

Article

Hybrid Layer of Improved Interfered Fluid Dynamic System and Nonlinear Model Predictive Control for Navigation and Control of Autonomous Underwater Vehicles

Jiqing Du , Dan Zhou and Sachiyo Arai *

Graduate School of Science and Engineering, Chiba University, Chiba 263-8522, Japan;
21wd1304@student.gs.chiba-u.jp (J.D.); 21wd1303@student.gs.chiba-u.jp (D.Z.)

* Correspondence: sachiyo@faculty.chiba-u.jp; Tel.: +81-43-290-3316

Abstract: This study introduces a hybrid control structure called Improved Interfered Fluid Dynamic System Nonlinear Model Predictive Control (IIFDS-NMPC) for the path planning and trajectory tracking of autonomous underwater vehicles (AUVs). The system consists of two layers; the upper layer utilizes the Improved Interfered Fluid Dynamic System (IIFDS) for path planning, while the lower layer employs Nonlinear Model Predictive Control (NMPC) for trajectory tracking. Extensive simulation experiments are conducted to determine optimal parameters for both static and dynamic obstacle scenarios. Additionally, real-world testing is performed using the BlueRov2 platform, incorporating multiple dynamic and static obstacles. The proposed approach achieves real-time control at a frequency of 100 Hz and exhibits impressive path tracking accuracy, with a root mean square (RMS) of 0.02 m. This research provides a valuable framework for navigation and control in practical applications.

Keywords: autonomous underwater vehicles; hybrid layer; IIFDS; NMPC; navigation and control



Citation: Du, J.; Zhou, D.; Arai, S. Hybrid Layer of Improved Interfered Fluid Dynamic System and Nonlinear Model Predictive Control for Navigation and Control of Autonomous Underwater Vehicles. *J. Mar. Sci. Eng.* **2023**, *11*, 2014. <https://doi.org/10.3390/jmse11102014>

Academic Editor: Abdellatif Ouahsine

Received: 15 September 2023

Revised: 14 October 2023

Accepted: 17 October 2023

Published: 19 October 2023



Copyright: © 2023 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

Autonomous underwater vehicles (AUVs) are increasingly utilized in various applications for ocean exploration. Path planning and trajectory tracking are fundamental requirements [1], and the smoothness, efficiency, and tracking deviation of the planned paths play crucial roles in these tasks [2]. Achieving real-time planning and control in complex environments is particularly important for efficient and safe AUV navigation [3]. Although both path planning and trajectory tracking tasks utilize the AUV motion model [4], they are treated as separate tasks within the general guide navigation control (GNC) framework [5].

Different path planning methods are employed for known and unknown obstacle environments [6]. In known static obstacle environments, algorithms like A* [7], Dijkstra [8], and RRT [9] are utilized to generate optimal paths using predetermined obstacle information. Dynamic obstacle environments, where obstacles move and change, present a more complex challenge. Randomized algorithms like PRM [10] and improved RRT [11], as well as SLAM [12] algorithms based on sensor data, are used. However, these methods often lack real-time capabilities.

Real-time path planning and tracking are essential for adapting to dynamic obstacles. To address real-time concerns, Jian, F.W. et al. introduced IIFDS [13], an APF-based method suitable for unknown dynamic environments. Extensive validation has been performed on various platforms, including unmanned boats and aerial vehicles [14,15].

AUV path tracking control has been widely researched. PID control is used in non-critical scenarios [16]. Linear quadratic regulator (LQR) control [17], while useful, may suffer from modeling inaccuracies. Nonlinear sliding mode control (SMC) can handle disturbances but may cause high-frequency oscillations [18].

Model Predictive Control (MPC) offers a trade-off between path tracking error and control constraints. Nonlinear MPC (NMPC) leverages dynamic model characteristics, making it suitable for path tracking control [19,20]. The computational time of the solver algorithm is predominantly consumed by matrix operations and constraints, factors that dictate parsing speed and variable precision. Sequential Quadratic Programming (SQP) is employed to address quadratic linear optimization problems concurrently featuring equality and inequality constraints, with the objective of identifying a locally optimal solution that satisfies the imposed constraints [21]. In the Baidu Apollo autonomous vehicle project, the SQP algorithm has been successfully applied [22].

Based on the extensive literature review presented, it becomes evident that the IIFDS offers the advantage of rapid computational capabilities, whereas NMPC excels in maintaining an optimal balance between tracking error and control effort. Building upon these insights, this paper introduces a composite control strategy that harnesses the strengths of both approaches, hereby referred to as the IIFDS-NMPC control structure (Figure 1). To validate the real-time control requirements, smooth trajectory tracking, and optimal performance in complex environments, we conducted a series of comprehensive simulations and experiments using this hybrid layer structure, encompassing scenarios with both static and dynamic obstacles.

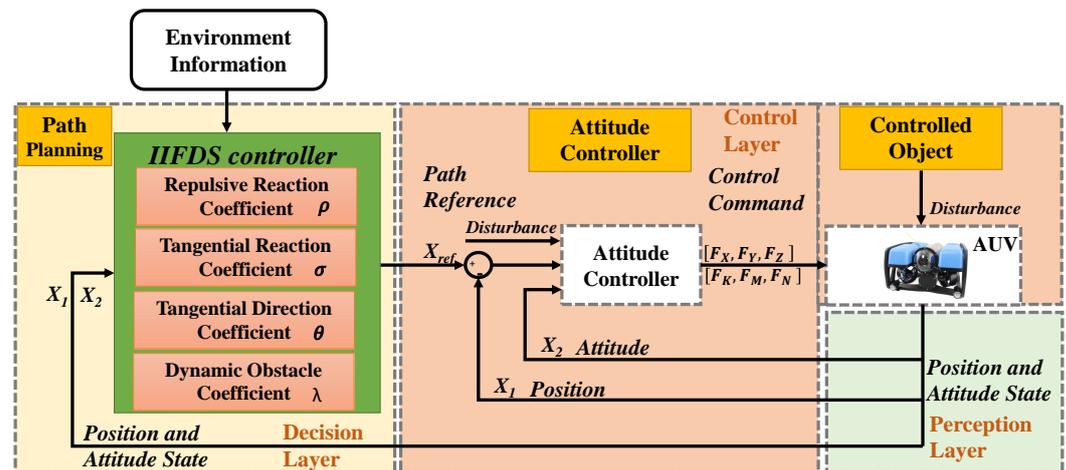


Figure 1. Schematic structure of the proposed hybrid layer controller.

Our contributions can be summarized as follows:

1. Real-Time Hybrid Control Structure: We propose a real-time hybrid control structure for path planning and trajectory tracking (depicted in Figure 1). In this architecture, the upper layer leverages the real-time IIFDS controller, while the bottom layer employs the NMPC controller. The computation time for each step remains consistently under 0.01 s, and the RMS tracking deviation is limited to a remarkable 0.02 m.
2. Detailed Design Process: We provide a comprehensive explanation of the complete design process. By following the steps outlined here, the proposed structure can be readily implemented across various platform systems, extending its applicability beyond just AUV systems.
3. Real-Time Feasibility: Extensive simulations and experiments validate the real-time feasibility of our approach. Testing was conducted in environments featuring both static and dynamic obstacles, with the designed path planning controller compared against A* methods. In static obstacle simulations, our method showcased a planning time reduction of 300 times compared to the A* algorithm. In dynamic obstacle scenarios, planning time was reduced by a factor of three. During real platform testing, our approach achieved a remarkable step-by-step tracking and control time of 0.0067 s, with an RMS tracking deviation as low as 0.01867 m.

- Physical Experimentation: To the best of our knowledge, this approach is employed for the first time in physical experiments with AUVs. The STM32 interface board acquires obstacle and environmental information from the ground station, while the main control NVIDIA-X2 core is utilized for path planning and the computation of optimal control quantities under various constraints.

The subsequent sections of this paper are organized as follows: Section 2 presents the kinematic and dynamic models of the AUV system along with the mathematical representations of obstacles. Section 3 delves into the methodology and algorithmic framework of our proposed hybrid layer structure. The simulation setup and experimental evaluations are detailed in Section 4, while Section 5 offers a comprehensive discussion of the results and performance analysis. Finally, in Section 6, we conclude the paper by summarizing the findings and outlining potential avenues for future research.

2. AUV Systems and Obstacle Scenario Modeling

To design the hybrid controller for the AUV known as BlueRov2, which is manufactured by BlueROV Robotics, located in Victoria, BC, Canada [23], as illustrated in Figure 2, it is essential to develop a comprehensive mathematical model that encompasses both the AUV itself and the obstacle scenarios it may encounter. This section aims to provide an overview of the AUV model, covering its kinematic and dynamic aspects. The kinematic model establishes the relationship between the AUV’s position, velocity, and acceleration during its motion. Conversely, the dynamic model captures the dynamic changes that occur during acceleration. Furthermore, the obstacle scenarios are described using model equations to provide a clear representation of the challenges the AUV may encounter.

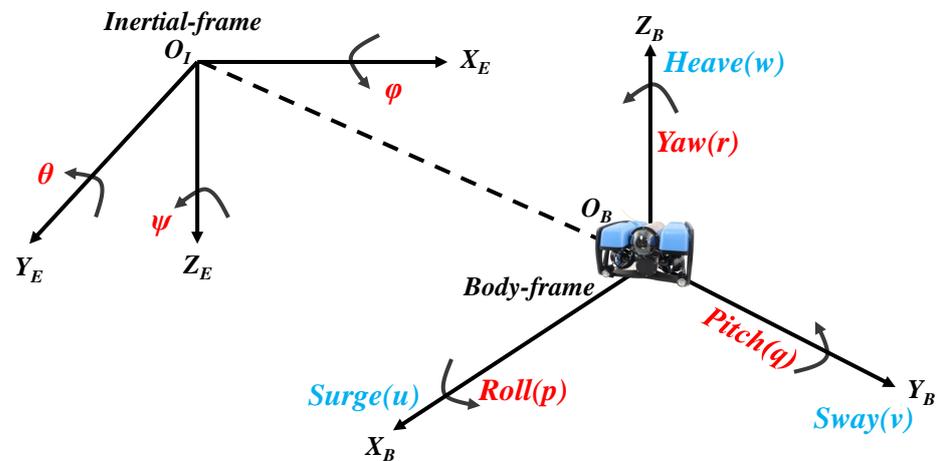


Figure 2. Inertial and motion coordinate system of autonomous underwater vehicles (AUVs), the left is the inertial-fixed coordinate system and the right is the body-fixed coordinate system.

2.1. Coordinate Systems of AUVs

The AUV system is designed to operate with six degrees of freedom (6DOF), allowing it to achieve independent movement in three linear directions (surge, sway, and heave) and three rotational directions (roll, pitch, and yaw). These six DOF provide the AUV with the capability to navigate and maneuver effectively in three-dimensional space. To describe the AUV’s motion, two frames of reference are commonly used: the inertial frame (I-frame) and the body frame (B-frame) [24]. It is assumed that the center of gravity and the center of buoyancy of the ROV are located at the center of the robot skeleton. This configuration is illustrated in Figure 2.

2.2. Kinematic System Model

The kinematic model primarily explains the geometric relationships among the motion variables of the AUV system, without considering the forces and torques that cause system motion and changes. Conversion can be performed between the body coordinate system and the earth coordinate system variables. The parameters of the AUV system include the position vector η , the velocity vector v , and the thrust vector τ [25]. The definitions of each variable are as follows.

$$\begin{aligned} \eta &= [x, y, z, \phi, \theta, \psi]^T \\ v &= [u, v, w, p, q, r]^T \\ \tau &= [X, Y, Z, K, M, N]^T \end{aligned} \tag{1}$$

The position vector η represents the coordinates in the Earth coordinate system, encompassing the absolute position of the origin (x, y, z) and the Euler angles (ϕ, θ, ψ) that indicate the orientation. In contrast, the body coordinate system is defined based on the centroid of the AUV. In this system, the velocity vector v describes the linear velocities (u, v, w) along the three axes of the body coordinate system, as well as the angular velocities (p, q, r) around those axes.

Furthermore, the thrust vector $\tau(X, Y, Z, K, M, N)$ denotes the external forces and torques acting upon the AUV system, expressed as a vector within the body coordinate system. To facilitate the transformation of variables between these two coordinate systems, a reversible transformation matrix $J(\eta) \in \mathbb{R}^{6 \times 6}$ is employed. This matrix enables the conversion between the respective coordinate systems.

$$\begin{aligned} \dot{\eta} &= J(\eta)v \\ J(\eta) &= \begin{bmatrix} R & 0 \\ 0 & T \end{bmatrix} \end{aligned} \tag{2}$$

The Euler angles $\eta = [\phi, \theta, \psi]^T$ correspond to the roll, pitch, and yaw angles of the AUV system. The velocity transformation matrix, denoted $R(\eta)$, facilitates the conversion from the body coordinate system to the Earth coordinate system. Similarly, the angular velocity transformation matrix, denoted $T(\eta)$, enables the conversion from the body coordinate system to the Earth coordinate system. Conversely, when transforming from the Earth coordinate system to the body coordinate system, an inverse matrix is utilized. The definitions of $R(\eta)$ and $T(\eta)$ are as follows:

$$R(\eta) = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \tag{3}$$

$$T(\eta) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \tag{4}$$

When $\theta = \pm 90^\circ$, $T(\eta)$ does not exist. Therefore, the angles of the AUV system have certain limitations. The roll angle ϕ lies within the range of $-\pi < \phi \leq \pi$, while the pitch angle θ is restricted to $-\pi/2 < \theta < \pi/2$.

2.3. Dynamic System Model

The AUV is a 6DOF nonlinear dynamic model with disturbance. The dynamics equation of the AUV system studies the relationship between external forces and motion, which can be expressed using the following formula,

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau + \omega \tag{5}$$

where $M = M_{RB} + M_A$, M_{RB} is a rigid body inertia matrix and M_A is an added mass inertia matrix. Among them, the standard M_{RB} is

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{xz} \\ mz_G & 0 & -mx_G & -I_{yx} & I_y & -I_{yz} \\ -my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{6}$$

If the center of mass and center of buoyancy of the AUV are concentric, the above formula can be greatly simplified.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{7}$$

The same type of C_{RB} is the coriolis force and added mass term centripetal force,

$$C(v) = C_{RB}(v) + C_A(v) \tag{8}$$

where

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & mv_z & -mv_y \\ 0 & 0 & 0 & -mv_z & 0 & mv_x \\ 0 & 0 & 0 & mv_y & -mv_x & 0 \\ 0 & mv_z & -mv_y & 0 & I_z\omega_z & -I_y\omega_y \\ -mv_z & 0 & mv_x & -I_z\omega_z & 0 & I_x\omega_x \\ mv_y & -mv_x & 0 & I_y\omega_y & -I_x\omega_x & 0 \end{bmatrix} \tag{9}$$

and

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z\dot{v}_z v_z & Y\dot{v}_y v_y \\ 0 & 0 & 0 & Z\dot{v}_z v_z & 0 & -X\dot{v}_x v_x \\ 0 & 0 & 0 & -Y\dot{v}_y v_y & X\dot{v}_x v_x & 0 \\ 0 & -Z\dot{v}_z v_z & Y\dot{v}_y v_y & 0 & -N\dot{\omega}_z \omega_z & M\dot{\omega}_y \omega_y \\ Z\dot{v}_z v_z & 0 & -X\dot{v}_x v_x & N\dot{\omega}_z \omega_z & 0 & -K\dot{\omega}_x \omega_x \\ -Y\dot{v}_y v_y & X\dot{v}_x v_x & 0 & -M\dot{\omega}_y \omega_y & K\dot{\omega}_x \omega_x & 0 \end{bmatrix} \tag{10}$$

According to the literature [26], we divide $D(|v|)$ for the surface resistance of the linear damping D and wave nonlinear quadratic damping $D_n(|v|)$ into two parts. The literature [24] states that the assumption of the AUV system has good symmetry, therefore,

$$D(|v|) = -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\} - \text{diag}\{X_{u|u}|u|, Y_{v|v}|v|, Z_{w|w}|w|, K_{p|p}|p|, M_{q|q}|q|, N_{r|r}|r|\} \tag{11}$$

The AUV system is subject to vertical downward gravity W acting on the center of gravity and vertical upward buoyancy B acting on the center of buoyancy on the vertical plane. The surface vehicle equilibrium condition is $W = B$, and it acts on the same plumb line. AUV systems are usually designed to be critical with near-zero buoyancy, i.e., gravity W is slightly less than buoyancy B . Restoring force determines the initial stability of the system; that is, when the AUV system is out of equilibrium due to external forces, the combined action of gravity and buoyancy can restore the system to equilibrium. The center of buoyancy is higher than the center of gravity, which can generate the torque of restoring equilibrium:

$$g(\eta) = \begin{bmatrix} (W - B) \sin(\theta) \\ -(W - B) \cos(\theta) \sin(\phi) \\ -(W - B) \cos(\theta) \cos(\phi) \\ -(y_g W - y_b B) \cos(\theta) \cos(\phi) + (z_g W - z_b B) \cos(\theta) \sin(\phi) \\ (z_g W - z_b B) \sin(\theta) + (x_g W - x_b B) \cos(\theta) \cos(\phi) \\ -(x_g W - x_b B) \cos(\theta) \sin(\phi) - (y_g W - y_b B) \sin(\theta) \end{bmatrix} \quad (12)$$

All the symbols used in the above equations are defined as follows: m is the mass of the AUV and $X_{u|u|}$, $Y_{v|v|}$, $Z_{w|w|}$, $K_{p|p|}$, $M_{q|q|}$, and $N_{r|r|}$ are hydrodynamic coefficients.

