2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Model Predictive Control for an Autonomous Underwater Robot with Fully Vectored Propulsion

Tianzhu Gao, Yudong Luo†, Chao Lv, Weirong Luo, Xianping Fu, Na Zhao, Xi Luo, and Yantao Shen

***Abstract*— Due to the low motion efficiency and maneuverability of underwater robots with six degrees of freedom, it is challenging for them to respond quickly to the attitude requirements during underwater autonomous manipulation. This paper presents a novel autonomous underwater robot with fully vectored propulsion and a model predictive control method to achieve more agile and efficient movements autonomously. In detail, we first design a robot with eight vector-distributed thruster layouts for fully vectored propulsion and construct the software architecture based on the robot operating system (ROS). Then, we establish the hydrodynamic model by adopting the Fossen approach and construct a 13-dimensional system state-space equation, which is discretized using the explicit fourth-order Runge-Kutta method. To achieve autonomous manipulation, model predictive control is employed along with physical constraints of the custom-built robot to enable real-time prediction and optimization of the robot's states for control purposes. Finally, numerical simulations and experiments of the Point-to-Point Motion are conducted to test the robot's performance. Experimental results reveal that the average error of each direction is 0.0027 m, 0.0031 m, and 0.0368 m in the x-axis, y-axis, and z-axis, respectively, and 0.8502°, 2.1941°, 0.2408° corresponding to three attitude angles, which verify the performance of employing MPC to control an autonomous underwater robot with fully vectored propulsion.**

I. INTRODUCTION

The underwater robots are of great significance to the development of marine resources. In recent years, an increasing number of underwater robots have been applied to tasks such as ocean exploration, observation, salvage, and archaeology [1]–[3]. In nearshore industries, diving operations can cause significant harm to divers, leading to various health issues. The development of Remotely Operated Vehicles (ROVs) in the 1990s opened up new possibilities for underwater operations by replacing human labor with robots, thereby saving costs and improving safety. However, using ROVs typically requires skilled operators and the support of a mother ship and cables, which limits their operation range and application scenarios. Therefore, developing autonomous underwater robots with manipulation capabilities is of great practical significance [4]. Several valuable advancements have been made, but high-precision underwater robot manipulation still remains challenging. It is worth noting that most underwater autonomous manipulation robots are designed based on the UVMS model, where the robotic arm is separated from the robot body. Although some researchers [5] have developed lightweight multi-link underwater manipulator arms to reduce the mutual influence between the arms and the robot, this design still faces difficulties in maintaining balance and hydrodynamic, which greatly affects control performance [5]–[7].

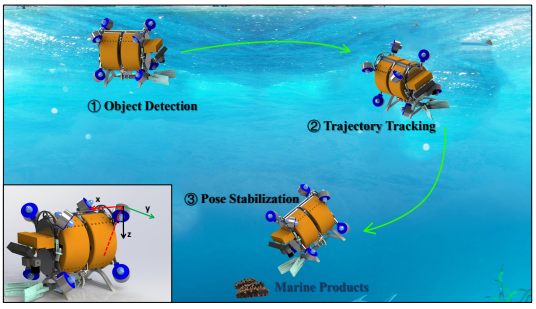

Fig. 1. The autonomous underwater with fully vectored propulsion for marine products grasping.

Another research approach is to develop six degrees of freedom underwater robots with autonomous manipulation capabilities, which can achieve the same manipulation abilities as UVMS by utilizing the full range of the robot's motion. However, current underwater robots with this capability often adopt a layout and control method with separate horizontal and vertical thrusters, resulting in low efficiency and poor maneuverability. These robots have failed to meet the requirements of fast response for various underwater poses during autonomous manipulations [8], [9]. To address these limitations, we design and develop an autonomous robot based on an 8-vectored thruster layout, with each thruster's thrust direction at a 45-degree angle relative to the robot's xy and xz planes. This layout enables eight thrusters to simultaneously act in any degree of freedom of the robot's movement, improving propulsion efficiency and robot flexibility. The concept is shown in Fig. 1.

However, the coupling between thrusters poses a new challenge for the robot controller. As an optimal control method, Model Predictive Control (MPC) excels in addressing multi-input multi-output system problems and can explicitly consider system constraints [10]. In recent years, MPC has shown great performance in robotics, made substantial progress [11], and gradually entered the field of underwater robots. For instance, by combining model predictive con-

*This work was supported by the Fundamental Research Funds for the Central Universities under Grant 3132023263 and the National Natural Science Foundation of China under Grant 52305009.

Tianzhu Gao, Yudong Luo†, Chao Lv, Weirong Luo, Xianping Fu, and Na Zhao are with Dalian Maritime University, Dalian, Liaoning 116026, China.

Xi Luo is with Yichang Testing Tech. Research Institution, Yichang, Hubei 410083, China.

