Time-Optimal Path Planning of a Hybrid Autonomous Underwater Vehicle Based on Ocean Current Neural Point Grid

Chenhua Hua¹, Nailong Wu¹*, Haodong Yuan², Xinyuan Chen¹, Yuqin Dong¹ and Xianhui Zeng¹

¹ College of Information Science and Technology, Donghua University, Shanghai 201620, China; 2201849@mail.dhu.edu.cn (C.H.); 2201822@mail.dhu.edu.cn (X.C.); 2201810@mail.dhu.edu.cn (Y.D.); xhzeng@mail.dhu.edu.cn (X.Z.)
² SANY Heavy Industry Co., Ltd., Kunshan 215300, China; flyzzxy@163.com
* Correspondence: nathan_wu@dhu.edu.cn

Abstract: Path planning is the precondition for Hybrid Autonomous Underwater Vehicles (HAUV) to enter the submerged area to undertake a mission. The influence of ocean currents on HAUV should be further investigated to obtain a time-optimal path. The improved A* algorithm and the neural network model are employed in this paper to plan a time-optimal path for the vehicle. The HAUV in glider mode is capable of traveling forward mainly through the zigzag motion in vertical plane. Since the vehicle can only receive the command orders when it surfaces from the water, the path is expected to include a series of discrete waypoints in the water surface. At the same time, the presence of submerged riverbeds is also taken into account to avoid hazards for HAUVs when it navigates in the water. It can be demonstrated that ocean currents can be used to decrease the operating time. The comparison results of the two methods verify that the size of the map affects the calculation time. In addition, the neural node represented method surpasses the modified A* method, especially when the map is too large.

Keywords: time-optimal; path planning; underwater motion trajectory; hybrid autonomous underwater vehicle

1. Introduction

1.1. Background

The path planning problem has been a popular research topic regarding mobile robots. Path planning algorithms are now widely used in the civilian and military fields. The civil applications of path planning algorithms include robotic transportation and distribution [1], the automatic driving of civilian UAVs [2,3], intelligent robotic obstacle avoidance planning [4], etc. In addition, the path planning method has also been used in military-related projects, including military UAV formation planning, missile trajectory planning [5], underwater vehicle trajectory tracking [6], etc.

For the path planning of mobile robots on land, many path planning algorithms have been proposed. Generally, path planning methods can be divided into two types. One is the non-bio-inspired algorithm, and the other is the bio-inspired algorithm. Non-bio-inspired algorithms include the graph search probability method [7], the simulated annealing algorithm [8], the artificial potential field method [9,10], the A* algorithm [11], the Dijkstra algorithm [12], and the Floyd algorithm [13]. Bio-inspired algorithms include neural network algorithms [14], particle swarm algorithms [15,16], ant colony algorithms [17,18], and genetic algorithms [19]. A new path planning method has been proposed to plan the motion trajectory of a planar articulated robot in a static workspace [7]. It contains the learning stage and the query phase. During the learning stage, a probabilistic roadmap is built. The nodes of the probabilistic roadmap correspond to collision-free configurations. The edges represent feasible paths between these configurations. In the query phase, the starting and the target configuration of a given mobile robot correspond to the two nodes of
the road map, and the path from the start point to the goal is searched via querying the road map. The simulated annealing algorithm and fuzzy logic algorithm are selected to plan the route for the mobile robot without collision [8]. The artificial potential field method can be combined with the membrane calculation and the genetic algorithm to generate a feasible and safe path [9]. A loose Dijkstra algorithm for real-time path planning of mobile robots in a large scale is developed to adapt to an obstacle-intensive work environment [12]. A new graph-based algorithm based on the encoded Floyd algorithm is designed to optimize the robot path [13]. In addition, a new intelligent path planning method based on artificial neural networks is also presented to avoid obstacles and improve the planning speed of mobile robots [14]. In order to combine the advantages of the ant colony algorithm and the A* algorithm, an adaptive A* hybrid algorithm is designed to obtain the optimal path [18]. Likewise, the improved D* algorithm based on particle swarm optimization is also employed to search for global routes in a dynamic workspace [15]. Furthermore, the neural network algorithm can be applied to fulfill the complete coverage path planning mission of robots in non-stationary environments [20] and the goal assignment task of swarm robots in a 3D environment [21]. The designed network algorithm is expected to generate a smooth and collision-free trajectory for the multi-robot systems [22]. However, the mentioned approach does not consider the time-varying effect of the complex environment on the operation of the vehicle.