2.4. Obstacle Model

Static underwater obstacles mainly include rocks, shipwrecks, and artificial obstacle lights. These obstacles can be simplified into spheres, cylinders, cuboids, and semi-spheres or their combinations. Moving obstacles such as large fish and underwater moving hulls can be regarded as spheres, and various typical obstacles can be regarded as a variety of standard convex objects. The shape of q can use a unified mathematical expression [27]:

$$\Gamma(P) = \left(\frac{x - x_0}{a}\right)^{2p} + \left(\frac{y - y_0}{b}\right)^{2q} + \left(\frac{z - z_0}{c}\right)^{2r} \quad (13)$$

In the given context, (x_0, y_0, z_0) represents the center of the obstacle, and (x, y, z) denotes the current position of the AUV. The coefficients $a, b,$ and $c,$ respectively, indicate the sizes of the obstacle's axis lengths, while the coefficients $p, q,$ and r determine the shape of the obstacle. The specific shapes and coefficient values for various convex obstacles are as follows: Sphere: $p = q = r = 1, a = b = c$; Cylinder: $p = q = 1, r > 1, a = b$; Cone: $p = q = 1, 0 < r < 1, a = b$; Cuboid: $p > 1, q > 1, r > 1$. Figure 3 illustrates typical convex obstacle shapes created by different coefficient combinations. Here, $\Gamma(P) < 1, \Gamma(P) > 1,$ and $\Gamma(P) = 1$ represent the interior region, exterior region, and surface of the obstacle, respectively. The simulation plots for various obstacles are shown below. In real scenarios, the obstacle environment can be created by combining different obstacles as depicted in Figure 3.

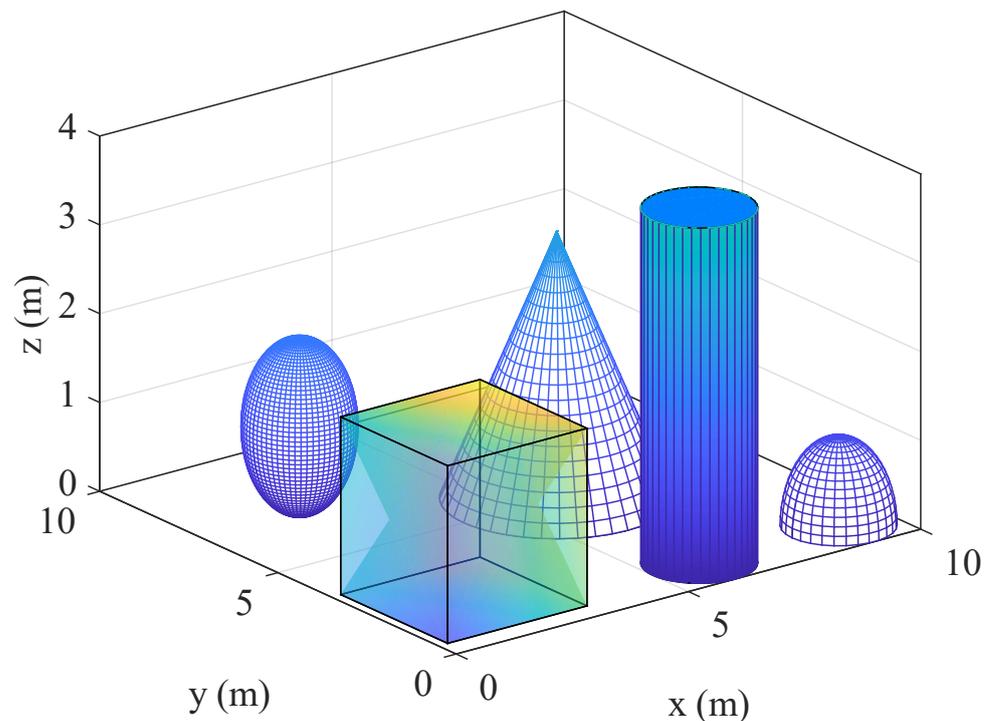


Figure 3. convex obstacles in an ocean environment.

3. Hybrid Layer Controller Design

3.1. Proposed Hybrid Structure of Controller

The proposed hybrid layer controller, as in Figure 4, comprises an IIFDS path planner and an NMPC tracking controller. The IIFDS is responsible for path planning by considering the agent’s current position, obstacles, and destination. It can iteratively update the overall path based on the initial agent position and surrounding environment information, or dynamically update the next reference position based on its own position and environment information, making it suitable for real-time path planning in dynamic environments. During the path planning process, drive and legality constraints are considered to ensure a feasible path. The NMPC controller effectively tracks the trajectory and rejects dynamic disturbances. Combining the two controllers enables real-time path planning and tracking from the start point to the destination, forming a double loop of navigation and control.

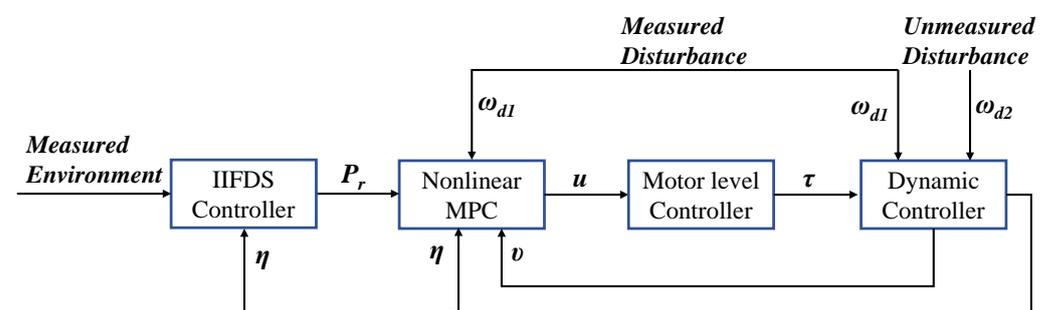


Figure 4. Hybrid structure of controller.

3.2. Upper Layer of IIFDS Navigation Controller Design

The algorithm is based on a variational approach, where the perturbation effect of obstacles on the flow field is represented using a modified quantization matrix \bar{M} , resulting in a perturbed flow field [13]. The introduction of tangential velocity in [14] expands the physical characteristics of the fluid, making the perturbed streamlines more suitable for AUV operation. Prior to planning the path, all obstacles are defined according to the formula in Equation (13). The position of the destination point is set as $P_d = (x_d, y_d, z_d)$ and the number of obstacles is denoted as K . If there are no obstacles in the current path, the original streamlines go directly from the initial position to the target point with a velocity magnitude of V just as Figure 5 shown. Hence, the initial flow field velocity $u(P)$ can be represented by Equation (14), where u represents the flow velocity. Subsequently, the perturbation effect of obstacles on the flow field is quantified using the modification matrix M . Finally, the initial flow field is corrected to obtain the perturbed flow field. This perturbed flow field remains stable, meaning that the flow streamlines can avoid obstacles and eventually reach the target point.

The design process of the proposed method can be summarized as follows:

Step 1. Calculate the initial fluid field $u(P)$ by considering the current position and the destination.

$$u(P) = -\left[\frac{V(x - x_d)}{d(P, P_d)}, \frac{V(y - y_d)}{d(P, P_d)}, \frac{V(z - z_d)}{d(P, P_d)}\right]^T \tag{14}$$

In the algorithm, $u(P)$ represents the initial flow field at each computed sampling point during the operation of the AUV. This initial flow field evolves continuously as the position changes. Here, V denotes the confluence velocity and $d(P, P_d)$ represents the distance between the AUV position P and the target position P_d .

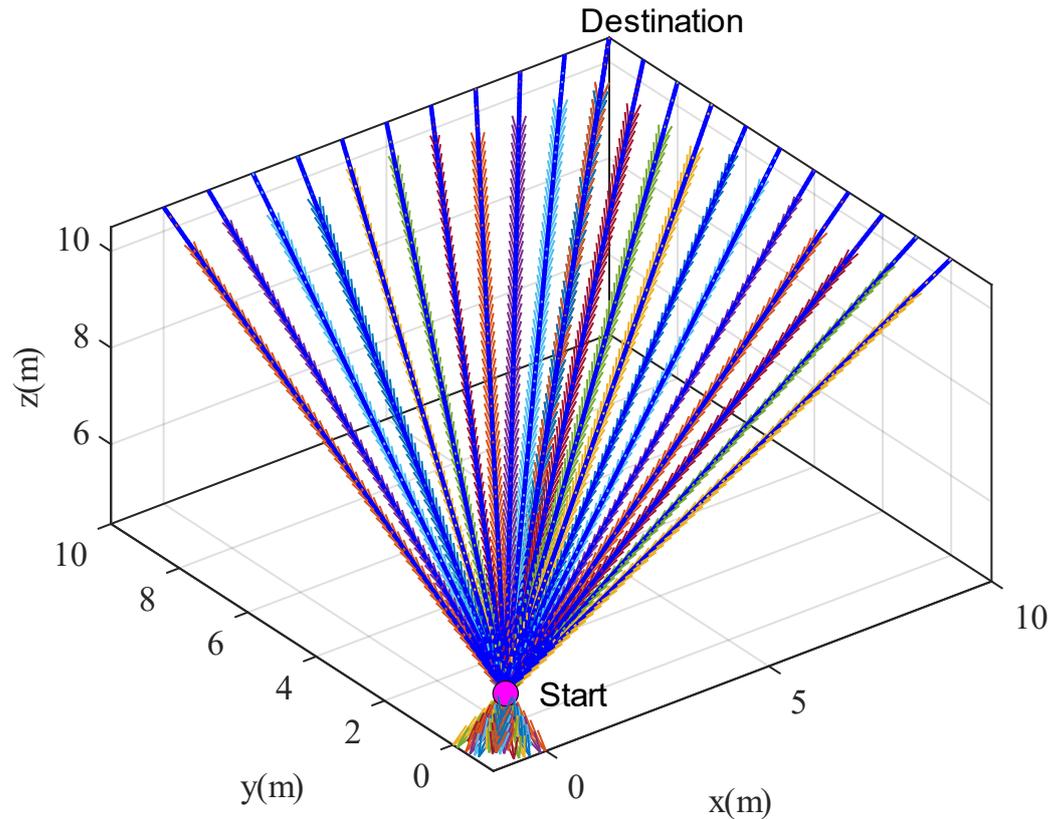


Figure 5. Initial flow field. Start position is (0,0,0), destination position is (10,10,10), and the direction is from AUV position to the destination.

$$d(P, P_d) = \sqrt{(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2} \tag{15}$$

Step 2. Calculate the disturbance modification matrix \bar{M} based on the presence of static or moving obstacles.

If there are K obstacles present in the environment, then the perturbed flow field velocity is obtained by modifying the initial flow field velocity $u(P)$ using the modification matrix \bar{M} for each obstacle.

$$\bar{M} = \sum_{k=1}^K \omega_k \bar{M}_k \tag{16}$$

where w_k represents the weight factor of the k_{th} obstacle.

$$\omega_k = \begin{cases} 1, K = 1 \\ \prod_{i=1, i \neq k}^K \frac{\Gamma_i - 1}{(\Gamma_i - 1) + (\Gamma_k - 1)}, K \neq 1 \end{cases} \tag{17}$$

where Γ_i and Γ_k represent the expressions for the i_{th} and k_{th} obstacles, respectively, calculated according to Equation (13). Due to the condition $\omega_{sum} = \sum_{k=1}^K \omega_k \leq 1$, it is necessary to normalize the weight of each obstacle at each position. The normalized weight of the k_{th} obstacle is given by

$$\omega_{k,nor} = \frac{\omega_k}{\omega_{sum}} \tag{18}$$

With this normalization, Equation (16) can be written as follows,

$$\bar{M} = \sum_{k=1}^K \omega_{k,nor} \bar{M}_k \tag{19}$$

Building upon the original IFDS [13], the introduction of the tangential matrix incorporates the perturbation matrix. Therefore, \bar{M}_k represents the modification matrix for the k_{th} obstacle.

$$\bar{M}_k = I - \frac{n_k n_k^T}{\Gamma_k^{\rho_k} n_k^T n_k} + \frac{t_k n_k^T}{\Gamma_k^{\sigma_k} |t_k| |n_k|} \tag{20}$$

where ρ_k is the repulsion coefficient, σ_k is the saturation coefficient, and θ_k is the tangential coefficient. These three factors collectively determine the avoidance time and sensitivity in obstacle avoidance.

$$\rho_k = \rho_k^0 \cdot e^{(1 - \frac{1}{d(P, P_d)^{\frac{1}{d(P, O_k)}}})} \tag{21}$$

$$\sigma_k = \sigma_k^0 \cdot e^{(1 - \frac{1}{d(P, P_d)^{\frac{1}{d(P, O_k)}}})} \tag{22}$$

where $\rho_k^0 > 0$ is the repulsion factor of the k_{th} obstacle, $\sigma_k^0 > 0$ is the tangential factor of the k_{th} obstacle, $d(P, P_d)$ is the distance between the AUV object and the target point, $d(P, O_k)$ is the distance between the AUV and the surface of the k_{th} obstacle, I represents the identity matrix of order three, and $n_k(P)$ is the outward normal vector to the surface of the k_{th} obstacle perpendicular to it.

$$n_k = (\frac{\partial \Gamma_k}{\partial x}, \frac{\partial \Gamma_k}{\partial y}, \frac{\partial \Gamma_k}{\partial z})^T \tag{23}$$

As for the other variables, $-\frac{n_k n_k^T}{\Gamma_k^{\rho_k} n_k^T n_k}$ is the repulsive matrix, $\frac{t_k n_k^T}{\Gamma_k^{\sigma_k} |t_k| |n_k|}$ is a tangential matrix, and t_k is the horizontal tangent vector, which can be obtained using the following expression. The tangent coordinate system $o' - x'y'z'$ is defined by setting $t_{k,1}$, $t_{k,2}$, and n_k as the x' , y' , and z' axes, respectively. The calculation process of t_k is as follows:

Define two mutually orthogonal vectors, $t_{k,1}$ and $t_{k,2}$, which are also orthogonal to n_k ,

$$\begin{cases} x' = t_{k,1} = [\frac{\partial \Gamma_k}{\partial y}, -\frac{\partial \Gamma_k}{\partial x}, 0]^T \\ y' = t_{k,2} = [\frac{\partial \Gamma_k}{\partial x} \frac{\partial \Gamma_k}{\partial z}, \frac{\partial \Gamma_k}{\partial y} \frac{\partial \Gamma_k}{\partial z}, -(\frac{\partial \Gamma_k}{\partial x})^2 - (\frac{\partial \Gamma_k}{\partial y})^2]^T \end{cases} \tag{24}$$

By using n_k , $t_{k,1}$, and $t_{k,2}$ as the basis vectors for a new coordinate system, any unit vector in the plane perpendicular to n_k can be expressed in the form

$$t'_k = [\cos\theta_k, \sin\theta_k, 0]^T \tag{25}$$

where $\theta'_k \in [-\pi, \pi]$ is the tangential direction factor. This angle represents the angle between any tangent vector and the x' axis. It determines the angle of avoidance direction. The z' axis is obtained through rotation. Therefore, in the inertial coordinate system, t'_k is transformed to the xoy standard coordinate system by using the tangent coordinate axis transformation matrix,

$$t_k = \Omega_T^I t'_k \tag{26}$$

where Ω_T^I is the transformation matrix from the tangent coordinate system to the inertial coordinate system,

$$\Omega_T^I = \begin{bmatrix} r_y & r_x r_z & r_x \\ r_2 & r_2 r_3 & r_3 \\ r_x & r_y r_z & r_y \\ r_2 & r_2 r_3 & r_3 \\ 0 & -r_2 & r_3 \end{bmatrix} \tag{27}$$

and $r_x = \frac{\partial \Gamma_k}{\partial x}$, $r_y = \frac{\partial \Gamma_k}{\partial y}$, $r_z = \frac{\partial \Gamma_k}{\partial z}$, $r_2 = \sqrt{r_x^2 + r_y^2}$, $r_3 = \sqrt{r_x^2 + r_y^2 + r_z^2}$.

Step 3. Compute the final perturbed fluid field using the results obtained from the previous two steps.

After modifying the initial flow field velocity $u(P)$, the perturbed flow field velocity becomes

$$\bar{u} = \bar{M}u(P) \tag{28}$$

During the actual operation of the AUV, it encounters not only static obstacles but also dynamic obstacles. In such cases, the velocity of dynamic obstacles needs to be considered in the flow field [15]. To incorporate the dynamic obstacle velocity, the process involves calculating the relative positions based on the static flow field, adding perturbation flow field effects, and finally incorporating the dynamic obstacle velocity.

The total velocity $v(P)$ of a dynamic obstacle is a combination of the obstacle’s velocity and its weight factor. If there is only one obstacle, the weight factor ω can be calculated according to Equation (17) as mentioned earlier. Assuming the velocity of the k_{th} obstacle is $v_{k,obs}$, its reference motion velocity is given by

$$\overline{v_{k,obs}} = e^{-\left(\frac{\Gamma_k(P)-1}{\lambda_k}\right)} v_{k,obs} \tag{29}$$

Step 4. Calculate the velocity of the dynamic obstacle under the action of the flow field.

The total vector velocity of the obstacle can be calculated using the following formula,

$$v(P) = \sum_{k=1}^K \omega_{k,nor}(P) \overline{v_{k,obs}} \tag{30}$$

Here, $v_{k,obs}$ represents the actual velocity of the k_{th} obstacle at position P at a specific time point. λ_k is the corresponding weight, where a higher value indicates a shorter time frame for the AUV to avoid the obstacle.

Step 5. Finally, determine the next position at the subsequent time step by considering the sample time and the current position.

