Contribution to the model and navigation control of an autonomous underwater vehicle

by

Julián González Agudelo

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Enginyeria Electrònica

Contribution to the Model and Navigation Control of an Autonomous Underwater Vehicle

Julián González Agudelo

PhD Thesis
in Electronic Engineering

July 2015

Contribution to the Model and Navigation Control of an Autonomous Underwater Vehicle

Julián González Agudelo

PhD Thesis

Electronic Engineering Department

Universitat Politècnica de Catalunya

Supervisor:
Dr. Spartacus Gomáriz

Co-Supervisor:
Dr. Carles Batlle

Thesis submitted in partial fulfillment of the requirement for the PhD Degree issued by the Universitat Politècnica de Catalunya, in its Electronic Engineering Program.

July 2015

A copy of this PhD thesis can be downloaded from: http://www.tdx.cat/
http://upcommons.upc.edu/

Date of defense:
Date of this version: July 1, 2015

Julián González Agudelo
SARTI Research Group
Universitat Politècnica de Catalunya
Rambla de l'Exposició 24 Edifici C, 08800
Vilanova i La Geltrú, Spain
<julian.gonzalez.agudelo@upc.edu>

Abstract

This thesis deals with the further development of an existing underwater vehicle for autonomous navigation. The vehicle was conceived to navigate over the sea surface and, at certain fixed points, to dive vertically in order to obtain a profile of a water column. The main objectives of the thesis are the improvement of the hardware and software of the vehicle in order to make it fully operational, and the design and implementation of control techniques for autonomous navigation.

The problem of autonomous navigation is addressed first with the calculation of an hydrodynamic model in 3DoF. An extensive study about the selection of the coefficients is performed, using a linearized model. The calculation of the coefficients is done using two approaches: a geometric one and another one based on least squares techniques applied to experimental data obtained during sea trials. The least squares method gives satisfactory results and the simulations fits the experimental data. The resulting hydrodynamic model is completed with the physical constraints of the actuators of the vehicle.

Solving the autonomous navigation problem requires the design of controllers for both the inner loop (dynamic) and the outer loop (kinematic). Several solutions based on type-1 TSK fuzzy control are presented for velocity control, yaw control, pure pursuit navigation, and path following. The fuzzy controller is used to manage different linear controllers designed for specific conditions. The hydrodynamic model plays an important role in the design of the controller for the inner loop. In addition, a gain scheduled controller is designed to validate a particular case of the fuzzy controller in the inner loop.

Regarding the finishing of the vehicle to be fully operational, the improvements begin with a new driver for the lateral thrusters because they lacked backwards movement capability. Additionally, upgrades in the handling of the vehicle had to devised. In this respect, a wireless on/off system is presented to power the vehicle, and a WiFi connection is adapted to manipulate the software of the vehicle remotely. Furthermore, a study of the currents and power of the immersion system in order to reduce the power consumption is performed, and the hardware is improved with the inclusion of some commercial devices, like an IMU, CTD, and acoustic localization system.

The software is improved in several aspects. First, some problems derived from previous works are debugged. The system is then restructured with a multithread development, which provides robustness and modularity. As the system needed an extension of the protocol communication for easy handling, a robust protocol communication is implemented with the possibility to execute scripts. Finally, the existing graphical user interface is simplified in order to provide only the information required by the operator.

In order to improve the buoyancy of the vehicle, several foams are designed, adjusted to the geometry of the vehicle, and a ballast system is also included for fine adjustment.

Finally, several tests in the laboratory, a swimming pool, a channel, and at sea are performed in order to check the performance of the vehicle. Results show a correct behavior of hardware and software, and also validate the performance of the controllers designed for autonomous navigation.

Keywords: Autonomous underwater vehicle, hydrodynamic model, autonomous navigation and control, TSK fuzzy control, PID control, gain scheduling control, vehicle hardware, vehicle software.

Acknowledgements

I would like to thank a large number of people that in many different ways have helped me in the accomplishment of this PhD thesis.

In first place, I wish to thank my supervisor Professor Spartacus Gomáriz, who first welcomed me in Barcelona, for his constant help and support. He has always provided excellent scientific guidance to cover all the topics. I appreciate the confidence he has placed in me, and his patience in listening to all my questions and problems and his effort in revising all my work. I also would like to thank my co-supervisor Professor Carles Batlle for his help in the resolution of the modeling, mathematical and control issues and his assistance in writing this work.

I would like to thank the people of the SARTI group, and my friends Ivan Masmitjà, Oriol Pallarés, David Sarriá, Normandino Carreras and César Galarza, with which I shared so many lunches, great laughs, and endless discussions on the most diverse topics. Thanks for sharing the enthusiasm and values of friendship and collaboration. My special thanks to Ivan for his involvement in the project, his initiative in improving the vehicle, and his collaboration in the field tests.

I wish to thank the scholarship program of the UPC for the FPI grant that was given to me for my stay and for the completion of the doctoral thesis in Barcelona.

During the Autumn of 2013 I had the chance to spend three wonderful months at the Group of Ocean Robotics of the Institute for Systems and Robotics (ISR) of the Instituto Superior Técnico, Lisbon, Portugal. I am indebted to Professor Antonio Pascoal for making this visit possible, for treating me like one of his students, and specially for his support to correct and to finish the hydrodynamic model of the vehicle.

My special gratitude to the professor who put me in touch with the SARTI group to perform this work, Professor Gerard Olivar of the National University of Colombia - Manizales Campus. I cannot forget all my friends and professors I met in this university, great people and talents, who shared with me their motivation, good research, and invaluable appreciation and friendship.

And last, but not least, my gratitude to my parents, to all my family and friends for trusting me and for their support during all these years. Finally, a very special and sincere thanks goes to Idalia, my love, for her unconditional support. I would like to dedicate this thesis to her and my son Joaquim, you are my inspiration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The study of the sea and oceans has gained considerable importance among fishers and biologists since the marine species change their behavior depending on environmental variables such as temperature, salinity, ph, chlorophyll, nitrate, among others [145] [80]. Some fishes, for example, are euryhaline, which implies that they can adapt to variations in the salinity concentration according to the season; and other types of fish are stenohaline, that is to say they do not support these variations and have to emigrate to other waters [38].

These environmental measurements also depend on the depth where terms like haloclines, vertical zone in which salinity changes rapidly, or thermoclines, vertical zone in which temperature changes rapidly, are studied [159] and it is necessary to obtain profiles of water columns in order to estimate the emigrations or fish population in a determinated area.

Additionally, as cited in [13], knowledge of the salinity distribution and its variability in the ocean are crucial in understanding the role of the ocean in the climate system. It is well known that ocean currents and air-sea heat fluxes regulate our climate.

For these oceanographic observations there are different tools used. One of them are the satellites, which make measurements of the ocean a global level from the aerospace. However, the observations are only over the surface but not underwater, therefore there is too much data that can not be registered.

Another alternative are the oceanographic ships because they are in direct contact with the ocean and can take measurements underwater. However the actual planning and mission deployment required to obtain a satisfactory spacial-time resolution data are very expensive.

As a solution to these needs, these tools are migrating towards the creation of autonomous marine vehicles such as Gliders, AUV (Autonomous Underwater Vehicles) and ASV (Autonomous Surface Vehicles) [86] [16]. Since they are unmanned they are of small dimensions and the mission deployment is more cheaper than the oceanographic ships. Thereby, the research in this field is focused on autonomous navigation, which envelops vehicle modeling and control.

1.1 Previous work

The *Technological Development Center of Remote Acquisition and Data Processing Systems* (SARTI Group), attached to the Electronics Department of the Universitat Politècnica de Catalunya, have been developing an autonomous underwater vehicle during the last years under the project Guanay. It is a vehicle conceived for shallow water, and it is conceived to navigate over the surface making vertical immersions at preestablished waypoints.

The project started with a first prototype [51]. However, the vehicle had some problems with the tightness and therefore a new structure was designed [48] and manufactured. This prototype was called GUANAY II. Next the hardware and software were reformed and improved [82], with good results but not enough to be fully operational when commanded manually.

1.2 Motivation

The idea of an autonomous underwater vehicle for oceanographic observation is almost finished, and this development is the principal motivation of the present thesis. The vehicle can be used as a complement to the *Expandable Seafloor Observatory* OBSEA (`www.obsea.es`) [94] which is located 4 km off the coast of Vilanova i La Geltrú (Barcelona). The GUANAY II can be used as an added element to the sensor network, taking into account that the principal aspect of a sensor network compared with a simple set of sensors is its cooperative behavior and the capacity of processing all the data in conjunction.

The motivation of this thesis is twofold. First the need to complete the development of an operating vehicle for marine observation, that is, the hardware and software; and second the need to develop navigation control techniques in order to give autonomy to the vehicle, taking into account its physical characteristics and operational requirements.

Regarding the first objective, the vehicle needs further developments in the buoyancy system, the hardware for the lateral thrusters so that it can go backwards, an inertial navigation system, a circuit to measure and study the power consumption, and an improvement of the software of the base station and the vehicle in order to have more simplicity and robustness.

Regarding the second objective, the navigation control is an important step in order to give autonomy to the vehicle. First of all, a good hydrodynamic model is required in order to estimate its performance against forces and torques. The principal problem with the modeling is selection of the hydrodynamic coefficients to be included and their calculation. Prestero [105] proposes a methodology to calculate them using a set of equations based on the vehicle's geometry, and these equations can be used for the GUANAY II. Additionally, other works calculate the hydrodynamic model using maximum-likehood estimation [134], least squares [112], and a Kalman filter [144], among other techniques.

Regarding the automatic control, the state of the art shows that the problems to be solved include techniques like *point stabilization*, *trajectory tracking*, *path following*, *path tracking*, among others. All of them can be achieved by means of several control strategies which, however, must be selected according to the vehicle's physical characteristics and required performance.

GUANAY II should have an electronic control system for autonomous navigation. Given a pre-established definition and scheduling of the path, the vehicle should be capable to follow it without human intervention in a robust way, able to deal with different environment perturbations and minimizing the energetic cost.

The review of the state of the art shows that controllers can be designed for a specific forward velocity in order to simplify the control. This is all right for a vehicle that navigates in open water normally, where the operating velocity is constant. However, if it navigates near the coast, in the subsoil or inside a harbor, it would be interesting to change the forward velocity, and thus to change the operation point for which the controller is designed. In this way, the path can be adjusted, keeping the benefits of the trajectory tracking technique.

Lyapunov methods have also been used in the literature, but they lead to quite cumbersome and complex control laws. Moreover, non-linear methods in general do not enjoy the same level of popularity as the their linear counterparts among the designers.

An alternative approach is presented by Silvestre [125], where controllers are designed for different velocities and they are later integrated into a single law using *linear matrix inequalities* in an *infinity norm* framework.

The approach selected in this work is the application of the *gain scheduling* methodology [119] and the use of fuzzy control as an alternative for the integration of the different linear controllers [120] [126]. The use of the fuzzy control method is feasible since it establishes rules for activation and deactivation (defining activation as the effective contribution of a rule to the control action) which can be controlled. This is known as "mapping" of the space of the input variables [32], which allows to assign control actions for different system operation points or to reduce large perturbations near their settling state.

Therefore the design of control systems for the tracking of paths for the navigation of autonomous underwater vehicles can be enclosed in the context of system control where the system is a piecewise linear system, and opens a way for the application of fuzzy control techniques with their capacity of assignation of differentiated control actions ("zonal" control).

1.3 Goal of the thesis

After the description of the research antecedents and motivation, the goal of this thesis is stated. The general purpose is summarized as:

"The improvement of the hardware and software of an autonomous underwater vehicle in order to be fully operational. The calculation of an hydrodynamic model. And the design and implementation of control techniques for autonomous navigation".

1.4 Thesis main contributions

Part I. Operability of Guanay II AUV

- Chapter 3. Buoyancy
  - Design of foams for buoyancy
  - Ballast system for fine adjustment
- Chapter 4. Hardware
  - Lateral thruster driver to go forward and backward
  - RF remote control on-off of the power system
  - Studies in the forces and electric currents for the immersion system
  - Addition of a WiFi connection for easy handling
  - Adaptation of commercial devices: IMU, CTD, acoustic locatization system.
- Chapter 5. Software
  - Multithread operation
  - Robustness and ampliation of the protocol communication
  - Incorporation of a script execution
  - Improvement of the graphical user interface

Part II. Hydrodynamic model and control

- Chapter 7. Modeling
  - Calculation of the hydrodynamic model of Guanay II AUV in 3DoF, comprising the selection of the coefficients and their calculation
  - Issues of physical constraints
- Chapter 8. Automatic control
  - Design of a type-1 TSK fuzzy controller to control the yaw and the forward velocity as an interpolator of different linear controllers
  - Design of a type-1 TSK fuzzy controller for pure pursuit navigation
  - Design of a type-1 TSK fuzzy controller in conjunction with another controller for path following

Part I

Operability of Guanay II AUV

Chapter 2

Review of unmanned vehicles and Guanay II AUV

Research in the sea and oceans is a challenging area of science in this time due to its complexity and its importance for climate chance, biodiversity and resource exploration. It comprises several disciplines and involves many tools that are in constant development and improvement for specialized tasks. For instance, regarding data recollection, there are three principal characteristics: the *temporal resolution*, which defines the volume of data taken in a given point; *spatial resolution*, which defines the precision of the measure; and the *total observable extension*, which defines the area or distance from which the instrument can take data in a specific zone.

Until quite recently, the instruments used for sounding the marine subsoil were principally dead-weights. The british explorer James Clark Ross was the first man to use this system in order to explore the Artic Ocean in 1840, where he determined depths up to 3700 m [115]. The worldwide Challenger expedition of 1872-1876 [140] used a similar system in order to sound the subsoil. It had the objective of investigating the physical and biological conditions in the oceanic basins, and it is considered by many as the beginning of modern oceanography.

Up to the middle of the XXth century, the mainstay of oceanographic exploration was the use of dedicated buoys and ships. This allowed the obtaining of samples with a greater temporal frequency and at the same time longer missions, capable to sense more volume of data in certain geographic areas. The main drawback of submarine surveys with ships is that they are quite expensive, both in terms of the needed infrastructure and of the operational costs.

In parallel, space technology started to be used for ocean exploration. The use of satellites to monitor the seas and oceans allows the survey of large regions. However, it has the drawback of a lower resolution, both in the temporal and spatial domains.

As response to these problems, unmanned vehicles were developed beginning in the 50's decade. These vehicles were designed first for military operations (as mine counter measures, anti-submarine warfare and rapid environmental assessment), and later their development was extended to industry and scientific surveys. The principal characteristic of these vehicles is their low cost with respect to manned vehicles, and the use of a robust infrastructure for many operations.

This thesis deals with the development and improvement of one of these vehicles, the Guanay II , specially in the area of modeling and navigation control. The next sections are structured as follows: Classification of unmanned vehicles, description of the class where the

GUANAY II belongs, and the motivation of the thesis, and finally the structure of the thesis.

2.1 Unmanned Vehicles

Figure 2.1 shows a classification of these unmanned vehicles: The first division is for surface vehicles and underwater vehicles. In the case of surface vehicles there are the ASVs (Autonomous Surface Vehicles), which can be propelled or use the waves energy for propulsion (Waveglider). In the case of underwater vehicles there are the ROVs (Remotely Operated Vehicles) which are tethered, and the AUVs (Autonomous Underwater Vehicles) which are untethered. The AUVs can be classified in Propelled AUVs, underwater gliders and biomimetic AUVs.

Each one coincide in a good temporal and spatial resolution, and have their differences in the total observable extension regarding the type of movement and the use of the energy for its propulsion. A short description of each one of these unmanned vehicles are exposed in the next section, their characteristics and capabilities in the different marine environments.

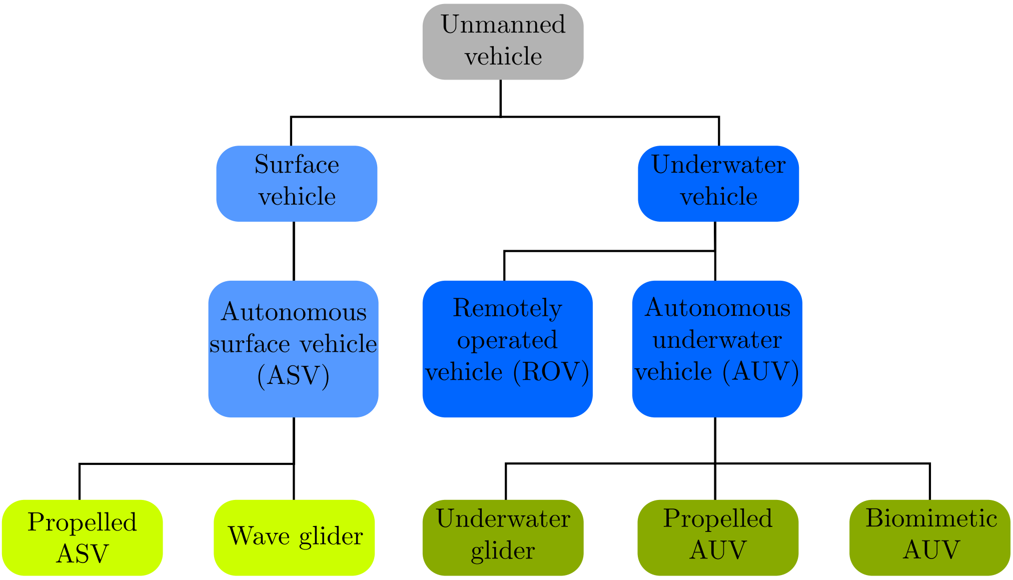
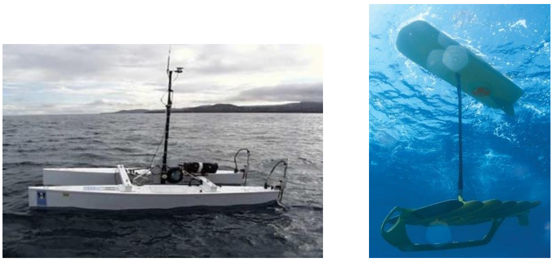


Figure 2.1: Underwater vehicles classification

2.1.1 Autonomous Surface Vehicles (ASV)

Autonomous surface vehicles (also called *Unmanned surface vehicles* - USV) are vehicles that operate on the surface of the water without a crew. They are valuable in oceanography, as they are more capable than moored of drifting weather buoys, but far cheaper than the equivalent weather ships and research vessels, and more flexible than commercial-ship contributions. An example of these is the Delfim vehicle [5]. Military applications for ASV include powered seaborne targets. A detailed state of the art can be found in [76] and [14].

ASV are self-propelled, but there is a special vehicle called *wave glider* [124], which uses the ocean's endless supply of wave energy for propulsion, so it has the ability to stay out at sea for long stretches and through many weather conditions.



(a) (b)

Figure 2.2: (a) The Delfim ASV. From [34]. (b) Wave glider.

2.1.2 Remotely Operated Vehicles (ROV)

The principal characteristic which serves to classify underwater vehicles is the autonomy. A ROV is a submarine robot that is operated remotely from a ship by a human [25]. It is a tethered robot, *i.e.* all the control systems and the energy transmission are supplied through a cable (also called umbilical) from the ship from which it is operated. The length of the tether changes with the working depth of the ROV, which can be up to 6000 m deep.

This fact makes ROVs non autonomous. However, the power supply through the wire means a bigger endurance for the vehicle, providing it with the power to use up to 8 thrusters in order to obtain a correct position and attitude, and to use one or two arms in order to do manipulation tasks. For this reason, ROVs rarely have a hydrodynamic shape, and instead are of compact design, generally with a rectangular aspect.

Vehicles that are submerged too deep use a complement called TMS (Tether Management System), which is immersed with the ROV and allows a better management of the cable, avoiding the entanglement of wires and decreasing the forces induced in the cable by the waves. Figure 2.3 shows the Quantum ROV [107] by SMD company, and its TMS in a regular mission at sea.

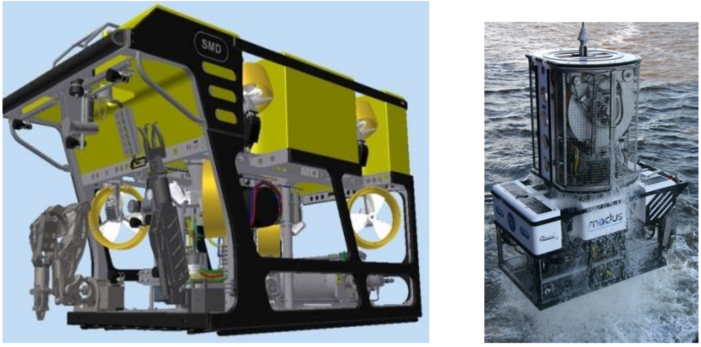


Figure 2.3: Quantum ROV by SMD

ROVs are used for several tasks, which can be classified according to observation and manipulation. Table 2.1 shows the different categories regarding size, onboard horsepower and depth capabilities [118].

Table 2.1: ROV categories

Class	Max. depth [m]	Power [hp]
Micro observation	100	5
Mini observation	300	10
Light/Medium Work Class	2000	100
Observation/Light Work Class	3000	20
Heavy Work Class/Large Payload	3000	300
Observation/Data Collection	3000	25
Heavy Work Class/Large Payload	3000	120

2.1.3 Autonomous Underwater Vehicles (AUV)

Autonomous underwater vehicles are conceived for autonomous navigation without the help of an operator. They do not use a tether for communication or power supply, and have instead an embedded control and a pack of batteries in order to power up all the sensors and actuators needed in the mission. This type of vehicle usually has an optimized hydrodynamic shape in order to improve its performance and decrease power consumption.

AUVs have been developed since the 50's decade, beginning in the Washington University in 1957 with the SPURV [154], and since then with many universities and institutions have taken up the task of developing improved vehicles adapted to new requirements.

Due to the limited power, AUVs are mainly dedicated to observation and data recollection, letting manipulation tasks to ROVs. However, in the last decade there have been AUV for intervation, called I-AUVs, like the Girona 500 [110]. AUVs can be developed for military, scientific or offshore uses. Their capabilities and applications are described below.

- Militar:
  - Mine counter measures (MCM)
  - Anti-submarine warfare (ASW)
  - Rapid environmental assessment (REA)
  - Surveillance
  - Search & Recovery
  - Port security
- Scientific:
  - Oceanography
  - Limnology
  - Habitat assessment
  - Hydrography
  - Bathymetric surveys
  - Archeology
  - Wreck location & mapping
  - Under ice surveying

- Offshore:
  - Bathymetric / sub-bottom / environmental surveys
  - Exploration and construction support
  - Post-hurricane inspection
  - Inspection of pipelines and platforms

AUV glider

A particular case of AUV is the AUV glider [137, p. 407], which uses small changes in its buoyancy in conjunction with wings to convert vertical motion to horizontal, and thereby propel itself forward with very low power consumption. While not as fast as conventional AUVs, gliders using buoyancy-based propulsion represent a significant increase in range and duration compared to vehicles propelled by electric motor-driven propellers, extending ocean sampling missions from hours to weeks or months, and to thousands of kilometers of range. Gliders follow an up-and-down, saw-tooth like profile through the water. Due its movement, gliders are used for data recollection in environmental surveys and other scientific studies.

An example of these is the *Scarlet Knight* [139], the first submarine robot to cross the Atlantic Ocean (see figure 2.4). This record was reached in 2009, traveling from New Jersey (USA) in April and finishing in Baiona (Spain) in December, 221 days in total. The vehicle only consumed the 60 % of the batteries, demonstrating the low consume of gliders [79].

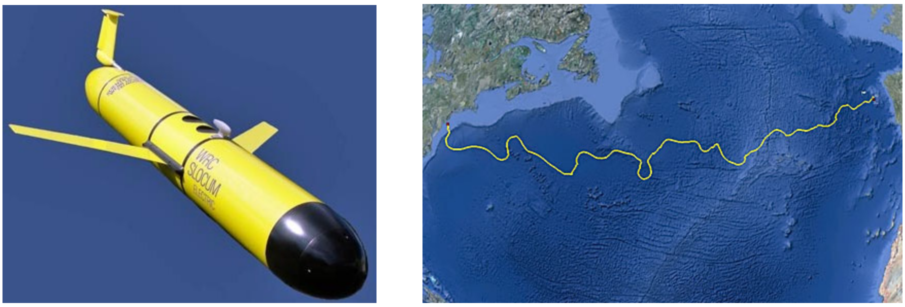


Figure 2.4: *Scarlet Knight Glider* and its journey in the Atlantic Ocean

Biomimetic AUV

Biomimetic AUVs are vehicles that copy the propulsion system directly from the animal world. There are different models, but all are based in the movements of marine species applied to robotics, for example, the movements of jellyfish, tuna or rays. In this way, the advantages of the animal world are exploited and embodied into a mechanical structure. A detailed state of the art and future research is explained in [148].

Figure 2.5 shows an example of these vehicles [78]. This is a soft-bodied robot created at MIT (2014) which employs a compliant body with embedded actuators emulating the slender anatomical form of a fish and has a fluidic actuation system that drives body motion.

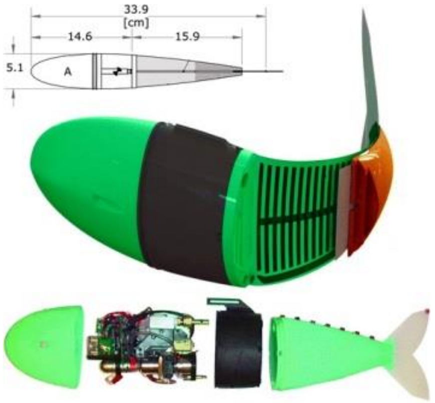


Figure 2.5: Autonomous soft robotic fish

Propelled AUV

Propelled AUVs are vehicles which were developed in parallel with ROVs [137, p. 408]. Instead of gliders, these vehicles have thrusters to realize their movement, generally with one thruster and a rudder for the attitude. This implies more energetic cost compared with gliders, so the endurance of the mission is generally about hours. However, they have the advantage of a bigger maneuverability which allows their use in tasks like mapping, searching, inspection of pipelines, among others, which are not possible with gliders. Figure 2.6 shows the AUV Munin [68, p. 84] designed for inspection of pipelines and platforms for the offshore industry.

Figure 2.6: Munin AUV by NTNU/SINTEF. Inspection of pipelines and platforms

2.2 Guanay II AUV

The Guanay II AUV is a vehicle developed by the *Technological Development Center of Remote Acquisition and Data Processing Systems* (SARTI for its acronym in spanish), and it is the basis for the research of this thesis.

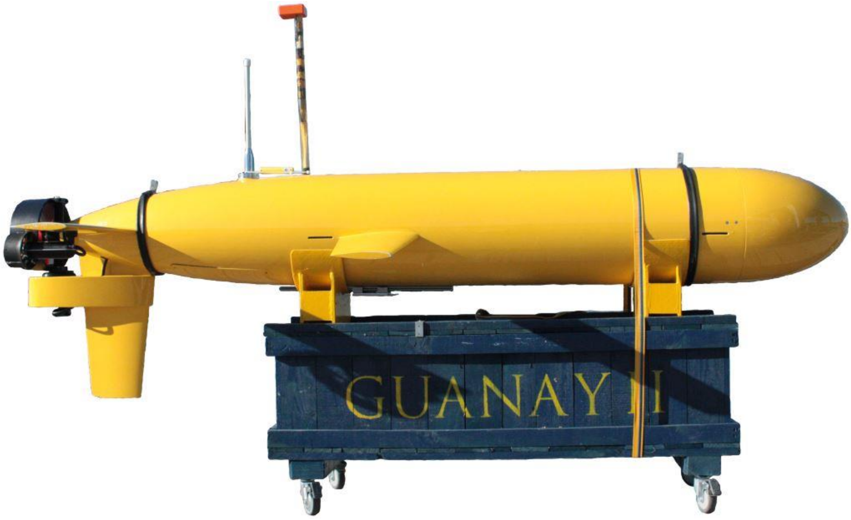


Figure 2.7: Guanay II AUV

2.2.1 Purpose

This vehicle was conceived to navigate on shallow water, up to 40 m. Its operational principle is to navigate over the sea surface following preestablished waypoints or a continuous path, stop at the required point, and then make a vertical immersion to take data on salinity, temperature, or other variables. The vehicle goes then up to the surface and continues with the path. This type of movement can be seen in figure 2.8. The data collection is useful for biologists and oceanographers for different studies, for instance the determination of haloclines or thermoclines, which are closely linked to the migration of fish.

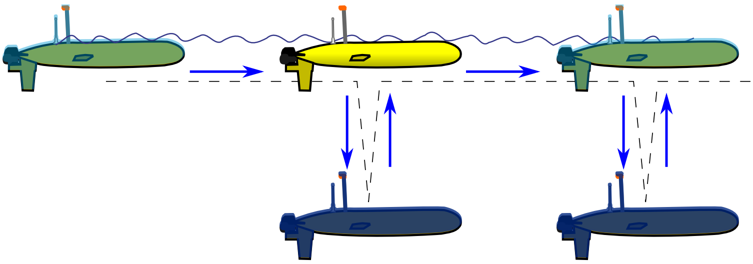


Figure 2.8: Principle of motion

2.2.2 Vehicle parts

As shown in figure 2.9, the external hull is a torpedo-shaped structure made of fiberglass, measuring 2.3 m long by 0.32 m wide. This hull is not watertight, as it has small holes through which water enters. Inside there is a module of aluminum, which is watertight and has all the hardware (see chapter 4 for further details). The module's tightness is ensured with an O-ring, and the attachment to the top is by pressure.

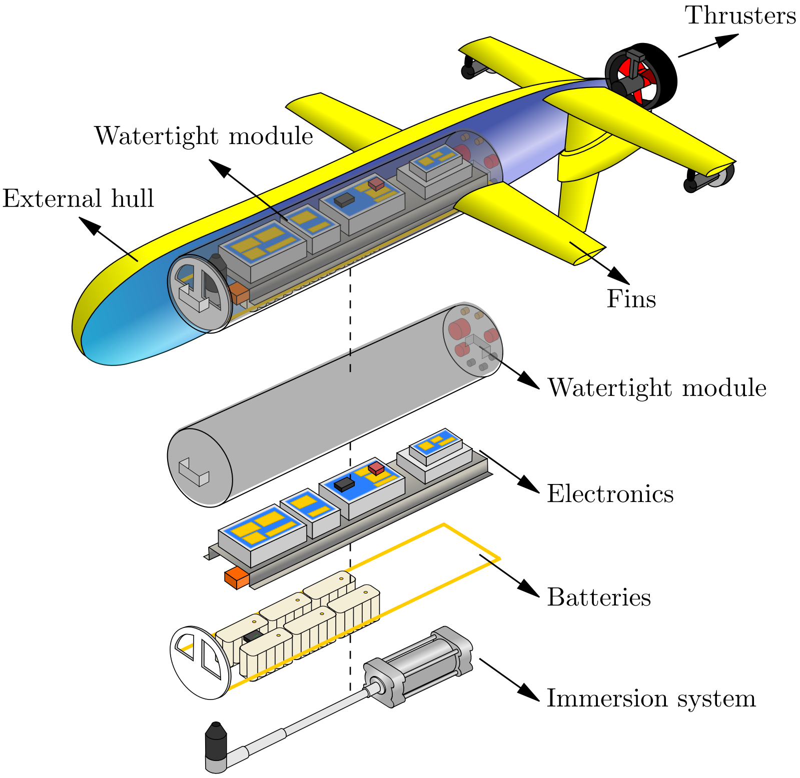


Figure 2.9: Parts of GUANAY II AUV

The vehicle has three thrusters, which classify the GUANAY II as a propelled AUV. The principal thrust engine, from Seaeye Company [141], is located in the stern and on the longitudinal axis of the vehicle. Two thrusters by Seabotix Company [142] have been placed on both sides of the stern to control the direction of the vehicle.

For immersion, the vehicle has a cylinder located inside of the watertight module. When the vehicle goes down the cylinder takes water to make the vehicle more dense than water. Conversely, when it goes up, the cylinder is emptied to make the vehicle less dense than water. The using of a cylinder has been chosen, instead of a vertical thruster, for two important reasons. First for energy saving, because a vertical thruster would imply a constant current consumption while the cylinder does not. And second, because the data collection of salinity needs minimum perturbations in the environment, which would not be guaranteed if a vertical thruster were used.

Finally, the vehicle has 5 fins, two pairs of them in a horizontal position to minimize rolling, and one in vertical position, a keel, to provide stability going forward.

2.2.3 Evolution of the project

The SARTI group has been working in the development of an AUV for some time (see Álvarez and Olivar [4] for one of the first publications). This project was born with the name *Cormorán* [114] because there is a family of aquatic birds with this name that catch the prey by diving from the surface. The funds of the project were shared with the IMEDEA institute. However, it was renamed to *Guanay*, a species of cormorants of South America, in order to differentiate it from the vehicle developed by the IMEDEA.

Works by Prat et al. [104] and Vidal et al. [152] show the design of the first prototype, which had a structure of PVC. This vehicle was finished and tested in a swimming pool with the work of Galo [46], and Viñolo and Pallarés [153], which was published in [51] [49] [50]. The resulting prototype had, however, serious tightness problems with. Figure 2.10 shows this first prototype developed by the SARTI group.

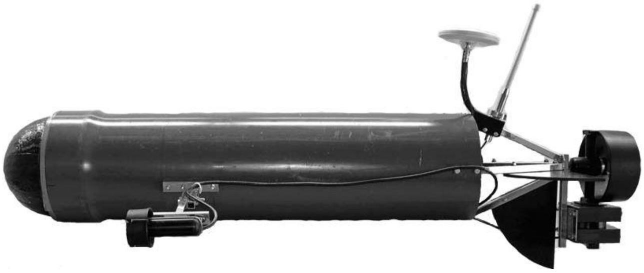


Figure 2.10: Guanay project: First prototype

Hydrodynamic studies were started in order to design the second prototype of the vehicle [48], which was called GUANAY II, and the external structure was changed to fiber glass, and the hardware was encased in a module with a robust tightness. The hardware was extended and improved by Masmitjà and Masmitjà [82], and further developed in [81].

2.2.4 Problems and needs for the operability

The latests developments made in the vehicle were very satisfactory. However, there remained some problems and needs to be solved in order to guarantee the operability of the vehicle. These needs can be divided into four groups: buoyancy, hardware, software, and autonomy.

- *Buoyancy.* The vehicle needed the addition of foams to improve buoyancy. These foams had to be designed taking into account the geometry of the vehicle and its weight. Additionally, a ballast system was required for fine adjustment.

- *Hardware.* The hardware adjustments can be further divided into 6 groups: Control unit, power, communication, navigation, actuators, and safety. First it was necessary to devise a new driver for the lateral thruster in order to enable it to thrust backwards, which would also improve the turning capabilities. Next, it was necessary to perform a complete study of capabilities for diving, that is, the electric power associated to diving maneuvers and the maximum depth attainable. A new system to start the vehicle for easy handling and a sensor for power consumption was also needed. Respect to communications, it was necessary to include a WiFi connection to access the operating system remotely, avoiding the need to open the watertight module. Finally, it needed some calibration of the sensors and the addition of some commercial devices.

- *Software.* The software had to be restructured and improved. First, it required multithread operation, that is, one thread for each task in order to interact with different sensors at different frequency acquisition. Second, it needed more commands to configure and to control the vehicle from the base station. The software lacked scripting capability, which is very useful to facilitate the work of the operator. Finally, the software had several bugs to be solved, and a new, more intuitive graphical interface had to be designed.

- *Autonomy.* The vehicle required a satisfactory controller for autonomous navigation. It was necessary to calculate the hydrodynamic model of the vehicle in 3DoF to determine its behavior in front of different situations in the surface, and a controller had to be designed using this model as a base.

Chapter 3

Buoyancy

The vehicle needs an adjustment in its buoyancy, because the components that it uses, like aluminum, batteries, electronics, thrusters and so on, are more dense than water. This chapter will explain the foam and ballast system added to the vehicle in order to produce a positive buoyancy for the vehicle and to stabilize it in pitch movement.

3.1 Foams

Archimedes' principle indicates that the upward buoyant force that is exerted on a body immersed in a fluid, whether fully or partially submerged, is equal to the weight of the fluid that the body displaces. This force is given by

$$B = \rho g V \tag{3.1}$$

where ρ is the density of the fluid, g the gravity and V the volume of the body immersed (or the volume of fluid that the body displaces).

3.1.1 Initial buoyant force

Table 3.1 shows the mass of the different components of the vehicle. They add up to 92.6 kg, which gives a total weight of

$$W = mg = 907.48\,\mathrm{N}. \tag{3.2}$$

On the other hand, the Guanay II AUV was modeled in the CAD software NX in order to compute its volume. The results showed that the total volume of the vehicle is

$$V = 0.075615995\,\mathrm{m}^3. \tag{3.3}$$

Assuming that the density of the seawater is 1025 kg/m^3, and applying equation (3.1), the buoyancy force on the AUV is

$$\begin{aligned} B &= \rho g V \\ &= 1025(9.8)(0.075615995) \\ &= 759.56\,\mathrm{N} \end{aligned} \tag{3.4}$$

As the weight W is bigger than the force B, the vehicle does not float and goes down.

Table 3.1: Vehicle masses in air

Vehicle part	Value	Units
External hull	27.1	kg
Main thruster	4.3	kg
Lateral thrusters	1.4	kg
Watertight module	59.8	kg

3.1.2 Calculation of the volume of foam

As the buoyancy force is not enough it is necessary the use of foams, which are much less dense than water. In this section we calculate the volume of foam V_f needed. We also take into account its density ρ_f.

When water goes into the cylinder the vehicle must dive, that is,

$$\begin{aligned} W &> B \\ (m + \rho_f V_f + \rho V_c)g &> \rho g(V + V_f) \\ m + \rho_f V_f + \rho V_c &> \rho(V + V_f) \\ V_f &< \frac{m - \rho V + \rho V_c}{\rho - \rho_f} \\ V_f &< \frac{16.6311}{1025 - \rho_f} \end{aligned}$$

where V_c is the volume of water that can take the cylinder (1.5 L). On the other hand when the cylinder expels the water the vehicle must float,

$$\begin{aligned} W &< B \\ (m + \rho_f V_f)g &< \rho g(V + V_f) \\ m + \rho_f V_f &< \rho(V + V_f) \\ V_f &> \frac{m - \rho V}{\rho - \rho_f} \\ V_f &> \frac{15.0936}{1025 - \rho_f} \end{aligned}$$

Hence

$$\frac{15.0936}{1025 - \rho_f} < V_f < \frac{16.6311}{1025 - \rho_f} \tag{3.5}$$

The inequality (3.5) indicates that the volume of foam must be between the two values in order to guarantee the principle of diving. The subtraction of these two values is approximately V_c, the volume of water that the cylinder can take.

Preliminary tests were made in a pool in order to confirm these calculations (see Figure 3.1). The volume of foam used was about 16 L (or 0.016 m^3). Taking into account that the density of the foam is about 20 kg/m^3, this volume are between the values of the inequality (3.5), thus the calculations coincides with the field tests. These tests also serve to align the center of gravity with the center of buoyancy so that the vehicle has no pitching. For this, 11 L must be at the stern and 5 L at the bow.

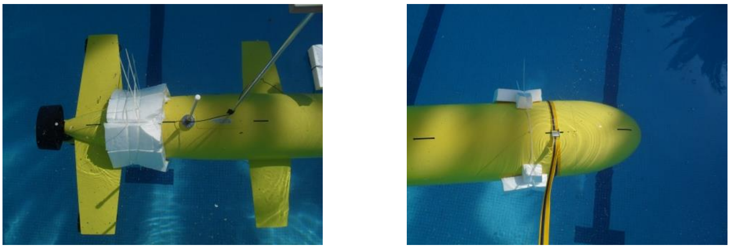


Figure 3.1: First foams used for the buoyancy

A material of polyurethane with high resilience, with commercial name H130, and with a density of $130\,\mathrm{kg/m^3}$ and operating depth up to $85\,\mathrm{m}$ was chosen for the foams. Using inequality 3.5 the volume of foam must be between $16.86\,\mathrm{L}$ and $18.58\,\mathrm{L}$. The design was made using SolidWorks software. The profile of each block of foam follows the Myring profile (see section 7.2.1) adapted to the bow, center and stern of the vehicle, with a volume of $1.11\,\mathrm{L}$, $4.58\,\mathrm{L}$ and $11.32\,\mathrm{L}$ respectively (see figures 3.2 - 3.6). The sum of all blocks is $17.01\,\mathrm{L}$. Taking this into account, the new mass of the vehicle is

$$\begin{aligned} m &= 92.6\,\mathrm{kg} + 130\,\mathrm{kg/m^3}(0.017\,\mathrm{m^3}) \\ &= 94.8\,\mathrm{kg} \end{aligned} \tag{3.6}$$

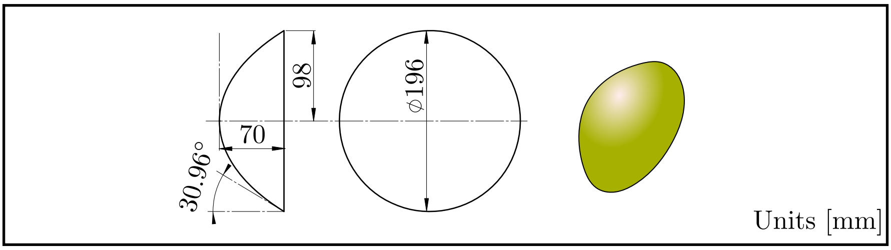


Figure 3.2: Foam A — Bow

3.1.3 Payload

In order to have a margin to put more weights without losing the buoyancy, another block of foam of $4.58\,\mathrm{L}$ (foam B - center) was designed for the center of the vehicle. The resulting payload can be used for more sensors or batteries.

Denoting by W_{pf} the weight of the pair payload-foam, and by B_{pf} the buoyancy force for this pair, the condition of equilibrium implies that

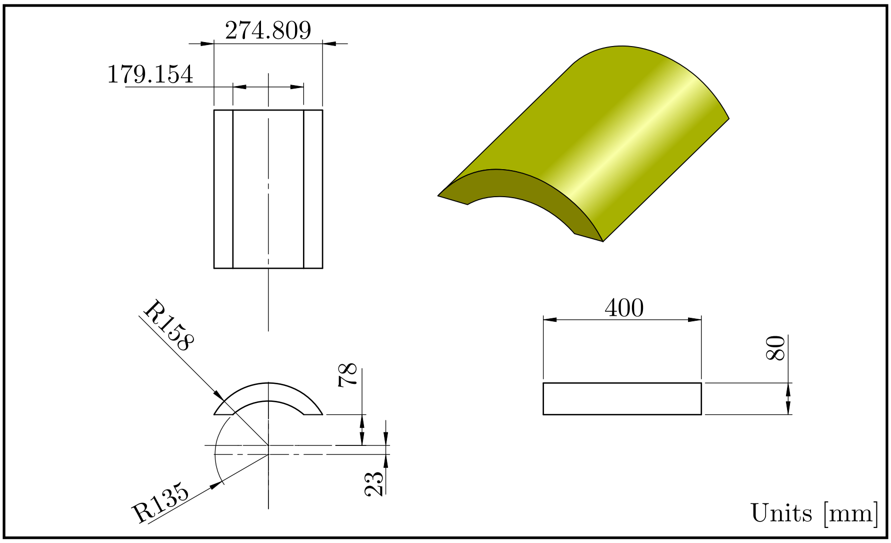


Figure 3.3: Foam B — Center

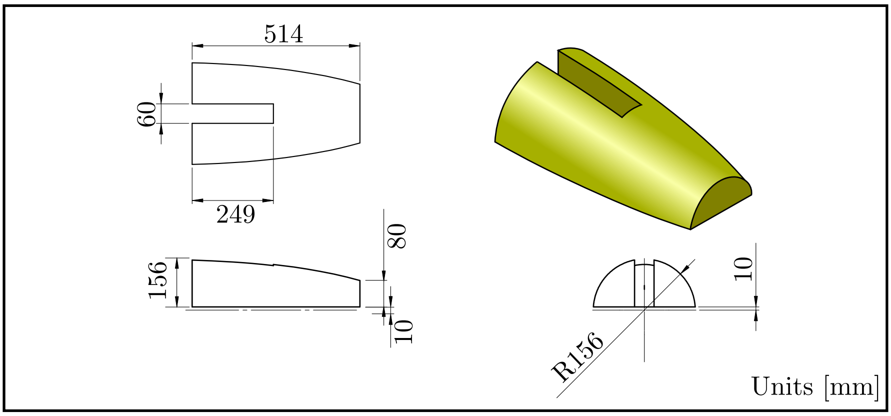


Figure 3.4: Foam C — Stern

$$
\begin{aligned}
W_{pf} &= B_{pf} \\
(m_p + m_f)g &= \rho g(V_p + V_f) \\
m_p - \rho V_p &= \rho V_f - m_f \\
\text{Payload} &= \rho V_f - m_f \\
\text{Payload} &= (\rho - \rho_f)V_f \\
&= (1025 - 130)(0.00458) \\
&= 4.1\,\text{kg} \qquad (3.7)
\end{aligned}
$$

Notice that the payload is not the weight in air but the weight in water.

3.2 Ballast system

The purpose of a ballast system is twofold. First, the possibility of adding a small weight in order to make small increases in the density of the vehicle and hence adjust the buoyancy. And second, to move this weight along the vehicle in order to adjust the pitch to zero, that is, to vertically align the center of gravity with the center of buoyancy.

We opted for an sliding lineal system, consisting of a guide of aluminum with an attached car that moves along it. The car has a screw to manually fix its position, and has a perforated platform to attach different weights. The model selected is a Drylin W by Igus [59] (see figure 3.5).

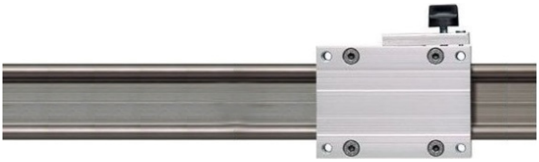


Figure 3.5: Drylin linear guide for the ballast system

The ballast system was situated at the bottom of the vehicle to lower the center of gravity, and also increase the stability in roll movement. Regarding the longitudinal position it is at the center of the vehicle in order to obtain a major rank to adjust the pitch inclination. The dimensions of the guide are 50 cm in length by 5 cm in width. Figure 3.6 shows the final disposition of the ballast system and the blocks of foam.

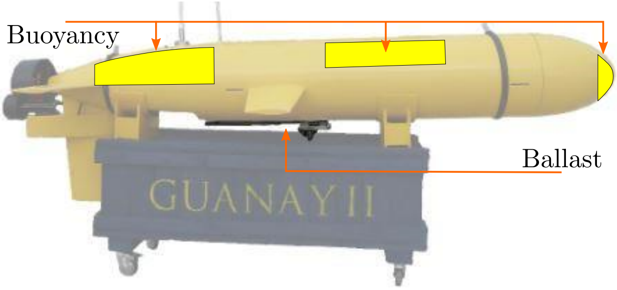


Figure 3.6: Ballast system

Chapter 4

Hardware

Guanay is a project conceived several years ago by the SARTI group of UPC, with the goal of having an autonomous underwater vehicle capable of performing different tasks. The evolution of the project passed first through a prototype using a basic hardware to monitor and control the vehicle [46] [153]. Next, another prototype was created, and this was called Guanay II. Its development continued with the improvements made by Masmitjà, first in [82] and continuing in [81].

The Guanay II AUV has in its interior a watertight module with all the electronics that communicate with the base station, sense the internal variables and control its actuators. They can be divided into 6 groups: Control unit, power, communication, navigation, actuators and safety module. Figure 4.1 shows its initial architecture.

The *control unit* consists of a PC104 embedded computer, model PM6100 [103]. Its processor is an AMD Geode LX800, 500 MHz and the data storage consists of a compact flash memory. The PC104 is equipped with an MSMX104+ expansion card [90] in order to have more serial ports. It has also been equipped with a DAS16JR/12 data acquisition card [98], which has 16 analog inputs (12 bits of resolution at 150 kbps) and 8 digital inputs in order to obtain data from analog and digital sensors. The operating system that has been used along the project is Windows XP and the programming is performed with NI-LabVIEW.

The *power system* of the vehicle is provided by a 2s3p NiCd battery pack by Saft-batteries [93], which gives a voltage of 24 V and has a capacity of 21 Ah. A DC/DC convertor from Mornsun series VRB [155] is used in order to supply voltages of 5 and 12 V to the different devices. Finally, a circuit to measure the state of the battery charge is used. This device uses a Coloumb counting control system using the BQ2013 (designed for NiCd batteries) in order to sense the incoming and outgoing energy from the batteries and is monitored through an RS232 port. This closes the group of the *power system*. A problem of this group is the starting switch, which consists of a connector used as an interrupter. Although it is an underwater connector (power series by Subconn [135]) it is in an area of difficult access, and hence the on/off procedure of the vehicle is a complex task in a real mission and impossible to manipulate if the vehicle is already in the water.

The *navigation system* consists of a GPS model Magallen DG14 [28]. To acquire the signals from the satellites, there is a 50 Ω passive WS3910 antenna [151] with a gain of 28 dB and a noise figure of 0.8 dB. The navigation is complemented by a digital compass/inclinometer model TCM2.6 [72]. It has a tilt range of ±80° and accuracy of 0.8° in tilt and 0.1° in compass. This device also has a sensor to measure the temperature. The *navigation system* lacks a sensor to measure the displacement when it is diving since the GPS only has coverage of satellites on the surface.

The *actuators system* is composed by the electronics to control the three thrusters and the cylinder for the immersion. The main thruster, a Seaeye SI-MCT01B [141], has its own driver and it is controlled from the unit control with a RS485 serial communication. On the other hand, both lateral thrusters (DC motors model BTD150 by Seabotix [142]) and cylinder (CRDNG series by Festo [132]) need separate boards to control them. One of them is a SSC32 driver of Lynxmotion [131], which can manage several servo signals. One signal, to control the cylinder, goes to an H-brigde drive to provide bidirectionality to the motor of the cylinder, needed to perform diving operations. The motor is a linear actuator model ATL-08 by Servomech [74]. The magnetic sensors, model CRSMEO-4 by Festo [106], are the responsible to detect the ends of the cylinder and stop its movement. One minor problem with this configuration is that these sensors are not feedback to the PC104, therefore the PC needs to wait long times to guarantee the movement to the ends of the cylinder. On the other hand, this servo driver gives other two signals that go to PWM drivers in order to control the lateral thrusters. One problem with this driver is that it is unidirectional, so that the lateral thrusters can not go in reverse. This has been a circumstantial problem because the maneuverability using only one lateral thruster is reduced, and it is necessary to complement the turning with the main thruster when going backwards.

The *safety system* permits monitoring variables which can become critical to the overall operation. They are: a *state of battery charge* device, described above; a *pressure sensor* model DS2806 [122] with a range of 0-10 bar and a sensitivity of 0.5 V/bar which allows to monitor the dive up to 100 m; a *humidity sensor* model HIH4000 [57] and a range of 0-100 %RH which allows to monitor the internal humidity; and a *temperature sensor* integrated with the compass (described above), to monitor the internal temperature. Some of these

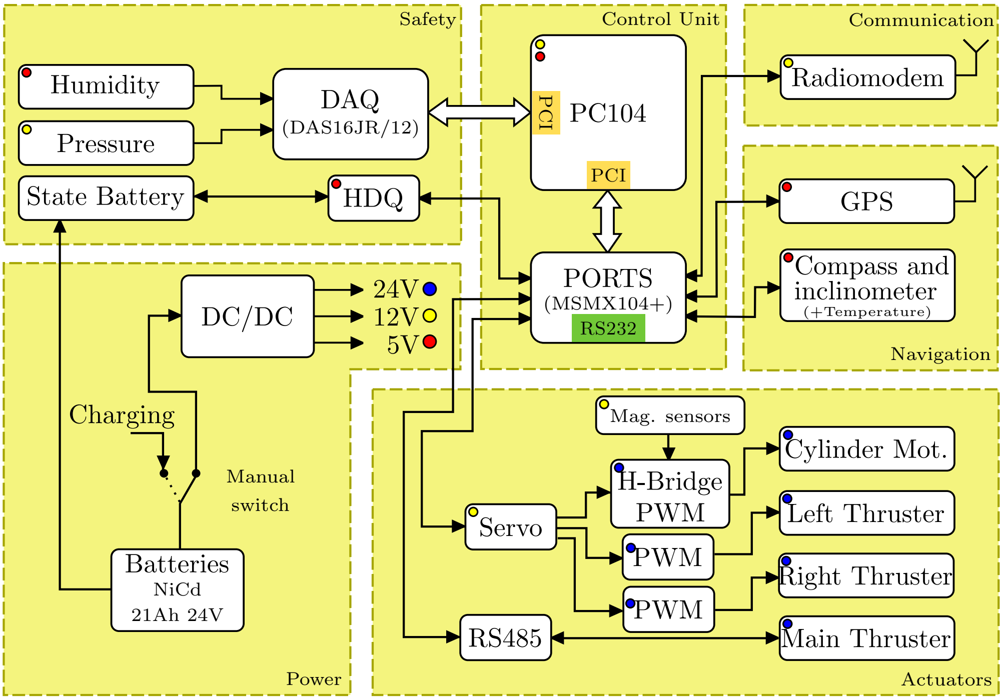


Figure 4.1: Initial hardware architecture of Guanay II AUV

variables are sensed through the DAQ which converts the analog signals to digital inputs for the PC104. Regularly, these sensors need a calibration; one of the most important is the depth sensor, which is very linked with the principle of movement of the vehicle.

The *communication system* is composed of a radio link with the base station. The device installed is a radio-modem TMOD-C48 by Farrell [146]. It works in a frequency range of 403-470 MHz (UHF) and a speed of 4800 bps.

This hardware has been useful for the operation of the Guanay II AUV, but has several problems that can be improved. In this chapter we describe the modifications and improvements made in the vehicle. First in the *actuators system* a driver is introduced in order to allow the bidirectionality in the lateral thrusters. Next, several changes in the *power system* were made in order to manipulate the starting of the vehicle, and to add a circuit to measure the power consumption. Next, some modifications in the immersion system were made, namely the calibration of the pressure sensor and the use of an air chamber to reduce the consumption. Additionally, some commercial devices were installed, like an inertial measurement unit in order to calculate the position underwater, a WiFi connection for a better operability and programming, a CTD and an acoustic localization system.

4.1 Lateral thruster driver

The Guanay II AUV has two lateral thrusters which are controlled by a driver which gives power in an unidirectional way. Tests in water have shown that radius of curvature using one lateral thruster is about 10 m, and it is necessary to use the main thruster in backwards in order to achieve a smaller radius.

On the other hand, this driver is commanded by a servo controller. The servo signals consists of a PWM of 20 ms of period and a pulse width from 0.5 ms to 2.5 ms. This means that the duty cycle is limited to the range 2.5-12.5 % and the signal needed another block to adjust it to a full range of duty cycle. In conclusion, it was advisable to dispense with this board.

Taking into account this two considerations, we have developed a new scheme which consists of a PIC microcontroller and a driver with H-bridge (see figure 4.2). The PIC can communicate with the PC104 using RS232 commands and has the particularity that gives directly a full range PWM signal. This device also has several digital pins in order to manage variables of errors. The driver with H-bridge receives the PWM signal, and using a power section amplifies it and assigns the polarity to the lateral thrusters.

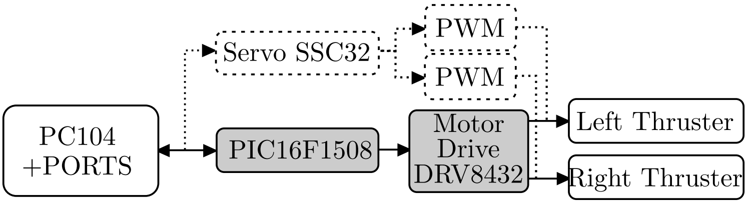


Figure 4.2: Hardware changes to control the lateral thrusters

4.1.1 DRV8432 motor drive

An H-bridge is a circuit that can control the current direction of a motor through four interrupters (generally MOSFET). The figure 4.3 shows the general operation of an H-bridge:

when the switches S1 and S4 are closed (and S2 and S3 are open) a positive voltage will be applied across the motor. By opening S1 and S4 and closing S2 and S3 switches, this voltage is reversed, allowing reverse operation of the motor.

