Research on Method of Collision Avoidance Planning for UUV Based on Deep Reinforcement Learning

Wei Gao 1, Mengxue Han 2, Zhao Wang 1, Lihui Deng 3, Hongjian Wang 1,* and Jingfei Ren 1

1 College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; gg19963996@foxmail.com (W.G.); 814551366@hrbeu.edu.cn (Z.W.); renjingfei@hrbeu.edu.cn (J.R.)
2 AVIC China Aero-Polytechnic Establishment, Beijing 100028, China; 13699154492@163.com
3 Tianjin Navigation and Instrument Institute, Tianjin 300130, China; gongchengrendlh@163.com
* Correspondence: cctime99@163.com

Abstract: A UUV can perform tasks such as underwater surveillance, reconnaissance, surveillance, and tracking by being equipped with sensors and different task modules. Due to the complex underwater environment, the UUV must have good collision avoidance planning algorithms to avoid various underwater obstacles when performing tasks. The existing path planning algorithms take a long time to plan and have poor adaptability to the environment. Some collision-avoidance planning algorithms do not take into account the kinematic limitations of the UUV, thus placing high demands on the performance and control algorithms of UUV. This article proposes a PPO-DWA collision avoidance planning algorithm for the UUV under static unknown obstacles, which is based on the proximal policy optimization (PPO) algorithm and the dynamic window approach (DWA). This algorithm acquires the obstacle information from forward-looking sonar as input and outputs the corresponding continuous actions. The PPO-DWA collision avoidance planning algorithm consists of the PPO algorithm and the modified DWA. The PPO collision avoidance planning algorithm is only responsible for outputting the continuous angular velocity, aiming to reduce the difficulty of training neural networks. The modified DWA acquires obstacle information and the optimal angular velocity from the PPO algorithm as input, and outputs of the linear velocity. The collision avoidance actions output by this algorithm meet the kinematic constraints of UUV, and the algorithm execution time is relatively short. The experimental data demonstrates that the PPO-DWA algorithm can effectively plan smooth collision-free paths in complex obstacle environments, and the execution time of the algorithm is acceptable.

Keywords: PPO; DWA; collision avoidance; path planning

1. Introduction

A UUV is composed of a body structure, power system, navigation system, perception system, control system, and communication system, which plays an important role in underwater search and rescue, terrain exploration, and other fields. Whether the UUV can quickly and safely reach the target point determines the effectiveness of task completion. Collision avoidance planning is a significant research topic in the field of UUV navigation. It enables robots to plan a collision-free trajectory from an initial pose to a target pose, even in uncertain environments [1]. The technologies in the field of global path planning based on known maps have developed tremendously over the past few decades. However, in the face of complex and unknown environments, the use of traditional collision avoidance planning algorithms is limited. The artificial potential field (APF) method does not take into account the kinematic limitations of a UUV, lacks adaptability to the environment, and may fail in complex obstacle environments. The rapidly exploring random tree (RRT) algorithm takes too long to calculate and requires high computational
power [2]. Collision avoidance planning algorithms based on intelligent search algorithms are prone to falling into local optima and have a longer planning time.

In recent years, with the progress of science and technology, deep reinforcement learning algorithms have made major breakthroughs in various fields. The deep reinforcement learning algorithms have good decision-making ability, which can explore and learn independently. It is feasible to use deep reinforcement learning algorithms to solve the problem of collision avoidance planning [3]. On the one hand, the self-learning mechanism of reinforcement learning (RL) can improve adaptability to the environment. Conversely, the end-to-end DRL strategy that accomplishes the collision avoidance planning task can take less time to make decisions. Deep reinforcement learning can address high-dimensional problems when the states and actions are continuous variables in the environment [4].

In order to improve the adaptability of collision avoidance algorithms to the environment, enhance the algorithms’ real-time planning ability, and meet the kinematic limitations of the UUV, this paper proposes a PPO-DWA collision avoidance planning algorithm for the UUV. The structure of this paper is as follows. Section 2 reviews the main collision avoidance planning algorithms. Section 3 completes the kinematic modeling of the UUV and introduces the collision avoidance algorithm DWA and deep reinforcement learning algorithm PPO. This is followed by a description of the problem definition and the PPO-DWA algorithm. Section 5 verifies the feasibility of the PPO-DWA method through experiments. Section 6 summarizes the entire article and discusses future research directions of this paper.

