Doctoral Thesis

Generating UAV/UUV Motion for Object Manipulation

NAOKI shirakura

Program of Information Science and Engineering
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Prof. Takahiro Wada
Human Robotics Laboratory
(Division of Information Science)

Submitted on June 18, 2021

A Doctoral Thesis
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

NAOKI Shirakura

Thesis Committee:

Supervisor Takahiro Wada
(Professor, Division of Information Science)
Kenji Sugimoto
(Professor, Division of Information Science)
Jun Takamatsu
(Visiting Professor, Division of Information Science)
Gustavo Alfonso Garcia Ricardez
(Visiting Associate Professor, Division of Information Science)
Sung-Gwi Cho
(Assistant Professor, Division of Information Science)
Tsukasa Ogasawara
(Executive Director, Nara Institute of Science and Technology)

Generating UAV/UUV Motion for Object Manipulation*

NAOKI Shirakura

Abstract

In research on mobile robots, completely autonomous control is one of the ultimate goals. However, full automation is still difficult because the real environment has uncertainties such as sensing noise and unexpected obstacles. Therefore, as an immediate goal, semi-automatic systems including remote control by humans are also being discussed.

Unmanned aerial vehicles (UAVs) and unmanned underwater vehicles (UUVs) are mobile robots that can perform tasks such as manipulating objects instead of humans in the air or underwater. However, since these robots require six degrees of freedom (DoF) control, it is difficult to perform object manipulation tasks while controlling the motion of the robots by manual operation. Therefore, simplification of manual operation is required.

In this dissertation, a motion generation method using simplified control commands is proposed for object manipulation with mobile robots such as UAVs and UUVs that require six DOF control. Existing controllers of UAVs can simplify the operation by semi-automatic control such as altitude hold. However, the existing controller does not support operations that require contact with the environment or force control. Therefore, as an example of such a manipulation task, we consider the motion generation method using simplified control commands during push motion with a UAV. For UUVs, we verify the effect of simplifying control commands through object manipulation tasks, including unique problems of mobile robots such as navigation to the target. We consider the problem that collecting floating objects with a UUV as a simple manipulation task. For simple operation, we develop a system to collect the floating object with UUV by selecting the target on the GUI.

To evaluate the effectiveness of the proposal in this dissertation, we conducted experiments using a dynamic simulator and an actual UAV and UUV. Through the experiment of pushing motion with a simulated UUV, it is confirmed that the pushing motion can be achieved only using simplified control commands. About the subject experiment of the UUV, it was confirmed that the workload was reduced, and the usability was improved as the effect of simplifying the command. Furthermore, we tried collection using the actual vehicle and confirmed its practicality.

Keywords:

Aerial manipulation, Cooperative system, Mobile robot, Pushing motion, Simplification, Teleoperation

*Doctoral Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, June 18, 2021.

物体操作のためのUAV/UUVの動作生成*

白倉 尚貴

内容梗概

移動ロボットの研究において完全な自律行動は一つの目標といえるが，センシングノイズや予期せぬ障害物など不確実性が多い実環境においては技術的にいまだ実現困難な課題が多い．そのため，より直近の目標として人による遠隔操作を含めた半自動システムについても議論されている．近年普及し始めた Unmanned Aerial Vehicle (UAV) や Unmanned Underwater Vehicle (UUV) は，空中または水中で人に代わって物体操作などの作業を行うことができる移動ロボットとして活躍が期待されている．しかしながら，これらのロボットは6自由度を同時に制御する必要があるため，完全な手動操作でロボット自体の制御と物体操作を同時に操作することは困難であり，制御の自動化や手動操作の簡略化が必要とされる．

本論文では，6自由度の制御を必要とする UAV や UUV について，より簡易な指令のみで目的の物体操作タスクを達成するための動作生成の手法について検討する．UAV 単体の制御については，オープンソースのコントローラにより目標位置の指定による自律制御や，姿勢制御のみの自動化など半自動操縦による操作の簡略化がすでに実現されている．しかしながら，既存のコントローラでサポートされているのは UAV 単体の制御のみであり，環境との接触や力制御を必要とする動作はサポートされていない．そこで，そのような物体操作の一例として，UAV による押し動作の際に簡易な指令を用いて動作生成を行う手法について提案する．UUV については，目標までのナビゲーションなど移動ロボット特有の問題を含めた物体操作タスクを通して，制御指令の簡略化による効果を検証した．水面の浮遊物を UUV によって回収するというシンプルな物体操作タスクを例に，UAV との連携によって簡易な操作で作業を達成可能なコントローラ及び GUI を提案した．

本論文における提案の有効性評価のため，動力学シミュレータ及び UAV と UUV の実機を用いた実験を行った．UAV による押し動作については，動力学シミュレーションにより設計通りの動作を確認した．UUV による浮遊物回収については，シミュレータを使用した被験者実験により，提案手法による作業負荷の軽減及び作業効率とユーザビリティの向上を確認した．

キーワード

空中マニピュレーション、協調システム、移動ロボット, 押し動作, 簡略化, テレオペレーション

*奈良先端科学技術大学院大学 先端科学技術研究科 博士論文, 2021年6月18日.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Mobile robots can move and execute tasks such as object manipulation and are expected to work instead of humans in dangerous environments. For example, rescue and disaster sites. In research on mobile robots, completely autonomous work is one of the ultimate goal. However, completely autonomous control in a real environment is still difficult because there are many uncertainties, such as noise and unexpected obstacles. Therefore, as an immediate goal, technologies for autonomous control such as localization and mapping methods [1, 2, 3, 4] and semi-automatic systems, including remote control by humans, are also being discussed.

Unmanned Ground Vehicle (UGV) is a classic mobile robot, which uses wheels to move. UGVs can move on the ground that has three degrees of freedom (DoF) of mobility. Also, the Unmanned Aerial Vehicle (UAV), which can fly and move in the air, and Unmanned Underwater Vehicles (UUVs), which can move underwater, are recently used for industry or general use. Since UAVs and UUVs can work in high place or underwater instead of humans, object manipulation with UAVs and AUVs are also expected [5, 6, 7]. Unlike UGVs, UAVs and UUVs move in the three-dimensional (3D) space with six DoF control, so it is difficult to perform object manipulation while controlling the mobile robot itself with manual operation. Besides, since the UGVs are on the ground, they can easily keep their position and work stably, but UAVs and UUVs are required effort to keep their position. Therefore, to perform object manipulation with remote-controlled UAVs or UAVs, a system that lets the operator concentrate on the target task is required.

This dissertation discusses a motion generation method to simplify the operation during object manipulation for a mobile robot with six DoF control such as UAV and UUV. The difficulty of object manipulation with UAV/UUV is that the object manipulation task and the motion generation of the mobile robot must be solved simultaneously. We propose a system that automates low-level control and allows the operator to control the vehicles using simplified commands directly related to the target task. For example, in the case of moving an object

with a UUV/UAV, the operator inputs only the goal position, and UAV/UUV trajectory generation and actuator control are automated. In this dissertation, we verify a system that controls a mobile robot based on simplified commands by the following two approaches: 1) We construct a method to achieve a specific object manipulation task with a UAV using only a simplified control command as input. 2) The efficacy of simplified control commands is verified through the application of object manipulation with a UUV.

Regarding UAVs, there are several existing controllers for UAVs [8]. we focus on the control which cannot be supported by open-source flight controllers. Existing controllers only support the control of a UAV alone, and does not support manipulation task that requires contact with the environment or force control. For example, pushing motion with a UAV. We verify a method to achieve a pushing motion with a UAV using simplified control commands. When designing a controller, we reuse the available parts of an existing controller for effective development. Then, we propose a method to calculate the input to the reused controller using a simplified command based on the task, such as pushing distance.

Regarding UUV, depending on the situation, the case where it can be approximated to the control in the 2D plane and the control in the 3D space are used properly, and in the control in the 3D space, in order to simplify the motion generation, humans We will verify a complex control method that provides real-time feedback based on the operation trajectory by manual operation. Compared to the case of UUV, there are many factors that hinder measurements such as refraction and attenuation of light and bubbles and turbidity and the self-position estimation method and sensing that are already established in the air. There is a problem that the technology cannot be used as it is. Therefore, we will verify a system that simplifies operations based on the remote control by humans. Consider cooperation with UAV as a system for remote control of UUV. The position of the UUV on the UAV viewpoint is controlled by using the viewpoint from the air by the UAV. The operator simply looks at the UAV's point of view about the control method at that time. Instead of operating the UUV, design a controller that can navigate the UUV to the target position with a simpler operation. Since the UAV's viewpoint can only be used while the UUV is on the surface of the

water, we verify the control by the camera mounted on the UUV underwater. Since the control has six DoF in the water, the motion generation is simplified, and the combination with the feedback from the camera is also verified.

1.2 Dissertation outline

Section 1.3 summarizes related research and current issues. Chapter 2 describes the pushing motion with a UAV. Chapter 3 describes the collecting floating object with a UUV. Chapter 4 discusses the possibility of object manipulation by UAV/UUV using simplified command and its effect based on the experimental results.

1.3 Related work

1.3.1 Object manipulation with a UAV

Manipulation using a UAV is studied in the field of aerial manipulation [5, 6]. In aerial manipulation, many research methods use specific mechanisms for manipulation. Car *et al.* [9] used two manipulators mounted on a multi-rotor UAV to grasp an object and achieve a peg-in-hole task. Zhao *et al.* [10] have developed a UAV composed of deformable multi-link rotors. The object is wrapped around by the multi-link body, and lifted. In this case, the manipulator does not have to be mounted on the UAV. Since the payload of the UAV is limited, mechanisms of manipulation must be selected according to the purpose.

In other research, a tilted-propeller hexa-rotor UAV was used for physical interaction control [11, 12]. A general multi-rotor UAV cannot generate a horizontal force without tilting. On the other hand, a tilted-propeller hexa-rotor UAV can control its 6D pose independently and can generate force in any direction. Also, it is possible to increase the DOF by mounting a manipulator as in the Bellicoso's research [13].

Kim *et al.* [14] have implemented compliance control using the manipulator mounted on the UAV. The manipulator is used to handle the reaction force. In the case of a pushing motion, the vibration when the UAV contacts the object complicates pushing operations. Therefore, a mechanism or control for reducing shocks is required.

In other related works, multiple robots including UAVs work cooperatively. Staub *et al.* [15] have constructed a collaborative work system with an aerial manipulator and a manipulator installed on the ground. Kim *et al.* [16] have studied cooperative work with multiple UAVs. Two UAVs grasp both sides of an object and place it in a target position. A controller to compensate the external force of the other robot is constructed. In these cases, the robot and the object move together. When pushing objects, it is required to consider each movement of the UAV and the object.

Wopereis *et al.* [17] have developed a mechanism to control UAVs with physical interaction. A parallel link is mounted on the UAV and controlled based on the reaction force caused by the contact with the environment. Meng *et al.* [18] have developed a controller for pushing against a fixed object using a hexa-rotor UAV. The pushing force is generated by tilting the UAV while interacting with the object using a mounted 1-DOF manipulator. Using this system, they succeeded in generating the desired force. These research methods focus on contact with a fixed object such as a wall. A similar control is required because a reaction force is also generated in the case of the pushing motion for manipulation. In addition, in the case of pushing non-fixed objects, it is required to consider the friction between the target object and the ground.

Several research methods use manual control for pushing and moving an object. Gioioso *et al.* [19] have developed a control interface for force-based control of a UAV. A pen-style haptic device is used as the force input device. An operator controls the UAV using this interface and pushes an object manually in simulation. However, in order to realize the pushing motion in the autonomous control, it is necessary to construct a physical model of the push operation by the UAV to determine the input.

1.3.2 Pushing motion with a robot

There is a previous work using a ground mobile robot for pushing objects. Yoo *et al.* [20] have proposed a path planning method for pushing motion using a trailer-like mobile robot. Meriçli *et al.* [21] have proposed planning pushing motions of various shape objects using an UGV. Hakamata *et al.* [22] have generated motions for pushing objects using a humanoid robot with a physical model of the

pushing motion.

1.3.3 Cooperation with a UAV

As the collaboration between a UAV and other robots, a collaboration between a UAV and a UGV is proposed for rescue and/or inspection in several studies [23, 24, 25, 26, 27]. In these studies, UAVs are used to assist the UGV by taking advantage of its wide field of view and 3D mobility. Since UUV control has 6 DoF, the difficulty of manual operation is higher than UGV control, which has 3 DoF. Therefore, in this dissertation, we also simplify the operation of the UUV to reduce workload.

Similarly, UAV-UUV systems are proposed for rescue [28] and/or underwater inspection [29]. Keila *et al.* proposed a framework for cooperation between UUV and UAV, and as an example of an application, underwater inspection has been performed [29]. Since the inspection environment is fixed, the inspection can be performed by UUVs moving along a pre-programmed path. The UAV is used to collect data measured by UUVs and does not share UAV's views with the UUVs. Most similarly to our work, Xuesu *et al.* proposed a UAV-USV collaboration system that uses a UAV viewpoint to guide the USV to a rescue target [28]. The system proposed in this dissertation aims to reach the rescue target, and motion generation of UUVs for manipulation is not considered. This method provides users with the UAV's views after homography translation. In contrast, our system shows the UAV's views as is. We actually evaluate the pros and cons of two types of views experimentally.

1.4 Purpose

The main proposal of this dissertation is to simplify motion generation to achieve manipulation tasks with simpler input for multi-DOF mobile robots such as UAVs and UUVs that require six DOF control.

2. Pushing Motion with a UAV

2.1 Problem setting

UAVs are expected to be used in places where people and ground robots cannot easily access. For example, UAVs are used to search for survivors in disaster sites and for inspection of bridges. In those situations, manipulation capabilities are often required.

One of the methods of aerial manipulation is using a manipulator mounted on a UAV. In this case, the motion of the manipulator affects the motion of the UAV. Hence, the combined control of manipulator and UAV has been studied [30]. Grasping and manipulating objects using a UAV with a mounted manipulator has been also studied [31], [32]. These studies focus on grasping and lifting objects for manipulation. If the object is unliftable, other approaches such as pushing without lifting are required. When pushing objects with UAVs, position controllers that are generally used for controlling UAVs cannot be used, since physical contact with the environment is not expected. Therefore, it is required to design a controller for pushing motions.

Previous studies have shown that pushing motion with a UAV is possible [33]. In those studies, controllers were designed for each hardware. On the other hand, there are various types of UAVs having different numbers of propellers and DoF in their control. Hence controllers that can be commonly used with these different hardware have been developed.

In this chapter, we propose a pushing motion method using those existing controllers. The contribution is to design a reusable pushing motion system that can be used in combination with existing controllers. By doing so, the controller can be redesigned with less effort, and many types of UAVs can push unliftable objects. The controller is designed based on a physical model that includes the reaction force from the target object. In the proposed method, an attitude control part of the existing controller is used for considering the reusability of the pushing controller. In this dissertation, pushing objects following a straight line is considered.

