ÉCOLE CENTRALE DE NANTES

UNIVERSITÀ degli STUDI di GENOVA

MASTER ERASMUS MUNDUS

EMARO+ "European Master in Advanced Robotics"

2024 / 2025

Master Thesis Report

Presented by

Ambra Ierardi

On Date August 29th, 2025

An Underwater Docking System for Sustained Presence at Sea

Jury

President:	Olivier KERMORGANT	Assoc. Professor (ECN)
Supervisor(s):	Antonio M. PASCOAL	Assoc. Professor (IST)
	Olivier KERMORGANT	Assoc. Professor (ECN)
(EMARO)	Enrico SIMETTI	Assoc. Professor (UniGe)

Laboratory: Institute for Systems and Robotics (IRS)-IST, Univ. Lisbon

Abstract

This thesis offers an integrated and comprehensive guidance and control framework for the autonomous underwater docking of a Medusa-class AUV that operates in four degrees of freedom (surge, sway, heave, and yaw), in the presence of constant but unknown ocean currents. Depending on whether the position and orientation of the docking station are known or unknown, two approach strategies are investigated: variable vehicle orientation with fixed planar speeds, and fixed AUV's orientation with variable surge and sway speeds. A reduced 4-DOF Newton-Euler formulation is used to model the vehicle's kinematic and dynamic behavior together with decoupled guidance laws for the vertical and horizontal planes, since the vehicle is considered to constantly maintain an horizontal plane orientation. For each of the four path-following phases (ship-based approach, beacon-based homing, circumnavigation, and terminal docking), an outer-loop guidance system calculates reference velocities and yaw commands for the AUV. To reduce current-induced drift, a smooth blending of circular and straight trajectories is incorporated. In order to achieve specific damping and natural frequency characteristics, the inner control loop uses cascaded PI and PID controllers with anti-windup to track desired surge, sway, depth and heading. These controllers are designed via linearization of the simplified dynamic model about selected working points. State estimation makes use of Kalman and Extended Kalman filters: a planar and vertical KF used during the ship-frame approach as well as the iUSBL-relying beacon-based homing phase; a unified EKF with range measurements and current velocity observations exploited during the bottom docking phase, eventually coupled with a specialized EKF to recover, when needed, docking-station orientation from dual-transponder acoustic data. The suggested methodology is validated by simulation results, which show satisfactory performance and good rate of convergence to the dock under all examined uncertainty scenarios, despite the simplified model used for the plant in the design of the controller, the noise applied on the physical parameters of the same, and the significant underestimation of the process and measurement noises in the estimation filter.

Acknowledgements

I want to express my deepest gratitude to my family, my mother and my father, for their never ending love, patience, and constant support during this journey. Your encouragement has been my anchor in moments of despair and my greatest source of strength, even this past year when thousands of kilometers were parting us.

My most sincere thanks go to my friends, who have been there to cheer me up and make me laugh even during the most difficult times, making those moments lighter.

I am especially grateful to my university colleagues, mates during this journey which began in Genoa, almost two years ago, and brought us to Nantes, with whom I have shared not only academic challenges but also countless memorable moments, becoming my family far from home. Their collaboration and friendship have enriched my experience inside and outside the university walls.

I owe special thanks to my supervisors. To Professor Antonio Pascoal, for being a guide throughout the entire process of this thesis. Thank you for your patience, generosity with your time, and for guiding me through the challenges I had to face during these six months, step by step. Working under your supervision has been both a privilege and a joy. To Professor Enrico Simetti, for your insightful feedback and readiness to support me even from another country, as well as the solid knowledge you provided during the first year of the master. Your remote guidance has been a precious help and a source of encouragement. And to Olivier Kermorgant, for the valuable teachings you provided during the past semester at École Centrale, which have been particularly useful in shaping and enriching the work presented in this thesis.

Notation

Symbol	Description	Units
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Position in inertial frame (NED)	m
ϕ, θ, ψ	Euler angles: roll, pitch, yaw	rad
u, v, w	Components of linear velocity in body frame (surge, sway, heave)	m/s
p, q, r	Components of angular velocity in body frame (roll, pitch, yaw rates)	rad/s
$\boldsymbol{\nu} = [u\ v\ w\ p\ q\ r]^\mathsf{T}$	Total velocity vector in body frame	m/s (linear), rad/s (angular)
$\boldsymbol{\eta} = [x\ y\ z\ \phi\ \theta\ \psi]^\mathsf{T}$	Pose vector in inertial frame	m (position), rad (angles)
$\boldsymbol{\tau} = [\tau_u\ \tau_v\ \tau_w\ \tau_r]^\mathsf{T}$	Control input vector	N (forces), N·m (moments)
M_{RB}	Rigid-body mass matrix	kg, kg·m, kg·m^2
$C_{RB}(\boldsymbol{\nu})$	Rigid-body Coriolis/centripetal matrix	kg/s, kg·m/s, kg·m^2/s
M_A	Added mass matrix	kg, kg·m, kg·m^2
$C_A(\boldsymbol{\nu})$	Added-mass Coriolis matrix	kg/s, kg·m/s, kg·m^2/s
$D(\boldsymbol{\nu})$	Hydrodynamic damping matrix	kg/s, kg·m/s, kg·m^2/s
$\mathbf{g}(\boldsymbol{\eta})$	Restoring force/moment vector (gravity and buoyancy)	N (forces), N·m (moments)
$\mathbf{v}_c = [v_{cx}\ v_{cy}\ v_{cz}]^\mathsf{T}$	Ocean current velocity vector	m/s
ψ_d	Desired yaw angle	rad
β	Desired trajectory orientation angle	rad
$R(\psi)$	Rotation matrix (body $\rightarrow$ inertial)	–
Δt	Sampling time step	s
$\hat{x}$	Estimated value of variable x	same as x

Acronyms

AUV Autonomous Underwater Vehicle

DOF Degree-of-Freedom

ILOS Integral Line-Of-Sight

LOS Line-of-Sight

ROV Remotely Operated Vehicle

INSTITUTES

KAIST Korea Advanced Institute of Science and Technology

MBARI Monterey Bay Aquarium Research Institute

SLAM Simultaneous Localization And Mapping

WHOI Woods Hole Oceanographic Institution

Abbreviations

2D 2-Dimensional

3D 3-Dimensional

ADCP Acoustic Doppler Current Profiler

DS Docking Station

DVL Doppler Velocity Logger

EKF Extended Kalman Filter

FSM Finite State Machine

GA Genetic Algorithm

GNSS Global Navigation Satellite System

iUSBL Inverted USBL

KF Kalman Filter

LCLR Line Capture Line Recovery

LTI Linear Time-Invariant

PF Path Following

PID Proportional-Integral-Derivative

PI Proportional-Integral

PSO Particle Swarm Optimization

SMC Sliding Mode Control

SRG Speed-Regulated Guidance

USBL UltraShort BaseLine

USV Unmanned Surface Vehicle

List of Figures

List of Tables

Contents

Introduction

For long-duration missions, autonomous docking of underwater vehicles is essential because it allows for in-water maintenance, data transfer, and recharging without the need for human intervention. In dynamic marine environments, recent studies have focused on accurate guidance, strong control, and trustworthy state estimation [14], [15]. This thesis tackles the docking issue for the Medusa AUV from a numerical point of view, allowing to extend the possibility to use this algorithm to other vehicles with the same dynamic equations, just by modifying the physical data relative to the single craft. The object of study is a research platform created at Instituto Superior Técnico, and is subject to continuous unknown ocean currents, considered constant since usually, in the examined environment, they are changing much slower than the AUV's dynamics, and operates in four active degrees of freedom: surge, sway, heave, and yaw.

Launched from a surface ship, the vehicle will navigate itself to a stationary docking station (DS) that has dual acoustic transponders and an inverted USBL (iUSBL). Two main guidance strategies are examined: one considering fixed orientation to match the DS and modifying planar velocities to compensate for currents, and the second one maintaining fixed surge and sway speeds while adjusting heading in real time, until the final homing part, where the yaw is adjusted according to the value required to perform a proper docking operation. In both cases, the depth is controlled in a decoupled way. Different homing behaviors result from evaluating three knowledge scenarios: approximated knowledge of the whole pose of the DS and partially unknown DS position or orientation.

By taking advantage of the AUV's built-in decoupling for vertical and horizontal motion characteristics, the docking scheme simplifies considerably. A four-phase path-following sequence is established: guide the AUV toward the DS while proceeding in the ship frame; proceed in the beacon-referenced homing phase, using iUSBL measurements in relation to the DS or switching to range, depending on the available sensor, going in a straight line until a threshold distance is reached; finally, enter the docking opening, or circumnavigate prior the docking phase in case the AUV was outside the DS field of view.

In order to smoothly enter the funnel-shaped dock, terminal homing involves executing a blended circular-straight trajectory.

A Newton–Euler 4-DOF formulation is used to model the vehicle dynamics. It incorporates rigid-body, added mass, hydrodynamic damping, and restoring forces, and it treats constant currents as disturbances. By linearizing the reduced version of the dynamic equations about reference working points, inner-loop controllers - PI for surge/sway and PID for depth/yaw - are designed to target specific closed-loop poles, through damping ratio and natural frequency. Under actuator limits, integrator saturation is avoided by an anti-windup scheme. A cascade of KFs and EKFs is used to estimate the state accurately in each phase. In order to enable alignment for docking, a specialized EKF recovers the DS orientation from the differential acoustic range to two transponders, in case of missing accurate information on the data.

Phase transitions and filter re-initialization are handled with a state-machine logic in a custom MATLAB/Simulink environment, validating this integrated guidance, control, and estimation framework.

The effectiveness of the suggested techniques is examined through statistical analyses, via the implementation of the Monte Carlo method, providing simulation results that show dependable convergence to the docking station in most uncertainty scenarios, including the simplified model and physical parameters uncertainty, on which the controllers are tuned, as well as underestimation of the covariances of process and measurement noises in the estimation filters, motivating further improvements in the algorithm, which looks promising.

Chapter 1

Big picture

The oceans cover over 70% of the Earth's surface [16], [17] and are fundamental to global ecosystems, regulating climate through precipitation and wind patterns, sustaining marine biodiversity, and supplying vital resources for humanity [18], [19]. Yet more than 80% of the seafloor remains unmapped [20], largely because traditional oceanographic research relies on large, costly research vessels that are logistically complex and pose safety risks to their crews.

Autonomous Underwater Vehicles have emerged as a promising, cost-effective alternative: their small size and modularity allow exploration in regions inaccessible to larger ships. However, AUVs rely on onboard electromechanical power - severely limiting mission duration to battery capacity. On the other hand, Remotely Operated Vehicles, while able to linger indefinitely, demand constant surface-ship support. Seafloor mapping is more than a mere scientific curiosity: has wide potential applications for the monitoring of our environment, the prediction of natural disasters and management of resources. The high-resolution data for great depths could, for example, be used to follow tectonic activity or model ocean circulation and at even greater distances it can provide monitoring of marine habitats threatened by human activities or climate change. Furthermore, autonomous exploration has immediate industrial implications ranging from subsea infrastructure inspection and underwater energy to environmental monitoring with a reduced need for human interfacing in order to bring down operational costs. To bridge this gap, underwater docking stations are being developed to let AUVs autonomously recharge and exchange data in situ, enabling extended and more flexible underwater missions.

Motivation

The ability for AUVs to autonomously extend their mission timelines through underwater docking is a critical enabler of persistent ocean monitoring and exploration. Nonetheless, autonomous docking has been a bottleneck due to the unreliable performance even in recent state-of-the-art solutions, particularly in uncertain underwater areas without GPS with sensor constraints and parameter uncertainties leading to challenges of control and estimation under currents, as well as non-linear system dynamics.

Docking stations help to overcome the fatigue of that limit by requiring an AUV to three-dimensionally detect and localize an unknown DS with only approximate position or orientation information; autonomously navigate in currents and environmental uncertainties through a funnel-shaped entrance of the DS without manual intervention from human; smoothly transit among different guidance phases such as long-range approach, field-of-view acquisition, circumnavigation and terminal homing while attaining stability and accuracy

In this thesis, these challenges are tackled for the MEDUSA AUV developed at LARSyS/DSOR (IST, Lisbon), while keeping the flexibility of a numerical approach. Focusing on a four degree-of-freedom model (surge, sway, heave, yaw), two primary docking scenarios are examined: one with variable orientation ψ and fixed planar speeds, where the AUV fixes its surge

and sway velocities and adapts yaw continuously to follow a trajectory toward the docking funnel, before changing policy allowing for a proper docking orientation; the other one with fixed orientation and variable planar speeds, where the DS orientation is known a priori but its position is approximate, and the AUV maintains a constant heading, according to the one of the docking station, while adjusting surge and sway speeds in real time to account for currents and path geometry.

These two scenarios allow for the evaluation of guidance performance under different operational constraints and environmental uncertainties, supporting greater flexibility.

To ensure robust performance, a two-layer control architecture is designed and validated: an outer-loop guidance system, that generates the desired velocity and orientation commands for each docking phase, and an inner-loop control layer (PI/PID with anti-windup), that tracks those references and compensates for disturbances.

Moreover, Kalman and Extended Kalman filters are integrated for online estimation of DS relative position, orientation and local currents, allowing the AUV to switch seamlessly between guidance phases and achieve precise, repeatable docking without surface support.

This work contributes to the development of fully autonomous AUV systems capable of long-duration, unattended missions, which is a key to scalable, cost-effective underwater operations in science and industry.

Chapter 2

State of the Art and Key Contributions

In the following section, a detailed picture of the different technologies used in the docking of Autonomous Underwater Vehicles is given, describing the possible configurations of the Docking Station, the different categories of AUVs, the various types of navigation and docking systems. Furthermore, the section includes a review of most important and recent solutions proposed by a variety of research institutes and universities from all around the world.
A typical scheme of the main elements necessary to perform autonomous navigation and docking is provided by [1], in figure 2.1.

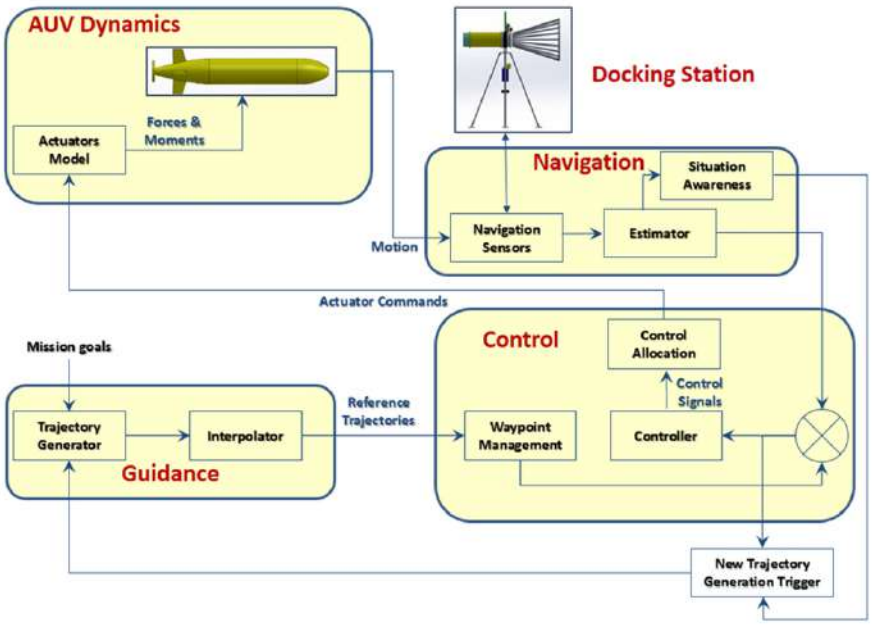


Figure 2.1: Typical AUV guidance, navigation and control scheme [1].

The principal blocks of interest are guidance, navigation and control.
The first one produces the desired velocity and attitude based on the current state and the goal position produced by the navigation system, the AUV's dynamics and the goals of the mission [1]. This block can work either online or offline: in the first case there is a continuous update of the DS's position and heading, so, in case of free floating or mobile Docking Station, it is still possible to detect the exact coordinates the vehicle should aim to perform docking, and compute the trajectory to reach that point; in the second case, the DS's pose and position are never updated, they're considered to be fixed and it is not computed any upgrade of the trajectory to follow, so this operating mode is suitable only for fixed stations [1].
The navigation block exploits sensors and an estimating subsystem, such as KF or EKF, to

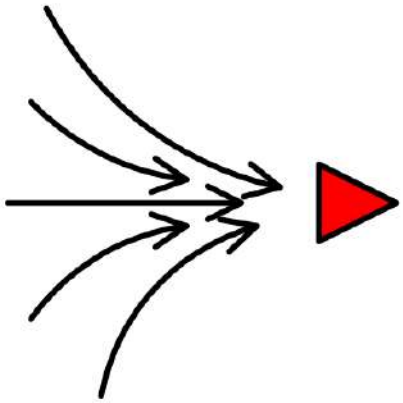
predict state variables and, if possible, also ocean currents and obstacle presence and positioning [1].
The control system produces the commands for forces and torques which are then processed by a thruster allocation system, to compute the desired activity for the thrusters, according to its state and the DS's information, and the velocities and orientations generated by the guidance block [1].

2.1 Docking Station configurations

In order to achieve the goal of safe docking of the AUV, two main configurations are used: unidirectional and omnidirectional, depending on the possible directions the AUV can move along to get attached to the DS [21], [2].

2.1.1 Unidirectional DSs

The unidirectional docking station is the the most common type of docking station to accommodate AUVs with a torpedo-like shape, which have to enter the station from an horizontal direction [1], as it is possible to notice from figure 2.2. This solution requires the knowledge of the orientation, as well as the position of the DS, so it's more demanding in terms of resources and computation, but it does not require any extremely complex mechanism on the nose of the AUV.



(a) Schematic of unidirectional docking.

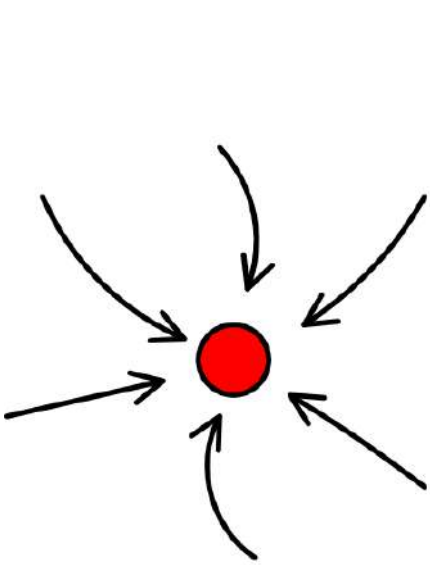
(b) Model of unidirectional docking.

Figure 2.2: Unidirectional docking stations [2].

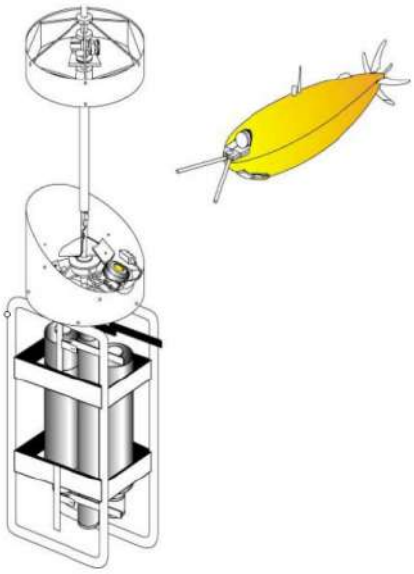
2.1.2 Omnidirectional DSs

The omnidirectional docking stations are vertical structures made of cables under tension or rigid poles, see figure 2.3, to which the AUV will dock thanks to a latching device mounted on its nose. Poles are generally used in case of fixed stations, while the other one is more suitable for mobile docks [1]. These configurations present the advantage of requiring only their location, not the orientation, but they require a complex mechanism on the AUV, especially challenging in smaller vehicles, to implement sonars to find the location, to communicate and to charge the battery as well [22],[23].

Another distinction is made depending on whether the DS is rigidly fixed to the seafloor or not, being free to float, like a buoyant, or mobile, towed to a ship [1].



(a) Schematic of omnidirectional docking.



(b) Model of omnidirectional docking.

Figure 2.3: Omnidirectional docking station [2].

2.1.3 Fixed DSs

Fixed structures, such as those shown in Figure 2.4, do not move with currents, so their location is always the same and the AUV doesn't strictly need to update that information, but they're more expensive to install [3].

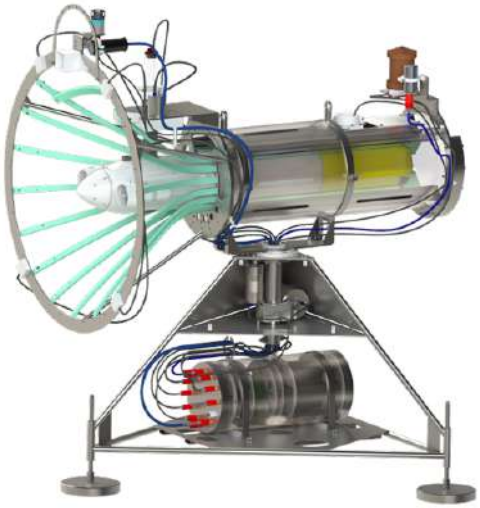


Figure 2.4: Fixed docking station [3].

2.1.4 Mobile and free floating DSs

These other solutions are sensitive to water current, so the AUV needs necessary an update on the position and pose of the DS, but they are easy to move, in order to change their location. Due to this non-trivial drawback, the most common docking stations are the fixed ones [4]. An example of a free floating DS is shown in Figure 2.5.

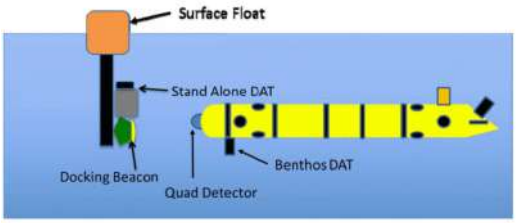


Figure 2.5: Free floating docking station [4].

2.2 Classifications of AUVs

Autonomous underwater vehicles are indispensable and self-sufficient instruments for underwater exploratory activities and operations. These autonomous intelligent mobile robots can perform complex tasks without human guidance and cope with a variety of challenges of working in a marine environment. AUVs have been categorized mainly on the following aspects: their design, size, operational capabilities, mission focus, and propulsion systems.

2.2.1 Classification by Design and Structure

The most common shape is the **Torpedo-like**, which are developed along their longitudinal axis, making them efficient for the forward motion, especially in case of missions requiring the AUV to travel long distances. Some examples are the REMUS series and the Bluefin Robotics AUVs, widely used for seabed mapping and environmental monitoring [24]. Another common design is the **Glider**, again developed along the longitudinal axis and ideal for long-term oceanographic expeditions, they are built exploiting buoyancy changes as propulsion, hence they are efficient in terms of energy. Some examples are the Slocum Glider and the Seaglider [25]. **Hover-capable AUVs** are vehicles capable of precise maneuvering, such as the ISE Explorer,useful in case of structure expeditions and close-range surveys [26]. Finally, **Hybrid AUVs-ROVs** combine the autonomy and the tethering system, providing versatile robots, useful for inspections and interventions [25].

2.2.2 Classification by Size

The size of an AUV determines the range it can cover, its payload capacity and in which situations it can be employed. **Large** AUVs, such as the Boeing Echo Voyager, are used in long missions and heavy tasks at significant depths [25]. **Medium-sized** AUVs are suitable for surveys close to the shores and similar mid-distance applications; an example is brought by the Bluefin-12 AUV [27]. **Small and portable** AUVs are lightweight solutions for effective short-range missions. An example is REMUS-100 [21]. As last, **Micro and miniature** AUVs, like the RoboClam, are specialized for tasks requiring a very accurate performance, such as biological sampling. It is just an example of autonomous fluid vehicle, being the fluid non properly seawater but human blood, they are still required to move in a fluid medium [26].

2.2.3 Classification by Operational Capability

Depending on the environment where the AUV has to work, different operational capabilities are required. In case the AUV has to work in **Shallow waters**, it will be designed for coastal zones, allowing it to be able to handle tasks such as harbor checking or environmental monitoring [28]. For operations in **Deep sea**, AUVs must be capable to withstand extremely high pressures, such as WHOI's Nereus [29]. As last, to explore extreme environments such as

the Arctic it is required an **Ice-capable** AUV model, such as the Autosub, which is tailored for polar operations, to test, for instance, ice thickness [28].

2.2.4 Classification by Mission Type

Depending on the specific mission the AUV must accomplish, different kind of robots are to be be used. **Survey** AUVs are useful for seabed mapping and habitat surveys, thanks to the high-resolution sonars and cameras equipping them [30]. In case an **Inspection** is to be performed, an AUV such as the Eelume is appropriate, since it is tailored to inspect pipelines and underwater structures [25]. For **Military** purposes, AUVs such as the Knifefish are the best choice, to perform mine detection [26]. To carry out **Scientific Researches**, it is required an AUV suitable for environmental monitoring, in order to study ocean currents and climate change [25]. In the end, **Commercial**-use AUVs are used by offshore industries for tasks such as rig inspections and, again, pipeline monitoring.

2.2.5 Classification by Propulsion Type

Based on the type of engine used to move the AUV, the efficiency and applicability of the AUV changes. AUVs driven by **Propellers**, like REMUS series, use traditional propellers, and are useful for general-purpose missions [27]. AUVs exploiting **Buoyancy** are efficient in terms of energy, because they rely on buoyancy changes, hence they allow extended missions [25]. **Bio-Inspired** AUVs like the RoboTuna, mimic marine life, being stealthy and efficient, suitable for specific tasks, but less common [25].

2.3 Docking Navigation system

If the docking system is free to float or towed to a ship, it is necessary to have a good communication system between it and the AUV, as well as proper navigation sensors. The mainly used navigation systems are acoustic, optic and electromagnetic based [1].

2.3.1 Acoustic-based Navigation systems

This method is based on the propagation of acoustic waves through water, in order to measure the distance and orientation between the AUV and the DS [1]. Usually, a transceiver is mounted on the AUV and emits a sound wave, to which the DS's transponder sends back a reply message. The distance and angle are computed based on the travel time of the wave and the main direction of the incoming sound wave. If the used technique is inverted ultra-short baseline (iUSBL), the returned signal is received on multiple transceivers, accurately separated, mounted on the AUV. The leaning range, i.e. the distance between each transceiver and the transponder, is calculated based on the travel time [8], [31], [7], [32], [33], [34].

2.3.2 Optical Navigation systems

This operational method requires multiple light sources on the Docking Station together with a vision system on the AUV, such as a quadrant detector or something more sophisticated as a camera and an image processing algorithm [1]. The drawback of this method is the sensitivity of the system to light itself: in case the DS is too close to the surface of the water in a sunny day processing the image might be too challenging, and so impossible to find the exact position of the opening of the dock. On the opposite side, if water is not clear it might be difficult to distinguish the light source [35].

2.3.3 Electromagnetic Navigation systems

This method is more robust with respect to the optical one [1], for any oceanographic condition, since it records electromagnetic waves generated from coils on the Docking Station, thanks to other coils on the AUV. This system is quite accurate, ensuring 20 cm accuracy, but electromagnetic waves can be sensed at up to 25-30 m of distance, due to the water properties [6].

2.4 Docking Guidance system

Guidance systems, or guidance laws [1], are the blocks capable of generating feasible and applicable velocity and orientation commands based on the information provided by the navigation system, the system providing the sensor information, in line with the mission objectives, to enable an AUV to move from the initial position to the docking station. Guidance systems can be subdivided into two main types: local and global methods, depending on the range of planning.
The docking process can be mainly divided into two steps [1]: the homing stage and the docking stage. In the first stage, the guidance system moves the vehicle from a far location (which might be between 1 km and 100 meters) to a position close to the docking station (around 20 meters to 10 meters). In the second and final stage, the guidance system computes the trajectory to guide the vehicle into the DS entrance and locks the vehicle into it.
There are two main categories of guidance methods commonly used for the homing and docking procedures [1].

2.4.1 Point-to-Point guidance laws

Point-to-point guidance laws are typically simple in design and operate by pointing the vehicle directly towards the DS. These methods are commonly used during the docking stage and are more suitable for docking operations where the approach can be made from any direction, such as docking with a pole-shaped DS. The simplest of these are the classic guidance laws like line-of-sight (LOS) [27], linear-terminal guidance [9], and pure-pursuit guidance [36]. The first one involves a reference point in addition to the interceptor and the target, in this case the DS, and it is supposed to achieve an intercept by constraining its motion along the line of sight between the reference point and the target [5]. The pure-pursuit guidance, instead, requires that the interceptor is supposed to align its velocity along the line of sight between the interceptor and the target [5]. A visual example can be seen at 2.6.

2.4.2 Optimization-based guidance laws

Guidance laws belonging to this category, unlike point-to-point guidance systems, can be applied to both homing and docking stages. The goal of these methods is to generate a geometric 3D path along with velocity and acceleration commands, to improve maneuverability. The main drawback of these methods is the computational load, which makes them less suitable for real-time docking operations. Examples of these methods include AUV path planning using evolutionary algorithms, such as genetic algorithms (GA), particle swarm optimization (PSO), and others [37], [38], [39], [40], as well as trajectory generation techniques like indirect/direct methods developed from optimal control theory [41], [42].

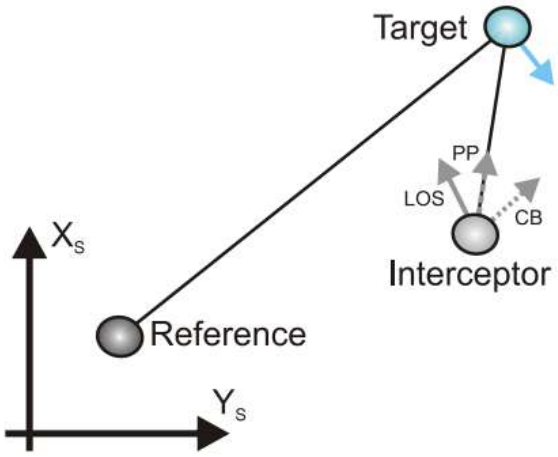


Figure 2.6: Comparison of velocity vectors of LOS and PP guidance laws [5].

2.5 Autonomous Underwater Docking solutions

In this section it is performed a review of some interesting solutions proposed by recent research studies, concerning AUV docking [1]. Usually, researches are performed in fixed, non-changing environments, since variable field conditions produce less reliable results in terms of accuracy.

In the studies on autonomous underwater docking conducted by the Oceanographic Systems Laboratory at the Woods Hole Oceanographic Institution (WHOI) of 1997 [24], a REMUS AUV was employed to dock into a funnel-shaped docking station (DS). The experiment accurately addressed all aspects of the docking problem, starting from the mechanical design of the DS, going through the battery charging and communication circuits design, the AUV navigation system and docking guidance development. The AUV docking process was divided into several steps. The first one involved the REMUS maneuvering, which consisted of a straight motion followed by a turn to position it 50 meters away from the DS, with the vehicle pointing directly into the DS funnel. In the second step, the AUV attempted to follow a reference path along the DS centerline using a path-following guidance law. In the final part, the AUV reached a 2-meter threshold in front of the DS. At this point, it straightened the fins and maintained constant thrust for 15 seconds, in order to enter through the DS funnel. The experiment exploited a USBL navigation system with a tolerance accuracy of 1-3° and an acoustic range of 2 km. The methodology employed did not account for the AUV's dynamic model, the DS's geometric model, or the effects of current disturbances. To minimize the impact of current disturbances, the DS was installed to face the current. The initial success rate was recorded at 21%, but it was later improved to 91% after further system improvements [4],[32].

