electronics

MDPI

Article

Hierarchical Motion Planning of AUVs in Three Typical Marine Environments

Xiangrui Ran, Hongyu Bian, Guocheng Zhang and Yushan Sun *

Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin 150001, China; ranxiangrui@hrbeu.edu.cn (X.R.); bianhongyu@hrbeu.edu.cn (H.B.); zhang_china2018@163.com (G.Z.)
* Correspondence: sunyushan@hrbeu.edu.cn

Abstract: Broad waters, harbor waters, and waterway waters make up more than 90% of autonomous underwater vehicles (AUV) navigation area, and each of them has its typical environmental characteristics. In this paper, a three-layer AUV motion planning architecture was designed to improve the planning logic of an AUV when completing complex underwater tasks. The AUV motion planning ability was trained by the improved deep deterministic policy gradient (DDPG) combined with the experience pool of classification. Compared with the traditional DDPG algorithm, the proposed algorithm is more efficient. Using the strategy obtained from the training and the motion planning architecture proposed in the paper, the tasks of AUVs searching in broad waters, crossing in waterway waters and patrolling in harbor waters were realized in the simulation experiment. The reliability of the planning system was verified in field tests.

Keywords: autonomous underwater vehicle; motion plan; marine environment; deep deterministic policy gradient; artificial expertise

Citation: Ran, X.; Bian, H.; Zhang, G.; Sun, Y. Hierarchical Motion Planning of AUVs in Three Typical Marine Environments. *Electronics* **2021**, *10*, 292. https://doi.org/10.3390/electronics10030292

Academic Editor: Jose Eugenio Naranjo
Received: 29 December 2020
Accepted: 20 January 2021
Published: 26 January 2021

1. Introduction

The autonomous underwater vehicle (AUV) is an important tool in the field of ocean development [1]. It can be used for environmental monitoring and topographic detection in broad waters, patrolling in harbor waters, and crossing in waterway waters. Broad waters, harbor waters, and waterway waters make up more than 90% of AUVs navigation area, and each of them has its typical environmental characteristics. Broad waters belong to the ideal area for AUVs navigation, with large space for avoidance and few restrictions on behavior and action. There are irregular dynamic and static obstacles in harbor waters, and AUVs can move within a narrow range. The complex waterway environment and obstacles such as fast-moving ships require real-time high performance AUV motion planning. When an AUV performs underwater tasks, it is faced with complex environments and situations such as sudden obstacles and unknown water environment. Therefore, it is urgent to improve AUV's motion planning capability under complex dynamic environment [2].

Eichhorn, Mike et al. [3] proposed several important requirements for optimal motion planning for the "SLOCUM Glider" AUV, as well as a solution based on fast graph algorithms. According to the marine environment of the Continental shelf of Newfoundland and Labrador, this paper planned the optimal navigation path of the waterway area.

Sun, Zhu et al. [4] proposed an optimized fuzzy control algorithm for AUVs motion planning. The model could be used for motion planning in complex underwater environment. The virtual acceleration and speed of an AUV were obtained by using the environmental information collected by sonar and the fuzzy system with an acceleration and braking module, so that the AUV could avoid dynamic obstacles automatically. However, considering that the choice of a fuzzy boundary is subjective, the generated path could not be guaranteed to be optimal. The following year, they [5] proposed a new discrete centralized planning strategy based on glasius bio-inspired neural network for AUVs full coverage motion planning. The algorithm had low computational cost and high efficiency.

It solved the problem that a single AUV is difficult to carry out full coverage task with long range. Ramos, Garcia-Garrido et al. [6] planned a path to optimize scientific impact and navigation efficiency according to the complex space-time structure of ocean flow field and dynamic system. They pre-planned the Silbo glider's mission to cross the North Atlantic from April 2016 to March 2017. The planning capability of the system in broad waters and waterway waters was demonstrated. MahmoudZadeh, Somaiyeh et al. [7] designed a hierarchical dynamic task planning framework for an AUV to complete task assignment within a limited time in an uncertain underwater environment. An advanced reactive task planner was developed to guide an AUV towards the target and finish the task on time. They designed a low-level motion planner to handle unexpected changes in dynamic terrain by regenerating the optimal trajectory. Based on the biogeography optimization (BBO) algorithm, the task was rearranged by updating the terrain, and the motion planning of broad water area was verified.

The above research content only focuses on a single water area and lacks detailed obstacle avoidance strategies. Different underwater environments have different characteristics, and AUVs motion planning needs a logical, detailed and accurate hierarchical structure.

In order to solve the problem of AUVs planning in three typical underwater environments, a three-layer AUV motion planning architecture is proposed, and the planning logic of AUV in complex tasks is given in the paper. At the same time, in order to reduce the complexity of programming, the improved reinforcement learning method is used to train the planning strategy of AUVs. DDPG algorithm is improved in the aspect of critic network. The improved algorithm combined with the classified experience pool is used to train the AUV motion planning ability. Compared with the traditional DDPG algorithm, the proposed algorithm is more efficient. In addition, other traditional methods, such as A* and ant colony algorithm, can only carry out global optimization and are easy to fall into local minimum. By contrast, the method proposed in this paper can obtain the local obstacle avoidance strategy for AUVs, and the intelligence level of AUVs can be improved.

The structure of the paper is as follows: The Section 2 establishes the three-layer motion planning architecture of the AUV. In Section 3, an improved reinforcement learning algorithm is proposed, and an algorithm model is established by combining with the artificial experience pool. In Section 4, the comparative experiment of the algorithm and the motion planning experiment of an AUV under three typical environments are carried out. The fifth chapter carries on the field experiments of an AUV motion planning. Section 6 concludes the study.

2. The Hierarchy of Motion Planning

AUV motion planning is divided into three levels of "task-behavior-action". The work which an AUV needs to complete is defined as tasks, such as: motion planning, target following, terrain detection, underwater search, etc. Target commands generated by an AUV underwater navigation are defined as actions, such as: left turn, right turn, forward, etc. The set of a series of actions generated by an AUV in order to complete the task is defined as the behavior, such as obstacle avoidance, target search, path following, etc. The three levels can be understood as: task decomposition layer, behavior planning layer, and action execution layer. The task decomposition layer breaks down the task commands received by the AUV into behaviors. The behavior planning layer plans the behavior based on the environmental information. The action execution layer uses reinforcement learning method to train AUV's movements to complete the motion planning. The AUV actuator is controlled to generate the action to achieve the target commends.

The motion planning task of AUVs is divided into three layers, as shown in Figure 1. The top layer is the task layer, which is the motion planning task. The second layer is the behavior layer, including obstacle avoidance behavior and navigation to the target. The third layer is the action layer, including: change velocities, change directions, up and down, emergency braking and back action.

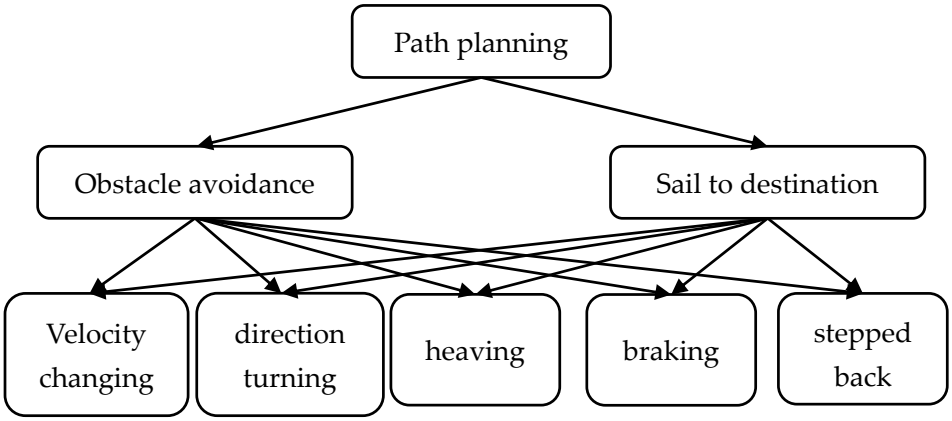

Figure 1. The hierarchy of motion planning.