We believe that the proposed method is fundamental to develop more complicated operations such as pushing objects and moving them in various directions.

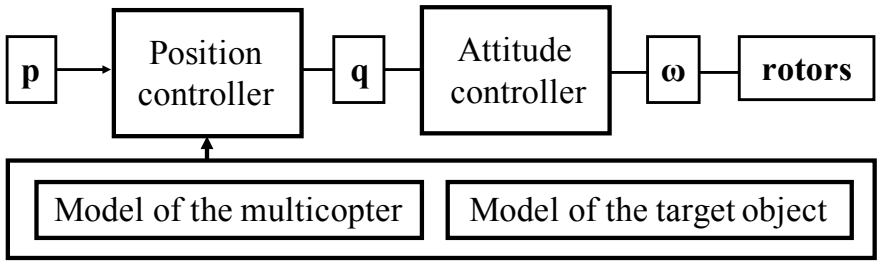


Figure 1. Overview of the proposed system.

2.2 Design of the controller

Fig. 1 shows the overview of the controller for pushing motions by a UAV. The controller is designed with two separate elements: a pushing motion controller and a UAV attitude controller. The purpose of the pushing motion is to move the target object. The moving distance of the target object $\boldsymbol{p}= [\Delta x\ \Delta y\ \Delta z]^T$ is given as the input of the system. The attitude and altitude of the UAV $\boldsymbol{q}= [\phi\ \theta\ \psi\ z]^T$ to achieve the pushing motion is calculated by the pushing motion controller based on the physical model. Then, $\boldsymbol{q}$ is input to the attitude controller as the target value of the attitude of the UAV. The attitude of the UAV is controlled by feedback control using an IMU sensor and a barometer.

2.3 Physical model of pushing motion

Fig. 2 shows the physical model of the pushing motion. In this dissertation, in order to achieve pushing an object in a straight line, the physical model is constructed in a two-dimensional space. The equation of motion in the vertical direction is given as:

$$M_{\mathrm{c}}\ddot{z} = f_{\mathrm{z}} - M_{\mathrm{c}}g\,, \tag{1}$$

where M_{c} is the mass of the UAV, z is vertical position of the UAV, g is gravitational acceleration, and f_{z} is the vertical component of the thrust generated by the propellers. Since the UAV keeps its altitude when pushing, f_{z} is calculated as follows:

$$f_{\mathrm{z}} = M_{\mathrm{c}}g\,. \tag{2}$$

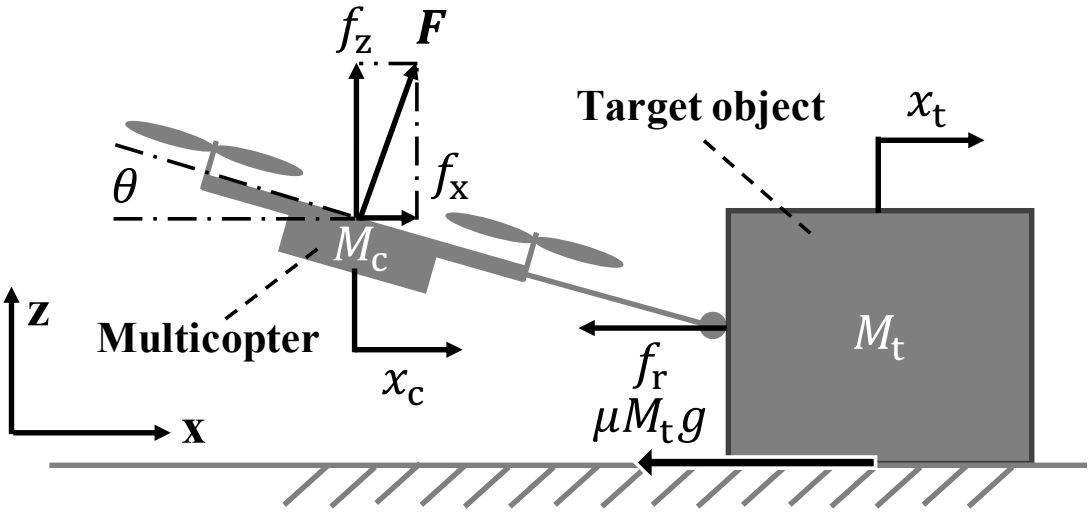


Figure 2. Physical model of the pushing motion.

The equation of motion in the horizontal direction can be written separately for the UAV and the target object. The equation of motion for the UAV is given as:

$$M_c \ddot{x}_c = f_x - f_r, \tag{3}$$

where f_x is the horizontal component of the thrust generated by the propellers, and f_r is the reaction force from the target object generated by the pushing motion. The equation of motion for the target object is given as:

$$M_t \ddot{x}_t = f_r - \mu M_t g\,, \tag{4}$$

where M_t is the mass of the target object, μ is the dynamic friction coefficient between the target object and the ground. Assuming that the UAV and the target object remain in contact during the pushing motion, the acceleration of each object has the following relation:

$$\ddot{x}_c = \ddot{x}_t = \ddot{x}\,. \tag{5}$$

According to Eq. (3), (4) and (5), the equation of motion of the target object is given as:

$$a(t) = \ddot{x} = \frac{f_{\mathrm{x}}(t) - \mu M_{\mathrm{t}} g}{M}\,, \tag{6}$$

where $a(t)$ is the acceleration of target object, $M = M_{\mathrm{c}} + M_{\mathrm{t}}$ is the sum of the copter mass and target mass, and $f(x)$ is the horizontal force generated by the UAV.

2.4 Assigning pushing force

To solve Eq. (6), it is required to determine $f_{\mathrm{x}}(t)$. $f_{\mathrm{x}}(x)$ is calculated as the horizontal component of the thrust of the UAV as follows:

$$f_{\mathrm{x}}(t) = f_{\mathrm{z}} \cdot \tan\theta \simeq f_{\mathrm{z}}\theta\,, \tag{7}$$

where θ is the pitch angle of the UAV. If θ is too large, the UAV will not be able to keep its posture. Therefore, it is necessary to push the object with the minimum θ required to push it. However, if θ is too small, the object will not move. In this dissertation, as we assume that the mass of the object is between 2.0 and 2.5 [kg], the mass of the UAV is 1.3 [kg] and the friction is 0.1, the minimum θ is 11 [deg], and 12 [deg] with some margin. Since $\theta = 12$ [deg] ~ 0.21 [rad] is small enough compared to 1.0, $\tan\theta$ can be approximated as a linear function using the Maclaurin's expansion.

The state of the pushing motion can be divided into the following three phases: during pushing the object, when the UAV is returning to the horizontal attitude, and braking by friction. Fig. 3 shows the model of the pushing force corresponding to the three phases. T_1, T_2 and T_3 are the durations of each of the three phases of the UAV.

In the part T_1, the pitch angle is controlled to the target value using PID control. Fig. 4 shows the time function of θ during the pushing motion. We assume that the PID parameters are adjusted so that the UAV's attitude control can be approximated to critical dumping, which has good response speed without overshoot [34]. The pitch angle in the part T_1 is formulated using a critical damping model as follows:

$$\theta(t) = \hat{\theta}(1 - e^{-\omega t})\,, \tag{8}$$

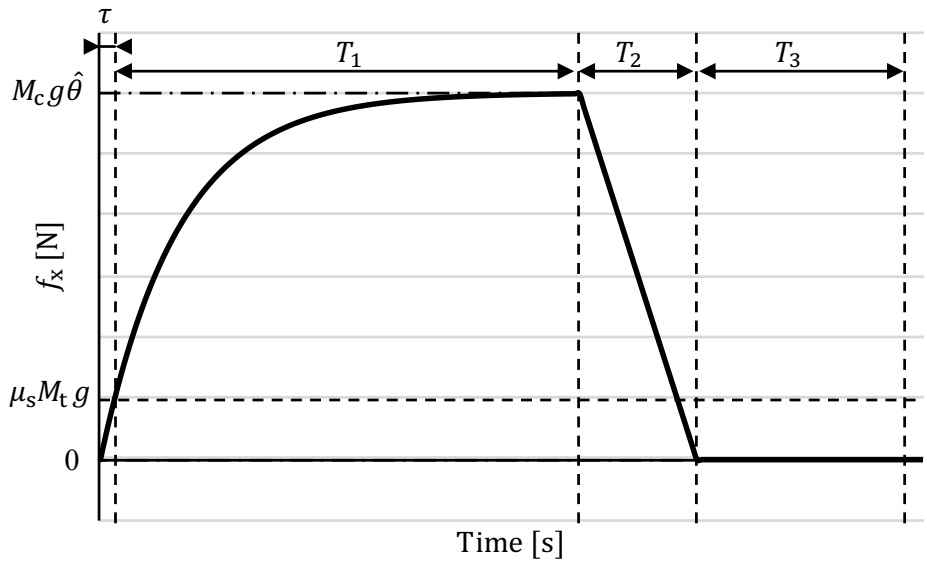


Figure 3. Model of the pushing force generated by the UAV.

where $\hat{\theta}$ is the target pitch angle of the UAV, and ω is a natural angular frequency. $\hat{\theta}$ is determined using the following condition:

$$f_z\hat{\theta} > f_s\,, \tag{9}$$

where $f_s = \mu_s M_t g$ is the friction force between the target object and the ground, and μ_s is the static friction coefficient.

The natural angular frequency ω can be calculated using the measured transition of the pitch angle. Fig. 5 shows a logarithmic scale of the transition of the pitch angle. The logarithmic scale of the pitch angle represented by the solid line in Fig. 5 is approximated using Eq. (8) as follows:

$$y = \omega t\,. \tag{10}$$

Eq. (10) means that ω is given as a gradient of the linear approximation. The dotted line in Fig. 4 is a linear approximation when the gradient is 5.0. Similarly, the dotted line in Fig. 5 is an exponential approximation when ω is 5.0. In this dissertation, the transition of the pitch angle during the pushing motion at the target 12 degrees is approximated by Eq. (8) and $\omega = 5$.

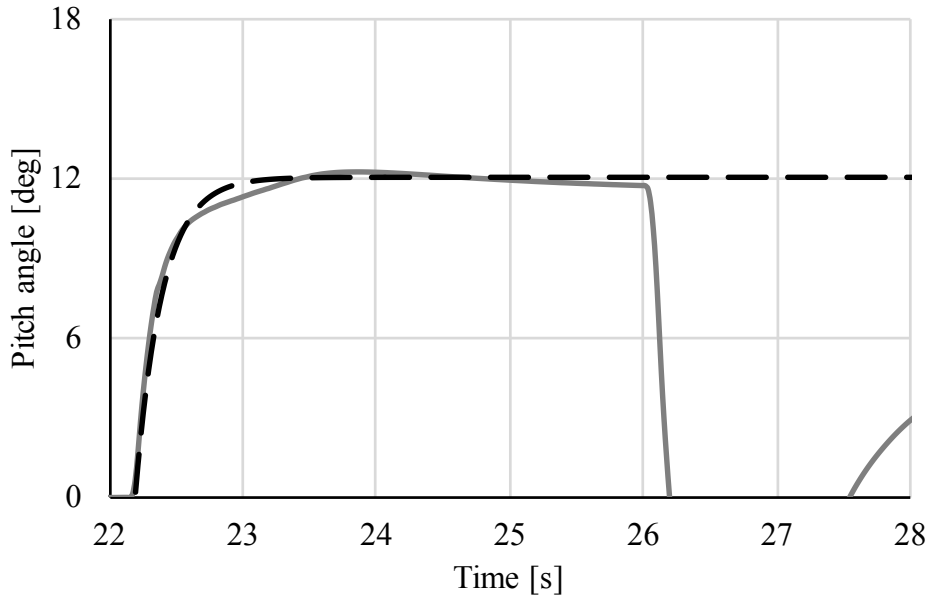


Figure 4. Pitch angle of UAV during pushing motion. The solid line is pitch angle of UAV. The UAV is controlled using the attitude part of the existing flight controller. the designed target pitch angle is 12 [deg]. The dotted line is exponential approximation.

Using Eq. (7) and (8) the pushing force in T_1 is given as:

$$f_{\mathrm{x1}}(t) = F_{\hat{\theta}}(1 - e^{-\omega t})\,, \tag{11}$$

where t is the time variable from the beginning of the pushing motion, $F_{\hat{\theta}} = M_{\mathrm{c}} g \hat{\theta}$ is the horizontal force when the pitch angle of the UAV is $\hat{\theta}$.

T_2 is the duration for the pitch angle of the UAV to return to the horizontal attitude. The pushing force during T_2 is given by linear approximation as:

$$f_{\mathrm{x2}}(t) = -\frac{F_{\hat{\theta}}}{T_2} t + F_{\hat{\theta}}(1 + \frac{T_1}{T_2})\,. \tag{12}$$

T_3 is the duration of the target object moving with inertia. Since the UAV does not apply the force to the target object in T_3, the pushing force is given as:

$$f_{\mathrm{x3}}(t) = 0\,. \tag{13}$$

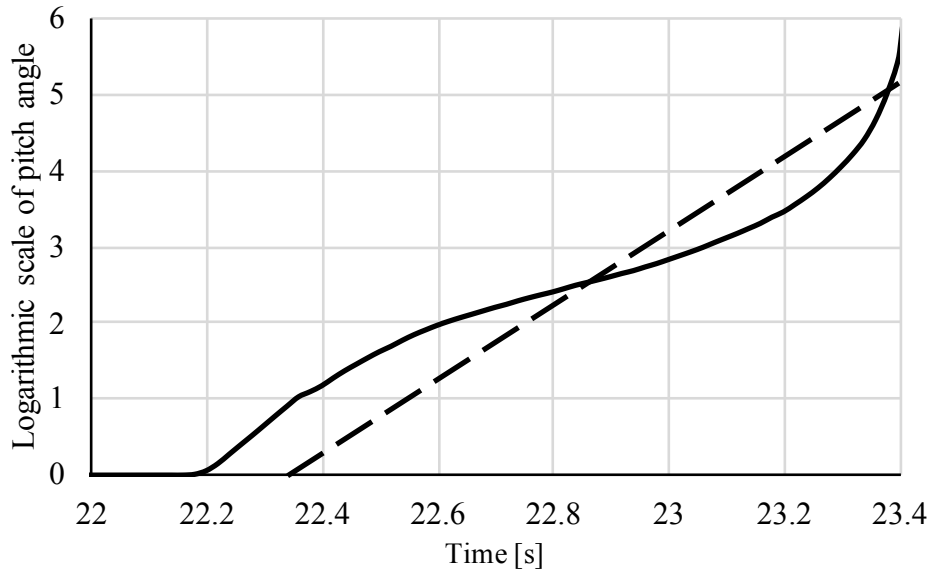


Figure 5. Logarithmic scale of transition of pitch angle. The solid line is the actual data. The dotted line is a linear approximation.