Yantao Shen is with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557, USA.

†Corresponding author. Email: *ydluo@dlmu.edu.cn.*

trol with flow state estimation [12], researchers proposed an active perception navigation framework for autonomous underwater docking under complex marine environments. Combined with the advantages of Model Predictive Control, [13] studied the cooperative transportation tasks of multiple underwater robots with communication constraints in confined spaces. Using nonlinear model predictive control, they investigated trajectory tracking and obstacle avoidance problems for under-actuated underwater robots in constrained spaces with obstacles [14]. A tracking and capture framework for motion targets based on model predictive control and Extended Kalman Filter (EKF) in a redundant structure UVMS was proposed [15]. For the hydrobatic mission of under-actuated AUV, optimal control strategies were analyzed using nonlinear model predictive control, and time-varying MPC were implemented on AUV hardware to demonstrate the control performance [16]. The control performance of model predictive control in ocean wave fields was proved based on linear wave theory. Compared to traditional feed-forward control, model predictive control showed smaller position errors [17].

Considering the current absence of work in model predictive control with fully-vectored autonomous underwater robots, we propose to control the fully vectored propulsion underwater robot using model predictive control to achieve a novel, efficient, and autonomous underwater robot for underwater manipulation tasks. **Our main contributions are as follows. i)** We propose the development of a fully vectored propulsion autonomous underwater robot, which can improve thruster efficiency, and enhance the robot's flexibility and agile manipulations on underwater targets. Together with onboard edge computing devices, sensors, and actuators, the developed robot can achieve autonomous manipulation capability. **ii)** We build the software architecture of the robot based on ROS, which ensures efficient information transfer among perception, planning, and control modules. **iii)** We perform theoretical modeling, parameter identification, and real-time prediction and optimization of the fully-vectored robot's state based on the model predictive control approach with physical constraints to achieve automatic control. **iv)** We conduct numerical studies and experiments of the robot's position and attitude response during the Point-to-Point Motion to validate the performance of the prototype of the robot. To the best of our knowledge, currently, there is no such autonomous underwater robot with fully vectored propulsion.

The paper is organized as follows. In Sec. II, we first describe the overall system design of the robot in terms of hardware design and software architecture. In Sec. III, we focus on the modeling and controller design for the robot. Sec. IV presents numerical simulation and experimental validation of the robot. Finally, we provide a summary of our research.

II. System Design and Parameter Identification

This section mainly introduces the system design of our fully vectored propulsion autonomous underwater robot, including hardware design and software architecture design. To facilitate modeling and control, the system parameters of the robot will be provided in this section, and the parameters that cannot be directly obtained will be acquired through system identification.

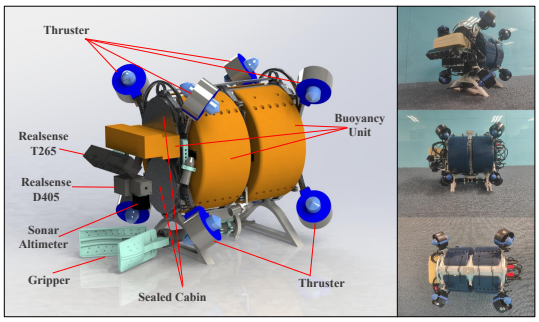

Fig. 2. The main components of the custom-built robot and prototype.

A. Hardware Design and Parameter Identification

Fig. 2 shows the main components and prototype of our proposed underwater robot. The main body is a cylindrical body with a length of 0.55 m and a diameter of 0.3 m. The core part of the robot includes the sensor unit, actuator unit, and processing unit.

The sensor units include a visual odometer T265 camera and an RGBD camera D405 in front of the robot, as well as a P30 sonar altimeter and XSENS AHRS. These sensing units are waterproofed and can collect color and depth images of 848 $\times$ 480 size at a frequency of 30 Hz in underwater environment, and provide feedback on the robot's pose information at a speed of no less than 70 Hz.

The actuator unit includes a single degree of freedom underwater gripper below the robot and eight vectored propulsion underwater thrusters. This layout allows the robot to have eight thrusters acting simultaneously in any degree of freedom motion, thus enabling the underwater robot to have flexible and efficient underwater 6-degree of freedom manipulation capabilities. To obtain the relationship between the input signal and force generated by the thruster, we designed an experiment, as shown in Fig. 3, where the thruster was fixed underwater to allow the blades to rotate freely, the force sensor was used to collect force generated by the thruster, while computers collected PWM signals. Finally, we can plot the relationship between the actual input voltage, PWM signal, and force generated by the thrusters, see the data plots in Fig. 3. According to the experimental results, it can be known that during forward rotation, a maximum thrust of 2.9 kg can be provided by a thruster, while for reverse rotation, a maximum thrust of 2.4 kg can be provided.

