

Article

Design, Modeling, and MPC-Based Control of a Fully Vectored Propulsion Underwater Robot

Tianzhu Gao ¹, Yudong Luo ^{1,*} , Na Zhao ¹, Yufu Gao ², Shengze Li ², Xianping Fu ¹, Xi Luo ³ and Yantao Shen ⁴¹ Information Science and Technology College, Dalian Maritime University, Dalian 116026, China; brain@dlmu.edu.cn (T.G.); zna@dlmu.edu.cn (N.Z.); fxp@dlmu.edu.cn (X.F.)² Marine Engineering College, Dalian Maritime University, Dalian 116026, China; cloudygao@dlmu.edu.cn (Y.G.); 1120241202@dlmu.edu.cn (S.L.)³ Yichang Testing Technology Research Institution, Yichang 443003, China; luoxi1234@gmail.com⁴ Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557, USA; ytshen@unr.edu

* Correspondence: ydluo@dlmu.edu.cn

Highlights

What are the main findings?

- A fully vectored propulsion underwater robot with an eight vector-distributed thruster layout is designed and experimentally validated, in which each thruster simultaneously contributes to both translational and rotational motion, thereby significantly enhancing maneuverability and motion flexibility.
- A MPC-based 6-DoF control framework, constructed using a simplified Fossen dynamic model with explicit input constraints, achieves synchronized position and attitude control in a real indoor tank, demonstrating high position accuracy (≤ 0.069 m), attitude accuracy ($\leq 2.41^\circ$), and robustness across waypoint tracking, continuous trajectory tracking, and artificial perturbations in the hovering state.

What is the implication of the main findings?

- The proposed mechatronic design for the robot's fully vectored propulsion and MPC-based 6-DoF control framework demonstrate a practically feasible and robust solution for autonomous underwater robots, providing a compact and agile platform suitable for underwater manipulation.

Abstract

This paper presents the design and implementation of a novel autonomous underwater robot with fully vectored propulsion based on model predictive control (MPC) to rapidly respond to the position and attitude required for autonomous operation. Specifically, the mechatronic design of the eight vector-distributed thruster layout for the robot's fully vectored propulsion is detailed, and the software architecture based on the robot operating system (ROS) is constructed. Then, the corresponding dynamics model is established by adopting the Fossen approach for the prediction and optimization of the control process. To achieve autonomous control, an MPC-based controller is designed and implemented to calculate the control input for the specified control objective. Finally, way-point tracking and trajectory-tracking experiments are carried out in an indoor tank equipped with a motion-capture system to validate the feasibility and effectiveness of the robot's design and control framework. In addition, the robustness of the robot system is verified by artificially perturbing the robot in the hovering state.



Academic Editor: Lu Liu

Received: 24 December 2025

Revised: 23 January 2026

Accepted: 30 January 2026

Published: 31 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: underwater robot; fully vectored propulsion; model predictive control

1. Introduction

At present, underwater robots are widely used in a variety of fields, such as hull inspection [1], marine ecosystem observation [2], aquaculture [3], archaeology [4], and scientific research [5], and have now transitioned from underwater monitoring to operations. As underwater operation tasks require robots to accurately manipulate targets in complex underwater environments, this introduces more technical challenges. In recent decades, remotely operated vehicles (ROVs) equipped with teleoperated manipulators have been the mainstay for underwater operations. However, this demands skilled operators and support from a mother vessel, which is costly and limits their operation range [6]. In contrast, autonomous underwater vehicles (AUVs) can reach a larger range and complete tasks autonomously without human intervention. However, traditional AUVs usually lack actuators for close-up manipulation. Therefore, developing underwater robots with autonomous manipulation capabilities can not only greatly reduce costs, but is also a future trend [7].

Currently, one of the main methods for addressing underwater operation tasks is to use underwater vehicle-manipulator systems (UVMS). In the well-known TRIDENT project, a floating underwater manipulation collaborative control framework was developed based on the task priority paradigm and dynamic programming technology using the Girona500 robot with a seven-degrees-of-freedom (7-DoF) underwater arm and dexterous hand, which successfully recovered the black box in the tank [8]. The Italian MARIS project developed an autonomous UVMS to increase the mission success rate and enhance robustness [9]. Although this provides a solution for underwater operation tasks, it also brings new issues. For example, the complex mechanical structure makes it difficult to establish an accurate dynamic model, and its motion-redundant system inherently increases the complexity of the controller. In addition, the coupling between the robotic arm and the robot body during operation will cause the robot's center of gravity to shift, introducing internal disturbances, and thus affecting the control performance [10].

Another research approach is to develop a highly maneuverable underwater robot that achieves the same manipulation abilities as a UVMS by utilizing its agile, omnidirectional 6-DoF motion. Reference [11] presents a micro-underactuated AUV with high agility, and experimentally demonstrates its maneuvering performance in confined environments based on the proposed navigation and control framework. Reference [12] proposes a hydrobatic AUV and demonstrates its large-angle maneuvering capability through both simulations and experiments. However, despite the improved agility achieved by these designs, the underactuated structure inherently limits the robot's ability to realize fully omnidirectional motion. In [13], a maneuverable underwater vehicle is developed to achieve near-seabed motion with low environmental disturbance through a flat-structured design. Reference [14] introduces a portable multi-thruster underwater vehicle capable of fully omnidirectional motion and experimentally demonstrates its tracking performance. However, the adoption of a layout with separate horizontal and vertical thrusters leads to motion inefficiency and cannot meet the requirements for rapid response to various underwater postures during autonomous operation. Therefore, one of our research goals is to design a novel type of underwater robot, comprising the robot body and simple actuators to form an agile, efficient, kinematically non-redundant underwater autonomous system for operation tasks.

To achieve autonomous motion, numerous controllers have been proposed, including proportional-integral-derivative (PID) [15], backstepping [16], sliding mode [17], and neural network [18] control methods. However, these methods exhibit inherent defects and limitations in practical applications, such as handling nonlinearities, robustness against external disturbances, chattering issues, and generalization capabilities [19,20]. Moreover, they cannot effectively handle constraints in accordance with actual scenarios. MPC, as an optimal control method, has the advantage of handling constraints, nonlinearities and model uncertainties [20], and being robust against external disturbances [21], and is therefore increasingly used to control robots in complex underwater environments. Reference [22], based on linear wave theory, shows that the use of MPC in the station-keeping problem can achieve smaller position errors compared to traditional feedback control. For agile maneuvering in AUVs, Reference [12] demonstrates that the effectiveness of MPC exceeds that of PID and a linear quadratic regulator (LQR). In [23,24], numerical simulations of the trajectory tracking of an AUV were performed based on MPC. In [21], the performance of MPC for underactuated underwater robots under conditions of limited underwater sensor perception, model uncertainty, and external disturbances was investigated, and simulation results were provided. In [25], MPC is used to study way-points tracking and obstacle-avoidance problems faced by underwater robots in confined spaces, and the influence of ocean currents on controller performance is considered; however, this was implemented with only 4-DoF. In addition, for model-based control strategies, the fidelity of the dynamic model is crucial for achieving high control performance. To this end, a variety of system identification approaches have been applied to obtain dynamic models of underwater robots, including theoretical analytical methods [26], computational fluid dynamics (CFD) [27], and data-driven modeling techniques [28]. To the best of our knowledge, no research has yet addressed the problem of synchronously controlling 6-DoF in an underwater robot using MPC and testing it in a physical environment. Therefore, another goal of our research is to model and design an MPC-driven 6-DoF controller for our fully vectored propulsion underwater robot to achieve a full range of autonomous motion and test its performance in underwater environments.

The contribution of this work is the development and implementation of an autonomous underwater robot with a fully vectored propulsion and control strategy. Firstly, we propose a novel concept of fully vectored propulsion and develop an innovative underwater robot based on an eight-thruster vectored layout. The advantage of this design over other existing underwater robots is that each thruster acts on all DoFs, enabling synchronous and rapid responses to position and attitude commands. Then, we establish the dynamic model for the novel system and construct the MPC controller in combination with physical constraints. Finally, we build an ROS-based software framework to achieve efficient communication between the robot components and verify the performance of the proposed robot through extensive underwater experiments. Building on our previous work [29], we refined the system design and physical constraints and supplemented a series of physical experiments in accordance with actual needs, including way-point tracking with small-angle and large-angle maneuvers, trajectory tracking of continuous curves, and robustness tests.