2. Related Work

The research of collision avoidance planning aims to design an optimal collision-free trajectory based on surrounding obstacle information. In this section, collision avoidance planning algorithms are divided into two categories: traditional algorithms and intelligence algorithms. The traditional collision avoidance planning algorithms mainly include a rapidly exploring random tree (RRT) [5], a dynamic window approach [6], an artificial potential field method [7], a bug algorithm [8], and intelligent collision avoidance algorithms, mainly include ant colony optimization algorithms [9], genetic algorithms (GA), and reinforcement learning algorithms [10]. The advantages and disadvantages of commonly used path planning algorithms are shown in Table 1.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dijkstra’s algorithm</td>
<td>Find the shortest path from the starting point to the ending point.</td>
<td>When the search dimension is high, the computational workload is high.</td>
</tr>
<tr>
<td>Rapidly exploring random tree</td>
<td>This algorithm has fast expansion and search speed, and is suitable for path planning in various complex environments.</td>
<td>This algorithm suffers from low node utilization and high path complexity due to its random sampling mechanism.</td>
</tr>
<tr>
<td>Probabilistic roadmap</td>
<td>This algorithm can construct effective path maps in high-dimensional space.</td>
<td>The algorithm cannot find a feasible path when there are too few sampling points.</td>
</tr>
<tr>
<td>Dynamic window Approach</td>
<td>This algorithm considers constraints such as environment and acceleration.</td>
<td>This algorithm lacks foresight and is prone to local extrema.</td>
</tr>
<tr>
<td>Artificial potential field</td>
<td>Simple structure and low computational complexity.</td>
<td>This algorithm is prone to falling into local minima and is difficult to handle complex environments.</td>
</tr>
<tr>
<td>Genetic algorithm</td>
<td>High global search ability and robustness.</td>
<td>This algorithm is prone to falling into local optima and difficult to achieve real-time planning.</td>
</tr>
<tr>
<td>Ant colony optimization</td>
<td>This algorithm has strong global search ability and can be parallelized and distributed for computation.</td>
<td>The algorithm has imbalanced optimization capabilities and is prone to falling into local optima and other shortcomings.</td>
</tr>
<tr>
<td>Neural network</td>
<td>This algorithm can achieve real-time planning.</td>
<td>The effectiveness of this algorithm relies on expert experience and lacks adaptability to the environment.</td>
</tr>
</tbody>
</table>
Reinforcement learning can achieve real-time planning and has strong adaptability to the environment. Both states and actions are discrete and finite, making it cumbersome to handle complex situations.

2.1. Traditional Algorithms

Traditional obstacle avoidance algorithms are limited in uncertain environments. They are characterized by having long planning times, easily falling into local optima, and unstable obstacle avoidance effects. The RRT algorithm is a randomized algorithm that can be directly applied to the planning of nonholonomic constraint systems. This method is suitable for high-dimensional systems because of its low complexity. However, the path planned by the RRT algorithm is not optimal, and the convergence speed of the RRT is slow. Informed RRT* was proposed to improve the convergence rate and final solution quality [11]. Wei Zhang et al. proposed a modified RRT collision avoidance planning algorithm, which could overcome the problems of non-optimal paths and low planning success rates in complex environments [12]. Yutong Lin et al. proposed and applied the improved RRT algorithm to solve the path planning problem of unmanned ships. Experimental results showed that the path length of the improved RRT algorithm was shorter and smoother than that of the traditional RRT algorithm. However, the obstacle environment in the experiment was too simple, and the planned path would have been too close to the obstacle [13]. One of the most commonly used methods for dynamic path planning is the DWA, which can meet the kinematic constraints of the mobile robot. However, the collision avoidance effect of DWA heavily depends on the setting of parameters. Matej Dobrevski et al. proposed a novel deep convolutional neural network, which can dynamically predict DWA parameters based on environmental changes [14]. APF can be implemented in real-time, but the biggest disadvantage of APF is that the attractive field and the repulsive potential field can easily cancel each other, resulting in the robot falling into local optimization. Riky Dwi Puriyanto et al. proposed a modified APF algorithm using the Gompertz function and the cone-shaped potential field [15]. Another drawback of the APF method is that the planned path is not the shortest. Guanghi Li et al. redefined potential functions and proposed a simultaneous forward search method to shorten the distance of the planned path. The simulation results confirmed that the modified APF can calculate a shorter and safer path to the target point [16]. Subir Kumar proposed the ModifiedCriticalPointBug(MCPB) algorithm based on the Bug algorithm, which could avoid run-time obstacles [17]. Yuanyuan Zhang et al. proposed an improved APF (FV-APF) algorithm to address the local minimal and goal unattainability problems of the classic APF method in the corridor environment by considering the angle constraint and improving the potential field function while combining simple fuzzy ideas. However, the path planned by this method has large corners, which is not conducive to controlling actual robots [18].

