Research Article

Reinforcement Learning-Based Autonomous Navigation and Obstacle Avoidance for USVs under Partially Observable Conditions

Nan Yan, Subin Huang , and Chao Kong

School of Computer and Information, Anhui Polytechnic University, Wuhu, China

Correspondence should be addressed to Subin Huang; subinhuang@ahpu.edu.cn

Received 27 February 2021; Accepted 26 April 2021; Published 4 May 2021

Academic Editor: Hegazy Rezk

Copyright © 2021 Nan Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Unmanned surface vehicles (USVs) have been widely used in research and exploration, patrol, and defense. Autonomous navigation and obstacle avoidance, as the essential technology of USVs, are the key conditions for successful mission execution. However, fine modeling of conventional algorithms cannot meet the real-time precise behavior control strategy of USVs in complex environments, which poses a great challenge to autonomous control policy. In this paper, a deep reinforcement learning-based UANO (USVs autonomous navigation and obstacle avoidance) method is proposed. The UANO achieves the autonomous navigation task of USVs by real-time sensing of partially complex ocean information around and real-time output of rudder angle control commands of USVs. In our work, we employ a double Q-network to achieve end-to-end control from raw sensor input to output of discrete rudder action, and design a set of reward functions that can be adapted to USV navigation and obstacle avoidance. To alleviate the decision bias caused by partial observable of USVs, we use the long short-term memory (LSTM) networks to enhance the ability to remember the ocean environment of USVs. Experiments demonstrate that UANO ensures a USV arrives at the target points with optimal path planning in complex ocean environments without any collisions occurring, and UANO outperforms deep Q-network (DQN) and random control policy in convergence speed, sailing distance, rudder angle steering consumption, and other performance measurements.

1. Introduction

USVs are primarily used to perform tasks that are dangerous and unsuitable for manned vessels. When the USV is equipped with a variety of customized sensors, communication devices, and other equipment, etc., it will have greater flexibility and intelligence to perform a variety of complex maritime tasks [1, 2]. USVs have fully demonstrated their huge advantages in reducing risks and improving mission efficiency no matter in the civil field or the military field. Combining USVs with other unmanned systems, they can build rich clusters of unmanned systems in ocean, capable of handling more complex maritime missions [3, 4].

USVs encounter different marine environments in different mission scenarios and often fail in their missions due to the harsh marine environment. Therefore, the autonomous navigation and obstacle avoidance capabilities of

USVs are highly required. That is to say, under certain constraints, the USV will depart from the initial location and adjust its navigation route in real time according to the changes in the external environment to reach the final destination.

Traditional methods (e.g., dynamic window method [5] and artificial potential field method [6]) have been used in the USV's navigation, path planning, and obstacle avoidance. However, the shortcomings of the traditional methods are obvious. They easily plunge into the local optimum in the complex ocean scenarios and thus fall into traps, resulting in a low probability of finding a reasonable route to the target location. Moreover, under partial observation conditions, conventional algorithms cannot plan the environment completely in advance, which is the reason for the difficulty of autonomous navigation and obstacle avoidance of USVs in complex environments.

In recent years, with the increase of computing power, artificial intelligence technology has been fully developed, an increasing number of researchers [7, 8] have started to adopt deep learning-based methods to achieve end-to-end learning for complex tasks after gradually converting the transformation of low-level feature representations into high-level feature representations through a multilayer process. Besides, reinforcement learning is based on trial-and-error mechanism from experiences to optimize control strategies, which is a widespread application in the autonomous control of unmanned systems. Tian Li et al. [9] provided an online learning framework for the smart grid by using reinforcement learning. Recurrent neural networks such as LSTM, which can find out the intrinsic hidden associative knowledge from time series, have been widely used in tasks such as text processing [10] and image analysis [11].

Inspired by the above deep learning and deep reinforcement learning, we proposed the UANO algorithm for USV's autonomous navigation and obstacle avoidance task. The contributions of this paper are as follows:

An end-to-end deep reinforcement learning-based UANO method is proposed to realize the navigation and obstacle avoidance tasks for USVs in complex marine environments by analyzing the operational actions of the USV and sensing the environment during the navigation process

To alleviate the problem of large bias of USV control policy under partial perception conditions, we use an LSTM network to perform temporal learning of the perceived environment to learn the hidden association information from the environment, thus allowing USV to learn more stably under limited perception conditions

We designed the dynamics model of the USV based on open source software, the scenario is more realistic, and the USV perceives the complex ocean environment in a radar-ranging way, which is more advanced than most other works

Our paper is organized as follows. In Section 2, we detail related work in the field of USVs autonomous control. In Section 3, we simulate the dynamical model of the USV and present the end-to-end UANO method in detail. The experimental results and comparison of the relevant algorithms are given in Section 4. Finally, in Section 5, the conclusions and future works are presented.

2. Related Work

In complex marine environments, USVs use a variety of sensors, such as radar sensors and visual sensors, to sense their surroundings and make their next decision in conjunction with their mission. Therefore, USVs should be capable of autonomous intelligent behavior control so that they can make the right decisions to ensure mission accomplishment in complex and harsh ocean environments.