The remainder of this paper is organized as follows. The custom-built underwater robot system is detailed in Section 2. Then, the dynamic model and MPC-based control framework are established in Section 3. Relevant physical experiments in a water tank are conducted and analyzed in Section 4. Finally, the conclusions and future work are presented in Section 5.

2. Underwater Robot System Design

This section details the system design of our proposed underwater robot with a fully vectored propulsion layout, including the mechanical structure, hardware configuration, and software architecture.

2.1. Mechanical Structure

Figure 1 shows different views of our proposed robot. If considering the main body as a cylinder, its length is 0.55 m and its diameter is 0.3 m. The total mass with the battery included is 36.65 kg. Figure 1a illustrates the main components of the robot, whose core structure consists of two sealed cabins made from aluminum alloy. The upper one is the main control cabin and the lower is the battery cabin. Both are enveloped by buoyancy blocks to provide buoyancy. The control cabin communicates with the ground control station via a neutrally buoyant cable to monitor the robot's status. The battery cabin, containing the battery pack and power management module, is connected to the control cabin via aviation connectors. This configuration not only powers the entire robot system but also lowers the robot's center of gravity. The robot's external framework, such as the main connecting components, support frame, and gripper accessories, is made of aluminum alloy, high-density polyethylene (HDPE), and 3D-printed nylon, specifically designed to support and suspend loads. Additionally, several blocks of stainless steel (316L grade) are employed as counterweights. After a series of waterproof tests, it was found that the robot can operate 150 m underwater.

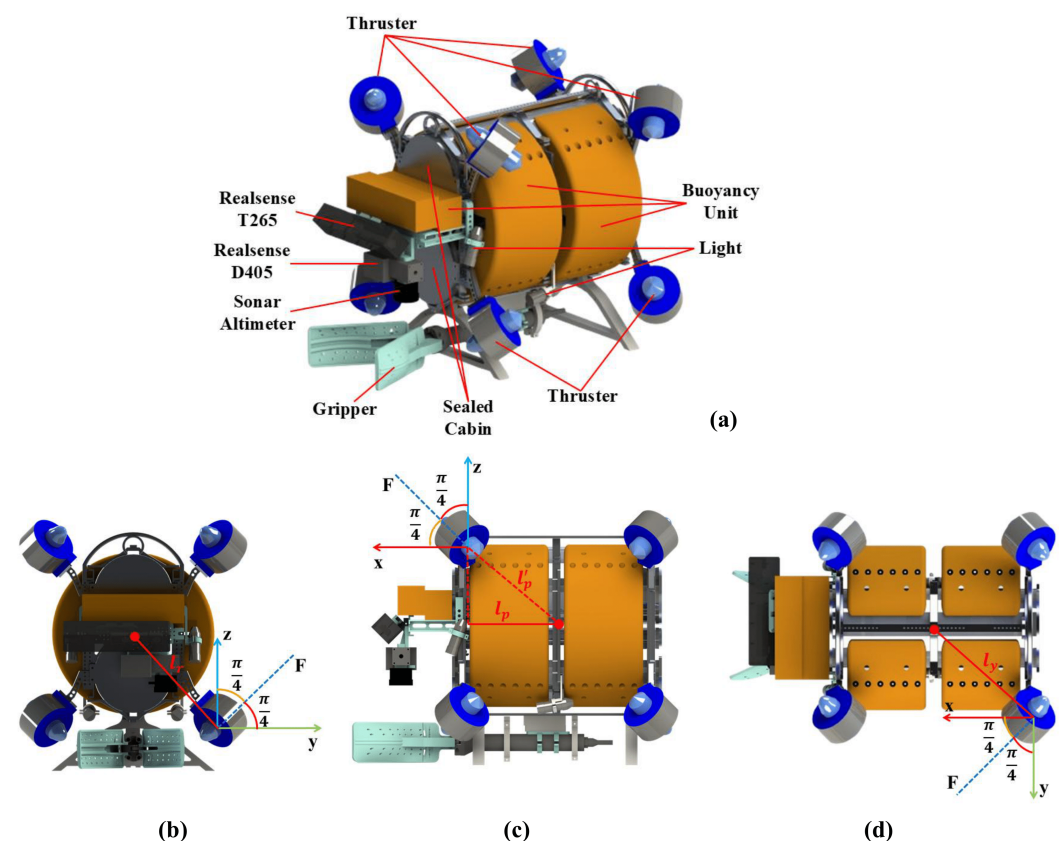


Figure 1. The custom-built underwater robot. (a) demonstration of all main components, (b) front view, (c) side view, and (d) top view. x , y , and z represent the robot's movement directions. "F" denotes the projection of the thrust generated by each thruster on each plane. l_r , l_p , and l_y denotes the moment arm of the thruster with respect to the roll, pitch, and yaw axes.

To enhance the robot's agility and motion efficiency, we designed a fully vectored layout for the propulsion system; the details are depicted in Figure 1b–d. It can be seen that the robot is equipped with eight thrusters, and four of them are arranged symmetrically around the center from all frontal views. It is worth noting that the thrust generated by each thruster forms an angle of $\frac{\pi}{4}$ with the x , y , and z axes when projected onto the xy , yz , and xz planes, respectively. Therefore, each thruster can simultaneously act on all degrees of freedom. This compact dynamic system is very suited to autonomous control and underwater manipulation.

2.2. Hardware Configuration and Software Architecture

We designed a specified hardware configuration and software framework to address the requirements for underwater operation tasks.

2.2.1. Hardware Configuration

The hardware configuration of the robot is mainly composed of five units, the sensor unit, processor unit, actuator unit, communication unit, and power unit, as shown in Figure 2a. The sensor unit includes a Realsense T265 visual odometry (Intel Corporation, Santa Clara, CA, USA) and a Realsense D405 RGBD camera (Intel Corporation, Santa Clara, CA, USA), both of which were waterproofed, as well as a P30 sonar altimeter (Blue Robotics Inc., Torrance, CA, USA), all mounted at the front of the robot. Additionally, an Xsens AHRS (Xsens Technologies B.V., Enschede, the Netherlands) is installed inside the main control cabin. In particular, the D405 camera is used to capture color and depth images. The Xsens AHRS is used to obtain the robot's attitude information (roll, pitch, yaw) and provides pose feedback with the T265. The P30 sonar altimeter is used to measure the robot's height from the seabed. The processor unit includes a ReComputer J4012-Edge AI device (Seeed Technology Co., Ltd., Shenzhen, Guangdong, China), equipped with an NVIDIA Jetson Orin NX 16 GB module (NVIDIA Corporation, Santa Clara, CA, USA), enabling real-time computation for tasks such as processing sensor feedback and executing control processes. Additionally, a Pixhawk embedded device is installed to receive commands to control devices such as thrusters, the gripper, and lights. The actuator unit consists of a single-DoF gripper mounted on the front underside of the robot that opens and closes to manipulate the target, and eight underwater thrusters that provide propulsion for the robot's movement. The communication unit includes two power line communication components. One is installed inside the robot's main control cabin, facilitating the conversion between network and electrical signals. The other is connected to the ground control station via an Ethernet cable, thereby enabling long-distance communication. The power unit consists of a battery pack and a power management module, providing power to other units.