2.2. Artificial Intelligence Algorithms

Intelligent algorithms can plan safe paths in complex obstacle environments with minimal planning time. As an intelligent bionics algorithm, the genetic algorithm shows good robustness in solving complex nonlinear optimization problems, such as the path planning problem of mobile robots. Therefore, it has been widely used in mobile robot collision avoidance planning [19]. Won-Seok Kang et al. proposed a stable collision avoidance planning algorithm for avoiding dynamic obstacles based on a genetic algorithm, which focuses on finding optimal movements for stability rather than finding the shortest paths or smallest movements [20]. A grid-based GA path planning algorithm is developed to seek a safe path requiring the least effort. The modified cost function can help the AUV to perform a long-range mission with maximum endurance [21]. The ant colony optimization algorithm (ACO) has the advantage of parallel computing's strong robustness, so it is used to solve the problem of avoiding obstacles. However, traditional ACO has shortcomings, such as slow convergence speed and easily falling into the local optimal value.
To overcome the above problems, a novel variant of ACO is proposed, which introduces a new heuristic mechanism with orientation information, a modified heuristic function, and a modified state transition probability rule [22]. Rajat Agrawal et al. have proposed the multi-object adaptive ant colony optimization for path planning in a static environment [23]. The objective function is formulated as a multiple objective problem by incorporating path length, safety factor, and energy consumption. To help mitigate the slow convergence of the original algorithm, this avoidance obstacle algorithm introduces the A* algorithm to modify the heuristic information function in the conventional ant colony optimization. Chaowei Liu et al. propose a path planning method for the UUV based on the QPSO algorithm. This algorithm is simple to model and easy to implement, and it also has fewer control parameters and rapid convergence. However, this method did not take into account the kinematic limitations of the UUV, and the planned path had too many corners [24].

The reinforcement learning algorithms can avoid the robot’s dependence on environmental maps and prior knowledge and show good self-learning capability and adaptability in practical applications. The avoidance obstacle algorithm based on OAE-Q(λ)-learning is proposed [25]. The method expands the concise obstacle area, which can avoid repeated invalid actions and accelerate the convergence speed of the algorithm. In order to meet the kinematic constraints of the mobile robots, the path planning method based on deep reinforcement learning is proposed [26]. The method can find the optimal strategy in the continuous action space. In order to achieve safe obstacle avoidance in agricultural scenarios, the Residual-like Soft actor–critic algorithm is proposed [27]. The method proposes an online expert experience pre-training model, which can alleviate the time-consuming problem of the exploration process of RL. Aiming to solve the Q-Learning algorithm that suffers from slow convergence speed and needs large storage space, K. Cai et al. proposed the distributed path planning algorithm [28]. The distributed path planning algorithm uses multiple mobile robots to explore local space and share information with other mobile robots. Changjian Lin et al. proposed an improved recurrent neural network for unmanned underwater vehicles’ online obstacle avoidance [29]. This algorithm obtains shorter paths, uses less energy through its actuators, and is resistant to noise. However, the collision avoidance strategy learned by this algorithm is based on expert data and lacks adaptability to the environment. Jian Xu proposed a novel event-triggered soft actor–critic algorithm for AUV collision avoidance [30]. This method can enable a UUV to avoid both unknown static obstacles and unknown dynamic obstacles under the condition of a limited detection range. In this method, the SAC algorithm is responsible for outputting three consecutive actions, increasing the network’s training difficulty. Had Behnaz proposed a DRL-based motion planning approach for an AUV, focusing on producing a short, safe, and feasible path [31]. However, this method does not change the longitudinal velocity of the UUV, and the obstacle environments are too simple. Prashant Bhopale proposed a modified Q-learning algorithm obstacle avoidance algorithm for a UUV [32]. This method reduces the chances of collision with obstacles, but the action space of this method is discrete, which may result in the resulting path not being optimal.

3. Materials

This section mainly introduces the kinematic modeling of the UUV and the basic principles of DWA and PPO to facilitate the proposal of a PPO–DWA collision avoidance planning algorithm.

3.1. UUV Model

This article takes the UUV as the research object, with a total length of 4.5 m, a total width of 1.1 m, and a height of 0.6 m. According to research needs, only the two-
dimensional motion in the horizontal direction of the UUV is considered. Therefore, the following assumptions are made:

1. Neglecting the influence of third-order and higher-order hydrodynamic coefficients on UUV;
2. Neglecting the influence of roll, pitch, and heave movements of UUV on horizontal motion.

