Journal of
Marine Science and Engineering

MDPI

Article

Three-Dimensional Path Planning for AUVs Based on Standard Particle Swarm Optimization Algorithm

Bangshun Zhan, Shun An, Yan He * and Longjin Wang *

College of Mechanical and Electrical Engineering, Qingdao University of Science and Technology, Qingdao 266061, China
* Correspondence: heyan@qust.edu.cn (Y.H.); wljwlj1984@126.com (L.W.)

Abstract: This paper proposes an improved standard particle swarm optimization 2011 for autonomous underwater vehicles (AUVs). A mutation operator with a threshold is introduced to solve the problem of particles falling into the local extreme, and a nonlinear adaptive parameter strategy is introduced to accelerate the convergence speed. The proposed algorithm considers "path length", "path safety", "path smoothness" and "physical constraints" synthetically. For the specific navigation environment of AUVs, the path planning simulation is conducted based on MATLAB/Simulink, and the navigation guidance and control closed-loop simulation system is established. Simulation results show the effectiveness of the proposed algorithm.

Keywords: AUVs; path planning; improved SPSO-2011 algorithm; three-dimensional guidance algorithm

check for updates

Citation: Zhan, B.; An, S.; He, Y.; Wang, L. Three-Dimensional Path Planning for AUVs Based on Standard Particle Swarm Optimization Algorithm. *J. Mar. Sci. Eng.* **2022**, *10*, 1253. https://doi.org/10.3390/jmse10091253

Academic Editor: Rafael Morales

Received: 21 July 2022
Accepted: 30 August 2022
Published: 5 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Nowadays, autonomous underwater vehicles (AUVs) are extensively employed in military and civilian areas, such as geological sampling [1], mine clearance [2], ocean floor survey [3], oil and gas exploration [4,5], and so on [6]. Because AUVs always need to be fully self-contained accomplish intelligently, and decision-making action during its mission, much research has been carried out worldwide with particular emphasis on autonomy, control, and navigation [7]. Recently, the development in the field of navigation, guidance and control systems has contributed significantly to the progress achieved in the development of AUVs. Naeem W. et al. [8] extensively discussed some effective guidance laws applicable to AUVs such as Lyapunov-based guidance, Proportional Navigation guidance (PNG), and Line-of-Sight (LOS) guidance. M. Breivik et al. [9] proposed the Pure Pursuit (PP) guidance and the Constant Bearing (CB) guidance in the case of AUVs. With an appropriate guidance strategy, an AUV can effectively perform its tasks by successively passing through the fixed waypoints toward the destination.

Path planning is an essential problem in autonomous control technology. To a certain extent, it reflects the intelligent level of AUVs [10]. At present, path planning can be divided into two categories: global and local path planning. The goal of global path planning is to find a feasible path in a known static environment, while local path planning aims at searching for an optimal or suboptimal path in unknown environments. In past decades, many approaches have been proposed for path planning of AUVs, including the Dijkstra algorithm [11], A* algorithm [12], artificial potential field algorithm (APF) [13], genic algorithm (GA) [14], particle swarm optimization (PSO) algorithm [15,16], neural network [17] and reinforcement learning [18,19]. However, the single path planning algorithm has limitations in specific scenarios. Therefore, combining different algorithms to solve the path planning problem for AUVs becomes more and more important. For example, Lim et al. proposed a PSO algorithm for the path planning of AUVs by introducing a selective differential evolution strategy where the complexity of computation was reduced significantly [20]. In [21], an ant colony optimization algorithm was developed by employing the pheromone elimination method, which can overcome the local extremum

easily. To solve the premature problem in the conventional genetic algorithm, Yan and Pan presented a modified genetic algorithm by building a route planning model, which can quickly find the optimal path [22]. Therefore, the combination of different path planning algorithms attracted more and more researchers' attention [23].

To quickly and accurately plan the safe optimal paths for AUVs in complex navigation environments, this paper proposes an improved standard PSO 2011 (SPSO-2011) algorithm. The contributions of this article are as follows:

- The proposed algorithm introduces a mutation operator with a threshold and a nonlinear adaptive parameter strategy, which can effectively shorten the length of the path and accelerate convergence speed compared with the PSO algorithm and the SPSO-2011 algorithm.
- A 3D guidance and control system is carried out to verify the effectiveness of the proposed algorithm. Simulation results show that the proposed algorithm has certain advantages in terms of path length and path rationality.

The remainder of this article is organized as follows. Section 2 establishes the AUV kinematic and dynamic models. Section 3 proposes a 3D guidance and control system for AUVs. Section 4 establishes the fitness function of the path. Section 5 proposes an improved SPSO-2011 algorithm. Simulation results are presented in Section 6. Finally, we offer some concluding remarks in Section 7.

2. AUV Kinematic and Dynamic Models

According to the AUV motion properties and neglecting the roll dynamics, the 5-degree-of-freedom (DOF) kinematic and dynamics models of an underactuated AUV can be expressed as [24]:

$$\begin{aligned}
\dot{x} &= u\cos(\theta)\cos(\psi) - v\sin(\psi) + w\sin(\theta)\cos(\psi),\\
\dot{y} &= u\cos(\theta)\sin(\psi) + v\cos(\psi) + w\sin(\theta)\sin(\psi),\\
\dot{z} &= -u\sin(\theta) + w\cos(\theta),\\
\dot{\theta} &= q,\\
\dot{\psi} &= r/\cos(\theta),
\end{aligned} \tag{1}$$

$$\begin{aligned}
\dot{u} &= \frac{m_{22}}{m_{11}}vr - \frac{m_{33}}{m_{11}}wq - \frac{d_{11}}{m_{11}}u + \frac{1}{m_{11}}\tau_u,\\
\dot{v} &= -\frac{m_{11}}{m_{22}}ur - \frac{d_{22}}{m_{22}}u,\\
\dot{w} &= \frac{m_{11}}{m_{33}}uq - \frac{d_{33}}{m_{33}}w,\\
\dot{q} &= \frac{m_{33}-m_{11}}{m_{55}}uw - \frac{d_{55}}{m_{55}}q - \frac{\rho g \bigtriangledown GM_L\sin(\theta)}{m_{55}} + \frac{1}{m_{55}}\tau_q,\\
\dot{r} &= \frac{m_{11}-m_{22}}{m_{66}}uv - \frac{d_{66}}{m_{66}}r + \frac{1}{m_{66}}\tau_r,
\end{aligned} \tag{2}$$

where x, y, z, θ and ψ are the positions and attitude of the AUV in the earth-fixed frame in Figure 1; u, v, w, q and r are the corresponding linear and angular velocities in the body-fixed frame; τ_u, τ_q and τ_r are the control forces and moments of the AUV produced by thrusters and propellers; m_{ii} $(i = 1, 2, 3, 5, 6)$ denote the inertia and added mass parameters of the AUV, and d_{ii} $(i = 1, 2, 3, 5, 6)$ denote the hydrodynamic damping coefficients, ∇ denotes the volume of fluid displaced by the AUV, g denotes the acceleration of gravity, ρ denotes the fluid density, G denotes the gravity of the AUV, and M_L denotes the distance between the center of gravity and the center of buoyancy [25].