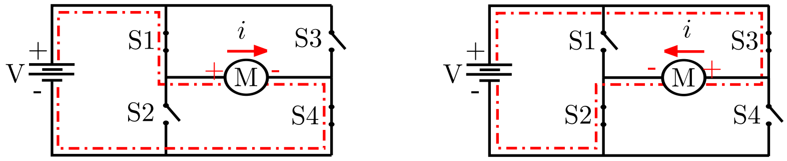


Figure 4.3: General operation of an H-bridge

The DRV8432 (by Texas Instruments [33]) is a chip that has two DC motor drivers with two H-bridges and an advanced protection system. This protection system includes the detection of overcurrent, short circuit, low voltage and two thermic systems. Its efficiency is about 97 %.

Regarding the H-bridge configuration, it allows 4 modes which are selected through `M1`, `M2` and `M3` pins:

- (`M3`,`M2`,`M1`)=(0,0,0). Dual full bridges (two PWM inputs in each full bridge) or four half bridges with cycle-by-cycle current limit.

- (`M3`,`M2`,`M1`)=(0,0,1). Dual full bridges (two PWM inputs in each full bridge) or four half bridges with OC latching shutdown.

- (`M3`,`M2`,`M1`)=(0,1,0). Parallel full bridges with cycle-by-cycle current limit.

- (`M3`,`M2`,`M1`)=(0,1,1). Dual full bridges (one PWM input each full bridge with complementary PWM on second half bridge) with cycle-by-cycle current limit.

From them we have selected the fourth option because we need a unique PWM and a full bridge for each thruster. At the same time it is convenient to use the cycle-by-cycle current limit as a protection. The manufacturer presents a series of resistances in order to configure this current, and in our case we have connected a resistance of 68 kΩ to the pin `OC_ADJ` for a maximum current of 4.1 A (current recommended by the manufacturer of the lateral thrusters).

Regarding the voltages, the chip has been powered by 12 V (pin `GVDD`) for its internal regulator and gates, and 24 V for the power of the H-bridges (pins `PVDD`).

Figure 4.4 presents the design made with this chip. Following its labels, one thruster is connected to `OUTA` and `OUTB`, and the other thruster to `OUTC` and `OUTD`. The signal inputs are `PWM_A` and `PWM_B`. Additionally, the circuit includes the resets `RES_AB` and `RES_CD` in order to disable the H-bridges. Finally, the signals `OTW` and `FAULT` are signals of errors of the chip which go to the ports and have their respective LED indicators.

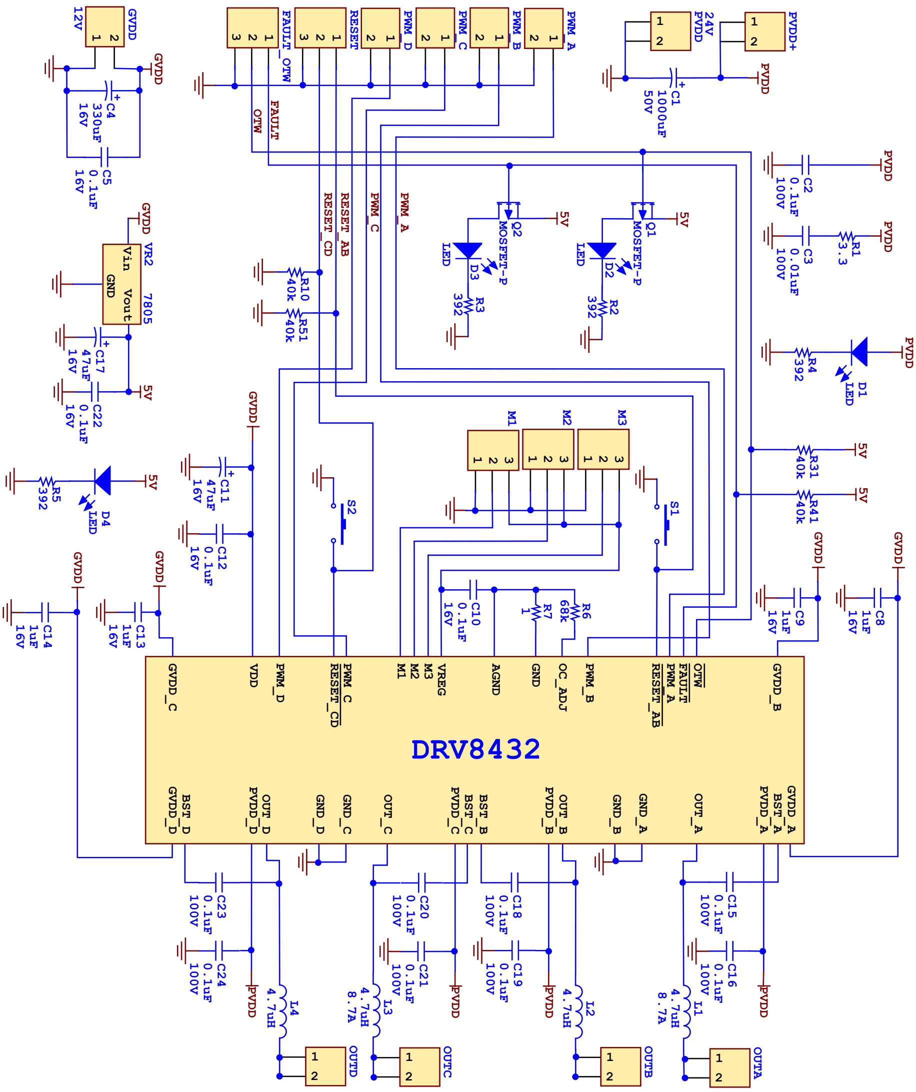


Figure 4.4: DRV8432 schematics

4.1.2 PIC16F1508

The microcontroller is the responsible to communicate with the PC104 through RS232 signals. At the same time it has to give two PWM signals to control the two thrusters using the DRV8432 described above. The PIC16F1508 by Microchip [102] is a microcontroller that can be used for this purpose. It has 20 pins which can be allocated for different purposes like digital I/O (18), A/D (10), PWM (4), UART (1), among others.

The PWM1 and PWM2 can be used to control the lateral thrusters. The PWM3 and PWM4 are reserved for future demands, such as the control of the cylinder and thus dispensing the servo controller. On the other hand, in order to establish a communication with the PC104, a MAX232 device is used to transform the UART signal of the PIC to the RS232 protocol. Figure 4.5 shows the circuit made for this stage.

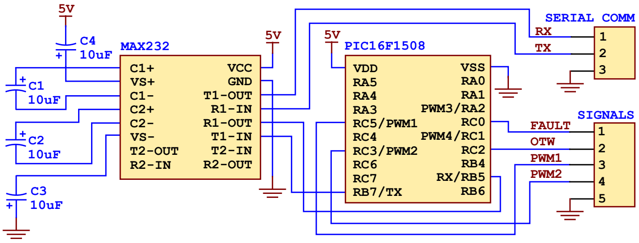


Figure 4.5: PIC16F1508 schematic

Two commands have been programmed in the PIC as a control protocol:

- "M1,*perc*". To set the left thruster by setting the PWM1. If there is no error the PIC will respond "OK".
- "M2,*perc*". To set the right thruster by setting the PWM2. If there is no error the PIC will respond "OK".

The argument *perc* is a value between -100 and 100, which corresponds with the maximum power in backwards (-24 V or 0 % of duty cycle) and the maximum power in forward (24 V or 100 % of duty cycle) respectively. In this order, the relation to calculate the duty cycle is defined by

$$D = \frac{100 + perc}{2}. \tag{4.1}$$

The code implemented can be found in appendix A.

4.1.3 Laboratory tests

Several laboratory tests have been made in order to check the correct operation of the board. Figure 4.6 shows different duty cycles for a PWM with a frequency of 50 kHz. The range of voltages are from -24 to 24 V. It also shows the average voltage. When the duty cycle is greater than 50 % the average voltage is positive, but when it is smaller the average voltage is negative.

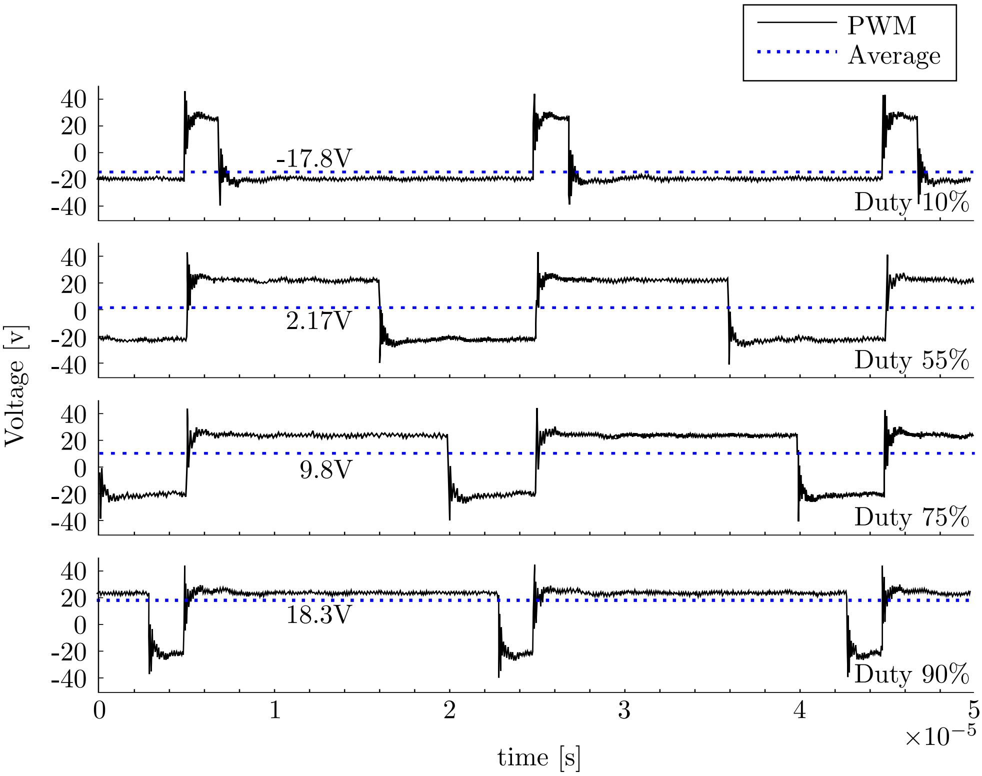


Figure 4.6: PWM signals with different duty cycles for the lateral thrusters

4.2 Power system

The power system of the Guanay II AUV can be improved in some aspects. In this section we describe a RF remote control on-off to facilitate the starting, and a sensor to monitor the power consumption by the actuators.

4.2.1 RF remote control on-off

One of the problems with the hardware architecture of the Guanay II AUV is the switch to start the vehicle. As can be seen in Figure 4.7 it consists of a HPDC4M interrupter by Subconn [135] that makes a bridge between the batteries and the electronics. If an operator needs to do a reset to all the system, the vehicle has to be pulled from the water, the bottom lid as to pe opened(in a reduced space), and the interrupter has to be disconnected/reconnected manually.

A RF remote control on-off has been implemented in order to solve this problem. It is a HIR6-433 device by RF-solutions [54], which consists of an encoder and a AM receptor. This device activates a relay that connects the batteries with the rest of electronics. Figure 4.8 shows the circuit made for this module. The circuit with its antenna has been located out of the watertight module, specifically in the main antenna of the vehicle, and it is wired through a Subconn connector.

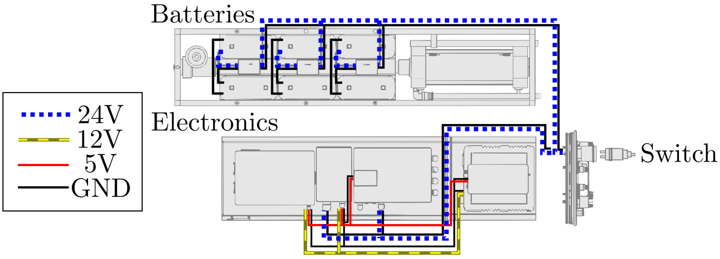


Figure 4.7: Original switch to power the electronics manually

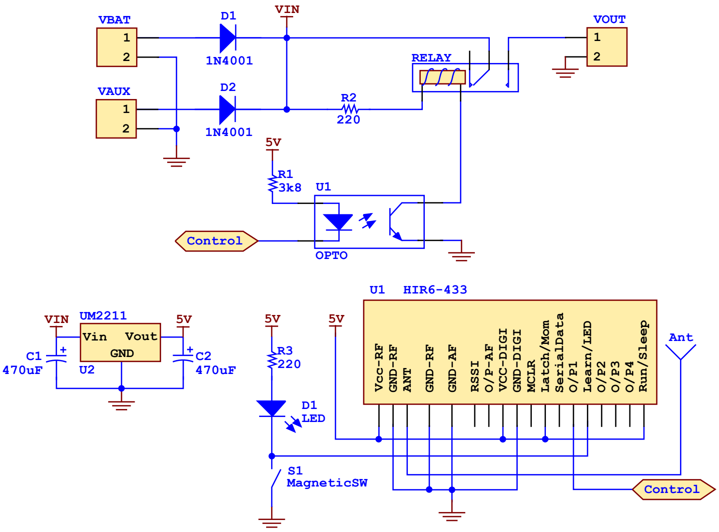


Figure 4.8: RF on-off schematic

4.2.2 Power consumption sensor

The Guanay II is a vehicle in development phase. For this reason, we have considered appropriate to add a power consumption sensor to monitor the consumption of the actuators of the vehicle. It is divided in two parts: First, a resistance of 0.01 Ω in series, whose small voltage is directly proportional to the current, so it can be used to measure the current; and second, a voltage divisor in order to measure the voltage. These two sensors are at the beginning of the power lines in the actuators box, so it measures the voltage and current of all the actuators together. Finally, the voltages of these signals are sensed by the DAQ board.

4.3 Immersion system

The immersion system of the Guanay II AUV is a important part for its navigation. We present first a calibration of the pressure sensor. Second, an study of the use of an air chamber is presented in order to reduce the energy consumption when the vehicle is emerging. And finally, the maximum operating depth of the vehicle is discussed.

4.3.1 Calibration of pressure sensor

The pressure sensor is powered by 12 V and following the indications of the manufacturer it has a sensitivity of 0.5 V/bar. Several tests have shown that this sensor needs a calibration in order to get good data. For this, we have connected the pressure sensor to a compressor where can be set to different pressures. Figure 4.9 shows a relation between the pressure and the voltage of the sensor.

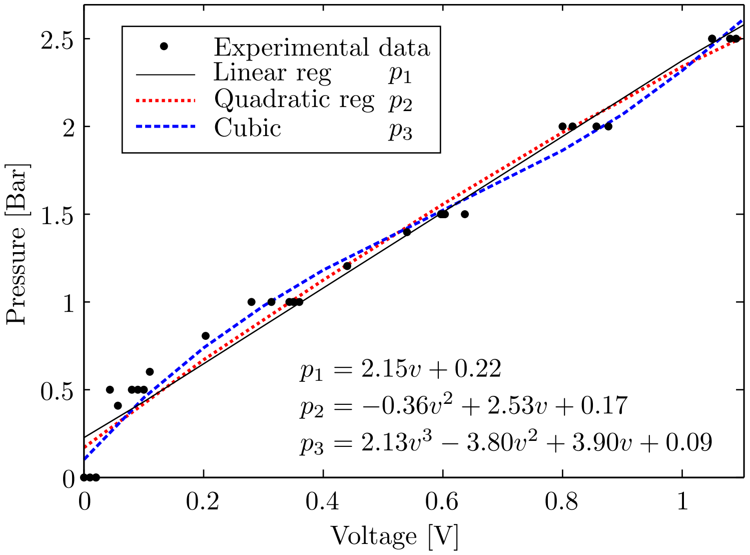


Figure 4.9: Relation of the pressure and voltage in the pressure sensor, and three polynomial regressions.

It also shows three different polynomial regressions. The first one is a linear regression, represented by a green line. It coincides with several values but has an offset of 0.22 bar, which is very different from reality, *i.e.* for 0 V the sensed pressure must be 0 bar.

The quadratic regression (displayed in red) also coincides at several points and has an offset of 0.17 bar. On the other hand, the cubic regression (represented by a cyan line) has an undesired oscillation but has a smaller offset. Although these two regressions have good performance the manufacturer mention that its regression must be linear, that is, for values greater than 2.5 bar the above regressions will not coincide with the reality.

It is important to note that the linearity can be seen clearly but it has a drastic change in values near to 0 V. Figure 4.10 shows another regression composed by two lines: One for voltages less than 0.08 V and the other for values greater than 0.08 V. This regression shows a better performance compared with the above regressions. Table 4.1 shows the root mean squared deviation of each regression respect to the experimental data. From them the segmented regression presents the smallest deviation.

4.3.2 Study of the use of a chamber

The vehicle uses the cylinder in the process of diving. It is a double actuated cylinder, model is CRDNG-100-200-PPV-A by Festo [132] (100 mm of diameter and 200 mm of length). One port is connected to the seawater in order to fill the cylinder, and the other port connects

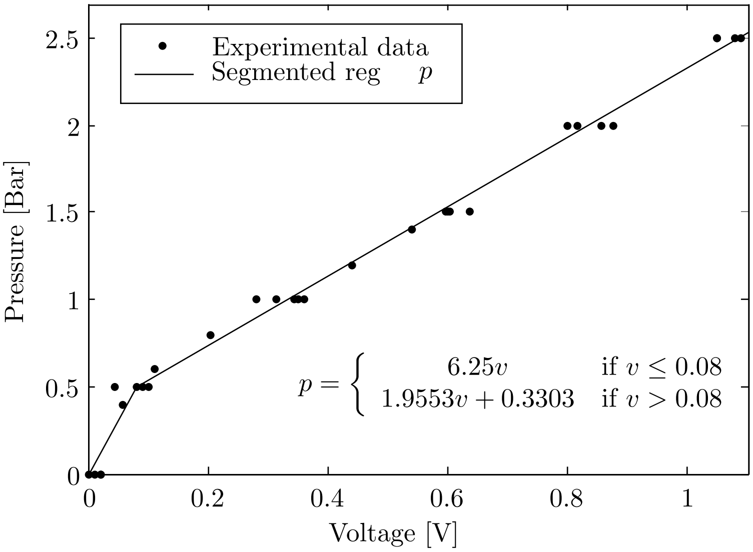


Figure 4.10: Segmented regression for the calibration of the pressure sensor

Table 4.1: Root mean squared deviation of the regressions of the pressure sensor

Regression	RMSD	Units
Linear	0.1189	bar
Quadratic	0.1122	bar
Cubic	0.0948	bar
Segmented	0.0699	bar

with the interior of the watertight module, as can be seen in figure 4.11. When the cylinder is full of water, the air pressure in the interior of the watertight module is slightly above atmospheric pressure.

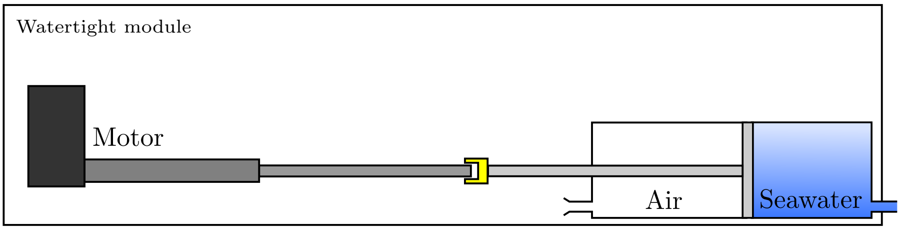


Figure 4.11: Immersion system

The consumption of energy is related to two cases: Going down and going up. When the vehicle is in the surface, the cylinder is subjected to the atmospheric pressure present in the seawater and the interior of the watertight module, and it uses little power in the immersion. On the other hand, when the vehicle is underwater, the cylinder is subjected to high pressure present due to the seawater column and the atmospheric pressure present in the watertight module. In this order, the cylinder uses more power consumption when the vehicle goes up. At higher depths there is a risk of shipwreck if the batteries have little power. Additionally, one has to take into account the features of the motor of the piston, a lineal motor ATL-08 by Servomech [74], whose maximum current is 9 A.

One solution is the use of an air chamber instead of connecting the cylinder with the interior of the watertight module. The idea is accumulate energy in the immersion (compress the air and concentrate pressure) in order to use it when the vehicle goes up. Although the batteries will use more energy in the immersion, they will use much less going up. This air chamber is a PVC tube of 2.66 cm of diameter and 65 cm of length.

The volume of the compressed air $V(x)$ consists of the sum of the air chamber V_{ch} and the cylinder $V_{cyl}(x)$, which is in function of the position of the piston x,

$$\begin{aligned} V(x) &= V_{ch} + V_{cyl}(x) \\ &= V_{ch} + \pi r^2 x \end{aligned} \tag{4.2}$$

where r is the radius of the cylinder.

Denoting by L the total movement of the piston, the air pressure can be calculated using Boyle's law:

$$\begin{aligned} P_a(x) &= P_e \frac{V(L)}{V(x)} \\ &= P_0 \frac{V_{ch} + \pi r^2 L}{V_{ch} + \pi r^2 x} \end{aligned} \tag{4.3}$$

where P_0 is the pressure when the cylinder is full of air and coincides with the atmospheric pressure (1 bar). The accumulated pressure can go from 1 to 5.3 bar.

When the piston is moving the sum of forces must be equal to zero in order obtain a constant velocity v. Denoting by F_a the force of the air chamber, F_w the force of the water and F_m the force of the piston (see figure 4.12), we obtain that

$$\begin{aligned} F_m + F_a - F_w &= 0 \\ P_m A + P_a A - P_w A &= 0 \\ P_m &= P_w - P_a. \end{aligned} \tag{4.4}$$

Notice that as we apply these forces to the same area A, this relation can be seen as a sum of pressures. In order to eject the water, the piston pressure P_m must be a little bigger than $P_w - P_a$, but when the cylinder takes the water in it must be a little lower than $P_w - P_a$.

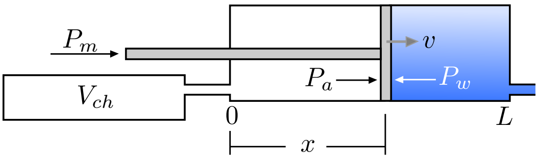


Figure 4.12: Pressures acting on the piston

This piston pressure is directly related to the direction of movement and the current given to the engine. This current must be the same whether the pressure to overcome is in one direction or the other. For example, the current used to move the piston forward when there are 2 bar in the air chamber and 3 bar in the sea, is the same current used to move the plunger in backwards when there are 3 bar in the air chamber and 2 bar in the sea. We define p as the pressure to overcome with respect to the direction of movement:

$$p = dP_m \tag{4.5}$$

where d is the direction of movement (that is $+1$ to empty or -1 to fill). Therefore, the current i is a function of this pressure:

$$\begin{aligned} i &= f(p) \\ i &= f(dP_m) \end{aligned} \tag{4.6}$$

In order to determine this relation several tests were made. They are separated into two groups:

1. *Without the use of the chamber*, as is observed in figure 4.11. The air is compressed in the watertight module and the pressure is approximately the atmospheric pressure (1 bar). Since the watertight module is large pressure changes very little.

2. *Using the air chamber*, as is observed in figure 4.12. Where the air is compressed from 1 to 5.3 bar.

The signals of the magnetic sensors have been adapted to the DAQ board in order to monitor the final position of the piston. The watertight module and the motor-cylinder ensemble have been placed inside of an hyperbaric chamber, which is an infrastructure to test the performance at different pressures of water. Figure 4.13 shows the infrastructure used.

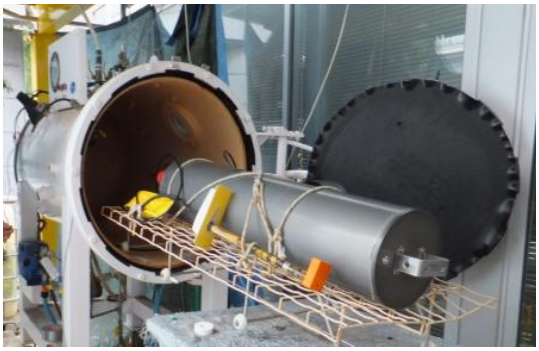


Figure 4.13: Watertight module and hyperbaric chamber

Table 4.2 summarizes the current and the time to move the total plunger stroke in the case of not using the air chamber, where the pressure changes are minimal and the current is almost constant.

The relation of the current to the pressure is shown in Figure 4.14. Each point corresponds to a specific condition of the water pressure P_w, the direction of movement d and the position of the piston x (empty cylinder of full cylinder). The pressure p combines these conditions using the equations (4.3) - (4.5). This figure also shows that for a pressure of 4 bar a current of 8.5 A is needed, which is near to the maximum current allowed by the motor.

Table 4.2: Cylinder currents and times without the use of an air chamber

d	P_w [bar]	i [A]	t [s]
-1	1	1.25	10.38
-1	2	0.5	10.5
-1	3	0.5	10.31
-1	4	0.5	9.2
1	1	1.25	11.15
1	2	2.2	12.25
1	3	4.0	14.65
1	4	6.0	18.75

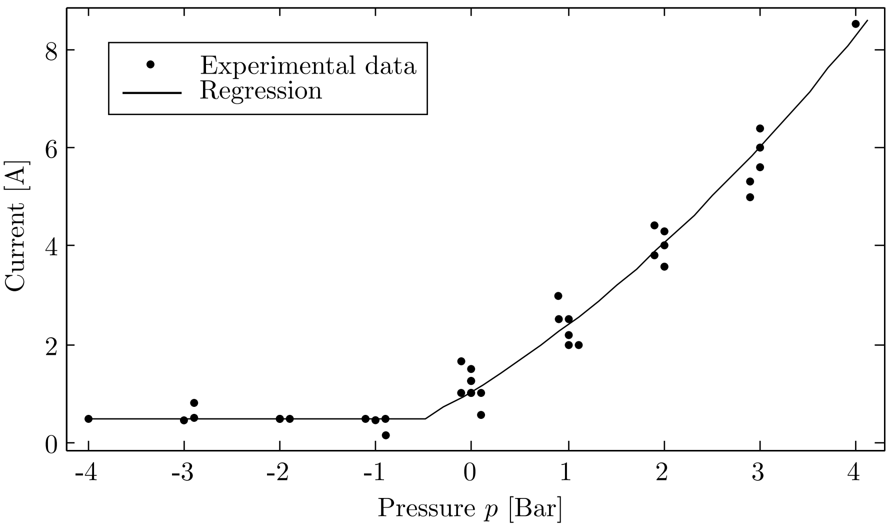


Figure 4.14: Current of the cylinder as a function of the pressure p

It is important to note that when p is negative the current tends to 0.5 A, which is the minimum current necessary to move the plunger. For instance, when the movement ejects the water ($d = 1$) and the air pressure is bigger than the water pressure ($P_w - P_a < 0$), the current used is minimum since the air pressure helps to move the plunger. The same result is achieved when the direction of movement is such that water ($d = -1$) goes in and the water pressure is bigger than the air pressure ($P_w - P_a > 0$). The regression analysis of the data has been separated into two segments:

$$i = \begin{cases} 0.1566p^2 + 1.1814p + 1.0203 & \text{if } p \geq -0.47 \\ 0.5 & \text{if } p < -0.47 \end{cases} \tag{4.7}$$

In the same way, using the total plunger stroke and the time of travel, the relation of the plunger velocity v and the pressure p has been calculated as

$$v = \begin{cases} -0.0002p^2 - 0.0017p + 0.0181 & \text{if } p \geq -1.4 \\ 0.02 & \text{if } p < -1.4 \end{cases} \tag{4.8}$$

Notice that both current and velocity are described with respect to the pressure p. On the other hand, this pressure is in function of the water pressure P_w, direction of movement d and position of the plunger x. In this way, it is possible to simulate the current i in time by setting conditions of water pressure and direction of movement.

Figure 4.15 shows the actual tests and simulations of the filling process for different conditions. When the water pressure is at the sea surface (1 bar) the cylinder needs 1 A of current without the air chamber, but needs a growing current from 1 to 5 A using the air chamber since the compression of the air impedes the movement. In this case the air chamber earns potential energy.

Following with the tests, when the water pressure corresponds to 10 m of deep (2 bar) the cylinder needs 0.5 A of current without the air chamber. This is because the seawater helps the movement. If the air chamber is used, the current starts at 0.5 A, but as the air is compressed the current grows up to 4 A.

In a similar way, when the water pressure is that of 20 m of depth (3 bar) the current is 0.5 A without the air chamber. If the air chamber is used the current starts at 0.5 A during a prolonged time, and grows when the air pressure overcomes the water pressure.

The analysis at 4 and 5 bar is similar to the above. The difference is that the cylinder needs less and less power to fill it due to the higher water pressure.

Figure 4.16 shows the actual tests and simulations of the emptying process for different

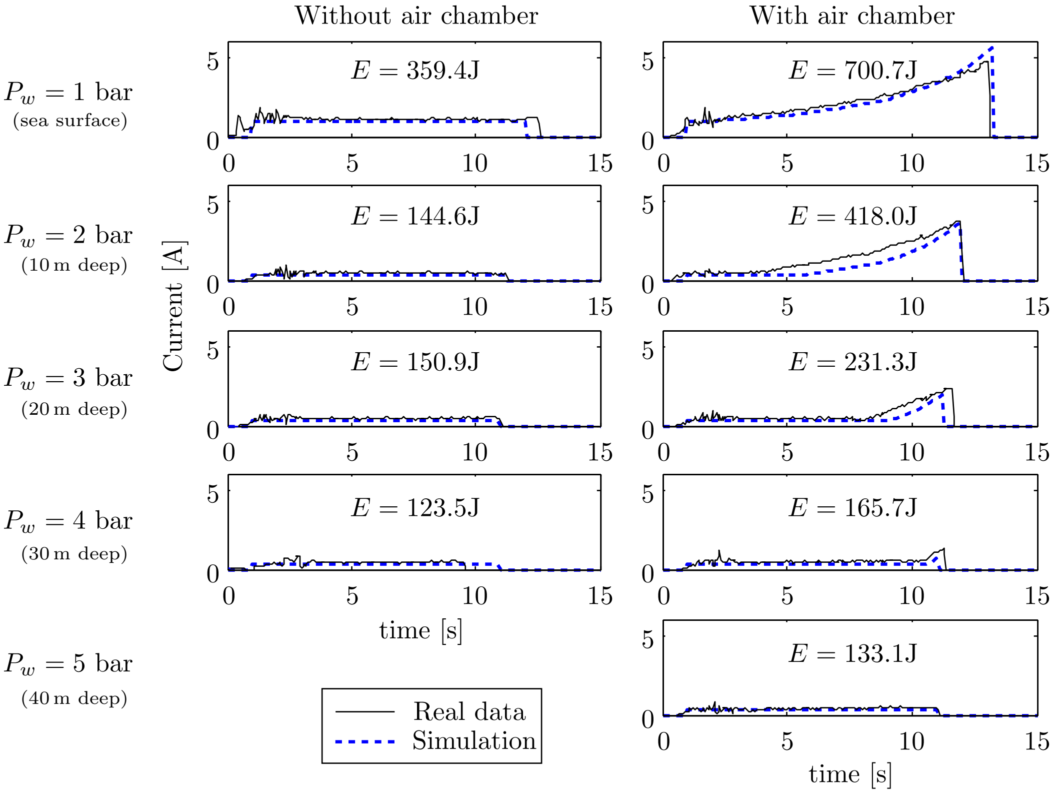


Figure 4.15: Current of the cylinder in the filling process at different conditions

conditions. In all cases where the air chamber is used, the initial condition is 5.3 bar of compressed air, which is a potential energy that will help to eject the water. When the water pressure is that of the sea surface (1 bar) the cylinder needs 1 A of current without the air chamber. If the air chamber is used the current starts at 0.5 A during a prolonged time, and grows when the air pressure decreases near to the atmospheric pressure.

In a similar way, the other cases of 10, 20 and 30 m of depth show a constant current without the use of the air chamber, and a small current that increases in time when the air chamber is used. At the same time, less time is needed to empty the water when using the air chamber.

These results satisfy the initial objective of reducing the current when the vehicle is underwater and is going to emerge. Figures 4.15 and 4.16 also show the energy used in each case to perform the action, which is given by

$$
\begin{aligned}
E &= \int_0^t \text{Power}\, dt \\
E &= 24 \int_0^t i\, dt \qquad (4.9)
\end{aligned}
$$

The equation calculates the energy based on the power used by the cylinder. The power consists of the product of the voltage and current, with a constant voltage of value 24 V, and

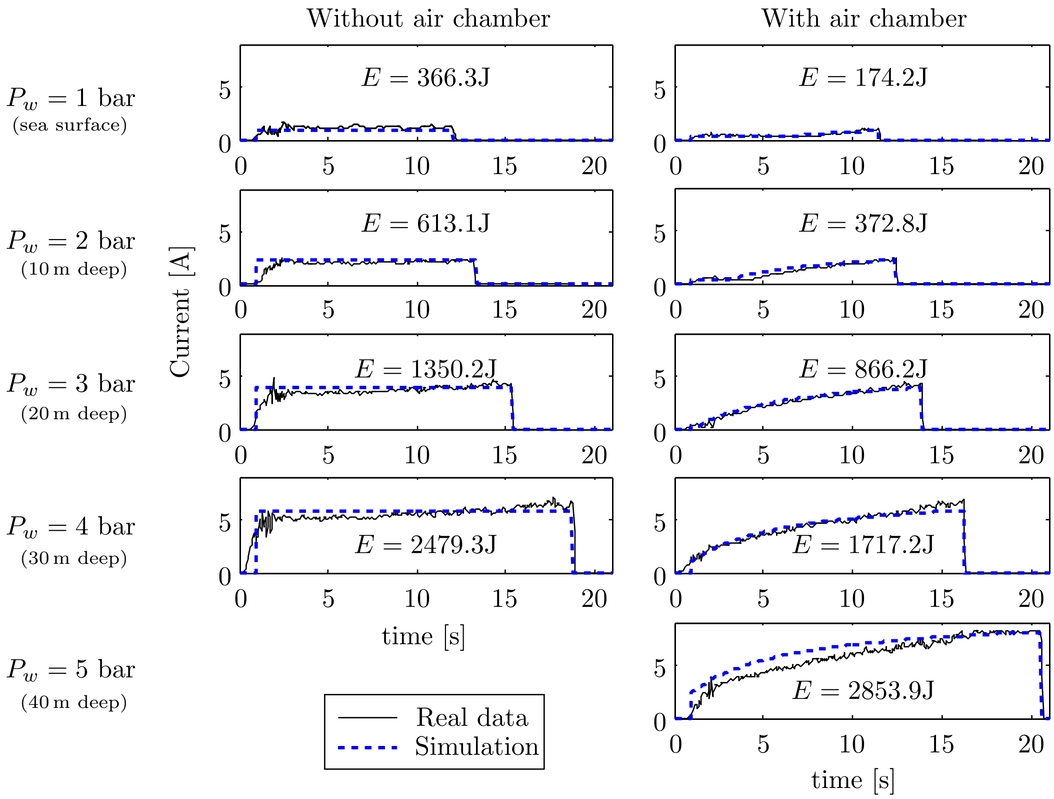


Figure 4.16: Current of the cylinder in the emptying process at different conditions

a current which depends on time.

The energy saving δ is calculated as the reduction of energy to empty the cylinder when the air chamber is used:

$$\delta = \frac{E_n - E_{ch}}{E_n} \times 100 \tag{4.10}$$

where E_n is the energy used without the air chamber and E_{ch} the energy with the air chamber. According to the above results, the energy savings are about 39.2%, 35.8% and 30.7% when the vehicle is at depths of 10, 20 and 30 m, respectively.

4.3.3 Maximum operating depth

As said, the Guanay II AUV was conceived to operate at shallow waters, and the maximum operating depth depends on several limitations. The first of them is the operating depth of the thrusters. The main thruster, being a thruster used for big ROVs, can be submerged up to 1000 m; instead the lateral thrusters, used for small ROVs, can be submerged up to 150 m. The second limitation is the tightness of the Guanay II. The module that contains all the electronics is made of aluminum, and the manufacturer estimates that its o-rings can support pressures up to 300 m. The next limitation is regarding the buoyancy. The foams have a safety factor for crush pressure, which corresponds to 85 m of depth. The final limitation is the force of the motor of the cylinder to eject water; in the last section we have shown that this limit is about 40 m of depth, which corresponds to the maximum motor current (9 A).

Table 4.3 resumes the maximum operating depths by component. As the cylinder motor is the most restrictive limitation, the maximum operating depth of the Guanay II is 40 m.

Table 4.3: Maximum operating depth by component

Component	Value	Units
Main thruster	1000	m
Lateral thrusters	150	m
Watertight module	300	m
Foams	85	m
Cylinder motor	40	m

4.4 Adapting commercial devices

4.4.1 Inertial Measurement Unit

An inertial measurement unit (IMU) is a device that has 3-axis acceleration and 3-axis gyroscope sensing. It is used to estimate the position and attitude of the vehicle as an alternative to GPS when the vehicle is underwater. The device used is a IG-500A-G3A1P1-B unit by SBG Systems [58] (see figure 4.17), which has a measurement range of ±2 g in the accelerometers and ±300 °/s in the gyroscopes.

The position can be calculated using integrals over the time from the acceleration and gyroscope data. This is called inertial navigation, the error in the position will increase over the time and has to be actualized by an absolute system, like a GPS. However, further tests in the Guanay II AUV have shown that the noise present in the device greatly affects the

Figure 4.17: Inertial measurement unit IG-500A

estimation of the position, even using a Kalman filter in conjunction with the GPS, and the results were not good at all. This is what we have preliminarily detected. We are budgeting the inclusion of a DVL, a device to measures the velocity underwater. Section 5.1.2 deals with the incorporation of the IMU in the vehicle software.

4.4.2 WiFi connection

Normally, the Guanay II is opened to modify the software by connecting a monitor, keyboard and mouse to the PC104. This process is time consuming and the manipulation of the vehicle must be as easy as possible. A similar situation is presented when it is necessary to access the stored data of the current session. To solve this, a wireless local area network (WLAN) was created using the Guanay II as one of its nodes. This wireless connection allows to remotely access not only the stored data but also to control the entire operating system. Additionally, it can be accessed from different devices simultaneously, for instance, from a laptop, a tablet and a smartphone.

For the network we have used a router by Tp-link, model TL-MR3220 [117]. It needs a power supply of 12 V and its speed is up to 150 Mbps for a robust Wi-Fi network. The router is located at the base station and has a range of about 25 m.

On the vehicle side, we have put a wireless USB adapter, model TL-WN727N by Tp-link [158] (see figure 4.18). Again, its speed can be up to 150 Mbps, so the remote access is fast. This adapter is connected to one of the USB ports of the PC104, and it small size makes it ideal for this type of application.

Figure 4.18: Wireless USB adapter TL-WN727N

In order to have the maximum range as possible, we have opted to locate it out of the watertight module, specifically in the main antenna of the vehicle (where the GPS and RF remote control antennas are). Figure 4.19 shows the device after placing it and added a resin to provide the necessary tightness.

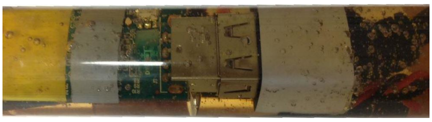


Figure 4.19: Detail of the wireless USB with resin to guarantee the tightness

4.4.3 CTD

A CTD or Sonde is an oceanography instrument used to determine the conductivity, temperature, and depth of the ocean, which are the most important variables sensed in the sea. The principle of movement of the GUANAY II AUV was adopted to measure water columns, and hence replace conventional monitoring systems from ships.

The model used is a XR-420 from RBR company [45] (see Figure 4.20). The range and precision of the sensors are: for conductivity (mS/cm) accuracy of ±0.003 and range of 0-85; for temperature (°C) accuracy of ±0.002 and range of -5 to 35; for pressure (m) accuracy of ±0.025 and range of 0-50.

Figure 4.20: CTD XR-420

The CTD can be connected to the control unit by RS232, or can work autonomously. It has been located at the bottom of the vehicle, linked to the rail of the ballast system.

4.4.4 Acoustic localization system

Several missions with the GUANAY II in the sea have shown the importance of a system to localize the vehicle when it is underwater. One solution has been implemented using an acoustic localization system, the Mantrak kit by Sonotronics [77] (see figure 4.21). It consists in a small piece of 10 cm that transmits acoustic pulses that can be heard up to 3 km. It is called pinger (model EMT-01-2), and it is located in the vehicle. The base station has a receptor (model USR-08) and a directional hydrophone (model DH-4) to heard the signals. The autonomy of the pinger is 18 months.

Figure 4.21: Mantrak kit. Acoustic locatizacion system

4.5 Conclusions

The implemented changes have improved the performing of the vehicle. Figure 4.22 shows the new hardware architecture of the Guanay II AUV. The improvements are summarized as follows:

- *Actuators system.* A new motor driver was designed and implemented for the lateral thrusters. The improvements lie in the possibility of going backwards and in the safety of current consumption. The driver consists in a PIC16F1508 microcontroller to establish the communication with the PC104, and a DRV8432 motor drive to control the thrusters. On the other hand, a complete study of the immersion system was made taking into account the addition of an air chamber to reduce power consumption. Finally, we discussed the maximum operating depth of the vehicle.

- *Power system.* A new on/off system was designed using a remote RF control. It allows a better manipulation without having to open the watertight module. Additionally, a power consumption sensor was added to monitor the consumption of the thrusters and cylinder.

- *Safety system.* The voltages of the magnetic sensors now are sensed by the PC104 in order to have a feedback with respect to the position of the piston.

- *Commercial devices.* The vehicle now has an inertial measurement unit in order to improve the navigation system. Concerning the mission, the vehicle has a CTD data logger to record the salinity, temperature and depth. Finally, regarding the communication, we have an acoustic localization system and a WiFi connection. The WiFi connection improves the software manipulation of the vehicle and downloading of the stored data.

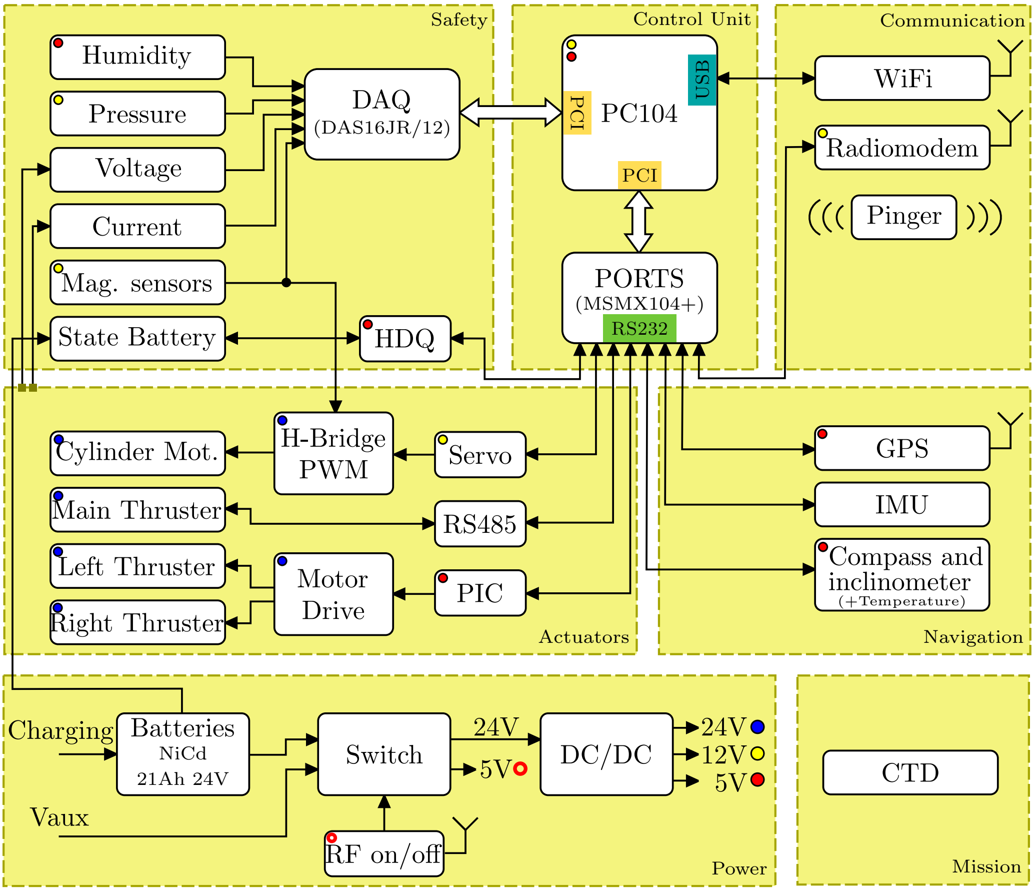


Figure 4.22: Final hardware architecture of Guanay II AUV

Chapter 5

Software

This chapter is dedicated to the description of the improvements made in the software. The programming language used is an object oriented language called LabVIEW [70], which allows programming with a powerful and very intuitive visual platform. Two parts of the software are modified, namely the vehicle and the base station.

The previous software worked as follows (see Masmitjà [82]). Firstly, the vehicle software had several modules which were structured as a machine state, and all data were grouped in one variable, a cluster. The flow diagram is shown in figure 5.1. It is cyclic: the vehicle communicates with the radio modem, next processes the information to set the actuators, and finally acquires the data from the sensors and continues again with the radio modem. One problem with this structure is the timing, that is, while the software is taking the data from the sensors, it is not possible to access the radio modem and therefore it is not possible to change the actuators. For this reason the software was very slow, and it was necessary a new system with time independent modules.

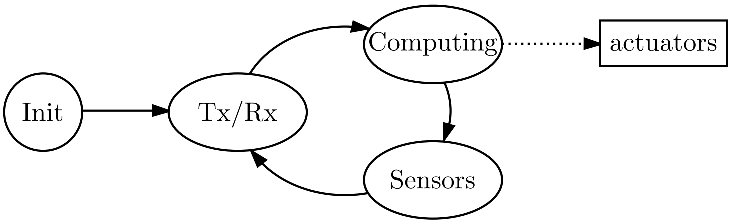


Figure 5.1: Initial software architecture of the vehicle

Concerning the protocol communication, the base station was able to send three types of commands: manual control, instruction to dive, and automatic control. Each frame was send periodically with a established period t. On the other side, when the vehicle received it, computed the command, and sent data to the base station using the same period t. However there is no timing synchronization for sending and receiving the messages. Additionally, the range of commands was very low.

Although the system had an automatic control algorithm, it was very short and did not work. For this reason the system needed a new system for the automatic control.

With respect to the coding, the vehicle software had a complex flow diagram which made it difficult debugging and adding new modifications. Similarly, the software in the base station had a very complex graphical user interface. The initial execution ran four SubVI (subroutine of LabVIEW) in parallel integrated in a unique window: controls, data sensors, graphics, and map. The way to see them was through tabs. However, with this configuration it was not possible to see two tabs at the same time, for instance, controls and data sensors.

This chapter is structured as follows. First we describe the modifications and improvements made in the vehicle software. Next we present the improvements in the software of the base station and the complementary software and, finally, we describe the new software architecture.

5.1 Vehicle software

After diagnosing the problems and bugs in the vehicle software we worked on it to solve them and to improve the robustness, versatility and speed of reaction.

As we said in the introduction, the software is implemented using the programming language of LabVIEW. The software was initially installed on a Windows XP operating system. However, the boot was very slow, taking about 7 min until establishing the RF connection with the base station. Hence, we decided to change the operating system to a Linux distribution. We selected Ubuntu 10.04 - Lucid, because it allows to configure several parameters and it is possible to dispense with unnecessary programs. For the correct operation, we had to create two new libraries, particularly for the IMU and DAQ, see section 5.1.2 for further details.

With respect to the programming in LabVIEW, we restructured the entire system recycling the existing code:

- We created one thread for each task, that is, one thread for each sensor, actuator and control system (see figure 5.2). Each thread has the blocks for initialization, operation, and error management to reset the thread if necessary. This restructuring allows to manage sensors and actuators with different frequency acquisition, and avoids the freezing of the entire system if some thread goes wrong. Additionally, it solves the problem of the state machine mentioned in the introduction.

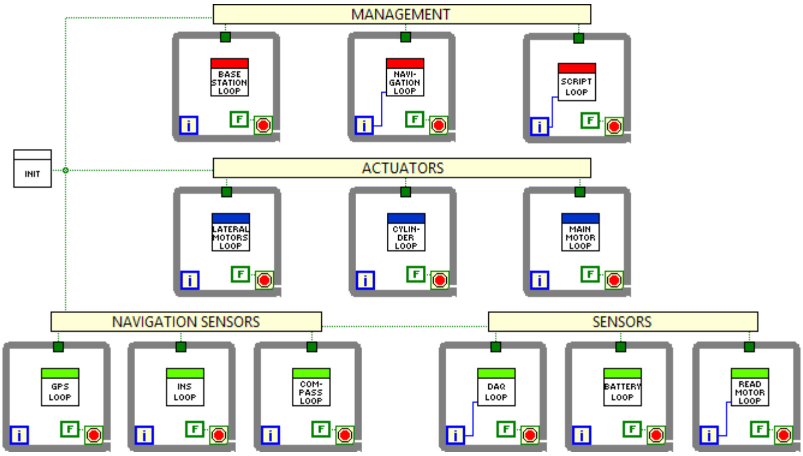


Figure 5.2: Main block of LabVIEW vehicle software. It displays different threads involved

- The cluster variable with all the data was ungrouped in several global variables. This change decentralizes the information and prevents conflicts between threads.

- There is a new option to emulate the global position, attitude, depth, and cylinder position. This option allows testing and debugging the software regardless of the sensors.

- There is a new option to run the vehicle in automatic mode.

- We solved several problems and bugs, specially with synchronization and variable assignation, giving robustness and fast response.

- We created more options and functions to improve the versatility.

- The GUI works in an independent thread. Since the graphical interface does not make sense in the vehicle software, this thread can be suppressed, if necessary, in the future.

We describe next some specific improvements in the software.

5.1.1 Command function

The vehicle had initially three commands, which were very limited to configure and control the vehicle. In the revised version we have added more parameters and consequently more commands to configure the operation of the vehicle. To implement it we use a subroutine dedicated to interpret and modify the parameters, the *command function.*

This function implements the communication between the threads. To avoid conflicts trying to access the same resources, this function works as a lock, so there is a synchronization accessing the parameters.

Additionally, this command function works as *message-response*, that is, every command has a response, with the function sending an OK in the case of instructions, or data information in the case of petitions. This operation improves the synchronization in messaging, one of the problems mentioned in the introduction (see figure 5.3).

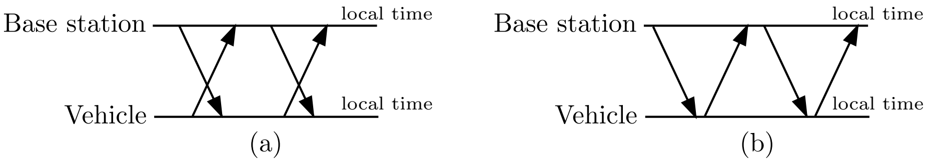


Figure 5.3: Temporal diagram of the protocol communication. a) Old approach. There is no synchronization between the nodes. b) New approach. For each message there is a synchronized response

Commands are divided in seven groups, as follows:

1. "`SET:`*variable*=*new_value*". This type of command is used to configure the parameters of the vehicle, like type of automatic control, maximum velocity, constraints to the main thruster, continuous data transmission, among others. If the assignation is correct the function returns `SETACK`.

2. "`GET:`*variable*". This type of command is used to get the actual value of a parameter. Additionally, it is used to get the sensors data.

3. “`MAN:`*MainT_RightT_LeftT*”. This is conceived to be used by the operator. The command modifies the propulsion of the three thrusters. The values must be between -100 and 100. The function returns data from the sensors, in order to show the performance of the vehicle to the operator.

4. “`IMM:`*TargetDepth_MaxTimeUW*”. This command initializes the immersion task of the vehicle. The arguments are the target depth and the maximum time allowed to be underwater. When one of these parameters is reached the vehicle goes up. The function returns `IMMACK`.

5. “`AUT:`*option*”. This command manages the autonomous operation of the vehicle. The response depends on the option argument. It starts/stops the operation, gets important information, or sets up the aborting operation, which sends the vehicle back home. Chapter 8 is dedicated to the autonomous navigation.

6. “`RUN:`*file*”. This command executes a script stored in a file, so the configuration of the parameters and the operation of the vehicle can be greatly simplified. Further details are shown in section 5.1.4. The function returns `RUNACK`.

7. “`CMD:`*command*”. This command sends a command to the console of the operating system. This is very useful when the WiFi connection fails (maybe due to the distance) and we want to get some information or to do some configuration. For instance, `CMD: cat file.txt` returns the data stored in a file.

A complete description of the commands and options can be found in appendix B.

5.1.2 Sensors

There are 6 different threads to sense the data of the sensors, and their main improvements are described next.

GPS

This thread manages the data from the GPS. It consists of a while loop which performs the lecture from the sensor, data storage in a file, and error management (see figure 5.4). Additionally, it includes the variables `emulLat` and `emulLong` to emulate a GPS frame for testing.

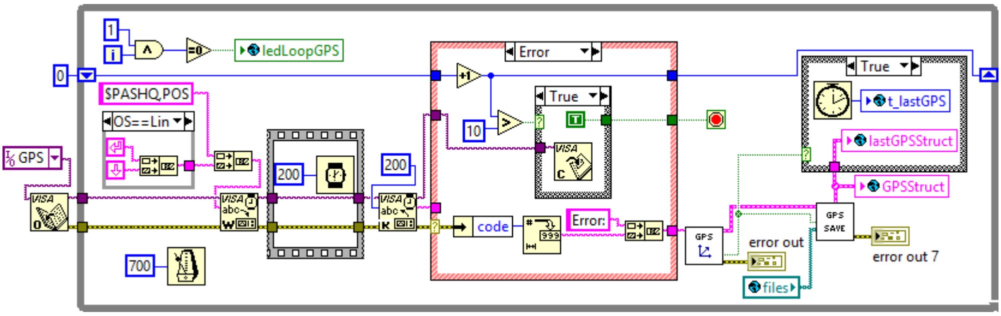


Figure 5.4: Principal loop of the GPS sensor

IMU

This thread manages the IMU. Since this sensor is a new one, this thread has been developed from scratch. A new library was required because the manufacturer only supports the LabVIEW blocks for Windows operating system. This development consists in the implementation of the low level protocol to communicate with the sensor by RS232 using frames up to 512 bytes.

The thread includes the lecture of the sensor, data storage, and error management. Figure 5.5 shows the main while loop of the thread.

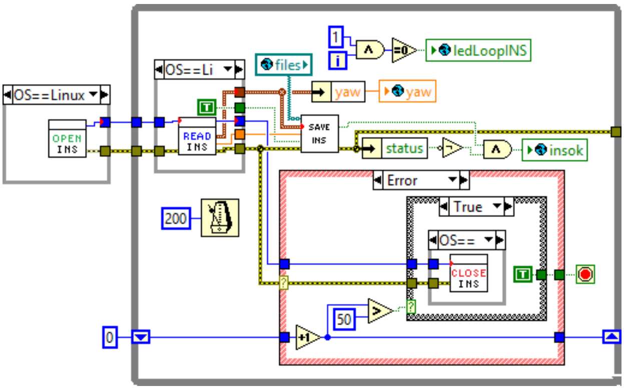


Figure 5.5: Principal loop of IMU sensor

Compass

This thread manages the data from the compass. It consists of a while loop which does the reading of the sensor, data storage in a file, and error management (see figure 5.6). Additionally, it includes the variable `emulYaw` to emulate a compass frame for testing. Finally, we have added the possibility to calibrate the measure loading a calibration curve stored in a file.

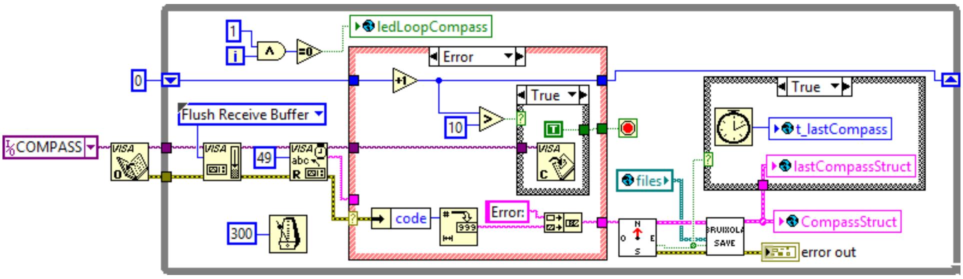


Figure 5.6: Main loop of the compass sensor

DAQ

This thread manages the information from the data acquisition board. Like in the IMU case, a new library was required to be created because the manufacturer only supports the LabVIEW blocks for Windows operating system. The communication at low level with the hardware was established using *libcomedi0* package [73].