The UUV coordinate system is shown in Figure 1. The movement of UUV is mainly carried out in three degrees of freedom: longitudinal, transverse, and bow. This article defines the vector \( \eta = [x, y, \varphi] \) to represent the generalized position information of UUV in a fixed coordinate system. The vector \( \nu = [u, v, \psi] \) represents the generalized velocity of UUV, where \( u \) is the longitudinal velocity of UUV, \( v \) is the lateral velocity of UUV, and \( \psi \) is the heading angular velocity. The schematic diagram of the horizontal movement of UUV is shown in the following figure, where \( \sqrt{v^2 + v^2} \) is the UUV synthesis speed and \( \beta = \arctan(v/u) \) is the drift angle.

![Figure 1. Schematic diagram of horizontal movement of UUV.](image_url)

In the absence of interference, the kinematic model of UUV in the horizontal plane is as follows:

\[
\begin{align*}
\dot{x} &= u \cos(\psi) - v \sin(\psi) \\
\dot{y} &= u \sin(\psi) + v \cos(\psi) \\
\psi &= \dot{\varphi}
\end{align*}
\]

In the actual process of underwater navigation, ocean currents cannot be eliminated and are the biggest interference factor affecting UUV. Therefore, in actual simulation research, it is necessary to consider the impact of ocean currents on UUV. However, in reality, ocean currents vary with time and space, but in the process of simulation research, this article simplifies them. Assuming that the ocean current is a constant current, the velocity in a fixed coordinate system is \( U_c \) and the direction angle is \( \psi_c \). The mathematical model for a constant horizontal current is as follows:

\[
\begin{align*}
u_c &= U_c \cos(\psi_c - \psi) \\
v_c &= U_c \sin(\psi_c - \psi)
\end{align*}
\]
Among them, \( u_c \) and \( v_c \) are the longitudinal and transverse components in the motion coordinate system, respectively.

If \( u \) and \( v \) are defined as the longitudinal and transverse velocities of UUV in a fixed coordinate system, \( u_v \) and \( v_v \) are the longitudinal and transverse components of UUV relative to the velocity measured by the surrounding ocean current. Thus, the kinematic equation of UUV under current interference is obtained as follows:

\[
\begin{align*}
\dot{x} &= (u_c + u_v) \cos(\psi) - (v_c + v_v) \sin(\psi) \\
\dot{y} &= (u_c + u_v) \sin(\psi) + (v_c + v_v) \cos(\psi) \\
\psi &= r
\end{align*}
\] (3)

3.2. DWA

DWA is a local collision avoidance planning algorithm studied by FOX and Thrun in 1997 [33]. This method samples the linear velocity and angular velocity in the constrained sampling space, generates a virtual path according to the sampling values of the two speeds, and then uses certain evaluation criteria to evaluate the generated virtual path. The biggest advantage of this method is that it can ensure the feasibility of the robot’s motion speed command. This method can also meet various constraints on the robot’s speed and can provide speed control commands based on the surrounding obstacles to achieve obstacle avoidance effect.

The DWA collision avoidance planning algorithm is mainly divided into two parts: (1) Generating a feasible velocity value space for the next moment based on the motor limits and safety constraints—the velocity window. The velocities within this range are the velocities that the mobile robot can reach, and these velocities form a dynamic window centered on the current velocity in the velocity space. (2) The speed in the dynamic window is sampled to obtain a series of discrete speed values. According to the sampled speed values, the future movement trajectory of the mobile robot is predicted, and then the predicted trajectory is scored to select the optimal speed combination at the next time.

3.3. PPO

Collision avoidance planning can be considered as a process of interaction between the robot and the obstacle environment. At every moment, the mobile robot needs to make decisions according to the observation information of the environment, and select the best action at the next moment until the mobile robot reaches the expected target point. Reinforcement learning is a common intelligent decision-making algorithm, so it is reasonable to use RL to solve collision avoidance planning problems.

