

A Bimanual Teleoperation Framework for Light Duty Underwater Vehicle-Manipulator Systems

Justin Sitler

Srikarran Sowrirajan

Brendan Englot

Long Wang

Abstract—In an effort to lower the barrier to entry in underwater manipulation, this paper presents an open-source, user-friendly framework for bimanual teleoperation of a light-duty underwater vehicle-manipulator system (UVMS). This framework allows for the control of the vehicle along with two manipulators and their end-effectors using two low-cost haptic devices.

The UVMS kinematics are derived in order to create an independent resolved motion rate controller for each manipulator, which optimally controls the joint positions to achieve a desired end-effector pose. This desired pose is computed in real-time using a teleoperation controller developed to process the dual haptic device input from the user. A physics-based simulation environment is used to implement this framework for two example tasks as well as provide data for error analysis of user commands*. The first task illustrates the functionality of the framework through motion control of the vehicle and manipulators using only the haptic devices. The second task is to grasp an object using both manipulators simultaneously, demonstrating precision and coordination using the framework. The framework code is available at https://github.com/stevens-armlab/uvms_bimanual_sim.

I. INTRODUCTION

Underwater robotics has many useful applications such as exploration [1], inspection [2], and aquaculture [3]. However, this research has typically involved large and expensive robots, such as the Girona 500 I-AUV [4], [5], SAUVIM [6], [7], or REMUS [8], [9]. The existing hardware is a huge barrier to entry for research in this field, and as such there has been some recent work developing light-duty platforms such as the Blue Robotics BlueROV2 [10], which has been used for research in works by Marais et. al. [11], [12] as well as McConnell et. al. [13–15]. Underwater robots are typically free-floating vehicles either teleoperated by a pilot [16–18] and referred to as remotely operated vehicles (ROVs), or autonomously controlled [19], [20] and referred to as autonomous underwater vehicles (AUVs). When equipped with a manipulator to perform intervention tasks, these robots are sometimes referred to as underwater vehicle-manipulator systems (UVMS).

The control of a UVMS is complicated due to the need for separate vehicle and manipulator control. A work package for teleoperation via wireless acoustic subsea communication is presented in [21], which presents a graphical user interface (GUI) with virtual and camera feedback to visualize the deployed system. The user can control the UVMS with a



Fig. 1: User workspace for bimanual teleoperation framework.

master device or use a mouse to send commands via the GUI. The system employs synthetic fixtures to assist the user through force and visual cues [22]. A user-friendly GUI was developed for semi-autonomous control of a BlueROV2 for floating debris collection in [23], in which vehicle commands are generated via interaction with the GUI.

Teleoperation of manipulator systems enables an operator to communicate specific maneuvers to a follower robot for execution. Telerobotics is popular in applications such as robotic surgery, where skilled surgeons conduct minimally-invasive surgery through visual feedback as presented in [24]. The 3D Systems Touch, formerly Phantom Omni, is a haptic device that captures pose information from the operator for teleoperation [25]. Bimanual teleoperation using haptic devices enables surgeons to perform base skull surgery, where haptic feedback offers an added dimension of realism to the surgeons, as presented in [26].

To lower the barrier to entry into underwater robotics we propose a simple, user-friendly framework for bimanual teleoperation of a light duty UVMS, an overview of which can be seen in Fig. 1. This paper derives the kinematic model for UVMS telemanipulation and implements a resolved motion rate controller. The mapping between the leader haptic device frames and the follower arm frame is also derived to

All authors are with the Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ, 07030, USA. Corresponding author: lwang4@stevens.edu. This research was supported in part by USDA-NIFA Grant No. 2021-67022-35977 and NSF Grant CMMI-2138896.

*<https://youtu.be/5pHZ5DaIIY4>

generate desired manipulator poses from the operator. This is implemented using two 3D Systems Touch haptic devices, whose low-cost and compact size make them easy to access and transport. In addition, the framework is implemented in a physics simulation environment to validate its performance in two different tasks. The first task is manual piloting of the vehicle, manipulators, and end-effectors to a desired pre-grasp pose. The second task involves grasping a small object using both manipulators simultaneously.

Our framework is unique from prior examples in that it uses two low-cost haptic devices for bimanual teleoperation instead of the expensive hardware and complex consoles used in the control of other bimanual UVMSs such as the Ocean One Avatar [17] or the Aquanaut [19]. The framework is also developed using open source tools such as Gazebo [27], the UUV Simulator [28], the Collaborative Robotics Toolkit (CRTK) [29], and our own open source code [30], so any researcher can easily adapt the framework to their own needs. In addition, we demonstrate a novel vehicle control method using the haptic devices to send vehicle commands, allowing for a single user to quickly and easily switch between vehicle, manipulator, and end-effector control.

The outline of this paper is as follows. Section II derives the kinematics for a generalized UVMS and the resolved motion rate controller which generates real-time manipulator joint commands from the desired end-effector pose. Section III introduces the mapping between leader-follower devices, from which end-effector commands are generated. Section IV discusses the simulation environment and UVMS models on which the bimanual control is implemented. Section V presents two example tasks executed in simulation using the bimanual controller. Finally, Section VI reflects on future improvements that could build on the work presented in this paper.

II. UVMS KINEMATICS

A. Forward Kinematics

A kinematic model of the UVMS manipulators is a prerequisite for teleoperational control. In this section, the direct kinematics and Jacobian matrix are derived in general for a serial manipulator of 6 degrees of freedom (DoF), denoted by the subscript $i = L, R$ representing the left and right manipulators respectively. Figure 2 shows the coordinate frames of a forward-facing configuration for two serial manipulators on the BlueROV2 Heavy platform.

