UNIVERSITY OF GENOVA

PHD PROGRAM IN ROBOTICS AND INTELLIGENT MACHINES

Task-Based Motion Control of an Autonomous Surface Vehicle in a Combined System with a Remotely Operated Vehicle (ASV-ROV)

by

Juri Khanmeh

Thesis submitted in partial fulfillment for the degree of *Doctor of Philosophy* (38° cycle)

February 2026

Giovanni Indiveri	Supervisor
Enrico Simetti	Supervisor
Antonio Sgorbissa	Head of the PhD program

Thesis Jury:

Alessandro Marino, *Università di Cassino e del Lazio Meridionale*	External examiner
Alessandro Ridolfi, *Università di Firenze*	External examiner
Giorgio Cannata, *Università di Genova*	Internal examiner

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

I would like to dedicate this thesis to …

All my wonderful friends and people I met in Germany and Italy, who have been very kind, supportive, and thoughtful to me. Thanks to you, I got my confidence back, I grew, and I feel stronger now.

My aunt and uncle. Thank you for being my home and filling me with love and care.

My siblings. Thank you for supporting my dream and taking care of my parents when I was far from home.

My parents, who always taught me to be positive and kind to others. Thank you for guiding me so far, and giving me the freedom to go, travel, discover the beauty of the world, and find my own happiness.

My old friends and family in Syria. Thank you for always cheering me up from this parallel world despite the tragedy you have been living. I wish I could see you in a better place, where I believe you have the life and the opportunities you deserve.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Juri Khanmeh
January 2026

Acknowledgements

This project was funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project “RAISE - Robotics and AI for Socio-economic Empowerment” (ECS00000035). Enrico Simetti and Giovanni Indiveri are part of RAISE personnel.

I would like to thank the German Research Center for Artificial Intelligence (DFKI) for providing the infrastructure for the experimental setup and for their technical support especially for the data acquisition part.

Abstract

Marine robotics plays a crucial role in carrying out a wide range of complex underwater missions. In this context, tethered Remotely Operated Vehicles (ROVs) provide several advantages, including efficient data transmission and a reliable physical connection that ensures safety in emergency situations. However, their operational range is inherently limited by the length of the tether. This study investigates the possibility of extending the operational range of ROVs by introducing a novel control strategy that enables cooperative operation with an Autonomous Surface Vehicle (ASV). The proposed ASV–ROV system addresses two primary challenges: the restricted mobility of the ROV and the potential risk of cable entanglement. To mitigate these issues, a new control framework is developed to ensure smooth ROV motion while minimizing conditions that could lead to tether entanglement. The control system allows the ASV to maintain alignment with the ROV and preserve a desired distance, while simultaneously regulating the length of cable deployed in the water. This thesis presents the control design of the ASV motion for a cooperative navigation with the ROV. The work includes the formulation of a novel control strategy that enables the ASV to autonomously track and follow the ROV while avoiding obstacles and adjusting its trajectory to maintain safe and efficient cooperation. The proposed control framework is validated through comprehensive software-in-the-loop simulations that represents offshore wind farm environments. The numerical results demonstrate the effectiveness and robustness of the system in improving cooperative navigation performance, reducing power consumption, and enhancing operational resilience under dynamic marine conditions. Furthermore, the work of this thesis investigates the impact of tether management on the overall stability and energy efficiency of the ASV–ROV system. An analytical assessment of tether dynamics is conducted, supported by experimental validation in a controlled test basin, where precise measurements about underwater tether behavior were collected. Directions for further research and system optimization are also discussed.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 The Demand of Remotely Operated Vehicle

Underwater robotic applications have expanded significantly over the past decade, driven by the growing number of construction projects taking place in seas and oceans. This trend has generated a substantial demand for autonomous systems capable of performing subsea intervention and infrastructure inspection missions. Among the most prominent underwater construction projects today are Subsea Production Systems (SPS) and offshore wind farm developments.

SPS installations are typically located on the seafloor at depths exceeding 2000 meters, where they extract and transport oil and gas through subsea manifolds and pipeline networks connected to production platforms. These pipelines can extend over large areas and must be regularly monitored and maintained to ensure a longer operational lifespan. Similarly, offshore wind farms are increasingly being deployed in marine environments due to the rising global interest in renewable and sustainable energy sources. Such installations can cover areas of up to 300 km^2 and also require routine inspections and, when necessary, repairs to address operational failures.

Remotely Operated Vehicles (ROVs) are unmanned underwater robots tethered to a control station and controlled by an operator onboard. They represent one of the primary technological solutions for conducting subsea operations. Owing to their mechatronic architecture, ROVs are highly modular, allowing the integration of customized sensors and manipulators depending on the specific application. Furthermore, the tether connection to the surface control station provides a communication link with significantly higher bandwidth compared to untethered underwater systems. This high-speed data transmission enables

real-time video streaming to the operator—an essential feature for inspection and repair missions, whether partially or fully autonomous.

Despite these advantages, ROV operations are not yet fully autonomous. Each mission requires transporting the robot to the operational site typically using a crewed surface vessel, deploying it, managing the tether, and recovering the system afterward. These procedures are logistically complex and costly, as they demand skilled personnel to operate both the ROV and the surface vessel, coordinate their movements, and determine the appropriate tether length to be deployed.

To address these limitations, one proposed cooperative approach involves integrating an ROV with an Autonomous Surface Vehicle (ASV) to form a fully autonomous ASV–ROV system. In this configuration, the ASV functions as an intermediary platform that transmits data and connects the ROV to a remote ground station, allowing the operator to control the ROV without being physically present on the vessel. The ASV autonomously manages deployment, follows the ROV during the mission, handles tether release, and retrieves the vehicle upon completion. Consequently, the ASV–ROV configuration has emerged as a promising and rapidly developing topic in marine robotics, offering significant operational and economic benefits.

1.2 Challenges

Although the ASV–ROV cooperative system offers numerous advantages, its most significant drawback lies in the physical coupling between the two vehicles. This physical link introduces additional complexity in coordinating the motion of the ASV and the ROV, especially in disturbance-prone environments such as the marine domain, where onboard power resources are limited. This constraint requires the ASV, in particular, to minimize energy consumption by avoiding unnecessary maneuvers, as it is a relatively large vehicle that consumes more power than the ROV.

Moreover, during offshore wind farm inspections, pillar-like obstacles such as wind turbine foundations present additional challenges. The ASV must not only avoid collisions but also ensure that the tether remains clear of obstacles and follow the ROV without allowing the cable to become entangled or snagged around the structures.

Another important aspect of this ASV–ROV configuration is umbilical (tether) management. When the ROV dives deeper and changes its vertical distance from the ASV, the ASV must release tether proportionally to the ROV's depth. Therefore, precise regulation of cable

length is essential, ensuring it is neither too short, which would restrict the ROV's motion, nor too long, which would increase hydrodynamic drag and resistance.

1.3 Related Work

Although the ASV-ROV cooperative system has been considered to be a promising configuration for marine robotics, there are limited studies covering this topic. Basically, the reseaches about ASV-ROV system is divided into three main subjects: the Navigation, Guidance and Control (NGC) of the ASV, the mechatronic and modelling of the system, and finally the ROV umbilical management. Regarding the first topic; the research published in Conte et al. (2016) proposes a guidance system for the ASV based on a moving target tracking algorithm with the aim of keeping a certain distance from the ROV, and experimental tests were published for a ROS based NGC system of the developed ASV prototype. Similarly in the work presented in Morinaga and Yamamoto (2023), the authors propose a reference waypoint to control the position of the ASV, preventing the cable from pulling the ROV.

Moving to the mechatronic definition of the system, the works presented in Kapetanovic et al. (2021) and Yamamoto et al. (2024) describe the structure of their ASV-ROV prototype; the first paper presents a system of heterogeneous cooperative autonomous robotic system for potential use in fish cage inspection and monitoring, the latter one proposes a system for monitoring seaweeds and CO2 measurements in the oceans. Concerning the modelling aspect, in the paper presented in Zhao et al. (2021a) a nonlinear model is proposed to study the effect of the tether for this combined ASV-ROV system. In addition, based on the previous study, both static and following ASVs are modeled in the work in Zhao et al. (2022). Another study published in Zhao et al. (2021b) presents the hydrodynamic models and control algorithms of the winch system for deploying/recovering the ROV in the ASV-ROV configuration. The study demonstrates the required umbilical length for different ROV targets and appropriate winch speeds using numerical approaches.

Regarding the ROV umbilical management, in the work presented in Morinaga and Yamamoto (2023), the optimal cable shape for this configuration was assumed to be a catenary, as it minimizes the tension at the ROV attachment point, and an equation for a reference cable length was presented. Another study concerning tether management is proposed in Viel (2022), where the authors consider a passive technique for self-management of the umbilical to estimate the shape of the tether near the attachment point to the ROV by fixing buoys and ballasts to force the cable to have a specific shape. Likewise, in the research published in Laranjeira et al. (2019), a vision-based cable shape identification method is

presented. In the work published in Drupt et al. (2022), another ROV was used in chain to estimate the shape of the umbilical for a negatively buoyant cable. The estimation method is based on the calculation of local tangents from the IMUs data. In Larsen et al. (2025), a tether force sensor for ROVs was developed. The sensor is mounted on the ROV and used to measure the external forces applied to the ROV by the tether. The measurements are then utilized in feed-forward control to enhance the motion of the ROV in a disturbed environment.

In the previous studies, and from a navigation point of view, only basic waypoint-based navigation strategies were implemented in both Conte et al. (2016) and Morinaga and Yamamoto (2023), and the navigation control law remained active even when the ROV operated in close proximity to the ASV. Nevertheless, in the marine environment, there are additional considerations that must be taken into account, such as obstacle avoidance, safety measures, and the minimization of power consumption, all of which can significantly influence the manner in which the ASV follows the ROV. In contrast, the reference umbilical length presented in Morinaga and Yamamoto (2023) is derived using an open-loop control scheme that lacks real-time feedback from the tether itself.

Based on the current state of the art, it is evident that this kind of ASV-ROV system requires a flexible control framework capable of handling multiple tasks. One effective approach to addressing this requirement is the task-priority method, which has been successfully applied to several underwater vehicle systems, as presented in Simetti et al. (2018); Sæbø et al. (2024); Thomas et al. (2021); Di Vito et al. (2021). In Simetti et al. (2018), the Task-Priority Approach was implemented for underwater vehicle–manipulator systems to perform safe waypoint navigation, grasping, landing, and inspection tasks. In Sæbø et al. (2024), the same approach was applied to a path-following controller for snake-like articulated vehicles, ensuring that each link remained aligned with the desired path at all times. Similarly, in Thomas et al. (2021), the authors adopted the Task-Priority framework to integrate multiple objectives, such as obstacle avoidance and seafloor alignment, for the execution of survey missions, homing operations, and docking maneuvers. The proposed control strategy demonstrated improved performance of autonomous vehicles under more challenging environmental conditions compared to previous works. In Di Vito et al. (2021), several sea trials were conducted, demonstrating the flexibility and robustness of the Task-Priority control approach in the kinematic control layer. Thus, the method proved to be efficient and promising for achieving a wide range of desired behaviors in underwater robotic systems.

1.4 Objective of Thesis

Our research in this thesis focuses on building a modular control framework for the coordination of coupled ASV-ROV systems, allowing to resolve conflicting operational requirements, such as precise vehicle tracking and collision avoidance. This helps to improve efficiency and safety by optimizing the ASV motion while following the ROV.

Another objective is to derive a tension model for the ROV tether, enabling the winch in the Tether Management System (TMS) to operate autonomously within a predefined predictive region while continuously monitoring its tension in real time. Rather than defining a reference length at every iteration and keeping the controller continuously active and highly sensitive to positional changes of the ASV and ROV, a predefined length range can be established such that the winch controller is activated only when the tether length falls outside this specified range.

The thesis addresses our contribution in applying the Task-Priority Approach to design and implement the navigation and guidance control system of the ASV for cooperative and coordinated motion with the ROV. This method enhances system modularity and flexibility by allowing the addition of missions, actions, and prioritized tasks that are simultaneously managed according to their assigned importance. In particular, the system is analyzed in simulated sea environment, where pillar-like obstacles are present.

Furthermore, ROV tether management is also addressed. The proposed method generates the reference cable length for the Tether Management System (TMS), which is mounted on the ASV, based on the ROV's position and sensor outputs, including the measured tether tension. And another objective is to design and carry out experiments to acquire real data related to the dynamic behaviour of the tether and enhance the computation of the reference cable length.

1.5 Organization of Thesis

The thesis begins with a general background section that provides an overview of marine vehicle modeling, the control architecture of the ASV, and the Task-Priority Approach. Subsequently, the system design is discussed in the mechatronic design chapter. Following this, the focus shifts to the control strategy developed to reduce tether tension. The first part addresses ASV motion control, while the second presents the computation of the desired ROV tether length. Next, the experimental setup is described for both the ASV–ROV simulation and the data acquisition experiments used for tether modeling. The experimental results are

then presented. The thesis concludes with a summary of the contributions and an overview of the resulting publications.

Chapter 2

Background

In this chapter, essential background information is provided to facilitate the understanding of the subsequent chapters. First, the equations of motion for general marine vehicles are presented. Next, specific models for the ROV and ASV are derived, with particular emphasis on the twin-thruster ASV model. The chapter then discusses the control framework, followed by a brief overview of the Task Priority Approach and the methodology for defining mission tasks within this framework. Finally, an example of an ASV action—path following—is presented to illustrate the implementation of the Task Priority Approach in a practical scenario.

2.1 Marine Vehicle Models

A commonly used dynamic model Fossen (1999) for marine robots includes forces and torques that can be classified as follows:

1. Radiation-induced forces:
    - Added mass due to the inertia of the surrounding fluid
    - Radiation-induced potential dumping due to the energy carried away by generated surface waves
    - Restoring forces due to Archimedes force (weight and buoyancy)
2. Environmental forces:
    - Ocean currents
    - Waves and wind

3. Propulsion forces:
    - Thruster/propeller forces
    - Control surfaces/rudder forces

After applying hydrodynamic forces, the dynamic motion equation for a marine vehicle can be derived by the following equation:

$$\begin{aligned} &\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{v}, \\ &\boldsymbol{v_r} \triangleq \boldsymbol{v} - \boldsymbol{v_c}, \\ &\mathbf{M}\dot{\boldsymbol{v}}_{\boldsymbol{r}} + \mathbf{C}(\boldsymbol{v_r})\boldsymbol{v_r} + \mathbf{D}(\boldsymbol{v_r})\boldsymbol{v_r} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}, \end{aligned} \tag{2.1}$$

$$\mathbf{M} = \mathbf{M_{RB}} + \mathbf{M_A}, \tag{2.2}$$

$$\mathbf{C}(\boldsymbol{v_r}) = \mathbf{C_{RB}}(\boldsymbol{v_r}) + \mathbf{C_A}(\boldsymbol{v_r}), \tag{2.3}$$

$$\mathbf{D}(\boldsymbol{v_r}) = \mathbf{D_P}(\boldsymbol{v_r}) + \mathbf{D_S}(\boldsymbol{v_r}) + \mathbf{D_W}(\boldsymbol{v_r}) + \mathbf{D_M}(\boldsymbol{v_r}), \tag{2.4}$$

where,

- $\boldsymbol{\eta}$ is the generalized position vector;
- $\mathbf{J}(\boldsymbol{\eta})$ Jacobian matrix relating the system velocities with the derivatives of $\boldsymbol{\eta}$;
- $\boldsymbol{v}$ is the generalized system velocity vector projected on the body-frame;
- $\boldsymbol{v_c}$ is ocean current velocity (assumed constant in earth fixed frame and irrotational);
- $\boldsymbol{v_r}$ is vehicle's relative velocity with resprect to the ocean current velocity;
- $\mathbf{M_{RB}}$ is the rigid body inertia matrix;
- $\mathbf{M_A}$ is the added mass inertia matrix due to the inertia of the surrounding fluid;
- $\mathbf{C_{RB}}(\boldsymbol{v_r})$ is the rigid body Coriolis and centripetal matrix;
- $C_A(\boldsymbol{v_r})$ is the hydrodynamic Coriolis and centripetal matrix;
- $\mathbf{D}(\boldsymbol{v_r})$ is the damping matrix;
- $\mathbf{D_P}(\boldsymbol{v_r})$ is the potential damping matrix;
- $\mathbf{D_S}(\boldsymbol{v_r})$ is the skin friction damping matrix;

	DOF	forces	linear vel.	positions
1	motion in the x-direction (surge)	X	u	x
2	motion in the y-direction (sway)	Y	v	y
3	motion in the z-direction (heave)	Z	w	z
		moments	angular vel.	Euler angles
4	rotation about x-axis (roll)	K	p	ϕ
5	rotation about y-axis (pitch)	M	q	θ
6	rotation about z-axis (yaw)	N	r	ψ

Table 2.1 Notation used for marine vehicles.

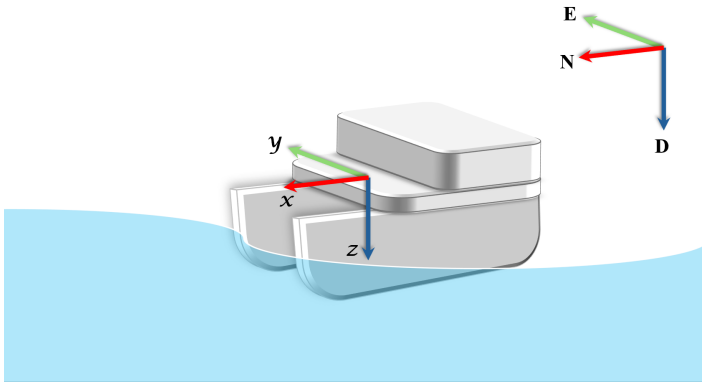


Figure 2.1 Reference Frames: the *white* object represents a marine vehicle. Body-fixed reference frames (x,y,z). NED frame: N pointing to north, E to east, and D to down.

- $\mathbf{D_W}(\boldsymbol{\nu_r})$ is the wave drift damping matrix;
- $\mathbf{D_M}(\boldsymbol{\nu_r})$ is the damping due to vortex shedding matix;
- $\mathbf{g}(\boldsymbol{\eta})$ is the vector of gravitational and buoyancy forces;
- $\boldsymbol{\tau}$ is the vector of external forces and moments acting on the vehilcle;

Table 2.1 introduces the general notations used for marine vehicles. Further details on the equations can be found in Fossen (1999). The following sections discuss the modeling of the ROV and the twin-thruster ASV in detail.

2.1.1 ROV

ROVs are underwater vehicles with 6 degrees of freedom (DoF), i.e. translation and rotation on three coordinated axes.

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^\top, \tag{2.5}$$

$$\boldsymbol{v} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\top, \tag{2.6}$$

where x, y, and z are the coordinates of the vehicle defined with respect to the North–East–Down (NED) reference frame (Figure 2.1), and ϕ, θ, and ψ are the Euler angles representing the vehicle's orientation relative to the NED frame. On the other hand, u, v, and w are the components of the vehicle's linear velocity vector expressed in the body-fixed frame (Figure 2.1), while p, q, and r are the components of the angular velocity vector expressed in the same frame.

According to the Fossen model Fossen (1999), the equation of ROV's motion is given by (2.1) where τ in this case, is a vector of external forces and moments acting on the ROV. The thruster model and control allocation are represented by the following equation:

$$\boldsymbol{\tau} = \mathbf{TKU}, \tag{2.7}$$

where:

- **T** thruster allocation matrix;
- **K** gain matrix, and it is a diagonal matrix with thrust coefficient values on the diagonal;
- **U** input vector, i.e. the propeller's RPM (revolutions-per-minute) value.

2.1.2 ASV

The ASV model is derived from the general case of the Fossen model Fossen (1999) after limiting the degrees of motion to 3 DoFs; surge, sway and yaw. Hence,

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}^\top, \tag{2.8}$$

$$\boldsymbol{v} = \begin{bmatrix} u & v & r \end{bmatrix}^\top. \tag{2.9}$$

Based on the work published in Simetti and Indiveri (2022), the proposed twin-thruster surface vehicle model expresses $\boldsymbol{\tau}_c(\boldsymbol{n})$ as follows:

$$\begin{aligned} \boldsymbol{\tau}_c(\boldsymbol{n}) &\triangleq \boldsymbol{B}\boldsymbol{f}(\boldsymbol{n}) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ d & -d \end{bmatrix} \begin{bmatrix} F_p(n_p) \\ F_s(n_s) \end{bmatrix}, \\ \boldsymbol{\tau}_c(\boldsymbol{n}) &= \begin{bmatrix} F_p(n_p) + F_s(n_s) \\ 0 \\ d(F_p(n_p) - F_s(n_s)) \end{bmatrix}, \end{aligned} \tag{2.10}$$

where:

- $\boldsymbol{n} \triangleq [n_p n_s]^\top$ is the vector of the RPM of the port and starboard motors;
- $F_p(n_p)$ and $F_s(n_s)$ are the port/starboard forces generated by the thrusters on the body;
- d is the distance between the thruster and the central point of the ASV along the transversal axis.

2.2 Control Architecture

The typical control architecture of autonomous marine vehicles can be schematically represented in Figure 2.2. As described in Indiveri and Pascoal (2020), the Mission Control System is the highest level of the system architecture, responsible for planning, scheduling, and generating mission tasks such as path following, trajectory tracking, and position holding. The Guidance block, on the other hand, is the functional subsystem that continuously computes the reference positions, velocities, and accelerations of the vehicle to be used by the Control subsystem. The Navigation block is responsible for estimating the vehicle's position, attitude, and other state variables by processing measurement data obtained from onboard sensors, thus providing the information required to operate the guidance system and close the feedback control loop. Finally, the Control block is responsible for generating the control forces and moments applied to the vehicle's actuators to ensure tracking of the references issued by the guidance system. The design of the control subsystem is based on the dynamic model of the vehicle.

Since this thesis focuses on the implementation of the ASV motion control system, our work is primarily concerned with the Guidance block. The Guidance block includes the

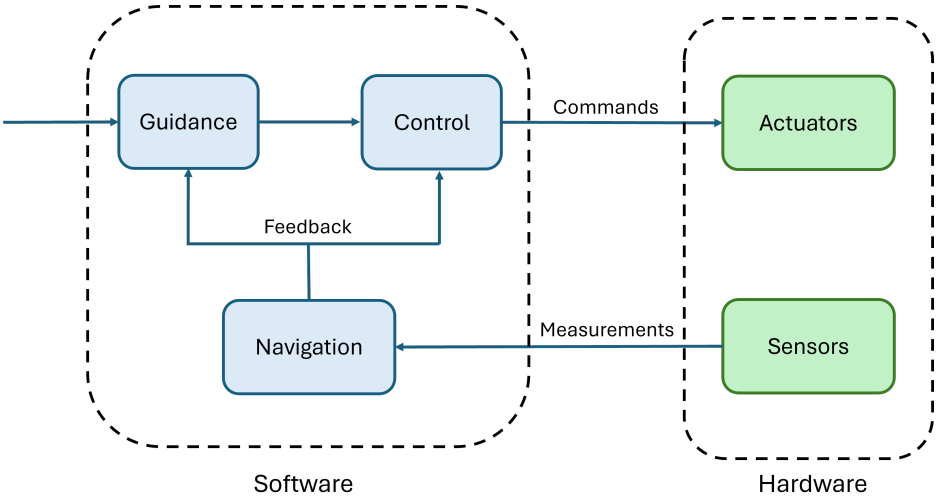


Figure 2.2 General GNC diagram.

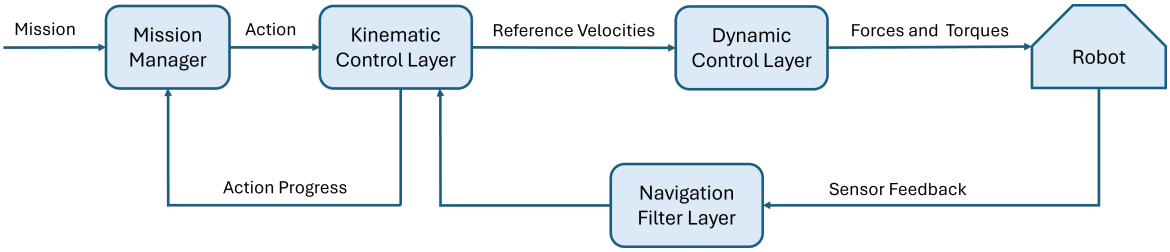


Figure 2.3 Control System Architecture of ASV

decision-making system, which sends commands to the control subsystem in a way that drives the actuators to produce the desired vessel behavior.

For the thesis project, the control framework of the ASV system is illustrated in the block diagram shown in Figure 2.3. The Mission Manager decides which actions should be executed at a given time. Subsequently, the Kinematic Control Layer (KCL) receives the selected action and computes all reference velocities using the vehicle's kinematic model. This is where the Task-Priority method is applied. While, the Dynamic Control Layer (DCL) receives the system reference velocities from the KCL and computes the torques and forces required to generate the commanded actuator velocities in the vehicle.

2.3 Task Priority Control

In this section, Task Priority Approach is reviewed based on Simetti et al. (2019).

2.3.1 Basic Definitions

The two key variables in defining the Task Priority algorithm are:

- the system configuration vector $\boldsymbol{c} \in \mathbb{R}^n$, which represents the degrees of freedom of the robot (e.g. joint positions for a manipulator, position, and orientation for a moving base, etc.);
- the system velocity vector $\dot{\boldsymbol{y}} \in \mathbb{R}^n$, which represents how the system can be actuated (e.g. joint velocities for a manipulator).

2.3.2 Control Objectives

Consider a scalar variable $x_o(\boldsymbol{c})$ related to a control objective o, then:

- the requirement for $x_o(\boldsymbol{c}) = x_{o,0}$, that is called an equality control objective,
- the requirement for $x_o(\boldsymbol{c}) \geq x_{o,m}$, or $x_o(\boldsymbol{c}) \leq x_{o,M}$ is called an inequality control objective,

where $x_{o,0}$ is a given reference value, and the m and M subscripts indicate a minimum and maximum value respectively.

2.3.3 Control Actions

The mission is built by a sequence of actions, and each action is composed of priority-ordered *tasks* to be simultaneously managed. There are different levels of control objectives according to their priorities:

- objectives related to physical constraints or safety measurements,
- objectives related to prerequisite of the main action,
- an action-defining objective,
- an optimization objective.

2.3.4 Activation Functions

For every control objective (and its corresponding reactive control task), there is an associated activation function called the Task Activation Function. The role of the Activation Function is to specify whenever the reactive control task associated to a control objective is relevant for the system.

When the task activation value is set to one, the associated task is *active*, and when the task activation value is set to zero the task is *inactive*. Otherwise, when the task activation value is between zero and one, the task is in the *transition zone* as shown in Fig. 2.4. For instance, assuming an enquility objective of the type $x(\boldsymbol{c}) \leq x_M$:

$$a^i(x) \triangleq \begin{cases} 0, & x(\boldsymbol{c}) < x_M - \Delta \\ s(x), & x_M - \Delta \leq x(\boldsymbol{c}) \leq x_M \\ 1, & x_M < x(\boldsymbol{c}) \end{cases}, \tag{2.11}$$

where $s(x)$ is any smooth, strictly decreasing function joining the two extrema, and Δ is a value that creates a buffer zone, where the inequality is already satisfied, but the activation value is still greater than zero, preventing chattering problems around the inequality control objective threshold x_M. Finally, for reactive control task associated to an equality control objective and nonreactive control tasks, the activation function is always set to one, $a^i(x) \equiv 1$.

The activation value is computed by combining two separate activation functions: internal and external. The internal one is computed only depending on the task's variable. The latter instead gives the possibility to deactivate tasks depending on external factors. Once all the control tasks have been identified, their priority must be established. The task priority control allows for establishing a strict priority order between the control tasks and their associated objectives. In this way, the lower priority tasks should not influence the achievement of higher priority ones. In other words, the innovative point of this control architecture is the ability of handling non-action defining objectives without over-constraining the system. For instance, to maintain a joint away from its end maximum or minimum limit, the associated control task needs to be activated only in the vicinity of the limit itself.