Conventional approaches for USVs autonomous navigation are using mathematical analysis. In [12], an improved

ant colony clustering algorithm is proposed, which makes full use of the limited computational resources of USVs and combines complexity clustering in different environments to select the search range, thus improving the path planning performance. In [13], an energy-efficient path method considering energy consumption is proposed, which is based on the ocean current data to calculate a suitable navigation path. Singh et al. [14] proposed an extension approach method that meets the requirements of ship obstacle avoidance in complex marine environments and different current strength environments. Mina et al. [15] presented a generalized multi-USV navigation, guidance, and control framework adaptable to specific USV maneuvering response capabilities for dynamic obstacle avoidance. Singh et al. [16] proposed a hybrid framework for guidance and navigation of swarm of USVs by combining the key characteristics of formation control and cooperative motion planning.

In recent years, deep learning and reinforcement learning have developed rapidly [17–19], which provides new ideas for solving path planning and obstacle avoidance problems in complex dynamic environments. Based on Q-learning and combined with deep learning, DeepMind proposed the DQN [20] algorithm, which has been applied in video games to surpass human beings, and DQN is now also widely used to address the behavioral control of intelligences in complex dynamic environments [21, 22]. Simultaneously, deep reinforcement learning-based path planning and obstacle avoidance tasks have been well-studied on unmanned vehicles with great success [23, 24].

As the advantages of deep and reinforcement learning are demonstrated in practical applications, a growing number of researchers are exploring the use of deep reinforcement learning to achieve autonomous navigation and obstacle avoidance in USVs. Liu et al. [25] presented a dynamic multiple step reinforcement learning algorithm that is based on virtual potential field path planning. Wang et al. [26] proposed a deep reinforcement learning method combined with a well-designed reward function to achieve USV obstacle avoidance in complex environments. Long et al. [27] proposed an end-to-end obstacle avoidance policy for generating efficient distributed multi-agent navigation, and they gave the expression of the obstacle avoidance navigation policy as a deep neural network mapped from the observed environmental information to the agent's movement speed with steering commands. Woo et al. [28] developed a path-following controller using deep reinforcement learning; the controller they built enables the path-following capability of the vehicle by interacting with the environment. Zhou et al. [29] investigated deep reinforcement learning algorithms in USV formation path planning for applications in complex ocean environments, including tasks such as reliable obstacle avoidance in complex marine environments. Liu et al. [30] built a novel cooperative search algorithm based on RL method and probability map method and greatly improved the USV detection capability.

In summary, the inherent deficiencies of traditional methods are not sufficient for autonomous navigation and obstacle avoidance of USVs in complex marine

environments. The advantages of deep reinforcement learning can make up for the deficiencies of traditional methods. In our work, we propose a deep reinforcement learning-based UANO method for USVs to achieve collision-free navigation from initial to target points in complex marine environments. In addition, we have verified the performance and generalization capability of various methods such as UANO, DQN, and random control policy on a highly simulated platform.

3. Method

3.1. The Kinematic Motion of USVs. Horizontal surface motion of the USV is the primary consideration for its autonomous navigation, so we have to focus on the 3-degree-of-freedom motion of the USV, namely, surge, sway, and yaw [31]. Based on that, $v = [u, v, r]^T \in \mathbb{R}^3$ and $\tau = [x, y, \phi]^T \in \mathbb{R}^3$ are selected as the velocity vector and position vector. In these, (ϕ) is the heading of the USV and (x, y) is the position of Earth global coordinates. The linear velocities $v = [u, v, r]^T \in \mathbb{R}^3$ correspond to surge, sway, and yaw. The kinematic motions of the USV can therefore be written as

$$\eta = Jv, \quad (1)$$

$$Mv = \tau - C(v)v - D(v)v + \tau_w, \quad (2)$$

where

$$J = J(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

and M is the mass matrix, $C(v)$ is the Coriolis and centripetal matrix, and $D(v)$ the damping matrix. The control input vector τ denotes the propulsion surge force and the yaw moment, which is given by

$$\tau = [\tau_u, 0, \tau_r]^T \in \mathbb{R}^3. \quad (4)$$

We consider the use of an underdriven the USV control system to further simplify the USV hardware configuration. By analyzing the propulsion force and moment vector τ , we can conclude that the independent sway control mechanism can be simplified away, which will make the USV more adaptable and have lower hardware requirements. The disturbance from the environment can be represented by the vector τ_w , which is given by

$$\tau_w = [\tau_{wu}, \tau_{wv}, \tau_{wr}]^T \in \mathbb{R}^3, \quad (5)$$

where τ_{wu}, τ_{wv} are the disturbance force on the surge and sway, respectively, while τ_{wr} is the disturbance moment on the yaw. Figure 1 shows major components of the USV.

3.2. Observation and Action of USVs

3.2.1. Observation of USVs. First and foremost, the USV needs to be aware of its surroundings while performing its mission so that it can make the right decisions, such as how

FIGURE 1: The core components of the USV, which can meet the behavior control of the USV in complex environments. It should be noted that USVs may change their heading angle or position during navigation due to currents and other effects.

the rudder angle and throttle should be manipulated. In general, USVs are configured with multiple sensors that work together to accomplish the task. USVs are equipped with three basic sensors in their missions, namely, camera, LIDAR, and RADAR. Among them, camera, as the most common sensor, has been deployed to USVs in large numbers, and its problem is that it cannot track distance information. RADAR makes up for the shortcomings of camera sensors, but still has the disadvantages of low accuracy and short visible range. LIDAR measures the relative distance between the edge of the object contour in the field of view and the device by emitting a laser beam, which is more accurate and has a longer range than radar. Using LIDAR information as the perceptual input of USVs can make the reinforcement learning navigation obstacle avoidance method of USV more stable and more accurate decision-making.