The Naval Command, Control and Ocean Surveillance Centre conducted an experiment using an Odyssey IIB Autonomous Underwater Vehicle (AUV) for optical terminal guidance. The AUV was equipped with a camera mounted on the nose and the Docking Station with an attached light source [43]. A pure-pursuit guidance system pointed the vehicle towards the light source at a constant speed between 1 and 1.5 m/s. Decoupled PID controllers managed in independent ways depth and heading control. The experiment achieved an acquisition range of 20-28 m with a positioning accuracy of approximately 1 cm. The influence of ocean currents was considered negligible in the proximity of the DS and was not taken into account in the experiment. The main limitation of this approach was the significant impact of water transparency on the achievable docking range. Furthermore, the system's performance was susceptible to interference from sunlight in shallow water environments, as it has already been considered in

the previous sections [1].
An electromagnetic navigation terminal guidance test with the Odyssey IIB AUV and funnel-shaped docking station was performed by Monterey Bay Aquarium Research Institute (MBARI) [6]. The coils installed on the docking station emitted horizontal and vertical magnetic fields, as it is possible to notice in figure 2.7, while the ones on the AUV were the receiving components of the system. The pure-pursuit guidance system controlled the maneuvers of the vehicle while a decoupled PID stable control loop was maintained [1]. The first step was a straight path, lasting 60 s, away from the docking station, as described in the experiment, followed by a 180° turn to be aligned with the docking station centerline. The AUV then traveled towards the docking station using dead reckoning in between until it was within the range of the electromagnetic field. Accuracy of up to 20 cm from the docking station centerline was achieved in this manner. However, the operating distance for the electromagnetic system was about 25 to 30 m long. With this experiment, it was also found that docking would not be successful if the alignment of AUV with the centerline of docking station is deviated by more than 30°.

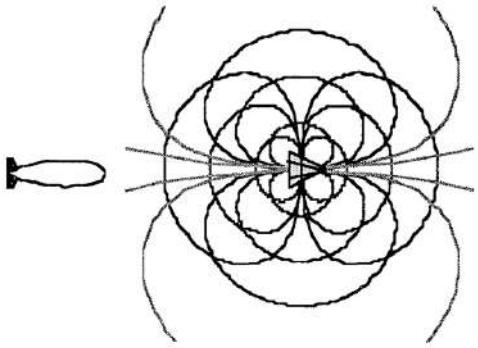


Figure 2.7: Schematic of EM homing system showing field lines and the dock [6].

Testing the pole-like docking station instrument at the WHOI [7] proved to be entirely different from the experiments previously conducted using unidirectional docking stations [1]. The authors claimed that their method could provide good homing and docking phases using straight LOS. The LOS guidance system is based on minimizing the difference of heading towards the DS, calculated from azimuth and elevation angles from the USBL sensor system. An inner-outer loop PID controller was created to allow the AUV to track the desired heading angle. The adaptability of this approach is that both steps can be homed without prior knowledge of the initial bearing between the AUV and the DS. Moreover, the authors included a way for the AUV to determine unsuccessful dockings and replan the homing sequence to have a better rate of success. The supervision and management of the whole autonomously controlled docking process were done using a hierarchical control architecture, using a FSM. This technique is quite promising, but there is an important flaw: the latch mechanism designed to lock the AUV to the DS pole was mounted on the nose of the AUV. This place typically requires the placement of lots of sensors like cameras and sonar and may lead to design conflict. The scheme of the communication system is in figure 2.8, where the acoustic link and the inductive power transmission between the AUV and the DS are highlighted, as well as the communication link between the DS and the ship and remote users, allowed by the use of a surface computer together with a satellite link.
In an open sea environment, a docking experiment was run by MBARI using a Dorado/ Bluefin 21" AUV, whose model is shown in 2.9, and a fixed-heading funnel-shaped DS [8].
The entire experiment was made up of several steps [1]: locating the DS, homing to it, computing a position fix, traveling to the initial position of final approach path, executing final approach, and finally latching on. The way the Dorado AUV navigated was to exploit a USBL

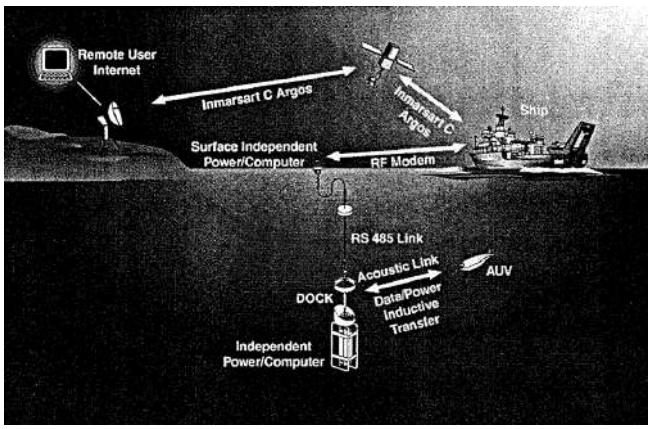


Figure 2.8: Implemented data paths between the dock, AUV, and remote users [7].

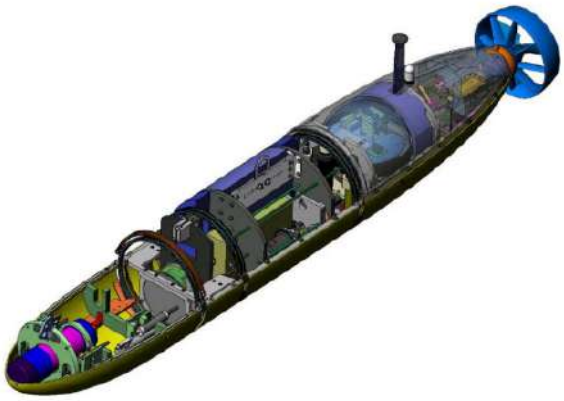


Figure 2.9: Solid model of the docking AUV showing USBL homing system as a cylinder at the bow [8].

and a DVL, using pure pursuit law for guidance during the homing stage. During docking, a path-following scheme that followed the DS centerline at a continued vehicle speed of 1 m/s and minimized cross-track error between the actual heading angle and the reference heading angle was developed with PID control. The cross-track error is here defined as measures of how much the vehicle was far from the desired orientation. No current disturbances were effectively managed by the pure pursuit guidance or the path-following guidance system. Therefore, the direction of the DS was deliberately pointed to minimizing the influence of the currents on its path.

A control solution for AUV homing and docking using a long baseline navigation sensor has been proposed by the University of Southampton in [44], [45]. The guidance system used the artificial potential field method for path generation, a line-of-sight guidance law for computing the reference heading, and a decoupled sliding mode controller for heading and depth control of the vehicle. A valid reduced-dynamic model of the AUV, which assumed passive control of the roll direction, was employed in this study. The effectiveness of the docking solution was tested solely through MATLAB simulations, so it's not a completely exhaustive research.

An evaluation of a visual navigation-based docking system was carried out by the Korea Advanced Institute of Science and Technology (KAIST) to guide the ISiMI AUV during its final docking phase into a fixed funnel-shaped DS equipped with five light sources [46], an example being reported at 2.10. The system used a pure-pursuit guidance algorithm to compute the reference heading angle, while a decoupled PID control loop was employed to track both the reference heading and depth. It's important to mention that this study did not account for the

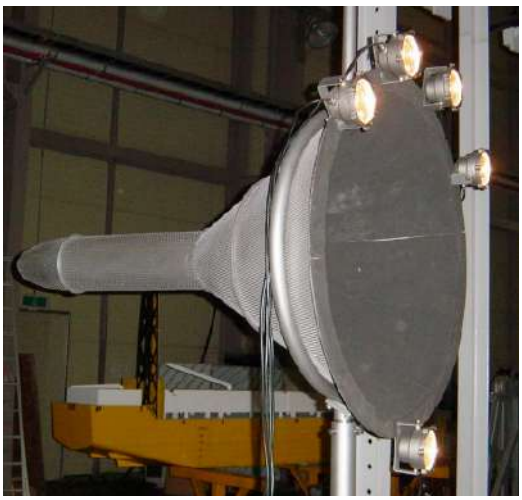


Figure 2.10: Five lights-installed funnel-shaped entrance [9].

effects of current disturbances, so it's not an entirely reliable study. Subsequently, the work was extended introducing a modified linear terminal guidance system, which incorporated a sideslip angle for the AUV during docking operations in the presence of ocean currents, partially making up for the lack of consideration of current disturbances [9],[47], [30], [48]. In these studies, the authors assumed the pose of the DS to be known a priori and employed both predefined uniform current disturbances and time-varying current fields recorded using an ocean current monitoring system. The primary objective in these experiments was to add an intentional cross-track error to mitigate the influence of current disturbances. In the end, the computation of a reference heading angle was performed, allowing the vehicle to align with the centerline of the DS during the terminal docking phase.
Ifremer and DCNS jointly developed a funnel-shaped docking solution for the Asterx AUV, a 4.5m long vehicle, weighting 800 kg, suitable for depths up to 2800 m, designed to dock into a moving underwater DS carried by a vessel, traveling at a speed of 2 knots [49]. The AUV was guided towards the DS using an hybrid system: an active acoustic positioning was used until it entered the optical range of the DS, at which point it homed in by recognizing LED patterns mounted on the DS. This work has been further advanced by Naval Group (formerly DCNS) in collaboration with RTSys, resulting in the development of a submarine-based launch/recovery system. The enhanced system includes integrated induction charging, Wi-Fi data communication, and an acoustic terminal guidance system replacing the original optical guidance system [50].
A two-step docking approach that integrates guidance and control strategies for an AUV used to operate on the field, with interventions, was developed by the University of Lisbon in [29]. In the first step, the control law was derived by assuming an underactuated dynamic model, meaning a model with less constraints, actuations, than the number of Degrees of Freedom (DOF), for the vehicle operating at relatively high speeds. This provided the AUV with a steering maneuver that guided it towards the final docking path, ensuring convergence to zero error for almost all initial conditions. In the second step, the vehicle reduced its speed to start performing the final part of the docking task, for which a fully-actuated dynamic model was used to achieve accurate docking performance. An adaptive control law was proposed in this phase to address uncertainties in the hydrodynamic parameters and ensure the desired accuracy in the vehicle's docking profile. This strategy, made of the combination of these two approaches, was developed for a 2D case in the horizontal plane, without considering the effects of current disturbances, especially on the vertical plane. However, the authors suggested that the proposed approach could be extended to accommodate 3D docking operations.

A three-layer hierarchical control architecture was developed by the University of Porto to enable the docking of an AUV with a mobile DS mounted on an ROV [51]. The higher-level control layer used hybrid automata, generalized FSM for modeling hybrid systems, namely discrete programs within analog environments, to supervise the entire docking operation, which included both interacting discrete events and continuous states represented by nonlinear ordinary differential equations. The supervisor component is the one switching between the different maneuver types, the middle block generated the necessary reference states, managed path following (PF), and performed the circular maneuvers. Some potential fields were also developed to perform obstacle avoidance. The lower-level controller employed sliding mode theory to facilitate the tracking of reference signals in the presence of state uncertainties and external disturbances. The effectiveness of the proposed control system has only been tested through MATLAB simulations so far.
Earlier, the University of Porto developed a control law that takes into account the position as the only controlled variable, while the orientation was unconstrained, for robust homing of the MARES AUV to a beacon [52]. This guidance law, derived from Lyapunov theory as opposed to using computationally intensive state estimation techniques to guide the AUV toward the beacon without requiring any "initialization", was designed to have the same performance regardless of whether or not initialization was performed. The approach was supposed to minimize all computation complexity and the requirement for sensors or equipment. Results from the tested homing experiment conducted in the Douro River subsequently demonstrated the asymptotic convergence of the vehicle to reference beacon.
Harbin Engineering University has also acted upon Lyapunov's stability theory toward a design of docking guidance controller [26]. The suggested guidance system used the visual positioning system and demonstrated ability in pool tests to generate reference heading and crabbing angles for compensation to horizontal and vertical deviations. It was then aimed at aligning the vehicle in the centerline of the DS in the end-phase docking process. The study thus excluded the explicit effects of the current disturbances. An 80% successful docking rate was recorded from pool test experiments.
A research paper, published by Zhejiang University, studies the guidance system for hybrid underwater gliders docking at the fixed funnel-shaped docking station [25]. This study developed the docking station as an active structure with actuators, enabling adjustment of its orientation for reception from different vehicle approach directions. The DS also works with an acoustic Doppler current profiler (ADCP) that measures the water current surrounding the station and transmits it to the AUV. This data incorporates with the guidance system of the vehicle to provide compensation from current disturbance. The AUV, in this case, is equipped with a USBL system along with visual navigation. A modified pure pursuit guidance algorithm has been formed. This algorithm was focused on the terminal part of the docking process and considered current compensation as well. The only function of the guidance system is to guide the vehicle with utmost accuracy to its docking position while the DS aligns itself to face the AUV during terminal docking. The actual trial in a swimming pool proved the feasibility and efficacy of the proposed cooperative approach in docking operations. Further experimentats should be carried out in open water conditions for real-world current disturbances testing.
Kongsberg Hydroid staged their docking test of REMUS-100 AUVs by towing a mobile funnel-shaped docking station (DS) behind a surface vessel [10]. The motion was stabilized through the water with depressor wings, which are stabilizing devices used to control motion and orientation of objects. Furthermore, a capture nose assembly module was designed in order to lock the AUV to the Docking Station, as it is possible to notice in figure 2.11.
This experiment followed a stepwise approach consisting of detection-tracking-interception-docking-undocking. The docking approach utilized a USBL navigation system, capable of calculating the rendezvous location as well as the intercept course with a LOS correction. A

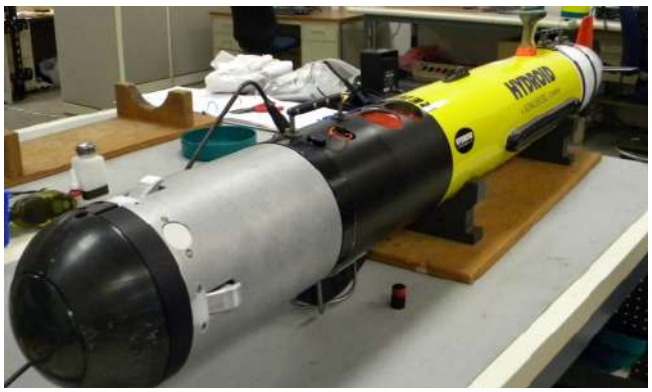
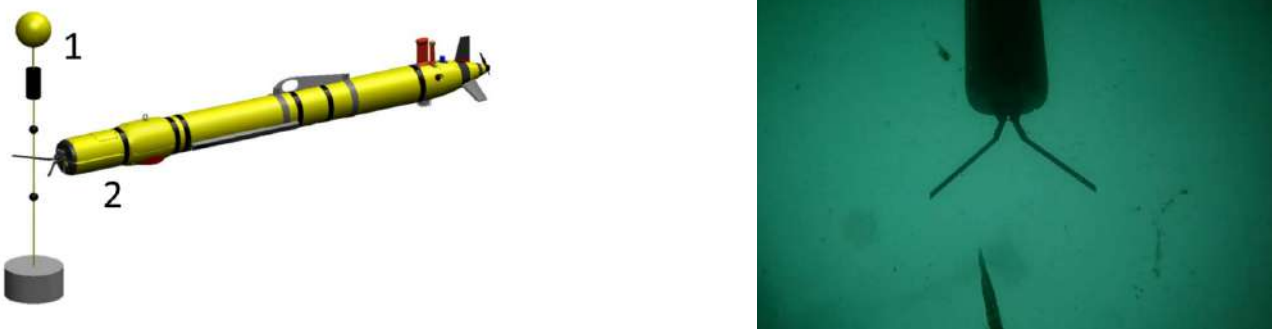


Figure 2.11: REMUS-100 with capture nose assembly [10].

2D guidance system, considering a fixed depth, was developed for the approach and latching process to the mobile DS. During the dock phase, the AUV self-positioned about 200 m behind the DS. It was also aligned to the central axis of the DS. This was done using horizontal angular measurements as input by the D-USBL system. Docking was tested with the AUV going with the current, against the current, and across the current. The vehicle was given maximum ten attempts and succeeded in docking within the first two attempts in the test conducted over 77% of the 11 tests carried out.
The Line Capture Line Recovery (LCLR), proposed by Hydroid, is a docking system adapted from the solution proposed in [10]. This DS is designed to welcome the REMUS 600 vehicle [11]. To aid in its operation, the underwater vehicle is equipped with a D-USBL, integrated with a pair of retractable whiskers at its nose. The AUV moves towards an acoustic transponder attached to a vertically suspended cable, as it is possible to notice in figure 2.12a and 2.12b, where **1** is the cable and **2** is the robot equipped with the sensor whiskers. As it approaches,



(a) Simulation.

(b) Real world picture.

Figure 2.12: Cable and whiskers [11].

the vehicle deploys the whiskers to widen the aperture for capturing the cable, while simultaneously opening the latch mechanism located between the two whiskers. Once the cable is intercepted, the latch hooks onto it, securing the AUV. The LCLR system can be used for stationary or towed deployment from a vessel, although recovery is likely to be affected by sea state conditions. The AUV captures the cable and is then pulled down into the DS for undersea installations or up a slope onto the stern ramp of a moving vessel. Hydroid now offers a commercial version of the LCLR system, the Sea Launcher.
Florida Atlantic University demonstrated an alternative line capture system for a REMUS-100 AUV deployed from a catamaran-style Unmanned Surface Vehicle (USV) [53]. The line was kept taut by a depressor wing with an acoustic transponder. The location and pose of the USV

was time-sampled and communicated to the AUV. The vertical line was then navigated using the D-USBL system in combination with fuzzy logic guidance, useful in case of difficulties in modeling the system's behavior due to uncertainties or noisy sensor data, and control systems. Sea trial results indicated that protected conditions enabled three successful retrieval attempts out of a total of eight failures thus showing that the proposed taut cable system is a promising alternative to a mobile direct towed approach.
The LOON-DOCK project, which is a continuation of the Universitat de Girona's SUNRISE FP7 project, involved an interfaced web system for creating a docking system that would provide a remote deployment for survey missions [28], [12]. It employed a two-stage navigation method using acoustic range-only measurements and visual beacon detection, which can be analyzed in figure 2.13, for the localization and pose determination of the DS by the Sparus II AUV. A 2D LOS guidance strategy was implemented with two proportional controllers for

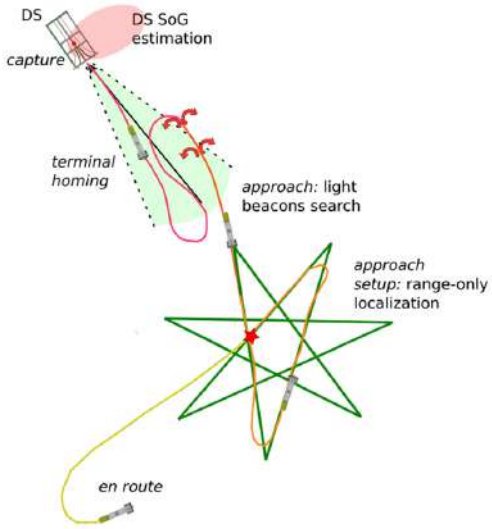


Figure 2.13: Docking phases scheme [12].

heading and velocity control to steer the AUV towards the DS. Experimentally the system was tested both in controlled water tank environment and in open sea conditions, and the results proved that the docking system was effective in minimal current disturbances; moderate to high current disturbances posed challenges, particularly if the DS heading was not in line with the current direction. In addition, the visual navigation system was susceptible to visibility in the water and was thus complicated during docking in such turbid water conditions.
The research work proposed by [13] focuses on the development of robust underwater docking scheme, enhancing operational sustainability of Autonomous Underwater Vehicles. AUVs per se are energy-constrained vehicles, having limited long-term missions, ideally without frequent and often human intervention to recharge their batteries. Therefore, the researchers intended to propose an approach navigation algorithm, tested on the OceanServer Iver3, and a terminal homing algorithm, validated on the Bluefin SandShark, from the initial position to the undersea station, for recharging and data transfer. The docking station consistd of two components: the DS itself and an adapter attached to the AUV, as depicted in figure 2.14, which allows to use the DS potentially with any AUV.
In the approach phase, Dubins path planning is combined with integral line of sight control; this allows the AUV to dynamically replan its trajectory in the presence of disturbances or failed docking attempts. Once close to the docking station, light-tracking is used to guide the AUV into the DS. This is further strengthened by employing simple, yet effective, visual and control systems to be accommodated within varied underwater conditions. Results from the experiments indicate high degrees of accuracy for the approach phase - the vehicle was able

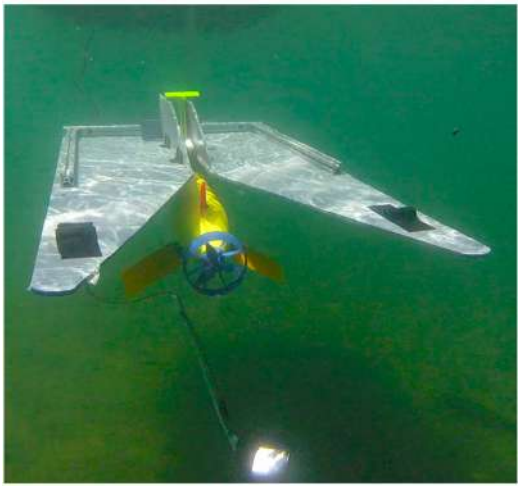
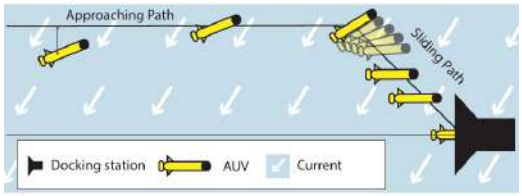
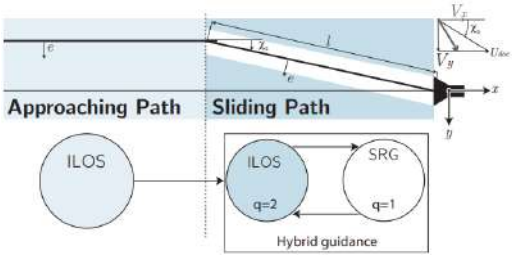


Figure 2.14: Adaptable docking system [13]

to get to the docking hand-off 2 m apart within all experiment runs, while it brought results of 1 m in 39 out of 48 experiments. The terminal homing part, although a bit less accurate, shows a success rate of about 70% in well-controlled conditions, while the failed attempts were concurrent to quite strong water jets. However, the fact that it combines replanning and retry mechanisms ensures that the cumulative success of subsequent attempts will be very high. This method indeed, as the researchers claims, is scalable and able to fit to different AUV platforms and thus lays the groundwork for continued persistent undersea operation. Integration of this docking technology with the recent advancements in underwater communication would be sure to provide for an entirely new platform for undersea operations.
The research carried out by [54] outlines a docking approach designed for an underactuated AUV, so not controlling directly sway. The purpose of this docking strategy is to perform an accurate alignment to the docking station in the presence of bumpy conditions such as cross-currents without relying on crab angle. According to the study results, uniquely tested so far on simulations, docking without crab angles becomes an attractive approach in cases where an entrance geometry of the docking station does not permit crab angles. The main docking process can therefore be subdivided into two major stages, as depicted in 2.15a and 2.15b, i.e. approaching path and sliding path. So the initial path that the AUV follows to dock is one



(a) Approaching and sliding path [54].



(b) Sketch of the path and its parameters [54].

with a parallel offset to the docking centerline while keeping a crab angle with respect to cross currents. When close to the docking, the vehicle enters into the sliding path and heads in the direction of the docking station centerline to enter smoothly. For this purpose, the paper takes two guidance laws, ILOS and SRG, in a combined control architecture formulation. The ILOS allows the AUV to preserve the intended trajectory, providing a straight line path even in presence of unknown currents, while the SRG ensures path precision by estimating the surge velocity of the AUV, aligning the vehicle to the docking station centerline. The combination

produces the advantage of ILOS for trajectory keeping with the strength of SRG for angular accuracy, thereby ensuring a dock that is both robust and reliable. Switching seamlessly between the two guidance laws on the basis of the vehicle's position and alignment with respect to the docking station is possible through the hybrid framework. This method has been established through simulations capable of adapting to changing strengths in currents, reaching the highest success in terms of alignment and position accuracy compared with the other methods.

2.6 Key Contributions

The key contributions of the thesis are described below:

- Development of a comprehensive framework for the analysis, design, and simulation of the different phases involved in the docking of an AUV on a Docking Station (DS)
- Study of two approaches to the docking maneuver: one operating on the single velocity components of the AUV, while keeping the orientation ready for the docking, from the very first segment of the docking phase, and the other one operating on the vehicle's orientation, while keeping the velocity components fixed, until the last part of the maneuver. The second mentioned case is proposed for three different scenarios according to the combination of knowledge available on the docking station (position and heading), while the first case addresses the situation where the DS's orientation is known in advance.
- Derivation of suitable estimation and control algorithms for all the case studies.
- Improvement of the state of the art in what concerns the development of a novel strategy to estimate the relative orientation between the AUV and the Docking Station, in case the orientation of the latter is not known a priori. The implemented method requires the use of two transponders on the DS and a single iUSBL on the AUV only, rather than employing a second iUSBL on the DS. This innovative characteristic holds good potential to significantly lower the cost of acoustics-based docking systems.

Chapter 3

Problem formulation

In the following chapter a description of the problem that has been addressed in this thesis is given to the reader, including useful basis, as the model's kinematics and dynamics, as well as the definition of inner and outer loop.

3.1 Examined Situations

This thesis addresses two cases of interest in the problem of underwater docking, for a Medusa Autonomous Underwater Vehicle actuated in 4 Degrees of Freedom, namely surge, sway, heave and yaw, in presence of constant unknown currents. The general idea is to launch the robot from the ship, and make it proceed through a path in order to get to the docking station. The two scenes differ in the way the vehicle approaches the docking station. Both of them get close to the desired goal position, decoupling the problem in the vertical plane and the horizontal one, since the pitch and the roll DOFs are not actuated, and so the two angles are considered to remain constantly zero, that is,

$$\phi = 0$$
$$\theta = 0$$

3.1.1 Case 1: Variable Orientation, Fixed Planar Speeds

In this scenario the vehicle moves on the plane xy with fixed surge and sway, while the yaw is computed accordingly to the desired trajectory and the estimated current, and the two fixed values of x and y velocity. Based on the accuracy of the knowledge of the docking station's position and orientation, three different subcases were examined. Only in the final part of the homing, the AUV changes attitude in order to adapt to the docking station orientation.

Unknown Orientation ψ_{DS}, Known Position P_{DS}

In this case, the data relative to the orientation of the docking station is completely unknown, while the position is well known.

Known Orientation ψ_{DS}, Unknown Position P_{DS}

The orientation in the North-East-Down reference frame is known with good accuracy, while the position is only known with substantial error.

Unknown Orientation ψ_{DS} and Position P_{DS}

This case is the connection of the two unfavorable previous cases: the orientation is completely unknown and the position is known with substantial error.

3.1.2 Case 2: Fixed Orientation, Variable Planar Speeds

In this case the vehicle moves with a chosen fixed orientation, based on the docking station orientation, and adjusts the planar velocities based on the currents and on the desired trajectory. This case clearly can be carried out just in the situation where the docking station orientation in the NED convention is known, so the studied situation is similar to the second one of the previous case, meaning that the only unknown data is the position, just approximately known, with substantial error.

3.1.3 Path-Following Phases

The path is subdivided in three or four different phases, depending on from which part the vehicle is approaching the docking station, with respect to the entrance, regardless of the working cases.

Phase 1: Approach in Ship frame

The vehicle leaves the ship, which is positioned at the origin of an inertial reference frame, and proceeds towards the docking station, based on the knowledge or prior estimation of its positioning. In this part, the robot relies on the measurements of the inverted USBL, giving the position in the ship reference frame, the DVL, providing the relative speed of the vehicle with respect to the fluid, the AHRS, providing the orientation of the AUV in the NED frame, and the depth cell. In order to position it in a world frame, the ship should send the information retrieved by the GNSS via the acoustic communication system. In this work, the GNSS data is not used because unnecessary for the aim of the docking, since everything is relative to the ship position, which is currently considered fixed. The robot proceeds in this phase until it can receive information on the position with respect to the ship. After the threshold distance, which in this simulation is set to 60 m, even though in the real would it would be of few kilometers, because of reasons of simulation time, the robot switches to the docking station frame.

Phase 2: Approaching in DS frame

When the vehicle is too distant from the ship, it switches behavior. In this phase, it refers its navigation to the docking station, so that becomes the new origin of the inertial frame. Depending on the single case of the ones previously depicted, the sensors used will be different. If the position of the docking station is known, the robot relies again on the iUSBL measurements for the position with respect to the beacon, while if the position is not known, it just relies on range-measurements. In both cases, the measurements coming from AHRS, DVL and depth cell are integrated to estimate the orientation and both robot's and current's velocities. When the vehicle gets enough close to the docking station, which in this work is simulated to be a radius of 45 m, in order not to be too close to it and be able to maneuver the vehicle properly, the AUV changes again behavior. Depending on whether the robot is in the Field of View (FoV) of the docking station or not, then it switches behavior to the third or fourth phase. The FoV is the opening of the docking station, which is where the robot will enter. Whether the AUV is or isn't in this area, it is decoded on the base of the number of pings the vehicle receives from the iUSBL's: if it gets back 2 pings, it means it's in the cone of visibility, so it can skip Phase 3 and go directly to Phase 4, which is the docking part; otherwise, if it gets only one ping back, it means that it's on another side of the opening, so it should go through the third phase prior to the last one.

Phase 3: Circumnavigate

If the vehicle is outside the FoV, it should go around it, following a circular path, while keeping the same attitude, until it detects that it has reached the Field of View of the DS. While performing this operation, the used sensors are the same used in the previous phase.

Phase 4: Terminal Homing

Once the vehicle gets into the FoV, or if it already was from Phase 2, it switches to this last behavior. The motion is different depending on wether the AUV had prior knowledge on the orientation of the DS or not: if the ψ_{DS} was already known, it just follows a blended path, that will be explained later, which is a mix of a circular and a straight path, in order to avoid a sharp transition between the previous and the current trajectory. The simulation ends once the vehicle reaches a threshold distance between the origin of the new inertial frame, meaning the internal point of the DS, and the estimated position of the center of mass of the robot itself. If the vehicle didn't have any prior knowledge on the DS orientation, then it first runs an estimation filter in order to identify the ψ_{DS} and then it proceeds with the adjustment of its orientation in order to align with the desired one, namely $\psi_{DS} + \pi$ since the two frames must be rotated of 180° one another, and it proceeds with the same step as the other case, that is the blended trajectory up until the vehicle gets into the docking structure.

3.2 Vehicle modeling

This section is a brief presentation of basic knowledge necessary to follow the solution to the previously stated problems, providing the definition of the reference frames, the kinematic and dynamic models.