2.5 Calculating pushing duration

The position of the target object is calculated by substituting Eq. (11), (12) and (13) into Eq. (6) and integrating. However, the pushing force is smaller than the frictional force in the section τ of Fig. 3, and the object does not move. Therefore, the integration range is from τ to the end of the part T_1. Using Eq. (11), τ is given as:

$$\tau = -\frac{1}{\omega}\log_e(1 + \frac{f_{\mathrm{s}}}{F_{\hat{\theta}}})\,. \tag{14}$$

When the initial velocity and the position are zero, the position of the target object is calculated as follows:

$$x(t) = C_1 T_1^2 + C_2 T_1 + C_3\,, \tag{15}$$

where C_1, C_2, C_3 are constants of the equation of the position of the target object. The parameters included in these constants are determined uniquely by determining the angle of the UAV during the pushing motion. From Eq. (15) to

T_1 is given as:

$$T_1 = \frac{-C_2 + \sqrt{C_2^2 - 4C_1(C_3 - D)}}{2C_1}. \tag{16}$$

T_1 is the time during which the target angle is given in actual control, $D = x(T_1 + T_2 + T_3) - x(0)$ is the target distance of the pushing operation. Therefore, if the UAV is controlled based on T_1 calculated by Eq. (16) and the target angle θ, the target object can be pushed to the target position $x(t)$.

2.6 Attitude control

Fig. 6 shows a dynamics model for the attitude control of a quad-rotor type of a UAV. In the case of a quad-rotor, four states of roll, pitch, yaw and altitude can be controlled independently using four rotors. The equations of motion for attitude and altitude are given as:

$$\begin{cases} M_{\mathrm{c}}\ddot{z} &= K_1(\omega_1 + \omega_2 + \omega_3 + \omega_4)\cos\phi\cos\theta - M_{\mathrm{c}}g \\ I_{\mathrm{x}}\ddot{\phi} &= K_2(\omega_3 - \omega_4) \\ I_{\mathrm{y}}\ddot{\theta} &= K_2(\omega_2 - \omega_1) \\ I_{\mathrm{z}}\ddot{\psi} &= K_3(\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2)\,, \end{cases} \tag{17}$$

where K_1, K_2 and K_3 are physical properties of the propellers. The equation between attitude and rotor is given as:

$$\mathrm{u} = \begin{bmatrix} u_{\mathrm{z}} \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}. \tag{18}$$

2.7 Control flow

The control of the UAV for the pushing motion consists of an approach phase and a push phase. The physical model of the pushing motion in Sec. 2.2 is used when calculating the pushing duration in the push phase. In the approach phase, the UAV moves to the starting point of the push motion. Since the external force is not exerted in this phase, an existing position controller can be used.

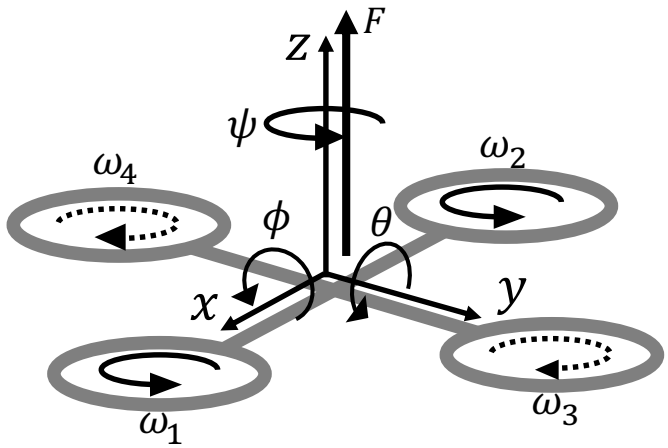


Figure 6. Model of UAV control.

Algorithm 1 shows the algorithm of the approach phase. In the approach phase, a tight approach is required to push the target object according to the physical model shown in Fig. 2. Furthermore, since the initial speed is zero, the UAV must stop before starting the pushing motion. Therefore, the UAV approaches the target object little by little in the Algorithm 1.

In the push phase, the attitude of the UAV is controlled based on the time calculated using Eq. (16). Algorithm 2 shows the algorithm to control the attitude of the UAV during the push phase. In this phase, the UAV is controlled using the existing attitude controller. The target attitude of the UAV $\boldsymbol{q} = [\phi\ \theta\ \psi\ z]^T = [0.0\ \hat{\theta}\ 0.0\ Z]^T$ is given to the existing attitude controller. The target altitude of the UAV Z is the height of the center of the target object. To move the target object to the target distance D, the attitude of the UAV is kept during T_1. After that, the UAV is returned to the horizontal attitude.

2.8 Simulation

2.8.1 Experimental setting

The pushing motion using the proposed method was tested in simulation. Fig. 7 shows the UAV and the target object in simulation. Erle-Copter which is a quadrotor UAV made by Erle Robotics was used for the experiment. *Ardupilot* which

Algorithm 1 Approach to the target object

```
procedure ApproachToTarget(d)  ▷ d is threshold of distance to target
    while ‖v‖ > d do
        p_c ⇐ CopterPosition
        p_t ⇐ TargetObjectPosition
        v ⇐ p_t − p_c
        v ⇐ v/2
        SetPosition(p_c + v) ▷ set target position of the UAV using an existing controller.
    end while
end procedure
```

Algorithm 2 Attitude control to push the object

```
procedure AttitudeControl(T_1, θ̂)
    while t < T_1 do                          ▷ t is time in the while loop
        SetAttitude(0.0, θ̂, 0.0, Z)
    end while
    while θ > 0.0 do
        SetAttitude(0.0, 0.0, 0.0, Z)         ▷ return horizontal
    end while
end procedure
```

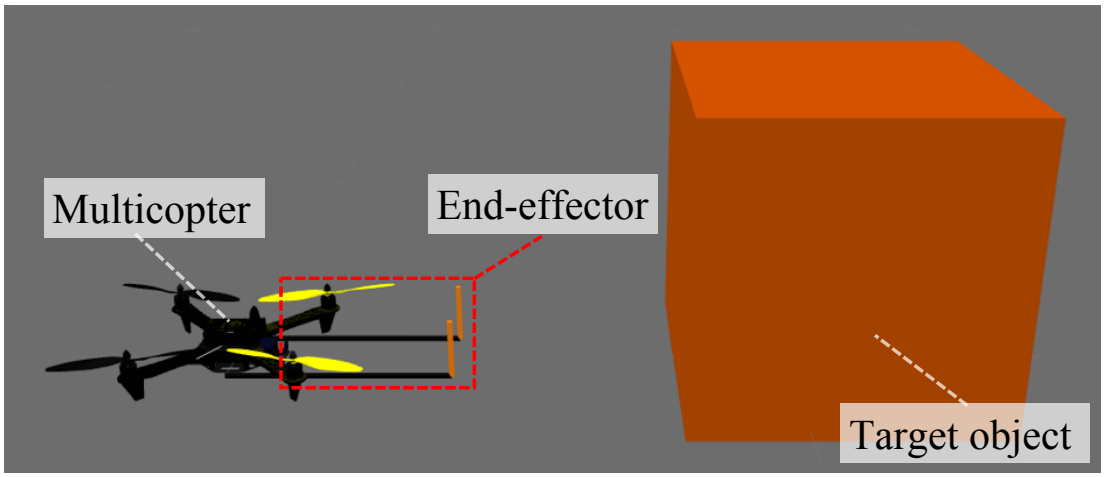


Figure 7. UAV which is used in simulation.

is an open-source flight controller is used for the control of the Erle-Copter. An end-effector consisting of two parallel sticks is mounted on the Erle-Copter to push the object. The simulation environment with Gazebo which is provided by Erle Robotics is used. The system is constructed based on ROS (Robot Operating System). The UAV is controlled using Algorithm 1, 2.

In this dissertation, the physical model was constructed for a pushing motion controller that can reuse the existing attitude controller. In the simulation, the object is pushed under different conditions to evaluate the effectiveness of the physical model. The conditions of the experiment are three patterns in total with different mass of target and pushing distance. In this simulation environment, the results vary slightly between each trial due to the lack of real-time control. Therefore, to test the accuracy of the proposed method, the experiment is repeated 10 times per condition. The friction coefficient between the target object and the floor is 0.1 in all patterns.

2.8.2 Result

Fig. 8 shows a demonstration of the simulation experiment. Fig. 9 shows the motion of the target object during the pushing motion. Table 1 shows the error and standard deviation of the simulation results. The error is calculated as Root Mean Square Error (RMSE) between goal position and target position. The

Table 1. Error and Standard Deviation of Simulation Experiments.

Mass of the target object [kg]	Pushing distance [m]	RMSE	SD
2.0	1.0	0.037	0.025
2.0	2.0	0.037	0.036
2.3	2.0	0.050	0.050

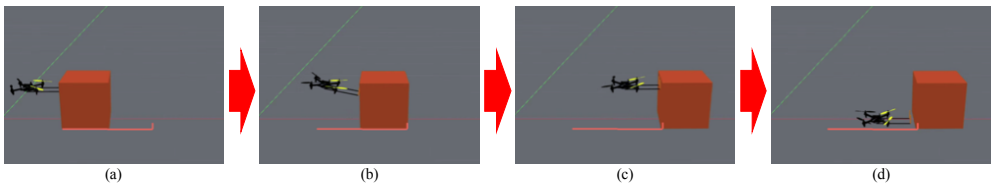
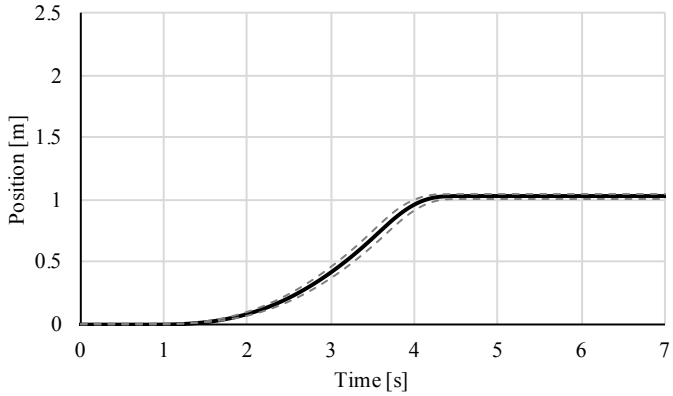


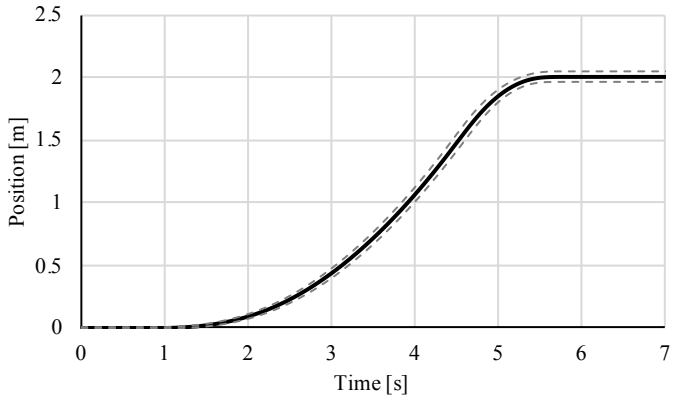
Figure 8. Simulation experiments. (a) start of the pushing motion, (b) pushing the target object, (c) end of pushing the object, and (d) landing after the pushing motion.

deviation is calculated as Standard Deviation (SD) of 10 trials.

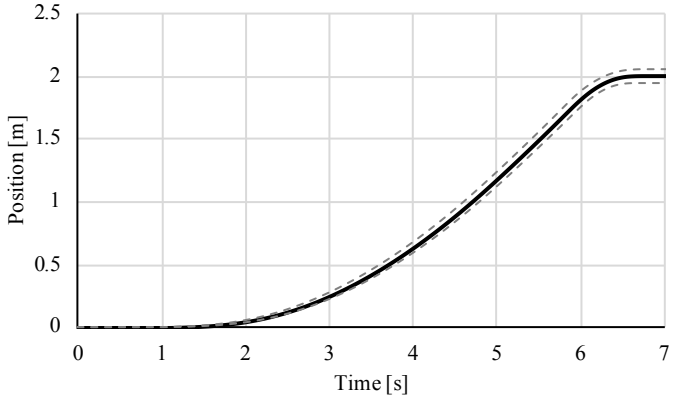
The target pushing distance is achieved as shown in Fig. 9 and Table 1. Since the standard deviation is small in Table 1, the pushing motion is achieved equally. Whereas, Table 1 also shows that as the weight and distance increase, the error and the variance tend to increase. The error is attributed to the lack of real-time control. Since ROS, used to construct the proposed system, does not guarantee real-time performance, the result has variance.



(a) 2.0 [kg], 1.0 [m].



(b) 2.0 [kg], 2.0 [m].



(c) 2.3 [kg], 2.0 [m].

Figure 9. Result of simulation experiments. The solid line is the average of the position of the target object, and the dotted line is the standard deviation.

3. Supporting UUV Teleoperation with a UAV

3.1 Problem setting

Marine debris is a serious environmental problem. In particular, plastic waste floating on the water surface is a non-negligible problem because it causes accidental ingestion in marine organisms, obstruction of ships navigation, and economic loss to marine industries. The current solution to this problem is manual collection using ships or divers. However, since the waterside is a harsh environment for humans, more efficient collection methods are required to reduce the labor and cost of collection.

The environmental organization *Ocean Cleanup* has proposed a method to collect floating debris by installing a net of more than 100 meters [35]. Although this method can efficiently collect floating debris in the vast ocean, it is not available in complicated topographies such as a port where marine debris tends to accumulate. To solve this problem, it is possible to collect floating debris using an UUV, which is suitable for the complicated topography of ports. Since a UUV can be operated underwater by remote control, it can reduce the labor for underwater work, and it can be used in various environments such as docks and seabed [36, 37].

When controlling UUVs, due to the view limitation of UUVs, it is difficult to understand the situation around them. For example, in sea rescue, aircraft or UAVs are often used to search for targets [28]. By searching from the air, it is possible to find objects with a wider view. This dissertation proposes an efficient system for collecting marine debris by making a UAV and a UUV collaborate.

Regarding detection of debris, a convolutional neural network (CNN) can detect floating objects on the water surface automatically, but more datasets are required to detect marine debris with various appearances [38]. In the case of teleoperation, which skips automatic detection, we would like to reduce the workload as much as possible by automating operation elements except for recognition, such as the control of the UUV. Hence, we design a GUI controller, where a user selects the target object on the UAV viewpoint, and then the UUV automatically collects it. By using this GUI, even people who have no experience in operating UUVs can collect floating debris easily and efficiently with teleoperation.

To collect debris, a method for estimating the position of debris on an actual scale, which is used for an accurate approach, is required. In this dissertation, we propose a simple method to measure the 3D-position of debris with a monocular camera mounted on a general UAV. By assuming that both the debris and the UUV are on the water surface, the normal and position of the water surface can be estimated based on the UUV pose, and the 3D position of the debris can be calculated from them.