In the framework of reinforcement learning, the agent will constantly make “mistakes” in the learning process, and it will absorb these experiences and constantly adjust its strategy to be closer to human thinking habits [34]. When a reinforcement learning algorithm is used to solve the task of collision avoidance planning, it is necessary to build the corresponding simulation environment for the task. The agent is constantly interacting with the environment, and sometimes the agent will adopt a very poor strategy, resulting in a collision between the mobile robot and obstacles. However, it can adjust its behavior through the reward information provided by the environment to obtain the maximum average cumulative reward [35]. The framework of RL is shown in Figure 2. The agent first generates the next action based on the current environment state, and the action will interact with the environment. The agent’s own state will change under the influence of the action, and the environment will also give feedback on the quality of the action.
Reinforcement learning has good decision-making ability, but this algorithm can only solve the situation where the state and action space are discrete, while the obstacle information returned by the sensor is the continuous value, and in order to exert accurate control over the robot, the action space is sometimes required to be a continuous value, so the use of traditional reinforcement learning algorithms in these scenarios is limited. Deep reinforcement learning combined with deep learning breaks through the limitations of the number of states and the number of actions, which has a higher intelligence level and the ability to solve complex problems. As a reinforcement learning algorithm of actor–critic structure, the PPO algorithm has the advantage of good robustness and simple parameter adjustment [36].

The PPO algorithm process is shown in Algorithm 1.

Algorithm 1 Proximal policy optimization

1: Input: initial policy parameters \( \theta \), initial value function parameters \( \phi \).
2: for \( i \in \{1,\ldots,N\} \) do
3: Run policy \( \pi_a \) for \( T \) timesteps, collecting \( \{s, a, r\} \)
4: Estimate advantages \( A_i = \sum_r y^i r - V(s) \)
5: \( \pi_{aw} \leftarrow \pi_a \)
6: for \( j \in \{1,\ldots,M\} \) do
7: \( J_{\pi_{aw}}(\theta) = \sum \pi_a(a \mid s) A_i - \lambda KL[\pi_{aw} \mid \pi_a] \)
8: Update \( \theta \) by a gradient method w.r.t. \( J_{\pi_{aw}}(\theta) \)
9: end for
10: for \( j \in \{1,\ldots,B\} \) do
11: \( L_\phi(\phi) = -\sum \pi_{aw}(a' \mid s) y^{-i} r - V(s) \)
12: Update \( \phi \) by a gradient method w.r.t. \( L_\phi(\phi) \)
13: end for
14: if \( KL[\pi_{aw} \mid \pi_a] > \beta_{aw} KL_{\pi_{aw}} \) then
15: \( \lambda \leftarrow \alpha \lambda \)
16: else if \( KL[\pi_{aw} \mid \pi_a] < \beta_{aw} KL_{\pi_{aw}} \) then
17: \( \lambda \leftarrow \lambda / \alpha \)
18: end if
19: end for

4. The Proposed Algorithm

4.1. Framework of PPO–DWA Collision Avoidance Planning Algorithm

The framework of the PPO–DWA collision avoidance planning algorithm is shown in the Figure 3. The PPO–DWA collision avoidance planning algorithm consists of two parts, namely the PPO collision avoidance planning method and the modified dynamic window method. These two parts are explained in Sections 4.2 and 4.3, respectively.
The framework of PPO-DWA collision avoidance planning algorithm.

The algorithm process is as follows:

1. Load obstacle environment information and trained network parameters;
2. Enter the execution stage, according to the mobile robot’s own sensors to detect the surrounding obstacle information;
3. The detected obstacle information is transmitted to the training decision stage, and the angular velocity of the next moment is outputted;
4. The modified DWA module accepts the obstacle information detected by the sensor and the angular velocity in the training decision-making stage to produce the next better linear velocity;
5. The execution action module accepts the optimal linear velocity output by the dynamic window method and the optimal angular velocity output in the training decision-making stage, and the robot can drive to a new position through the environmental interaction part;
6. Determine whether the distance between the robot and the target is less than the threshold set in advance; if it is less than the threshold, the robot reaches the target point, and the algorithm ends. Otherwise, it enters step (2).

The PPO-DWA collision avoidance planning algorithm takes into account the yaw limitation of the UUV when designing collision avoidance rules, which can reduce the performance requirements for the UUV. At the same time, the turning motion of the algorithm is trained in the environment, so it has good adaptability to the environment.