In our work, we mainly use a laser ranging sensor and a position sensor to work together to accomplish the navigation and obstacle avoidance task of the USV. The main purpose of the laser ranging sensor is to sense obstacles in the surroundings of the USV. In our work, the laser ranging sensor works on the principle that the USV shoots several rays in front of it, which are reflected back when they hit an obstacle, thus giving the distance between the USV and the obstacle. The position sensor gets real-time position information of the USV in the sea, which allows calculating the real-time distance between the USV and the target location to assist the USV in making the right decision.

As shown in Figure 2, based on the position sensor, the USV gets its own position information in real time. However, it cannot get the position information of the surrounding obstacles. In order to be able to perceive information about the surrounding obstacles, we have

FIGURE 2: Two types of sensors used in our work: the position sensor returns location information in real time; the laser ranging sensor senses the surrounding ocean environment.

configured a laser ranging sensor. The laser ranging sensor is able to detect obstacles in a timely manner within an effective perception range. The accuracy of the laser ranging sensor is affected by its detection distance and the number of rays. In Figure 1, the effective detection distance of the laser ranging sensor is 30 meters, and the rays are uniformly directed forward, with 13 rays.

3.2.2. Action of USVs. The USV is a typical underdriven kinematic system that uses the throttle and rudder angle to counteract swells and surges, to achieve path planning and obstacle avoidance, and to reach the target position to carry out the mission. In our paper, we consider a kinematic model of the USV, where the behavioral control action taken by the USV is to operate the rudder angle, by changing the rudder angle to accomplish a change in the USV's heading, and assuming that the USV cruises at a fixed horsepower. If dynamics-propeller forces are considered as the output of the algorithm, then the algorithm will have a stronger generalization capability after convergence, and USV can learn how to back up to avoid obstacles in more complex environments through the cooperation of dynamics-propeller forces and rudder angle. If the throttle is fixed, then the USV moves forward by default and USV may be caught in a local dilemma due to poor decision-making. However, dynamics-propeller and rudder angel as the algorithm's input will make model more difficult to train, and for this reason, we fixed the propulsion of the USV.

Existing work often defines the motion space of a USV as up, down, left, right, and right [32, 33]. In this paper, we redefine a more fine-grained set of discrete motion spaces as follows:

$$A = [a_1, a_2, \dots, a_7, \dots, a_{12}, a_{13}]. \quad (6)$$

In equation (6), the action space of the USV A is a vector that contains 13 elements with the value ranging from -60° to 60° , and with increments of 10° . This design is more consistent with the actual manipulation of USVs. Note that, in our work, we fixed the throttle size to allow the USV to sail at a fixed speed.

3.3. Deep Reinforcement Learning-Based UANO Algorithm. In the above, we have analyzed the observation space and action control of the USV. In this part, we start to design the UANO algorithm to realize the navigation and obstacle

avoidance of the USV in the complex ocean environment. Firstly, we introduce Markov Decision Processes (MDPs) that are typically used to solve time-series complex decision tasks for modeling; secondly, we evaluate the advantages of deep reinforcement learning with double Q-learning, and finally, we combine the observation space and control of the USV to derive the UANO algorithm.

3.3.1. Markov Decision Process. In this paper, we use Markov Decision Process (MDP) for modeling USV navigation and obstacle avoidance task. An MDP describes an intelligence learning process of decision-making strategies by interacting with the environment, and an MDP can be represented by a 5-element tuple:

$$M = [S, A, P, \gamma, R], \quad (7)$$

where S is the state space sensed by sensors during the navigation and obstacle avoidance of the USV and A is the action space of the USV, which is defined by equation (6). R is our own defined reward function; the USV will constantly adjust its behavior strategy according to the reward and will eventually achieve accurate navigation and obstacle avoidance. γ is the discount factor that determines the present reward of the discounted future reward. P is the transition probability function defined as follows:

$$P(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \dots). \quad (8)$$

MDP for an USV and environments is shown in Figure 3. The USV selects an action a_t under the observed environment state s_t ; the environment will update the state $s_{(t+1)}$ based on the behavior and the state transition probability P and returns an immediate reward $r_{(t+1)}$ to the USV.

3.3.2. Enhanced State Awareness. In Markov Decision Process, the decision of the agent is based on the fully sensed state; however, in the complex marine environment, the information about the surrounding environment perceived by the USV changes with the behavioral control made by the USV, and the USV will make decisions based on the current perception, which will make the USV forget the information perceived in the previous moment and thus unable to make accurate decision control. For this reason, we used a GRU (Gated Recurrent Unit, a variant of LSTM) network to increase the memory capacity of previously perceived

FIGURE 3: The MDP framework for USV to interact with environments.

information. GRU enables the USV to effectively remember the previous states and complements the scarce effective states, thus alleviating the problem of incomplete perception. The whole process is shown in Figure 4.

Based on the GRU unit, the USV can augment the current perceptual state, while the current perceptual features also characterize the implicit knowledge including surges. As shown in Figure 5, the USV is caused to move sideways and shake under the action of the surge, and the memory capability of the GRU unit allows the USV to learn the implicit knowledge of the current surge to achieve obstacle avoidance.