The processing unit includes a ReComputer J4012-Edge AI device equipped with an NVIDIA Jetson Orin NX 16 GB module capable of providing 100 TOPS of AI computing power, as well as an embedded device Pixhawk for controlling drives such as thrusters and grippers. In addition, the buoyancy foams provide about 30 kg of buoyancy, allowing the robot to suspend under the water. Then, by following the method in [18] and considering the robot as

a standard cylinder, we can obtain the model parameters of the underwater robot. The specific parameters and values are shown in Tab. I, and will be used in the modeling part.

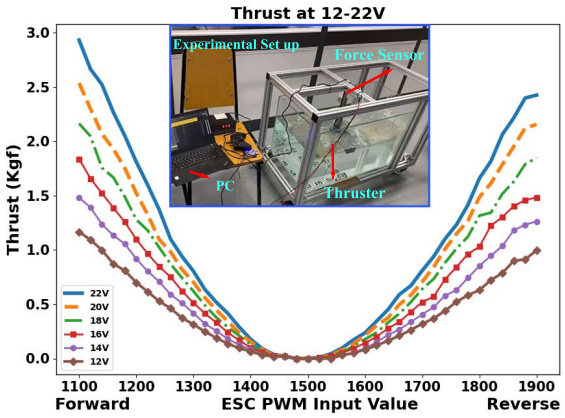

Fig. 3. The corresponding relationship among the input voltage, PWM signal, and the thrust.

TABLE I

PARAMETERS OF THE ROBOT

Parameters	Nomenclature	Value (unit)
Total mass	m	35.65 kg
Rotational Inertia, p	I_{xx}	1.19 kg·m^2
Rotational Inertia, q	I_{yy}	1.03 kg·m^2
Rotational Inertia, r	I_{zz}	0.61 kg·m^2
Added mass, u	$X_{\dot{u}}$	7.8 kg
Added mass, v	$Y_{\dot{v}}$	36.8 kg
Added mass, w	$Z_{\dot{w}}$	36.8 kg
Added mass, p	$K_{\dot{p}}$	0 kg·m^2
Added mass, q	$M_{\dot{q}}$	2.3 kg·m^2
Added mass, r	$N_{\dot{r}}$	2.3 kg·m^2
Hydrodynamic Damping, u	X_u	5.1 kg/s
Hydrodynamic Damping, v	Y_v	36.7 kg/s
Hydrodynamic Damping, w	X_w	36.7 kg/s
Hydrodynamic Damping, p	K_p	0 kg·m^2/(s·rad)
Hydrodynamic Damping, q	M_q	1.7 kg·m^2/(s·rad)
Hydrodynamic Damping, r	N_r	8.8 kg·m^2/(s·rad)

B. Software Architecture Design

As shown in Fig. 4, we have developed a software system framework for robots based on Ubuntu 20.04 and ROS Noetic. Through ROS topics, we have achieved connections between software and hardware systems, ensuring efficient information transfer between modules during autonomous underwater manipulations. The software system mainly includes three modules, perception, planning, and control. The function of the perception module is to measure and estimate the state information of the target and robot in real-time through sensors such as visual, acoustic, and inertial units carried by the robot, providing a basis for the planner. The function of the planning module is to process and optimize the state information provided by the perception module in real-time based on task objectives and transmit instructions to the control module. The function of the control module is to calculate the control inputs required for the robot to move in real-time based on the instructions provided by the planning module.

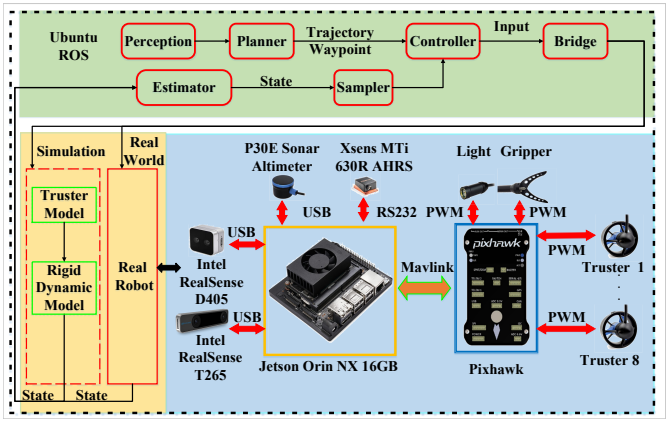

Fig. 4. Software and hardware framework of the underwater robot.

III. MODELING AND CONTROL

A. Dynamic Modeling of the Underwater Robot

For the actual needs of underwater manipulation tasks, both the position and attitude need to be concerned synchronously. A standard dynamics model of underwater robot is established based on the Fossen method [19].