2.3.5 Task priority inverse kinematics

The control task is then defined as tracking a given feedback reference rate $\dot{\tilde{x}}$, which enables the associated scalar variable to be driven toward its corresponding objective. The link between the system velocity vector $\dot{\boldsymbol{y}}$ and the considered control objective x is given by the

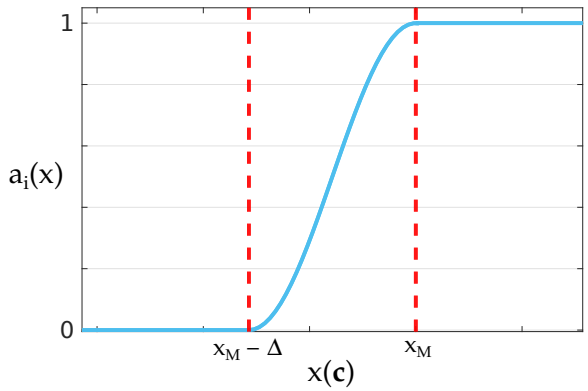


Figure 2.4 Task Activation Value (TAV): The phase between $x_M - \Delta$ and x_M is the *transition zone*. The activation function starts activating at $x_M - \Delta$, and it is fully activated at x_M.

Jacobian relationship:

$$\dot{\boldsymbol{x}} = J(\boldsymbol{c})\dot{\boldsymbol{y}}, \tag{2.12}$$

Since such a relationship exists between each of the control objectives and the system velocity vector $\dot{\boldsymbol{y}}$, it is clear that a procedure is required to determine the *best* value of $\dot{\boldsymbol{y}}$. This procedure needs to take into account the *priority* of the control objectives and their related tasks.

Solving the control problem locally using the Task Priority Inverse Kinematic (TPIK), the solution for a high-priority task defines a *null space*, within which lower-priority tasks can be executed without affecting the satisfaction of higher-priority objectives. Specifically, if J_1 denotes the Jacobian of a high-priority task, the control input for a lower-priority task can be computed as

$$\mathbf{u}_2 = \mathbf{u}_{2,0} + (I - J_1^{\dagger} J_1)\mathbf{u}_{2,0}, \tag{2.13}$$

where $J_1^{\dagger}$ is the pseudo inverse of J_1, and $\mathbf{u}_{2,0}$ is the unconstrained control input for the second task. This procedure can be iteratively applied to any number of tasks without over-constraining the system, creating a strictly prioritized, modular, and scalable multi-objective control framework.

2.3.6 Practical Implementation

To clarify the implementation of Task Priority, we can take an example of an *ASV Path Following* mission. This mission consists of two successive actions; 'Move to Starting Point', then 'Path Following' as shown in Figure 2.5. When the mission is complete, the state is switched to *Hold*, which is a new mission, and it is composed of an action called 'Hold'.

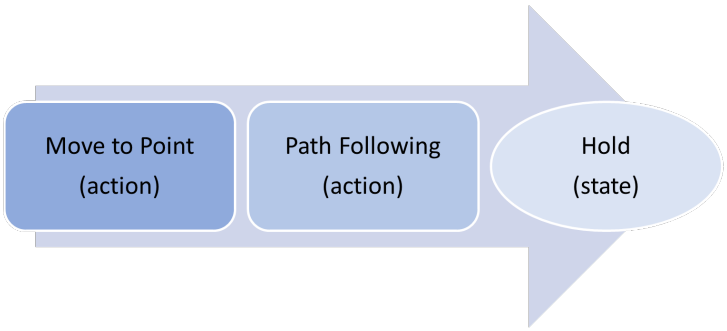


Figure 2.5 *ASV Path Following* mission is composed of the two successive actions: moving to the starting point of the path, then following the path until the end. After arriving to the end of the path the state is switched to *Hold*, which is a new mission.

Each action consists of a set of tasks that must be concurrently executed during the running action. For instance, 'Path Following' action has the "Cartesian Distance" task and the "Align to Target" task, see Figure 2.6b. Because the path following algorithm defines a moving point on the path as a target for the vehicle. Therefore, the vehicle has to follow that target by aligning towards that target and converging to it. For safety measurements we defined two other tasks with higher priority; "Safety Boundry" and "Absolute Axis Alignment Safety" tasks for avoiding the the border of the safety area. Table 2.2 describes the tasks implemented in this context.

As it was mentioned before, the *ASV Path Following* is composed of two actions; 'Move to Point', and 'Path Following'. Hence, in the unified tasks list, there are all the implemented list of tasks. As we can see in Figure 2.7, the tasks are listed from the highest priority in the up to the lowest at the bottom. And each colour represents a priority level. That is, the tasks that have the same colour have the same priority level.

Technically speaking, the tasks are activated and deactivated according to the action sequence by the external activation functions. They are graphically illustrated by the sign of stop in Figure 2.8a. On the other hand, the activation of internal activation functions are represented by the green disk. The Figure 2.8a illustrates a particular case when the vehicle starts executing the first action 'Move to Starting Point'. When the first action is comoplete, the next action starts Figure 2.8b. As we can observe from Figure 2.8a and 2.8b, all the internal activation functions are activated except the safety tasks. The internal activation function of those tasks are activated only in particular case when the ASV is closed to the border of the safety region, see Figure 2.8c.

For more details on the TPIK approach, readers can refer to Simetti and Casalino (2016) and Simetti et al. (2018) in particular for underwater applications.

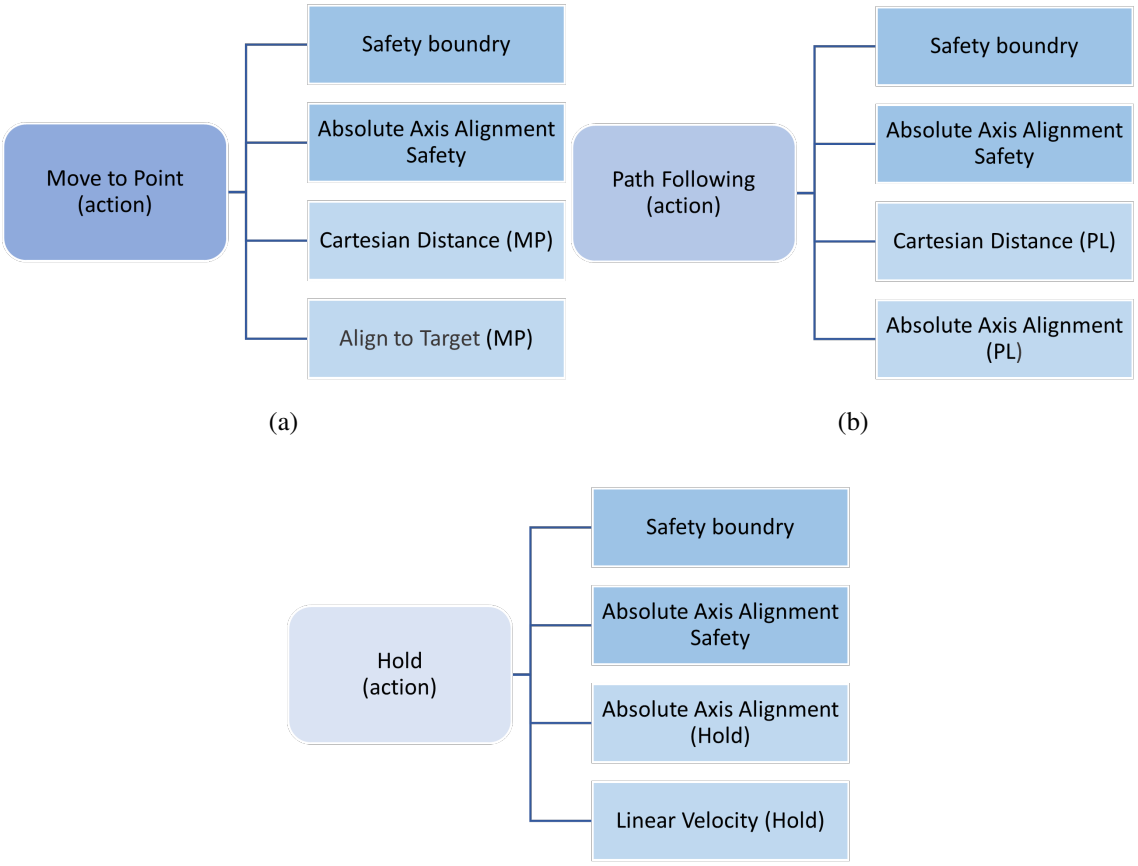
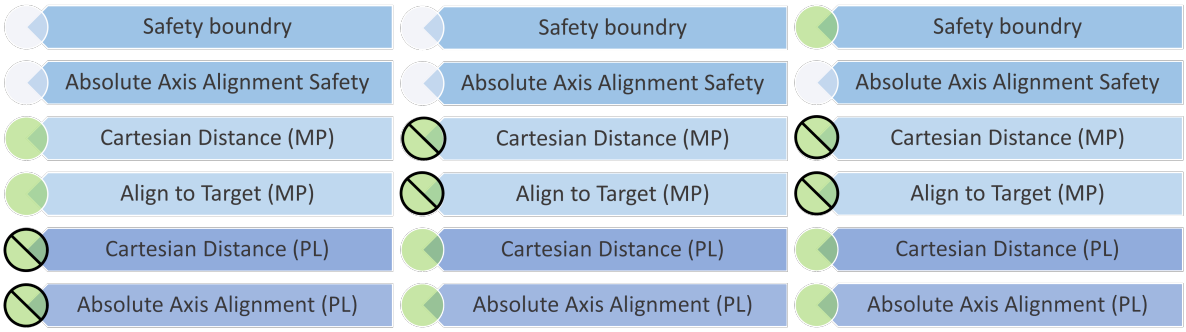


Figure 2.6 Examples of the structure of an action: **(a)** Move to Point (MP) action. **(b)** Path following (PL) action **(c)** Hold action.

Figure 2.7 Unified Task list: Each color represents a priority level, ordered from highest (up) to the lowest (below). Each task has a circle that represents its activation.

	Task	Explanation
1	Safety Boundry	A task that implements the repulsion mechanic from the boundaries.
2	Absolute Axis Alignment	A task that implements the absolute axis alignment of a given frame, (e.g. forcing one vehicle axis to be aligned to a certain axis fixed in the absolute frame).
3	Align to Target	A task that implements the alignment of a given frame towards a relative axis task, (e.g. aligning one vehicle axis towards a target that changes with time).
4	Cartesian Distance	A task that implements the linear position for a given frame.
5	Linear Velocity	A task that implements the linear velocity for a given frame.

Table 2.2 Task definition.



(a) (b) (c)

Figure 2.8 Unified Task list: the green circle represents the activated tasks in each phase. (a) The vehicle is on the first phase applying the first action; 'Move to Starting Point'. (b) The vehicle is on the second phase applying the second action; 'Path Following'. (c) The vehicle is on the second phase applying the 'Path Following' action, and close to the border.

Chapter 3

Mechatronic Design

In this chapter, we introduce the system from a mechanical perspective. Starting with an overview of the entire system, we discuss the vehicles and mechanical components in each section. In particular, the chapter includes the ASV, the ROV, the Tether Management System (TMS), and the Launch and Recovery System (LARS). Also, the sensors required to implement the control policy are described at the end of the chapter.

3.1 System Description

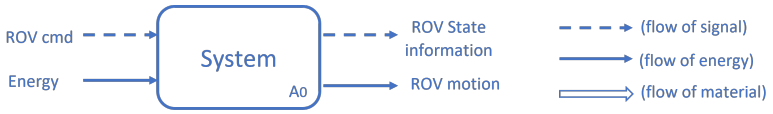
Describing the system by a simple input-output diagram, it can be represented by a diagram as shown in Figure 3.1. Assuming a user is piloting the ROV remotely from a control station, the ROV should follow the commands, navigate through the water, and provide real-time feedback to the user, including video streaming and sensor readings.

To accomplish this objective, an ASV is required to transport the ROV to the mission area, deploy it, and provide a suitable cable length that transmits and receives all commands and feedback between the ROV and the control station. Hence, if Block A0 in Figure 3.1 is decomposed, the system can be represented as shown in Figure 3.2. From this system architecture, the ASV is evident as it serves as a bridge to the ROV.

Additionally, a block exists between the ASV and the ROV, which corresponds to the Launch and Recovery System (LARS) and Tether Management System (TMS) block. The LARS is responsible for deploying and recovering the ROV from the water, while the TMS regulates the release and retraction of the tether. As shown in Figure 3.2, the LARS and TMS block is powered by the ASV's power supply. Moreover, the ASV transmits commands to reel/unreel the cable or deploy/recover the ROV, and it receives feedback on the released cable length from the TMS.

The Interface Block represents the Graphical User Interface (GUI) used by the control station operator, which transmits commands to the ASV. The data flow begins wirelessly from the Interface to the ASV, continues via the cable to the TMS and LARS, and finally through the tether to the ROV. The feedback follows the same path in reverse, from the ROV back to the control station.

In the following section, the ASV, ROV, and LARS/TMS blocks are discussed. The Interface block is covered later in Section 5.1.1.



(a) (b)

Figure 3.1 System Design Diagram: **(a)** Simplified System Block *A*0: the basic Input-Output flow. **(b)** Description of Arrows.

3.2 The ULISSE ASV

The ASV in our cooperative ASV-ROV system studied in this thesis is the ULISSE catamaran. The ULISSE ASV is a 3 m-long and 1.8 m-wide catamaran constructed from fiberglass. It represents the second generation of catamarans developed by the Interuniversity Research Center on Integrated Systems for the Marine Environment (ISME) at the University of Genova, and was used for executing autonomous geotechnical surveys in the H2020 WiMUST project. The two hulls have identical geometry and are rigidly connected by two carbon fiber bars passing beneath the deck line. The catamaran is designed to transport payloads up to 200 kg.

Each hull contains a compartment that holds the batteries (approximately 3.2 kWh per hull) and the control electronics, including inertial sensors (compass, accelerometers, and gyroscopes). A Global Navigation Satellite System (GNSS) antenna is mounted on top of the roll bar. Propulsion is provided by two Torqeedo Cruise 2R electric propeller motors, each rated at 2 kW with approximately 50% propulsive efficiency. The two motors are identical and rotate clockwise for positive RPMs and counterclockwise for negative RPMs.

The vehicle is controlled by a *PCM* – 3362 single-board computer, equipped with a 1.67 GHz Intel® Atom™ N450 processor. While the computational power is moderate, it is sufficient to run path-following and low-level control algorithms required for the WiMUST survey. The vehicle has been designed for maximum reconfigurability, allowing custom decks

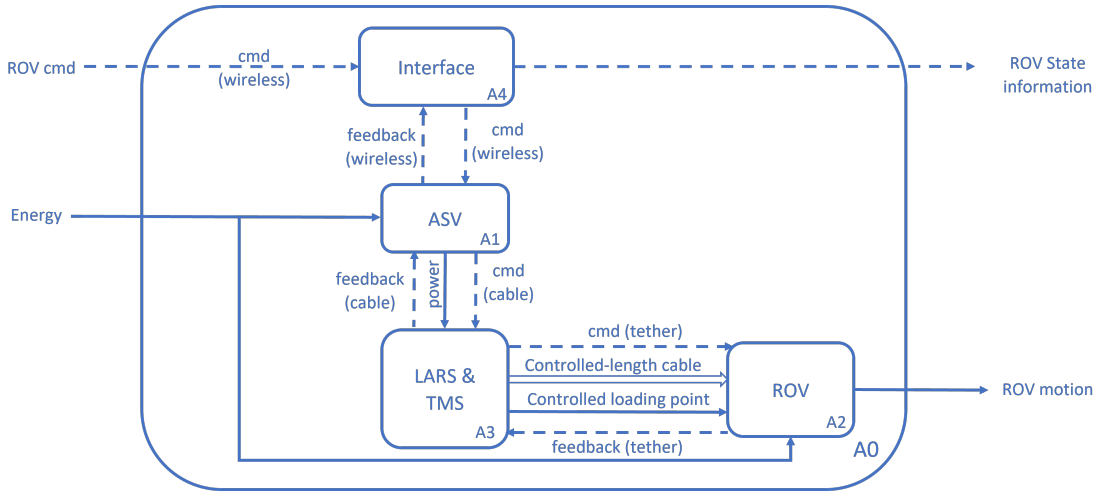


Figure 3.2 Decomposition of the Main System Block *A*0.

to be mounted to host mission-specific payloads. The catamaran is capable of autonomous navigation and utilizes information from onboard sensors, such as GPS, IMU, and compass, as well as data exchanged with other network nodes via Wi-Fi for surface nodes and acoustic modems for underwater nodes. The motion control software of the ULISSE catamaran has been developed within the Robot Operating System (ROS) framework Antonelli et al. (2018).

Based on the thruster model discussed in Section 2.1.2, it is important to note that this type of vessel exhibits nonlinear behavior due to several factors. As reported in Simetti and Indiveri (2022), twin-screw vessels such as ULISSE experience a lateral force generated by the differential pressure between the two hulls. Furthermore, a lateral force, known as the "paddle wheel effect," is generated because ULISSE's propellers are not counter-rotating. These effects collectively result in asymmetric turning radii for the ASV. Consequently, the ULISSE ASV model used in this thesis was adopted from Simetti and Indiveri (2022), where all aforementioned nonlinear effects were considered, and the system parameters were experimentally identified.

3.3 BlueROV2

The BlueROV2 is one of the most widely used ROVs in academic and research projects, as it combines affordability, flexibility, and open-source support, making it ideal for prototyping, experimentation, and data collection in underwater environments. One notable feature of the BlueROV2 is its open-source and customizable design, which makes it particularly suitable for academic robotics research and the development of advanced control, autonomy, or AI-

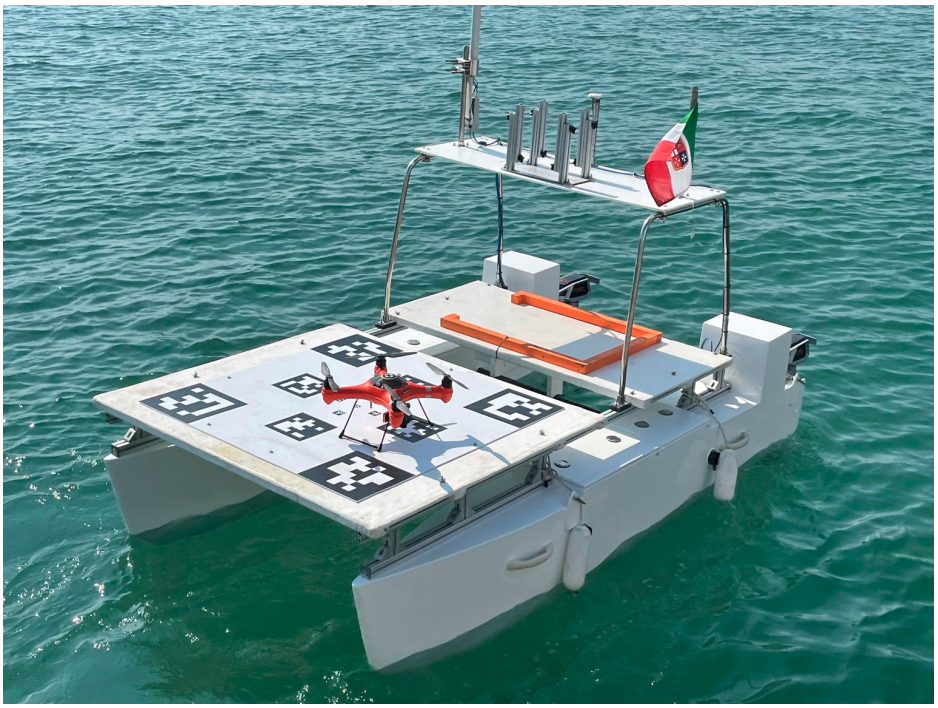


Figure 3.3 Ulisse catamaran.

based underwater systems. Various sensors (e.g., sonar, DVL, water quality probes, grippers, hydrophones) can be added or replaced easily via standard interfaces. The BlueROV2 used in this project is the Heavy Configuration, which is equipped with eight thrusters, enabling full six-degree-of-freedom control and enhanced stability even in moderate currents.

The tether used for the BlueROV2 is the Fathom Tether, a high-quality cable designed specifically for ROVs and other subsea applications. According to the official site of the product, it is neutrally buoyant, has a breaking strength of 660 N, and is embedded with water-blocking fibers to seal any leaks. Its diameter is 7.6 mm. For detailed physical specifications, the reader can refer to the tether cable's technical information page[1].

For the simulation software, the physical and geometrical parameters of the BlueROV2 are taken from its datasheets. Hydrodynamic coefficients of the ROV, including the matrices $M, C(\boldsymbol{v}_r), D(\boldsymbol{v}_r), T, K$, as well as the tether material properties, were taken from the work published in Zand (2009).

[1] https://bluerobotics.com/store/cables-connectors/cables/fathom-rov-tether-rov-ready/

(a) (b)

Figure 3.4 ROV component taken from Blue Robitics: **(a)** BlueROV2 Heavy Configuration. **(b)** Fathom Tether. Image credit: BlueRobotics

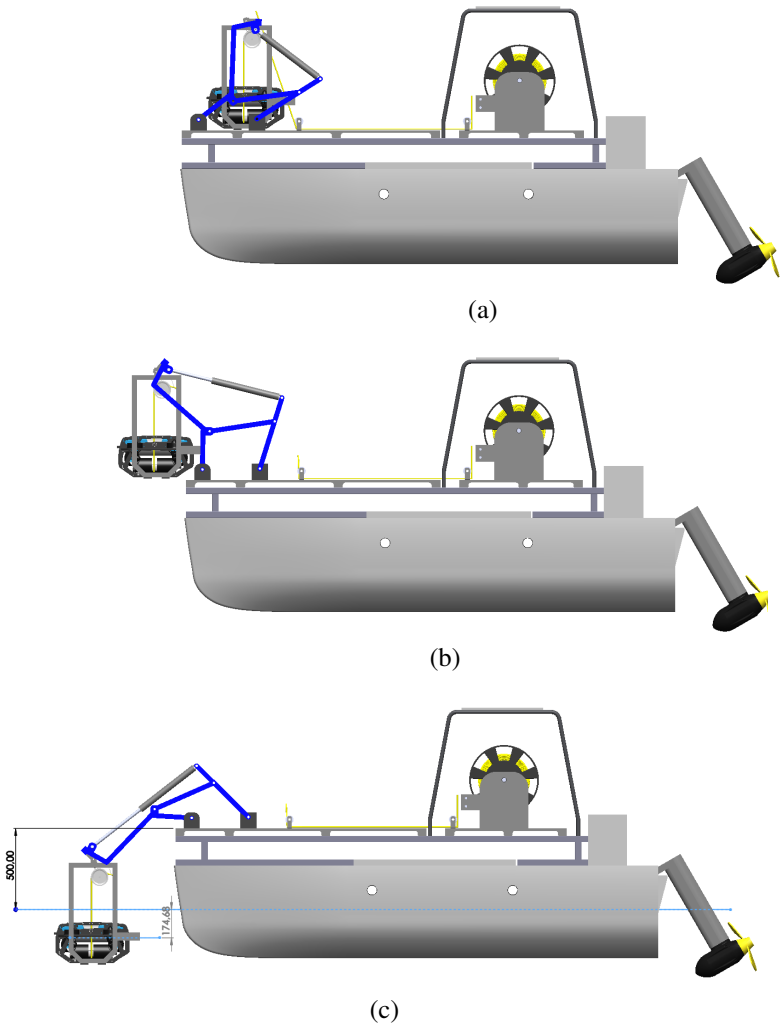
3.4 Tether Management System and Launch And Recovery System

The Tether Management System (TMS) follows the general design of a motor-powered cable reel. Accordingly, it is motorized using a DC motor equipped with a gearbox. Since the cable is relatively long (up to 300 m), a synchronized level-wind mechanism is incorporated to arrange the cable neatly on the drum. The level winder is a mechanical component that automatically moves from side to side, guiding the cable evenly onto the spool. This is crucial for preventing tangles and maximizing the amount of cable that can be stored.

Regarding the Launch and Recovery System (LARS), the specific design adapted for the ULISSE catamaran is illustrated in Figures 3.5 and 3.6. This LARS is composed of a horizontal guide that holds a carriage; in which the ROV is placed, The vertical parts of the LARS is composed of four passive arms and one linear actuator on the left side. However, the right side is not activated and has one passive arm. This system aims to lift the ROV from the deck and deploy it in the water. Thanks to its four-arm design (for the left side of the vertical part), the carriage can reach under the water level using only one actuator. Hence, the ROV can be deployed in water without placing stress on the TMS. A tension sensor,

discussed in Section 3.5.1, is mounted on the moving frame of the system. Both the TMS and LARS are powered by the ASV's batteries (24 V).

We collaborated in the design of these TMS and LARS with GRAALtech[2] company, which is responsible for the manufacturing.



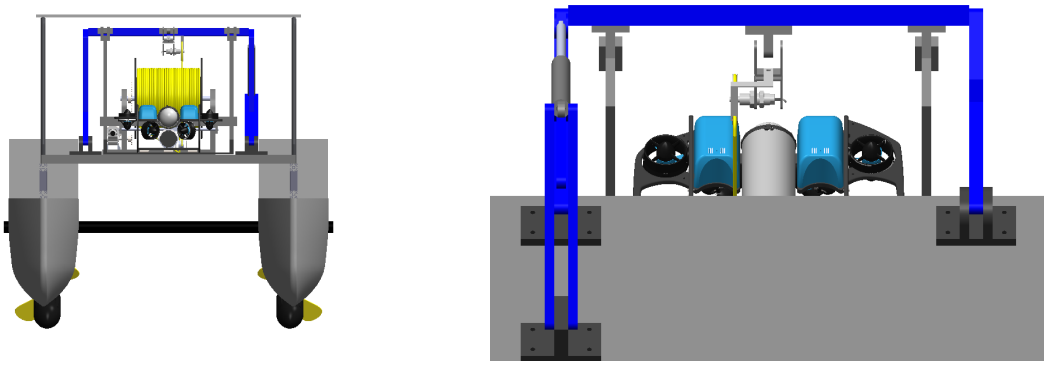
(a)

(b)

(c)

Figure 3.5 TMS and LARS mounted on ULISSE ASV (side view): **(a)** Initial position of the ROV on the deck.**(b)** Starting to deploy the ROV in water. **(c)** Fulley deployed ROV in water. The *light blue* line is the level of water.

[2]https://www.graaltech.com/



(a) (b)

Figure 3.6 TMS and LARS on ULISSE ASV: **(a)** Front view. **(b)** Back view (zoomed). The tension sensor is mounted on the horizontal guide.

3.5 Additional Sensors

A number of sensors were added on the system; in particular a tension sensor in the LARSfor acquiring the tension of the ROV tether, and an underwater localization system in order to acquire the position of the ROV, which is a fundamental requirement for the ASV motion control.

3.5.1 Tension Sensor

For collecting tension data, the radial force sensor[3] is employed. The sensor features two orthogonal measurement axes and a splash-proof surface suitable for wet or dusty environments (Figure 3.7a). This specific tension sensor enables estimation of the resultant tension force vector F_R without requiring knowledge of the exact wrapping angle α, by measuring the two radial components F_x and F_y. The relation

$$F_R = \sqrt{F_x^2 + F_y^2} = 2F_z \sin\left(\frac{\alpha}{2}\right)$$

is taken directly from the sensor datasheet.

