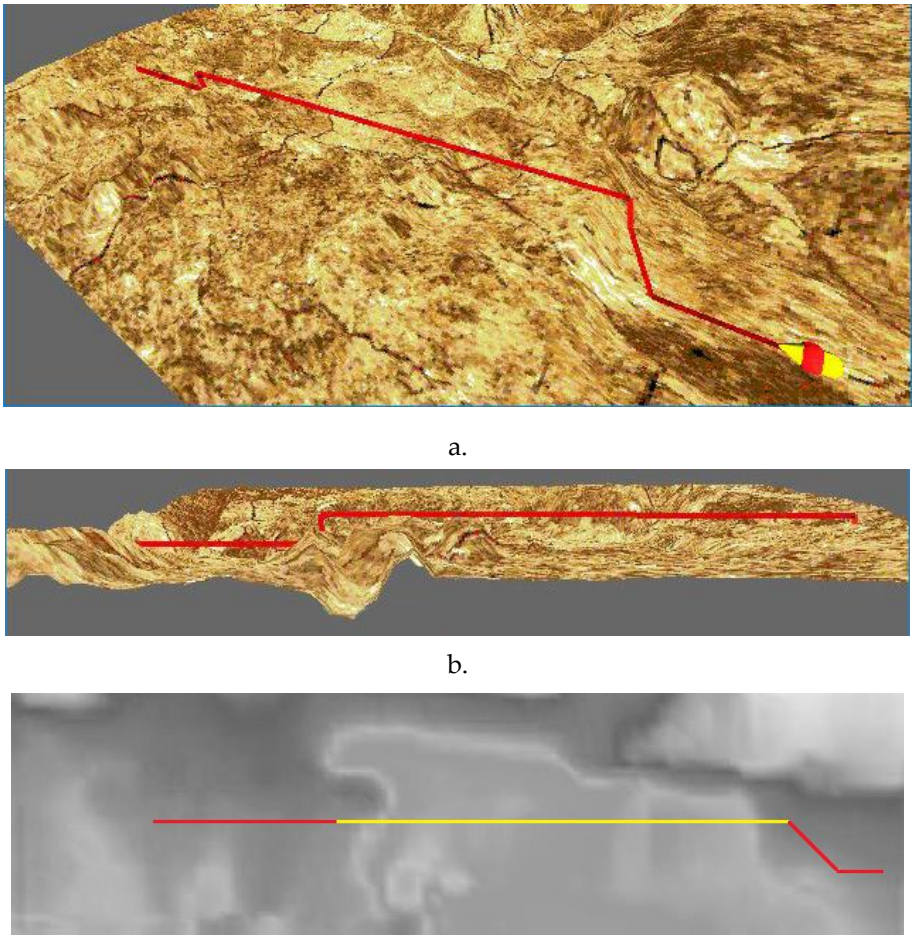


Figure 14. Simulation results of vertical obstacle avoidance: path. (**a**) 3D view; (**b**) Front view; (**c**) Top view based on a grayscale.

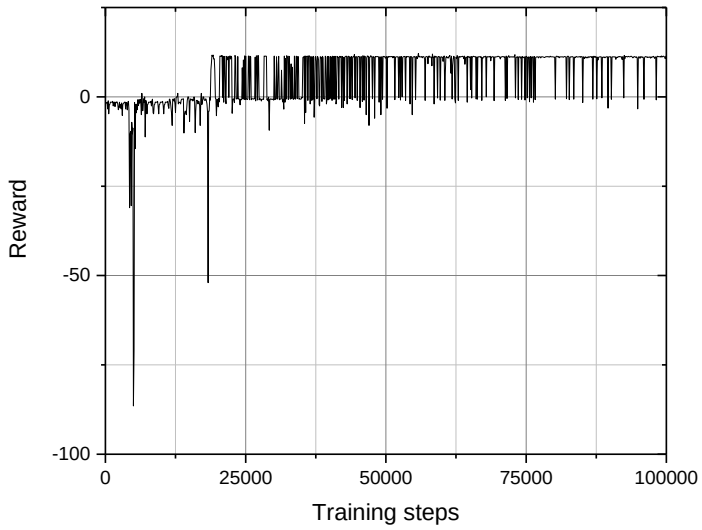


Figure 15. Simulation results of vertical obstacle avoidance: cumulative rewards.