This thread reads the voltage of 6 signals and normalize them to the correct units. 4 of them are analogue (pressure, humidity, drivers voltage, and drivers current), and 2 are digital (magnetic sensors to sense the ends of the cylinder). The routine also includes some variables to emulate the sensing. Figure 5.7 shows the principal routine of the thread.

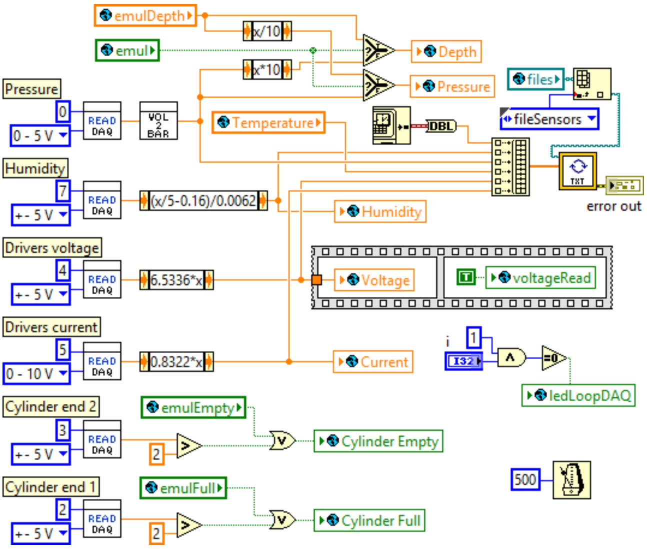


Figure 5.7: Principal loop of the data acquisition board

Battery

This thread manages the information from the battery. It consists of a while loop which reads the sensor, performs data storage in a file, and error management (see figure 5.8). The measure of the charge of the battery is made by Coulomb counting. This means that the measure is relative to the incoming/outcoming current and it is not cent per cent reliable. For this reason we have added a special initialization. Thanks to the voltage measure it is possible to estimate if the battery is empty of full by measuring 23 or 28 V respectively, so there is a module to verify these limits and to rewrite the charge lecture.

Data of main thruster

This thread manages the data acquisition from the main thruster, *i.e.* current and revolutions per minute. It consists of a while loop which does the lecture of the sensor, data storage in a file, and error management (see figure 5.9).

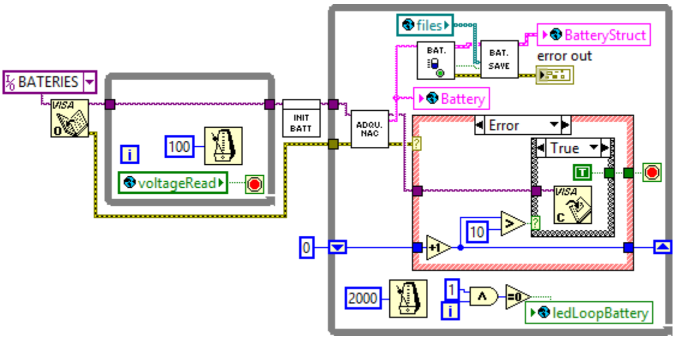


Figure 5.8: Principal loop of the battery sensor

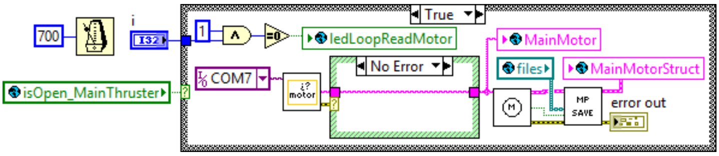


Figure 5.9: Principal loop to acquire data from the main thruster

5.1.3 Actuators

There are 3 different threads to control the actuators: Main thruster, lateral thrusters, and cylinder.

Main thruster

This thread manages the actuation of the main thruster. It consists of a while loop which has the protocol to control it, and error management (see figure 5.10).

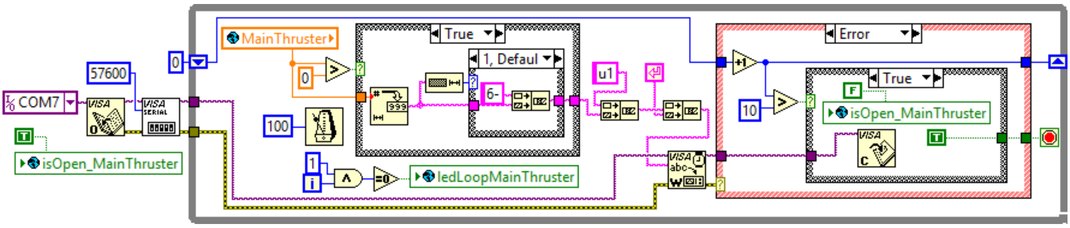


Figure 5.10: Principal loop to actuate the main thruster

Lateral thrusters

This thread manages the actuation of the lateral thrusters. As we saw in section 4.1.2 there are two instructions to control them. Figure 5.11 shows the while loop with this protocol and the error management.

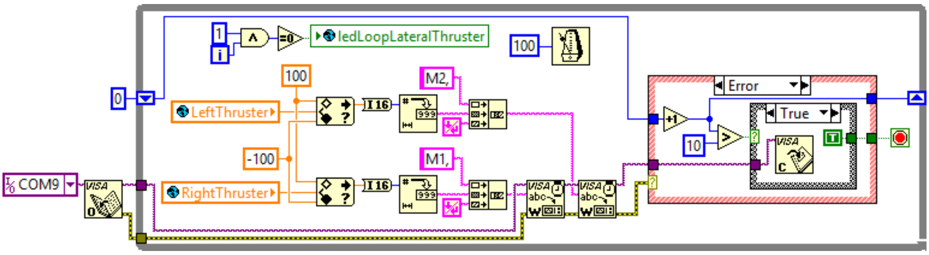


Figure 5.11: Principal loop to actuate the lateral thrusters

Cylinder

This thread manages the actuation of the cylinder thrusters. For this version this thread uses the magnetic sensors, which are sensed by the DAQ board (section 5.1.2), in order to know when the cylinder is full or empty, and thus to continue with the operation going down or going up. Figure 5.12 shows the principal loop of this thread to change between the states and to manage the errors.

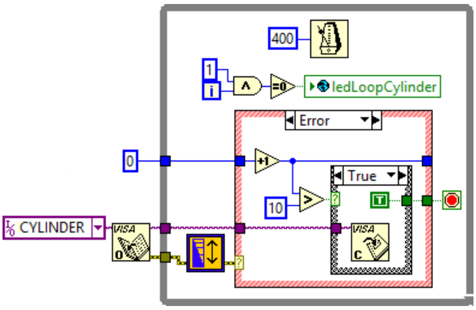


Figure 5.12: Principal loop to actuate the cylinder

5.1.4 Management

This section presents the high level management of the entire system. Formerly the code consisted of a single module, the base station communication. For this version we have created three threads: base station communication, navigation, and scripts.

Base station communication

This thread has the task of communicating the vehicle with the operator in the base station. It receives a frame by the radio modem, does the instruction, and finally sends a response to the base station.

The thread consists of a while loop which reads the bytes in the port until an end of line (EOL) is received. Next, using the *command function* it sends the instruction to the appropriate module. Like the above, this thread has an error management for possible problems with the port. Figure 5.13 shows the principal loop of this module.

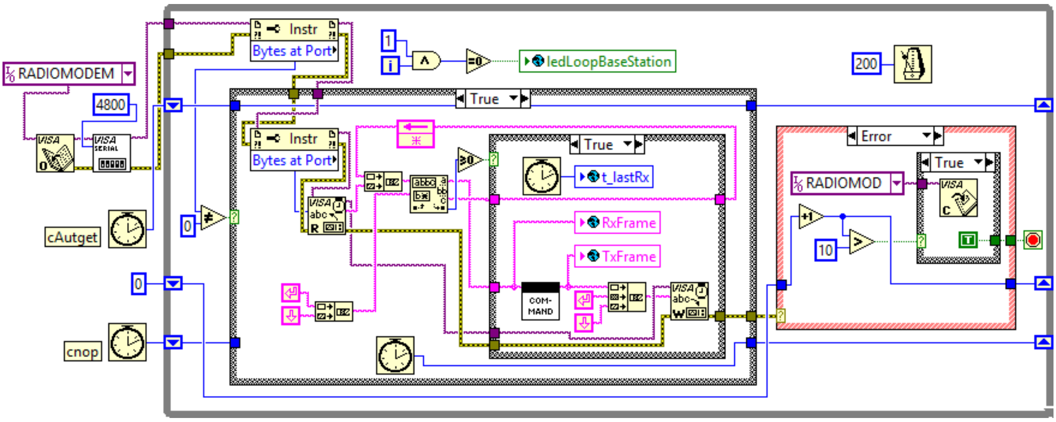


Figure 5.13: Main loop of the base station module

Scripts

In order to simplify the operation of the vehicle we have added this thread to implement several instructions sequentially. The idea is to save a file in the hard disk of the vehicle, containing a sequence of instructions. The operator selects then the file with the command `SET:SCRIPTFILE=`*file.txt*, and finally runs the script with the command `RUN`.

While in previous thread the standard output (stdout) was the base station, in this thread it is a log file. The file stores the time, command, response, and the waypoints reached if the vehicle works in automatic mode. Figure 5.14 shows the principal loop of this thread.

Navigation

This thread controls the navigation of the vehicle. Among the high level management threads, and as a safety feature and to avoid conflicts, this is the unique thread that modifies the action of the cylinder and thrusters.

The way to act on the actuators depends on the operation mode, which can be manual or automatic. For this thread we have created an automatic module with several options and types of guidance (Chapter 8 is dedicated to the autonomous navigation). Waypoints are taken from a file in the vehicle.

As a security feature the thrusters are stoped if any of following conditions is met: 1) The vehicle is underwater (configurable by the parameter `DEPTHSURFACE`), 2) There is no GPS signal (configurable by the parameter `MAXTGPS`), 3) There is no compass signal (configurable by the parameter `MAXTCOMPASS`), 4) There is no communication with the base station (configurable by the parameters `MAXTMOT` and `MAXTAUTGET` for manual mode and automatic mode respectively).

This thread also calculates the actual velocity and maximum depth reached. Finally, all values of the actuators are saved to a file. Figure 5.15 shows the principal loop of the thread.

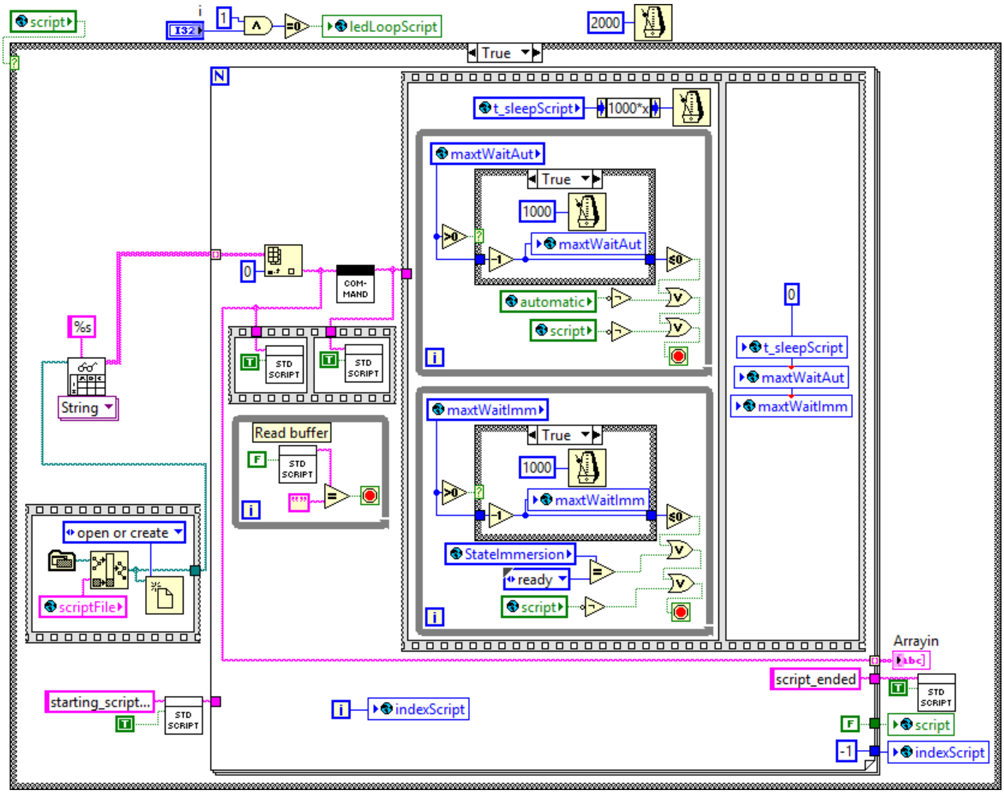


Figure 5.14: Principal loop of the Script module

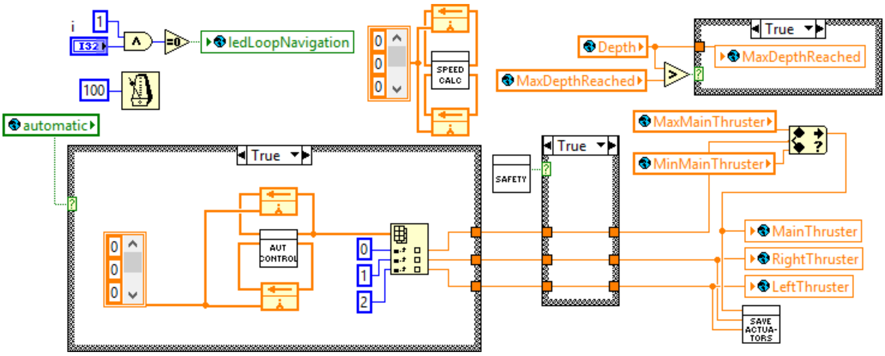


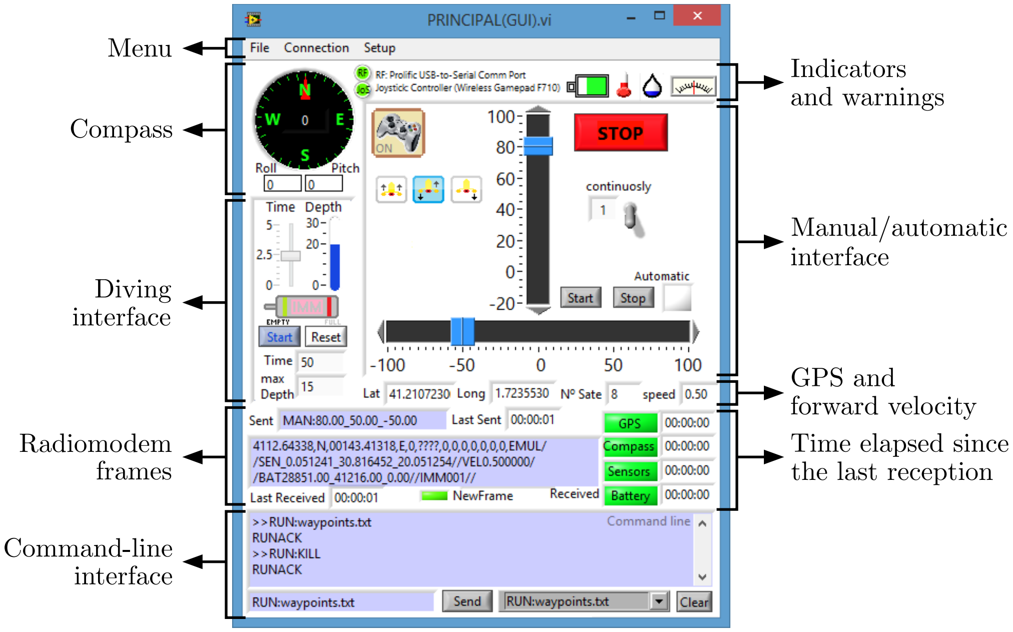
Figure 5.15: Principal loop of the Navigation module

5.2 Base station software

The operator is in the base station. For this reason it is necessary to have a software with a robust graphical user interface (GUI) which facilitates the operation of the Guanay II AUV.

The software works with 3 threads. One of them is dedicated to manage events related with the radio modem in order to establish communication with the vehicle. The second thread manages the visualization of indicators and warnings. And the third thread is used to interact with the joystick. This thread is necessary because the joystick does not has event handlers.

The GUI is very intuitive, since all controls and sensors are shown in a single reduced window without losing functionality, as it can be shown in figure 5.16. There are several panels:



RF: Prolific USB-to-Serial Comm Port
Joystick Controller (Wireless Gamepad F710)

Roll 0 Pitch 0

continuosly 1

Time 50
max Depth 15

Lat 41.2107230 Long 1.7235530 Nº Sate 8 speed 0.50

Sent MAN:80.00_50.00_-50.00 Last Sent 00:00:01

```
4112.64338,N,00143.41318,E,0,????,0,0,0,0,0,0,0,EMUL/
/SEN_0.051241_30.816452_20.051254//VEL0.500000/
/BAT28851.00_41216.00_0.00//IMM001//
```

Last Received 00:00:01 NewFrame Received

GPS	00:00:00
Compass	00:00:00
Sensors	00:00:00
Battery	00:00:00

```
>>RUN:waypoints.txt
RUNACK
>>RUN:KILL
RUNACK
```

RUN:waypoints.txt Send RUN:waypoints.txt

Figure 5.16: Base station software. Graphical user interface

- *Manual/automatic interface.* The panel has two bars to control the thrusters. The vertical bar controls the main thruster, and the horizontal bar controls the lateral thrusters. These bars can be handled by the keyboard, the mouse, or more preferably, by the joystick.

  The horizontal bar works in conjunction with 3 buttons, located on the left, which determine the way that the thrusters are actuated. Finally, the panel includes buttons and indicators for the automatic mode.

- *Diving interface.* This panel has the bars and controls to set the diving motion. It has indicators to know the elapsed time of the last immersion, the maximum depth reached, and if the cylinder is full or empty.

- *Radiomodem frames.* This panel shows the last frame sent and the last frame received.

Additionally, it shows the elapsed time since the last frame reception and receiving data from the different sensors.

- *Compass.* It shows the heading, roll and pitch of the vehicle.

- *GPS and velocity.* It shows the latitude and longitude of the vehicle, the number of satellites used, and the forward velocity of the vehicle.

- *Indicators and warnings.* It shows the state of some vehicle sensors (battery, inside temperature, inside humidity, and pressure), and the state of the base station ports (radio modem and joystick).

- *Command-line interface.* This interface allows the operator to send commands to the vehicle (see appendix B for the communication protocol). On the right, there is a combo box with some useful commands (configurable with predefcommands.txt), and a button to clear the command history of the console.

- *Menu.* It has a menu to diagnose the connection with the vehicle. Additionally it has a button to start Google Earth in parallel.

5.3 Complementary software

We added some complementary software to facilitate the operation of the vehicle.

- *Google Earth.* This software is used in the base station to see the path tracked by the vehicle. It works in real time: the software of the base station updates a kml file continuously with the vehicle position and the waypoints. Then Google Earth opens this file and shows the travel in the map.

- *FTP connection.* The vehicle has installed vsftpd, which is a software to establish a FTP server. This allows an easy way to download the stored files in the vehicle after a mission. A client node is required in the base station (like FileZilla or a browser).

- *SSH connection.* The base station has installed putty, a client to establish a secure shell connection (SSH). This software is used to access the console of the vehicle operating system remotely.

- *TeamViewer.* This software is used to control remotely the operating system of the vehicle. Since it shares the desktop, it allows for an easier manipulation.

5.4 Conclusions

The new software has improved and simplified the manipulation of the Guanay II AUV. Figure 5.17 shows the new software architecture in both the vehicle and the base station. The main improvements are summarized as follows.

- The software of the vehicle has been developed to work in Linux. Libraries were created to manage the IMU and the DAQ.

- The vehicle uses one thread for each task. This helps to interact with different frequency acquisition and to strengthen the software.

- There are more commands to control the vehicle. They are implemented using the *command function*, which works as a lock to avoid conflicts between the threads and improves the timing synchronization.
- Three threads form the high level management. The first one communicates with the base station, the second executes scripts, and the third, the navigation module, has the modules for manual/automatic control, and it sends the instructions to the thrusters and cylinder.
- The GUI of the base station is very intuitive, simply and complete. It shows all controls and sensors. Additionally, it has a command-line interface.
- Complementary software have been installed to improve the communication with the vehicle and the file transfer.

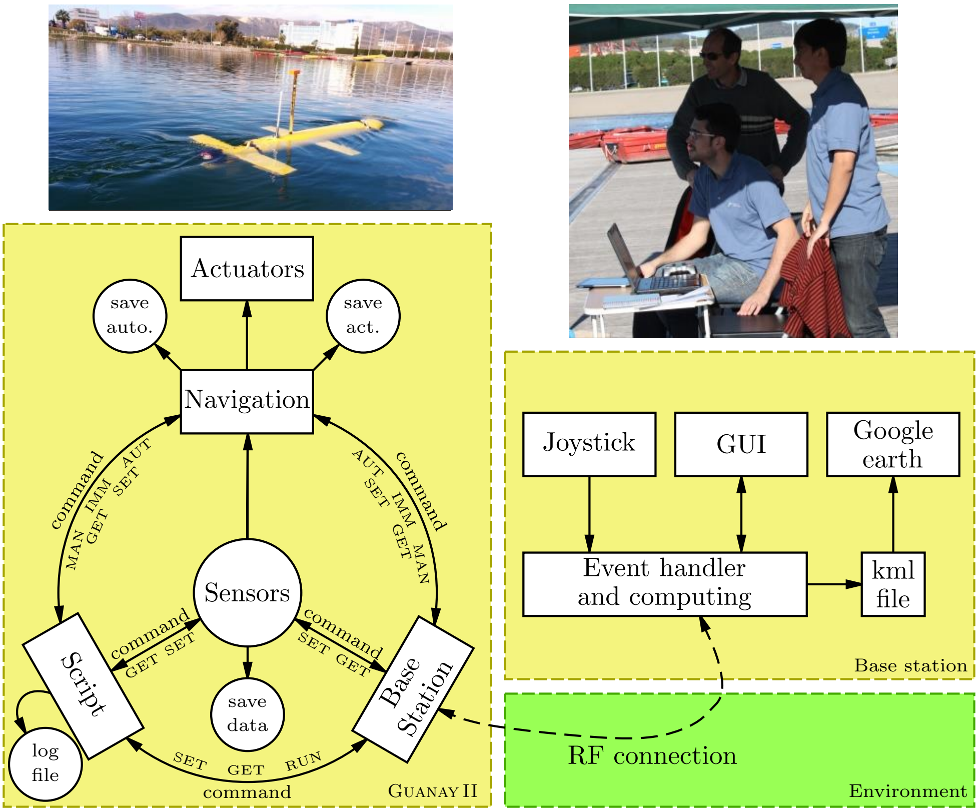


Figure 5.17: Final software architecture

Chapter 6

Field Tests: Operability

In this chapter we present different field tests to investigate the correct operation of the vehicle. The structure of this chapter is as follows. First we present some tests made at the laboratory. Following, we present some tests made in the Mar Menor lagoon as part of a joint experiment with several institutions. Next we discuss tests in a swimming pool and then tests continue with an immersion at open sea. Finally, tests at the Catalonia Olympic Channel are presented.

6.1 Laboratory tests

All changes, new boards and new instruments were fitted in the watertight module and the antenna. Figure 6.1 shows the final disposition of all of them. The antenna is filled with resin to make it watertight. The changes in the power system help to dispense with the opening of the watertight module when the battery has low charge.

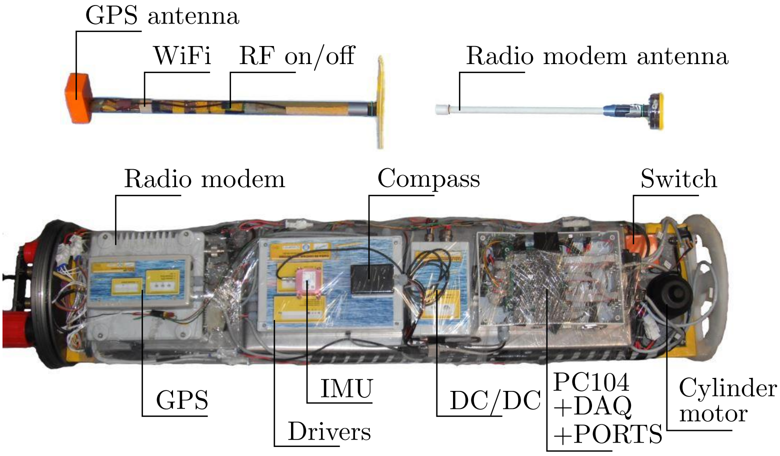


Figure 6.1: Final disposition of the electronics

6.1.1 Booting

One of the first tests was the booting of the operating system (see figure 6.2). When the vehicle used Windows XP-SP3 from the starting until the communication establishment took about 7 min, but using Ubuntu 10.04 Linux operating system it took about 2.5 min, which means a 64% reduction of the initial time.

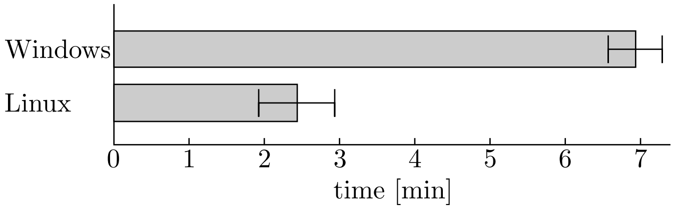


Figure 6.2: Booting time until the communication with the base station is established

6.1.2 Energy and Consumption

We have tested the energy consumption of the vehicle, which allows us to estimate the vehicle autonomy.

Table 6.1 shows the power consumption of the different devices. There are different configurations. When the vehicle does not use any actuator its consumption is 30.15 W. Using the three trusters at 100% it consumes 550.15 W. The cylinder's consumption depends of the depth. If the vehicle is at 40 m deep, its consumption is 234.15 W.

Table 6.1: Maximum consumptions of the Guanay II

Device	Voltage [V]	Power [W]
GPS	5	2
PC104	5	9.9
Compass	5	0.1
Radio modem	12	18
Actuators Driver	5	0.15
Main Thruster	24	300
Right Thruster	24	110
Left Thruster	24	110
Cylinder	24	204

On the other hand, the battery yields 24 V and has a capacity of 21 Ah. Thus, theoretically, the autonomy without using any actuator is

$$t = \frac{C}{P} = \frac{24 \times 21\text{Wh}}{30.15\text{W}} = 16.7\text{h}$$

where C is the capacity, and P the power consumption.

We have made a test to corroborate this time at the laboratory. The test consisted in leaving the vehicle started until the battery is dead. Additionally the radio modem sent frames every second. The result is shown in figure 6.3. The capacity of the batteries reduced uniformly, and the total time was about 16 h, which largely coincides with the theoretical calculation.

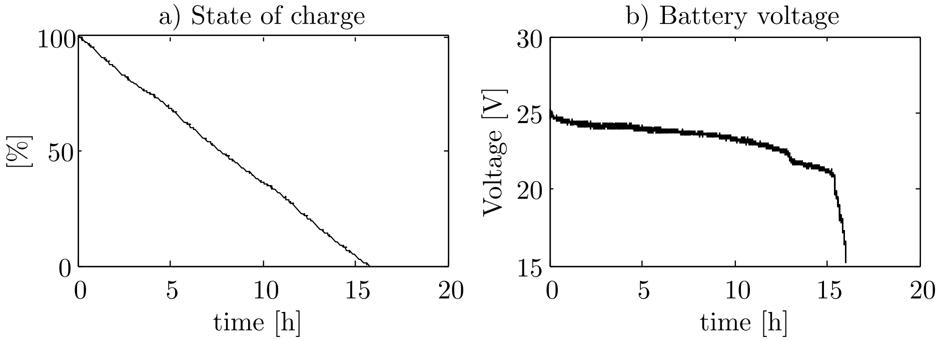


Figure 6.3: Autonomy test without the use of actuators. a) Battery life. b) Battery voltage

6.2 Mar Menor Lagoon

The vehicle has been tested in the *"Underwater robotic experiment in the Mar Menor Coastal lagoon"*, a joint experiment with AUVs developed in Murcia (Spain) between several institutions from Spain, Portugal and USA, in order to measure and assess the influence of sea water, in the Mar Menor, and adjacent area to the Mediterranean sea (see figure 6.4). Many marine species are unable to enter this lagoon because of the strong environmental gradient, mainly imposed by the salinity. The principal goal was to measure the salinity around the bounds of the sea in order to identify the plume. Figure 6.5 shows one of the tracks carried out with the Guanay II and the salinity measurements from the experiment.

Results show a clean movement through the water, and a good stabilization both in roll

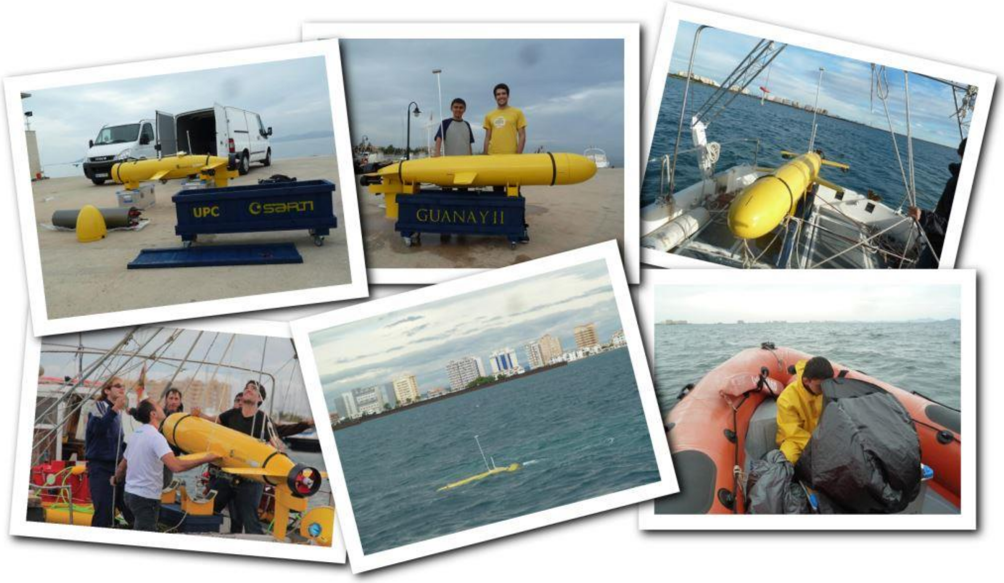


Figure 6.4: Photos during the underwater robotic experiment in the Mar Menor coastal lagoon

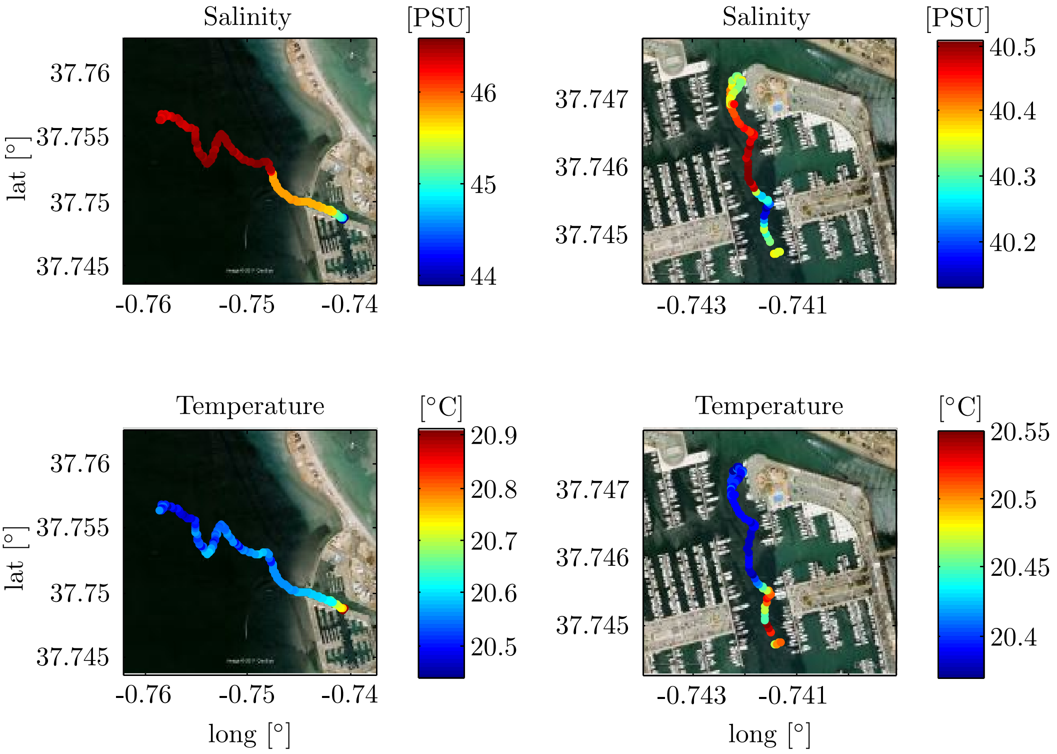


Figure 6.5: Mar Menor: records of salinity and temperature

and pitch. The total distance in one trial was about 2.5 km with an average speed of 0.54 m/s.

This was the first time that the vehicle navigated in open sea, and the test helped to visualize the problems and needs of the Guanay II in order to be more robust. The needs included a new system to charge the batteries from the exterior, and a better connection with the PC104, like a shared desktop, among others. These problems were solved and they are presented in this thesis.

6.3 Swimming pool

Several tests in a swimming pool were performed to verify the correct operation of the changes and new features added to the vehicle.

One of first tests allowed to adjust the buoyancy and check the new foams. The results were satisfactory, and we could test some vertical diving movements (up to 2 m of deep).

At the pool we also observed the improvements brought about by the introduction of the WiFi connection, facilitating an easier manipulation of the vehicle. At the same time, the introduction of the RF remote control on/off allowed to turn on, turn off, and restart the vehicle more easily. Figure 6.6 shows some pictures during these tests.

The thrusters were also tested, but since the swimming pool had small dimensions the navigation was not performed properly.

Figure 6.6: Photos during the tests in swimming pool

6.4 Immersion at the OBSEA

Another test with the vehicle was made in the OBSEA [94], a marine observatory located 4 km off the coast of Vilanova i la Geltrú, Spain. The test consisted of realizing several immersions at different depths taking temperature data, as can be seen in Figure 6.7. The results showed good performance. The vehicle reached the depths 5 m, 10 m, 15 m and 20 m (maximum depth of the sea at this point) in a satisfactory way. The descent speed was about 0.07 m/s and the ascending one about 0.11 m/s.

Regarding attitude, the vehicle had a slight deviation between the center of buoyancy and the center of gravity when the cylinder takes in water, prompting an inclination in the pitch of about 10° going down and 5° going up. Although there is an unwanted small pitching angle, it does not produce any effective horizontal displacement of the vehicle since the vertical speed is slow and the design of the fins is for stabilization only. The roll movement is close to zero degrees. The yaw movement, despite not being automatically controlled, was stable over several time periods, with some deviations due to the currents.

Figure 6.7 also shows the consumption during the immersions. When the vehicle takes water in, the cylinder consumes 55 W (25V@2.2A) during a short period, and has a higher consumption when climbing. This is because it has to overcome the pressure caused by coming up from the depths: For 5 m it uses 130 W (24.5V@5.3A), in 10 m it uses 158 W (24.3V@6.5A), in 15 m it uses 167 W (24.2V@6.9A), in 20 m it uses 187 W (24V@7.8A). We noticed that these powers were greater than the powers seen at the hyperbaric chamber (see section 4.3.2). However, we realized that the constant use of the cylinder caused the salt water impurities to adhere to the walls, causing an increase in the current. To avoid this problem the cylinder needs rigorous cleaning periodically.

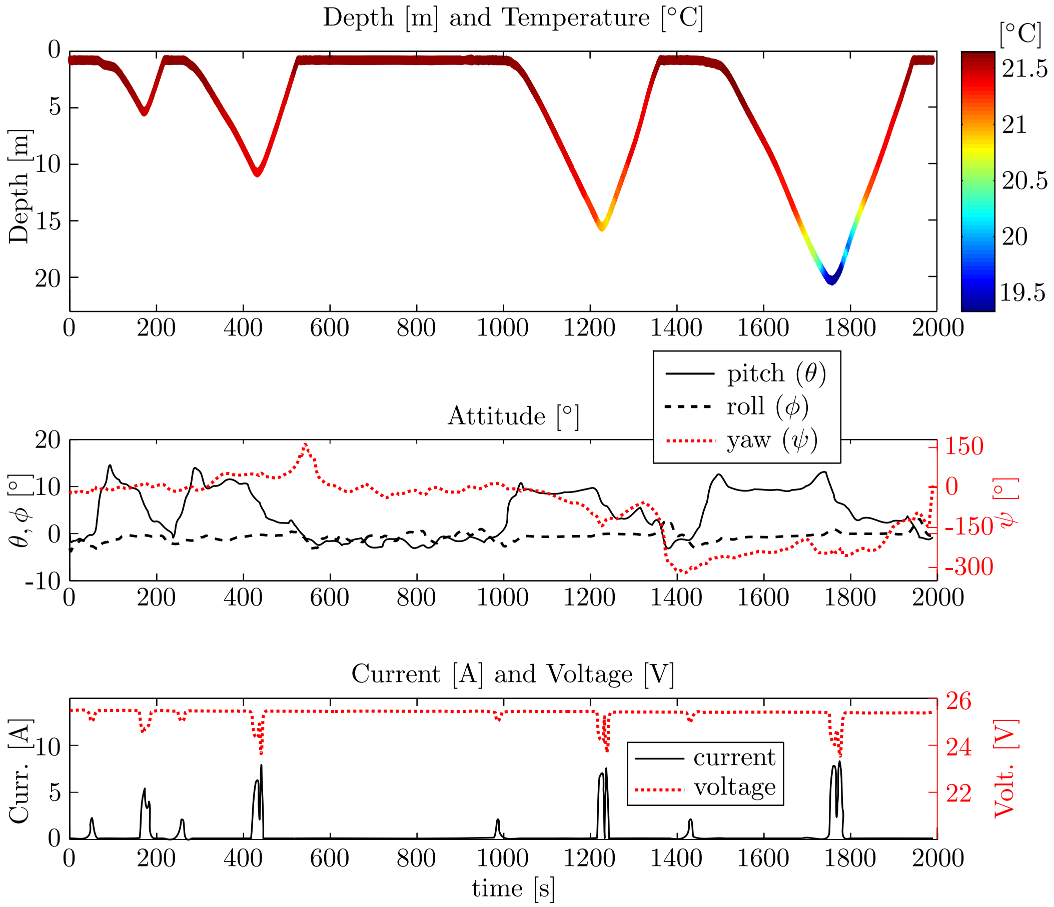


Figure 6.7: OBSEA: Depth test

6.5 Catalonia Olympic Channel

The Catalonia Olympic Channel is a seawater channel located in Castelldefels [157]. Its dimension is 1200 m in length and 200 m in width, which is enough to perform navigation maneuvers with the vehicle. Additionally, its platform allows to set up the infrastructure, including the base station, comfortably.

The tests made in this site allowed to validate all changes and new features, namely the driver for the H-bridge for the lateral thrusters, the WiFi communication, the remote control to turn on/off, and the improvements in the software, among others. Figure 6.8 shows some pictures made during these tests.

Thanks to the characteristics of the channel (low currents and thus low perturbations) we used it to test the navigation and maneuverability of the vehicle. This allowed us to determine the hydrodynamic model of the vehicle. Additionally, we used the channel to test the controllers for autonomous navigation. The second part of this thesis provides a detailed study of these issues.

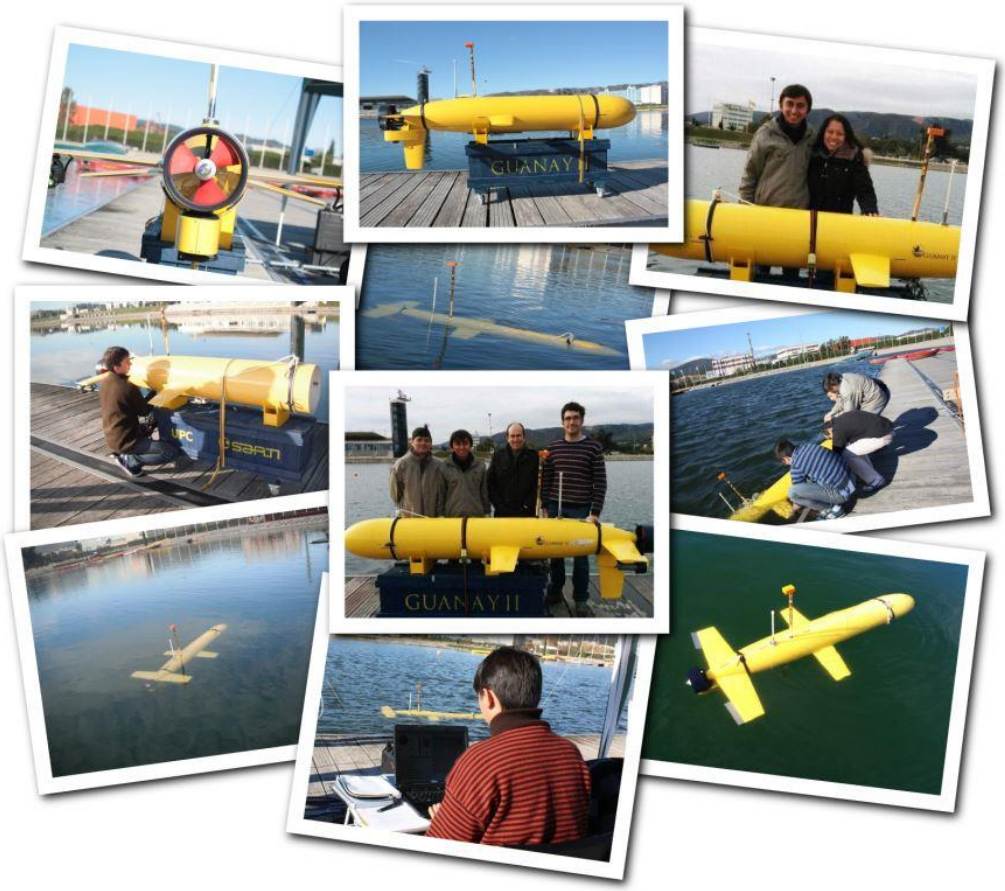


Figure 6.8: Photos during the tests on the Catalonia Olympic Channel

6.6 Lessons learned

The experience acquired during the laboratory and field tests have taught us several lessons, which would have been very helpful if we had known before. Take into account that the development of this vehicle, unlike a commercial one with minimum guarantees of operation, was made in the research group, and we have to found all the little mistakes and solve them.

The principal lesson learnt is that a campaign in the sea is very exigent, and minor inconveniences presented in the laboratory or swimming pool can be a serious problem at sea. The work at sea is difficult due to the waves that move the ship, the reduced spaces that make it difficult to work comfortably, the absence of electricity, the weather conditions which may be rainy or poor visibility, or due to the stress of all these things together.

For this reason the manipulation of the vehicle must be as simple as possible. In the case of the hardware the starting system needs to be simple and remote. The vehicle needs a WiFi connection to control its desktop remotely, likewise it needs a joystick to input motion commands easily. All connections must be very robust and nowhere delicate, because there may be sharp movements in the vehicle. The tightness must be ensured at all points. The way to charge the batteries must be of type connect/disconnect, so that there is no need to disassemble things. Additionally, a protocol, including a tracking system, should be established for emergency cases.

In the case of the software all of the modules must be robust. The software at the base station must be intuitive and very simple without losing functionality. The fewer clicks the better, trying to do as much as possible automatically. Adding functions to emulate a real test in the laboratory helps to detect and solve bugs.

With all of these improvements the operator can work more relaxed and the desired minimum operating conditions to work at open sea are achieved.

6.7 Conclusions

In this chapter we have showed the performance of the Guanay II AUV after implementing the improvements of buoyancy, hardware and software. First, at the laboratory, we found that with the change of the operating system to Linux, the starting time, including establishing communication with the base station, was reduced by 64%. Similarly, we have tested the power consumption of the different devices at the laboratory, with the result that the vehicle has an autonomy of 16 h without using any actuator.

The vehicle have been tested at sea, in the Mar Menor lagoon, where we observed an steady movement through the water and a good stabilization both in roll and pitch. The CTD was tested obtaining good results. This test also served to identify some possible improvements in hardware and software, regarding the handling of the vehicle, which were then implemented.

The vehicle was improved with new foams designed to be resistant under water. Additionally, it was upgraded with a WiFi connection and an RF remote control on/off for easy handling. Tests in a swimming pool showed good performance and easy manipulation.

Further tests at sea showed a good performance of the immersion system. The vehicle was tested up to 20 m of deep, reaching a descending and ascending speeds of 0.07 m/s and 0.11 m/s, respectively.

Finally, several tests were performed at the Catalonia Olympic Channel, mainly to check the behavior of the thrusters when following established paths. The new driver of the lateral thrusters worked very well, providing better maneuverability. Additionally, the improvements in the software helped to control the vehicle faster and in an easier way.

From these results we concluded that the vehicle was fully operative, both in surface and under water, and that the operator can control, and get information from, the vehicle in a convenient and flexible way.

Part II

Hydrodynamic Model and Control

Chapter 7

Modeling

This chapter presents a mathematical model for the GUANAY II AUV. Mathematical models are necessary in order to simulate the performance of the vehicle in open loop and to design controllers for autonomous navigation.

For vehicles in general, one can, in principle, separate the modeling of the kinematics from that of the dynamics. Kinematics refers to the vehicle's movement given its acceleration, from which its velocity and position can be computed. On the other hand, dynamics refers to the forces and torques that generate those accelerations. In the case of marine vehicles there are several forces and torques generated by the movement inside a fluid that must be taken into account, and the calculation of some coefficients related to them is one of the main steps in the modeling process.

Modeling of marine vehicles is developed in works by Fossen [42] [43], Wadoo [156], Perez [100], Nikolai [67], among others, where the coefficients, forces and torques needed in the characterization of a general ship are described. For a given vehicle one encounters two main difficulties: first the selection of the coefficients and then their calculation.

It is difficult to know which coefficients are relevant for the characterization of a specific vehicle, such as the GUANAY II AUV. The master's thesis of Prestero [105] can be used as a guide s because it uses a set of coefficients adapted to the Remus AUV, which is a vehicle with the same geometry as the GUANAY II. Prestero models and linearizes the Remus in the vertical plane, therefore he doesn't need the linear damping coefficients and he only uses the quadratic damping. However, although this assumption simplifies the modeling, the dynamics when it is linearized in the horizontal plane is not very good.

Additionally, it is important to remember that the matrices and coefficients of the external forces and torques acting on a marine vehicle correspond to an approximation of a more complex dynamics of the surrounding fluid. Triantafyllou and Hover, professors of MIT, mention that "*the method of hydrodynamic coefficients is a somewhat blind series expansion of the fluid force... the intractability of the governing equations of motion of viscous fluid prohibits, at least today and in the near future, a computation of these forces*" [147, Chap. 3], and mention that the actual assignation of the coefficients are taken from a Taylor series, and other necessary terms that can be found empirically, like the coefficient $Y_{|v|v}$, which is associated with the modulus of the speed in the y-axis.

With respect to this, we have found an unstability in the modeling of the GUANAY II when it is moving backwards, which, up to our knowledge, is not mentioned in the literature in a detailed way. Our study shows that the model must be symmetric in order to stabilize the movement of the vehicle when it goes backwards.

The second problem is the calculation of the coefficients, which are strongly coupled, making its individual computation quite difficult. Moreover, the navigation sensors of the Guanay II, that is GPS and compass, are not good for recording the vehicle dynamics because the frequency of the GPS and the compass is 1 Hz, and the calculation of velocity and acceleration with this data is not good at all. The thesis of Prestero [105] proposes a method of calculation using directly the geometry of the vehicle. For this, he proposes the computation of a set of integrals and multiplications for vehicles that have the Myring profiles [91], which coincide with the Guanay II geometry and can be used as reference. Each of these formulas calculate the coefficients one by one, which is very useful. However, these theoretical computations must be compared with experimental data, and the results can be quite different. In fact, Prestero mentions that some coefficients for his vehicle were corrected with the results from field experiments, but he does not mention the method used to do that.

Other authors do the identification with different methods using experimental data and fitting it to the equations of marine vehicles. The work of Åström and Källström [134] proposes a method of identification using the Maximum-likehood estimation (MLE). The work of Ridao et al. [112] shows a method of identification of underwater vehicles using Least Squares (LS). The method of Least Squares can approach the identification in a satisfactory way, and it does not need the coefficients to be well decoupled into sub-systems. On the other hand, Fauske et al. [39] propose a particular formula based in non-linear least squares (NLS), but this method can only be used for decoupled sub-systems. The work of Alessandri et al. [3] shows a technique using two steps, namely Least Squares (LS) and Extended Kalman Filter (EKF). Again, this method requires the coefficients of the system to be fairly decoupled, making it difficult to implement. Tiano et al. [144] also propose the use of the Kalman Filter for the identification of the Hammerhead AUV. Finally, the use of a cost function is also proposed by Tiano in [143].

Knowing all these problems and limitations in the modeling of marine vehicles, both in selection of coefficients and in their calculation, in this work we use the set of coefficients of Prestero due its similarity with the Guanay II, but including the linear damping coefficients in order to exploit in a better way the dynamics when it is linearized in the horizontal plane. The formulas of Prestero are used for a theoretical calculation as a preliminary way, but afterwards we employ an LS method derived from Ridao's paper, using experimental data of the vehicle in order to compare with the theoretical values.

Throughout this work, we use he notation of SNAME (1950) [129]*, which is widely employed in the literature.*

7.1 Review of modeling of marine vehicles

7.1.1 Kinematics

Kinematics treats the geometrical aspects of motion, once the forces and torques, and hence the accelerations, have been specified. A marine vehicle is a 6 degrees of freedom (DoF) system, since 6 independent coordinates are necessary to determine the position and orientation, three for axial movement and three for angular movement: "surge", the displacement in the x-direction, "sway" in the y-direction, "heave" in the z-direction, "roll", the rotation about the x-axis, "pitch", the rotation about the y-axis, and "yaw", the rotation about the z-axis.

Reference frames

It is convenient to work with two reference frames, one frame fixed to the vehicle, called "body frame", in order to manipulate all the forces and accelerations induced in the vehicle; and another frame fixed to the Earth, in order to calculate the vehicle movement. There are several frames fixed to the Earth, such as the ECI frame, the ECEF frame and the NED frame [43]. In this thesis we will use the NED frame.

The body frame consists of a frame with origin in the center of buoyancy CB of the vehicle, and such that the x-axis points to the bow, the y-axis points to starboard and the z-axis points to the bottom. On the other hand, the NED frame has its origin in the surface of the sea, with the x-axis pointing to the North, the y-axis to the East and the z-axis pointing down. Figure 7.1 shows these two frames. The positions, velocities and forces, as defined by SNAME (1950) [129], are:

$$
\begin{array}{lll}
\text{NED position:} & \mathbf{p}^n = \begin{bmatrix} n \\ e \\ d \end{bmatrix} & \text{Attitude (Euler angles):} \quad \boldsymbol{\Theta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \\
\text{Body-fixed linear velocity:} & \mathbf{v}^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} & \text{Body-fixed angular velocity:} \quad \boldsymbol{\omega}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\
\text{Body-fixed force:} & \mathbf{f}^b = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} & \text{Body-fixed torque:} \quad \mathbf{m}^b = \begin{bmatrix} K \\ M \\ N \end{bmatrix}
\end{array}
\tag{7.1}
$$

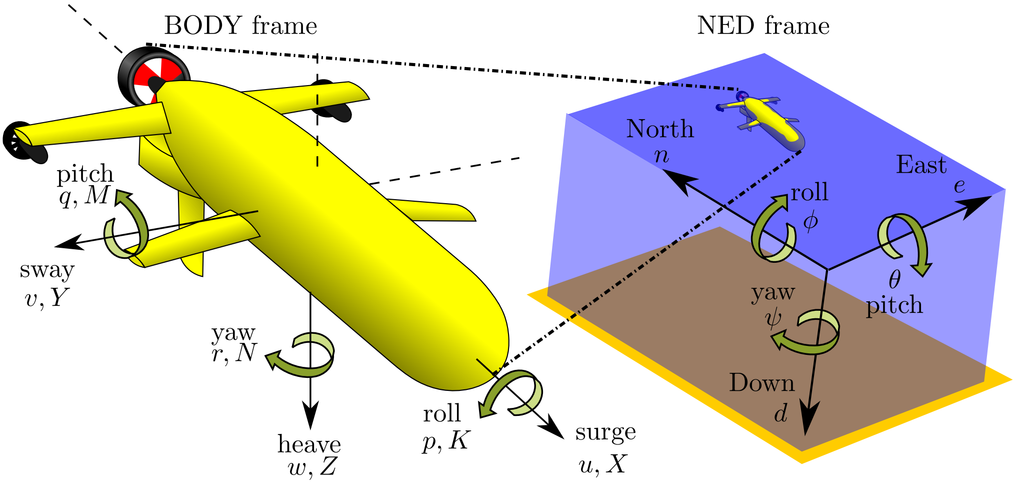


Figure 7.1: Body frame and NED frame

The general motion of a marine vehicle in 6 DoF is described by the following vectors:

$$
\boldsymbol{\eta} = \begin{bmatrix} \mathbf{p}^n \\ \boldsymbol{\Theta} \end{bmatrix}, \qquad \boldsymbol{\nu} = \begin{bmatrix} \mathbf{v}^b \\ \boldsymbol{\omega}^b \end{bmatrix}, \qquad \boldsymbol{\tau} = \begin{bmatrix} \mathbf{f}^b \\ \mathbf{m}^b \end{bmatrix}. \tag{7.2}
$$

Transformation between BODY and NED

A rotation matrix $\mathbf{R}$ is used to rotate the coordinates in order to change from a frame to another. The parameters of the rotation are the Euler angles roll (ϕ), pitch (θ) and yaw (ψ), and multiplying the rotation matrices for each one of them (the interested reader is advised to consult Fossen [43]) the following general formula to rotate from the body frame to the NED frame is obtained:

$$\mathbf{R}_b^n(\mathbf{\Theta}) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \tag{7.3}$$

where subscript $\cdot_b$ and superscript $\cdot^n$ represent the notation "from body frame to NED frame", and $s\cdot = \sin(\cdot)$ and $c\cdot = \cos(\cdot)$. The rotation matrix $\mathbf{R}$ is orthogonal, and hence the inverse rotation matrix is given by $\mathbf{R}^{-1} = \mathbf{R}^\intercal$.

7.1.2 Dynamics

Dynamics deals with the forces and torques that generate the accelerations. For simplicity they are calculated in the body frame and are given in function of the velocity vector $\boldsymbol{\nu}$. Fossen [42] proposes the following equation for the dynamics:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{g_0} + \mathbf{w} \tag{7.4}$$

where

$\mathbf{M}$	- system inertia matrix (including added mass)
$\mathbf{C}(\boldsymbol{\nu})$	- Coriolis-centripetal matrix (including added mass)
$\mathbf{D}(\boldsymbol{\nu})$	- damping matrix
$\mathbf{g}(\boldsymbol{\eta})$	- vector of gravitational/buoyancy forces and torques
$\boldsymbol{\tau}$	- vector of control inputs
$\mathbf{g_0}$	- vector used for pretrimming (ballast control)
$\mathbf{w}$	- vector of environmental disturbances (wind, waves and currents)

The terms in (7.4) take into account the inertial and non-inertial contributions from the rigid body of the vehicle and the contributions from the fluid around the vehicle.