For convenience, the kinematics of the UVMS manipulators are defined relative to the vehicle frame $\{ROV\}$ except where otherwise noted. The end-effector pose of manipulator i is defined as:

$$\mathbf{x}_i \triangleq (\mathbf{p}_i, \mathbf{R}_i), \quad (1)$$

where $\mathbf{p}_i$ is the position of the end-effector and $\mathbf{R}_i$ is the 3×3 rotation matrix representing the orientation of the end-effector. The end-effector pose can also be represented as the 4×4 transformation matrix:

$$\left({}^{ROV}\mathbf{T}_{ee}\right)_i \triangleq \begin{bmatrix} \mathbf{R}_i & \mathbf{p}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (2)$$

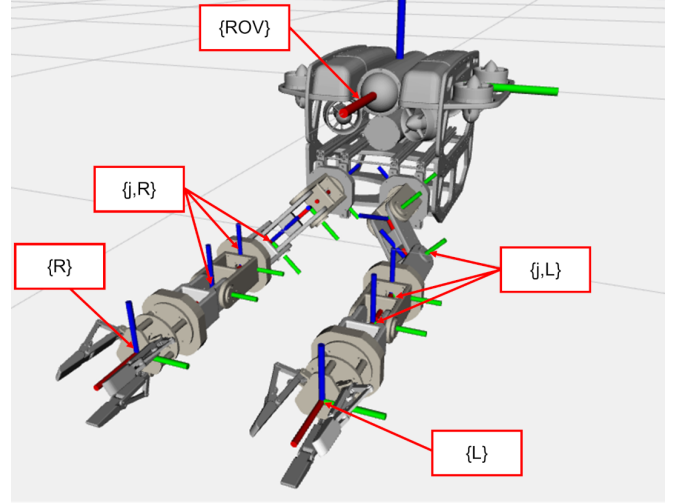


Fig. 2: Coordinate frames used in the kinematic model. Frame indices $\{j, i\}$ begin with $j = 1$ at the base of the manipulator and $j = 6$ equivalent to the end-effector frames $\{L\}$ and $\{R\}$.

The joint variables of each manipulator are independent from each other and are defined as:

$$\mathbf{q}_i \triangleq [q_{1,i} \quad q_{2,i} \quad \cdots \quad q_{6,i}]^T. \quad (3)$$

The position and orientation of the end-effector can be found sequentially:

$$\left({}^{ROV}\mathbf{T}_{ee}\right)_i = {}^{ROV}\mathbf{T}_{1,i} {}^{1,i}\mathbf{T}_{2,i} {}^{2,i}\mathbf{T}_{3,i} {}^{3,i}\mathbf{T}_{4,i} {}^{4,i}\mathbf{T}_{5,i} {}^{5,i}\mathbf{T}_{6,i}, \quad (4)$$

where ${}^{j-1,i}\mathbf{T}_{j,i}$ represents the relative transformation between frames $\{j-1, i\}$ and $\{j, i\}$ as defined in Fig. 2. The Jacobian for manipulator i relates the end-effector twist ξ_i to the joint velocities $\dot{\mathbf{q}}_i$ and is derived geometrically:

$$\xi_i \triangleq \mathbf{J}_i \dot{\mathbf{q}}_i, \quad \mathbf{J}_i = \begin{bmatrix} \mathbf{J}_{p,i} \\ \mathbf{J}_{o,i} \end{bmatrix}, \quad \xi_i \triangleq [\mathbf{v}_i^T, \boldsymbol{\omega}_i^T]^T \quad (5)$$

$$\mathbf{J}_{p,i} = \begin{bmatrix} \hat{\mathbf{s}}_{1,i} \times (\mathbf{p}_i - \mathbf{p}_{1,i}) & \cdots & \hat{\mathbf{s}}_{6,i} \times (\mathbf{p}_i - \mathbf{p}_{6,i}) \end{bmatrix} \quad (6)$$

$$\mathbf{J}_{o,i} = [\hat{\mathbf{s}}_{1,i} \quad \cdots \quad \hat{\mathbf{s}}_{6,i}], \quad (7)$$

where $\mathbf{v}_i$ represents the linear velocity of the end-effector, $\boldsymbol{\omega}_i$ represents the angular velocity of the end effector, $\mathbf{p}_{j,i}$ represents the position of frame $\{j, i\}$ and $\hat{\mathbf{s}}_{j,i}^\dagger$ represents the axis of rotation of the corresponding joint.

B. Resolved Motion Rate Control

Control of both manipulators is achieved by a resolved motion rate algorithm that separately tracks joint commands. This was first proposed in [31] and has also been adapted by Marani et. al. for singularity avoidance in control of the SAUVIM underwater robot's manipulator [32]. The desired pose, denoted as $\mathbf{x}_{d,i} \triangleq (\mathbf{p}_{d,i}, \mathbf{R}_{d,i})$, is specified in real-time by operator input as described in Section III. Therefore, the position error $\mathbf{e}_{p,i}$ of manipulator i between the current pose $\mathbf{x}_i$ is given by:

$$\mathbf{e}_{p,i} = \mathbf{p}_{d,i} - \mathbf{p}_i. \quad (8)$$

$^\dagger(\cdot)$ denotes a unit vector.