4.2. PPO Collision Avoidance Planning Algorithm

This section designs the PPO collision avoidance planning method for the mobile robot. The method is a local path planner that can perform online real-time planning. The framework of the PPO collision avoidance planning method is shown in the Figure 4.
4.2.1. State Space

The state space includes the observation information from the forward-looking sonar and the target point information. Due to the unique nature of underwater environments, acoustic equipment is often used as a sensing instrument for UUVs to perceive the environment. This article carries the SeaBat 8125-H forward-looking sonar model from the RESON company, which can detect a horizontal 120° fan-shaped area with a vertical opening angle of 17°. In normal mode, it contains a total of 240 beams, divided into three layers, each layer containing 80 beams, with a beam angle of 0.5° and a maximum range of 120 m. Because the data returned by the forward-looking sonar is too large, and the high-dimensional obstacle data is not required in the collision avoidance planning task, the obstacle data can be cut by zones to simplify the calculation. This paper divides 120° sensor data into 10 sector regions and uses the minimum distance in each sector region to represent the obstacle distance value in that region. In the collision avoidance process of the mobile robots, the relative angle of the obstacle closest to the mobile robot is directly related to whether the mobile robot can avoid the obstacle, so this is also considered as a part of the state space.

In local collision avoidance planning, the target point information is given as a polar coordinate system, as shown in Figure 5. Using $\rho_t$ to express the distance information between the target point and the UUV at time $t$, $\theta_t$ is the angle of the UUV heading away from the target at time $t$. The distance $\rho_t$ and angle $\theta_t$ are as follows:
Figure 5. Representation diagram of target point information.

The expression for \( \rho \) and \( \theta \) is as follows:

\[
\rho = \sqrt{(x - x')^2 + (y - y')^2} \quad (4)
\]

\[
\theta = \theta - \arctan \frac{y - y'}{x - x'} \quad (5)
\]

\((x, y)\) is the coordinate of the robot in the local tangent plane coordinates, \( \theta \) is the heading angle of the robot, and \((x', y')\) is the coordinate of the target point in the local tangent plane coordinates.

4.2.2. Action Space

The actions in most of the literature are designed to discretize the linear speed and angular velocity or simply discretize the angular velocity. Blind discretization may not take into account the impact of motor performance, which will cause the mobile robot to adjust its heading and linear speed significantly, and it will take a long time for the mobile robot to reach the target action [37]. In order to apply more precise control to the UUV, this paper adopts the continuous action space. However, if the linear velocity and angular velocity are both continuous, the reward function of reinforcement learning is difficult to converge. Therefore, the PPO collision avoidance planning algorithm keeps the linear velocity unchanged and only changes the heading of the UUV; that is, the action space only includes the angular velocity item. Considering the actual situation, the angular velocity of the UUV is limited between \(-5° / s\) and \(5° / s\).

4.2.3. Reward Function

In the framework of the PPO collision avoidance planning algorithm, the design of the reward function is a key point. When training the mobile robot for collision avoidance planning through deep reinforcement learning, if rewards are given only when the UUV reaches the target point, the training process will face the problem of spare reward. Therefore, the corresponding reward value will be given to the agent when the agent makes
decisions. Aiming to enable the mobile robot to avoid obstacles and drive towards the target point, the reward function is set as follows:

\[ R = R_e + R_r + R_c + R_g \]  \hspace{1cm} (6)

\( R_e \) represents the reward received by the agent when it reaches the final target point. If the UUV reaches the target point, then \( R_e = 30 \), otherwise \( R_e = 0 \); \( R_r \) represents the reward received by the agent when colliding with obstacles in the environment. If the UUV collides with an obstacle, then \( R_r = -50 \), otherwise \( R_r = 0 \); \( R_c \) represents the distance reward between the robot and the target point, \( \text{dis}_{\text{goal}} \) represents the distance between the robot and the target, and \( k_{\text{dis}} \) is the corresponding coefficient term. This article sets \( R_c \) as an auxiliary reward, with \( k_{\text{dis}} = 1 \); \( R_g \) represents the heading angle reward, and its specific expression is as follows:

\[ R_g = \begin{cases} k_{\text{dis}} \cdot \cos(\text{angle}_{\text{dis}}) & \text{if } \text{dis}_{\text{goal}} \geq \text{dis}_{\text{obs}} \\ -k_{\text{dis}} \cdot \cos(\text{angle}_{\text{dis}}) & \text{if } \text{dis}_{\text{goal}} < \text{dis}_{\text{obs}} \end{cases} \]  \hspace{1cm} (7)

\( \text{dis}_{\text{obs}} \) represents the minimum distance between the mobile robot and the obstacle, \( \text{angle}_{\text{dis}} \) represents the angle of the robot heading deviation from the target point, \( \text{angle}_{\text{obs}} \) represents the angle of the robot heading deviation from the obstacle which is closest with the robot. \( k_{\text{dis}} \) and \( k_{\text{obs}} \) are corresponding coefficients, both of which are set to 5.