Velocity and position of the final perturbed flow field are

$$\bar{u}(P) = \begin{cases} \bar{M}(P)(u(P) - v(P)) + v(P), & \text{for the dynamic obstacle} \\ \bar{M}(p)u(P), & \text{for the static obstacle} \end{cases} \tag{31}$$

$$P_{next} = P + \bar{u}(P)\Delta_t \tag{32}$$

Among them, Δ_t is the sampling control period. A smaller value of Δ_t leads to more accurate results, but it also increases computational complexity. The parameters ρ_k , σ_k , and θ_k shape the streamlines. Adjusting σ_k and θ_k avoids stagnation points and traps.

In IIFDS, the tangential matrix enables fluid flow in multiple directions around obstacles. θ_k determines the path plane; when it is close to 0 or π , the path is horizontal (Path 1, Path 5), favoring side avoidance. When it is close to $\pi/2$, the path is vertical (Path 3), favoring top avoidance, as in Figure 6.

For the parameters σ and ρ , which represent the tangential reaction coefficient and repulsion coefficient, respectively, the larger their values, the earlier the planned path avoids obstacles, and the AUV stays farther away from obstacles, making the planned path safer, as in Figures 7 and 8.

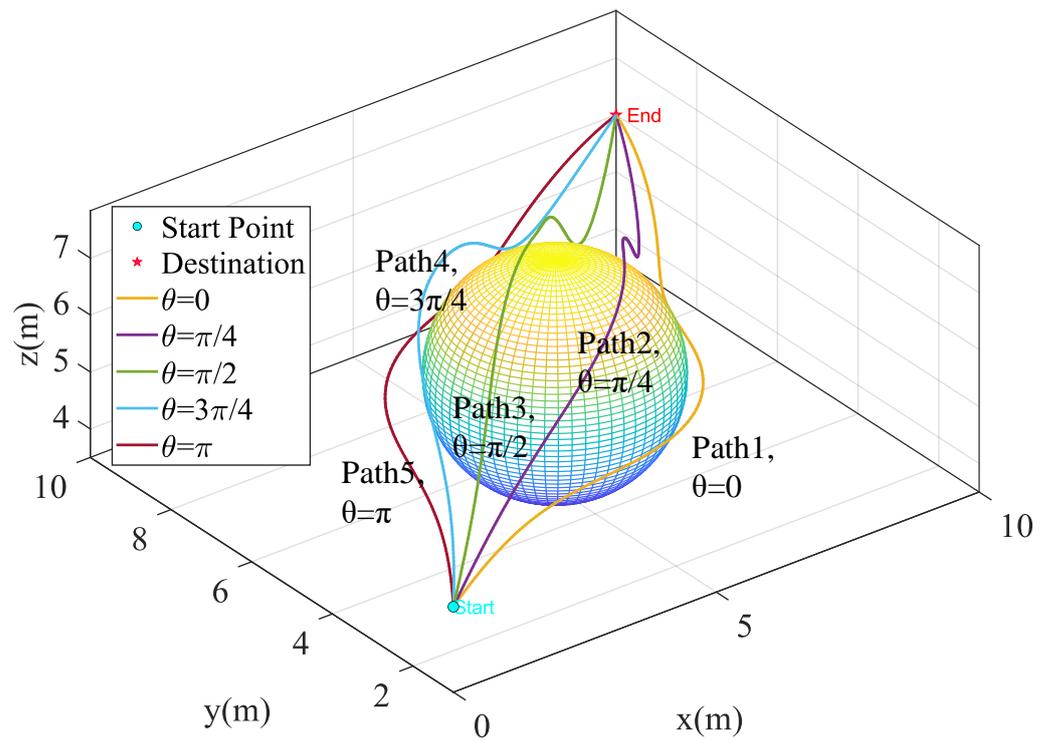


Figure 6. Different direction angle θ for path planning.

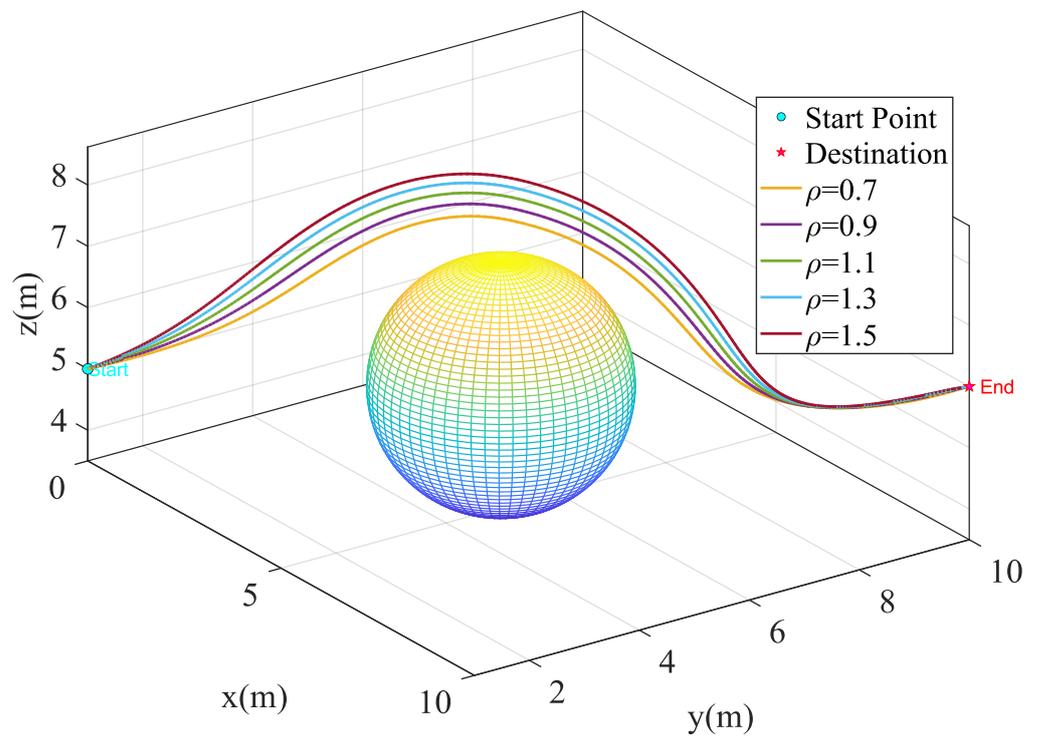


Figure 7. Different reaction coefficients ρ for path planning.

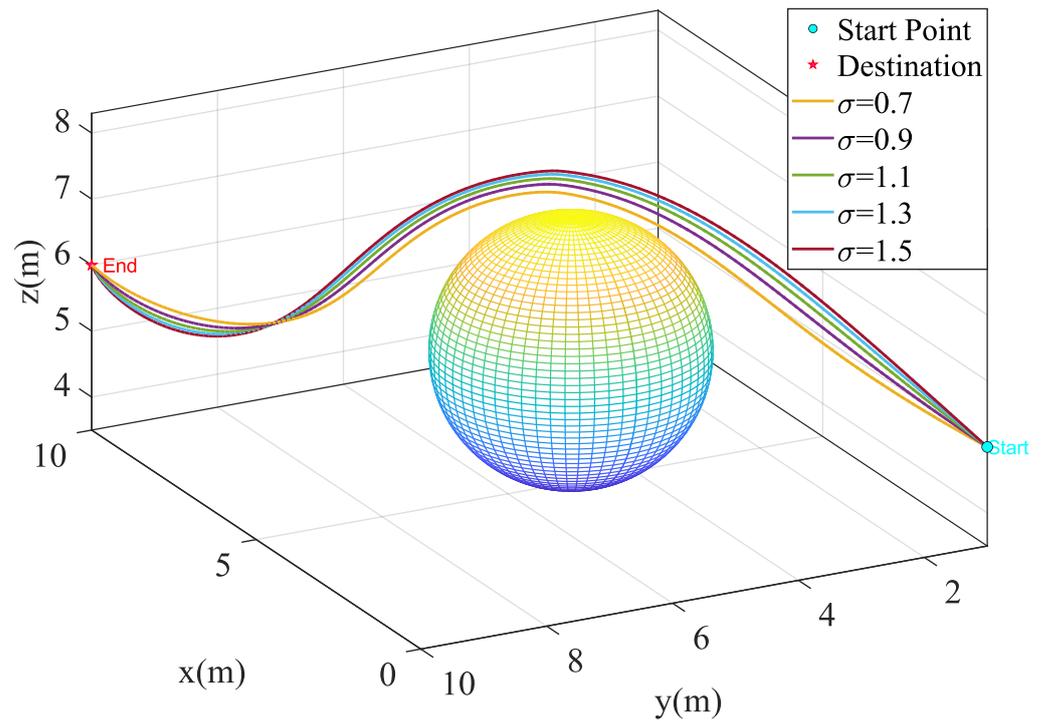


Figure 8. Different repulsive reaction coefficients σ for path planning.

In the case of a moving obstacle environment, as shown in Figure 9, the parameter λ is a positive value. A larger value indicates that the AUV avoids obstacles earlier, reacts faster, and achieves better avoidance effects. However, when the AUV is farther away from the obstacles, the relative reference velocity increases. This increases the possibility of collision between the AUV and the obstacles. In other words, the higher the difficulty level of avoiding obstacles for the AUV, the greater the associated risk. This is because the moving obstacles appear relatively stationary in the frame of reference of the navigation vector field.

In general, randomly selected parameters affect the length of the path and the distance from obstacles, potentially rendering the path planning infeasible. Therefore, parameter optimization is crucial for the performance of the system. By selecting appropriate parameters, the IIFDS method can always strike a balance between safety and smoothness, effectively avoiding obstacles.

3.3. Bottom Layer of NMPC-Based Trajectory Tracking Controller Design

The reference path is planned using the outer-loop path planning layer, and the reference path can be described in the following form,

$$P_r(t) = (x_d(t), y_d(t), z_d(t)) \tag{33}$$

The path tracking error is

$$P_e(t) = P(t) - P_r(t) \tag{34}$$

and the control object is to guarantee the deviation of the position to converge to zero, i.e., $\lim_{t \rightarrow \infty} P_e(t) \rightarrow 0$. The reference path dynamically changes based on the movement of the plant and the distance between obstacles. If there are any dynamic obstacles present, the corresponding fluid velocity is adjusted accordingly. The inner-loop NMPC controller then follows the receding path. This process can be described in the following steps:

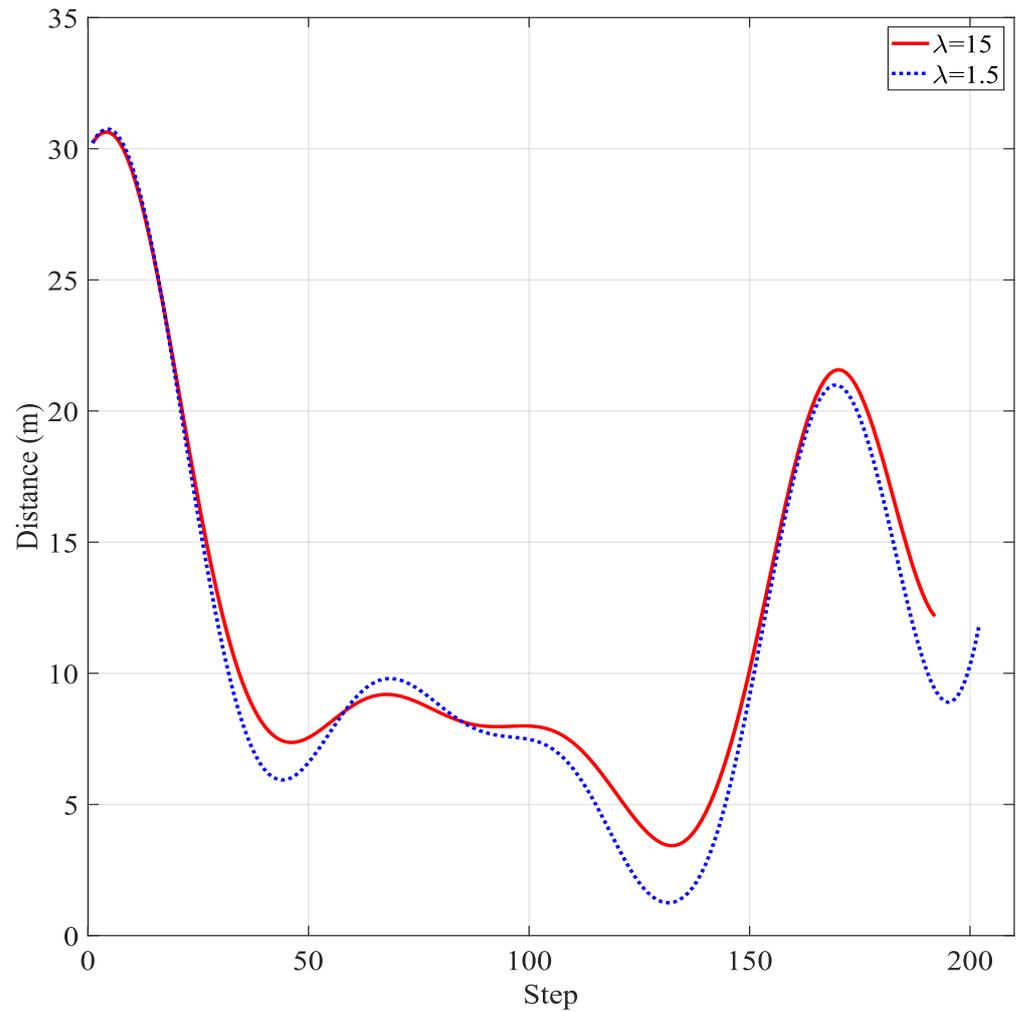


Figure 9. Dynamic reaction coefficient λ affects the distance.

Step 1. Express the model in the nonlinear state space equation.

The nonlinear state model can be described as follows:

$$x = [\eta, v]^T \tag{35}$$

and the first deviating equation consists of state, control input, and disturbance,

$$\dot{x} = [\dot{\eta}, \dot{v}]^T = f(x, u, \omega, t) \tag{36}$$

The Nonlinear MPC model's function is

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), \omega(t)) \\ y(t) = h(x(t)) \end{cases} \tag{37}$$

Step 2. Prediction process.

Regarding the state space of the nonlinear model, the next state can be determined using the Runge–Kutta (RK) method. In the prediction process, both the fourth-order and sixth-order RK methods can be employed. However, it is crucial to strike a balance between computational cost and accuracy. Considering this trade-off, utilizing the RK4 method is deemed sufficient [28].

$$\begin{aligned} x_{t+1} &= f_d(x_t, u_t, \omega_d, t) = x_t + \Delta t \cdot f(x_t, u_t, \omega_{dt}, t) \\ &= x_t + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{38}$$

$$\begin{aligned}
 k_1 &= \Delta_t \cdot f(x_t, u_t, w_{dt}) \\
 k_2 &= \Delta_t \cdot f(x_t + \frac{k_1}{2}, u_t, w_{dt}) \\
 k_3 &= \Delta_t \cdot f(x_t + \frac{k_2}{2}, u_t, w_{dt}) \\
 k_4 &= \Delta_t \cdot f(x_t + k_3, u_t, w_{dt})
 \end{aligned}
 \tag{39}$$

The characteristics of nonlinear AUV dynamics are captured during each time step of model prediction. Furthermore, the prediction is based on the actual state and control variables x_t, u_t , rather than operating points x_0, u_0 . This grants Nonlinear Model Predictive Control (NMPC) a natural advantage over other control strategies that rely on linear approximations. Additionally, the controller prediction model incorporates the propeller characteristics and disturbances from the ROV dynamics Ordinary Differential Equations (ODEs). Therefore, the discrete model f_d possesses the capability to predict nonlinear ROV dynamics in the presence of 6DOFdisturbances. This implies that NMPC can better adapt to the actual dynamic characteristics and disturbances of the ROV system, rather than being limited to a specific operating point and linear approximation. Such features make NMPC particularly effective in complex, real-time control environments.

Step 3. Control object and the cost function.

According to the NMPC frame, the path tracking problem can be seen as a constraint optimization problem, as it is for the cost function, and N is the predictive time domain [29],

$$J = \sum_{i=1}^N \left(\omega_1 \|x_p(k+i|k) - x_{pref}\| + \omega_2 \|x_v(k+i|k) - x_{vref}\| + \omega_3 \|u(k+i|k) - u_{ref}\| \right)
 \tag{40}$$

$$\text{s.t } x_0 = x_{init}, 0 \leq x_t \leq x_{max}
 \tag{41}$$

Just as in Equation (40), ω_1, ω_2 are the corresponding weight matrices to position deviation x_p and the velocity deviation x_v , ω_3 is the weight matrix on control Δu , and $\omega_1 \in \mathbb{R}^{6*1}, \omega_2 \in \mathbb{R}^{6*1}$, and $\omega_3 \in \mathbb{R}^{8*1}$. These three control objectives the nonlinear MPC would like to achieve at the same time. Firstly, the objective of the cost function is to minimize the tracking error, ensuring that the AUV approaches and stably maintains as close as possible to the reference position. The cost function adopts a formulation represented by $\omega_1 \|x_p(k+i|k) - x_{pref}\|$, which penalizes the disparity between the current state and the positional reference. Secondly, during underwater operations, angular velocity and linear velocity need to be maintained at zero after reaching the target point. Consequently, x_{vref} is likewise set to 0_{6*1} . Last but equally significant, the optimal controller is subject to three constraints. The first two constraints, respectively, ensure stable path-keeping and remaining stationary. The design of the optimal controller aims at minimizing the overall control effort. The intermediate cost J constitutes the sum of all control variables, governed by the optimal control strategy u^* to minimize the control effort of the AUV. The reference control variables u_{ref} are set to 0_{8*1} .