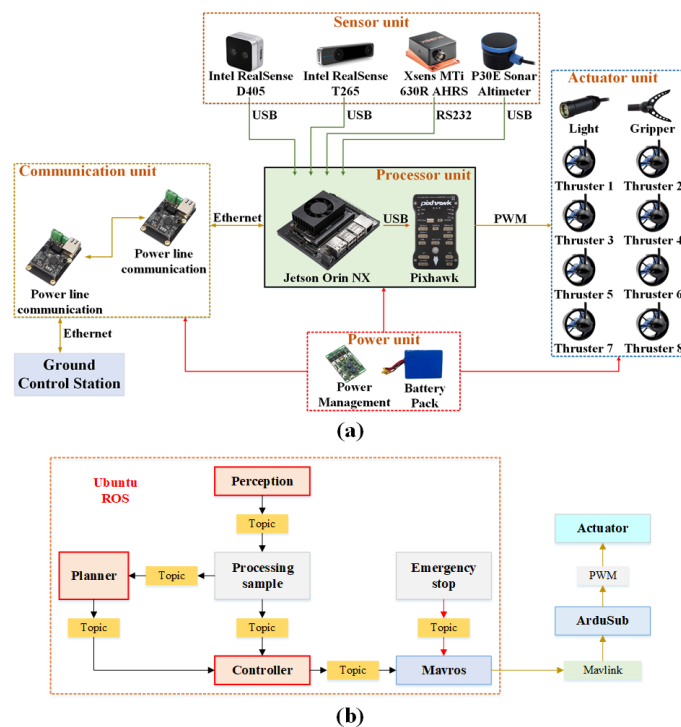


Figure 2. (a) Hardware configuration and (b) software framework of the underwater robot.

2.2.2. Software Architecture

The system architecture is developed based on Ubuntu 20.04 and ROS Noetic, and efficient information exchange between different components is achieved through ROS topics. As shown in Figure 2b, the software system comprises three units: perception, planner, and controller. The perception unit is responsible for processing the information fed back by the robot's sensors in real-time and estimating the states of the robot and the operation target. The planner unit processes and optimizes the state information provided by the perception component. It then plans a sequence of motion commands based on the task objectives and forwards them to the controller unit. The controller unit receives instructions from the planner unit and computes appropriate control inputs in real-time based on the robot's state. Additionally, the open-source software ArduSub (<https://www.ardusub.com/> (accessed on 15 August 2023)) is installed on the Pixhawk, and a bridge is established with ROS through the Mavros (<https://wiki.ros.org/mavros> (accessed on 15 August 2023)) node. It receives computational results from the controller unit via Mavlink messages and drives the actuator units. To avoid damage to the robot during experiments, we also established an emergency stop component for manually stopping all actuator units.

3. Dynamic Modeling and MPC-Based Control

This section presents the dynamic model and control architecture developed for our proposed underwater robot. A standard model of the underwater robot is first established. Then, according to the engineering implementation requirements, several assumptions and simplifications are made, and the thrust allocation matrix is formulated to map the force and torque to thruster output. Finally, model predictive control is constructed to address the 6-DoF synchronized motion problem of the proposed robot.

3.1. Dynamic Modeling of the Underwater Robot

Figure 3 illustrates the body coordinate system and world coordinate system used to represent the translation and rotation motion of the robot. The origin of the body coordinate system is located at the robot's center of mass. The world coordinate system is represented

using the East–North–Up (ENU) coordinate system. The robot’s position and orientation in the world coordinate system are denoted by η , and corresponding linear and angular velocities in body coordinate system are represented by v . Then, we establish the robot’s dynamic model based on the Fossen method [30] as

$$\dot{\eta} = J(\eta)v \quad (1)$$

$$(M_{RB} + M_A)\dot{v} + (C_{RB}(v) + C_A(v))v + D(v)v + g(\eta) = \tau \quad (2)$$

where $J(\eta)$ represents the rotation matrix between the pose vector η and the velocity vector v . M_{RB} denotes the mass and inertia matrix, M_A signifies the added mass, $C_{RB}(v)$ and $C_A(v)$ represent the Coriolis and centripetal force matrices caused by the rigid body and added mass, respectively. $D(v)$ is the hydrodynamic damping coefficient matrix, $g(\eta)$ denotes the gravitational and buoyancy forces, and τ is the vector of forces and torques acting on the robot.

To avoid singularities when the pitch angle is $\frac{\pi}{2}$, we express the robot’s orientation using unit quaternions. Define x as the robot’s state vector, including pose and velocity; then, according to (1) and (2), we can obtain the continuous state space in the form of $\dot{x} = f_{dyn}(x, u) = [\dot{p}_W \ \dot{q}_W \ \dot{v}_B \ \dot{\omega}_B]^T$, where $p_W = [x \ y \ z]^T$ and $q_W = [q_w \ q_x \ q_y \ q_z]^T$ represent the position and orientation of the robot in the world coordinate system, respectively. Similarly, $v_B = [u \ v \ w]^T$ and $\omega_B = [p \ q \ r]^T$ denote the linear and angular velocity vectors in the body coordinate system.

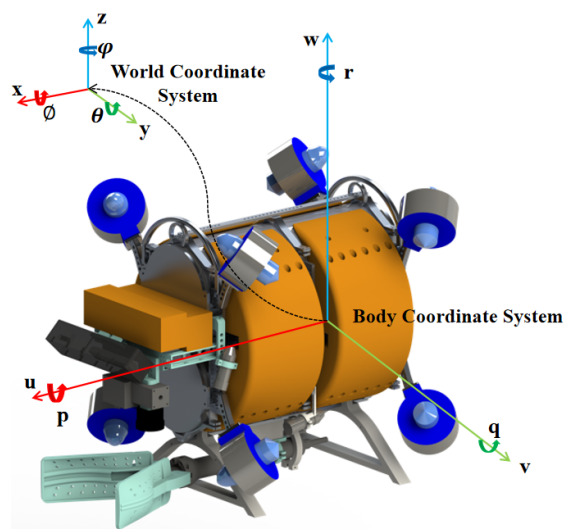


Figure 3. Coordinate systems definition of the proposed underwater robot.

Then, we made some approximations of the dynamic model. The inertia matrix, added mass matrix, and hydrodynamic damping coefficient matrix are simplified into diagonal matrices, with non-diagonal elements set to zero. High-order damping terms are neglected, and a simple linear relationship is assumed between the hydrodynamic damping and both the linear and angular velocities of the robot. In addition, the robot is considered to be neutrally buoyant in water, and the center of gravity and the center of buoyancy of the robot coincide with its geometric center. Consequently, the restoring forces can be neglected

and the effects of Coriolis and centripetal forces are also neglected. Thus, we have the final dynamic model.

$$\begin{cases}
 \dot{x} = (1 - 2(q_y^2 + q_z^2))u + 2(q_x q_y - q_w q_z)v + 2(q_x q_z + q_w q_y)w \\
 \dot{y} = 2(q_x q_y + q_w q_z)u + (1 - 2(q_x^2 + q_z^2))v + 2(q_y q_z - q_w q_x)w \\
 \dot{z} = 2(q_x q_z - q_w q_y)u + 2(q_y q_z + q_w q_x)v + (1 - 2(q_x^2 + q_y^2))w \\
 \dot{q}_w = \frac{1}{2}(-pq_x - qq_y - rq_z) \\
 \dot{q}_x = \frac{1}{2}(pq_w + rq_y - qq_z) \\
 \dot{q}_y = \frac{1}{2}(qq_w - rq_x + pq_z) \\
 \dot{q}_z = \frac{1}{2}(rq_w + qq_x - pq_y) \\
 \dot{u} = (\tau_u - uX_u)/(m + X_{\dot{u}}) \\
 \dot{v} = (\tau_v - vY_v)/(m + Y_{\dot{v}}) \\
 \dot{w} = (\tau_w - wZ_w)/(m + Z_{\dot{w}}) \\
 \dot{p} = (\tau_p - pK_p)/(I_{xx} + K_{\dot{p}}) \\
 \dot{q} = (\tau_q - qM_q)/(I_{yy} + M_{\dot{q}}) \\
 \dot{r} = (\tau_r - rN_r)/(I_{zz} + N_{\dot{r}})
 \end{cases} \quad (3)$$

where m represents the total mass of the robot, $\{I_{xx}, I_{yy}, I_{zz}\}$ denote the rotational inertia in each respective direction, $\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}$ represents the added mass, $\{X_u, Y_v, Z_w, K_p, M_q, N_r\}$ correspond to the hydrodynamic damping coefficients and $\{\tau_u, \tau_v, \tau_w, \tau_p, \tau_q, \tau_r\}$ signify the forces and torques in each respective direction. All physical parameters in the above dynamic model (3) are obtained through a combination of theoretical analysis, SolidWorks 2022-based geometric measurements, and parameter identification experiments. More detailed parameter values and the corresponding analysis procedures can be found in [29].