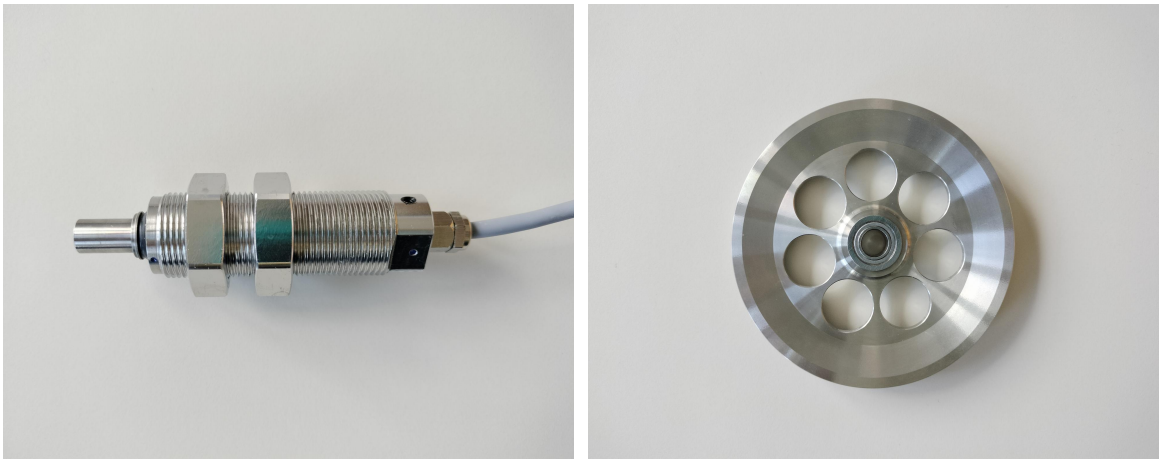
This tension sensor requires the following accessories to obtain accurate force measurements:

- Pulley: A lightweight stainless-steel U-groove pulley was used in conjunction with the sensor, specifically designed to be compatible with the RFS® 150 XY. The pulley

[3] https://honigmann.com/k7/g75/i456/Tension-sensors-ALL-2-axis-radial-force-sensor-RFS150-XY.html

is required to measure the tension of the cable. As shown in Figure 3.7b, this pulley features a low-mass construction with lightening holes, and has an outer diameter of 108 mm, a root diameter of 95 mm, and a groove width of 12 mm. It is optimized for cables with a radius of 4 mm and has a bearing journal with a diameter of 10 mm (see Figure 3.7b).

- Tensiotron® TS 621: A conditioning circuit used to process the raw signals from the sensor and output amplified and filtered data. For more information about the board, the reader can refer to the link[4].



(a) (b)

Figure 3.7 Tension Sensor Components: **(a)** Sensor RFS® 150 XY. **(b)** Pulley.

3.5.2 Underwater Localization System

Generally, the ASV requires the position of the ROV to track it. We selected the Underwater GPS G2 Standard Kit from WaterLinked. It is a short baseline (SBL) system and consists of an array of transceivers (Figure 3.9a) mounted on a pole beneath the surface vessel, and a transponder installed on the ROV (Figure 3.9b). A topside unit calculates the position from the ranges and bearings measured by the transceiver.

If the target position is required in Earth coordinates such as latitude/longitude or UTM, the SBL positioning system is combined with a GPS receiver and an electronic compass, both mounted on the vessel. By combining the vessel's location and orientation with the

[4]https://www.honigmann.com/page.php?m_pro=14&lang=2

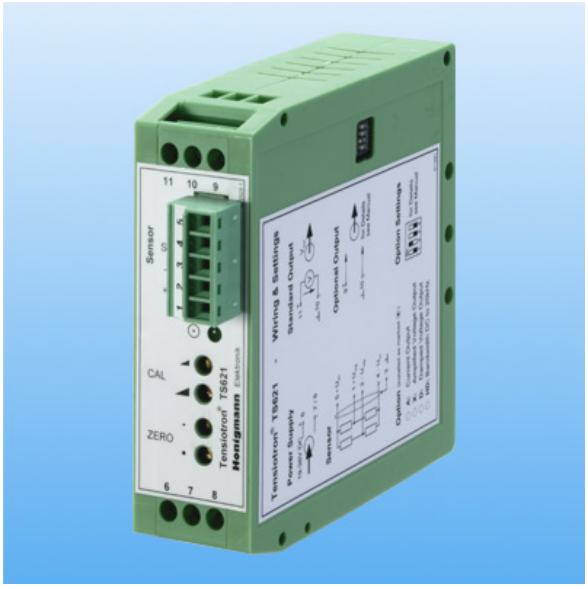


Figure 3.8 Tensiotron® TS 621: An electronic precision measuring amplifier used for tension measurement. Image credit: Honigmann.

relative position data from the SBL system, the position of the tracked target — in this case, the ROV — is determined in Earth coordinates.

(a) (b)

Figure 3.9 Accoustic Positioning System: **(a)** Antenna: three transducers. **(b)** LocatorU1: a transponder. Image credit: WaterLinked.

Chapter 4

Control Policy

In this chapter, a control strategy to minimize the umbilical tension in the ASV-ROV system is presented. The primary factor is to track the changes in the relative distance between the ASV and the ROV. The discussion can be divided into two main aspects. The first focuses on the horizontal distance, and the second addresses the vertical distance. When the ROV changes its horizontal distance relative to the ASV, the latter should follow the ROV rather than release additional cable into the water, as discussed in Zhao et al. (2022), where the "static ASV" strategy was compared to the "following ASV" strategy. According to the numerical results, the latter approach significantly reduces umbilical tension. Conversely, when the ROV moves vertically, the TMS should release more cable as the ROV descends and retract cable as the ROV ascends. In this scenario, the ASV's position does not influence the cable length management strategy. The following sections of this chapter present a detailed analysis of ASV motion control and subsequently address the application and adaptation of the task-priority approach to cooperative ASV–ROV systems for tethered objectives. With respect to the tether management, the discussion begins with the motivation for regulating the tether length, followed by an overview of the preferred tether shape based on the ROV's operational scenarios. Finally, the chapter concludes with the enhanced reference cable length equation, grounded in the state-of-the-art literature.

4.1 ASV motion control

To automate the execution of underwater missions performed by the ROV during the inspection phase, the ASV should follow the ROV autonomously and act as a bridge by tracking it, transmitting commands, and releasing the appropriate length of cable. In this configuration,

the ROV leads the motion, and the ASV follows from behind, considering the following criteria:

- Power Consumption: Both the ASV and ROV are powered by onboard batteries, so their energy sources are limited. It is therefore essential to use energy efficiently by minimizing unnecessary movements. Moreover, a long, extended cable generates significant drag underwater, which increases the energy required to compensate for this drag.
- Wear and Stress: The ROV generally requires a sufficiently long tether released in the water to allow unrestricted motion. Otherwise, if the cable length is shorter than the distance between the ASV and ROV, the thrusters may operate without effective motion, leading to both energy consumption and additional stress on the tether.
- Obstacle Avoidance: During offshore wind farm inspection missions, the ASV control system should account for pillar-like obstacles. In this scenario, the cable cannot pass over the obstacles, so the ASV should maintain a safe distance (far from the obstacles) and orient itself appropriately (align towards both the ROV and the obstalce) to prevent cable entanglement while following the ROV.

In the following sections, the ROV following and obstacle avoidance control policies are described in detail, while the final section addresses the implementation of the Task Priority method.

4.1.1 ROV Following

When the ROV navigates from one location to another, the most efficient approach is to allow the ASV to follow, enabling the ROV to move freely without cable constraints. However, when the ROV performs tasks underwater, it may maneuver within a relatively limited range around the ASV. In such cases, the ASV should maintain its position. The area surrounding the ASV is defined as the “Neighborhood Zone”, which is the volume of a cylinder whose axis is centered on the ASV’s position, as illustrated in Figure 4.1.

Once the ROV moves outside the “Neighborhood Zone”, the ASV should align and approach the ROV until it returns within the zone. If the alignment error between the ASV and ROV is large (i.e., the ROV is outside the blue slice shown in Figure 4.1), the ASV should rotate towards the ROV to ensure that the cable passes over the front side of the ASV. In this context, the alignment error threshold should not be excessively small; otherwise, the ASV may become overly sensitive and oscillate in an attempt to maintain precise alignment.

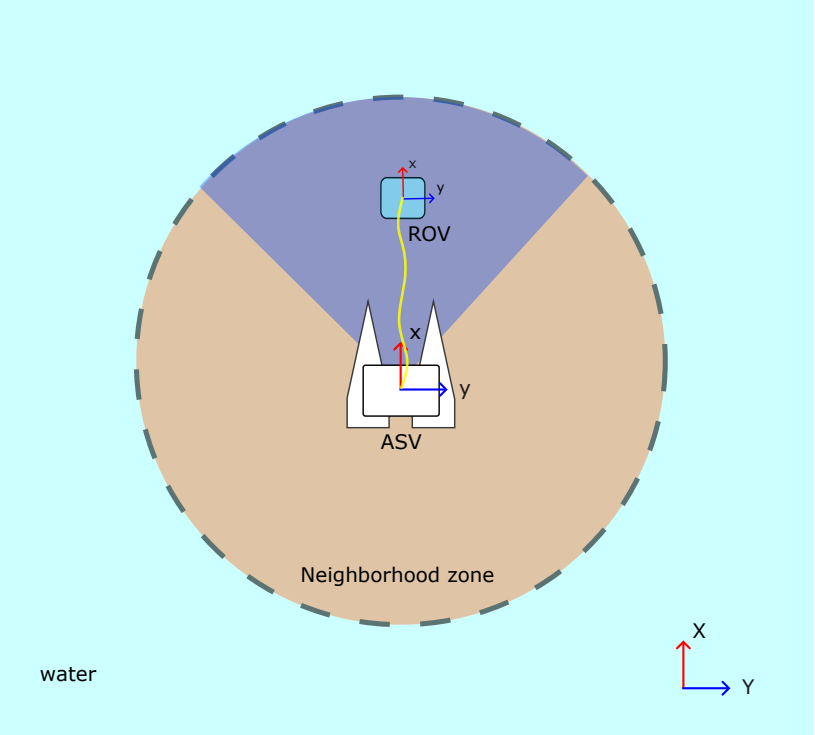


Figure 4.1 Neighbourhood Zone: While the ROV is moving inside the "Neighborhood Zone" (the *light brown* disk), the ASV does not move nor align towards the ROV. The *blue* slice represents the area where the ROV navigates without having the ASV aligned towards the ROV.

4.1.2 Obstacle Avoidance

Regarding the obstacle avoidance scenario, the ASV must avoid obstacles like any mobile vehicle by generating a velocity vector opposite to the obstacle. In addition, for this type of tethered ASV-ROV configuration, the ASV must simultaneously align with both the ROV and the obstacle, maintaining a sufficient distance from the ROV to prevent the cable from becoming entangled around pillar-like obstacles. Figure 4.2 illustrates the ASV control criteria depending on the ROV's relative position with respect to the obstacle and the ASV.

When the ROV is inside the "Neighborhood Zone" (the *light brown* circle in Figure 4.2a), the ASV should hold its position. If the ROV enters the "Obstacle Zone" (the *green* circle surrounding the obstacle, Figure 4.2b), the ASV should align with the alignment point, indicated by the red flag in Figure 4.2c. The alignment point represents the ASV's target position, located along the line connecting the centers of both the obstacle and the ROV (*dashed green* line) keeping a predefined safe distance from the ROV. After aligning towards the alignment point, the ASV should move to that position and then re-align towards the ROV,

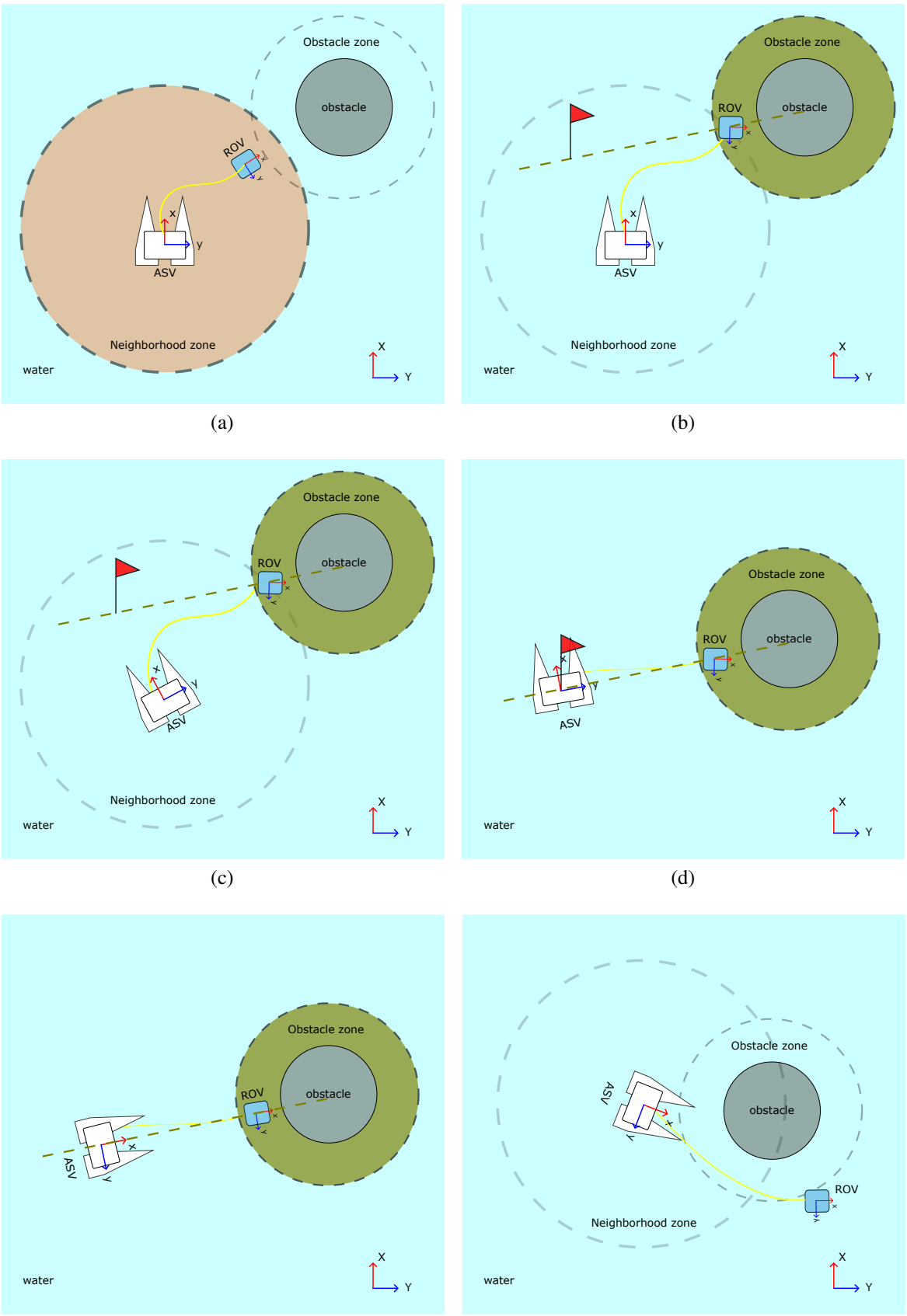


Figure 4.2 Obstacle Avoidance control strategies.

as shown in Figures 4.2d and 4.2e. Once the ROV exits the “Obstacle Zone”, the ASV should follow the ROV, as illustrated in Figure 4.2f, until the ROV returns to the “Neighborhood Zone”, as shown in Figure 4.2a.

According to Schwanz et al. (2012), the minimum distance between wind turbines is five times the diameter of the turbine blades. Considering that the smallest turbine blade diameter can reach 50 m, the spacing between turbines is sufficiently large to allow the ASV to consider one pillar-like obstacle at a time. In other words, the ASV can detect and respond to one obstacle at a time.

4.1.3 Task Priority Adaptation

The task-priority framework, originally developed for basic navigation and station-keeping on the ULISSE ASV, is extended in this thesis to support coordinated ASV–ROV motion. Before applying Task Priority in the new kinematic control layer of the ASV system, the previous criteria should be converted into mission, actions, and tasks.

Defining Missions, Actions, and Prioritized Tasks

During the inspection process, the ASV initiates the mission by holding the ROV and transporting it toward the inspection region, then launches the ROV into the water. Subsequently, it tracks the ROV for cooperative navigation while the ROV performs underwater inspection. Upon completion of the inspection, the ROV is recovered from the water, and finally, the ASV returns to the base station.

In this thesis, we consider the *Inspection* as the main mission that the ASV should be accomplished, and the corresponding control actions of this mission are defined as the following: ‘Go to Starting Point’, ‘Releasing the ROV’, ‘Cooperative Navigation with ROV’, ‘Recovering the ROV’, and ‘Go Back to Starting Point’ to return to the base station as demonstrated in the diagram in Figure 4.3. In this thesis work, we focus on the ‘Cooperative Navigation with ROV’ action (Figure 4.4). The task “Absolute Axis Alignment (Alignment Point)” is to keep the ASV x axis pointing towards the ROV-Obstacle axis point (which is illustrated as the Red Flag in Fig. 4.2d), while the task “Cartesian Distance (Alignment Point)” is to move and reach the alignment point of the obstacle. On the other hand, “Absolute Axis Alignment (ROV)” is for keeping the ASV x axis pointing to the ROV, and finally task “Cartesian Distance (ROV)” is to keep the distance between the ASV and the ROV under a specific threshold.

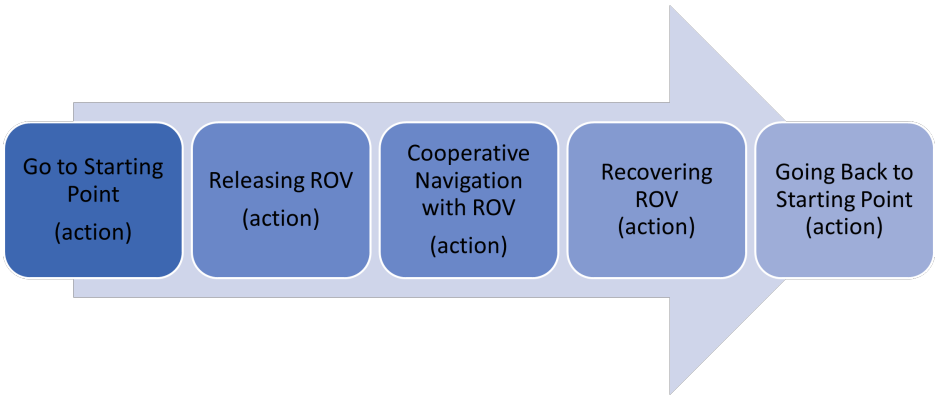


Figure 4.3 The mission *Inspection*.

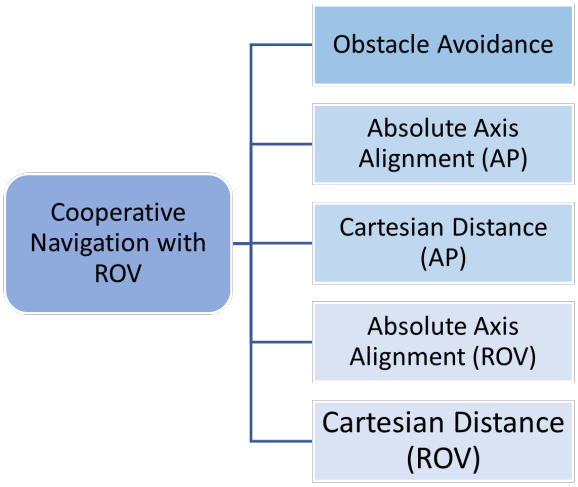


Figure 4.4 The action 'Cooperative Navigation with ROV'.

The tasks are listed in Table 4.1 from the highest priority to the lowest priority. During the execution of the action 'Cooperative Navigation with ROV', these tasks are not activated all the time. Table 4.1 shows the activated tasks according to each case in Fig. 4.2. For instance, the task "Obstacle Avoidance" is activated only in case (f) when the ASV is close to the pillar, while the task "Align to ROV" is activated in case (d); after reaching the Red Flag to align to the ROV and the obstacle, and in case (f); where the ASV should follow the ROV. As mentioned in section 2.3, the activation of the tasks are accomplished by the Task Activation Functions.

Detailed Motion

Examining the motion of the ASV in more detail, the following control design requirements for the ASV-ROV configuration are identified:

Table 4.1 Activated Task

	Cases*					
Task	(a)	(b)	(c)	(d)	(e)	(f)
Obstacle Avoidance						X
Align to Red Flag		X				
Move to Red Flag			X			
Align to ROV				X		X
Move to ROV						X

*Cases from (a) to (f) are illustrated in

- The ASV should begin aligning with the ROV before the ROV reaches the edge of the "Neighborhood Zone", ensuring that once the ROV exits the "Neighborhood Zone", the ASV is already oriented toward it.
- The ASV should move toward the ROV only when the heading error is sufficiently small, since the Ulisse ASV is a twin-thruster vehicle and cannot generate sway motion.
- The ASV should move toward the ROV once the ROV fully exits the "Neighborhood Zone".
- The ASV should proceed to the alignment point when the ROV is inside the "Obstacle Zone" and the ASV is already oriented toward the ROV.

These specifications are achieved by appropriately setting the parameters of the Task Activation Functions. As described in Section 2.3, the parameters ξ_1 and ξ_2 determine when the activation function begins and reaches full activation. For example, for the External Activation Function of the "Align to ROV" task, ξ_1 and ξ_2 should satisfy $\xi_1 < \xi_2 < r_n$ (where r_n is the radius of the "Neighborhood Zone"). The exact values of ξ_1 and ξ_2 are determined based on the use-case specifications or by trial and error.

Moreover, thanks to the activation function, not only the transition of the activation function can be set, but also the activation of the tasks can be dependent on real-time variables. For instance, the External Activation Function of the "Move to ROV" task is multiplied by the heading error of the "Align to ROV" task to ensure that it is activated only when the heading error is sufficiently small. Similarly, the tasks "Move to ROV" and "Align to Red Flag" are influenced by external factors. Table 4.2 summarizes the activation function parameters for each task.

Table 4.2 External Activation Functions

Task	External Activation Function Variable
Align to Red Flag	ROV-Obstacle distance ASV-Red Flag distance
Move to Red Flag	ROV-Obstacle distance ASV-Red Flag heading error
Align to ROV	ROV-Obstacle distance ASV-ROV distance
Move to ROV	ASV-ROV heading error

4.2 Desired ROV Tether Length

One of the important aspects to be studied in the ASV-ROV cooperative system is the length of the tether in every moment. This section covers the static and kinetic studies, and the preferred cable shape. When the ASV and the ROV are in the static case, the basic consideration to be taken into account is to make the cable length longer than the distance between the two vehicles, i.e.,

$$l_{ref} > d, \tag{4.1}$$

where l_{ref} is the reference cable length to be released in the water and d is the distance between the ASV and the ROV. This reduces both the power consumption and the wear and stress on the umbilical over time. However, when the ROV, ASV, or both vehicles are in motion, a force is applied to the cable due to hydrodynamic drag. The drag force is expressed by the following equation:

$$F_D = \frac{1}{2}\rho v^2 C_D A \tag{4.2}$$

where

- F_D is the drag force;
- ρ is the density of the fluid;
- v is speed of the object with relative to the fluid;
- C_D is the drag coefficient;
- A is the cross-sectional area.

To estimate the order of magnitude of the force in our scenario, we considered the worst-case situation, in which the ROV is positioned directly beneath the ASV and both vehicles are

Table 4.3 Drag Force on the underwater cable.

v (m/s)	0.5	0.5	1.0	1.0	1.5	1.5
l_{cable} (m)	100	300	100	300	100	300
F_D (N)	115	340	445	1.3k	1k	3k

moving at a maximum speed of 1.5 m/s. In this situation, the cable can be modeled as a vertical cylinder moving perpendicular to its axis.

Since A represents the orthographic projection of the object, the frontal area projected onto a plane perpendicular to the direction of motion can be approximated as the area of a rectangle, calculated as the product of the cable's length and diameter. Regarding the drag coefficient C_D, for an unfaired cable, it typically ranges around 1.2, as noted in Molland (2008). Table 4.3 summarizes the drag calculations based on these assumptions.

The results indicate that, in the worst-case scenario with a velocity $v = 1$ m/s and a cable length $l_{cable} = 100$ m, the drag force can reach approximately 1 kN, which is relatively high.

Generally, when ROV is deployed in water, the practical estimation for the required cable length is to take the double of the distance between the ROV and deployment point. However, based on the worst case calculation of the drag force as we did in Table 4.3, we can notice that doubling the length the cable will increase the drag drastically. Hence, the tether length must be minimized and bounded.

To constrain equation (4.1) with an upper bound, the subsequent subsection discusses the primary types of underwater tethers.

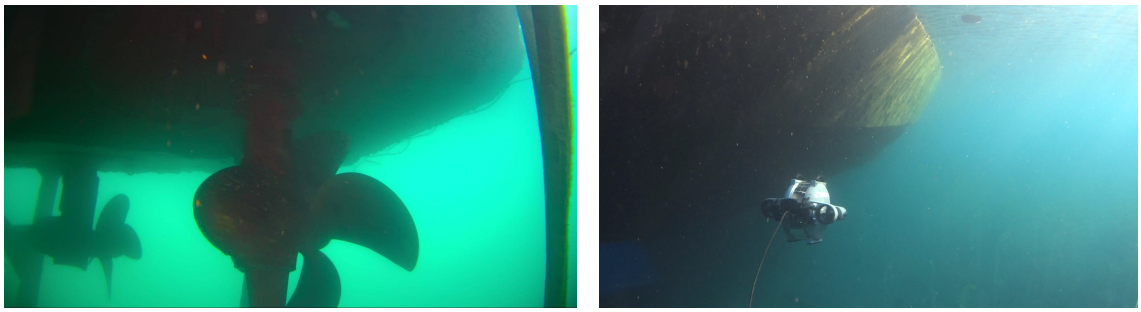
4.2.1 Preferable Cable Shape

Based on ROV's mission, the relative position of the obstacle determines the preferred cable shape which decreases the probability of cable entanglement with those obstacles. Table 4.4 summarizes the obstacle type and position of several scenarios, including: pipeline inspection (Figure 4.5), coastal monitoring (Figure 4.6), ship hull inspection (Figure 4.7), structure inspection (Figure 4.8), search and recovery (Figure 4.9), bridge inspection (Figure 4.10), offshore wind farm inspection (Figure 4.11), and harbour/dam inspection (Figure 4.12).

By observing Table 4.4, we conclude that there are three main types of tether: *U-shape* (Figure 4.13a), *L-shape* (Figure 4.13b) and *Floating*.

Figure 4.5 Pipeline Inspection Scene. Image credit: Satu Teknik Sdn Bhd.

Figure 4.6 Coastal Monitoring Scene. Image credit: BlueRobotics.



(a) (b)

Figure 4.7 Ship Hull Monitoring Scene: **(a)** The ship hull near the thrusters. **(b)** An ROV inspecting the ship hull. Image credit: Deep Trekker.

Figure 4.8 Manual Underwater Structure Inspection. Image credit: Aquatiq Diving.

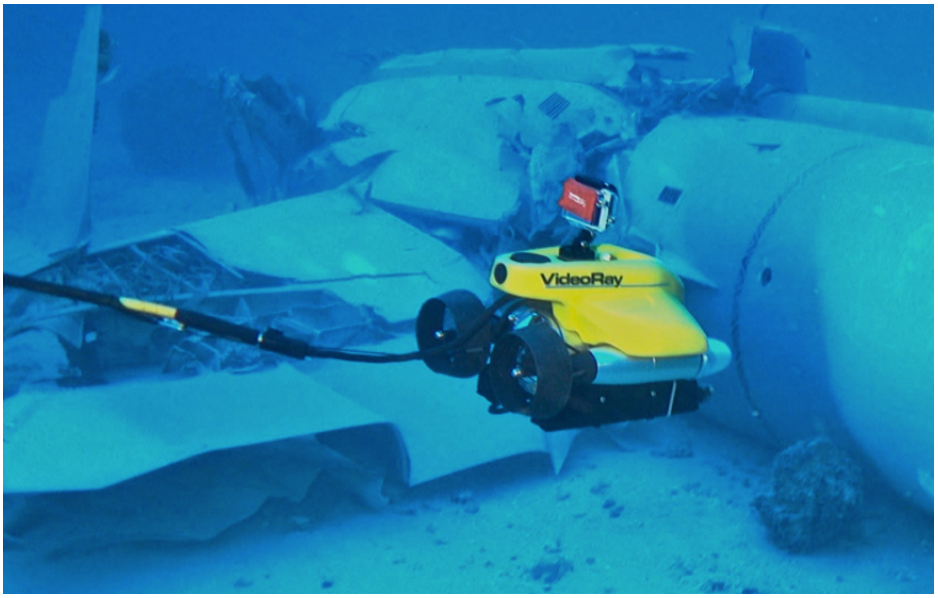


Figure 4.9 Search and Recovery Scene: Image credit: VideoRay.