We focus on the motion generation and control of the UUV. And we use a simple mechanism that can collect floating objects by the motion of the UUV only, such as scooping up. This simplicity enables the installation on various types of UUV and does not require any additional power supplies. The motion of the UUV to collect debris after approaching it is generated by teaching playback in advance; the generation does not concern the users. In this dissertation, we focus on the process until the UUV catches debris, and the processes after collecting debris, such as carry the collected debris, are future work.

To evaluate the collecting efficiency of our system, we conducted an experiment of collecting floating debris using a simulator and compared the time to collect all debris with the case of using only the view from the UUV and the related work [28]. We also evaluated usability and workload of our system using evaluation index such as the System Usability Scale (SUS) [39] and the NASA Task Load Index (NASA-TLX) [40]. Furthermore, our system successfully collected floating debris in a pool using real robots; thus, our experiments demonstrate the practicality of the proposed method.

3.2 System overview

Fig. 10 shows the overview of the proposed system. In the proposed system, the UAV shown in Fig. 10 (A) is used to search for marine debris. Searching from the air with a UAV provides a 360-degree view to locate debris despite the limited view of the UUV. The UUV then initiates the scooping motion shown in Fig. 10 (C) to collect the debris using the collection mechanism shown in Fig. 10 (B). The operator teaches the scooping trajectory and the initial condition (e.g., relative position $\boldsymbol{p}_0$ between debris and a UUV) once. The UUV then playbacks the trajectory when the initial condition is satisfied.

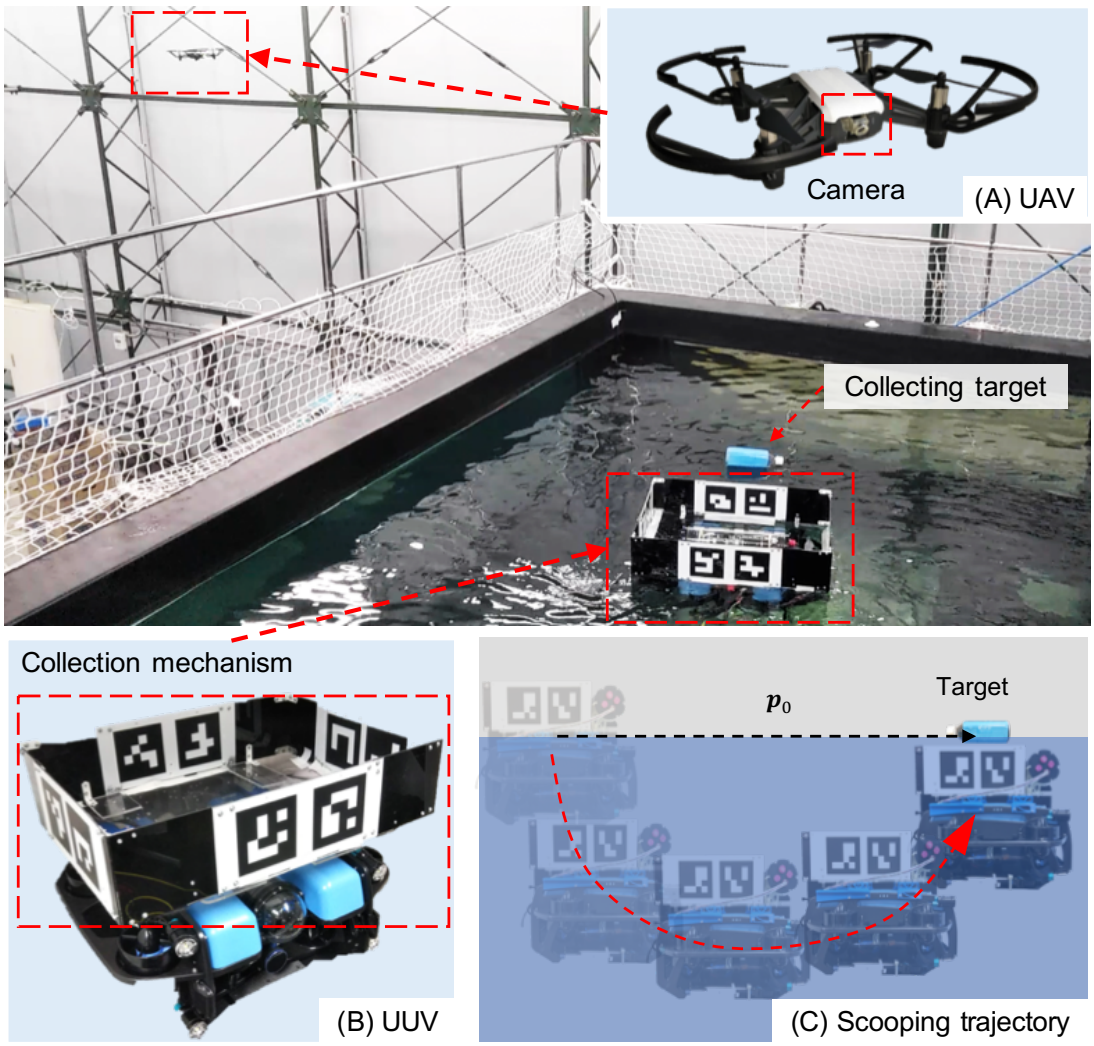


Figure 10. Overview of our collaborative UAV-UUV system to collect marine debris. (A) The UAV with a camera for localization of debris. (B) The UUV with a mechanism for collecting debris. (C) The scooping trajectory to collect debris.

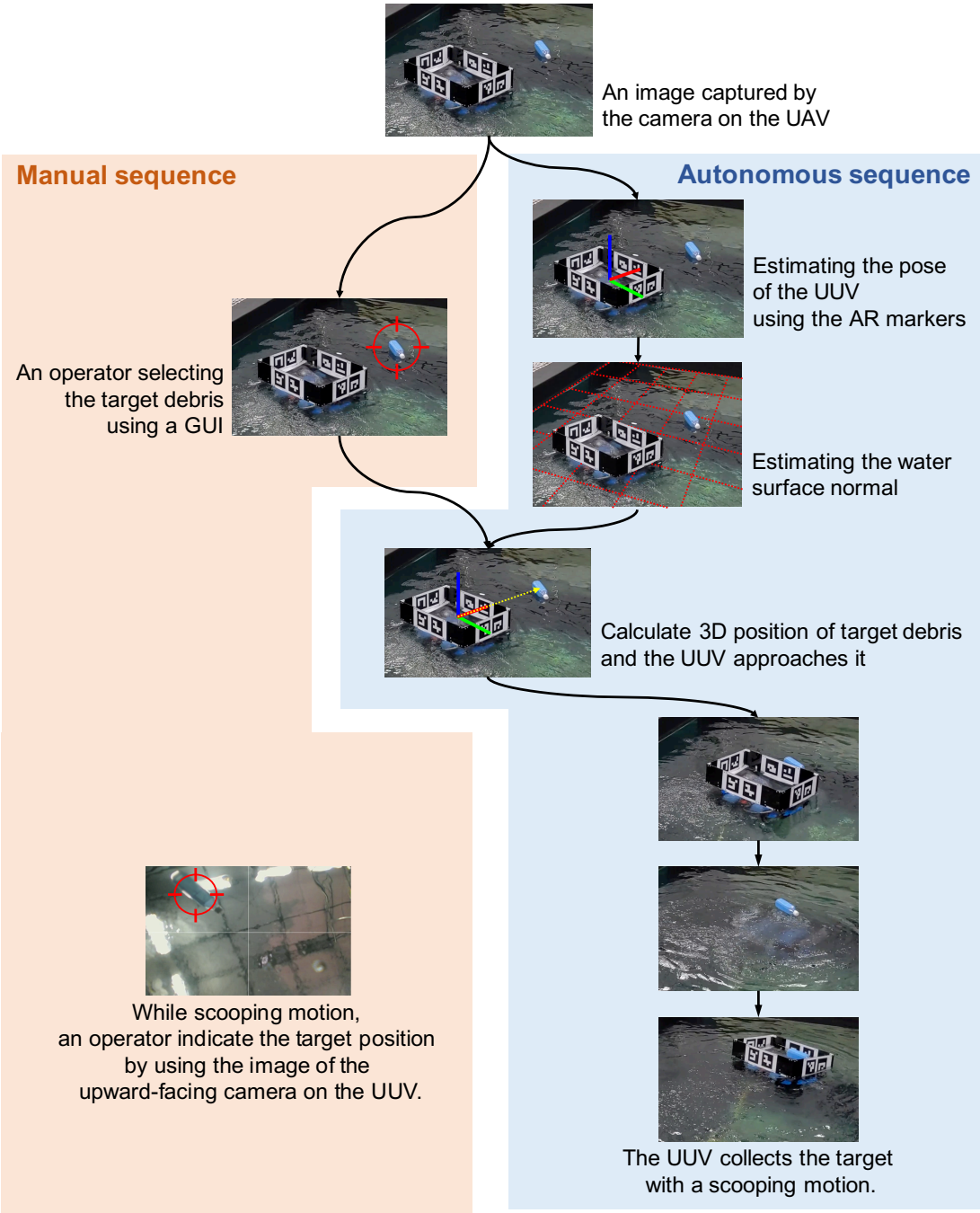
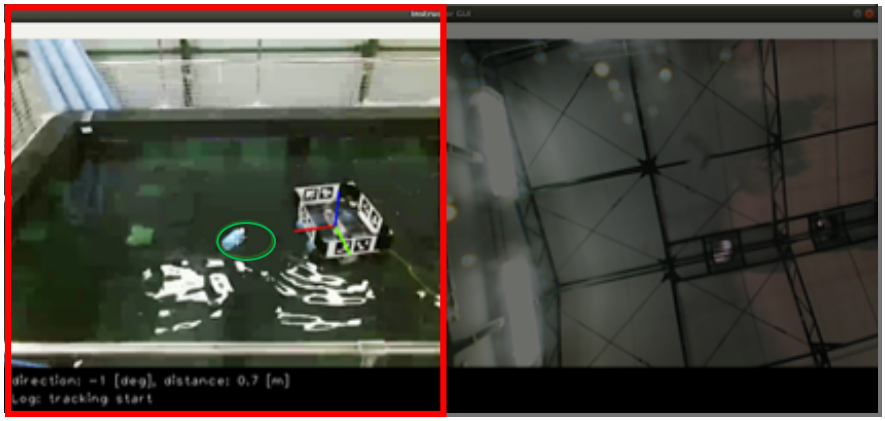


Figure 11. Whole process of the proposed system.

(a) Step 1: Press the target to approach it until the red circle turns green.



(b) Step 2: Release the target and start scooping motion.

(c) Step 3: Correct the trajectory by pressing the target on the right side of the GUI screen.

Figure 12. Semi-automatic collection procedure of our GUI controller.

Fig. 11 shows the process of the proposed system. The system consists of autonomous and manual sequence parts. By using the image from the UAV's camera, the pose of the UUV in the UAV frame can be measured using AR markers. The distance and normal of the water surface are estimated from the pose of the UUV in the UAV frame. The target debris on the camera image is selected manually by an operator. The 3D position of the target can be calculated as the intersection between the view direction and the water surface. The UUV will approach the target based on the calculated 3D position. After the UUV is sufficiently close to the target, the UUV collects the target with the scooping motion shown in Fig. 10 (C).

Fig. 12 shows the GUI used to control the system. A camera image from the UAV is displayed on the left side of the GUI. The operator selects the target by pressing and holding on one piece of debris on the GUI (Fig. 12 (a)). Since debris is not stationary due to the water flow, the position of the target debris can be updated by dragging. The red circle in Fig. 12 (a) is displayed at the relative position $\boldsymbol{p}_0$ between the debris and the UUV recorded at the same time as the scooping motion. When the target debris enters the red circle, the circle turns green to inform the operator that the UUV is sufficiently close to the target for collection (Fig. 12 (b)). Then, by releasing the target on the GUI, the scooping motion is started. The right side of the GUI shows an image of the upward-facing camera on the UUV to observe the target object during scooping motion. The operator can adjust the scooping trajectory of the UUV by pressing the target on the right side of the GUI (Fig. 12 (c)).

3.3 Projection to the water surface

Fig. 13 shows the perspective projection from points on the camera image to the 3D position on the water surface. The 3D position of the selected debris on the UAV frame ${}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{t}} = [\ {}^{\mathrm{a}}p_{\mathrm{t}x}\ {}^{\mathrm{a}}p_{\mathrm{t}y}\ {}^{\mathrm{a}}p_{\mathrm{t}z}\ 1\ ]^T$ is constrained on a line as follows:

$$ {}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{t}} = a\boldsymbol{v} = a\boldsymbol{M}^{-1}\ {}^{\mathrm{ac}}\boldsymbol{p}_{\mathrm{t}}\,, \tag{19} $$

where $\boldsymbol{M}$ is the camera matrix of the UAV's camera, ${}^{\mathrm{ac}}\boldsymbol{p}_{\mathrm{t}}$ is the position of the target debris on the UAV camera frame, $\boldsymbol{v}$ is a vector from the UAV's camera center to the image point of the selected debris, and a is any scalar value. Since

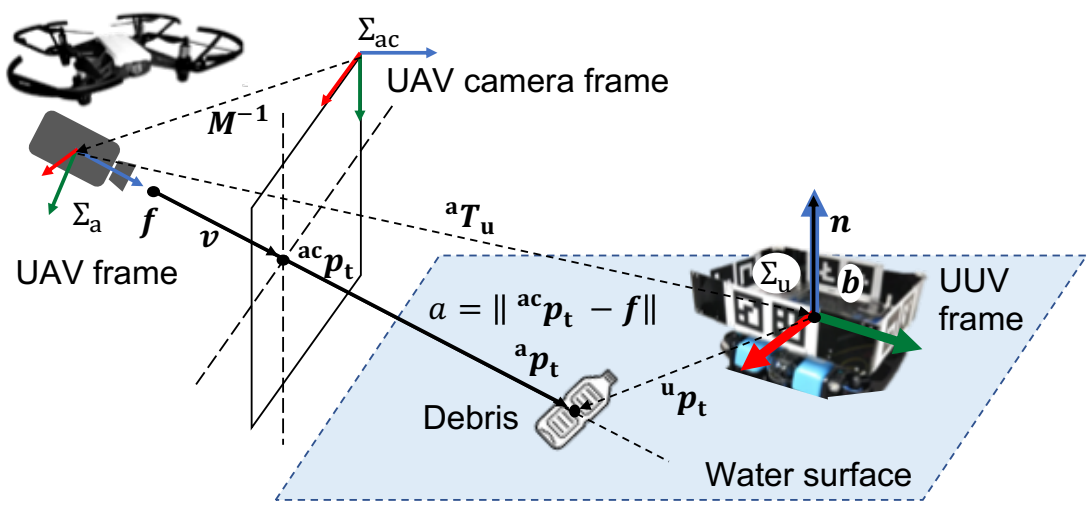


Figure 13. A perspective projection from a point on the camera image to a 3D position on the water surface.

debris floats on the water surface, the following constraint is given:

$$\boldsymbol{n} \cdot ({}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{t}} - \boldsymbol{b}) = 0\,, \tag{20}$$

where $\boldsymbol{n}$ is a normal vector of the water surface, and $\boldsymbol{b}$ is any point on the water surface. If the UUV is horizontal on the water surface, $\boldsymbol{n}$ and $\boldsymbol{b}$ can be calculated as follows:

$$\boldsymbol{b} = {}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{u}}\,, \tag{21}$$

$$\boldsymbol{n} = {}^{\mathrm{u}}\boldsymbol{T}_{\mathrm{c}}\boldsymbol{p}_{\mathrm{u}z}\,, \tag{22}$$

where ${}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{u}}$ is the position of the UUV on the UAV frame, $\boldsymbol{p}_{\mathrm{u}z} = [\ 0 \quad 0 \quad 1\ ]^T$ is a vertically upward unit vector in the UUV frame, and ${}^{\mathrm{u}}\boldsymbol{T}_{\mathrm{c}}$ is a transformation from the UUV to the UAV camera frame. As mentioned above, ${}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{u}}$ and ${}^{\mathrm{u}}\boldsymbol{T}_{\mathrm{c}}$ are obtained from the relative position of the UAV and the UUV. We obtain them using AR markers.