3.2. Thrust Allocation

The underwater robot proposed in this paper adopts eight vector-distributed thrusters to control the 6-DoF motion. Therefore, after obtaining the control inputs, it is necessary to design an allocation matrix to map the control inputs containing force and torque to the control inputs of thrusters. For ease of calculation, we expressed the actual values of the forces, torques, and control inputs of the thrusters in percentage form. $\mathbf{u}_{dyn} = [u_u \ u_v \ u_w \ u_p \ u_q \ u_r]^T$ represents the percentage of the actual control input $\boldsymbol{\tau}$. $\mathbf{u}_{thruster} = [u_1 \ u_2 \ u_3 \ u_4 \ u_5 \ u_6 \ u_7 \ u_8]^T$ represents the percentage of the control input of thrusters f . The conversion of force and torque to thrust is defined as

$$\mathbf{T}_\alpha \mathbf{F}_{max} \mathbf{u}_{thruster} = \boldsymbol{\tau}_{max} \mathbf{u}_{dyn} \quad (4)$$

where $\boldsymbol{\tau}_{max} = \text{diag}(\tau_{max}(u), \tau_{max}(v), \tau_{max}(w), \tau_{max}(p), \tau_{max}(q), \tau_{max}(r))$ represents the maximum force and torque produced by thrusters, $\mathbf{F}_{max} = \text{diag}(f_{max}(1), f_{max}(2), \dots, f_{max}(8))$ represents the maximum thrust of each thruster, and $\mathbf{T}_\alpha$ is a 6×8 dynamics allocation matrix that characterizes the mapping from the forces generated by individual thrusters to the forces and torques acting on the robot. To maintain the angular momentum

balance of the robot, the thrusters are arranged in alternating clockwise and counterclockwise configurations, and the allocation matrix T_α is defined accordingly as

$$T_\alpha = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{\sqrt{2}}{\sqrt{3}}l_r & \frac{\sqrt{2}}{\sqrt{3}}l_r & \frac{\sqrt{2}}{\sqrt{3}}l_r & \frac{\sqrt{2}}{\sqrt{3}}l_r & -\frac{\sqrt{2}}{\sqrt{3}}l_r & -\frac{\sqrt{2}}{\sqrt{3}}l_r & -\frac{\sqrt{2}}{\sqrt{3}}l_r & -\frac{\sqrt{2}}{\sqrt{3}}l_r \\ -\frac{1}{\sqrt{3}}l_p & \frac{1}{\sqrt{3}}l_p & \frac{1}{\sqrt{3}}l_p & -\frac{1}{\sqrt{3}}l_p & -\frac{1}{\sqrt{3}}l_p & \frac{1}{\sqrt{3}}l_p & \frac{1}{\sqrt{3}}l_p & -\frac{1}{\sqrt{3}}l_p \\ -\frac{\sqrt{2}}{\sqrt{3}}l_y & -\frac{\sqrt{2}}{\sqrt{3}}l_y & \frac{\sqrt{2}}{\sqrt{3}}l_y & \frac{\sqrt{2}}{\sqrt{3}}l_y & -\frac{\sqrt{2}}{\sqrt{3}}l_y & -\frac{\sqrt{2}}{\sqrt{3}}l_y & \frac{\sqrt{2}}{\sqrt{3}}l_y & \frac{\sqrt{2}}{\sqrt{3}}l_y \end{bmatrix} \quad (5)$$

where l_r , l_p , and l_y denote the moment arms of the thruster with respect to the roll, pitch, and yaw axes, respectively, as illustrated in Figure 1. In addition, it should be noted that in the pitch plane, as shown in Figure 1c, the direction of the thrust generated by the thruster is parallel to the nominal moment arm l'_p , and therefore the effective moment arm l_p is defined as the perpendicular component of l'_p . As a result, once we have the percentage of the dynamic control inputs u_{dyn} , we can calculate the percentage of the thruster control inputs from (4) as

$$u_{thruster} = F_{max}^{-1} T_\alpha^\dagger \tau_{max} u_{dyn}. \quad (6)$$

3.3. Model Predictive Control

The general form of model predictive control involves predicting the future states of a system within a defined finite prediction horizon T based on the system's initial states and the system dynamics $\dot{x} = f_{dyn}(x, u)$, and then determining the optimal control input at each time step. In our application scenario, the control objective is to maneuver the robot synchronously to a specified target position with the desired orientation, and to maintain specified linear and angular velocities. Therefore, the control objective can be formulated as $x_d = [x, y, z, q_w, q_x, q_y, q_z, u, v, w, p, q, r]^T$. To obtain numerical results, the prediction horizon T is represented as N discrete time steps. Using a prediction step size of $\delta t = T/N$, the robot's dynamic model (3) can be discretized by employing the explicit fourth-order Runge–Kutta method, $x_{k+1} = f_{RK4}(x_k, u_k, \delta t)$. The discretized model will be utilized in the prediction and optimization process of MPC.

Therefore, the MPC controller designed for our proposed robot can be described as follows:

$$\begin{aligned} & \min_{u^*} (x_N - x_d)^T Q (x_N - x_d) \\ & + \sum_{k=0}^N \left\{ (x_k - x_d)^T Q (x_k - x_d) + u_k^T R u_k \right\} \\ & s.t. \begin{cases} x_{k+1} = f_{RK4}(x_k, u_k, \delta t) \\ x_0 = x_{init} \\ u_{min} \leq u_k \leq u_{max} \end{cases} \end{aligned} \quad (7)$$

where x_0 is the state of the robot at the beginning of each optimization process. Q and R are the weight matrices for the loss function. u_{min} and u_{max} represent the lower and upper limits of the percentage of force and torque inputs for each degree of freedom, respectively.

By solving the constructed MPC, the optimal control sequence $u^* = [u_0^*, u_1^*, \dots, u_{N-1}^*]$ can be obtained in real-time. Subsequently, the first control vector u_0^* is selected as the current control input u_{dyn} , which is then converted into a percentage of thruster control inputs $u_{thruster}$ through (6). Finally, the thruster control inputs are applied to the actual underwater robot system. During the implementation process, similar to the approaches in [31], we employ CasADi [32] to formulate the MPC problem and utilize ACADOS [33] to

generate the C++ interface, which is subsequently solved in real-time using the qpOASES solver [34], thereby implementing real-time control for our proposed robot.

4. Experiments and Results

In this section, to validate the effectiveness of the proposed robot system design and the performance of the MPC-based motion control framework under different control objectives and disturbance conditions, and to demonstrate that the robot has the potential to accomplish autonomous operation tasks, we conducted a series of experiments in an indoor water tank, including way-point tracking, trajectory tracking, and robustness testing.

4.1. Experimental Setup

All the following experiments were conducted in an indoor rectangular water tank with dimensions of $5 \times 3 \times 1.5 \text{ m}^3$. The tank was surrounded by a NOKOV motion capture system equipped with six underwater cameras, providing the robot's position and velocity information within the tank with sub-millimeter accuracy at a frequency of 100 Hz, as shown in Figure 4. To ensure accurate data quality, only a square area ($1.5 \times 1.5 \text{ m}^2$) in the center of the water tank was used as a reliable testing area. Additionally, the Xsens AHRS was used to provide attitude and angular velocity in the ENU coordinate system at a frequency of 70 Hz, with accuracies of $0.2^\circ, 0.2^\circ, 1^\circ$ in the roll, pitch, and yaw directions, respectively. Define $\{X_w, Y_w, Z\}$ and $\{X_t, Y_t, Z\}$ as representing the Xsens AHRS and motion capture coordinate systems, respectively. It is worth noting that there is an offset of approximately 10° between the positive direction X_w and the positive direction X_t .