With the development of underwater science and technology, path planning algorithms have been further applied in the marine environment [23–29]. To search for the best navigation path in a complex marine environment, an A*-based method is used for path searching from quadtree maps, and the influence of islands is considered in the heuristic function to obtain better search results [30]. Taking into account the potential underwater obstacles or “no entry” areas in the water, an accurate path planning scheme is proposed for the AUV mission planner [31]. A dynamics-constrained global–local (DGL) hybrid path planning scheme is introduced to improve the navigation performance of an autonomous surface vehicles (ASV) in the global path planning and the local hierarchical jointed architecture [32]. In addition, the path planning of the underwater glider is also obtained utilizing the differential evolution to path planning [33]. In order to solve the problem of dynamic path planning of the environmental monitoring USV in complex sea areas effectively, a hybrid algorithm is presented via combining the global path planning with local path planning [34]. Considering the state uncertainty during the random energy sensing path planning, an energy sensing feedback planning method is proposed to deal with the motion uncertainty in the LRAUV kinematic model, as well as the sensor noise [35]. To facilitate the vessel navigation in ocean current and the dynamic scenes with multiple marine vehicles, a time-optimal path for a marine vehicle is introduced [36–38]. In addition, the energy consumption of the vehicle can be reduced by means of ocean currents. The developed dynamic neural network model can guide the underwater vehicle to follow the favorable ocean currents in case that it consumes more energy all the time to combat the impact of surface currents [39]. During the path planning of the multi-AUV systems, a grid of neural point elements can be employed to represent the ocean currents in spatial locations. Thus, the overall sailing distance for the marine vessel’s formation can be reduced [40]. Considering the shortcomings of the PSO algorithm, the PSO algorithm is hybridized with differential evolution (DE) to improve its path searching ability. The generated path can make the AUV safely pass through a marine environment with obstacles [41]. A swarm hyper-heuristic algorithm (SHH) with online learning capability is proposed to solve the path planning problem of AUVs [42]. Aiming at the situation of an AUV driving in turbulent, cluttered, and uncertain environments, a path planning scheme based on annular space decomposition (ASD) is proposed. The method decomposes the search space into annular areas, places control points in each area, and a trajectory is then generated from this set of control points by using splines [43].
1.2. Related Work

The path planning problem is a hot topic in marine applications of underwater robots. Many academics have enhanced the classic path planning approaches and offered novel path planning methods to deal with underwater robot path planning problems. A path planning model based on the differential evolution (DE) algorithm is proposed, which can effectively adapt to the complex and uncertain marine environment. The method can generate the shortest maneuvering safe path [23]. The particle swarm optimization algorithm and reinforcement learning mechanism are combined to form a new algorithm, RMPSO, which considers the ocean current model to save energy for AUVs [24]. A submerged path planning method based on an improved interfered fluid dynamical system (IIFDS) is proposed for the 3D ocean environment, which considers the influence of ocean currents on AUVs and also prevents AUVs from colliding with dynamic obstacles [25]. For the case of AUVs operating in marine environments with dynamic and uncertain flow fields, an interval optimization (IO) scheme is proposed, which generates time-optimal paths [26]. An ensemble algorithm combining the belief function method (BF) and the velocity synthesis algorithm (VS) is proposed, which considers the influence of time-varying current on AUV. Furthermore, it can guide an AUV in marine environments [27]. For marine environments with strong ocean currents and enhanced space-time variability, a genetic algorithm is developed to find paths with minimal energy costs [28]. A new path planning method is proposed based on ocean current prediction, which can utilize the favorable ocean currents to find more optimal trajectories [29]. Ref. [39] proposes a dynamic neural network that can quickly adapt to dynamically changing environments to solve the shortest time paths in a map. In the neural network, the connection weight between adjacent neurons takes into account the influence of the ocean current on the AUV, and it only increases in the direction that can effectively utilize the ocean current, so that the connection weight has the characteristics of asymmetric anisotropy.

1.3. Article Structure

The rest of the article is organized as follows. In the second section, the motion form of a Hybrid Autonomous Underwater Vehicle (HAUV) in glider mode and the process of path planning are introduced. In the third section, the improved path planning method and the waypoint calculation method are proposed. In the fourth section, the simulation research is carried out. A summary is presented in the fifth section.

The main contributions of the article are summarized from the following three aspects:

1. In the paper, two path planning algorithms are run in different grid maps to plan time-optimal paths. The two algorithms are also compared in terms of program running time. In fact, a suitable path planning method can not only improve the efficiency of the path search, but it can also obtain time-optimal solution.
2. The path is numerically separated to produce a series of waypoints related to the underwater motion of the HAUV in glider mode. The article simulates the motion trajectory of the HAUV based on the waypoints.
3. The influence of the riverbed on the HAUV is considered when waypoints are planned, so that the HAUV is capable of avoiding hitting the riverbed while traveling.