Using Eq. (19) and Eq. (20), a is calculated as follows:

$$a = \frac{\boldsymbol{n} \cdot \boldsymbol{b}}{\boldsymbol{v} \cdot \boldsymbol{n}}\,. \tag{23}$$

The 3D position of the debris in the UUV coordinate ${}^{\mathrm{u}}\boldsymbol{p}_{\mathrm{t}}$ is calculated using

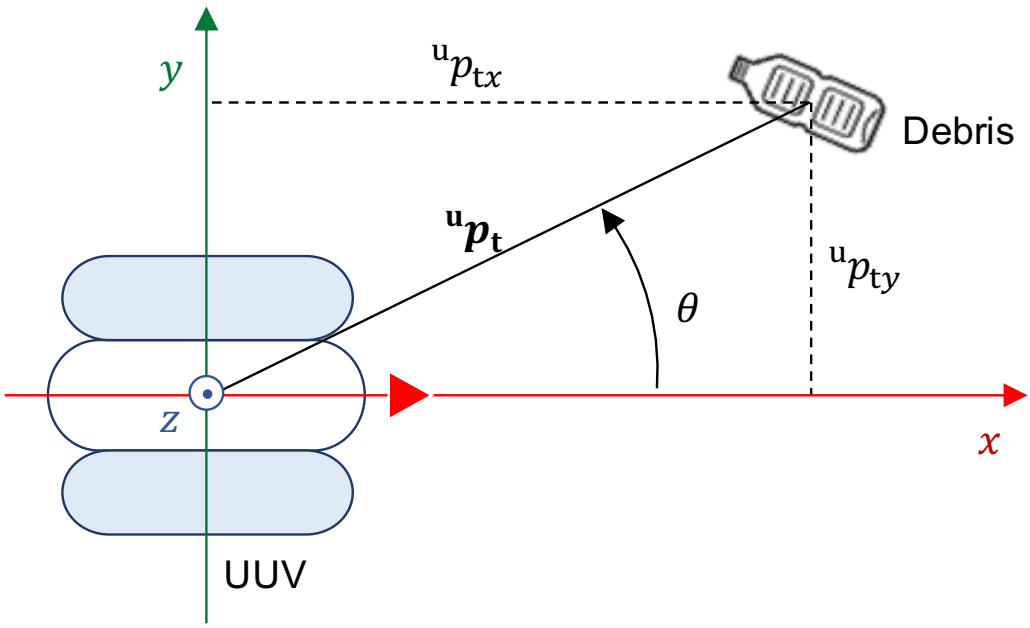


Figure 14. Example of the relative position of debris and the UUV.

the transformation from the camera to the UUV ${}^{\mathrm{a}}\boldsymbol{T}_{\mathrm{u}}$ as follows:

$$ {}^{\mathrm{u}}\boldsymbol{p}_{\mathrm{t}} = {}^{\mathrm{a}}\boldsymbol{T}_{\mathrm{u}}\,{}^{\mathrm{a}}\boldsymbol{p}_{\mathrm{t}}. \tag{24} $$

3.4 Approaching target with UUV

To successfully collect debris, the UUV first needs to come close enough to the target debris. Fig. 14 shows a simple situation of the relative position of the UUV and debris. The UUV coordinate system is defined as a right-handed coordinate system in which the upward direction is $+z$ direction and the forward direction is $+x$ direction. In this case, since the movement of the UUV is limited to the water surface, the control of the UUV can be considered as a 2D-control. Therefore, a control input $\boldsymbol{u}$ of the UUV is defined as follows:

$$ \boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \\ u_r \end{bmatrix}, \tag{25} $$

where u_x and u_y are the horizontal control inputs, and u_r is the rotational control input. The UUV state $\boldsymbol{x}$ is defined as the positional difference between the UUV and the target. The state is formulated as follows:

$$\boldsymbol{x} = -\begin{bmatrix} {}^{\mathrm{u}}p_{\mathrm{t}x} \\ {}^{\mathrm{u}}p_{\mathrm{t}y} \\ \tan^{-1}({}^{\mathrm{u}}p_{\mathrm{t}y}/{}^{\mathrm{u}}p_{\mathrm{t}x}) \end{bmatrix}, \tag{26}$$

where ${}^{\mathrm{r}}p_{\mathrm{t}x}$, ${}^{\mathrm{r}}p_{\mathrm{t}y}$ are the x-y components of the position of the target. The deviation is defined using $\boldsymbol{x}$ as follows:

$$\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}\,, \tag{27}$$

where $\hat{\boldsymbol{x}}$ is a target state of the UUV, that is the state to start scooping motion. When $\boldsymbol{e}$ is less than a threshold value, it indicates that the UUV has come close enough to the target debris. To make $\boldsymbol{e}$ be less than the threshold value, the input is calculated using a PID controller, as follows:

$$\boldsymbol{u} = \boldsymbol{K}_{\mathrm{p}}\boldsymbol{e} + \boldsymbol{K}_{\mathrm{d}}\frac{d}{dt}\boldsymbol{e} + \boldsymbol{K}_{\mathrm{i}}\int \boldsymbol{e}dt\,, \tag{28}$$

where $\boldsymbol{K}_{\mathrm{p}}$, $\boldsymbol{K}_{\mathrm{d}}$, and $\boldsymbol{K}_{\mathrm{i}}$ are the gains of the PID controller and are determined by trial and error using a simulator and a real robot to converge quickly and not diverge.

3.5 Scooping motion for collection

Fig. 15 shows the overview of the collecting motion generation. The motion generator is divided into a teaching step that records a manual scooping motion and a playback step that generates scooping motion automatically based on the recorded motion. In the teaching step, debris is collected using the collection mechanism with manual operation. In the initial starting point before the trajectory teaching, the relative 2D position and orientation angle on the water surface between the UUV and the target debris $\boldsymbol{p}_0$ are recorded. $\boldsymbol{p}_0$ is used to decide the goal position when the UUV approach to the target as follows:

$$\hat{\boldsymbol{x}} = -\boldsymbol{p}_0\,. \tag{29}$$

The control input during the manually controlled trajectory are also recorded as time series data with a time stamp. The control input consists of four channels:

up-down, front-rear, left-right, and a continuous value of vertical rotational angle. By playing back the recorded data according to the time stamp, the motion of the UUV to collect the target debris is generated.

3.6 Teaching playback with feedback

The target object may move during the scooping motion due to the waves and the water flow created by the UUV. Therefore an upward camera on the top of the UUV was installed. By using the position of the target on the upward camera image, the scooping trajectory is sequentially adjusted, and a robust collection is achieved. To collect the target with the collection mechanism, it is ideal to approach from directly below the target. Therefore, using the target position on the UUV upward camera frame ${}^{\mathrm{uc}}\boldsymbol{p}_{\mathrm{t}}$, the deviation $\boldsymbol{e}_{\mathrm{u}}$ is defined as follows:

$$\boldsymbol{e}_{\mathrm{u}} = \boldsymbol{A}({}^{\mathrm{uc}}\boldsymbol{p}_{\mathrm{t}} - \boldsymbol{c}) \,, \tag{30}$$

where $\boldsymbol{A}$ is the transformation matrix from the upward camera image frame to the UUV frame, and $\boldsymbol{c}$ is a center of the camera image. To approach from directly below the target, the scooping trajectory of the UUV is corrected using a PID controller. Hence, using the original recorded trajectory $\boldsymbol{u}(t)$, the corrected trajectory $\tilde{\boldsymbol{u}}(t)$ is calculated as follows:

$$\tilde{\boldsymbol{u}}(t) = \boldsymbol{u}(t) + \boldsymbol{K}_{\mathrm{up}}\boldsymbol{e}_{\mathrm{u}} + \boldsymbol{K}_{\mathrm{ud}}\frac{d}{dt}\boldsymbol{e}_{\mathrm{u}} + \boldsymbol{K}_{\mathrm{ui}}\int \boldsymbol{e}_{\mathrm{u}}dt \,, \tag{31}$$

where $\boldsymbol{K}_{\mathrm{up}}$, $\boldsymbol{K}_{\mathrm{ud}}$, and $\boldsymbol{K}_{\mathrm{ui}}$ are the gains for trajectory correction.

3.7 Experiment

To evaluate our semi-automatic floating debris collecting system, the following two experiments were conducted: experiment to evaluate collecting efficiency in a simulator, and the debris collecting test with real robots to evaluate the practicality of our system.

3.7.1 Implementation

Fig. 16 shows the hardware configuration of the proposed system. The *Blue Robotics BlueROV2* was used as a UUV. The collecting mechanism is a 42 cm

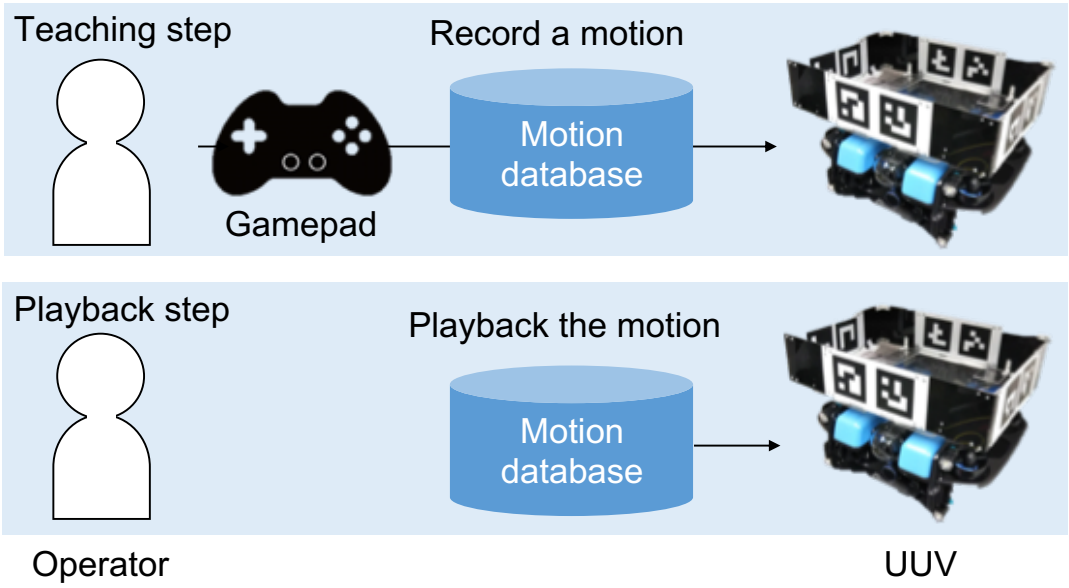


Figure 15. Overview of the motion generator using teaching-playback.

× 57 cm × 19 cm ($X \times Y \times Z$) cage-like mechanism with AR markers and was installed on top of the UUV. The upward-facing camera *BUFFALO BSW200MBK* on the UUV is connected to a *Raspberry Pi 4* and installed on the UUV with a *Blue Robotics Watertight Enclosure (3 Series)*. A micro quadrotor UAV *DJI Tello* was used. The camera was mounted on the front of the UAV at a downward angle of 45 degrees.

The UUV is controlled by Ubuntu 18.04.3 LTS PC *iiyama W350SS* equipped with an Intel(R) Core (TM) i7-4710MQ processor, and 7.6 GB-RAM. The UAV is controlled by Ubuntu 18.04.1 LTS PC *VAIO SVS13A2AJ* equipped with an Intel(R) Core (TM) i7-3520M processor, and 11.6 GB-RAM. The UAV and its control PC communicate through Wi-Fi. The UUV, its control PC and the *Raspberry Pi 4* for the upward-facing camera were connected through a *Fathom ROV Tether*, which acts as a network and traction cable. We used *Fathom-X Tether Interface Boards* to convert *Fathom ROV Tether* to an ethernet cable. A gamepad *dualshock 4*, which was used to teach the scooping motion was connected to the UUV control PC. All hardware except *Tello* is on the same network through a switching hub.

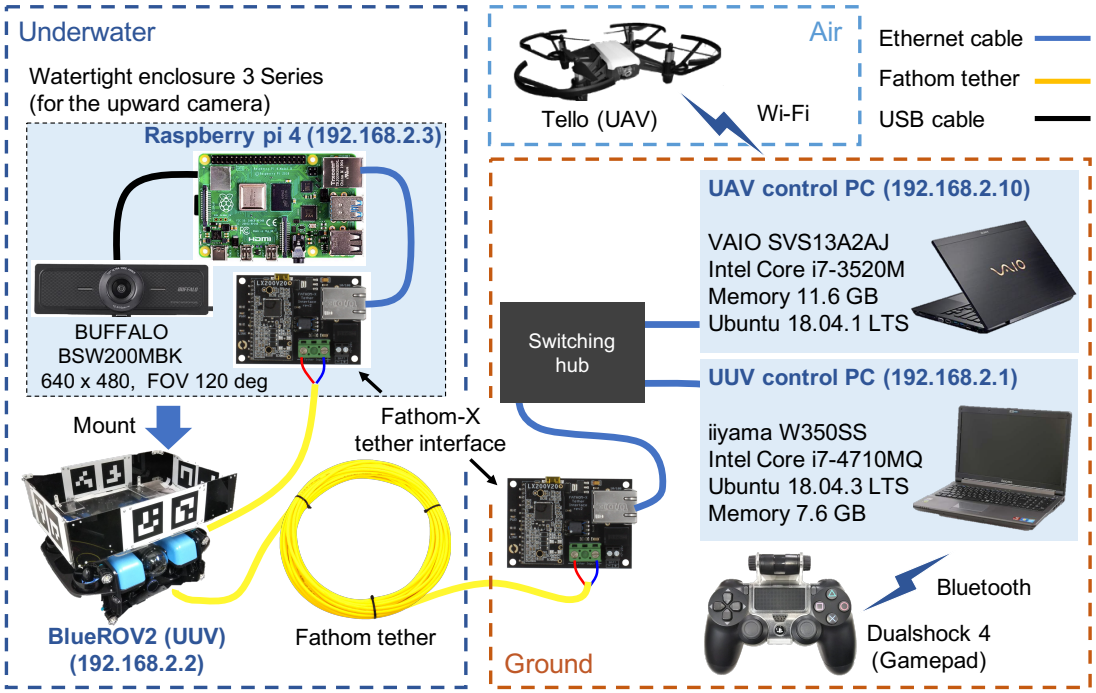


Figure 16. Hardware and software configuration of the proposed system.