The orientation error $\mathbf{e}_{\mu,i}$ of manipulator i is found by first deriving the relative error rotation matrix $\mathbf{R}_{e,i}$ between the current orientation $\mathbf{R}_i$ and desired orientation $\mathbf{R}_{d,i}$:

$$\mathbf{R}_{e,i} = \mathbf{R}_{d,i} \mathbf{R}_i^T \quad (9)$$

$$\theta_i = \arccos \frac{\text{tr}(\mathbf{R}_{e,i}) - 1}{2} \quad (10)$$

$$\mathbf{e}_{\mu,i} = \frac{\theta_i}{2 \sin \theta_i} \begin{bmatrix} \mathbf{R}_{e,i}(3,2) - \mathbf{R}_{e,i}(2,3) \\ \mathbf{R}_{e,i}(1,3) - \mathbf{R}_{e,i}(3,1) \\ \mathbf{R}_{e,i}(2,1) - \mathbf{R}_{e,i}(1,2) \end{bmatrix}. \quad (11)$$

The position and orientation errors are used in resolved motion rate control to calculate the desired end-effector twist ξ_i . As the end-effector approaches the desired position and the errors approach zero, the magnitude of the end-effector twist decreases:

$$v_{\text{mag}} = \begin{cases} v_M & \|\mathbf{e}_{\mathbf{p},i}\| > \frac{\lambda_p}{\delta_p} \\ v_m + (v_M - v_m) \frac{\|\mathbf{e}_{\mathbf{p},i}\| - \delta_p}{\delta_p(\lambda_p - 1)} & \|\mathbf{e}_{\mathbf{p},i}\| \leq \frac{\lambda_p}{\delta_p} \end{cases} \quad (12)$$

$$\omega_{\text{mag}} = \begin{cases} \omega_M & \|\mathbf{e}_{\mu,i}\| > \frac{\lambda_\mu}{\delta_\mu} \\ \omega_m + (\omega_M - \omega_m) \frac{\|\mathbf{e}_{\mu,i}\| - \delta_\mu}{\delta_\mu(\lambda_\mu - 1)} & \|\mathbf{e}_{\mu,i}\| \leq \frac{\lambda_\mu}{\delta_\mu} \end{cases} \quad (13)$$

$$\mathbf{v}_i = v_{\text{mag}} \frac{\mathbf{e}_{\mathbf{p},i}}{\|\mathbf{e}_{\mathbf{p},i}\|} \quad (14)$$

$$\boldsymbol{\omega}_i = \omega_{\text{mag}} \frac{\mathbf{e}_{\mu,i}}{\|\mathbf{e}_{\mu,i}\|}, \quad (15)$$

where v_M and v_m represent the maximum and minimum linear velocity of the end-effector; ω_M and ω_m represent the maximum and minimum angular velocity of the end-effector; δ_p and δ_μ represent the allowable position and orientation error thresholds; and λ_p and λ_μ represent the relative error radius in which the end-effector switches from a constant (resolved) rate to a reduced proportional rate. The joint velocities can be calculated from the twist defined in Eq. 5:

$$\dot{\mathbf{q}}_i = \mathbf{J}_i^{+*} \xi_i, \quad (16)$$

where $\mathbf{J}_i^{+}$ denotes Moore-Penrose pseudo-inverse of $\mathbf{J}_i$. The manipulator joint commands can then be found iteratively:

$$\mathbf{q}_i = \mathbf{q}_i + \dot{\mathbf{q}}_i dt. \quad (17)$$

III. TELEOPERATION

Two 3D Systems Touch haptic devices are designated as the leaders for bimanual teleoperation of the UVMS. The devices each have 6 degrees of freedom of motion and are equipped with 2 buttons on the stylus. The software package presented in [29] provides the necessary APIs to communicate with the devices using Robot Operating System (ROS) [33], and retrieve information such as the pose of the stylus and the state of the buttons. The teleoperation framework can be separated into commanding manipulator trajectory and vehicle teleoperation. The type of teleoperation command executed by the UVMS is determined by the

sequence or combination of buttons pressed on the stylus of one or both devices. The grippers on the left or right manipulator are toggled between open and closed states by clicking both buttons simultaneously on the corresponding left or right haptic device.

A. Commanding Manipulator Trajectory

Teleoperation of the manipulators is commanded when only the proximal button, referred to as the manipulator button, on the corresponding stylus is engaged. In this proposed system, both follower manipulators can be controlled simultaneously or individually. When teleoperation of a follower serial manipulator is initiated, the haptic stylus pose at this instant, known as the anchor pose, is registered as $({}^H\mathbf{T}_{s,a})_i$. The subsequent poses of the moving stylus, or current pose, while the button stays engaged is represented as $({}^H\mathbf{T}_{s,c})_i$ as seen in Fig 4a. The relationship between the anchor and current pose of the haptic stylus is as follows:

$$({}^H\mathbf{T}_{s,c})_i = ({}^H\mathbf{T}_{s,a})_i ({}^{s,a}\mathbf{T}_{s,c/a})_i, \quad (18)$$

where $({}^{s,a}\mathbf{T}_{s,c/a})_i$ represents the change in pose from the anchor frame to the current frame of the corresponding stylus, as represented in the anchor frame. This relative change in pose corresponds to the follower manipulator's relative change in pose using a similarity transformation by:

$$({}^{ee,a}\mathbf{T}_{s,c/a})_i = ({}^{ee,a}\mathbf{T}_{s,a})_i ({}^{s,a}\mathbf{T}_{s,c/a})_i ({}^{ee,a}\mathbf{T}_{s,a})_i^{-1}, \quad (19)$$

where $({}^{ee,a})$ is the end-effector frame corresponding to the manipulator's pose when the manipulator button is initially pressed. $({}^{ee,a}\mathbf{T}_{s,a})_i$ is computed using the following expression:

$$({}^{ee,a}\mathbf{T}_{s,a})_i = ({}^{ROV}\mathbf{T}_{ee,a})_i^{-1} ({}^{ROV}\mathbf{T}_H)_i ({}^H\mathbf{T}_{s,a})_i, \quad (20)$$

where $({}^{ROV}\mathbf{T}_{ee,a})_i$ is the manipulator anchor pose as written in the vehicle base frame.