3.2.1 Reference Frames and Basic Notation

A detailed analysis of the Medusa AUV's motion begins with the definition of the coordinate systems used to describe its motion in space. For the purpose of modeling the docking maneuver, three distinct reference frames are introduced, based on the conventions proposed by Fossen et al. [14] , and illustrated in Figure 3.1.

The first is the inertial frame, denoted as $\{I\}$, which adopts the North-East-Down (NED) configuration. In this system, the x_I axis points towards geographic north, y_I extends eastward, and z_I is oriented vertically down, perpendicular to the Earth's surface. Since the AUV operates at relatively low speeds and is not significantly affected by the Earth's rotation, this frame can be considered fixed with respect to the planet's surface. In this thesis, it is assumed to be located at sea level, making the z_I coordinate representative of water depth. Furthermore, it will be considered coincident with the ship's frame, that is the initial position of the AUV. The ship, in this figure is seen from the side.

The second frame, labeled $\{B\}$, is the body-fixed reference frame, centered at the vehicle's center of mass. It is aligned with the primary geometric axes of the AUV: x_B runs along the longitudinal axis from the rear (stern) to the front (bow); y_B extends laterally toward the starboard side; and z_B points downward, completing a right-handed system. This frame moves and rotates with the vehicle, and is used to express the vehicle's velocity and orientation. In the figure 3.1, the vehicle is seen from the side.

The third reference frame, referred to as $\{D\}$, is associated with the docking station. It is fixed at the target position the AUV must reach by the end of the maneuver. Its axes are mutually orthogonal: z_{DS} points downward, consistent with the previous frames, while x_{DS}

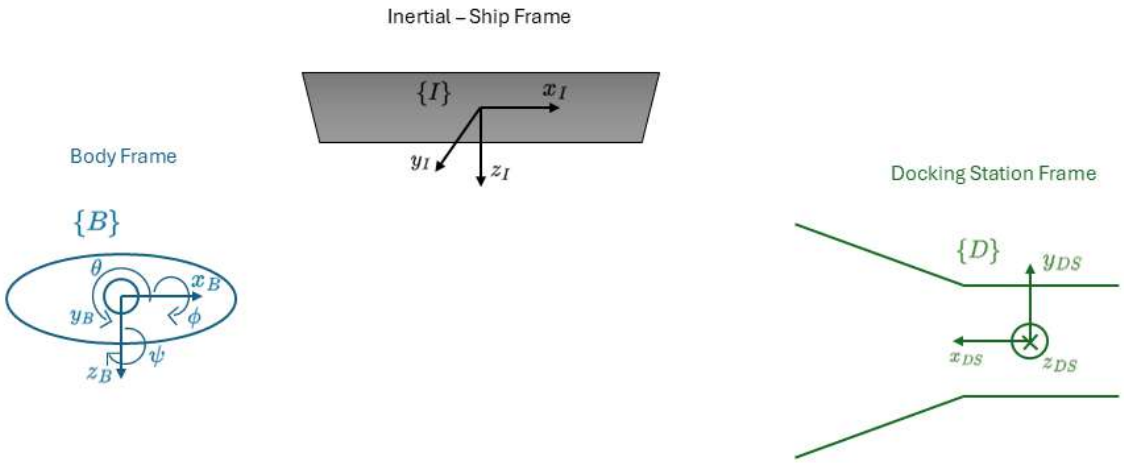


Figure 3.1: Reference Frames.

is directed outward from the docking port through which the vehicle is expected to approach and enter, while the y_{DS} completes the right-handed set of axis. In the figure 3.1, the docking station is seen from the top perspective, and the proof is the entering z_{DS} axis.

Most AUVs have six-degree-of-freedom (6-DOF), experiencing both translational and rotational motion along and about the three principal axes. To describe the AUV's kinematics and dynamics, this work adopts the standardized terminology defined by the Society of Naval Architects and Marine Engineers (SNAME), which provides a consistent framework for representing the vehicle's state, velocities, and applied loads.

The following notation is used throughout the analysis:

- Body-fixed frame ($\{B\}$): A coordinate system rigidly attached to the vehicle, typically centered at its geometric center of mass.
- Inertial frame ($\{I\}$): A fixed reference frame used as a global coordinate system to describe absolute motion.
- Docking station frame ($\{D\}$): A local reference frame anchored to the docking station, centered at the target docking location.

To characterize the vehicle's state, these variables are used:

- The position vector indicates the translational position of the origin of frame $\{B\}$ relative to $\{I\}$, expressed in inertial coordinates:

$$\boldsymbol{\eta}_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- The orientation vector describes the rotational attitude of $\{B\}$ with respect to $\{I\}$, using Euler angles, roll (ϕ), pitch (θ), and yaw (ψ):

$$\boldsymbol{\eta}_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

- The linear velocity vector denotes the velocity of the vehicle's center of mass, expressed in the body frame $\{B\}$, where u is along the x_B, v along y_B and w along z_B:

$$\boldsymbol{\nu}_1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

- The angular velocity vector represents the rotational velocity components about the body axes, also expressed in $\{B\}$, where p is about the x_B, q about y_B and r about z_B:

$$\boldsymbol{\nu}_2 = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

- The external force vector contains the components of the external forces acting on the vehicle in the body frame $\{B\}$, where each component is acting along the relative homonymous axis:

$$\mathbf{F}_{RB} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

- The external torque vector represents the moments applied on the vehicle's roll, pitch, and yaw axes respectively, also expressed in $\{B\}$:

$$\mathbf{N}_{RB} = \begin{bmatrix} K \\ M \\ N \end{bmatrix}$$

3.2.2 Kinematic Model

The kinematic model, as presented by Fossen et al. [14], characterizes the motion of the vehicle from a purely geometric standpoint, independent from the dynamic forces acting on the AUV. This formulation establishes a direct relationship between the linear and angular velocities of the body-fixed frame $\{B\}$ and its corresponding position and orientation with respect to the inertial frame $\{I\}$, through the use of the **Jacobian** matrix, which is composed by non-linear functions of $\boldsymbol{\eta}_2$, with the top left submatrix made by a rotation matrix, describing the transformation from the body frame to the inertial one (see A), while the bottom right submatrix is composed by an Euler attitude transformation matrix (B). Considering the variables described in the previous subsection 3.2.1, the kinematic equations can be written in the following way:

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} {}^I_B R(\boldsymbol{\eta}_2) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & Q(\boldsymbol{\eta}_2) \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix} \tag{3.1}$$

where the left-hand side is the vector $\dot{\boldsymbol{\eta}}$, composed by the derivatives of first order of the position and orientation vectors juxtaposed, while on the right-hand side, post-multiplied to the Jacobian matrix, lies the vector of the velocities along the axis of the DOFs.
However, the considered model has only 4 degrees of freedom, since it is considered to only rotate about the $z - axis$ and so isn't allowed to perform rotations on the roll (θ) and pitch (ψ) angles. Because of this, the kinematic model simplifies considerably:

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} {}^I_B R(\boldsymbol{\eta}_2) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 \\ r \end{bmatrix} \tag{3.2}$$

Which, formulated in equations, is:

$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{z} = w \\ \dot{\psi} = r \end{cases}$$

Note that the rate of variation of the yaw angle only depends on the angular rate r, since it's a decoupled problem in absence of motion along the pitch and roll angles, as well as the vertical motion only depends on the vertical speed w.

3.2.3 Dynamic Model

The study of dynamics focuses on analyzing how external forces and moments influence the motion of a vehicle. A Newton–Euler formulation has been adopted to model the vehicle's dynamics and derive its equations of motion in the body-fixed frame $\{B\}$. According to [14] the rigid-body dynamics are expressed as:

$$M_{RB}\dot{\boldsymbol{\nu}} + C_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{RB} \tag{3.3}$$

where M_{RB} is the rigid-body mass matrix and C_{RB} is the Coriolis and centripetal matrix arising from the rotation of the body frame $\{B\}$ with respect to the inertial frame $\{I\}$. The external force and moment vector is given by:

$$\boldsymbol{\tau}_{RB} = \begin{bmatrix} X \\ Y \\ Z \\ K \\ M \\ N \end{bmatrix} \tag{3.4}$$

which contains the total forces and torques acting on the body frame $\{B\}$. This vector can be decomposed into:

$$\boldsymbol{\tau}_{RB} = \boldsymbol{\tau} + \boldsymbol{\tau}_A + \boldsymbol{\tau}_D + \boldsymbol{\tau}_R + \boldsymbol{\tau}_{dist} \tag{3.5}$$

with the components defined as follows:

- $\boldsymbol{\tau}$: control input forces and moments;
- $\boldsymbol{\tau}_A$: forces and moments due to added mass effects, given by:

$$\boldsymbol{\tau}_A = -M_A\dot{\boldsymbol{\nu}} - C_A(\boldsymbol{\nu})\boldsymbol{\nu} \tag{3.6}$$

- $\boldsymbol{\tau}_D$: hydrodynamic damping effects such as drag and skin friction:

$$\boldsymbol{\tau}_D = -D(\boldsymbol{\nu})\boldsymbol{\nu} \tag{3.7}$$

- $\boldsymbol{\tau}_R$: restoring forces and moments from gravity and buoyancy:

$$\boldsymbol{\tau}_R = -g(\boldsymbol{\eta}) \tag{3.8}$$

- $\boldsymbol{\tau}_{dist}$: environmental disturbances (e.g., wind, ocean currents, waves).

Substituting expressions 3.5–3.8 into 3.3, the overall equation of motion can be expressed compactly as:

$$\underbrace{M_{RB}\dot{\boldsymbol{\nu}} + C_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}}_{\text{rigid-body forces}} + \underbrace{M_A\dot{\boldsymbol{\nu}} + C_A(\boldsymbol{\nu})\boldsymbol{\nu} + D(\boldsymbol{\nu})\boldsymbol{\nu}}_{\text{hydrodynamic forces}} + \underbrace{g(\boldsymbol{\eta}) + g_0}_{\text{hydrostatic forces}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{dist}. \tag{3.9}$$

Given that the vehicles are actuated only by the three forces and a torque about the z-axis, the input vector $\boldsymbol{\tau}$ is redefined as $\boldsymbol{\tau} = [\tau_u, \tau_v, \tau_w, \tau_r]^T$. Based on these assumptions and neglecting external disturbances, because the current is modeled as a constant velocity, and therefore it does not give any contribution to the dynamics of the robot, the dynamic model in Equation 3.9 simplifies to:

$$\begin{cases} m_u\dot{u} - m_v vr + d_u u = \tau_u \\ m_v\dot{v} + m_u ur + d_v v = \tau_v \\ m_w\dot{w} + d_w w - g_w = \tau_w \\ m_r\dot{r} - m_{uv}uv + d_r r = \tau_r \end{cases} \tag{3.10}$$

The parameters in Equation 3.10 are defined as:

$$m_u = m - X_{\dot{u}}, \quad m_v = m - Y_{\dot{v}}, \quad m_w = m - Z_{\dot{w}}, \quad m_r = I_z - N_{\dot{r}} \tag{3.11}$$

$$m_{uv} = m_u - m_v \tag{3.12}$$

$$d_u = -X_u - X_{|u|u}|u|, \quad d_v = -Y_v - Y_{|v|v}|v|, \quad d_w = Z_w + Z_{|w|w}|w|, \quad d_r = -N_r - N_{|r|r}|r| \tag{3.13}$$

where $X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, N_{\dot{r}}, X_u, Y_v, Z_w, N_r, X_{|u|u}, Y_{|v|v}, Z_{|w|w}, N_{|r|r}$ are vehicle-specific hydrodynamic coefficients that will be specified later, while g_w is the restoring term, composed by the algebraic sum of the weight and the buoyancy. The key SIMULINK blocks of the kinematic and dynamic models of MEDUSA are shown in Figure 3.2.

3.3 Medusa Vehicles and Docking Station

The MEDUSA crafts are Autonomous Underwater Vehicles (AUVs) developed by the Laboratory for Robotics and Engineering Systems (LARSyS) and the Dynamical Systems and Ocean Robotics Laboratory (DSOR), both part of the Institute for Systems and Robotics (ISR) at Instituto Superior Técnico (IST) in Lisbon, Portugal. These vehicles serve as research platforms for the DSOR group, under which the present work is conducted. Accordingly, they will be used as the basis for the simulations presented in this thesis. Computer-generated 3D models of the MEDUSA vehicle and its docking system (DS) are shown in Figure 3.3. The main structure of the vehicle consists of two longitudinal acrylic tubes, aligned vertically one above the other and connected via an aluminium frame. These tubes house all electronic components, while the frame provides structural support for the thrusters and sensors mounted on the vehicle.

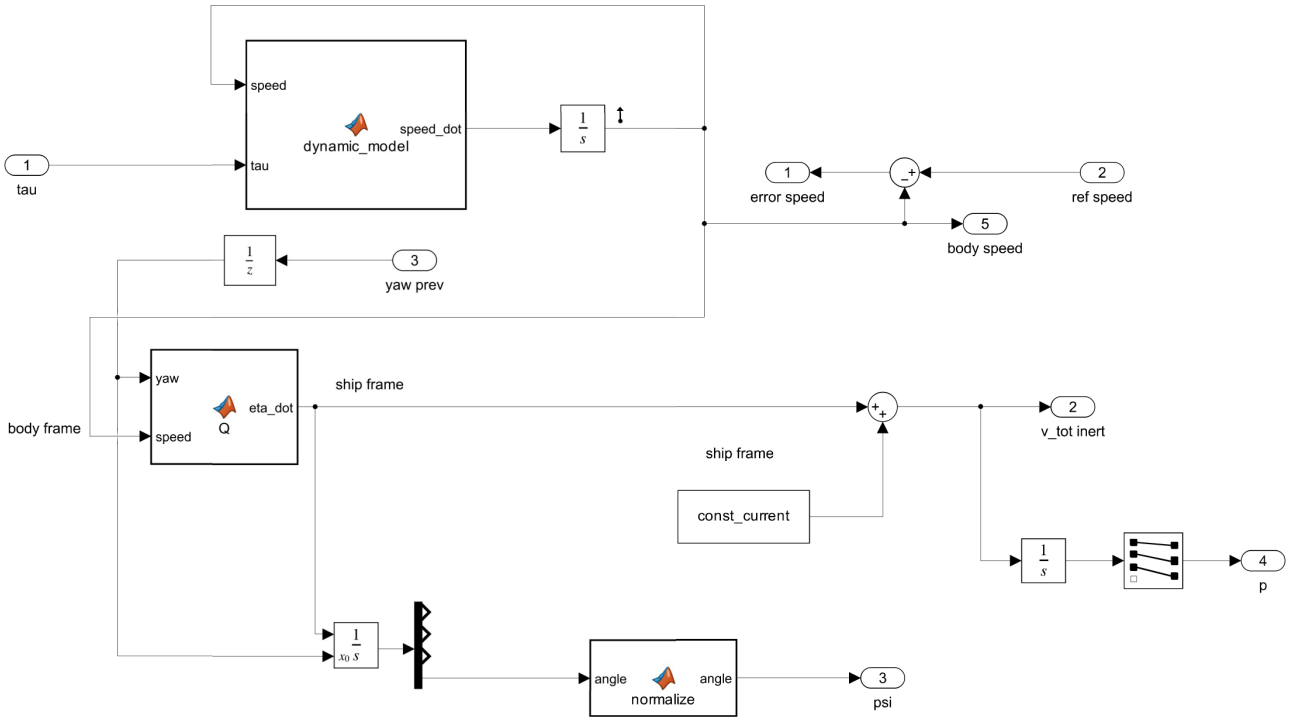
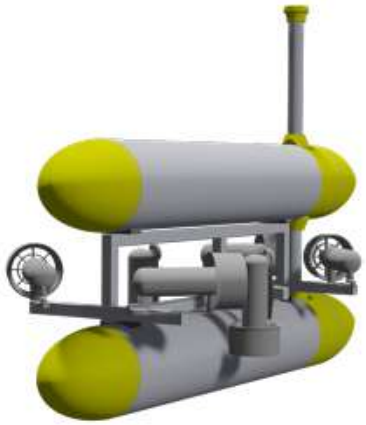


Figure 3.2: Dynamic and Kinematic model.



(a) CAD model of the MEDUSA vehicle

(b) CAD model of the Docking Station (DS)

Figure 3.3: 3D models of the MEDUSA vehicle and the Docking Station

The actuation system comprises six thrusters: four located in the horizontal plane and two in the vertical direction. The four horizontal thrusters are arranged in a vectorial configuration, where each thruster is mounted at a 45° angle relative to the vehicle's longitudinal axis, as illustrated in Figure 3.3a. This configuration allows the vehicle to be fully actuated within the 2D horizontal plane. The Docking Station (DS), see Figure 3.3b, has been specifically designed to match the MEDUSA's geometry. Its funnel-shaped entrance facilitates the vehicle's entry into the dock, ensuring alignment and capture during docking maneuvers. The sensor suite integrated into the vehicle includes a GNSS, a Doppler Velocity Logger (DVL), a Depth Cell, Attitude and Heading Reference System (AHRS), an acoustic modem, and an Ultrashort Baseline (USBL) positioning system. The DS is also equipped with an acoustic modem and USBL system, enabling a communication link between the docking station and the AUV. The hydrodynamic added masses and drag coefficients used in modeling the vehicle have been experimentally identified by the DSOR team and are summarized in Table 3.1. Furthermore, the vehicle's moment of inertia about the z-axis and its mass have been established as $I_z = 4.14$ kg·m^2 and $m = 30$ kg, while the restoring term g_w is equal to -0.8339 N, which is negative thanks to the physical properties designed for safety reasons, in order to achieve a condition such that, in case of null thrust, this would let the AUV float up to the surface, rather than sinking.

$X_{\dot{u}}$	-25 kg	X_u	-0.2 kg/s	$X_{\|u\|u}$	-19.5 kg/m
$Y_{\dot{v}}$	-2.325 kg	Y_v	-55.117 kg/s	$Y_{\|v\|v}$	-147.9 kg/m
$Z_{\dot{w}}$	-19.4311 kg	Z_w	33.6804 kg/s	$Z_{\|w\|w}$	89.4105 kg/m
$N_{\dot{r}}$	-8.690 kg·m^2	N_r	-4.14 kg·m/s	$N_{\|r\|r}$	-6.23 kg·m

Table 3.1: Identified hydrodynamic parameters for the MEDUSA vehicle

The second two Z parameters are with opposite sign with respect to the other directions because of how the equations were modeled in equation 3.10.

3.4 Inner vs Outer Loop

In robotics it is usual to separate motion control in two decoupled problems, in order to simplify the solution: the *outer loop*, or kinematic loop, and the *inner loop*, or control loop, as described in [15] and depicted in figure 3.4.

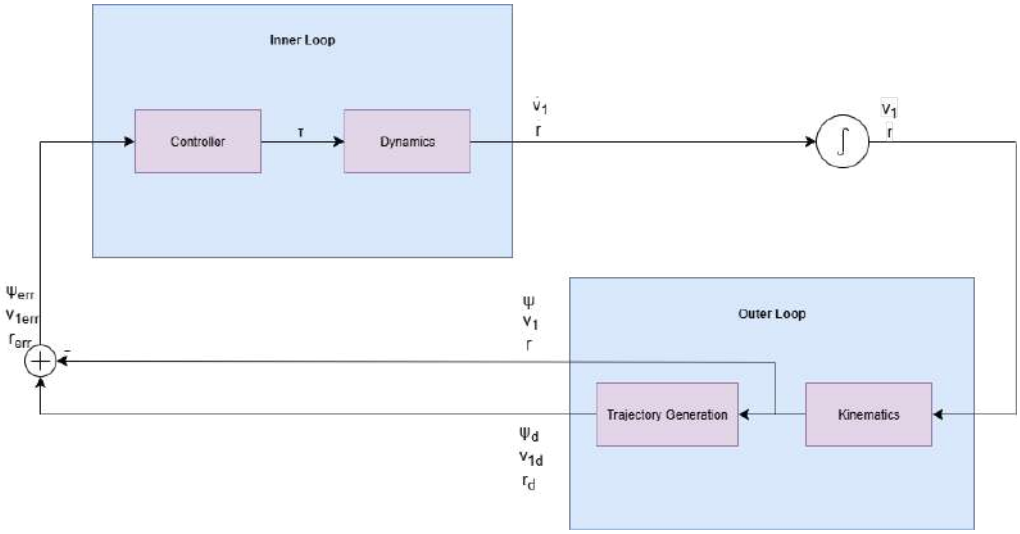


Figure 3.4: Inner-Outer Loop

3.4.1 Outer Loop

The outer loop is responsible for computing the desired linear and angular speeds and orientation of the vehicle, which will be given as inputs, subtracted of the estimate of the current values, to the control unit. This subsystem is also named *guidance strategy* and produces the reference quantities to drive the robot along the path with the desired trajectory and speed profile.
The vehicle's linear kinematics will follow this law:

$$ {}^{I}\boldsymbol{v_T} = {}^{I}_{B}R(\boldsymbol{\psi}){}^{B}\boldsymbol{\nu_1} + {}^{I}\boldsymbol{v_c} \tag{3.14} $$

where the current has the following components:

$$ {}^{I}\boldsymbol{v_c} = \begin{bmatrix} v_{c_x} \\ v_{c_y} \\ v_{c_z} \end{bmatrix} \tag{3.15} $$

Then, assuming the vertical current negligible, since its order of magnitude is usually significantly lower than the one on the xy axis, and the motion on the vertical plane decoupled, the equations of interest become the ones on the horizontal plane:

$$ \begin{cases} v_{T_x} = |{}^{I}\boldsymbol{v_T}|cos(\beta) &= ucos(\psi) - vsin(\psi) + v_{c_x} \\ v_{T_y} = |{}^{I}\boldsymbol{v_T}|sin(\beta) &= usin(\psi) + vcos(\psi) + v_{c_y} \end{cases} \tag{3.16} $$

3.4.2 Inner Loop

The inner loop is composed by the dynamics of the model and the control of the same. As previously mentioned, it receives as input the difference between the reference quantities, given by the outer loop, and the estimate of the current values provided by the Kalman or Estimated Kalman filter, and it produces the desired forces and torques to track the desired values, in the limits given by the physical saturation.

Chapter 4

Proposed Solution

In this chapter, a description of the implementation proposed in this work, to improve the state of the art, is given to the reader, including the guidance system, the control choices, the contribution to the retrieval of the docking station orientation, based on a single iUSBL, the estimate filters, the custom simulation environment and the state machine created to identify the current behavior of the AUV.

4.1 Guidance System - Outer Loop

4.1.1 Case 1: Variable Orientation, Fixed Planar Speeds

In all of the three situations of knowledge about the docking station's position and orientation, the procedure of the outer loop is the same: in the first, second and, if outside the FoV, also the third behavior, the outer loop will provide the desired yaw angle, based on the estimated current velocity, on the fixed u and v values and on the direction the robot should follow, i.e. β, exploiting the equation 3.16: the easiest way to proceed is to find ψ_d out from a range of 360 values between 0 and 2π, as the value minimizing the error between

$$\theta \coloneqq \arctan\left(\frac{v_{T_y}}{v_{T_x}}\right),$$

given by the examined candidate ψ, and the desired trajectory orientation β:

$$\alpha = \theta - \beta \tag{4.1}$$

$$\psi_d = \arg\min_{\psi} |\arctan\left(\frac{sin(\alpha)}{cos(\alpha)}\right)| \tag{4.2}$$

The trigonometric operation is required in order to be sure to obtain a normalized value between $-\pi$ and π.
This value will become the reference one for the yaw controller, i.e. ψ_d. In the final part, approaching the docking station, the law is changed: at this point, the vehicle has to modify its orientation, according to the one of the docking station, whether the latter was already known or it is sensed, depending on the knowledge on the DS, as mentioned in the previous chapter. In this new section, once achieved the correct orientation, the latter is kept constant by a controller, while the body speeds are computed runtime in order to follow the desired trajectory. This is achieved by inverting the vector equation 3.14, cropped to two dimensions out of three, because, again, kinematics is decoupled in the vertical and horizontal planes:

$$\begin{bmatrix} v_{T_x} \\ v_{T_y} \end{bmatrix} = \underbrace{\begin{bmatrix} cos(\psi_{dock}) & -sin(\psi_{dock}) \\ sin(\psi_{dock}) & cos(\psi_{dock}) \end{bmatrix}}_{{}^{I}_{B}R(\psi_{dock})_{xy}} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} v_{c_x} \\ v_{c_y} \end{bmatrix} \tag{4.3}$$

Thanks to the property of orthogonality of the rotational matrix R, it's possible to retrieve the vector of the required body speeds just by subtracting the estimated current velocity, here reported for simplicity as v_{c_x} and v_{c_y}, and premultiplying the result by the transposed rotational matrix:

$$\begin{bmatrix} v_{T_x} \\ v_{T_y} \end{bmatrix} - \begin{bmatrix} v_{c_x} \\ v_{c_y} \end{bmatrix} = {}^{I}_{B}R(\psi_{dock})_{xy} \begin{bmatrix} u \\ v \end{bmatrix} \tag{4.4}$$

leading to:

$$\begin{bmatrix} u \\ v \end{bmatrix} = {}^{I}_{B}R(\psi_{dock})^{T}_{xy} \left(\begin{bmatrix} v_{T_x} \\ v_{T_y} \end{bmatrix} - \begin{bmatrix} v_{c_x} \\ v_{c_y} \end{bmatrix} \right) \tag{4.5}$$

which in equation form will be:

$$\begin{cases} u = (v_{T_x} - v_{c_x})cos(\psi) + (v_{T_y} - v_{c_y})sin(\psi) \\ v = -(v_{T_x} - v_{c_x})sin(\psi) + (v_{T_y} - v_{c_y})cos(\psi) \end{cases} \tag{4.6}$$

Furthermore, the direction β is chosen according to the phase: in the approaching phases, it is computed as

$$\beta = \arctan\left(\frac{y_{ds} - y_{auv}}{x_{ds} - x_{auv}}\right) \tag{4.7}$$

In the circumnavigation phase, β is taken as the tangent direction to the closest point in the virtual circumference built with center in the docking station and radius equal to the threshold of minimum distance from the DS itself.
In the final homing phase, the β is equal to the desired docking ψ_{dock}, and the trajectory will be a blend between a straight line and the circumference, in order to avoid abrupt transitions between the two behaviors, which would bring the AUV out from the desired path to enter the DS properly, but how to implement this will be further explained later in this section.

4.1.2 Case 2: Fixed Orientations, Variable Planar Speeds

In the case where the docking station's orientation is known, it's possible to proceed throughout all the four phases with already the proper ψ_{dock}, and compute runtime the necessary body speeds to pursue the desired trajectory and compensate for the current velocity.
In what concerns the directions β of the four phases, it's computed in the same way as in section 4.1.1, so it won't be repeated here. The planar body velocities are computed in the same way for all the four phases, given the estimated ${}^{I}\boldsymbol{v_c}$, the desired total speed $|{}^{I}\boldsymbol{v_T}|$ and the ψ_{dock}, as well as the trajectory orientation previously mentioned β, exactly as computed in the terminal homing phase treated in the previous section, in Equation 4.6.

4.1.3 Vertical Plane

In the vertical plane, being completely decoupled from the horizontal one, and assuming pitch and roll constantly equal to zero, since they're not actuated, the situation is much easier and it's equal for both scenarios: both when the orientation is fixed and the body velocities are variable, and the opposite case. In this case the key idea is simply to directly reduce to zero the error between the depth of the robot and the depth (or it's approximate estimate) of the docking station. For this reason, there's no need for any specific guidance system, since it's not required to perform any particular trajectory. Furthermore, the reference body speed for the AUV's vertical plane will always be zero, because, once reached the correct depth, the robot should not go through any vertical displacement, so the desired vertical velocity will be $w = 0$.

4.1.4 Blended Velocities in phase 4

A smooth blending function can be defined based on a normalized and clamped distance. Given a distance d and a maximum threshold D, the scalar s is computed as follows:

$$s = 1 - \max\left(0, \min\left(1, \frac{d}{D}\right)\right) \tag{4.8}$$

This formulation ensures that $s = 1$ when $d = 0$, $s = 0$ when $d \geq D$, the transition from $s = 1$ to $s = 0$ is linear and inverted with respect to the normalized distance.

To produce a non-linear and smooth interpolation, the following cubic polynomial is applied:

$$w = 3s^2 - 2s^3 \tag{4.9}$$

This function, known as the *smooth step* function, has the following properties:

- $w(0) = 0$, $w(1) = 1$,
- Zero derivatives at the endpoints: $w'(0) = 0$, $w'(1) = 0$,
- Continuous and differentiable over the interval $[0, 1]$,
- Provides an "ease-in, ease-out" transition profile.

The smooth step function helps to avoid abrupt changes and ensures a gradual variation of the output weight w.

The weight then is used to blend the circular component of the velocity, in the following way:

$$v_{blend} = (1 - w)v_{circ} + v_{line} \tag{4.10}$$

In this way, the vehicle enters the fourth phase when it's not yet on the line, and this way it can start adjusting the direction of motion, so the velocity has still a component of the circular velocity in order to keep steering in a seamless way, plus a straight line component to help the vehicle go straight in a more efficient way. Once the vehicle gets closer and closer to the line, the circular component decreases in order to let the vehicle move in a straight line. This algorithm turned out to be efficient in order not to make the AUV lose the correct path and not to make it steer in an abrupt trajectory, which would occur in absence of the transition zone, using just a straight line from the very beginning of the homing phase.

4.1.5 Local Path Definition

Circular Path

The circular path was built by finding the closest point in the reference circumference and considering the tangent to that point as the reference direction of the total velocity vector, namely the angle β.

Straight Path

The straight path has been built using the Line-Of-Sight algorithm, coupled with the Δ-implementation which can be found in section 3.2.3 of [15], already described in 2.4.1. The specific procedure is here reminded.

Given $\mathbf{r}$, the current position, $\mathbf{s}^*$, the reference point in the straight line of the docking station, that is the closest point on that line to the vehicle, ψ_{dock}, the reference heading (e.g., aligned

with the docking path) and Δ, the lookahead distance, and computing the orthogonal distance $\mathbf{y}_1$ of the AUV to s^* in the following way, as the norm of the vector $\mathbf{y}_{1,I}$

$$^I\mathbf{y}_{1,I} = {}^I\mathbf{r} - {}^I\mathbf{s}^* \tag{4.11}$$

resulting in

$$y_1 = \sqrt{(y_{1,I}^{(1)})^2 + (y_{1,I}^{(2)})^2} \tag{4.12}$$

The vector is then corrected with the sign according to this policy:

$$\text{sign} = \begin{cases} -1, & \text{if } r_1 < s_1^* \\ 1, & \text{if } r_1 \geq s_1^* \end{cases}$$

$$y_1 \leftarrow y_1 \cdot \text{sign} \tag{4.13}$$

This step assigns a direction to the displacement y_1, depending on whether the target is to the left or right of the current position. The Line-Of-Sight angle α is computed this way

$$\alpha = \arctan(\frac{y_1}{\Delta}) \tag{4.14}$$

using the signed cross-track error y_1 and a constant lookahead distance Δ. Finally, the desired heading β is computed as

$$\beta = \psi_{\text{dock}} + \alpha \tag{4.15}$$

combining the dock alignment angle and the LOS correction.