Fig. 17 shows the overview of nodes running on PCs and data communication. The proposed system runs on a robot operating system (ROS). The *DJI* official software development kit *Tello-Python* was used to control *Tello* and acquire the camera image. *Mjpeg-streamer* ran on *Raspberry Pi 4* to share the image of the UUV's upward-facing camera through the network. The proposed GUI controller node ran on the UAV control PC and acquired latest frame of the camera image by multi-thread processing. The main thread of the GUI controller node calculated and sent the 3D position of the target to the UUV controller. The UUV controller received the 3D position of the target debris ${}^{r}\boldsymbol{p}_{\mathrm{t}}$ and had the motion database for the teaching playback, and sent control commands to the UUV through a wired network using a *MAVLink* protocol. The *MAVLink* interface for ROS, *MAVROS*, was used. To display the approaching target circle (red and green circles in Fig. 12), the UUV controller sent the control target position $\hat{\boldsymbol{x}}$ and the current state $\boldsymbol{x}$ to the GUI.

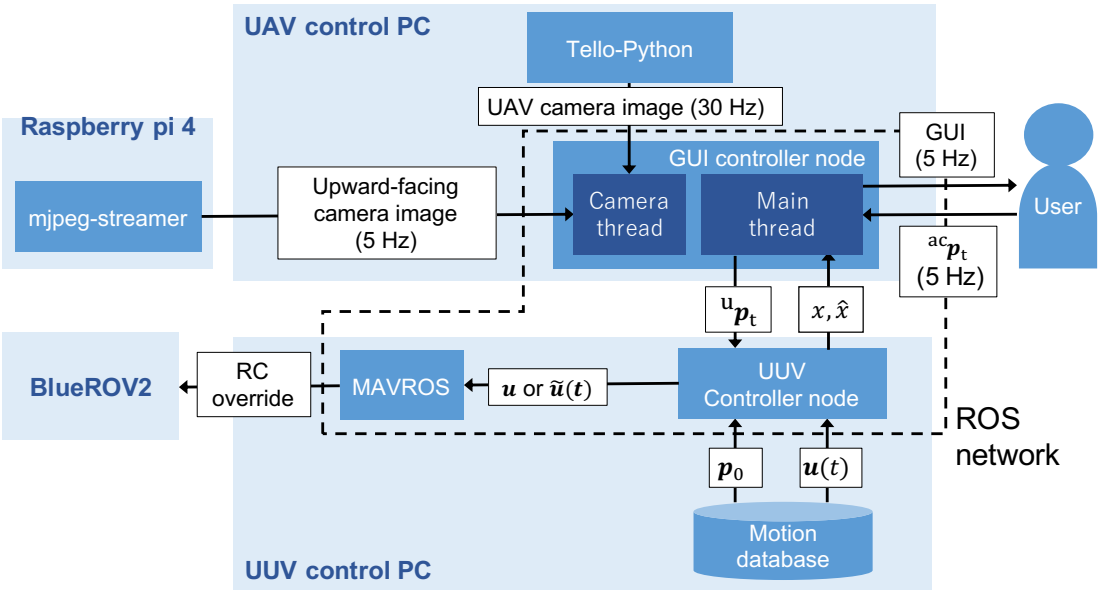


Figure 17. Software configuration of the proposed system.

3.7.2 Evaluation of the collecting efficiency, usability, and workload

To evaluate the following three effects, our system was tested in a simulation environment:

- To evaluate the improvement of collection efficiency with cooperation of UAV and UUV, total collection time and travel distance are compared.
- To confirm that easiness to use was realized with the proposed GUI, SUS is used as an index to evaluate usability.
- To evaluate the effect of labor reduction with the proposed semi-automation of the collection process, NASA-TLX is used as an index to evaluate the difference in workload.

Fig. 18 shows the environment of the simulation experiment. We used *Gazebo* as a dynamics simulator. To simulate the underwater environment, a plugin and an environment provided by *Blue Robotics* and *VRX Simulation* are used. In the simulation environment, five plastic bottles with a radius of 6 cm and a height of 26 cm are randomly located in a range of 10 m × 10 m and the

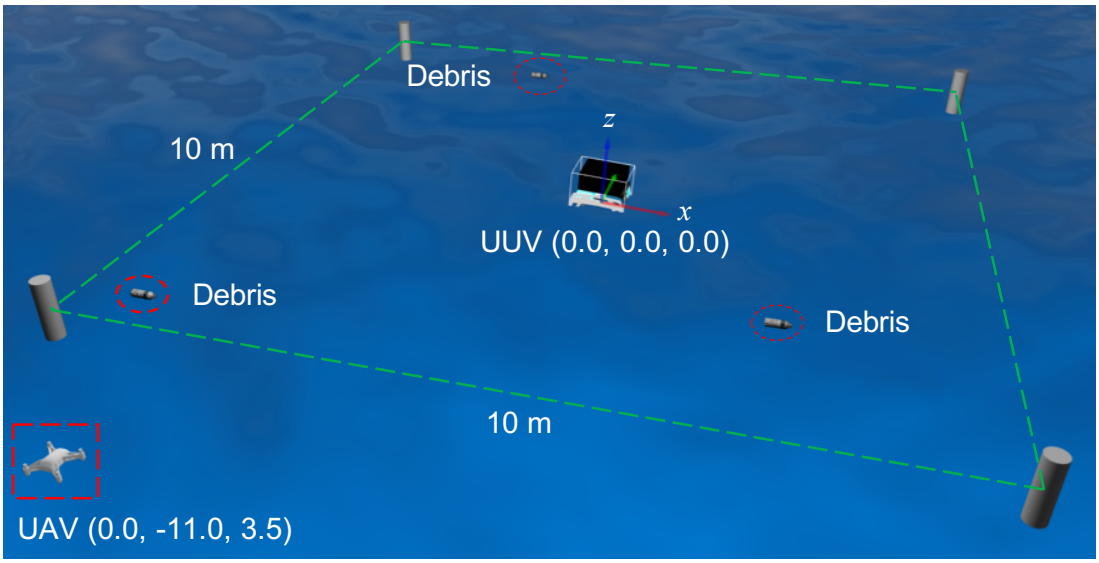


Figure 18. Environment of the simulation experiment.

UUV is located at the origin. The UAV was placed in the following location to overlook all range of 10 m $\times$ 10 m: an altitude of $+3.5$ m and -11.0 m offset from the UUV in the y-direction. Buoyancy and vertical sway due to waves have been simulated, but plastic bottles' movement due to water flow has not been simulated. This dissertation assumes that the collection is completed when debris enters the collection box. Therefore, in the simulation environment, the bottle disappears when it enters the box.

Ten Japanese subjects performed the simulation experiment. For comparison, each subject controlled the UUV in the following four conditions:

1) **UUV alone:** Subjects use the GUI shown in Fig. 12 whose screen of the left side is replaced with a forward camera image on the UUV. The operation and control procedure are the same as shown in Fig. 12.

2) **Gamepad:** Subjects control the UUV using a joystick of a gamepad shown in Fig. 19 to collect debris while looking at the GUI shown in Fig. 12.

3) **Proposed GUI:** Subjects collect the debris with the GUI shown in Fig. 12 (proposed method).

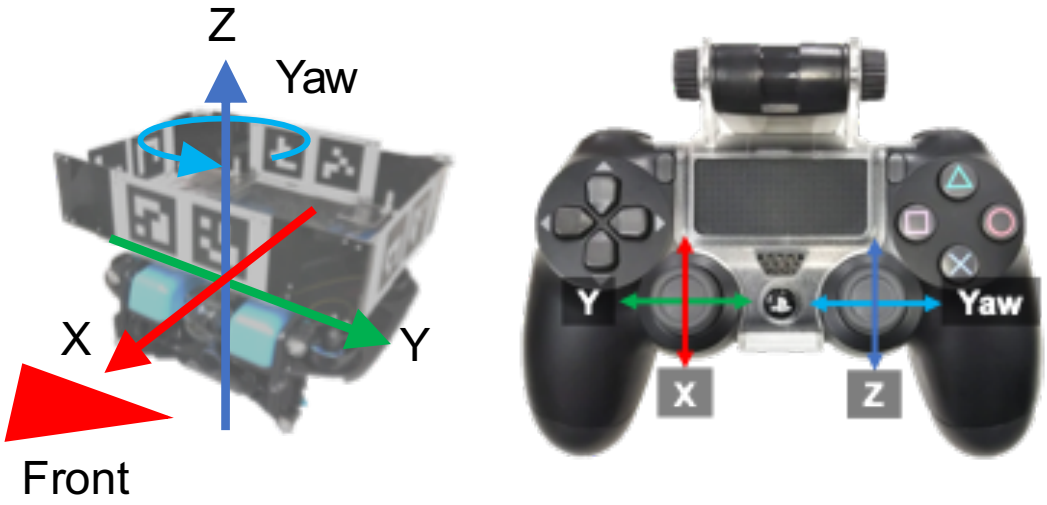


Figure 19. Configuration of the manual control operation.

4) **Homography GUI:** Subjects use the GUI shown in Fig. 12, whose screen on the left side is replaced with the image after homography transformation.

Comparing with condition 1 verifies the effect of cooperation with UAV, condition 2 verifies the usability of the proposed GUI and the effect of reducing the workload, and condition 4 corresponds to the previous study [28] which is used for rescue.

The flow of the experiment for each subject is as follows: First, we told the subject the minimum information to complete the experiment, such as the usage of the control GUI and experimental procedures. After that, the subject try to collect only one plastic bottle under the four conditions as a practice. After the practice, the subjects collect five plastic bottles in order from conditions 1 to 4 and record the time, route, and traveling distance from the start of the operation to complete the collection. The subjects answered the NASA-TLX and SUS questionnaire after the experiment was completed under all conditions in order that the contents of the questionnaire do not affect the subject's intentions. The Japanese version of NASA-TLX [41] was used according to the subjects' native language. The original English text and the Japanese translation were written together in the SUS questionnaire. In the SUS questionnaire, we also accepted

Table 2. Result of the user experiment.

Subjecs	Condition							
	1. UUV alone		**2. Gamepad**		**3. Proposed GUI**		**4. Homographic GUI**	
	T/[s]	D/[m]	T/[s]	D/[m]	T/[s]	D/[m]	T/[s]	D/[m]
1	207.10	49.50	99.99	34.11	99.20	38.69	103.20	43.11
2	176.79	46.69	93.80	42.05	92.60	37.10	77.80	32.73
3	204.50	51.24	152.69	62.32	117.30	48.00	88.70	38.22
4	181.20	47.30	92.70	42.49	123.20	47.62	91.10	38.07
5	164.70	43.03	72.80	31.96	124.20	50.49	119.50	47.33
6	236.91	51.74	223.90	55.36	115.10	43.62	140.30	47.47
7	188.00	45.88	82.50	36.19	82.90	37.21	100.60	42.51
8	150.60	41.70	87.41	39.24	103.59	40.13	93.20	36.72
9	145.48	38.82	97.39	46.76	94.60	40.43	99.20	41.90
10	181.38	45.51	77.80	37.20	107.40	41.24	90.30	36.66
Average	183.67	46.14	108.10	42.77	106.01	42.59	100.39	40.47
SD	26.09	3.92	43.89	9.14	13.17	4.47	16.89	4.57

free-form answers for each conditions.

To make the conditions consistent, we asked the subjects to adhere to the following rules:

- Subjects must select the plastic bottle that seems to be the closest as a target.
- Collection must be done as quickly and accurately as possible.
- If the collection fails, the subjects try again until they succeed.
- In condition 1, if no plastic bottle was visible, the subjects rotate the UUV counterclockwise using the gamepad. If even one plastic bottle is found, the subjects control the UUV with the GUI.
- In conditions using the GUI, i.e., conditions 1, 3 and 4, the subjects collect debris in the steps 1-3 in Fig. 12.

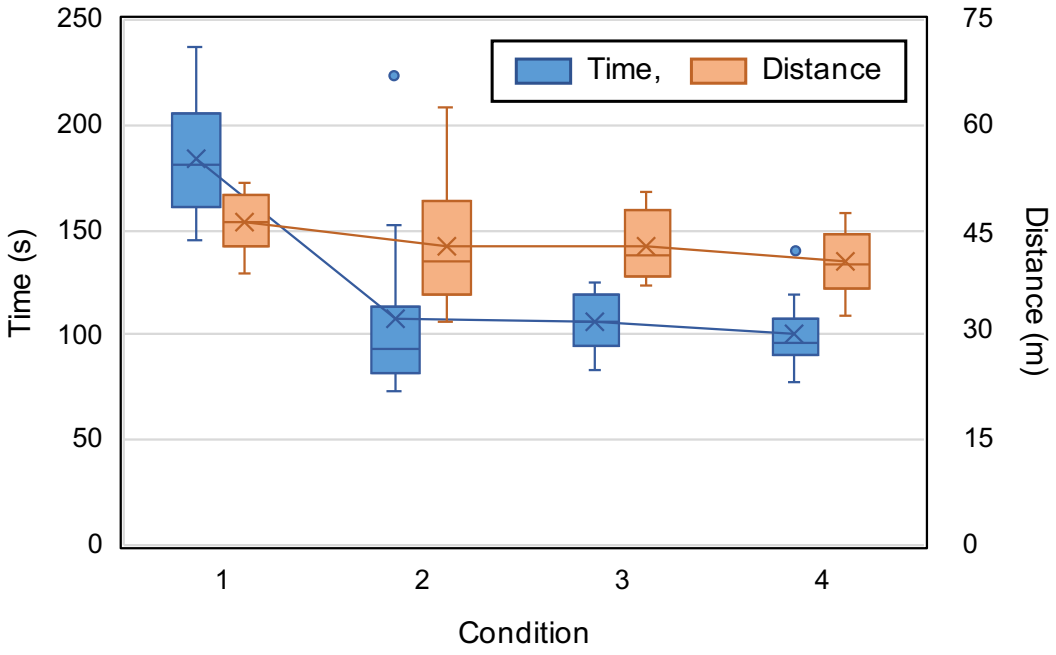
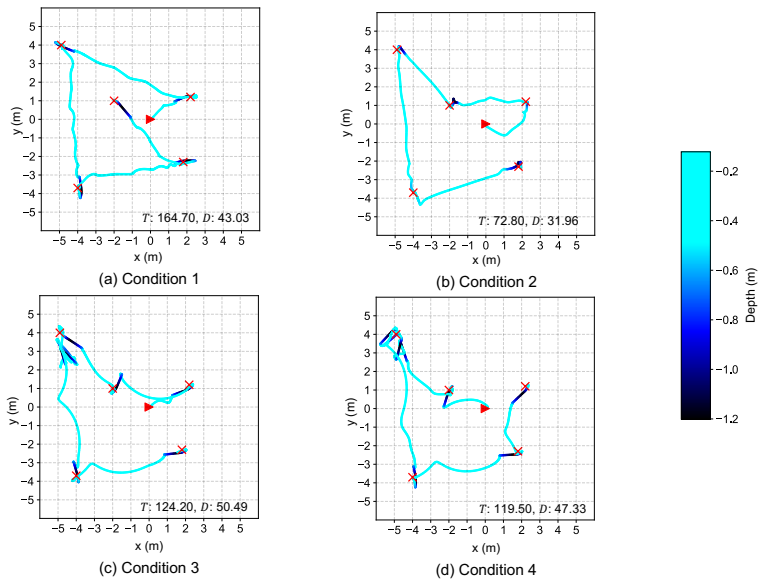