2. Problem

2.1. HAUV Overview

The underwater glider has the advantages of a long operation time and a wide detection depth, and it has been used for marine environmental monitoring and data collection [44]. The HAUV has two working modes, namely, the glider mode and the AUV mode. The HAUV in glider mode meets the needs of long-distance navigation and low energy consumption, while the HAUV in AUV mode meets the requirements of rapid maneuverability and operation at a certain depth.

The subject of the paper is the HAUV in glider mode. In glider mode, the HAUV can undertake multi-cycle gliding motions in the water. When the HAUV dives into the
sea and glides along a V-shaped trajectory to sea level, it completes a period of gliding motion. As the HAUV travels along a predetermined path through the water, the sawtooth motion of the HAUV causes it to emerge from and dive into the ocean as it travels. As shown in Figure 1, when the HAUV fulfills a multi-cycle saw-tooth motion, the outgoing waypoints should be set. The HAUV receives commands at the water surface to determine the direction and depth of the next gliding movement.

Figure 1. Multi-cycle saw-tooth motion of the HAUV in glider mode.

2.2. Path Planning

2.2.1. Path Planning under Ocean Currents

As a drone that often travels long distances in the sea, the HAUV in glider mode has a low speed and is easily affected by ocean currents. The HAUV deviates from its intended trajectory when influenced by ocean currents. The characteristics of the HAUV will affect its tracking performance from the start point to the end point. Therefore, the HAUV path planning problem is how to find a time-saving feasible path in the ocean grid map. To ensure the operating safety of the HAUV, the searching space cannot include any obstacles.

The ocean searching space is a three-dimensional area, the three-dimensional ocean area can be divided into two parts: a two-dimensional ocean plane and an underwater area containing the riverbed. The process of establishing a path in the ocean plane can be divided into three steps. A grid map that can be used for path search is created based on the two-dimensional ocean plane. Then, a suitable algorithm is employed to search for the path, and finally the time-optimal path is separated to obtain the waypoints on the surface plane.

The entire two-dimensional ocean plane is processed to generate a grid map that can represent the two-dimensional ocean plane. The grid map consists of uniform and discrete grids. In the grid map, the point where a row and a column intersect is defined as a grid node. After that, the path planning algorithm is used to plan the time-optimal path from the start point \( n_{\text{start}} \) to the end point \( n_{\text{goal}} \) in the grid map. The path consists of a series of grids, which is expressed as:

\[
p = \left( n_{\text{start}}, \ldots, n_i, n_{i+1}, \ldots, n_k, n_{\text{goal}} \right)
\]

(1)

where \( n_j = (x_j, y_j) \) is the coordinate of grid \( j \).

The time-optimal path satisfies the requirement of consuming the minimum total time from the start point to the end point on the grid map, which is expressed in mathematical notation as:

\[
\min TIME = \min \sum_{i=\text{start}}^{k} \text{time}(i, i+1)
\]

(2)

where \( \text{time}(n, n+1) \) denotes the travel time for the HAUV moving from node \( n \) to adjacent node \( n + 1 \).

2.2.2. HAUV Travel Planning

The attitude of the HAUV in glider mode is controllable. The HAUV is expected to reach the end point along a predetermined trajectory. The treated path is separated into a
series of waypoints, and the vehicle arriving at each waypoint is capable of transmitting messages between the ground control station and the HAUV. Thus, the control command of the HAUV can be renewed at each waypoint, and the behavior control orders can be calculated for the vehicle during the next gliding motion.

In the submerged area containing the riverbed, the riverbed can pose a threat to the underwater travel of the HAUV, as shown in Figure 2. If the diving depth of the HAUV is not reasonably controlled when setting the waypoint, it may hit the riverbed and become stuck. Assuming that the HAUV executes each command correctly, the travel planning goal of the HAUV is to obtain proper waypoints on the path so that the HAUV can safely travel from the start point to the end point in the three-dimensional marine environment.

3. Method
3.1. Global Path Planning
3.1.1. The Grid Map

Obstacles and ocean currents should be considered in the path planning on the two-dimensional ocean surface plane, and hence the grid map can be divided into a feasible space set and an impassable space set. Because the HAUV can only operate under water, land is regarded as an obstacle. On the water surface plane, the obstacle area represents the impassable area, and the non-obstacle area can be processed to obtain the optimal path. The two-dimensional ocean plane can be separated into M rows and N columns using straight lines in the horizontal direction and the vertical direction. As for the three-dimensional marine environment, the water depth data are also added into the grid map. The grid node where the water depth data is less than 0 m is set as an obstacle node. Water flow data for each grid node are also added to the map. Each grid has the same side length $\Delta L$. The ocean current is defined using a vector in the grid map. As shown in Figure 3, a two-dimensional ocean plane map contains 14 rows by 25 columns nodes. Ocean currents and barriers are plotted on the plane with uniform grids.