4.2 Control System - Inner Loop

The control system is the same for all the examined cases, and it's specific for the controlled quantity, adapting it to the single Degree of Freedom. All of them are implemented using an anti-windup strategy in order not to saturate the engines.
The controller is introduced in cascade before the plant itself, which is the dynamic model. It gets as input the error, which is defined according to the convention as the difference between the reference quantity, computed by the outer loop, and the estimated current quantity, since the AUV doesn't have direct access to the real value but can only estimate it through proper filters, starting from the values perceived by the sensors:

$$e \coloneqq x_d - \hat{x} \tag{4.16}$$

where x is the general example quantity. One of the key objectives of using PI or PID controllers is to ensure that the steady-state error approaches zero as time goes to infinity:

$$\lim_{t \to \infty} e(t) = 0 \tag{4.17}$$

This is achieved primarily through the integral action, which accumulates the error over time and applies corrective input to eliminate any persistent deviation. As a result, PI and PID controllers are particularly effective in rejecting constant disturbances and ensuring accurate reference tracking.
The general scheme of the feedback control loop in *Laplace* domain is in Figure 4.1, where the reference signal is $R(s)$, $Y(s)$ is the current measured quantity, which is computed as the cascade of the signal $U(s) = E(s)K(s)$ and the plant $P(s)$, resulting in $Y(s) = U(s)P(s)$, $E(s)$ is the error derived by the subtraction of the latter two, $E(s) = R(s) - Y(s)$, $U(s)$ is

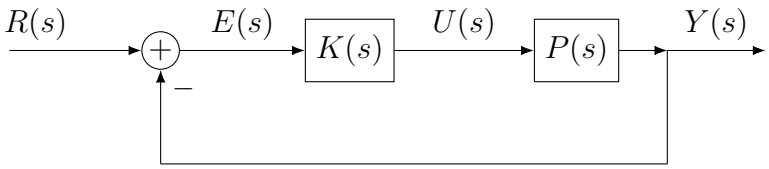


Figure 4.1: Standard unity feedback control loop

the controlled signal, $K(s)$ is the controller transfer function and $P(s)$ is the plant transfer function.
The feedback loop transfer function is obtained by closing the chain in the following way:

$$\begin{aligned} Y(s) &= U(s)P(s) = \\ &= E(s)K(s)P(s) = \\ &= (R(s) - Y(s))K(s)P(s) = \\ &= R(s)K(s)P(s) - Y(s)K(s)P(s) \end{aligned} \tag{4.18}$$

Which, grouping the components with $Y(s)$ on the left side, results in:

$$Y(s)(1 + K(s)P(s)) = R(s)K(s)P(s) \tag{4.19}$$

leading to the computation of the general feedback system transfer function:

$$P_{CL}(s) = \frac{Y(s)}{R(s)} = \frac{K(s)P(s)}{1 + K(s)P(s)} \tag{4.20}$$

valid for whichever plant $P(s)$, controller $K(s)$, measure $Y(s)$ and reference $R(s)$. Then, based on the performance required for the closed loop system, the transfer function of the controller will be chosen in a custom way.
The stream is the following: first the gains are computed based on the performance required in closed loop, then they're fed to the actual controllers together with the relative error, producing the necessary torque which will be sent to the simulation of the physical model. The visual idea of the structure is given in the picture 4.2. The plant of the system $P(s)$, is the model itself, with the exception that it's considered simplified, by neglecting the mixed speed terms. Because of this, the set of equations in 3.10 simplifies to the following:

$$\begin{cases} m_u\dot{u} + d_u u = \tau_u \\ m_v\dot{v} + d_v v = \tau_v \\ m_w\dot{w} + d_w w - g_w = \tau_w \\ m_r\dot{r} + d_r r = \tau_r \end{cases} \tag{4.21}$$

In this way, the controller is a bit less accurate, but, being a robust controller, it performs well.

4.2.1 PI for Surge Control

To control the surge, a Proportional Integrative (PI) controller was implemented, whose general functioning is explained in C.1. The reason of this choice is that what matters for this degree of freedom is to let the AUV's surge speed follow the desired quantity, which is, indeed, a velocity. Recalling the PI controller general transfer function in the *Laplace* domain,

$$K_{PI_u}(s) = K_{P_u} + \frac{K_{I_u}}{s} \tag{4.22}$$

It's then necessary to define the plant function $P_u(s)$. Since the model is non-linear, it's necessary to linearize it about an equilibrium point, which is represented by the computed

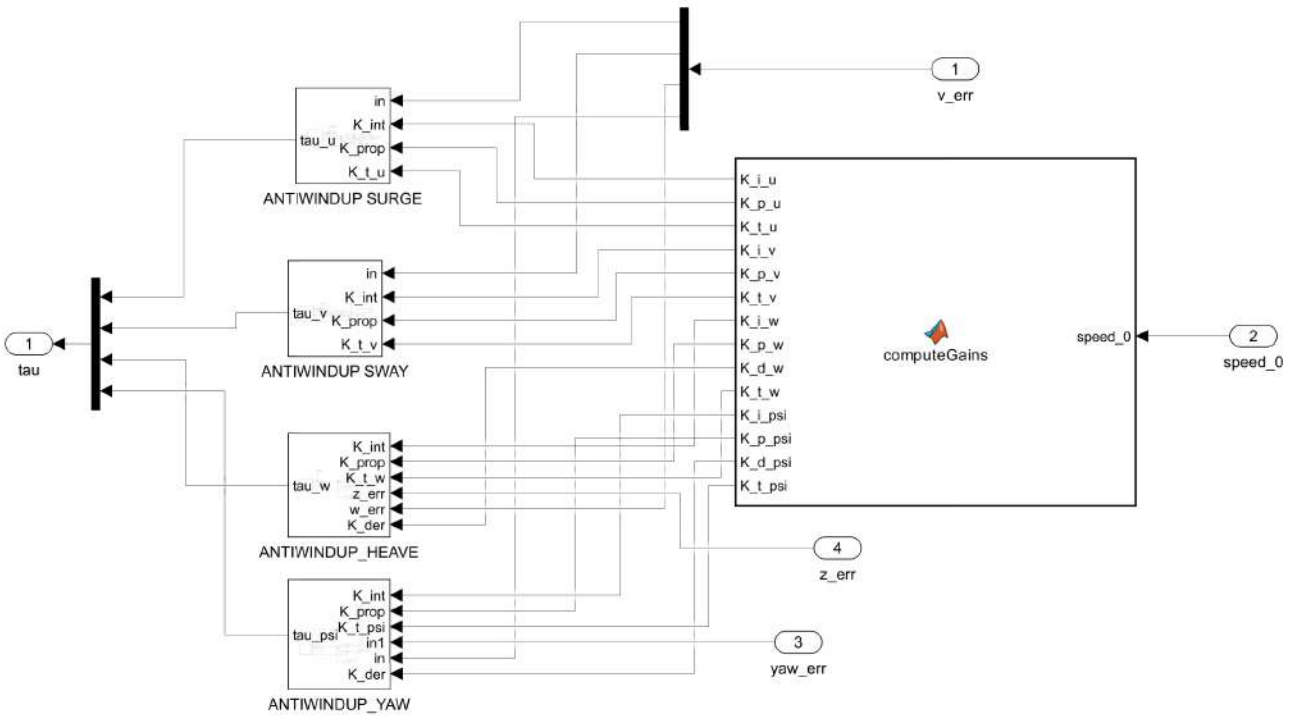


Figure 4.2: Controller stream.

desired value given by the outer loop. The simplified model equation of the surge component is:

$$\dot{u} = \frac{X_u}{m_u}u + \frac{X_{|u|u}}{m_u}|u|u + \frac{\tau_u}{m_u} \tag{4.23}$$

First, it's necessary to compute the equilibrium torque, i.e. the torque providing a null acceleration along the x_B component, and which ideally leaves the u unchanged on the equilibrium value u_0:

$$0 = \frac{X_u}{m_u}u_0 + \frac{X_{|u|u}}{m_u}|u_0|u_0 + \frac{\tau_{u0}}{m_u} \tag{4.24}$$

which gives the following equilibrium force

$$\tau_{u0} = -X_u u_0 - X_{|u|u}|u_0|u_0 \tag{4.25}$$

Now it's possible to linearize the equation 4.23 about the equilibrium point given by the following set of conditions:

$$u = u_0 + \delta u \qquad \dot{u} = \dot{u_0} + \delta\dot{u} \qquad \tau_u = \tau_{u0} + \delta\tau_u \tag{4.26}$$

with $\dot{u_0} = 0$ since u_0 is considered constant in the range of time around which the linearization is performed. The method to linearize an equation like 4.23 is described in D, and here it is exploited to retrieve the linearized version of the dynamics of u:

$$\begin{aligned}\dot{u} = \delta\dot{u} &= \left.\frac{\partial f}{\partial u}\right|_{u_0,\tau_{u0}} \delta u + \left.\frac{\partial f}{\partial u}\right|_{u_0,\tau_{u0}} \delta\tau_u \\ &= \underbrace{\left(\frac{X_u}{m_u} + \frac{X_{|u|u}}{m_u}2|u_0|\right)}_{A_u}\delta u + \underbrace{\frac{1}{m_u}}_{B_u}\delta\tau_u\end{aligned} \tag{4.27}$$

which in the *Laplace* domain becomes:

$$s\tilde{U}(s) = A_u\tilde{U}(s) + B_u\tilde{T}_u(s) \tag{4.28}$$

leading to the following plant P_u(s):

$$P_u(s) \coloneqq \frac{\tilde{U}(s)}{\tilde{T}_u(s)} = \frac{B_u}{s - A_u} = \frac{1}{m_u s - X_u - X_{|u|u} 2|u_0|} \tag{4.29}$$

Plugging it, together with the $K_{PI}(s)$ expression 4.22, in the feedback equation 4.20, the closed loop transfer function for the surge control is given by:

$$P_{u_{CL}} = \frac{P_u(s) K_{PI_u}(s)}{1 + P_u(s) K_{PI_u}(s)} = \frac{s\frac{K_{P_u}}{m_u} + \frac{K_{I_u}}{m_u}}{s^2 + s(\frac{K_{P_u} - X_u - X_{|u|u} 2|u_0|}{m_u}) + \frac{K_{I_u}}{m_u}} \tag{4.30}$$

Now, the idea is to obtain a behavior of the system such that its feedback poles follow the desired natural frequency and damping; in order to achieve this, the system should get a transfer function with the following shape:

$$P_{u_{CL_d}}(s) = \frac{N(s)}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{4.31}$$

Regardless of the numerators, there should be a match of the denominators, in order to obtain the desired behavior in frequency. Because of this, it's important to choose K_{I_u} and K_{P_u} in order to satisfy the following relations:

$$\frac{K_{P_u} - X_u - X_{|u|u} 2|u_0|}{m_u} = 2\xi\omega_n \qquad \frac{K_{I_u}}{m_u} = \omega_n^2 \tag{4.32}$$

leading to the following values:

$$K_{P_u} = 2\xi\omega_n m_u + X_u + X_{|u|u} 2|u_0| \qquad K_{I_u} = m_u \omega_n^2 \tag{4.33}$$

In this work, the chosen ω_n and ξ values are respectively 0.2 and 0.6, for every controller.
The structure of the PI controller for the surge is the following:

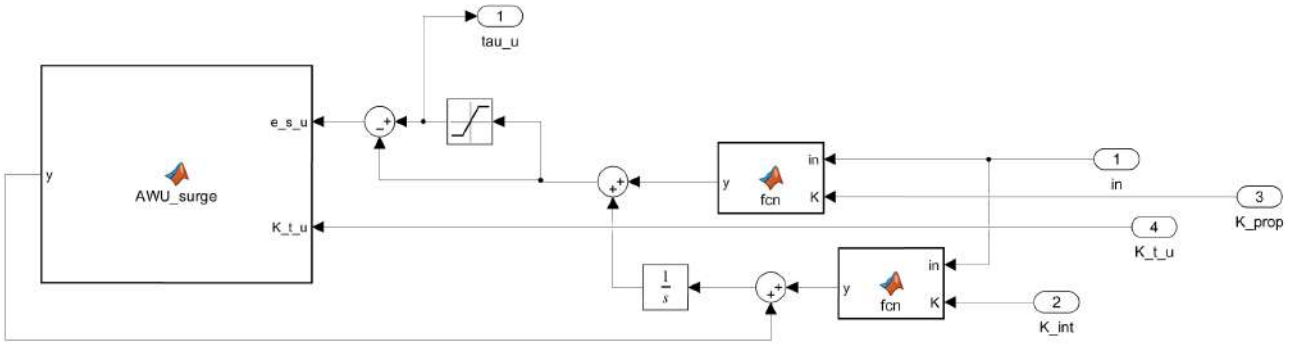


Figure 4.3: PI with Anti-Windup for surge control.

4.2.2 PI for Sway Control

The controller for the sway DoF follows exactly the same scheme of the sway one, since the goal is to force the system to follow the desired value of velocity along the y_B of the vehicle and the equation of the dynamics along this direction is exactly equal to the one along the x.

To avoid redundancy, the final results are directly proposed to the reader. The plant of the system, based on the simplified dynamic model 4.21, is the following:

$$P_v(s) \coloneqq \frac{\tilde{V}(s)}{\tilde{T}_v(s)} = \frac{1}{m_v s - Y_v - Y_{|v|v} 2|v_0|} \tag{4.34}$$

which, plugged in the closed loop equation, together with the same PI controller structure as the surge one, results in the following feedback transfer function:

$$P_{v_{CL}} = \frac{P_v(s) K_{PI_v}(s)}{1 + P_v(s) K_{PI_v}(s)} = \frac{s\frac{K_{P_v}}{m_v} + \frac{K_{I_v}}{m_v}}{s^2 + s(\frac{K_{P_v} - Y_v - Y_{|v|v} 2|v_0|}{m_v}) + \frac{K_{I_v}}{m_v}} \tag{4.35}$$

bringing to the following required control gains:

$$K_{P_v} = 2\xi\omega_n m_v + Y_v + Y_{|v|v} 2|v_0| \qquad K_{I_v} = m_v \omega_n^2 \tag{4.36}$$

The structure of the PI for the sway, being completely equal to the surge one from the equations point of view, will not be here reported for sake of shortness.

4.2.3 PID for Depth Control

To control the heave, a Proportional Integrative Derivative (PID) controller was implemented, whose general structure is explained in C.2. This time, it was chosen a controller acting on the z error, rather than on the velocity w, in order to be sure to bring to zero the error between the desired depth and the actual value. Recalling the PID controller transfer function in the *Laplace* domain,

$$K_{PID_z}(s) = K_{P_z} + \frac{K_{I_z}}{s} + sK_{D_z} \tag{4.37}$$

then it's necessary to define the plant transfer function $P_z(s)$. In a similar way, the starting point is the linearization of the nonlinear model about an the equilibrium point, represented by the goal depth value. The model equation in question is:

$$\dot{w} = -\frac{Z_w}{m_w} w - \frac{Z_{|w|w}}{m_w} |w| w + \frac{\tau_w}{m_w} + \frac{g_w}{m_w} \tag{4.38}$$

So the idea is to compute the plant transfer function for the vertical speed w and then just apply an integrator in cascade, in order to operate on the z value rather than on the speed. The first step, again, is to compute the equilibrium vertical force:

$$0 = -\frac{Z_w}{m_w} w_0 - \frac{Z_{|w|w}}{m_w} |w_0| w_0 + \frac{\tau_{w_0}}{m_w} + \frac{g_w}{m_w} \tag{4.39}$$

resulting in:

$$\tau_{w_0} = Z_w w_0 + Z_{|w|w} |w_0| w_0 - g_w \tag{4.40}$$

Then, one must proceed by linearizing about the following set of equilibrium conditions:

$$w = w_0 + \delta w \qquad \dot{w} = \dot{w}_0 + \delta\dot{w} \qquad \tau_w = \tau_{w_0} + \delta\tau_w \tag{4.41}$$

again, with $\dot{w}_0 = 0$, considering w_0 constant. Then, proceeding with the actual linearization of the dynamic along z,

$$\begin{aligned} \dot{w} = \delta\dot{w} &= \left.\frac{\partial f}{\partial w}\right|_{w_0, \tau_{w0}} \delta w + \left.\frac{\partial f}{\partial w}\right|_{w_0, \tau_{w0}} \delta\tau_w \\ &= \underbrace{(-\frac{Z_w}{m_w} - \frac{Z_{|w|w}}{m_w} 2|w_0|)}_{A_w} \delta w + \underbrace{\frac{1}{m_w}}_{B_w} \delta\tau_w \end{aligned} \tag{4.42}$$

that, moving to the *Laplace* domain, becomes

$$s\tilde{W}(s) = A_w\tilde{W}(s) + B_w\tilde{T_w}(s) \tag{4.43}$$

leading to the following transfer function for the plant $P_w(s)$:

$$P_w(s) \coloneqq \frac{\tilde{W}(s)}{\tilde{T_w}(s)} = \frac{B_w}{s - A_w} = \frac{1}{m_w s + Z_w + Z_{w|w|}2|w_0|} \tag{4.44}$$

Now, since the objective is to control z rather than w, the next step is to integrate, which, in the *Laplace* domain, just consists of a multiplication by $\frac{1}{s}$, resulting in the following plant:

$$P_z(s) = \frac{1}{s(m_w s + Z_w + Z_{|w|w}2|w_0|)} \tag{4.45}$$

Plugging it in the equation 4.20, together with the equation in *Laplace* of the PID gain 4.37, the closed loop transfer function of the overall system is given by:

$$P_{z_{CL}} = \frac{P_z(s)K_{PID_z}(s)}{1 + P_z(s)K_{PID_z}(s)} = \frac{s^2\frac{K_{D_z}}{m_w} + s\frac{K_{P_z}}{m_w} + \frac{K_{I_z}}{m_w}}{s^3 + s^2(\frac{Z_w + Z_{|w|w}2|w_0| + K_{D_z}}{m_w}) + s\frac{K_{P_z}}{m_w} + \frac{K_{I_z}}{m_w}} \tag{4.46}$$

This time, the desired feedback behavior of the system is one of a third-order system, with again damping and natural frequency as the ones before, but with a third pole in p in the negative semi-plane of the *Gaussian plane* for the complex numbers, with the following shape:

$$P_{z_{CL}}(s) = \frac{N(s)}{(s^2 + 2\xi\omega_n s + \omega_n^2)(s + p)} = \frac{N(s)}{s^3 + s^2(p + 2\xi\omega_n) + s(2\xi\omega_n p + \omega_n^2) + \omega_n^2 p} \tag{4.47}$$

Once again, to obtain this desired behavior it's required to match the coefficients of the corresponding powers of s:

$$\frac{Z_w + Z_{|w|w}2|w_0| + K_{D_z}}{m_w} = p + 2\xi\omega_n \qquad \frac{K_{P_z}}{m_w} = 2\xi\omega_n p + \omega_n^2 \qquad \frac{K_{I_z}}{m_w} = \omega_n^2 p \tag{4.48}$$

leading to:

$$K_{D_z} = m_w(p + 2\xi\omega_n) - Z_w - Z_{|w|w}2|w_0| \qquad K_{P_z} = m_w(2\xi\omega_n p + \omega_n^2) \qquad K_{I_z} = m_w\omega_n^2 p \tag{4.49}$$

Once again, ω_n and ξ were chosen respectively as 0.2 and 0.6, while p equal to 10, in order to have the pole then in -10, to have stability.
The structure of the PID controller for the depth is the following:

4.2.4 PID for Yaw Control

The Proportional Integrative Derivative controller for the Yaw has exactly the same scheme of the one designed for the depth, since the goal is to make sure the system follows the desired value of yaw, prior to the yaw rate, so, for the sake of brevity, just the final result will be here reported. The plant of the system, based on the simplified dynamics of the yaw rate from 4.21, before integrating it to obtain the yaw dynamics, is as follows:

$$P_r(s) \coloneqq \frac{\tilde{R}(s)}{\tilde{T_r}(s)} = \frac{1}{m_r s - N_r - N_{r|r|}2|r_0|} \tag{4.50}$$

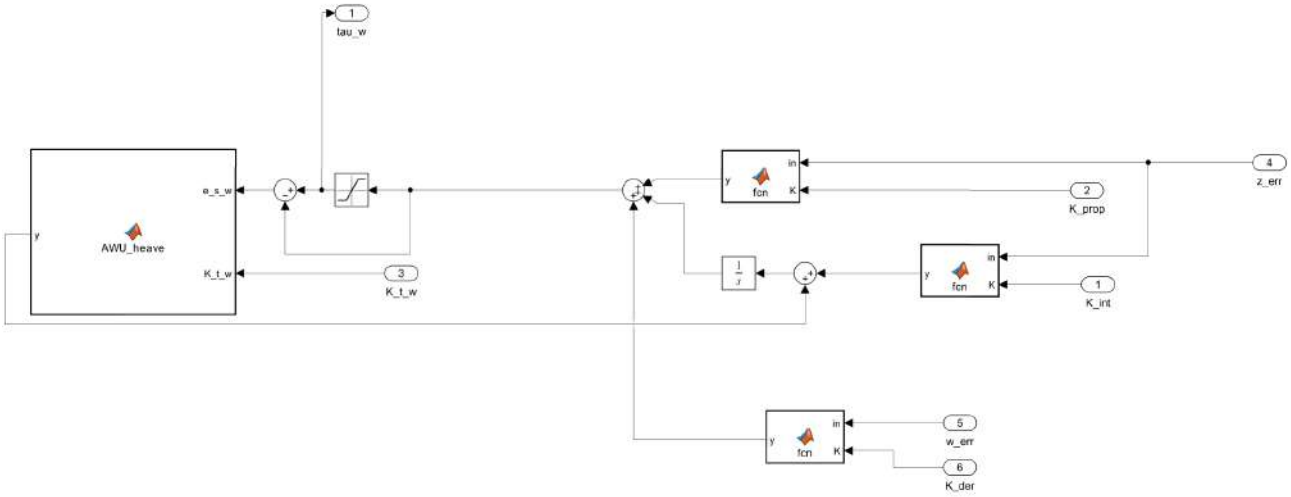


Figure 4.4: PID with Anti-Windup for the depth control.

which, once integrated in the *Laplace* domain, results in:

$$P_\psi(s) = \frac{1}{s}P_r(s) = \frac{1}{s(m_r s - N_r - N_{|r|r}2|r_0|)} \tag{4.51}$$

which, plugged in the closed loop equation, together with the same structure of the PID controller as the depth one, results in the following closed loop transfer function

$$P_\psi(s) = \frac{P_\psi(s)K_{PID_\psi}(s)}{1 + P_\psi(s)K_{PID_\psi}(s)} = \frac{s^2\frac{K_{D_\psi}}{m_r} + s\frac{K_{P_\psi}}{m_r} + \frac{K_{I_\psi}}{m_r}}{s^3 + s^2(\frac{K_{D_\psi} - N_r - N_{|r|r}2|r_0|}{m_r}) + s\frac{K_{P_\psi}}{m_r} + \frac{K_{I_\psi}}{m_r}} \tag{4.52}$$

bringing to the following control gains:

$$K_{D_\psi} = m_r(2\xi\omega_n + p) + N_r + N_{|r|r}2|r_0| \qquad K_{P_\psi} = m_r(2\xi\omega_n p + \omega_n^2) \qquad K_{I_\psi} = m_r\omega_n^2 p \tag{4.53}$$

The structure of the PID for the yaw control will not be here reported, being exactly equal to the depth one, for sake of brevity.

4.2.5 Anti-Windup Strategy

Integral windup is a common issue in controllers with integral action (such as PI and PID), especially when actuators are subject to saturation limits. Windup occurs when the actuator output $u(t)$ is saturated, but the integrator continues to accumulate error, causing delayed recovery, overshoot, and degraded performance when the saturation ends.

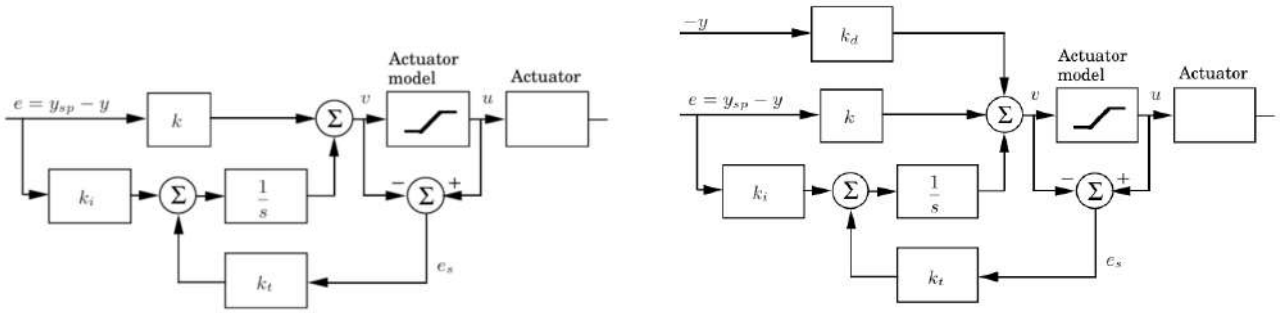
One effective anti-windup strategy is illustrated in Figure 4.5, as it is possible to find in [55]: the system includes an additional feedback loop that becomes active only during saturation. The idea is to compare the actual actuator output $u(t)$ with the raw controller output $v(t)$, producing a feedback signal

$$e_s(t) = u(t) - v(t) \tag{4.54}$$

This signal $e_s(t)$ is then fed back into the integrator input through a gain k_t, modifying the integrator dynamics as follows:

$$\dot{I}(t) = e(t) + k_t e_s(t) \tag{4.55}$$

When the actuator is not saturated, $u(t) \approx v(t)$, hence $e_s(t) \approx 0$, and the feedback loop has no effect. When saturation occurs, $e_s(t) \neq 0$, and the feedback term drives the integrator in the opposite direction, effectively preventing further windup. This ensures that the controller output $v(t)$ remains close to the saturation limits, and it can react immediately when the error changes sign.



(a) PI controller with anti-windup

(b) PID controller with anti-windup

Figure 4.5: PI and PID controllers with anti-windup.

Gain Selection

The feedback gain k_t determines how quickly the integrator is reset during saturation. A large value of k_t results in faster unwinding, but excessive gain may amplify measurement noise and introduce instability. A practical choice is:

$$k_t \approx \frac{1}{T_i} \quad \text{for PI controllers,} \qquad k_t \approx \frac{1}{\sqrt{T_i T_d}} \quad \text{for PID controllers}$$

where

$$T_i = \frac{K_P}{K_I} \qquad T_d = \frac{K_D}{K_P} \tag{4.56}$$

The two structures were implemented in the PI and PID controllers used in this thesis, in order to obtain the forces and torques necessary for every quantity, to be fed to the actuators in the dynamics of the system. For what concerns the PIDs, the error derivative is taken exploiting the physical properties: rather than deriving the error of the quantity, namely the depth z and the angle ψ, it was taken the error on the velocities, respectively the speed along z, w, and the yaw rate, r, comparing it with the desired value, equal to zero in both cases since, at expired transient, the AUV should not rotate or change depth.
It's important to remark that, to be coherent, the errors were computed starting from the quantities in the body frame, rather than the inertial ones, since, starting from this frame, the torque that is then fed to the non-linear dynamics will be computed, which is again referred in the body frame.

4.3 Yaw Normalization

In robotics it is very common and actually standard to normalize angles between $-\pi$ and π, especially in applications such as mobile robotics, SLAM, path planning and state estimation. Normalization is crucial because of angles' periodicity: angles differing by integer multiples of 2π actually represent the same orientation. This numerical difference, if not managed, would result in a huge error from the controller point of view, while, from a physical perspective, the

two angles would be exactly the same. This would result in a control input proportional to this error when not needed, which represents a conceptual error. By normalizing angles, they are kept bounded between a standard range, i.e. $(-\pi, \pi)$, ensuring continuous transitions between adjacent angles and making comparison for error computation reliable. The chosen method for normalization is, throughout the whole work, the following one:

$$\alpha_{norm} = \arctan\left(\frac{\sin(\alpha)}{cos(\alpha)}\right) \tag{4.57}$$

4.4 Retrieval of Docking Station Orientation

In the cases where the docking station's orientation is not well known prior the beginning of the mission, a strategy is necessary to let the AUV understand how to change its orientation in the terminal homing phase. The developed method exploits the use of just one iUSBL on the AUV rather than two different iUSBLs, one on the vehicle and one on the DS, communicating the relative orientation. This contributes to save costs considerably. In order to compute the relative angle, defined as

$$\psi_r \coloneqq \psi_{DS} - \psi_{AUV} \tag{4.58}$$

the necessary equipment is composed of two different transponders on the docking station, one on the left side of the opening and the other one on the right side, the green balls in the image 4.6: this way the vehicle captures two sets of acoustic waves coming as feedback from the docking structure, one for each side, which are the two dark red vectors in the figure, ${}^{DS}P_R$ and ${}^{DS}P_L$. The two vectors, defined in the Docking Station frame, are then subtracted in order to obtain the vector connecting the two, ${}^{DS}\Delta P$, in the following way:

$${}^{DS}\Delta P = {}^{DS}P_R - {}^{DS}P_L \tag{4.59}$$

This vector, because of how the Docking Station frame is built, is exactly aligned with the y_{DS} axis. This is critical to retrieve the orientation between the two bodies. Since the magnitude of the ${}^{DS}\Delta P$ vector is known, again for construction reasons, and is named d, it's then possible to retrieve how the frame of the docking station is oriented with respect to the one of the AUV, by considering one of the planar components of the DS frame.

$$\begin{cases} {}^{A}\boldsymbol{\Delta P_x} = d\cos(\psi_r)\,{}^{\boldsymbol{A}}\boldsymbol{x} + d\sin(\psi_r)\,{}^{\boldsymbol{A}}\boldsymbol{y} \\ {}^{A}\boldsymbol{\Delta P_y} = -d\sin(\psi_r)\,{}^{\boldsymbol{A}}\boldsymbol{x} + d\cos(\psi_r)\,{}^{\boldsymbol{A}}\boldsymbol{y} \end{cases} \tag{4.60}$$

where ${}^{A}\boldsymbol{\Delta P_x}$ is the x component of the ${}^{DS}\boldsymbol{\Delta P}$ vector projected in the frame $\{A\}$ and, similarly, ${}^{A}\boldsymbol{\Delta P_y}$ is the y component of the ${}^{DS}\boldsymbol{\Delta P}$ vector projected in the frame $\{A\}$. Here the ${}^{A}\boldsymbol{\Delta P_y}$ vector was examined, but in a similar procedure the it's possible to use ${}^{A}\boldsymbol{\Delta P_x}$. The two components, along ${}^{\boldsymbol{A}}\boldsymbol{x}$ and ${}^{\boldsymbol{A}}\boldsymbol{y}$, will then be used as measurements to be fed to the Extended Kalman filter in charge of computing the relative orientation ψ_r.