Rigid body

Rigid body dynamics must take into account the Coriolis and centrifugal effects brought in by the fact that the body frame is not an inertial one:

$$\mathbf{M_{RB}}\dot{\boldsymbol{\nu}} + \mathbf{C_{RB}}\boldsymbol{\nu} = \boldsymbol{\tau}_{\mathbf{hyd}}. \tag{7.5}$$

Here $\mathbf{M_{RB}}$ is the rigid body inertia matrix, $\mathbf{C_{RB}}$ is the rigid body Coriolis and centripetal matrix and $\boldsymbol{\tau}_{\mathbf{hyd}}$ is the generalized vector of external forces and torques. The matrices $\mathbf{M_{RB}}$ and $\mathbf{C_{RB}}$ are shown in equations (7.6) and (7.7). Notice that $\mathbf{M_{RB}}$ does not depend of the vector state $\boldsymbol{\nu}$.

$$\mathbf{M_{RB}} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{7.6}$$

$$\mathbf{C_{RB}} = \begin{bmatrix} 0 & 0 & 0 & m(y_g q + z_g r) & -m(x_g q - w) & -m(x_g r + v) \\ 0 & 0 & 0 & -m(y_g p + w) & m(z_g r + x_g p) & -m(y_g r - u) \\ 0 & 0 & 0 & -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g q) \\ -m(y_g q + z_g r) & m(y_g p + w) & m(z_g p - v) & 0 & -I_{yz} q - I_{xz} p + I_z r & I_{yz} r + I_{xy} p - I_y q \\ m(x_g q - w) & -m(z_g r + x_g p) & m(z_g q + u) & I_{yz} q + I_{xz} p - I_z r & 0 & -I_{xz} r - I_{xy} q + I_x p \\ m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) & -I_{yz} r - I_{xy} p + I_y q & I_{xz} r + I_{xy} q - I_x p & 0 \end{bmatrix} \tag{7.7}$$

where m is the vehicle mass, I the 3×3 inertia tensor, and (x_g, y_g, z_g) is the center of gravity.

Hydrodynamic forces and torques

The hydrodynamic forces and torques induced by the water are complicated to calculate because it is a fluid in an open space and, as said in Triantafyllou and Hover, "*the method of hydrodynamic coefficients is a somewhat blind series expansion of the fluid force in an attempt to provide a framework on which to base experiments and calculations to evaluate these terms.*" [147, Chap. 3]. Two types of parametrizations for the hydrodynamics forces are generally used in classical modeling theory: a truncated Taylor-series expansion, proposed by Davison and Shiff [27] using first order terms, and later on Abkowitz [1] using terms up to third order; and the so-called *second order modulus terms* proposed by Fedayevsky and Sobolev [40], and later by Norrbin [95]. Perez [100] calculates these forces as the sum of 5 components,

$$\boldsymbol{\tau}_{\mathbf{hyd}} = \boldsymbol{\tau}_{1\mathrm{w}} + \boldsymbol{\tau}_{2\mathrm{w}} + \boldsymbol{\tau}_{\mathrm{r}} + \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{\tau}_{\mathrm{hs}} \tag{7.8}$$

where

- First-order wave excitation forces ($\boldsymbol{\tau}_{1\mathrm{w}}$): These forces are the zero-mean oscillatory forces caused by the waves.

- Second-order wave excitation forces ($\boldsymbol{\tau}_{2\mathrm{w}}$): These forces include mean wave-drift loads, slowly varying (difference frequencies) and rapidly varying (sum frequencies) wave loads.

- Radiation forces ($\boldsymbol{\tau}_{\mathrm{r}}$): These forces appear as a consequence of the change in the momentum of the fluid and the waves generated due to the motion of the hull. These forces are proportional to the accelerations and velocities of the ship. Due to this, the radiation forces are separated into the so-called *added-mass forces* (forces proportional to accelerations) and *potential-damping forces* (forces proportional to velocities).

- Viscous forces ($\boldsymbol{\tau}_{\mathrm{v}}$): These are nonlinear damping forces that appear due to nonlinear non-conservative phenomena by which kinetic energy of the hull is transferred to the fluid due to viscous effects (skin friction, flow separation and eddy making).

- Hydrostatic forces ($\boldsymbol{\tau}_{\mathrm{hs}}$): These are restoring forces due to gravity and buoyancy that tend to preserve the equilibrium of the ship.

Another way to represent this sum of forces is as follows (Fossen [43]):

$$\boldsymbol{\tau}_{\mathbf{hyd}} = -\underbrace{\mathbf{M_A}\dot{\boldsymbol{\nu}} - \mathbf{C_A}\boldsymbol{\nu}}_{\text{added mass}} - \underbrace{\mathbf{D_p}\boldsymbol{\nu}}_{\text{damping}} - \underbrace{\mathbf{g}(\boldsymbol{\eta}) + \mathbf{g_0}}_{\text{restoring forces}} + \underbrace{\boldsymbol{\tau}}_{\text{c. inputs}} + \underbrace{\mathbf{w}}_{\text{disturbances}} \tag{7.9}$$

where the matrices $\mathbf{M_A}$ and $\mathbf{C_A}$ represent the added mass and are described in the equations (7.11) - (7.13). The matrix $\mathbf{D_p}$ is the potential damping. Some authors (see Fossen [43]) represent it as the sum of the linear damping $\mathbf{D_l}$ and nonlinear damping $\mathbf{D_n}$, while others (see Prestero [105]) represent it only with the nonlinear damping matrix. The linear damping is described in equation (7.14). The nonlinear damping is a matrix whose coefficients depend on the Reynolds number (see Norrbin [95]). The matrix $\mathbf{g}(\boldsymbol{\eta})$ is described in equation (7.15). Notice that the matrices $\mathbf{M_A}$ and $\mathbf{D_l}$ do not depend on the vector state $\boldsymbol{\nu}$.

All of these matrices depend on several coefficients, whose strict definition is the partial derivative of a force (say X, Y or Z) or a torque (say K, M, or N) that actuate in the vehicle respect a velocity or an acceleration evaluated at the origin, such as:

$$X_{\dot{u}} = \left.\frac{\partial X}{\partial \dot{u}}\right|_{\dot{u}=0}, \qquad X_u = \left.\frac{\partial X}{\partial u}\right|_{u=0}, \qquad X_{|u|u} = \left.\frac{\partial^2 X}{\partial u \partial |u|}\right|_{u=0}. \tag{7.10}$$

Figure 7.2 shows a typical relation between the force X and the forward velocity u. In the absence of any propeller thrust, the X force decreases, due to increasing drag, if the velocity increases.

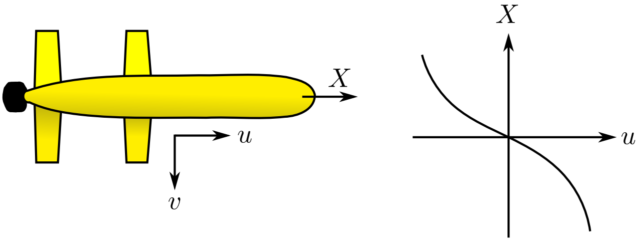


Figure 7.2: Plot of X vs. u

$$\mathbf{M_A} = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{7.11}$$

$$\mathbf{C_A} = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0-b_3 & b_2 & \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \tag{7.12}$$

where

$$\begin{aligned}
a_1 &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\
a_2 &= Y_{\dot{u}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\
a_3 &= Z_{\dot{u}}u + Z_{\dot{v}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\
b_1 &= K_{\dot{u}}u + K_{\dot{v}}v + K_{\dot{w}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \\
b_2 &= M_{\dot{u}}u + M_{\dot{v}}v + M_{\dot{w}}w + M_{\dot{p}}p + M_{\dot{q}}q + M_{\dot{r}}r \\
b_3 &= N_{\dot{u}}u + N_{\dot{v}}v + N_{\dot{w}}w + N_{\dot{p}}p + N_{\dot{q}}q + N_{\dot{r}}r
\end{aligned} \tag{7.13}$$

$$\mathbf{D_l} = -\begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ K_u & K_v & K_w & K_p & K_q & K_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix} \tag{7.14}$$

$$\mathbf{g}(\boldsymbol{\eta}) = -\begin{bmatrix} (W-B)\sin\theta & \\ -(W-B)\cos\theta\sin\phi & \\ -(W-B)\cos\theta\cos\phi & \\ -(y_gW - y_bB)\cos\theta\cos\phi & +(z_gW - z_bB)cos\theta\sin\phi \\ (z_gW - z_bB)\sin\theta & +(x_gW - x_bB)\cos\theta\cos\phi \\ -(x_gW - x_bB)\cos\theta\sin\phi & -(y_gW - y_bB)\sin\theta \end{bmatrix} \tag{7.15}$$

where W is the vehicle weight, B is the buoyancy of the vehicle, (x_g, y_g, z_g) is the center of gravity and (x_b, y_b, z_b) is the center of buoyancy.

7.1.3 Simplified 3 DoF horizontal model

The Guanay II AUV moves on the surface, and thus the model can be simplified to 3 DoF: surge, sway and yaw; and the movement in the other coordinates can be disregarded. Thus, the position and velocity vectors become:

$$\boldsymbol{\eta} = \begin{bmatrix} n \\ e \\ \psi \end{bmatrix}, \qquad \boldsymbol{\nu} = \begin{bmatrix} u \\ v \\ r \end{bmatrix}. \tag{7.16}$$

Assuming that y_g is zero, the matrices of rigid body can be simplified as follows:

$$\mathbf{M_{RB}} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{7.17}$$

$$\mathbf{C_{RB}} = \begin{bmatrix} 0 & 0 & -m(x_gr + v) \\ 0 & 0 & mu \\ m(x_gr + v) & -mu & 0 \end{bmatrix} \tag{7.18}$$

On the other hand, due to xz-plane symmetry, several coefficients can be neglected, leaving only the coefficients of the force X with respect to the forward velocity, and the coefficients of the components Y and N with respect to the velocities v and r. Additionally, neglecting the linear damping matrix $\mathbf{D_l}$ as does Prestero [105], the matrices of hydrodynamic forces and torques are simplified as follows:

$$\mathbf{M_A} = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix} \tag{7.19}$$

$$\mathbf{C_A} = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -(Y_{\dot{v}}v + Y_{\dot{r}}r) & X_{\dot{u}}u & 0 \end{bmatrix} \tag{7.20}$$

$$\mathbf{D_n} = -\begin{bmatrix} X_{|u|u}|u| & 0 & 0 \\ 0 & Y_{|v|v}|v| & Y_{|r|r}|r| \\ 0 & N_{|v|v}|v| & N_{|r|r}|r| \end{bmatrix} \tag{7.21}$$

Notice that the matrix $\mathbf{g}(\boldsymbol{\eta}) = [0\ 0\ 0]^\intercal$ since $\phi = \theta = 0$.

In the case of the GUANAY II, the control inputs $\boldsymbol{\tau}$ consist of a force for surge movement using the three thrusters, and a torque for yaw movement using the lateral thrusters:

$$\boldsymbol{\tau} = \begin{bmatrix} \text{Prop} \\ 0 \\ \text{Torque} \end{bmatrix} \qquad \begin{aligned} \text{Prop} &= X_{\text{main}} + X_{\text{lft}} + X_{\text{rgt}} \\ \text{Torque} &= a_{\text{fin}}(X_{\text{lft}} - X_{\text{rgt}}) \end{aligned} \tag{7.22}$$

where a_{fin} is the distance from the lateral thrusters to the central axis, and X_{main}, X_{lft} and X_{rgt} are the forces of the main, left and right thrusters, respectively.

Finally, from equations (7.4), (7.5) and (7.8), and assuming that the disturbances $\mathbf{w}$ are small, the dynamics of the vehicle is described as follows:

$$(\mathbf{M_{RB}} + \mathbf{M_A})\dot{\boldsymbol{\nu}} + (\mathbf{C_{RB}} + \mathbf{C_A})\boldsymbol{\nu} + \mathbf{D_n}\boldsymbol{\nu} = \boldsymbol{\tau} \tag{7.23}$$

7.1.4 Stability in forward motion

Stability of an equilibrium situation means that the system returns to the original condition of equilibrium after a disturbance of the smallest amount. Considering the case of forward movement without the deflection of control surfaces or any application of torque, a vehicle that can maintain the straight line after a disturbance is stable, and is unstable if it cannot return to the forward movement situation (see figure 7.3). The *stability parameter* C defined by

$$C = Y_v(N_r - mx_g u) + N_v(mu - Y_r) \tag{7.24}$$

is commonly used. [1] [147]

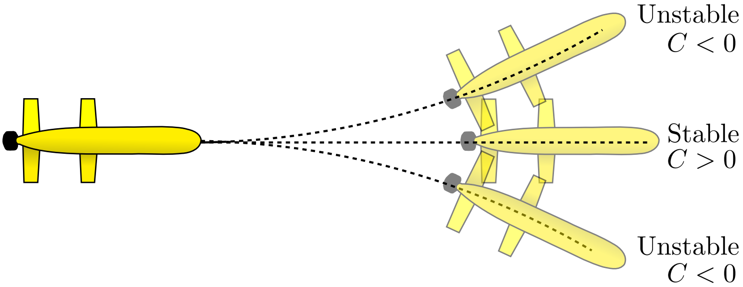


Figure 7.3: Stability in forward motion

7.2 Guanay II AUV

In order to calculate the vehicle coefficients we must first define the profile of the vehicle, determine its mass and buoyancy, and finally identify the necessary control fin parameters.

7.2.1 Vehicle profile

The external structure of the vehicle has been designed using the Myring equations [91], which yield good navigation performance for given dimensions. These equations are based on the following parameters (see Fig. 7.4):

- a, bow length;
- b, mid-body length. This section has constant radius along x-axis;
- c, stern length;
- n, exponential parameter which can be varied to give different body shapes;
- 2θ, angle of the tip of the stern;
- d, maximum hull diameter.

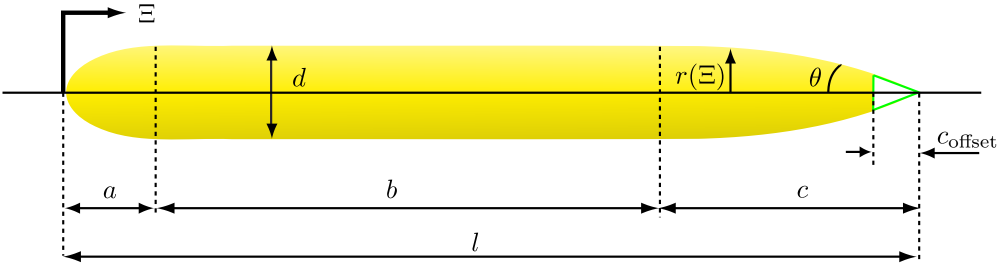


Figure 7.4: Myring profile: vehicle hull radius as a function of axial position

These equations assume that the origin is in the forward end. The radii that define the hull at the bow are given by

$$r(\Xi) = \frac{1}{2}d\left[1 - \left(\frac{\Xi - a}{a}\right)^2\right]^{\frac{1}{n}} \tag{7.25}$$

where $r(\Xi)$ are the radii for the different distances, and Ξ is the distance measured from the forward end of the vehicle. Also, the stern profile is determined by

$$r(\Xi) = \frac{1}{2}d - \left[\frac{3d}{2c^2} - \frac{\tan\theta}{c}\right](\Xi - l_f)^2 + \left[\frac{d}{c^3} - \frac{\tan\theta}{c^2}\right](\Xi - l_f)^3 \tag{7.26}$$

Guanay II AUV is based on the Myring B hull contour, with the dimensions of all the parameters shown in table 7.1.

Table 7.1: Myring parameters for GUANAY II

Parameter	Value	Units	Description
a	0.325	m	Bow length
b	1.12	m	Midbody length
c	0.924	m	Stern length
c_{offset}	0.07	m	Stern offset
n	2	n/a	Exponential Coefficient
θ	0.436	rad	Included stern angle
d	0.326	m	Maximum hull diameter
l	2.3	m	Vehicle total length

7.2.2 Vehicle mass

From equation (3.6) we know that the vehicle mass is 94.8 kg. But, as the vehicle is immersed and is filled with water, it is necessary to take into account the water between the watertight module and the hull structure. The amount of water is calculated as follows.

The volume inside the hull structure can be calculated with the radii of the equations (7.25) and (7.26) subtracting 5 mm of thickness:

$$V_{\text{hull}} = \int_{x_{\text{bow}}}^{x_{\text{tail}}} \pi(r(x) - 0.005)^2 dx = 0.1473\,\text{m}^3. \tag{7.27}$$

The watertight module consist of a cylinder of 0.25 m in diameter and 1.09 m in length:

$$V_{\text{cyl}} = \pi r^2 h = 0.0535\,\text{m}^3. \tag{7.28}$$

The foam, as described in section 3.1 consists of three blocks of 0.00111 m^3, 0.00458 m^3 and 0.01132 m^3. Finally, we must take into account 5.2 liters of cables and some small pieces, *i.e.* 0.0052 m^3. Thereby, the total vehicle mass taking into account the inside water is

$$\begin{aligned} V_{\text{water}} &= V_{\text{hull}} - V_{\text{cyl}} - V_{\text{foams}} - V_{\text{others}} \\ &= 0.1473 - 0.0535 - 0.017 - 0.0052 \\ &= 0.0716\,\text{m}^3 \end{aligned} \qquad \begin{aligned} m &= 94.8 + m_{\text{water}} \\ &= 94.8 + \rho_{\text{water}} V_{\text{water}} \\ &= 94.8 + 1025(0.0716), \end{aligned}$$

that is

$$m = 168.2\,\text{kg}. \tag{7.29}$$

7.2.3 Centers of buoyancy, gravity and inertia tensor

The GUANAY II AUV was modeled with the CAD software NX, which allows to define the different position of the components and their weights. The tools allow the computation of the centers of buoyancy and gravity, and the inertia tensor. The center of gravity can vary because it is often necessary to change some parts and re-ballast the vehicle. The average values are given in tables 7.2 and 7.3, and the value of the inertia tensor is

$$I_z = 52.7\,\text{kg}{\cdot}\text{m}^2. \tag{7.30}$$

Table 7.2: Center of buoyancy wrt origin at vehicle nose

Parameter	Value	Unit
x_b	1.21	m
y_b	0	m
z_b	0	m

Table 7.3: Center of gravity wrt origin at CB

Parameter	Value	Unit
x_g	0	m
y_g	0	m
z_g	0.01	m

7.3 Linear system

Control design for non-linear models, although well developed in the literature, is inherently difficult to use and sometimes yields very complicated and unintuitive control laws which are not easily grasped and tuned, even for experienced engineers. An alternative is the linearization of the model around a working point, which yields a linear model whose behaviour is not very different from that of the full, non-linear one, provided that the system state is sufficiently close to the working point. In our case, the idea is to define a linear system around a fixed velocity vector and consider only small deviations from it.

The purpose of this section is to linearize the model using matrix notation, and to discuss the need to include the linear damping matrix in order to better exploit the dynamics in the horizontal plane.

Let $\boldsymbol{\nu}_0$ be the constant velocity of the vehicle in forward movement:

$$\boldsymbol{\nu}_0 = \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} \tag{7.31}$$

and $\Delta\boldsymbol{\nu}$ the small variations around $\boldsymbol{\nu}_0$

$$\Delta\boldsymbol{\nu} = \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta r \end{bmatrix}. \tag{7.32}$$

The linearized model can be calculated from the equation (7.23) as

$$(\mathbf{M_{RB}} + \mathbf{M_A})\Delta\dot{\boldsymbol{\nu}} + \left.\frac{\partial \mathbf{f_c}}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}_0} \Delta\boldsymbol{\nu} + \left.\frac{\partial \mathbf{f_d}}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}_0} \Delta\boldsymbol{\nu} = \Delta\boldsymbol{\tau} \tag{7.33}$$

where

$$\mathbf{f_c} = (\mathbf{C_{RB}} + \mathbf{C_A})\boldsymbol{\nu}, \qquad \mathbf{f_d} = \mathbf{D_n}\boldsymbol{\nu}. \tag{7.34}$$

The calculation using these equations implies to solve the vectors $\mathbf{f_c}$ and $\mathbf{f_d}$ before to apply the partial derivatives, which can be calculated easily for this case. However, it would be interesting to use a tool to break down that derivative before solve these vectors.

Definition 1 *If* $\mathbf{x} = [x_1\ x_2\ \dots\ x_n]^T$ *is a vector of* n *variables and* $\mathbf{F}(\mathbf{x})$ *is a matrix of* $p \times q$ *elements that are a function of* $\mathbf{x}$*, we define the partial derivative of* $\mathbf{F}(\mathbf{x})$ *with respect to* $\mathbf{x}$ *as*

$$\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}} = \left[\begin{array}{cccc} \frac{\partial \mathbf{F}(\mathbf{x})}{\partial x_1} & \frac{\partial \mathbf{F}(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial \mathbf{F}(\mathbf{x})}{\partial x_n} \end{array}\right] \tag{7.35}$$

Notice that each $\dfrac{\partial \mathbf{F}(\mathbf{x})}{\partial x_i}$ is a $p \times q$ matrix.

Theorem 1 *Let* $\mathbf{x} = [x_1\ x_2\ \dots\ x_n]^T$ *be a vector of* n *variables,* $\mathbf{H}(\mathbf{x})$ *a vector of* q *elements dependent on* $\mathbf{x}$*, and* $\mathbf{F}(\mathbf{x})$ *a matrix of* $p \times q$ *elements also dependent on* $\mathbf{x}$*. Then*

$$\frac{\partial (\mathbf{FH})}{\partial \mathbf{x}} = \mathbf{F}\frac{\partial \mathbf{H}}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}(I_n \otimes \mathbf{H}) \tag{7.36}$$

where I_n *is the identity matrix of* n *elements and* $\otimes$ *is the Kronecker product.*

Proof 1 *The multiplication* $\mathbf{FH}$ *by definition is*

$$\begin{aligned} \mathbf{FH} &= \begin{bmatrix} f_{11} & \cdots & f_{1q} \\ \vdots & \ddots & \vdots \\ f_{p1} & \cdots & f_{pq} \end{bmatrix} \begin{bmatrix} h_1 \\ \vdots \\ h_q \end{bmatrix} \\ &= \begin{bmatrix} f_{11}h_1 + \cdots + f_{1q}h_q \\ \vdots \\ f_{p1}h_1 + \cdots + f_{pq}h_q \end{bmatrix} \end{aligned}$$

The partial derivative (or Jacobian matrix) of this multiplication is

$$\frac{\partial (\mathbf{FH})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial (f_{11}h_1 + \cdots + f_{1q}h_q)}{\partial x_1} & \cdots & \frac{\partial (f_{11}h_1 + \cdots + f_{1q}h_q)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial (f_{p1}h_1 + \cdots + f_{pq}h_q)}{\partial x_1} & \cdots & \frac{\partial (f_{p1}h_1 + \cdots + f_{pq}h_q)}{\partial x_n} \end{bmatrix}$$

Using the multiplicative and additive properties of the partial derivatives we have:

$$\frac{\partial (\mathbf{FH})}{\partial \mathbf{x}} = \begin{bmatrix} \begin{pmatrix} f_{11}\frac{\partial h_1}{\partial x_1} + h_1\frac{\partial f_{11}}{\partial x_1} + \\ + \cdots + \\ + f_{1q}\frac{\partial h_q}{\partial x_1} + h_q\frac{\partial f_{1q}}{\partial x_1} \end{pmatrix} & \cdots & \begin{pmatrix} f_{11}\frac{\partial h_1}{\partial x_n} + h_1\frac{\partial f_{11}}{\partial x_n} + \\ + \cdots + \\ + f_{1q}\frac{\partial h_q}{\partial x_n} + h_q\frac{\partial f_{1q}}{\partial x_n} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ \begin{pmatrix} f_{p1}\frac{\partial h_1}{\partial x_1} + h_1\frac{\partial f_{p1}}{\partial x_1} + \\ + \cdots + \\ + f_{pq}\frac{\partial h_q}{\partial x_1} + h_q\frac{\partial f_{pq}}{\partial x_1} \end{pmatrix} & \cdots & \begin{pmatrix} f_{p1}\frac{\partial h_1}{\partial x_n} + h_1\frac{\partial f_{p1}}{\partial x_n} + \\ + \cdots + \\ + f_{pq}\frac{\partial h_q}{\partial x_n} + h_q\frac{\partial f_{pq}}{\partial x_n} \end{pmatrix} \end{bmatrix}$$

$$\begin{aligned} \frac{\partial (\mathbf{FH})}{\partial \mathbf{x}} = &\begin{bmatrix} (f_{11}\frac{\partial h_1}{\partial x_1} + \cdots + f_{1q}\frac{\partial h_q}{\partial x_1}) & \cdots & (f_{11}\frac{\partial h_1}{\partial x_n} + \cdots + f_{1q}\frac{\partial h_q}{\partial x_n}) \\ \vdots & \ddots & \vdots \\ (f_{p1}\frac{\partial h_1}{\partial x_1} + \cdots + f_{pq}\frac{\partial h_q}{\partial x_1}) & \cdots & (f_{p1}\frac{\partial h_1}{\partial x_n} + \cdots + f_{pq}\frac{\partial h_q}{\partial x_n}) \end{bmatrix} + \\ &+ \begin{bmatrix} (h_1\frac{\partial f_{11}}{\partial x_1} + \cdots + h_q\frac{\partial f_{1q}}{\partial x_1}) & \cdots & (h_1\frac{\partial f_{11}}{\partial x_n} + \cdots + h_q\frac{\partial f_{1q}}{\partial x_n}) \\ \vdots & \ddots & \vdots \\ (h_1\frac{\partial f_{p1}}{\partial x_1} + \cdots + h_q\frac{\partial f_{pq}}{\partial x_1}) & \cdots & (h_1\frac{\partial f_{p1}}{\partial x_n} + \cdots + h_q\frac{\partial f_{pq}}{\partial x_n}) \end{bmatrix} \end{aligned}$$

$$\frac{\partial(\mathbf{FH})}{\partial \mathbf{x}} = \underbrace{\begin{bmatrix} f_{11} & \cdots & f_{1q} \\ \vdots & \ddots & \vdots \\ f_{p1} & \cdots & f_{pq} \end{bmatrix}}_{p\times q} \underbrace{\begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1} & \cdots & \frac{\partial h_q}{\partial x_n} \end{bmatrix}}_{q\times n} +$$

$$+ \underbrace{\begin{bmatrix} \frac{\partial f_{11}}{\partial x_1} & \cdots & \frac{\partial f_{1q}}{\partial x_1} & \cdots & \frac{\partial f_{11}}{\partial x_n} & \cdots & \frac{\partial f_{1q}}{\partial x_n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial f_{p1}}{\partial x_1} & \cdots & \frac{\partial f_{pq}}{\partial x_1} & \cdots & \frac{\partial f_{p1}}{\partial x_n} & \cdots & \frac{\partial f_{pq}}{\partial x_n} \end{bmatrix}}_{p\times qn} \underbrace{\begin{bmatrix} h_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ h_q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h_1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h_q \end{bmatrix}}_{qn\times n}$$

$$\begin{aligned} \frac{\partial(\mathbf{FH})}{\partial \mathbf{x}} =& \mathbf{F}\frac{\partial \mathbf{H}}{\partial \mathbf{x}} + \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial x_1} & \cdots & \frac{\partial \mathbf{F}}{\partial x_n} \end{bmatrix} \begin{bmatrix} \mathbf{H} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{H} \end{bmatrix} \\ \frac{\partial(\mathbf{FH})}{\partial \mathbf{x}} =& \mathbf{F}\frac{\partial \mathbf{H}}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}(I_n \otimes \mathbf{H}) \end{aligned}$$

The proof is complete. ■

It is important to note that this result can be used to linearize the hydrodynamical model of the vehicle in a modular way, which is very useful since the coefficients are highly coupled, for example in the model with 6DoF.

If we apply this result to the equations (7.34) we have

$$\begin{aligned} \frac{\partial \mathbf{f_c}}{\partial \boldsymbol{\nu}} &= \frac{\partial(\mathbf{C_{RB}}\boldsymbol{\nu})}{\partial \boldsymbol{\nu}} + \frac{\partial(\mathbf{C_A}\boldsymbol{\nu})}{\partial \boldsymbol{\nu}} \\ \frac{\partial \mathbf{f_c}}{\partial \boldsymbol{\nu}} &= \mathbf{C_{RB}} + \frac{\partial \mathbf{C_{RB}}}{\partial \boldsymbol{\nu}}(I_3 \otimes \boldsymbol{\nu}) + \mathbf{C_A} + \frac{\partial \mathbf{C_A}}{\partial \boldsymbol{\nu}}(I_3 \otimes \boldsymbol{\nu}) \\ \frac{\partial \mathbf{f_c}}{\partial \boldsymbol{\nu}} &= \mathbf{C_{RB}} + \mathbf{C_A} + \left(\frac{\partial \mathbf{C_{RB}}}{\partial \boldsymbol{\nu}} + \frac{\partial \mathbf{C_A}}{\partial \boldsymbol{\nu}}\right)(I_3 \otimes \boldsymbol{\nu}) \end{aligned} \tag{7.37}$$

and

$$\begin{aligned} \frac{\partial \mathbf{f_d}}{\partial \boldsymbol{\nu}} &= \frac{\partial(\mathbf{D_n}\boldsymbol{\nu})}{\partial \boldsymbol{\nu}} \\ \frac{\partial \mathbf{f_d}}{\partial \boldsymbol{\nu}} &= \mathbf{D_n} + \frac{\partial \mathbf{D_n}}{\partial \boldsymbol{\nu}}(I_3 \otimes \boldsymbol{\nu}) \end{aligned} \tag{7.38}$$

where

$$\begin{aligned}
\frac{\partial \mathbf{C_{RB}}}{\partial \boldsymbol{\nu}} &= \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & -m & 0 & 0 & -mx_g \\ 0 & 0 & m & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -m & 0 & m & 0 & 0 & mx_g & 0 & 0 \end{array}\right] \\
\frac{\partial \mathbf{C_{A}}}{\partial \boldsymbol{\nu}} &= \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & Y_{\dot{v}} & 0 & 0 & Y_{\dot{r}} \\ 0 & 0 & -X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & X_{\dot{u}} & 0 & -Y_{\dot{v}} & 0 & 0 & -Y_{\dot{r}} & 0 & 0 \end{array}\right] \\
\frac{\partial \mathbf{D_{n}}}{\partial \boldsymbol{\nu}} &= -\left[\begin{array}{ccc|ccc|ccc} X_{|u|u}\mathrm{sgn}(u) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Y_{|v|v}\mathrm{sgn}(v) & 0 & 0 & 0 & Y_{|r|r}\mathrm{sgn}(r) \\ 0 & 0 & 0 & 0 & N_{|v|v}\mathrm{sgn}(v) & 0 & 0 & 0 & N_{|r|r}\mathrm{sgn}(r) \end{array}\right]
\end{aligned} \tag{7.39}$$

Using the Laplace transform, the linearized model 7.33 can be writen as:

$$\begin{aligned}
&\Bigg((\mathbf{M_{RB}} + \mathbf{M_{A}})s + \mathbf{C_{RB}} + \mathbf{C_{A}} + \mathbf{D_{n}} + \\
&\qquad + \left(\frac{\partial \mathbf{C_{RB}}}{\partial \boldsymbol{\nu}} + \frac{\partial \mathbf{C_{A}}}{\partial \boldsymbol{\nu}} + \frac{\partial \mathbf{D_{n}}}{\partial \boldsymbol{\nu}}\right)(I_3 \otimes \boldsymbol{\nu})\Bigg)\Bigg|_{\boldsymbol{\nu}_0} \boldsymbol{\nu}(s) = \boldsymbol{\tau}(s)
\end{aligned} \tag{7.40}$$

Solving for the matrices we have

$$\left[\begin{array}{ccc} m_u s - 2|u|X_{|u|u} & -m_v r & -m_v v - 2m_{Y_{\dot{r}}} r \\ m_u r & m_v s - 2|v|Y_{|v|v} & m_{Y_{\dot{r}}} s + m_u u - 2|r|Y_{|r|r} \\ -m_{uv} v + m_{Y_{\dot{r}}} r & m_{N_{\dot{v}}} s - m_{uv} u - 2|v|N_{|v|v} & m_r s + m_{Y_{\dot{r}}} u - 2|r|N_{|r|r} \end{array}\right]_{\boldsymbol{\nu}_0} \boldsymbol{\nu}(s) = \boldsymbol{\tau}(s) \tag{7.41}$$

$$\left[\begin{array}{ccc} m_u s - 2|u_0|X_{|u|u} & 0 & 0 \\ 0 & m_v s & m_{Y_{\dot{r}}} s + m_u u_0 \\ 0 & m_{N_{\dot{v}}} s - m_{uv} u_0 & m_r s + m_{Y_{\dot{r}}} u_0 \end{array}\right] \boldsymbol{\nu}(s) = \boldsymbol{\tau}(s) \tag{7.42}$$

where

$$\begin{aligned}
m_u &= m - X_{\dot{u}} \\
m_v &= m - Y_{\dot{v}} \\
m_{uv} &= m_u - m_v = Y_{\dot{v}} - X_{\dot{u}} \\
m_{Y_{\dot{r}}} &= mx_g - Y_{\dot{r}} \\
m_{N_{\dot{v}}} &= mx_g - N_{\dot{v}}
\end{aligned} \tag{7.43}$$

7.3.1 Stability

The stability can be determined calculating the poles of the linearized model 7.40. If all the poles have negative real part the system is stable, but if at least one pole has positive real part then the system is unstable. Equating the determinant of the matrix in equation (7.42) to zero we get

$$\left|\begin{array}{ccc} m_u s - 2|u_0|X_{|u|u} & 0 & 0 \\ 0 & m_v s & m_{Y_{\dot{r}}} s + m_u u_0 \\ 0 & m_{N_{\dot{v}}} s - m_{uv} u_0 & m_r s + m_{Y_{\dot{r}}} u_0 \end{array}\right| = 0$$

$$(m_u s - 2|u_0|X_{|u|u})\left(m_v s(m_r s + m_{Y_{\dot{r}}} u_0) - (m_{N_{\dot{v}}} s - m_{uv} u_0)(m_{Y_{\dot{r}}} s + m_u u_0)\right) = 0 \tag{7.44}$$

The first pole is

$$\begin{aligned} m_u s - 2|u_0|X_{|u|u} &= 0 \\ s &= 2|u_0|\frac{X_{|u|u}}{m_u}. \end{aligned} \tag{7.45}$$

In order to guaranty its negativity it is necessary that

$$\frac{X_{|u|u}}{m - X_{\dot{u}}} < 0 \tag{7.46}$$

This condition is always achieved since the dynamic drag implies the negativity of the coefficients $X_{|u|u}$ and $X_{\dot{u}}$.

The calculation of the other two poles is as follows:

$$\begin{aligned} m_v s(m_r s + m_{Y_{\dot{r}}} u_0) - (m_{N_{\dot{v}}} s - m_{uv} u_0)(m_{Y_{\dot{r}}} s + m_u u_0) &= 0 \\ (m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})s^2 + (m_{Y_{\dot{r}}}(m_{uv} + m_v) - m_{N_{\dot{v}}} m_u)u_0 s + m_u m_{uv} u_0^2 &= 0 \\ (m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})s^2 + (m_{Y_{\dot{r}}} m_u - m_{N_{\dot{v}}} m_u)u_0 s + m_u m_{uv} u_0^2 &= 0 \\ (m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})s^2 + m_u(m_{Y_{\dot{r}}} - m_{N_{\dot{v}}})u_0 s + m_u m_{uv} u_0^2 &= 0 \\ (m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})s^2 + m_u(N_{\dot{v}} - Y_{\dot{r}})u_0 s + m_u m_{uv} u_0^2 &= 0 \\ s^2 + \underbrace{\frac{m_u(N_{\dot{v}} - Y_{\dot{r}})u_0}{(m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})}}_{B} s + \underbrace{\frac{m_u m_{uv} u_0^2}{(m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})}}_{C} &= 0 \end{aligned} \tag{7.47}$$

Thus, the other two poles are

$$s_{1,2} = -\frac{B}{2} \pm \frac{\sqrt{B^2 - 4C}}{2}. \tag{7.48}$$

If $B^2 - 4C < 0$ the poles are complex conjugate. Thus the real part $-\frac{B}{2}$ determines its negativity, which is achieved if $B > 0$. Also notice, that $C > 0$ because $B^2 < 4C$.

On the other hand, if $B^2 - 4C \geq 0$ the poles are real, and they are negative if

$$\begin{aligned} -\frac{B}{2} \pm \frac{\sqrt{B^2 - 4C}}{2} &\leq 0 \\ B &\geq \pm\sqrt{B^2 - 4C} \\ B &\geq 0 \end{aligned}$$

and also

$$\begin{aligned} B &\geq \pm\sqrt{B^2 - 4C} \\ B^2 &\geq B^2 - 4C \\ C &\geq 0 \end{aligned}$$

For this reason, the stability is directly linked to the positivity of the values B and C. Take as an example the Sirene AUV [127], a vehicle developed at the "Instituto Superior Técnico (IST)". Its B value is zero while C is negative, which indicates that it is unstable and cannot maintain the straight line without the use of a torque. The coefficients of Sirene AUV are described in table 7.4.

Table 7.4: Sirene AUV coefficients (IST)

Parameter	Value	Units
m	4000	kg
I_z	2660	kg·m^2
$X_{\dot{u}}$	-290	kg
$Y_{\dot{v}}$	-310	kg
$Y_{\dot{r}}$	0	kg·m/rad
$N_{\dot{v}}$	0	kg·m
$N_{\dot{r}}$	-95	kg·m^2/rad
X_u	-360	kg/s
Y_v	-420	kg/s
Y_r	0	kg·m/rad·s
N_v	0	kg·m/s
N_r	-110	kg·m^2/rad·s
$X_{\|u\|u}$	-805	kg/m
$Y_{\|v\|v}$	-1930	kg/m
$Y_{\|r\|r}$	0	kg·m/rad^2
$N_{\|v\|v}$	0	kg
$N_{\|r\|r}$	-555	kg·m^2/rad^2

7.3.2 Transfer functions

The transfer functions relate the velocity vector $\boldsymbol{\nu}(s)$ to the inputs $\boldsymbol{\tau}(s)$ in the complex domain s. From equation (7.42) we can extract:

$$\frac{\boldsymbol{\nu}(s)}{\boldsymbol{\tau}(s)} = \begin{bmatrix} m_u s - 2|u_0|X_{|u|u} & 0 & 0 \\ 0 & m_v s & m_{Y_{\dot{r}}} s + m_u u_0 \\ 0 & m_{N_{\dot{v}}} s - m_{uv} u_0 & m_r s + m_{Y_{\dot{r}}} u_0 \end{bmatrix}^{-1}$$

$$\frac{\boldsymbol{\nu}(s)}{\boldsymbol{\tau}(s)} = \left[\begin{array}{c:c} \dfrac{1}{m_u s - 2|u_0|X_{|u|u}} & \begin{matrix} 0 & \qquad 0 \end{matrix} \\ \hdashline \begin{matrix} 0 \\ \\ 0 \end{matrix} & \dfrac{\begin{bmatrix} m_r s + m_{Y_{\dot{r}}} u_0 & -m_{Y_{\dot{r}}} s - m_u u_0 \\ -m_{N_{\dot{v}}} s + m_{uv} u_0 & m_v s \end{bmatrix}}{(m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})s^2 + m_u(N_{\dot{v}} - Y_{\dot{r}})u_0 s + m_u m_{uv} u_0^2} \end{array}\right] \tag{7.49}$$

The transfer functions of major interest are that of the forward velocity with to respect the propulsion, and that of the angular velocity with respect to the applied torque,

$$\frac{u(s)}{\mathrm{Prop}(s)} = \frac{1}{m_u s - 2|u_0|X_{|u|u}} \tag{7.50}$$

$$\frac{r(s)}{\mathrm{Torque}(s)} = \frac{m_v s}{(m_v m_r - m_{Y_{\dot{r}}} m_{N_{\dot{v}}})s^2 + m_u(N_{\dot{v}} - Y_{\dot{r}})u_0 s + m_u m_{uv} u_0^2} \tag{7.51}$$

They can be used to study the behavior of the vehicle near a specific forward velocity u_0. Figure 7.5 shows the performance of the Sirene AUV for the u velocity. The model was linearized at 0.5 m/s and simulated in two cases, namely step response and response to a sinusoidal signal. The nominal propulsion applied is 200 N, which is the value needed to reach u_0. On the other hand, Figure 7.6 shows similar simulations for the angular velocity r, linearizing the model at 0 m/s and using only the torque input.

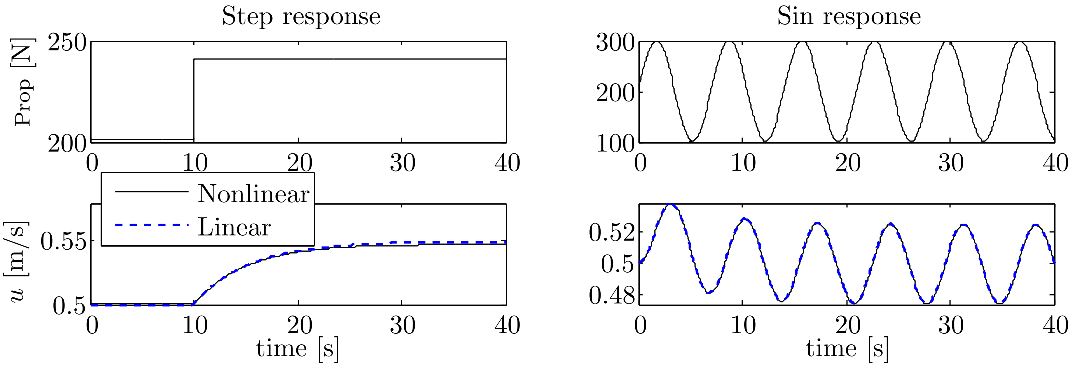


Figure 7.5: u velocity in Sirene AUV: non-linear and linear model at u_0 =0.5 m/s

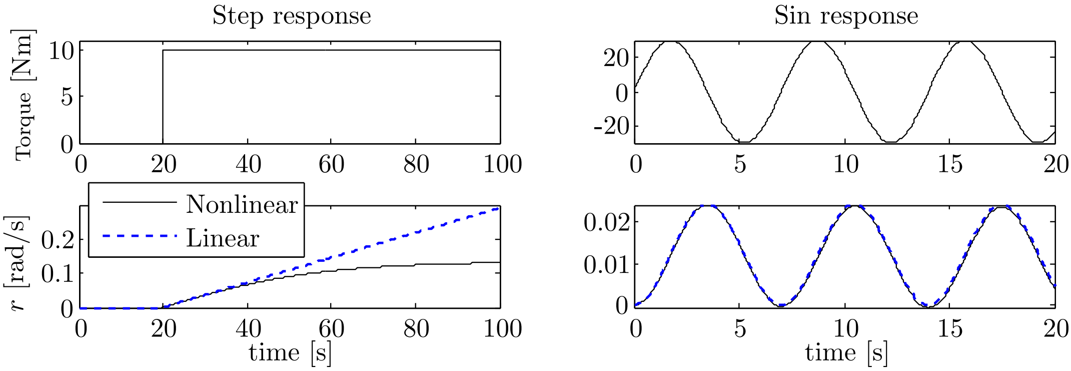


Figure 7.6: r velocity in Sirene AUV: non-linear and linear model at u_0 =0 m/s

Problem with the transfer function $\frac{r(s)}{\mathbf{Torque}(s)}$

The final value theorem establishes that the asymptotic time value for a function $F(s)$ with all the poles in the left half-plane can be calculated as

$$f(\infty) = \lim_{s \to 0} sF(s) \tag{7.52}$$

In the case of the vehicle dynamics, calculating the variations in the angular velocity r by a step in the torque, and supposing the poles in the left half-plane, we have

$$\begin{aligned}\Delta r_{(t=\infty)} &= \lim_{s\to 0} s\left(\frac{r(s)}{\text{Torque}(s)}\text{Torque}(s)\right)\\ &= \lim_{s\to 0} s\frac{m_v s}{(m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}})s^2 + m_u(N_{\dot v} - Y_{\dot r})u_0 s + m_u m_{uv} u_0^2}\cdot\frac{1}{s}\\ &= \lim_{s\to 0}\frac{m_v s}{(m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}})s^2 + m_u(N_{\dot v} - Y_{\dot r})u_0 s + m_u m_{uv} u_0^2}\end{aligned}$$

$$\Delta r_{(t=\infty)} = \begin{cases}\infty & \text{if } u_0 = 0\\ 0 & \text{if } u_0 \neq 0\end{cases} \tag{7.53}$$

If we assume that a constant torque is applied, the angular velocity should tend to a constant value greater than zero because it is rotating. However, the above result indicates that the angular velocity tends to zero or infinity. This is because the linear system is a good approximation to the non-linear one near the nominal velocity of linearization, but it is not accurate under other conditions. Notice that the result is due to the transfer function has a zero located at the origin. This problem can be seen in the figure 7.6, where the linear model tends to infinity when a step is applied.

7.4 Linear system 2: Adding linear damping coefficients

The linear system showed in the last section was extracted using the equations proposed by Prestero [105] since he uses a vehicle with similar characteristics to Guanay II AUV. However, problems with the transfer function of the angular velocity have shown that the dynamic model is not well approximated in the linearization at the xy-plane. In order to solve this, in this section we include a linear damping coefficient, as recommend Fossen [43].

Simplifying the model to 3 DoF and neglecting some coefficients due the xz-plane symmetry, the matrix 7.14 can be represented as

$$\mathbf{D_l} = -\begin{bmatrix} X_u & 0 & 0\\ 0 & Y_v & Y_r\\ 0 & N_v & N_r\end{bmatrix} \tag{7.54}$$

The model with 3 DoF is represented as

$$(\mathbf{M_{RB}} + \mathbf{M_A})\dot{\boldsymbol{\nu}} + (\mathbf{C_{RB}} + \mathbf{C_A})\boldsymbol{\nu} + (\mathbf{D_l} + \mathbf{D_n})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{7.55}$$

Similar to the last section, the linearized model that is obtained is given by

$$\begin{aligned}\Bigg((\mathbf{M_{RB}} + \mathbf{M_A})s + \mathbf{C_{RB}} + \mathbf{C_A} + \mathbf{D_l} + \mathbf{D_n} +\\ \left(\frac{\partial \mathbf{C_{RB}}}{\partial \boldsymbol{\nu}} + \frac{\partial \mathbf{C_A}}{\partial \boldsymbol{\nu}} + \frac{\partial \mathbf{D_n}}{\partial \boldsymbol{\nu}}\right)(I_3 \otimes \boldsymbol{\nu})\Bigg)\Bigg|_{\boldsymbol{\nu}_0} \boldsymbol{\nu}(s) = \boldsymbol{\tau}(s)\end{aligned} \tag{7.56}$$

Solving for the matrices we get

$$\begin{bmatrix} m_u s - 2|u_0|X_{|u|u} - X_u & 0 & 0\\ 0 & m_v s - Y_v & m_{Y_{\dot r}} s + m_u u_0 - Y_r\\ 0 & m_{N_{\dot v}} s - m_{uv} u_0 - N_v & m_r s + m_{Y_{\dot r}} u_0 - N_r\end{bmatrix} \boldsymbol{\nu}(s) = \boldsymbol{\tau}(s) \tag{7.57}$$

7.4.1 Stability

The poles of the linear system 7.57 are given by

$$\begin{vmatrix} m_u s - 2|u_0|X_{|u|u} - X_u & 0 & 0 \\ 0 & m_v s - Y_v & m_{Y_{\dot r}} s + m_u u_0 - Y_r \\ 0 & m_{N_{\dot v}} s - m_{uv} u_0 - N_v & m_r s + m_{Y_{\dot r}} u_0 - N_r \end{vmatrix} = 0$$

$$\begin{aligned} &(m_u s - 2|u_0|X_{|u|u} - X_u)\big((m_v s - Y_v)(m_r s + m_{Y_{\dot r}} u_0 - N_r) + \\ &\qquad - (m_{N_{\dot v}} s - m_{uv} u_0 - N_v)(m_{Y_{\dot r}} s + m_u u_0 - Y_r)\big) = 0 \end{aligned} \tag{7.58}$$

The first pole is

$$\begin{aligned} m_u s - 2|u_0|X_{|u|u} - X_u = 0 \\ s = 2|u_0|\frac{X_{|u|u}}{m_u} + \frac{X_u}{m_u} \\ s = 2|u_0|\frac{X_{|u|u}}{m - X_{\dot u}} + \frac{X_u}{m - X_{\dot u}} \end{aligned} \tag{7.59}$$

This pole is always negative since the dynamic drag implies the negativity of the coefficients $X_{\dot u}$, X_u and $X_{|u|u}$.

The calculation of the other two poles is as follows:

$$\begin{aligned} &(m_v s - Y_v)(m_r s + m_{Y_{\dot r}} u_0 - N_r) - (m_{N_{\dot v}} s - m_{uv} u_0 - N_v)(m_{Y_{\dot r}} s + m_u u_0 - Y_r) = 0 \\ &(m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}})s^2 + \\ &+ (m_v(m_{Y_{\dot r}} u_0 - N_r) - m_r Y_v + m_{Y_{\dot r}}(m_{uv} u_0 + N_v) - m_{N_{\dot v}}(m_u u_0 - Y_r))s + \\ &- Y_v(m_{Y_{\dot r}} u_0 - N_r) + (m_{uv} u_0 + N_v)(m_u u_0 - Y_r) = 0 \\ &(m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}})s^2 + \\ &+ (m_{Y_{\dot r}} u_0(m_v + m_{uv}) - m_v N_r - m_r Y_v + m_{Y_{\dot r}} N_v - m_{N_{\dot v}}(m_u u_0 - Y_r))s + \\ &- Y_v(m_{Y_{\dot r}} u_0 - N_r) + (m_{uv} u_0 + N_v)(m_u u_0 - Y_r) = 0 \\ &(m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}})s^2 + \\ &+ (m_{Y_{\dot r}} u_0 m_u - m_v N_r - m_r Y_v + m_{Y_{\dot r}} N_v - m_{N_{\dot v}}(m_u u_0 - Y_r))s + \\ &- Y_v(m_{Y_{\dot r}} u_0 - N_r) + (m_{uv} u_0 + N_v)(m_u u_0 - Y_r) = 0 \\ &(m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}})s^2 + \\ &+ (m_u(N_{\dot v} - Y_{\dot r})u_0 - m_v N_r - m_r Y_v + m_{Y_{\dot r}} N_v + m_{N_{\dot v}} Y_r)s + \\ &- Y_v(m_{Y_{\dot r}} u_0 - N_r) + (m_{uv} u_0 + N_v)(m_u u_0 - Y_r) = 0 \\ &As^2 + Bs + C = 0 \end{aligned} \tag{7.60}$$

where

$$\begin{aligned} A &= m_v m_r - m_{Y_{\dot r}} m_{N_{\dot v}} \\ B &= m_u(N_{\dot v} - Y_{\dot r})u_0 - m_v N_r - m_r Y_v + m_{Y_{\dot r}} N_v + m_{N_{\dot v}} Y_r \\ C &= -Y_v(m_{Y_{\dot r}} u_0 - N_r) + (m_{uv} u_0 + N_v)(m_u u_0 - Y_r) \end{aligned} \tag{7.61}$$

The A value is generally positive due the large positive values of m_v and m_r. Thus, similar to the development made with the equation (7.48), the stability is achieved when $B \geq 0$ and $C \geq 0$. Taking again as an example the Sirene AUV (see table 7.4) its B value is positive but the C value continues to be negative, which indicates instability.

7.4.2 Transfer functions

From equation (7.57) we can extract

$$\frac{\boldsymbol{\nu}(s)}{\boldsymbol{\tau}(s)} = \begin{bmatrix} m_u s - 2|u_0|X_{|u|u} - X_u & 0 & 0 \\ 0 & m_v s - Y_v & m_{Y_{\dot r}} s + m_u u_0 - Y_r \\ 0 & m_{N_{\dot v}} s - m_{uv} u_0 - N_v & m_r s + m_{Y_{\dot r}} u_0 - N_r \end{bmatrix}^{-1}$$

$$\frac{\boldsymbol{\nu}(s)}{\boldsymbol{\tau}(s)} = \left[\begin{array}{c:c} \dfrac{1}{m_u s - 2|u_0|X_{|u|u} - X_u} & \begin{matrix} 0 & \qquad 0 \end{matrix} \\ \hdashline \begin{matrix} 0 \\ \\ 0 \end{matrix} & \dfrac{\begin{bmatrix} m_r s + m_{Y_{\dot r}} u_0 - N_r & -m_{Y_{\dot r}} s - m_u u_0 + Y_r \\ -m_{N_{\dot v}} s + m_{uv} u_0 + N_v & m_v s - Y_v \end{bmatrix}}{As^2 + Bs + C} \end{array} \right] \tag{7.62}$$

where A, B, and C are the coefficients of the equations (7.61). From these transfer functions we extract two important functions, one for the forward velocity and the other for the angular velocity

$$\frac{u(s)}{\text{Prop}(s)} = \frac{1}{m_u s - 2|u_0|X_{|u|u} - X_u} \tag{7.63}$$

$$\frac{r(s)}{\text{Torque}(s)} = \frac{m_v s - Y_v}{As^2 + Bs + C} \tag{7.64}$$

Some simulations of the Sirene AUV have been made in order to study the performance. Figure 7.7 shows the performance for the u velocity. The model was linearized at 0.5 m/s and simulated in two cases, step response and response to a sinusoidal signal. On the other hand, figure 7.8 shows similar simulations for the angular velocity r, linearizing the model at 0 m/s and using the torque as input. Observe that, unlike the simulation in figure 7.6, in this simulation the value of r converges for the step response.

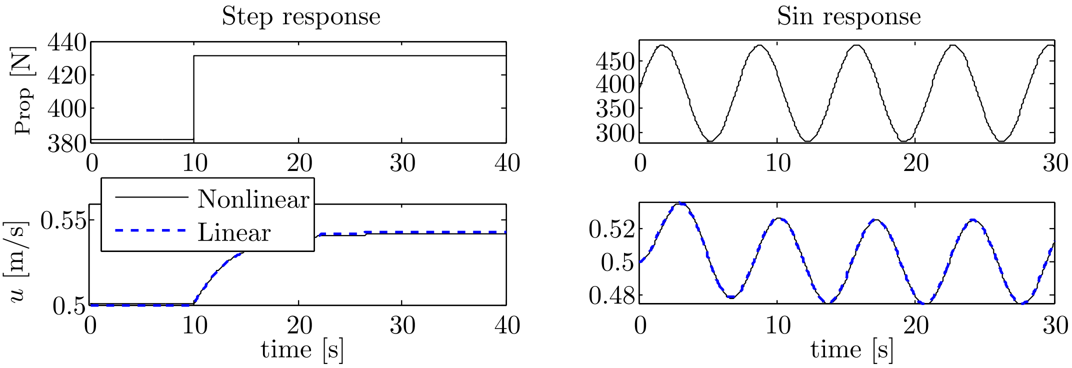


Figure 7.7: u velocity in Sirene AUV (considering linear damping coefficients): non-linear and linear model at u_0 =0.5 m/s

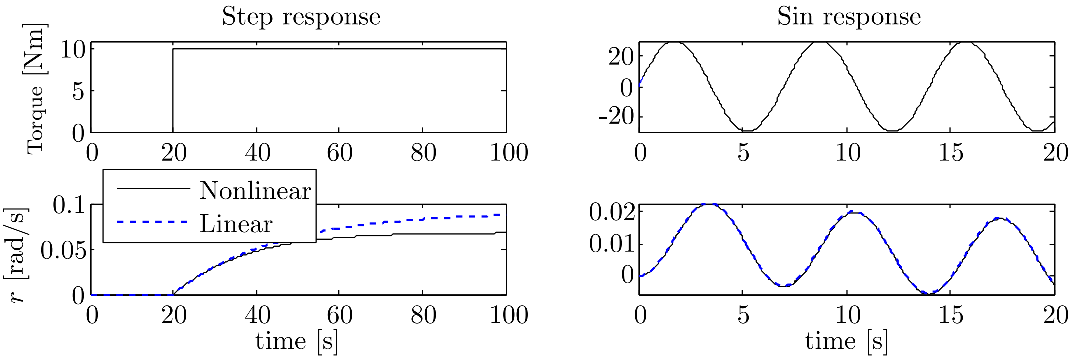


Figure 7.8: r velocity in Sirene AUV (considering linear damping coefficients): non-linear and linear model at u_0 =0 m/s

If we calculate the final value of Δr for a step response, we have

$$\begin{aligned}\Delta r_{(t=\infty)} &= \lim_{s\to 0} s\left(\frac{r(s)}{\text{Torque}(s)}\text{Torque}(s)\right)\\ &= \lim_{s\to 0} s\frac{m_v s - Y_v}{As^2+Bs+C}\cdot\frac{1}{s}\\ &= \lim_{s\to 0}\frac{m_v s - Y_v}{As^2+Bs+C}\end{aligned}$$

$$\Delta r_{(t=\infty)} = \frac{Y_v}{Y_v(m_{Y_{\dot r}}u_0 - N_r) - (m_{uv}u_0 + N_v)(m_u u_0 - Y_r)} \tag{7.65}$$

which confirms the convergence of r.