3.3.3. Reward Function Design. An effective and reasonable reward function design is one of the necessary conditions for reinforcement learning to successfully solve complex tasks. In solving the navigation and obstacle avoidance task of the USV, it is necessary to fully consider various situations that the USV will encounter during the whole process and design the reward function on the basis of the above. From the analysis, we can know that the USV will encounter the following situations during the navigating: (i) the USV collision during navigating; (ii) the USV can navigate correctly and reach the target position; (iii) the USV navigates arbitrarily without collision. For the first case, when a collision occurs while the USV is navigation, we can give a larger penalty reward $r_{\text{collision}}$. For the second case, based on the distance between the USV and the target position calculated by the position sensor, the closer the target position is, the more positive the reward obtained will be, so the USV can get a larger positive reward when the USV reaches the target position without any collision. Conversely, if the USV keeps moving away from the target position, then we will give a negative reward to the USV. This reward is recorded by r_{distance} . For the last case, where the USV navigates aimlessly but does not collide, then we will give the USV a smaller penalty reward called $r_{\text{aimlessly}}$, so that the penalty will gradually increase over time.

In summary, we can summarize the reward function of the USV in the navigation obstacle avoidance task as follows:

$$r = r_{\text{collision}} + r_{\text{distance}} + r_{\text{aimlessly}}. \quad (9)$$

It is worth noting that, at the beginning of the training, collisions will frequent occurrences, and the round will end quickly. After a long training period, the USV has learned how to avoid collisions, and $r_{\text{collision}}$ will always be zero in the reward function r .

3.3.4. Design of UANO Algorithm. In the above section, we use MDP for modeling USV navigation and obstacle avoidance task. The proposed UANO algorithm contains the MDP framework, and eventually, an optimal strategy π is learned by UANO algorithm to achieve autonomous navigation of the USV. According to the MDP interaction framework, the USV will perform the ruddering operation (Action A) based on the perceived information (State S), and will obtain a reward r at each step, and at the end of each episode then we will count the total reward obtained in this episode.

The calculation of reward is shown in the following equation:

$$R(h) = r(s_t, a_t) + \gamma * r(s_{t+1}, a_{t+1}) + \dots = \sum_{t=1}^T \gamma^{k-1} r(s_t, a_t), \quad (10)$$

where $R(h)$ is the sum of discounted rewards after time t and γ is the discount factor. Here, we give three definitions of end of an episode in navigation and obstacle avoidance of the USV: (i) if the USV navigates to the target position, then the episode ends; (ii) if a collision occurs while the USV navigates, then the episode ends; (iii) if the number of actions performed by the USV exceeds the maximum, then the episode ends.

UANO algorithm's main target is to learn an optimal strategy π from the episodes. It can be expressed as follows:

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E}_{p^{\pi}(h)} [R(h)], \quad (11)$$

where $p^{\pi}(h)$ is the probability density of the episode. The control policies learned from UANO algorithm are determined by the state-action value function $Q^{\pi}(s, a) \in \mathbb{R}$, so equation (11) also can be expressed as follows:

$$Q^{\pi}(s, a) = \operatorname{argmax}_{\pi} \mathbb{E}_{p^{\pi}(h)} [R(h) | s_1 = s, a_1 = a]. \quad (12)$$

When the USV takes a specific action a_t applied on the current state s_t , the USV will get an expected immediate reward $r(s_t, a_t)$

$$r(s_t, a_t) = \mathbb{E}_{p(s_{t+1}|s_t, a_t)} [r(s_t, a_t)]. \quad (13)$$

By recursion, $Q^{\pi}(s, a)$ can be expressed as

$$Q^{\pi}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{\pi(a_{t+1}|s_{t+1})p(s_{t+1}|s_t, a_t)} [Q^{\pi}(s_{t+1}, a_{t+1})]. \quad (14)$$

Using the above equation, we can obtain the method to update the state-action value function called Q-learning, which can be expressed as follows:

FIGURE 4: The overall GRU processing flow, whose generated h contains previous state information that improves the USV's decision accuracy.

FIGURE 5: The UAV uses sensors to sense the current environmental state during navigation, aided by a GRU unit to uncover implicit knowledge of the environment, such as direction of currents, and leads the USV to make the right behavioral decisions.

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma \max Q_k(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \quad (15)$$

However, the Q -table-based approach cannot solve the large-scale problem of USVs navigation and obstacle avoidance, so deep learning can be integrated to solve the problem of insufficient expressive power of Q -tables. A deep Q -network consists of a multilayer neural network; when giving it a state s , the network will give a vector of action values $Q(s, a; \theta)$, where θ are the parameters of the network. The two breakthroughs of DQN are the target network and experience replay. The target network parameters are aligned with the online parameters whenever a number of steps have been executed and then remain fixed at other steps and wait for the next synchronization update. The experience replay stored the observed transitions and sampled uniformly from the experience pool to update the network. The two designs, target network and experience replay, greatly improve the performance of the algorithm and allow extending the algorithm to more complex problems [34].

$$Q(S_t, a_t)^{\text{DQN}} = r(s_t, a_t) + \gamma \max_a Q(S_{t+1}, a; \theta_t^-). \quad (16)$$

From equation (16), the Q -value at state S_t takes the maximized state-action Q -value of the next state at each step of the Q -value update process, which leads to a serious overestimation of the Q -value when updating, leading to instability and poor results in the training process. Briefly, updating Q -values and evaluating Q -values are both based on the same policy θ^- , leading to an overcoupling situation.