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

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{2}$$

where $\boldsymbol{\eta}$ represents the position and attitude of the robot in the world coordinate system, $\boldsymbol{v}$ represents the linear velocity and angular velocity of the robot in the body coordinate system, $\boldsymbol{J}(\boldsymbol{\eta})$ and $\boldsymbol{M}$ are the rotation matrix and inertia matrix, respectively, $\boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}$ is the Coriolis force, $\boldsymbol{D}(\boldsymbol{v})\boldsymbol{v}$ represent the hydrodynamic damping, $\boldsymbol{g}(\boldsymbol{\eta})$ indicates the restoring force, and $\boldsymbol{\tau}$ is the force and torque produced by the thrusters.

In order to simplify the design of the controller, some assumptions about the standard dynamics model are made. First, we assume that the center of gravity and the center of buoyancy of the robot coincide with the geometric center of the robot and that the robot is a rigid, neutrally buoyant body. Then the restoring force $\boldsymbol{g}(\boldsymbol{\eta})$ and the Coriolis force $\boldsymbol{C}(\boldsymbol{v})\boldsymbol{v}$ can be ignored. Besides, we also ignored high-order damping and assumed a simple linear relationship between hydrodynamic damping and the velocity as well as the angular velocity of the robot. Finally, we simplify the standard robot model (1) and (2) to a 13-dimensional dynamics equation (3)∼(5), which can be expressed as

$$\begin{aligned}
\dot{x} &= (1 - 2(2q_y + 2q_z))u + 2(q_x q_y - q_w q_z)v \\
&\quad + 2(q_x q_z + q_w q_y)w \\
\dot{y} &= 2(q_x q_y + q_w q_z)u + (1 - 2(2q_x + 2q_z))v \\
&\quad + 2(q_y q_z - q_w q_x)w \\
\dot{z} &= 2(q_x q_z - q_w q_y)u + 2(q_y q_z + q_w q_x)v \\
&\quad + (1 - 2(2q_x + 2q_y))w
\end{aligned} \tag{3}$$

$$
\begin{aligned}
\dot{q_w} &= \frac{1}{2}(-pq_x - qq_y - rq_z) \\
\dot{q_x} &= \frac{1}{2}(pq_w + rq_y - qq_z) \\
\dot{q_y} &= \frac{1}{2}(qq_w - rq_x + pq_z) \\
\dot{q_z} &= \frac{1}{2}(rq_w + qq_x - pq_y)
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\dot{u} &= (\tau_u - uX_u)/(m + X_{\dot{u}}) \\
\dot{v} &= (\tau_v - vY_v)/(m + Y_{\dot{v}}) \\
\dot{w} &= (\tau_w - wZ_w)/(m + Z_{\dot{w}}) \\
\dot{p} &= (\tau_p - pK_p)/(I_{xx} + K_{\dot{p}}) \\
\dot{q} &= (\tau_q - qM_q)/(I_{yy} + M_{\dot{q}}) \\
\dot{r} &= (\tau_r - rN_r)/(I_{zz} + N_{\dot{r}})
\end{aligned} \tag{5}
$$

where $\{x, y, z\}$ represents the robot's position in the world coordinate system. To avoid singularity, a unit quaternion (q_w, q_x, q_y, q_z) is adopted to describe the robot's attitude $\{\phi, \theta, \psi\}$. $\{u, v, w\}$ and $\{p, q, r\}$ represent linear velocity and angular velocity of the robot in the body coordinate system, respectively. m is the mass of the robot, $\{I_{xx}, I_{yy}, I_{zz}\}$ is the rotational inertia, $\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}$ is the added mass, and $\{X_u, Y_v, Z_w, K_p, M_q, N_r\}$ is the hydrodynamic damping coefficients. $\tau = \{\tau_u, \tau_v, \tau_w, \tau_p, \tau_q, \tau_r\}$ represents the force and torque in each direction.

B. Thrust Allocation

The designed underwater robot uses a structure of fully-vectored propulsion. For the motion of the robot in any direction, all the thrusters are executed synchronously. The dynamics of control input in each direction could be converted to the control input of the thrusters based on an allocation matrix.

For the convenience of calculation, the percentage of control input is applied instead of using the actual value of control input. $\boldsymbol{u}_{dyn} = \begin{bmatrix} u_u, u_v, u_w, u_p, u_q, u_r \end{bmatrix}^\mathsf{T}$ represents the percentage of the actual control input of dynamics $\boldsymbol{\tau}$. $\boldsymbol{u}_{thruster} = \begin{bmatrix} u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8 \end{bmatrix}^\mathsf{T}$ represents the percentage of the control input of thrusters $\boldsymbol{f}$. The conversion of force and torque to thrust is defined as

$$
\boldsymbol{T}_\alpha \boldsymbol{f}_{max} \boldsymbol{u}_{thruster} = \boldsymbol{\tau}_{max} \boldsymbol{u}_{dyn} \tag{6}
$$

where $\boldsymbol{\tau}_{max} = diag(\tau_{max}(u), \tau_{max}(v), \tau_{max}(w), \tau_{max}(p), \tau_{max}(q), \tau_{max}(r))$ represents the maximum value of force and torque produced by thrusters, $\boldsymbol{f}_{max} = diag(f_{max}(1), f_{max}(2), f_{max}(3), f_{max}(4), f_{max}(5), f_{max}(6) f_{max}(7), f_{max}(8))$ represents the maximum thrust of each thruster. $\boldsymbol{T}_\alpha$ is a dynamics allocation matrix of 6×8.