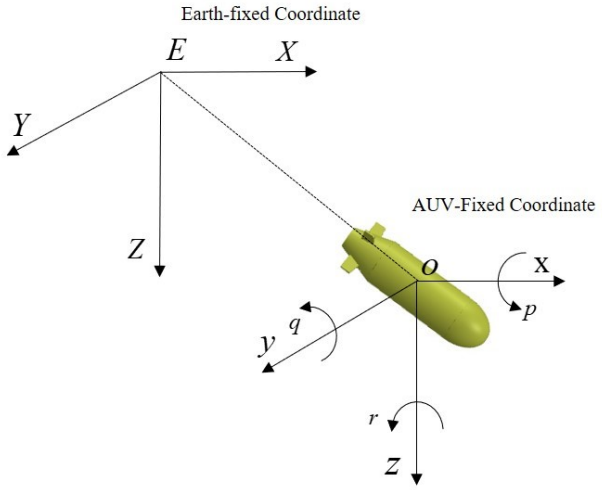


Figure 1. AUVs with body and earth-fixed reference frames.

3. Three-Dimensional Guidance System

The 3D guidance system is motivated by the 2D guidance law. This paper simplifies a 3D guidance system into two independent and decoupled 3-DOF cases; namely, the horizontal-plane guidance and the vertical-plane guidance. The horizontal-plane guidance strategy makes AUVs converge to a straight line on the horizontal plane via generating the reference heading angle; similarly, the vertical-plane guidance strategy generates the reference pitch angle to converge to a straight line on the vertical plane. The block diagram of the 3D guidance system is given in Figure 2.

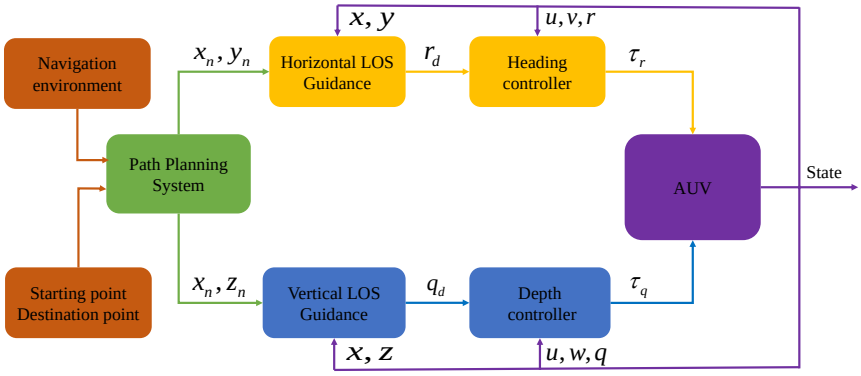


Figure 2. The block diagram of the 3D guidance system.

To obtain the 3D guidance algorithm, the original 2D guidance algorithm is extended. The inertial frame is transformed into the path frame by two rotations, and the tracking error vector is built to clearly represent the distance from the AUV to the path.

The first step is to rotate the inertial frame by α around the z axis, and the second step is to rotate the resulting frame from the first rotation by β around the y axis, where α and β, defined by (3) and (4), are the slope of the trajectory straight line before each rotation. The rotations can be described by two rotation matrices, $\boldsymbol{R}_z(\alpha)$ and $\boldsymbol{R}_y(\beta)$, given in (5) and (6), respectively.

$$\alpha = \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right), \tag{3}$$

$$\beta = \arctan\left(\frac{z_i - z_{i+1}}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}\right), \tag{4}$$

$$\boldsymbol{R}_z(\alpha) = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

$$\boldsymbol{R}_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{pmatrix}. \tag{6}$$

Then, the tracking error vector $\boldsymbol{e} = [e_x, e_y, e_z]^T$ can be computed in the path frame as (7), where e_x, e_y and e_z are the along, cross and vertical tracking errors, respectively.

$$\boldsymbol{e} = \boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\alpha)(\boldsymbol{p} - \boldsymbol{W}_i). \tag{7}$$

In order to make the AUV follow the desired path, the lookahead guidance algorithm is employed by introducing a virtual target [26]. In this algorithm, the error components e_y and e_z are used to guide the AUV to the virtual target. To guarantee that the tracking errors e_y and e_z converge to zero, the guidance angles ψ_d and θ_d are set as

$$\psi_d = \alpha + \arctan\left(\frac{-e_y}{\delta_y}\right), \tag{8}$$

$$\theta_d = \theta + \arctan\left(\frac{e_z}{\delta_z}\right). \tag{9}$$

where δ_y and δ_z are designed to force the position of the AUV to the xz and xy planes of the path frame, respectively. Then, we can obtain the desired angular velocities q_d and r_d by proportional controller gains k_ψ and k_θ.

The steps of the lookahead guidance algorithm for AUVs are listed in Algorithm 1.

Algorithm 1 Lookahead guidance algorithm.