This transformed relative pose is essential to compute the pose of the teleoperated manipulator written in the base frame of the vehicle using the following expression:

$$({}^{ROV}\mathbf{T}_{ee,d})_i = ({}^{ROV}\mathbf{T}_{ee,a})_i ({}^{ee,a}\mathbf{T}_{s,c/a})_i, \quad (21)$$

where $({}^{ROV}\mathbf{T}_{ee,d})_i$ is the end-effector desired pose $\mathbf{x}_{d,i}$ in the form of a 4×4 homogeneous transformation. The desired pose is used in the resolved motion rate controller introduced in Section II.

B. Vehicle Teleoperation

Teleoperation of the vehicle occurs when both leader haptic devices have only the distal button on their stylus pressed. When the buttons are pressed, the initial positions of the styluses are registered and corresponding positions are rewritten as vectors denoting the relative position. A vector in the haptic base frame is represented component-wise in the axes of the same frame. Note that the rotation of the stylus pose is ignored for this mode. To command motion of the vehicle, the vectors of both devices must have the same dominant component, and both vectors must have a norm above a nominal distance. The condition for the vector norm

*The pseudo-inverse of the Jacobian is used in the general case; however, in this paper the manipulator is presumed to have 6 DoFs, in which case the Jacobian is square and the standard inverse can be used instead.

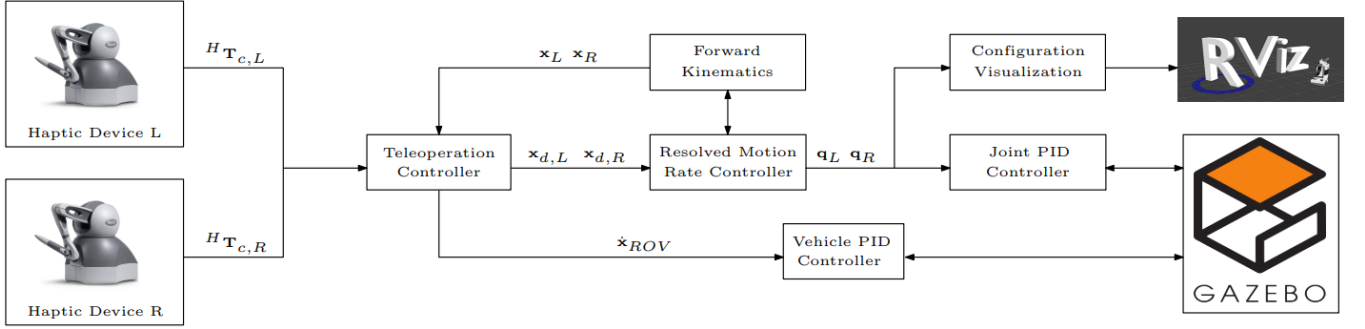
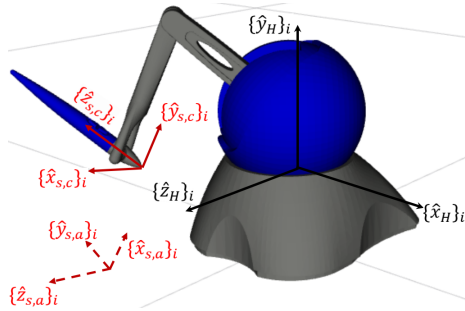
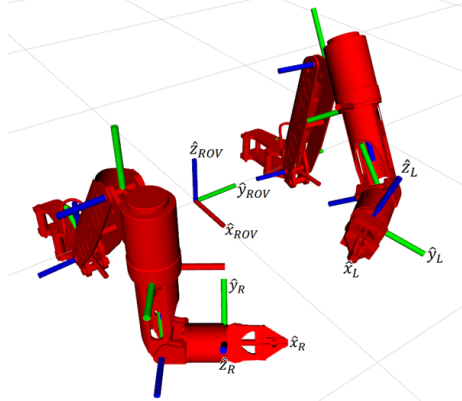


Fig. 3: Control architecture for bimanual teleoperation of UVMS.



(a) Coordinate frames of single leader haptic device, with anchor frame indicated.



(b) Coordinate frames of dual follower manipulators, visualized using RViz.

Fig. 4: Coordinate frames of haptic devices and manipulators, including anchor frames.

is to create a dead-zone, which will prevent the vehicle from moving in unwanted directions by filtering out small user motions when initiating vehicular teleoperation. While both leader device vectors share the same dominant component, they must also share the same direction to command a constant velocity along that direction. In the $\{\mathcal{ROV}\}$ base frame, this translates to a linear velocity in the direction of its components. Additionally, the yaw of the vehicle can also be controlled when the two haptic vectors have the dominant $\hat{\mathbf{z}}_{H,i}$ axes but are in opposite directions. In this case, the angular velocity of the vehicle in the $\hat{\mathbf{z}}_{ROV}$ axes is commanded.

C. Control Architecture

Figure 3 displays the control architecture for bimanual teleoperation of a UVMS. The two haptic devices communicate their poses and button states in real-time to the teleoperation controller. When controlling the manipulators, the teleoperation controller communicates the desired poses to the resolved motion rate controller, which computes the joint velocities to determine and command joint positions of the follower manipulators to the joint PID controller in Gazebo. An RViz [34] visualization of the manipulator poses is also deployed simultaneously. The teleoperation controller also sends vehicle velocity commands directly to its PID controller in Gazebo.

IV. SIMULATION MODEL

To test the proposed teleoperational framework, the bimanual UVMS is imported into the Gazebo simulator, a 3D dynamic environment for modeling robots, actuators, and sensors in a realistic world [27]. The physics engine is capable of modeling gravity, inertia, friction, and collision detection between objects. Self-collision detection is possible with Gazebo, but is not utilized in this paper. The actuation of the serial manipulators is achieved via joints which define the kinematic and dynamic relationship between links. Another important attribute of this simulator is that it is open-source and compatible with many custom plugins, including integration with ROS to control the robot.