The percentage of the dynamics control inputs $\boldsymbol{u}_{dyn}$ can be converted to the percentage of thruster control input, which is expressed as

$$
\boldsymbol{u}_{thruster} = \boldsymbol{f}_{max}^{-1} \boldsymbol{T}_\alpha^{-1} \boldsymbol{\tau}_{max} \boldsymbol{u}_{dyn}. \tag{7}
$$

C. Model Predictive Control

The general approach of model predictive control involves predicting the system behavior based on the system's dynamics model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ for N time steps ahead within a defined time horizon and optimizing a set of control input sequences, then applying only the first term from this optimized sequence to the actual system. At the next system time, this procedure is repeated iteratively.

In our application scenario, the control objective is for the underwater robot to synchronously track both its position and attitude, represented as $\boldsymbol{x}_{target} = \begin{bmatrix} x, y, z, q_w, q_x, q_y, q_z \end{bmatrix}^\mathsf{T}$. Due to the energy limit of the thruster that we use, the constraints of control input need to be considered. As the general expression, our specific problem is formulated as a quadratic form:

$$
\begin{aligned}
&\min_{\boldsymbol{u}} (\boldsymbol{x}_N - \boldsymbol{x}_{target})^\mathsf{T} Q (\boldsymbol{x}_N - \boldsymbol{x}_{target}) \\
&+ \sum_{k=0}^{N} \left\{ (\boldsymbol{x}_k - \boldsymbol{x}_{target})^\mathsf{T} Q (\boldsymbol{x}_k - \boldsymbol{x}_{target}) + \boldsymbol{u}_k^\mathsf{T} R \boldsymbol{u}_k \right\} \\
&s.t. \begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{f}_{RK4}(\boldsymbol{x}_k, \boldsymbol{u}_k, \delta t) \\ \boldsymbol{x}_0 = \boldsymbol{x}_{init} \\ \boldsymbol{u}_{min} \le \boldsymbol{u}_k \le \boldsymbol{u}_{max} \end{cases}
\end{aligned} \tag{8}
$$

where Q and R represent the weight matrix of the cost function. We use the fourth-order Runge-Kutta method to transform the dynamics model (3) (4) (5) into a discrete form, where δt represents the size of the prediction time step. $\boldsymbol{x}_0$ represents the state of the robot at the beginning of each optimization process. $\boldsymbol{u}_{min}$ and $\boldsymbol{u}_{max}$ represent the lower and upper limits of the control input, respectively. Similar to the work of [20], we also transformed model predictive control into solving SQP problem and implemented our robot control system using Acados [21] and CasADi [22].

IV. Experiments and Results

In order to verify the performance of the proposed robot based on the constructed MPC controller, we conducted both numerical studies and experiments to demonstrate the robot's performance in terms of position and attitude, respectively.

A. Numerical Simulations

1) Simulation setup: The simulation was performed on a laptop configured with an Intel(R) Core(TM) i5-10210U CPU and 16 GB of memory. During the simulation, the robot initially stood still, then started to move from the initial position $[x_0, y_0, z_0] = [2, 2, -2]$ m at a pose angle $[\phi_0, \theta_0, \psi_0] = [0°, 30°, 30°]$ towards the target position $[x, y, z] = [0, 0, 0]$ m. The expected pose angle of the robot upon arrival was $[\phi, \theta, \psi] = [0°, 0°, 0°]$, after which it maintained this pose at the target position.