The task layer is the highest level of AUV learning system, the last learning layer and the slowest learning layer. Therefore, in practical design, task design at the root task level is relatively simple, making it easy for the AUV to learn experience and make decisions. Decision design is the most important part of the root task level, so a better decision strategy should be designed for the root task layer.

The behavior layer realizes the analysis of environmental information, the update of knowledge and the choice of behavior. The AUV must analyze the environmental information collected by the sensor and send the analysis result to the decision-making strategy module, and then make the choice of behavior according to the corresponding policy decision. At the same time, the learning system can update the analysis results to its knowledge database, and then send the knowledge database to the decision-making strategy module of this layer.

The action layer selects actions according to its own decision-making strategies. It is the bottom layer of the hierarchy and also the execution layer of the motion planning task. There are options for multiple actions that are executed immediately after selected.

The decision-making process of the system is a top-down process. The task layer does not directly select policies according to the detected environment state, but makes decisions from the top to the bottom layer, until the basic action instructions to be executed are finally decided.

Decision making is a bottom-up learning process. First, learn the action layer. Then, the behavioral layer learns to select strategies for each behavior. Finally, the task layer learns how to implement the AUV's final task decision. This process can be extended based on decision problems.

The planning logic of an AUV navigation is as follows:

1. Load the existing map environment and initialize the parameters.
2. The planning mission is confirmed and the AUV begins to sail.
3. According to the multi-constraint model, the optimal navigation behavior to the target point is planned.
4. Real-time sensor data is received to determine the AUV location and supplement the environmental map.
5. Whether the AUV has reached the target point: Yes, end the task. No, proceed to the next step.
6. Whether unknown obstacle is detected: No, go to 7. Yes, go to 8.
7. Whether an AUV deviates from the path of global planning: No, maintain this behavior. Yes, preplanning to navigate to the target. Go to 9.
8. According to the multi-constraint model, the optimal obstacle avoidance behavior is programmed. Go to 9.
9. Plan actions;
10. The AUV performs one step of the planned action, go to 4.

3. Motion Planning Algorithm Modeling

The AUV's ability to reach the target and avoid obstacles is trained by improved deep deterministic policy gradient (DDPG) [8], combining artificial experience [9] with reinforcement learning algorithms. Artificial experience is used to improve the training efficiency of reinforcement learning. Reinforcement learning is used to optimize the obstacle avoidance strategy of artificial experience.

3.1. The Improved DDPG

DDPG combines Actor Critic [10] and Deep Q Network (DQN) [11], which is to apply the memory banks in the DQN structure and the idea of two neural networks update on Actor Critic. The Critic network in DDPG only evaluates the overall actions' performance. However, there are 6-dof outputs which are coupled with each other in the motion system of AUVs, including: longitudinal velocity u, lateral velocity v, angular velocity ω, heeling angle φ, trim angle θ, and heading angle ψ. The critic network does not evaluate the actions in each dimension, so the difference between the state value of the optimal and non-optimal actions is small, resulting in low learning efficiency of the algorithm. This study proposes to establish the average motion critic network to solve this problem.

DDPG is divided into two parts: Main Net and Target Net, as shown in Figure 2. Each part includes an ActorNet and six CriticNets. The ActorNet outputs actions of 6-dof, and the CriticNet evaluates each action. Considering a standard Reinforcement Learning (RL) problem, a finite Markov Decision Process (MDP) which comprises a current state s_t, an action space a_t, a reward function r, and the next state s_{t+1} is established.

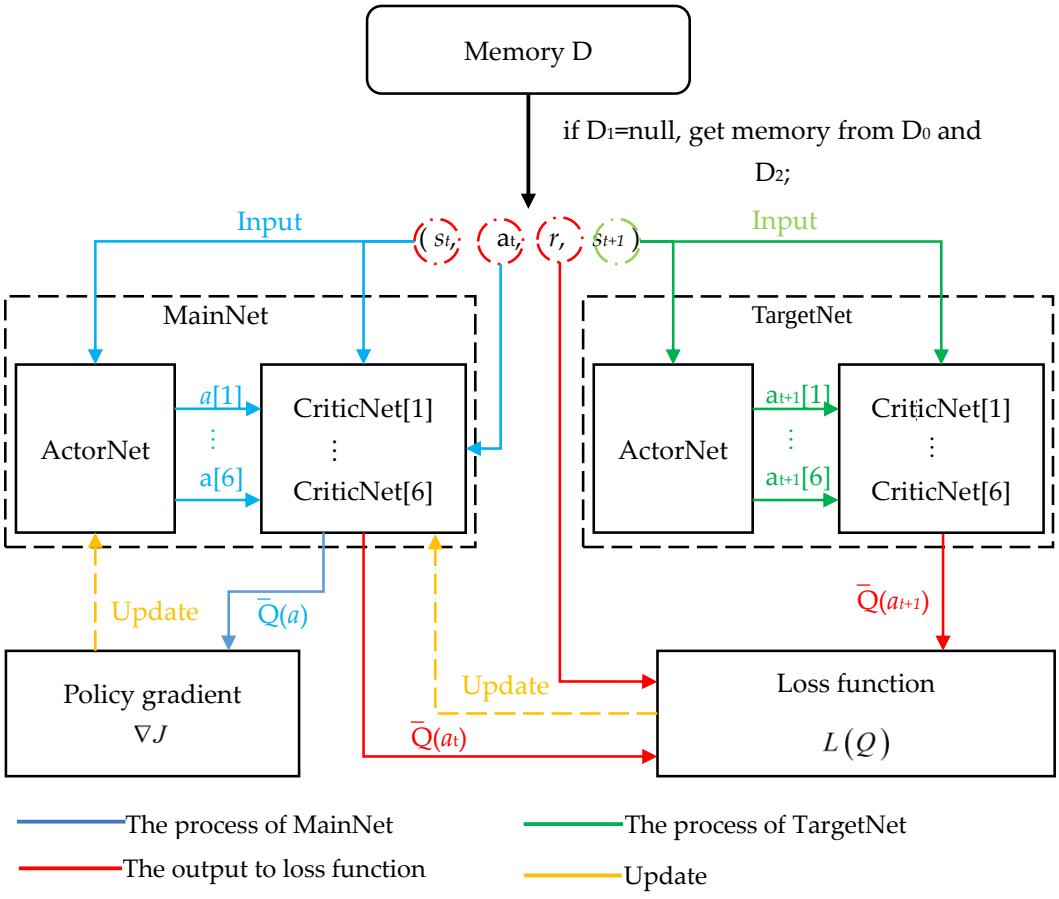

Figure 2. The improved deep deterministic policy gradient (DDPG) algorithm.