To simulate underwater vehicles in Gazebo, the UUV Simulator is used, which contains custom Gazebo plugins for underwater objects such as thrusters and sensors [28]. In addition, the package is compatible with ROS nodes to allow for features such as dynamic position (DP) control of the vehicle, trajectory generation, and manipulator control. The package also expands the dynamic model to include hydrodynamics and hydrostatics such as buoyancy and drag forces.

The purpose of the simulation is to provide preliminary validation of the framework's performance in two different tasks. The first task is manual piloting of the vehicle, manipulators, and end-effectors to a desired pre-grasp pose. In this task, the ability to execute commands of all components of the robot is demonstrated. In addition, the teleoperation command, resolved motion rate controller command, and

actual position of the end-effectors are compared to assess the accuracy of the framework. The second task involves grasping a small object using both manipulators simultaneously. This demonstrates precise, coordinated motion of the whole robot with just a single novice operator, as well as the ability of the simulator to model the physics of a multi-point grasp.

A. Vehicle Model

The RexROV is the platform utilized in this simulation. This vehicle was used because it is one of the default vehicles included in the UUV Simulator packages. In addition, the size and thruster configuration of the vehicle is similar to that of light-duty or work class ROVs such as the Saab Lynx or Cougar models [35]. However, the proposed framework can easily be generalized to any light-duty underwater vehicle, such as the BlueROV2. This is to be consistent with the goal of lowering the barrier to entry into UVMS control through compatibility with a variety of UVMS designs. The RexROV model has a standard eight-thruster configuration which gives full six-DoF control of the vehicle. In addition, the model also has simulated on-board cameras which allow the user to control the vehicle using realistic visual feedback.

To control the vehicle pose, a combined PID position and velocity controller is implemented. This hybrid controller maintains an upright orientation of the vehicle by using position control to set both the roll and pitch of the vehicle to zero. The user cannot control these vehicle DoFs. The controller's velocity PID control allows the user to control the linear velocities and yaw rate of the vehicle. The controller combines the orientation control with the user's desired velocity control to publish a desired vehicle wrench. A thruster controller generated by the UUV Simulator package subscribes to the vehicle wrench and calculates the actuation (thruster) forces required to achieve this wrench. The thruster controller is automatically generated based on the geometry of the thruster placement and the thruster gains.

The PID controller allows the user to send velocity commands to the vehicle in one of two ways. The default method is via gamepad controller or joystick, which is a common method of controlling small ROVs such as the BlueROV2. However, we also developed a novel vehicle control method that uses the haptic devices to generate desired vehicle commands when the user presses the vehicle button on both manipulators. This allows for a single operator to effortlessly switch between vehicle and manipulator control.

B. Manipulator Model

The framework provided can be used for any desired configuration or manipulator, with the only required modification being that of the kinematic model parameters to match the geometry of the manipulators. For example, the Reach Alpha by Reach Robotics [36] could be used instead of the simple design shown on the BlueROV2 in Fig. 2.

The manipulators are controlled via a PID position controller for each joint, including the gripper joints. The desired end-effector poses from the teleoperation tracking

are calculated independently based on the transformation between the user's current pose and the anchor pose set when the manipulator button is engaged. Since there are only two buttons on each haptic device, the gripper is actuated when the user presses both buttons at the same time. The desired poses are sent to a resolved motion rate controller which calculates the joint commands to achieve the desired pose. As such, there are two types of error to be considered: pose error of the end-effector and joint position errors of the PID controller for each joint.

V. SIMULATION RESULTS

Task 1: Motion Control of UVMS

The first simulation task is to implement the proposed framework for real-time teleoperational control of the entire UVMS. In this task, the robot will move into a pre-grasp position to interact with a subsea panel, which is a common task required for intervention ROVs. This task has two modes: the vehicle motion and the manipulation mode. In the first mode, the vehicle is piloted into position using both haptic devices, and the manipulators cannot be controlled. In the second mode, the manipulators and grippers are controlled independently, and the vehicle cannot be controlled. The operator can freely switch between either mode as needed to move the end-effectors into the desired pre-grasp pose.

Figure 5 summarizes the framework implementation over the duration of the simulation. First, the UVMS is spawned and the thruster and manipulator controllers are activated. Once this occurs, both the vehicle and the manipulators will remain stationary until the user gives a command via the haptic devices, as shown in Fig. 5a. Next, the vehicle velocity control is demonstrated using the haptic devices to pilot the robot into an ideal position. This is done by pressing the vehicle button on each haptic device and using the haptic device pens to move the vehicle in strafing motions or rotational (yaw) motion. Figure 5b shows the vehicle after it has been piloted to its pre-grasp position. Once the vehicle is in the desired position, the manipulators are engaged and moved to the desired position by pressing the manipulator button on the pen. The grippers are opened and closed by pressing both buttons on the corresponding pen. During this task, the vehicle camera view is used to provide a realistic simulated view of what the operator would see; the default third person view can be used for assistance in positioning, but it is important to note that in a real teleoperational task this would likely not be available to the operator. The camera view used in the simulation is shown in Fig. 5c. Finally, Fig. 5d shows the haptic device motion during the simulation, indicating the actual commands given by the operator.