Figure 4.10 Bridge Inspection. Image credit: AUS-ROV.

Figure 4.11 Offshore Wind Farm Scene. Image credit: Principle Power.

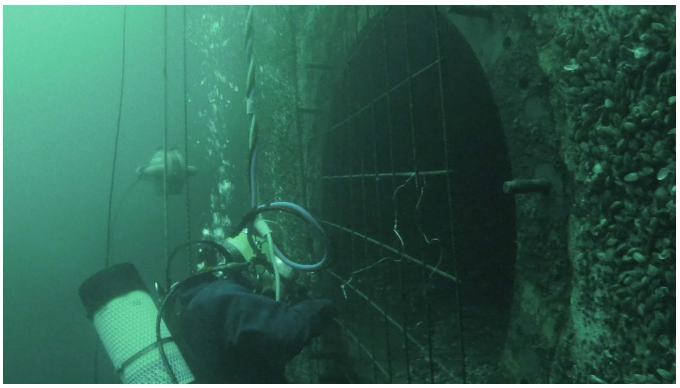


Figure 4.12 Manual Dam Inspection by Divers. Image credit: Deep Trekker.

Table 4.4 Context Analysis of ROV Applications.

Application	**Operation Range***	**Obstacle Type**	**Obstacle Position****	**Optimal Cable Shape**
Pipeline inspection	deep	seaweed	below	L-shape
Coastal monitoring	shallow	coral reefs, seaweed	below	L-shape
Ship hull inspection	deep/shallow	hull	aside, above	U-shape
Infrastructure inspection	deep	objects, pillars	aside, above, below	Floating
Search and recovery	deep	objects on seabed	aside, below	L-shape
Bridge inspection	shallow	pillar, tunnel, rock	aside, below	Floating
Offshore wind farm inspection	deep	pillars	aside, below	L-shape
Harbour/dam inspection	deep	walls, pillars	aside	L-shape

*ROV's operation range (shallow or deep water)
**the position with respect to the ROV

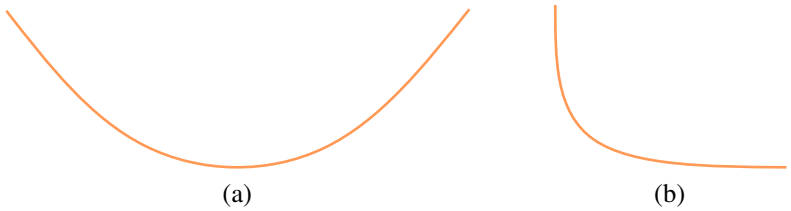


Figure 4.13 Illustration of ROV's tether shape: (a) U-shape. (b) L-shape.

U-shape

This configuration is desirable when the obstacle is positioned above the ROV, making it preferable for the cable to pass beneath the ROV, forming a *U-shape*. For instance, this is typical in ship hull and structure inspections. To achieve a *U-shape*, a positively buoyant tether is required. This can be accomplished either by adding ballasts to the tether or by using cables that are inherently positively buoyant.

L-shape

This configuration is preferred for missions such as pipeline inspection, coastal monitoring, search and recovery, and marine debris exploration. The rationale behind a semi-stretched cable is to avoid contact with the seabed, including obstacles such as corals, rocks, or wrack that could entangle the cable. For this research, focused on inspection tasks, the L-shape tether is adopted. For this configuration, the cable length should satisfy the following inequality:

$$l_{ref} < b + c = |\sin(\varphi) + \cos(\varphi)| \cdot a \leq \sqrt{2} \cdot a \tag{4.3}$$

where a, b, c, and φ are defined in Figure 4.14.

Floating

In certain cases, a floating tether is required in shallow or unknown environments, such as structure and bridge inspections near the water surface. In these situations, modifications to the tether are necessary, such as attaching air tubes to the ROV tether Sakaue et al. (2022).

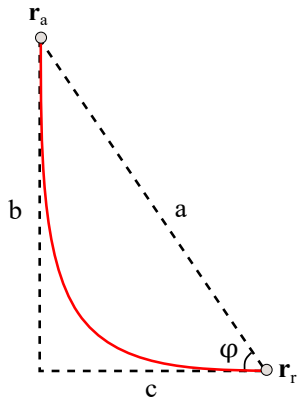


Figure 4.14 ROV cable: $\mathbf{r}_a$ and $\mathbf{r}_r$ are the position of the ASV and ROV respectively. a, b and c are the legs of the triangle. The angle φ is facing the leg b, $a = d$ (the distance). The red curve represents the cable connected between the ASV and the ROV.

4.2.2 Reference Cable Length

In Morinaga and Yamamoto (2023), the authors propose a target cable length l_t for ASV-ROV system for having an approximate *L-shape* of the cable as

$$l_t = \{\beta + \gamma(\theta_R - \theta_A)\}||\mathbf{r_R} - \mathbf{r_A}|| \tag{4.4}$$

where

- θ_R and θ_A are the orientation of ROV and ASV respectively;
- $\mathbf{r_R}$ and $\mathbf{r_A}$ are the position of ROV and ASV respectively;
- $\beta > 0$ and $\gamma > 0$ are coefficients for lengthening the cable according to the differences in linear distance and orientation.

According to the study conducted in Morinaga and Yamamoto (2023), if the ASV follows the target values specified by the control law and the released cable satisfies the length given in Eq. (4.4), then, by properly setting the parameters β and γ, the cable is expected to approximate a catenary shape between the endpoints at the ROV and ASV. The tuning parameters should be as small as possible to reduce the likelihood of entanglement.

To simplify the estimation of β, γ is set to zero $\gamma = 0$, assuming that the heading error $(\theta_R - \theta_A)$ does not affect the reference cable length. Eqs. (4.1) and (4.3) imply that

$$d < l_{ref} \leq \sqrt{2} \cdot d \tag{4.5}$$

Hence, $\beta \in]1, 1.41[$. Notice that, from Eq. (4.4), the reference length is proportional to the distance. In other words, the further the ROV is, the more cable must be released, which is reasonable because the required precision of the ROV's position decreases with distance. Therefore, it is better to increase the rate of additional cable as the distance increases. Equation (4.4) is valid for $d \geq 10$ m. For $d < 10$ m, an additive value should be included so that the reference length becomes

$$l_t = \{\beta + \gamma(\theta_R - \theta_A)\}||\mathbf{r_R} - \mathbf{r_A}|| + \delta \tag{4.6}$$

where $\delta \in [0, 5]$, being negligible when d increases. Hence, Eq. (4.6) is valid for $d \geq 10$ m as well.

For tuning and validating Equation (4.6), it requires acquiring real data of the ROV and the tether. Indeed, the next chapter of the thesis 5.2 covers the the experimental setup for the data acquisition carried out in German Research Center for Artificial Intelligence (DFKI) in Bremen using the BlueROV2. However, the next section adresses the modelling of the winch for lower level of the control to drive the winch motor and make it follow the reference cable length, assuming that the its value is computed by Equation (4.6).

4.2.3 Tether Winch Control

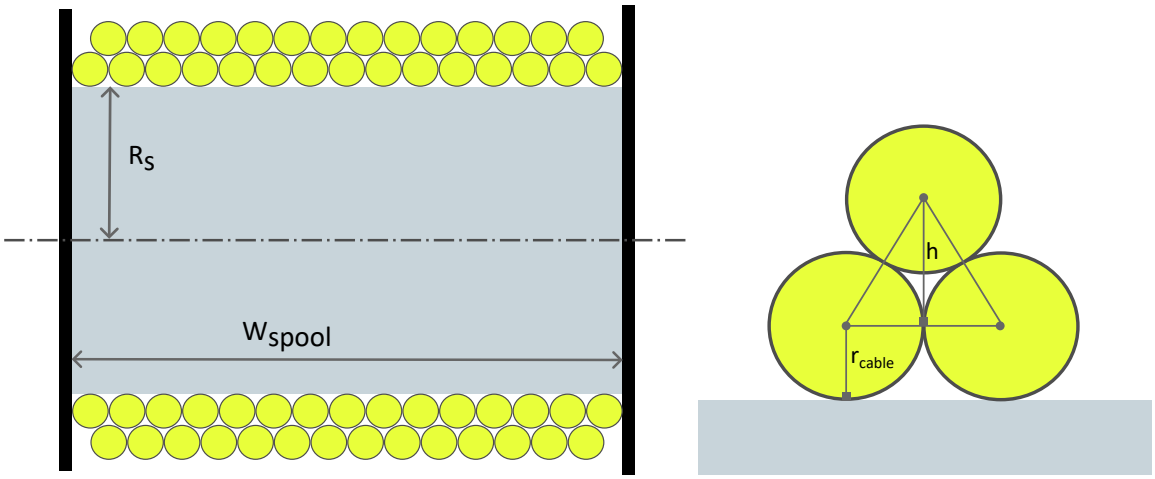
The linear velocity of the pullying cable V_{cable} is given by the general formula:

$$V_{cable} = W_{motor} R_i \tag{4.7}$$

where W_{motor} is the angular velocity of the motor that drives the spool, and R_i is the radius of the spool R_s plus an additional length coming from the number of layers of cable winded around the spool. For this, it is necessary to identify the radius R_i for each layer i of the spool. Each layer of the cable has a fixed number of cable loops which is calculated by the spool width divided by the cable diameter. That is to say, for each layer we can wind a specific length of cable.

$$n_{loop/layer} = \frac{w_{spool}}{d_{cable}} \tag{4.8}$$

$$l_{i_{max}} = 2\pi R_i n_{loop/layer} \tag{4.9}$$



(a) (b)

Figure 4.15 ROV Tether Drum: In 4.15a, the 2D sectional view of the spool is illustrated. The *area* represents the body of the spool, *black* is the side disk that prevents the cable from falling from the sides. The yellow circles represents the section of the ROV tether cable. R_s and w_{spool} are the radius and the width of the spool respectively. In 4.15a, the zoom view of the cable is illustrated. r_{cable} is the radius of the cable, and h is the hight between two consecutive layer of cable.

$$\begin{aligned} R_i &= R_s + h_i \\ h_i &= h_{i-1} + \frac{\sqrt{3}}{2} \times d_{cable} \\ h_{i=1} &= \frac{d_{cable}}{2} \end{aligned} \tag{4.10}$$

w_{spool} is width of spool, d_{cable} is the diameter of the cable, $l_{i_{max}}$ is the maximum cable length in layer i. Hence, if we compute during the initialization phase the possible cable winding lengths for each layer and store them in a look-up table, a mapping between the released cable length and the corresponding cable layer i can be obtained. For instance, the cable length range for layer i is $]l_{i-1_{\max}}, l_{i_{\max}}]$. Once the layer i is identified, using Equation 4.10, it is possible to calculate R_i, and subsequently V_{cable} using Equation 4.7.

The model assumes an automatic cable reel with a level-wind mechanism, considering that the cable is wound perfectly — one loop next to another — and neglects potential entanglements that could occur during the winding operation.

Chapter 5

Experimental and Simulation Setup

The experiments conducted in this project are categorized into two main groups: numerical simulations for evaluating the ASV control strategy, and experimental tests in the test basin for acquiring data related to the ROV and its tether. Consequently, this chapter (which includes the description of the experimental setup) is also divided into two principal sections: the first focuses on the ASV–ROV numerical simulation setup, while the second presents the tether data acquisition experimental setup. In the first section, the software architecture of the ASV system is described, and key configuration parameters are identified. In the second section, the first three subsections outline the setup of the three data sources: the BlueROV2, the Motion Capture System (MoCap), and the tension sensor. Subsequently, the overall data acquisition process is discussed, detailing the simultaneous recording of data from all three sources. Finally, the last subsection addresses the data processing procedures performed after acquisition.

5.1 ASV-ROV Numerical Simulation

The tool for the initial testing of the proposed ASV-ROV cooperative motion is the simulator that is developed in the lab GRAAL[1]. The simulator includes the dynamic models of the ASV and the ROV, hence by using this simulator, all the parameters of the activation functions were tuned and set correctly. In the next paragraph, we talk about the software architecutre of this simulator and the extensions added to test our ASV control strategies.

[1] https://www.graal.dibris.unige.it/

5.1.1 Software Architecture

To implement the control strategy defined in Section 4.1 for the ASV motion, a dedicated software architecture was designed, as illustrated in Figure 5.1 Simetti et al. (2018); Simetti and Casalino (2016). A software-in-the-loop simulator had been previously developed for the ULISSE ASV. For the purpose of this work, the simulator was extended by integrating the ROV and TMS models, enabling the evaluation of the ASV–ROV cooperative system. Figure 5.1 depicts the overall software architecture of the ASV.

The Mission Manager layer is responsible for planning and managing transitions between operational states, namely the "Hold" and "ROV Following" modes. In the Kinematic Control Layer (KCL), the Task-Priority Inverse Kinematics (TPIK) approach is implemented. The Dynamic Control Layer (DCL) receives the reference velocities generated by the KCL and converts them into thruster commands for the ASV. The Navigation Layer collects sensor measurements to estimate the position, orientation, and velocity of both vehicles, providing feedback to the control layers.

Regarding the simulation part, the simulator models all sensors mounted on both vehicles while accounting for measurement disturbances and parameter uncertainties affecting each vehicle. The software has been developed using the ROS2 (Robot Operating System) framework. As shown in Figure 5.1, both the simulator and the hardware driver share the same ROS2 interface. This modular design enables a seamless transition from simulation to real hardware, facilitated by the prior validation performed through software-in-the-loop simulations.

Furthermore, the software interface allows dynamic reloading of configuration files following any modifications and enables resetting of ROS topics by reinitializing publishers and subscribers. This functionality is particularly useful, as it permits parameter adjustments and test repetitions without recompiling the entire system, thereby reducing compiling time.

Three configuration files are employed within the system: the KCL, DCL, and Navigation configuration files. The KCL configuration file defines the system states, actions, and tasks, as well as task parameters such as the "Neighborhood Zone" and "Obstacle Zone" radius. The DCL configuration file specifies the control mode—such as Thruster Mapping, Classic PID Control, or Computed Torque Control and the corresponding control loop parameters. The Navigation configuration file defines the filtering approach, including the Luenberger Observer and the Extended Kalman Filter (EKF), together with their respective configuration parameters.

The finite state machine (FSM) of the ASV system is illustrated in Figure 5.2. In the original configuration, the FSM comprised four main states: *Hold*, *Move to Target*, *Path*

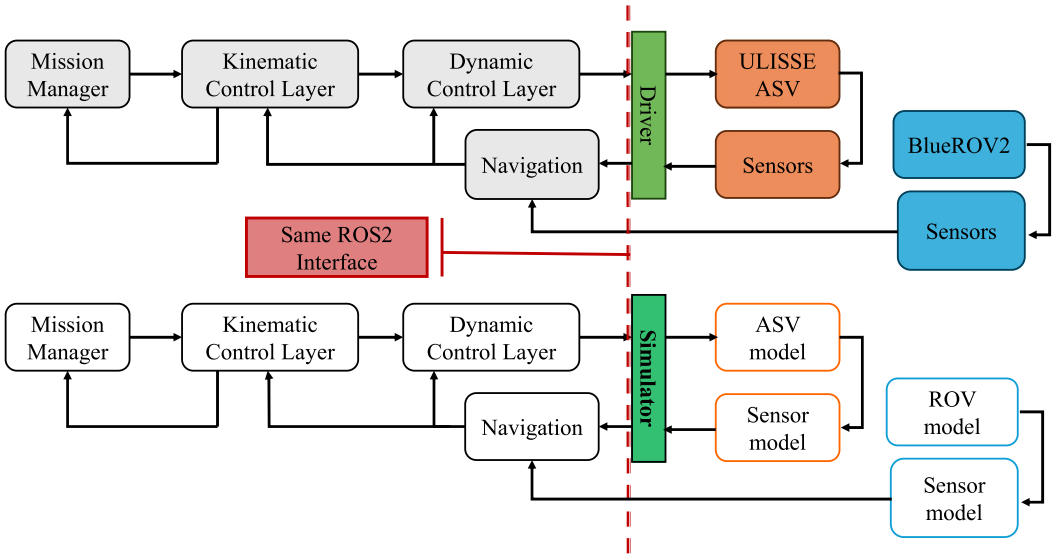


Figure 5.1 Software architecture of the ASV: for the real system (above), and the simulator (below). Both the developed simulator and the real hardware system have the same ROS2 interface which makes it simple for a realistic Software-In-The-Loop assessment of the system performance.

Following, and *Halt*. The *Hold* state maintains the ASV in a fixed position, while the *Move to Target* state commands the vehicle to navigate toward a specified position. The *Path Following* state enables the ASV to track a predefined trajectory, and the *Halt* state serves as the initial mode of operation. The *Halt* state can also be used to deactivate all controllers and thrusters, for instance, in the event of a program failure.

For the purpose of this thesis, an additional state, *ROV Cooperative Navigation*, was introduced to support coordinated ASV–ROV operations. State transitions are user-controlled and are triggered through commands sent via the graphical user interface (GUI).

5.1.2 Simulator Setup

Regarding the numerical simulation setup, the simulations were carried out using the dynamic models of both the ULISSE and BlueROV2 vehicles. Furthermore, the simulations incorporated the actual control software employed for these marine vehicles, enabling a realistic software-in-the-loop (SIL) evaluation of their performance. Initially, `Rviz2` was used to visualize the simulation scene in three dimensions, as shown in Figure 5.3. Subsequently, the `StoneFish` simulator was integrated to provide a more realistic rendering of the ocean environment, as illustrated in Figure 5.4. A pair of wind turbines was added to the simulated

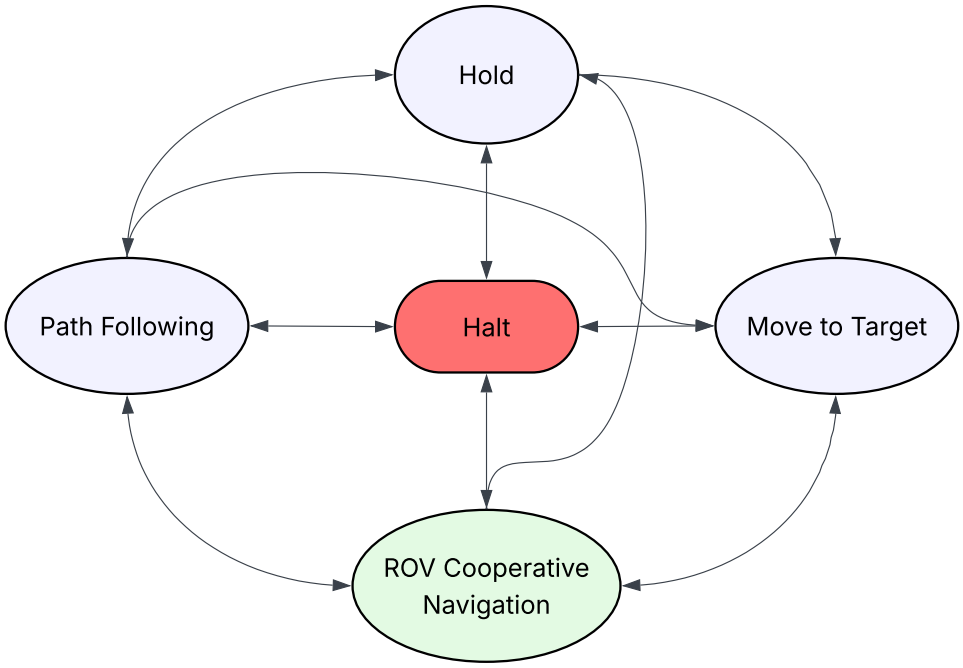


Figure 5.2 Finite State Machine of the ASV: The *red* block is the starting state, the *purple* blocks are the existing states, and the *green* block is the developed state for the project of the thesis.

environment, each with a blade diameter of 10 m and an inter-turbine spacing of 20 m. The values of the parameters used during the recorded simulations are reported in Table 5.1.

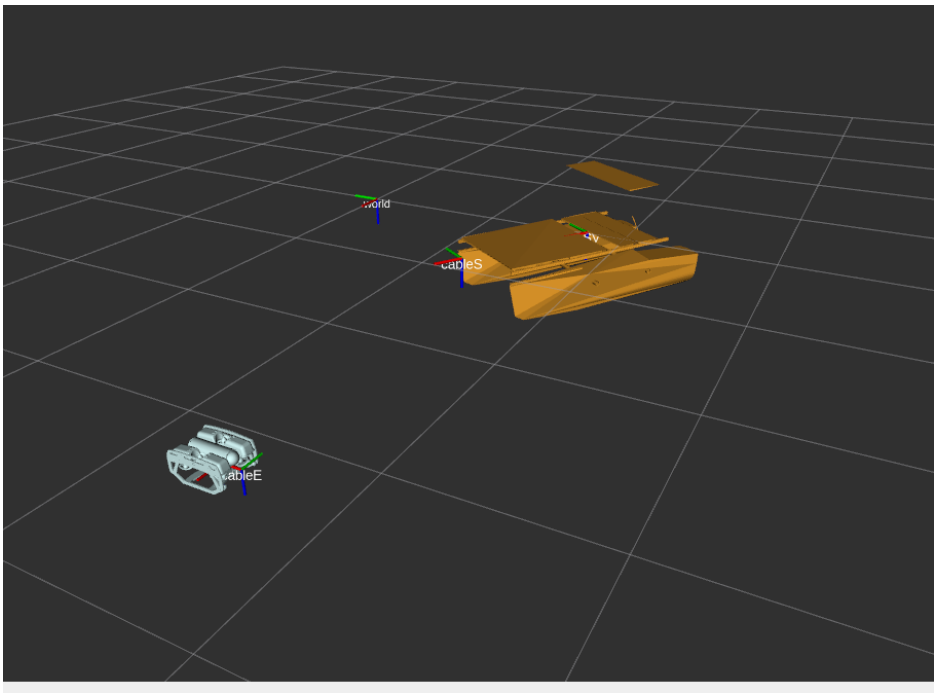
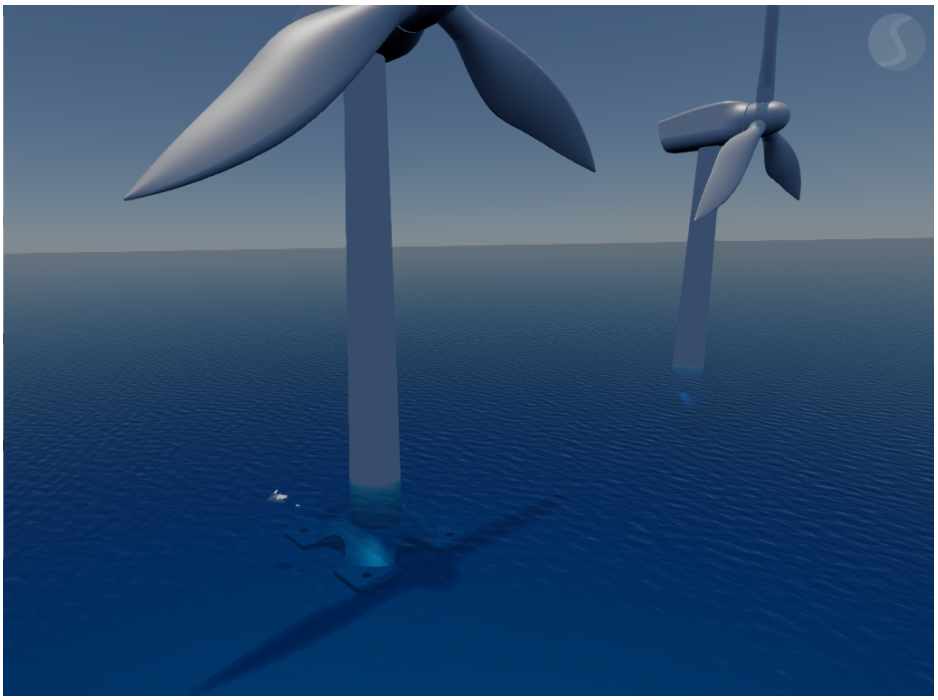


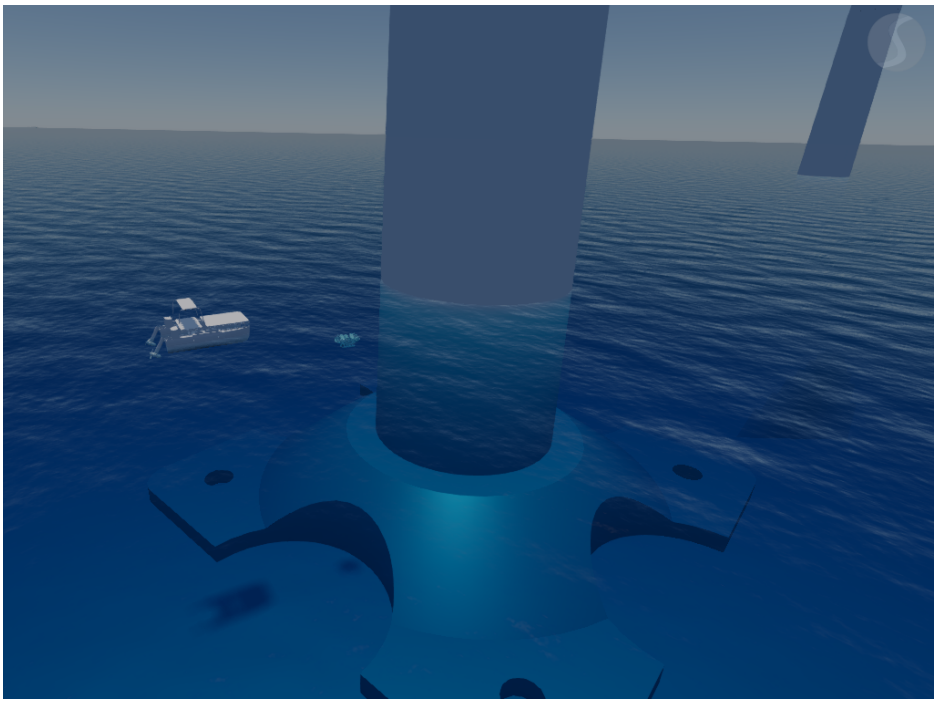
Figure 5.3 Rviz2 for the simulation visualizer: The grey vehicle is the ROV, and the orange one is the ASV.

5.2 Tether Data Acquisition Setup

As explained in Section 4.2.2, the objective of this experiment is to collect ROV and tether data during ROV operations in water and to analyze the behaviour of the tether, with particular focus on its relationship to the measured tension. The experiments were conducted by deploying an ROV in the test basin and remotely piloting it while collecting high-precision position data using a motion capture system (mo-cap), as illustrated in Figure 5.5. The mo-cap system tracked the motion of both the ROV and its tether. Additionally, a tension sensor was mounted on the surface near the tether drum to measure the tension applied to the cable. The measurement setup consisted of the tether passing around a pulley attached to the shaft of the tension sensor, after which it extended into the water to connect with the ROV. In parallel, command and feedback messages transmitted between the ROV and the control station via the MAVLink protocol were recorded. The following sections provide detailed descriptions of the data acquisition systems and methodologies used.



(a)



(b)

Figure 5.4 Stonefish for the simulation visualizer: The white vehicle is the ASV, and the blue one is the ROV. **(a)** Wind farm scene. **(b)** Zoom on ASV-ROV system near the wind turbine pillar.