The parameters of the controller that were used for simulation are as follows. The sampling period is $T = 0.1$ s, the prediction horizon is $N = 20$, and the weights of the cost function are $Q = \{1, 1, 1, 1, 1, 1, 0.5, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01\}$ and $R = \{0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$. To ensure that the

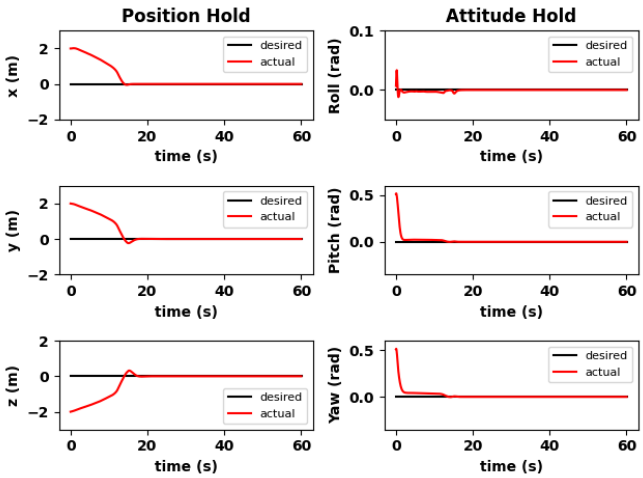

Fig. 5. Position and attitude results in numerical simulation.

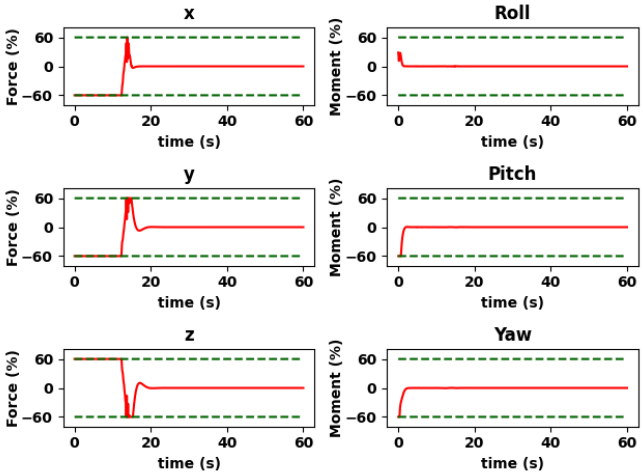

Fig. 6. The control input of each direction in numerical simulation.

control input does not exceed the execution capacity of the motor as much as possible, the constraint in each direction is set to be 60% of the maximum control input. The execution frequency of the controller set in the simulation is 20 Hz.

2) Results and discussion: Fig. 5 and Fig. 6 representatively demonstrate the results acquired from the simulation. In detail, Fig. 5 shows the response of the robot in position and attitude during the simulation. It can be seen that after about 2 seconds, the attitude angles including roll, pitch, and yaw are smoothly adjusted to the target attitude and the robot remains stable subsequently. After about 13 seconds, the robot reached the target position and then remained stable after a slight oscillation process.

Fig. 6 illustrates the control inputs of the robot in each direction, where the solid lines and the dashed line represent the desired control inputs and corresponding constraints, respectively. It can be observed that the control input is always figured out within the constraints, that is, 60% of the maximum input. When the robot reached the expected

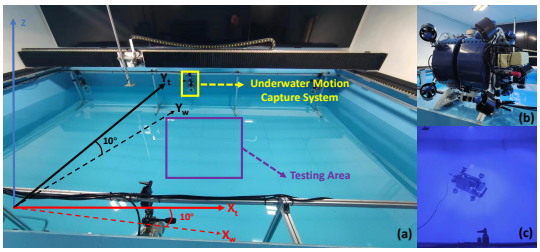

Fig. 7. The testing environment of the robot in a real tank: (a) a rectangular water tank equipped with a motion capture system, (b) the prototype of the robot, and (c) the robot in the tank. $\{X_w, Y_w, Z\}$ and $\{X_t, Y_t, Z\}$ represent the Xsens AHRS coordinate system and tank coordinate system, respectively.

position and attitude, the control input gradually approached zero and remained in this state.

The numerical simulation results show the effectiveness of the constructed controller and prove the feasibility of applying model predictive control to the underwater robot with designed fully vectored propulsion.

B. Experiments

1) Experimental setup: The experiment was conducted in a 5 m $\times$ 3 m $\times$ 1.5 m indoor rectangular tank, surrounded by a set of NOKOV Motion Capture System with six underwater cameras, and the sampling rate of the system is 100 Hz. In order to obtain accurate position and velocity information, we only used a square area of 1.5 m $\times$ 1.5 m close to the central area of the tank as the captured area, with an accuracy of sub-millimeter, as shown in Fig. 7. The attitude data was provided by the Xsens AHRS that delivers true-north-referenced yaw or heading with a sampling rate of 70 Hz and an accuracy of 0.2°, 0.2°, 1° in the roll, pitch, yaw direction, respectively. It is worth noting that there is an offset of 10° between the positive direction of the X_t in the tank coordinate system and the X_w direction of the Xsens AHRS coordinate system.

A Point-to-Point Motion experiment was carried out to evaluate the performance of the proposed MPC controller. As shown in Fig. 8, the robot initially moved from a random position (a) in the tank at a random attitude to the target position $[x, y, z] = [0, 0, 0.6]$ m, and reached the target (c) attitude $[\phi, \theta, \psi] = [0°, 0°, 0°]$, after which it remained pose at the target point.