During the path tracking process of an AUV, both the operational state and control variables of the AUV are subject to certain constraints. These constraints consist of both hard and soft limits. It is imperative to adhere to these constraints, with particular emphasis on the hard constraints, as they must be strictly followed for safety reasons. The system state x_t and the control variables u_t are constrained as follows:

$$\begin{aligned}
 x_{min} &\leq x_t \leq x_{max} \\
 u_{min} &\leq u_t \leq u_{max}
 \end{aligned}
 \tag{42}$$

where x_{min}, u_{min} are the lower bound of the state vector and the control input, and x_{max}, u_{max} are the upper bound of the state vector and the control value. In the control process, the bound domain can be rewritten as

$$\begin{aligned}
 u_{min} &\leq u(k) \leq u_{max} \\
 x_{min} &\leq f(x_t, u_t, w_d) \leq x_{max}
 \end{aligned}
 \tag{43}$$

The aforementioned NMPC equations undergo optimization at each time step, with only the first optimized control signal u^* being applied to the AUV. The system's new control trajectory is optimized in a receding horizon manner. In other words, after the AUV sends the first control signal, the prediction horizon transitions from $[0, T]$ to $[t, t + T]$.

In summary, the objective of this controller is to enable the AUV to track the reference position, ultimately stabilizing at a stationary state while achieving optimal control.

3.4. Algorithm and Pseudocode

The algorithm for path planning and trajectory tracking is outlined in Algorithm 1, while the NMPC algorithm is encapsulated in Algorithm 2.

Algorithm 1 : Hybrid Layer of Navigation and Tracking Controller of the AUV.

- 1: Initialize the initial state x_0 , the destination position p_d , the velocity of AUV v , the predictive horizon N , the weight matrix $\omega_1, \omega_2, \omega_3$, control value bound u_{max}, u_{min} , state bound x_{min}, x_{max} ;
 - 2: Compute the distance from current position P to destination position P_d , using Equation (15).
 - 3: **while** AUV has not reached the destination **do**
 - 4: Calculate the initial fluid field $u(P)$, using Equation (14),
 - 5: Obtain obstacle information from the environment, including the position $O(x, y, z)$ and velocity $v(x, y, z)$, through the perception layer. Based on the current position, as well as the positions and velocities of obstacles (if they are dynamic), calculate the disturbance flow field matrix \bar{M} , using Equations (16)–(27).
 - 6: Calculate the overall vector velocity of dynamic obstacles in the flow field $\bar{v}_{k,obs}$, if applicable, using Equations (29) and (30).
 - 7: Calculate the next potential position for the next path P_{next} , using Equations (31) and (32).
 - 8: Compute the optimal control quantity based on the reference path and Equations (33)–(43).
 - 9: Calculate the control quantity using Equation (40) and Algorithm 2 based on the current position P and the next position P_{next} , which is the path to be tracked.
 - 10: Measure all the states of the scenario, compute the distance from current position P to destination position P_d , using Equation (15).
 - 11: **end while**
-

Algorithm 2 : The Process of Computing the Control Value u in NMPC.

- 1: Initialize all initial state variables x_0 , initialize all control variables u , set up the weight matrices $\omega_1, \omega_2, \omega_3$, and define the range of control constraints u_{min}, u_{max} , set the predictive horizon N , obtain the reference state x_{ref} from the planned path.
 - 2: **while** Not meeting the specified prediction horizon cycles N , **do**
 - 3: Calculate the next state variable based on Equation (38),
 - 4: Restrict the state variables within the constraints x_{min}, x_{max} defined by Equation.
 - 5: Compute the cost based on the reference path and weight matrix, according to Equation (40).
 - 6: **end while**
 - 7: Sum up the cost values obtained in the aforementioned N instances to calculate the total cost, yielding the value of J based on Equation (40).
 - 8: Compute the control input u based on the cost J and constraints u_{max}, u_{min} , in this process, the optimal solver is SQP, and the computed control variables need to adhere to the constraints specified by Equation (43).
 - 9: Apply the first dimension control input $u[1 :]$ to control the system.
-

4. Computer Simulation Experiment

To validate the proposed method for path planning and AUV trajectory tracking, we utilize the AUV model described in Section 2 and employ the hybrid layer controller outlined in Section 3. The simulation experiments are conducted within the Matlab 2018 Rb environment, with all AUV parameters defined in Table 1. The experimental system is equipped with an 11th Gen Intel(R) Core(TM) i9-11900H @ 2.50 GHz processor, 32.0 GB RAM, and Windows 10 operating system.

Table 1. Parameters of the AUV.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
m	11.5 kg	I_y	0.16 kg.m ²	K_p	-0.12 kg.m ² /rad	L_{DZw}	-5.18 Ns/m	Q_{DYv}	-21.66 Ns ² /m ²
L	0.4571 m	I_z	0.16 kg.m ²	$M_{\dot{q}}$	-0.12 kg.m ² /rad	L_{DKp}	-0.07 Ns/rad	Q_{DZw}	-36.99 Ns ² /m ²
W	0.4361m	$X_{\dot{u}}$	-5.5 kg	N_r	-0.12 kg.m ² /rad	L_{DMq}	-0.07 Ns/rad	Q_{DKp}	-1.55 Ns ² /rad ²
H	0.2539 m	$Y_{\dot{v}}$	-12.7 kg	L_{DXu}	-4.03 Ns/m	L_{DNr}	-0.07 Ns/rad	Q_{DMq}	-1.55 Ns ² /rad ²
I_x	0.16 kg.m ²	$Z_{\dot{w}}$	-14.57kg	L_{DYv}	-6.22 Ns/m	Q_{DXu}	-18.18 Ns ² /m ²	Q_{DNr}	-1.55 Ns ² /rad ²

The test environment encompasses both static obstacle and dynamic obstacle scenarios. The crucial metrics for evaluation include path planning time and tracking control deviation, path planning length, smoothness, and the range of control values during tracking that should also be considered.

In Table 1, m is the mass of the AUV; L,W, and H, respectively, represent the length, width, and height of the AUV; $I_x, I_y,$ and I_z are the moments of inertia about the $x_b, y_b,$ and z_b axes; $X_{\dot{u}}, Y_{\dot{v}},$ and $Z_{\dot{w}}$ represent the hydrodynamic forces for linear motion; and $K_p, M_{\dot{q}},$ and N_r represent the hydrodynamic forces for rotational motion. $L_{DXu}, L_{DYv},$ and L_{DZw} are the damping coefficients in the translational directions, while $L_{DKp}, L_{DMq},$ and L_{DNr} are the damping coefficients in the rotational directions. $Q_{DXu}, Q_{DYv},$ and Q_{DZw} are the acceleration damping coefficients in the translational directions, and $Q_{DKp}, Q_{DMq},$ and Q_{DNr} are the acceleration damping coefficients in the rotational directions.

4.1. Static Obstacle Avoidance and Path Planning

For the evaluation, we set the initial point as a fixed value (0, 0, 0) and consider four destination points: (10, 10, 10), (10, 10, 9), (10, 9, 10), and (9, 10, 10). In our test scenario, we introduce random spheres as obstacles, each with a unique position and size. Specifically, the obstacle positions are (2, 5, 3), (2, 2, 2), (8, 8, 8), (6, 6, 6), (1, 7, 4), and (3, 7, 7), accompanied by corresponding radii of 0.5, 1, 1, 0.6, 0.6, and 0.5, respectively.

The IIFDS parameters $\rho = 1.5, \sigma = 0.5,$ and $\theta = 0.5$ were selected. It has been noted in the literature [7] that the A* method is frequently employed for path planning. To evaluate their performance in the aforementioned environment, both the A* and IIFDS algorithms were tested.

From Table 2, it is evident that for the same starting and target points, the A* algorithm exhibits longer path planning times and generates longer paths. This observation highlights the advantages of the IIFDS algorithm over the A* algorithm in terms of faster response times and shorter planned paths. Particularly, when there are direct obstacles along the path, the execution time for the A* algorithm in path planning can increase significantly. For instance, when considering target points such as (10, 10, 9), both algorithms yield similar path lengths of approximately 17.854283 m and 19.102506 m, respectively. However, the time taken differs significantly, A* requiring 346.934560 s while IIFDS only needs 0.573924 s. This indicates a substantial difference of approximately 605×. When considering the four destination points, the length of the planned paths does not differ significantly. However, there is a significant difference in the time consumption.

Table 2. Performance comparison of different algorithms in static obstacle scenarios, and all bolded data represents extreme values.

Destination	Time (s)		Length (m)		Mean Direction Angle (deg)		Max Direction Angle (deg)	
	IIFDS	A*	IIFDS	A*	IIFDS	A*	IIFDS	A*
(10,10,10)	1.014391	131.837413	18.565024	19.127552	0.543250	8.4375	1.867074	49.867805
(10,10,9)	0.573924	346.934560	17.854283	19.102506	0.562810	15.309670	1.781431	49.867805
(10,9,10)	0.672776	36.556171	17.743238	18.147479	0.489524	9.000000	1.845807	49.867805
(9,10,10)	0.568327	45.466717	18.192898	18.214942	0.702289	12.649041	2.781137	67.500000

From Figure 10 and Table 2, it is evident that for all anticipated planned paths, the A* algorithm exhibits larger turning angles and more pronounced changes in direction. These large turning angles can lead to severe control oscillations, thereby significantly impacting the stability of the control system. Furthermore, such abrupt fluctuations can also cause damage to the actuation mechanism. On the other hand, the IIFDS path planning algorithm generates smoother and more seamless paths, which are characterized by fewer abrupt changes in direction.

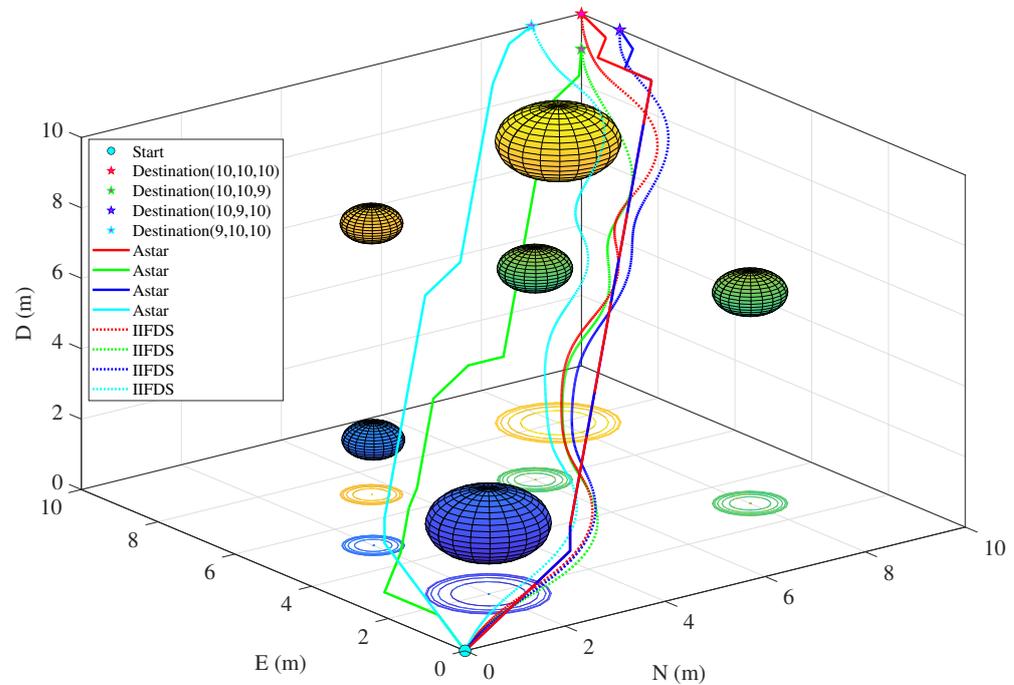


Figure 10. Comparison of path planning of A* and IIFDS methods. All solid lines in the figure represent paths planned by the Astar method, while dashed lines represent paths planned by the IIFDS method. The colors indicate different endpoints: red represents the endpoint (10,10,10), green represents the endpoint (10,10,9), blue represents the endpoint (10,9,10), and cyan represents the endpoint (9,10,10).

Based on the observations from Figure 10 and Table 2, the following conclusions can be drawn. The IIFDS path planning algorithm exhibits better real-time performance, as it generates shorter paths with higher smoothness. These characteristics make it more suitable for real-time control applications.

4.2. Dynamic Obstacle Avoidance and Path Planning

The IIFDS algorithm is designed to address dynamic obstacles by enabling the real-time monitoring of their positions and velocities. Based on its own position, the algorithm plans reference waypoints for the next control cycle. Different IIFDS parameters yield

varying results in terms of path length and obstacle avoidance effectiveness. Therefore, in this scenario with the same start/end points and dynamic obstacles, different IIFDS parameters are set to assess their impact on path planning. This analysis aims to identify the most suitable parameters for practical obstacle avoidance environments. The start and end coordinates remain as (0,0,0) and (10,10,10), respectively.

The dynamic obstacle environment consists of three spherical obstacles located initially at (5,5,7.5), (3,2,6), and (8,8.5,8.5), with radii of (0.5,0.4,0.5), respectively. These obstacles move at a linear velocity of 2 m/s along the x-axis with cosine direction variation and 1 m/s along the y-axis with sine direction variation, while maintaining a fixed position along the z-axis. The IIFDS path planning controller is configured with five different parameter sets, as in Table 3. Each of the aforementioned parameter sets produces a distinct planned path (path1, path2, path3, path4, path5).

Table 3. Different IIFDS parameters for dynamic obstacles scenario of path planning.

Parameters	Dynamic Coefficient λ	Reaction Coefficient ρ	Reaction Coefficient σ	Tangential Reaction Coefficient θ
Path1	5	1.5	0.5	0.1
Path2	15	1.5	0.5	0.1
Path3	5	0.5	0.5	0.1
Path4	5	1.5	1.5	0.1
Path5	5	1.5	0.5	0.5

Based on Figure 11 and Table 4, it is evident that among the five parameter sets, path3 stands out with the shortest path length of 17.334920 m. With the exception of path4, which experiences a substantial increase in length, the path lengths of the remaining four sets are approximately equal. Thus, it can be inferred that the parameter σ significantly affects the path length, whereas the other parameters have a relatively minor influence on it. Regarding planning time, path3 has the shortest duration, 24.268841 s, suggesting that the parameter ρ has a significant effect on the planning time. Furthermore, as the parameter value decreases, the planning time decreases accordingly. The path labeled “path3” exhibits both the lowest average angle, 0.1617 deg, and the smallest maximum angle, 1.200126 deg, of direction. Referring to Figure 11, it is evident that a significant alteration occurs in the path when the reaction coefficient shifts from 0.5 to 1.5. The change in this particular parameter is much more pronounced compared to the three other parameters, which also underwent a threefold change. Hence, it becomes apparent that a more gradual adjustment is necessary for this parameter.

Table 4. Performance comparison of different algorithms in dynamic obstacle scenarios, and all bolded data represents extreme values.

Parameters	Length (m)	Time (s)	Mean Direction Angle (deg)	Max Direction Angle (deg)
(Path1) $\lambda = 5, \rho = 1.5, \sigma = 0.5, \theta = 0.1$	17.488965	31.574731	0.736420	9.629480
(Path2) $\lambda = 15, \rho = 1.5, \sigma = 0.5, \theta = 0.1$	17.512065	31.236525	0.867715	9.679838
(Path3) $\lambda = 5, \rho = 0.5, \sigma = 0.5, \theta = 0.1$	17.334920	24.268841	0.161700	1.200126
(Path4) $\lambda = 5, \rho = 1.5, \sigma = 1.5, \theta = 0.1$	19.144417	31.804764	0.965932	6.882223
(Path5) $\lambda = 5, \rho = 1.5, \sigma = 0.5, \theta = 0.5$	17.480835	30.959117	0.753329	9.309696

4.3. Parameter Selection and Trajectory Tracking

From the paths planned in the above scenarios, a typical path is selected correspondingly to use the NMPC controller for tracking control, to test the influence of various variable parameters on the tracking effect. The NMPC is utilized for tracking planned paths in environments with both dynamic and static obstacles. The cost function is designed to incorporate various objectives, including position deviation, velocity deviation, and control input magnitude. Different weight parameters are set to compare the tracking performance.

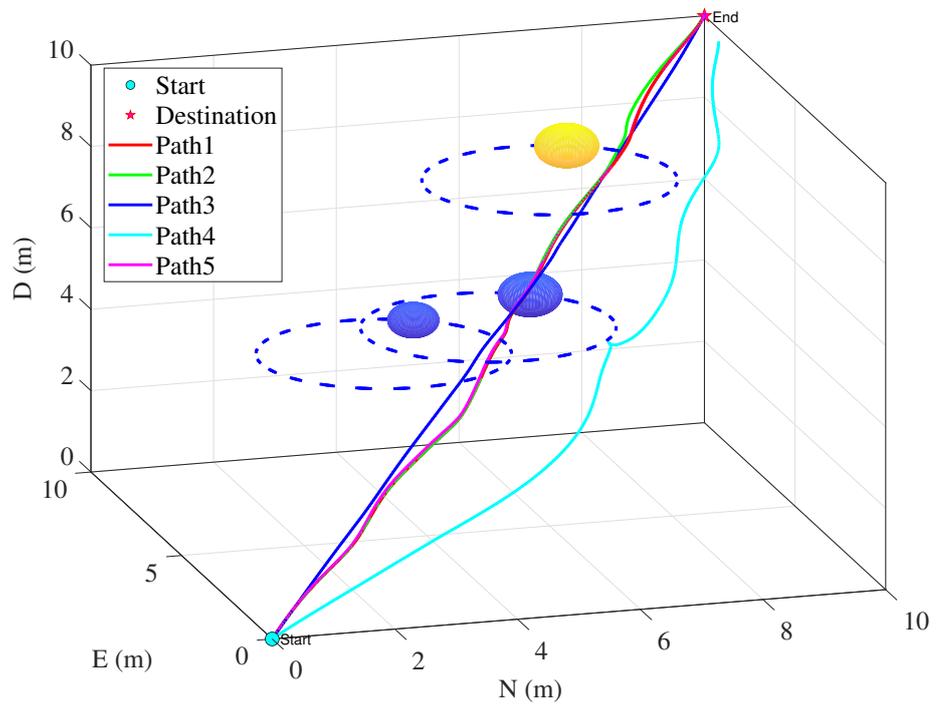


Figure 11. Different parameters of IIFDS for path planning, the paths of different colors correspond to different planning parameters as indicated in Table 4.