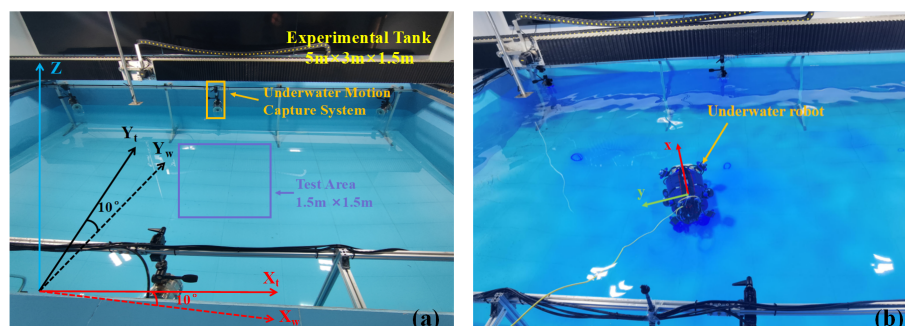


Figure 4. The testing environment of the underwater robot in a water tank: (a) a rectangular water tank equipped with six underwater capture cameras, and (b) the robot in the tank.

4.2. Way-Points Tracking Tests

Two way-point tracking experiments were conducted by guiding the robot to follow a series of predefined waypoints within the tank and maintain a specific attitude at each point. In *Experiment I*, the robot's ability to move towards the desired waypoints while maintaining small-angle maneuvers was preliminarily determined. Subsequently, *Experiment II* further validated the robot's ability to perform large-angle maneuvers during the motion.

4.2.1. Experiments I

The robot initially started from a random position and orientation in the tank, and sequentially moved to several waypoints, maintaining stability for a period after reaching each target, as shown in Figure 5. From 0 s to 26 s, the robot moved to the first waypoint $[x_1, y_1, z_1] = [-0.1, -0.1, 0.6] \text{ m}$. From 26 s to 66 s, it moved to the second waypoint $[x_2, y_2, z_2] = [-0.1, 0.1, 0.6] \text{ m}$. From 66 s to 106 s, it reached the third waypoint $[x_3, y_3, z_3] = [0.1, 0.1, 0.6] \text{ m}$, and from 106 s to 146 s, it arrived at the fourth waypoint $[x_4, y_4, z_4] = [0.1, -0.1, 0.6] \text{ m}$. After 146 s, it returned to the first one. Throughout these

movements, the desired orientation was consistently maintained at $[\phi_d, \theta_d, \psi_d] = [0^\circ, 0^\circ, 0^\circ]$. The controller parameters used for the experiments were set as follows: the prediction horizon for MPC was $T = 0.5$ s with $N = 5$ and the weights for the loss function were $Q = [1, 1, 1, 1, 1, 0.5, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]$ and $R = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$. The entire optimization process was executed at a frequency of 20 Hz, and the frequency of thruster execution was set to 50 Hz. To ensure that each thruster was not overloaded, we set the control input limits to 60% of the maximum capacity.

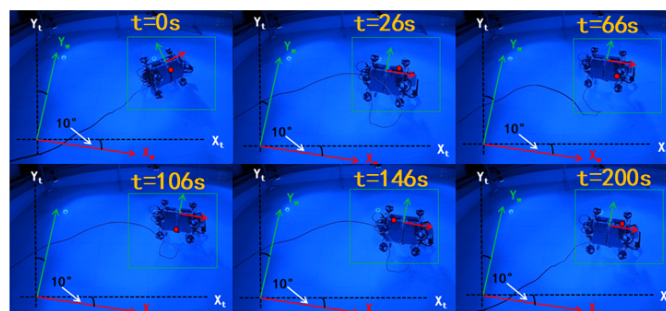


Figure 5. Illustration of the robot's movement in way-points tracking experiment I. The red points indicate the center of the tank's coordinate system.

Figure 6 illustrates the experimental results of position and attitude during way-points tracking. It can be seen that after receiving the control objective, the robot reached the target within 2 s of the initial position, then swayed around the target for a period of time and gradually stabilized within 10 s. The control inputs throughout the entire motion process are plotted in Figures 7 and 8, where red solid lines represent the actual values, green dashed lines represent the upper and lower bounds of the constraints, and the actual forces and torques always remain within the set constraints. The robot can maintain relatively stable for 15 s after receiving each new instruction, and we calculated the average steady-state error in *Experiment I*, which are $[x_{err}, y_{err}, z_{err}] = [0.0035, 0.0024, 0.0267]$ m in x , y , and z directions, respectively, and $[\phi_{err}, \theta_{err}, \psi_{err}] = [1.06^\circ, 2.15^\circ, 0.35^\circ]$ in roll, pitch, and yaw orientations.

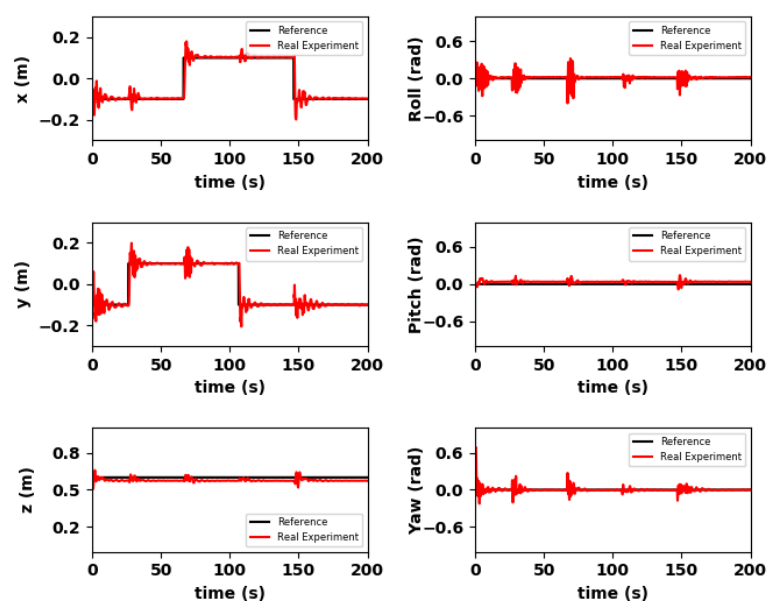


Figure 6. Way-points tracking experiment I: tracking results of the underwater robot's position and attitude.

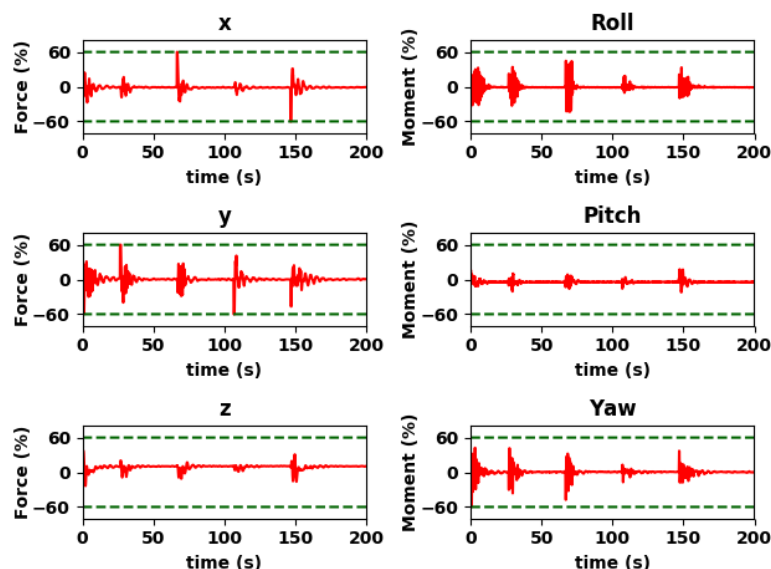


Figure 7. Way-points tracking experiment I: actual control inputs and constraints, expressed as percentages.