Extract data from the experience pool for learning. Input the 6-dof actions at form (s_t, a_t, r, s_{t+1}) into six critic networks of Main Net, and each critic network calculates the value function of each action, respectively, by Bellman equation:

$$Q^{\pi}(s_t[i], a_t[i]) = E_{r_t, s_{t+1} \sim E}\left[r(s_t[i], a_t[i]) + \gamma E_{a_{t+1} \sim \pi}[Q^{\pi}(s_{t+1}[i], a_{t+1}[i])]\right] \quad (1)$$

where, i is between 1 and 6. If the target policy is deterministic, it can be described as a function μ:$s \leftarrow a$, and the inner expectation is as followings:

$$Q^{\mu}(s_t[i], a_t[i]) = E_{r_t, s_{t+1} \sim E}[r(s_t[i], a_t[i]) + \gamma[Q^{\mu}(s_{t+1}[i], \mu(s_{t+1}[i]))]] \quad (2)$$

Get the average Q value $\overline{Q}(s_t, a_t)$:

$$\overline{Q}^{\mu}(s_t, a_t) = \frac{1}{6}\sum_{i=1}^{6} Q^{\mu}(s_t[i], a_t[i]) \quad (3)$$

According to Q-learning [12], considering function approximators parameterized by θ^Q, the loss function is:

$$L(Q) = \overline{R} + \gamma \max_a \overline{Q}\left(s_{t+1}, \mu(s_{t+1}) \middle| \theta^Q\right) - \overline{Q}\left(s_t, a_t \middle| \theta^Q\right) \quad (4)$$

Train the neural network to minimize the loss function so that the actual $\overline{Q}$ value tends to the target $\overline{Q}$ value.

On basis of policy gradient, a parameterized actor function $\mu(s|\theta^{\mu})$ which specifies the current policy by deterministically mapping states to a specific action is maintained. The parameters of the motion estimation network are updated by following function:

$$\begin{aligned} \nabla_{\theta^{\mu}} J &\approx \frac{1}{N}\sum_t \nabla_{\theta^{\mu}} \overline{Q}(s, a|\theta^Q)\Big|_{s=s_t, a=\mu(s_t|\theta^{\mu})} \\ &= \frac{1}{N}\sum_t \nabla_a \overline{Q}(s, a|\theta^Q)\Big|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_t} \end{aligned} . \quad (5)$$

3.2. Obstacle Avoidance Strategy Based on Artificial Experience

The AUV adopts the method of artificial experience to avoid obstacles during underwater navigation.15

(1) The calculation formula of the heading angle is as follows:

$$\beta = \begin{cases} \beta' + \arctan(y_2 - y_1)/(x_2 - x_1) & x_1 < x_2 \\ \beta' + \arctan(y_2 - y_1)/(x_2 - x_1) + \pi & x_1 > x_2 \\ \pi/2 & x_1 = x_2 \text{ and } y_1 < y_2 \\ -\pi/2 & x_1 = x_2 \text{ and } y_1 > y_2 \\ 0 & x_1 = x_2 \text{ and } y_1 = y_2 \end{cases} . \quad (6)$$

where, β is the target heading. β' is the current heading. (x_1, y_1) is the current position of an AUV. (x_2, y_2) is the target position of an AUV.

(2) When an AUV detects a static obstacle within 5 m in front, avoid it. Make an edge in the direction perpendicular to the AUV navigation, while reserving a 2 m safe range to rectangle the obstacle. The rectangle angle nearest to an AUV is taken as the target point and the heading angle is calculated by Formula 6, as shown in Figure 3.

(3) When the AUV detects a dynamic obstacle within 5 m in front, avoid it. The node at the end of the dynamic obstacle is taken as the target point, and the heading angle is calculated based on Formula 6, as shown in Figure 4.

The artificial experience algorithm uses simple logic to avoid obstacles. However, it is not optimal; the logic of the artificial experience algorithm needs to be trained.

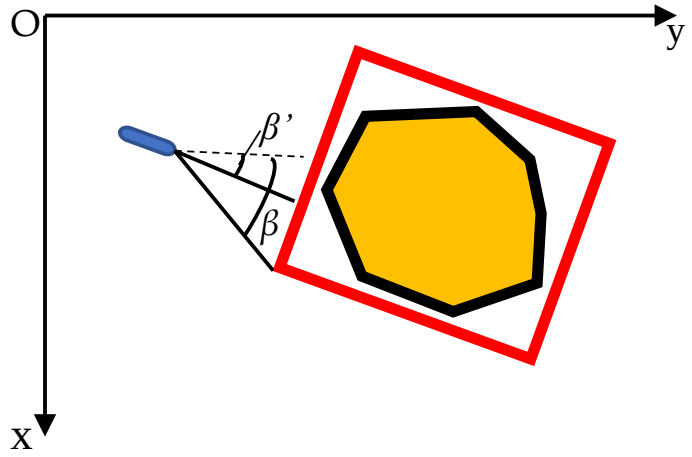

Figure 3. Avoid static obstacles.

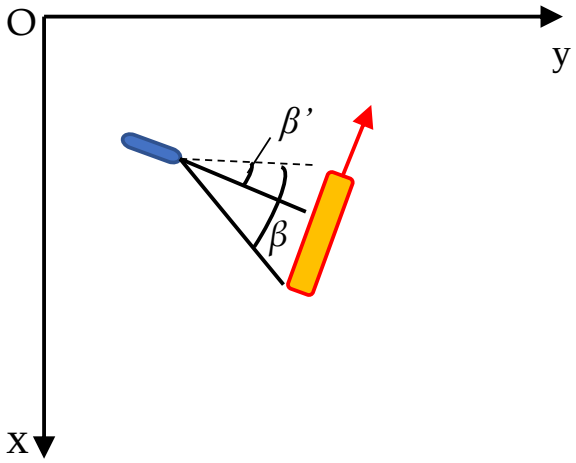

Figure 4. Avoiding dynamic obstacles.

3.3. Sample Space Classification

The sample space of DDPG algorithm is divided into four categories: the original sample space D_0, high error sample space D_1, diversity sample space D_2, and artificial sample space D_3;

In DDPG algorithm, the network samples randomly from the buffer for training. The buffer is finite in size in which (s_t, a_t, r, s_{t+1}) is stored and sampled randomly according to the exploration strategy. However, due to the limited of sampling space size and the general learning effect of samples in the early stage, the learning rate will slow down in the later stage, and the behavior cannot be significantly improved, making the control range of the controller unstable. Therefore, in the second stage of learning, the sample space should be increased to get better samples, so as to obtain more control experience. Before that, the bisimulation is showed as following.

Definition 1. *The bisimulation [13]: If $E \subseteq S \times S$ is the relationship of bisimulation, then for $s_1, s_2 \in S$:*

$$(1)\ \forall a \in A, r(s_1, a) = r(s_2, a); \tag{7}$$

$$(2)\ \forall a \in A, \forall C \in \frac{S}{E}, \sum_{t \in C} P(t|s_1, a) = \sum_{t \in C} P(t|s_2, a); \tag{8}$$

where, the equivalent set of states S with respect to E is denoted by $\frac{S}{E}$. $P(t|s_1, a)$ represents the probability that the system takes an action a and moves to the next state at time t in the process of MDP. If the two states satisfy the bisimulation relationship, it can be denoted as $s_1 \sim s_2$.