Input: $(\boldsymbol{W}_i = [x_i, y_i, z_i]^T, \boldsymbol{W}_{i+1} = [x_{i+1}, y_{i+1}, z_{i+1}]^T, \boldsymbol{p} = [x, y, z]^t, \psi, \theta, \delta_y, \delta_z, k_\psi, k_\theta, i = (1, 2 \cdots, n))$

Output: (r_d, q_d)

for $i = 1$ to $n - 1$ **do**
- $\alpha \leftarrow \arctan(\frac{y_{i+1} - y_i}{x_{i+1} - x_i})$
- $\boldsymbol{W}_i^R = \boldsymbol{R}_z(\alpha)\boldsymbol{W}_i$
- $\triangleright \boldsymbol{W}_i^R = [x_i^R, y_i^R, z_i^R]^T$
- $\boldsymbol{W}_{i+1}^R \leftarrow \boldsymbol{R}_z(\alpha)\boldsymbol{W}_{i+1}$
- $\beta \leftarrow \arctan(\frac{z_i^R - z_{i+1}^R}{x_{i+1}^R - x_i^R})$
- $\triangleright \boldsymbol{e} = [e_x, e_y, e_z]^T$
- $\boldsymbol{e} = \boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\alpha)(\boldsymbol{p} - \boldsymbol{W}_i)$
- $\psi_d = \alpha + \arctan(\frac{-e_y}{\delta_y})$
- $\theta_d = \theta + \arctan(\frac{e_z}{\delta_z})$
- $r_d = k_\psi(\psi_d - \psi)$
- $q_d = k_\theta(\theta_d - \theta)$
- **return** (r_d, q_d)

This paper designs two independent controllers for the underactuated AUV based on the desired surge velocity u_d, the desired pitch angle rate q_d, and the desired heading angle rate r_d to control the horizontal-plane and vertical-plane motions, respectively [27].

Define the surge velocity error z_1, the pitch angle rate error z_2 and the heading angle rate error z_3 as:

$$\begin{aligned} z_1 &:= u - u_d, \\ z_2 &:= q - q_d, \\ z_3 &:= r - r_d. \end{aligned} \tag{10}$$

Step 1: Differentiating (10) with respect to time, respectively, and we have

$$\begin{aligned} m_{11}\dot{z}_1 &= m_{11}\dot{u} - m_{11}\dot{u_d}, \\ m_{55}\dot{z}_2 &= m_{55}\dot{q} - m_{55}\dot{q}_d, \\ m_{66}\dot{z}_3 &= m_{66}\dot{r} - m_{66}\dot{r}_d. \end{aligned} \tag{11}$$

Step 2: The control law can be designed as:

$$\begin{aligned} \tau_u &= -k_1 z_1 - m_{22}vr + m_{33}wq + d_{11}u + m_{11}\dot{u_d}, \\ \tau_q &= -k_2 z_2 - (m_{33} - m_{11})uw + d_{55}q + (\rho g \bigtriangledown GM_L \sin(\theta)) + m_{55}\dot{q}_d, \\ \tau_r &= -k_3 z_3 - (m_{11} - m_{22})uv + d_{66}r + m_{66}\dot{r}_d, \end{aligned} \tag{12}$$

where $k_1 > 0$, $k_2 > 0$ and $k_3 > 0$ are the parameters to be designed.

Step 3: Define a Lyapunov function candidate as:

$$V(z_1, z_2, z_3) = \frac{1}{2}m_{11}{z_1}^2 + \frac{1}{2}m_{55}{z_2}^2 + \frac{1}{2}m_{66}{z_3}^2. \tag{13}$$

Differentiating the Lyapunov function with respect to time, and we have

$$\dot{V}(z_1, z_2, z_3) = z_1 m_{11}\dot{z}_1 + z_2 m_{55}\dot{z}_2 + z_3 m_{66}\dot{z}_3. \tag{14}$$

Substituting Equations (2), (11) and (12) into (15), and we have

$$\begin{aligned} \dot{V}(z_1, z_2, z_3) &= z_1 m_{11}\dot{z}_1 + z_2 m_{55}\dot{z}_2 + z_3 m_{66}\dot{z}_3 \\ &= z_1 m_{55}(\dot{u} - \dot{u_d}) + z_2 m_{55}(\dot{q} - \dot{q}_d) + z_3 m_{66}(\dot{r} - \dot{r}_d) \\ &= -k_1{z_1}^2 - k_2{z_2}^2 - k_3{z_3}^2 \leq 0. \end{aligned} \tag{15}$$

According to the Lyapunov stability theorem, the errors z_1, z_2 and z_3 can converge to zero, and the closed-loop system is asymptotically stable.

4. Multi-Objective Optimization

If $\boldsymbol{P}_0$ is the starting point and $\boldsymbol{P}_N$ is the destination point, then the path can be expressed as follows:

$$\boldsymbol{W} = (\boldsymbol{P}_0, x_1, y_1, z_1, \ldots, x_n, y_n, z_n, \boldsymbol{P}_N), \tag{16}$$

where $\boldsymbol{P}_n = (x_n, y_n, z_n)$ is a middle waypoint of the path, both the line segment from $(x_{n-1}, y_{n-1}, z_{n-1})$ to (x_n, y_n, z_n) and the line segment from (x_n, y_n, z_n) to $(x_{n+1}, y_{n+1}, z_{n+1})$ do not cut obstacle areas.

According to the given navigation environment and the physical property of AUVs, some formal constraints are formulated as follows:

$$\begin{aligned} x_{min} < x_n < x_{max}, \quad n &= \overline{1, n}, \\ y_{min} < y_n < y_{max}, \quad n &= \overline{1, n}, \\ z_{min} < z_n < z_{max}, \quad n &= \overline{1, n}, \\ \gamma &< \gamma_{max}, \\ \varphi &> \varphi_{min}, \end{aligned} \tag{17}$$

where x_{min} and x_{max} are constraints for x_n, y_{min} and y_{max} are constraints for y_n, z_{min} and z_{max} are constraints for z_n, γ_{max} is the maximum allowed angle of the planning path on the horizontal plane, and φ_{min} is the minimal allowed angle of the planning path on the vertical plane.

4.1. Path Length

The total length of the path is usually discussed because it determines the sailing time and energy expenditure. The total length of the path can be calculated by:

$$L = \sum_{n=1}^{N} \|\boldsymbol{P}_n - \boldsymbol{P}_{n-1}\|, \tag{18}$$

where $\|\boldsymbol{P}_n - \boldsymbol{P}_{n-1}\|$ indicates the Euclidean distance between $\boldsymbol{P}_n = (x_n, y_n, z_n)$ and $\boldsymbol{P}_{n-1} = (x_{n-1}, y_{n-1}, z_{n-1})$.

During path planning, the path length ratio is usually used to calculate the path length, specifically as follows:

$$A_1 = \frac{L}{\|\boldsymbol{P}_N - \boldsymbol{P}_0\|}, \tag{19}$$

where $\|\boldsymbol{P}_N - \boldsymbol{P}_0\|$ represents the Euclidean distance between the starting point $\boldsymbol{P}_0$ and the destination point $\boldsymbol{P}_N$.