Table 5.1 KCL configuration files of the Cooperative Navigation with ROV state.

Parameter	**Value**	**Unit**	**Explanation**
minHeadingError	0.25	rad	The starting angle for the activation of ASV alignment towards the ROV task.
maxHeadingError	0.75	rad	The angle where the activation function of ASV alignment towards the ROV task is fully active.
minAcceptanceRadius	4.0	m	The "Neighborhood Zone" radius.
obstacleGoalAcceptanceRadius	3.0	m	The distance from the alignment point where the ASV can be considered to reach the alignment point.
minObstacleZoneRadius	4.0	m	The radius of the "Obstacle Zone" where the ROV is considered to be fully inside the "Obstacle Zone".
maxObstacleZoneRadius	10	m	The starting distance of the "Obstacle Zone".

5.2.1 BlueROV2 Onboard Systems and Telemetry Logging

ROV The core of the setup is the BlueROV2[2], a low-cost, highly modular remotely operated vehicle, powered by a lithium-ion battery and equipped with a Newton Subsea Gripper. The gripper is irrelevant to this experiment but influential in ballast placement as shown in Figure 5.6a. The ROV was connected via a neutrally buoyant tether, specifically the Fathom ROV Tether, which consists of four twisted pairs and has a diameter of 7.6 mm, a minimum working bend diameter of 75 mm.For more information regarding the physical specifications, the reader can refer to the tether cable's technical details[3]. As for the hydrodynamic coefficients of the ROV and the tether material properties, these were identified and published in Zand (2009).

Sensors The BlueROV2 is equipped with a range of onboard sensors, including a 6-DOF IMU, Dual 3-DOF magnetometers, an internal barometer, an external pressure/depth, a temperature sensor, and current and voltage sensors. The control architecture consists of an Autopilot, specifically the Navigator ControllerFlight[4] running the ArduSub[5] firmware, and a

[2]https://bluerobotics.com/store/rov/bluerov2/
[3]https://bluerobotics.com/store/cables-connectors/cables/fathom-rov-tether-rov-ready/
[4]https://bluerobotics.com/store/comm-control-power/control/navigator/
[5]https://www.ardusub.com/

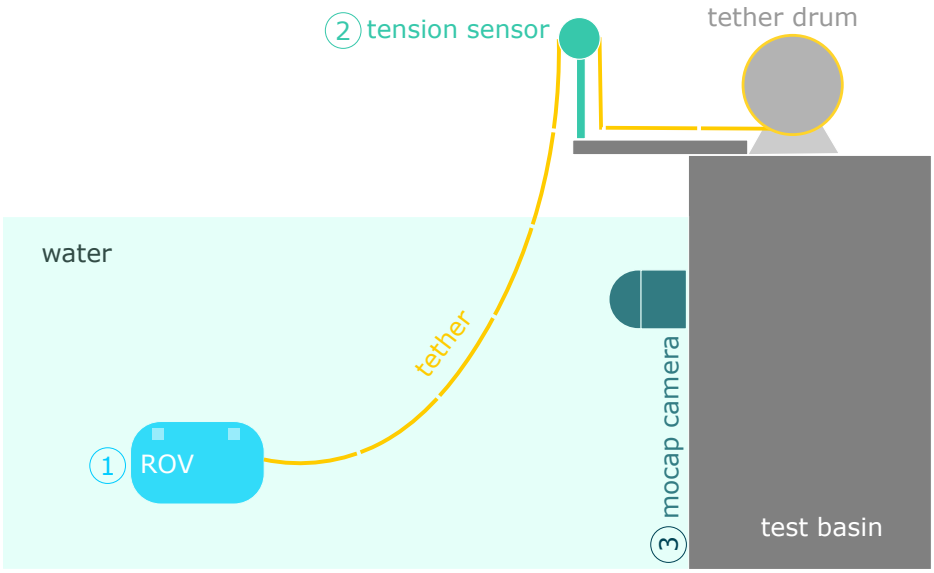
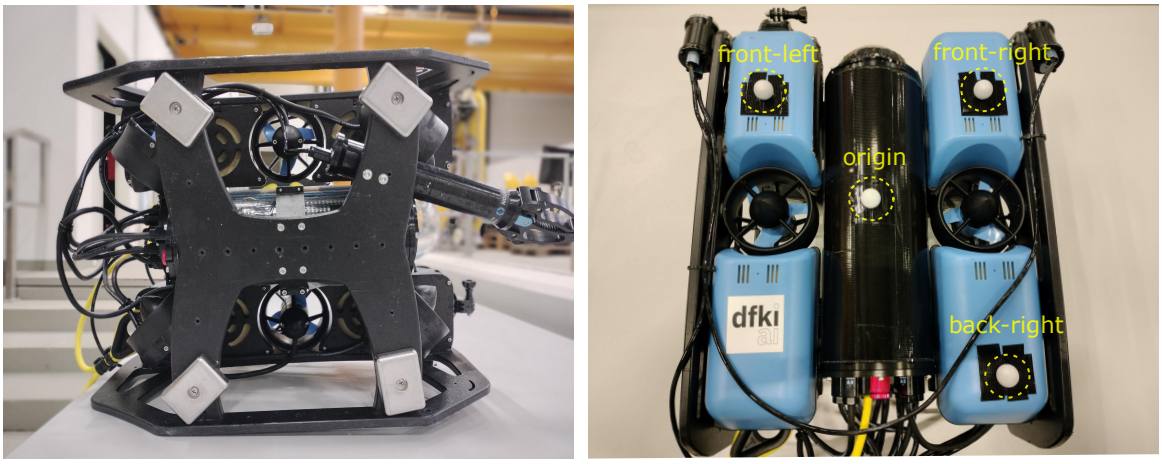


Figure 5.5 Experimental Setup: The ROV is deployed in the test basin and operated remotely, while the tension sensor measures the tether tension at the highest point outside the water, near the tether drum. The mo-cap system tracks the position of both the ROV and its tether. Data are acquired from three sources: the ROV, the tension sensor, and the mo-cap system.

companion computer, running BlueOS[6]. The companion computer facilitates communication between the autopilot and the topside computer by relaying over an Ethernet connection MAVLink messages[7]. For more information regarding the BlueROV2's hardware connections, readers are encouraged to explore the provided link for the BlueROV2's hardware connections[8].

ROS Topics Although MAVLink messages can be monitored using the *MAVLink Inspector* tab in the QGroundControl interface, this tool does not support message logging. To address this limitation, we developed custom Robot Operating System (ROS) nodes that subscribe to MAVLink messages, convert them into ROS-compatible formats, and publish them on ROS topics. These topics are then recorded using the ROS2 bagging system, enabling time-synchronized logging of MAVLink data for offline analysis.

[6]https://docs.bluerobotics.com/ardusub-zola/software/onboard/BlueOS-1.0/
[7]https://ardupilot.org/dev/docs/mavlink-basics.html
[8]https://www.ardusub.com/introduction/hardware-options/connection-diagrams.html



(a) (b)

Figure 5.6 BlueROV2 Setup: **(a)** Bottom View: The position of the 4 ballasts for balancing the ROV and making it neutrally buoyant. **(b)** Upper View: Fixing 4 mo-cap markers to define a rigid body in the Qualisys Track Manager (QTM) software.

5.2.2 Pose Estimation via Optical Motion Capture System

Test Basin The experiments were conducted in the test basin of the Maritime Exploration Hall in the Robotics Innovation Center (RIC) in the German Research Center for Artificial Intelligence GmbH (DFKI). The basin measures 23 m x 19 m x 8 m filled with 3.4 million liters of saltwater. Precisely, the experiments were executed in the water having the following conditions: conductivity 2.06 ± 0.01 S/m, temperature 22.68 ± 0.01 °C, and pressure 103520.0 ± 0.3 Pa. Based on the Practical Salinity Scale of 1978 (PSS-78), the salinity and density of the water can be calculated. This facility provides controlled and observable conditions, enabling repeatable experiments regardless of external weather variations. The basin is equipped with an underwater motion tracking system for 3D motion capture. A total of twelve cameras detect optical markers on the robots and compute their precise position and orientations within the basin, as illustrated in Figure 5.7a.

Motion Capture System The mo-cap system installed in the basin of Maritime Exploration Hall at RIC has the following specifications: manufacturer: Qualisys AB; number of cameras: 12; resolution: 12 megapixels per camera; sampling rate: 250 Hz; accuracy: approximately 4 mm. The cameras are distributed along three edges of the basin near the surface of water, and tilted downward at an angle of approximately 10 °, as shown in Figure 5.7b. Five cameras are mounted on each of the long edges, and two cameras are installed on the short edge, as illustrated in Figure 5.8. After calibrating each camera and adjusting the *Exposure & Flash*

Time and *Marker Threshold* parameters, the system was configured with an *Exposure & Flash Time* of 500 μs and a *Marker Threshold* of 20%.

Markers A mocap system is a technology that records the movement of people or objects by tracking the movement of markers placed on a subject's body. Two different types of markers were used for these experiments. For the BlueROV2, [9] were mounted, as depicted in Figure 5.6b. These markers were selected for their durability and reliability in prolonged underwater use, offering better performance than traditional taped mo-cap markers. In general, the markers are placed asymmetrically on the most visible surface of the vehicle, as seen by all cameras. According to our setup, the 12 cameras are mounted near the top of the pool and tilted downwards, thus the upper surface of the ROV is the most visible. The specific marker positions were chosen to enable the identification of the ROV's *body-frame coordinate system*. In particular

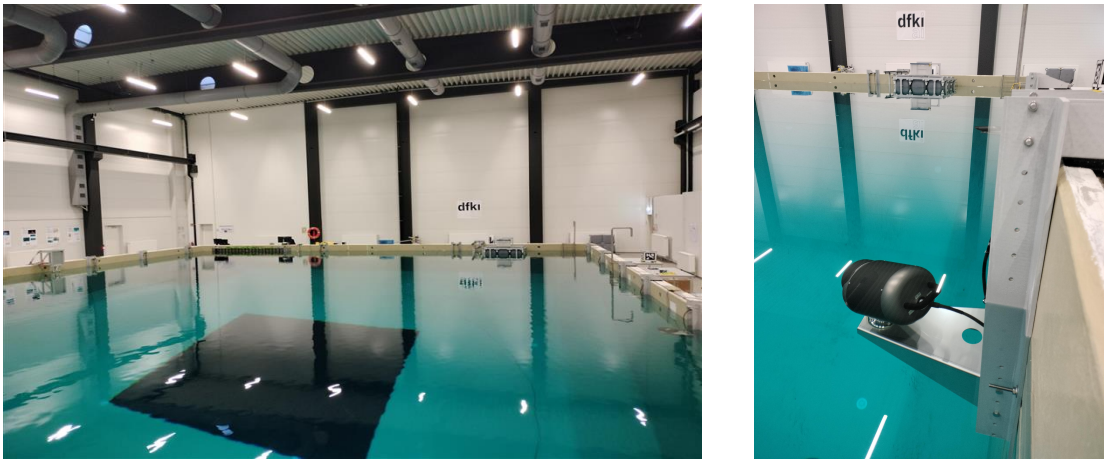
- The vector from the back-right marker to the front-right marker defines the x-axis direction.
- The vector from the front-left marker to the front-right marker defines the y-axis direction.
- An additional origin marker defines the origin of the body-frame.

The spherical markers were attached to the ROV using bolts fixed with strong waterproof tape (PATTEX Power Tape). Once the bolts were secured to the ROV surface, the markers were mounted by simply tightening them onto the bolts.

The second type of used marker was the *retro-reflective underwater tape*. This tape was selected for marking the tether instead of using spherical markers, as it has minimal impact on the cable's dynamical behavior. Strips of retro-reflective tape, each approximately 1.5 cm wide (Figure 5.9a), were attached to the tether at regular intervals of 50 cm. Since the maximum cable length released during the experiments was about 13 m, only the first 20 m of the tether were marked, as shown in Figure 5.9b.

Qualisys Track Manager (QTM) QTM is the mo-cap software used for object tracking, post-processing, and data recording. It enables the definition of rigid bodies, allowing the BlueROV2 to be modeled and tracked in real time. Consequently, the vehicle's position and velocity were computed and published as ROS topics in real time. However, the tether

[9] https://www.qualisys.com/accessories/markers/super-spherical-markers/



(a) (b)

Figure 5.7 Mo-cap System Setup: **(a)** Test basin covered by 12 mo-cap cameras. **(b)** A mo-cap camera fixed inside the basin.

was not modeled as a rigid body due to its flexible nature and complex dynamic behaviour. Because no predefined model was available—and generating one was not feasible—the tether tracking data were manually labeled during the post-processing phase. This procedure is discussed in further detail in Section 5.2.5.

5.2.3 Tension Sensing Setup

Sensor For collecting the tension data, the radial force sensor RFS® 150 XY[10] is utilized. The sensor features two orthogonal measurement axes, with a splash-proof surface for wet or very dusty environments (Figure 3.7a). This specific tension sensor allows us to estimate the resultant tension force vector without requiring knowledge of the exact wrapping angle by combining two radial components. The sensor was calibrated using precise weights: $[1, 2, 5, 10, 20]$ kg. The sensor was calibrated to ensure the measurements corresponded to the calibration weights. Another important procedure before starting the experiments is the axis alignment between the sensor axis (x and y) and the horizontal and vertical axes. By applying a vertical force on the sensor, we keep rotating the sensor around the shaft axis until the force measurements on the horizontal axis are almost zero.

Platform The sensor and pulley assembly was mounted on a custom platform constructed using 45 mm $\times$ 45 mm aluminum profiles, as shown in Figure 5.10. To ensure that the tether

[10] https://honigmann.com/k7/g75/i456/Tension-sensors-ALL-2-axis-radial-force-sensor-RFS150-XY.html

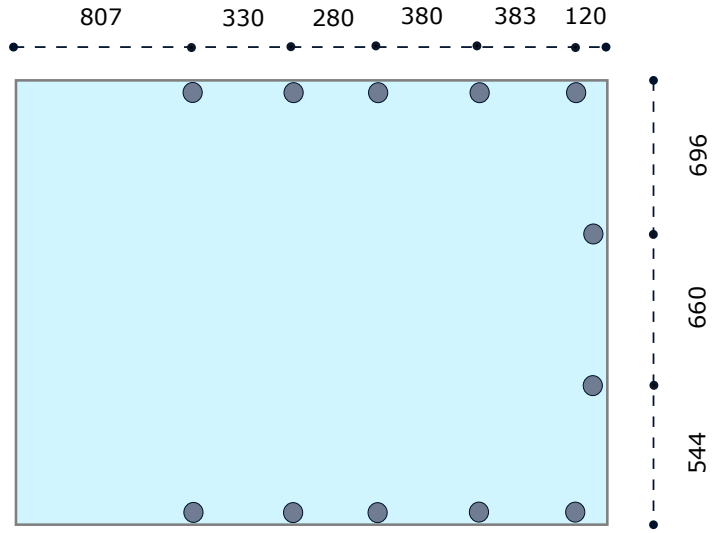


Figure 5.8 2D drawing of the upper view of the test basin: The gray dots are the mo-cap camera positions. (Units are in *cm*)

.

enters the pulley groove along a consistent plane, two guiding hooks were installed upstream, aligned with the groove. This configuration prevents the tether from slipping off during cable deployment and maintains a controlled wrapping geometry. The sensor was positioned slightly higher than the hooks to guarantee a minimum wrapping angle around the pulley, thereby improving the consistency of the tension measurements.

Acquisition software HCC-Easy software[11] was used for data acquisition and logging. The program provides an integrated platform for real-time data acquisition, recording, and analysis of tension data.

5.2.4 Data Acquisition

All data were acquired simultaneously, and the final architecture of the data acquisition system is illustrated in Figure 5.11. To record the MAVLink messages transmitted from the onboard computer of the ROV, a custom ROS package was developed. This package includes a node that connects to the MAVLink UDP port, receives the incoming MAVLink messages, converts them into time-stamped ROS messages, and publishes them on dedicated ROS topics.

Simultaneously, QTM receives image frames from the motion capture (MoCap) cameras, computes the position and velocity of the ROV, and publishes these data on ROS topics.

[11] https://www.honigmann.com/page.php?m_pro=401&lang=2

(a) (b)

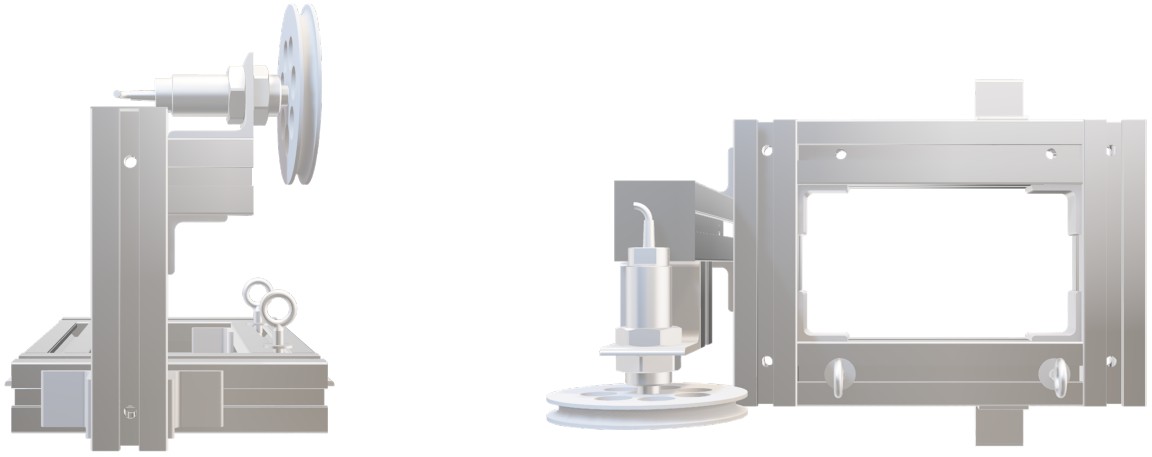
Figure 5.9 Tether Setup: **(a)** A single tape (marker) width $[1.0 - 1.5]$ cm for the mo-cap system. **(b)** The entire marked tether.

The *ROS2 Bag Recorder* node subscribes to all relevant topics and stores the messages in a ROS bag file. Tracking data not detected in real time—such as the tether markers—were manually labeled in QTM after the recording phase and exported following post-processing. Meanwhile, the *HCC-Easy* software handled the tension data from the sensor and stored them in `.HED` file format.

Data acquisition was performed using three different computers: the first, connected to the MoCap system, ran QTM; the second, connected to the BlueROV2, executed the ROS nodes; and the third, connected to the tension sensor, operated the *HCC-Easy* software. All three computers were connected to the same local network to ensure clock synchronization across the systems.

5.2.5 Post-Processing

Labeling Mo-cap Markers QTM can track the trajectories of rigid bodies if their markers are defined as a rigid body in the system prior to recording. In this configuration, the rigid body can be detected in real time, and its position and velocity can be published as ROS topics during the experiment. However, the tether markers are not defined as a rigid body because the distances between them change dynamically according to the tether's shape. Moreover, due to the complexity of the underwater cable model, the tether cannot be detected automatically in real time. For these reasons, the trajectories of the tether markers were manually labeled after the experiment.



(a) (b)

Figure 5.10 3D model of the assembly of the platform and the tension sensor. The assembly is made from: a tension sensor, a pulley, a platform made by aluminum frames, and two hooks for ensuring the cable aligns to the vertical plane of the pulley. **(a)** Side view. **(b)** Upper view.

Manual labeling involves assigning names to the trajectories detected by the motion capture system. A screenshot of the QTM interface is shown in Figure 5.12. In the main window (left side), the ROV rigid body is shown in blue, and the tether markers are depicted in green. Yellow lines connect the tether markers to provide a visual representation of the tether. On the top-right panel, the list of *Labelled Trajectories* is displayed: blue entries correspond to ROV markers detected automatically, while green entries represent tether markers labeled manually.

Initially, all trajectories not associated with rigid bodies are marked as *Undefined Trajectories*. By selecting a trajectory (represented as a dot in QTM) and assigning it a label, it becomes a labeled trajectory. This process is repeated for all tether markers. The tether markers were labeled according to the format ‘MX’, where ‘M’ indicates ‘marker’ and ‘X’ denotes the distance from the ROV along the cable. For example, the first five tether markers were named M0.5, M1.0, M1.5, M2.0, M2.5, and so on.

The middle-right panel of QTM lists all *Undefined Trajectories*. Due to light reflections and other disturbances affecting the motion capture cameras, QTM sometimes detects incorrect trajectories, which initially appear under *Undefined Trajectories*. By filtering these trajectories, erroneous or unnecessary entries can be discarded and moved to the *Discarded Trajectories* list in the bottom-right panel (see Figure 5.12).

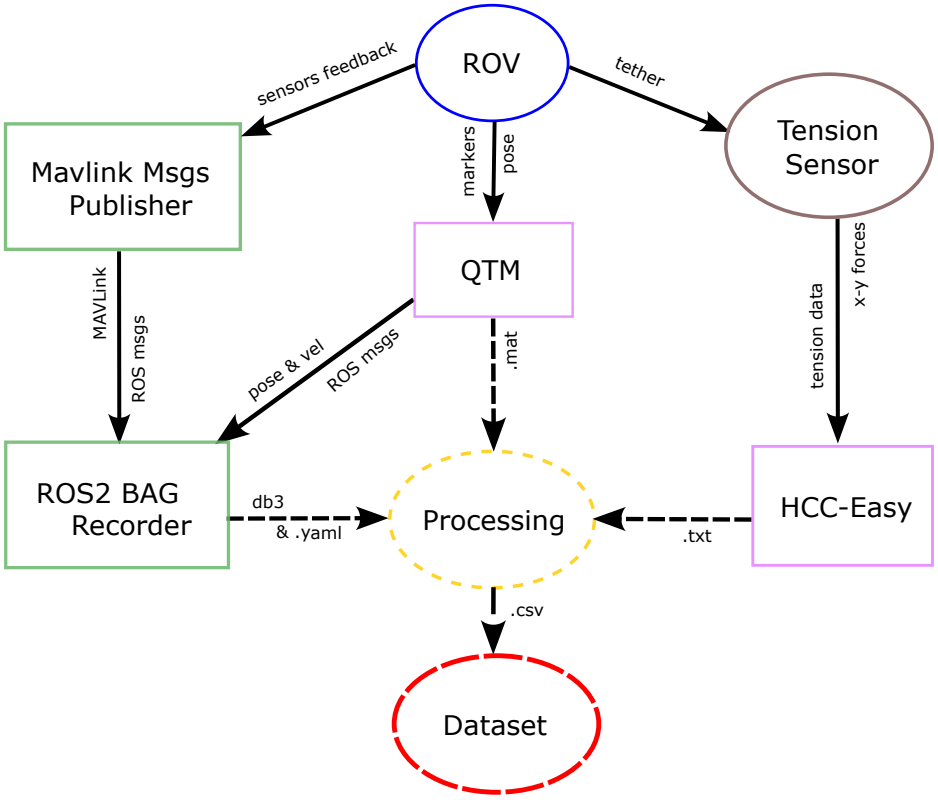


Figure 5.11 Software architecture of the data acquisition system. Color legend: *blue* represents the robot, *pink* represents software, *brown* represents sensors, *green* represents the developed packages, *yellow* represents the post-processing stage, and *red* represents the final dataset. The labels near the dashed arrows indicate the corresponding data formats.

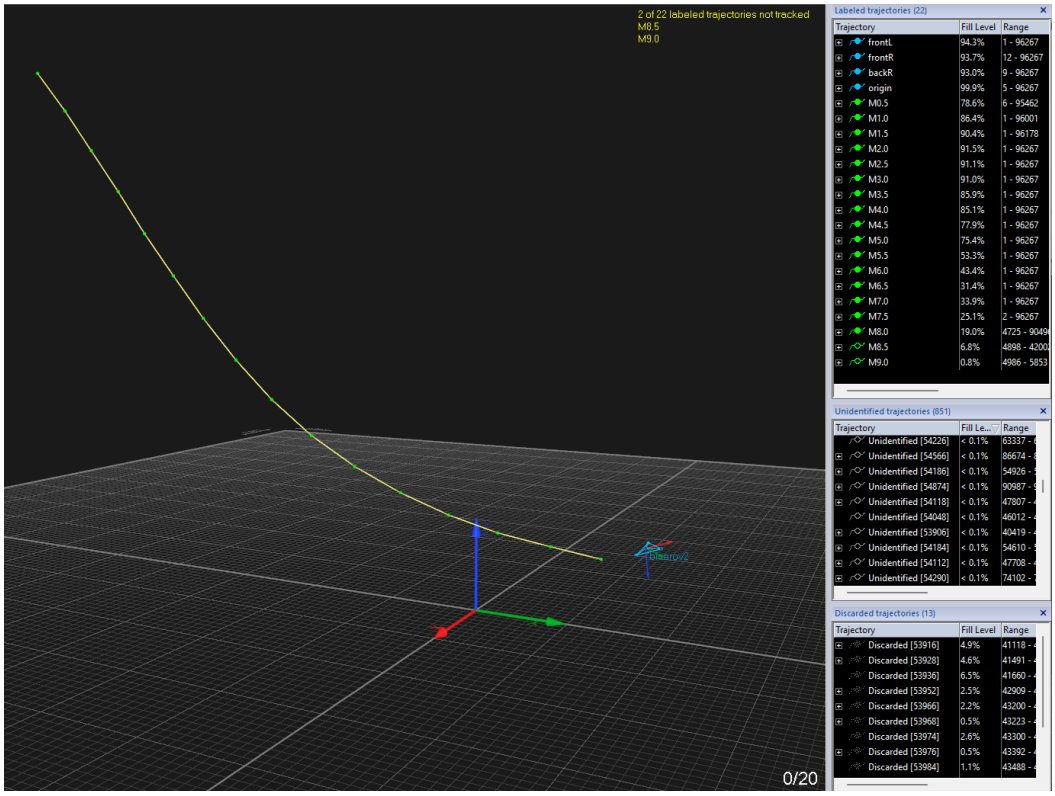


Figure 5.12 Manual annotation of the markers using QTM: Each marker of the tether had to be annotated since there was no automatic labelling due to the lack of the tether model. The blue lines represent the rigid body of ROV, and the green dots are the markers of the tether, while the yellow lines interpolate the tether markers with straight lines to represent the cable. In the upright panel *Labelled trajectories*, the labelled markers are listed; the *blue* ones are the ROV markers, and the *green* ones are for the tether. The last two markers (M8.5 and M9.0) are not filled because they are not detected by the cameras at the moment. The unlabelled and discarded trajectories are listed below in *Undefined trajectories* and *Discarded trajectories*.

Gap-filling and Smoothing Tracking Data QTM offers the capability to process the recorded data, like smoothing the trajectories by removing spikes. In contrast, when the markers are not detected, there are gaps in the trajectory. In such a case, gap-filling functions was required to export continuous trajectories.

Filtering Tension Signal HCC-Easy provides a signal filtering function to remove noise and spikes. However, the recorded data was relatively clear and acceptable, so the tension data presented in this paper is the raw data without applying any additional digital filters.

Data Synchronization Since the data is acquired from three different independent sources, one of the challenging parts in acquiring the dataset was the time synchronization. The ROS messages stored from the ROS topics are all time-stamped. The data exported from QTM has the time stamp of the first frame and the rate of the stored frames. Regarding the tension measurements, the data were time-stamped starting from zero. By knowing the time of file creation, we managed to shift the time and obtain the correct time stamp for each frame. To get the time-stamp of the file creation, we used FileTime[12] MATLAB toolbox fil (2025).