Lemma 1. *Bisimulation metric [14]: D is defined as the set of metrics on the state set S, and let* $d \in D$. *For* $\forall s_1, s_2 \in S$, *define:*

$$G(d)(s_1, s_2) = \max_{a \in A}(d_a(s_1, s_2) + \gamma T_K(d)(P(s_1, a)P(s_2, a))) \tag{9}$$

where, $0 \leq \gamma \leq 1$, $d_a(s_1, s_2) = |r(s_1, a) - r(s_2, a)|$, $T_K(d)$ *is the Kantorovich metric [15], which is defined by the following linear program:*

$$\max_{i=1...|S|} \sum_{i=1}^{|S|} (P(s_i) - Q(s_i))u_i. \tag{10}$$

Subject to: $\forall i, j.u_i - u_j \leq d(s_i, s_j)$,$\forall i.0 \leq u_i \leq 1$.

On the basis of the above questions, this study constructed two sample spaces besides the original sample space D_0, namely, high error sample space D_1 and diversity sample space D_2. In the space D_1, the temporal-difference (TD) error is used as the heuristic information to sort samples in the sample space so as to improve the probability of selecting samples with large error. The bisimulation measurement method is used to measure the distance between samples in D_2. During the algorithm learning process, the training samples will be selected in proportion from the D_1 and the D_2, respectively, so as to give consideration to the diversity and value of samples.

The AUV's position within 5 m of the obstacle is set as the dangerous state. The system selects an action based on artificial experience. When the AUV encounters an obstacle and correctly outputs the heading angle according to the artificial experience, the reward value will be 1, which will be stored in the artificial experience pool D_3.

During the process of training, the sample (s_t, a_t, r, s_{t+1}) is put into D_0. TD error of the action function is calculated, as shown in formula (10). When the TD error is larger than a threshold value, it indicates that the sample has a great influence on the change of the action value function, and it can be considered that the modified sample has high value. Thus, the sample is put in D_1. When the samples in D_1 are sufficient, the training samples are not only selected from D_0, but from D_0 and D_1, respectively, in a certain proportion. Meanwhile, Algorithm 1 is used to calculate the distance between samples in D_0. Low-similarity samples are put into D_2 to ensure the diversity of selected samples. After that, Samples from D_1 and D_2 are selected, respectively, in a certain proportion for learning. The distance between samples is calculated by Algorithm 1. The larger the distance, the lower the similarity of samples. When the state of AUV is in danger, the samples is put in D_3.

Algorithm 1 Distance measurement algorithm between states.

Input states s_1 and s_2;
Initialize: $d(s_1, s_2) = 0$, distance parameters γ and ζ.
for $k = 1$ in range $k \leq \frac{\ln \zeta}{\ln \gamma}$ **do**
for $i = 1$ in range $|A|$ **do**
  $T_K(d)(p(s_1, a_i), p(s_2, a_i))$;
End
$d(s_1, s_2) = \max_{a \in A}\{d_a(s_1, s_2) + \gamma T_K(d)(p(s_1, a_i), p(s_2, a_i))\}$;
End

3.4. Algorithm Model Establishment

The algorithm mainly includes three models [11]: input model, output model, and reward value model.

The input model of the algorithm is the state of an AUV, including position coordinates, velocity, and heading of an AUV. The output of the algorithm is the action of an AUV. The reward value model is designed based on multiple constraints. The constraints of motion planning are: following target constraint, obstacle constraint, and current constraint.

A multi-constraint model is established, which includes the position coordinates and attitude of the AUV, ocean currents, and collision distance of obstacles. By adding the weight coefficient of economy, safety and concealment, the navigation value is obtained.

$$C = \varsigma \cdot (\sin\alpha - l_{ob}/L_{\max}) + \zeta \cdot (\cos\alpha + l_{ta}/L_{\max}) \tag{11}$$

C is the constraint. The larger the C, the more costly the AUV navigation. So, C should be as small as possible. α is the angle between the current and the AUV attitude. l_{ta} is the distance between the current position of the AUV and the target position. $L_{\max}$ is the distance from the initial position of the AUV to the target position. l_{ob} is the distance between the current position of the AUV and the obstacle. ς is the security coefficient, and the value is (0,1). The larger ς is, the more security is considered in the planning. The AUV tends to yaw at an angle of 90° with the current. Therefore, the attitude of AUV is planned to avoid a 90° angle to the current. ζ is the economic coefficient, and the value is (0,1). The larger ζ is, the more economical it is considered in the planning. In short, the AUV should try to navigate downstream and avoid obstacles to reach the target as soon as possible.

The multi-constraint model of AUV is designed as the reward value during the training of DDPG algorithm, $R = -C$. The higher the C value, the higher the AUV navigation cost, the worse the reward value. The smaller the C value, the smaller the AUV navigation cost and the better the reward value. The heading of the AUV is treated as an action, which is a continuous output value. The planned combination of actions is the behavior and, in this article, the path.

In addition, if the AUV hits the obstacle, the round ends, and the reward R = −1. If the AUV reaches the target, the round ends, and the reward R = 1. During the path following, the system will be rewarded in real time:

$$R = -l_d/L_{\max} \tag{12}$$

l_d represents the vertical distance between the AUV position and the target path.

4. Simulation Experiment

4.1. Contrast Experiment

AUV motion planning is simulated by using the algorithm designed in this paper. AUV's position coordinates, AUV's current heading angle, speed, distance between the AUV and the target, distance between the AUV and the obstacle are taken as input, and AUV's actions are output to train AUV's ability to follow towards the target and avoid obstacles. The training process is as follows in Algorithm 2:

Algorithm 2 The training process.

Initialize parameters.
for episode in MAX_EPISODES do
Put low-similarity samples into D_1;
Targets and obstacles appear randomly within the environment;
for step in MAX_EP_STEPS:
Learning and output action β= a $=\mu(s_t|\theta^\mu)$;
According to AUV's heading and velocity, the AUV kinematic model is used to calculate its position at the next moment;
Calculate the constraint C, $R = -C$;
if AUV reaches the target:
R = 1;
done = true;
step = 0;
break;
else if collides with obstacles:
R = −1
done = true;
step = 0;
break;
else:
on_goal = 0;
Store (s, a, r, s') in D_0;
if AUV is in critical condition:
Select a certain number of samples from D_3 Randomly;
else if $D_1 \neq$ null:
Select a certain number of samples from D_1 and D_2 Randomly;
else if $D_2 \neq$ null:
Select a certain number of samples from D_0 and D_2 Randomly;
else
Select a certain number of samples from D_0 Randomly;
Update the critic **network**:
$L(Q) = R + \gamma \max_a \overline{Q}(s_{t+1}, \mu(s_{t+1})|\theta^Q) - \overline{Q}(s_t, a_t|\theta^Q)$;
if $L(Q) - \overline{Q}(s_t, a_t|\theta^Q) > p$:
Put (s, a, r, s') in D_2;
if AUV is in critical condition:
Put (s, a, r, s') in D_3;
Update the actor network:
$\nabla_{\theta^\mu} J = \frac{1}{N}\sum_t \nabla_a \overline{Q}(s, a|\theta^Q)\Big|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_t}$;
Update the network parameters:
$\theta^{Q'} = \rho\theta^Q + (1-\rho)\theta^{Q'}, \theta^{\mu'} = \rho\theta^\mu + (1-\rho)\theta^{\mu'}$;
end for step;
end for episode;

After the training, complete obstacle avoidance and following strategies were obtained. The results are shown in Figure 5.