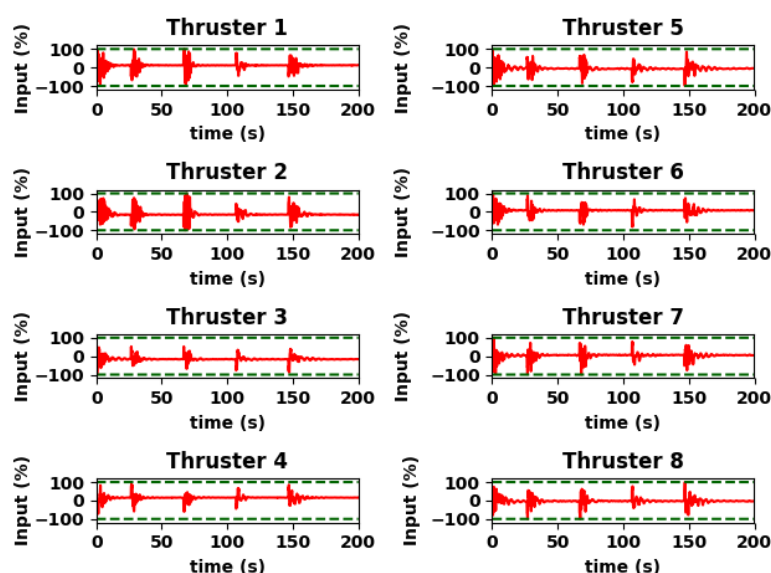


Figure 8. Way-points tracking experiment I: actual input of eight thrusters.

4.2.2. Experiments II

From 0 s to 54 s, the robot initially moved from a random position with a random attitude to the first waypoint $[x_1, y_1, z_1] = [0, 0, 0.6]$ m and maintained the first target attitude $[\phi_1, \theta_1, \psi_1] = [0^\circ, 0^\circ, 0^\circ]$. Then, from 54 s to 100 s, it moved to the second waypoint $[x_2, y_2, z_2] = [0.2, -0.2, 0.6]$ m, then executed a 90° rotation and maintained the second target attitude $[\phi_2, \theta_2, \psi_2] = [0^\circ, 0^\circ, 90^\circ]$. In this experiment, we set $Q = [0.5, 0.5, 0.5, 1, 1, 1, 0.5, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]$ and $R = [0.2, 0.2, 0.2, 0.3, 0.1, 0.1]$.

The experimental results are shown in Figures 9 and 10, where we can see that upon receiving the control objective, the robot can reach the desired position and attitude within 6 s and maintain stability. Similarly, we further calculated the average steady state error in *Experiment II*, which is $[x_{err}, y_{err}, z_{err}] = [0.0162, 0.0275, 0.0691]$ m and $[\phi_{err}, \theta_{err}, \psi_{err}] = [1.43^\circ, 1.31^\circ, 2.41^\circ]$.

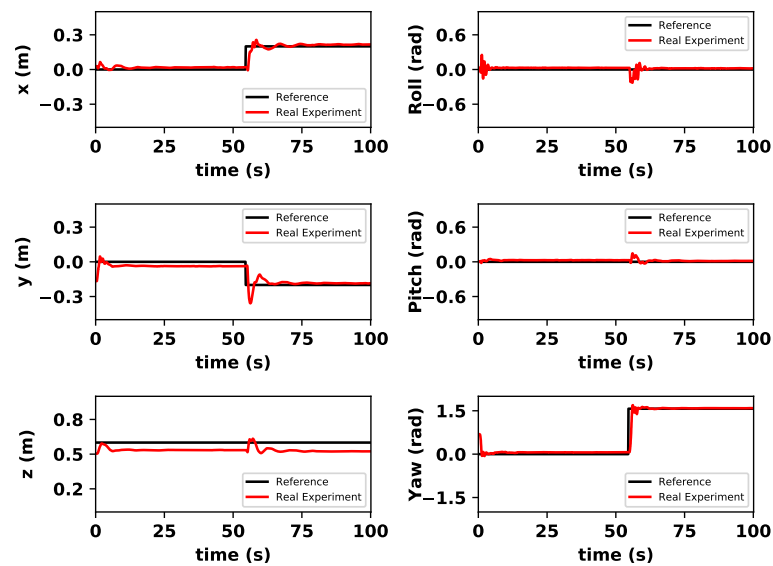


Figure 9. Way-points tracking experiment II: tracking results of the underwater robot's position and attitude.

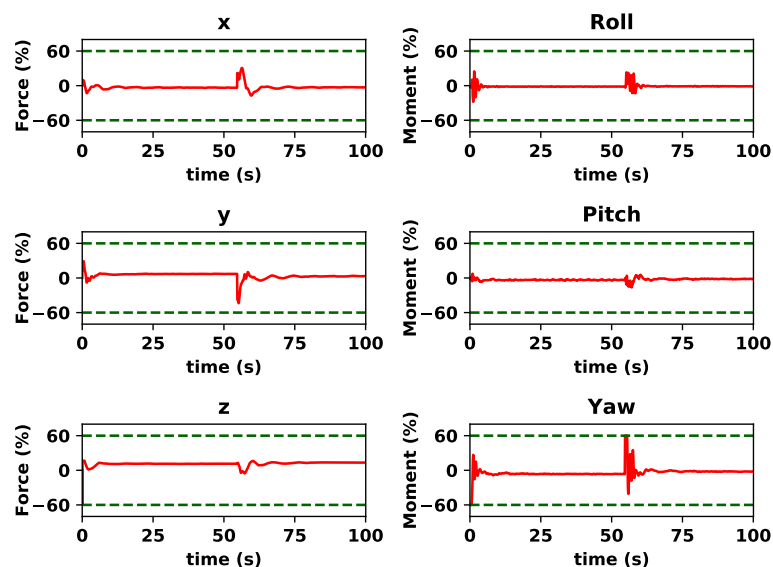


Figure 10. Way-points tracking experiment II: actual control inputs and constraints, expressed as percentages.

From the above experiments, it can be seen that the maximum value of the average steady-state error in position and attitude does not exceed 0.0691 m and 2.41° for both small-angle and large-angle maneuvers. Furthermore, it is noted that the average error on z axis is relatively large compared to the x and y axes. This is because the robot's motion in z axis is influenced by both gravity and buoyancy, and the thrusters we used have a range of dead zone. When the control input is very small, the thrusters cannot be effectively activated, but this can be solved by using thrusters with higher precision.

4.3. Trajectory Tracking Tests

In this experiment, the robot initially started from a random position with a random attitude, and tracked a desired circular trajectory with a radius of 0.1 m, while maintaining a fixed height of 0.6 m. The desired trajectory is expressed as

$[x_d(t), y_d(t), z_d(t)] = [0.1 \sin(2\pi/60)t, 0.1 \cos(2\pi/60)t, 0.6]$ m, and the desired orientations are $[\phi_d(t), \theta_d(t), \psi_d(t)] = [0^\circ, 0^\circ, 0^\circ]$. The controller's parameters are identical to those in way-points tracking *Experiment II*. The tracking results are plotted in Figure 11, where the black line represents the desired trajectory, and the red line represents the actual motion trajectory of the robot. It can be seen that upon receiving the control command, the robot can respond within 2 s and gradually reduces the tracking error to approaching the desired trajectory within 30 s. After stabilized, the maximum errors in x , y , and z are 0.0213 m, 0.0343 m, and 0.0744 m, respectively, and that in ϕ , θ and ψ are 1.88° , 1.41° , and 2.41° , respectively. Furthermore, the control inputs are plotted in Figure 12, showing that the constructed controller can always determine the optimal force and torque within the constraints.