4. Field Experiment

The field experiment was carried out to prove that the training path was reliable, which was completed in an inland river area of Qinghai Province, China. As the length of a river was much longer than its width, the AUV was required to set the depth and avoid obstacles to navigate safely along the river bank to collect the information of the test water area efficiently and quickly.

Six collision avoidance sonars were mounted on AUV's head, tail, and sides to ensure the safety of the AUV, as shown in Figure 16. They measured the distance from the AUV to the obstacle in four directions. When the distance was less than 3 m, AUV would avoid obstacles. The obstacle avoidance strategy of this experiment adopted the manual experience method:

$$\beta = \begin{cases} \beta' + \pi/18 & \text{obstacles detected ahead, and no obstacle detected left} \\ \beta' - \pi/18 & \text{obstacles detected ahead, and no obstacle detected right} \\ \beta'' & \text{no obstacle detected ahead} \end{cases} \quad (15)$$

where β and β' are the target and current heading angles of the AUV. β'' is the planned heading angle of the system. $\pi/18$ is the empirical value obtained from multiple field tests.

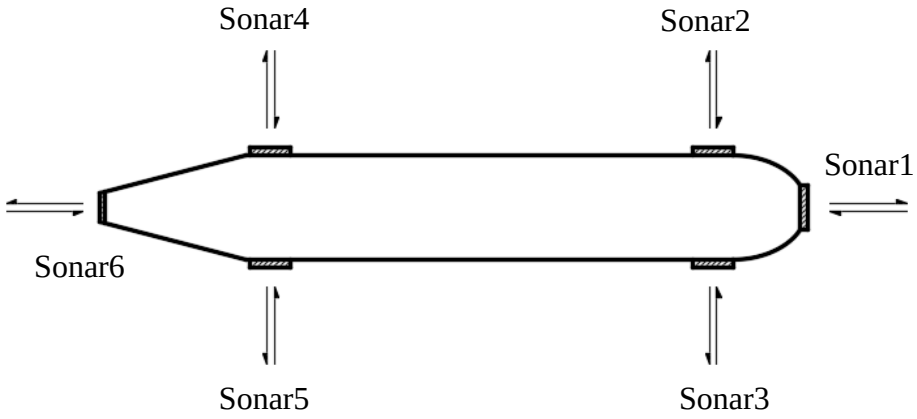


Figure 16. The layout of sonars on AUV.

The simulation test results, as shown in Figure 11, were used to complete the field experiment of the AUV (Figure 17). After launching the AUV into the water and sending the nodes of the path to the AUV, the AUV navigated to the path node. The test results are shown in Figure 18.

Figure 17. Field experiment of AUV path planning.

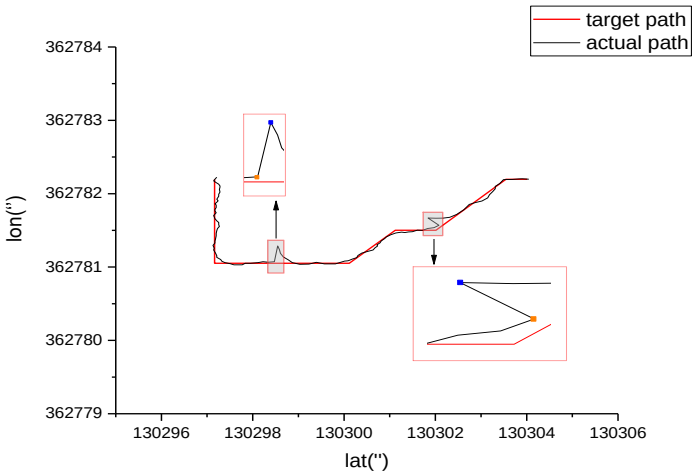


Figure 18. Results of the AUV tracking the path.