4.2. Path Smoothness

Path smoothness should be considered to avoid the mutations in navigation direction in path planning. Mutations in navigation direction will not only increase energy consumption, but also reduce navigation accuracy. Therefore, in order to avoid the mutations in navigation direction, the fitness function is defined as:

$$\begin{aligned} A_2 = &\sum_{n=1}^{N} \frac{\gamma_n}{min(\|(x_{n-1}, y_{n-1}), (x_n, y_n)\|, \|(x_n, y_n), (x_{n+1}, y_{n+1})\|)} \\ &+ \sum_{n=1}^{N} \frac{\varphi_n}{\sqrt{(z_n - z_{n-1})^2 + ((x_n - x_{n-1})^2)}}, \end{aligned} \tag{20}$$

where $\|\cdot\|$ represents the Euclidean distance, γ_n and φ_n represent the angle of the planning path on the horizontal plane and the angle of the planning path on the vertical plane, which are calculated according to the following formula:

$$\begin{aligned} \gamma_n &= \left| \arctan\left[\frac{z_n - z_{n-1}}{\sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}} \right] \right|, \\ \varphi_n &= \pi - \left| \arccos\left[\frac{(x_n - x_{n-1})(x_{n+1} - x_n) + (y_n - y_{n-1})(y_{n+1} - y_n)}{\sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2} + \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2}} \right] \right|. \end{aligned} \tag{21}$$

4.3. Path Safety

A safe path must avoid all the obstacles. The obstacles are assumed to be static with known position. The underwater obstacle is usually assumed to be a stationary target and positioned. The solutions should be penalized when the generated path cuts any obstacles. The safety fitness function of the path can be expressed as:

$$A_3 = \begin{cases} +\infty, & \text{if } \boldsymbol{P}_n\boldsymbol{P}_{n-1} \bigcap OB \neq \emptyset, \\ 0, & \text{if } \boldsymbol{P}_n\boldsymbol{P}_{n-1} \bigcap OB = \emptyset, \end{cases} \tag{22}$$

where A_3 is the fitness function of the path, OB represents the collection of all obstacle areas, if the planned path does not intersect all obstacles, the fitness function of the path is 0, conversely, it is $+\infty$.

4.4. The Construction of the Total Fitness Function

The path planning algorithm takes the path length, the path safety and the path smoothness as the main optimization goal. It is common to aggregate the multiple objectives into a single fitness function, which is nonzero and monotonic. The total fitness function of the path comes from a weighted sum of objectives above, and the smaller fitness function value represents a better path. Define the total fitness function as:

$$\begin{aligned} &F = \eta_1 A_1 + \eta_2 A_2 + \eta_3 A_3, \\ &\sum_{i=1} \eta_i = 1, \end{aligned} \tag{23}$$

where F represents the total fitness function of the path, η_i (i = 1, 2, 3) are weight factors that determine the importance of each objective.

5. The Improved SPSO-2011 Algorithm

Evolutionary algorithms have been widely applied to solve multi-objective optimization engineering problems. One of the most popular evolutionary algorithms is the particle swarm optimization (PSO) algorithm.

The SPSO-2011 exploits the idea of rotational invariance to improve the standard PSO. The improved SPSO-2011 algorithm still utilizes the update rules of the SPSO-2011 algorithm and uses the uniform random initialization. Velocity and position of the particle are updated by

$$\begin{aligned} \boldsymbol{V}_i(t+1) &= \omega \boldsymbol{V}_i(t) - \frac{c1+c2}{3}\boldsymbol{X}_i(t) + \frac{c1\boldsymbol{P}_{best_i}(t) + c2\boldsymbol{G}_{best}}{3}, \\ \boldsymbol{X}_i(t+1) &= \boldsymbol{X}_i(t) + \boldsymbol{V}_i(t+1), \end{aligned} \tag{24}$$

where $i = 1, 2, \ldots, M$, with M equal to swarm size and $t = 1, 2, \ldots, T$ with T equal to the maximum number of iterations, $\boldsymbol{V}$ represents the velocity of the particle, $\boldsymbol{X}$ the position of the particle, ω is the inertia weight, $c1$ is the cognitive learning factor, $c2$ is the social learning factor, $\boldsymbol{G}_{best}$ denotes the best position that the whole swarm has found so far, and $\boldsymbol{P}_{best}$ denotes the best position that the particle has found so far.

5.1. The Improved Adaptive Parameters

The inertia weight ω is applied to balance the global and local search capabilities. The larger the w, the faster the particle convergence rate and the stronger the global search ability, but it reduces the local search ability of the particles and ultimately leads to lower global convergence accuracy. The smaller w will improve the convergence accuracy of the particles, but reduces the convergence speed of the particles, which easily makes the particle into the local optimum in the search process and cannot jump out of the local extreme value [28]. In the SPSO-2011 algorithm, the inertia weight is a fixed value, but it cannot consider both the search accuracy and the search range. A nonlinear adaptive strategy can ensure that all particles can quickly spread into the whole search space in the early search stage to determine the approximate range of the global extrema, accelerating the particle convergence in the late search stage [29]. In the improved SPSO-2011 algorithm, the inertia weight w linearly increases in the early stage, which increases the search range, prevents falling into the local optimum, and then the inertia weight w nonlinearly reduces in the late stage to improve the search accuracy. The nonlinear adaptive inertia weight was proposed as the followed formula:

$$\begin{aligned} &\omega = \omega_r + (\frac{t}{T}) * (\omega_{max} - \omega_{min}), 0 < t < \frac{T - iter_{min}}{2}, \\ &\omega = \omega_{min} + e^{\frac{-(k*t)^2}{T}} * (\omega_{max} - \omega_{min}), \frac{T - iter_{min}}{2} < t < T, \\ &\omega = \omega_{max}, \omega > \omega_{max}, \end{aligned} \tag{25}$$

where ω_r represents the initial value, ω_{min} is the minimum value, ω_{max} is the maximum value, and $iter_{min}$ represents the minimum number of iterations.