![Figure 2. Three-dimensional underwater area with the riverbed.](image)

![Figure 3. A grid map with ocean water surface features such as obstacles and ocean currents.](image)

3.1.2. Velocity Modeling

In the grid map, each grid has eight adjacent nodes. The center node can be connected with the eight adjacent nodes in different directions, namely, the vertical direction, the horizontal direction and the diagonal direction.

The Euclidean distance $d(x, y)$ between node $x = (x_1, y_1)$ and node $y = (x_2, y_2)$ in a two-dimensional plane is expressed as:
The distance $L$ between the node $n$ and the adjacent node $n+1$ in the grid map is expressed as:

$$L = \Delta L \cdot d(n, n+1)$$

Ocean current data are adopted to obtain the total velocity in the grid map [38]. The ocean current vector $\vec{V}_c$, HAUUV speed vector $\vec{V}_g$, and the HAUUV’s total velocity vector $\vec{V}_{all}$ with the superimposed ocean current vector of node $n$ are shown in Figure 4.

![Figure 4. Ocean current vector $\vec{V}_c$, HAUUV speed vector $\vec{V}_g$, and total velocity vector $\vec{V}_{all}$ of the node $n$ in the grid map.](image)

The ocean current vector $\vec{V}_c$ and the HAUUV’s own speed $V_g$ are unchangeable fixed values over a longer period of time. During the path searching in the grid map, the HAUUV’s speed $V_g$ can be renewed. When the HAUUV moves from the node $n$ to the neighboring node $n+1$, the heading angle $\theta_{all}$ of the node $n$ should be equal to the angle between the vector connected from the node $n$ to the adjacent node $n+1$ and the horizontal line. During the optimization process, $\theta_{all}$ starts from $0^\circ$ and gradually increases by $45^\circ$ to $315^\circ$. The total velocity $V_{all}$ of node $n$ is an unknown quantity. The total velocity calculation formula of each grid is expressed as:

$$\vec{V}_{all} = \vec{V}_c + \vec{V}_g$$

The ocean current distribution between node $n$ and node $n+1$ is set as the ocean current at node $n$. The total velocity $V_{all}$ for the HAUUV moving from each node $n$ to the adjacent node $n+1$ is solved using Equation (8). The equation is created by combining $\theta_{all}$, $\theta_c$, $\theta_g$, $V_c$, and $V_{all}$. The equation takes $V_{all}$ as the independent variable. Based on Equation (5), an equation is created on the horizontal line, which is expressed as:

$$V_{all} \cos \theta_{all} - V_c \cos \theta_c = V_g \cos \theta_g$$

and an equation is created on the vertical line, which is expressed as:

$$V_{all} \sin \theta_{all} - V_c \sin \theta_c = V_g \sin \theta_g$$

Then, the sum of squares of the two equations is calculated and simplified into a quadratic equation with the independent variable, which is expressed as:

$$V_{all}^2 - 2V_c \cos (\theta_c - \theta_{all}) V_{all} + \left(V_c^2 - V_g^2\right) = 0$$

Then, the root can be obtained, which is expressed as:

$$\begin{align*}
x_1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\
x_2 &= \frac{-b - \sqrt{b^2 - 4ac}}{2a}
\end{align*}$$

By bringing $a = 1$, $b = -2V_c \cos (\theta_c - \theta_{all})$, $c = V_c^2 - V_g^2$ into the solution equation, the value of the independent variable can be determined.
If the quadratic equation has a positive real solution, it means that the HAUV can travel from the current node \( n \) to the adjacent node \( n + 1 \) successfully. The negative or complex solutions is meaningless. If two positive real solutions appear, the larger one is chosen as \( \forall \). For each node \( n \), since the adjacent nodes are different, a quadratic equation for the velocity \( \forall \) should be established at node \( n \) for each time.

If the HAUV can travel from the current node \( n \) to the adjacent node \( n + 1 \), the travel time can be calculated. The travel speed \( \forall \) and the moving distance \( L \) for the HAUV moving from node \( n \) to the node \( n + 1 \) are obtained. The travel time \( \text{time}(n,n+1) \) for the HAUV moving from node \( n \) to adjacent node \( n + 1 \) is expressed as:

\[
\text{time}(n,n+1) = \frac{L}{\forall} \tag{10}
\]

3.1.3. The Improved A* Algorithm

As a direct search method for solving the shortest path, traditional A* algorithm is the most popular searching method in a static map. The traditional A* algorithm formula is expressed as:

\[
f(n) = g(n) + h(n) \tag{11}
\]

where \( g(n) \) represents the minimum cost for the robot moving from the start point to node \( n \) in the map, and \( h(n) \) indicates the minimum estimated cost for the robot moving from node \( n \) to the end point. It can search one path with the shortest distance in the map at the cost of distance.