In Figure 5, the abscissa represents the episodes of training, whereas the ordinate represents the total reward of each episode. A total of 10,000 episodes are created during simulation training. Each training episode is updated with 500 steps. Although the learning curves of both artificial experience-DDPG and DDPG algorithm converges to a satisfactory value, the learning process of artificial experience-DDPG is more stable. It can be seen form Figure 5 that artificial experience-DDPG is more effective than conventional DDPG. Figure 5 shows that when the reward value converges to 230–250, the training is successful. In addition, in order for the AUV to learn all the motions, there is a strategy of randomly selecting the actions during the training. So, the curves in Figure 5 do not converge.

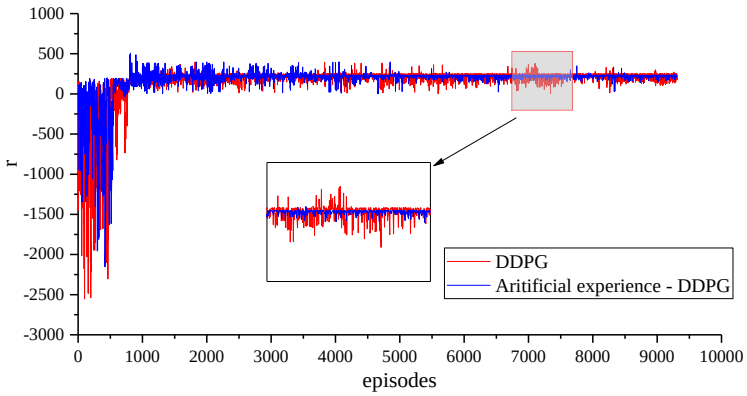

Figure 5. Accumulation of training reward value.

4.2. Simulation Experiment Platform

The architecture of the simulation experiment platform is shown in Figure 6. The system consists of several computers, which are visual simulation computers for the operation environment modeling system, motion planning computers for the operation planning system, and sensor data processor for the operation perception processing system. They integrate into a system via network switches. Motion planning control, environment awareness and information transmission of typical environment model are accomplished through network communication.

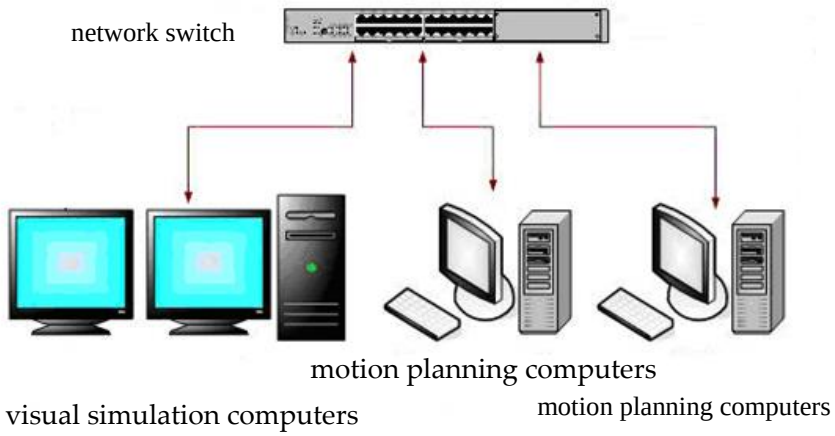

Figure 6. Hardware architecture of the autonomous underwater vehicle (AUV) simulation platform.

A total of three typical environmental models, the AUV dynamics and kinematics models, obstacle models, and target models, were established in the environmental modeling system. The perception processing system detects the target in the environment, filters the perception data, and sends the processed data to the planning system. The planning system plans behaviors and actions according to perceptual data, environmental map and the AUV real-time location. The planning instruction is sent to the environmental system. The AUV is controlled to move in the environment to realize target following and obstacle avoidance. When the sensing system detects a sudden obstacle or a moving obstacle, the planning system carries out re-planning to realize the real-time obstacle avoidance behavior.

4.3. Motion Planning Simulation Experiment

After the simulation platform was built, models of broad waters, waterway waters and harbor waters were established. Training experience was introduced, and simulation experiments were carried out respectively.

In broad waters, the AUV was assigned an area search task. The AUV was greatly influenced by topography. The AUV path following and obstacle avoidance were simulated. The result is shown in Figure 7.

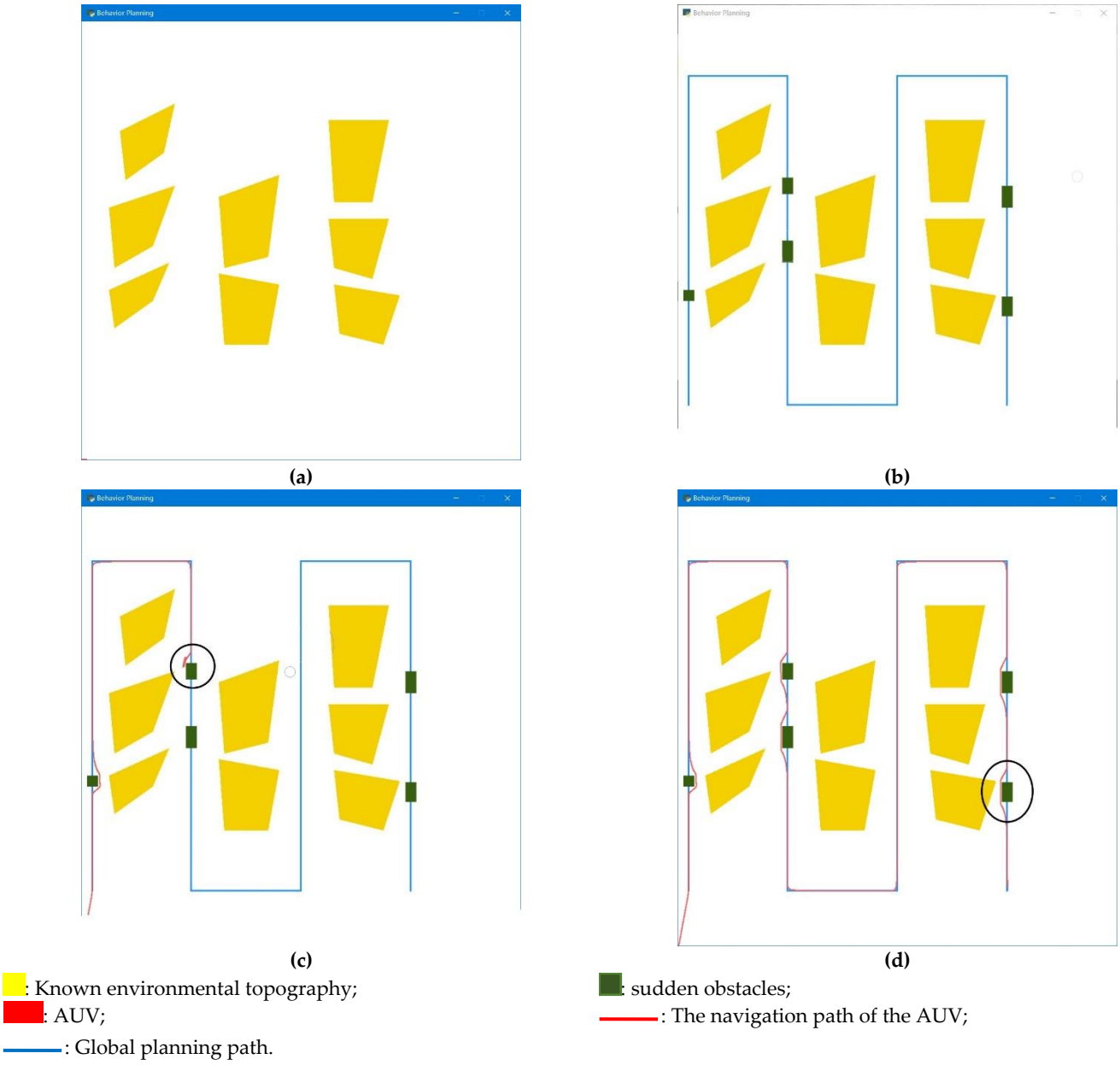

Figure 7. Motion planning of patrol inspection in broad waters. (**a**) marine environment, (**b**) global planning, (**c**) path following and obstacle avoidance, (**d**) end of the test.