4.3. The Modified DWA Method

Although the PPO collision avoidance planning algorithm can guide the UUV to safely reach the target point, the linear velocity of the UUV is fixed, which can lead to poor collision avoidance performance. When obstacles are too dense, the UUV may not have enough time to make turns. The ideal collision avoidance effect is to reduce the linear velocity of the UUV when the distance between the surrounding obstacles and the UUV is small and to drive toward the target at a faster speed when the robot is far from the obstacle. Therefore, this section proposes the PPO-DWA collision avoidance planning algorithm, which can adjust the linear velocity of the mobile robot by combining the modified DWA algorithm.

The improved DWA algorithm has the same linear velocity sampling space and trajectory estimation method as the traditional DWA algorithm. However, there is no angular velocity sampling in the velocity sampling space; only linear velocity is sampled. This is because the PPO collision avoidance planning algorithm can output the optimal angular velocity in the current state, so only linear velocity sampling is required.

The velocity vector sampling space consists of the following parts:

1. Constraint on maximum and minimum speeds:

\[ V_v = \{ v \mid v \in [v_{\text{min}}, v_{\text{max}}] \} \]  \hspace{1cm} (8)

\( v_{\text{min}} \) and \( v_{\text{max}} \) are the minimum and maximum linear velocities of the robot along the direction of the vehicle body, respectively.

2. Limitation on motor performance:

\[ V_s = \{ v \mid v \in [v_{\text{min}} - \dot{v}_s \Delta t, v_{\text{max}} + \dot{v}_s \Delta t] \} \]  \hspace{1cm} (9)

\( \Delta t \) is the time step; \( v_{\text{min}} \) represents the linear velocity of the robot at the current moment; \( \dot{v}_s \) and \( \dot{v}_s \) are the maximum linear deceleration and maximum linear acceleration, respectively.

3. Limitation on safety during mobile robot driving:
The objective function expression for the modified DWA algorithm is as follows:

\[
G(v, w) = \sigma(\beta \cdot \text{dist}(v, w_p) + \gamma \cdot \text{velocity}(v, w))
\]

where \(w_p\) is the angular velocity output by the PPO collision avoidance planning algorithm, \(\beta\) and \(\gamma\) are the proportions of the \(\text{dist}(v, w_p)\) and the \(\text{velocity}(v, w)\) in the total objective function, respectively.

The original DWA algorithm needs to sample the linear velocity and angular velocity at the same time, and its time complexity is \(O(n^3)\). When predicting the trajectory of each group of sampling speeds, it needs to increase the complexity of the algorithm, which will waste more computing resources and take longer to execute the algorithm. The PPO–DWA collision avoidance planning algorithm combines the characteristics of two algorithms, which can change the linear speed while ensuring real-time planning.

5. Experiments and Results

The experiments are conducted to verify the PPO–DWA algorithm. The programming language used in the experiment is Python, and the compiler version is Python 3.6. The open-source libraries include Pygame 2.4.0, Tensorflow 1.10.0, and Numpy 1.16.0. In the obstacle map, the yellow block represents the starting point and the blue block represents the ending point. The velocity \(U\) in the ocean current interference model is set to 0.1 m/s, and the direction angle \(\psi\) is set to \(\pi/4\) rad.

This experiment takes the UUV model in Section 3 as the research object and selects an area near Songhua Lake in Jilin Province as the obstacle environment. In order to demonstrate the advantages of the algorithm in this article, this section uses the PPO collision avoidance planning method designed in 4.2 and the APF algorithm as the comparison. The performance of algorithms is evaluated from the aspects of path length, the number of steps to reach the target point, and the average solution time of the algorithm. The experimental results are shown in Figures 6 and 7.

Figure 6 shows the experiment results of the PPO algorithm, the APF algorithm and the PPO–DWA algorithm in the obstacle environment 1. From the results of Figure 6a, it can be seen that three collision avoidance planning algorithms can plan a smooth and safe path. Figure 6b–d, respectively, show the comparison curves of the heading angle \(\psi\), longitudinal speed \(u\), and yaw rate \(r\) for the two algorithms, and Figures 6e,f respectively, show the comparison curves of longitudinal thrust and turning moment. Table 2 shows the performance of the two collision avoidance planning algorithms.
Figure 6. Experiment results in the obstacle environment 1.
Figure 7. Simulation results in the obstacle environment 2.