ROV Orientation Correction Both the IMU and the mo-cap system estimate the orientation of the ROV in real time. The IMU computes the orientation using an Extended Kalman Filter (EKF), while the mo-cap system determines it by detecting the markers that define the rigid body. Figure 5.13 shows the ROV's orientation angles obtained from both sources. However, since the reference frames were initially different, the two representations are not directly comparable. Specifically, the IMU provides the ROV's orientation in roll–pitch–yaw (RPY) form with respect to an inertial frame aligned with the North–East–Down (NED) axes. In contrast, the mo-cap system outputs the orientation of the ROV frame relative to its own local base frame. To align the data, a rotation matrix is applied to transform the mo-cap orientation from its local frame to the NED frame, as illustrated in Figure 5.14. After this transformation, both RPY representations refer to the same reference frame and can be meaningfully compared.

In general, the mo-cap orientation is relatively accurate, particularly for the yaw angle. However, discontinuities are observed in the roll angle due to occasional loss of marker detection, as seen in Figure 5.14 around the time intervals [25, 75] s and [160, 180] s, where sudden flips occur. The IMU, on the other hand, provides continuous and reliable estimates for roll and pitch, but the yaw angle drifts over time. This drift is especially pronounced

[12]https://www.mathworks.com/matlabcentral/fileexchange/24671-filetime

when the ROV approaches metallic structures due to magnetometer interference, and has been linked to Gyro-Z bias estimation in the EKF, as discussed in the Blue Robotics forum[13].

To ensure consistency in the dataset, the final ROV orientation combines the roll and pitch angles from the IMU with a yaw angle corrected to align with the mo-cap measurement.

Calculating Tension Force The tension sensor measures the forces F_x and F_y generated by the tether on the pulley (Figure 12), which is not equal to the tension force. In this section, we explain how to calculate the tether tension using the annotation of angles and arms as illustrated in Figure 5.15. According to the datasheet of the tension sensor, the resultant force F_R is calculated based on the tension force F_z and the wrapping angle α:

$$F_R = 2F_z \sin\left(\frac{\alpha}{2}\right) = \sqrt{F_x^2 + F_y^2}, \tag{5.1}$$

where F_x and F_y are the measured forces by the sensor on the vertical and horizontal axes, respectively. In our application, the wrapping angle is not fixed since the tether follows the motion of the ROV. Hence, to calculate the wrapping angle α, the angles φ, ε and β, as illustrated in 5.15,should be calculated using the following equations:

$$\varphi = \tan^{-1}\left(\frac{F_x}{F_y}\right); \tag{5.2}$$

$$\varepsilon = \cos^{-1}\left(\frac{C}{\sqrt{A^2+B^2}}\right) - \beta; \tag{5.3}$$

$$\beta = \tan^{-1}\left(\frac{A}{B}\right); \tag{5.4}$$

$$\frac{\alpha}{2} + \varepsilon = \varphi. \tag{5.5}$$

By substituting (5.2), (5.3), and (5.4) into (5.5), the wrapping angle α can be expressed as

$$\alpha = 2(\varphi - \varepsilon) = 2\left[\tan^{-1}\left(\frac{F_x}{F_y}\right) + \tan^{-1}\left(\frac{A}{B}\right) - \cos^{-1}\left(\frac{C}{\sqrt{A^2+B^2}}\right)\right], \tag{5.6}$$

where A, B, and C are known geometrical parameters with values $A = 285$ mm, $B = 160$ mm, and $C = 54$ mm. Once the wrapping angle α and the resultant force F_R are determined, the

[13]https://discuss.bluerobotics.com/t/yaw-drift-due-to-increase-of-ekf-gyro-bias-estimation/5490

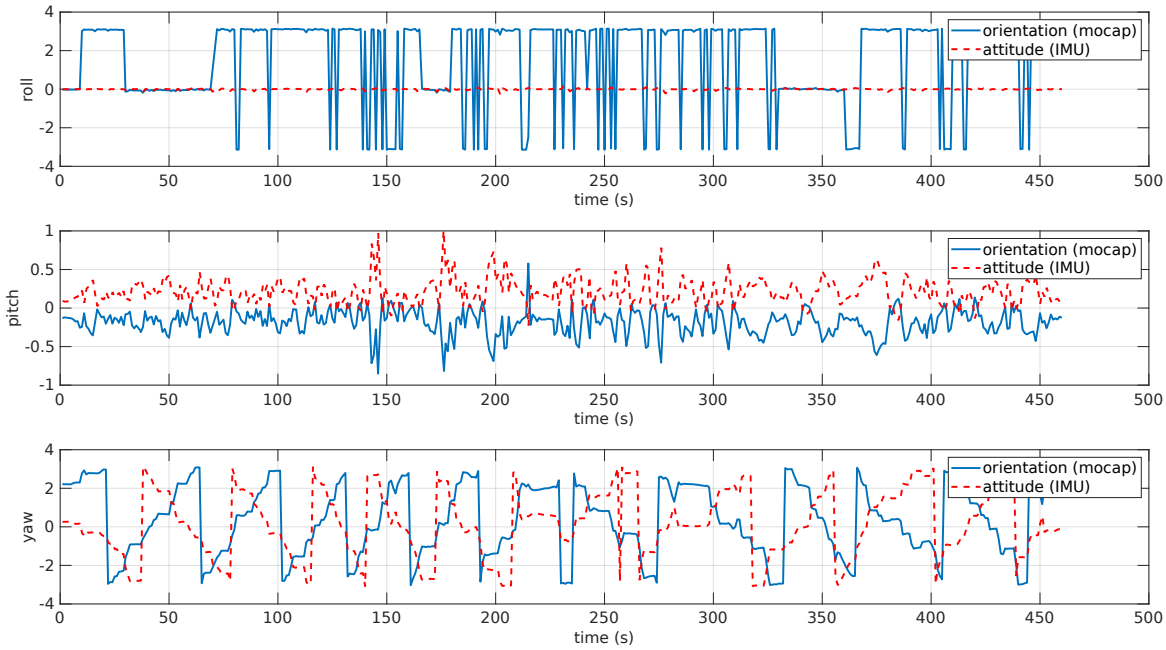


Figure 5.13 ROV orientation angles: raw data from mo-cap and IMU are plotted on different reference frames. The angles are in radians. The data from the mo-cap is projected on mo-cap reference frame, while data from IMU sensor is projected on the NED frame. Hence, the data are completely misaligned.

tension force F_z is calculated as

$$F_z = \frac{\sqrt{F_x^2 + F_y^2}}{2\sin(\alpha/2)}. \tag{5.7}$$

Exporting Data QTM can export data in several formats, including C3D, TSV, AVI, FBX, or MATLAB files. In this work, mo-cap data were exported in MATLAB file format. MAVLink messages were saved using a ROS2 bag recorder[14], producing *metadata.yaml* and *DB3* files. Tension sensor data were exported as *.txt* files by *HCC-Easy*. To unify the data for analysis, all data were processed in MATLAB and converted to Comma-Separated Values (.csv) format after the synchronization. The final curated dataset is therefore provided entirely in .csv format.

[14]https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Recording-And-Playing-Back-Data/Recording-And-Playing-Back-Data.html

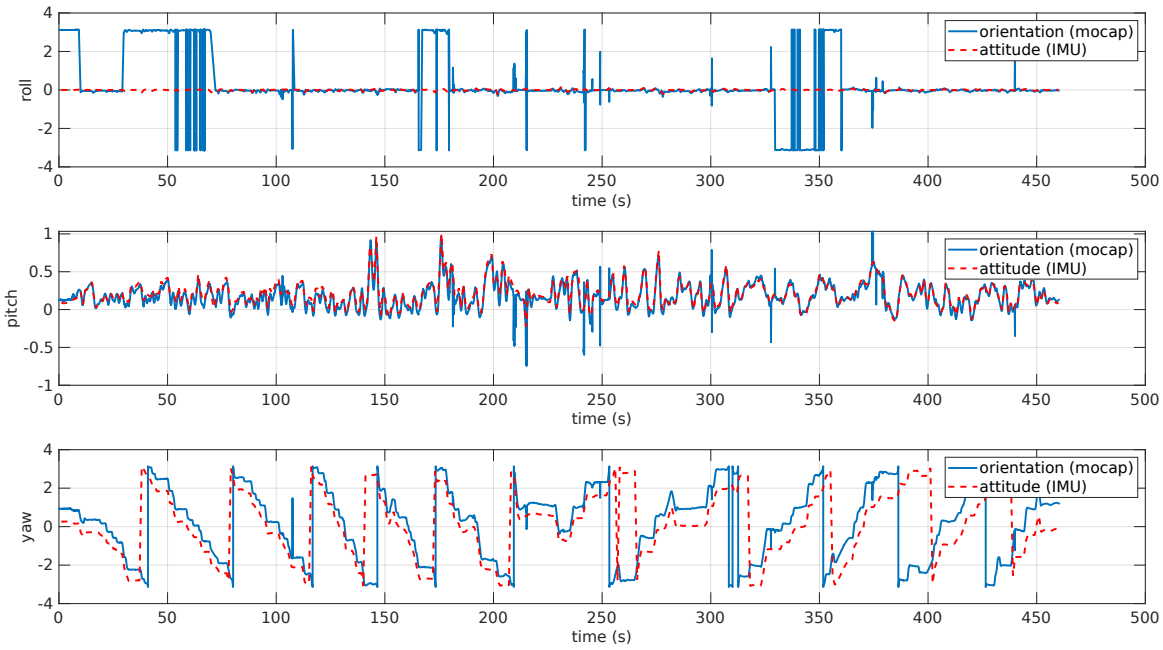


Figure 5.14 ROV orientation angles: data from the mo-cap and IMU projected on the same NED frame. The angles are in radians. The two plots from roll and pitch are aligned after the first step of post-processing. On the other hand, there is still a drift on the yaw data.

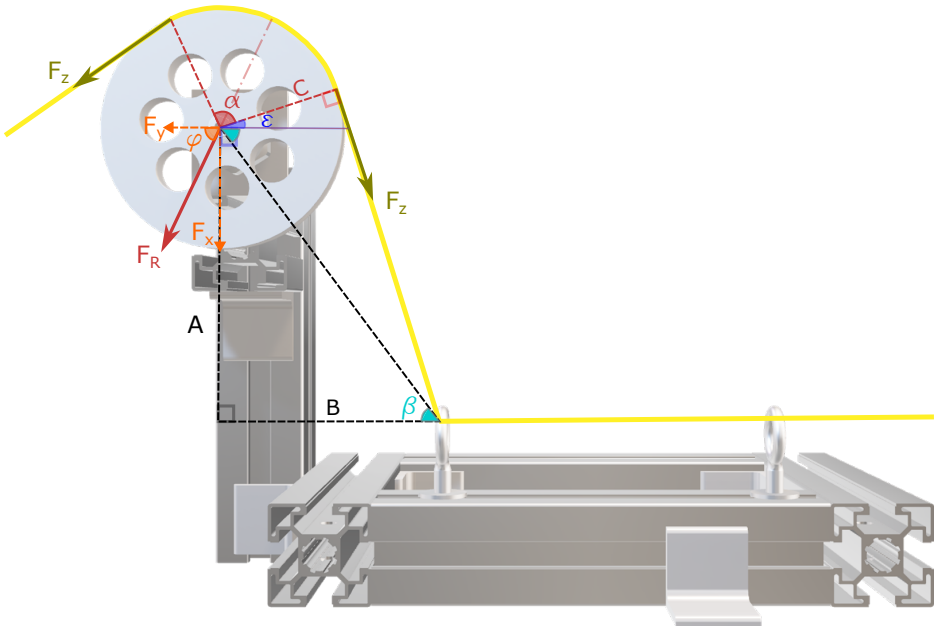


Figure 5.15 Tether Tension calculation: F_z is the tension force of the cable, F_R is the resultant force, F_X and F_Y are the projections of the resultant force on the vertical and horizontal axes, respectively, α is the wrapping angle, and C is the radius of the pulley. A and B are the vertical and the horizontal distances between the center of the pulley and the center of the left hook.

.

Chapter 6

Results

This chapter presents the results of the ASV–ROV simulation and tether modeling. In the simulation results, two control strategies were evaluated: the "Neighborhood Zone" approach and the "Obstacle Avoidance" policy. In the tether modeling section, the acquired dataset from the experimental test is presented and subsequently validated through a series of illustrative plots.

6.1 ASV-ROV Simulation Results

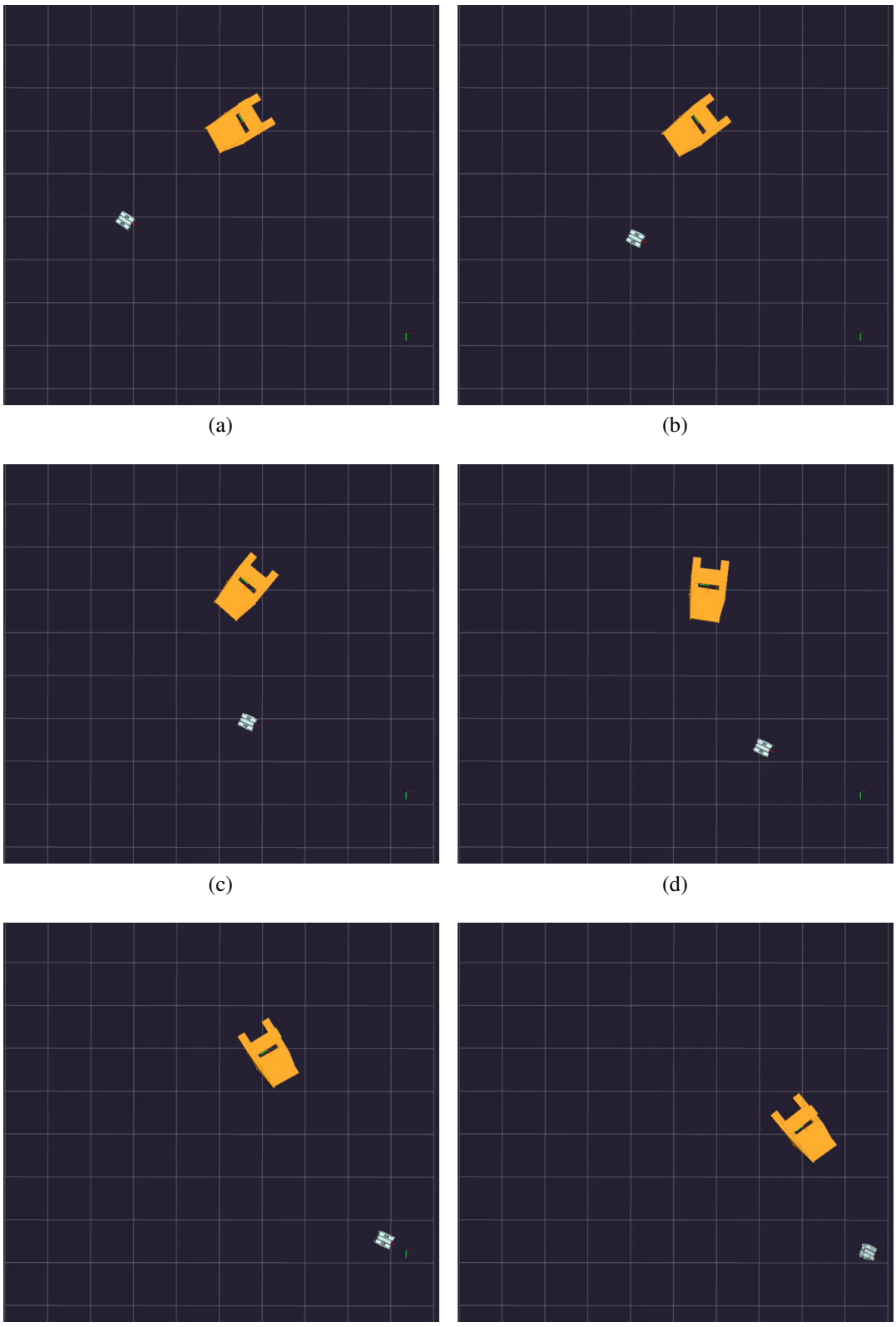
Using the simulator software presented in Section 5.1.1, the proposed control policy described in Section 4.1 was evaluated. First, the "Neighborhood Zone" policy was tested, followed by the validation of the "Obstacle Avoidance" strategy.

6.1.1 Neighbourhood Zone

The simulation starts with an initial position for the ASV and the ROV close to the zero of the world coordinate. When the ROV is given a target point outside the "Neighbourhood Zone", the ASV starts aligning and then approaching the ROV as we have defined in the action. Besides, when the ROV goes deeper in the altitude, more cable is released in the water. A few screenshots from one of the simulations are reported hereafter in Figure 6.1 to show the overall motion of the two robots. The full video of this simulation is available online[1].

To validate the control system, plots of the Task Activation Value and the error values for both the "Align-To-Target" and "Cartesian-Distance" tasks were generated during a

[1] https://www.youtube.com/watch?v=eIhYmUTOLgE



(a) (b)

(c) (d)

(e) (f)

Figure 6.1 Screenshots of ASV/ROV cooperative motion simulation (x-y plane projection). The grey vehicle is the ROV, commanded to move towards the bottom-right corner, and the orange one is the ASV. The full video of this simulation is available at https://www.youtube.com/watch?v=eIhYmUTOLgE

representative test, as shown in Figure 6.2. From the four plots in these figures, it can be observed that when the distance between the two vehicles exceeds the red threshold, the activation function of the “Cartesian-Distance” task begins to increase, while the “Align-To-Target” task is fully activated. Conversely, when the distance error falls below the red threshold, the Cartesian-Distance task activation value drops to zero, whereas the activation value for the angular position decreases only once the distance error completely crosses the threshold.

It is important to note that, during these simulations, the activation functions ensured that tasks were not continuously engaged throughout the mission. This design allows ASV tasks to be activated only when the ROV is outside the defined “Neighbourhood Zone”. The smooth activation functions contribute to a correspondingly smooth motion of the ASV.

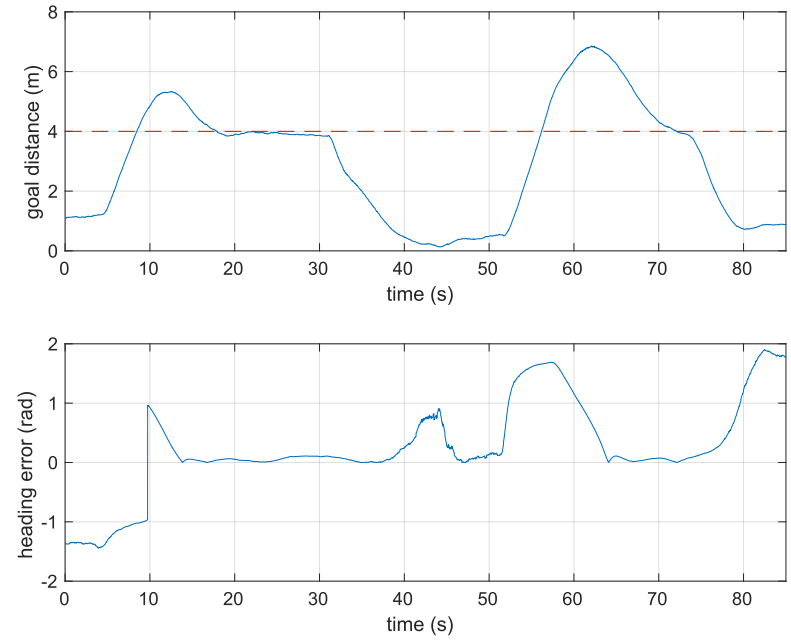
After conducting several trials in which the ROV was moved to different positions and altitudes, it was confirmed that the proposed ROV-following strategy operated successfully, as demonstrated in the preceding discussion.

6.1.2 Obstacle Avoidance

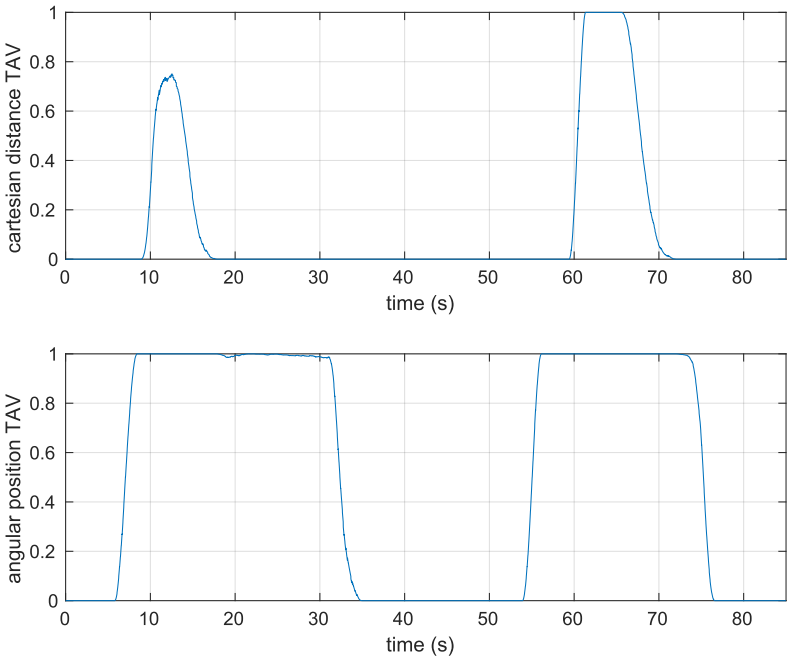
As the “Neighbourhood Zone" policy was successfully validated, the next step was to check the pillar-like obstacle avoidance scenarios. The obstacle avoidance scenario was tested in a series of simulations that were recorded and analysed. After running several trials by moving the ROV to different positions, inside/outside of “Obstacle Zone” and “Neighborhood Zone”, the proposed strategies were verified and confirmed to be working properly, as demonstrated in the corresponding video available online[2]. With the 3D visualizer (Figure 6.3), the overall motion control was verified. However, to check the activation of the tasks in more detail, Task Activation Functions (TAV)s were plotted in Figure 6.4, and since the activation of the tasks is based on several factors, including the distance values and the heading errors, those variables were also plotted in Figure 6.5 and 6.6.

According to the plot in Figure 6.5, the simulation starts from case (a), where the distance between the ROV and the obstacle exceeds the “Obstacle Zone” (the dashed purple line in Figure 6.5). At $t = 6$ s, the ROV enters the “Obstacle Zone”, and the distance between the ASV and the generated alignment point is over the threshold (the green dashed line in Figure 6.5. Consequently, the tasks “Align to the Alignment Point" and “Move to Alignment Point" are activated as shown in the plots in Figure 6.4. Both tasks are activated around the same time $t = 6$ s; however, the “Align to the Alignment Point" task has a higher priority

[2]https://youtu.be/76mCFr8oJ6w



(a)



(b)

Figure 6.2 **(a)** Distance between the vehicles (in meters), and the difference in heading (in radians) between the ASV and the ROV during the experiment. The red dashed line in the goal distance plot represents the threshold which is the radius of the “Neighbourhood Zone" in practice. **(b)** Task Activation Value (TAV) for the Align-To-Target (lower) and the Cartesian-Distance (upper) tasks throughout the experiment.

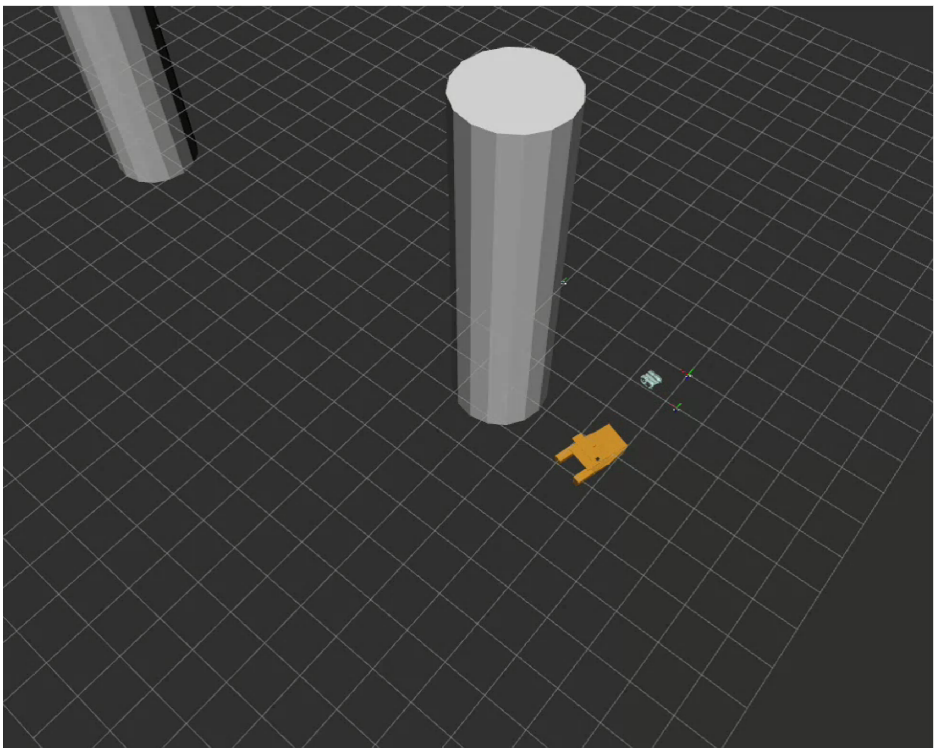


Figure 6.3 A recorded simulation: the grey vehicle is the ROV, the orange one is the ASV, and the white cylinder is the pillar-like obstacle. The full video of this simulation is available at https://youtu.be/76mCFr8oJ6w

than the “Move to Alignment Point" task, hence the first task is executed as the heading error between the ASV and the Alignment Point decreases, see Figure 6.6. Once the heading error is small enough, the ASV uses its degree of freedom to solve the second activated task, which is “Move to Alignment Point" task. This can be seen from the distance plot around $t = 20$ s in Figure 6.5, where the distance between the ASV and the Red Flag decreases. This phase is represented by the case (d) in Figure 4.2. As specified in the control policy, the ASV aligns to the ROV after reaching the Red Flag, so at $t = 24$ s, the “Align to ROV" task is activated as evidenced in Figure 6.4, and the heading error between the ASV and the ROV drops as depicted in Figure 6.6. After that, the ROV moves around the “Obstacle Zone”, generating a different alignment point, causing an increase in the distance between the ASV and the new alignment point position, which exceeds the threshold at $t = 36$ s, refer to Figure 6.5. Therefore, the ASV repeats the cases (b), (d), and (e) as shown in Figure 6.5 and Figure 6.6.

At the end of the simulation, the ROV moves out of both the “Obstacle Zone” and the “Neighborhood Zone”, which is represented by the case (f) in Figure 4.2. In this case, the alignment point tasks are deactivated, but since the ROV is outside the “Neighborhood Zone”,

the tasks “Move to ROV" and “Align to ROV" are activated at $t = 73$ s as shown in Figure 6.4.

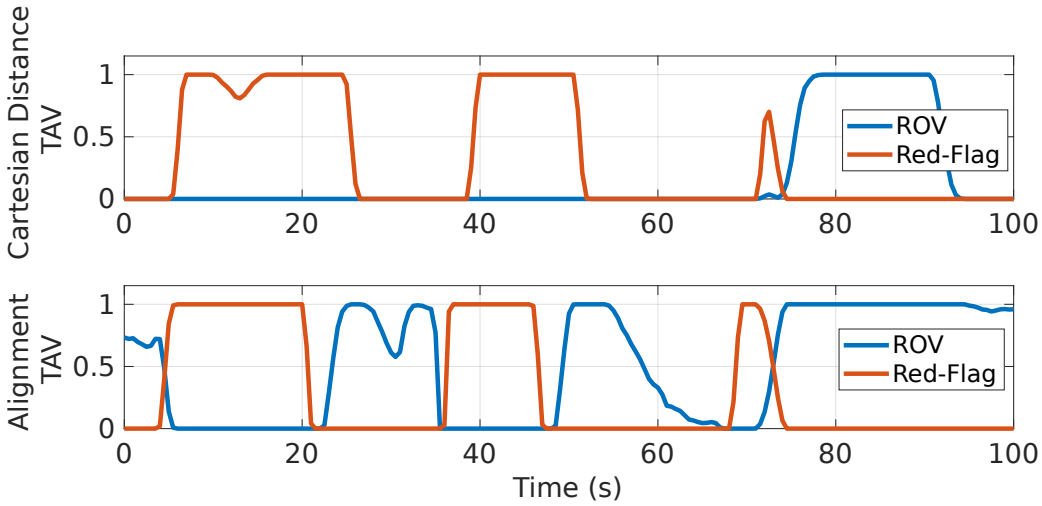


Figure 6.4 Task Activation Value (TAV) vs. Time: In the upper plot, the *blue* line represents the TAV of “Move to ROV" task, and the *red* one represents the TAV of “Move to Alignment Point" task. Similarly, for the lower plot, the *blue* line represents the TAV of “Align to ROV" task, and the *red* one represents the TAV of “Align to the Alignment Point task.