Moreover, the prediction horizon directly influences the performance of the control system and computational burden. The open-loop sampling time is generally required to cover the transient response of the open-loop system. According to Figure 9 in the referenced literature [30], the open-loop response time of the BlueRov2 system is approximately 5 steps.

According to the MATLAB interpretation [31], providing a prediction horizon of 1–2 times the open-loop response time is deemed appropriate. Simulation results indicate that the computational time for 10 prediction horizons increases by nearly 2 times, but the performance remains essentially unchanged; it is suitable to set the horizon to $N = 5$ for all scenarios.

The initial state $x_0 = \mathbf{0}_{12 \times 1}$, the initial control input is $u = \mathbf{0}_{8 \times 5}$, in which 5 represents the prediction horizon, the upper and lower bounds for the control input are set to 40 and -40 respectively, and the state variable is constrained within the range of 0 to 10. According to reference [20] and the Equation (40), it is evident that a higher weight imposes a stronger constraint on the corresponding variable, resulting in a smaller range of variation. Therefore, by setting different numerical values, the impact of various weights on the simulation results is compared, and an appropriate weight is chosen as an experimental parameter.

Case 1. Different weight parameters for velocity and drive force.

Different weight coefficients ω_2, ω_3 are selected to compare the variations in velocity and control inputs during the path tracking process. The evaluation metrics include maximum values and stability.

Here, the planning path IIFDS path3 is chosen as the reference path. The velocity weight matrix ω_2 is set to $\mathbf{90}_{6 \times 1}, \mathbf{30}_{6 \times 1},$ and $\mathbf{10}_{6 \times 1}$, while the control input weight matrix is set to $\mathbf{0.2}_{6 \times 1}, \mathbf{1}_{6 \times 1}$ and $\mathbf{5}_{6 \times 1}$, respectively. These weight ranges are selected based on reference [20] and empirical knowledge.

Figure 12 illustrates the impact of different weights on velocity. It is evident that when $\omega_2 = \mathbf{90}_{6 \times 1}$, the velocities u and r are minimized, but other velocity components, particularly v , exhibit significant fluctuations. In contrast, with a weight coefficient of

$\omega_2 = 10_{6 \times 1}$, the constraints on velocity are relatively lax, often resulting in abrupt transitions from one extreme to another. Therefore, among the three parameters mentioned, $\omega_2 = 30_{6 \times 1}$ demonstrates the most stable behavior.

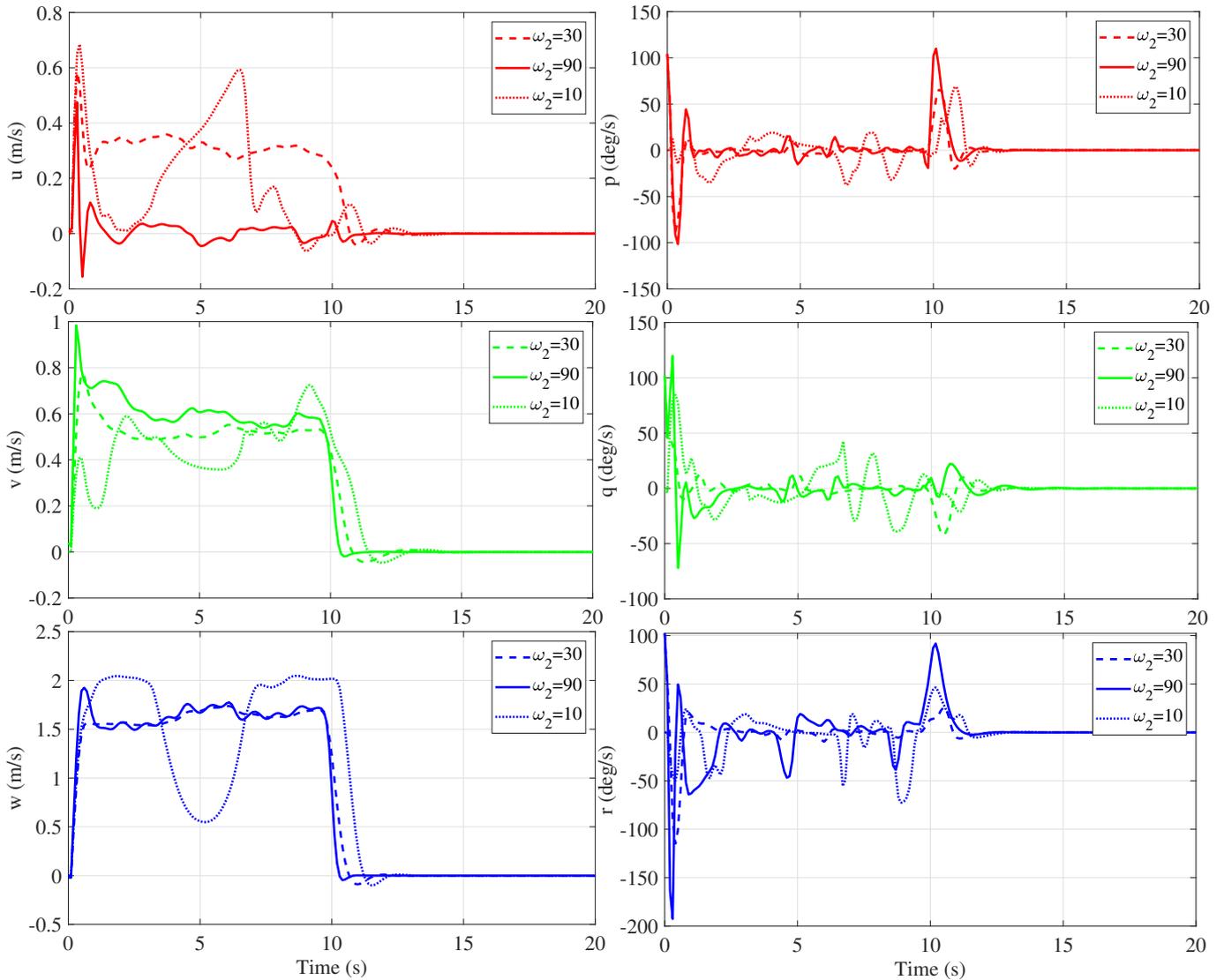


Figure 12. The impact of variable weights on velocity.

Figure 13 depicts the influence of weights on driving force. For $\omega_3 = 0.2_{6 \times 1}$, the driving forces experience minimal constraints, leading to both substantial fluctuations and frequent reaching of limit values. In the case of $\omega_3 = 5_{6 \times 1}$, the driving forces are subject to maximum constraints, yet it is noteworthy that both the minimum and maximum driving forces occur under this parameter scenario among the three weights. It is observed that at $\omega_3 = 1_{6 \times 1}$, all driving forces exhibit relatively smooth control processes.

Case 2. Different weight parameters for tracking the path planned using A*.

Among them, the static path is selected to track the path towards the target point (10, 10, 10) using A*. As shown in Table 5 and Figure 14, it can be observed that for different control parameters, the average path deviation is minimized when ω_1 is set to 2000, with a value of 0.01384 m. The maximum path tracking deviation is 0.068 m. As the ω_1 value decreases, the tracking performance deteriorates. Among the paths, path4 exhibits the worst tracking performance, with a path tracking deviation of 0.4282 m when ω_1 is set to 50. The maximum path tracking deviation for this case is 1.429 m. Furthermore, from

the four tracked paths, it is evident that NMPC has the ability to anticipate paths with larger turning angles and generate smoother angular trajectories. From Figure 15, it can be observed that the position and orientation stabilize around the 13th s. Similarly, from Figure 16, the velocity and angular velocity also stabilize around the 13th s. Furthermore, as shown in Figure 17, the propulsive forces reach stability around the 13th s. All three variables exhibit mutual consistency. Additionally, the eight driving force variables remain stable between -40 and 20 throughout the execution process, consistently within the range of positive and negative 40 .

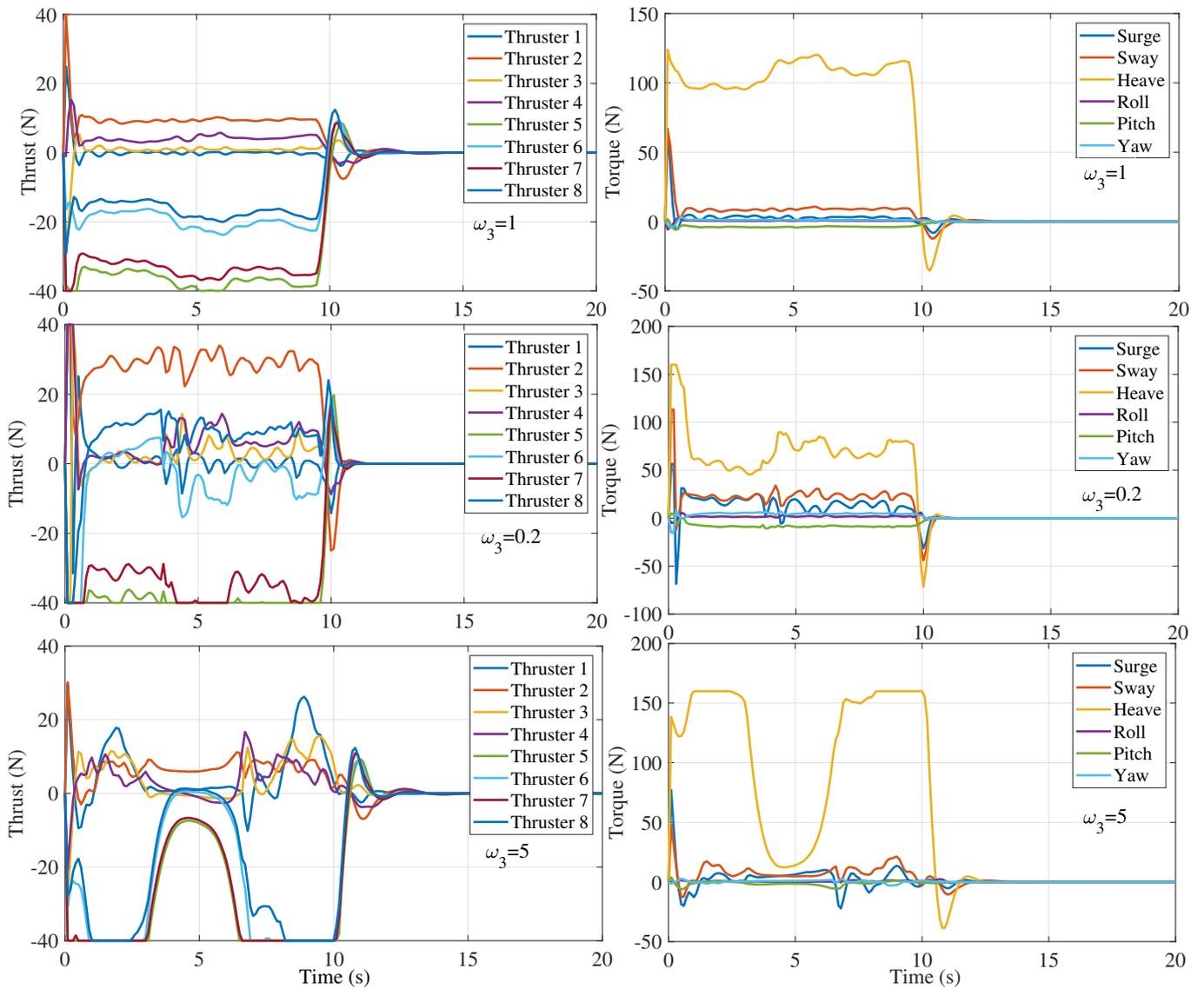


Figure 13. The influence of different weights on driving force.

Table 5. Different NMPC parameters for trajectory tracking.

Path	Parameters	Mean Variance (m) A*/IIFDS	Max Variance (m) A*/IIFDS
path1	$\omega_1 = 2000_{6*1}$ $\omega_2 = 30_{6*1}$ $\omega_3 = 1_{8*1}$	0.01384/0.01386	0.068/0.058
path2	$\omega_1 = 1000_{6*1}$ $\omega_2 = 30_{6*1}$ $\omega_3 = 1_{8*1}$	0.04752/0.01586	0.3867/0.072
path3	$\omega_1 = 400_{6*1}$ $\omega_2 = 30_{6*1}$ $\omega_3 = 1_{8*1}$	0.18369/0.01628	0.9737/0.082
path4	$\omega_1 = 50_{6*1}$ $\omega_2 = 30_{6*1}$ $\omega_3 = 1_{8*1}$	0.4282/0.01725	1.429/0.087

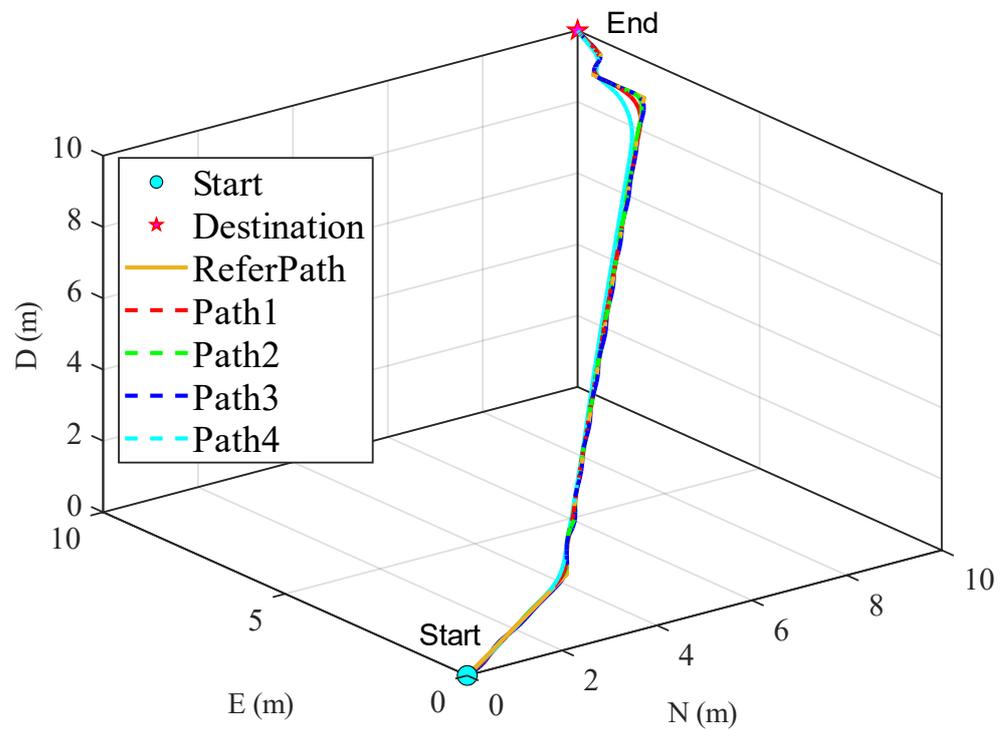


Figure 14. Comparison of different weight parameters for A* planned path tracking, the paths of different colors correspond to different weight coefficients in Table 5.

Case 3. Different weight parameters for tracking the path planned by IIFDS.

The path for tracking the dynamic obstacle was selected from Table 3, specifically path3. Different $\omega_1, \omega_2, \omega_3$ parameters were chosen from Table 5 to track the path. The tracking results can be seen in Figure 18 and Table 5. It is evident that the parameter ω_1 has a negligible impact on the tracking performance.