To reduce the overcoupling of Q -learning, Hasselt et al. [35] proposed double Q -learning to update the Q -value, and DeepMind [36] extends it to deep reinforcement learning. The advantage of double Q -learning over DQN is the uncoupling of part of the DQN algorithm with the idea of decomposing the operation of selecting the maximum target into the selection and evaluation of actions. The target network in the DQN network can be used as an additional network to provide a second value function. Therefore, [36] proposed evaluating the greedy strategy based on the online network and used the target network to estimate its value. Equation (17) shows the Q -value update process of double DQN.

$$Q(S_t, a_t)^{\text{DoubleDQN}} = r(s_t, a_t) + \gamma Q\left(S_{t+1}, \arg \max_a Q(S_{t+1}, a; \theta_t), \theta_t^-\right). \quad (17)$$

As equation (18) shows, the above Q -network which contains neural networks can be trained by minimizing a

sequence of loss functions that changes at each iteration; y_i is the $Q(S_t, a_t)^{\text{DoubleDQN}}$.

$$L(\theta_t) = \mathbb{E}_{s,a} [(y_i - Q(s, a; \theta_t))^2]. \quad (18)$$

The entire data flow of the UANOA algorithm is shown in Figure 6. The USV first senses the surrounding environment s_t based on sensors and performs an action a_t , and then a reward r_t and next state s_{t+1} are generated, and this information is stored in the experience pool. The deep double Q-network will sample the data in the experience pool to learn, and as training proceeds, the parameters of the deep double Q-network will eventually converge.

In the above, we have designed the inputs of the USV, which are the radar sensor and position sensor, and the outputs of the USV, which are the 13 ruddering actions, and we have designed the reward function for the navigation and obstacle avoidance tasks of the USV, and also, we have completed the derivation of the UANOA algorithm. Therefore, based on the above, we can get the complete UANOA algorithm (Algorithm 1).

4. Experiment

In this section, we elaborate the UANOA algorithm designed above in a virtual experimental environment, and validate and analyze the effectiveness of our algorithm and compare it with other deep reinforcement learning algorithms (DQN, random policy).

4.1. Training Environment Construction and Algorithm Parameter Setting. We firstly construct a trainable USV obstacle avoidance and navigation virtual environment. The virtual environment is being implemented using the Unity Machine Learning Agent Toolkit [37]. We used Unity3D to implement the kinematic modeling of USV and the construction of complex scenes, and the Unity Machine Learning Agent Toolkit combined with a deep learning framework [38] to implement and train the UANOA algorithms.

Figure 7 shows the environment-aware scenario of the USV we built based on the virtual engine. Compared with the traditional 2D simulation-based conditions [39, 40], the USV navigation and obstacle avoidance tasks constructed based on 3D scenarios are more realistic, and the algorithms are more versatile and easier to deploy. The experimental environment is shown in Figure 7.

Previously, we have completed the design of the UANOA algorithm, including the input and output of the USV, the design of the reward function. We described that the USV senses the state of its surroundings through laser ranging sensors and position sensors. The laser ranging sensor as input to the algorithm is a 13-dimensional vector, each dimension representing the distance of a ray detected to the current USV from an obstacle, and filled with 0 if no obstacle is detected. Here, we will give the details of the network structure and parameters used in the UANOA algorithm.

The UANOA algorithm is based on a deep reinforcement learning algorithm, and the network structure can be divided into two parts. The first part is the observation

embedding part, which is divided into four main layers. The first layer consists of 256 neurons, which mainly receives the data sensed by the laser ranging sensor. The second layer is composed of 256 neurons. The third layer has 128 neurons and concatenates the position sensor data, which is a three-dimensional vector. The activation function of each layer is set to Relu, and the location sensor data is normalized. The fourth layer is the GRU layer, which is a variant of LSTM with a simpler network structure, and we also set 128 neurons in this layer with the activation function Tanh. The second part of the UANOA algorithm parameters is the RL network part, which is mainly responsible for receiving the output data of the observation embedding and outputting the USV actions. This part is mainly divided into three layers; the first layer is composed of 128 neurons with Relu activation function, the second layer is composed of 64 neurons with Relu activation function, and the last layer is the output layer and performs softmax processing on the output results, which mainly outputs the actions of the USV. Table 1 shows the model structure and parameters in the UANOA algorithm.

In Section 3.3.3, we have completed the design of the reward function, and here we give the specific values of the reward function. When a collision occurs in the USV, then a -2 reward is given, and when the USV does not collide and travels towards the target position, then a $+0.1$ reward is given; if the USV is far from the target point, give a -0.1 reward. Also, to encourage the USV to keep making movements, we give a -0.01 reward for each moment. Also, in the experiment, the throttle size of the USV was fixed, allowing the USV to travel at a speed of 5 meters per second.

4.2. Experimental Results of UANOA Algorithm. In the above, we have designed the experimental scenario, in which the lower left position is the starting position of the USV, and the upper right position is our target point; the whole ocean environment is arranged with many obstacles; when the USV can avoid obstacle and navigation to the target position within a limited number of steps, then it is counted to have completed the task.

Figure 8 shows the reward and average reward training process of the UANOA algorithm in ten thousand training episodes. Figure 8(a) shows the UANOA algorithm, DQN, and random policy reward trend in every episode. From the figure, we can see that the UANOA algorithm converges more easily than other algorithms, and DQN algorithm converges slowly only at the end of training, while random policy is not able to converge at all. Figure 8(b) shows the average reward trend of UANOA algorithm, DQN, and random policy in every episode. From the figure, we can see that the average reward obtained of the UANOA algorithm gradually converges to 0.1, which means that the USV keeps traveling towards the target position, DQN algorithm also tends to converge, but is less stable compared to the UANOA algorithm, while random policy cannot converge in average reward.