The A* algorithm based on optimal time is proposed [38]. The estimated distance of the A* algorithm in the grid map is set as the Euclidean distance or Octile distance, and the estimated distance of the A* algorithm in the non-uniform network map constructed by RRTs is set as the Euclidean distance [38]. In this paper, the estimated distance of the A* algorithm is set as the Euclidean distance in a uniform grid map. The modified A* algorithm’s formula is expressed as:

\[
f^*(n) = g^*(n) + h^*(n) \tag{12}
\]

where \( f^*(n) \) represents the minimum time cost estimated for moving the HAUV from the start point to the end point via node \( n \).

\[
g^*(n) = g^*(n-1) + \text{time}(n-1,n) \tag{13}
\]

where node \( n \) is the adjacent node of node \( n - 1 \), \( g^*(n) \) represents the minimum time cost for moving the HAUV from the start point to node \( n \) in the map. Firstly, the feasibility of moving HAUV from node \( n - 1 \) to node \( n \) is determined. If the vehicle cannot travel from node \( n - 1 \) to node \( n \), skip node \( n \) and process the next neighboring node.

\[
h^*(n) = \frac{\Delta L \cdot d(n,\text{goal})}{\max(V_c) + V_g} \tag{14}
\]

where \( h^*(n) \) represents the minimum estimated time cost for HAUV moving from node \( n \) to the end point. \( h^*(n) \) defines the influence of ocean currents on the future travel of HAUV, and the minimum estimated time for HAUV traveling from node \( n \) to the end point under the influence of the maximum ocean current can be calculated.
3.1.4. Neural Network Model

A new dynamic neural network model is proposed to search time-saving paths solution in ocean environments [39]. In this paper, the model is utilized to improve the searching process of the HAUV, especially the travel time optimization.

\[ \omega_{ij} = \begin{cases} e^{-\gamma \cdot \text{time}(i,j)}, & d(i,j) < r \\ 0, & \text{others} \end{cases} \quad (15) \]

where \( \omega_{ij} \) represents the connection weight from a certain neuron \( j \) to \( i \), as shown in Figure 5. Grids represent neurons on a grid map. The centers of two adjacent neurons \( i \) and \( j \) are connected. \( d(i,j) \) gives the Euclidean distance between two centers. \( r \) means that the neuron \( i \) can only touch neurons within a circle with radius \( r \). If \( r = 2 \), the neuron \( i \) can only contact with eight neighboring neurons.

**Figure 5.** The connection weight \( w_{ij} \) from adjacent neuron \( j \) to neuron \( i \) in the grid map.

\[ I_i = \begin{cases} v, & \text{if } i \text{ is destination} \\ -v, & \text{if } i \text{ is obstacle} \\ 0, & \text{others} \end{cases} \quad (16) \]

where \( I_i \) defines a labeled input, which means that neuron \( i \) is either a drivable unit, an obstacle unit, or a destination unit, when \( v \gg 1 \).

\[ g(x) = \begin{cases} 1, & x \geq 1 \\ \beta \cdot x, & 0 < x < 1, \beta > 0 \\ 0, & \text{others} \end{cases} \quad (17) \]

where \( g(x) \) represents a piecewise function, which can ensure the neural activity value \( x_i(t) \epsilon [0,1] \) of neuron \( i \).

\[ x_i(t) = g \left( \max_{j \in S_i} \left( \omega_{ij} \cdot x_j(t-1) \right) + I_i \right) \quad (18) \]

where \( x_i(t) \) represents the neural activity value of neuron \( i \) in the current iteration, \( x_i(t-1) \) is the neural activity value of neuron \( j \) in the previous iteration. It can be obtained by multiplying the neural activity value \( x_j(t-1) \) of a surrounding neuron \( j \) in the previous iteration by the connection weight \( \omega_{ij} \) from a surrounding neuron \( j \) to the neuron \( i \). When \( r = 2 \), neuron \( i \) has eight neighboring neurons. \( S_i \) is the set of eight adjacent neurons of neuron \( i \) when \( r = 2 \). All neurons in the map can be used to calculate the value of the same direction at the same time, or the eight values in a clockwise direction.

\[ E_n(i) = k|\omega_{ik} \cdot x_k(t-1) = \max_{j \in S_i} (\omega_{ij} \cdot x_j(t-1)) \quad (19) \]

where \( E_n(i) \) represents the stimulating neuron of neuron \( i \). \( E_n(i) \) can record the propagation direction. If the adjacent neuron \( j \) corresponding to the maximum value is regarded as the
stimulating neuron of neuron \( i \), it means that the neural activity of neuron \( i \) is propagated by adjacent neuron \( j \).