The error associated with Task 1 is shown in Fig. 6. Figure 6a shows the joint commanded and actual positions of each joint for the left manipulator. The PID controller error is very small for most joints but more pronounced in joint 2 and, to a lesser extent, joint 3. This makes sense because the second joint is the one which bears the greatest load due to gravity in the default configuration. This error is not significant because the operator can easily compensate using visual feedback

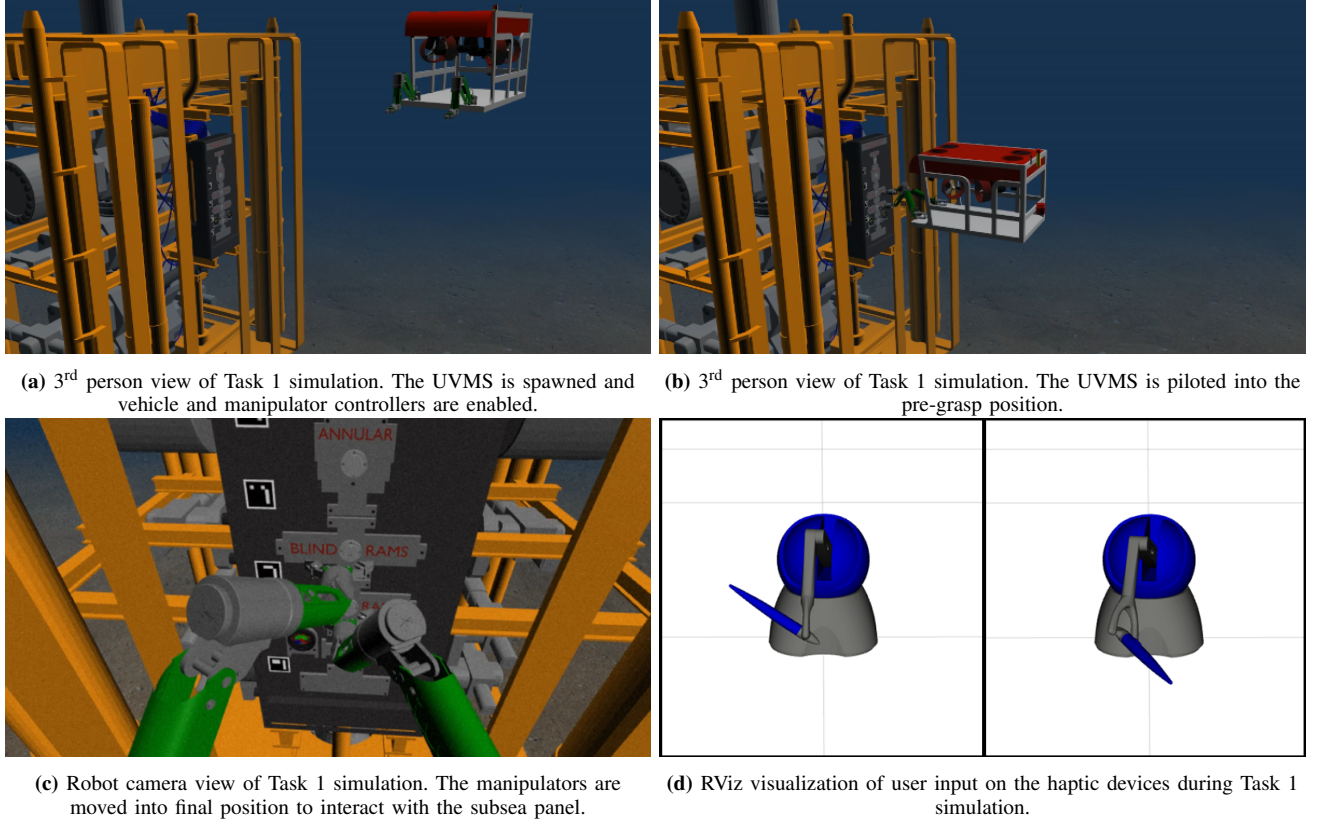


Fig. 5: Task 1 demonstration: controlling vehicle and manipulators using haptic devices.

from the vehicle cameras to get the manipulators into the desired position. The PID gains of these joints can also be adjusted to reduce this error if desired.

The commanded and actual end-effector positions in the ROV base frame are shown in Fig. 6b. The error is greatest in the Z direction, small in the X direction, and very small in the Y direction. This is consistent with the joint 2 errors, representing the robot sagging under the effects of gravity and causing primarily the Z position to be lower in actuality than commanded. As before, this can be compensated for manually by the operator or by adjusting the PID gains.

Task 2: Bimanual Manipulation of an Object

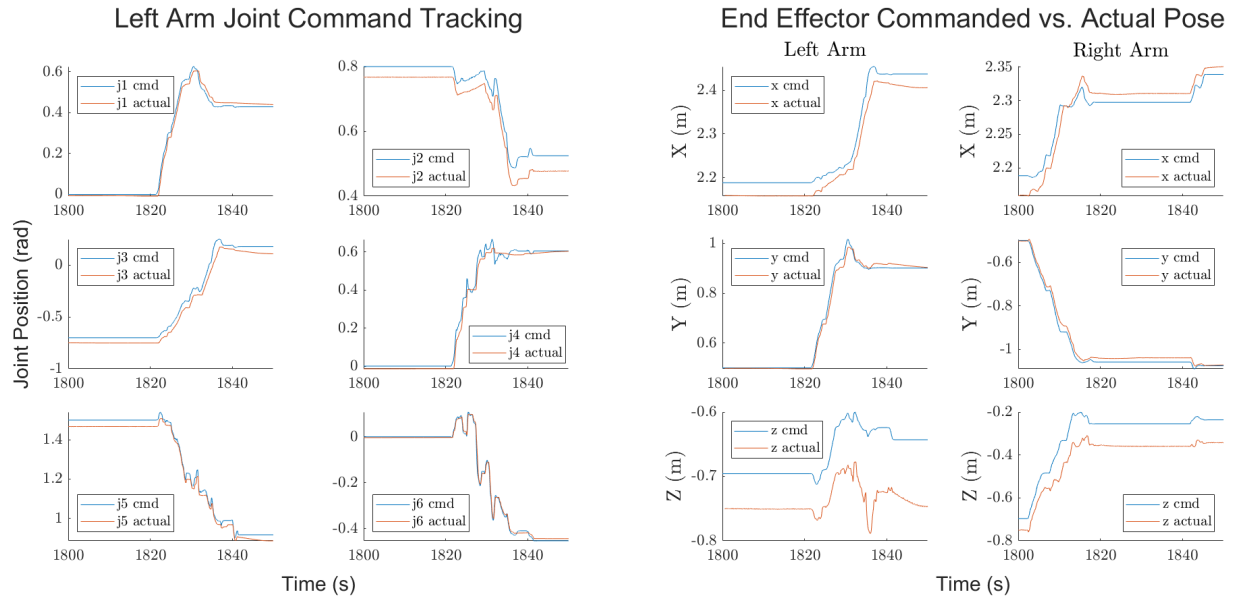
The second task uses the framework to perform a more challenging task to grasp an object using both manipulators simultaneously. This task is similar in concept to the previous task and uses the same two modes, but with a greater emphasis on precise manipulation and the physics of grasping. The object in this case is a simple rectangular box spawned into Gazebo for demonstration purposes. In this task, the object is grasped with both manipulators, which requires coordination and precision. The fact that this can be grabbed by a novice operator demonstrates the usefulness of the proposed framework. Another advantage to using bimanual grasping is that the object can be stabilized using both manipulators. A single gripper would have difficulty grabbing such an object because of its smooth surface, allowing the object to slip both length-wise as well as laterally out of the gripper. However,