In the SPSO-2011 algorithm, the update speed of the particle swarm is mainly adjusted through the cognitive learning factor and the social learning factor, but due to the lack of changes in learning factors, it is easy to make the particle fall into the local optimum. In the improved SPSO-2011 algorithm, two learning factors are devised with dynamic adjustment methods to avoid the local optimum. The dynamic adjustment methods are as follows:

$$\begin{aligned} c1 &= c1_{max} - K_1 * \frac{t}{T} * (c1_{max} - c1_{min}), \\ c2 &= c2_{min} + K_2 * \frac{t}{T} * (c2_{max} - c2_{min}), \end{aligned} \tag{26}$$

where $c1_{min}$ represents the minimum value of the cognitive learning factor, $c2_{min}$ represents the minimum value of the social learning factor, $c1_{max}$ represents the maximum value of the cognitive learning factor, $c2_{max}$ represents the maximum value of the social learning factor, and K_1 and K_2 represent two adjustable and positive constants.

5.2. Mutation Operator

The mutation is employed to give new information to the population and increase population diversity. In this paper, the improved SPSO-2011 algorithm introduces a mutation operator with a threshold. In the improved SPSO-2011 algorithm, when the cumulative number of iterations is greater than the threshold and the individual fitness function value is always greater than the global fitness function in the cumulative process, the mutation is performed. Otherwise, the cumulative number of iterations is 0. The improved method can effectively increase global search ability and avoid falling into the local optimum. The random mutation can be expressed by the following equation:

$$X_i(t+1) = rand * D, \tag{27}$$

where $rand$ is the random number in the $[0, 1]$ and D is the range of map.

The flow chart of the improved SPSO-2011 algorithm is shown in Figure 3.

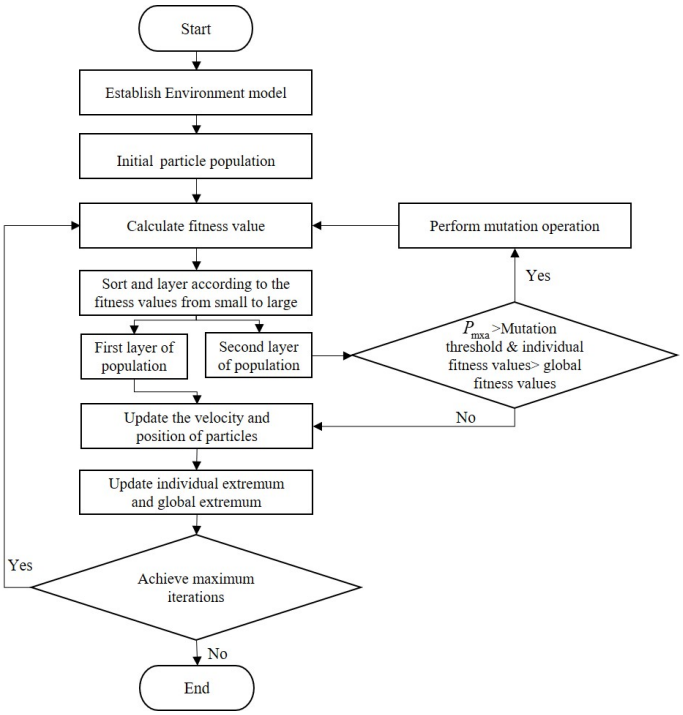


Figure 3. The improved SPSO-2011 algorithm flow chart.

6. Simulation Results

To verify the effectiveness of the proposed algorithm, this paper contrasts with the PSO and SPSO-2011 algorithms in terms of planning time and path length. For certain AUV, a 3D guidance and control strategy is designed, and a closed-loop simulation system is established based on MATLAB/Simulink. Comparative simulations are carried out to verify the effectiveness of the proposed algorithm.

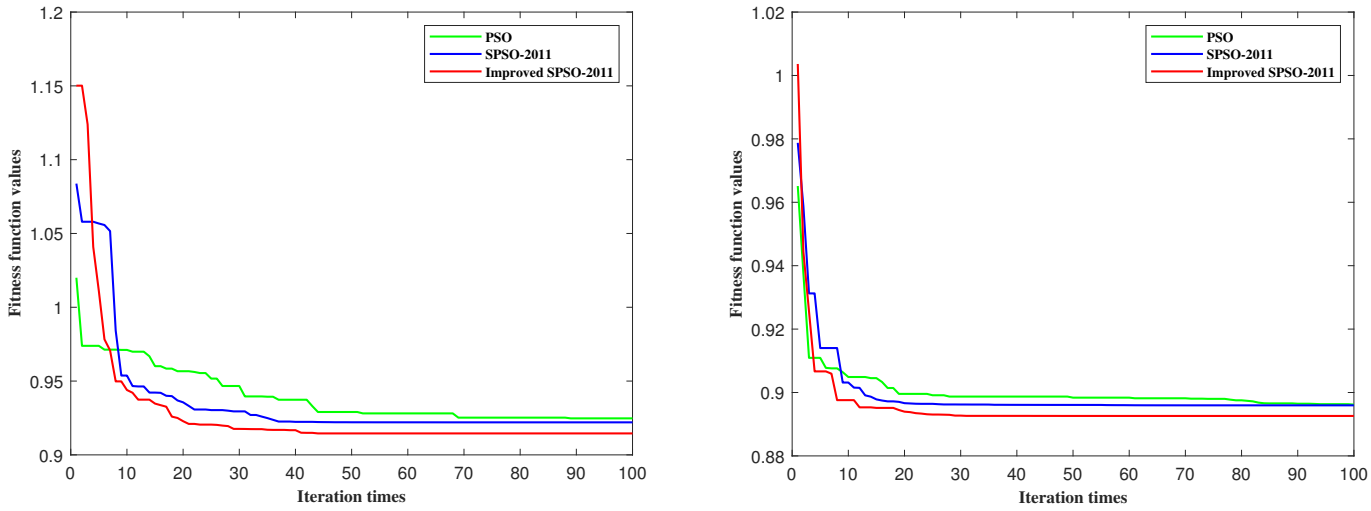
The simulations are carried out in different 3D spaces with different obstacles and different starting and destination points. In general, the underwater obstacles mainly include sea ridges, underwater shipwrecks, marine suspended matter and other objects. To facilitate the problem description, the sphere and cuboids are used to describe the underwater obstacles. The parameters of the proposed Improved SPSO-2011 algorithm are given in Table 1. The simulations are carried out on a 500 m $\times$ 500 m $\times$ 100 m 3D space with some equivalent obstacles. In 3D space, the starting point is $(2, 10, 7)$, and the destination point is $(480, 430, 30)$. All obstacles are replaced by regular balls or cuboids. The main parameters of the improved SPSO-2011 algorithm are: $\gamma_{max} < |\pm \pi/3|$, $\varphi_{min} > |\pm \pi/6|$, $M = 200$, $\omega_r = 0.7$, $\omega_{min} = 0.4$, $\omega_{max} = 0.9$, $c1_{min} = c2_{min} = 0.9$, $c1_{max} = c2_{max} = 2.2$, $K_1 = K_2 = 0.2$, $iter_{min} = 70$, $T = 100$, $k = 1.5$ and $P_{max} = 7$. Model parameters of the AUV are given by $m_{11} = 25$ kg, $m_{22} = 17.5$ kg, $m_{33} = 30$ kg, $m_{55} = 22.5\,\text{kgm}^2$, $m_{66} = 15\,\text{kgm}^2$, $d_{11} = 30\,\text{kgs}^{-1}$, $d_{22} = 30\,\text{kgs}^{-1}$, $d_{33} = 30\,\text{kgs}^{-1}$, $d_{55} = 25\,\text{kgm}^2\text{s}^{-1}$, $d_{66} = 25\,\text{kgm}^2\text{s}^{-1}$, and $\rho g \nabla GM_L = 5$ [30,31]. The initial states of the AUV are set as $[x(0), y(0), z(0), \theta(0), \psi(0)] = [2(m), 10(m), 7(m), 0(rad), 0(rad)]$. The constant surge velocity of the AUV is set as $u_d = 1.5\ \text{ms}^{-1}$. The 3D guidance algorithm and controller parameters are chosen as $k_1 = 1$, $k_2 = 1$, $k_3 = 1$, $k_\psi = 1$, $k_\theta = 1$, $\delta_y = 10$ and $\delta_z = 10$.