Hence, in the next sections we will use the model described above in order to have a more accurate, stable dynamics of the linear model.

7.4.3 Stability comparison

Until now, we have presented three conditions of stability in the forward movement, see sections 7.1.4, 7.3.1 and 7.4.1. They deal with the positivity of some values, but in particular the positivity of C, which is the most characteristic and sensitive value. These three forms are summarized as:

$$C = Y_v(N_r - mx_g u_0) + N_v(mu_0 - Y_r) \tag{7.66}$$

$$C = m_u m_{uv} u_0^2 \tag{7.67}$$

$$C = -Y_v(m_{Y_{\dot r}}u_0 - N_r) + (m_{uv}u_0 + N_v)(m_u u_0 - Y_r) \tag{7.68}$$

The first one, proposed in the literature by [1] and [147], is derived for a linearization where the matrices $\mathbf{C_A}$ and $\mathbf{D_n}$ are not taken into account.

The second one, calculated from the equations of Prestero [105], shows the C value for a linearization where only the matrix $\mathbf{D_l}$ is not included.

Finally, the third one, shows the C value for a linearization where all of the matrices mentioned above are taken into account.

Given all of these options, one might ask what is the most appropriate value for C. The answer lies in the type of model chosen, that is, depending of the matrices selected for the model we have to use one calculation or another. In the case of the GUANAY II we selected the relation (7.68) because the resulting model includes more dynamics and has good properties, as discussed in the previous section.

7.5 Coefficient calculation

7.5.1 Theoretical calculation

The work of Prestero [105] shows the coefficient calculation for the Remus AUV, a vehicle developed at the WHOI which also uses the Myring profiles for its geometry. For the calculation, he uses concepts from fluid dynamics like the Reynolds number, the Slender body theory, and also the theory of Newman [92], Hoerner [55], Blevins [15], among others, which provide empirical formulas for different coefficients.

The importance of these empirical formulas is that they do not depend of the general equation of marine dynamics, which have highly coupled coefficients, but on the other hand they can be calculated one by one. In this section we will use this knowledge in order to estimate the coefficients of GUANAY II , taking advantage of the fact that its geometry is very similar to that of Remus.

Axial damping

The axial damping is calculated using the following formula

$$X_{|u|u} = -\frac{1}{2}\rho c_d A_f \tag{7.69}$$

where ρ is the density of the surrounding fluid, in this case the seawater, A_f is the vehicle frontal area, and c_d is an empirical coefficient of value 0.3 [105, pag. 26]. This yields a value of -12.8334 kg/m.

Crowsflow drag

The crossflow drag is calculated using the following formulas

$$\begin{aligned}
Y_{|v|v} &= -\tfrac{1}{2}\rho c_{dc}\int_x 2R(x)dx - \rho S_{\text{fin}} c_{df} \\
N_{|v|v} &= -\tfrac{1}{2}\rho c_{dc}\int_x 2xR(x)dx - x_{\text{fin}}\rho S_{\text{fin}} c_{df} \\
Y_{|r|r} &= -\tfrac{1}{2}\rho c_{dc}\int_x 2x|x|R(x)dx - x_{\text{fin}}|x_{\text{fin}}|\rho S_{\text{fin}} c_{df} \\
N_{|r|r} &= -\tfrac{1}{2}\rho c_{dc}\int_x 2x^3R(x)dx - x_{\text{fin}}^3\rho S_{\text{fin}} c_{df}
\end{aligned} \tag{7.70}$$

where ρ is the seawater density, c_{dc} the drag coefficient of a cylinder, which is estimated at 1.1 [105, pag. 26], $R(x)$ is the hull radius as a function of axial position as given by equations (7.25) and (7.26), S_{fin} is the fin platform area, x_{fin} is the axial position of the fin, and c_{df} is an empirical coefficient for crossflow drag, which is estimated at 0.56 [105, pag.

26]. This calculation yields the values -425.6532 kg/m, -42.8456 kg/m, -43.2557 kg·m/rad^2 and -43.6539 kg·m^2/rad^2 respectively.

Axial added mass

To estimate axial added mass, the vehicle hull shape is approximated to an ellipsoid. One can then use the expression

$$X_{\dot{u}} = -\frac{4\beta\rho\pi}{3}\left(\frac{d}{2}\right)^3 \tag{7.71}$$

where ρ is the seawater density, and β is an empirical parameter determined by the ratio of the vehicle length to its diameter, as shown in Table 7.5 [105, pag. 28]. For Guanay II this parameter has value 0.2473. This calculation yields an axial added mass of -4.5983 kg.

Table 7.5: Axial added mass parameter

l/d	β
1.0	0.5000
2.0	0.4200
3.0	0.3660
5.0	0.2956
7.0	0.2510
10.0	0.2071

Crossflow added mass

Slender body theory is used for this computation, which assumes that the length of the vehicle is large compared with its diameter. In this way, the crossflow added masses can be calculated as the sum of cylindrical slices. We define the variable m_a which represents the added mass per unit length; it has 2 expressions, depending on whether the transverse section of the vehicle is just a cylinder or it corresponds to a section that has also fins, as shown in equation (7.72).

$$m_a(x) = \begin{cases} \pi\rho R(x)^2 & \text{for } x \text{ cylinder} \\ \pi\rho\left(a_{\text{fin}}^2 - R(x)^2 + \frac{R(x)^4}{a_{\text{fin}}^2}\right) & \text{for } x \text{ fin} \end{cases} \tag{7.72}$$

In this way, the crossflow added masses can be calculated from the following integrals along the x-axis:

$$\begin{aligned} Y_{\dot{v}} &= -\int_x m_a(x)dx \\ Y_{\dot{r}} = N_{\dot{v}} &= -\int_x x m_a(x)dx \\ N_{\dot{r}} &= -\int_x x^2 m_a(x)dx \end{aligned} \tag{7.73}$$

This calculation yields the values -509.7220 kg, -176.3057 kg and -224.5006 kg·m^2/rad respectively.

All of these coefficients are summarized in Table 7.6. These calculations do not take into account the linear damping coefficients mentioned in Section 7.4.

Table 7.6: Theoretical coefficients of the GUANAY II AUV

Parameter	Value	Units
$X_{\dot{u}}$	-4.5983	kg
$Y_{\dot{v}}$	-509.7220	kg
$Y_{\dot{r}}$	-176.3057	kg·m/rad
$N_{\dot{v}}$	-176.3057	kg·m
$N_{\dot{r}}$	-224.5006	kg·m^2/rad
$X_{\|u\|u}$	-12.8334	kg/m
$Y_{\|v\|v}$	-425.6532	kg/m
$Y_{\|r\|r}$	-43.2557	kg·m/rad^2
$N_{\|v\|v}$	-42.8456	kg
$N_{\|r\|r}$	-43.6539	kg·m^2/rad^2

It is important to note that $X_{\dot{u}} > Y_{\dot{v}}$, which leads to the value $m_{uv} < 0$ (see equation 7.43). And thus, the system is unstable in the forward movement, as is shown in the conclusions of the equations (7.47) and (7.48). This result shows that there is a problem with the calculated coefficients, because the GUANAY II is a very stable vehicle (its keel is long and is located at the stern). Figure 7.9 shows a simulation using these coefficients. The constant propulsion of 40 N leads to a forward velocity near to 2 m/s. However, if a small perturbation acts on the vehicle, like a momentary action in a side thruster, the angular velocity r becomes a nonzero constant and the vehicle turns indefinitely. The simulation also shows a significant decrease in the forward velocity ($u = 0.2$m/s) and a large increase in the angular velocity ($r =$27.5 °/s), leading to the vehicle standing at one point and turning on itself. This significant error is due to the high negativity of m_{uv}, which makes the system much more unstable.

7.5.2 Calculation with experimental data

As an alternative to the theoretical values, this section presents the computation of the coefficients using data from actual navigation of the vehicle. These data must be taken with the major precision possible, in a environment with the lowest presence of noise and doing the simplest movements for greater simplicity in determining its dynamics.

The fitting of the experimental data to the hydrodynamic model presents an important challenge since the coefficients are highly coupled, as can be seen from equation (7.23), and it is difficult to separate the action of each coefficient. The other problem is the calculation of velocities and accelerations. The GUANAY II has an IMU but, as will be seen in section 4.4.1, its precision is not enough due to the accumulative error in the integration. The GPS and compass are more precise but they get data at a frequency of only 1 Hz.

For this reason, it is necessary to use a simple identification method that can be adapted to these difficult circumstances . The Least Squares (LS) optimization technique is a good option to minimize the error and is easy to implement [112].

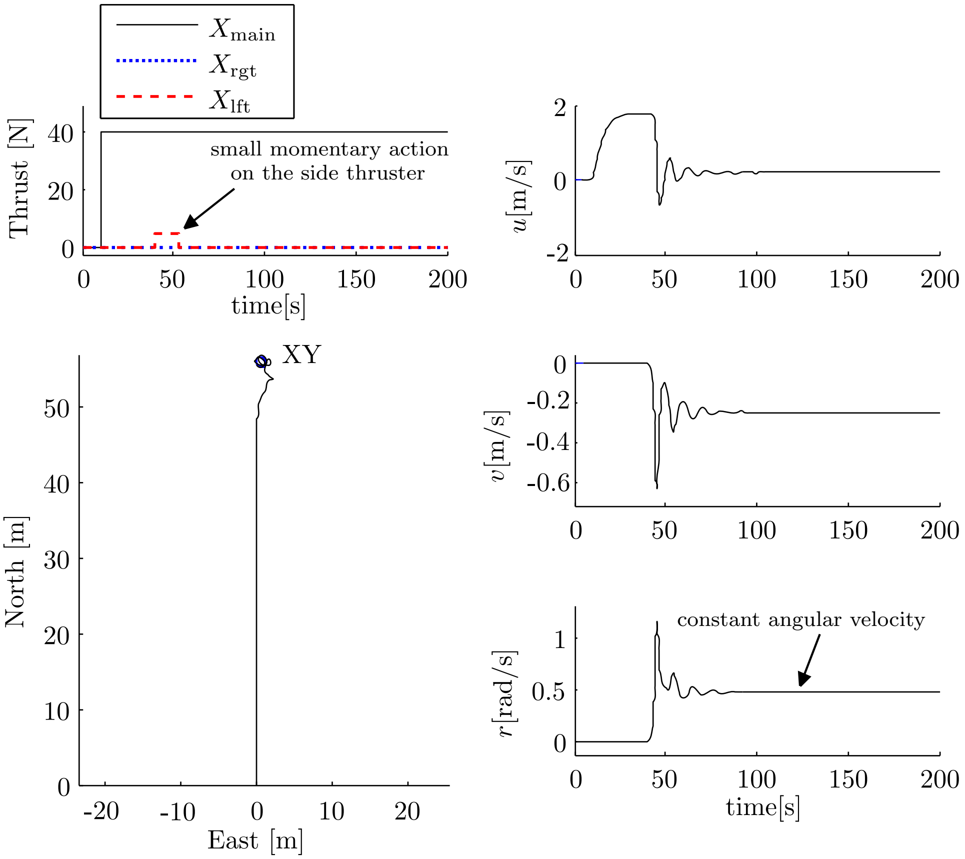


Figure 7.9: Simulation using the theoretical coefficients

Thruster model

The GUANAY II has 3 thrusters: a main thruster for the forward movement, and other two thrusters at the end of the fins for the turns. The main thruster has its own drive, which sets the angular velocity to a value ordered by RS-485 commands, reaching 127 N at 960 rpm [141]. On the other hand, the commands to the lateral thrusters set a PWM signal which gives a rate of the nominal power, reaching 28 N at 110 W [142]. We will assume that the angular velocity of the lateral thrusters is proportional to this input in order to compare it with the main thruster.

For the modeling of the thrusters, we use the work of Yoerger [160], which proposes the following system

$$\begin{aligned} \dot{\Omega} &= \beta \mathrm{Torque} - \alpha\Omega|\Omega|, \\ \mathrm{Thrust} &= C_t\Omega|\Omega|. \end{aligned} \tag{7.74}$$

Here Ω is the propeller angular velocity, Torque is the input torque, α and β are constant model parameters, and C_t is a proportionality constant. The first equation represents the transient dynamics of the angular velocity, and the second equation the force that it produces.

Defining λ_{main}, λ_{lft} and λ_{rgt} as the angular velocities normalized between the values -100 and 100 (as a percentage value) for the main thruster, left thruster, and right thruster respectively, the force generated for each thruster is:

$$\begin{aligned} X_{\text{main}} &= c_{\text{main}}\lambda_{\text{main}}|\lambda_{\text{main}}| \\ X_{\text{lft}} &= c_{\text{lft}}\lambda_{\text{lft}}|\lambda_{\text{lft}}| \\ X_{\text{rgt}} &= c_{\text{rgt}}\lambda_{\text{rgt}}|\lambda_{\text{rgt}}| \end{aligned} \tag{7.75}$$

where c_{main}, c_{lft} and c_{rgt} are proportionality constants. Knowing the maximum force of each thruster, a first estimation of these coefficients yields 0.0127, 0.0028 and 0.0028, respectively.

Least squares method

Solving equation (7.23), we have

$$(m - X_{\dot{u}})\dot{u} - X_u u - X_{|u|u}u|u| - (m - Y_{\dot{v}})vr + Y_{\dot{r}}r^2 = \text{Prop} \tag{7.76}$$

Assuming a forward movement, the components v and r will be zero, and then

$$\begin{aligned} (m - X_{\dot{u}})\dot{u} - X_u u - X_{|u|u}u|u| &= \text{Prop} \\ \dot{u} - \frac{X_u}{m - X_{\dot{u}}}u - \frac{X_{|u|u}}{m - X_{\dot{u}}}u|u| &= \frac{\text{Prop}}{m - X_{\dot{u}}} \\ \text{Prop}\frac{1}{m - X_{\dot{u}}} + u\frac{X_u}{m - X_{\dot{u}}} + u|u|\frac{X_{|u|u}}{m - X_{\dot{u}}} &= \dot{u} \\ \begin{bmatrix} \text{Prop} & u & u|u| \end{bmatrix} \begin{bmatrix} \frac{1}{m-X_{\dot{u}}} & \frac{X_u}{m-X_{\dot{u}}} & \frac{X_{|u|u}}{m-X_{\dot{u}}} \end{bmatrix}^{\mathsf{T}} &= \dot{u}. \end{aligned}$$

Using the sensors of the GUANAY II we can calculate the velocities, accelerations and propulsion in the field tests. In this way, this equation can be reinterpreted as a set of equations in a matrix system as follows:

$$\underbrace{\begin{bmatrix} \text{Prop}_1 & u_1 & u_1|u_1| \\ \text{Prop}_2 & u_2 & u_2|u_2| \\ \text{Prop}_3 & u_3 & u_3|u_3| \\ \vdots & \vdots & \vdots \\ \text{Prop}_n & u_n & u_n|u_n| \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} \frac{1}{m - X_{\dot{u}}} \\ \frac{X_u}{m - X_{\dot{u}}} \\ \frac{X_{|u|u}}{m - X_{\dot{u}}} \end{bmatrix}}_{\mathbf{S}} = \underbrace{\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \vdots \\ \dot{u}_n \end{bmatrix}}_{\mathbf{Q}} \tag{7.77}$$

where the values $(\text{Prop}_i, u_i, \dot{u}_i)$ are each of the records taken in the field test at time i. Given this system of equations $\mathbf{PS} = \mathbf{Q}$, the LS method uses the pseudoinverse of $\mathbf{P}$, denoted as $\mathbf{P}^+$, for the fitting:

$$\mathbf{P}^+ = (\mathbf{P}^{\mathsf{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathsf{T}} \tag{7.78}$$

Applying this pseudoinverse we have

$$\begin{aligned}
\mathbf{PS} &= \mathbf{Q} \\
\mathbf{P}^{+}\mathbf{PS} &= \mathbf{P}^{+}\mathbf{Q} \\
(\mathbf{P}^{\intercal}\mathbf{P})^{-1}\mathbf{P}^{\intercal}\mathbf{PS} &= (\mathbf{P}^{\intercal}\mathbf{P})^{-1}\mathbf{P}^{\intercal}\mathbf{Q} \\
\mathbf{S} &= (\mathbf{P}^{\intercal}\mathbf{P})^{-1}\mathbf{P}^{\intercal}\mathbf{Q}
\end{aligned} \tag{7.79}$$

The LS method shows that this calculation minimizes the sum of the squares of the errors made in the results of every single equation.

In a similar way, from equation (7.23) we have

$$(m - Y_{\dot{v}})\dot{v} - Y_{\dot{r}}\dot{r} - Y_v v - Y_{|v|v}v|v| - Y_r r - Y_{|r|r}r|r| + (m - X_{\dot{u}})ur = 0 \tag{7.80}$$

$$(I_z - N_{\dot{r}})\dot{r} - N_{\dot{v}}\dot{v} - N_v v - N_{|v|v}v|v| - N_r r - N_{|r|r}r|r| - (Y_{\dot{v}} - X_{\dot{u}})uv - Y_{\dot{r}}ur = \text{Torque} \tag{7.81}$$

These two equations are highly coupled and have a lot of coefficients. Additionally, they are related to the velocities v and r which correspond to lateral movements and turns. Assuming that we know the coefficient $X_{\dot{u}}$, the equations can be reordered in a way that the left side contains the unknown coefficients and the right side the known values:

$$(m - Y_{\dot{v}})\dot{v} - Y_{\dot{r}}\dot{r} - Y_v v - Y_{|v|v}v|v| - Y_r r - Y_{|r|r}r|r| = -(m - X_{\dot{u}})ur \tag{7.82}$$

$$\begin{aligned}
&(m - Y_{\dot{v}})uv + (I_z - N_{\dot{r}})\dot{r} - N_{\dot{v}}\dot{v} - N_v v - N_{|v|v}v|v| - N_r r + \\
&\qquad - N_{|r|r}r|r| - Y_{\dot{r}}ur = \text{Torque} + (m - X_{\dot{u}})uv
\end{aligned} \tag{7.83}$$

Note that we split $(Y_{\dot{v}} - X_{\dot{u}})$ as $(m - X_{\dot{u}}) - (m - Y_{\dot{v}})$ in order to have the same term $(m - Y_{\dot{v}})$ in the two equations. One way to calculate the coefficients using the LS method is as follows:

$$\left[\begin{array}{cccccccccccc}
\dot{v}_1 & 0 & 0 & \text{-}\dot{r}_1 & \text{-}v_1|v_1| & \text{-}r_1|r_1| & 0 & 0 & \text{-}v_1 & \text{-}r_1 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\dot{v}_n & 0 & 0 & \text{-}\dot{r}_n & \text{-}v_n|v_n| & \text{-}r_n|r_n| & 0 & 0 & \text{-}v_n & \text{-}r_n & 0 & 0 \\
\hdashline
u_1 v_1 & \dot{r}_1 & \text{-}\dot{v}_1 & u_1 r_1 & 0 & 0 & \text{-}v_1|v_1| & \text{-}r_1|r_1| & 0 & 0 & \text{-}v_1 & \text{-}r_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_n v_n & \dot{r}_n & \text{-}\dot{v}_n & u_n r_n & 0 & 0 & \text{-}v_n|v_n| & \text{-}r_n|r_n| & 0 & 0 & \text{-}v_n & \text{-}r_n
\end{array}\right]
\begin{bmatrix}
m - Y_{\dot{v}} \\ I_z - N_{\dot{r}} \\ N_{\dot{v}} \\ Y_{\dot{r}} \\ Y_{|v|v} \\ Y_{|r|r} \\ N_{|v|v} \\ N_{|r|r} \\ Y_v \\ Y_r \\ N_v \\ N_r
\end{bmatrix} =
\left[\begin{array}{c}
-(m - X_{\dot{u}})u_1 r_1 \\
\vdots \\
-(m - X_{\dot{u}})u_n r_n \\
\hdashline
\text{Torque}_1 + (m - X_{\dot{u}})u_1 v_1 \\
\vdots \\
\text{Torque}_n + (m - X_{\dot{u}})u_n v_n
\end{array}\right] \tag{7.84}$$

where the values $(\text{Torque}_i, u_i, v_i, r_i, \dot{v}_i, \dot{r}_i)$ are each of the records taken in the field test at time i.

In order to have a better fitting of the experimental data of lateral movements and turns with the equations, it is better to use the three equations. Assuming that we know the coefficients $X_{\dot{u}}$, X_u and $X_{|u|u}$, we can add the information of the equation (7.76) to the above matrix equation (7.84) as follows:

$$\underbrace{\begin{bmatrix} -v_1 r_1 & 0 & 0 & r_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -v_n r_n & 0 & 0 & r_n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline \dot{v}_1 & 0 & 0 & -\dot{r}_1 & -v_1|v_1| & -r_1|r_1| & 0 & 0 & -v_1 & -r_1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dot{v}_n & 0 & 0 & -\dot{r}_n & -v_n|v_n| & -r_n|r_n| & 0 & 0 & -v_n & -r_n & 0 & 0 \\ \hdashline u_1 v_1 & \dot{r}_1 & -\dot{v}_1 & u_1 r_1 & 0 & 0 & -v_1|v_1| & -r_1|r_1| & 0 & 0 & -v_1 & -r_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n v_n & \dot{r}_n & -\dot{v}_n & u_n r_n & 0 & 0 & -v_n|v_n| & -r_n|r_n| & 0 & 0 & -v_n & -r_n \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} m - Y_{\dot{v}} \\ I_z - N_{\dot{r}} \\ N_{\dot{v}} \\ Y_{\dot{r}} \\ Y_{|v|v} \\ Y_{|r|r} \\ N_{|v|v} \\ N_{|r|r} \\ Y_v \\ Y_r \\ N_v \\ N_r \end{bmatrix}}_{\mathbf{S}} =$$

$$\underbrace{\begin{bmatrix} \text{Prop}_1 - (m - X_{\dot{u}})\dot{u}_1 + X_u u_1 + X_{|u|u} u_1|u_1| \\ \vdots \\ \text{Prop}_n - (m - X_{\dot{u}})\dot{u}_n + X_u u_n + X_{|u|u} u_n|u_n| \\ \hdashline -(m - X_{\dot{u}}) u_1 r_1 \\ \vdots \\ -(m - X_{\dot{u}}) u_n r_n \\ \hdashline \text{Torque}_1 + (m - X_{\dot{u}}) u_1 v_1 \\ \vdots \\ \text{Torque}_n + (m - X_{\dot{u}}) u_n v_n \end{bmatrix}}_{\mathbf{Q}} \tag{7.85}$$

Data preprocessing

The GPS gives the position in latitude/longitude coordinates. The position in meters can be calculated as follows:

$$\text{North:} \quad n = r \text{ lat} \tag{7.86}$$

$$\text{East:} \quad e = r \text{ long} \cos(\text{lat}) \tag{7.87}$$

where r is the earth radius, and the latitude and longitude are given in radians. Using equations (7.16), it is possible to calculate the different accelerations and velocities by applying derivatives:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \frac{dn}{dt} & \frac{de}{dt} & \frac{d\psi}{dt} \end{bmatrix}^{\mathsf{T}} \tag{7.88}$$

$$\nu = \mathbf{R}(\psi)\,\dot{\boldsymbol{\eta}} \tag{7.89}$$

$$\dot{\boldsymbol{\nu}} = \begin{bmatrix} \frac{du}{dt} & \frac{dv}{dt} & \frac{dr}{dt} \end{bmatrix}^{\mathsf{T}} \tag{7.90}$$

Note that since the GPS and compass give their data at discrete time, these derivatives give the average velocity $\boldsymbol{\nu}$ between two samples and average acceleration $\dot{\boldsymbol{\nu}}$ between three samples.

Experimental data and fitting

Several field tests of the Guanay II AUV were taken in the Catalonia Olympic Channel (see section 6.5). Since it is very quiet the perturbations due the currents are small. Additionally, the channel size is enough to maneuver the vehicle. The tests consisted in recollecting data from two movements to apply the LS technique described above: a) Forward movement, in order to calculate the first group of coefficients in equation (7.77); and b) Movement with turns, in order to calculate the second group of coefficients in equation (7.85).

The total amount of data that was taken corresponds to about 2 hours, and all the sensors worked correctly. These tests also revealed a small difference between the powers of the lateral thrusters, with the right thruster performing at more power than the left thruster. This implies a recalculation of the coefficients c_{rgt} and c_{lft}. Due to this, the process of coefficient identification consisted in an iterative calculation, using the following steps:

1. Assignation of the values c_{lft} and c_{rgt} for the lateral thrusters.

2. Calculation of the coefficients $X_{\dot{u}}$, X_u and $X_{|u|u}$ taking the experimental data in forward movement, and using the least square method with equations (7.77) and (7.79).

3. Calculation of the other hydrodynamic coefficients taking the experimental data for turns, and using the least square method with equations (7.85) and (7.79).

4. Check the performance of the coefficients calculating the mean squared error (MSE) of the fitted data with the vehicle simulations. If $\hat{X}$ is the vector of n values of the simulation, and X the vector of the experimental values, then the MSE is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{X}_i - X_i)^2 \tag{7.91}$$

5. In function of the results, repeat from step 1 using other values for c_{lft} and c_{rgt}.

After this process, we found that the correct values for the coefficients c_{lft} and c_{rgt} are 0.0027 and 0.0029, respectively. The final values for all the hydrodynamic coefficients are the Table 7.7[1]. The MSE of the steps 2 and 3 are:

$$\text{Step 2: MSE} = 0.002306 \tag{7.92}$$

$$\text{Step 3: MSE} = 320.2684 \tag{7.93}$$

Step 2 uses a $\mathbf{Q}$ around the acceleration $\dot{u}$, so that the root square of 0.002306 is an error regarding the forward acceleration. Step 3 uses a $\mathbf{Q}$ around the forces and torques, and hence the root square of 320.2684 (that is $\sim$ 17.9) is an error regarding these forces. This is the reason of the large difference between these two errors.

At this point, we can evaluate the condition of stability presented in Section 7.4.1. The calculation of the equations (7.61) using these coefficients yields $B > 0$ and $C > 0$, which implies stability in the forward movement.

Table 7.7: Final coefficients for Guanay II AUV

Parameter	Value	Units
m	168.2000	kg
I_z	52.7000	kg·m^2
$X_{\dot{u}}$	-452.6809	kg
$Y_{\dot{v}}$	-415.9546	kg
$Y_{\dot{r}}$	-518.5884	kg·m/rad
$N_{\dot{v}}$	18.3217	kg·m
$N_{\dot{r}}$	-231.9244	kg·m^2/rad
X_u	-0.2992	kg/s
Y_v	-45.3418	kg/s
Y_r	105.3055	kg·m/rad·s
N_v	-3.3604	kg·m/s
N_r	-56.9228	kg·m^2/rad·s
$X_{\|u\|u}$	-152.6010	kg/m
$Y_{\|v\|v}$	-399.4271	kg/m
$Y_{\|r\|r}$	-899.5579	kg·m/rad^2
$N_{\|v\|v}$	21.1978	kg
$N_{\|r\|r}$	-284.1011	kg·m^2/rad^2
a_{fin}	0.5000	m
c_{main}	0.0127	kg·m/s^2
c_{lft}	0.0027	kg·m/s^2
c_{rgt}	0.0029	kg·m/s^2

[1] For the reader's interest, these coefficients are calculated by running `GuanayIICoefCalculation.m`, which is attached in the CD.

Figures (7.10 - 7.17) show some intervals of the experimental data and the simulation made with the vehicle. All of them present the propulsion used in each experiment and the velocities obtained. The initial conditions in the simulations correspond to those of the experimental data. Tests are separated in two groups: forward movement and movement with turns.

Figures 7.10 and 7.11 show two tests with 60% of power in the main thruster. The velocity reached is about 0.5 m/s in both cases, and the settling time is about 5 s. The simulation has a little error respect the experimental data, but it is very adjusted to the reality.

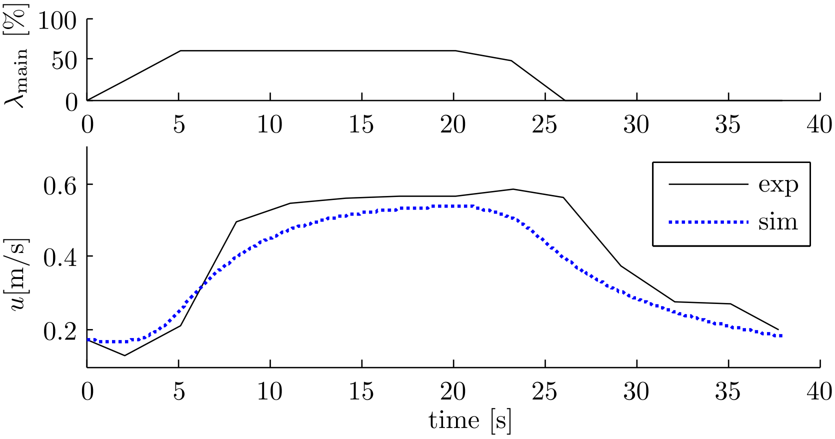


Figure 7.10: Real movement and simulation in straight line (a)

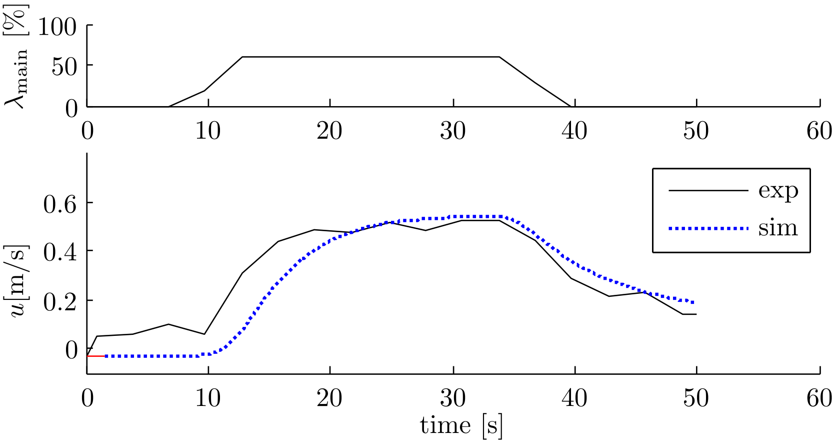


Figure 7.11: Real movement and simulation in straight line (b)

When the vehicle is set to 100% in the main thruster the velocity reaches 1 m/s with a settling time around 5 s; see Figure 7.12. This test has an undesired initial velocity of 0.25 m/s, due the currents in the water channel.

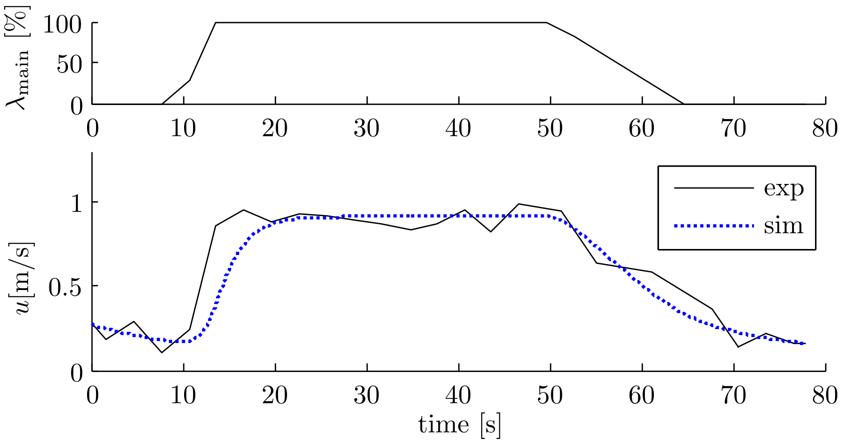


Figure 7.12: Real movement and simulation in straight line (c)

If the vehicle is commanded to perform turns, it implies a movement in 3 degrees of freedom for u, v and r. Figure 7.13 shows the performance of the right thruster at 100 % of power. The radius of curvature is about 10 m although the vehicle describes a semi-oval in the experimental data maybe due the currents. The forward velocity is 0.45 m/s and the angular velocity is -0.05 rad/s.

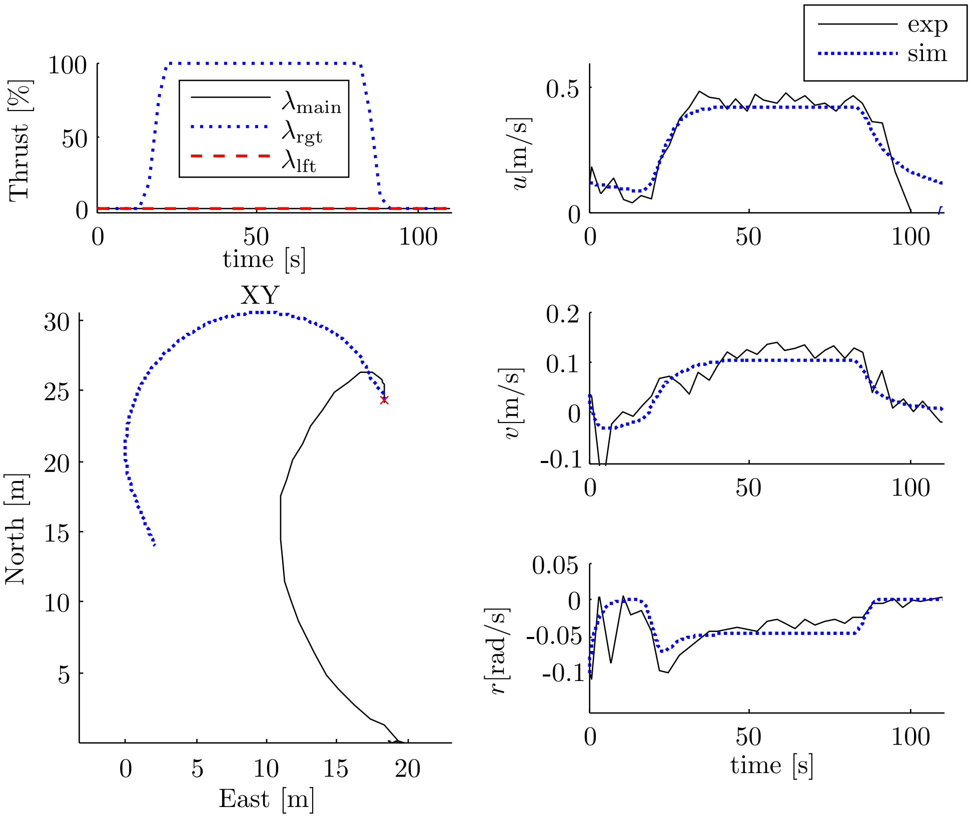


Figure 7.13: Left movement and simulation using the right thruster

Figure 7.14 shows the performance of the left thruster at 100 % of power. The radius of curvature is about 10 m although the vehicle describes again an oval in the experimental data due the currents. The forward velocity is 0.4 m/s and the angular velocity is 0.04 rad/s.

Notice that in this figure as well as in the other tests the XY path displays a significant difference between the simulation and the experimental data. This is because they only coincides at the beginning and as the simulation is an integration over the time, the position and direction will be more and more different. The noise in the GPS and compass are responsible for this error.

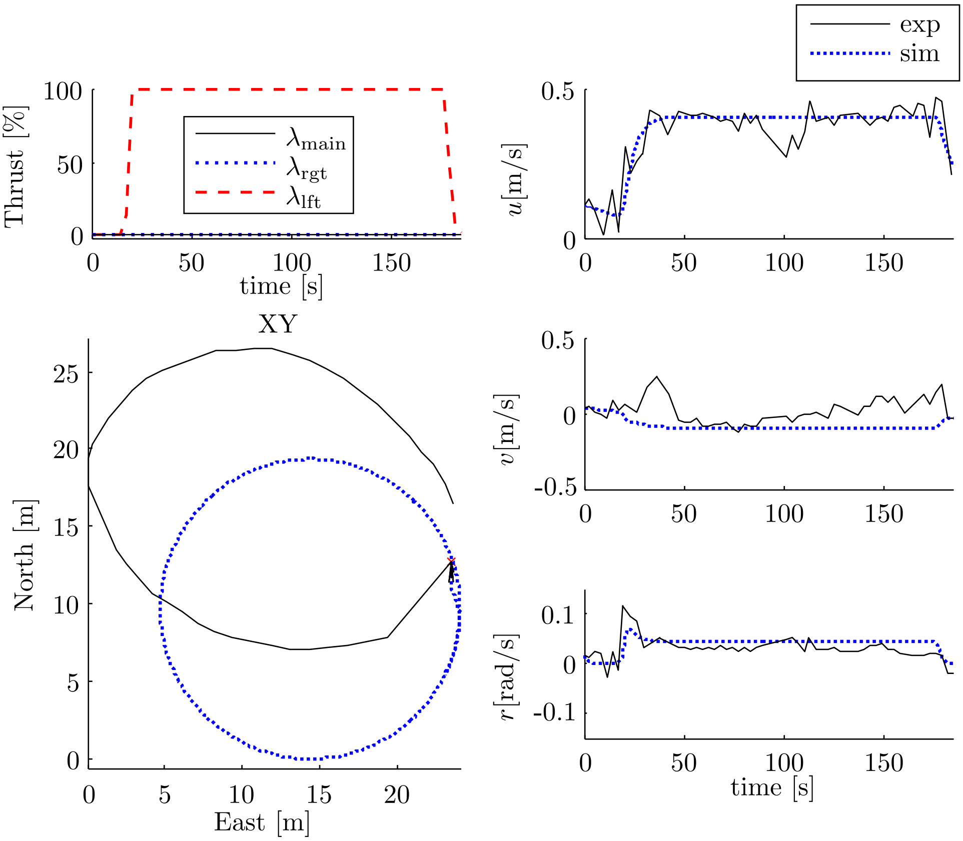


Figure 7.14: Right movement and simulation using the left thruster

Figure 7.15 shows different power combinations of the three thrusters, starting with the lateral thrusters at 50 % in left and -50 % in right, and the main thruster at 30 %. In this period the simulation almost describes a complete circle, while in the field test describes a semi-oval. Notice that at this point the position and direction is very different between them. The propulsion continues with 100 % in the main thruster and 0 % in the lateral thrusters. In this period the vehicle describes a straight line.

Regarding the velocities, the simulation fits very well the experimental data.

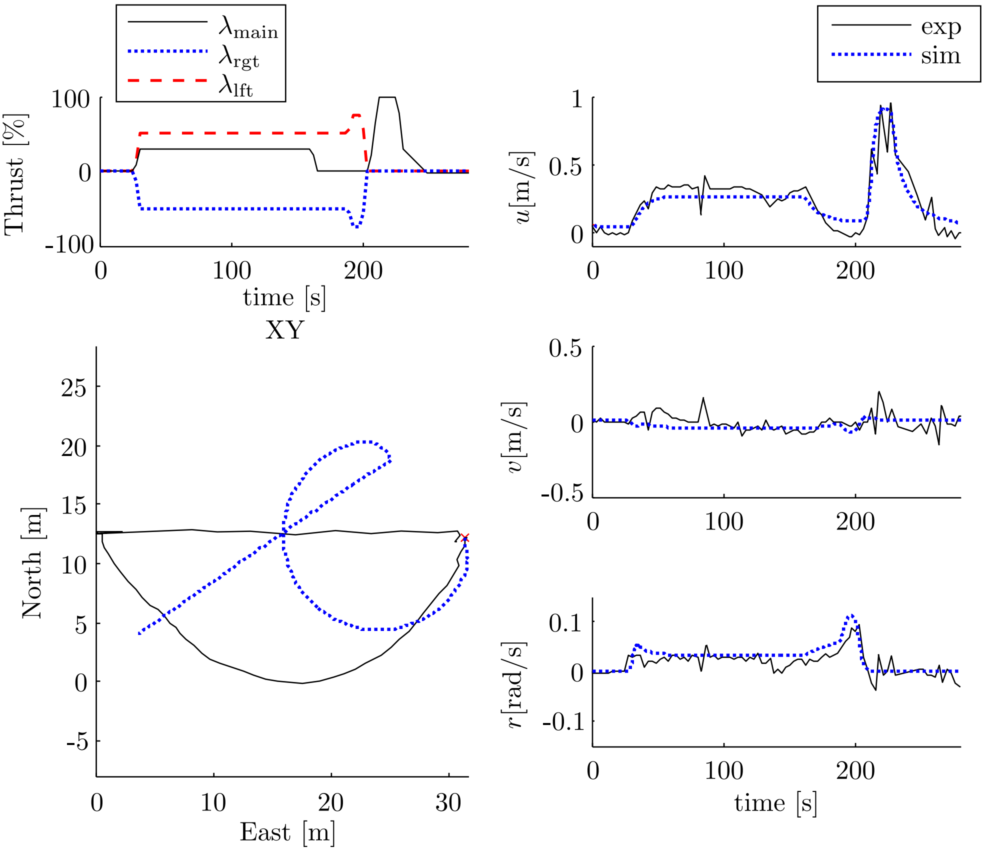


Figure 7.15: Real movement and simulation in straight line and turns (a)

Figure 7.16 shows the test for one turn and two straight lines. The main thruster uses 50 % of power for the two lines. Notice that using this propulsion the forward velocity is 0.6 m/s. However, the field test in Figure 7.11 shows a velocity of 0.5 m/s using more propulsion (60 %). This slight difference is due to the currents. On the other hand, the simulation shows a velocity below 0.5 m/s which is the expected value.

Regarding the turn, both simulation and experimental data display a half turn using about 50 % in the right thruster and -50 % in the left thruster during 50 s.

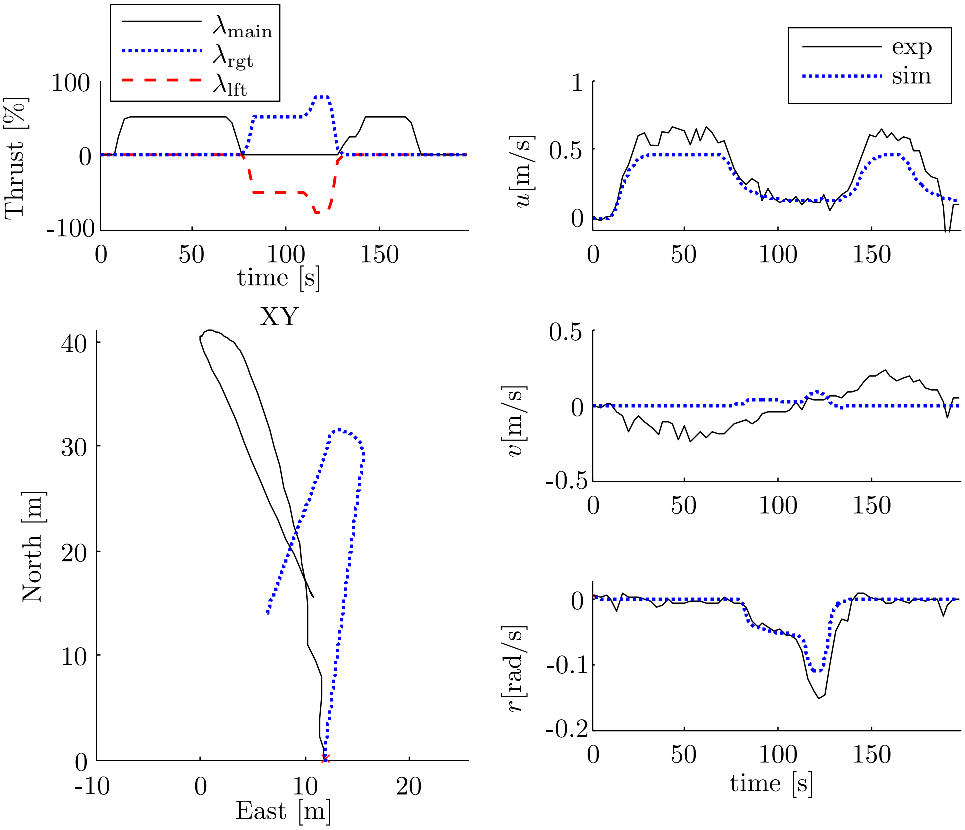


Figure 7.16: Real movement and simulation in straight line and turns (b)

Finally, figure 7.17 shows the movement of the vehicle with different propulsion values. As in the previous tests, the experimental forward velocity is greater than in the simulation, but the angular velocitiy r is essentially the same.

In the interval 0-80 s the thrusters act to perform a straight line with a small curve. In the period 80-200 s the lateral thrusters act more strongly and the vehicle performs a curve and continues with a straight line. In the period 200-230 s all the thrusters are stopped, the vehicle loses velocity and navigates freely. In the period 230-270 s the main thruster is almost stopped and the lateral thrusters are about 50 %. In this case the angular velocity is very high, and looking at the XY plane, both simulation and experimental data show a change of direction of about 110 °. In the final period of 270-400 s, the main thruster acts more strongly while the lateral thrusters act with small forces. In this case the vehicle performs a straight line with a small curve.

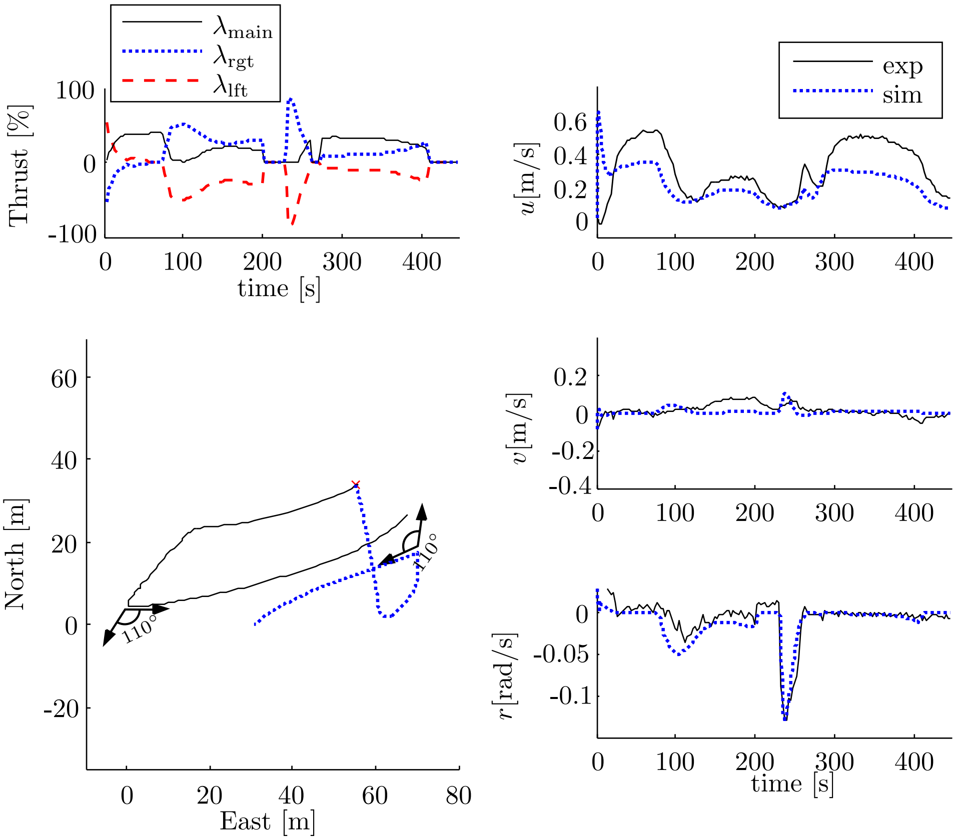


Figure 7.17: Real movement and simulation in straight line and turns (c)

7.5.3 Comparison

A comparison between the theoretical coefficients and the ones calculated with the experimental data is shown in Table 7.8. It presents a significant difference in most of the coefficients, but a similar value in coefficients like $Y_{\dot{v}}$, $N_{\dot{r}}$ and $Y_{|v|v}$.

Regarding the performance of each set, we have discarded the theoretical coefficients due two important reasons:

1. Section 7.4 have shown the importance of adding the linear damping coefficients in order to have a better linearization of the system. The coefficients calculated with the experimental data take this into account, but the theoretical values do not consider them.

2. The Guanay II AUV has a long keel (located at the stern) which gives it stability in the forward movement. This stability is guarantied for the final experimental coefficients, but not for the theoretical ones.

Table 7.8: Comparison: Final coefficients and theoretical calculation

Parameter	Final value	Theoretical	Units
$X_{\dot{u}}$	-452.6809	-4.5983	kg
$Y_{\dot{v}}$	-415.9546	-509.7220	kg
$Y_{\dot{r}}$	-518.5884	-176.3057	kg·m/rad
$N_{\dot{v}}$	18.3217	-176.3057	kg·m
$N_{\dot{r}}$	-231.9244	-224.5006	kg·m^2/rad
X_u	-0.2992	—	kg/s
Y_v	-45.3418	—	kg/s
Y_r	105.3055	—	kg·m/rad·s
N_v	-3.3604	—	kg·m/s
N_r	-56.9228	—	kg·m^2/rad·s
$X_{\|u\|u}$	-152.6010	-12.8334	kg/m
$Y_{\|v\|v}$	-399.4271	-425.6532	kg/m
$Y_{\|r\|r}$	-899.5579	-43.2557	kg·m/rad^2
$N_{\|v\|v}$	21.1978	-42.8456	kg
$N_{\|r\|r}$	-284.1011	-43.6539	kg·m^2/rad^2

7.6 Physical constraints

This Section shows the map that gives the possible input values to the modeled vehicle using the maximum forces of the three thrusters.

The range of action of the three thrusters, using equations (7.75) and the final coefficients (see table 7.7), are

$$\begin{aligned} X_{\text{main}} &\in [-127, 127]\ \text{N} \\ X_{\text{lft}} &\in [-27, 27]\ \text{N} \\ X_{\text{rgt}} &\in [-29, 29]\ \text{N} \end{aligned} \tag{7.94}$$

The combination of these three actuators will give an input of two values, a forward propulsion and a torque, following equations (7.22). Defining the superindexes $^+$ and $^-$ as the maximum and minimum possible value, respectively, for a thruster, the possible extreme values of power are:

1. $(X^+_{\text{main}}, X^+_{\text{lft}}, X^+_{\text{rgt}})$ which gives a propulsion and torque of (183 N, -1 Nm). This is the maximum propulsion possible and implies a little torque.

2. $(X^+_{\text{main}}, X^+_{\text{lft}}, X^-_{\text{rgt}})$ which gives a propulsion and torque of (125 N, 28 Nm). This is the maximum propulsion possible at the maximum torque.

3. $(X^+_{\text{main}}, X^-_{\text{lft}}, X^+_{\text{rgt}})$ which gives a propulsion and torque of (129 N, -28 Nm). This is the maximum propulsion possible at the minimum torque.

4. $(X^+_{\text{main}}, X^-_{\text{lft}}, X^-_{\text{rgt}})$ which gives a propulsion and torque of (71 N, 1 Nm). This is an undesirable combination because it consumes a lot of energy, and the propulsion and torque can be achieved with a more economic combination.

5. $(X^-_{\text{main}}, X^+_{\text{lft}}, X^+_{\text{rgt}})$ which gives a propulsion and torque of (-71 N, -1 Nm). This is an undesirable combination because it consumes a lot of energy, and the propulsion and torque can be achieved with a more economic combination.

6. $(X^-_{\text{main}}, X^+_{\text{lft}}, X^-_{\text{rgt}})$ which gives a propulsion and torque of (-129 N, 28 Nm). This is the minimum propulsion possible at the maximum torque.

7. $(X^-_{\text{main}}, X^-_{\text{lft}}, X^+_{\text{rgt}})$ which gives a propulsion and torque of (-125 N, -28 Nm). This is the minimum propulsion possible at the minimum torque.

8. $(X^-_{\text{main}}, X^-_{\text{lft}}, X^-_{\text{rgt}})$ which gives a propulsion and torque of (-183 N, 1 Nm). This is the minimum propulsion possible and implies a little torque.

Additionally, using equations (7.22), it is easy to extract the following results:

$$
\begin{aligned}
&\text{Prop} = -2\text{Torque} + 181 \qquad \text{if } X_{\text{main}} = X^+_{\text{main}} \text{ and } X_{\text{lft}} = X^+_{\text{lft}} \\
&\text{Prop} = +2\text{Torque} + 185 \qquad \text{if } X_{\text{main}} = X^+_{\text{main}} \text{ and } X_{\text{rgt}} = X^+_{\text{rgt}} \\
&\text{Prop} = -2\text{Torque} - 181 \qquad \text{if } X_{\text{main}} = X^-_{\text{main}} \text{ and } X_{\text{lft}} = X^-_{\text{lft}} \\
&\text{Prop} = +2\text{Torque} - 185 \qquad \text{if } X_{\text{main}} = X^-_{\text{main}} \text{ and } X_{\text{rgt}} = X^-_{\text{rgt}} \\
&\text{Torque} = +28 \qquad \text{if } X_{\text{lft}} = X^+_{\text{lft}} \text{ and } X_{\text{rgt}} = X^-_{\text{rgt}} \\
&\text{Torque} = -28 \qquad \text{if } X_{\text{lft}} = X^-_{\text{lft}} \text{ and } X_{\text{rgt}} = X^+_{\text{rgt}}
\end{aligned}
\tag{7.95}
$$

These equations represent the constraints of the torque and propulsion for the Guanay II AUV. Figure 7.18 summarizes all these limits.

7.6.1 Calculation of $(X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}})$ from $(\text{Prop}, \text{Torque})$

The calculation of the thruster forces from a propulsion and a torque represents a difficulty because the Guanay II uses 3 thrusters, so there are fewer equations than unknowns and the system is underdetermined. In this Section we propose the use of a cost function in order to do an optimal assignment.