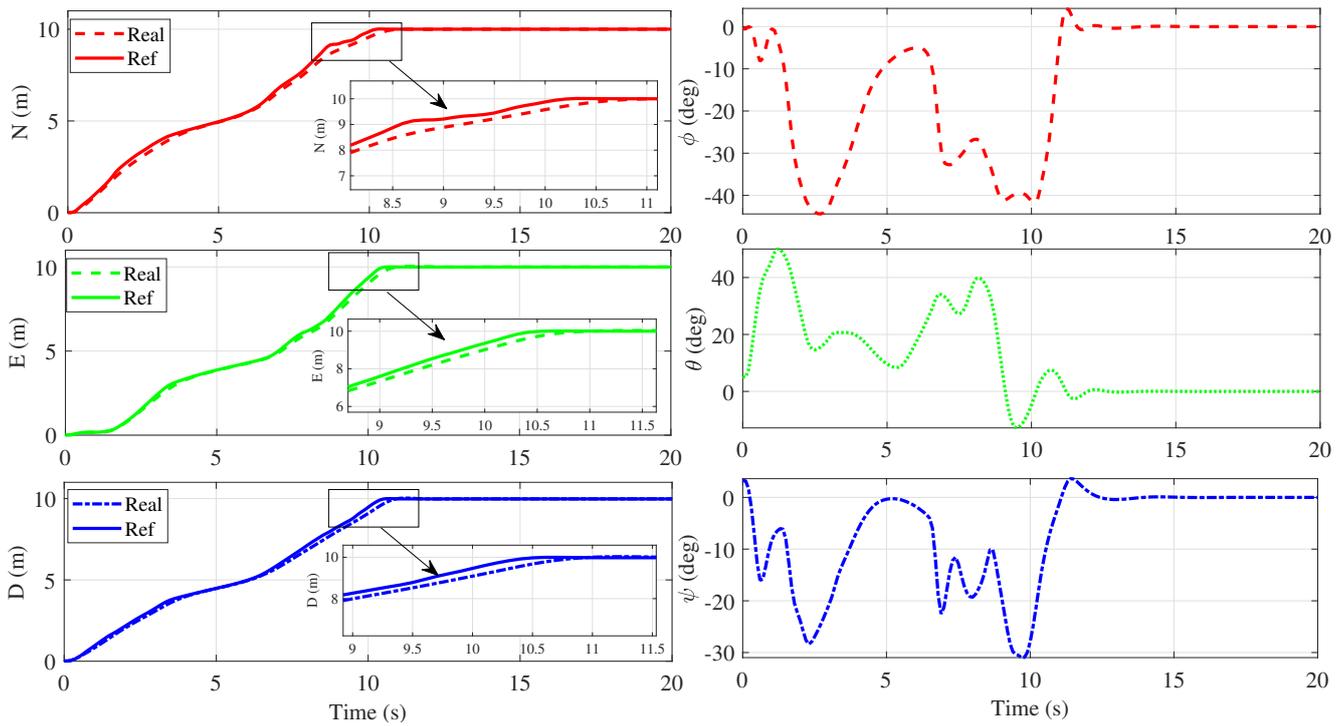


Figure 15. Position of the trajectory tracking for A* Path, the red, green, and blue on the left represent position N, E, D respectively, and on the right, the red, green, and blue represent attitude angle ϕ , θ , ψ respectively.

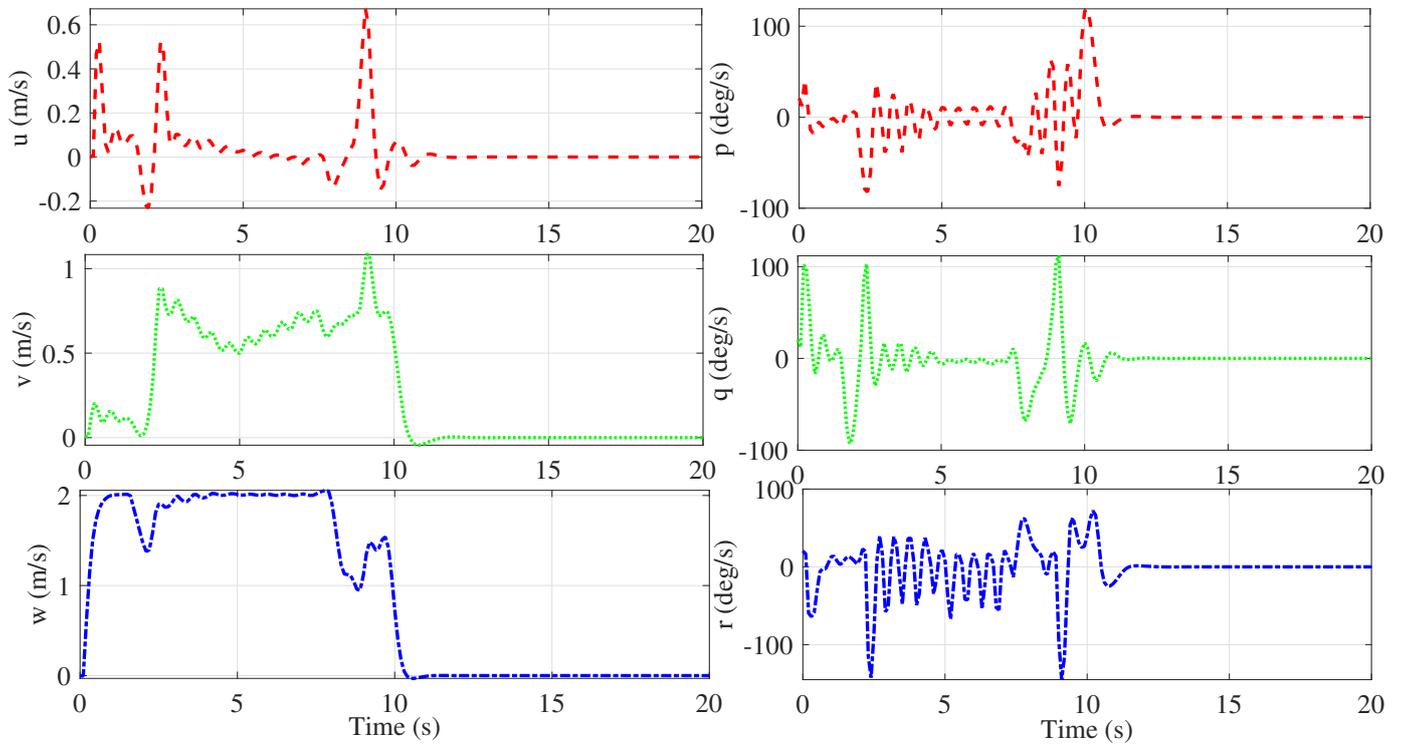


Figure 16. Velocity of the trajectory tracking for A* Path, the red, green, and blue on the left represent linear velocities u , v , w respectively, and on the right, the red, green, and blue represent angular velocities p , q , r respectively.

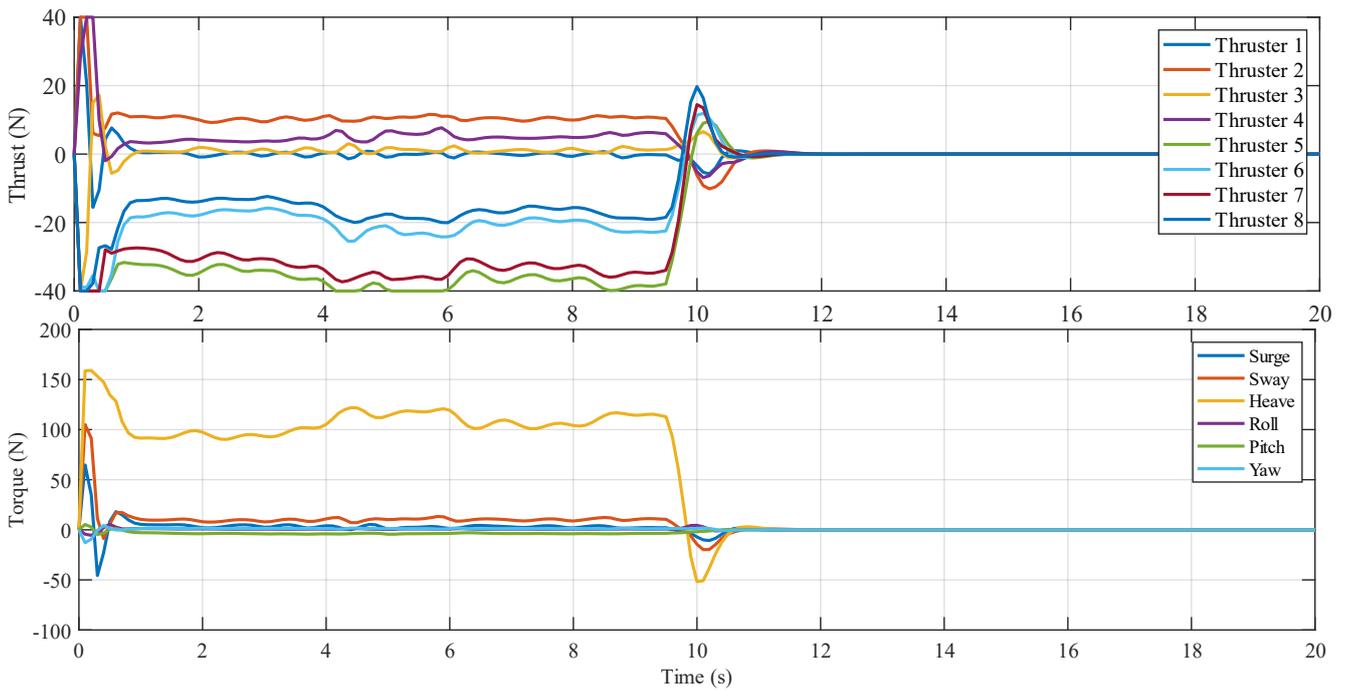


Figure 17. Thruster of the trajectory tracking for A* path.

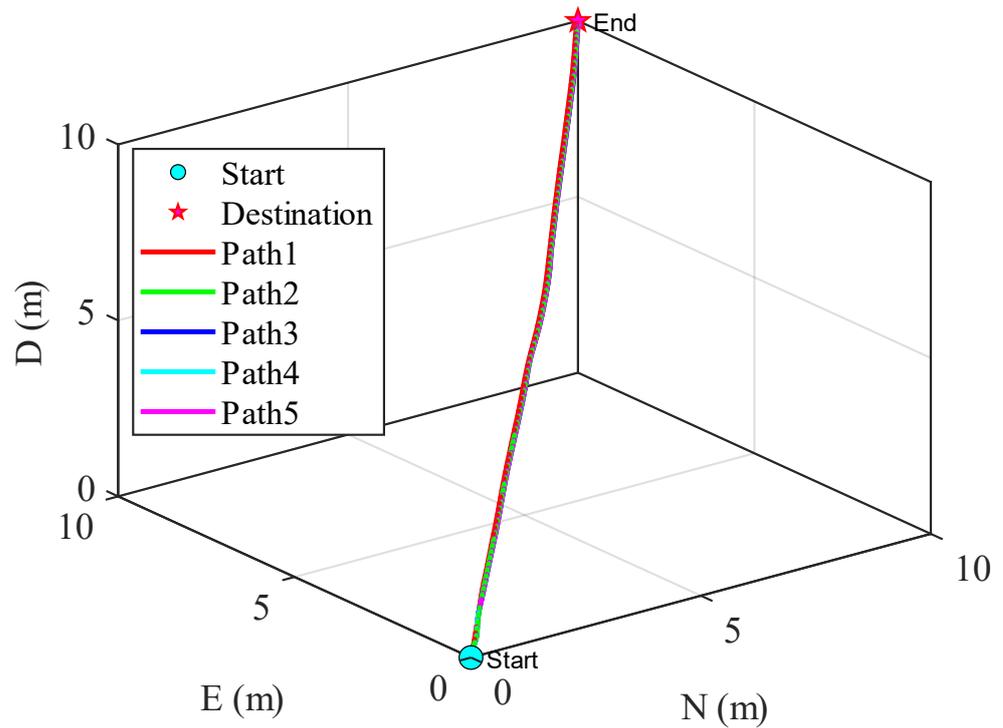


Figure 18. Comparison of different weight parameters for IIFDS planned path tracking, the paths of different colors correspond to different weight coefficients in Table 5.

We selected the best-performing set of tracking results from all the aforementioned paths and observed the various state variables and control quantities during the tracking process. From Figures 15 and 19, it can be observed that the position exhibits a nearly linear variation during the tracking process. The attitude quickly adjusts to the desired angle at the beginning of the tracking process and remains relatively stable throughout. Similarly, as seen in Figures 16 and 20, the velocity and angular velocity rapidly adjust to their operating values and maintain constant values. Upon approaching the vicinity of the target point, they quickly return to zero. As seen in Figures 17 and 21, the thrust force, constrained by a maximum value, initially undergoes several adjustments to reach the maximum value of 40 N and then swiftly returns to a constant value [32]. After reaching the target point, a reverse force may appear to adjust the attitude, which aligns with the control logic.

Considering the overall situation, a larger weight coefficient imposes greater constraints on the corresponding quantity. However, a balance needs to be struck between the magnitude and stability of the corresponding quantity. Based on the above description, the parameter $\omega_1 = 2000$ exhibits excellent performance in path tracking. The weight coefficients $\omega_1 = 2000_{6 \times 1}$, $\omega_2 = 30_{6 \times 1}$, and $\omega_3 = 1_{6 \times 1}$ represent a relatively well-performing set of weights.

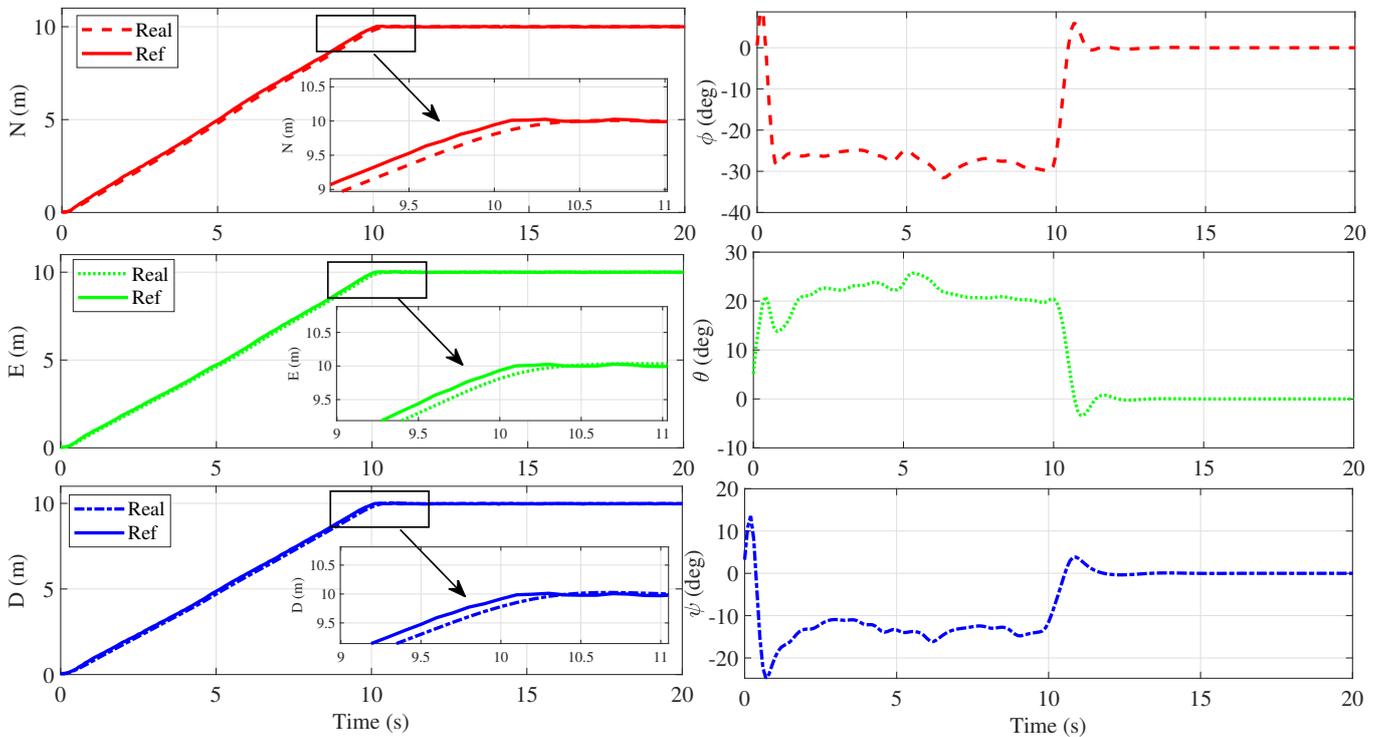


Figure 19. Position of the AUV in the tracking process, the red, green, and blue on the left represent linear velocities N, E, D respectively, and on the right, the red, green, and blue represent angular velocities ϕ , θ , ψ respectively.

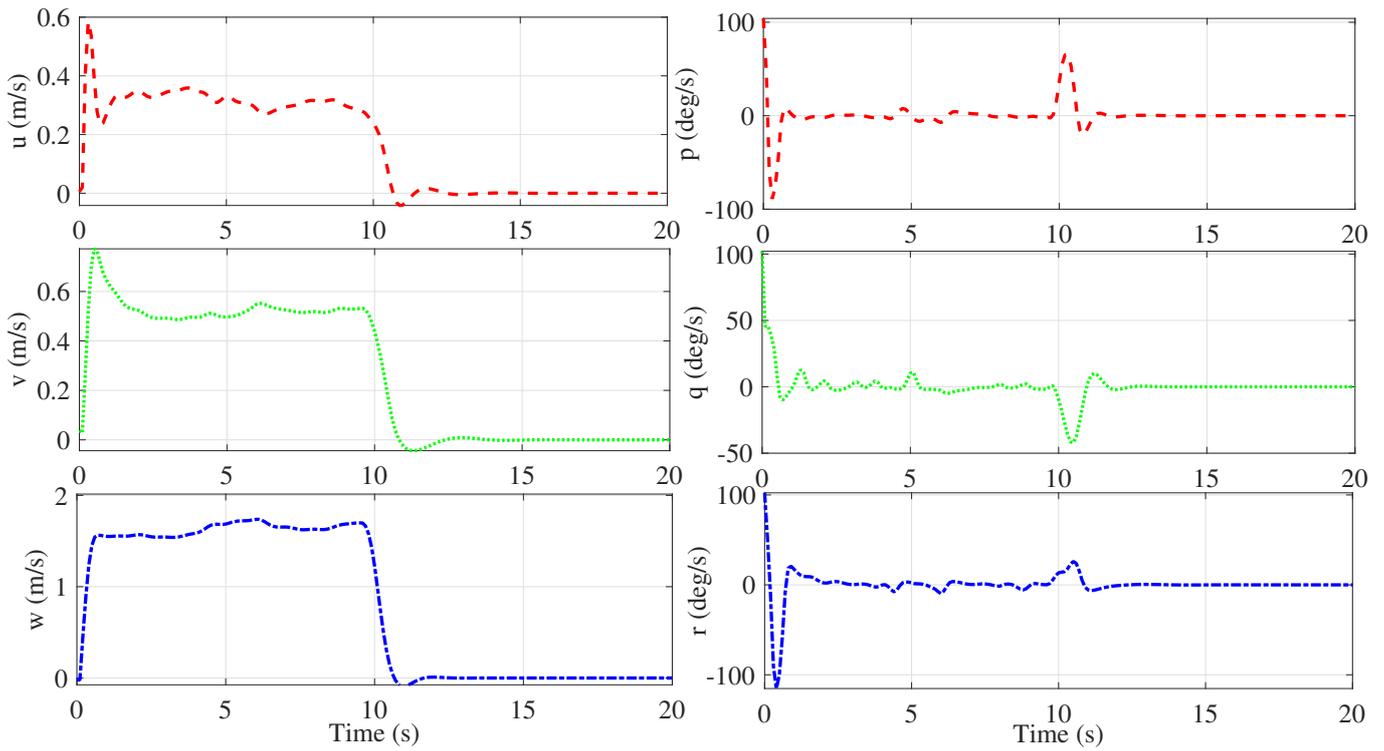


Figure 20. Velocity of the AUV in the tracking process, the red, green, and blue on the left represent linear velocities u , v , w respectively, and on the right, the red, green, and blue represent angular velocities p , q , r respectively.