Figure 18 shows that the AUV tracked the planning path after navigating to the first path node. In this experiment, the AUV resurfaced twice to correct its position. Based on the dead reckoning, DVL, magnetic compass, depth meter, and altimeter were used to determine the position of the AUV. Table 1 shows the specifications of the DVL. When the AUV rose to the surface, GPS was used to correct its position, as shown in the rectangular box in Figure 18. The AUV's position was corrected from the orange point to the blue point. The distance of this test voyage was short, resulting in a small error accumulation, so there was little deviation between positions of the dead reckoning and the actual.

Table 1. DVL specifications.

Model	Frequency	Accuracy	Maximum Altitude	Minimum Altitude	Maximum Velocity	Maximum Ping Rate
NavQuest 600 Micro	600 kHz	1% ± 1 mm/s	110 m	0.3 m	±20 knots	5/s

Figure 19 records the data collected by sonar 1, 2, and 3. In order to display the data clearly, the data over 20 m was cut out. Since the field experiment was carried out in inland waters, the underwater conditions were relatively stable. As could be seen from Figure 19, the distance measured by sonar was always more than 3 m, and the AUV had been sailing in a safe area without triggering the obstacle avoidance strategy.

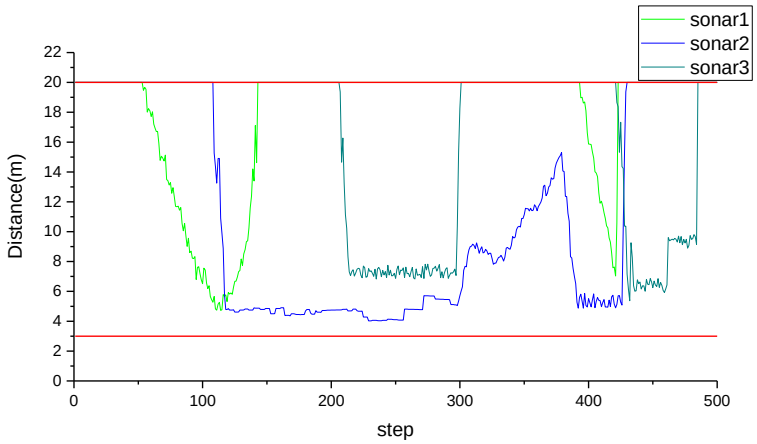


Figure 19. Records from the sonars.

5. Conclusions

In the study, a real underwater environment model was established. The HDQN, combined with the prioritized experience replay, improved the learning rate of the agent and shortened the learning time. The path planning task was divided into three layers with the idea of layering so as to solve the problem of dimension disaster. The idea of an artificial potential field was added to improve the problem of sparse rewards. The different paths could be obtained by modifying and setting the rewards. The field experiment proved that the path obtained by simulation training was safe and effective. However, the framework proposed in the study could not realize real-time obstacle avoidance of AUVs in an unknown environment. This would be the next problem that needs to be researched and solved.

Author Contributions: Conceptualization, G.Z.; Data curation, Y.S. and G.Z.; Formal analysis, X.R.; Funding acquisition, Y.S.; Methodology, X.R.; Resources, Y.S.; Software, G.Z., H.X. and X.W.; Writing—original draft, X.R.; Writing—review and editing, X.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Equipment Pre-research Project (grant number 41412030201), the China National Natural Science Foundation (grant number 51779057, 51709061).

Acknowledgments: The author would like to thank the reviewers for their comments on improving the quality of the paper. Special thanks also go to the employees of Harbin Engineering University for their assistance during the field experiments.