The proposed cost function is

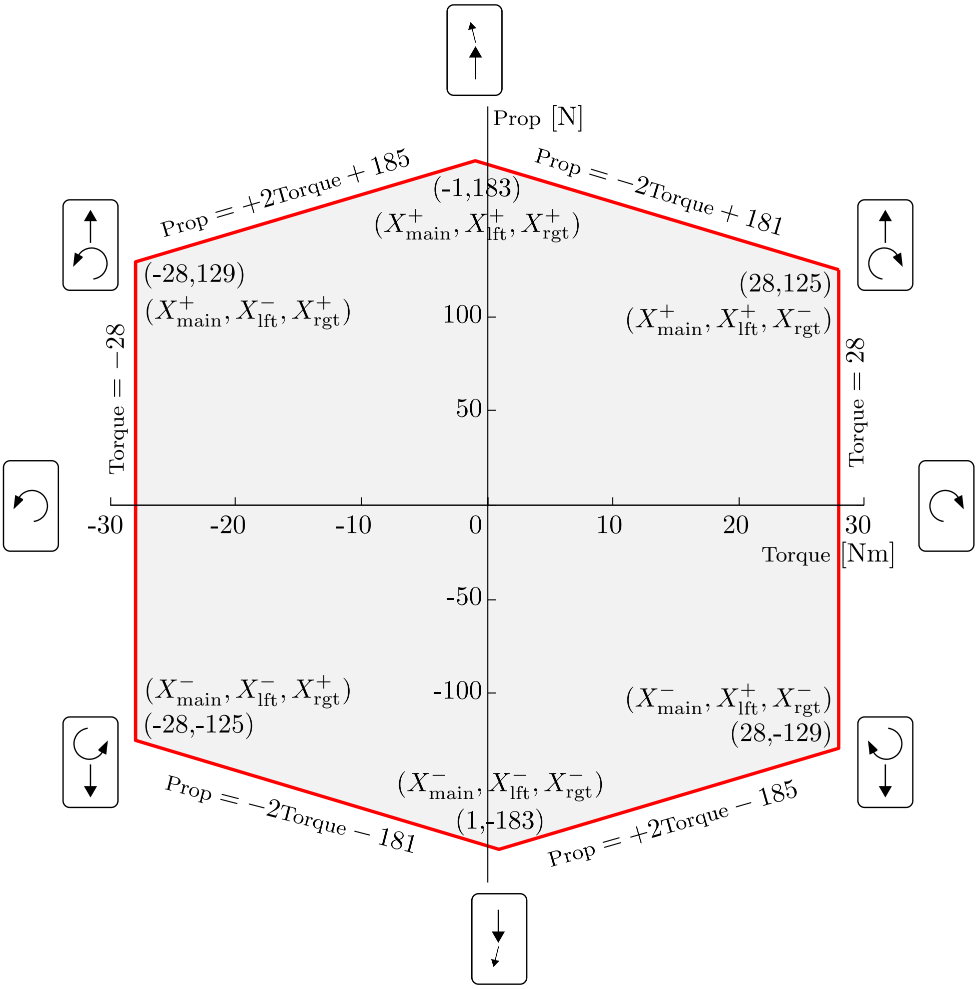

Prop [N]
Torque [Nm]
-30
-20
-10
0
10
20
30
100
50
-50
-100
Prop = +2Torque + 185
Prop = −2Torque + 181
Prop = −2Torque − 181
Prop = +2Torque − 185
Torque = −28
Torque = 28
(-1,183)
$(X^+_{\text{main}}, X^+_{\text{lft}}, X^+_{\text{rgt}})$
(-28,129)
$(X^+_{\text{main}}, X^-_{\text{lft}}, X^+_{\text{rgt}})$
(28,125)
$(X^+_{\text{main}}, X^+_{\text{lft}}, X^-_{\text{rgt}})$
$(X^-_{\text{main}}, X^-_{\text{lft}}, X^+_{\text{rgt}})$
(-28,-125)
$(X^-_{\text{main}}, X^+_{\text{lft}}, X^-_{\text{rgt}})$
(28,-129)
$(X^-_{\text{main}}, X^-_{\text{lft}}, X^-_{\text{rgt}})$
(1,-183)

Figure 7.18: Physical constraints of GUANAY II AUV

$$\text{Cost} = k_1|X_{\text{main}}| + k_2|X_{\text{lft}}| + k_3|X_{\text{rgt}}| \tag{7.96}$$

where k_i are the weights for the thrusters. For uniformity, we define these weights as $k_i = 1$. Using the equations (7.22) we can rewrite it as

$$\text{Cost} = k_1\left|\text{Prop} + \frac{\text{Torque}}{a_{\text{fin}}} - 2X_{\text{lft}}\right| + k_2\left|X_{\text{lft}}\right| + k_3\left|X_{\text{lft}} - \frac{\text{Torque}}{a_{\text{fin}}}\right| \tag{7.97}$$

Thus the function Cost is composed of a series of lines, which change according to their critical points. Knowing that this function is concave upward, one of these critical points is the value that minimizes the function, that is:

$$\begin{aligned} \text{Prop} + \frac{\text{Torque}}{a_{\text{fin}}} - 2X_{\text{lft}\,1} &= 0 \\ X_{\text{lft}\,2} &= 0 \\ X_{\text{lft}\,3} - \frac{\text{Torque}}{a_{\text{fin}}} &= 0 \end{aligned} \tag{7.98}$$

Solving these equations for X_{lft} and calculating X_{rgt} and X_{main} from it, there are three possible solutions that minimize the cost function:

$$(X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}}) = \begin{cases} \left(0\,,\ \frac{\alpha}{2}\,,\ \frac{\beta}{2}\right), \\ \left(\alpha\,,\ 0\,,\ -\frac{\text{Torque}}{a_{\text{fin}}}\right), \\ \left(\beta\,,\ \frac{\text{Torque}}{a_{\text{fin}}}\,,\ 0\right). \end{cases} \tag{7.99}$$

where

$$\alpha = \text{Prop} + \frac{\text{Torque}}{a_{\text{fin}}} \qquad \beta = \text{Prop} - \frac{\text{Torque}}{a_{\text{fin}}} \tag{7.100}$$

Notice that these critical points correspond with the triplets where $X_{\text{main}} = 0$, where $X_{\text{lft}} = 0$, and where $X_{\text{rgt}} = 0$. However, although one of them minimizes the function, the result may be outside the constraint region defined in 7.94. Take as an example a propulsion of 10 N and torque of 25 Nm, which give the critical points $(X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}}) = \{(0, 30, -20), (60, 0, -50), (-40, 50, 0)\}$. The combination that minimizes the cost function is $(0, 30, -20)$, but 30 exceeds the limit of the left thruster and the solution is incorrect. Thus, the possible solutions must include the constraints of the thrusters. If the cost function is composed by a series of lines but the critical points are out of our domain, the value that minimizes the function is one of the limits, that is $X_{\text{main}} = X^-_{\text{main}}$, $X_{\text{main}} = X^+_{\text{main}}$, $X_{\text{lft}} = X^-_{\text{lft}}$, $X_{\text{lft}} = X^+_{\text{lft}}$, $X_{\text{rgt}} = X^-_{\text{rgt}}$, and $X_{\text{rgt}} = X^+_{\text{rgt}}$. Using the equations (7.22) and solving these values, the possible solutions that minimize the cost function tacking into account the maximum and minimum power of the thrusters are

$$T = \{t_1, \ldots, t_9\} = \begin{cases} \left(0\,,\ \frac{\alpha}{2}\,,\ \frac{\beta}{2}\right), \\ \left(\alpha\,,\ 0\,,\ -\frac{\text{Torque}}{a_{\text{fin}}}\right), \\ \left(\beta\,,\ \frac{\text{Torque}}{a_{\text{fin}}}\,,\ 0\right), \\ \left(X^+_{\text{main}}\,,\ \frac{\alpha - X^+_{\text{main}}}{2}\,,\ \frac{\beta - X^+_{\text{main}}}{2}\right), \\ \left(\alpha - 2X^+_{\text{lft}}\,,\ X^+_{\text{lft}}\,,\ X^+_{\text{lft}} - \frac{\text{Torque}}{a_{\text{fin}}}\right), \\ \left(\beta - 2X^+_{\text{rgt}}\,,\ X^+_{\text{rgt}} + \frac{\text{Torque}}{a_{\text{fin}}}\,,\ X^+_{\text{rgt}}\right), \\ \left(X^-_{\text{main}}\,,\ \frac{\alpha - X^-_{\text{main}}}{2}\,,\ \frac{\beta - X^-_{\text{main}}}{2}\right), \\ \left(\alpha - 2X^-_{\text{lft}}\,,\ X^-_{\text{lft}}\,,\ X^-_{\text{lft}} - \frac{\text{Torque}}{a_{\text{fin}}}\right), \\ \left(\beta - 2X^-_{\text{rgt}}\,,\ X^-_{\text{rgt}} + \frac{\text{Torque}}{a_{\text{fin}}}\,,\ X^-_{\text{rgt}}\right). \end{cases} \tag{7.101}$$

where $t_i = (X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}})$ is a triplet (or 3-tuple) of the thrusters. From these solutions we take the triplets that are in the proper domain, that is

$$T_{\text{dom}} = \left\{ t = (X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}}) \in T \quad \middle| \quad \begin{aligned} X_{\text{main}} &\in [X^-_{\text{main}}, X^+_{\text{main}}], \\ X_{\text{lft}} &\in [X^-_{\text{lft}}, X^+_{\text{lft}}], \\ X_{\text{rgt}} &\in [X^-_{\text{rgt}}, X^+_{\text{rgt}}] \end{aligned} \right\}. \tag{7.102}$$

Finally, the triplet of thrusters that minimizes the cost function is

$$(X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}}) = \left\{ t_i \in T_{\text{dom}} \mid \mathsf{Cost}(t_i) = \min_{t \in T_{\text{dom}}} \mathsf{Cost}(t) \right\} \tag{7.103}$$

Continuing with the example, the best solution to reproduce a propulsion of 10 N and torque of 25 Nm, and that is within the constraints, is $(X_{\text{main}}, X_{\text{lft}}, X_{\text{rgt}}) = (6, 27, -23)$.

7.6.2 Steady state

An input $(\mathsf{Prop}, \mathsf{Torque})$ produces a change of the vector velocity $\boldsymbol{\nu}$ over time, and converges to a value called steady state. Figures 7.19, 7.20 and 7.21 show the calculation of the steady state in u, v and r respectively for different propulsion values and torques. From these we can conclude that the maximum forward velocity is about 1.1 m/s, and is achieved using all the thrusters at 100 %; and that the maximum angular velocity is about 0.2 rad/s (or 1.91 rpm) using one lateral thruster at 100 % and the other at -100 %.

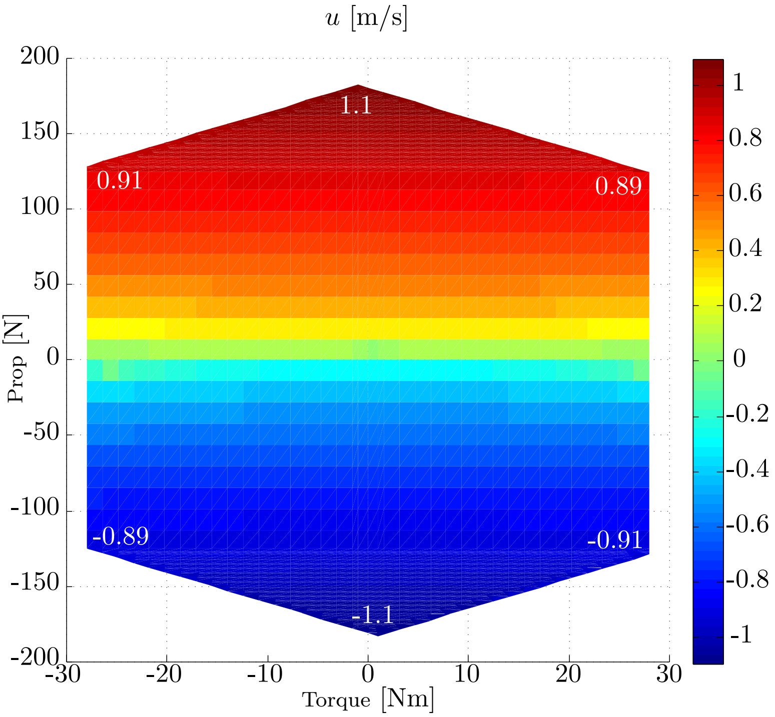


Figure 7.19: Steady state map — forward velocity u

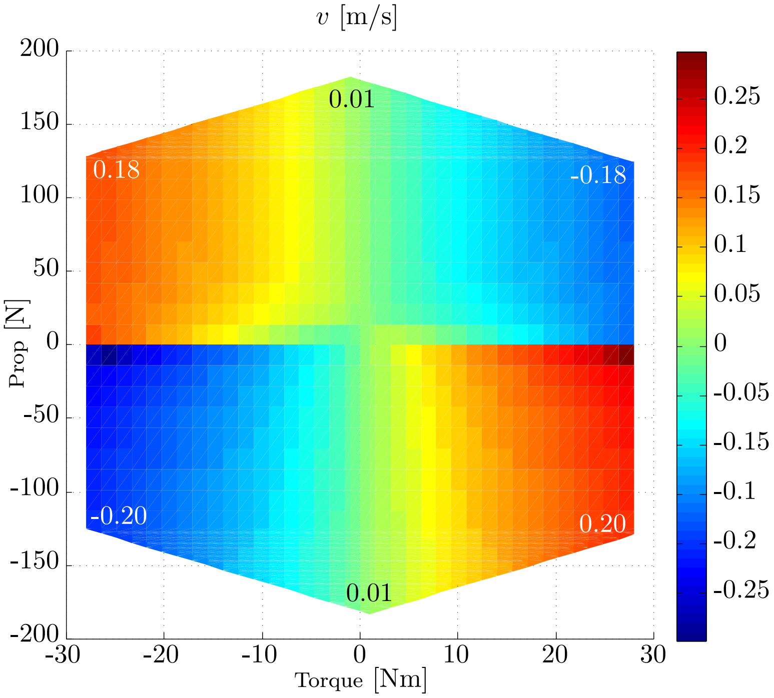


Figure 7.20: Steady state map — lateral velocity v

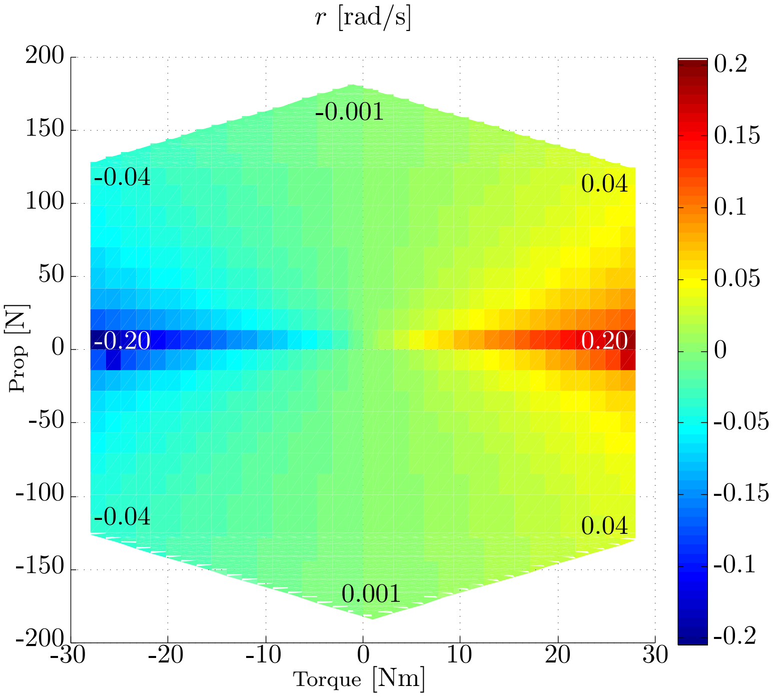


Figure 7.21: Steady state map — angular velocity r

7.6.3 Radius of curvature

When the vehicle turns it describes a certain radius R. The path traveled is calculated as

$$l = R\theta,$$

where θ is the angle of the circular section given in radians. It is easy to see that the derivative of θ is the same angular velocity r, because the vehicle's movement is perpendicular to the radius. Also, the derivative of l corresponds to the norm of the velocity. Then

$$\begin{aligned}\frac{dl}{dt} &= R\frac{d\theta}{dt}\\ \sqrt{u^2+v^2} &= Rr\\ R &= \frac{\sqrt{u^2+v^2}}{r}.\end{aligned} \tag{7.104}$$

While r increases the radius decreases, and while u increases the radius increases. This leads us to calculate the maximum velocity u possible given a specific radius R. We have calculated different radii at steady state from the velocities at steady state. From this calculation we have extracted the maximum values for u. Figure 7.22 displays the relation of them respect the radius R.

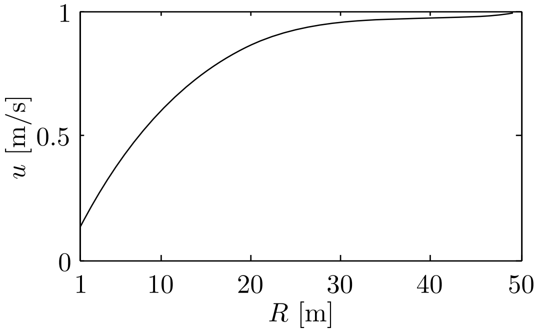


Figure 7.22: Maximum forward velocity u regarding the radius of curvature

Using a polynomial regression, we obtain that

$$u_{\max} = 0.0000162R^3 - 0.00183R^2 + 0.0698R + 0.0677. \tag{7.105}$$

7.7 Conclusions

In this chapter we have related several contributions to the hydrodynamic model of the Guanay II AUV. First we focused on a model proposed by Prestero, which does not contemplate the linear damping matrix. Next, this model was linearized and an analysis of stability was developed. We have seen that the transfer functions of the yaw respect to the torque has a zero located at the origin, which is not good because the angular velocity does not tend to a constant value when a step is applied.

In order to solve this problem we have taken into account a hydrodynamic model with the linear damping matrix. Again, a linearization and an study of stability were made. In this case the response of the yaw is more situated to the desired values.

After that we have dealt with the calculation of the coefficients of the Guanay II, first using a theoretical method that takes into account the geometry of the vehicle. However, simulations with these coefficients were not good at all, so we used a method, based on least squares of experimental data, to calculate them. The results were very good, with simulations coinciding with experimental data with a very small error.

Finally, we have dealt with the physical constraints of the vehicle. The limits are shown in a map of propulsion and torque. At the same time we showed different maps of the steady state of the velocity components using all ranges of propulsion and torque. We closed the constraints analysis with a study about the minimum radius of curvature respect to the forward velocity.

Chapter 8

Automatic Control

Guanay II must be capable of autonomous navigation. After defining and scheduling a path, the vehicle must be able to follow it without human intervention, and to cope with any environmental perturbations present in the sea. Works of Fossen [43] and Antonelli et al. [9] present the concept of *Guidance, Navigation and Control System (GNC)*, which is the set of programs responsible of carrying out a mission. As the name says, it is divided into three main layers or subsystems (see figure 8.1):

- *Guidance system.* This is the high level control of the vehicle during the mission. The mission is decomposed into a sequence of sub-goals which generate the desired settings: a reference trajectory (position, velocity and acceleration). It has a waypoint generator which establishes the desired waypoints according to information about the mission, operator decision, weather, amount of power available, etc.

  There are several lines of research in this area, among which one can cite path planning, mission planning, obstacle avoidance, and multi-vehicle collaboration.

- *Navigation system.* This system receives the sensor data and computes the actual position, velocity and acceleration.

  Lines of research work on GPS/IMU integration, short baseline (SBL), ultra-short baseline (USBL), long baseline (LBL), simultaneous localization and mapping (SLAM), among others (see [96] [101] [11] [111]).

- *Control system.* This system processes information to infer the state of the vehicle and to generate an appropriate command for the actuators so as to reduce the difference between the actual and desired trajectories.

A complete survey of GNC systems can be found in Kendowl [64], and Moreno et al. [89]. The form to combine and to manage these systems relies on the *control architecture.* See [113], [89], [99], [85] for further details and classification.

We reviewed in chapter 5 the software of the vehicle. This corresponds with the *guidance system* of the Guanay II. It has the high level management that communicates with the operator and defines a set of instructions or sub-goals. In this chapter on automatic control we will focus in the intermediate layer, the *control system.*

The *control system* can have different operation modes, and depending on the type of operation performed by the vehicle, the controller can combine (or be switched into) the different modes: *path following*, *path tracking*, *trajectory tracking*, *point stabilization*, *pure*

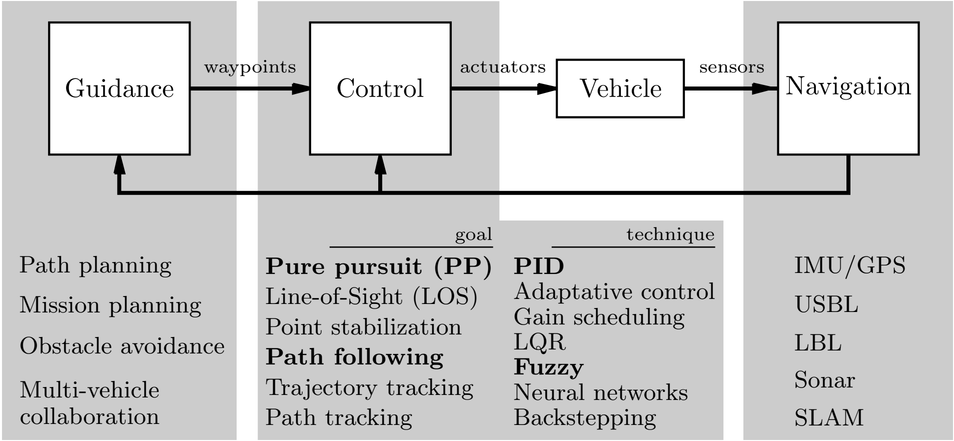


Figure 8.1: Guidance, Navigation and Control, and the main associated research lines. This chapter focuses on the topics in bold.

pursuit, *line-of sight* and *constant bearing*. These type of movements are tightly linked with the guidance system. Further details can be found in Breivik and Fossen [20], and Duc and Pan [30].

The way to deal with these control problems covers all type of controllers, including linear control [150] [12], sliding control [161] [123] [130], fuzzy control [128] [31] [108], gain scheduling control [119] [149]; LQR [84], adaptative control [6] [7]; robust control [26], nonlinear laws based on Lyapunov functions (backstepping) [18] [8] [36] [71] [29], and neural networks [65] [56] [41], among others.

Most of the works about the state of the art [150] [84] [123] design the controllers referencing the hydrodynamic model of the vehicle when it travels at a specific forward velocity in order to simplify the control design. This is right for a vehicle that habitually navigates at open sea where its velocity will be mainly constant. However, if it navigates near the coast, on the sea floor, or in the interior of ports and channels, the variation of the forward velocity becomes relevant, and it is then important to be able to vary working point of the controller in order to adjust the paths to these environments and keep an uniform performance across several environments.

Some of the works that propose a solution to this problem use techniques based of Lyapunov functions [71] [19]. These solutions lead to a loss of simplicity of the control laws. Moreover, these nonlinear techniques do not enjoy the same diffusion and popularity as their linear counterparts.

An alternative is the work of Silvestre and Pascoal [125], where they design linear controllers for different forward velocities, and thereafter use a gain scheduling controller to integrate them. The application of this methodology is the motivation of the present work, but applying a fuzzy controller to integrate the different linear controllers.

Regarding fuzzy controllers, several papers use it for obstacle avoidance, like in Dong et al. [31] and Liu et al. [75], who design memberships in function of the distance to the obstacle and the forward velocity. On the other hand, there are other works focused on using the advantages of Gain Scheduling controllers but applying the fuzzy framework to manage them, like in Zhang et al. [163] and Jun et al. [62], who develop a fuzzy controller for AUVs in a way that it manages several linear controllers to be actuated at specific conditions of velocity.

However, *they do not calculate the parameters of the controller using an analytic procedure but using a linguistic interpretation.* In Reddy et al. [109] we can see a comparative of the fuzzy controller technique in regard to gain scheduling. They show that fuzzy controllers have similar behaviour and performance when compared to gain scheduling ones.

Thus, due to the good performance of the fuzzy controller serving as interpolator of linear controllers, we propose in this work the use of a fuzzy controller of type TSK in order to manage different linear controllers designed for specific conditions of forward velocity. We also present an analytic development to calculate different parameters. The fuzzy controller allows us to establish activation zones (understood as the effective contribution of one rule of the controller) which can be controlled through fuzzy sets (Driankov et al. [32]; Takagi and Sugeno [138]; Mittal et al. [88]).

For the *control system*, we split the controller into two loops: inner loop and outer loop. The first loop is responsible for setting the yaw ψ and the forward velocity u, given a reference $(\psi_{\text{ref}}, u_{\text{ref}})$. At this level we develop the fuzzy controller mentioned above. The second loop is responsible to set the reference for yaw and forward velocity for a given path.

The structure of this chapter is as follows: First we deal with linear controllers to control the yaw and forward velocity on the inner loop. These controllers are based on the hydrodynamic model. Next, we introduce the fuzzy controller as interpolator of these controllers. This development is compared with the gain scheduled controller. Next we present some problems related to the physical limitations of the GUANAY II and we propose a change in the inner loop for heading using a fuzzy controller based on the yaw error. From this point we deal with the outer loop, which consists of two methods to implement a *pure pursuit*, and one method to implement *path following*.

8.1 Review of control system

8.1.1 Basic motion tasks

The classification of the control system regarding the motion is divided into 7 tasks. They are very linked with the guidance system because they define the type of movement, but we classify them in the control system because they start after the definition of the way-points.

Trajectory tracking

Trajectory tracking is a frequent control problem both in marine vehicles and in land and aerial vehicles. Its principal goal is to follow a virtual vehicle, that is, to follow a trajectory with timing conditions.

Let $p(t)$ be the actual position of the vehicle, and let $p_d(t) : [0, \infty]) \rightarrow \mathbb{R}^2$ be the position of the reference virtual vehicle. The error

$$e = \|p(t) - p_d(t)\| \tag{8.1}$$

needs to be decreased to zero in a finite time [2].

Path following

Path following is similar to *trajectory tracking* but without the timing restrictions. Its goal is to follow a geometric reference path starting from a given initial configuration.

Let $p_d(\zeta) \in \mathbb{R}^2$ be a path parametrized by $\zeta \in \mathbb{R}$, and let $v_r(\zeta) \in \mathbb{R}$ be a desired velocity. The error

$$e = \|p(t) - p_d(\zeta(t))\| \tag{8.2}$$

needs to be decreased to zero [37]. Notice that there is not an explicit restriction on time.

Path tracking

Path tracking, like *trajectory tracking*, has a timing law to converge to the path, but not necessarily using a virtual vehicle. Encarnaçao and Pascoal [37] present it as a combination between *trajectory tracking* and *path following*. The desired position $p_d(\pi)$ is parametrized by π, which is defined as follows.

$$\pi(p,t) = \arg\min_{\zeta\in\mathbb{R}} \left[(1-\lambda^2)||p - p_d(\zeta)||_P^2 + \lambda^2(t-\zeta)^2\right] \tag{8.3}$$

where λ is a weight used to interpolate between the two motion tasks. In this way, when λ is zero we obtain *path following*, but when λ is 1 we obtain *trajectory tracking*.

Point stabilization

Point stabilization consists in stabilizing the position of a vehicle at a given point and attitude. For vehicles with nonholonomic constraints (like GUANAY II) it is not achievable with constant state-feedback control and they need of using more complex controllers [21]. This is not a problem for fully-actuated vehicles, contrary to underactuated ones [52].

Pure pursuit (PP)

Pure pursuit guidance only considers the target (waypoint) and the vehicle. The method is quite intuitive, whereas the heading angle is calculated only based on the current position of the vehicle and the next waypoint. The concept of pure pursuit can be traced in early technical literature regarding the problem of a missile pursuing a target [121]. In the pure pursuit course, the missile velocity vector is always directed toward the instantaneous target position.

Line-of-Sight (LOS)

The basic idea behind *line-of-sight (LOS)* guidance algorithms is to define a LOS setpoint on the straight line between two waypoints $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$. The vector from the vehicle's current position to the setpoint is known as the LOS vector and its direction implicitly also becomes the course of the vehicle [60].

All of these motion tasks are summarized in figure 8.2. Take into account that it is possible to combine these tasks, for instance, LOS with path following [44].

8.1.2 Fuzzy logic

Fuzzy logic is a form of many-valued logic which deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values), fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions. The term "fuzzy logic" was introduced by Zadeh [162].

A fuzzy set μ of X is a function from the reference set X to the unit interval, i.e.

$$\mu : X \to [0,1].$$

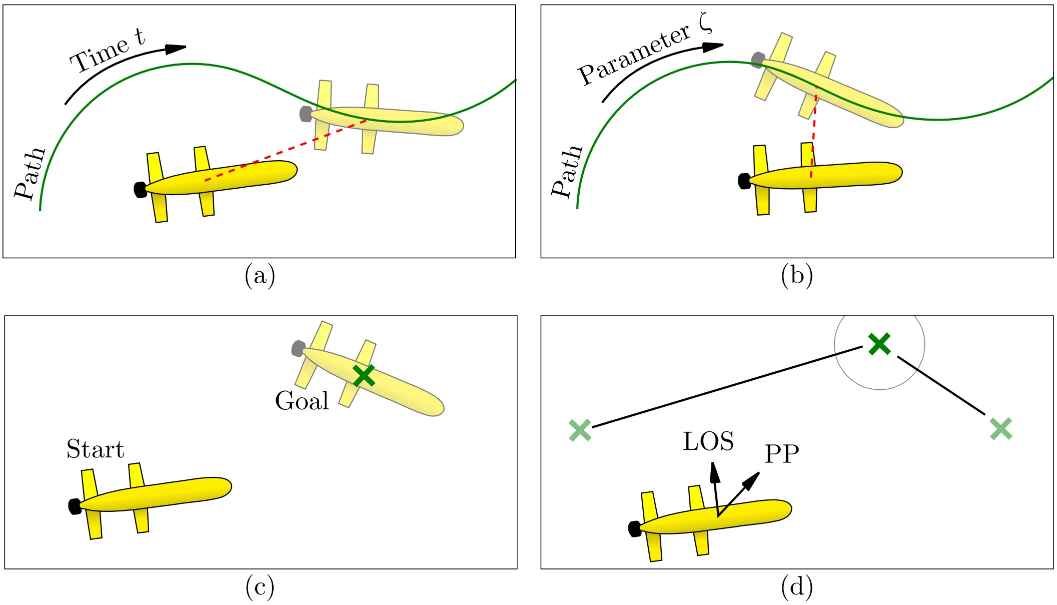


Figure 8.2: Basic motion tasks. a) Trajectory/path tracking. b) Path following. c) Point stabilization. d) Pure pursuit and Line-of-Sight

$F(X)$ denotes the set of all fuzzy sets of X. In general, fuzzy sets merely have an intuitive basis as a formal description of vague data. Figure 8.3 shows an example of this concept. A human can be considered 'tall' if his height is near to 2 m, but he is not 'tall' if his height is 1 m. In the case of 1.5 m we say that the condition of 'tall' is partial truth, and numerically is 0.7.

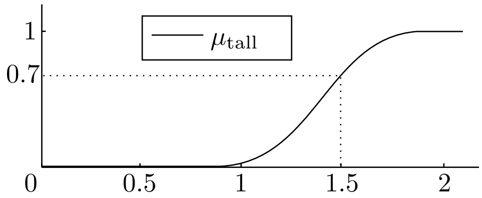


Figure 8.3: Representation of a fuzzy set

The basic operations on fuzzy sets are intersection, union and complement. They are defined as follows (see figure 8.4).

$$\begin{aligned} (\mu \cap \mu')(x) &= \min\{\mu(x), \mu'(x)\} && \text{(intersection)} \\ (\mu \cup \mu')(x) &= \max\{\mu(x), \mu'(x)\} && \text{(union)} \\ \bar{\mu}(x) &= 1 - \mu(x) && \text{(complement)} \end{aligned} \tag{8.4}$$

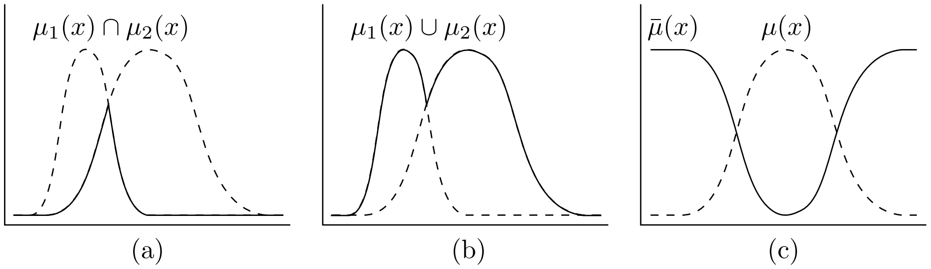


Figure 8.4: Basic operations on fuzzy sets. a) intersection. b) union. c) complement

Types of fuzzy sets

There are two types of fuzzy sets, namely: Type-1 and type-2. The first one has memberships functions in two dimensions, that is, the membership grade for each element is a crisp number in $[0, 1]$.

The type-2 fuzzy set is characterized by a three dimensional membership function and a footprint of uncertainty (FOU) [120] [61]. Hence, the membership value (or membership grade) for each element of this set is a fuzzy set in $[0, 1]$. This allows to associate uncertainties of the sensors measurements, linguistic uncertainties (words means different things to different people), or uncertainties associated with the change in the operation conditions due to varying load and environment conditions [53]. Figure 8.5 shows a graphical representation of a type-2 fuzzy set. It is bounded by a upper membership function (UMF) and a lower membership function (LMF).

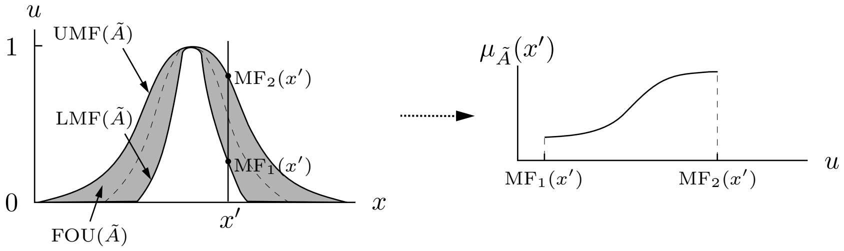


Figure 8.5: type-2 fuzzy set

8.1.3 Fuzzy control

As explained by Kruse [69], fuzzy control is seen as a way of defining non-linear table-based control systems, where the definition of the non-linear transition function can be made without the need to specify each entry of the table individually. The development can be viewed as a kind of knowledge-based interpolation technique. The analysis of the table-based controller can be performed by standard methods of control engineering.

Let $\xi_1, \ldots, \xi_n$ by the input variable, which consists of the measurements of different sensors. Let γ by the control variable. The measurements are used to determine an actual value of γ. In addition, we assume that the measured input ξ_i, $i = 1, \ldots, n$, is a value of the set X_i, and that the control variable is in the set Y. The solution of the control problem should be a control function $\phi : X_1 \times \ldots \times X_n \to Y$ that assigns to each input tuple $(x_1, \ldots, x_n) \in X_1 \times \ldots \times X_n$ an adequate control value $y = \phi(x_1, \ldots, x_n)$.

The idea of fuzzy control is to simulate the behaviour of a person who is able to control the given process. The development of a model of a human 'control expert' is called *knowledge-based analysis*. An adequate architecture for the knowledge-based model of a controller is illustrated in figure 8.6.

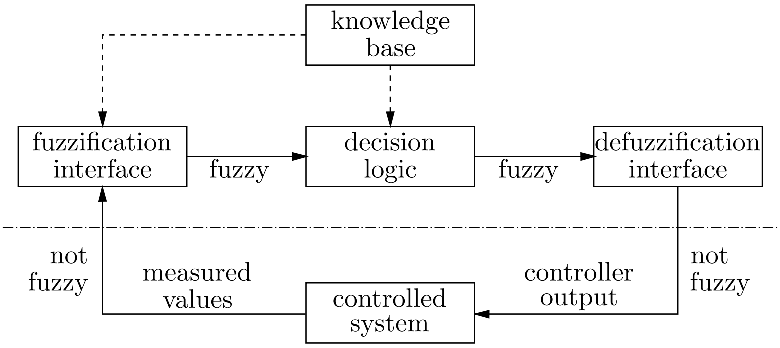


Figure 8.6: Architecture of a fuzzy controller

- The *fuzzification interface* receives the current input variable and converts it into a linguistic term or into a fuzzy set.
- The *knowledge base* contains information about the boundaries, and the fuzzy sets with their corresponding linguistic terms. In addition, the knowledge base contains a *rule base* consisting of linguistic control rules.
- The *decision logic* represents the processing unit. It determines the corresponding output value from the measured input according to the knowledge base.
- The *defuzzification interface* has the task of determining a crisp output value — taking the information about the control variable provided by the decision logic into account. Finally, if necessary, it carries out a transformation of the output value into the appropriate domain.

An expert has to specify his knowledge in the form of linguistic rules. First, we have to determine for each input domain $X_1, \ldots, X_n$ and the output Y appropriate linguistic terms like *approximately zero*, *positive small*, etc. In the mathematical model each linguistic term has to be represented by a fuzzy set. For this reason we define distinct p_1 fuzzy sets $\mu_1, \ldots, \mu_{p_1} \in F(X_1)$ on the set X_1 and associate a linguistic term to each set.

Next we specify the rule base. A single rule has the form

$$R_m\text{: } \textbf{if } \xi_1 \textbf{ is } A_1 \textbf{ and } \ldots \textbf{ and } \xi_n \textbf{ is } A_n \textbf{ then } \gamma \textbf{ is } B$$

where $A_1, \ldots, A_n$ and B represent the linguistic terms that correspond with the fuzzy sets μ according to the fuzzy partitions of the set $X_1 \times \ldots \times X_n$, and Y. The operation **and** is computed using the intersection. Then, letting α_r by the degree of applicability of rule R_m we obtain

$$\alpha_m = \min\{\mu_{i_{1,m}}(x_1), \ldots, \mu_{i_{n,m}}(x_n)\} \tag{8.5}$$

where $\mu_{i_{1,m}}(x_1), \ldots, \mu_{i_{n,m}}(x_n)$ are all the fuzzy sets of the input variables involved in the rule R_m.

The fuzzy set of the output is

$$\begin{aligned} \mu^{\text{output}(R_m)}_{x_1,\ldots,x_n} : Y \to& [0,1], \\ y \mapsto& \min\{\alpha_m, \mu_{i_m}(y)\} \end{aligned} \tag{8.6}$$

where $\mu_{i_r}(y)$ is the fuzzy set of the linguistic term of output.

At the end we will have k rules that determine the knowledge base. For each rule we will have a degree of applicability and a fuzzy set for the output. The way to combine all the fuzzy sets obtained from the rules into one output fuzzy set is determined by the maximum of all of them (union operator). Finally the decision logic yields the fuzzy set

$$\begin{aligned} \mu^{\text{output}}_{x_1,\ldots,x_n} : Y \to& [0,1], \\ y \mapsto& \max_{m\in\{1,\ldots,k\}} \Big\{ \min\{\alpha_m, \mu_{i_m}(y)\} \Big\}. \end{aligned} \tag{8.7}$$

There are two different inference systems to compute the output variable γ.

Mamdani's approach

This approach is widely used. It establishes three ways to compute the defuzzification:

- *The Max Criterion Method.* Using this method, we choose an arbitrary value $y \in Y$ for which the fuzzy set $\mu^{\text{output}}_{x_1,\ldots,x_n}$ reaches a maximum membership function. This is defined as follows.

$$\gamma = \text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right) = \{y \in Y \mid \forall y' \in Y : \mu^{\text{output}}_{x_1,\ldots,x_n}(y') \leq \mu^{\text{output}}_{x_1,\ldots,x_n}(y)\} \tag{8.8}$$

- *The Mean of Maxima Method (MOM).* A supposition for the mean of maxima method is that Y is an interval and the set $\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)$ is non-empty and measurable. If the set $\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)$ is finite, we obtain the output value by the formula

$$\gamma = \frac{1}{\left|\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)\right|} \sum_{y\in\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)} y, \tag{8.9}$$

  which may be written as

$$\gamma = \frac{1}{\int\limits_{y\in\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)} dy} \cdot \int\limits_{y\in\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)} y\, dy \tag{8.10}$$

  in the case of an infinite set $\text{Max}\left(\mu^{\text{output}}_{x_1,\ldots,x_n}\right)$.

- *Centre of Gravity Method (COG).* Using the centre of gravity demands the same suppositions as the mean of maxima method. We define the value located under the centre of gravity of the area that is given by the function $\mu^{\text{output}}_{x_1,\ldots,x_n}$ as output value γ. Technically this value is computed by the formula

$$\gamma = \frac{1}{\int\limits_{y\in Y} \mu^{\text{output}}_{x_1,\ldots,x_n}(y)dy} \cdot \int\limits_{y\in Y} y \cdot \mu^{\text{output}}_{x_1,\ldots,x_n}(y)dy. \tag{8.11}$$

  One important advantage of this method is that the controller shows nearly always a smooth behaviour.

The approach of Takagi-Sugeno-Kang

The approach to fuzzy control first proposed by Takagi and Sugeno (TSK) can be seen as a modification of Mamdani's model.

For Sugeno fuzzy controller, fuzzy partitions for the input domains have to be specified in the same way as for the Mamdani's fuzzy controller. A fuzzy partition of the output domain is not needed, since the control rules are given in the form of

$$R_m\text{: } \textbf{if } \xi_1 \textbf{ is } A_1 \textbf{ and } \dots \textbf{ and } \xi_n \textbf{ is } A_n \textbf{ then } \gamma = f_m(\xi_1, \dots, \xi_n)$$

where f_m is a mapping from $X_1 \times \ldots \times X_n$ to Y, $m = 1, \ldots, k$. Generally it is assumed that f_m is linear, i.e. $f_m(x_1, \ldots, x_n) = a_1x_1 + \cdots + a_nx_n + a_{n+1}$, and it is called of first order.

The task of the decision logic is to determine the degree of applicability α_m of each premise and to compute the value $f_m(x_1, \ldots, x_n)$ for the input tuple $(x_1, \ldots, x_n) \in X_1 \times \ldots \times X_n$ for each rule R_m. This is done using (8.5) for type-1 TSK fuzzy logic systems. In addition the decision logic delivers the crisp control value γ according to the formula

$$\gamma = \frac{\sum_{m=1}^{k} \alpha_m \cdot f_m(x_1, \ldots, x_n)}{\sum_{m=1}^{k} \alpha_m}. \tag{8.12}$$

This controller does not has deffuzifier, which reduces the computational cost greatly. Since the output is calculated using a crisp function it does not use linguistic terms and it can be used as an interpolator of controllers clearly differentiated [88]. These controllers can be designed using analytic developments.

An example of this control can be shown in figure 8.7. The system is nonlinear and it has two state variables, x and y. Consider two linear controllers, LC$_1$ and LC$_2$, which have associated the equilibrium points (x_1, y_1) and (x_2, y_2). These controllers guarantee the stability of the system in a closed zone near to the equilibrium point. Outside of this linear zone the fuzzy controller (FC) acts as an interpolator using the fuzzy sets to define de zones, and using the rules to establish the decision of the knowledge base.

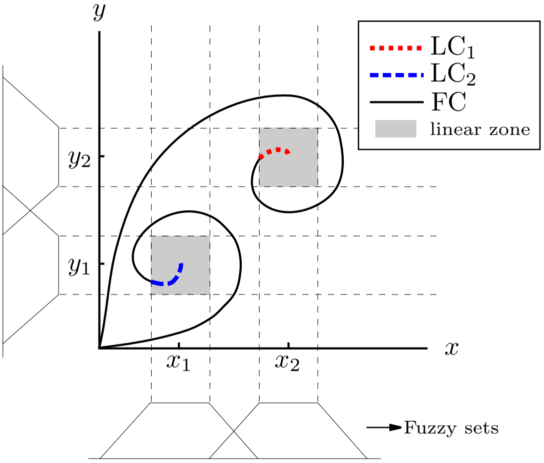


Figure 8.7: Map of control laws using a type-1 TSK fuzzy controller

8.2 Linear controls for different velocities

A linear control is a simple way to control the variables of a linear system. Its design is well documented. However, the Guanay II AUV is a highly non-linear system and, consequently, if we design a linear control for one linearization of the vehicle around to a specific velocity, say u_1, the performance will be optimal near this velocity u_1, but not if the vehicle has a very different velocity u_2.

In this section we propose different linear controls, since a non-linear system can be approximated by a piecewise linear model. For this, it is necessary to introduce the following definition.

Definition. It is said that a set of linear controls is *zonally differentiated* if it can be shown that each control is more optimal than the others in a specific zone. That is, one control has good performance in the conditions for which it was designed, but not the others (which have good performance in other zones). For marine vehicles, these zones represent the different forward velocities u for which the model is linearized.

In this section we develop a set of zonally differentiated controls in order to control the yaw angle of the Guanay II AUV for different forward velocities.

8.2.1 Transfer function of the yaw

The yaw angle ψ can be calculated through the angular velocity r through the rotation matrix $\mathbf{R}$ (see equation 7.3), which gives

$$r = \dot{\psi}. \tag{8.13}$$

Applying the Laplace transform we have

$$r(s) = s\,\psi(s). \tag{8.14}$$

Using this equation and equation (7.64), the transfer function of the yaw respect to the torque can be calculated as

$$\underset{u_0}{G_\psi}(s) = \frac{\psi(s)}{\text{Torque}(s)} = \frac{m_v s - Y_v}{As^3 + Bs^2 + Cs} \tag{8.15}$$

where the subindex u_0 in the notation $G_{\psi_{u_0}}(s)$ represents the velocity at which the model is linearized.

This general transfer function has one zero and three poles. Notice that the denominator depends of the forward velocity u_0 (the values A, B, and C are defined in equation 7.61), but the numerator does not. Notice also that one pole is an integrator. Then, using the coefficients of the vehicle, the model is linearized at different velocities. In this case, we opted for three velocities: 0.3, 0.9 and 2 m/s, which have been selected as an abstraction of “low”, “high” and “very high” velocity, respectively. Equations (8.16) show the transfer functions obtained and figure 8.8 shows the position of their poles and zeros.

Notice that one pole is very close to zero in all the cases. Hence, the system can be simplified to a transfer function with one integrator, one real pole, and no zeros.

$$\begin{aligned} \underset{0.3}{G_\psi}(s) &= \frac{0.003323s + 0.000258}{s^3 + 0.8107s^2 + 0.05834s} \\ \underset{0.9}{G_\psi}(s) &= \frac{0.003323s + 0.000258}{s^3 + 1.9490s^2 + 0.21170s} \\ \underset{2.0}{G_\psi}(s) &= \frac{0.003323s + 0.000258}{s^3 + 4.0350s^2 + 0.73540s} \end{aligned} \tag{8.16}$$

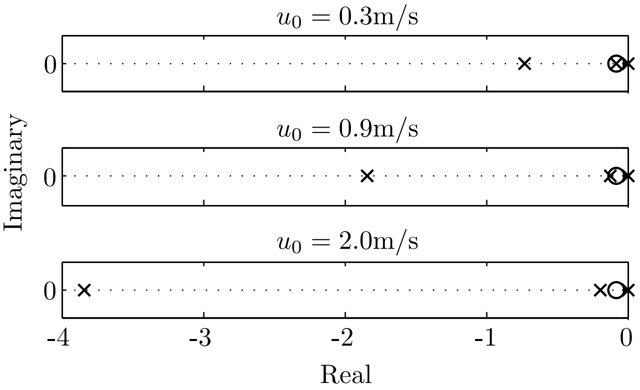


Figure 8.8: pole-zero map of the transfer function $G_{\psi_{u_0}}(s)$ for different u_0

8.2.2 Yaw control

The yaw control consists of a definition of a reference ψ_{ref}, which is compared with the actual angle ψ in order to establish an error. The principal idea is to bring this error to zero using a controller $C(s)$ that actuates the lateral thrusters. The general block diagram can be seen in figure 8.9.

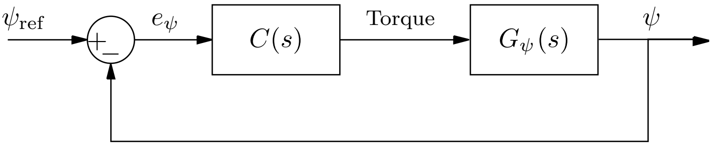


Figure 8.9: Block diagram of the closed loop system for the yaw control

The the transfer function of the yaw with respect to the reference, $H_\psi(s)$, can be calculated as

$$H_\psi(s) = \frac{\psi(s)}{\psi_{\text{ref}}(s)} = \frac{C(s)\,G_\psi(s)}{1 + C(s)\,G_\psi(s)}. \tag{8.17}$$

P controller

A proportional controller consists in scaling the error e_ψ with a constant k to actuate the thrusters:

$$\underset{u_0}{C_{\text{P}}}(s) = k \tag{8.18}$$

where the subscript P denotes “proportional” and u_0 is the velocity at which the model has been linearized to design the controller.

This type of controller is good to control the yaw since for this type of plant zero error is guaranteed in steady state in presence of a step as input. A simple calculation of the final value theorem is enough to show it.

On the other hand, the root locus of this feedback system consists in the movement of three poles. From figure 8.8 we can extract that the integrator will go to zero, and the remaining two poles will meet each other at a breakaway point. From this point, not having more zeros, two vertical branches are created which go to infinity. The closed loop poles can be calculated by equating the denominator of $H_\psi(s)$ to zero:

$$\begin{aligned} 1 + C_\mathrm{P}(s)\, G_\psi(s) &= 0 \\ 1 + k\frac{m_v s - Y_v}{As^3 + Bs^2 + Cs} &= 0 \\ As^3 + Bs^2 + Cs + k(m_v s - Y_v) &= 0 \\ As^3 + Bs^2 + (C + m_v k)s - Y_v k &= 0 \end{aligned} \tag{8.19}$$

For the controller design, the breakaway point is the best position for the poles because the dominant pole is as far away as possible from the imaginary axis, which gives a fast response. Also, as they are real, the response does not present overshoot or oscillations. This point is achieved when the two poles are the same, that is

$$\begin{aligned} (s + a)^2(s + b) &= 0 \\ s^3 + (2a + b)s^2 + (a^2 + 2ab)s + a^2 b &= 0 \end{aligned} \tag{8.20}$$

Comparing the equations (8.19) and (8.20), we obtain the system of equations

$$\begin{aligned} 2a + b &= B/A \\ a^2 + 2ab &= (C + m_v k)/A \\ a^2 b &= -kY_v/A \end{aligned} \tag{8.21}$$

which has three unknowns, a, b, and k.

Solving these equations for the values of the transfer functions (8.16), we obtain

$$\begin{aligned} \underset{0.3}{C_\mathrm{P}}(s) &= 39.8731 \\ \underset{0.9}{C_\mathrm{P}}(s) &= 241.5303 \\ \underset{2.0}{C_\mathrm{P}}(s) &= 1043.8723 \end{aligned} \tag{8.22}$$

Figure 8.10 shows the root locus of the linearized systems for 0.3, 0.9 and 2 m/s using a proportional controller. Particularly, it shows the pole displacements using the three controllers mentioned above. For the first system $G_{\psi_{0.3}}(s)$, the controller $C_{\mathrm{P}0.3}(s)$ moves the poles to the breakaway point, but the other controllers have a high gain that move the poles to the vertical branches. In the second root locus for $G_{\psi_{0.9}}(s)$, the controller that moves the poles to the breakaway point is $C_{\mathrm{P}0.9}(s)$, while $C_{\mathrm{P}0.3}(s)$ is a low gain and $C_{\mathrm{P}2.0}(s)$ is very high. Finally, the root locus for the linearization $G_{\psi_{2.0}}(s)$ shows that $C_{\mathrm{P}2.0}(s)$ is its best controller, while the other two have a low gain.

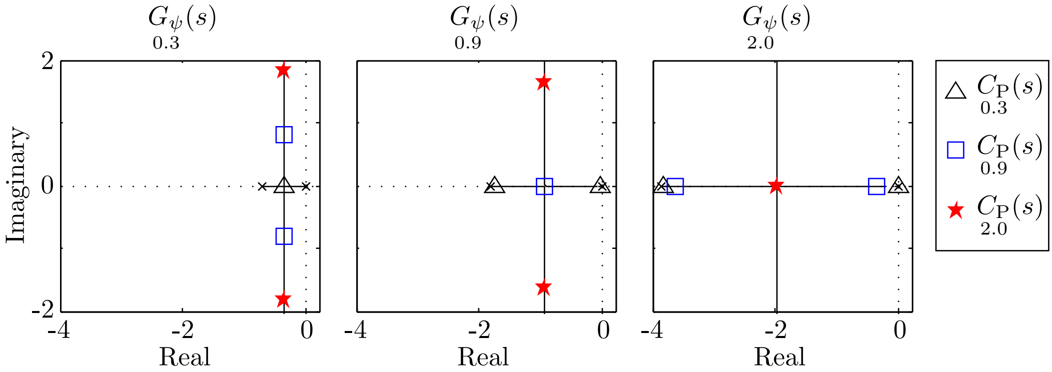


Figure 8.10: Root locus of the different $G_{\psi}(s)$ and pole displacement using the controllers $C_P(s)$

Although these root locus show that the controllers seem to be zonally differentiated, it is important to test their performance in time. Figure 8.11 shows the step responses of the different linearizations using the different controllers in a closed loop. Travelling at 0.3 m/s, the controller $C_{P0.3}(s)$ does not yield any overshoot. However, controller $C_{P0.9}(s)$ seems to provide a better response, although it has some overshoot. A similar situation is obtained in the second linearization: at 0.9 m/s the controller designed for this case, $C_{P0.9}(s)$, yields a good response without overshoot. However, controller $C_{P2.0}(s)$ leads to a lower settling time and to a small overshoot. Finally, the responses in the third linearization, at speed 2 m/s, clearly show that the best controller is $C_{P2.0}(s)$, since it has the lower settling time and has no overshoot.

Knowing these particular situations, we can say that this set of proportional controllers is not zonally differentiated.

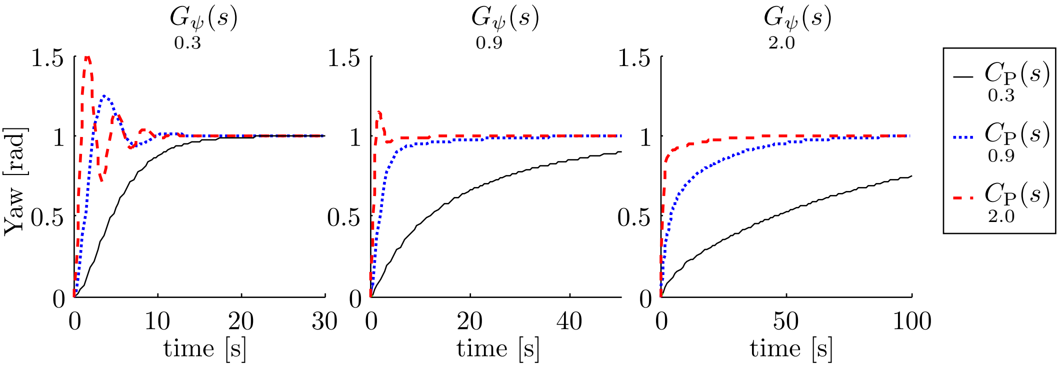


Figure 8.11: Step response of the different feedback systems using the controllers $C_P(s)$

PD Controller

A proportional-derivative controller (PD) consists of a gain for the yaw error e_ψ and a zero which allows fast response:

$$\underset{u_0}{C_{\mathrm{PD}}}(s) = k_d s + k_p \tag{8.23}$$

where the subscript PD denotes "proportional-derivative" and u_0 is the velocity chosen for the linearization. This type of control has the great advantage that it can move the two dominant poles to more negative values, which represents better performance.

The root locus has two types of diagrams for stable systems:

1. Zero located between the two poles. In this case, the dominant pole will go to zero, and the other to infinity. The advantage is that no matter the gain, the resulting poles are real. The disadvantage is that the resulting position of the dominant pole, and consequently the speed of response, is limited by the position of the zero.

2. Zero located to the left of the two poles. In this case, the movement of the poles has three parts. First, the poles approach each other until they reach a breakaway point; then one has two complex branches enclosing the zero in an oval; and third one real pole goes to zero and the other to infinity. The advantage with this design is that the two poles can be at the left of the zero, and consequently one gets a fast response. The disadvantage is that the fast response requires high gains, and also the possibility of having conjugated poles that create large oscillations.

The closed loop poles can be calculated from $H_\psi(s)$ equating its denominator to zero:

$$\begin{aligned}
1 + C_{\mathrm{P}}(s)\, G_\psi(s) &= 0 \\
1 + (k_d s + k_p)\frac{m_v s - Y_v}{As^3 + Bs^2 + Cs} &= 0 \\
As^3 + Bs^2 + Cs + (k_d s + k_p)(m_v s - Y_v) &= 0 \\
As^3 + (B + m_v k_d)s^2 + (C + m_v k_p - Y_v k_d)s - Y_v k_p &= 0
\end{aligned} \tag{8.24}$$

For the controller design, it is better to locate the zero at the left of the two poles in order to obtain a fast response, and the gain should be a constant which leads the poles to the breakaway point at the left of the zero. Similarly to the P controller design, this point is achieved when two poles are the same.

Comparing the disposition of two equal poles and a real pole (equation 8.20) and the equation (8.24), we obtain the equation system

$$\begin{aligned}
2a + b &= (B + m_v k_d)/A \\
a^2 + 2ab &= (C + m_v k_p - Y_v k_d)/A \\
a^2 b &= -Y_v k_p / A
\end{aligned} \tag{8.25}$$

which has four unknowns, a, b, k_d, and k_p. As the equation system is under-determined, it is necessary to define one of these values. In this case we will set the zero, that is $-k_p/k_d$, at the left of the more negative pole through the formula

$$-\frac{k_p}{k_d} = 1.1\, p_1 \tag{8.26}$$

where p_1 is the more negative pole.