Table 2. Performance comparison in the obstacle environment 1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Path Length/m</th>
<th>Steps</th>
<th>Execution Time/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPO</td>
<td>672</td>
<td>6731</td>
<td>1.5</td>
</tr>
<tr>
<td>PPO–DWA</td>
<td>675</td>
<td>3854</td>
<td>5.15</td>
</tr>
<tr>
<td>APF</td>
<td>653</td>
<td>6542</td>
<td>1.6</td>
</tr>
</tbody>
</table>

From the above experimental results, it can be seen that the path lengths of the three collision avoidance planning algorithms in obstacle environment 1 are similar, with the artificial potential field method having the shortest path length. However, from the heading angle and yaw rate curves, it can be seen that the artificial potential field method can cause frequent and significant changes in the heading of UUV, which puts high...
requirements on the performance and control algorithm of the UUV. At the same time, the path planned by the artificial potential field method may be too close to obstacles. When encountering interference from waves and currents, the UUV may collide with obstacles. In obstacle environment 2, from Figure 7 and Table 3, it can be seen that the artificial potential field method cannot reach the target point. The PPO collision avoidance planning algorithm and PPO–DWA collision avoidance planning algorithm can successfully reach the target point in both obstacle environments, and the path lengths planned by the two algorithms are similar. But, the PPO–DWA collision avoidance planning algorithm can dynamically adjust the linear velocity, so the number of steps to reach the target point is less; that is, the time to reach the target point is shorter. Due to the addition of the DWA module, the average solution time of the PPO–DWA algorithm is much greater than that of the PPO algorithm. However, it is still at the millisecond level, and real-time performance can still be guaranteed. The PPO–DWA collision avoidance planning algorithm can dynamically adjust the linear velocity, which can enable the UUV to reach the target point as soon as possible with guaranteed performance.

Table 3. Performance comparison in the obstacle environment 2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Path Length/m</th>
<th>Steps</th>
<th>Execution Time/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPO</td>
<td>748</td>
<td>7470</td>
<td>1.4</td>
</tr>
<tr>
<td>PPO–DWA</td>
<td>750</td>
<td>3964</td>
<td>4.8</td>
</tr>
<tr>
<td>APF</td>
<td>/</td>
<td>/</td>
<td>1.5</td>
</tr>
</tbody>
</table>

6. Conclusions

In this work, the PPO–DWA collision avoidance planning algorithm can output continuous angular velocity and linear velocity values, which can take less time to plan a smooth, safe, and reasonable path according to the surrounding obstacle information. The PPO–DWA collision avoidance planning algorithm consists of the PPO algorithm and modified DWA. Firstly, the PPO collision avoidance planning algorithm is proposed. Aimed at the task of collision avoidance planning, the simulation environment is built, and the action space, state space, and reward function are designed. Secondly, combined with the modified DWA, the PPO–DWA collision avoidance planning algorithm is proposed. In order to control the linear velocity, the evaluation function and velocity sampling space of the DWA method are modified according to the requirement. Then, the modified DWA can output the best linear velocity the next time according to the obstacle information and the angular velocity information from the PPO algorithm. Finally, it is demonstrated through experiments that the PPO–DWA collision avoidance planning algorithm can plan smooth and feasible paths in irregular real obstacle environments. The PPO–DWA can change the linear and angular velocities of the UUV according to the environment without increasing the training difficulty of the network. At the same time, the action space output by this algorithm can meet the kinematic limitations of the UUV, and there will be no situation where the UUV adjusts its heading significantly in a short period of time.

This article only considers the problem of collision avoidance planning in static obstacle environments. However, in real scenarios, dynamic obstacles inevitably exist in the environment, which limits the use of the algorithm proposed in this article. Subsequently, deep reinforcement learning will be applied to the study of collision avoidance planning for dynamic obstacles, with a focus on processing detection data and predicting the trajectory of dynamic obstacles.

Author Contributions: W.G., methodology, investigation, software, and writing—original and draft; M.H., conceptualization, writing—reviewing and editing; Z.W., writing—original and draft, software; L.D., formal analysis and data curation; H.W., supervision, funding, and acquisition. J.R., methodology, investigation. All authors have read and agreed to the published version of the manuscript.
Funding: This research work is supported by the National Science and Technology Innovation Special Zone Project (21-163-05-ZT-002-005-03), the National Key Laboratory of Underwater Robot Technology Fund (No. JCKYS2022SSXQR-09), and a special program to guide high-level scientific research (No. 3072022QBJ0403).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the anonymous reviewers and the handling editors for their constructive comments that greatly improved this article from its original form.

Conflicts of Interest: Author Mengxue Han was employed by the AVIC China Aero-Polytechnology Establishment. Author Lihui Deng was employed by the Tianjin Navigation and Instrument Institute. The all authors declare no conflict of interest.

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.