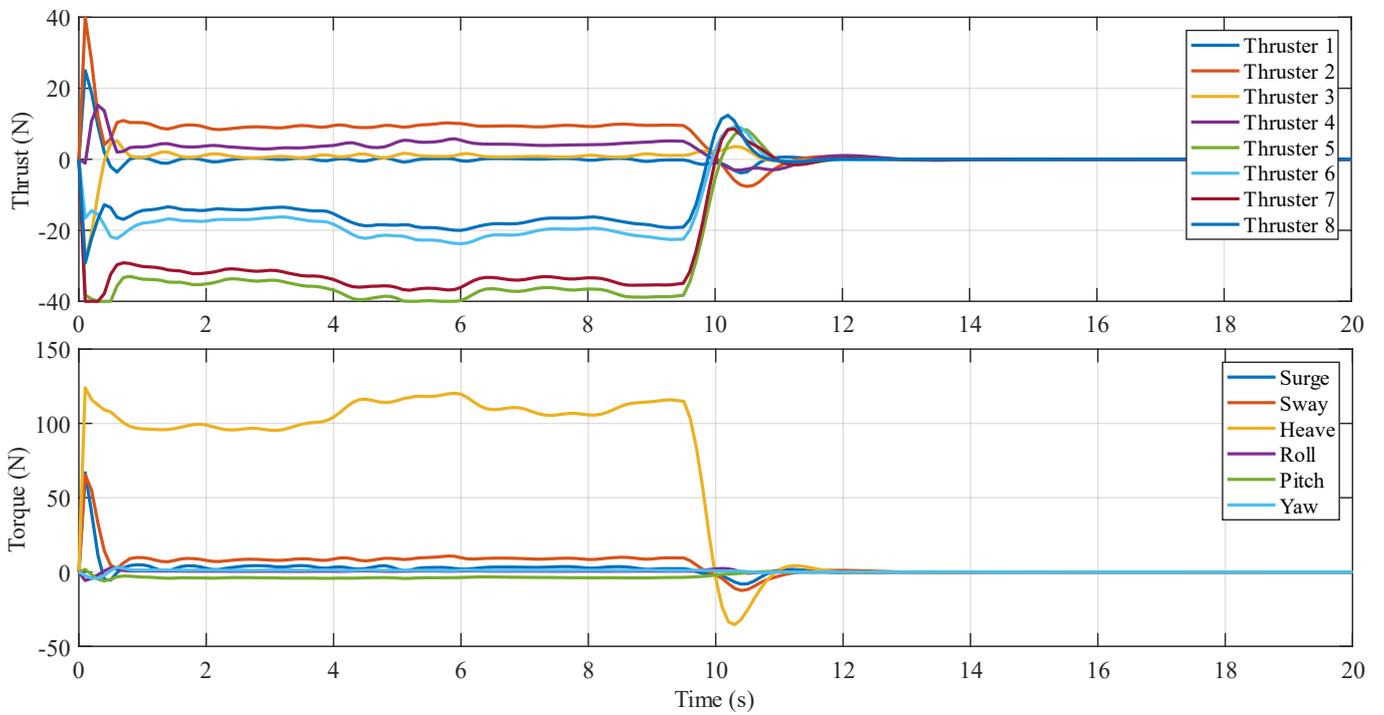


Figure 21. Thruster of the AUV in the tracking process.

5. Experimental Results

To validate the effectiveness of the designed IIFDS-NMPC control framework in practical applications, we conducted the following test scenario. In a water tank measuring 10 m long, 3 m wide, and 1.2 m deep, six obstacles were randomly placed. Among them, four were static obstacles, while the other two were dynamic obstacles moving in a straight line at a velocity of $v_{obs} = 1$ m/s. All obstacles are considered to be spheres with a radius of 0.25 m. The positioning of the AUV and obstacles was carried out using the approach described in [33], which combines image- and radar-based localization systems. The positioning system and the AUV system were independent systems that communicated with each other through a wireless serial port connection.

The AUV’s main structure followed that of BlueRov2-heavy, with the main control unit replaced by a self-designed control board based on STM32F407 and NVIDIA NX2.

The central processor in NX2 is a 6-core NVIDIA CarmelArmV 64-bit CPU. An STM32 unit is responsible for gathering all external environmental information and transmitting it to the CPU. The CPU, based on this information, executes the hybrid layer control structure designed in this paper, namely the IIFDS-NMPC control flow. The Apollo platform incorporates three solvers (SQP, Interior Point OPTimizer (IPOT), Operator Splitting Quadratic Programming (OSQP)), with OSQP as the default choice. This preference is driven by its faster solution-finding capabilities, as it does not account for curvature constraints. However, a significant drawback is its inability to guarantee curvature continuity for smoothed points. As a result, the AUV system seeks a compromise between computation time and accuracy, choosing the SQP solver. Utilizing the built-in optimal solver SQP in Python + Pyomo to solve the optimal control with constraints, it generates control commands and sends the first set of control commands to the STM32 interface board. Subsequently, the interface board forwards the control commands downstream to the drive unit. In this process, the STM32 serves as the interface board, interacting with all peripherals and transmitting data to the processor. Meanwhile, the role of the NVIDIA NX2 is to perform path planning and calculate the optimal control quantities for NMPC. The hardware architecture diagram is shown in Figure 22, and the optimization algorithm for NMPC used to obtain control signals and the IIFDS were implemented as in Figure 23. The AUV was powered by batteries during the tests, and the control operated at a frequency of 100 Hz.

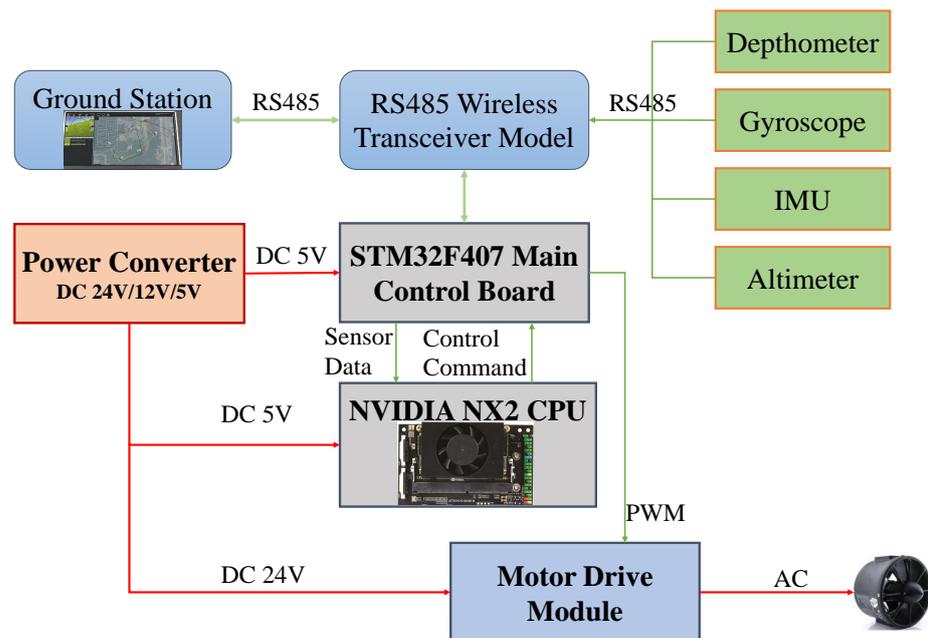


Figure 22. Hardware control system block diagram for AUV.

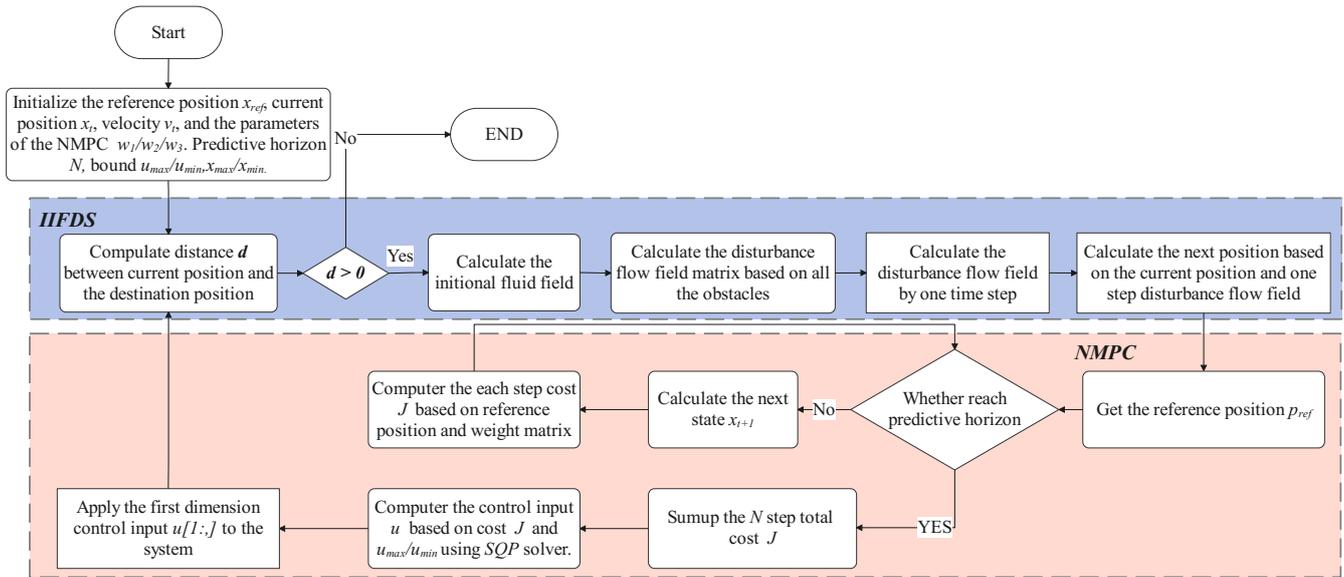


Figure 23. Path planning and tracking control flow chart.

As the center of the AUV is located at a depth of 0.5 m, the obstacles are set at a fixed depth of 0.5 m, and the obstacle avoidance parameter θ for IIFDS was set to 0. The other parameters were set according to the optimal simulation parameters for path3 in Table 3, with $\lambda = 5$, $\rho = 0.5$, and $\sigma = 0.5$. The weight parameters for NMPC were set according to path1 in Table 5, with $\omega_1 = 2000_{6 \times 1}$, $\omega_2 = 30_{6 \times 1}$, and $\omega_3 = 1_{8 \times 1}$. The choice in prediction horizon ($N = 5$) has a crucial impact on the actual control cycle in practical applications. Therefore, through comparative measurements, a prediction horizon of $N = 5$ was determined.

The starting point was $(0, 0, 0)$, and the target point relative to the starting point had coordinates of $(8, 2, 0)$. The initial speed of the AUV was 1 m/s. The actual path planning and tracking results are shown in Table 6 and Figure 24, respectively.

Figure 25 illustrates the position and orientation measurements, which reach the target position around the 9th s. According to the collected data, the RMS of the path tracking is calculated to be 0.01867 m. Additionally, based on feedback from the microcontroller unit (MCU), the actual average computation period is determined to be 0.0063 s, which significantly satisfies the control frequency requirement of 100 Hz. Moreover, as the number of obstacles increases, the theoretical computation period is expected to further increase, providing additional redundancy for obstacle calculations. From Figures 26 and 27, it can be observed that the velocity and propulsive force also reach stable values around the 9th s. The orientation measurements, velocity measurements, and propulsive forces are synchronized in time. Furthermore, the propulsive forces remain within the range of positive and negative 40 N, while the torque stays within the range of positive and negative 160 N, complying with the actual constraints of the driver. The actual experimental video can be obtained through the following link: <https://youtu.be/TyMuQIc91a8> accessed on 7 October 2023.

Table 6. Real navigation and control result.

Path Tracking Deviation RMS (m)	Average Computation Period (s)
0.01867	0.0063

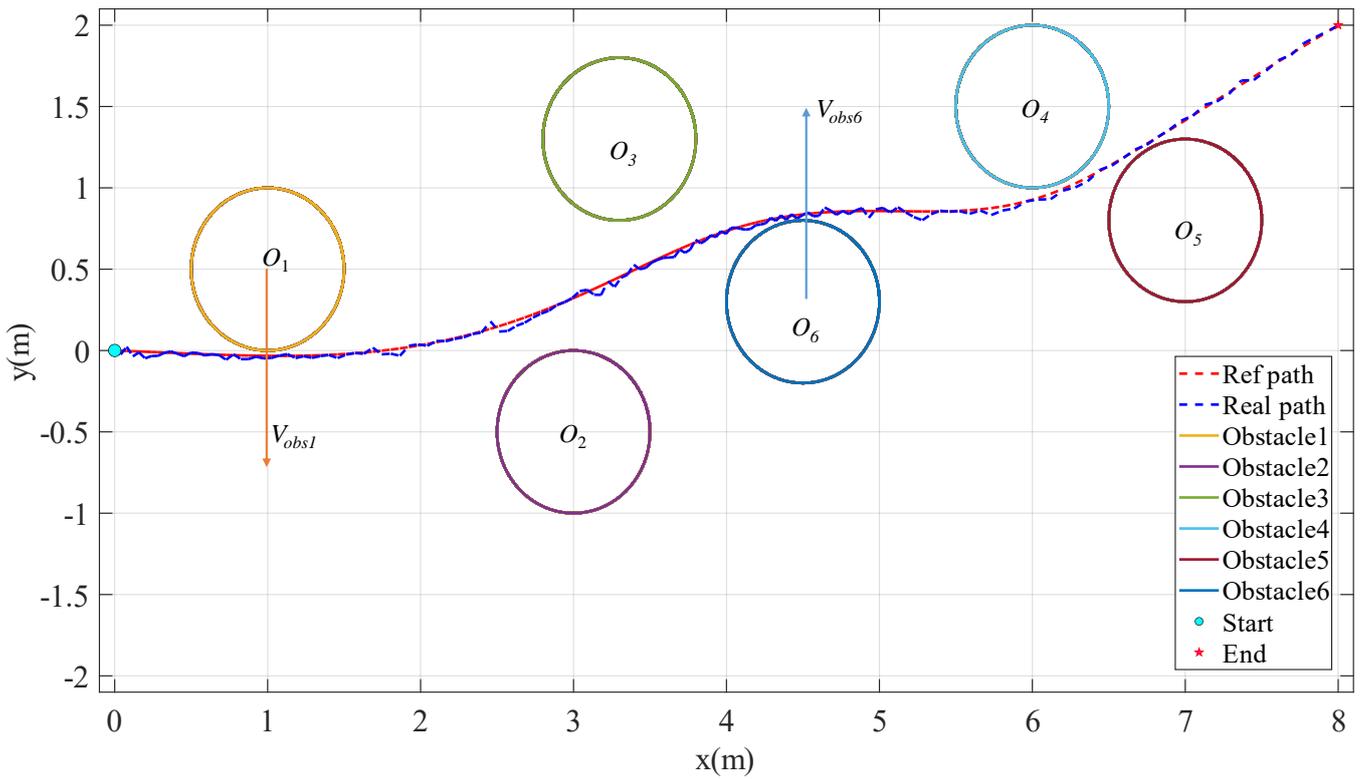


Figure 24. Real path planning and tracking in water pool.