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

References

1. Fan, J.; Li, Y.; Liao, Y.; Jiang, W.; Wang, L.; Jia, Q.; Wu, H. Second Path Planning for Unmanned Surface Vehicle Considering the Constraint of Motion Performance. *J. Mar. Sci. Eng.* **2019**, *7*, 104. [CrossRef]
2. Kirkwood, W.J. AUV Technology and Application Basics. In Proceedings of the OCEANS 2008-MTS/IEEE Kobe Techno-Ocean, Kobe, Japan, 8–11 April 2008.
3. Mullen, L.; Cochenour, B.; Laux, A.; Alley, D. Optical modulation techniques for underwater detection, ranging and imaging. *Proc. SPIE.* **2011**, *8030*, 803008.
4. Williams, D.P. On optimal AUV track-spacing for underwater mine detection. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4755–4762.
5. Li, B.; Zhao, R.; Xu, G.; Wang, G.; Su, Z.; Chen, Z. Three-Dimensional Path Planning for an Under-Actuated Autonomous Underwater Vehicle. In Proceedings of the 29th International Ocean and Polar Engineering Conference, Honolulu, HI, USA, 16–21 June 2019.
6. Hernández, J.D.; Vidal, E.; Vallicrosa, G.; Galceran, E.; Carreras, M. Online path planning for autonomous underwater vehicles in unknown environments. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1152–1157.
7. Petres, C.; Pailhas, Y.; Patron, P.; Petillot, Y.; Evans, J.; Lane, D. Path Planning for Autonomous Underwater Vehicles. *IEEE Trans. Robot.* **2007**, *23*, 331–341. [CrossRef]
8. Sun, B.; Zhu, D.; Yang, S.X. An Optimized Fuzzy Control Algorithm for Three-Dimensional AUV Path Planning. *Int. J. Fuzzy Syst.* **2018**, *20*, 597–610. [CrossRef]
9. Sociological Research. Artificial Intelligence-A Modern Approach. *Appl. Mech. Mater.* **2009**, *263*, 2829–2833.
10. Wehbe, B.; Hildebrandt, M.; Kirchner, F. Experimental evaluation of various machine learning regression methods for model identification of autonomous underwater vehicles. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4885–4890.
11. Kawano, H.; Ura, T. Motion planning algorithm for nonholonomic autonomous underwater vehicle in disturbance using reinforcement learning and teaching method. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), Washington, DC, USA, 11–15 May 2002; Volume 4, pp. 4032–4038.
12. Yang, G.; Zhang, R.; Xu, D.; Zhang, Z. Local Planning of AUV Based on Fuzzy-Q Learning in Strong Sea Flow Field. In Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization, Sanya, China, 24–26 April 2009; Volume 1, pp. 994–998.

13. Liu, B.; Lu, Z. AUV path planning under ocean current based on reinforcement learning in electronic chart. In Proceedings of the 2013 International Conference on Computational and Information Sciences, Shiyang, China, 21–23 June 2013; pp. 1939–1942.
14. Cheng, Y.; Zhang, W. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* **2018**, *272*, 63–73. [CrossRef]
15. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
16. Cui, R.; Li, Y.; Yan, W. Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT*. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *46*, 993–1004. [CrossRef]
17. Xu, H.; Zhang, G.C.; Sun, Y.S.; Pang, S.; Ran, X.R.; Wang, X.B. Design and Experiment of a Plateau Data-Gathering AUV. *J. Mar. Sci. Eng.* **2019**, *7*, 376. [CrossRef]
18. Barto, A.G. *Reinforcement Learning. A Bradford Book*; MIT Press: Cambridge, MA, USA, 1998; Volume 15, pp. 665–685.
19. Miletić, S.; Boag, R.J.; Forstmann, B.U. Mutual benefits: Combining reinforcement learning with sequential sampling models. *Neuropsychologia* **2020**, *136*, 107261. [CrossRef] [PubMed]
20. Watkins CJ, C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
21. Wei, Q.; Song, R.; Xu, Y.; Liu, D. Iterative Q-learning-based nonlinear optimal tracking control. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016.
22. Wei, Q.; Liu, D.; Song, R. Discrete-time optimal control scheme based on Q-learning algorithm. In Proceedings of the 2016 Seventh International Conference on Intelligent Control and Information Processing (ICICIP), Siem Reap, Cambodia, 1–4 December 2016; pp. 125–130.
23. Cherry, J.M.; Adler, C.; Ball, C.; Chervitz, S.A.; Dwight, S.S.; Hester, E.T.; Weng, S. SGD: Saccharomyces Genome Database. *Nucleic Acids Res.* **1998**, *26*, 73–79. [CrossRef] [PubMed]
24. Zhang, J.; McInnes, C.R. Reconfiguring smart structures using approximate heteroclinic connections. *Smart Mater. Struct.* **2015**, *24*, 105034. [CrossRef]
25. Zhang, J.; McInnes, C.R. Using instability to reconfigure smart structures in a spring-mass model. *Mech. Syst. Signal Process.* **2017**, *91*, 81–92. [CrossRef]