Solving this system for the values of the transfer functions (8.16), we obtain

$$
\begin{aligned}
\underset{0.3}{C_{\mathrm{PD}}}(s) &= 408.8266(s+0.8039)\\
\underset{0.9}{C_{\mathrm{PD}}}(s) &= 1012.4791(s+2.0165)\\
\underset{2.0}{C_{\mathrm{PD}}}(s) &= 2109.6664(s+4.2279)
\end{aligned}
\tag{8.27}
$$

Figure 8.12 shows the root locus of the linearized systems for 0.3, 0.9 and 2 m/s using the different zeros of the PD controllers. The green lines mark the root locus when the zero is placed at -0.8, and the green triangles show the particular case when controller $C_{\mathrm{PD}0.3}(s)$ is used. The blue lines mark the root locus when the zero is placed at -2.0, and the blue squares show the particular case when controller $C_{\mathrm{PD}0.9}(s)$ is used. Finally, the red lines mark the root locus when the zero is placed at -4.2, and the red stars show the particular case when controller $C_{\mathrm{PD}2.0}(s)$ is used.

For the first system $G_{\psi_{0.3}}(s)$, controller $C_{\mathrm{PD}0.3}(s)$ moves the real poles to a more negative zone. Controller $C_{\mathrm{PD}0.9}(s)$ moves the poles to a much more negative zone but the poles are complex, leading to overshoot and oscillations. Finally, controller $C_{\mathrm{PD}0.9}(s)$ moves the poles to $-3.9 \pm 3.8i$ (not seen in the figure), leading again to underdamped responses but placing the poles at a very negative values.

For the second system $G_{\psi_{0.9}}(s)$, controller $C_{\mathrm{PD}0.3}(s)$ is not very good because the dominant pole is near the imaginary axis. Controller $C_{\mathrm{PD}0.9}(s)$ is in a better situation because it moves the poles to a more negative zone while keeping them real. Finally, controller $C_{\mathrm{PD}2.0}(s)$ moves the poles to $-4.4 \pm 3.1i$ (not seen in the figure), which is much more to the left but with complex values.

For the third system $G_{\psi_{2.0}}(s)$, the two controllers $C_{\mathrm{PD}0.3}(s)$ and $C_{\mathrm{PD}0.9}(s)$ are not very good because their dominant poles are not moved far away from the imaginary axis. On the other hand, controller $C_{\mathrm{PD}2.0}(s)$ moves the real poles to very negative values.

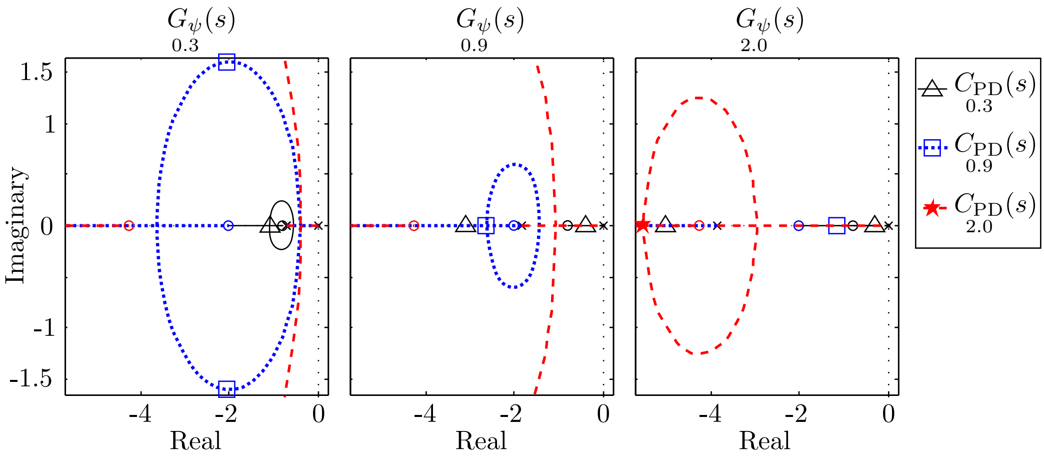


Figure 8.12: Root locus of the different $G_\psi(s)$ and pole displacement using the controllers $C_{\mathrm{PD}}(s)$

The above analysis shows that when a controller is used in the zone for which it was designed the poles are moved to negative real values. On the other hand, crossing the controllers, the poles can be moved to much more negative values but they become complex. This situation makes it difficult to see if the controllers are zonally differentiated.

Figure 8.13 helps to distinguish this. It shows the step responses of the different linearizations using the different controllers in a closed loop. At 0.3 m/s, controller $C_{\mathrm{PD}0.3}(s)$ does not yield overshoot. However, controller $C_{\mathrm{PD}2.0}(s)$ seems to have a better response although it has some overshoot. In the second linearization, at 0.9 m/s, the controller designed for this case, $C_{\mathrm{PD}0.9}(s)$, yields a better response than the others. However, controller $C_{\mathrm{PD}2.0}(s)$ leads to a very similar response with a small overshoot. Finally, the responses in the third linearization, for 2 m/s, clearly show that the best controller is $C_{\mathrm{PD}2.0}(s)$ since it has the lower settling time and has no overshoot. From all this, it is not clear that this set of controllers is zonally differentiated.

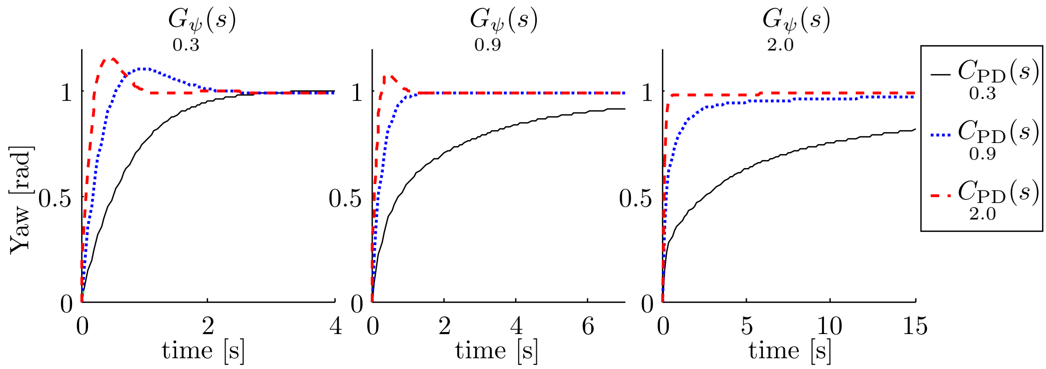


Figure 8.13: Step response of the different feedback systems using the controllers $C_{\mathrm{PD}}(s)$

Zonally differentiated controllers

The controllers designed above have shown that, as the forward velocities increases, the controller needs higher gain. On the other hand, when a controller designed for high velocities is used at low velocities the response is underdamped but faster. At the same time, the use of three velocities makes it difficult to see a good differentiation between the controllers.

To solve this, we have opted to use two velocities to design the controllers, 0.3 m/s and 2 m/s. The controllers chosen for each velocity are $C_{\mathrm{PD}0.3}(s)$ and $C_{\mathrm{P}2.0}(s)$ respectively. Figure 8.14 shows the root locus for each case. When the vehicle travels at 0.3 m/s the first controller, $C_{\mathrm{PD}0.3}(s)$, moves the two poles to -1 while the second, $C_{\mathrm{P}2.0}(s)$, to $-0.4 \pm 1.8i$. In this case, the benefits of the first controller are clear. On the other hand, when the vehicle travels at 2 m/s the first controller moves the dominant pole to -0.3 while the second controller moves the two poles to -2; in this case, the benefits of the second controller are clear.

Regarding the response in time, figure 8.15 shows the step response using these controllers. For the first linearization, controller $C_{\mathrm{PD}0.3}(s)$ yields a better performance than the second one, which has an underdamped and slow response. For the second linearization, the controller $C_{\mathrm{P}2.0}(s)$ yields a faster response than the first one.

In conclusion, we can say that this set of controllers is zonally differentiated, which is what we wanted to obtain. Figure 8.16 shows the step response but this time on the nonlinear model. The vehicle is forced to navigate to two velocities, 0.3 and 2 m/s, obtaining similar results as above.

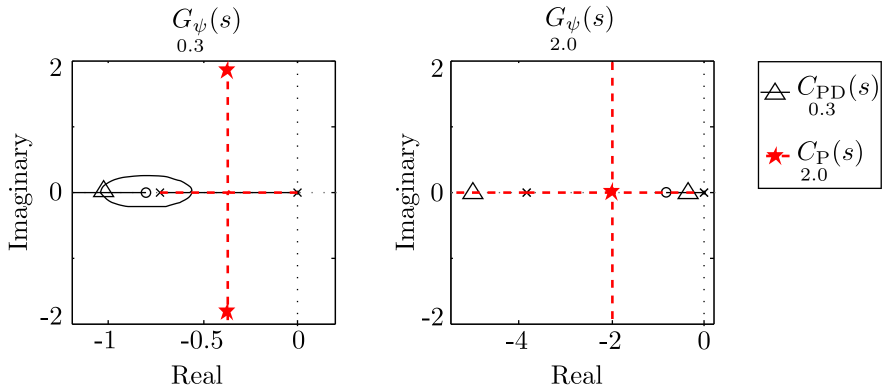


Figure 8.14: Root locus for $G_{\psi_{0.3}}(s)$ and $G_{\psi_{2.0}}(s)$, and pole displacement using the controllers $C_{\mathrm{PD}0.3}(s)$ and $C_{\mathrm{P}2.0}(s)$

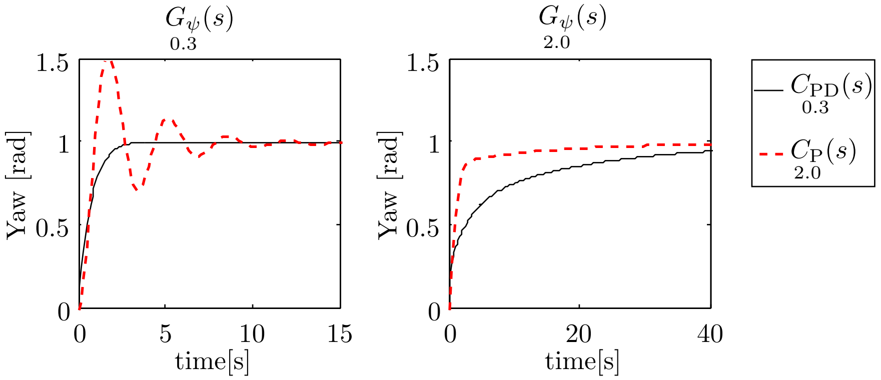


Figure 8.15: Step response of the systems $G_{\psi_{0.3}}(s)$ and $G_{\psi_{2.0}}(s)$ using the controllers $C_{\mathrm{PD}0.3}(s)$ and $C_{\mathrm{P}2.0}(s)$ in a feedback loop

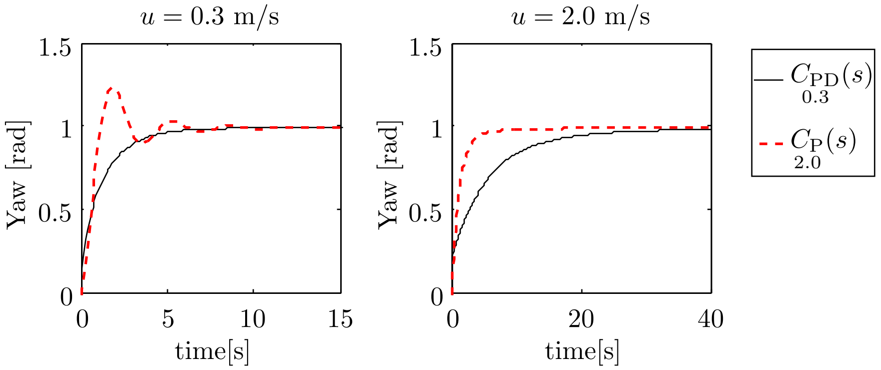


Figure 8.16: Step response of the nonlinear model using the controllers $C_{\mathrm{PD}0.3}(s)$ and $C_{\mathrm{P}2.0}(s)$ in a feedback loop, and forced to navigate at 0.3 m/s and 2 m/s

8.2.3 Transfer function of the forward velocity

In the chapter on modeling, we introduced the transfer function of the forward velocity with respect to the propulsion (see equation 7.63), which only has one pole. This can be expressed as

$$\underset{u_0}{G_u}(s) = \frac{u(s)}{\text{Prop}(s)} = \frac{1}{m_u s - 2|u_0|X_{|u|u} - X_u} \tag{8.28}$$

where the subindex u_0 in the notation $G_{u u_0}(s)$ represents the velocity at which the model is linearized. Equations (8.29) show three of them when the model is linearized at 0.3, 0.9 and 2 m/s.

$$\begin{aligned} \underset{0.3}{G_u}(s) &= \frac{0.001611}{s+0.148} \\ \underset{0.9}{G_u}(s) &= \frac{0.001611}{s+0.4429} \\ \underset{2.0}{G_u}(s) &= \frac{0.001611}{s+0.9836} \end{aligned} \tag{8.29}$$

8.2.4 Control of the forward velocity

The velocity control consists of a definition of a reference u_{ref}, which is compared with the actual velocity u in order to establish an error. The principal idea is to bring this error to zero using a controller $T(s)$ that actuates the thrusters. The general block diagram can be seen in figure 8.17.

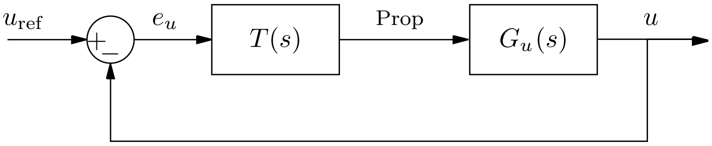


Figure 8.17: Block diagram of the closed loop system for the velocity control

Denoting by $H_u(s)$ the transfer function of the velocity with respect to the reference, it can be calculated as

$$H_u(s) = \frac{u(s)}{u_{\text{ref}}(s)} = \frac{T(s)\,G_u(s)}{1+T(s)\,G_u(s)}, \tag{8.30}$$

and the error function is

$$\frac{e_u(s)}{u_{\text{ref}}(s)} = \frac{1}{1+T(s)\,G_u(s)}. \tag{8.31}$$

The first condition to design the controller is to guarantee zero error in steady state for a step response. Using a proportional controller, and employing the final value theorem, we have

$$\begin{aligned} \lim_{t\to\infty} e_u(t) &= \lim_{s\to 0} s\,\frac{e_u(s)}{u_{\text{ref}}(s)}\,\frac{1}{s} \\ &= \lim_{s\to 0} \frac{1}{1+k\,G_u(s)} \\ &= \frac{2|u_0|X_{|u|u} + X_u}{2|u_0|X_{|u|u} + X_u - k} \end{aligned}$$

As the result is not zero, it is necessary to use another control. The proportional-integral controller (PI) is a good option for this case because it adds an integrator. Applying the same calculation we have

$$\begin{aligned}\lim_{t\to\infty} e_u(t) &= \lim_{s\to 0} s\,\frac{e_u(s)}{u_{\text{ref}}(s)}\,\frac{1}{s}\\ &= \lim_{s\to 0}\frac{1}{1+\frac{k_p s+k_i}{s}\,G_u(s)}\\ &= \lim_{s\to 0}\frac{s(m_u s-2|u_0|X_{|u|u}-X_u)}{m_u s^2+(k_p-2|u_0|X_{|u|u}-X_u)s+k_i}\\ &= 0\end{aligned}$$

which proves that the error is zero in steady state.

Taking this into account, we set this controller as follows,

$$\underset{u_0}{T_{\text{PI}}}(s) = \frac{k_p s + k_i}{s} \tag{8.32}$$

where the subscript PI denotes "proportional-integral" and u_0 is the velocity at which the model as been linearized.

Notice that the use of a PI controller implies a root locus with two poles and one zero, which is similar to the disposition obtained in the yaw control using a PD controller. Hence, the same design process can be used for this controller. Figure 8.18 shows the root locus of two controllers calculated for two transfer functions at 0.3 m/s and 2 m/s. The main idea is that they be zonally differentiated, in order to have the best pole disposition using the correct controller at the right conditions of velocity.

For the first transfer function, $G_{u0.3}(s)$, the poles are moved to -0.35 using the controller $T_{\text{PI}0.3}(s)$, while they are moved to $-0.34\pm 0.68i$ using the controller $T_{\text{PI}2.0}(s)$. The real poles represent a better option compared with the complex conjugated poles.

On the other hand, for the second transfer function, $G_{u2.0}(s)$, the dominant pole is moved to -0.08 using the controller $T_{\text{PI}0.3}(s)$, which is very near to the imaginary axis, while the poles are moved to -1.4 using the controller $T_{\text{PI}2.0}(s)$, which guarantees fast response. Notice that this design moves the poles at the right breakaway point and not to the left; this is because the left breakaway point needs a large gain which compromises the zonal differentiation (a large gain implies better performance at the first transfer function $G_{u0.3}(s)$).

The designed controllers are

$$\begin{aligned}\underset{0.3}{T_{\text{PI}}}(s) &= \frac{337.6657(s+0.2200)}{s},\\ \underset{2.0}{T_{\text{PI}}}(s) &= \frac{327.7492(s+1.0820)}{s}.\end{aligned} \tag{8.33}$$

Regarding the response in time, figure 8.19 shows the step response for 0.3 m/s and 2 m/s using these controllers. For the first linearization, controller $T_{\text{PI}0.3}(s)$ yields a small overshoot but better performance than the second one, which has a more underdamped response. On the other hand, for the second linearization, controller $T_{\text{PI}2.0}(s)$ yields a faster response than the first one. Similar results are obtained when the nonlinear model is used, as can be seen in figure 8.20.

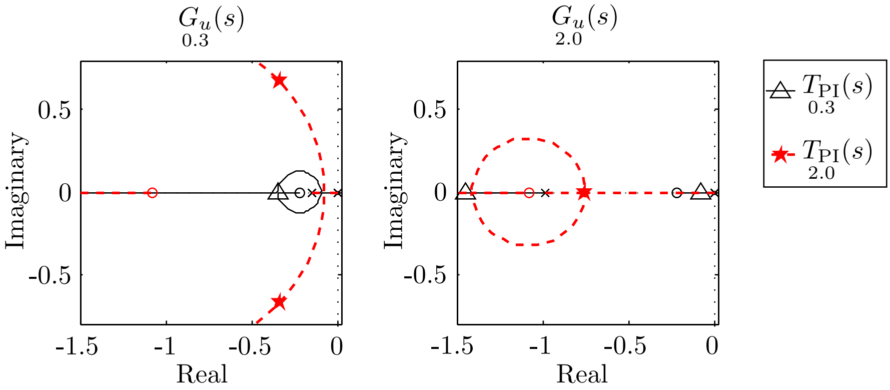


Figure 8.18: Root locus for $G_{u0.3}(s)$ and $G_{u2.0}(s)$, and pole displacement using the controllers $T_{\mathrm{PI}0.3}(s)$ and $T_{\mathrm{PI}2.0}(s)$

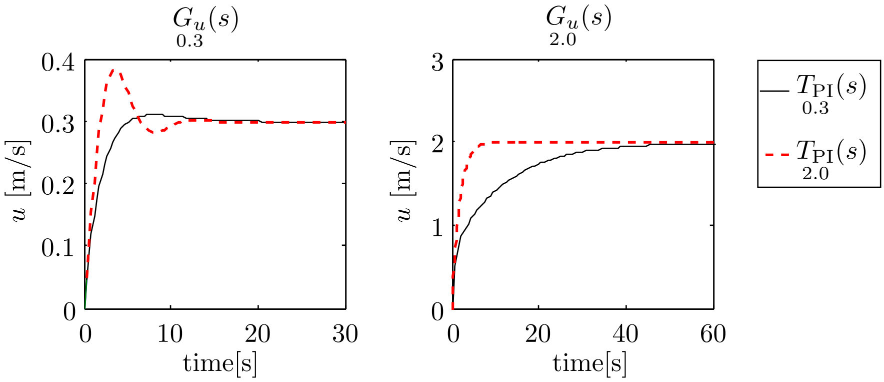


Figure 8.19: Step response of the systems $G_{u0.3}(s)$ and $G_{u2.0}(s)$ using the controllers $T_{\mathrm{PI}0.3}(s)$ and $T_{\mathrm{PI}2.0}(s)$ in a feedback loop

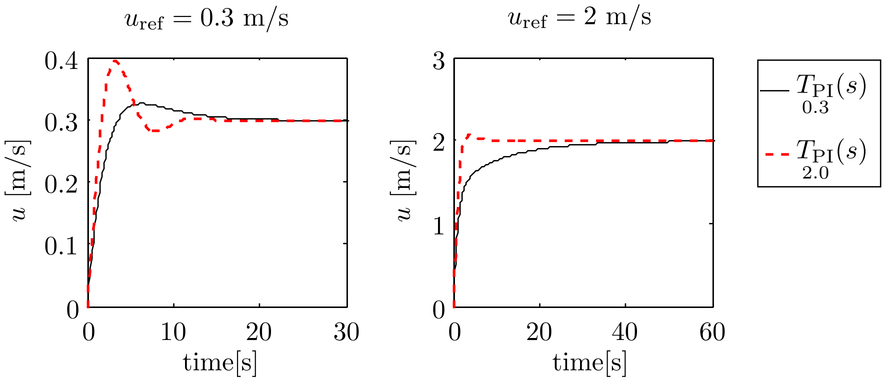


Figure 8.20: Step response of the nonlinear model using the controllers $T_{\mathrm{PI}0.3}(s)$ and $T_{\mathrm{PI}2.0}(s)$ in a feedback loop, and using 0.3 m/s and 2 m/s as input references

8.3 Inner loop: Fuzzy control (first approach)

Alternatively to classical control engineering, it seems convenient to simulate the behaviour of a person who is able to control the given process. We call this development of a model of a human 'control expert' *knowledge-based analysis.* To make such an analysis the expert may be questioned directly. The expert then specifies his knowledge in form of *linguistic rules.* Instead of directly interviewing the expert it is also possible to observe his behaviour and extract from the observation protocol the necessary information. The results of this procedure can be used to provide appropriate (linguistic) rules that control the process [69] [32].

The fuzzy control is a tool used for these purposes. As we introduced in the review of this chapter the type-1 TSK fuzzy controller uses crisp functions in the output instead of linguistic terms, which is very effective and reduces the computational calculation. Its importance stems from its ability to be adapted to the development of linear controllers, like the controllers P, PD and PI of section 8.2.

In the case of the GUANAY II this control can be used to control the velocity and the yaw. The above results have shown the importance of using different controllers depending of the forward velocity u. The use of this control expert can be used to change between the different controllers, in other words, fuzzy control is presented as an interpolating controller. A block diagram of this concept is presented in figure 8.21.

It has two controllers. The first one, called $C^{(u)}_{\text{Fuzzy1}}$, takes the velocity error e_u and uses the linear controller $T(s)$ to control the velocity. The parameters of $T(s)$ are dynamically modified by a fuzzy block which has u_{ref} as input. Notice that we could use u as input, but the reference u_{ref} is similar to the step response with the simulations made using the lineal controllers, while u changes over time causing different responses.

The second controller, called $C^{(\psi)}_{\text{Fuzzy1}}$, takes the yaw error e_ψ and uses the linear controller $C(s)$ to control the yaw. The parameters of $C(s)$ are dynamically modified by a fuzzy block which has the forward velocity u as input. This input u is the basis of the designed controllers in the previous section.

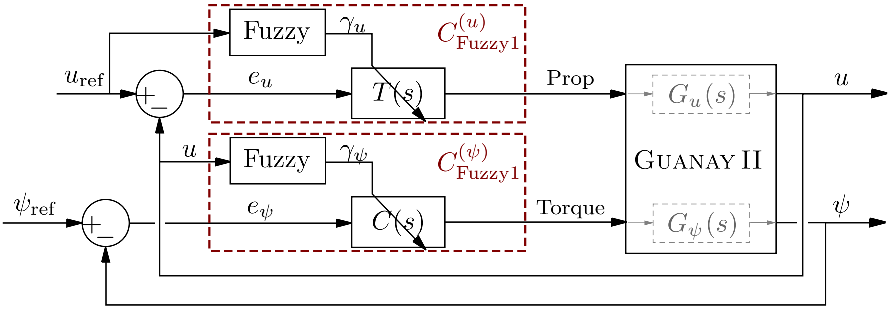


Figure 8.21: Fuzzy control — velocity and yaw control regarding the forward velocity u

The development of controllers $C^{(u)}_{\text{Fuzzy1}}$ and $C^{(\psi)}_{\text{Fuzzy1}}$ is detailed below.

8.3.1 Fuzzification

The velocities u and u_{ref} need to be transformed to a linguistic terms in order to be controlled in a fuzzy way. Generally, these terms have 5 o 7 connotations (for instance, large negative, small negative, zero, small positive, and large positive), but the linear controllers designed in the last section have shown that the GUANAY II only needs two: 0.3 m/s and 2 m/s, considered as 'low' and 'high' velocity. The guarantee of zone differentiation prevents the use of more connotations.

There are many ways to define the fuzzy sets. Fuzzy sets of 'low' and 'high' are presented in equations (8.34) and (8.35) respectively. They are segmented in piecewise linear functions. Likewise, figure 8.22 shows a graphical representation of these membership functions.

$$\mu_l(u) = \begin{cases} 1 & \text{if} \quad u < 0.3 \\ \frac{10(2-u)}{17} & \text{if} \quad 0.3 \leq u \leq 2 \\ 0 & \text{if} \quad u > 2 \end{cases} \tag{8.34}$$

$$\mu_h(u) = \begin{cases} 0 & \text{if} \quad u < 0.3 \\ \frac{10u-3}{17} & \text{if} \quad 0.3 \leq u \leq 2 \\ 1 & \text{if} \quad u > 2 \end{cases} \tag{8.35}$$

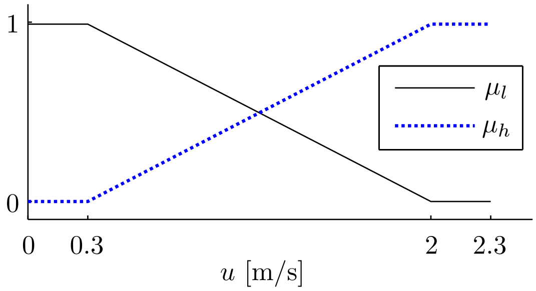


Figure 8.22: Fuzzy set. Membership functions μ_l and μ_h

8.3.2 Inference

The expert specifies his knowledge in form of linguistic rules. The advantage of this conception is the ease to define the rules and outputs. For instance, in the case of the GUANAY II , we know that some controllers have to be used at low velocities, $C_{\text{PD}0.3}(s)$ and $T_{\text{PI}0.3}(s)$ (see equations 8.27 and 8.33), and other controllers at high velocities, $C_{\text{P}2.0}(s)$ and $T_{\text{PI}2.0}(s)$ (see equations 8.22 and 8.33). The linguistic rules of $C^{(\psi)}_{\text{Fuzzy1}}$ to control the yaw are:

$$\begin{array}{ll} R_1: & \textbf{if } u \textbf{ is } \mu_l \textbf{ then } C(s) \textbf{ is } C_{\text{PD}0.3}(s) \\ R_2: & \textbf{if } u \textbf{ is } \mu_h \textbf{ then } C(s) \textbf{ is } \ C_{\text{P}2.0}(s) \end{array}$$

while the linguistic rules of $C^{(u)}_{\text{Fuzzy1}}$ to control the velocity are given by

$$\begin{array}{ll} R_1: & \textbf{if } u_{\text{ref}} \textbf{ is } \mu_l \textbf{ then } T(s) \textbf{ is } T_{\text{PI}0.3}(s) \\ R_2: & \textbf{if } u_{\text{ref}} \textbf{ is } \mu_h \textbf{ then } T(s) \textbf{ is } T_{\text{PI}2.0}(s) \end{array}$$

From equation (8.5) we have seen the fuzzy operator "min" as a logical conjunction between the membership functions of two or more input variables. In the case of GUANAY II , as we only need one variable, this value is simplified to the membership function of the rule. That is, to control the yaw we have

$$\begin{aligned} \alpha_{\psi 1} &= \mu_l(u) \\ \alpha_{\psi 2} &= \mu_h(u) \end{aligned} \tag{8.36}$$

and to control the velocity we have

$$\begin{aligned}\alpha_{u1} &= \mu_l(u_{\text{ref}}) \\ \alpha_{u2} &= \mu_h(u_{\text{ref}})\end{aligned} \tag{8.37}$$

On the other hand, from equation (8.12) we obtain the output of the controller using the different α_m as weights. Both the control of the yaw and the control of the velocity is determined by a vector of three parameters: k_p, k_i and k_d (see table 8.1). The analytic functions are

$$\begin{aligned}f_{\psi m} &= \begin{bmatrix} k_{p_m}^{(\psi)} & k_{i_m}^{(\psi)} & k_{d_m}^{(\psi)} \end{bmatrix} \\ f_{um} &= \begin{bmatrix} k_{p_m}^{(u)} & k_{i_m}^{(u)} & k_{d_m}^{(u)} \end{bmatrix}\end{aligned} \tag{8.38}$$

and the crisp functions to control the yaw and velocity are

$$\begin{aligned}\gamma_\psi &= \frac{\alpha_{\psi 1} f_{\psi 1} + \alpha_{\psi 2} f_{\psi 2}}{\alpha_{\psi 1} + \alpha_{\psi 2}} \\ \gamma_u &= \frac{\alpha_{u1} f_{u1} + \alpha_{u2} f_{u2}}{\alpha_{u1} + \alpha_{u2}}\end{aligned} \tag{8.39}$$

Table 8.1: Fuzzy control 1 — Outputs of the different rules

Rule	$C_{\text{Fuzzy1}}^{(\psi)} : C(s)$			$C_{\text{Fuzzy1}}^{(u)} : T(s)$		
	$k_p^{(\psi)}$	$k_i^{(\psi)}$	$k_d^{(\psi)}$	$k_p^{(u)}$	$k_i^{(u)}$	$k_d^{(u)}$
R_1	328.7	0	408.8	337.7	74.3	0
R_2	1043.9	0	0	327.7	354.6	0

8.3.3 Simulations

Several simulations with the nonlinear model were made in order to test the performance of the fuzzy controllers. Figure 8.23 shows the step response for the yaw using the different controllers and navigating at different velocities. Likewise, table 8.2 presents the overshoot and settling time (tolerance of 5 %) for each case.

When the vehicle is navigating at 0.3 m/s the controller $C_{\text{Fuzzy1}}^{(\psi)}$ performs as well as the response of the controller $C_{\text{PD}0.3}(s)$ which has a good settling time (3.8 s) and no overshoot. However, the controller $C_{\text{P}2.0}(s)$ yields an overshoot of 26.1 %. There is a similar characteristic when the vehicle goes at 2 m/s; the fuzzy controller $C_{\text{Fuzzy1}}^{(\psi)}$ yields a similar response to that of the controller $C_{\text{P}2.0}(s)$ which has a good settling time (4.2 s) and no overshoot. However, the controller $C_{\text{PD}0.3}(s)$ has an undesired settling time, 19.4 s. Finally, when the vehicle is navigating at 1 m/s, the fuzzy controller yields an intermediate response between the linear controllers.

On the other hand, figure 8.24 shows the step response for the forward velocity using the different controllers and references. Likewise, table 8.3 presents the overshoot and settling time (tolerance of 5 %) for each case.

When the reference is 0.3 m/s, the controller $C_{\text{Fuzzy1}}^{(u)}$ equals the response of the controller $T_{\text{PI}0.3}(s)$, which has a settling time of 10.6 s and overshoot of 8.9 %. However, the controller $T_{\text{PI}2.0}(s)$ yields a high overshoot, 31.7 %. There is a similar behaviour when the reference

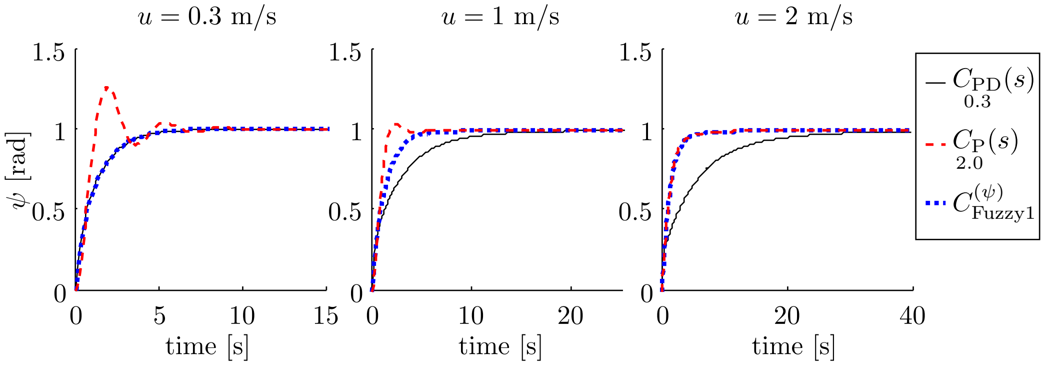


Figure 8.23: Comparative of the step response for the yaw using the fuzzy controller and linear controllers

Table 8.2: Comparative of the overshoot and settling time for the yaw using the fuzzy controller and linear controllers

Controller	Overshoot [%]			Settling time [s]		
	0.3 $\frac{\text{m}}{\text{s}}$	1 $\frac{\text{m}}{\text{s}}$	2 $\frac{\text{m}}{\text{s}}$	0.3 $\frac{\text{m}}{\text{s}}$	1 $\frac{\text{m}}{\text{s}}$	2 $\frac{\text{m}}{\text{s}}$
$C_{\text{PD0.3}}(s)$	—	—	—	3.8	9.4	19.4
$C_{\text{P2.0}}(s)$	26.1	3.9	—	4.2	1.6	4.2
$C^{(\psi)}_{\text{Fuzzy1}}$	—	—	—	3.8	4.0	4.2

is 2 m/s; the fuzzy controller $C^{(u)}_{\text{Fuzzy1}}$ performs as well as the controller $T_{\text{PI2.0}}(s)$, which has a settling time of 2.1 s and overshoot of 3.4 %. However, the controller $T_{\text{PI0.3}}(s)$ has an undesired settling time, 19.8 s. Finally, when the reference is 1 m/s the fuzzy controller yields an intermediate response between the linear controllers.

In conclusion, these results show the advantage of using the type-1 TSK fuzzy controller,

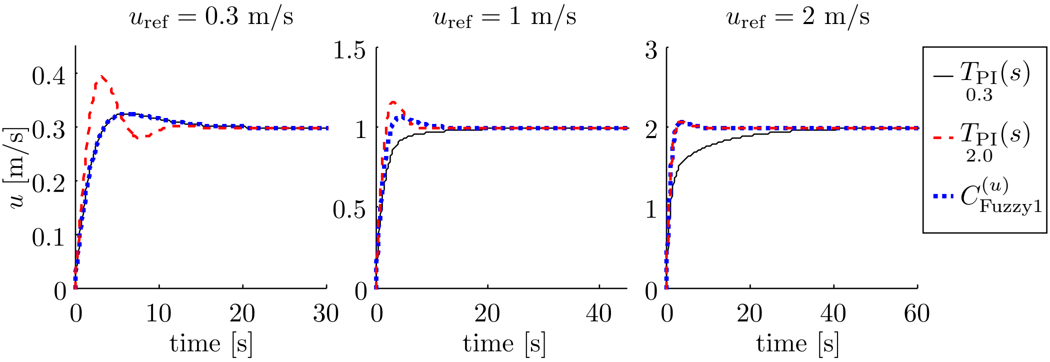


Figure 8.24: Comparative of the step response for the forward velocity using the fuzzy controller and linear controllers

Table 8.3: Comparative of the overshoot and settling time for the forward velocity using the fuzzy controller and linear controllers

Controller	Overshoot [%]			Settling time [s]		
	$0.3\ \frac{\text{m}}{\text{s}}$	$1\ \frac{\text{m}}{\text{s}}$	$2\ \frac{\text{m}}{\text{s}}$	$0.3\ \frac{\text{m}}{\text{s}}$	$1\ \frac{\text{m}}{\text{s}}$	$2\ \frac{\text{m}}{\text{s}}$
$T_{\text{PI}0.3}(s)$	8.9	—	—	10.6	6.4	19.8
$T_{\text{PI}2.0}(s)$	31.7	16.5	3.4	8.8	5.8	2.1
$C^{(u)}_{\text{Fuzzy1}}$	8.9	6.7	3.4	10.6	6.3	2.1

because it adapts the controller to the correct zone depending of the forward velocity, while the linear controllers only have good responses in an specific zone.

8.4 Inner loop: Gain scheduled control

This control problem can be solved using a gain scheduled controller. This controller is very similar to fuzzy control, in the sense that parameters of linear controllers are changed by a specific variable, in this case the velocity. Figure 8.25 shows a block diagram for the control of the GUANAY II.

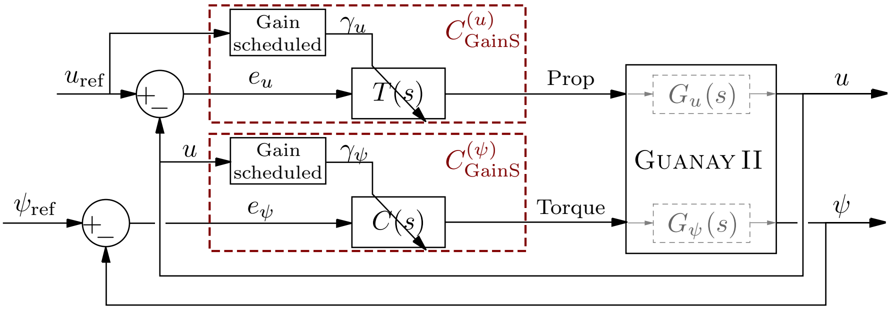


Figure 8.25: Gain scheduling control — velocity and yaw control regarding the forward velocity u

Similar to the fuzzy controller, we will vary $C(s)$ taking into account that it is $C_{\text{PD}0.3}(s)$ at low velocities (see equation 8.27), and $C_{\text{P}2.0}(s)$ at high velocities (see equation 8.22). This controller is called $C^{(\psi)}_{\text{GainS}}$. And, likewise, $T(s)$ will vary taking into account that it is $T_{\text{PI}0.3}(s)$ at the small velocity reference, and $T_{\text{PI}2.0}(s)$ at the large velocity reference (see equations 8.33). This controller is called $C^{(u)}_{\text{GainS}}$. Equations (8.40) show a polynomial regression of the changes in the parameters $k_p^{(\psi)}$, $k_d^{(\psi)}$, $k_p^{(u)}$, and $k_i^{(u)}$. Finally, figure 8.26 shows a graphical representation of these functions. The block $C^{(u)}_{\text{GainS}}$ is used to control the forward velocity, and $C^{(\psi)}_{\mathrm{GainS}}$ is used to control the yaw.

$$\begin{aligned}
k_p^{(\psi)}(u) &= -106.7873u^3 + 368.4161u^2 + 42.7484u + 311.9691 \\
k_d^{(\psi)}(u) &= 60.8593u^3 - 209.9647u^2 - 24.3628u + 417.1362 \\
k_p^{(u)}(u_{\mathrm{ref}}) &= 1.4917u_{\mathrm{ref}}^3 - 5.1462u_{\mathrm{ref}}^2 - 0.5971u_{\mathrm{ref}} + 337.9239 \\
k_i^{(u)}(u_{\mathrm{ref}}) &= -41.8110u_{\mathrm{ref}}^3 + 144.2478u_{\mathrm{ref}}^2 + 16.7375u_{\mathrm{ref}} + 68.0233
\end{aligned} \tag{8.40}$$

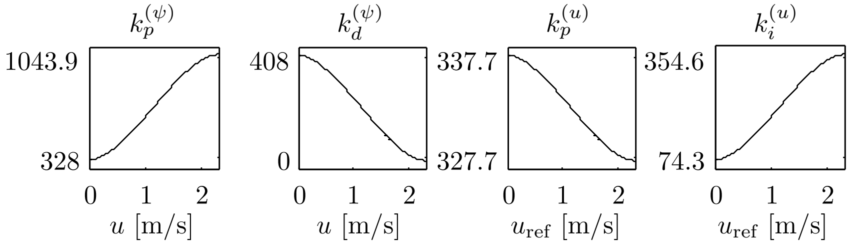


Figure 8.26: Gain scheduled parameters to control the yaw and forward velocity

8.4.1 Simulations

Several simulations with the nonlinear model were made in order to test the performance of the gain scheduled controller. Figures 8.27 and 8.28 show the performance of the yaw control and forward velocity control respectively. They compare several step responses using the gain schedule controller and each linear controller. In all of them, the advantage of using the gain schedule controller is apparent.

Notice that these simulations are very similar to the comparative made using the fuzzy controller in section 8.3.3, and therefore the analysis of the results is the same. In this way the gain scheduled controller validates the TSK fuzzy controller as an interpolator of different linear controllers.

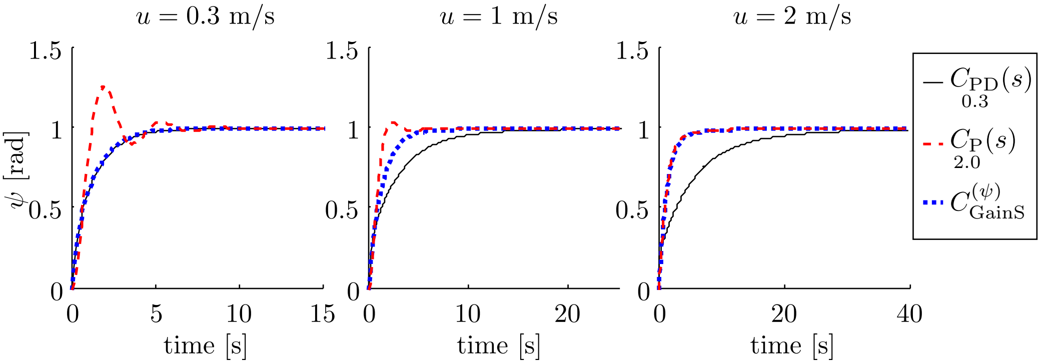


Figure 8.27: Comparative of the step response for the yaw using the gain schedule controller and the linear controllers

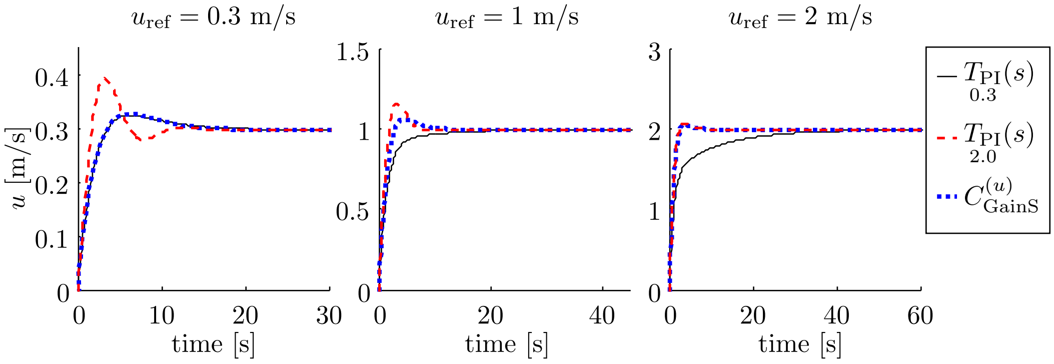


Figure 8.28: Comparative of the step response for the forward velocity using the gain schedule controller and linear controllers

8.5 Problems with constraints

In the last simulations we saw the importance of using a fuzzy controller. However, the propulsion of the thrusters was not detailed. This section deals with the thrusters and the associated physical constraints when the yaw is controlled. For a detailed analysis of the physical constraints see section 7.6.

Figure 8.29 shows different simulations comparing the torques used to follow a yaw reference. All simulations consider that the vehicle is navigating at 1 m/s. In subfigure (d) we can see the step response using different controllers, including the fuzzy controller $C^{(\psi)}_{\text{Fuzzy1}}$, and one can realize that this is the same response as above (figure 8.23). However, the torque used to accomplish this turn is very large, reaching undesired and impossible values of 3000 Nm (see subfigure (a)). This is because the simulations in the last section were made without considering any limitations for the thrusters.

When thrusters are saturated according to their physical limitations, relating to subfigure (b), during the first 20 s the maximum possible torque is applied, and thereafter the different controllers decrease their actions. In this case, the yaw response, see subfigure (e), suffers an important change respect to the first simulation: The saturation avoids the fast response and there is no significant difference between the controllers in terms of overshoot and settling time. We can conclude that, since the GUANAY II AUV has low power to turn (remember that the maximum torque of 28 Nm gives an angular velocity 0.2 rad/s in steady state, and lower values in transient state), the controllers can not have large gains, and therefore, it is difficult to establish zonally differentiated controllers.

Finally, we have seen on field tests that the vehicle suffers important disturbances when it senses the attitude. We have modeled these disturbances and we found a white noise of about ±5.84 including small waves (high frequency). Subfigures (c) and (f) show a simulation with this condition. The problem with the controllers becomes evident in subfigure (c), where the thrusters are saturated in two ways. This evidences the high gain of the controllers. Moreover, subfigure (f) shows the yaw response, which is compromised when using $C^{(\psi)}_{\text{Fuzzy1}}$ and $C_{\text{PD}0.3}(s)$ controllers, due to the derivative term.

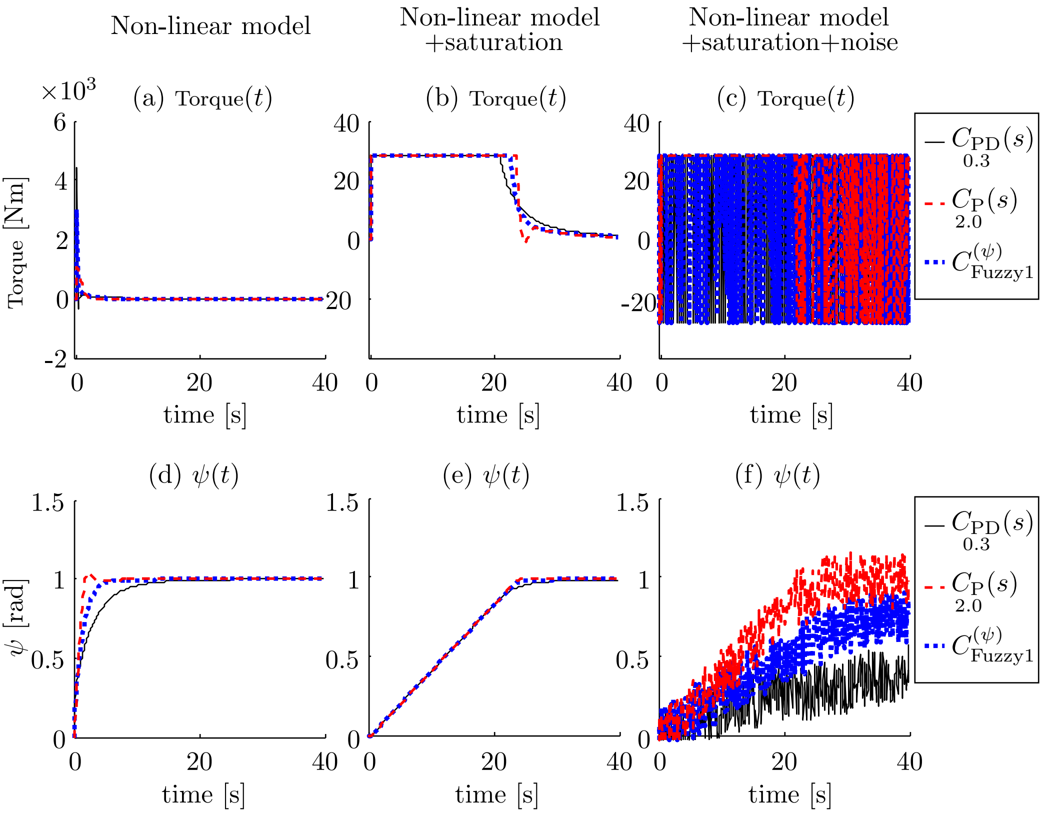


Figure 8.29: Torque comparative of the step response for the yaw using different controllers. The vehicle is navigating at 1 m/s. (a), (d) torque without constraints and yaw response. (b), (e) torque with physical constraints and yaw response. (c), (f) torque with physical constraints and yaw with noise.

8.6 Inner loop: Fuzzy control (second approach)

We saw in the last section two principal problems with the yaw control. The first one is the slow response due to physical constraints of the thrusters. This problem evidences that the controller design using different linearizations at different velocities, which was the approach of the previous sections, is not useful. The reason is that the vehicle has low maneuverability when it is navigating at high velocities, as was seen in section 7.6.2, and therefore, regardless of the selected controller, the response is very similar at high velocities.

The second problem is the noise present in the measure. As the controller has a derivative action, the noise is amplified and the torque saturates in two ways. A fast switching is produced, where the maximum torque is used in an instance (28 Nm), and the minimum torque in the next, causing damages to the thrusters. This problem is also due to small waves (high frequency) which are also amplified by the derivative action.

Knowing these problems, we propose in this section another approach to control the yaw, C_{Fuzzy2}. It consists of using the yaw error as input of the fuzzy controller, which will adjust the parameters of the linear controller. The controller does not worry about the forward velocity. The idea is to use a big gain when the error is big, and a small gain when the error is small. In this order, the noise is minimized when the error is small. Figure 8.30 shows the block diagram of this control.

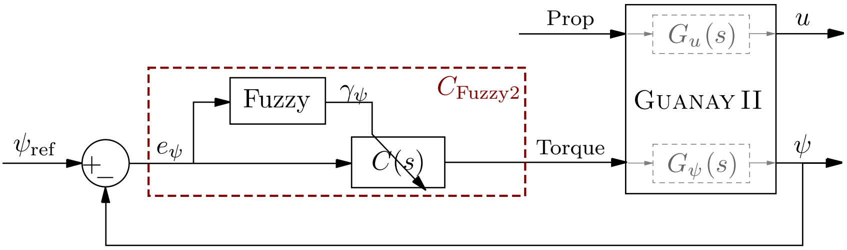


Figure 8.30: Fuzzy control regarding the yaw error

8.6.1 Calculation of gains

We know that the torque must be between ±28 Nm. Using a proportional controller, the torque is calculated as

$$\text{Torque} = e_\psi \, k_p. \tag{8.41}$$

This means that if the proportional gain k_p is 328.7, the torque saturates at $28/328.7 = 0.085$ rad (or 4.9°) of error, which is one of the problems seen above. By contrast, the idea of this controller is to saturate the torque action only with large yaw errors, and to avoid fast switching with small errors. For this, we will define some limits in the torque action, and they will be compared with the noise of the yaw measure, which is about ±0.102 rad (or ±5.84°).

Assume that the yaw error e_ψ is smaller than 2°. In order to avoid excessive switching, we assume that the noise amplified in the torque cannot exceed 10% of the maximum torque, that is 2.8 Nm. Then, the proportional controller is

$$C(s) = k_{p1} = \frac{\text{Torque}}{e_\psi} = \frac{2.8}{0.102} = 27.45. \tag{8.42}$$

Next, assume that the yaw error e_ψ is about 10°. In this case we assume that the noise amplified in the torque cannot exceed 15% of the maximum torque, that is 4.2 Nm. Then

$$C(s) = k_{p2} = \frac{\text{Torque}}{e_\psi} = \frac{4.2}{0.102} = 41.18. \tag{8.43}$$

Finally, assume that the yaw error e_ψ is about 20°. In this case we assume that the controller action must be larger, but not enough to amplify the noise more than the 60% of the maximum torque, that is 11.2 Nm. Then

$$C(s) = k_{p3} = \frac{\text{Torque}}{e_\psi} = \frac{16.8}{0.102} = 164.7. \tag{8.44}$$

8.6.2 Fuzzification

We define 'low', 'medium' and 'high' error as linguistic terms to control the yaw in a fuzzy way. They are linked with 2°, 10° and 20° respectively. We have chosen triangular memberships because we have seen that a constant change in the parameters of the linear controller is enough to achieve good results. The fuzzy set is as follows:

$$\mu_l(e_\psi) = \begin{cases} 1 & \text{if} \quad e_\psi < 2^\circ \\ \frac{10-e_\psi}{8} & \text{if} \quad 2^\circ \le e_\psi \le 10^\circ \\ 0 & \text{if} \quad e_\psi > 10^\circ \end{cases} \tag{8.45}$$

$$\mu_m(e_\psi) = \begin{cases} 0 & \text{if} \quad e_\psi < 2^\circ \\ \frac{e_\psi-2}{8} & \text{if} \quad 2^\circ \le e_\psi \le 10^\circ \\ \frac{20-e_\psi}{10} & \text{if} \quad 10^\circ < e_\psi \le 20^\circ \\ 0 & \text{if} \quad e_\psi > 20^\circ \end{cases} \tag{8.46}$$

$$\mu_h(e_\psi) = \begin{cases} 0 & \text{if} \quad e_\psi < 10^\circ \\ \frac{e_\psi-10}{10} & \text{if} \quad 10^\circ \le e_\psi \le 20^\circ \\ 1 & \text{if} \quad e_\psi > 20^\circ \end{cases} \tag{8.47}$$

Likewise, figure 8.31 shows a graphical representation of these membership functions.

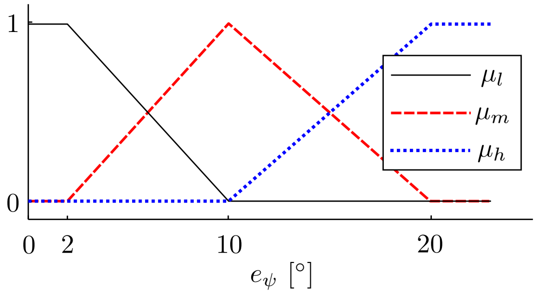


Figure 8.31: Yaw error fuzzy set. Membership functions μ_l, μ_h and μ_h

8.6.3 Inference

The linguistic rules to control the yaw using this error as input are

$$\begin{array}{ll} R_1: & \textbf{if } |e_\psi| \textbf{ is } \mu_l \textbf{ then } C(s) \textbf{ is } k_{p1} \\ R_2: & \textbf{if } |e_\psi| \textbf{ is } \mu_m \textbf{ then } C(s) \textbf{ is } k_{p2} \\ R_3: & \textbf{if } |e_\psi| \textbf{ is } \mu_h \textbf{ then } C(s) \textbf{ is } k_{p3} \end{array}$$

Similar to the controllers in previous sections, we use type-1 TSK controller to the inference of rules. From equation (8.5) we regard the fuzzy operator “min” as a logical conjunction between the membership functions of two or more input variables. In the case of GUANAY II, as we only need one variable, this value is simplified to the membership function of the rule. That is, to control the yaw we have

$$\begin{aligned} \alpha_{\psi 1} &= \mu_l(e_\psi) \\ \alpha_{\psi 2} &= \mu_m(e_\psi) \\ \alpha_{\psi 3} &= \mu_h(e_\psi) \end{aligned} \tag{8.48}$$

On the other hand, from equation (8.12) we obtain the output of the controller, which has the α_m as weights. Now the crisp function is

$$\gamma_\psi = \frac{\alpha_{\psi 1} k_{p1} + \alpha_{\psi 2} k_{p2} + \alpha_{\psi 3} k_{p3}}{\alpha_{\psi 1} + \alpha_{\psi 2} + \alpha_{\psi 3}}. \tag{8.49}$$

8.6.4 Simutations

We made several simulations in order to see the performance of the different controllers. The simulations include the constraints on thrusters and a white noise of ±5.84° in the compass.

Figure 8.32 shows the step response of the yaw when k_{p1} is used. The idea is to turn 1 rad, and to examine the settling time and the performance of the torque applied. For the calculation of the settling time we apply a low pass filter in order to remove the noise.