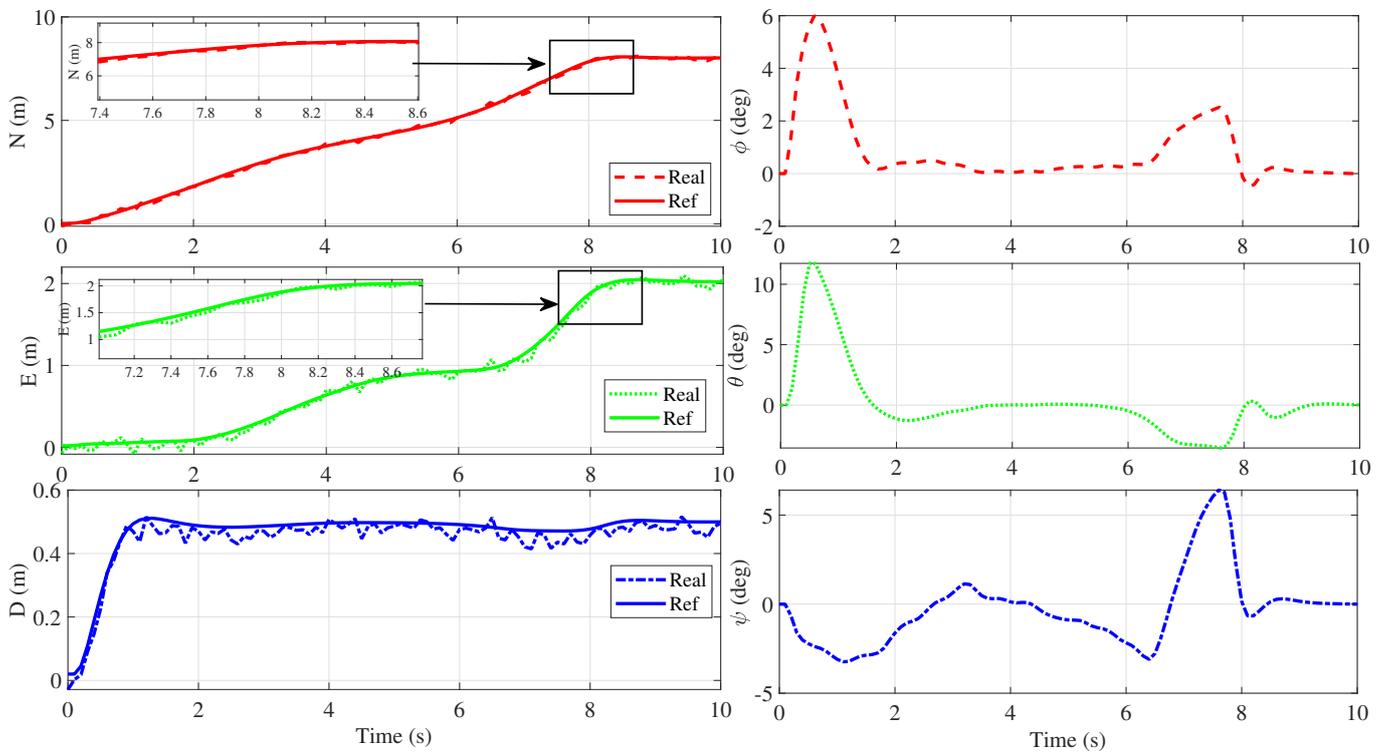


Figure 25. Position of the AUV during tracking process in water pool, the red, green, and blue on the left represent linear velocities N , E , D respectively, and on the right, the red, green, and blue represent angular velocities ϕ , θ , ψ respectively.

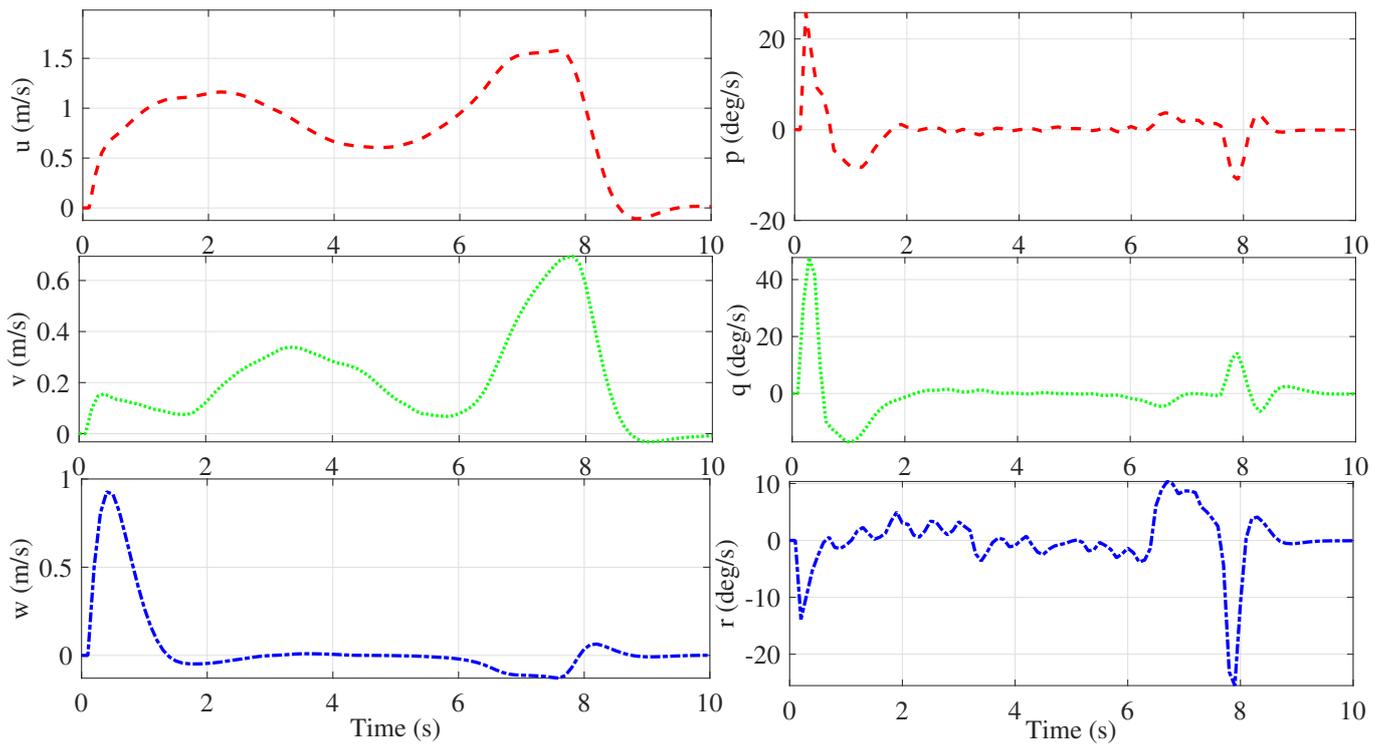


Figure 26. Velocity of the AUV during tracking process in water pool, the red, green, and blue on the left represent linear velocities u , v , w respectively, and on the right, the red, green, and blue represent angular velocities p , q , r respectively.

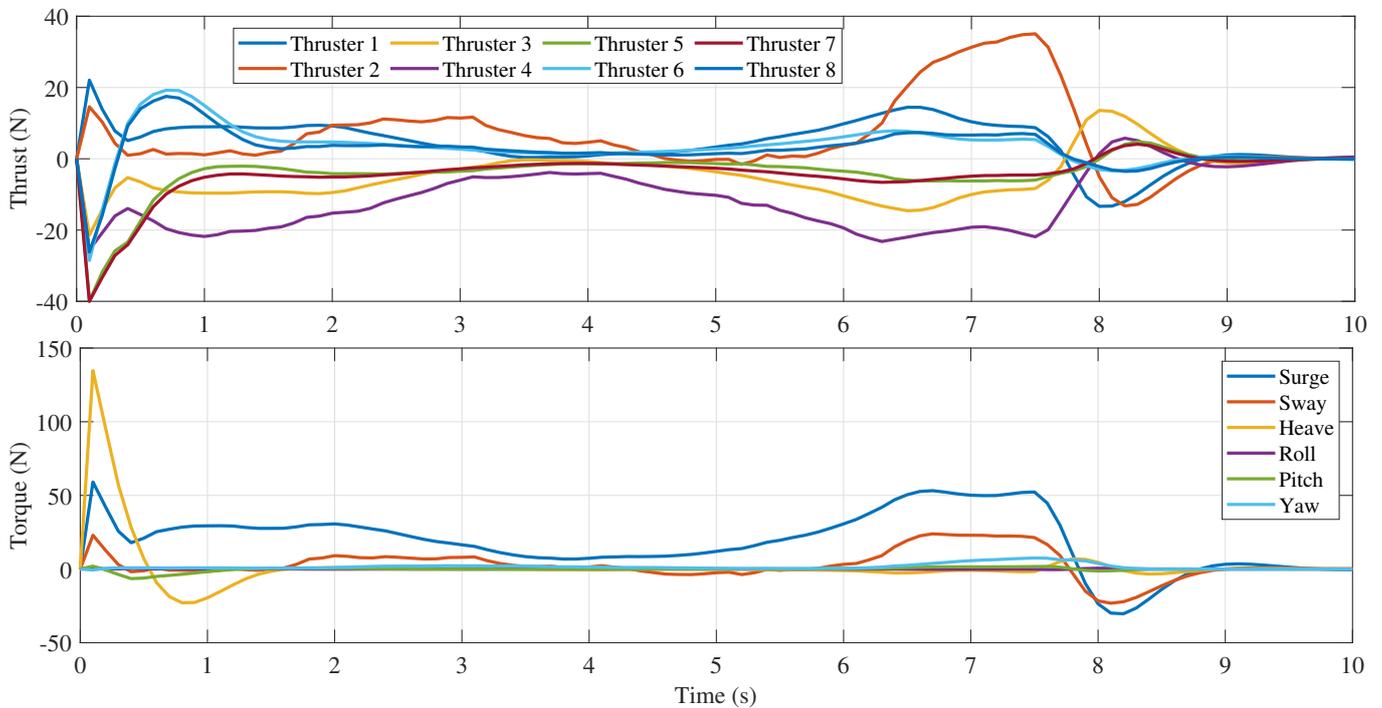


Figure 27. Thruster of the AUV during tracking process in water pool.

6. Conclusions

In this study, we proposed a dual-layer control strategy, IIFDS-NMPC, for path planning and tracking control in complex obstacle environments. The upper layer utilized IIFDS for path planning, while the lower layer employed NMPC for trajectory tracking. By simulating dynamic and static obstacle environments, we selected the planning parameters for IIFDS to generate paths that are as short as possible while minimizing the time taken, and the planned path was the smoothest. NMPC weights have a direct impact on the tracking deviation and control command during the tracking process. Through comparative experiments, the selected parameters are $\omega_1 = 2000$, $\omega_2 = 30$, and $\omega_3 = 1$, ensuring the path tracking followed the planned trajectory as closely as possible, while keeping the control and state variables within their respective limits. Through real platform testing experiments using the BlueRov2, the designed dual-layer control strategy fulfilled the maximum control requirement of 100 Hz, and the path tracking accuracy was within a range of 0.02 m. This control strategy meets the requirements for practical applications.

Although the path planning strategy, IIFDS, attempted several fixed measurements that proved suitable for the application, the parameters of this planner remained unchanged throughout the entire planning process and did not consider dynamic adjustments based on the AUV's position and the changing external obstacle environment. These parameters are not optimal. Therefore, future research will focus on dynamically adjusting the optimal parameters.

Author Contributions: Conceptualization, J.D.; methodology, J.D. and S.A.; software, D.Z. and J.D.; validation, J.D.; formal analysis, J.D.; investigation, J.D.; resources, J.D.; data curation, J.D.; writing—original draft preparation, D.Z. and J.D.; writing—review and editing, S.A. and D.Z.; visualization, J.D.; supervision, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JST SPRING, Grant Number JPMJSP2109.

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

1. Jalal, F.; Nasir, F. Underwater navigation, localization and path planning for autonomous vehicles: A review. In Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 12–16 January 2021; pp. 817–828.
2. Shen, C.; Shi, Y.; Buckham, B. Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach. *IEEE/ASME Trans. Mechatronics* **2016**, *22*, 1163–1173. [[CrossRef](#)]
3. Lim, H.S.; King, P.; Chin, C.K.; Chai, S.; Bose, N. Real-time implementation of an online path replanner for an AUV operating in a dynamic and unexplored environment. *Appl. Ocean. Res.* **2022**, *118*, 103006. [[CrossRef](#)]
4. Vagale, A.; Oucheikh, R.; Bye, R.T.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles I: A review. *JMSE* **2021**, *26*, 1292–1306. [[CrossRef](#)]
5. Pollini, L.; Razzanelli, M.; Pinna, F. Development of the Guidance Navigation and Control System of the Folaga AUV for Autonomous Acoustic Surveys in the WiMUST Project. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–6.
6. Cheng, C.; Sha, Q.; He, B.; Li, G. Path planning and obstacle avoidance for AUV: A review. *Ocean. Eng.* **2021**, *235*, 109355. [[CrossRef](#)]
7. Sui, F.; Tang, X.; Dong, Z.; Gan, X.; Luo, P.; Sun, J. ACO+ PSO+ A*: A bi-layer hybrid algorithm for multi-task path planning of an AUV. *Comput. Ind. Eng.* **2023**, *175*, 108905. [[CrossRef](#)]
8. Ru, J.; Yu, S.; Wu, H.; Li, Y.; Wu, C.; Jia, Z.; Xu, H. A multi-AUV path planning system based on the omni-directional sensing ability. *JMSE* **2021**, *9*, 806. [[CrossRef](#)]
9. Yu, F.; Shang, H.; Zhu, Q.; Zhang, H.; Chen, Y. An efficient RRT-based motion planning algorithm for autonomous underwater vehicles under cylindrical sampling constraints. *Auton. Robot.* **2023**, *47*, 281–297. [[CrossRef](#)]
10. Wang, L.; Zhu, D.; Pang, W.; Zhang, Y. A survey of underwater search for multi-target using Multi-AUV: Task allocation, path planning, and formation control. *Ocean. Eng.* **2023**, *278*, 114393. [[CrossRef](#)]

11. Li, J.; Li, C.; Chen, T.; Zhang, Y. Improved RRT algorithm for AUV target search in unknown 3D environment. *JMSE* **2022**, *10*, 826. [[CrossRef](#)]
12. Zhou, L.; Wang, M.; Zhang, X.; Qin, P.; He, B. Adaptive SLAM Methodology Based on Simulated Annealing Particle Swarm Optimization for AUV Navigation. *Electronics* **2023**, *12*, 2372. [[CrossRef](#)]
13. Wu, J.; Wang, H.; Liu, Y.; Zhang, M.; Wu, T. Learning-based fixed-wing UAV reactive maneuver control for obstacle avoidance. *Aerosp. Sci. Technol.* **2022**, *126*, 107623. [[CrossRef](#)]
14. Wu, J.; Wang, H.; Zhang, M.; Yu, Y. On obstacle avoidance path planning in unknown 3D environments: A fluid-based framework. *ISA Trans.* **2021**, *111*, 249–264. [[CrossRef](#)] [[PubMed](#)]
15. Yao, P.; Zhao, S. Three-dimensional path planning for AUV based on interfered fluid dynamical system under ocean current. *IEEE Access* **2018**, *6*, 42904–42916. [[CrossRef](#)]
16. Li, Y.; Jiang, Y.Q.; Wang, L.F.; Cao, J.; Zhang, G.C. Intelligent PID guidance control for AUV path tracking. *J. Cent. South Univ.* **2015**, *22*, 3440–3449. [[CrossRef](#)]
17. Bae, S.B.; Shin, D.H.; Kwon, S.T.; Joo, M.G. An LQR controller for autonomous underwater vehicle. *JICRC* **2014**, *20*, 132–137.
18. Li, D.; Du, L. Auv trajectory tracking models and control strategies: A review. *J. Mar. Sci. Eng.* **2021**, *9*, 1020. [[CrossRef](#)]
19. Shen, C.; Shi, Y.; Buckham, B. Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5796–5805. [[CrossRef](#)]
20. Cao, Y.; Li, B.; Li, Q.; Stokes, A.A.; Ingram, D.M.; Kiprakis, A. A nonlinear model predictive controller for remotely operated underwater vehicles with disturbance rejection. *IEEE Access* **2020**, *8*, 158622–158634. [[CrossRef](#)]
21. Gill, P.E.; Murray, W.; Saunders, M.A. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.* **2005**, *47*, 99–131. [[CrossRef](#)]
22. Available online: Apollo studio <https://apollo.baidu.com/> (accessed on 17 August 2023).
23. SNAME[1950]. Nomenclature for treating the motion of a submerged body through a fluid. *Soc. Nav. Archit. Mar. Eng. Tech. Reserach Bull.* **1995**, 1–15.
24. Wu, C.J. 6-dof Modelling and Control of a Remotely Operated Vehicle. Ph.D. Thesis, Flinders University, College of Science and Engineering, Bedford Park, Australia, 2018.
25. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: New York, NY, USA, 2011.
26. Yang, R.; Clement, B.; Mansour, A.; Li, M.; Wu, N. Modeling of a complex-shaped underwater vehicle for robust control scheme. *JINT* **2015**, *80*, 491–506. [[CrossRef](#)]
27. Wu, J.; Wang, H.; Li, N.; Su, Z. Formation obstacle avoidance: A fluid-based solution. *IEEE Syst. J.* **2019**, *14*, 1479–1490. [[CrossRef](#)]
28. Luque, J.C.C.; Donha, D.C.; de Barros, E.A. AUV parameter identification. *IFAC Proc.* **2009**, *42*, 72–77. [[CrossRef](#)]
29. Zhang, Y.; Liu, X.; Luo, M.; Yang, C. MPC-based 3-D trajectory tracking for an autonomous underwater vehicle with constraints in complex ocean environments. *Ocean. Eng.* **2019**, *189*, 106309. [[CrossRef](#)]
30. Tanveer, A.; Ahmad, S.M. Cross-Coupled Dynamics and MPA-Optimized Robust MIMO Control for a Compact Unmanned Underwater Vehicle. *JMSE* **2023**, *11*, 1411. [[CrossRef](#)]
31. How to Use MPC. Available online: <https://www.youtube.com/watch?v=8U0xiOkDcmw&list=PLn8PRpmsu08ozoeoXgxPSBKLyd4YEHww8> (accessed on 18 August 2023).
32. Blue Robotics. Available online: <https://bluerobotics.com/store/rov/bluerov2/> (accessed on 5 August 2023).
33. Han, G.; Long, X.; Zhu, C.; Guizani, M.; Bi, Y.; Zhang, W. An AUV location prediction-based data collection scheme for underwater wireless sensor networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6037–6049. [[CrossRef](#)]

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.