The parameters of the neural network model are selected as \( \beta = 1, \gamma = 0.0005, v = 200, r = 2 \) to ensure the effectiveness of the neural network model [39]. The start point and end point are determined on a map, and then the neural network model is performed. After multiple iterations of calculation, all neurons in the map will converge (neural activity values do not change). Since all the neurons values are stable, the path is planned by following the stimulus neurons one by one.

3.2. HAUV Travel Planning

The dive depth \( h \) of the HAUV is proportional to the horizontal distance \( f \) in one cycle of the gliding motion. The relationship between \( h \) and \( f \) is expressed as:

\[
\frac{h}{f} = \frac{1}{5} \quad (20)
\]

A three-dimensional deep sea area is selected. The process of HAUV travel planning consists of three steps. Firstly, the path is created on the water surface and divided into small line segments by the inflection points. Then, the small paths are separated sequentially in the grid map according to Equation (20) to generate a series of waypoints, as shown in Figure 6. Finally, the HAUV is assumed to be able to execute commands correctly at each waypoint and the adjacent waypoints are connected in the three-dimensional marine environment to form the desired underwater trajectory.

![Figure 6. The path with waypoints on the ocean water surface containing obstacles and ocean currents.](image)

The maximum dive depth value of the HAUV itself is set to \( \lambda \). The maximum distance \( \mu \) from sea level to the underwater riverbed is calculated on each small path. The minimum value of \( \lambda \) and \( \mu \) is chosen on each small path, the maximum dive depth \( \epsilon \) of the HAUV on the small path can then be determined. The dive depth of the HAUV between adjacent waypoints should be set properly to be less than \( \epsilon \) when planning waypoints on the small path to ensure that the HAUV is safe during the diving phase.

4. Simulation

4.1. Data Description

The Ocean Surface Current Analysis (OSCAR) data are obtained from JPL Physical Oceanography DAAC and developed by ESR. The data cover the surface of the Earth. Data from 26 January and 5 February 2016 are selected as ocean current data for the experiment.

4.2. HAUV Description

The object of the study is the HAUV in glider mode. In the path planning searching process, the HAUV is regarded as a moving Newtonian particle. In the experiment, the HAUV’s own speed is set at 0.2–0.5 m/s. The maximum diving depth of the HAUV itself
is set at 1000 m. It is supposed that the HAUUV can execute the command correctly at each water exit point.

4.3. Simulation Results

The path planning is carried out in the experiment to evaluate the effectiveness of the modified algorithms in path planning. The simulation is implemented using Matlab R2020a. In Matlab, the parallel calculation channel can be manually opened before running the program, or the parallel calculation channel can be automatically opened when running the program.

A certain three-dimensional ocean area is chosen on Earth. A grid map with static ocean currents is designed based on the ocean plane. No boxes are drawn on the grid map. Each ocean current arrow indicates the presence of a grid. The side length ∆L of the uniform grid is 1000 m.

Three levels of ocean current modes are set on the same map, namely, the strong ocean current mode, middle ocean current mode, and weak ocean current mode. In the strong ocean current mode \( \frac{\text{max}(V_c)}{V_g} = 2.5 \), which means that the maximum ocean current speed is more than twice the speed of the HAUUV itself. It is set that \( \frac{\text{max}(V_c)}{V_g} = 1.5 \) in the middle ocean current mode, which means that the maximum ocean current speed exceeds the speed of the HAUUV itself. In the weak ocean current mode \( \frac{\text{max}(V_c)}{V_g} = 0.5 \), which means that the maximum ocean current speed is lower than the HAUUV’s own speed.

4.3.1. Global Path Planning

1. Comparison of traditional A* algorithm and improved A* algorithm

It is set that \( V_g = 0.4107 \) m/s in the grid map. The paths are that the HAUUV at the grids (1,1), (26,8), and (1,17) travels to the grids (20,17), (15,8), and (20,17). The grid map is in the strong ocean current mode. The traditional A* algorithm and the improved A* algorithm are performed to establish the paths.

The results generated by the traditional A* algorithm and the improved A* algorithm are shown in Figure 7. Obstacles are represented by black zones. The size of the ocean current is represented by the length of the arrow. The path generated by the traditional A* algorithm consists of a series of cyan hexagons. The path generated by the improved A* algorithm consists of a series of magenta circles.

![Figure 7](image.png)

**Figure 7.** The routes established by the traditional A* algorithm and the improved A* algorithm from the grids (1,1), (26,8), and (1,17) to the grids (20,17), (15,8), and (20,17) in the grid map. The paths generated by the traditional A* algorithm are indicated by the cyan lines. The paths generated by the improved A* algorithm are indicated by the magenta lines.