4.5 Estimation Filters

In many real-world applications, such as robotics, navigation, and control systems, it's often needed to estimate the state of a system based on noisy or incomplete measurements. Estimation filters like Kalman Filter (KF) and Extended Kalman Filter (EKF) play a crucial role in this context. These filters allow to combine sensor data and system models to produce more accurate and reliable estimates of the true system state. The Kalman Filter is optimal for linear systems with Gaussian noise, while the Extended Kalman Filter extends this approach to

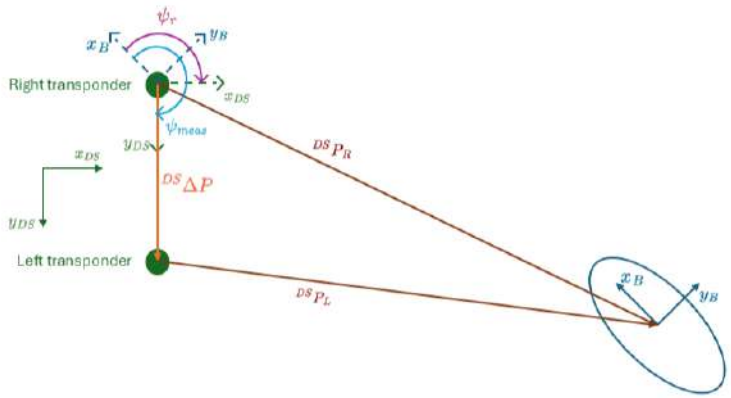


Figure 4.6: ψ_r retrieval method.

nonlinear systems by linearizing the model around the current estimate. Exploiting these filters improves performance, stability, and decision-making in dynamic and uncertain environments. For this reason, in this thesis different estimation filters were implemented, each of them custom for the relative situation, based on sensor availability, the phase of the trajectory and the knowledge concerning the docking station.
In the following subsections, the procedure to derive each estimation filter will be shown, whilst the KF and EKF functioning will be left to the reader in the appendix E.

4.5.1 Phase 1

The first phase is exactly the same for every situation, from an estimation point of view. In this phase it's possible to rely on the position measurements coming from the inverted USBL, the velocity ones from the DVL and the orientation from AHRS, as well as the depth cell. The retrieved quantities are position, orientation, yaw rate and current. Two separate Kalman Filters were implemented, one concerning the xy plane and one on the vertical one, since everything is decoupled.

Planar Filter

The model of the dynamics of the planar part is given by:

$$\begin{cases} \dot{x} = v_{m_x} + v_{c_x} + \xi_x \\ \dot{y} = v_{m_y} + v_{c_y} + \xi_y \\ \dot{\psi} = r + \xi_\psi \\ \dot{r} = \xi_r \\ \dot{v}_{c_x} = \xi_{v_{c_x}} \\ \dot{v}_{c_y} = \xi_{v_{c_y}} \end{cases} \tag{4.61}$$

where x, y are the coordinates of the planar position measured by the iUSBL, v_{m_x}, v_{m_y} are the xy components of the relative velocity, with respect to the fluid, measured by the Doppler Velocity Log, v_{c_x}, v_{c_y} are the horizontal components of the current velocity, ψ the orientation, r the rotational rate, and the ξ are the state noises. Note the dynamics of the currents and of the yaw rate: they could be simplified in this way thanks to the assumption of constant current (or very slowly varying over time) and quasi-zero rotational velocity along the z axis, since this

is the desired behavior at expired transient.
The model of the observation, again on the xy plane, is the following:

$$\begin{cases} y_1 = x + \eta_1 \\ y_2 = y + \eta_2 \\ y_3 = \psi + \eta_3 \end{cases} \tag{4.62}$$

where the $\eta_{1,2,3}$ are the modeled measurement noise.
Discretizing the dynamics, through the Forward Euler method, which is suitable for this problem, since the Δt is much smaller than the dynamics of the system and results in a fast and easy discretization of the model, assuming the noise constant on a limited interval, the equation 4.61 becomes the following:

$$\begin{cases} \dfrac{x_{k+1} - x_k}{\Delta t} = v_{m_{x_k}} + v_{c_{x_k}} + \xi_x \\ \dfrac{y_{k+1} - y_k}{\Delta t} = v_{m_{y_k}} + v_{c_{y_k}} + \xi_y \\ \dfrac{\psi_{k+1} - \psi_k}{\Delta t} = r_k + \xi_\psi \\ \dfrac{r_{k+1} - r_k}{\Delta t} = \xi_r \\ \dfrac{v_{c_{x_{k+1}}} - v_{c_{x_k}}}{\Delta t} = \xi_{v_{cx}} \\ \dfrac{v_{c_{y_{k+1}}} - v_{c_{y_k}}}{\Delta t} = \xi_{v_{cy}} \end{cases} \Rightarrow \begin{cases} x_{k+1} = x_k + (v_{m_{x_k}} + v_{c_{x_k}} + \xi_x)\Delta t \\ y_{k+1} = y_k + (v_{m_{y_k}} + v_{c_{y_k}} + \xi_y)\Delta t \\ \psi_{k+1} = \psi_k + (r_k + \xi_\psi)\Delta t \\ r_{k+1} = r_k + \xi_r \Delta t \\ v_{c_{x_{k+1}}} = v_{c_{x_k}} + \xi_{v_{cx}} \Delta t \\ v_{c_{y_{k+1}}} = v_{c_{y_k}} + \xi_{v_{cy}} \Delta t \end{cases} \tag{4.63}$$

From this discretization, it's possible to proceed with the composition of the vectors and matrices which characterize the Kalman filter. Here will follow the state, input, output, state noise and measurement noise vectors in this order:

$$\mathrm{x} = \begin{bmatrix} x \\ y \\ \psi \\ r \\ v_{c_x} \\ v_{c_y} \end{bmatrix}, \quad \mathrm{u} = \begin{bmatrix} v_{m_x} \\ v_{m_y} \end{bmatrix}, \quad \mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}, \quad \mathrm{w} = \begin{bmatrix} \xi_x \\ \xi_y \\ \xi_\psi \\ \xi_r \\ \xi_{v_{cx}} \\ \xi_{v_{cy}} \end{bmatrix}, \quad \mathrm{v} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}$$

while the state, input, output, state and output noise covariances, as well as state noise coefficients matrices are, respectively:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$Q = \begin{bmatrix} \xi_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \xi_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi_\psi^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi_r^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \xi_{v_{cx}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \xi_{v_{cy}}^2 \end{bmatrix}, \qquad R = \begin{bmatrix} \eta_1^2 & 0 & 0 \\ 0 & \eta_2^2 & 0 \\ 0 & 0 & \eta_3^2 \end{bmatrix}, \qquad G = \Delta t I_{6\times 6}$$

Vertical Filter

The dynamics of the model on the vertical part follows a similar scheme, except that there's no angular contribution, that is,

$$\begin{cases} \dot{z} = v_{m_z} + v_{c_z} + \xi_z \\ \dot{v}_{c_z} = \xi_{v_{c_z}} \end{cases} \tag{4.64}$$

where z is the depth coordinate measured by the depth cell, v_{m_z} the velocity measured by the DVL, v_{c_z} the vertical current velocity, and the $\xi_{z,v_{c_z}}$ are the state noises.
The model of the observation is given by:

$$y = z + \eta_z \tag{4.65}$$

where η_z is the depth cell measurement noise. Discretizing the dynamics, again with the Forward Euler method, the equation 4.64 becomes

$$\begin{cases} \dfrac{z_{k+1} - z_k}{\Delta t} = v_{m_{z_k}} + v_{c_{z_k}} + \xi_z \\ \dfrac{v_{c_{z_{k+1}}} - v_{c_{z_k}}}{\Delta t} = \xi_{v_{c_z}} \end{cases} \Rightarrow \begin{cases} z_{k+1} = z_k + v_{m_{z_k}}\Delta t + v_{c_{z_k}}\Delta t + \xi_z \Delta t \\ v_{c_{z_{k+1}}} = v_{c_{z_k}} + \xi_{v_{c_z}}\Delta t \end{cases} \tag{4.66}$$

Now it is possible to proceed with the composition of the vectors and matrices to be used in the Kalman filter. In what follows the state, input, output, state noise and measurement noise vectors or scalars, in this order are given by

$$\mathrm{x} = \begin{bmatrix} z \\ v_{c_z} \end{bmatrix}, \quad \mathrm{u} = v_{m_z}, \quad \mathrm{y} = y, \quad \mathrm{w} = \begin{bmatrix} \xi_z \\ \xi_{v_{c_z}} \end{bmatrix}, \quad \mathrm{v} = \eta_z$$

while the state, input, output, state and output noise covariances, as well as state noise coefficient matrices are, respectively:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} \xi_z^2 & 0 \\ 0 & \xi_{v_{c_z}}^2 \end{bmatrix}, \quad R = \eta_z^2, \quad G = \Delta t I_{2\times 2}$$

4.5.2 Phase 2, Phase 3 Phase 4

The second, third and fourth phases were implemented in different ways according to the knowledge on the docking station position and orientation, but, in each case, for every behavior, the same filters were implemented.

Known DS Position

This case, since the position of the docking station is well known and the sensor used are exactly the same ones of the Phase 1, is implemented exactly as the previous filter, in every of the final three behaviors, both for the horizontal and vertical planes; hence, for sake of brevity, the equations, matrices and vectors will not be here reported.

Unknown DS Position

In all the other cases, where the DS position is just approximately known, the estimation filters for the second, third and fourth behaviors are equal one another but different with respect to Phase 1, because the exploited position sensor now is the *range-measurement*, instead of the iUSBL. Because of this, the equations of the measurement change, and, for reasons of

compactness, only one filter was implemented, merging the xy information with the vertical ones. Also, in this phase, since the AUV is assumed to be already near the seabed, it's possible to measure the current velocity, other than the speed of the vehicle itself relative to the fluid, performed through alternative shooting the water and the seabed with sound waves.
Since the range measurement is non-linear, as it will be shown later on, the use of a Kalman filter is no more possible; rather an Extended Kalman filter will be used.
The model of the filter dynamics is the following:

$$\begin{cases} \dot{x} = v_{m_x} + v_{c_x} + \xi_x \\ \dot{y} = v_{m_y} + v_{c_y} + \xi_y \\ \dot{z} = v_{m_z} + v_{c_z} + \xi_z \\ \dot{v}_{c_x} = \xi_{v_{c_x}} \\ \dot{v}_{c_y} = \xi_{v_{c_y}} \\ \dot{v}_{c_z} = \xi_{v_{c_z}} \\ \dot{\psi} = r + \xi_\psi \\ \dot{r} = \xi_r \end{cases} \tag{4.67}$$

where the quantities here represented are the same one of the previous filter.
The observation model now becomes a bit different:

$$\begin{cases} y_1 = ||x^2 + y^2 + z^2|| + \eta_1 \\ y_2 = v_{c_x} + \eta_2 \\ y_3 = v_{c_y} + \eta_3 \\ y_4 = v_{c_z} + \eta_4 \\ y_5 = \psi + \eta_5 \\ y_6 = z + \eta_6 \end{cases} \tag{4.68}$$

where the first measurement represents the 3D range distance between the AUV and the beacon.
The dyanmics is still linear, so it's possible to proceed with the Forward Euler method discretization, obtaining the following set of equations:

$$\begin{cases} \dfrac{x_{k+1} - x_k}{\Delta t} = v_{m_{x_k}} + v_{c_{x_k}} + \xi_x \\ \dfrac{y_{k+1} - y_k}{\Delta t} = v_{m_{y_k}} + v_{c_{y_k}} + \xi_y \\ \dfrac{z_{k+1} - z_k}{\Delta t} = v_{m_{z_k}} + v_{c_{z_k}} + \xi_z \\ \dfrac{v_{c_{x_{k+1}}} - v_{c_{x_k}}}{\Delta t} = \xi_{v_{c_x}} \\ \dfrac{v_{c_{y_{k+1}}} - v_{c_{y_k}}}{\Delta t} = \xi_{v_{c_y}} \\ \dfrac{v_{c_{z_{k+1}}} - v_{c_{z_k}}}{\Delta t} = \xi_{v_{c_z}} \\ \dfrac{\psi_{k+1} - \psi_k}{\Delta t} = r_k + \xi_\psi \\ \dfrac{r_{k+1} - r_k}{\Delta t} = \xi_r \end{cases} \Rightarrow \begin{cases} x_{k+1} = x_k + (v_{m_{x_k}} + v_{c_{x_k}} + \xi_x)\Delta t \\ y_{k+1} = y_k + (v_{m_{y_k}} + v_{c_{y_k}} + \xi_y)\Delta t \\ z_{k+1} = z_k + (v_{m_{z_k}} + v_{c_{z_k}} + \xi_z)\Delta t \\ v_{c_{x_{k+1}}} = v_{c_{x_k}} + \xi_{v_{c_x}}\Delta t \\ v_{c_{y_{k+1}}} = v_{c_{y_k}} + \xi_{v_{c_y}}\Delta t \\ v_{c_{z_{k+1}}} = v_{c_{z_k}} + \xi_{v_{c_z}}\Delta t \\ \psi_{k+1} = \psi_k + (r_k + \xi_\psi)\Delta t \\ r_{k+1} = r_k + \xi_r \Delta t \end{cases} \tag{4.69}$$

Now, it's possible to define the vectors and also the matrices characteristic of the Kalman filter, since the dynamic part is still linear, so they will be Linear Time-Invariant (LTI), and

so possible to compute it offline, prior to the beginning of the simulation, in order to save computational time. Here following, the state, input, output, state noise and measurement noise vectors, in this order:

$$\mathrm{x} = \begin{bmatrix} x \\ y \\ z \\ v_{c_x} \\ v_{c_y} \\ v_{c_z} \\ \psi \\ r \end{bmatrix}, \quad \mathrm{u} = \begin{bmatrix} v_{m_x} \\ v_{m_y} \\ v_{m_z} \end{bmatrix}, \quad \mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix}, \quad \mathrm{w} = \begin{bmatrix} \xi_x \\ \xi_y \\ \xi_z \\ \xi_{v_{c_x}} \\ \xi_{v_{c_y}} \\ \xi_{v_{c_z}} \\ \xi_\psi \\ \xi_r \end{bmatrix}, \quad \mathrm{v} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \\ \eta_6 \end{bmatrix}$$

while the state, input, state noise coefficients, state and output noise covariances matrices are:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad G = \Delta t I_{8\times 8},$$

$$Q = \begin{bmatrix} \xi_x^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \xi_y^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi_z^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi_{v_{c_x}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \xi_{v_{c_y}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \xi_{v_{c_z}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \xi_\psi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_r^2 \end{bmatrix}, \qquad R = \begin{bmatrix} \eta_1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \eta_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \eta_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \eta_4^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \eta_5^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \eta_6^2 \end{bmatrix}$$

For what concerns the output, because of the non-linear *range-measurement*, the whole output procedure must be linearized about the current estimate, making, as previously mentioned, the whole filter an Extended Kalman filter.
Considering again the equation 4.68 and expanding the first equation, what's obtained is

$$\begin{cases} y_1 = h_1(\mathrm{x}) = \sqrt{x^2 + y^2 + z^2} + \eta_1 \\ y_2 = h_2(\mathrm{x}) = v_{c_x} + \eta_2 \\ y_3 = h_3(\mathrm{x}) = v_{c_y} + \eta_3 \\ y_4 = h_4(\mathrm{x}) = v_{c_z} + \eta_4 \\ y_5 = h_5(\mathrm{x}) = \psi + \eta_5 \\ y_6 = h_6(\mathrm{x}) = z + \eta_6 \end{cases} \tag{4.70}$$

The next step is to compute the partial derivatives of the measurement equations in order to

then evaluate them in the current estimated point and compose the output Jacobian matrix:

$$H_k = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2+z^2}} & \frac{y}{\sqrt{x^2+y^2+z^2}} & \frac{z}{\sqrt{x^2+y^2+z^2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Bigg|_{\mathrm{x}=\hat{\mathrm{x}}_\mathrm{k}} \tag{4.71}$$

This matrix is computed at each time step since some elements of the matrix have to be evaluated on the current estimate, so it cannot be precomputed offline as the other ones.

4.5.3 Relative Orientation

A specific filter was designed for the estimation of the relative orientation between the AUV and the docking station. This filter estimates the angle and the bias which might be present in the gyroscope measurements, as it's possible to analyze from the following dynamics:

$$\begin{cases} \dot{\psi}_r = r_m + \mu + \xi_{\psi_r} \\ \dot{\mu} = \xi_\mu \end{cases} \tag{4.72}$$

where ψ_r is the relative angle between the structure and the vehicle, μ is the bias, inverted of sign, of the gyro, r_m is the measured angular rate, always referred to the relative angle, $\xi_{\psi_r,\mu}$ are the state noises. For what concerns the observation, the following are non-linear output equations, coming from the second component of the system at 4.60:

$$\begin{cases} y_1 = -d\sin(\psi_r) + \eta_1 \\ y_2 = d\cos(\psi_r) + \eta_2 \end{cases} \tag{4.73}$$

where the first quantity is the ${}^{A}\boldsymbol{\Delta P_y}$ component along ${}^{\boldsymbol{A}}\boldsymbol{x}$, whilst the second one is along ${}^{\boldsymbol{A}}\boldsymbol{y}$. The model dynamics is linear, so it's possible to use again the Forward Euler method to discretize the equations and to compose the offline filter matrices.

$$\begin{cases} \dfrac{\psi_{r_{k+1}} - \psi_{r_k}}{\Delta t} = r_{m_k} + \mu_k + \xi_{\psi_r} \\ \dfrac{\mu_{k+1} - \mu_k}{\Delta t} = \xi_\mu \end{cases} \Rightarrow \begin{cases} \psi_{r_{k+1}} = \psi_{r_k} + r_{m_k}\Delta t + \mu_k \Delta t + \xi_{\psi_r}\Delta t \\ \mu_{k+1} = \mu_k + \xi_\mu \Delta t \end{cases} \tag{4.74}$$

At this point, it's possible to compose the vectors or scalars and the matrices of the filter, which, for the state dynamics, are still linear and possible to compute offline. The state, input, output, state noise and measurement noise vectors are as follows:

$$\mathrm{x} = \begin{bmatrix} \psi_r \\ \mu \end{bmatrix}, \quad \mathrm{u} = r_{m_k}, \quad \mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathrm{w} = \begin{bmatrix} \xi_{\psi_r} \\ \xi_\mu \end{bmatrix}, \quad \mathrm{v} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$$

while the state, input, state and output noise covariances and state noise coefficients matrices are

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}, \qquad Q = \begin{bmatrix} \xi_{\psi_r}^2 & 0 \\ 0 & \xi_\mu^2 \end{bmatrix}, \qquad R = \begin{bmatrix} \eta_1^2 & 0 \\ 0 & \eta_2^2 \end{bmatrix}, \qquad G = \Delta t I_{2\times 2}$$

In what concerns the output, consisting of two non-linear equations, they have to be linearized about the current estimate, making the filter an EKF.
Starting from equation 4.73, the output Jacobian matrix is given by

$$H_k = \begin{bmatrix} -d\cos(\psi_r) & 0 \\ -d\sin(\psi_r) & 0 \end{bmatrix}\Bigg|_{\mathrm{x}=\hat{\mathrm{x}}_\mathrm{k}} \tag{4.75}$$

This matrix is computed at each time step because of the evaluation of the entries on the current state estimate.
Even if the output Jacobian has a non-maximum rank, it's necessary to use both equations because of the highly non-observable nature of this problem.

4.6 Simulation Environment

The numerical simulations were carried out in an environment custom made for this problem. For every of the four scenarios, namely the different cases of knowledge on the docking station combined to the quantity used to maneuver the AUV, a specific simulation environment was created on Simulink, coupled with the data coming from a MATLAB script.
The general structure of the Simulink environment is made by three different subsystems, as it's possible to notice in upper part of the Figure 4.7, one for Phase 1, one for Phase 2 and a block for Phase 3 merged with the last one, for reasons of compactness. All the data relative to position, orientation and body speeds are collected in the block on the left side of the scheme, which will register them in the MATLAB workspace, in order to make a proper plot of the scene at the end of the simulation. The data relative to the errors, both the estimate and control ones, are collected in two separate blocks, respectively in the bottom right and bottom left blocks. The state machine logic is in the MATLAB function block named `STATE_MACHINE`. The logic to transition between one phase and the next one is implemented exploiting enabled subsystems.

4.6.1 Phase 1

The first phase, being equal for every scenario, will be reported only once. As it's possible to notice in 4.8, based on the estimation of the position with respect to the ship, when the threshold distance `max_dist_from_ship` is reached, the `stop_flag` is raised to 1 and this determines when this subsystem is turned off, having a logic inverter after the flag signal, connected to the enable of the same block, see Figure 4.7. Also, some `Data Store` blocks are used in order to store the final estimated value from this step and use it in the next behavior, to be able to start the new estimate with a more accurate expected value and covariance.

4.6.2 Phase 2

In the next phase, the structure is similar to the first one. There's a slight distinction to be made between the case when only the orientation of the DS is not known, that is the very first treated case, and the other ones. In the first mentioned case the structure is exactly equal to the former stage, in order to recycle the implemented material, with the only exception that the KFs, both for the planar and vertical estimate, use the last estimate and covariance of the previous filter, while in the Phase 1 the initial conditions were zero, since there was no clue on the possible values. While in the other cases, the structure is again very similar, with the only difference that only one EKF was implemented for both directions, planar and vertical, as it was mentioned in 4.5.2, with the proper initial expected values and covariances coming from

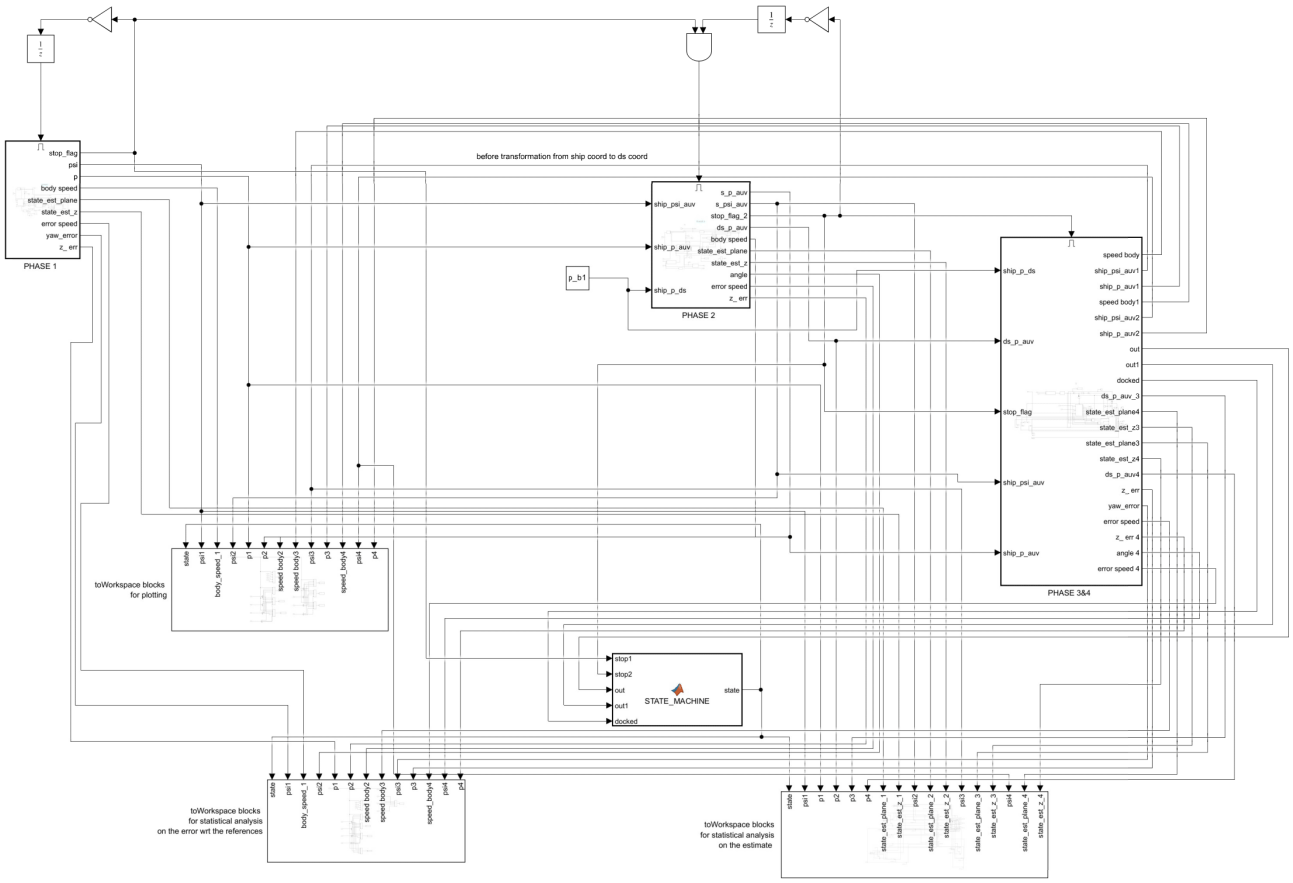


Figure 4.7: Simulation environment.

the previous stage, as it's possible to notice in Figure 4.9. The second phase, as it's possible to see in the big scheme in 4.7, is enabled only when the `stop_flag` of the first subsystem is 1, meaning that the first phase is done, and the `stop_flag2`, of the block itself, is 0, that is prior the AUV reaches the threshold distance from the DS; this in order to guarantee that the block stops running and registering data once the vehicle is close enough to the structure and switches to the next stage.

4.6.3 Phase 3&4

These two phases are grouped together, as in Figure 4.10, because, prior deciding whether to enter the third or the fourth stage, a physical check, from the sensor's point of view, must be done: once the system enters this block, after the `stop_flag2` reached 1, the whole block is turned on and a custom made simulator of the iUSBL is activated.

iUSBL Simulator

This simulator computes whether the robot is in the cone of the docking station or not, based on the following algorithm.
The starting point is defining the cone direction based on the actual angle value of the docking station α_{DS} with respect to the inertial frame, or *NED* convention, which gives the unit direction

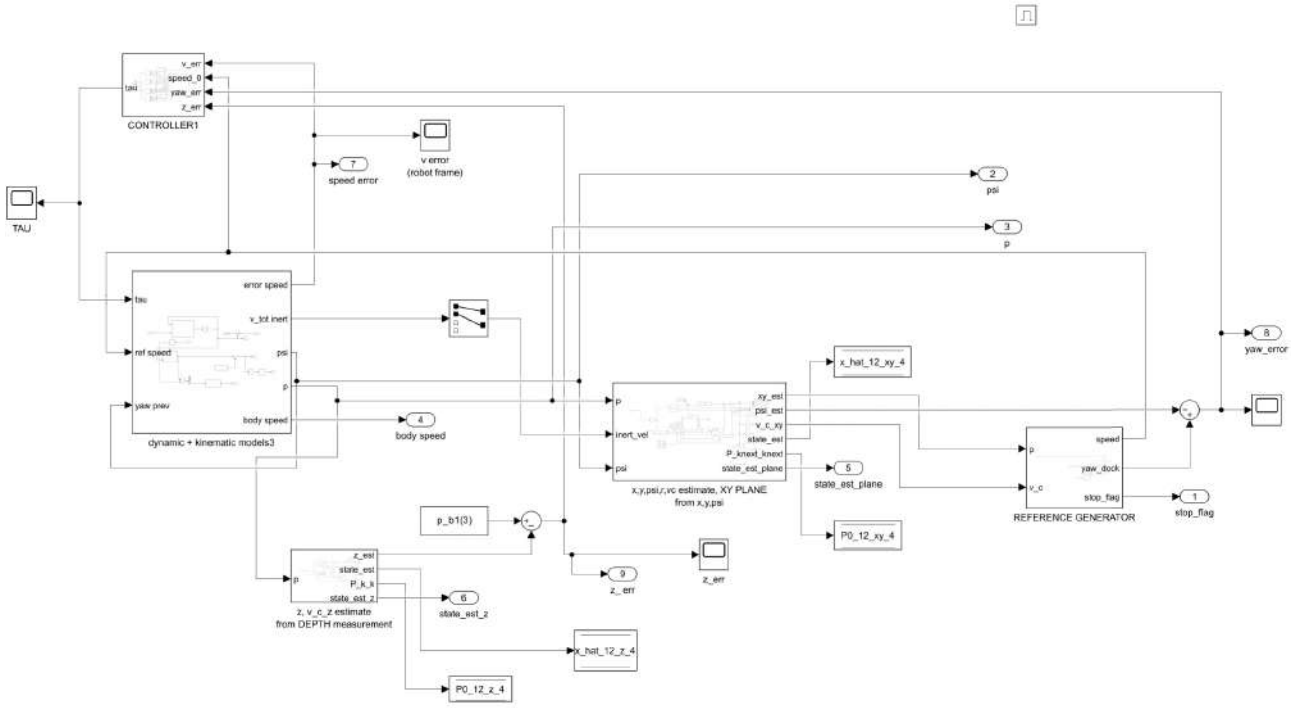


Figure 4.8: Phase 1 scheme.

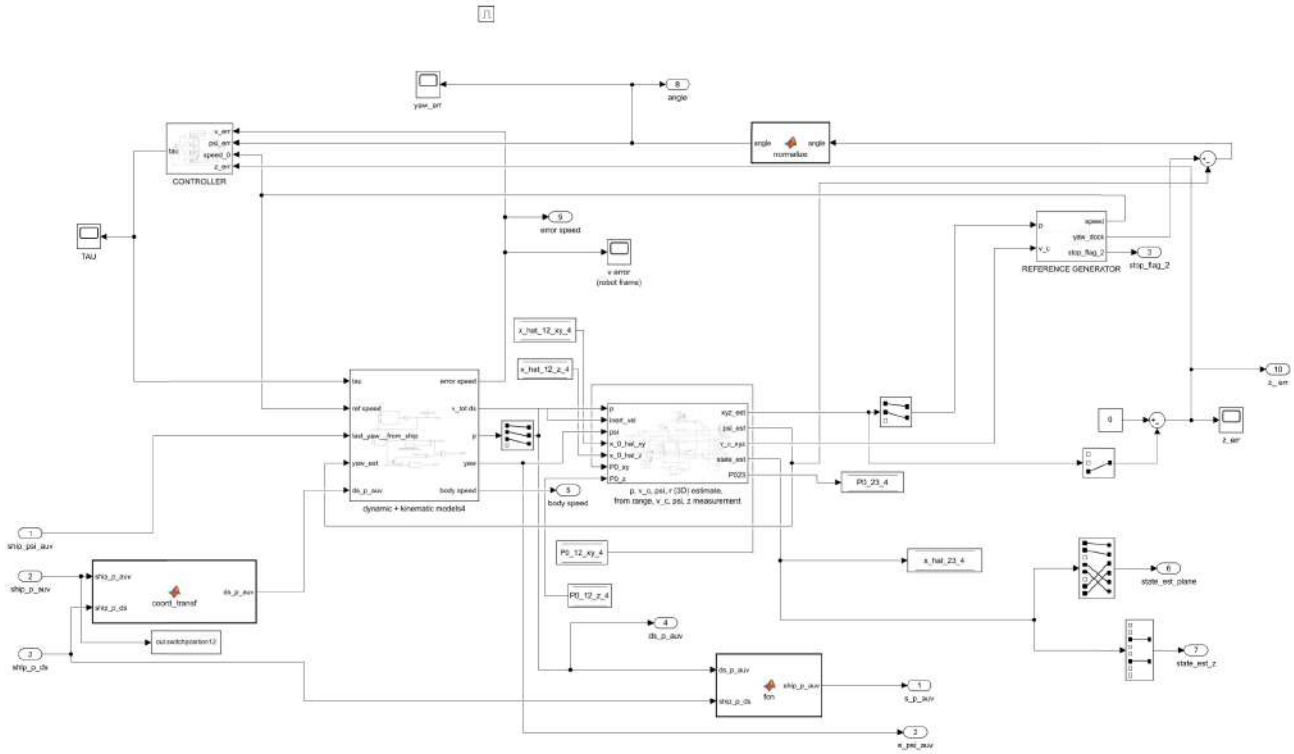


Figure 4.9: Phase 2 - Unknown DS position model structure.