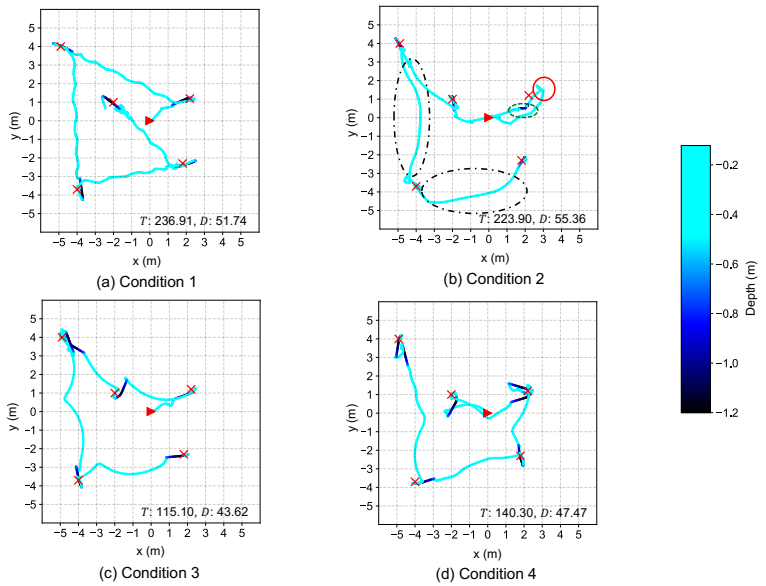
Figure 20. Comparison of collection time and traveling distance in the simulation experiment.

3.7.3 Results of collection time and traveling distance

Table 2 shows the total collection time T and traveling distance D. When all subjects used condition 1, which uses only the UUV viewpoint, T was the largest, but there was almost no difference in the travel distance D. Regarding other conditions, condition 3 (proposal method) and condition 4 showed the same tendency but the average collection time and average total travel distance was slightly shorter in condition 4 shown as Fig. 20. This may be because the homography transformation made the positional relationship clearer. The clarified positional relationship of the UUV and debris may have made it easier to select the shortest path and adjust the position. Besides, regarding manual operation condition 2, the difference between the subjects was large, and T's variance was the largest among the four conditions. As for each condition's average, condition 1 took about 1.7 times longer than the others, but there was almost no difference for other conditions.



(a) Subject 5



(b) Subject 6

Figure 21. Results of trajectories of the UUV executed in simulation experiments. The five red crosses of each graph indicate location of five pieces of debris. The red triangles show the initial pose of the UUV. The line color indicates depth of the UUV. Results of subjects 5 and 6 are shown as examples of the subjects who have different tendencies for condition 2.

Fig. 21 shows the trajectories of the UUV for subjects 5 and 6, which have different tendencies for condition 2, as an example. Since the field of view of condition 1 is limited, the order of collection was different from other conditions. Even though there is no difference in D, there is a large difference in T because the UUV has to rotate on the spot after collecting one target until the next target is found in condition 1. With other conditions, by collaborating with the UAV, since all plastic bottles can be observed at all times, the UUV can immediately move on to the next target after collecting one target.

For condition 2, the order of collecting targets and D are almost the same as other conditions, but T is about three times. In condition 2 in Fig. 21 (b), the solid red circle indicates that the UUV had passed once collecting the first target. Also, the green dotted circle indicates that the collection had failed, and the scooping motion was repeated. The black dash-dotted circle indicates that the UUV could not head straight toward the next target. These facts indicate that the operation proficiency level for the gamepad affects the result of condition 2.

As a result of the comparison of T and D, it was confirmed that the time required for collecting multiple floating debris could be shortened, and the collection efficiency can be improved by cooperating with the UAV. Also, in the case of manual operation using a gamepad, it was confirmed that the collection time varies greatly among individuals, and some subjects can collect debris more efficiently.

3.7.4 Result of the NASA-TLX questionnaire

Fig. 22 shows the results of the NASA-TLX questionnaire. The TLX score expresses load as a value from 0 to 100, and the larger the value is, the higher workload is. As a result, the proposed method has a significant difference ($p < .05$) from condition 2. From the comparison of conditions 2 and 3 in Fig. 22, the workload of condition 3 was lower than condition 2. Furthermore, even in the case of subjects 5 and 10 who showed better results with condition 2 than condition 3 about the collection time in Table 2, it was confirmed that the workload was reduced in condition 3. Regarding condition 1, it cannot be said that there is a significant difference, but it has a value close to the significance level ($p = .052$). Therefore, using the viewpoint from the UAV may have the effect of reducing

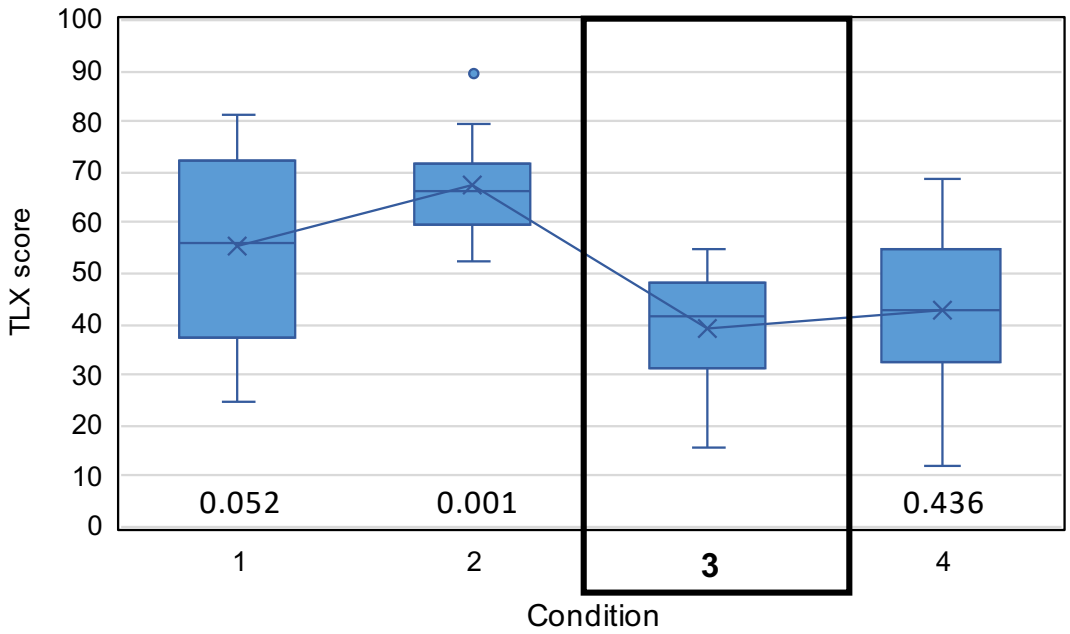


Figure 22. TLX score of each condition. Outliers are eliminated by keeping the minimum and maximum values within 1.5 times the box length. The number in the graph shows the p-value of the t-test between the each conditions and the proposed methods (condition 3). Significance level is 0.05.

the workload. Regarding condition 4, there was no significant difference, but the mean and variance were smaller in the proposed method. In addition, the TLX score for each element in Fig. 23 indicates that condition 3 has a lower Mental Demand (MD) than condition 4. Since MD represents how much mental and perceptual thinking is required for the task, the difference in MD shows that our GUI is intuitive and does not require extra thinking to use.

3.7.5 Result of the SUS questionnaire

Fig. 24 shows the results of the SUS questionnaire. The SUS score expresses usability as a value from 0 to 40, and the larger the value is, the higher the usability is. As a result, it was confirmed that condition 3 (proposed method) is significantly different from other conditions. Also, Fig. 24 shows that the proposed method has the highest SUS score. The cause of the difference can

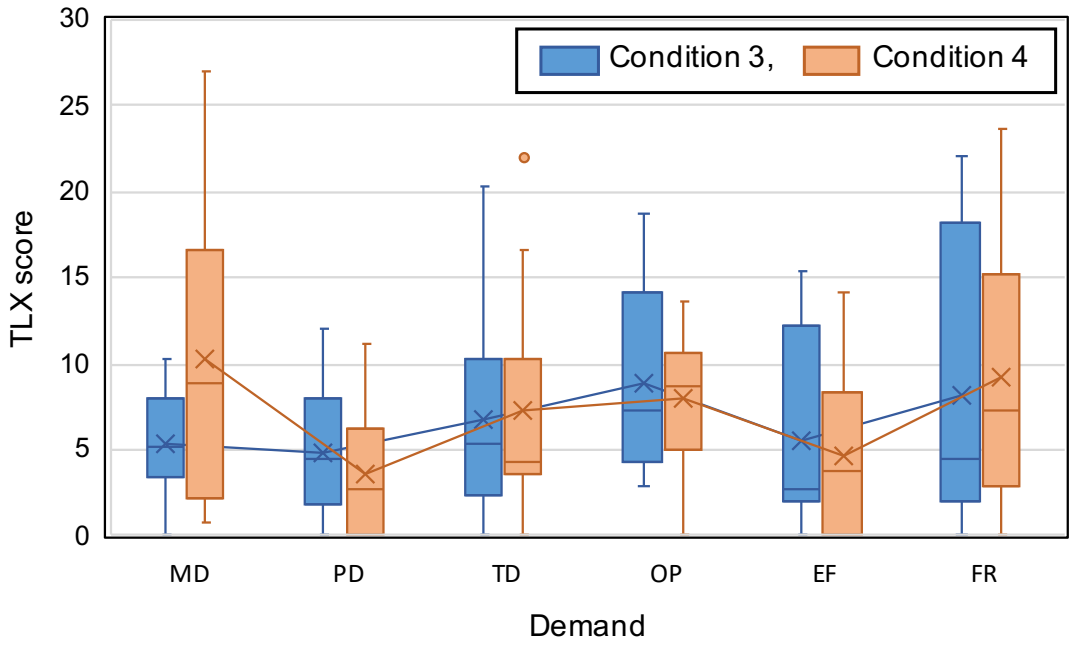


Figure 23. TLX score comparison of conditions 3 and 4 for each demand.

be considered as follows from the contents of the SUS questionnaire's free-form answers:

- Condition 1. The field of view is narrow, and it takes time to find the target. It is not easy to track the target because the viewpoint moves with the UUV.
- Condition 2. If the user gets used to the operation, it can be collected efficiently, but it is not easy at first. The operation becomes confusing when the viewpoint changes.
- Condition 4. The position relationship of plastic bottles is easy to grasp, but it feels unnatural. The original image is easier to use.

Regarding conditions 1 and 2, there were significant differences in efficiency and load from the proposed method, so they may also affects usability. Regarding condition 4, there was no significant difference in efficiency or load, but a significant difference was confirmed in usability. This may be because the unnaturalness of the deformed image affect usability. In the proposed GUI, since the

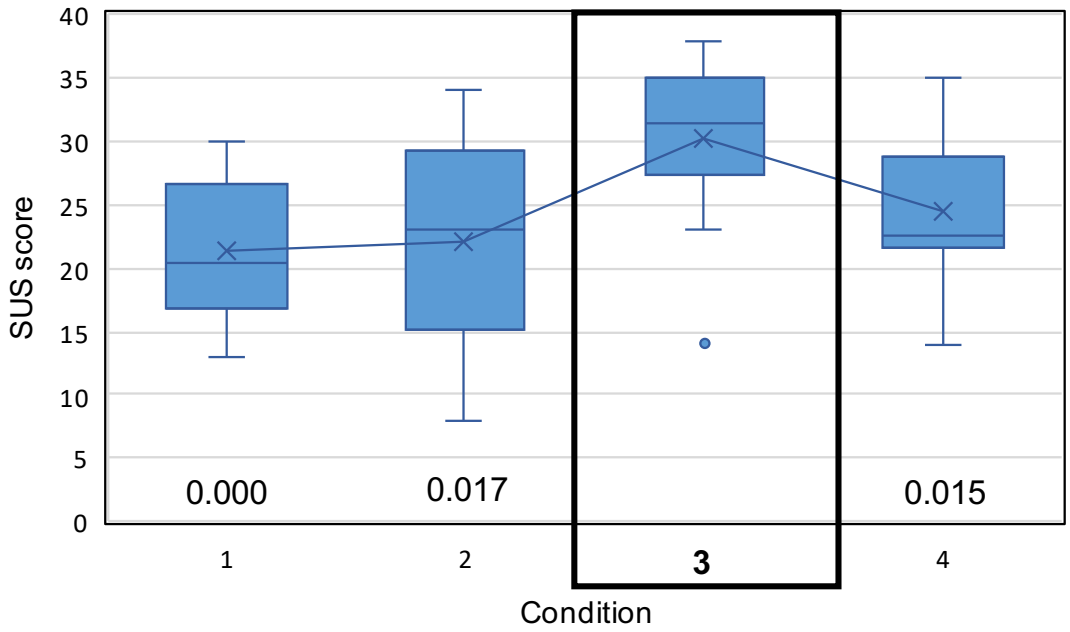


Figure 24. SUS score of each condition. Outliers are eliminated by keeping the minimum and maximum values within 1.5 times the box length. The number in the graph shows the p-value of the t-test between the each conditions and the proposed methods (condition 3). Significance level is 0.05.

original natural image is given to the user as a viewpoint, the usability does not decrease.

3.7.6 Debris collecting test with real robots

To verify the practicality of our system, we tried collection using real vehicles. An experimental pool with a width of 4.0 m, a length of 2.5 m, and a depth of 1.5 m was used as an experimental environment. In the pool, one plastic bottle with a radius of 6 cm and a height of 26 cm was placed. We tried to collect the plastic bottle 17 times using the proposed system. In each trial, targeting using the GUI, approaching the target, and generating the scooping motion are executed, and the number of successful collections is recorded. The initial positions of the UUV and the plastic bottle were not too close to each other, and the plastic bottle was located away from the wall of the pool to prevent the UUV from colliding the

wall. Regardless of the collection result, the position of the UUV and the plastic bottle was reset for each trial.