Table 1. The parameters of the improved SPSO-2011 algorithm.

Variables	Value
γ_{max}	$\pi/3$
φ_{min}	$\pi/6$
ω_r	0.7
ω_{min}	0.4
ω_{max}	0.9
$iter_{min}$	70
$c1_{min}$	0.9
$c2_{min}$	0.9
$c1_{max}$	2.1
$c2_{max}$	2.1
P_{max}	7
K_1	0.2
K_2	0.2
M	200
T	100
k	1.5

The parameters of 3D space and the positions of starting and destination points are shown in Table 2. Simulations of the PSO algorithm, the SPSO-2011 algorithm and the improved SPSO-2011 algorithm are performed using MATLAB, and each three path planning algorithms were performed eight times. The optimal fitness function values of three path planning algorithms in different 3D spaces with different obstacles and different starting and destination points are shown in Figure 4. It can be clearly observed that the proposed improved SPSO-2011 algorithm has a faster convergence rate and the minimum fitness function values 0.9145 and 0.8926 for 3D space 1 and 2, respectively. This confirms that the proposed improved SPSO-2011 algorithm has better performance than the PSO and SPSO-2011 algorithms. The optimal planned paths of three path planning algorithms are shown in Figure 5. The shortest distance of the paths, and the average distance of the paths of three path planning algorithms, are shown in Table 2. As seen in Figure 5 and Table 2,

under the proposed improved SPSO-2011 algorithm, the planned path has the shorter path length and higher path smoothness in 3D spaces 1 and 2. From the values of the average distance of the paths in Table 2, the proposed algorithm has the shortest average distance of the paths, and thus can effectively avoid falling into the local optimum.

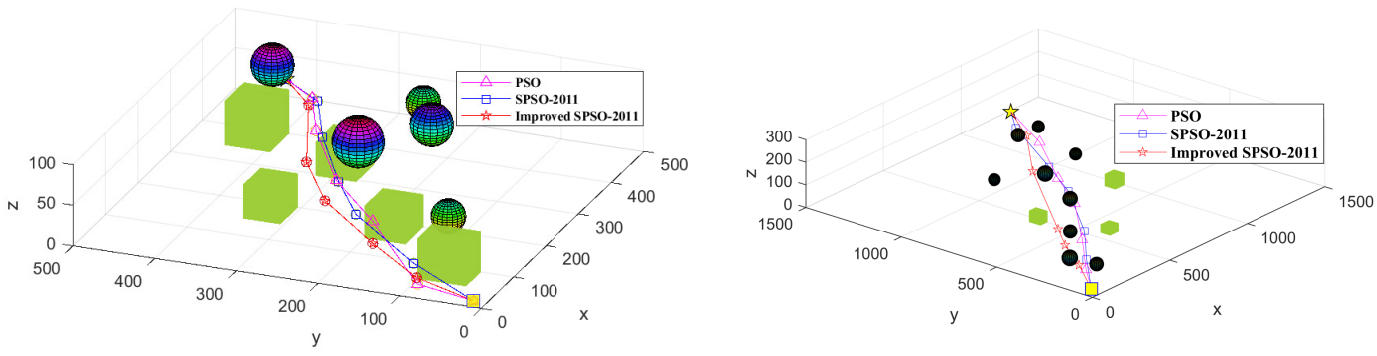


(**a**) 3D space 1 (**b**) 3D space 2

Figure 4. The fitness function curves of three path planning algorithms in different 3D spaces.

Table 2. Performance comparisons of path planning algorithms in different 3D spaces.

Algorithm	3D Spaces	The Shortest Distance of Paths (Meters)	The Average Distance of Paths (Meters)
PSO	3D space1: 500 m × 500 m × 100 m	663.54	665.98
SPSO-2011	The starting point: $(2, 10, 7)$	660.10	669.99
Improved SPSO-2011	The destination point: $(480, 430, 30)$	655.12	657.26
PSO	3D space2: 1500 m × 1500 m × 300 m	1828.35	1834.77
SPSO-2011	The starting point: $(18, 20, 25)$	1829.83	1836.91
Improved SPSO-2011	The destination point: $(1200, 1400, 60)$	1820.89	1827.71



(**a**) 3D space 1 (**b**) 3D space 2

Figure 5. Compared with the optimal path of three path planning algorithms.

To verify the path performability, the planned path in the 3D space 1 is selected for tracking control by employing the proposed 3D guidance and control algorithm. The simulation results are illustrated in Figures 6 and 7. The 3D following trajectory of the AUV is illustrated in Figure 6a. The corresponding following trajectories in the xy, xz and yz planes are illustrated in Figure 6b–d, respectively. Figure 7a shows the desired heading angle and actual heading angle. Figure 7b shows the desired pitch angle and actual pitch angle. From Figures 6 and 7, we can observe that the proposed 3D guidance and control algorithm can make the AUV track the planned path accurately and the desired heading and pitch angles are reasonable.