The time cost of the planned paths in the grid map is displayed in Table 1. It can be seen from Table 1 that the results of the traditional A* algorithm are not time-optimal solutions compared to the results of the improved A* algorithm. The path established by the improved A* algorithm follows favorable ocean currents, allowing the HAUUV to reach the end point with the help of the ocean current. In addition, the improved A* algorithm reduces the time cost of the path with the help of ocean currents. Thus, the results of the
improved A* algorithm are more favorable to HAUV navigation in the ocean plane than the results of the traditional A* algorithm.

Table 1. Time cost in seconds of the routes established by two algorithms in the grid map.

<table>
<thead>
<tr>
<th>The Route</th>
<th>Time Cost of the Route Established by the Traditional A* Algorithm</th>
<th>Time Cost of the Route Established by the Improved A* Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>The route from (1,1) to (20,17)</td>
<td>∞</td>
<td>45,973</td>
</tr>
<tr>
<td>The route from (26,8) to (15,8)</td>
<td>16,462</td>
<td>13,196</td>
</tr>
<tr>
<td>The route from (1,17) to (20,17)</td>
<td>35,565</td>
<td>30,091</td>
</tr>
</tbody>
</table>

The path from the grid (1,1) to the grid (20,17) established by the traditional A* algorithm is not practicable in the ocean plane, and the HAUV cannot reach the destination along the path. The path from the grid (1,1) to the grid (20,17) established by the improved A* algorithm can guide the HAUV to the destination by following ocean currents. The path from the grid (26,8) to the grid (15,8) established by the traditional A* algorithm can also transport the HAUV to the end point; however, this path takes more time than that established by the improved A* algorithm. The path from the grid (1,17) to the grid (20,17) established by the improved A* algorithm saves time by taking use of the favorable currents, which is superior to that established by the traditional A* algorithm.

2. Paths generated by the improved A* algorithm and neural network model

Ten grid maps with areas ranging from small size to large size are established. Each map is set to the strong ocean current mode. For example, the plane is divided into 14 rows and 25 columns. Marks are made on the plane to form a grid map. The grid map is defined as $14 \times 25$ in size. The remaining nine grid maps are $18 \times 38$, $19 \times 34$, $32 \times 70$, $47 \times 104$, $106 \times 237$, $70 \times 156$, $46 \times 104$, $71 \times 158$, and $21 \times 47$. The start point and end point are diagonal on the map and far apart. The improved A* algorithm and the one represented by a neural network model are carried out to obtain the optimal paths on each map.

Three types of grid maps are used for effect demonstration. The first type is the grid map where only the ocean currents exist. The second type is the grid map with a few obstacles. The third type is the grid map with a large number of obstacles.

It is set that $V_g = 0.2865$ m/s in a $32 \times 70$ map. The path has the HAUV start at the grid (2,31) and travel to the grid (66,11). In the $21 \times 47$ map, $V_g = 0.3921$ m/s. The path has the HAUV start at the grid (8,20) and travel to the grid (47,2).

It is set that $V_g = 0.2774$ m/s in the $47 \times 104$ map. The path has the HAUV start at the grid (100,3) and travels to the grid (5,45). In the $70 \times 156$ map, $V_g = 0.3006$ m/s. The path has the HAUV start at the grid (102,2) and travel to the grid (5,44).

It is set that $V_g = 0.4894$ m/s in the $106 \times 237$ map. The path has the HAUV start at the grid (5,104) and travel to the grid (229,30). In the $70 \times 156$ map, $V_g = 0.2969$ m/s. The path has the HAUV start at the grid (5,68) and travel to the grid (110,2).

Figures 8-10 show that two algorithms can be used to solve the HAUV path planning problem. The results of the neural network model consist of a series of yellow fork shapes. The paths generated by the improved A* algorithm consist of a series of magenta circles. The paths indicated in Figure 8a guide the HAUV along favorable ocean currents to the end point in the $32 \times 70$ map. The paths depicted in Figure 8b are time-optimal solutions that direct the HAUV to travel quickly to the end point in the $21 \times 47$ map. The paths indicated in Figure 9a guide the HAUV safely to the end point in the $47 \times 104$ map. The paths depicted in Figure 9b act as time-saving paths that satisfy the requirement of time optimality. In the $106 \times 237$ map with a large number of obstacles, the paths indicated in Figure 10a guide the HAUV to avoid obstacles and follow favorable ocean currents towards the end point. In the $70 \times 156$ map with a large number of obstacles, the paths indicated in Figure 10b direct the HAUV to the end point quickly.
3. Comparison of working time

The time that the improved A* algorithm and the neural network model take to work in the above ten maps is recorded. The parallel computation channel is turned off in Matlab, after which the improved A* algorithm is implemented. The neural network model is run in two cases. The parallel computing channel is turned on in Matlab, and the neural network model is carried out afterwards. The parallel computing channel is turned off in
Matlab, and the parallel computing channel is set to be automatically turned on when the neural network model is implemented.