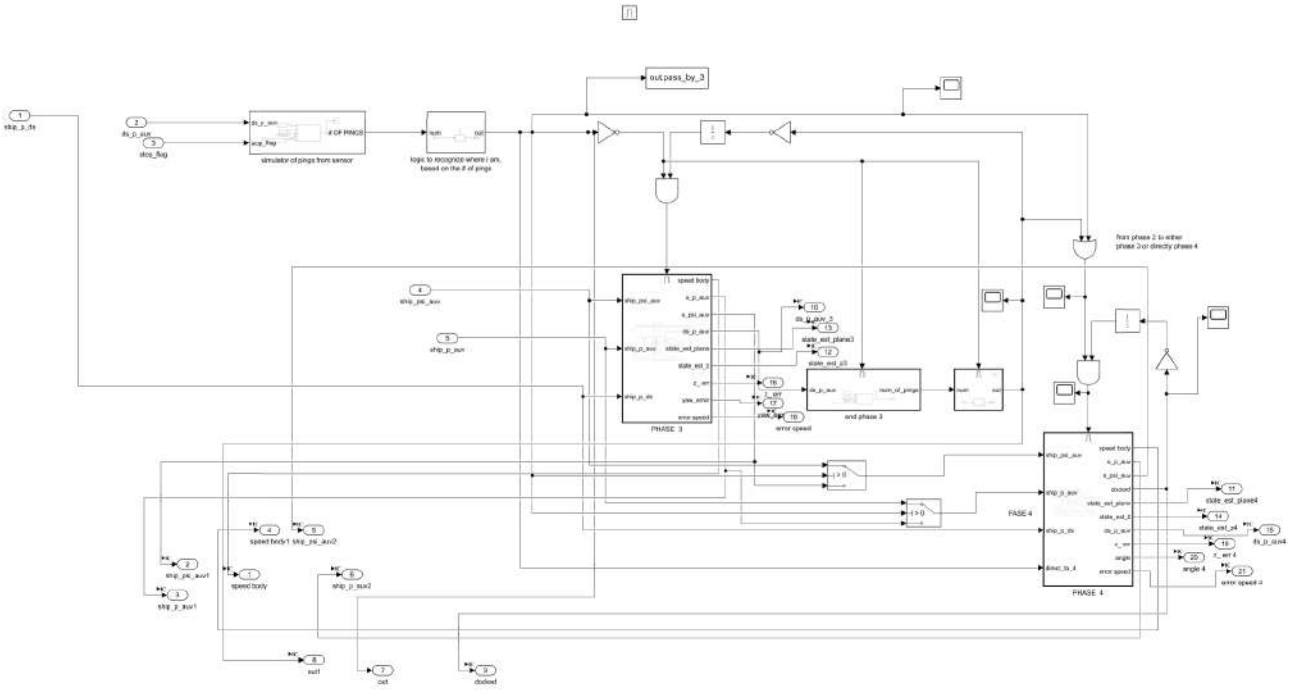


Figure 4.10: Phase 3&4

vector:

$$^{I}\boldsymbol{d} = \begin{bmatrix} \cos(\alpha_{\text{DS}}) \\ \sin(\alpha_{\text{DS}}) \end{bmatrix} \tag{4.76}$$

The vector is computed from the real physical data of the orientation of the DS, whether or not the AUV knows it, since it's a quantity which is required just to simulate the feedback of the sensor. The vector pointing from the DS to the AUV is computed as the difference between their positions:

$$^{DS}\boldsymbol{v} = {}^{DS}\boldsymbol{p}_{\text{auv}} - {}^{DS}\boldsymbol{p}_{\text{ds}}. \tag{4.77}$$

To check if the AUV lies in the direction of the cone, the cosine of the angle between $^{DS}\boldsymbol{v}$ and $^{DS}\boldsymbol{d}$ was computed using the normalized dot product:

$$\cos(\theta) = \frac{^{DS}\boldsymbol{v} \cdot {}^{DS}\boldsymbol{d}}{\|^{DS}\boldsymbol{v}\|}. \tag{4.78}$$

The AUV is considered to be inside the cone if this angle is smaller or equal to the half-opening angle δ, i.e., if the cosine of θ is bigger than the cosine of δ:

$$\frac{^{DS}\boldsymbol{v} \cdot {}^{DS}\boldsymbol{d}}{\|^{DS}\boldsymbol{v}\|} \geq \cos(\delta), \tag{4.79}$$

and, additionally, it's required that the $||^{DS}\boldsymbol{v}||$ is greater than zero, being at the denominator, so it's necessary to require also

$$\|^{DS}\boldsymbol{v}\| > 0. \tag{4.80}$$

If both conditions are satisfied, the AUV is considered inside the cone and the number of pings is set accordingly:

$$\texttt{num\_of\_pings} = \begin{cases} 1, & \text{if the AUV is outside the FoV - Phase 3} \\ 2, & \text{if the AUV is inside the FoV - Phase 4.} \end{cases} \tag{4.81}$$

Phase 3

In all the different cases, the structure of the third phase is equal to the one it had in the previous scenario, so for sake of brevity it won't be here reported. The only thing changing is the guidance policy, being that the AUV is no more going in a straight line but in a circular one; but, besides this, everything is equal.
This block is activated at the final moment of the Phase 2, only in case the output of the iUSBL is only one ping and stays active until the vehicle doesn't receive two pings back from the simulator of the sensor. Once this happens, the output flag of the structure to interpret the number of pings goes to 1 and the `AND` enabling the block of the Phase 3 goes to 0, shutting down the logic.
Once again, the estimate filters are initialized with the values coming from the former phase.

Phase 4

This block is turned on in two possible conditions: either from the end of the third phase, if the AUV was outside the field of view of the DS, or at the end of the second one, if the AUV was already in the FoV. This is interpreted through an `OR` logic, with an `AND` in cascade, to check whether the `docked` variable is high or not: if the flag is 1, the `AND` goes to 0, because of the inversion of the output of the `docking` variable, and deactivates the whole subsystem, otherwise it lets it run, until the docking is performed. Furthermore, once again, the filter uses as initial estimate the last estimates of the previous stage, both expected value and covariance. Again, in this case, things are slightly different for the case of known or unknown DS position: the main difference is always the use of two distinct KF for the vertical and planar directions in the first scenario, exactly as the previous stages. Another difference in the structure comes out based on whether the AUV already knows the DS orientation with good accuracy or not: if not, the structure requires the EKF mentioned in 4.5.3 and shown in 4.11, in the block after the dynamic model, named *relative yaw/bias estimate from vector measurement*, in order to estimate the relative orientation between itself and the structure. Since the orientation of the docking station is estimated starting from this block, it takes some seconds before getting a stable value of it, therefore the actual reference guidance will be activated, through an enable, once a required threshold between two consecutive estimates is achieved, handled through the `check_yaw_dock` MATLAB function block. If the orientation of the Docking Station is already known, as in Figure 4.12, there is no need for the relative orientation estimation filter.
The AUV is considered docked once it reaches a required threshold from the center of mass of the docking station: once that condition is satisfied, the simulation is stopped.

4.7 State Machine

In order to visualize in an easier way the behavior in which the robot is at each time step, a sequential state machine logic was built. Here in the flowchart it's possible to notice the steps in each behavior, or phase, and the conditions to transition from one step to the next one. The phases are the ones described in 3.1.3 and the output of the MATLAB function block `STATE_MACHINE`, at the center of the figure 4.7, is exactly this value, going from 1 to 4 depending on the behavior the AUV is currently having. This value determines the switch between the different values registered by the `toWorkspace` block, in order to perform the plot or evaluate the errors at the proper stage of the simulation.

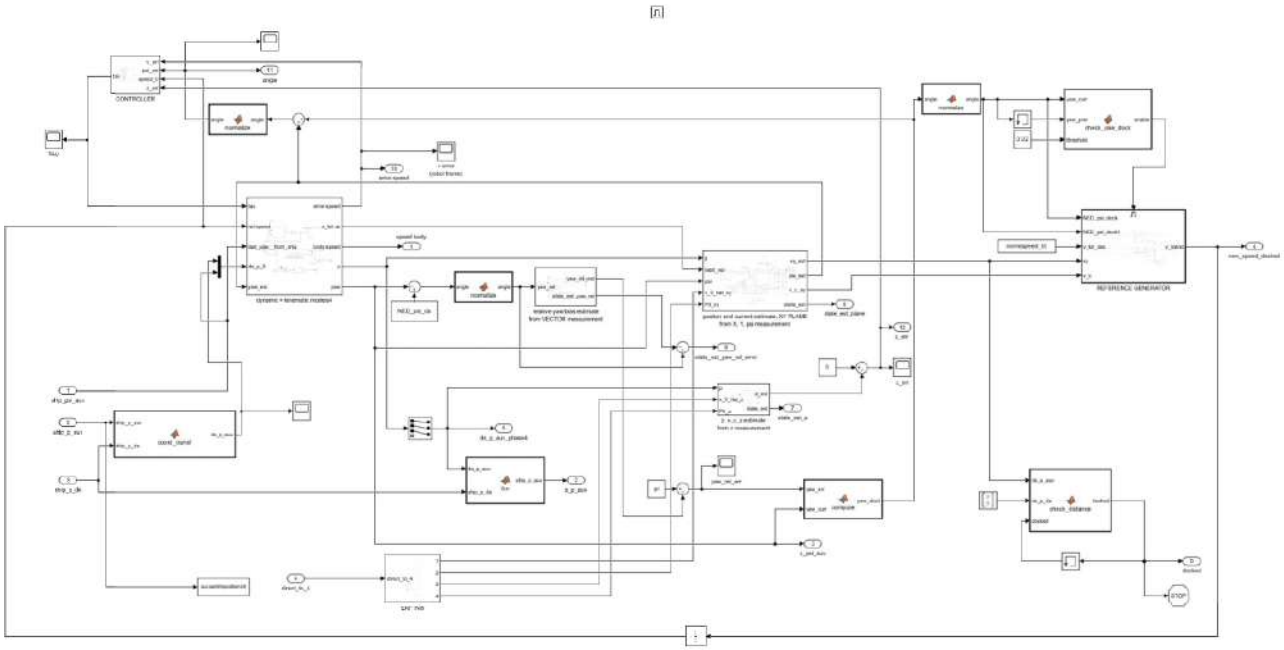


Figure 4.11: Phase 4 - Unknown DS Orientation

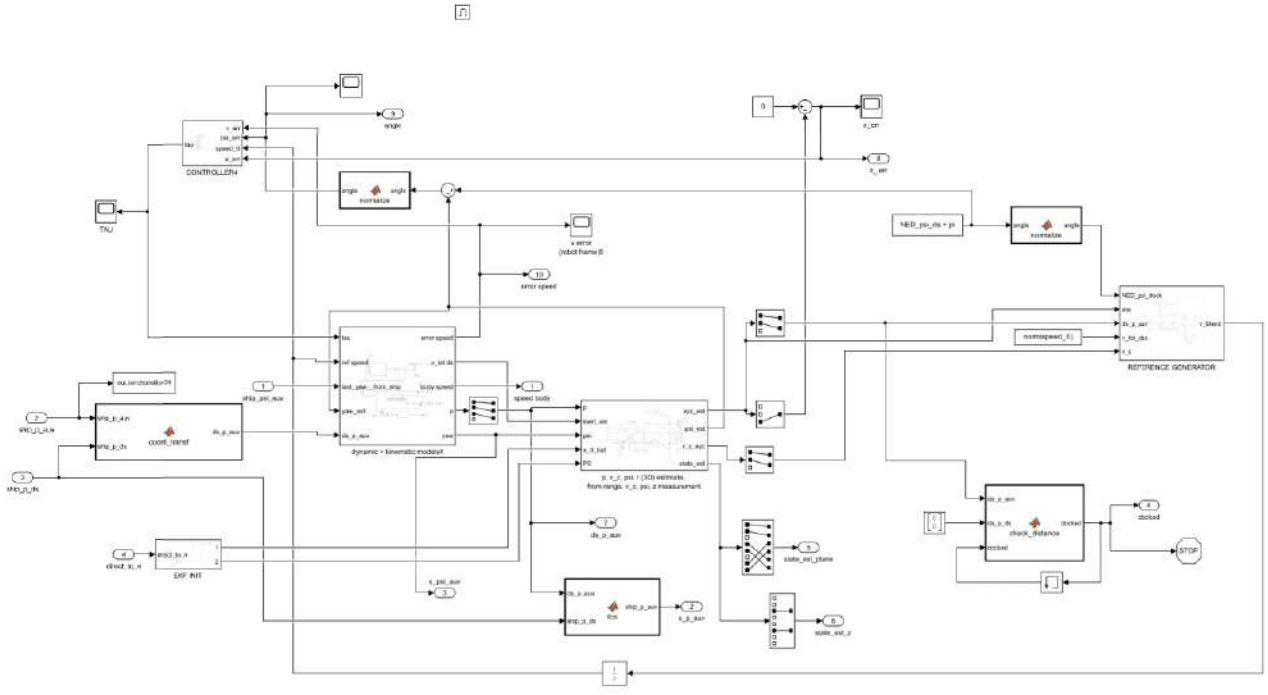


Figure 4.12: Phase 4 - Known DS Position

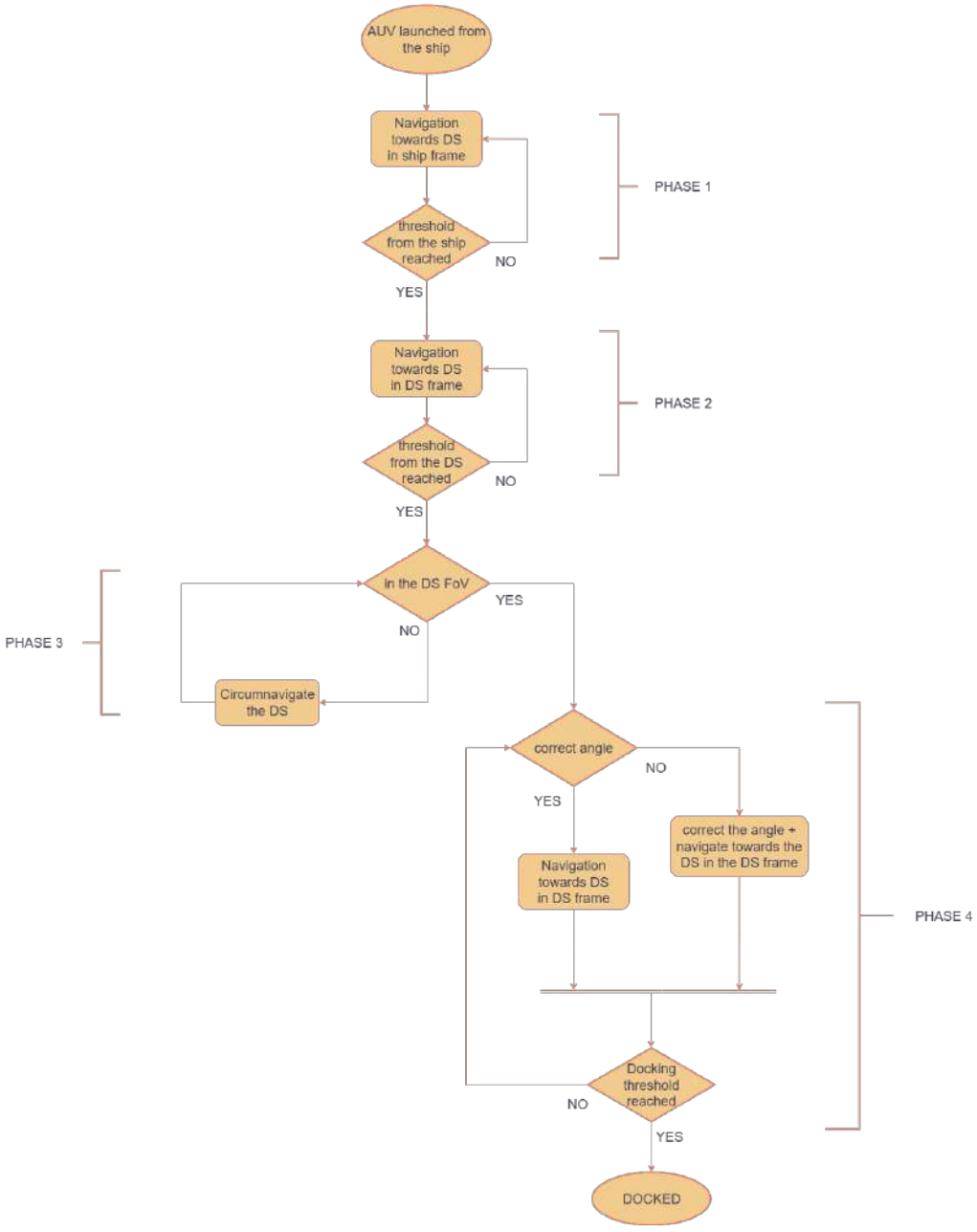


Figure 4.13: Flowchart of the processes.

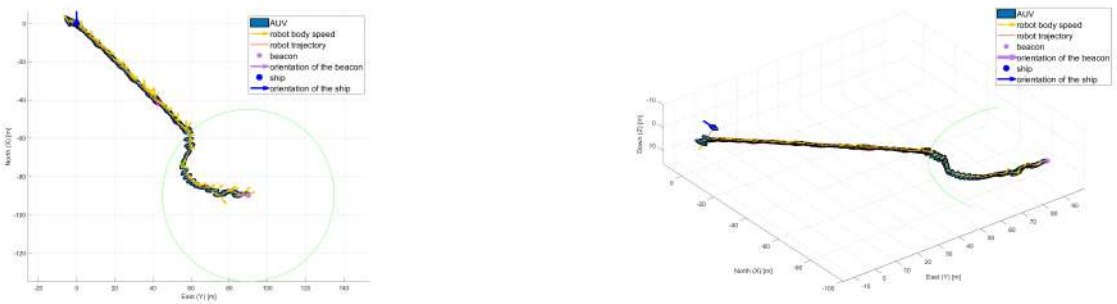
Chapter 5

Numerical Simulations

In this section the results of the numerical simulations, run in the previously described custom-developed environment in Simulink, are presented in order to evaluate and compare the performance of the different algorithms, in particular of the controllers and of the estimation filters. The idea is to visualize the statistics on the errors between the reference values and the actual ones, through the repetition of randomic experiments, with the *Monte Carlo* method.

5.1 Plots in 2D/3D

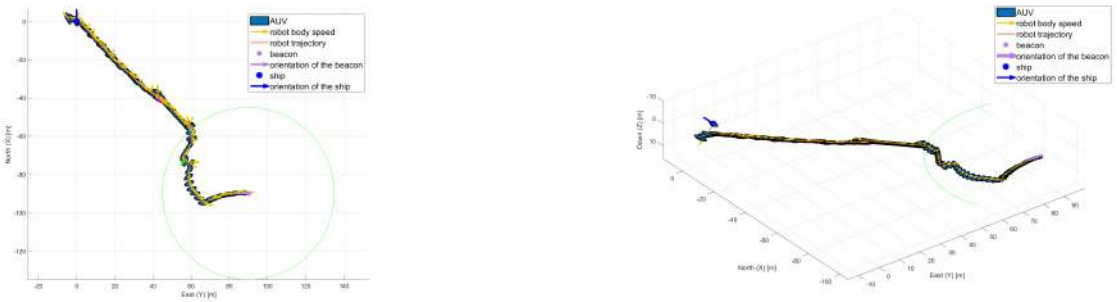
In this section, an overview of the trajectory performed by the AUV is given, both in 2D and in 3D, capturing the different phases of the motion, using the same setting of initial conditions and docking conditions which are also used later on to run the simulations for the Monte Carlo statistics. The performances of the single algorithms will be deeply analyzed in the next sections. In the meanwhile, it's possible to distinguish, in all the proposed algorithms, a first trajectory, represented by a straight line, up to the pink dot, identifying the switch between the first and second phases, followed by another straight trajectory, this time performed with respect to the DS frame, representing the second phase. Once the vehicle reaches the minimum threshold from the DS, i.e. the light blue dot on the green circle, it switches to the circumnavigation behavior, until it enters the Field of View of the DS, switching point represented by the light green dot, when it starts moving with a blended trajectory, mixing a circular and straight path, in order to perform the terminal docking phase, on the lilac dot, representing the DS itself.



(a) Case 1 - Plot 2D.

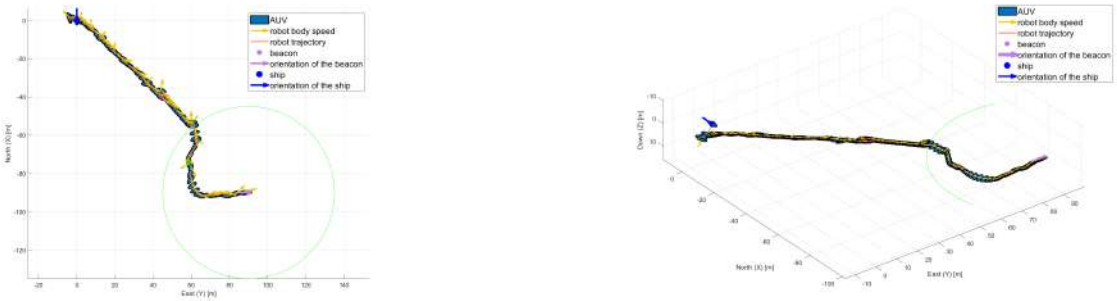
(b) Case 1 - Plot 3D.

Figure 5.1: Plots Case 1, 2D vs 3D.



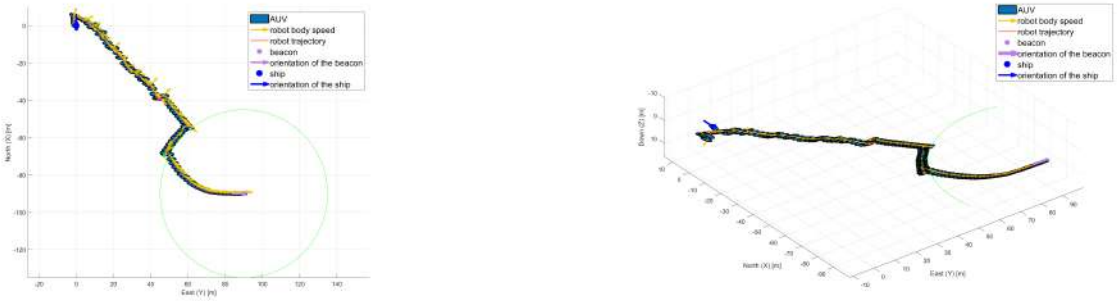
(a) Case 2 - Plot 2D. (b) Case 2 - Plot 3D.

Figure 5.2: Plots Case 2, 2D vs 3D.



(a) Case 3 - Plot 2D. (b) Case 3 - Plot 3D.

Figure 5.3: Plots Case 3, 2D vs 3D.



(a) Case 4 - Plot 2D. (b) Case 4 - Plot 3D.

Figure 5.4: Plots Case 4, 2D vs 3D.

5.2 Monte Carlo Method

The Monte Carlo method is a statistical technique that uses repeated random sampling to approximate numerical results. It is frequently used in circumstances where variability, stochasticity, or uncertainty are important factors.

The performance of the controllers and the state estimation filters were assessed in this work using the Monte Carlo method. In particular, the technique was applied to calculate the controlled variable's and the estimation error's time-varying mean and standard deviation. This was accomplished by running 100 times the simulation with various noise and uncertainty realizations. The ensemble of results offers a statistically significant description of the behavior of the system, even though the comparatively small number of runs was forced by computational limitations.

By measuring the variability brought on by stochastic disturbances and capturing how the error changes over time, this method allows for a thorough examination of the estimator and controller performance. The reliability and consistency of the suggested control and estimation schemes can be better understood by visualizing the mean trajectory and the confidence intervals that are obtained from the standard deviation.

5.2.1 Randomic Number Seed

In order to make the simulations even more stochastic, for every *Randomic Number* block representing Gaussian noise effect, it was chosen a seed, namely the number starting from which the random sequence is generated, obtained converting the current pc time into an integer value.

5.2.2 Statistics

The mean and the standard deviation of the output variables were calculated, along the number of simulations for every sample instant, for a deeper statistical understanding of the system behavior in an uncertain environment with Monte Carlo analysis. Since only a hundred samples were used, the standard deviation calculation was corrected based on Bessel's correction to deliver unbiased variance measurement against the population. This adjustment of the divisor, from N to $N-1$, corrects for the fact that with a small number of samples, the sample variance tends to underestimate the actual variance. These statistics provide a first order description of the magnitude and variability expected in the system error, providing a reliable link between input uncertainties into model sensitivity.
The sample standard deviation follows this structure

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{5.1}$$

where $\bar{x}$ is the sample mean, computed as

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{5.2}$$

being N the number of samples, in this case 100, and x_i the value of each sample.

5.3 Considered Data

In this section, the data relative to the whole simulations will be reported.

5.3.1 Model Noise

The model itself was considered as noisy, meaning that it's considered not accurately known in the wholeness of its data. In particular, the controller has been built with slightly different data with respect to the ones of the physical model, as reported in the Table 5.1. This worsens for sure the performance of the controller but it is not a critic problem for the algorithm, being these controllers robust enough.

Parameter	Physical Value	Controller Value	Unit
m	30	30	kg
I_z	5	4.14	kg·m^2
$X_{\dot{u}}$	-25	-25	kg
$Y_{\dot{v}}$	-2	-2.325	kg
$Z_{\dot{w}}$	-20	-19.4311	kg
$N_{\dot{r}}$	-10	-8.690	kg·m^2
X_u	-0.2	-0.2	kg/s
Y_v	-55	-55.117	kg/s
Z_w	35	33.6804	kg/s
N_r	-5	-4.14	kg·m/s
$X_{\|u\|u}$	-20	-19.5	kg/m
$Y_{\|v\|v}$	-150	-147.9	kg/m
$Z_{\|w\|w}$	100	89.4105	kg/m
$N_{\|r\|r}$	-7	-6.23	kg·m

Table 5.1: Comparison between physical values and noisy controller values.

5.3.2 Estimation Filters

In order to implement a simulation more similar to the real world, the covariances of the estimation filters were considered lower with respect to the real noise introduced in the simulation: this is because in real life it's not possible to know deterministically the exact noise, so it's better to overestimate it, rather than the opposite. The exact noise of each simulation and the motivation of the choice is detailed in the next subsection.

5.3.3 Case 1

In the first case, since the vehicle uses the same estimation filter throughout the whole task, from launching to docking, it was necessary to lower the noise covariances, in order to achieve an acceptable performance. In table 5.2 the values used for the planar filter, both the matrix ones and the real ones, will be shown. It's possible to notice that for x and y, being generally higher values with respect to the current and the orientation, as well as the rate r, it's allowed to use a higher noise, both on the measurement and on the state. These parameters were chosen empirically. For this case, the same filter is used throughout the whole simulation, so the noise values will be the same in each phase. For what concerns the vertical axis, the values are reported in table 5.3. Furthermore, the values of the filter used to retrieve the relative angle ψ_r between the DS and the AUV was implemented with the following noise parameters, reported in table 5.4.

5.3.4 Case 2

In what concerns the first approaching phase in the ship frame, the estimation filter works exactly as the one used for Case 1, with the exception that it works correctly also with a higher real value of the noises, since it's still the part far away from docking. Here, in table 5.5 are reported the values for the planar phase, while for the vertical one, it's exactly as the previous case 5.3.3. For what concerns the next phases, the values of the noises of the estimation filter for this case are the ones in table 5.6. In order to perform a correct docking, it's important to have a good estimate of the real covariance, so it's not possible to have it too much higher with respect to the model one.

Parameter	Simulation Value	Estimation Filter Value	Unit
ξ_x	0.003	0.001	m
ξ_y	0.003	0.001	m
ξ_ψ	0.0003	0.0001	rad
ξ_r	0.0003	0.0001	rad/s
$\xi_{v_{c_x}}$	0.0003	0.0001	m/s
$\xi_{v_{c_y}}$	0.0003	0.0001	m/s
η_x	0.03	0.01	m
η_y	0.03	0.01	m
η_ψ	0.003	0.001	rad

Table 5.2: Simulation and Estimation Filter Square Root of the Covariances - Case 1 - Planar.

Parameter	Simulation Value	Estimation Filter Value	Unit
ξ_z	0.005	0.001	m
ξ_{vcz}	0.005	0.001	m/s
η_z	0.005	0.001	m

Table 5.3: Simulation and Estimation Filter Square Root of the Covariances - Case 1 - Vertical.

Parameter	Simulation Value	Estimation Filter Value	Unit
ξ_{ψ_r}	0.0035 (0.2°)	0.0017 (0.1°)	rad
ξ_μ	0.02	0.01	rad/s
η_1	0.0087 (0.5°)	0.0017 (0.1°)	rad
η_2	0.0087 (0.5°)	0.0017 (0.1°)	rad

Table 5.4: Simulation and Estimation Filter Square Root of the Covariances - Relative Yaw ψ_r - Cases 1&3.

Parameter	Simulation Value	Estimation Filter Value	Unit
ξ_x	0.005	0.001	m
ξ_y	0.005	0.001	m
ξ_ψ	0.0005	0.0001	rad
ξ_r	0.0005	0.0001	rad/s
$\xi_{v_{c_x}}$	0.0005	0.0001	m/s
$\xi_{v_{c_y}}$	0.0005	0.0001	m/s
η_x	0.05	0.01	m
η_y	0.05	0.01	m
η_ψ	0.005	0.001	rad

Table 5.5: Simulation and Estimation Filter Square Root of the Covariances - Case 2 - Planar - Phase 1.

5.3.5 Case 3

In the third analyzed case, the first phase is just like the one for Case 2, as well as the next phases. In addition, there is also the the estimator for the relative orientation between the DS and the AUV, with the same values equal to the ones in 5.4.

5.3.6 Case 4

In the last scenario, the values of the covariances are exactly equal to the ones in Table 5.6.