6.2 Tether Modelling Results

In this section, the acquired dataset, which was first introduced in section 5.2, is presented in detail. Then, data validation is carried out by plotting different components of the data collection. Finally, the use of the dataset is discussed for the control of the tether length.

6.2.1 Data Records

The final dataset is available on Zenodo Khanmeh et al. (2025a). The uploaded dataset comprises the complete set of experimental data collected simultaneously and is designed to support a range of applications; not only for the purpose of this thesis but also for more general use, by selecting the relevant data subsets.

The dataset is organized according to different experiments as shown in Figure 6.7. These experiments can be classified into two main categories: those conducted with a limited tether and those conducted with an unlimited tether. In the former case, the tether length was varied from 5 m to 15 m. In the latter, three experiments were executed: serpentine path, random path, and circular path. Each folder (experiment) contains the following data file:

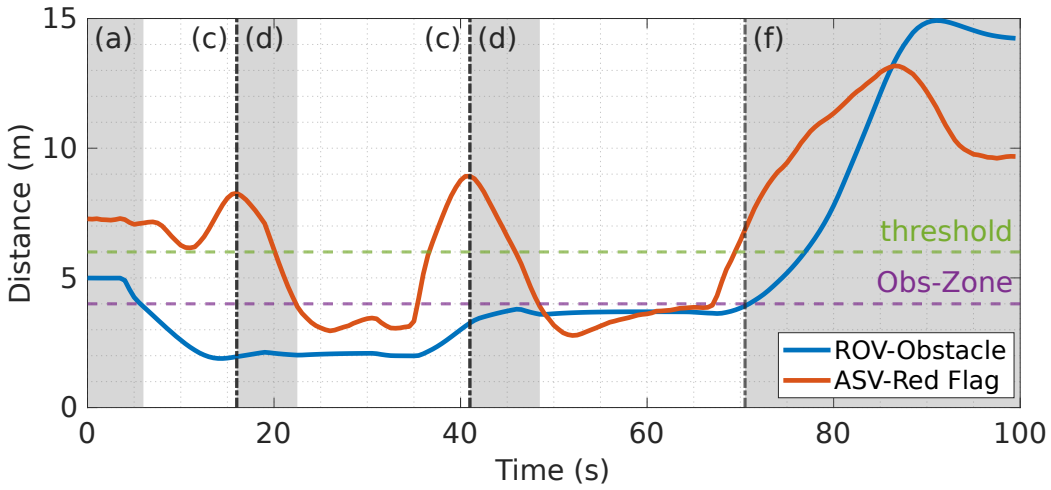


Figure 6.5 Distance variables vs. Time: The *blue* line represents the distance between the ROV and the obstacle, while the *red* line represents the distance between the ASV and the Alignment Point. The dashed *purple* line is the radius of the “Obstacle Zone”, and the dashed *green* line is the threshold for the task activation function of the “Move to Alignment Point" task. The shaded regions (a), (d), and (f) represent the corresponding cases in Figure 4.2.

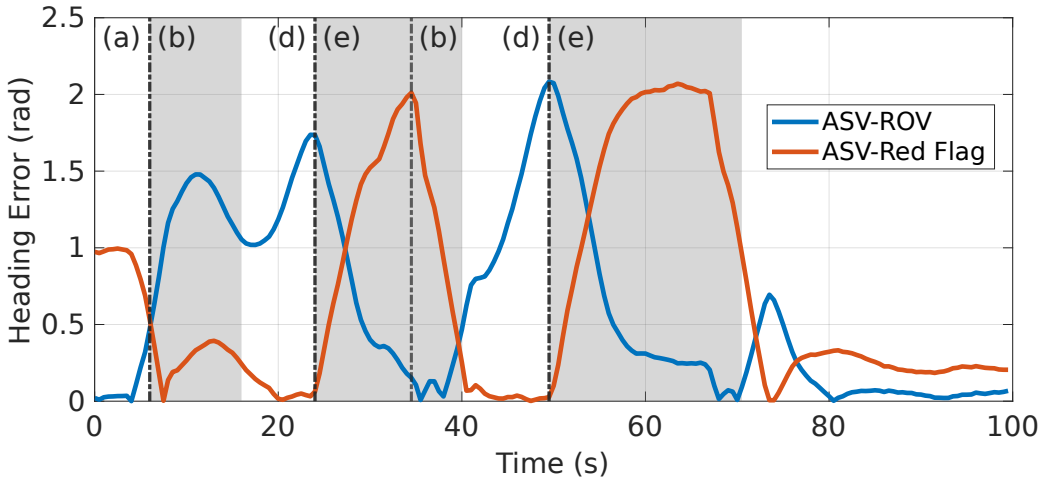


Figure 6.6 Alignment Error vs. Time: The *blue* line represents the heading error between the x-axis of the ASV and the ROV’s position, while the *red* line represents the heading error between the x-axis of the ASV and the Alignment Point position. The shaded regions (b) and (e) represent the corresponding cases in Fig. 4.2.

1. MAVLink messages from the ROV:
    - Attitude
    - Battery Status
    - IMU RAW
    - IMU SCALED
    - RC IN
    - RC OUT
    - Pressure
2. Tracking data from the mo-cap system:
    - ROV's position (x,y,z)
    - ROV's orientation (x,y,z,w)
    - ROV's velocity (linear and angular)
    - Tether's shape; which is represented by the tether's markers positions (x,y,z)
3. Tension measurements from the tension sensor
    - Horizontal force of the tether
    - Vertical force of the tether

The whole dataset is in Comma-Separated Values (.csv) format, and the first column of each file is the time.

6.2.2 Technical Validation

In this section, we present several plots generated by a MATLAB script to visualize and verify the quality of the acquired dataset. Since the data were collected from three different sources and computers, the first step was to verify the synchronization of the data after post-processing. In Figure 6.8, we plot both the ROV position acquired from the ROS topics, which was recorded online in ROS Bag, and the ROV marker's position exported offline using QTM after the recording. This comparison verifies the synchronization between the ROS bag data and the QTM data.

To verify the synchronization with the tension data, the ROV position and the tension measurements were plotted together, as shown in Figure 6.9. Knowing the position of the

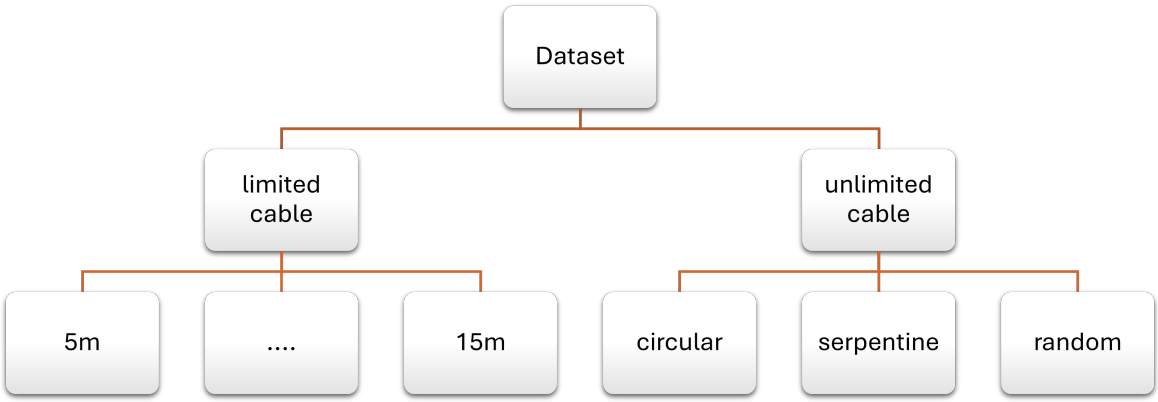


Figure 6.7 Structure of the dataset folder and its subfolders.

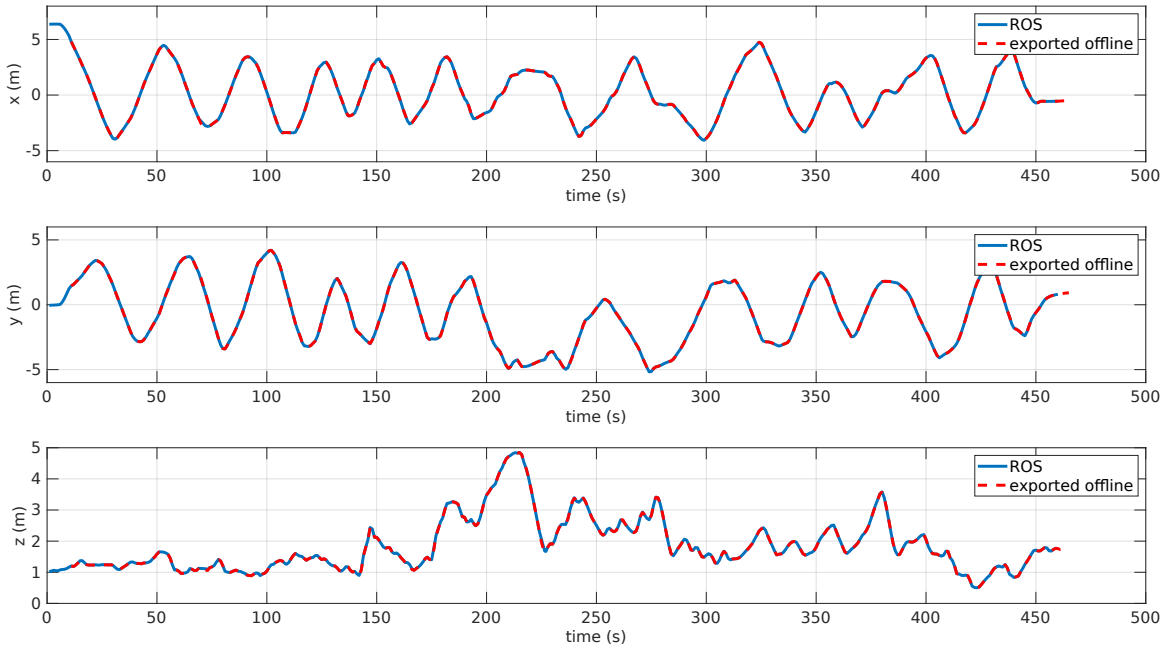


Figure 6.8 ROV's cartesian position [x,y,z] - experiment: circular path. The blue line is the position of the rigid body and is published in real-time in ROS topic. The red dashed line is the position of the marker of the origin of ROV exported by QTM after the recording.

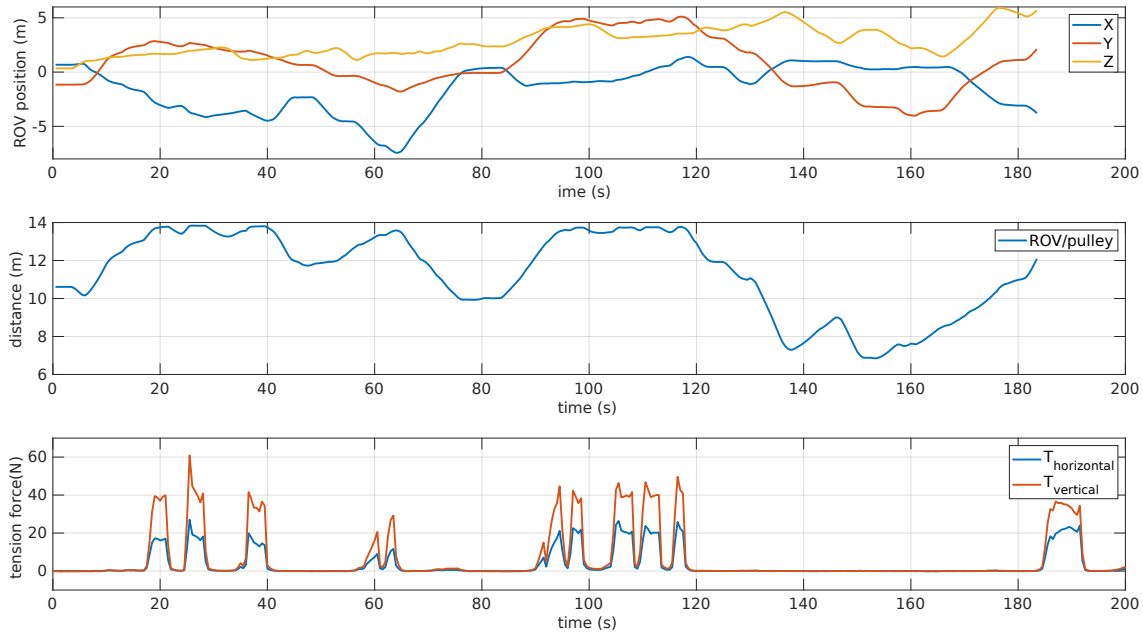


Figure 6.9 Tension vs. Distance - experiment: random motion with fixed tether length (14m).

pulley with respect to the mo-cap system frame ($x = 3.299$ m, $y = -8.510$ m, $z = 8.459$ m), the distance between the ROV and the pulley was computed and plotted, as illustrated in Figure6.9. In this experiment, the maximum tether length was fixed at 14 m; consequently, when the distance between the pulley and the ROV approaches 14 m, the tension increases. The plot in Figure6.9 spans a relatively large timescale of 200 s. However, the data were also examined at smaller timescales, confirming that the signals are well synchronized.

After verifying the synchronization of the entire dataset, various plots were generated. Starting with the tether's markers, Figure 6.10 shows the Cartesian positions of the ROV origin marker along with the first four tether markers. When the tether is vertical, the distance between successive markers is nearly constant. This is clearly visible in the third subplot (z-axis) of Figure 6.10. The discontinuous trajectories are caused by the temporary loss of marker detection by the mo-cap cameras during those intervals.

A sample of the ROV's position and velocity data is presented in Figure 6.11. These measurements are obtained online from QTM, which detects the rigid body and publishes its position and velocity to the corresponding ROS topics, as illustrated in the data acquisition diagram shown in Figure 5.11. Another interesting plot is the 3D track of the markers, reported in Figure 6.12. In this plot, the 3D trajectory of each marker is illustrated. Consequently, the continuity of each marker's trajectory was validated.

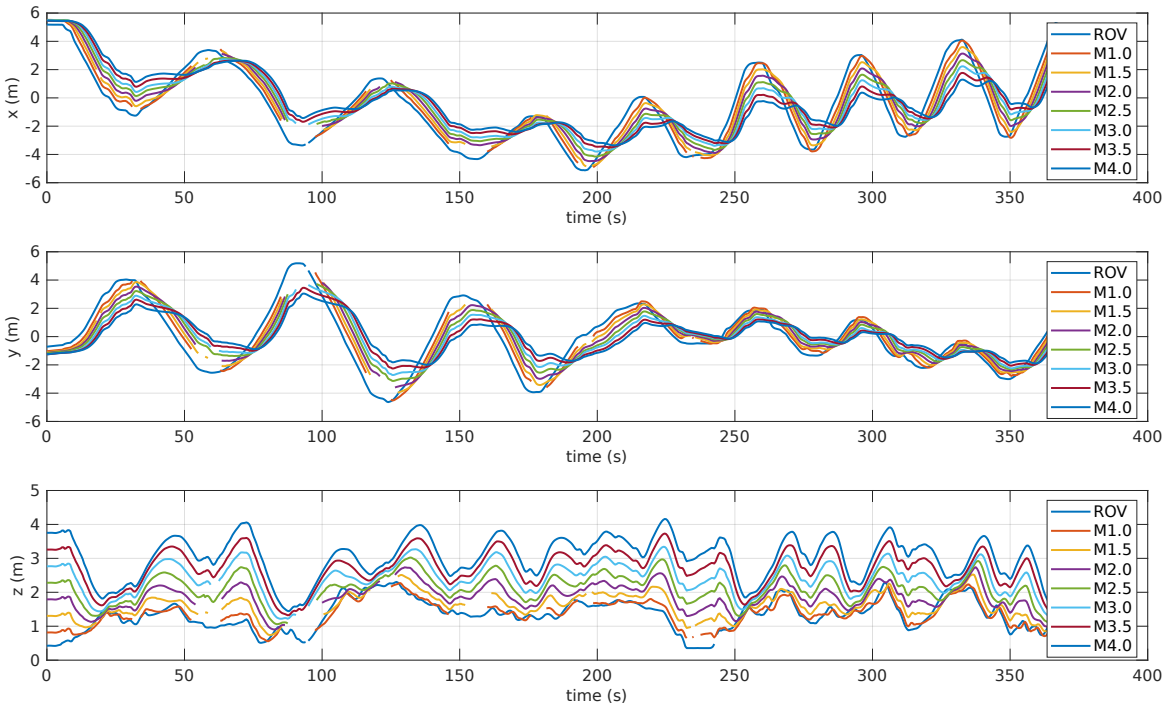


Figure 6.10 Markers' Position [x,y,z] - experiment: circle path. In the legend of the plot, MX.0 stands for the marker on the tether at X m far from the ROV.

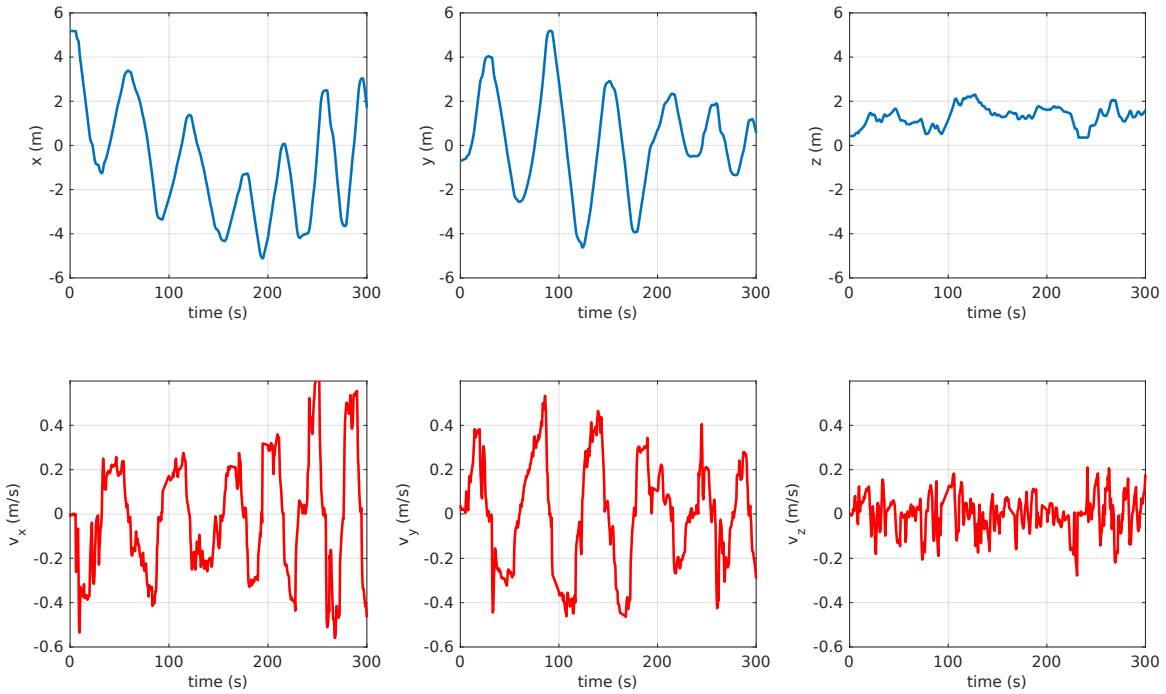


Figure 6.11 Rigid body tracking data: ROV's position and velocity. The *blue* lines represent the ROV's position, and the *red* lines represent its velocities.

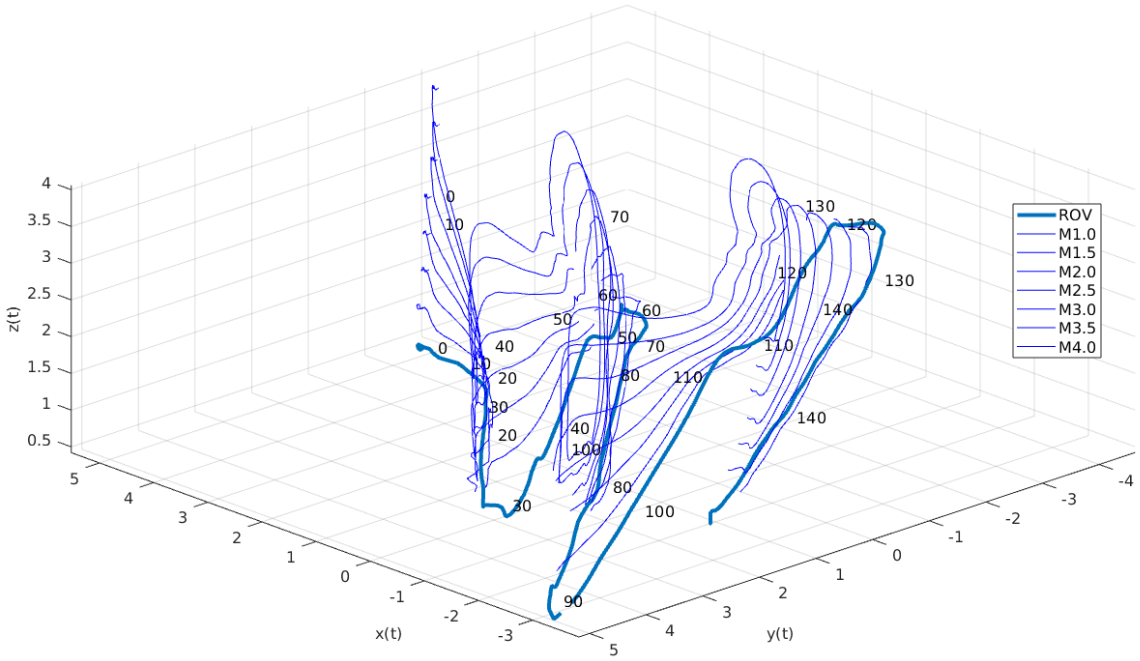


Figure 6.12 Time-labeled 3D trajectory of the markers. Experiment: serpentine path. To provide additional context, time labels are placed along the 3D trajectory at 10 s intervals. The thick blue line represents the ROV's trajectory.

Another important aspect to consider is the orientation of the ROV. After applying the yaw angle shift discussed in the post-processing section, the resulting plots were generated, as shown in Figure 6.13, where the three angles are aligned. To avoid confusion in the dataset, only the final version of the orientation — obtained from the IMU after shifting the yaw to align with the mo-cap system — was uploaded. Once the ROV orientation was corrected, we plotted the 3D position and orientation of the ROV along with its tether, as shown in Figure 6.14. The ROV's body frame in Figure 6.14 was plotted using the plotframe[3] MATLAB toolbox Abrahams (2023).

The final plot represents the tension force retrieved from the tension sensor, the battery voltage from the BlueROV2, and the distance between the ROV and the pulley of the tension. The correlation between the drop of the battery voltage and the maximum tension force is remarkable, as shown in Figure 6.15.

The dataset provides a comprehensive benchmark of synchronized measurements that can be selectively exploited depending on the specific application. In the next subsection,

[3]https://www.mathworks.com/matlabcentral/fileexchange/156419-plotframe-plot-a-3-d-cartesian-coordinate-system?s_tid=srchtitle

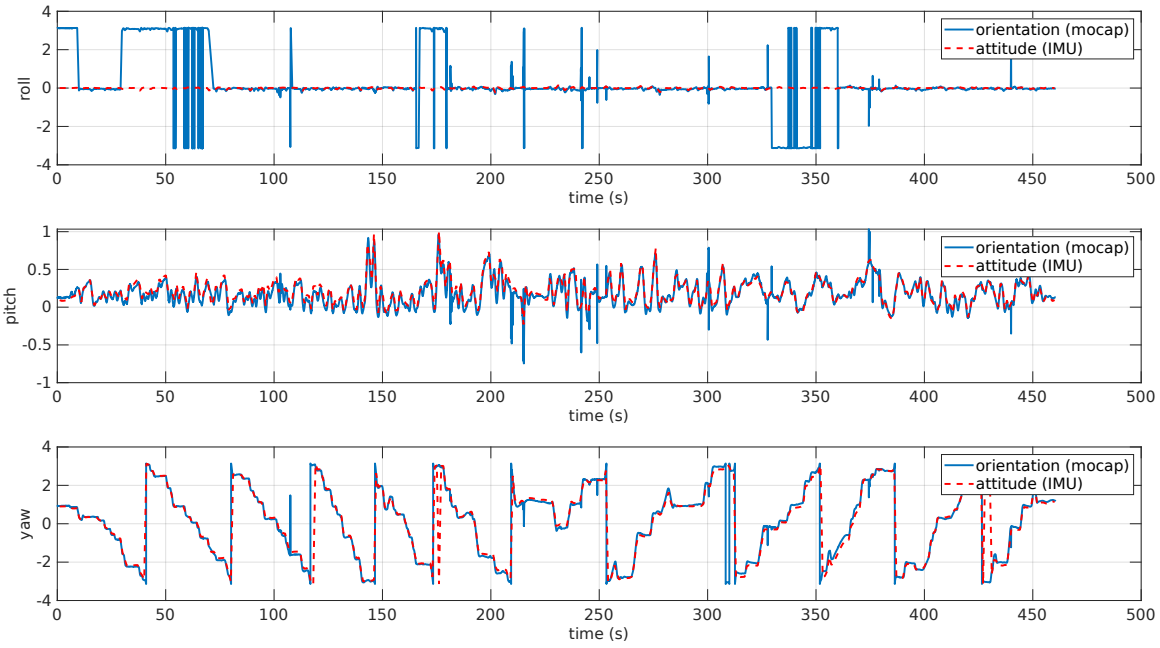


Figure 6.13 ROV orientation angles after post-processing (shifted yaw). The angles are expressed in radians. The three orientation components (roll, pitch, and yaw) are aligned.

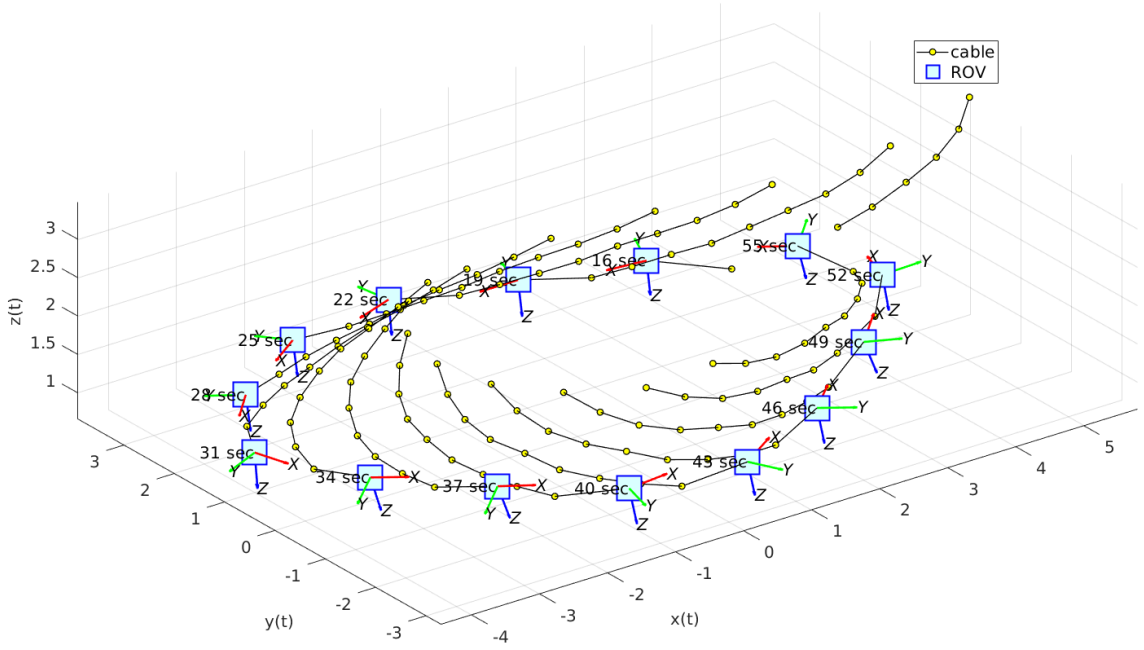


Figure 6.14 Time-labeled 3D view of the ROV and its tether (time interval: 3 s).