Figure 11 shows the working time of both algorithms. The working time of the improved A* algorithm is indicated by the blue line. The working time of the neural network model is indicated by the black line when the parallel computing channel is automatically turned on. The neural network model working time when parallel computing is turned on early is indicated by the cyan line. In the case that parallel computing channel is turned off in advance, the working time of the improved A* algorithm is shorter than that of the neural network model when the area is small, and the opposite is true when the area is too large. In the case that the parallel computation channel is opened in advance, the working time of the neural network model is shorter than that of the improved A* algorithm, regardless of the map size. The algorithm can be selected to build the path depending on the actual situation.

4. Advantages

Three ocean current modes are set on the above ten grid maps. The start and end points are unchanged. The time cost of each path is counted, and defined as the HAUV’s travel time on the graph.

Figure 12 shows that both algorithms work properly in all three ocean current modes. The HAUV’s travel time is the shortest when the grid map is in the strong ocean current mode of the three ocean current modes. It indicates that the path produced by the two approaches in the strong ocean current mode can cost less time than the paths established in the other modes.

Figure 12. HAUV travel time on the paths established by two algorithms in the ten grid maps. (a) HAUV travel time on the paths established by the improved A* algorithm in the ten grid maps. (b) HAUV travel time on the paths established by neural network model in the ten grid maps.
4.3.2. HAUV Travel Planning

The underwater trajectory of the HAUV is simulated in the three-dimensional marine environment corresponding to the $71 \times 158$ grid map. The $71 \times 158$ map is set to be in the strong ocean current mode. It is set that $V_g = 0.3063$ m/s in $71 \times 158$ map. The start point (156, 2) and the end point (40, 61) are determined in the $71 \times 158$ map. Since the neural network model works efficiently, the neural network model is used to search for the time-optimal path in $71 \times 158$ map.

Figure 13 shows the trajectory of the HAUV moving from (156, 2, 0) to (40, 61, 0) in the three-dimensional marine environment. The end point is represented by a blue star, while the start point is represented by a blue circle. The waypoint is shown as a green star, and the HAUV’s underwater motion trajectory is displayed by the green lines.

![Figure 13](image_url)

Figure 13. The underwater trajectory from (156, 2, 0) to (40, 61, 0) in the three-dimensional marine environment corresponding to the $71 \times 158$ map.

In the grid map, twenty-three inflection points, including the start point and the end point, occur on the path. The adjacent inflection points form a small path, and the maximum dive depth $\epsilon$ of the HAUV on each small path is 1000 m. The number of waypoints is 46, including the start point and the end point. In the trajectory, the maximum depth that the HAUV can dive is 848.5281 m, and the minimum dive depth is 200 m. Figure 13 demonstrates that the trajectory of the HAUV underwater is feasible and safe because it does not touch the riverbed. It means that the waypoint formulation strategy proposed in the paper is effective.

5. Conclusions

In the paper, the path planning for the HAUV in glider mode is studied. HAUVs tend to deviate from the intended path under the influence of ocean currents, so the ocean currents should be taken into account during the path planning. If an inappropriate path planning algorithm is selected when searching for a path, the calculation time may be too long, and the working efficiency can be reduced. The article adopts an improved A* algorithm and a neural network model to obtain the time-optimal path. Ocean currents are efficiently exploited in path planning to reduce path travel time. The program running time of the improved A* algorithm and the program running time of the neural network model are compared. The results show that the running time of the neural network model is less than the running time of the improved A* algorithm when the map is larger. In addition, the underwater motion trajectory of the HAUV is simulated in the three-dimensional marine environment. The results show that the generated waypoints can guide the HAUV to avoid the riverbed and travel safely to the end point.
**Author Contributions:** Conceptualization, C.H. and N.W.; methodology, C.H. and N.W.; writing—original draft paper, C.H. and N.W.; writing—review and editing, C.H. and N.W.; formal analysis, X.Z.; supervision, H.Y.; software, X.C. and Y.D.; visualization, X.C. and Y.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research carried out in this paper was supported by the National Natural Science Foundation of China under grant number 52101346 and the Fundamental Research Funds for the Central Universities of Donghua University.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

**References**


25. Yao, P.; Zhao, S. Three-dimensional path planning for autonomous underwater vehicle based on interval optimization in uncertain flow fields. *Ocean. Eng.* **2021**, *234*, 108675. [CrossRef]