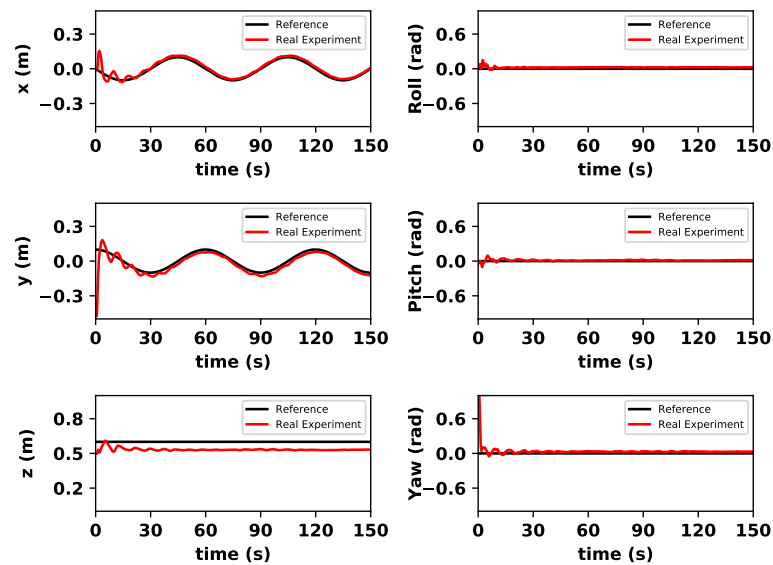


Figure 11. Trajectory tracking experiment: tracking results of the underwater robot's position and attitude.

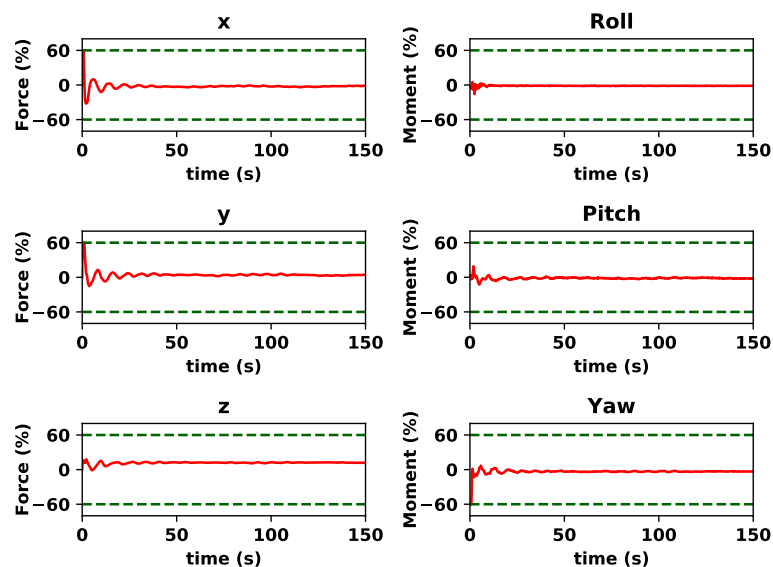


Figure 12. Trajectory tracking experiment: actual control inputs and constraints, expressed as percentages.

4.4. Robustness Tests

To test the robustness of the proposed controller, we randomly pushed the robot from different directions when it was hovering. Figures 13 and 14 representatively demonstrate the results that the robot was initially hovering, and we pushed it at 14 s, 42 s, 58 s, and 84 s to cause it to deviate from the target position. It can be seen from the several peaks that, after experiencing disturbances, the robot can quickly return to the target position and orientation through the controller's adjustments. During the entire adjustment process, all control inputs were found within the constraints.

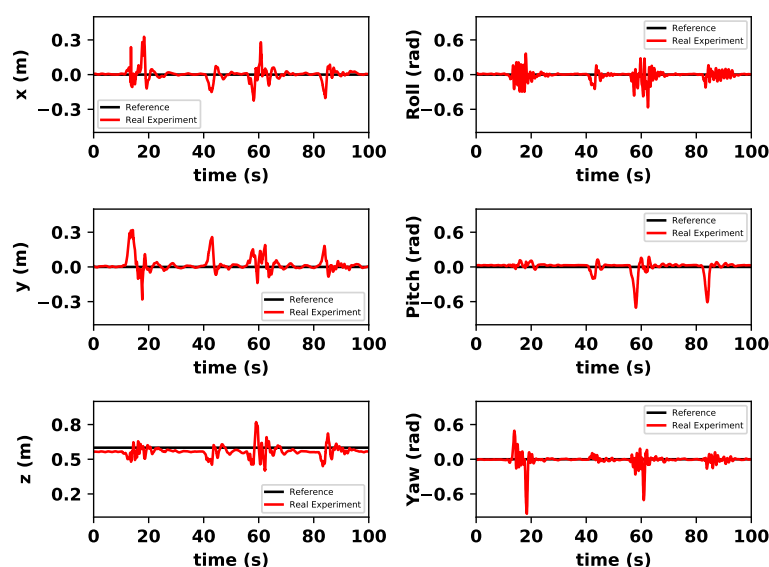


Figure 13. Robustness testing experiments: tracking results of the underwater robot's position and attitude.

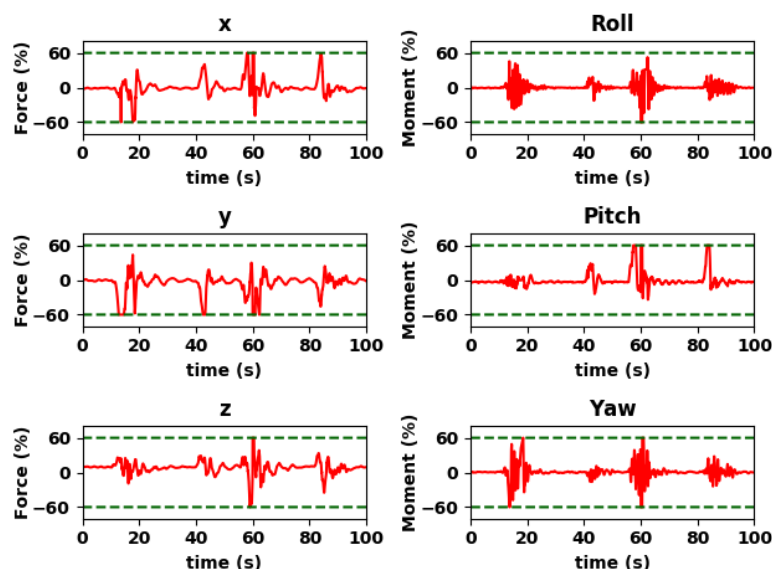


Figure 14. Robustness testing experiment: actual control inputs and constraints, expressed as percentages.

5. Conclusions and Future Work

In this paper, an autonomous underwater robot with fully vectored propulsion was proposed and prototyped. The hardware platform and software architecture required for autonomous underwater manipulation were developed. The proposed robot's control

system, which is based on model predictive control, aims to enable the robot to perform synchronized movements in 6-DoF, making it potentially applicable to various ocean manipulation tasks. The experimental results in the tank demonstrated that the robot has precise and flexible movement capabilities and can adapt to a certain degree of disturbance.

While the proposed robot's system design and the MPC-based control framework demonstrate satisfactory performance in 6-DoF motion control, the robot's continuous aggressive maneuvering ability has not yet been fully explored. The employed thrusters limit the robot's achievable motion. Therefore, future work will focus on integrating energy-aware thrust allocation strategies into the proposed control framework to enhance the robot's ability to perform aggressive maneuvers. Another future direction is to evaluate the robot's performance in real marine environments and to achieve underwater manipulation.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L. and T.G.; software, T.G.; validation, T.G., Y.G. and S.L.; formal analysis, T.G.; investigation, T.G., Y.L. and N.Z.; resources, T.G. and Y.G.; data curation, T.G.; writing—original draft preparation, T.G.; writing—review and editing, Y.L., N.Z., X.F., X.L. and Y.S.; visualization, S.L.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L., N.Z. and X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 52305009, Grant 62176037 and Grant 62002043.