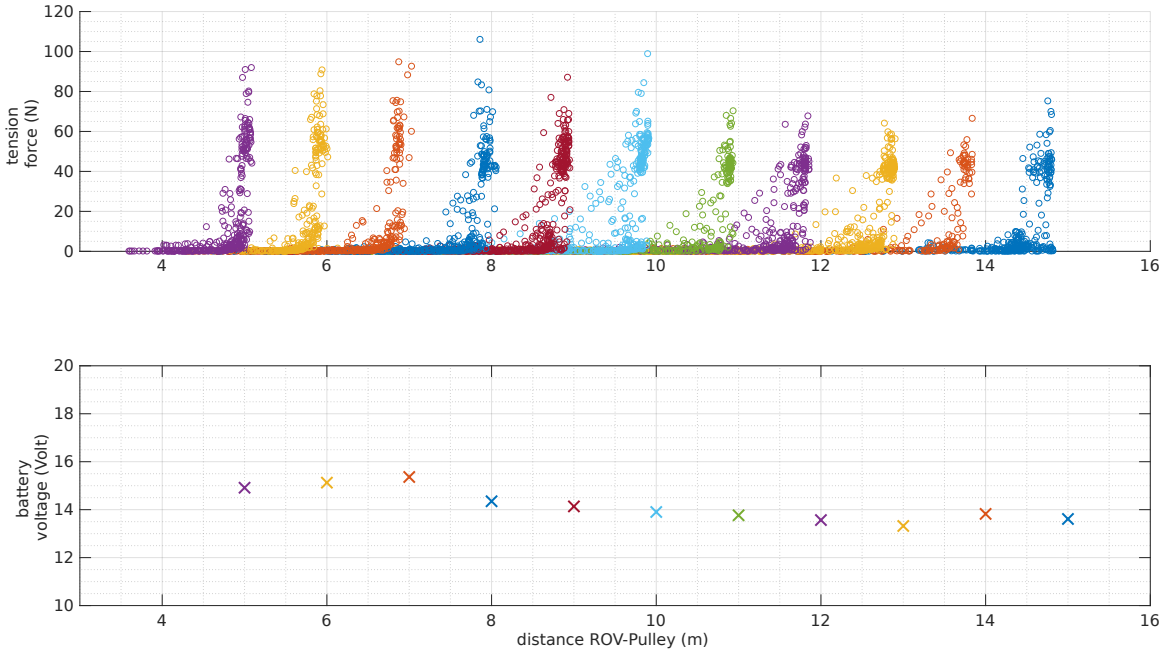


Figure 6.15 Tension force, ROV battery voltage, and the distance between the ROV and the pulley. The plot represents eleven experiments, in each experiment the tether length is fixed to a specific length $(5,6,7,..,15)$ m. Each color in the plot represents one experiment, for instance, in the *red* plot, the tether is fixed at 9 m since the maximum length is 9 m.

we discuss how this dataset was used for the tether winch control in ASV-ROV combined system.

6.2.3 Tether Length Control

In ASV–ROV cooperative vehicle systems, the TMS must release an appropriate length of cable, because if the cable is too short, it restricts the ROV's motion, whereas if it is too long, it increases hydrodynamic drag. Therefore, the reference cable length needs to be computed in real time to optimize the power consumption of the coupled vehicles.

The dataset was exploited to derive a tether model based on the measured tension, the distance between the ROV and the pulley, and the length of the released tether. A three-dimensional model was obtained through surface curve fitting, as shown in Figure 6.16. The resulting surface fit equation is given by

$$f(x,y) = a\exp(bx+cy) \tag{6.1}$$

where a, b, and c are fitting parameters. x is the cable length, y is the distance between the ROV and the pulley, and f represents cable tension. The ranges of the coefficients and the

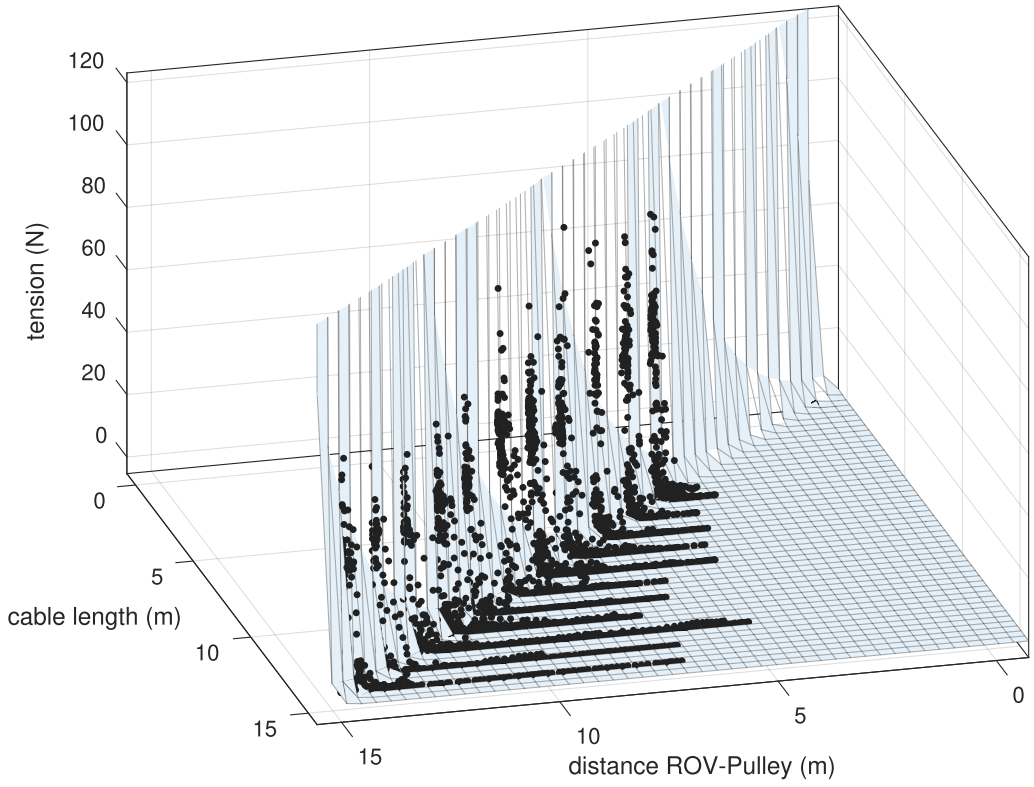


Figure 6.16 Curve fitting: a 3D plot was generated for the following data: tension, cable length, and the distance between the ROV and the tension sensor pulley. The *black* dots are the acquired data, and the *light blue* mesh is the fitted surface.

corresponding goodness-of-fit metrics are summarized in Tables 6.1 and 6.2, respectively.

The fitted surface represented in equation (6.1) is also represented as a 2D heatmap in Figure 6.17. The accepted region of cable length, which corresponds to the admissible cable lengths given the distance between the ROV and the pulley, is defined in this model. As shown in Figure 6.17, the accepted region is determined by the area bounded between two curves (the red and the pink lines in Figure 6.17). Ensuring that the cable length remains within this region guarantees that the tether is not excessively long, since maximum bounds are imposed by the upper limit (the red line in Figure 6.17), while being sufficiently longer than the distance between the ROV and the pulley to avoid constraining the ROV's motion (the pink line in Figure 6.17). Since the model is derived in an ideal condition (no sea current

Table 6.1 Fitted coefficients a, b, and c of the exponential surface model relating the ROV distance from the pulley, tether length, and tension. The table also reports the corresponding lower and upper bounds.

Coefficient	**Value**	**Lower Bound**	**Upper Bound**
a	0.4682	0.4724	0.4766
b	7.6912	7.6701	7.7123
c	-7.2578	-7.2784	-7.2372

Table 6.2 Goodness-of-fit statistics for the exponential surface model relating the ROV distance from the pulley, tether length, and tension.

Statistical Measure	**Value**	**Unit**
Confidence bounds (CB)	95%	-
Sum of Squared Errors (SSE)	2.3351×10^3	N^2
Coefficient of Determination (R^2)	0.9989	-
Adjusted Coefficient of Determination (Adjusted R^2)	0.9989	-
Root Mean Square Error (RMSE)	0.5	N

or waves), by measuring the tether tension in a real sea test, we can estimate the tension generated by sea current or the entanglement of the tether around an object.

After defining the accepted region, the model is used for the tether length controller by activating the tether drum motor only when the state (distance - cable length) is close to the accepted region's boundaries (the red and the pink lines in Figure 6.17). Figure 6.18 shows the result of one numerical simulation carried out by moving the ROV in different depths to change the vertical distance from the ASV.

The simulation starts where the ROV is 7.5 m away from the ASV at $t = 0$ s. When the ROV increases the distance to 10.5 m at $t = 20$ s, the cable length tracks the change and becomes 11.8 m. On the other hand, when the distance decreases from 11.8 m to 8 m, the cable length decreases from 13.5 m to 10.5 m, meaning that the winch controller winds the tether accordingly. In conclusion, we can see from Figure 6.18b, that the cable length is always in the accepted region thanks to the tether drum controller that winds and unwinds the cable according to the defined accepted region.

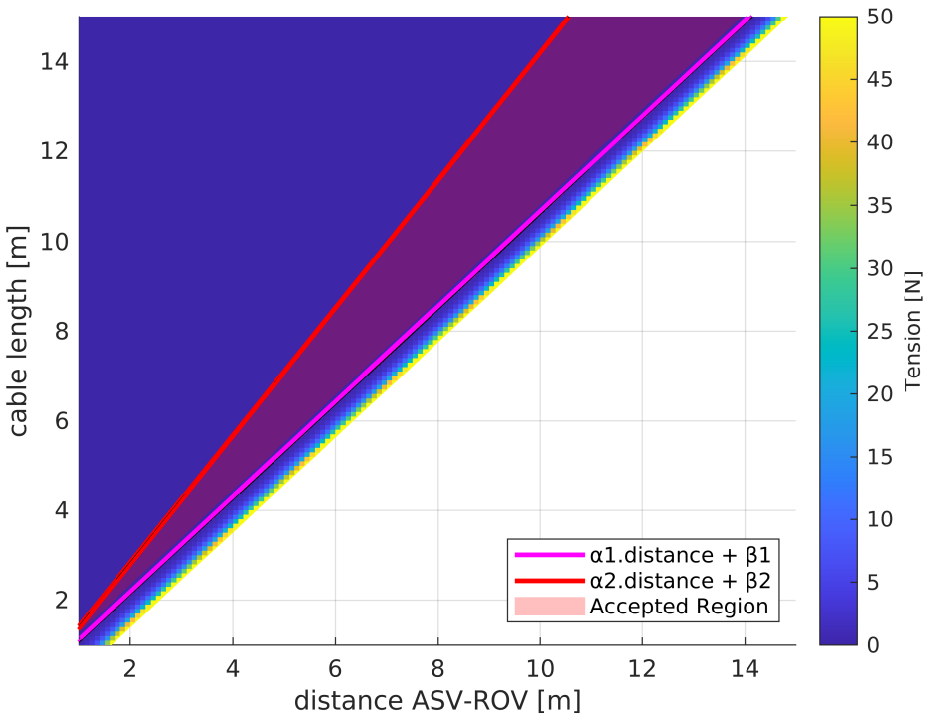
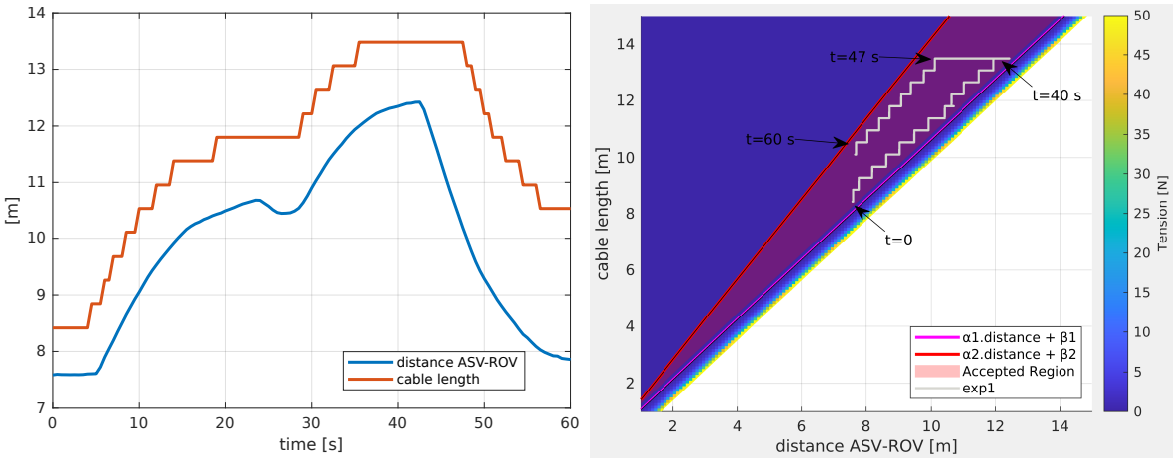


Figure 6.17 Heat map for tension model: After deriving the model of the BlueROV2 tether, the final surface equation (6.1) is plotted in 2D mode. The *red* and *pink* colors are not related to the Tension scale.



(a) (b)

Figure 6.18 A simulation study for checking the control policy of the tether. **(a)** Plot of distance between the ASV and the ROV, and the cable length with respect to time. **(b)** The distance between the ASV and the ROV vs. cable length.

Chapter 7

Conclusion

This thesis investigates control strategies aimed at reducing the internal tension within the coupled ASV–ROV system. The proposed approach focuses on coordinating the motion of the ASV and regulating the tether length to minimize tension, thereby reducing overall power consumption. By acquiring the ROV's position, the ASV computes its relative distance from the ROV and autonomously responds to variations in this distance. With respect to horizontal displacement, the ASV adjusts its position to either align with or follow the ROV. Conversely, for vertical displacement (when the ROV changes its depth), the ASV regulates the tether length by winding or unwinding the cable through the TMS.

This work concentrates on the design and implementation of novel control strategies that enable the ASV to effectively track the ROV in environments containing pillar-like underwater obstacles. Furthermore, a reference tether length is defined and translated into TMS motor commands to maintain the desired cable length within an accepted range.

The final chapter summarizes the main contributions of this thesis.

7.1 Summary

Chapter 1 introduces the thesis by addressing the challenges associated with employing the ROV as an independent platform for automating subsea intervention tasks. Based on the state of the art, the ASV–ROV configuration is identified as one of the most promising robotic systems. However, the physical connection between the two vehicles complicates their motion, particularly in environments affected by disturbances such as sea currents, waves, and wind. Following the review of the state of the art, the chapter concludes by presenting the thesis objectives and the organization of the subsequent chapters.

Before delving into the core of the thesis, essential background information is provided in Chapter 2. This chapter presents the models of marine vehicles, particularly the ASV and the ROV. Moreover, the Task-Priority method is briefly introduced to facilitate comprehension in the subsequent chapters, where this method is implemented to achieve the control policy within the ASV kinematic control layer.

In Chapter 3, the studied system is examined from a mechatronic perspective. The chapter begins with an overview of the complete system, followed by a detailed description of each robot, subsystem, and sensor involved.

After defining the system and its components, Chapter 4 presents the proposed control policy. This chapter focuses on the motion control of the ASV to ensure accurate tracking of the ROV, particularly within an offshore wind farm scenario. It includes the formulation of the ASV motion control strategy that enables the vehicle to follow the ROV without entangling the tether around underwater structures. In addition, the shape and length of the ROV tether are analyzed, and a suitable configuration is selected for the wind farm inspection use case.

Chapter 5 describes the experimental setups for both the simulation and data acquisition phases. Regarding the ASV–ROV numerical experiments, the software architecture of the ASV system is presented; this same architecture is used in the simulator since the tests are performed in a software-in-the-loop framework. The simulator configuration parameters used for the recorded experiments are also defined. Concerning the tether data acquisition process, the employed devices, sensors, and software are first introduced, followed by an explanation of the data acquisition and post-processing procedures.

Chapter 6 presents the experimental results, covering both the ASV–ROV simulations in the wind farm environment and the analysis of the acquired dataset. In the simulation section, illustrations and video links of the recorded experiments are provided, and an in-depth analysis of the ASV controller performance is conducted by presenting plots of distance and heading errors alongside the activation functions of each task. Regarding the tether experiments part, the acquired dataset is presented, accompanied by several informative plots. Moreover, a tether drum controller is proposed based on the model derived from the obtained dataset.

7.2 Main Contributions

The primary contribution of this work lies in the integration of the Task-Priority framework into the kinematic control of the ASV–ROV configuration. This integration establishes a

robust and modular foundation that enables the coordinated operation of the two vehicles while maintaining flexibility for system expansion. The Task Priority-based approach allows the definition of multiple prioritized control objectives, ensuring that high-priority tasks are consistently satisfied while lower-priority ones are executed when possible. This hierarchical structure not only enhances system robustness and adaptability but also provides an intuitive means of defining control actions in a manner that closely resembles human decision-making logic. Consequently, new behaviors, missions, or task layers can be incorporated seamlessly into the existing control architecture, making the framework scalable and easily extendable for future research or industrial applications.

From the tether study perspective, a comprehensive and experimentally validated dataset has been developed through a complex and carefully designed acquisition setup. This dataset focuses on characterizing the dynamic behavior of underwater tethers under ideal conditions. It includes measurements of key parameters such as ROV's position, velocity, tether tension, and motion profiles, which are essential for analyzing tether dynamics and validating numerical models. The dataset can serve as a valuable open resource for researchers interested in studying tethered underwater systems, facilitating model validation, simulation benchmarking, and the development of advanced control and estimation strategies.

Furthermore, a tether model based on the inter-vehicle distance, cable tension, and cable length is developed. This model enables the determination of acceptable cable lengths to drive the motor of the ROV's tether drum. In this manner, not only is the tension exerted on both vehicles reduced, but also the power consumption is minimized, as the winch is activated only when the cable length approaches the predefined limits.

7.2.1 Published Papers

During the PhD program, the following papers Khanmeh et al. (2024, 2025b,c,d) were published in collaboration with colleagues and supervisors.

References

G. Conte, G.P. De Capua, and D. Scaradozzi. Designing the NGC system of a small ASV for tracking underwater targets. *Robotics and Autonomous Systems*, 76:46–57, 2016. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2015.11.008.

Akihiro Morinaga and Ikuo Yamamoto. Trajectory planning strategies and simulation for autonomous-surface-vehicle–remotely-operated-vehicle (ASV-ROV) cooperative system. *Sensors and Materials*, 35:411–422, Feb 2023.

Nadir Kapetanovic, Dula Nad, and Nikola Miskovic. Towards a heterogeneous robotic system for autonomous inspection in mariculture. In *OCEANS 2021: San Diego – Porto*, pages 1–6, 2021. doi: 10.23919/OCEANS44145.2021.9705776.

Ikuo Yamamoto, Akihiro Morinaga, Taishi Oshino, Kyohei Kinoshita, and Yasunori Izumi. Research on ASV/ROV cooperative system with AI for seagrass distribution survey. *IFAC-PapersOnLine*, 58(20):325–328, 2024. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2024.10.074. URL https://www.sciencedirect.com/science/article/pii/S2405896324018299. 15th IFAC Conference on Control Applications in Marine Systems, Robotics and Vehicles CAMS 2024.

Chenyu Zhao, Philipp Thies, Johanning Lars, and James Cowles. ROV launch and recovery from an unmanned autonomous surface vessel – hydrodynamic modelling and system integration. *Ocean Engineering*, 232:109019, 2021a. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2021.109019. URL https://www.sciencedirect.com/science/article/pii/S0029801821004546.

Chenyu Zhao, Philipp R. Thies, and Lars Johanning. Offshore inspection mission modelling for an ASV/ROV system. *Ocean Engineering*, 259:111899, 2022. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2022.111899.

Chenyu Zhao, Philipp R. Thies, and Lars Johanning. Investigating the winch performance in an ASV/ROV autonomous inspection system. *Applied Ocean Research*, 115:102827, 2021b. ISSN 0141-1187. doi: https://doi.org/10.1016/j.apor.2021.102827. URL https://www.sciencedirect.com/science/article/pii/S0141118721002996.

Christophe Viel. Self-management of rov umbilical using sliding buoys and stop. *IEEE Robotics and Automation Letters*, 7(3):8061–8068, 2022. doi: 10.1109/LRA.2022.3187267.

Matheus Laranjeira, Claire Dune, and Vincent Hugel. Embedded visual detection and shape identification of underwater umbilical for vehicle positioning. In *OCEANS 2019-Marseille*, pages 1–9. IEEE, 2019.

Juliette Drupt, Claire Dune, Andrew Comport, Sabine Seillier, and Vincent Hugel. Inertial-measurement-based catenary shape estimation of underwater cables for tethered robots. pages 6867–6872, 10 2022. doi: 10.1109/IROS47612.2022.9981980.

Jannic S. Larsen, Esben T. Uth, Mikkel Edling, Simon Pedersen, and Jesper Liniger. Development of a novel tether force sensor for rov automation. In *OCEANS 2025 Brest*, pages 1–7, 2025. doi: 10.1109/OCEANS58557.2025.11104269.

E Simetti, G Casalino, F Wanderlingh, and M Aicardi. Task priority control of underwater intervention systems: Theory and applications. *Ocean Engineering*, 164:40–54, 2018.

Bjørn Kåre Sæbø, Kristin Y Pettersen, and Jan Tommy Gravdahl. Kinematic task-priority path following for articulated marine vehicles. *IFAC-PapersOnLine*, 58(20):65–72, 2024.

Cris Thomas, Enrico Simetti, and Giuseppe Casalino. A unifying task priority approach for autonomous underwater vehicles integrating homing and docking maneuvers. *Journal of Marine Science and Engineering*, 9(2), 2021. ISSN 2077-1312. doi: 10.3390/jmse9020162.

Daniele Di Vito, Daniela De Palma, Enrico Simetti, Giovanni Indiveri, and Gianluca Antonelli. Experimental validation of the modeling and control of a multibody underwater vehicle manipulator system for sea mining exploration. *Journal of Field Robotics*, 38(2): 171–191, 2021.

Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis, 1999. URL http://resolver.tudelft.nl/uuid:2ea31f56-6db4-475b-afe6-d146cebbd7e9.

Enrico Simetti and Giovanni Indiveri. Control oriented modeling of a twin thruster autonomous surface vehicle. *Ocean Engineering*, 243:110260, 2022. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2021.110260.

Giovanni Indiveri and António M. Pascoal. *Control of Autonomous Underwater Vehicles*, pages 1–6. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020. ISBN 978-3-642-41610-1. doi: 10.1007/978-3-642-41610-1_16-1. URL https://doi.org/10.1007/978-3-642-41610-1_16-1.

E. Simetti, G. Casalino, F. Wanderlingh, and M. Aicardi. A task priority approach to cooperative mobile manipulation: Theory and experiments. *Robotics and Autonomous Systems*, 122:103287, 2019. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2019.103287. URL https://www.sciencedirect.com/science/article/pii/S0921889019302763.

Enrico Simetti and Giuseppe Casalino. A novel practical technique to integrate inequality control objectives and task transitions in priority based control. *Journal of Intelligent & Robotic Systems*, 84:877–902, 2016.

Gianluca Antonelli, Filippo Arrichiello, Andrea Caiti, Giuseppe Casalino, Daniela De Palma, Giovanni Indiveri, Matteo Razzanelli, Lorenzo Pollini, and Enrico Simetti. ISME activity on the use of autonomous surface and underwater vehicles for acoustic surveys at sea. *Acta IMEKO*, 7(2):24–31, 2018. doi: 10.21014/acta_imeko.v7i2.539. URL https://acta.imeko.org/index.php/acta-imeko/article/view/IMEKO-ACTA-07(2018)-02-05.

Jonathan Zand. Enhanced navigation and tether management of inspection class remotely operated vehicles. Master's thesis, University of British Columbia, Department of Mechanical Enginnering, 2009.

D. Schwanz, R.E. Henke, and Roberto Chouhy Leborgne. Wind power integration in southern brazil: Steady-state analysis. pages 1–6, 09 2012. ISBN 978-1-4673-2672-8. doi: 10.1109/TDC-LA.2012.6319089.

Anthony F. Molland. Chapter 10 - underwater vehicles. In Anthony F. Molland, editor, *The Maritime Engineering Reference Book*, pages 728–783. Butterworth-Heinemann, Oxford, 2008. ISBN 978-0-7506-8987-8. doi: https://doi.org/10.1016/B978-0-7506-8987-8.00010-X.

Tomoki Sakaue, Toru Nagakita, Tomoya Kaneda, Yu Yamashita, Koju Nishizawa, Kenshi Kanbara, Hajime Hanaoka, Shu Shirai, Shiro Kikuchi, and Daisaku Uchijima. Development of usv used in underground floors surveying of the contaminated buildings at fukushima daiichi nps. In *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 224–229, 2022. doi: 10.1109/SSRR56537.2022.10018645.

Filetime. https://www.mathworks.com/matlabcentral/fileexchange/24671-filetime, 2025. [MATLAB Central File Exchange].

Juri Khanmeh, Bilal Wehbe, Enrico Simetti, and Giovanni Indiveri. Tracking data of a remotely operated vehicle and its tether using underwater motion capture system and a tension sensor. *Zenodo*, 2025a. doi: 10.5281/zenodo.17360027.

George Abrahams. plotframe: Plot a 3-d cartesian coordinate system. https://www.linkedin.com/in/georgeabrahams, 2023. URL https://www.linkedin.com/in/georgeabrahams. [https://github.com/WD40andTape/plotframe/releases/tag/1.0.0 GitHub. Retrieved December 14, 2023].

Juri Khanmeh, Francesco Wanderlingh, Giovanni Indiveri, and Enrico Simetti. Design and control of a cooperative system of an autonomous surface vehicle and a remotely operated vehicle (asv-rov). In *2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–7, 2024. doi: 10.1109/MESA61532.2024.10704912.

Juri Khanmeh, Bilal Wehbe, Francesco Wanderlingh, Enrico Simetti, and Giovanni Indiveri. Control strategies for minimizing the effect of the tension of the tether of a cooperative system autonomous surface vehicle and remotely operated vehicle (asv/rov). In *Intelligent Autonomous Systems 19 - Proceedings of the 19th International Conference IAS-19*, volume 2, 2025b.

Juri Khanmeh, Francesco Wanderlingh, Enrico Simetti, and Giovanni Indiveri. Task-based motion control of an autonomous surface vehicle connected to a remotely operated vehicle for inspection purposes. In *2025 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*, pages 363–368, 2025c. doi: 10.1109/MetroSea66681.2025.11245666.

Juri Khanmeh, Bilal Wehbe, Enrico Simetti, and Giovanni Indiveri. Tracking data of a remotely operated vehicle and its tether using a motion capture system and a tension sensor. *Scientific Data*, Dec 2025d. ISSN 2052-4463. doi: 10.1038/s41597-025-06347-0. URL https://doi.org/10.1038/s41597-025-06347-0.