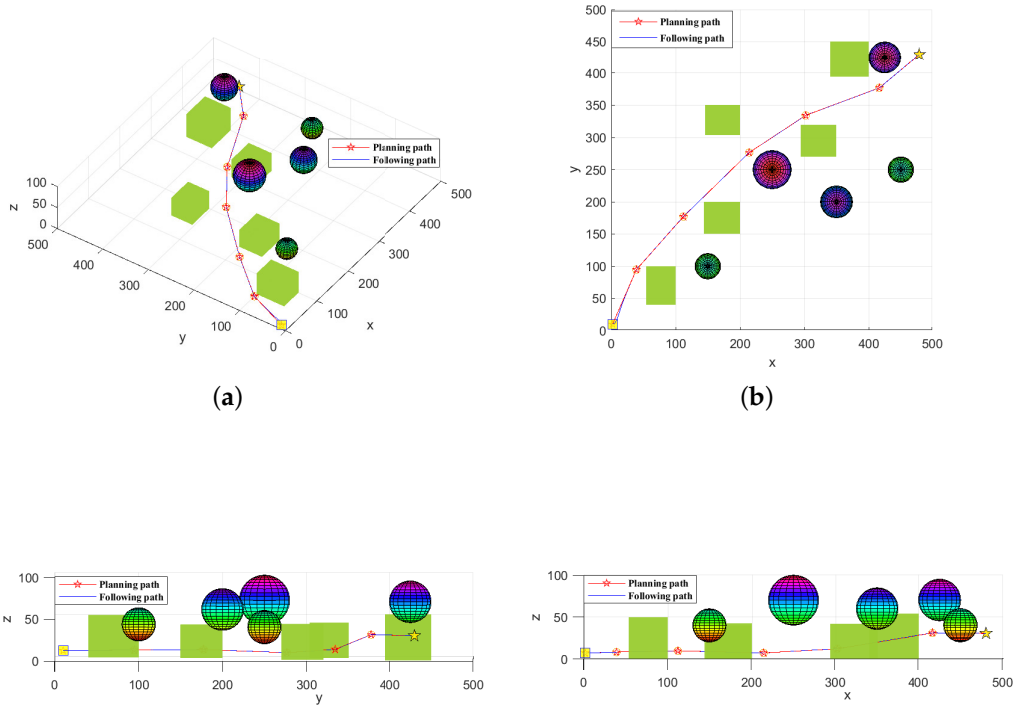


Figure 6. (**a**) The following optimal path in the 3D space 1. (**b**) The following optimal path in the xy plane. (**c**) The following optimal path in the yz plane. (**d**) The following optimal path in the xz plane.

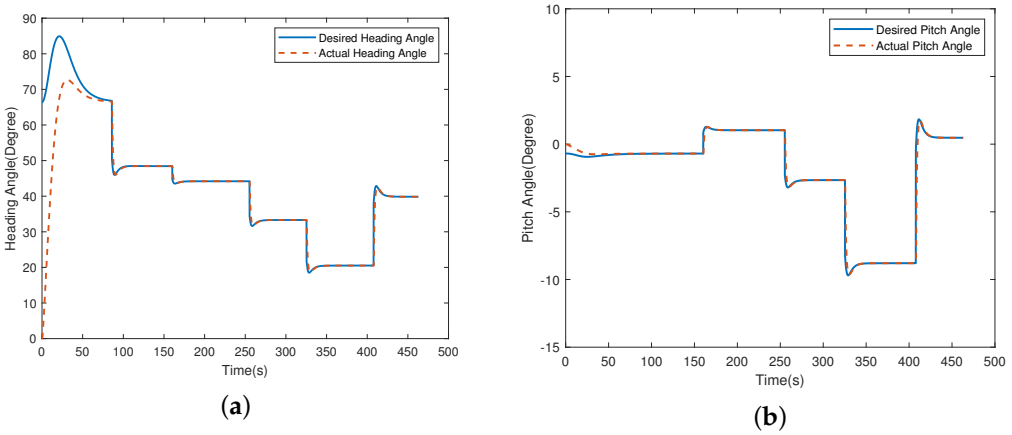


Figure 7. Tracking results of guidance angles. (**a**) Desired heading angle and actual heading angle. (**b**) Desired pitch angle and actual pitch angle.

Simulation results demonstrate the effectiveness of the proposed algorithm. Compared with the PSO algorithm and SPSO-2011 algorithm, the proposed algorithm has the following advantages:

- Under the proposed algorithm, the phenomenon of local optima can be avoided effectively and thus plan shorter path lengths.
- The proposed algorithm has the faster convergence rate and higher accuracy.
- The proposed 3D guidance algorithm can accurately track the planned path, indicating that the planned path has good performability.

7. Conclusions

This paper proposes an improved SPSO-2011 algorithm. A mutation operator with a threshold is introduced to prevent falling into local optima and a nonlinear adaptive parameter strategy is introduced to accelerate convergence speed. The construction of the total fitness function considers "path length", "path smoothness", "path safety" and "physical constraints", which can effectively avoid collision with threats, obtain a shorter path, and save energy consumption. Compared with the PSO algorithm and the SPSO-2011 algorithm, simulation results show that the proposed algorithm has greater advantages in terms of path length and planning time. To verify the performability of the planned path, a 3D guidance algorithm and controller are designed for AUVs. Simulation results show that the AUV can accurately track the planned path based on the improved SPSO-2011 algorithm.

Author Contributions: Conceptualization, L.W.; methodology, B.Z.; software, B.Z.; validation, S.A. ; resources, Y.H. and L.W.; writing—original draft preparation, B.Z.; writing—review and editing, S.A. and B.Z.; supervision, L.W.; project administration, Y.H.; funding acquisition, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: Authors wish to thank the funding from the Taishan Scholar Project of Shandong Province (ts20190937) and National Natural Science Foundation of China (52176076).