5.3.7 Kinematic & Settings Parameters

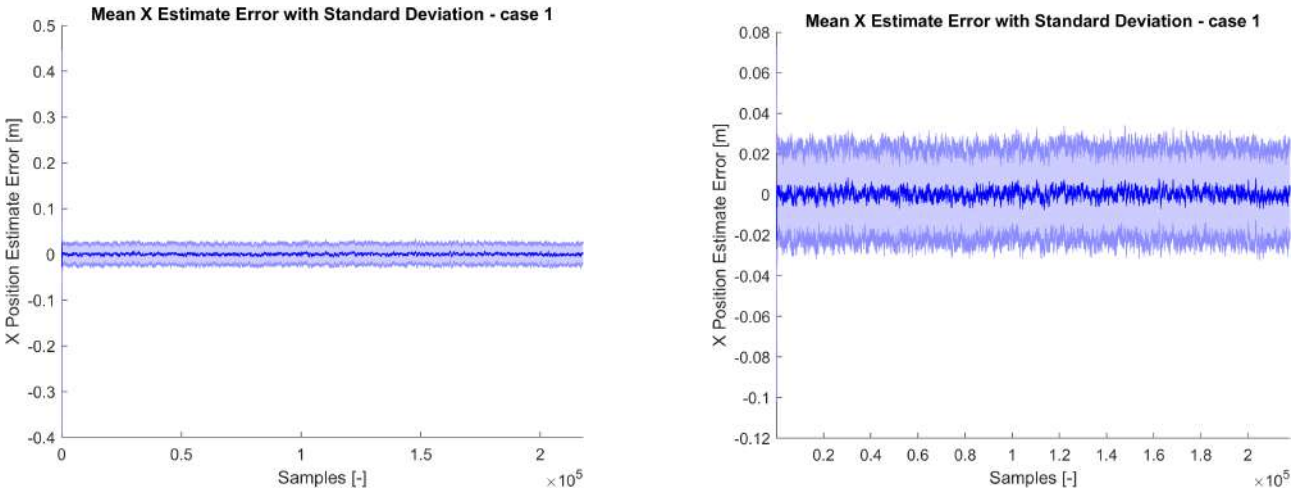
In order to achieve a proper rate of dockings in such a short distance, it was necessary to properly choose some kinematic quantities. For what concerns the terminal homing phase, it is crucial to use a proper value for Δ, in order to avoid the vehicle to perform abrupt maneuvers and avoid the AUV to lose the track of the virtual straight line that goes out of the DS. Since the radius in which the vehicle has to align is of 45 m, which is a relatively small distance, the Δ should be not too small to avoid a too high settling distance on the docking line. For this reason, many empirical values were tried, and the most suitable ones are reported in 5.7. Similarly, the maximum orthogonal threshold from which to start the blending of the velocities in phase 4, was selected empirically, because of the opening of the DS, since the vehicle usually switches to the last behavior at approximately 15m from the line in its normal direction. For this reason, the value considered giving the best ration of dockings was $D = 15$. In case of changing the radius from which the AUV starts the final phase, it would be necessary to change these sets of values; a good starting point would be a value close to the orthogonal distance of the vehicle from the line, and from there investigate empirically.
The values of the two thresholds at which switching from Phase 1 to 2 (60m) and from Phase 2 to 3 (45m) were chosen this way due to simulation reasons, in order to achieve a compromise between a realistic simulation and computational effort.
The Δt value was chosen to be 0.005s, again in order to compromise between computational effort and accuracy. From this number, it's possible to get to the effective time interval, just by multiplying it for the number of samples elapsed up to the examined moment.
The velocity, relative to the fluid, measured through the use of the Doppler Velocity Log, was considered to be noisy as well, in order to make a more realistic simulation. The values of the covariances were the ones in Table 5.8. These values are reasonable, since they are, in average, two orders of magnitude lower than the velocity itself.
The saturation values used for the *Anti-Windup* structure, was chosen once for all the analysis, based on empirical tests. Every degree of freedom has its own custom saturation value, as reported in 5.9.

5.4 Estimation Filter Errors

Here follows a subgroup of the whole set of variables examined to evaluate the performance of the estimation filters. In general, the variables examined were, other than the displayed x (5.6), v_{c_x} (5.7), ψ (5.8) and relative orientation ψ_r (5.9), also y, z, v_{c_y}, v_{c_z}, r. Due to lack of space, and being less relevant, the latter ones were not here plotted.

5.4.1 x Estimation Filter Performance

In figure 5.6, it's possible to notice the mean error and the confidence interval of the estimation of x, throughout the four implemented algorithms. The mean value is always around 0, meaning that the x is correctly estimated, with a confidence interval equal to one standard deviation, varying in the different cases. In 5.6a, being the estimate filter always the same throughout the four phases, the policy does not change, so it's natural that the error trend does not change over the whole simulation. In order to allow the docking, it was necessary to avoid having too high noise both on the state and on the measurements, as mentioned in 5.3. In the second, third and fourth cases (5.6b, 5.6c, 5.6d), the algorithm makes a distinction between a first approaching part in the ship frame and the second one in the docking station one, implementing two different estimation filters. In the first phase it was not required a too low noise since the relevant task is just to approach in the direction of the DS, while in the second one, requiring a precise performance of the AUV, it was necessary to lower the noise: this justifies the lower noise in the second half of the simulation for these three cases.
Due to a large initial uncertainty on the value, having no clue on the real value of the quantities, the standard deviation of the error is very high, in all the quantities; because of this, the graph looks like the one in Figure 5.5a. In this situation it's difficult to appreciate the details of the statistics of the error after the initial high covariance. For this reason, the errors statistics were cropped in the first time samples for every shown quantity, to show better the details, as it's possible to notice in 5.5b. The comparison between the two views is shown in 5.5.



(a) Case 1.

(b) Case 2.

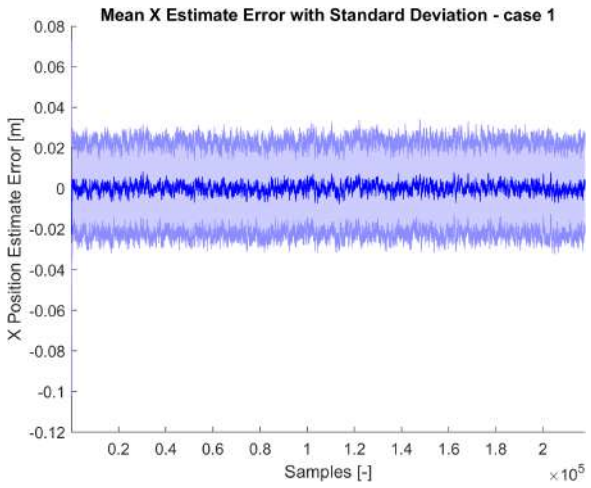
Figure 5.5: Real and cropped error dynamics - x error.

5.4.2 v_{c_x} Estimation Filter Performance

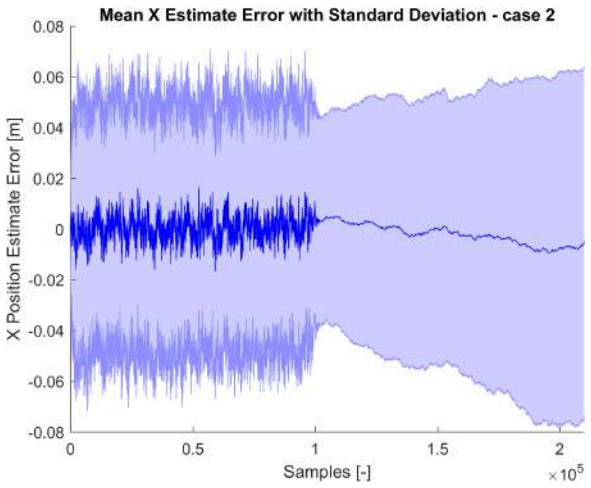
In the set of figures 5.7 it's possible to make again a distinction between the first case and the last three ones: in the first case, having the same estimation filter for all the the behaviors, the mean error trend is homogeneous throughout the whole simulation. For what concerns the other cases, in correspondence of the switch between the first filter and the other ones, there's a switch in the magnitude of the mean error and its standard deviation. This happens because the noise both in the modeling and the simulation is much smaller, since in this part, having a difficult measurement such as the range for the position, it is very important to estimate correctly the current, so a covariance of one magnitude order smaller was chosen, bringing to a very accurate estimate.

Parameter	Simulation Value	Estimation Filter Value	Unit
ξ_x	0.0002	0.0001	m
ξ_y	0.002	0.001	m
ξ_z	0.002	0.001	m
$\xi_{v_{cx}}$	0.00002	0.00001	rad
$\xi_{v_{cy}}$	0.00002	0.00001	rad
$\xi_{v_{cz}}$	0.00002	0.00001	rad
ξ_ψ	0.002	0.001	m/s
ξ_r	0.02	0.01	m/s
η_1	0.002	0.001	m
η_2	0.0002	0.0001	rad
η_3	0.0002	0.0001	rad
η_4	0.0002	0.0001	rad
η_5	0.002	0.001	m/s
η_6	0.002	0.001	m/s

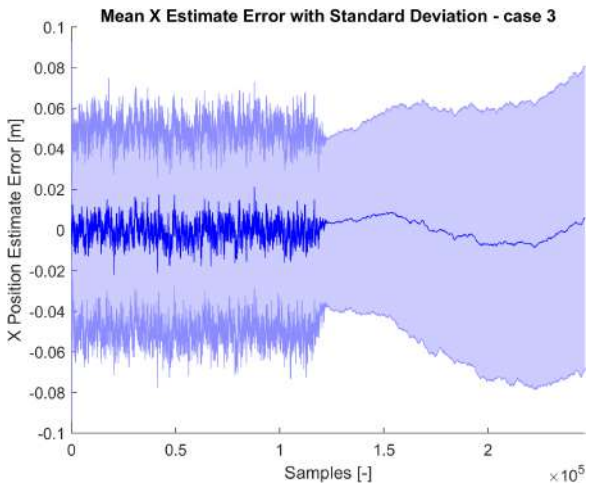
Table 5.6: Simulation and Estimation Filter Covariances - Case 2 - Phase 2&3&4.



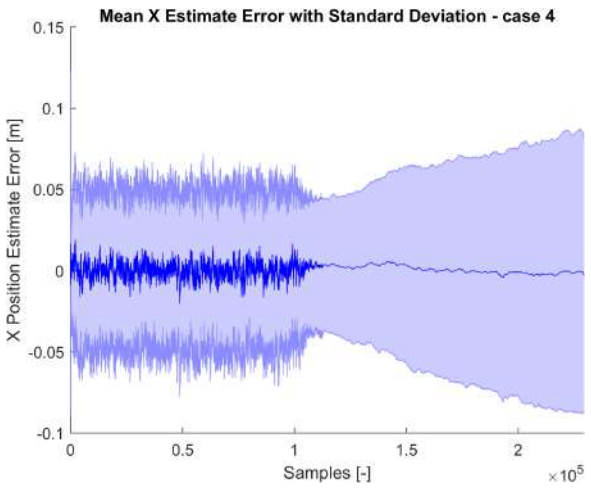
(a) Case 1.



(b) Case 2.

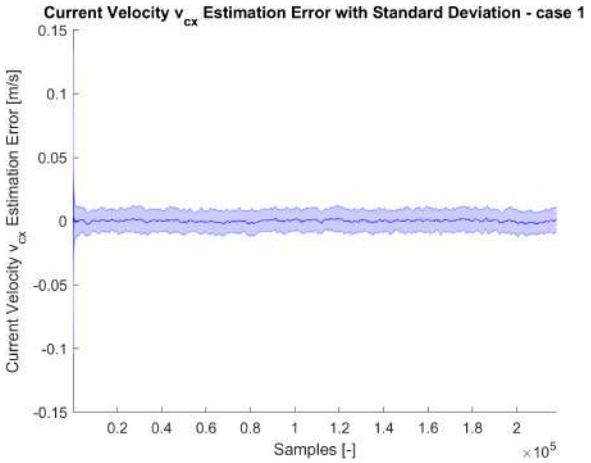


(c) Case 3.

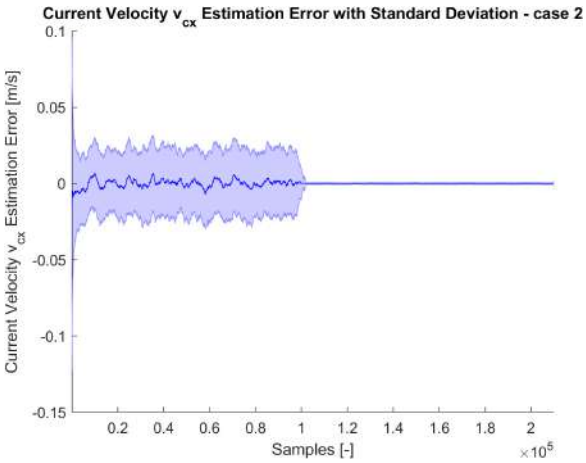


(d) Case 4.

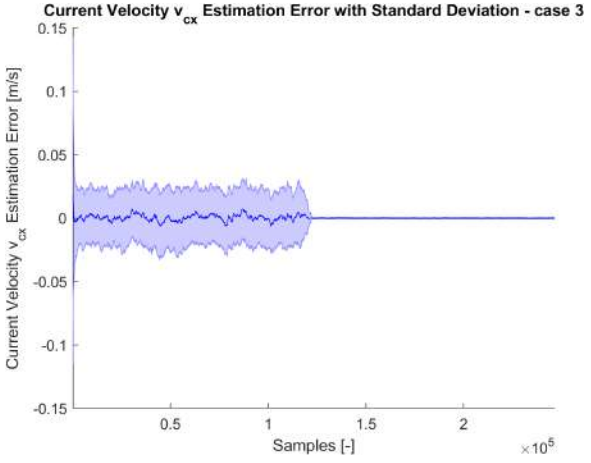
Figure 5.6: Estimate error of the x coordinate of the position of the vehicle, compared to the real value.



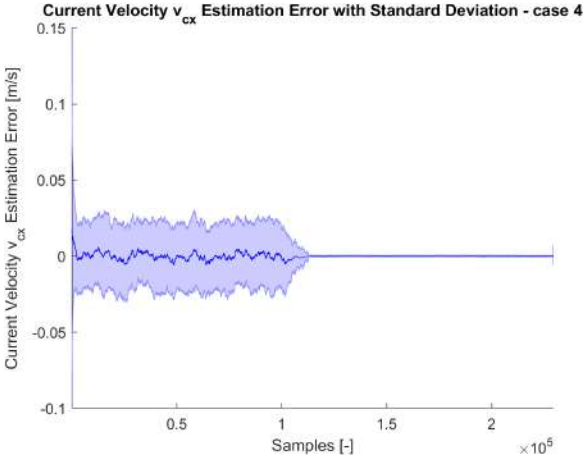
(a) Case 1.



(b) Case 2.



(c) Case 3.



(d) Case 4.

Figure 5.7: Estimate error of the v_{c_x} coordinate of the velocity of the ocean current, compared to the real value.

	Case 1	**Case 2**	**Case 3**	**Case 4**
Δ	30	10	20	13

Table 5.7: Δ values used for different scenarios.

$\eta_{v_{m_x}}$	$\eta_{v_{m_y}}$	$\eta_{v_{m_z}}$	**Unit**
0.001	0.001	0.001	m/s

Table 5.8: Noise covariance values used for x, y and z.

τ_x	τ_y	τ_z	τ_r
10 N	30 N	30 N	0.1 N$\cdot m$

Table 5.9: Saturation values for Anti-Windup.

5.4.3 ψ Estimation Filter Performance

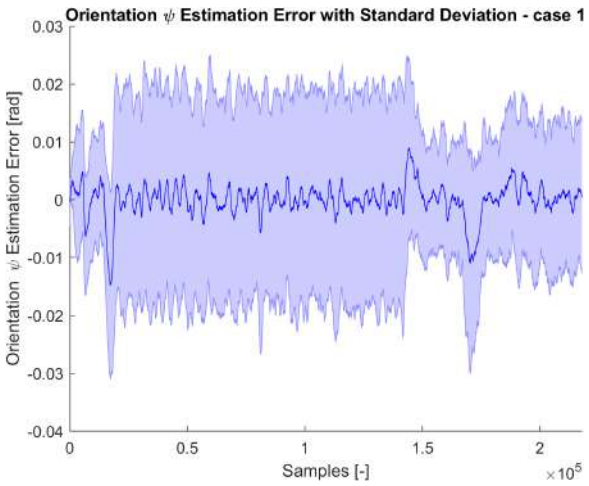
In Figures 5.8, it's shown how the trend of ψ estimate error goes over time. In Case 1, 5.8a, it is similar throughout the whole simulation again because one single estimate filter was used. In the other cases, the trend switches between phase 1 and phase 2, having a much higher noise in the second part, because the covariance of the modeled matrices of the estimation filters are one order of magnitude higher than the first filter, as well as the simulation noises. In particular, in 5.8b there are two visible spikes , which may be attributed to the change of policy in the next phases, between phase 2 and phase 3.

5.4.4 ψ_r Estimation Filter Performance

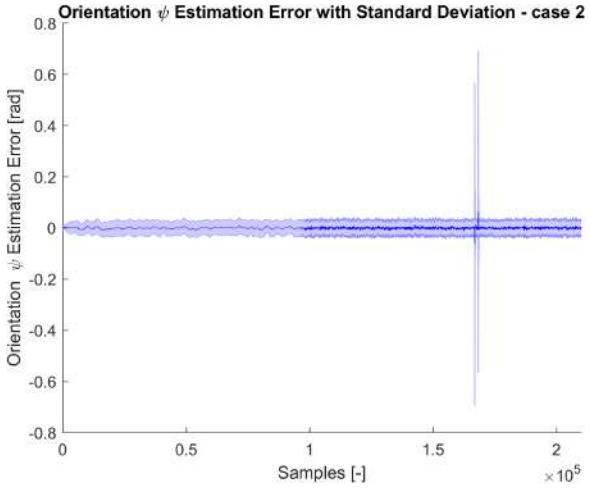
The Figures in 5.9 show the trend of the estimate error for the relative angle between the Docking Station and the AUV. t There's a general trend around zero, meaning that in average the angle is correctly estimated, but it is polluted by spikes. This behavior is accentuated in Figure 5.9b. This phenomenon might be due to the higher noise in the ψ_r estimation in this setting.

5.5 Control Errors

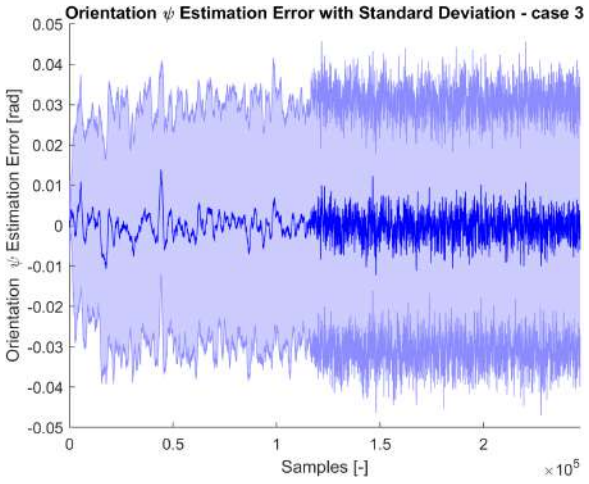
In this section, a subgroup of the whole set of variables examined to evaluate the performance of the controllers is reported. In general, the variables recorded were the errors on, other than the displayed u (5.10), v (5.11) and ψ (5.12), also z. The quantities are named *Tracking Errors* since they're the error between the reference quantity and the actual one; for this reason this is the error of the tracking, being the goal of the controller to drive the variable to the desired value. Due to lack of space, and being less relevant, the depth was not here plotted. While evaluating the performance of the controllers, it's important to keep in mind that the latter have been tuned on a noisy version of the parameters of the physical model, see Table 5.1, and considering the simplified model for the plant, on every direction, making the definition of the system, from the controller point of view, less accurate, not matching perfectly with the real physical model. Despite this, the performance is more than acceptable in all the examined cases.



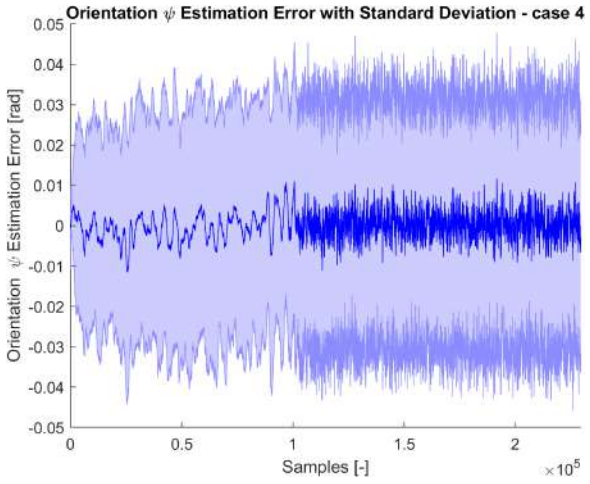
(a) Case 1.



(b) Case 2.

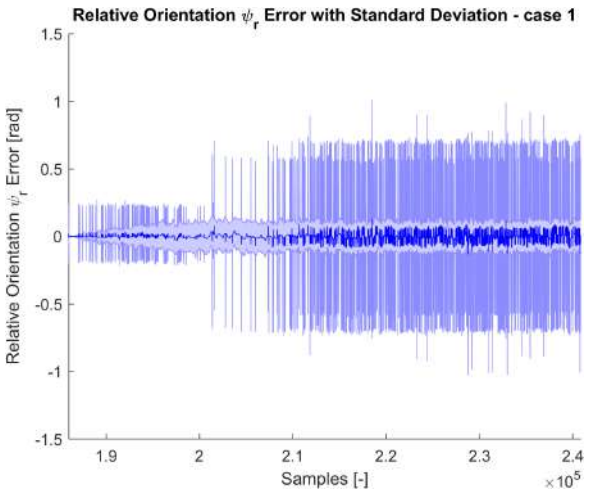


(c) Case 3.

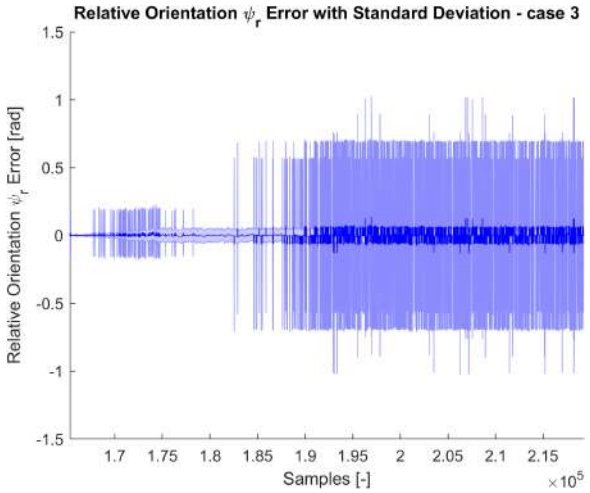


(d) Case 4.

Figure 5.8: Estimate error of the orientation of the vehicle ψ, compared to the real value.



(a) Case 1.

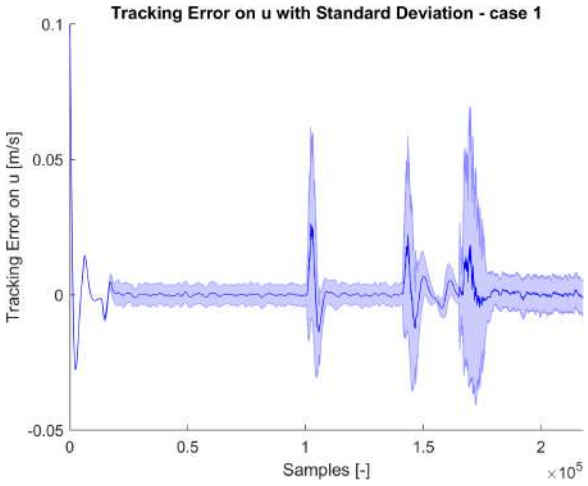


(b) Case 3.

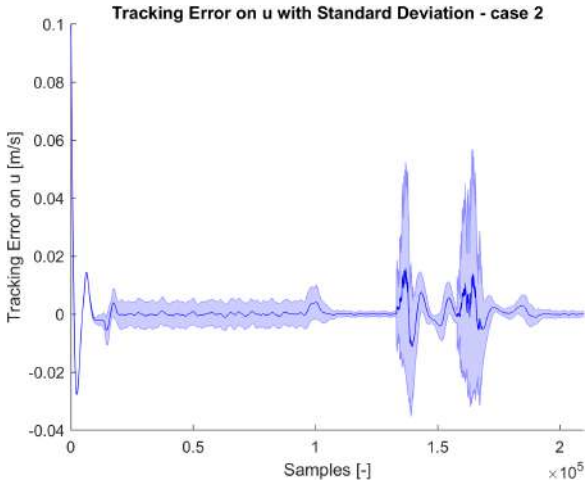
Figure 5.9: Estimation error of the relative orientation ψ_r of the docking station with respect to the vehicle.

5.5.1 u Controller Performance

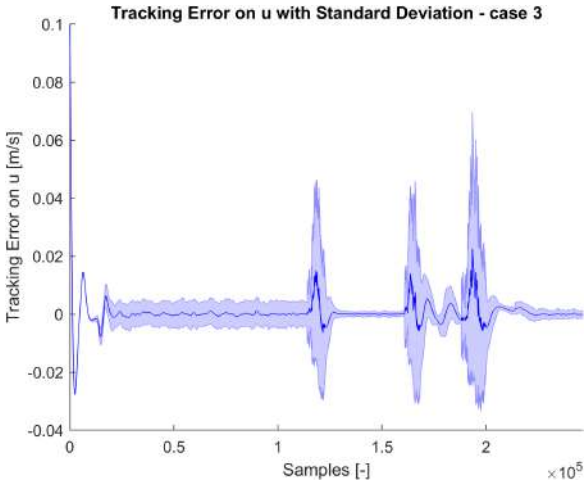
The tracking error on the surge speed follows a similar trend for all the first three cases, as it's possible to notice from Figures 5.10: the error starts with an average value of 0.1 since the initial reference value is equal to 0.1 and the initial longitudinal speed is 0. In short time, the error goes to zero, until the AUV keeps the first behavior. Transitioning to the second one, the error increases due to the change of reference frame, which takes a bit of time to stabilize the new procedure. Similarly, when transitioning to the third case, due to the change in the trajectory to follow, again there's a temporary increase in the error, which is then briefly shut to zero. Finally, entering the last phase, again there's a spike in the error, easily fixed. The behavior, as previously mentioned, is similar throughout every behavior, with the exception that in Case 4, being the speed not fixed but variable since the idea is to move with the required yaw angle according to the known docking station orientation, the initial magnitude of the error is a bit higher, as well as throughout the whole simulation, but the spikes in the transition phases are less marked.



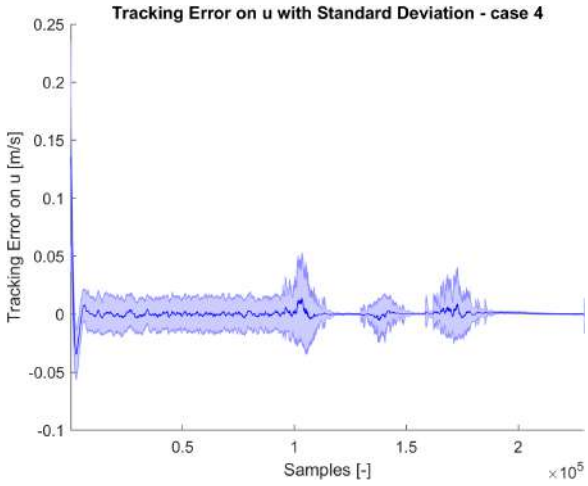
(a) Case 1.



(b) Case 2.



(c) Case 3.

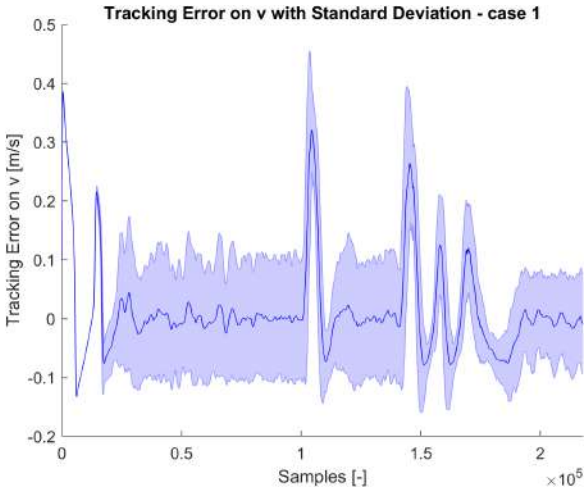


(d) Case 4.

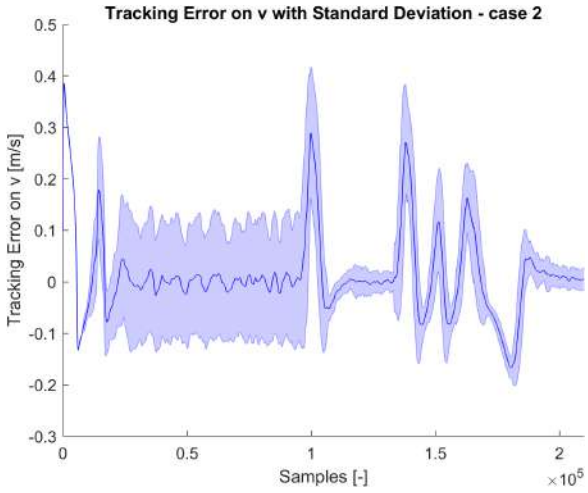
Figure 5.10: Control error of the surge velocity u, compared to the reference one u_d.

5.5.2 v Controller Performance

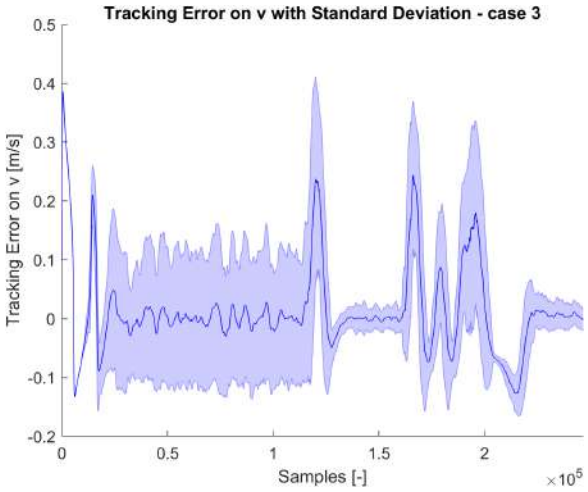
The dynamic on the error of the sway velocity is pretty similar to the along the longitudinal direction, see Figures 5.11. Again, overall the error is mostly equal to zero in average, with some spikes during the transition phases. The spikes are less accentuated in the last algorithm, namely the one with variable velocities. This is probably due to the switch between the policies, which is done in a more smooth way in the last case but also due to the saturation of the actuators. Also, the achievement of a desired yaw in order to follow the desired trajectory, might be in conflict with the control on the translational velocity.



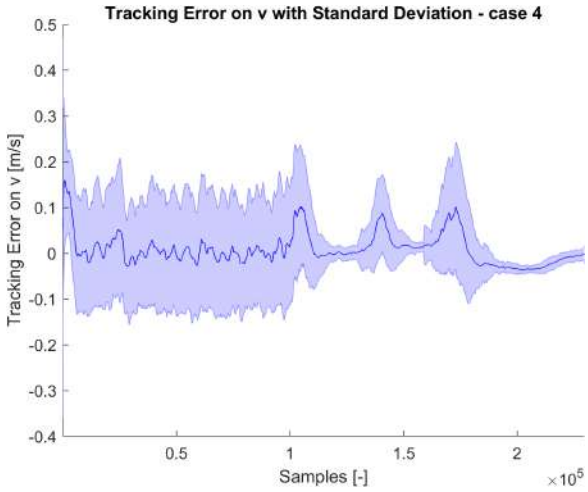
(a) Case 1.



(b) Case 2.



(c) Case 3.