The parameters of the controller and their value are detailed as follows. The sampling period is $T = 0.5$ s, the prediction horizon is $N = 5$, and the values of weight parameters and constraints are the same as those set in the numerical simulation. The execution frequency of the controller is 20 Hz and the execution frequency of the trusters is 50 Hz.

2) Experimental results and analysis: Fig. 9 illustrates the response of the robot's position and attitude within a time duration of 60 seconds. It can be seen from the plots of x, y, and z that the robot approached the desired position from its initial position within 2 seconds, then it gradually stabilized within 10 seconds, which can be observed from the three attitude plots. During this process, all control inputs

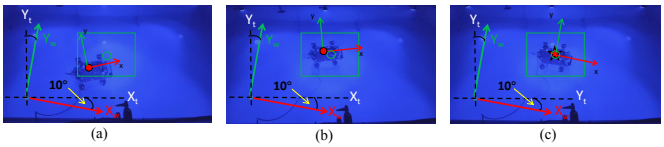

Fig. 8. Illustration of the movement of the robot in the Point-to-Point Motion experiment. The red circle and green circle represent the current position of the robot and its desired position, respectively.

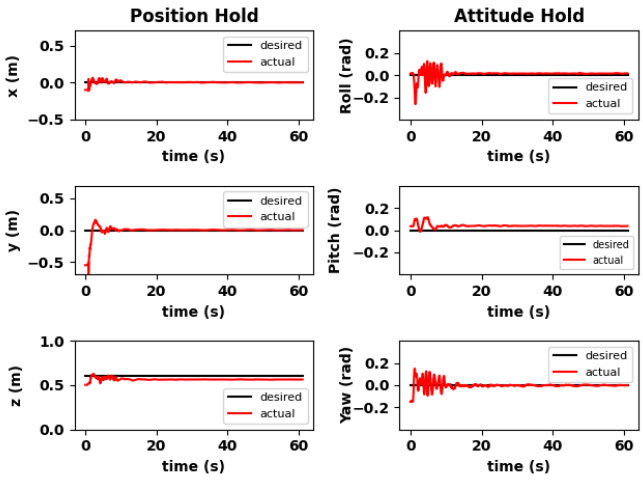

Fig. 9. Position and attitude results during the experiment.

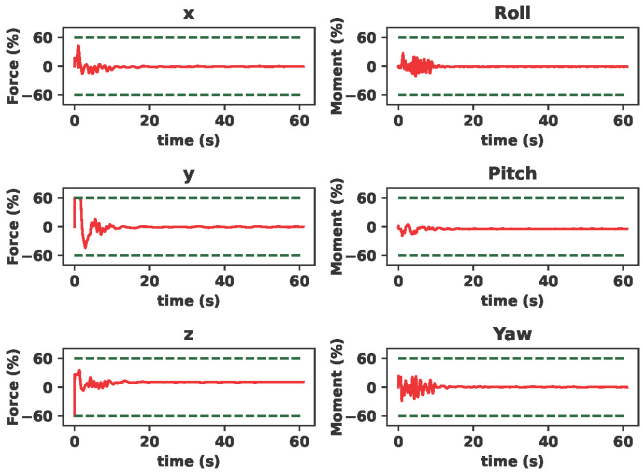

Fig. 10. The percentage of control input in each direction during the experiment.

including forces and torques in three axes were figured out within the constraints, as shown in Fig. 10.

Then, we collected the average error corresponding to the robot's position and attitude after 15 seconds, which are defined by adding a subscript err to each variable. Thus we have $[x_{err}, y_{err}, z_{err}] = [0.0027, 0.0031, 0.0368]$ m and $[\phi_{err}, \theta_{err}, \psi_{err}] = [0.8502°, 2.1941°, 0.2408°]$. Overall, the errors of the three attitude angles are relatively small, and even the pitch angle, where the maximum error occurs, with an error of $2.1941°$, is acceptable. Compared to the error on

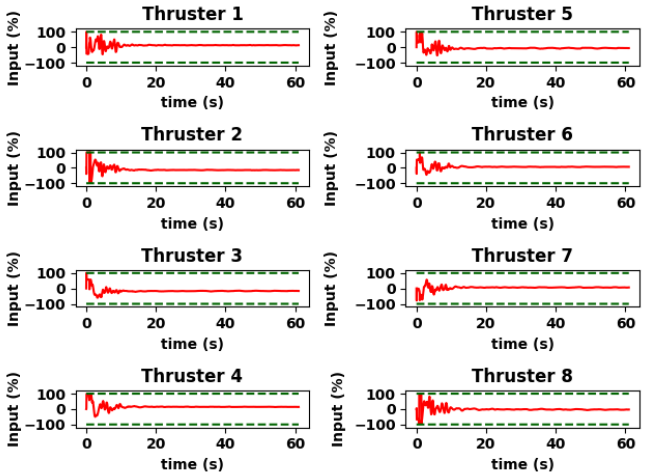

Fig. 11. The input of each thruster during the experiment.