Figure 9 shows the USV navigation distance and rudder angle switching frequency training process of the UANOA

FIGURE 6: The main core parts and data streams of the UANO algorithm.

```

Initialize replay memory  $D$ 
Initialize evaluate  $Q$  function of the USV with random weights  $\theta$ 
Initialize target  $Q$  function of the USV with random weights  $\theta^- = \theta$ 
for  $episode = 1, 2, \dots, M$  do
  for  $t = 1, \dots, T$  do
    With probability  $\epsilon$  select a random USV rudder action
    otherwise select  $a_t = \operatorname{argmax}_a Q(s_t, a; \theta)$ 
    Get reward  $r_t$  and next state  $s_{t+1}$  by executing rudder action  $a_t$ 
    Store experience  $(s_t, a_t, r_t, s_{t+1})$  in  $D$  where  $s$  is processed by the LSTM network
    Sample random minibatch of experience  $(s_t, a_t, r_t, s_{t+1})$  from  $D$ 
    Set  $y_j = \begin{cases} r(s_t, a_t) & \text{if episode terminates at step } t+1 \\ r(s_t, a_t) + \gamma Q(s_{t+1}, \operatorname{argmax}_a Q(s_{t+1}, a; \theta^-)) & \text{otherwise} \end{cases}$ 
    Perform a gradient descent step on  $(y_t - Q(s_t, a_t; \theta))^2$  with respect to the
    weights  $\theta$ 
    Every  $C$  steps reset  $\theta^- = \theta$ 
  end
end

```

ALGORITHM 1: UANO algorithm for navigation and obstacle avoidance of the USV.

FIGURE 7: Experimental environment in our work.

TABLE 1: The overall network structure of the UANO algorithm, which consists of a perception part and a decision part.

Network structure	Network type	Number of neurons	Activation function
First layer	Linear	256	Relu
Second layer	Linear	256	Relu
Third layer	Linear (concat)	131	Relu
Fifth layer	GRU	128	Tanh
Sixth layer	Linear	128	Relu
Seventh layer	Linear	128	Relu
Eighth layer	Linear	13	Softmax

FIGURE 8: Comparison of the UANO algorithm, DQN, and random policy in reward and average reward during the USV navigation and obstacle avoidance training.

FIGURE 9: Comparison of the UANO algorithm, DQN, and random policy in max USV navigation distance and rudder angle switching frequency during the USV navigation and obstacle avoidance training.

algorithm in ten thousand training episodes. Figure 9(a) shows the USV navigation distance in every episode. From the figure, we can see that as the UANO algorithm trained,

the USV is able to navigate forward gradually, eventually reaching 140 m at ten thousand training episodes, which is also an approximate length of the USV to the target location,

FIGURE 10: Performance of the USV navigation and obstacle avoidance behavior of UANO algorithm under the different number of the episode of training.

and DQN can make USV travels gradually towards the target position, and slightly lower than the UANO algorithm in the same episode, while random policy is only able to travel about 20 m before a collision occurs or the maximum number of steps is reached. Figure 9(b) shows the rudder angle switching frequency in every episode. From the figure, we can see that with the training process the UANO algorithm is able to make the USV's rudder angle switch less frequently, which means the USV has been able to learn how to avoid obstacles in complex ocean scenarios and USV becomes more stable during navigation; DQN algorithm also has the ability to reduce the rudder angle switching frequency, but the performance is not as good as our algorithm. Random policy performs poorly and fails to meet the task requirements.

Figure 10 shows the performance of the USV navigation and obstacle avoidance behavior of UANO algorithm under the different number of episodes of training. In the figure, (a) to (d) is the initial stage of training; we can see that the USV is not yet able to navigate towards the target position and eventually sailing into a narrow spot and touches an obstacle. And, (e) to (h) is the middle stage of training; in this stage, we can see that the USV has mastered certain navigation and obstacle avoidance skills and is able to travel farther and closer to the target location. The final convergence of the UANO algorithm is shown in (i) to (l). At this time, the USV has been able to complete navigation and obstacle avoidance operations in the whole complex

environment and has been able to plan the optimal navigation route.

4.3. Generalization Capability of UANO Algorithm. In the above experiments, our UANO algorithm has fully converged and can achieve navigation and obstacle avoidance tasks in complex marine environments. However, the current reinforcement learning-based UANO algorithm only achieves the USV navigation and obstacle avoidance in a single training scenario, and the USV may encounter more complex environments in real situations, which requires that our algorithm has a strong generalization capability, namely, whether the navigation and obstacle avoidance tasks can be continued in different scenarios.

In this section, we design three scenarios (shown in Figure 11) to verify the generalization ability of UANO algorithm, and the three scenarios correspond to three different difficulties, including the primary scenario, the medium scenario, and the advanced scenario.

For the evaluation of the generalization capability of the UANO algorithm, we mainly test the UANO algorithm for 1000 episodes in the above three different scenarios and then analyze it from the following two metrics: (i) the navigation distance of USV; (ii) the success rate of obstacle avoidance of USV. Simultaneously, we take DQN algorithm and random policy as the comparison reference. For the first metric, we mainly counted the maximum navigation

FIGURE 11: Three different scenario ocean environments for testing the generalization capability of the UANO algorithm. (a) Primary scenario. (b) Medium scenario. (c) Advanced scenario.

FIGURE 12: The navigation statistical results of the UANO algorithm and comparison algorithms in three different ocean scenarios.

TABLE 2: The obstacle avoidance success rate of the UANO algorithm and comparison algorithms in three different ocean scenarios.