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable

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

References

1. Konoplin, A.Y.; Konoplin, N.Y. System for automatic soil sampling by underwater vehicle. In Proceedings of the 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), St. Petersburg, Russia, 16–19 May 2017.
2. Xu, Y.R.; Xiao, K. Technology development of autonomous ocean vehicle. *Acta Autom. Sin.* **2018**, *33*, 518.
3. Villar, S.A.; Acosta, G.G.; Sousa, A.L.; Rozenfeld, A. Evaluation of an efficient approach for target tracking from acoustic imagery for the perception system of an autonomous underwater vehicle. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 24. [CrossRef]
4. Gao, Z.Y.; Guo, G. Fixed-time leader-follower formation control of autonomous underwater vehicles with event-triggered intermittent communications. *IEEE Access* **2018**, *3*, 27902–27911. [CrossRef]
5. Wynn, R.B.; Huvenne, V.A.; Le Bas, T.P.; Murton, B.J.; Connelly, D.P.; Bett, B.J.; Ruhl, H.A.; Morris, K.J.; Peakall, J.; Parsons, D.R.; et al. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Mar. Geol.* **2014**, *352*, 451–468. [CrossRef]
6. Fossen, T.I.; Breivik, M.; Skjetne, R. Line-of-sight path following of underactuated marine craft. *IFAC Proc. Vol.* **2003**, *36*, 211–216. [CrossRef]
7. Sahoo, A.; Dwivedy, S.K.; Robi, P.S. Advancements in the field of autonomous underwater vehicle. *Ocean. Eng.* **2020**, *181*, 145–160. [CrossRef]
8. Naeem, W.; Sutton, R.; Ahmad, S.M.; Burns, R.S. A review of guidance laws applicable to unmanned underwater vehicles. *J. Navig.* **2003**, *56*, 15–29. [CrossRef]
9. Breivik, M.; Fossen, T.I. Guidance laws for autonomous underwater vehicles. *Underw. Veh.* **2009**, *4*, 51–76.
10. Zhao, Y.J.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [CrossRef]
11. Khachiyan, L.; Gurvich, V.; Zhao, J.H. *Extending Dijkstra's Algorithm to Maximize the Shortest Path by Node-Wise Limited Arc Interdiction*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 221–234.
12. Ammar, A.; Bennaceur, H.; Châari, I.; Koubâa, A.; Alajlan, M. Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments. *Soft Comput.* **2016**, *20*, 4149–4171. [CrossRef]
13. Fan, X.J.; Guo, Y.J.; Liu, H.; Wei, B.; Lyu, W. Improved artificial potential field method applied for AUV path planning. *Math. Probl. Eng.* **2020**, *2020*, 6523158. [CrossRef]

14. Sun, Y.; Zhang, R.B. Research on global path planning for AUV based on GA. In *Mechanical Engineering and Technology*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 311–318.
15. Cai, L.; Jia, J.P. Path planning with PSO for autonomous vehicle. *Adv. Eng. Mater. Appl. Mech.* **2016**, 263–266.
16. Das, P.K.; Behera, H.S.; Das, S.; Tripathy, H.K.; Panigrahi, B.K.; Pradhan, S.K. A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment. *Neurocomputing* **2016**, *207*, 735–753. [CrossRef]
17. Lin, C.; Wang, H.; Yuan, J.; Yu, D.; Li, C. An improved recurrent neural network for unmanned underwater vehicle online obstacle avoidance. *Ocean. Eng.* **2019**, *189*, 106327. [CrossRef]
18. Wang, Z.; Zhang, S.W.; Feng, X.N.; Sui, Y.C. Autonomous underwater vehicle path planning based on actor-multi-critic reinforcement learning. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2021**, *235*, 1787–1796. [CrossRef]
19. Cheng, C.; Zhu, D.; Sun, B.; Chu, Z.; Nie, J.; Zhang, S. Path planning for autonomous underwater vehicle based on artificial potential field and velocity synthesis. In Proceedings of the IEEE/2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 3–6 May 2015.
20. Lim, H.S.; Fan, S.; Chin, C.K.; Chai, S.; Bose, N. Particle swarm optimization algorithms with selective differential evolution for AUV path planning. *Int. J. Robot. Autom.* **2020**, *9*, 94–112. [CrossRef]
21. Yang, L.; Li, K.S.; Zhang, W.S.; Wang, Y.; Chen, Y.; Zheng, L.F. Three-dimensional Path Planning for Underwater Vehicles Based on an Improved Ant Colony Optimization Algorithm. *J. Eng. Sci. Technol. Rev.* **2015**, *8*, 24–33. [CrossRef]
22. Yan, S.K.; Pan, F. Research on route planning of AUV based on genetic algorithms. In Proceedings of the 2019 IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI), Xi'an, China, 22–24 November 2019.
23. Cheng, C.X.; Sha, Q.X.; He, B.; Li, G. Path planning and obstacle avoidance for AUV: A review. *Ocean. Eng.* **2021** *235*, 109355. [CrossRef]
24. Do, K.D. Coordination control of underactuated ODINs in three-dimensional space. *Robot. Auton. Syst.* **2013** *61*, 853–867. [CrossRef]
25. Pettersen, K.Y.; Egeland, O. Time-varying exponential stabilization of the position and attitude of an underactuated autonomous underwater vehicle. *IEEE Trans. Autom. Control* **1999** *44*, 112–115. [CrossRef]
26. Breivik, M.; Fossen, T.I. Principles of guidance-based path following in 2D and 3D. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 12–15 December 2005.
27. Lekkas, A.M. *Guidance and Path-Planning Systems for Autonomous Vehicles*; Norwegian University of Science and Technology: Trondheim, Norway, 2014.
28. Xu, S.H; Li,X.X. An adaptive changed inertia weight particle swarm algorithm. *Sci. Technol. Eng.* **2012**, *9*, 1671–1815.
29. Cheng, Z.; Dong, M.N.; Yang, T.K.; Han, L.J. Extraction of solar cell model parameters based on self-adaptive chaos particle swarm optimization algorithm. *Trans. China Electrotech. Soc.* **2014** *29*, 245–252.
30. Do, K.D.; Pan, J. *Control of Ships and Underwater Vehicles: Design for Underactuated and Nonlinear Marine Systems*; Springer: London, UK, 2009.
31. Fossen, T.I. Marine Control Systems–Guidance. Navigation, and Control of Ships, Rigs and Underwater Vehicles. Marine Cybernetics, Trondheim, Norway, Org. Number NO 985 195 005 MVA, ISBN: 82 92356 00 2. 2002. Available online: www.marinecybernetics.com (accessed on 8 November 2002).