Table 3 shows the success rate of the debris collection experiment using real robots. The results show that the collection of one piece of debris was successful 15 times of 17 trials, about 88.23%.

Fig. 25 and Fig. 26 show the motions of the UUV when the collection succeeded and failed. In the case of a successful collection, the UUV approaches the target (within 0 to 6 s), the UUV is sufficiently close to the target and the operator triggers the execution of the scooping motion (at 8 s), and completes the scooping motion and succeeded to collect the target (at 14 s). In the case of a failed collection, the UUV does not come close enough to the target, and the target cannot be tracked using the upward camera on the UUV. Therefore, if the target can be captured with the upward camera, the collection will almost certainly succeed.

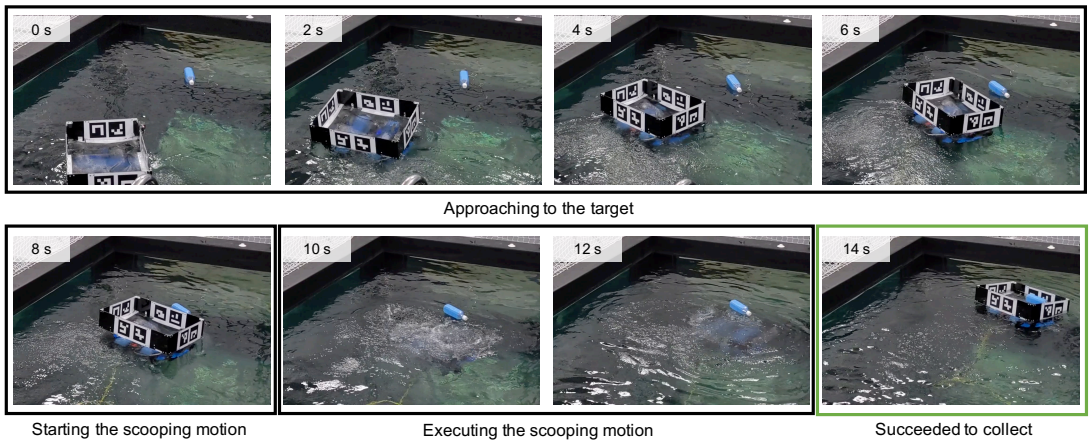


Figure 25. Motion of the UUV in the success case of the experiment using the real robot.

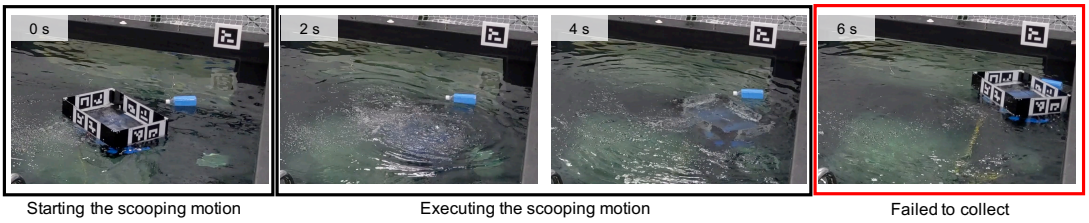


Figure 26. Motion of the UUV in the failure case of the experiment using the real robot.

Table 3. Result of the experiment using real robots.

Success	**Failure**	**Total**	**Success rate**
15	2	17	$15/17 \approx 88.23\%$

4. Conclusion and Future Work

4.1 Conclusion

This dissertation discussed the simplification of motion generation of six DoF mobile robot for manipulation tasks. As a six DoF mobile robot, we focused on UAVs and UUVs. We considered the following themes: 1) simplifying motion generation when a manipulation task with a UAV alone, 2) simplifying motion generation, including mobile robot sensing, navigation problems, and evaluating the effectiveness of the simplification.

The push operation by UAV was performed as an example of the manipulation task with a UAV, including contact with the environment that the existing controller of UAVs cannot cover. A physical model for pushing motions using a UAV was constructed. The system reuses the existing attitude controller for the pushing motion. Simulation experiments using the proposed method showed that the pushing motion was possible with simple command such as target distance even if the mass of the target object is different.

We considered a manipulation task of collecting floating debris on the water surface as a simple application of a manipulation task with UUVs. We developed a system that can collect the floating object by just selecting the target on the screen of the camera looking down from the UAV. We designed coordinate transformation method and a GUI to estimate the 3D position of floating debris on the water surface using images from a monocular camera mounted on a UAV. To make the collection robust, we developed a controller that generates a robust collection trajectory using teaching playback with feedback.

To evaluate the efficacy of the collection system, we conducted both simulation and real-robot experiments. In the simulation experiments, 10 subjects collected multiple pieces of floating debris under different conditions. As a result, it was confirmed that UAV-UUV cooperation enables more efficient collection than UUV alone. Furthermore, from the results of the NASA-TLX and SUS questionnaires, it was confirmed that our GUI reduced the workload and realized the ease of use. In addition, from the comparison between the homographic-converted image [28] and the original image, it was confirmed that the usability improved when the original image is provided to the user. By contrast, it was indicated that the

homographic-converted image may have assisted subjects in fully comprehending the situation, such as the positional relationship between debris and UUV, and slightly improved the collection efficiency. We also tested the proposed system using real robots in an experimental pool, and we confirmed the practicality of a semi-automatic collection of floating debris with a successful collection rate of 88.23%.

From each result, we achieved simplification of motion generation of a multi-degree-of-freedom mobile robot in a manipulation task, construction of a system that can be used for a specific application, and confirmation of the effect of simplification of motion generation by subjective evaluation. By combining each base idea, it is possible to generate a UAV/UUV motion during an object manipulation task using simple control command at the application level.

4.2 Future work

In future work, we are planning to use the actual UAV to perform the push operation. To apply in real-world experiments, feedback control and an end-effector such as a passive damping mechanism are required due to sensing noise and unknown parameters. Furthermore, we described a method of pushing unliftable objects in only one direction, but we would like to consider pushing in various directions as a future work. In the case of pushing motions in various directions, it is necessary to consider not only simply extending the proposed method in two directions but also the rotation of the target objects.

About manipulation with the UUV, we plan to improve the successful collection rate by modifying the feedback controller. Our system currently uses simple PID control, but when the deviation is small, the actuator's dead zone may make it difficult to converge to the target. It may cause collection failure or usability reduction, so the method that considers how to handle the actuator's dead zone is required [42]. Furthermore, we will try to collect marine debris in an actual environment instead of the experimental pool using our system. Currently, we are thinking of situations that are remotely controlled from a ship, so an onboard computer does not require much computing power. When our system is fully automated by combining the automatic object detection such as Lili's method [38], all processing can be executable onboard by installing a computer suitable for

deep learning, such as an embedded AI computing device *NVIDIA Jetson*.

Acknowledgement

First, I wish to thank my thesis supervisor, Professor Takahiro Wada. He accepted and took care of me, despite my urgent request. I would like to thank him for the opportunity to graduate he gave me.

Second, I am thankful to Professor Kenji Sugimoto for reviewing this dissertation and advice. He has given me helpful advice from a new perspective since I was a master's student.

Particularly, Visiting Professor Jun Takamatsu gave me a great deal of instruction for my study. He always gave me sound advice on writing a thesis, research policies, and technical issues. I am really appreciated his great support.

I am also thankful for Visiting Associate Professor Gustavo Alfonso Garcia Ricardez. He gave me a lot of advice on correcting English and research.

Assistant Professor Sung-Gwi Cho has given me much advice on college life and research since he was a student. I would like to thank him for supporting me.

Furthermore, I would like to all members of the Robotics laboratory for their kindness and supporting me.

Additionally, this work would not be possible without the support of my family.

Finally, I am grateful to Executive Director of the Nara Institute of science and technology, Tsukasa Ogasawara who was my supervisor. He always gave me his best support and much advice to my research until he was retired from his professor post. Unfortunately, I could not complete this dissertation during his tenure, but I would like to acknowledge his support.

References

[1] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. I–I, Ieee, 2004.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, mar 2018.

[5] M. Orsag, C. Korpela, P. Oh, and S. Bogdan, *Aerial Manipulation*. Springer, 2018.

[6] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[7] S. Sivčev, J. Coleman, E. Omerdić, G. Dooly, and D. Toal, "Underwater manipulators: A review," *Ocean Engineering*, vol. 163, pp. 431–450, 2018.

[8] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, "A survey of open-source uav flight controllers and flight simulators," *Microprocessors and Microsystems*, vol. 61, pp. 11–20, 2018.

[9] M. Car, A. Ivanovic, M. Orsag, and S. Bogdan, "Impedance based force control for aerial robot peg-in-hole insertion tasks," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 6734–6739, 2018.

[10] M. Zhao, K. Kawasaki, T. Anzai, X. Chen, S. Noda, F. Shi, K. Okada, and M. Inaba, "Transformable multirotor with two-dimensional multilinks:

Modeling, control, and whole-body aerial manipulation," *The International Journal of Robotics Research*, pp. 1085–1112, August 2018.

[11] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, "6d physical interaction with a fully actuated aerial robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 5190–5195, 2017.

[12] G. Jiang and R. Voyles, "Hexrotor uav platform enabling dextrous aerial mobile manipulation," in *Proc. of the Int. Micro Air Vehicle Conf. and Competitions*, pp. 21–26, 2013.

[13] C. D. Bellicoso, L. R. Buonocore, V. Lippiello, and B. Siciliano, "Design, modeling and control of a 5-dof lightweight robot arm for aerial manipulation," in *Proc. of the Mediterranean Conf. on Control and Automation*, pp. 853–858, 2015.

[14] M. J. Kim, R. Balachandran, M. D. Stefano, K. Kondak, and C. Ott, "Passive compliance control of aerial manipulators," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4177–4184, 2018.

[15] N. Staub, M. Mohammadi, D. Bicego, Q. Delamare, H. Yang, D. Prattichizzo, P. R. Giordano, D. Lee, and A. Franchi, "The tele-magmas," *IEEE Robotics and Automation Magazine*, vol. 25, pp. 66–75, December 2018.

[16] H. Kim, H. Seo, C. Y. Son, H. Lee, S. Kim, and H. J. Kim, "Cooperation in the air," *IEEE Robotics and Automation Magazine*, vol. 25, pp. 76–85, December 2018.

[17] H. W. Wopereis, W. L. van de Ridder, T. J. Lankhorst, L. Klooster, E. M. Bukai, D. Wuthier, G. Nikolakopoulos, S. Stramigioli, J. B. Engelen, and M. Fumagalli, "Mulimodal aerial locomotion," *IEEE Robotics and Automation Magazine*, vol. 25, pp. 57–65, December 2018.

[18] X. Meng, Y. He, Q. Li, F. Gu, L. Yang, T. Yan, and J. Han, "Contact force control of an aerial manipulator in pressing an emergency switch process," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2107–2113, 2018.

[19] G. Gioioso, M. Mohammadi, A. Franchi, and D. Prattichizzo, "A force-based bilateral teleoperation framework for aerial robots in contact with the environment," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 318–324, 2015.

[20] K. Yoo and W. Chung, "Pushing motion control of n passive off-hooked trailers by a car-like mobile robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4928–4933, 2010.

[21] T. Meriçli, M. Veloso, and H. L. Akın, "Push-manipulation of complex passive mobile objects using experimentally acquired motion models," *Autonomous Robots*, vol. 38, no. 3, pp. 317–329, 2015.

[22] Y. Hakamata, S. Tsuichihara, J. Takamatsu, and T. Ogasawara, "Whole body motion generation of humanoid robot using a predicted reaction force," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots*, pp. 95–100, 2016.

[23] S. Hood, K. Benson, P. Hamod, D. Madison, J. M. O'Kane, and I. Rekleitis, "Bird's eye view: Cooperative exploration by UGV and UAV," in *Proc. of IEEE Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 247–255, 2017.

[24] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Trans. on Vehicular Technology*, vol. 65, no. 12, pp. 9585–9596, 2016.

[25] G. Christie, A. Shoemaker, K. Kochersberger, P. Tokekar, L. McLean, and A. Leonessa, "Radiation search operations using scene understanding with autonomous UAV and UGV," *J. of Field Robotics*, vol. 34, no. 8, pp. 1450–1468, 2017.

[26] T. Miki, P. Khrapchenkov, and K. Hori, "UAV/UGV autonomous cooperation: UAV assists UGV to climb a cliff by attaching a tether," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 8041–8047, 2019.

[27] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Cooperative forest monitoring and fire detection using a team of UAVs-UGVs," in *Proc. of Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 1206–1211, 2016.

[28] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 6105–6110, 2017.

[29] K. Lima, E. R. B. Marques, J. Pinto, and J. B. Sousa, "Dolphin: a task orchestration language for autonomous vehicle networks," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 603–610, 2018.

[30] S. D. Lucia, G. D. Tipaldi, and W. Burgard, "Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping aproach," in *Proc. of the European Conf. on Mobile Robots*, pp. 1–6, 2015.

[31] G. Garimella and M. Kobilarov, "Towards model-predictive control for aerial pick-and-place," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 4692–4697, 2015.

[32] L. Lin, Y. Yang, H. Cheng, and X. Chen, "Autonomous vision-based aerial grasping for rotorcraft unmanned aerial vehicles," *Sensors*, vol. 19, no. 15, p. 3410, 2019.

[33] C. Papachristos, K. Alexis, and A. Tzes, "Efficient force exertion for aerial robotic manipulation: Exploiting the thrust-vectoring authority of a tri-tiltrotor uav," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 4500–4505, 2014.

[34] M. A. Johnson and M. H. Moradi, *PID control*. Springer, 2005.

[35] "The ocean cleanup." [online]. Available: `https://theoceancleanup.com/`. [Accessed: 18- Sep- 2019].

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

[37] S. Kawamura, "Underwater robot development for manipulation task and their uses in biwa lake," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 14–19, 2015.

[38] L. Zhang, Y. Zhang, Z. Zhang, J. Shen, and H. Wang, "Real-time water surface object detection based on improved Faster R-CNN," *Sensors*, vol. 19, no. 16, p. 3523, 2019.

[39] J. Brooke, "SUS: a " quick and dirty'usability," *Usability evaluation in industry*, p. 189, 1996.

[40] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," vol. 52, pp. 139–183, 1988.

[41] S. Haga and N. Mizukami, "Japanese version of NASA task load index," *The Japanese journal of ergonomics*, vol. 32, no. 2, pp. 71–79, 1996.

[42] T. Yang, N. Sun, and Y. Fang, "Adaptive fuzzy control for a class of mimo underactuated systems with plant uncertainties and actuator deadzones: Design and experiments," *IEEE Transactions on Cybernetics*, 2021.