Environment type	Primary scenario	Medium scenario	Advanced scenario
UANO success rate	95%	86%	67%
DQN success rate	91%	79%	49%
Random policy success rate	0	0	0

distance of the USV in three scenarios and the closest distance to the target point.

As shown in Figure 12(a), the UANO algorithm can navigate 140 m, 120 m, and 90 m in the primary to advanced scenarios, respectively, while the DQN algorithm can navigate 120 m, 75 m, and 55 m in the primary to advanced scenarios. The random strategy algorithm can navigate less than 80 m, 60 m, and 40 m in the primary to advanced scenarios. As shown in Figure 12(b), the UANO algorithm is able to navigate to 20 m, 38 m, and 45 m from the target

position in the primary to advanced scenarios, and the DQN algorithm is able to navigate to 35 m, 55 m, and 62 m from the target location in the primary to advanced scenarios. The random policy, on the other hand, navigates to 55 m, 65 m, and 120 m from the target location, respectively. For the second metric, we count the collision avoidance success rate of the USV in 1000 episodes for testing. As shown in Table 2, the UANO algorithm can achieve a collision avoidance success rate of 95% in the primary scenario, 86% in the medium scenario, and 67% in the advanced scenario. The

DQN algorithm can achieve a collision avoidance success rate of 91%, 79%, and 49% in these three scenarios. However, the random policy success rate is 0 in these three scenarios.

5. Discussions

The autonomous navigation and obstacle avoidance capability of USVs is of great scientific importance and practical value. A qualified autonomous navigation and obstacle avoidance algorithm is able to fully perceive the surrounding information and make decisions in real time with the different sensors. In this paper, we propose the UANO algorithm for autonomous navigation and obstacle avoidance of the USV, which uses laser ranging sensors and position sensors to sense the surrounding environment information to make the corresponding rudder angle operation in real time. Our UANO algorithm can perform real-time navigation and obstacle avoidance tasks in complex sea conditions after 10,000 rounds of training, and compared with algorithms such as DQN, our algorithm has a stronger generalization capability, which can meet the USV's ability to protect itself in the face of unexpected situations.

6. Conclusions and Future Work

In this paper, a reinforcement learning-based UANO method has been proposed which integrated with the real control model of USVs for the autonomous navigation and obstacle avoidance. Compared with previous studies, we learned the USV's rudder control in a complex ocean environment by using a laser ranging sensor and a position sensor. We have conducted numerous experiments to demonstrate the convergence and effectiveness of the algorithm, and our algorithmic architecture can be easily extended to other complex tasks for USVs, such as autonomous berthing and target tracking.

However, the USV autonomous navigation and obstacle avoidance task is still a very complex and difficult issue. In our work, we use the laser ranging and position sensor to perceive the surrounding environment; this means of perception is incomplete and susceptible to interference. If we can integrate other kinds of sensor information (i.e., images), we can use more valid information to derive more robotic control strategies. Besides, our algorithm only outputs the rudder angle of the USV, while the throttle operation of the USV is also very important, because the throttle determines the sailing speed of the USV, which allows the USV to decide whether to accelerate or decelerate according to the environment, thus improving the mission execution success rate. It should be noted that all of our experiments are based on the virtual environment; the distance gap between virtual and real makes the algorithm cannot be used directly; we will subsequently build a real USV navigation dataset and test our algorithm in the dataset, through the test results to continue to adjust the algorithm, so as to finally realize the application on a real USV. We take these issues as our future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research reported in this paper was supported by the National Natural Science Foundation of China Youth Fund under Grant no. 61902001. This work was jointly supported by the Anhui Province University Outstanding Youth Talent Support Program under Grant no. gxyq2020031 and the Natural Science Project of Anhui Polytechnic University under Grant no. Xjky072019A01.

References

- [1] Y. Peng, Y. Yang, J. Cui et al., "Development of the USV "JingHai-I" and sea trials in the southern yellow sea," *Ocean Engineering*, vol. 131, pp. 186–196, 2017.
- [2] Y. Xie, X. Liang, L. Lou, and X. Guo, "Self-organization method of usv swarm target strike task based on ant colony algorithm," in *Proceedings of the 2019 3rd International Symposium on Autonomous Systems (ISAS)*, pp. 388–393, IEEE, Shanghai, China, May 2019.
- [3] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6105–6110, IEEE, Vancouver, Canada, September 2017.
- [4] G. Shao, Y. Ma, R. Malekian, X. Yan, and Z. Li, "A novel cooperative platform design for coupled USV-UAV systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4913–4922, 2019.
- [5] Y. Zhang, D. Qu, J. Ke, and X. Li, "Dynamic obstacle avoidance for usv based on velocity obstacle and dynamic window method," *Journal of Shanghai University (Natural Science Edition)*, vol. 23, no. 1, pp. 1–16, 2017.
- [6] S. Xie, P. Wu, Y. Peng et al., "The obstacle avoidance planning of usv based on improved artificial potential field," in *Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA)*, pp. 746–751, IEEE, Hailar, China, July 2014.
- [7] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 6016–6025, IEEE Computer Society, Salt Lake City, UT, USA, June 2018.
- [8] A. Hämmäläinen, K. Arndt, A. Ghadirzadeh, and V. Kyrki, "Affordance learning for end-to-end visuomotor robot control," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019*, pp. 1781–1788, IEEE, Macau, SAR, China, November 2019.
- [9] T. Li, Y. Li, and B. Li, "Reinforcement learning based novel adaptive learning framework for smart grid prediction," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8192368, 8 pages, 2017.