using both manipulators at once allows the object to be caged by both grippers, forming a secure grasp to prevent lateral motion.

Figure 7 shows Task 2 in execution. At top, the third-person view of the robot shows the object being lifted off the seafloor. At bottom, the robot view is shown, which is what the operator will typically use in a real teleoperational task. Once grasped, the object can be transported by moving the vehicle itself or by moving the manipulators; however, the operator must be careful to move the object with both manipulators in coordination, or the grasp will fail.

VI. CONCLUSION

In this paper, an open-source bimanual teleoperation framework for a light duty UVMS is introduced. The novelty of the framework lies in the use of low-cost and readily available haptic devices to control both the vehicle and manipulator motions. The requisite UVMS kinematics are defined and a resolved motion rate controller is given to independently control both manipulators. The teleoperational mapping between the haptic devices and the desired manipulator pose is also derived. The framework is validated in simulation in two tasks. The first task is to demonstrate the motion control of the vehicle, manipulators, and grippers using the haptic devices only. The performance of the robot is assessed by analyzing the joint and end-effector position errors. The second task is to grasp an object using both manipulators simultaneously, which demonstrates the precise



(a) Comparison of individual left arm joint commanded and actual position during Task 1 simulation.

(b) Comparison of end-effector commanded and actual position during Task 1 simulation.

Fig. 6: Task 1 manipulator performance analysis.

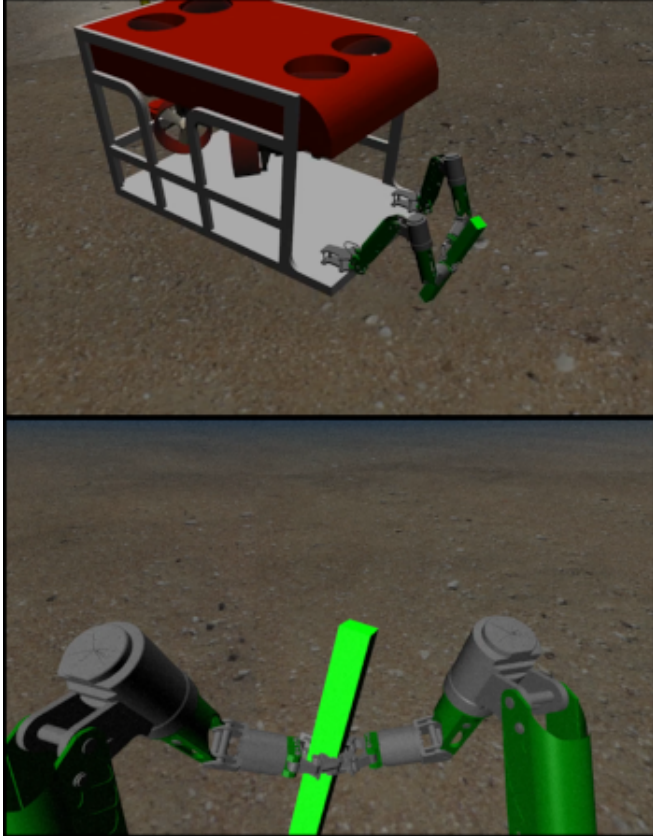


Fig. 7: Task 2 demonstration: both manipulators are engaged to grasp a rectangular object, shown in green.

and coordinated control of the UVMS using the proposed framework.

Future development of the framework includes coordination between macro-level and micro-level control of the UVMS. The manipulator and vehicle control presented in the current framework constitutes micro-level control. Macro-level control would encompass simultaneous localization and mapping (SLAM) and autonomous execution of user-generated waypoint trajectories. Such a framework would be very valuable for completing more complex tasks and for lowering the barrier to entry into underwater robotics. In addition, the incorporation of haptic feedback to assist the operator could provide useful capabilities such as virtual fixtures to guide the operator through a pre-determined task or indicate contact with an object. Finally, testing the framework on an experimental platform would demonstrate the real-world utility of this framework.