the x-axis and y-axis, the error on the z-axis is relatively large. The main reason that results in this difference is that the performance in the z-axis direction is the result of the resultant force generated by weight and buoyancy, while the x- and y-axes are not, which can be proved by the force plots of x, y, and z in Fig. 10. The presence of this resultant force makes the motor output required for z-axis motion small. In this case, due to the dead zone of the thruster we employed in our prototype, the thruster cannot be effectively started. However, this problem can be solved by replacing it with a high-precision thruster to reduce the response dead zone. Additionally, we collected the required outputs of eight motors and plotted them in Fig. 11, where we can find that the output of all thrusters does not exceed the upper and lower boundaries. The experimental results prove the feasibility of the robotic system design and the effectiveness of the controller.

V. Conclusion

In this paper, we designed and manufactured an autonomous underwater robot with a fully vectored propulsion structure, and built the hardware platform and software architecture that are required for autonomous underwater manipulation. We also constructed the control system for the robot based on model predictive control. The numerical simulation and experimental results in the tank showed that the robot has very accurate motion ability and can potentially be applied to ocean manipulation tasks. Our future research work will evaluate the robot's performance in various disturbance conditions and conduct manipulation tasks in the field.

Acknowledgment

We are grateful to Professor Minyi Xu from the Marine Engineering College of Dalian Maritime University and his students Bo Liu, Peng Xu, and Cong Zhao for providing the tank and underwater motion capture system.

REFERENCES

[1] A. A. R. Newaz, T. Alam, G. M. Reis, L. Bobadilla, and R. N. Smith, "Long-term autonomy for auvs operating under uncertainties in dynamic marine environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6313–6320, 2021.

[2] K. Baxevani, G. E. Otto, H. G. Tanner, and A. C. Trembanis, "Development and field testing of an optimal path following asv controller for marine surveys," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 6861–6866.

[3] J. Rutledge, W. Yuan, J. Wu, S. Freed, A. Lewis, Z. Wood, T. Gambin, and C. Clark, "Intelligent shipwreck search using autonomous underwater vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6175–6182.

[4] Y. R. Petillot, G. Antonelli, G. Casalino, and F. Ferreira, "Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles," *IEEE Robotics & Automation Magazine*, vol. 26, no. 2, pp. 94–101, 2019.

[5] Y. Wang, S. Wang, Q. Wei, M. Tan, C. Zhou, and J. Yu, "Development of an underwater manipulator and its free-floating autonomous operation," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 815–824, 2015.

[6] P. Cieslak, P. Ridao, and M. Giergiel, "Autonomous underwater panel operation by girona500 uvms: A practical approach to autonomous underwater manipulation," in *2015 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 529–536.

[7] E. Simetti and G. Casalino, "Manipulation and transportation with cooperative underwater vehicle manipulator systems," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 782–799, 2016.

[8] H. Huang, Q. Tang, J. Li, W. Zhang, X. Bao, H. Zhu, and G. Wang, "A review on underwater autonomous environmental perception and target grasp, the challenge of robotic organism capture," *Ocean Engineering*, vol. 195, p. 106644, 2020.

[9] S. Bhat and I. Stenius, "Hydrobatics: A review of trends, challenges and opportunities for efficient and agile underactuated auvs," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, 2018, pp. 1–8.

[10] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[11] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio *et al.*, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, p. eabl6259, 2022.

[12] R. Vvekanandan, D. Chang, and G. A. Hollinger, "Autonomous underwater docking using flow state estimation and model predictive control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1062–1068.

[13] S. Heshmati-Alamdari, G. C. Karras, and K. J. Kyriakopoulos, "A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 917–930, 2021.

[14] S. Heshmati-Alamdari, A. Nikou, and D. V. Dimarogonas, "Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1288–1301, 2020.

[15] Y. Dai, S. Yu, Y. Yan, and X. Yu, "An ekf-based fast tube mpc scheme for moving target tracking of a redundant underwater vehicle-manipulator system," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2803–2814, 2019.

[16] S. Bhat, C. Panteli, I. Stenius, and D. V. Dimarogonas, "Nonlinear model predictive control for hydrobatics: Experiments with an underactuated auv," *Journal of Field Robotics*, 2022.

[17] D. C. Fernández and G. A. Hollinger, "Model predictive control for underwater robots in ocean waves," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 88–95, 2016.

[18] K. Orpen, "Dynamic modeling and simulation of an autonomous underwater vehicle (auv)," 2021.

[19] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[20] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[21] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.

[22] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.