In Figure 7, the yellow polygons represent the known environmental terrain. The green polygons represent sudden obstacles. The blue lines are the path of the global plan. The red lines indicate the AUV's path. In this simulation experiment, the AUV path following and obstacle avoidance are realized. Figure 7 shows that in broad waters, the AUV actions can be correctly planned based on the algorithm designed in this paper, enabling the AUV to follow the path and avoid obstacles.

In the waterway, the task of transversal crossing was assigned to the AUV. The AUV was greatly influenced by passing ships. The AUV behavior of avoiding moving obstacles was realized. The result is shown in Figure 8.

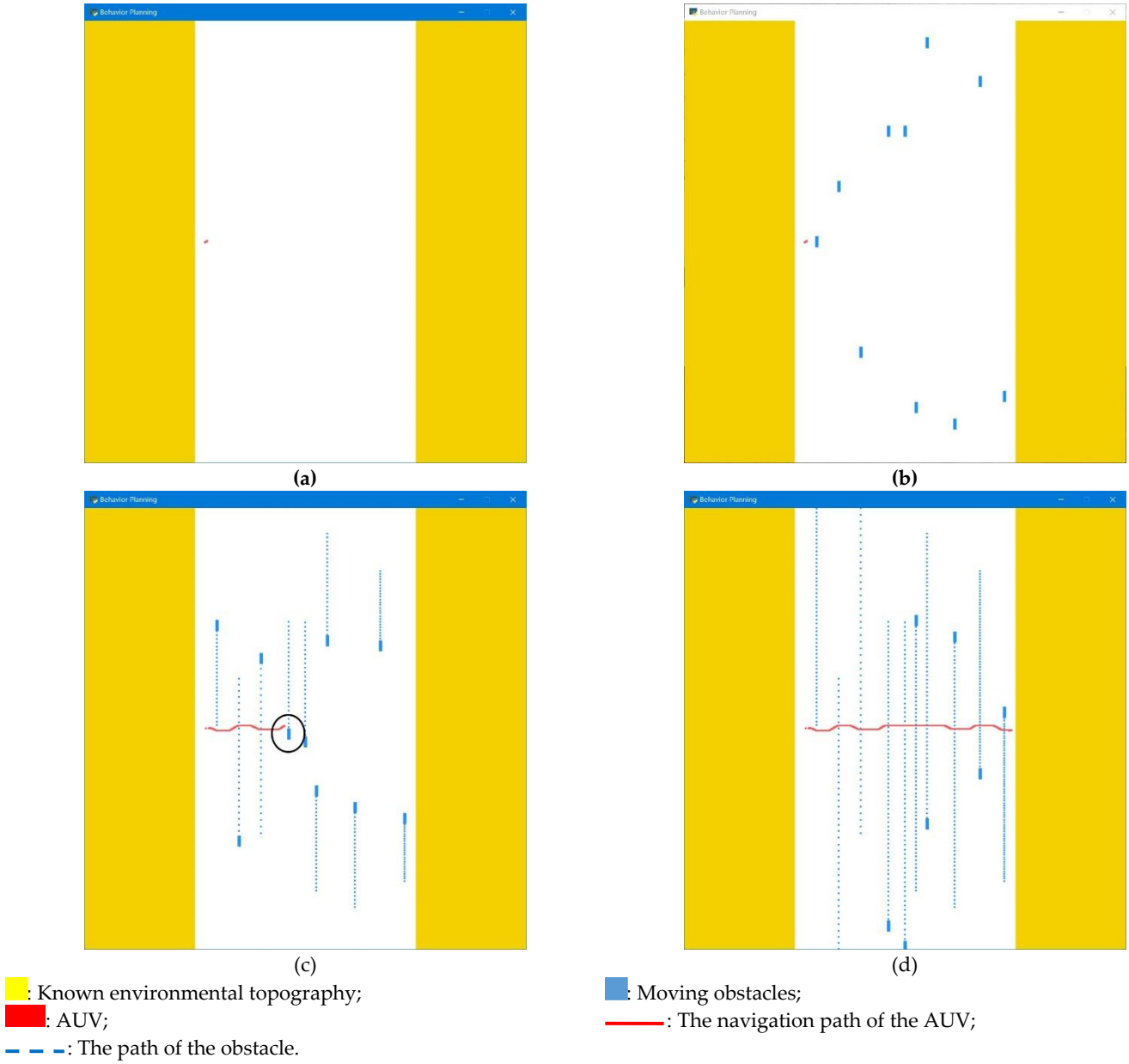

Figure 8. Motion planning of transversal crossing in waterway. (**a**) marine environment, (**b**) moving obstacle, (**c**) obstacle avoidance, (**d**) end of the test.

In Figure 8, the yellow polygons represent the known environmental terrain. The blue polygons represent moving obstacles. The blue lines are the path of the obstacles. The red lines indicate the AUV's path. The motion planning of AUV waterway crossing and obstacle avoidance is realized in the simulation experiment. Figure 8 shows that in the waterway, the algorithm designed in this paper can correctly plan AUV's actions, enabling the AUV to navigate and avoid moving obstacles.

In the harbor waters, the task of patrolling was assigned to the AUV. The AUV was influenced by both terrain and stationary and moving obstacles. However, different from the obstacles with regular movement in waterway, the obstacles in harbor waters area were complex and irregular. AUV patrol and obstacle avoidance were realized. The result is shown in Figure 9.

In Figure 9, the yellow polygons represent the known environmental terrain. The blue polygons represent moving obstacles. The blue lines are the path of the obstacles. The red lines indicate the AUV's path. In this simulation experiment, the AUV patrol

and obstacle avoidance are realized. Figure 9 shows that in harbor waters, the algorithm designed in this paper can correctly plan AUV's actions, enabling the AUV to realize patrol and obstacle avoidance.