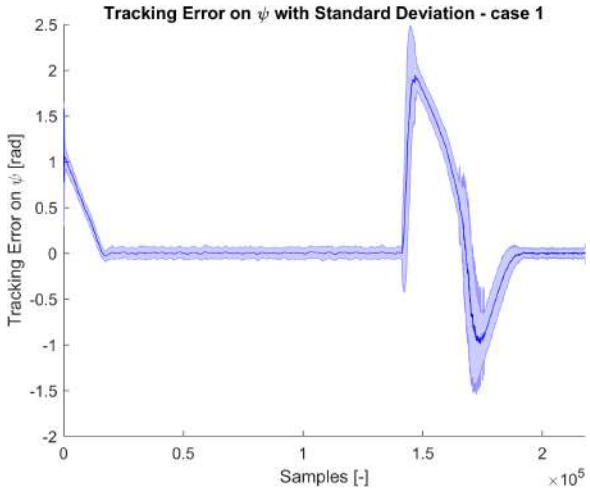
(d) Case 4.

Figure 5.11: Control error of the sway velocity v, compared to the reference one v_d.

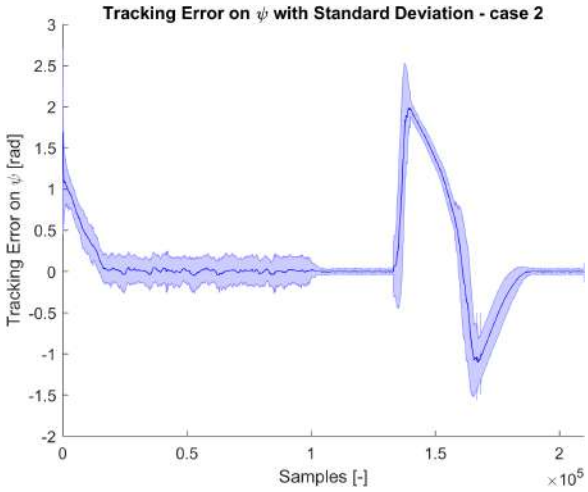
5.5.3 ψ Controller Performance

This error is computed as the difference between the reference yaw angle and the estimated one. In Figures 5.12 two distinct behaviors are seen, in the two opposite cases of having either velocity or orientation fixed on the AUV. In the first scenario, Figures 5.12a, 5.12b and 5.12c, there's a quite important spike in the final part, that is when the AUV switches to the correct orientation to perform the docking. In the case where the AUV has freedom to choose arbitrarily the orientation, that is the last one, in Figure 5.12d, the trend is different: the error at first is much higher, in this case equal to $\frac{\pi}{2}$ due to the initial configuration of the AUV and to the

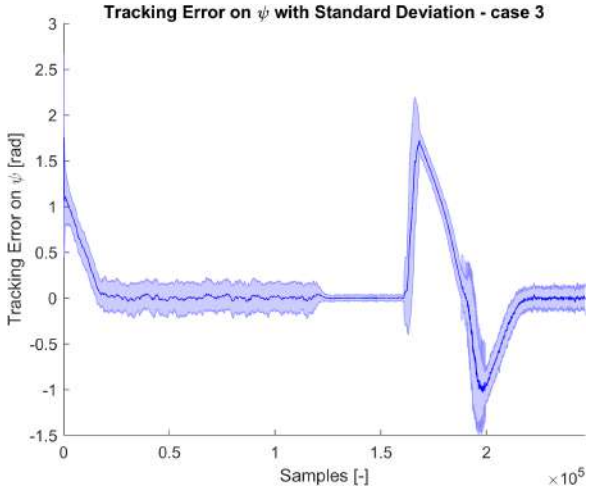
Docking Station one, then it is driven to zero, and doesn't grow anymore, avoiding the spikes of the first three cases, due to the change in the vehicle's policy.



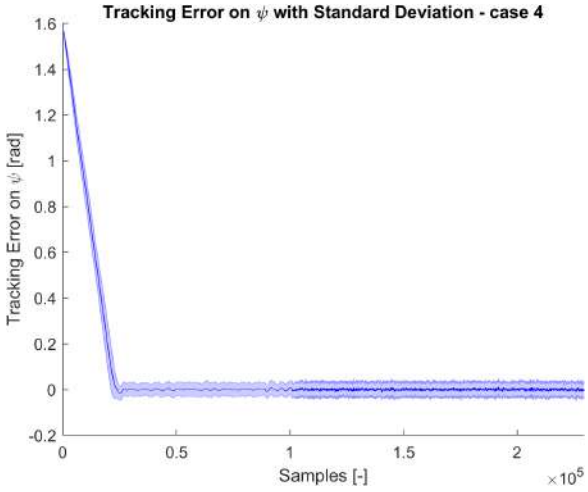
(a) Case 1.



(b) Case 2.



(c) Case 3.



(d) Case 4.

Figure 5.12: Control error of the orientation ψ, compared to the reference one ψ_d.

5.6 Docking Frequency

As a last result, it was recorded, out of 100 attempts for each examined case, the total number of successful dockings. A successful docking means that the vehicle is able to approach to the docking station from the correct direction and with the correct orientation. The operation is considered done once the vehicle gets its estimated center of mass to a minimum range of 30 *cm* from the center of mass of the docking station. At this point the simulation is automatically stopped. As it's possible to see in Table 5.10, the docking rate varies considerably based on the knowledge of the docking station orientation, showing that the algorithm in this thesis developed is a good starting idea but still has to be refined. These statistics are relative to the simulation run with the data presented in Section 5.3. Considering that the estimation systems had to deal with considerably higher noises with respect to the ones on which the filters were tuned, see Tables 5.2, 5.5, 5.6 and 5.4, the estimation and the controllers can be considered robust enough. Clearly, when the simulation noise gets closer and closer to the

Case	Number of Dockings
Case 1	53
Case 2	79
Case 3	57
Case 4	96

Table 5.10: Number of successful dockings for each case out of 100 simulations.

one of the covariance matrices, the results improve notably. The most successful algorithm is the one using variable velocities, moving with the proper ψ according to the docking station orientation, confirming that the algorithm developed based on the differential signal from the two transponders needs further improvements.

Chapter 6

Future Work

6.1 Integration of GNSS Data for Global Localization

The first possible future work to implement in this project is the integration of GNSS data to allow AUV localization in a global reference frame instead of just the ship frame. The current approach, which relies on relative localization with respect to the ship, may become inaccurate if the ship moves while operations are in progress. By combining the ship's GNSS measurements with the data from its onboard sensors using methods like the already implemented Kalman filter, accordingly modified based on the equations coming from the GNSS, the AUV's position could be tracked more accurately in a fixed global frame. This improvement would support more dependable long-term deployments and increase the robustness of navigation and estimation, especially in dynamic scenarios.

6.2 Sliding Mode Control

The ability of Sliding Mode Control (SMC) to effectively manage system uncertainties and external disturbances has made it a popular control strategy. The fundamental principle of SMC is to apply a control law that rapidly switches between various inputs in order to force the system's behavior to "slide" along a predetermined surface, known as the sliding surface. Even in the event of unforeseen disruptions or imperfect model accuracy, the system remains at this surface and behaves according to the intended dynamics.

Sliding mode control is especially attractive because of its robustness and simplicity. SMC can still function effectively when some parameters are unknown or changing, in contrast to many other control strategies that necessitate exact knowledge of the system model, such as Model Predictive Control or Linear Quadratic Regulator.

Because of this, it is particularly helpful in real-world applications where flawless modeling is rarely possible.

However, the so-called chattering phenomenon, which is brought on by the high-frequency switching of the control signal, is one of the primary problems with SMC. This may result in deterioration or even instability in real-world systems. Over time, several smoother SMC versions have been put forth to lessen this.

Sliding mode control is a logical option for situations where dependability is more important than style since it provides a strong and useful tool for managing nonlinear systems under uncertainty, and could be interesting to implement it as future development of this work.

6.3 Cramér-Rao Lower Bound & Fisher Information Matrix

Utilizing analytical tools like the Fisher Information Matrix (FIM) and the Cramér-Rao Lower Bound (CRLB) to enhance and supplement statistical analysis is another exciting avenue for future research. Methods based on CRLB provide a theoretical lower bound on the variance of any unbiased estimator, even though the Monte Carlo approach was used in this work to estimate the mean and standard deviation of both the estimation error and the controlled variable - based on a small number of simulation runs because of computational limitations. This would eliminate the need for repeated simulations to assess the basic bounds of estimation accuracy. In addition to lowering computational effort, CRLB or FIM-based analysis could offer a standard by which to measure how closely the actual estimators match optimal performance, providing a more thorough and insightful statistical description of the system.

6.4 Extend the Blended Trajectory Use

Another useful implementation would be to use the blended trajectory algorithm also between the second and the third phases of the algorithm, in order to avoid the abrupt change in terms of direction, when passing from the straight line to the circumference, which generates delays in the control of the vehicle. The idea, would be completely analog to the one implemented in 4.1.4.

Conclusion

The work in this thesis has tackled the issue of autonomous underwater docking for the Medusa-class AUV using a full guidance, control, and estimation framework developed through design, modeling and testing. The work was developed in a phased approach, proceeding from the operational scenarios definition to cover the solution proposal implementation and verification in specific simulation environment.

Beginning by formulating the docking problem, this is modeled as a control problem where two main operational strategies characterized by distinct constraints imposed on vehicle orientation and velocity control are identified. Subsets of this (long-range approach, homing, circumnavigation, terminal docking) were assumed as quasi-separate phases in the overall docking sequence, which might be managed with guidance and control laws tailored to their respective demands. This modular structure guaranteed the framework to be invariant with respect to different initial states, environmental effects and sensor setups.

The vehicle model was built on a four degrees-of-freedom Newton–Euler representation, which offers sufficient fidelity with minimum computational demand. Utilizing a simplified version of this model, affected by physical parameters noise, parallel PI and PID controllers with anti-windup handling were synthesized and parameterized to stabilize the system while tracking a reference input in presence of actuator saturation, to prevent the integrator's saturation which would provide a longer response time of the correction of the error. State estimation was also treated independently using Kalman and Extended Kalman filters, with different configurations tailored towards propagating the AUV information and dock orientation over time in the presence of highly noisy readings from the sensors.

One of the key contributions in this work is building a custom simulation environment for thorough testing of the presented algorithms, under realistic operational conditions. Other characteristics of this environment were considered in terms of conditions under which controlled process and measurement noise could be added, providing an evaluation of the degree to which the guidance, control, and estimation layers maintained robustness under degraded conditions and studying the effect of parameter mismatch between the controller and plant model, as well as the underestimated noise on Kalman and Extended Kalman filters. It was also employed to structure and control simulation data for reproducible and systematic comparison across different operational scenarios. This detailed numerical simulation analysis is the foundation of the quantitative evaluation in this approach.

The other key contribution was the modeling and implementation of a new method to retrieve the orientation of the docking station in the unlucky but probable case where the exact orientation is not exactly known, while keeping the number of sensors low, by resorting to only one iUSBL rather than two distinct ones, one on the AUV and the other on the DS, communicating each other the relative orientation.

This thesis describes a promising solution for docking onboard the Medusa-class platform as well as establishing an adaptable methodology to be used one for more broadly applicable vehicles with similar dynamics. The modular design, emphasis on resilience, and the realistic testing tools make it possible to design Status for future growth. Specifically, it provides a firm

foundation for the research directions presented in 6, including adding more sensor modalities, improving control strategies (Sliding Mode controllers), and expanding the idea of blended trajectories to more advanced cases.

In summary, this thesis underscores the structured and scientific nature of methodology that combines stringent modeling, focused control and estimation schemes, and a mature simulation environment to take on the natural challenges apparent in autonomous underwater docking. This approach not only directly extends the work so far done on the Medusa-class AUV, but also serves as a foundation in the development of reliable, long-endurance and fully-autonomous underwater operation across many platforms, opening for future improvements.

Appendix A

Rotational Matrix

The rotation matrix ${}^{I}_{B}R(\boldsymbol{\eta}_2)$ is an orthogonal matrix, which means that

$$ {}^{I}_{B}R(\boldsymbol{\eta}_2)^{T}\,{}^{I}_{B}R(\boldsymbol{\eta}_2) = I_{3\times 3} \text{ and } \quad \det({}^{I}_{B}R(\boldsymbol{\eta}_2)) = 1 $$

and represents how the frame $\{B\}$ is rotated with respect to the inertial frame $\{I\}$. This matrix is constructed through a sequence of rotations about the body axes in the order: a rotation of yaw (ψ) about the z-axis, then rotating of pitch (θ) about the y-axis, and last one, rotate of roll (ϕ) about the x-axis:

$$ {}^{I}_{B}R(\boldsymbol{\eta}_2) = R_z(\psi)R_y(\theta)R_x(\phi) $$

Each individual rotation is defined as:

$$ R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} $$

The resulting complete rotation matrix is:

$$ {}^{I}_{B}R(\boldsymbol{\eta}_2) = \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\phi)\sin(\theta)\sin(\psi) & -\cos(\psi)\sin(\phi) + \sin(\theta)\sin(\psi)\cos(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} $$

Appendix B

Euler Attitude Transformation Matrix

The Euler attitude transformation matrix, denoted by $Q(\boldsymbol{\eta}_2)$, relates the angular velocities expressed in the body frame $\{B\}$, namely $\boldsymbol{\nu}_2 = [p, q, r]^T$, to the time derivatives of the Euler angles, $\dot{\boldsymbol{\eta}}_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$, and it is defined as

$$Q(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix},$$

It should be noted that the matrix Q becomes singular at $\theta = \pm\frac{\pi}{2}$. However, this singularity does not pose a problem, as such angle values are outside the operational range considered for the vehicle in this study, since it only has one rotational degree of freedom, namely the yaw angle ψ.

Appendix C

Controllers

C.1 PI Controller

The Proportional-Integrative (PI) controller corrects the input error quantity feeding the same error and the integrative of the latter, premultiplied by the proper gains, to the system as control input. The proportional term reacts to the current error, while the integral term accumulates past errors. The PI controller improves both transient and steady-state response by reducing the steady-state error while maintaining a simple structure.

The control law of a PI controller in continuous time is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

where $u(t)$ is the control signal, $e(t) = r(t) - y(t)$ the control error, in which $r(t)$ is the reference signal, computed by the outer loop, and $y(t)$ is the measured output quantity, K_p is the proportional gain, K_i is the integral gain.
The proportional term provides an immediate response to the error, while the integral term removes the steady-state error by integrating the error over time. Unlike the full PID controller, the PI controller does not include a derivative term, making it more robust to high-frequency noise.

C.2 PID Controller

The Proportional-Integrative-Derivative (PID) controller corrects the input error quantity through the computation of a control input made by the sum of the error, its integrative and its derivative, premultiplied by proper gains. The proportional term reacts to the current error, the integral one eliminates accumulated past errors, while the derivative one anticipates future errors' behavior. This combination allows the controller to achieve a good balance between fast response, minimal overshoot, and zero steady-state error, providing a more sophisticated solution with respect to the simple PI.

The continuous-time control law of a PID controller is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t)$$

where $u(t)$ is the control signal, $e(t) = r(t) - y(t)$ is the control error, in which $r(t)$ is the reference signal, computed by the outer loop, $y(t)$ is the measured output of the quantity of interest, K_p is the proportional gain, K_i is the integral one and K_d is the derivative one.

The proportional term provides immediate response, the integral term eliminates steady-state error, and the derivative term improves system stability and transient response by predicting future error trends. Proper tuning of the three gains is crucial for optimal performance and stability of the closed-loop system.

Appendix D

Linearizaton of a Non-Linear System

Consider a nonlinear function that depends on both the state vector $x \in \mathbb{R}^n$ and the input vector $u \in \mathbb{R}^m$, such that:

$$y = f(x, u) \tag{D.1}$$

To simplify analysis or design (e.g., for control purposes), it is often desirable to linearize this function around an operating point (x_0, u_0). This allows the use of linear system theory in a local neighborhood of the operating point.

First-Order Taylor Expansion

The function $f(x, u)$ can be approximated by its first-order Taylor expansion around (x_0, u_0):

$$f(x, u) \approx f(x_0, u_0) + \left.\frac{\partial f}{\partial x}\right|_{x_0,u_0} (x - x_0) + \left.\frac{\partial f}{\partial u}\right|_{x_0,u_0} (u - u_0) \tag{D.2}$$

Linearized Input-Output Relationship

Defining deviation (perturbation) variables:

$$\delta x = x - x_0, \quad \delta u = u - u_0, \quad \delta y = y - f(x_0, u_0)$$

The linearized equation becomes:

$$\delta y \approx \left.\frac{\partial f}{\partial x}\right|_{x_0,u_0} \delta x + \left.\frac{\partial f}{\partial u}\right|_{x_0,u_0} \delta u \tag{D.3}$$

This expression provides a linear approximation of the nonlinear function $f(x, u)$ around the point (x_0, u_0), valid for small perturbations δx and δu.

Remarks

- The matrices $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ are known as the Jacobians of f with respect to x and u, respectively.
- This linearization is valid only locally, near the chosen operating point.

Appendix E

Kalman Filter Overview

The Kalman Filter is an efficient recursive algorithm used to estimate the internal state of a dynamic system from a series of noisy measurements. It is widely used in control systems, robotics and navigation due to its ability to provide optimal estimates in a mean square error sense under uncertainty, assuming Gaussian noise and linear system dynamics.

Given a system described by a linear state-space model, in the following form:

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \tag{E.1}$$
$$z_k = C_k x_k + v_k \tag{E.2}$$

where x_k is the state vector at time k, u_k is the control input, z_k is the measurement vector, A_k, B_k, and C_k are system matrices, w_k and v_k are process and measurement noise, assumed to be zero-mean Gaussian with covariance $\tilde{Q}_k$ and R_k, respectively, where $\tilde{Q}_k$=GQ_kG^T, being G the matrix of the coefficient due to discretization, related to the state noise.

E.1 Kalman Filter Steps

The Kalman Filter algorithm operates in two main phases: prediction and update.

Prediction Step

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1} \tag{E.3}$$
$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^{\top} + GQ_{k-1}G^T \tag{E.4}$$

Update Step

$$K_k = P_{k|k-1}C_k^{\top}(C_kP_{k|k-1}C_k^{\top} + R_k)^{-1} \tag{E.5}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - C_k\hat{x}_{k|k-1}) \tag{E.6}$$
$$P_{k|k} = (I - K_kC_k)P_{k|k-1} \tag{E.7}$$

where $\hat{x}_{k|k-1}$ is the predicted state estimate, $\hat{x}_{k|k}$ is the updated (a posteriori) state estimate, P is the error covariance matrix and K_k is the Kalman gain.

E.2 Extended Kalman Filter (EKF)

The standard Kalman Filter assumes linear dynamics. In many real-world systems, the state transition and observation models are non-linear:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{E.8}$$
$$z_k = h(x_k) + v_k \tag{E.9}$$

In such cases, the Extended Kalman Filter (EKF) linearizes the non-linear functions $f(\cdot)$ and $h(\cdot)$ around the current estimate using their Jacobians.

Prediction

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \tag{E.10}$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^\top + G Q_{k-1} G^T \tag{E.11}$$

Update

$$K_k = P_{k|k-1} H_k^\top (H_k P_{k|k-1} H_k^\top + R_k)^{-1} \tag{E.12}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1})) \tag{E.13}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{E.14}$$

Where:

- $F_k = \frac{\partial f}{\partial x}\big|_{x=\hat{x}_{k-1|k-1}}$ is the Jacobian of f
- $H_k = \frac{\partial h}{\partial x}\big|_{x=\hat{x}_{k|k-1}}$ is the Jacobian of h

The EKF provides a practical solution for non-linear systems, although it may suffer from linearization errors. Alternative methods like the Unscented Kalman Filter (UKF) address some of these limitations.

Bibliography

[1] A. Yazdani, K. Sammut, O. Yakimenko, and A. Lammas, "A survey of underwater docking guidance systems," *Robotics and Autonomous Systems*, vol. 124, p. 103382, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889019300181

[2] J. Kim, *Dual control approach for automatic docking using monocular vision*, 2007, vol. 68, no. 09.

[3] J. Esteba, P. Cieślak, N. Palomeras, and P. Ridao, "Sparus docking station: A current aware docking station system for a non-holonomic auv," *Journal of Field Robotics*, vol. 41, pp. 1765–1779, 2024.

[4] B. Fletcher, S. Martin, G. Flores, A. Jones, A. Nguyen, M. H. Brown, and D. L. Moore, "From the lab to the ocean: Characterizing the critical docking parameters for a free floating dock with a remus 600," in *OCEANS 2017 - Anchorage*, 2017, pp. 1–7.

[5] M. Breivik and T. I. Fossen, "Guidance laws for planar motion control," *Centre for Ships and Ocean Structures, Department of Engineering Cybernetics, Norwegian University of Science and Technology*, 2005.

[6] M. Feezor, F. Yates Sorrell, P. Blankinship, and J. Bellingham, "Autonomous underwater vehicle homing/docking via electromagnetic guidance," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 515–521, 2001.

[7] H. Singh, J. G. Bellingham, F. Hover, S. Lemer, B. A. Moran, K. V. der Heydt, and D. Yoerger, "Docking for an autonomous ocean sampling network," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 498–514, 2001.

[8] R. S. McEwen, B. W. Hobson, L. McBride, and J. G. Bellingham, "Docking control system for a 54-cm-diameter (21-in) auv," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 550–562, 2008.

[9] J. Y. Park, B. H. Jun, P. M. Lee, Y. K. Lim, and J. H. Oh, "Docking problem and guidance laws considering drift for an underactuated auv," in *Proceedings of the IEEE OCEANS 2011 Conference*, 2011, pp. 1–7.

[10] F. Raspante, "Underwater mobile docking of autonomous underwater vehicles," in *OCEANS 2012 Conference*, 2012, pp. 1–15.

[11] J. Piskura, M. Purcell, R. Stokey, T. Austin, D. Tebo, R. Christensen, and F. Jaffre, "Development of a robust line capture, line recovery (lclr) technology for autonomous docking of auvs," in *OCEANS 2016 Conference*, 2016, pp. 1–5.

[12] N. Palomeras, G. Vallicrosa, A. Mallios, J. Bosch, E. Vidal, N. Hurtos, M. Carreras, and P. Ridao, "Auv homing and docking for remote operations," *Ocean Engineering*, vol. 154, pp. 106–120, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801818301367

[13] B. R. Page, R. Lambert, J. Chavez-Galaviz, and N. Mahmoudian, "Underwater docking approach and homing to enable persistent operation," *Frontiers in Robotics and AI*, vol. 8, 2021. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2021.621755

[14] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011.

[15] N. T. Hung, F. Rego, J. Quintas, J. Cruz, M. Jacinto, D. Souto, A. Potes, L. Sebastião, and A. Pascoal, "A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments," *Journal of Field Robotics*, vol. 40, no. 3, pp. 747–779, Dec. 2022.

[16] M. Pidwirny, "Introduction to the oceans," 2006, fundamentals of Physical Geography, Date Viewed: 2013 March.

[17] B. Worm *et al.*, "Impacts of biodiversity loss on ocean ecosystem services," *Science*, 2006, accessed: 2023-12-24. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1132294

[18] N. G. Education, "Learn about the ocean – anytime, anywhere," 2021, visited on 14/05/2024. [Online]. Available: https://blog.education.nationalgeographic.org/2021/06/04/learn-about-the-ocean-anytime-anywhere/

[19] K. L. Nash *et al.*, "Oceans and society: feedbacks between ocean and human health," *Reviews in Fish Biology and Fisheries*, vol. 32, no. 1, pp. 161–187, 2022.

[20] NOAA, "Historical maps and charts audio podcast," National Ocean Service website, 2017, visited on 13/08/2017. [Online]. Available: https://oceanservice.noaa.gov/podcast/july17/nop08-historical-maps-charts.html

[21] R. Coulson, J. C. Lambiotte, G. Grenon, T. Pantelakis, J. Curran, and A. An, "Development of a modular docking sub-system for 12" class autonomous underwater vehicles," *Oceans '04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600)*, vol. 3, pp. 1745–1749 Vol.3, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:37498450

[22] V. P. Shah, "Design considerations for engineering autonomous underwater vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[23] B. W. Hobson, R. S. McEwen, J. Erickson, T. Hoover, L. McBride, F. Shane, and J. G. Bellingham, "The development and ocean testing of an auv docking station for a 21 auv," in *IEEE OCEANS 2007 Conference*, 2007, pp. 1–6.

[24] R. P. Stokey, M. Purcell, N. C. Forrester, T. C. Austin, R. G. Goldsborough, B. Allen, and C. von Alt, "A docking system for remus, an autonomous underwater vehicle," *Oceans '97. MTS/IEEE Conference Proceedings*, vol. 2, pp. 1132–1136 vol.2, 1997. [Online]. Available: https://api.semanticscholar.org/CorpusID:109770827

[25] C. Yang, S. Peng, S. Fan, S. Zhang, P. Wang, and Y. Chen, "Study on docking guidance algorithm for hybrid underwater glider in currents," *Ocean Engineering*, vol. 125, pp. 170–181, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801816303286

[26] Y. Li, Y. Jiang, J. Cao, B. Wang, and Y. Li, "Auv docking experiments based on vision positioning using two cameras," *Ocean Engineering*, vol. 110, pp. 163–173, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801815005521

[27] P. A. Wilson, "Autonomous homing and docking tasks for an underwater vehicle," *IFAC Proceedings Volumes*, vol. 42, no. 18, pp. 304–309, 2009, 8th IFAC Conference on Manoeuvring and Control of Marine Craft. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016319127

[28] N. Hurtós, A. Mallios, N. Palomeras, J. Bosch, G. Vallicrosa, E. Vidal, D. Ribas, N. R. Gracias, M. Carreras, and P. Ridao, "Loon-dock: Auv homing and docking for high-bandwidth data transmission," *OCEANS 2017 - Aberdeen*, pp. 1–7, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:45033261

[29] P. Batista, C. Silvestre, and P. Oliveira, "A two-step control approach for docking of autonomous underwater vehicles," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 10, pp. 1528–1547, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.3158

[30] J.-Y. Park, B.-H. Jun, P.-M. Lee, Y.-K. Lim, and J.-H. Oh, "Modified linear terminal guidance for docking and a time-varying ocean current observer," *2011 IEEE Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, pp. 1–6, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:45388035

[31] B. Allen, T. Austin, N. Forrester, R. Goldsborough, A. Kukulya, G. Packard, M. Purcell, and R. Stokey, "Autonomous docking demonstrations with enhanced remus technology," in *OCEANS 2006*, 2006, pp. 1–6.

[32] R. Stokey, B. Allen, T. Austin, R. Goldsborough, N. Forrester, M. Purcell, and C. von Alt, "Enabling technologies for remus docking: an integral component of an autonomous ocean-sampling network," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 487–497, 2001.

[33] K. Vickery, "Acoustic positioning systems: A practical overview of current systems," in *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles, AUV'98*, 1998, pp. 5–17.

[34] G. M. de Goede and D. Norris, "Recovering unmanned undersea vehicles with a homing and docking sonar," in *Proceedings of the IEEE OCEANS 2005 Conference*, 2005, pp. 2789–2794.

[35] D. Stramski, E. Boss, D. Bogucki, and K. J. Voss, "The role of seawater constituents in light backscattering in the ocean," *Progress in Oceanography*, vol. 61, no. 1, pp. 27–56, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079661104000692

[36] W. Naeem, R. Sutton, and S. Ahmad, "Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for cable/pipeline tracking," *Proceedings of the Institution of Marine Engineers, Part C: Journal of Marine Science and Environment*, pp. 25–35, 2004.

[37] S. MahmoudZadeh, A. M. Yazdani, K. Sammut, and D. Powers, "Online path planning for auv rendezvous in dynamic cluttered undersea environment using evolutionary algorithms," *Applied Soft Computing*, vol. 26, pp. 515–521, 2018.

[38] S. M. Zadeh, D. M. W. Powers, and K. Sammut, "An autonomous reactive architecture for efficient auv mission time management in realistic severe ocean environment," 2016. [Online]. Available: https://arxiv.org/abs/1604.08336

[39] S. M. Zadeh, K. Sammut, D. M. W. Powers, A. Atyabi, and A. M. Yazdani, "A hierarchal planning framework for auv mission management in a spatio-temporal varying ocean," 2017. [Online]. Available: https://arxiv.org/abs/1604.07898

[40] S. MahmoudZadeh, D. M. Powers, K. Sammut, A. M. Yazdani, and A. Atyabi, "Hybrid motion planning task allocation model for auv's safe maneuvering in a realistic ocean environment," *Journal of Intelligent Robotic Systems*, pp. 1–18, 2019.

[41] M. Chyba, T. Haberkorn, S. B. Singh, R. N. Smith, and S. K. Choi, "Increasing underwater vehicle autonomy by reducing energy consumption," *Ocean Engineering*, vol. 36, pp. 62–73, 2009.

[42] M. B. Loc, H. S. Choi, S. S. You, and T. N. Huy, "Time optimal trajectory design for unmanned underwater vehicle," *Ocean Engineering*, vol. 89, pp. 69–81, 2014.

[43] S. Cowen, S. Briest, and J. Dombrowski, "Underwater docking of autonomous undersea vehicles using optical terminal guidance," in *Oceans '97. MTS/IEEE Conference Proceedings*, vol. 2, 1997, pp. 1143–1147 vol.2.

[44] P. Jantapremjit and P. Wilson, "Optimal control and guidance for homing and docking tasks using an autonomous underwater vehicle," in *International Conference on Mechatronics and Automation*, 2007, pp. 243–248.

[45] ——, "Guidance-control based path following for homing and docking using an autonomous underwater vehicle," in *OCEANS Conference*, 2008, pp. 1–6.

[46] S. Ferrari and G. Foderaro, "A potential field approach to finding minimum exposure paths in wireless sensor networks," in *International Conference on Robotics and Automation*, 2010, pp. 335–341.

[47] J.-Y. Park, B. huan Jun, P. mook Lee, and J. Oh, "Experiments on vision guided docking of an autonomous underwater vehicle using one camera," *Ocean Engineering*, vol. 36, no. 1, pp. 48–61, 2009, autonomous Underwater Vehicles. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801808002242

[48] J. Park, B. Jun, K. Kim, P. Lee, J. Oh, and Y. Lim, "Improvement of vision guided underwater docking for small auv isimi," in *OCEANS 2009 Conference*, 2009, pp. 1–5.

[49] V. Rigaud and B. Nicolas-Meunier, "From manned to autonomous and hybrid underwater systems - a review of existing operational systems, development trends and ifremer/dcns cooperative projects," in *2015 IEEE Underwater Technology*. Chennai, India: Natl Inst Ocean Technology, 2015, pp. 1–10.

[50] M. Richard, R. Bourdon, and A. Vettoretti, "Drone docking underwater – acoustic based system extends drone performance," *Subsea Technology Magazine*, pp. 13–14, 2018.

[51] J. Braga, "Control of underwater vehicles on autonomous docking maneuvers," Ph.D. dissertation, Faculdade de Engenharia da Universidade do Porto, 2010.

[52] B. Ferreira, A. Matos, N. Cruz, and A. Moreira, "Homing a robot with range-only measurements under unknown drifts," *Robot. Auton. Syst.*, vol. 67, pp. 3–13, 2015.

[53] E. Sarda and M. Dhanak, "Launch and recovery of an autonomous underwater vehicle from a station-keeping unmanned surface vehicle," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 4, pp. 1007–1017, 2018.

[54] A. Sans-Muntadas, K. Y. Pettersen, E. Brekke, and V. F. Henriksen, "A hybrid approach to underwater docking of auvs with cross-current," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–7.

[55] K. J. A. ström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, draft v2.4a ed. Lund, Sweden and Pasadena, CA: Self-published, September 2006, © 2006 Karl Johan Åström and Richard M. Murray. All rights reserved.