Data Availability Statement: The original contributions presented in this study are included in the paper. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model Predictive Control
ROS	Robot Operating System
ROVs	Remotely Operated Vehicles
AUVs	Autonomous Underwater Vehicles
UVMS	Underwater Vehicle-Manipulator Systems
PID	Proportional-Integral-Derivative
LQR	Linear Quadratic Regulator
HDPE	High-Density Polyethylene
ENU	East-North-Up

References

1. Hong, S.; Chung, D.; Kim, J.; Kim, Y.; Kim, A.; Yoon, H.K. In-water visual ship hull inspection using a hover-capable underwater vehicle with stereo vision. *J. Field Robot.* **2019**, *36*, 531–546.
2. Girdhar, Y.; McGuire, N.; Cai, L.; Jamieson, S.; McCammon, S.; Claus, B.; Soucie, J.E.S.; Todd, J.E.; Mooney, T.A. CUREE: A Curious Underwater Robot for Ecosystem Exploration. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023.
3. Chalkiadakis, V.; Papandroulakis, N.; Livanos, G.; Moirogiorgou, K.; Giakos, G.; Zervakis, M. Designing a small-sized autonomous underwater vehicle architecture for regular periodic fish-cage net inspection. In Proceedings of the 2017 IEEE International Conference on Imaging Systems and Techniques (IST), Beijing, China, 18–20 October 2017; pp. 1–6.
4. Rutledge, J.; Yuan, W.; Wu, J.; Freed, S.; Lewis, A.; Wood, Z.; Gambin, T.; Clark, C. Intelligent shipwreck search using autonomous underwater vehicles. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 6175–6182.
5. Yoerger, D.R.; Govindarajan, A.F.; Howland, J.C.; Llopiz, J.K.; Wiebe, P.H.; Curran, M.; Fujii, J.; Gomez-Ibanez, D.; Katija, K.; Robison, B.H.; et al. A hybrid underwater robot for multidisciplinary investigation of the ocean twilight zone. *Sci. Robot.* **2021**, *6*, eabe1901.

6. Mostafa, A.; Wang, Y.; Tangirala, S.; Zhang, H.; Lu, F. A 5 kW Hull-Compatible Inductive Charging System With 360° Folded Spatial Unipolar Coupler for Autonomous Underwater Vehicles (AUVs). *IEEE Trans. Ind. Appl.* **2023**, *59*, 7001–7012.
7. Petillot, Y.R.; Antonelli, G.; Casalino, G.; Ferreira, F. Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles. *IEEE Robot. Autom. Mag.* **2019**, *26*, 94–101.
8. Simetti, E.; Casalino, G.; Torelli, S.; Sperinde, A.; Turetta, A. Floating underwater manipulation: Developed control methodology and experimental validation within the trident project. *J. Field Robot.* **2014**, *31*, 364–385.
9. Simetti, E.; Wanderlingh, F.; Torelli, S.; Bibuli, M.; Odetti, A.; Bruzzone, G.; Rizzini, D.L.; Aleotti, J.; Palli, G.; Moriello, L.; et al. Autonomous underwater intervention: Experimental results of the MARIS project. *IEEE J. Ocean. Eng.* **2017**, *43*, 620–639.
10. Cai, M.; Wang, S.; Wang, Y.; Wang, R.; Tan, M. Coordinated Control of Underwater Biomimetic Vehicle–Manipulator System for Free Floating Autonomous Manipulation. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 4793–4803.
11. Duecker, D.A.; Bauschmann, N.; Hansen, T.; Kreuzer, E.; Seifried, R. Towards Micro Robot Hydrobatcs: Vision-based Guidance, Navigation, and Control for Agile Underwater Vehicles in Confined Environments. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 1819–1826.
12. Bhat, S.; Panteli, C.; Stenius, I.; Dimarogonas, D.V. Nonlinear model predictive control for hydrobatcs: Experiments with an underactuated AUV. *J. Field Robot.* **2023**, *40*, 1840–1859.
13. Liu, K.; Ding, M.; Pan, B.; Yu, P.; Lu, D.; Chen, S.; Zhang, S.; Wang, G. A maneuverable underwater vehicle for near-seabed observation. *Nat. Commun.* **2024**, *15*, 10284.
14. Sha, Z.; Wang, X.; Yang, M.; Lei, H.; Zhang, F. A Portable Autonomous Underwater Vehicle With Multi-Thruster Propulsion: Design, Development, and Vision-Based Tracking Control. *IEEE Robot. Autom. Lett.* **2025**, *10*, 3046–3053.
15. Chemori, A.; Kuusmik, K.; Salumäe, T.; Kruusmaa, M. Depth control of the biomimetic u-cat turtle-like auv with experiments in real operating conditions. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4750–4755.
16. Dong, B.; Lu, Y.; Xie, W.; Huang, L.; Chen, W.; Yang, Y.; Zhang, W. Robust performance-prescribed attitude control of foldable wave-energy powered auv using optimized backstepping technique. *IEEE Trans. Intell. Veh.* **2022**, *8*, 1230–1240.
17. Qiao, L.; Zhang, W. Trajectory tracking control of AUVs via adaptive fast nonsingular integral terminal sliding mode control. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1248–1258.
18. Cui, R.; Yang, C.; Li, Y.; Sharma, S. Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1019–1029.
19. Gan, W.; Zhu, D.; Hu, Z.; Shi, X.; Yang, L.; Chen, Y. Model Predictive Adaptive Constraint Tracking Control for Underwater Vehicles. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7829–7840.
20. Shen, C.; Shi, Y. Distributed implementation of nonlinear model predictive control for AUV trajectory tracking. *Automatica* **2020**, *115*, 108863.
21. Heshmati-Alamdari, S.; Nikou, A.; Dimarogonas, D.V. Robust Trajectory Tracking Control for Underactuated Autonomous Underwater Vehicles in Uncertain Environments. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 1288–1301.
22. Fernández, D.C.; Hollinger, G.A. Model Predictive Control for Underwater Robots in Ocean Waves. *IEEE Robot. Autom. Lett.* **2017**, *2*, 88–95.
23. 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.
24. Yan, Z.; Gong, P.; Zhang, W.; Wu, W. Model predictive control of autonomous underwater vehicles for trajectory tracking with external disturbances. *Ocean Eng.* **2020**, *217*, 107884.
25. Heshmati-alamdari, S.; Karras, G.C.; Marantos, P.; Kyriakopoulos, K.J. A Robust Model Predictive Control Approach for Autonomous Underwater Vehicles Operating in a Constrained Workspace. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 6183–6188.
26. Edge, C.; Sakib Enan, S.; Fulton, M.; Hong, J.; Mo, J.; Barthelemy, K.; Bashaw, H.; Kallevig, B.; Knutson, C.; Orpen, K.; et al. Design and Experiments with LoCO AUV: A Low Cost Open-Source Autonomous Underwater Vehicle. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 1761–1768.
27. Hong, L.; Wang, X.; Zhang, D.S. CFD-based hydrodynamic performance investigation of autonomous underwater vehicles: A survey. *Ocean Eng.* **2024**, *305*, 117911.
28. Macatangay, X.; Gabriel, S.A.; Hoseinnezhad, R.; Fowler, A.; Bab-Hadiashar, A. Machine learning for modeling underwater vehicle dynamics: Overview and insights. *IEEE Access* **2024**, *12*, 139486–139504.
29. Gao, T.; Luo, Y.; Lv, C.; Luo, W.; Fu, X.; Zhao, N.; Luo, X.; Shen, Y. Model Predictive Control for an Autonomous Underwater Robot with Fully Vectored Propulsion. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 10482–10488.

30. T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
31. Torrente, G.; Kaufmann, E.; Föhn, P.; Scaramuzza, D. Data-driven MPC for quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3769–3776.
32. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36.
33. Verschueren, R.; Frison, G.; Kouzoupis, D.; Frey, J.; Duijkeren, N.v.; Zanelli, A.; Novoselnik, B.; Albin, T.; Quirynen, R.; Diehl, M. Acados—A modular open-source framework for fast embedded optimal control. *Math. Program. Comput.* **2022**, *14*, 147–183.
34. Ferreau, H.J.; Kirches, C.; Potschka, A.; Bock, H.G.; Diehl, M. qpOASES: A parametric active-set algorithm for quadratic programming. *Math. Program. Comput.* **2014**, *6*, 327–363.

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.