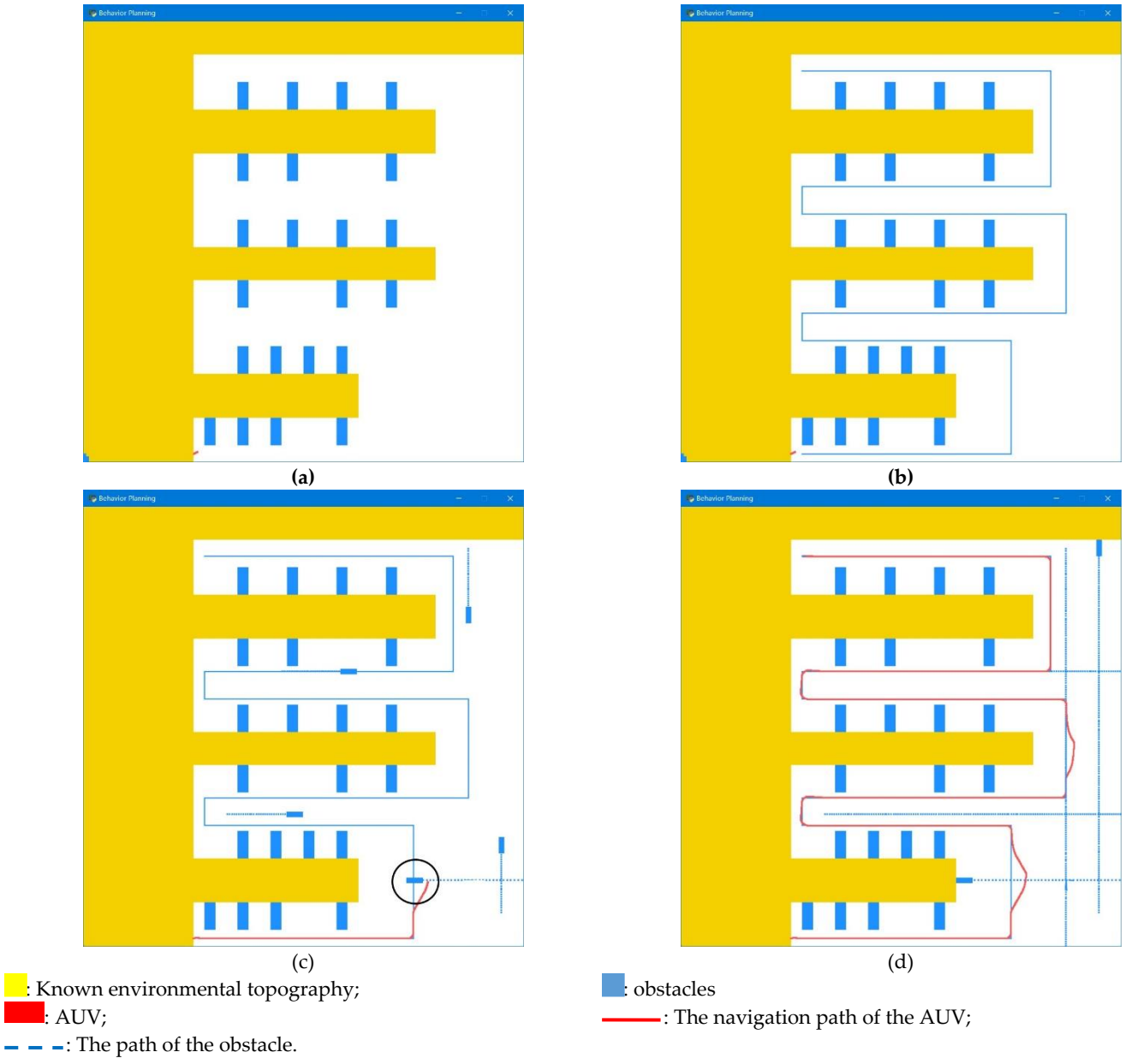

Figure 9. Motion planning of patrolling in harbor waters. (**a**) marine environment, (**b**) global planning, (**c**) path following and obstacle avoidance, (**d**) end of the test.

To sum up, compared with the traditional algorithm, the algorithm designed in this paper has a great improvement in training efficiency. With the experience gained from algorithm training, the AUV motion planning under three typical environments can be well realized, so as to complete the scheduled tasks. In addition, other traditional methods, such as A* and ant colony algorithm, can only carry out global optimization and are easy to fall into a local minimum. By contrast, the method proposed in this paper can obtain the local obstacle avoidance strategy for AUVs, and the intelligence level of AUVs can be improved.

5. Field Tests

The applicability of the algorithm was verified by experiments in the field. The AUV system in the experiment was composed of sensing, planning and control system [16]. The AUV carried out motion planning under the guidance and supervision of global knowledge, and completed the assigned task. As the decision maker of the AUV, the planning system received the information of sensor and navigation system in real time, analyzed and gave corresponding planning instructions, and then sent them to the control system to control the movement of AUV. Within each step, the data of the AUV's current state information (position information, attitude information, planning instructions, etc.) was recorded. Parameters of the AUV were shown in Table 1. The system hardware architecture of the AUV is shown in Figure 10. At the end of the test, the data results were derived to verify the technical indicators.

Table 1. Parameters of the AUV.

Parameters	Value
Length	2060 mm
Maximum Diameter	400 mm
Weight	174.6 kg
Maximum Depth	200 m
Cruising Speed	$\leq$2 m/s
Endurance Time	8 h
Video Storage Time(1080p/30fps)	$\geq$10 h
Sensors	Fiber Optic Gyroscope (FOG) TCM5 magnetic compass Water Leak Sensor Power Supply Monitoring Ranging Sonar Depth Sensor Video Camera
Operating System	VxWorks5.5 (bottom layer) Windows 7 (surface layer)

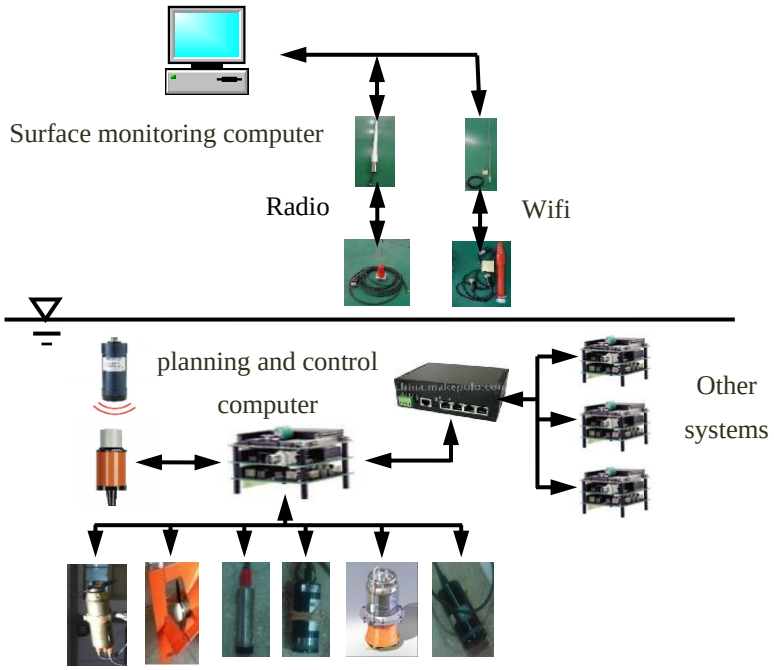

Figure 10. The system hardware architecture of the AUV.

Area search experiment in broad waters was shown in Figure 11 which was completed in an inland river area in Qinghai Province, China. The experimental area is 1000 m × 1000 m. The maximum water depth is 30 m. The AUV navigated along the predetermined route to complete the search task in the test area. The test results are shown in the Figure 12.

Figure 11. Area search test.

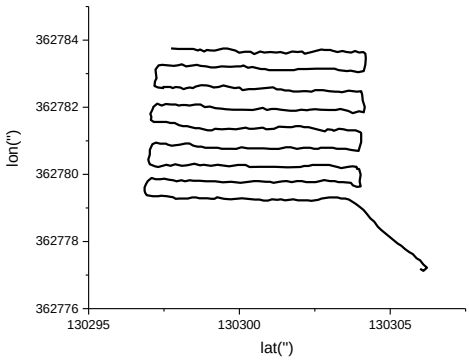

Figure 12. The results of area search test.

The test results show that, based on the AUV motion planning algorithm designed in this paper, the system outputs planning instructions to the control system in real time, and the control system controls the AUV navigation, which can realize the task of the AUV path following in broad waters.