- [10] H.-J. Song, H.-K. Kim, J.-D. Kim, C.-Y. Park, and Y.-S. Kim, "Inter-sentence segmentation of youtube subtitles using long-short term memory (lstm)," *Applied Sciences*, vol. 9, no. 7, p. 1504, 2019.
- [11] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder-decoder architecture for detection of image forgeries," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286–3300, 2019.
- [12] X. Liu, Y. Li, J. Zhang, J. Zheng, and C. Yang, "Self-adaptive dynamic obstacle avoidance and path planning for usv under complex maritime environment," *IEEE Access*, vol. 7, pp. 114945–114954, 2019.
- [13] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos, "An energy-efficient path planning algorithm for unmanned surface vehicles," *Ocean Engineering*, vol. 161, pp. 308–321, 2018.
- [14] Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "A constrained a* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents," *Ocean Engineering*, vol. 169, pp. 187–201, 2018.
- [15] T. Mina, Y. Singh, and B.-C. Min, "Maneuvering ability-based weighted potential field framework for multi-usv navigation, guidance, and control," *Marine Technology Society Journal*, vol. 54, no. 4, pp. 40–58, 2020.
- [16] Y. Singh, M. Bibuli, E. Zereik, S. Sharma, A. Khan, and R. Sutton, "A novel double layered hybrid multi-robot framework for guidance and navigation of unmanned surface vehicles in a practical maritime environment," *Journal of Marine Science and Engineering*, vol. 8, no. 9, p. 624, 2020.
- [17] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [18] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor uav using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2009.
- [19] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," 2017, <https://arxiv.org/abs/1709.06560>.
- [20] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," 2013, <https://arxiv.org/abs/1312.5602>.
- [21] A. R. Fayjie, S. Hossain, D. Oualid, and D.-J. Lee, "Driverless car: autonomous driving using deep reinforcement learning in urban environment," in *Proceedings of the 2018 15th International Conference on Ubiquitous Robots (UR)*, pp. 896–901, IEEE, Jeju, South Korea, July 2018.
- [22] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S.-i. Maeda, "Dqn-tamer: human-in-the-loop reinforcement learning with intractable feedback," 2018, <https://arxiv.org/abs/1810.11748>.
- [23] A. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.
- [24] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Brisbane, Australia, May 2018.
- [25] J. Liu, W. Qi, and X. Lu, "Multi-step reinforcement learning algorithm of mobile robot path planning based on virtual potential field," *Communications in Computer and Information Science*, vol. 60, pp. 528–538, 2017.
- [26] W. Wang, X. Luo, Y. Li, and S. Xie, "Unmanned surface vessel obstacle avoidance with prior knowledge-based reward shaping," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 9, p. e6110, 2021.
- [27] P. Long, W. Liu, and J. Pan, "Deep-learned collision avoidance policy for distributed multiagent navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 656–663, 2017.
- [28] J. Woo, C. Yu, and N. Kim, "Deep reinforcement learning-based controller for path following of an unmanned surface vehicle," *Ocean Engineering*, vol. 183, pp. 155–166, 2019.
- [29] X. Zhou, P. Wu, H. Zhang, W. Guo, and Y. Liu, "Learn to navigate: cooperative path planning for unmanned surface vehicles using deep reinforcement learning," *IEEE Access*, vol. 7, pp. 165262–165278, 2019.
- [30] Y. Liu, Y. Peng, M. Wang, J. Xie, and R. Zhou, "Multi-usv system cooperative underwater target search based on reinforcement learning and probability map," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7842768, 12 pages, 2020.
- [31] Y. Cheng and W. Zhang, "Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels," *Neurocomputing*, vol. 272, pp. 63–73, 2018.
- [32] X. Zhang, C. Wang, Y. Liu, and X. Chen, "Decision-making for the autonomous navigation of maritime autonomous surface ships based on scene division and deep reinforcement learning," *Sensors*, vol. 19, no. 18, p. 4055, 2019.
- [33] H. Xu, N. Wang, H. Zhao, and Z. Zheng, "Deep reinforcement learning-based path planning of underactuated surface vessels," *Cyber-Physical Systems*, vol. 5, no. 1, pp. 1–17, 2019.
- [34] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [35] H. Hasselt, "Double Q-learning," *Advances in Neural Information Processing Systems*, vol. 23, pp. 2613–2621, 2010.
- [36] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," 2015, <https://arxiv.org/abs/1509.06461>.
- [37] A. Juliani, V.-P. Berges, E. Vckay et al., "Unity: a general platform for intelligent agents," 2018, <https://arxiv.org/abs/1809.02627>.
- [38] M. Abadi, P. Barham, J. Chen et al., "Tensorflow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, K. Keeton and T. Roscoe, Eds., USENIX Association, Savannah, GA, USA, November 2016.
- [39] H. Mousazadeh, H. Jafarbiglu, H. Abdolmaleki et al., "Developing a navigation, guidance and obstacle avoidance algorithm for an unmanned surface vehicle (usv) by algorithms fusion," *Ocean Engineering*, vol. 159, pp. 56–65, 2018.
- [40] T. Liu, Z. Dong, H. Du, L. Song, and Y. Mao, "Path following control of the underactuated usv based on the improved line-of-sight guidance algorithm," *Polish Maritime Research*, vol. 24, no. 1, pp. 3–11, 2017.