REFERENCES

- [1] J. A. Trotter, C. Pattiaratchi, P. Montagna, M. Taviani, J. Falter, R. Thresher, A. Hosie, D. Haig, F. Foglini, Q. Hua *et al.*, "First roV exploration of the perth canyon: Canyon setting, faunal observations, and anthropogenic impacts," *Frontiers in Marine Science*, vol. 6, no. 173, 2019.
- [2] V. H. Fernandes, A. A. Neto, and D. D. Rodrigues, "Pipeline inspection with auv," in *IEEE/OES Acoustics in Underwater Geosciences Symposium (RIO Acoustics)*, 2015, pp. 1–5.
- [3] P. Rundtop and K. Frank, "Experimental evaluation of hydroacoustic instruments for roV navigation along aquaculture net pens," *Aquacultural Engineering*, vol. 74, pp. 143–156, 2016.
- [4] D. Ribas, P. Ridao, L. Magí, N. Palomeras, and M. Carreras, "The girona 500, a multipurpose autonomous underwater vehicle," in *Oceans 2011 IEEE-Spain*. IEEE, 2011, pp. 1–5.
- [5] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *IEEE/ASME Transactions on mechatronics*, vol. 17, no. 1, pp. 46–53, 2012.

- [6] J. Yuh, S. Choi, C. Ikehara, G. Kim, G. McMurty, M. Ghasemi-Nejhad, N. Sarkar, and K. Sugihara, "Design of a semi-autonomous underwater vehicle for intervention missions (sauvim)," in *Proceedings of 1998 international symposium on underwater technology*. IEEE, 1998, pp. 63–68.
- [7] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009, autonomous Underwater Vehicles.
- [8] R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard *et al.*, "Development of the remus 600 autonomous underwater vehicle," in *Proceedings of OCEANS 2005 MTS/IEEE*. IEEE, 2005, pp. 1301–1304.
- [9] F. Jaffre, R. Littlefield, M. Grund, and M. Purcell, "Development of a new version of the remus 6000 autonomous underwater vehicle," in *OCEANS 2019-Marseille*. IEEE, 2019, pp. 1–7.
- [10] "Blue robotics, inc." <https://bluerobotics.com/>, accessed: 2023-11-14.
- [11] W. J. Marais, S. B. Williams, and O. Pizarro, "Anisotropic disturbance rejection for kinematically redundant systems with applications on an uvm," in *IEEE Robotics and Automation Letters*, vol. 6, no. 4. IEEE, 2021, pp. 7017–7024.
- [12] —, "Go with the flow: energy minimising periodic trajectories for uvm," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 01–07.
- [13] J. McConnell, J. D. Martin, and B. Englot, "Fusing concurrent orthogonal wide-aperture sonar images for dense underwater 3d reconstruction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1653–1660.
- [14] J. McConnell and B. Englot, "Predictive 3d sonar mapping of underwater environments via object-specific bayesian inference," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6761–6767.
- [15] J. McConnell, F. Chen, and B. Englot, "Overhead image factors for underwater sonar-based slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4901–4908, 2022.
- [16] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey," *Robotics and Autonomous Systems*, vol. 150, p. 103973, 2022.
- [17] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, P. Mullins, M. Barham, C. R. Voolstra, K. N. Salama, M. L'Hour, and V. Creuze, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robotics Automation Magazine*, vol. 23, no. 4, pp. 20–29, 2016.
- [18] M. V. Jakuba, C. R. German, A. D. Bowen, L. L. Whitcomb, K. Hand, A. Branch, S. Chien, and C. McFarland, "Teleoperation and robotics under ice: Implications for planetary exploration," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–14.
- [19] J. E. Manley, S. Halpin, N. Radford, and M. Ondler, "Aquanaut: A new tool for subsea inspection and intervention," in *OCEANS 2018 MTS/IEEE Charleston*, 2018, pp. 1–4.
- [20] K. P. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, and G. A. Demetriou, "Control architectures for autonomous underwater vehicles," *IEEE Control Systems Magazine*, vol. 17, no. 6, pp. 48–64, 1997.
- [21] C. Sayers, D. R. Yoerger, R. P. Paul, and J. Lisiewicz, "A manipulator work package for teleoperation from unmanned untethered vehicles—current feasibility and future applications," in *International Advanced Robotics Program Workshop on Subsea Robotics*. Citeseer, 1996.
- [22] C. P. Sayers and R. P. Paul, "An operator interface for teleprogramming employing synthetic fixtures," *Presence: Teleoperators & Virtual Environments*, vol. 3, no. 4, pp. 309–320, 1994.
- [23] N. Shirakura, T. Kiyokawa, H. Kumamoto, J. Takamatsu, and T. Ogasawara, "Collection of marine debris by jointly using uav-uuv with gui for simple operation," *IEEE Access*, vol. 9, pp. 67 432–67 443, 2021.
- [24] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster, "A telerobotic system for transnasal surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 996–1006, 2014.
- [25] "3d systems, inc." <https://3dsystems.com/>, accessed: 2023-11-21.
- [26] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M. Srinivasan, "Haptics in minimally invasive surgical simulation and training," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 56–64, 2004.
- [27] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [28] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *OCEANS 2016 MTS/IEEE Monterey*. IEEE, sep 2016. [Online]. Available: <https://doi.org/10.1109/OCEANS.2016.7761080>
- [29] Y.-H. Su, A. Munawar, A. Deguet, A. Lewis, K. Lindgren, Y. Li, R. H. Taylor, G. S. Fischer, B. Hannaford, and P. Kazanzides, "Collaborative robotics toolkit (crtk): Open software framework for surgical robotics research," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, 2020, pp. 48–55.
- [30] "Haptic bimanual teleoperation simulation," <https://github.com/stevens-armlab/uvm-bimanual-sim>, accessed: 2024-01-18.
- [31] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on man-machine systems*, vol. 10, no. 2, pp. 47–53, 1969.
- [32] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1973–1978.
- [33] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.
- [34] "Rviz," <https://wiki.ros.org/rviz>, accessed: 2024-01-22.
- [35] Saab Seaeye Ltd, <https://www.saabseaeye.com/solutions/underwater-vehicles/compare>, accessed: 2024-01-08.
- [36] Reach Robotics, <https://reachrobotics.com/>, accessed: 2023-07-05.