The test of the AUV following along the wall in the waterway was shown in the Figure 13, which was completed in Zhejiang Province, China. The experimental area is 1500 m × 50 m. The maximum water depth is 8m. The ranging sonar mounted by the AUV was used to detect the distance between the AUV and one side of the waterway in real time, and the distance was kept to 3 m. The ranging sonar mounted by the AUV is shown in Figure 14. The test results are shown in the Figure 14. The sonar parameters are shown in Table 2.

As shown in the Figure 15, based on the motion planning system designed in this paper, the AUV can maintain a distance of about 3 m from one side of the waterway. The reliability of the proposed planning system is verified.

Figure 13. The test of AUV following along the wall.

Figure 14. The ranging sonar.

Table 2. The sonar parameters.

The Sonar Parameters
Name: 200 kHz–50 kHz Underwater ranging transducer.
Model: DYW-50/200-NA.
Frequency: 200 kHz ± 5 kHz/50 kHz ± 3 kHz.
Range: 200 kHz 0.6–120 m./ 50 kHz 3–500 m.

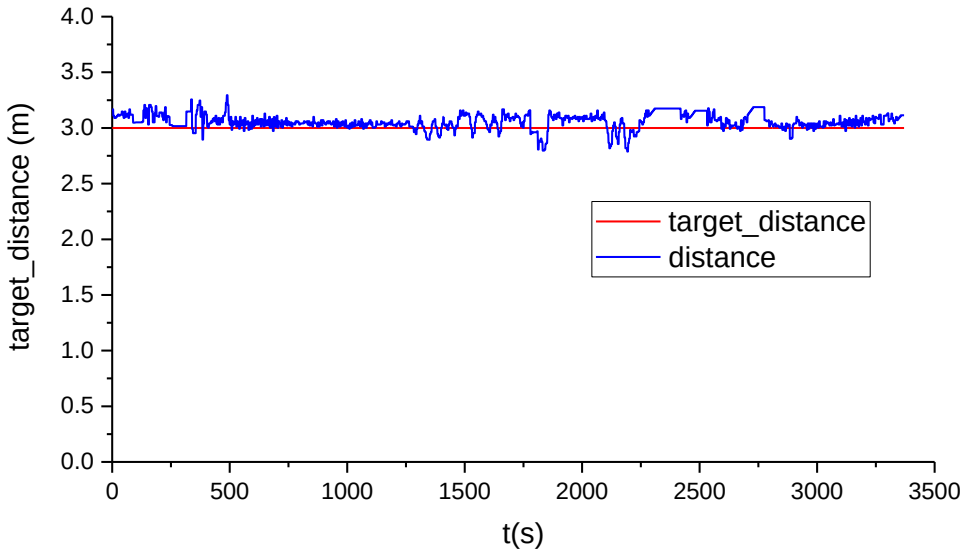

Figure 15. The AUV test results for wall tracking.

6. Conclusions

In order to improve the AUV's motion planning ability in underwater complex tasks, this paper proposes a three-layer AUV motion planning architecture. The logical order of AUV underwater motion planning is pointed out. In the motion critic network, the DDPG algorithm is improved. The AUV obstacle avoidance and target following motion planning ability is trained by combining with classified experience pool. Compared with the traditional DDPG algorithm, the training efficiency of this method is higher. The motion planning simulation test platform of the AUV under three typical environments was built. By using the strategy obtained from training and based on the motion planning architecture designed in this paper, the tasks of the AUV scanning in broad waters, navigating in waterway waters and patrolling in harbor waters were realized in the simulation experiment. The area search test and the waterway following test were carried out in the field. Experimental results were obtained to verify the reliability of the planning system.

Author Contributions: Conceptualization, X.R.; methodology, X.R.; software, X.R.; validation, Y.S., H.B., and G.Z.; formal analysis, X.R.; investigation, H.B.; resources, Y.S.; data curation, X.R.; writing—original draft preparation, X.R.; writing—review and editing, X.R.; visualization, X.R.; supervision, Y.S.; project administration, Y.S.; funding acquisition, Y.S., G.Z., and H.B. 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.

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

References

1. Sun, Y.; Ran, X.; Zhang, G.; Wang, X.; Xu, H. AUV path following controlled by modified Deep Deterministic Policy Gradient. *Ocean Eng.* **2020**, *210*, 107360. [CrossRef]
2. Sun, Y.; Ran, X.; Zhang, G.; Xu, H.; Wang, X. AUV 3D Path Planning Based on the Improved Hierarchical Deep Q Network. *J. Mar. Sci. Eng.* **2020**, *8*, 145. [CrossRef]
3. Eichhorn, M. Solutions for practice-oriented requirements for optimal path planning for the AUV "SLOCUM Glider". In Proceedings of the OCEANS 2010 MTS/IEEE, Seattle, WA, USA, 20–23 September 2010; pp. 1–10.
4. 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]
5. Sun, B.; Zhu, D.; Tian, C.; Luo, C. Complete Coverage Autonomous Underwater Vehicles Path Planning Based on Glasius Bio-inspired Neural Network Algorithm for Discrete and Centralized Programming. *IEEE Trans. Cogn. Develop. Syst.* **2019**, *11*, 73–84. [CrossRef]
6. Ramos, A.G.; García-Garrido, V.J.; Mancho, A.M.; Wiggins, S.; Coca, J.; Glenn, S.; Schofield, O.; Kohut, J.; Aragon, D.; Kerfoot, J.; et al. Lagrangian coherent structure assisted path planning for transoceanic autonomous underwater vehicle missions. *Sci. Rep.* **2018**, *8*, 1–9. [CrossRef] [PubMed]
7. Mahmoudzadeh, S.; Powers, D.M.; Sammut, K.; Atyabi, A.; Yazdani, A. A hierarchal planning framework for AUV mission management in a spatiotemporal varying ocean. *Comput. Electr. Eng.* **2018**, *67*, 741–760. [CrossRef]
8. Xu, Y.-H.; Yang, C.-C.; Hua, M.; Zhou, W. Deep Deterministic Policy Gradient (DDPG)-Based Resource Allocation Scheme for NOMA Vehicular Communications. *IEEE Access* **2020**, *8*, 18797–18807. [CrossRef]
9. Simpson, S.A.; Cook, T.S. Artificial Intelligence and the Trainee Experience in Radiology. *J. Am. Coll. Radiol.* **2020**, *17*, 1388–1393. [CrossRef] [PubMed]
10. Peters, J.; Schaal, S. Natural Actor-Critic. *Neurocomputing* **2008**, *71*, 1180–1190. [CrossRef]
11. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nat. Cell Biol.* **2015**, *518*, 529–533. [CrossRef] [PubMed]
12. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
13. Givan, R.; Dean, T.; Greig, M. Equivalence notions and model minimization in Markov decision processes. *Artif. Intell.* **2003**, *147*, 163–223. [CrossRef]
14. Ferns, N.; Panangaden, P.; Precup, D. Metrics for Finite Markov Decision Processes. In Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, 7–11 July 2004; pp. 162–169.

15. Gibbs, A.L.; Su, F.E. On Choosing and Bounding Probability Metrics. *Int. Stat. Rev.* **2002**, *70*, 419–435. [CrossRef]
16. Sun, Y.; Ran, X.; Zhang, G.-C.; Wu, F.; Du, C. Distributed control system architecture for deep submergence rescue vehicles. *Int. J. Nav. Arch. Ocean Eng.* **2018**, *11*, 274–284. [CrossRef]