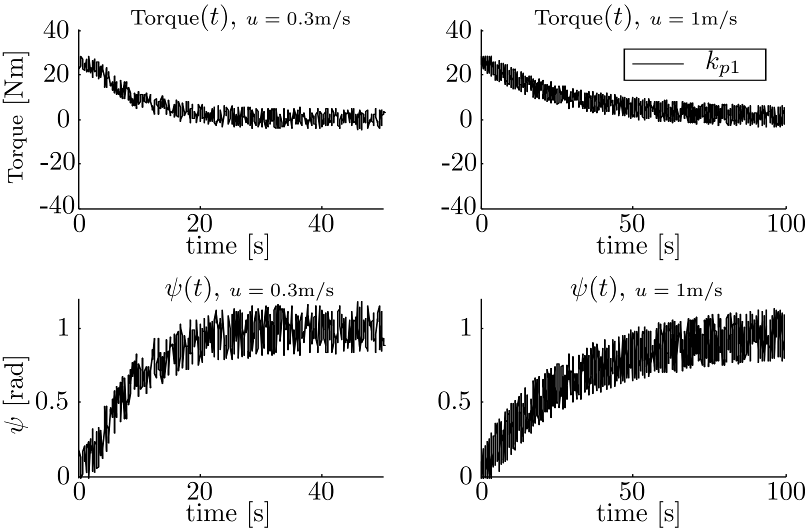


Figure 8.32: Step response of the yaw and torque using k_{p1}

When the vehicle travels at 0.3 m/s, the settling time is about 29.9 s, and the standard deviation of the torque at steady state is about 2.9 Nm, which is small. On the other hand, when the vehicle travels at 1 m/s the response takes more time: the settling time is about 98 s, while the noise in the torque is similar to the previous case, 3.0 Nm.

Figure 8.33 shows the step response when k_{p2} is used. In this case, we see improvements in the settling time, that is 19.3 s at 0.3 m/s and 65.3 s at 1 m/s, but as a counterpart, the standard deviation in the torque increases, 4.3 Nm at 0.3 m/s and 4.1 Nm at 1 m/s. Notice also that the torque saturates at the beginning for a short period.

The next simulation, displayed in figure 8.34, shows the performance when k_{p3} is used, which is the largest gain. In this case we found a better response for the yaw, because the settling time is about 11.1 s at 0.3 m/s and 30.3 s at 1 m/s. However, as a drawback, we see that the torque is almost saturated both ways, and the standard deviation is very high, about 16.4 Nm at 0.3 m/s and 14.5 Nm at 1 m/s.

Finally, figure 8.35 shows the performance using the fuzzy controller C_{Fuzzy2}. As mentioned, it is a combination of k_{p1}, k_{p2} and k_{p3}, and takes the best of each one. When the error is high, the controller acts like k_{p3} and the torque saturates for a long time, but as the

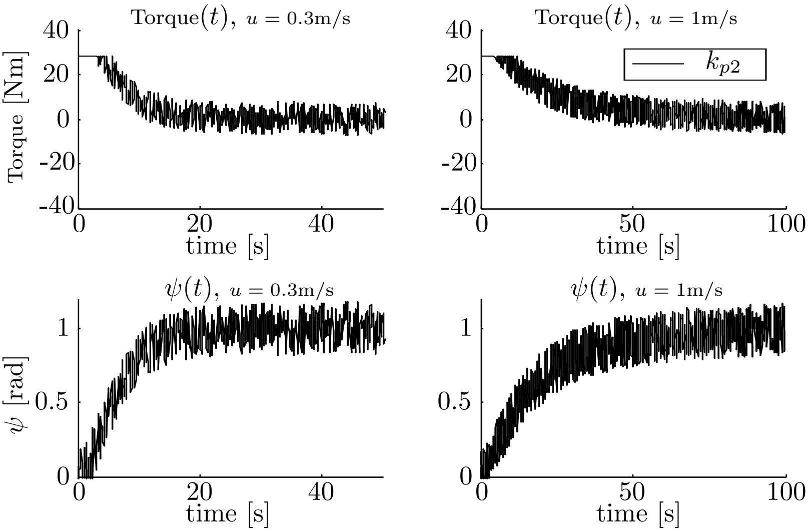


Figure 8.33: Step response of the yaw and torque using k_{p2}

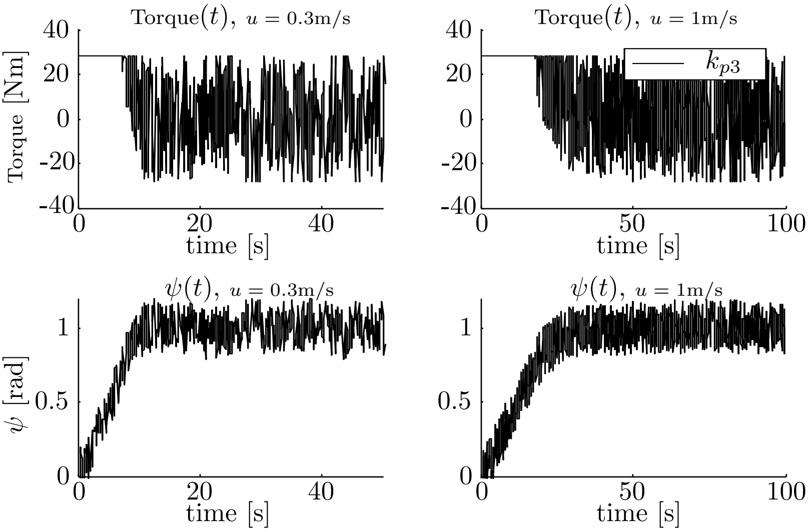


Figure 8.34: Step response of the yaw and torque using k_{p3}

error decreases the controller changes its gain until k_{p1}. In this way, the noise in the torque is small, like with k_{p1}. The simulation shows settling times of 12.2 s at 0.3 m/s and 47.5 s at 1 m/s. The standard deviation in the torque is about 3.6 Nm at 0.3 m/s and 4.9 Nm at 1 m/s.

In conclusion, a large gain is good to achieve a specific angle, but by contrast a low gain helps to reduce the noise in the torque. In this way, the fuzzy controller is presented as a interpolator between them regarding the yaw error. Table 8.4 resumes the performance of this controller compared with the other proportional controllers. Finally, notice that their standard deviation almost coincides with the design values of the controllers.

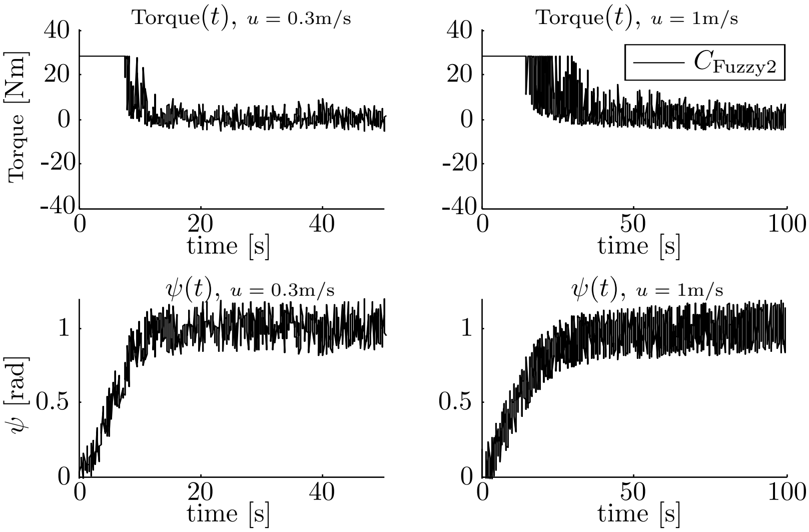


Figure 8.35: Step response of the yaw and torque using Fuzzy controller

Table 8.4: Comparative of the settling time and noise in the torque using the fuzzy controller and proportional controllers

Controller	Settling time [s]		Stdev in torque [Nm]	
	$0.3\ \frac{\text{m}}{\text{s}}$	$1\ \frac{\text{m}}{\text{s}}$	$0.3\ \frac{\text{m}}{\text{s}}$	$1\ \frac{\text{m}}{\text{s}}$
k_{p1}	29.9	98.0	2.9	3.0
k_{p2}	19.3	65.3	4.3	4.1
k_{p3}	11.1	30.3	16.4	14.5
C_{Fuzzy2}	12.2	47.5	3.6	4.9

8.7 Outer loop: Pure pursuit — by radius of curvature

After seeing different ways to implement the inner loop, in this section and the following we present several approaches to design the outer loop. This outer loop takes a position reference as input, and has u_{ref} and ψ_{ref} as outputs, the values which are taken by the inner loop.

In order not to confuse the symbol e of "east" with the symbol e of "error", we will use the notation (x, y) for the position on NED frame, that is

$$\boldsymbol{\eta} = \begin{bmatrix} n \\ e \\ \psi \end{bmatrix} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}. \tag{8.50}$$

That said, first we present a way to do a pure pursuit taking into account the radius of curvature that the vehicle can perform given a certain velocity.

8.7.1 Calculation of $\psi_{\mathbf{ref}}$

Let $\mathbf{p}_k = [x_k \; y_k]^\intercal$ be the waypoint to follow. The error vector is

$$\mathbf{e}_{xy} = \begin{bmatrix} x_k - x \\ y_k - y \end{bmatrix}, \tag{8.51}$$

where $[x \; y]^\intercal$ is the actual position. Then, the direction that the vehicle should point to must be the angle of this error vector, that is

$$\psi_{\text{ref}} = \tan^{-1}\left(\frac{y_k - y}{x_k - x}\right). \tag{8.52}$$

8.7.2 Calculation of $u_{\mathbf{ref}}$

The GUANAY II has a low maneuverability, so it needs to reduce the velocity to turn. In section 7.6.3 we saw the relation between the forward velocity and the radius of curvature. The idea of setting u_{ref} is to calculate the circular section between the actual position and the waypoint such that its tangent o coincides with the actual yaw ψ, and then to calculate the forward velocity given the radius of curvature of such circle. Figure 8.36 displays this concept.

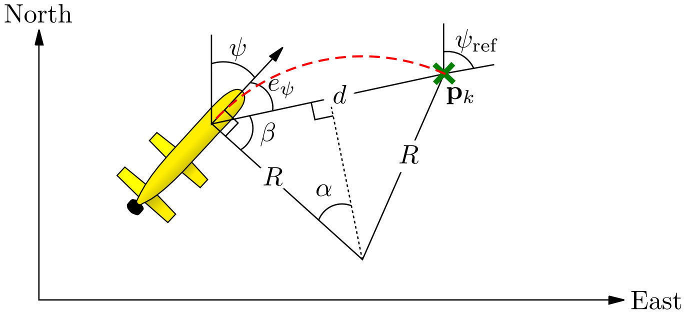


Figure 8.36: Pure pursuit regarding the radius of curvature

The distance between actual position and the waypoint is

$$\begin{aligned} d &= \sqrt{\mathbf{e}_{xy}^\intercal \cdot \mathbf{e}_{xy}} \\ &= \sqrt{(x_k - x)^2 + (y_k - y)^2}. \end{aligned} \tag{8.53}$$

From figure 8.36 we can extract that

$$\begin{aligned} \sin\alpha &= \frac{d/2}{R} \\ R &= \frac{d}{2\sin\alpha}. \end{aligned} \tag{8.54}$$

The angle α can be calculated as $\alpha = \frac{\pi}{2} - \beta$. Additionally, this angle is always positive, it is in the first quadrant, and it is closely related to the yaw error e_ψ since the yaw is tangent to the circle. If the yaw error is in the first quadrant we have $e_\psi = \frac{\pi}{2} - \beta$. Solving β we obtain that

$$\beta = \tfrac{\pi}{2} - \alpha = \tfrac{\pi}{2} - e_\psi$$
$$\alpha = e_\psi$$
$$\sin\alpha = \sin e_\psi$$

When e_ψ is in other quadrant the calculation is a little different, but as general case we can conclude that

$$\sin\alpha = |\sin e_\psi|.$$

Replacing in (8.54) we have

$$R = \frac{d}{2|\sin e_\psi|}. \tag{8.55}$$

Using (7.105) we calculate the maximum forward velocity u given this radius R:

$$u_R = 0.0000162R^3 - 0.00183R^2 + 0.0698R + 0.0677. \tag{8.56}$$

Finally, let u_m be the desired velocity for the mission. Then the velocity reference is

$$u_{\text{ref}} = \min\{u_R, u_m\}. \tag{8.57}$$

In order to switch to the next waypoint, the vehicle must be at a certain distance from the present waypoint, that is, one must define a radius of acceptance. The block diagram of this outer loop is displayed in figure 8.37. The *speed profile* represents the calculation of velocity reference u_R given the distance d and the yaw error e_ψ. The inner loop can be any of those mentioned in the previous sections.

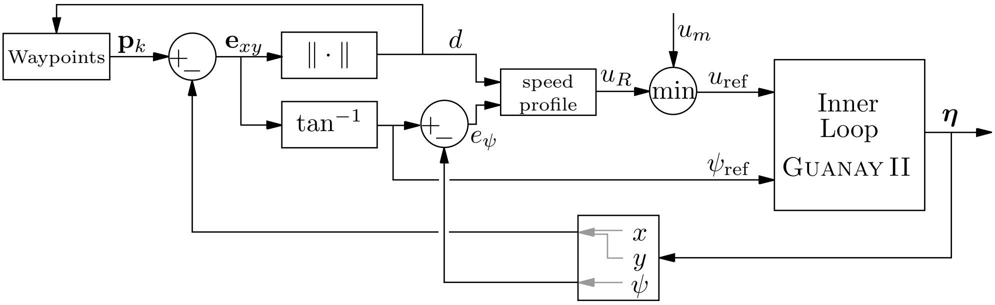


Figure 8.37: Outer loop — pure pursuit regarding the radius of curvature

8.7.3 Simulations

In what follows we show different simulations using this type of controller. We have chosen three desired velocities u_m: 0.3, 0.6 and 1 m/s. For the inner loop we have selected $C^{(u)}_{\text{Fuzzy1}}$ for the forward velocity and $C^{(\psi)}_{\text{Fuzzy1}}$ for the yaw (see section 8.3). Additionally, the simulation includes a white noise of ±5.84° in the compass.

Figure 8.38 displays the response using path 1, consisting of three waypoints with a total perimeter of about 100 m. The radius of acceptance chosen for each waypoint is 5 m. When u_m is 0.3 m/s the vehicle describes a good trajectory, and the turns are smooth and well defined. The velocity u is constant throughout the path. When u_m is 0.6 m/s, the figure shows that the vehicle needs a larger radius to turn to the second waypoint. In this case the velocity u is almost constant, it only has two little velocity reductions at the time of the turns. Finally, when u_m is 1 m/s the vehicle has difficulty making the travel. The radius of curvature to turn to the second waypoint is very large. This turn reduces the velocity reference to 0.5 m/s, and it grows up after the waypoint is reached. Notice that the noise in the compass yields a noise in the calculation of the velocity reference u_{ref}.

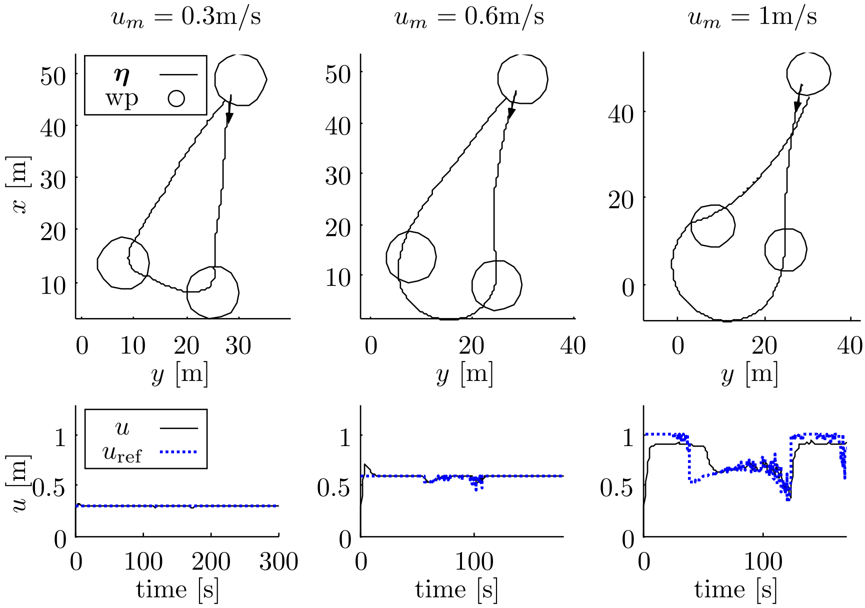


Figure 8.38: Responses using the pure pursuit regarding the radius of curvature. Path 1

Figure 8.39 displays the response using path 2. This path consists of four points with a total perimeter of about 90 m. The radius of acceptance is also 5 m. These simulations shows a similar behaviour to previous case: there is a good result at low velocities, but when it increases the vehicle describes a path with more radius of curvature and the velocity is reduced.

Figure 8.40 displays the response using path 3. This path consists of four more distanced waypoints with total perimeter of about 250 m. The radius of acceptance for the waypoints is again 5 m. Again we found that the vehicle has good performance at low velocities, but it is not very good at high velocities. Particularly, in the simulation at 1 m/s the velocity is almost constant.

All of these results show a compromise between the power to turn and the forward velocity. And as we see in section 7.6.2 the Guanay II does not has good power to turn, so the curves are very open at high velocities.

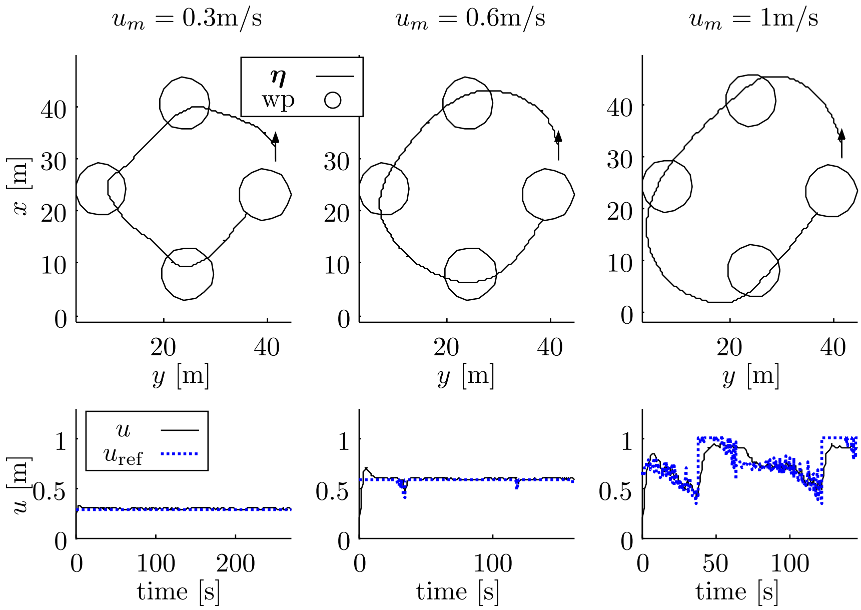


Figure 8.39: Responses using the pure pursuit regarding the radius of curvature. Path 2

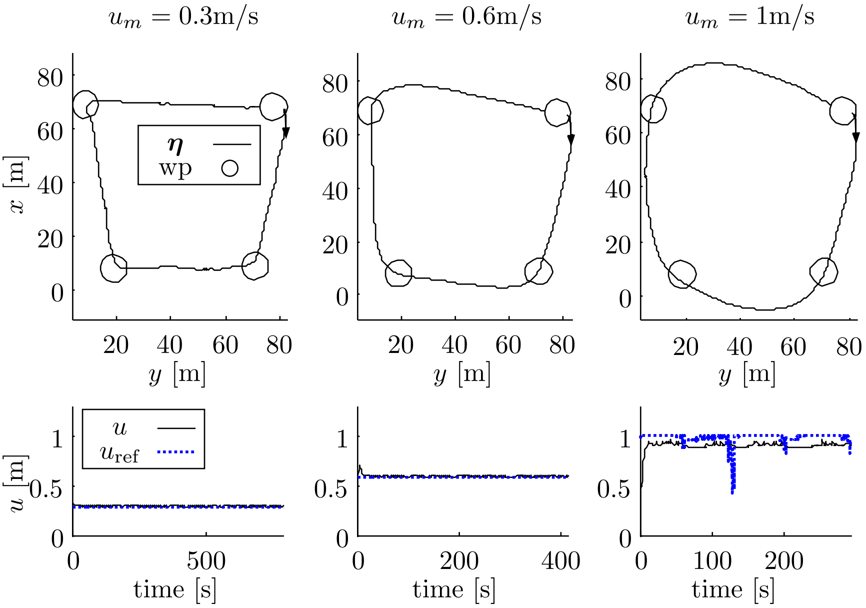


Figure 8.40: Responses using the pure pursuit regarding the radius of curvature. Path 3

8.8 Outer loop: Pure pursuit — fuzzy control

The discussion of the last section lead us to conclude that in the compromise “power to turn-forward velocity” the curves must have more relevance and the velocity needs to be decreased considerably in order to follow the waypoints in a better way. We propose in this section an outer loop method using fuzzy controllers. The idea is to reduce the velocity with respect to the three parameters: the yaw error e_ψ, the distance to the waypoint d, and the angle that the vehicle has to point to after it reaches the waypoint. This concept is shown in figure 8.41. When the vehicle reaches the waypoint $\mathbf{p}_k$ it changes its direction to point to $\mathbf{p}_{k+1}$. The new yaw reference is ψ_{ref2}, and $e_{\psi 2}$ is the change of angle between ψ_{ref} and ψ_{ref2}.

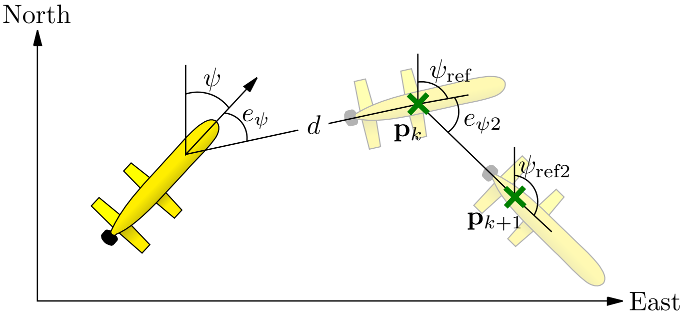


Figure 8.41: Pure pursuit — fuzzy control

8.8.1 Calculation of ψ_{ref}

The calculation of the yaw error is the same calculation of the last section. See (8.52) for further details.

Additionally, the yaw reference after reaching the waypoint is

$$\psi_{\text{ref2}} = \tan^{-1}\left(\mathbf{p}_{k+1} - \mathbf{p}_k\right) = \tan^{-1}\left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right), \tag{8.58}$$

where $\mathbf{p}_k$ is the next waypoint and $\mathbf{p}_{k+1}$ is the next to it. Then, the yaw error after reaching the waypoint will be

$$e_{\psi 2} = \psi_{\text{ref2}} - \psi_{\text{ref}}. \tag{8.59}$$

8.8.2 Calculation of u_{ref}

The calculation of u_{ref} is defined using a type-1 TSK fuzzy controller. The inputs of this controller are three: e_ψ, the yaw error; d, the distance to the waypoint; and $e_{\psi 2}$, the yaw error to follow the second waypoint. Following we present the values and membership functions used for the fuzzification. Figure 8.42 summarizes these membership functions.

$$\begin{aligned}
\mu_{1l}(e_\psi) &= \begin{cases} 1 & \text{if} \quad e_\psi < 15^\circ \\ \frac{45 - e_\psi}{30} & \text{if} \quad 15^\circ \leq e_\psi \leq 45^\circ \\ 0 & \text{if} \quad e_\psi > 45^\circ \end{cases} \\
\mu_{1h}(e_\psi) &= \begin{cases} 0 & \text{if} \quad e_\psi < 15^\circ \\ \frac{e_\psi - 15}{30} & \text{if} \quad 15^\circ \leq e_\psi \leq 45^\circ \\ 1 & \text{if} \quad e_\psi > 45^\circ \end{cases}
\end{aligned} \tag{8.60}$$

$$\begin{aligned}
\mu_{2l}(d) &= \begin{cases} 1 & \text{if} \quad d < 3 \\ \frac{10-d}{7} & \text{if} \quad 3 \le d \le 10 \\ 0 & \text{if} \quad d > 10 \end{cases} \\
\mu_{2h}(d) &= \begin{cases} 0 & \text{if} \quad d < 3 \\ \frac{d-3}{7} & \text{if} \quad 3 \le d \le 10 \\ 1 & \text{if} \quad d > 10 \end{cases}
\end{aligned} \tag{8.61}$$

$$\begin{aligned}
\mu_{3l}(e_{\psi 2}) &= \begin{cases} 1 & \text{if} \quad e_{\psi 2} < 15^\circ \\ \frac{60-e_{\psi 2}}{45} & \text{if} \quad 15^\circ \le e_{\psi 2} \le 60^\circ \\ 0 & \text{if} \quad e_{\psi 2} > 60^\circ \end{cases} \\
\mu_{3h}(e_{\psi 2}) &= \begin{cases} 0 & \text{if} \quad e_{\psi 2} < 15^\circ \\ \frac{e_{\psi 2}-15}{45} & \text{if} \quad 15^\circ \le e_{\psi 2} \le 60^\circ \\ 1 & \text{if} \quad e_{\psi 2} > 60^\circ \end{cases}
\end{aligned} \tag{8.62}$$

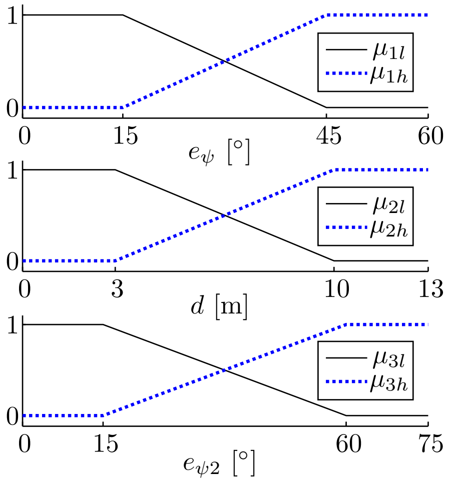


Figure 8.42: Membership functions of the fuzzy set of the Outer loop

The rules are composed for the combination of these variables. Assuming that the yaw error is small and that the vehicle is far from the waypoint it must travel at maximum velocity, that is, the velocity of the mission u_m. If the yaw error is large and the vehicle is far from de waypoint it must first reduce the velocity to do the turn and continue in the right direction. The velocity of reference for this is $0.3u_m$. Another case happens when the yaw error is large and the distance to the waypoint is small. In this case the vehicle must stop to turn without losing the waypoint. Finally, we have the case when both the yaw error and the distance are small, and in this case we look at the angle to follow to the second waypoint. If the error to the second waypoint is large the velocity is reduced to $0.2u_m$, while if this error is small it is only reduced to $0.8u_m$. The change in direction between waypoints may be considerable and it is usually necessary to make a speed reduction. Table 8.5 summarizes these rules.

Table 8.5: The control rules for the velocity reference

Rule	$\lvert e_\psi \rvert$	d	$\lvert e_{\psi 2} \rvert$	u_{ref}
R_1	small	small	small	$0.8u_m$
R_2	small	small	big	$0.2u_m$
R_3	big	small	—	0
R_4	small	big	—	u_m
R_5	big	big	—	$0.3u_m$

Using the equation (8.5) these rules gives us the following weights:

$$\begin{aligned}
\alpha_1 &= \min\{\mu_{1l}(e_\psi), \mu_{2l}(d), \mu_{3l}(e_{\psi 2})\}\\
\alpha_2 &= \min\{\mu_{1l}(e_\psi), \mu_{2l}(d), \mu_{3h}(e_{\psi 2})\}\\
\alpha_3 &= \min\{\mu_{1h}(e_\psi), \mu_{2l}(d)\}\\
\alpha_4 &= \min\{\mu_{1l}(e_\psi), \mu_{2h}(d)\}\\
\alpha_5 &= \min\{\mu_{1h}(e_\psi), \mu_{2h}(d)\}.
\end{aligned} \tag{8.63}$$

And finally, using the crisp function (8.12) of the TSK controller the velocity reference is

$$u_{\mathrm{ref}} = \frac{0.8\alpha_1 + 0.2\alpha_2 + \alpha_4 + 0.3\alpha_5}{\sum_{m=1}^{5} \alpha_m} u_m. \tag{8.64}$$

Again, in order to define the moment to switch to the next waypoint, one must establish a radius of acceptance. The block diagram of this outer loop is displayed in figure 8.43. The inner loop can be any of those mentioned in the previous sections.

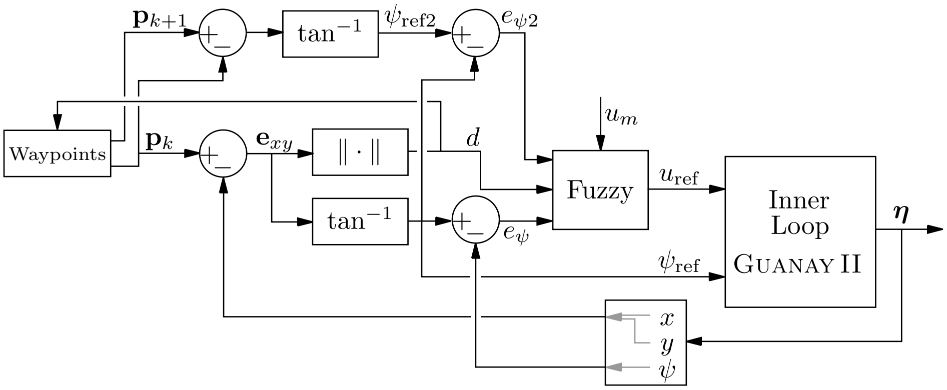


Figure 8.43: Outer loop — pure pursuit using a fuzzy controller

8.8.3 Simulations

Following we show different simulations using this type of controller. For them we have chosen three velocities u_m equal to 0.3, 0.6 and 1 m/s. For the inner loop we have selected

$C^{(u)}_{\text{Fuzzy1}}$ (see section 8.3) for the forward velocity and C_{Fuzzy2} (see section 8.6) for the yaw. Additionally, the simulation includes a white noise of ±5.84° in the compass.

Figure 8.44 displays the response using path 1. The radius of acceptance is 5 m. At first glance, we see that it performs better than the controller of the previous section (figure 8.38), specially when u_m is big. Using this fuzzy controller the velocity is reduced considerably when the vehicle is near to the waypoint, and it can negotiate the curve in a better way. Take as an example the trajectory when u_m is 1 m/s: at the beginning it goes fast, but near to the first waypoint it reduces its velocity tol 0.3 m/s. It reaches the waypoint and continues with slow velocity until it points to the next. At this time it increases its speed and its travel continues.

Figure 8.45 displays the response using path 2, and figure 8.46 for path 3. Again, we see an improvement with respect to the controller of the last section.

Concluding, the inclusion of the type-1 TSK fuzzy controller has allowed a better management of the forward velocity thanks to the versatility and easy implementation of it. This management is very important to the GUANAY II AUV because the vehicle has limited power to turn, and thus it can not navigates at a constant velocity all the time.

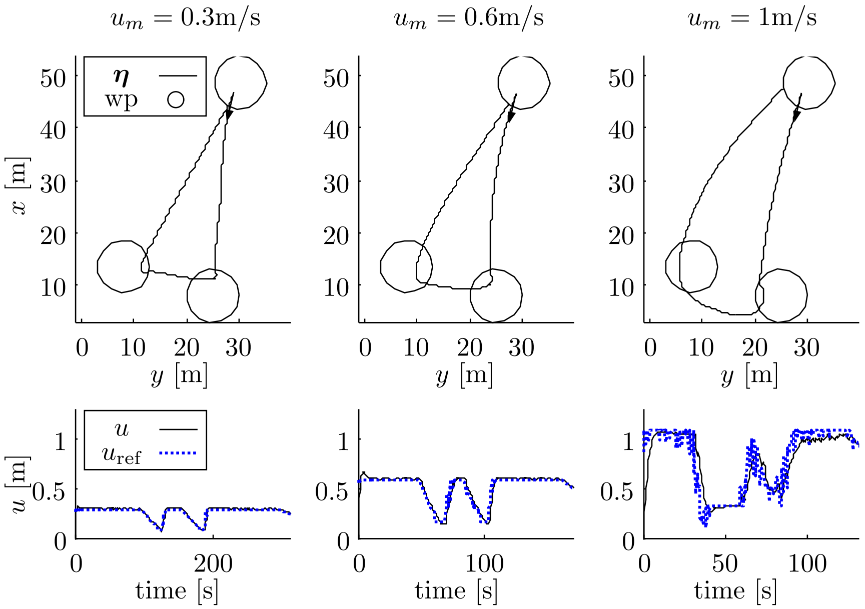


Figure 8.44: Responses using pure pursuit with fuzzy controller. Path 1

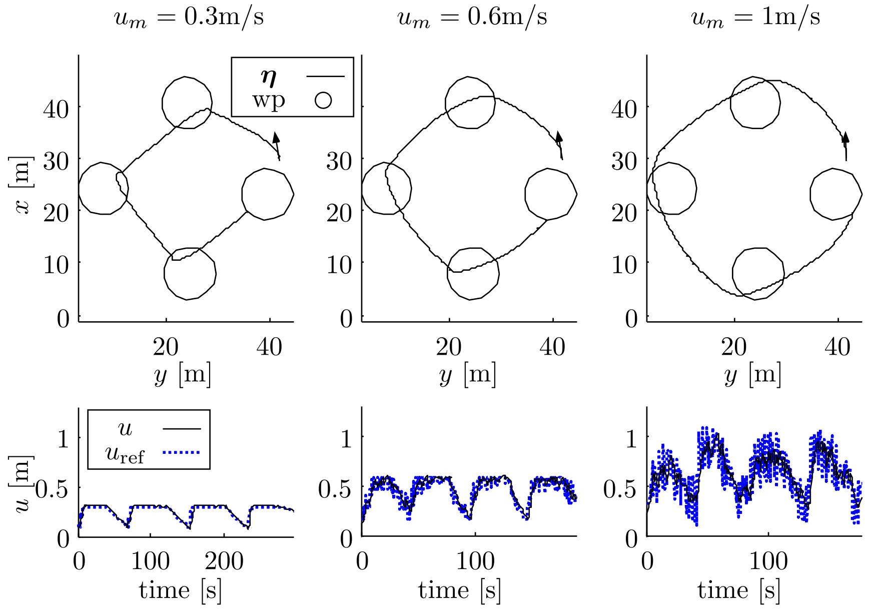


Figure 8.45: Responses using pure pursuit with fuzzy controller. Path 2

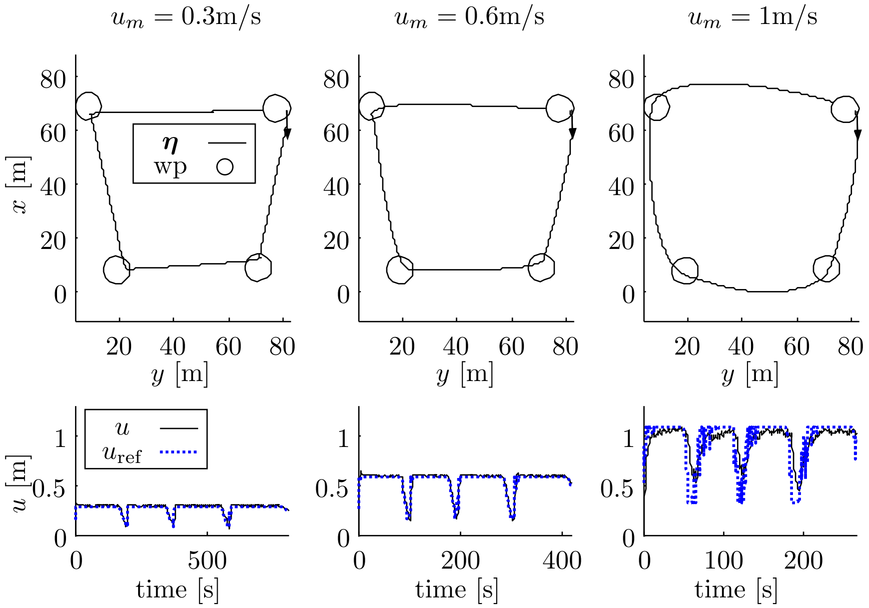


Figure 8.46: Responses using pure pursuit with fuzzy controller. Path 3

8.9 Outer loop: Path following

In this section we present another form to follow a path. In this case the path is defined by a continuous line of class C^1. The idea is to be as close as possible to the path, but there is no a specific radius of acceptance. To control the yaw we use the algorithm proposed by Maurya et al. [83] consisting in a proportional-integral controller in function of the distance to the path; for the forward velocity we use a fuzzy controller and a limitation with respect to the radius of curvature. Following we present the construction of each part.

8.9.1 Calculation of ψ_{ref}

The distance e is defined as the error to the path, which we want to be zero, and it is the length of a segment perpendicular to the path. Although the path is smooth, we will assume that the path is defined for many waypoints in order to simplify the calculation. The distances between the waypoints are small, so the arcs can be considered as straight lines. The path following problem that we consider can be simply explained by referring to figure 8.47.

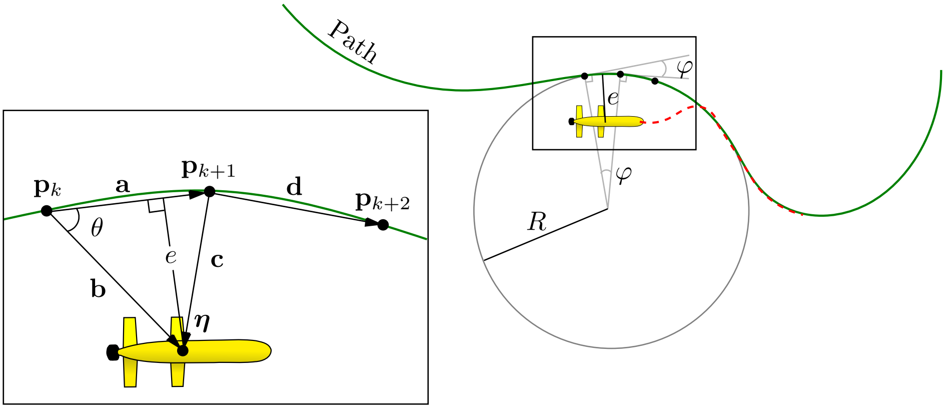


Figure 8.47: Path following

Let the vectors

$$\mathbf{a} = \mathbf{p}_{k+1} - \mathbf{p}_k, \qquad \mathbf{b} = \boldsymbol{\eta} - \mathbf{p}_k,$$
$$\mathbf{c} = \boldsymbol{\eta} - \mathbf{p}_{k+1}, \qquad \mathbf{d} = \mathbf{p}_{k+2} - \mathbf{p}_{k+1}. \tag{8.65}$$

Let θ be the angle between $\mathbf{a}$ and $\mathbf{b}$. Then

$$\theta = \beta - \alpha \tag{8.66}$$

where

$$\beta = \tan^{-1}(\mathbf{b}) = \tan^{-1}\left(\frac{b_y}{b_x}\right) \qquad \alpha = \tan^{-1}(\mathbf{a}) = \tan^{-1}\left(\frac{a_y}{a_x}\right) \tag{8.67}$$

Notice that α is the direction of the path.

The distance to the path e will be

$$e = \|\mathbf{b}\| \sin\theta. \tag{8.68}$$

Maurya proposes a proportional-integral controller around this distance which guarantees the convergence to zero:

$$\xi = -\frac{1}{U}\left(K_1 e - K_2 \int_0^t e(\tau)d\tau\right), \tag{8.69}$$

where K_1 and K_2 are positive gains defining the desired natural frequency and the damping factor, and

$$U = \sqrt{u^2 + v^2} \tag{8.70}$$

is the total speed, which is calculated from the forward velocity u and the lateral velocity v. We calculate an angle γ respect to the path using this control variable ξ:

$$\gamma = \sin^{-1}(\xi_s) \tag{8.71}$$

where

$$\xi_s = \text{sat}(\xi) = \begin{cases} \xi & \text{if } |\xi| \leq 1, \\ 1 & \text{if } \xi > 1, \\ -1 & \text{if } \xi < -1. \end{cases} \tag{8.72}$$

Finally, the reference angle ψ_{ref} is calculated as

$$\psi_{\text{ref}} = \gamma + \alpha. \tag{8.73}$$

Figure 8.48 shows a representation of this calculation. Note that ξ_s will be always contrary to e and forces the vehicle towards the path. The saturation is needed to guarantee that the argument of $\sin^{-1}(\cdot)$ lies in the interval $[-1, 1]$. Maurya et al. [83], calculating the derivative $\dot{e}$ and applying this controller, make the proof of the convergence of the error to zero for a straight line path, even in the presence of currents (thanks to the integral action).

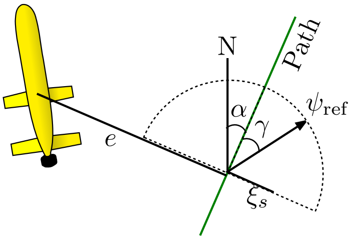


Figure 8.48: Calculation of ψ_{ref} on path following

8.9.2 Calculation of u_{ref}

The calculation of u_{ref} is defined using a fuzzy controller. Additionally, as the paths are smooth, we complement it with a control that takes into account the radius of curvature of the path. The inputs of the fuzzy controller are two: e_ψ, the yaw error, and e, the distance to the path. We present next the values and membership functions used for the fuzzification. Figure 8.49 summarizes these membership functions.

$$\begin{aligned} \mu_{1l}(e_\psi) &= \begin{cases} 1 & \text{if} \quad e_\psi < 15^\circ \\ \frac{45-e_\psi}{30} & \text{if} \quad 15^\circ \le e_\psi \le 45^\circ \\ 0 & \text{if} \quad e_\psi > 45^\circ \end{cases} \\ \mu_{1h}(e_\psi) &= \begin{cases} 0 & \text{if} \quad e_\psi < 15^\circ \\ \frac{e_\psi-15}{30} & \text{if} \quad 15^\circ \le e_\psi \le 45^\circ \\ 1 & \text{if} \quad e_\psi > 45^\circ \end{cases} \end{aligned} \tag{8.74}$$

$$\begin{aligned} \mu_{2l}(e) &= \begin{cases} 1 & \text{if} \quad e < 3 \\ \frac{10-e}{7} & \text{if} \quad 3 \le e \le 10 \\ 0 & \text{if} \quad e > 10 \end{cases} \\ \mu_{2h}(e) &= \begin{cases} 0 & \text{if} \quad e < 3 \\ \frac{e-3}{7} & \text{if} \quad 3 \le e \le 10 \\ 1 & \text{if} \quad e > 10 \end{cases} \end{aligned} \tag{8.75}$$

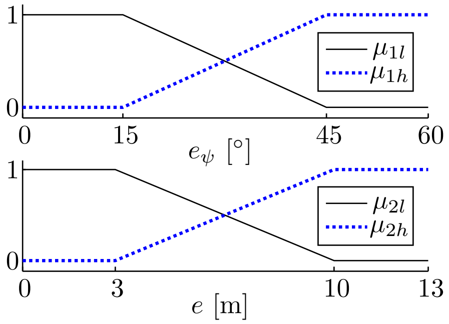


Figure 8.49: Membership functions of the fuzzy set of path following

The rules are composed for the combination of these two variables. Assuming that the yaw error is small and that the vehicle is close to the path it must travel at maximum velocity, that is, the velocity of the mission u_m. If the yaw error is large but the vehicle is close to the path, it must first reduce the velocity to do the turn and continue in the right direction, and the velocity of reference is set to $0.5u_m$. Another case happens when the yaw error is small and the distance to the path is large. In this case the vehicle points perpendicular to the path (since ξ_s saturates to ± 1 and then $\gamma = \pm\pi/2$), so it must travel slowly in order to avoid an underdamped response. The velocity of reference is $0.3u_m$. Finally, we have the case when both the yaw error and the distance are large. Here the vehicle must stop to turn completely because it is far from the path and pointing in a wrong direction. Table 8.6 summarizes these rules.

Table 8.6: The control rules for the velocity reference in path following

Rule	$\lvert e_\psi \rvert$	e	u_{ref}
R_1	small	small	u_m
R_2	big	small	$0.5u_m$
R_3	small	big	$0.3u_m$
R_4	big	big	0

Using the equation (8.5) these rules gives us the following weights:

$$\begin{aligned}\alpha_1 &= \min\{\mu_{1l}(e_\psi), \mu_{2l}(d)\}\\ \alpha_2 &= \min\{\mu_{1h}(e_\psi), \mu_{2l}(d)\}\\ \alpha_3 &= \min\{\mu_{1l}(e_\psi), \mu_{2h}(d)\}\\ \alpha_4 &= \min\{\mu_{1h}(e_\psi), \mu_{2h}(d)\}.\end{aligned} \tag{8.76}$$

Finally, the crisp function of the TSK controller (8.12) is used to calculate the velocity reference:

$$u_{\text{fuzzy}} = \frac{\alpha_1 + 0.5\alpha_2 + 0.3\alpha_3}{\sum_{m=1}^{4} \alpha_m} u_m. \tag{8.77}$$

On the other hand, we know that the vector **a** is perpendicular to the radius of the path. Letting φ by the angle between the vectors **a** and **d**, that is

$$\varphi = \tan^{-1}(\mathbf{d}) - \tan^{-1}(\mathbf{a}), \tag{8.78}$$

it is easy to see that $|\varphi|$ is equal to the opening of the circular segment (see figure 8.47). Then

$$\begin{aligned}\|\mathbf{a}\| &= R|\varphi|\\ R &= \frac{\|\mathbf{a}\|}{|\varphi|}\end{aligned} \tag{8.79}$$

From (7.105) we calculate the maximum forward velocity u given this radius R:

$$u_R = 0.0000162R^3 - 0.00183R^2 + 0.0698R + 0.0677. \tag{8.80}$$

Finally, we take the minimum value of the calculated velocities as a constraint of the control:

$$u_{\text{ref}} = \min\{u_{\text{fuzzy}}, u_R\}. \tag{8.81}$$

At the beginning we said that the path is composed by many waypoints which are very close between them. For the calculation of ψ_{ref} and u_{ref} we need three consecutive waypoints. The way to switch to the next waypoint consists in calculating the angle between the vectors **a** and **c** (see figure 8.47) and to switch when it exceeds $\pi/2$. The block diagram of this outer loop is displayed in figure 8.50. The *speed profile* represents the calculation of velocity reference u_R given the norm of **a** and the angle φ. The inner loop can be any of those mentioned in the previous sections.

8.9.3 Simulations

Following we show different simulations using this type of controller. For them we have chosen three velocities u_m: 0.3, 0.6 and 1 m/s. For the inner loop we have selected $C^{(u)}_{\text{Fuzzy1}}$ (see section 8.3) for the forward velocity and C_{Fuzzy2} (see section 8.6) for the yaw. Additionally, the simulation includes a white noise of ±5.84° in the compass.

Figure 8.51 displays the response using path 4, which consists of a loop making an eight-shaped path. When u_m is 0.3 m/s the vehicle describe a good trajectory, and the turns are smooth and well defined. However, there is small error respect to the path. The velocity u is almost constant throughout the path. When u_m is 0.6 m/s the figure shows a underdamped

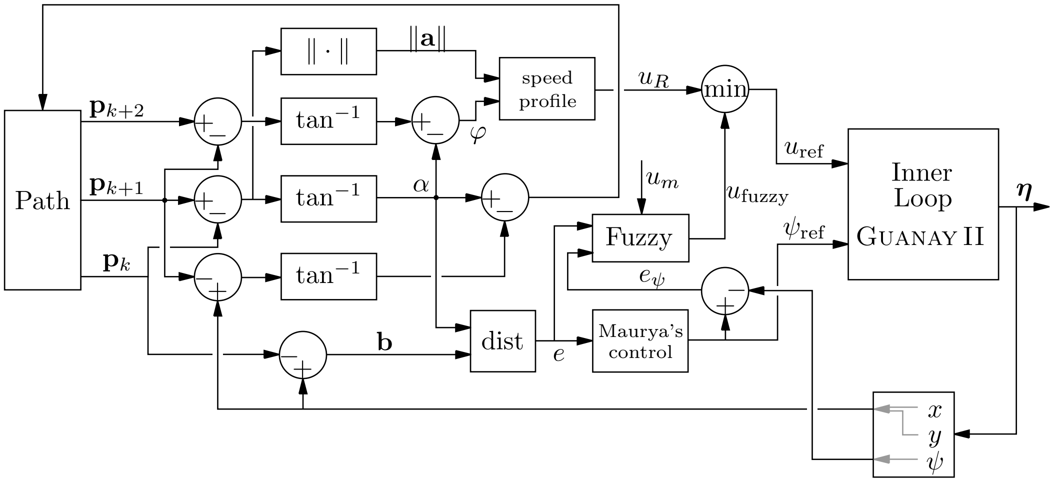


Figure 8.50: Outer loop — path following

response at the beginning but it is smooth for the rest of the path. Again, we see a considerable error respect to the path. Regarding the velocity, it is almost constant but has to be reduced at some times. Finally, when u_m is 1 m/s the vehicle has difficulty making the travel. The underdamped response is more accentuated and it misses smooth motion. The velocity grows and falls linked to the underdamped response seen in the path.

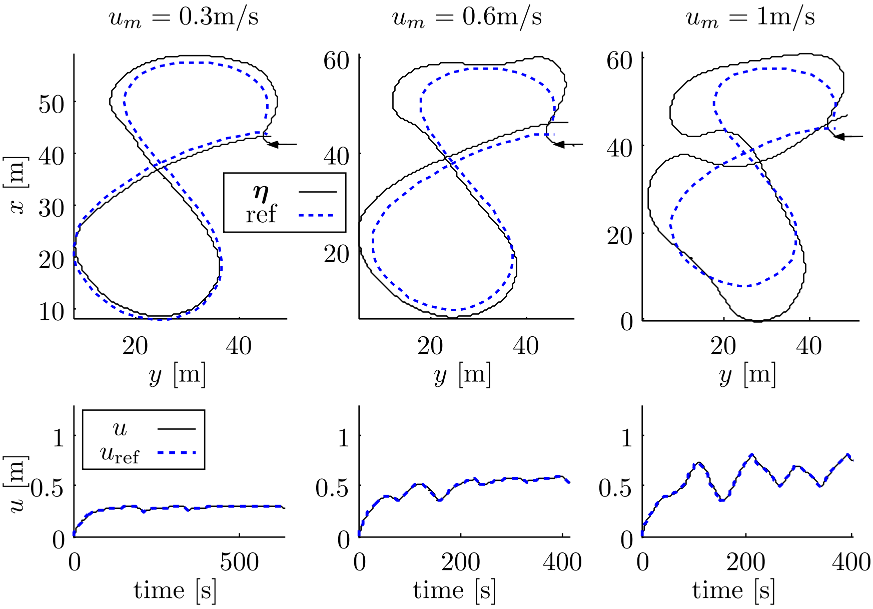


Figure 8.51: Responses using the path following controller. Path 4

Notice that the radius of the path is small and it is difficult to follow it at high velocities. Figure 8.52 displays the response using path 5, which is a more extended one path with several straight lines and curves. At first glance the performance is better than for path 4. When u_m is 0.3 m/s the vehicle follows the path very well and the velocity is constant. When u_m is 0.6 m/s the vehicle performs sluggishly at the beginning but it continues the path with a small error. In this case the velocity is almost constant and has little dips at the turns. Finally, when u_m is 1 m/s the underdamped response is more obvious but the vehicle follows the path correctly. The velocity takes some dips in order to perform the curves.

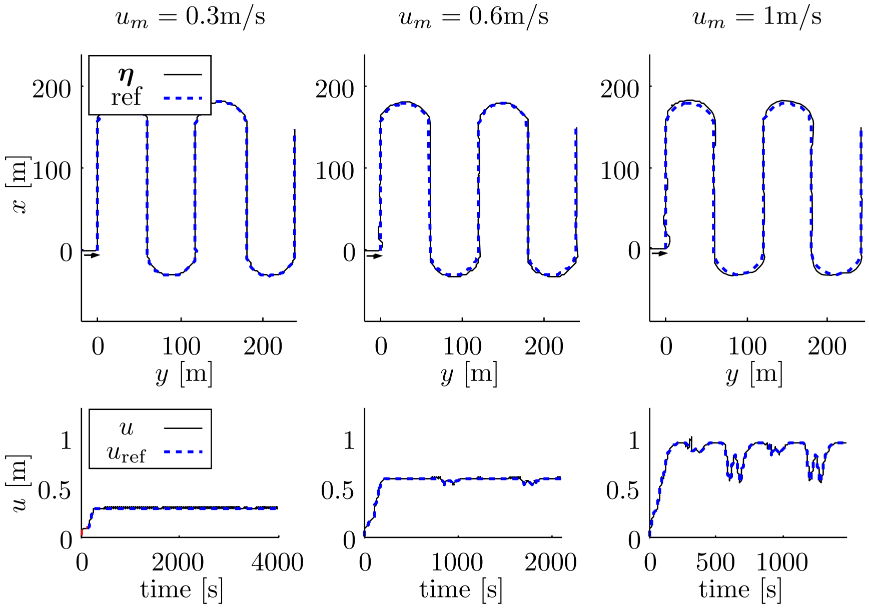


Figure 8.52: Responses using the path following controller. Path 5

Finally, notice that the TSK fuzzy controller help to reduce the velocity at the right times. The absence of this block would cause the results to be similar to section 8.7, where the radius of curvature was too high depending on the forward velocity.

8.10 Conclusions

In this chapter we showed several controllers needed to give an automatic movement to the vehicle and to follow a preestablished set of waypoints.

First we focused on the calculation of different linear controllers with respect to the forward velocity u using the non-linear model. The non-linear is nearly a piecewise linear model. The idea was to develop a set of zonally differentiated controllers to control both the yaw and forward velocity given certain references for them.

Next, we proposed a way to merge the set of controllers into one controller with a type-1 TSK fuzzy controller. The advantage of this conception of merging lies on that TSK fuzzy controller uses analytic functions in the defuzzification instead of linguistic terms, and thus it can be linked to different controllers which are designed from the dynamic model. The performance of this controller was compared to that of another controller based on gain scheduling. The results were very similar.

Although we noted the importance of the fuzzy controller for combining linear controllers, and this was one of the goals of the thesis, we showed that the control of the yaw was not possible to implement satisfactorily due to physical constraints. The goodness of this controller was presented by simulations assuming that the vehicle had enough power for the torque and that its propulsion can go up to 2 m/s.

Next, we proposed another way to control the yaw, but this time using a fuzzy controller which was not based in the non-linear model but in maximum torque and the yaw error. Simulations showed a good performance.

Finally we presented two methods for pure pursuit and one method for path following. These methods correspond to the outer loop and their outputs are the desired references of the yaw and the forward velocity.

The first method makes a calculation of the path trying to preserve the forward velocity. For this it takes into account the radius of curvature that the vehicle can perform.

The second method uses a fuzzy controller based on the yaw error, the distance to the waypoint, and the angle to the second waypoint. It is more strict with the forward velocity in order to do better turns.

The last method, used for path following, consisted in a proportional-integral controller to define the yaw reference which was proposed in the literature. To control the forward velocity we presented a controller which takes into account two controllers: First a fuzzy controller with the yaw error and distance to the path as inputs; and second, a controller that calculates the maximum velocity considering to the radius of the path.

All of the outer loops were tested and compared with simulations. The results were satisfactory.

Chapter 9

Field Tests: Automatic Control

In this chapter we compare the different controllers on field tests. As we mentioned in section 6.5, the tests were taken in the Catalonia Olympic Channel because it is very quiet and has large enough dimensions. The chapter ends with the results from a test off the coast of Vilanova i La Geltrú.

9.1 Outer loop: Pure pursuit — by radius of curvature

This section shows the tests where the outer loop of pure pursuit by radius of curvature was used. The inner loop is divided in three subsections. First we present the performance of the fuzzy controller C_{Fuzzy1}; second we compare this fuzzy controller with the gain scheduled controller; finally, we present the performance of the second approach for fuzzy controllers C_{Fuzzy2}, which is with respect to the yaw error.

9.1.1 Inner loop: Fuzzy control 1

We present first the field test using the fuzzy controller $C^{(u)}_{\text{Fuzzy1}}$ for the forward velocity and $C^{(\psi)}_{\text{Fuzzy1}}$ for the yaw (see section 8.3). The radius of acceptance is 8 m and the path used is number 1 (with 3 waypoints).

Figure 9.1 shows the test results when the velocity of the mission, u_m, is 0.3 m/s. At first glance we can see that the vehicle follows correctly the three waypoints. Likewise, the forward velocity is almost constant and coincides with 0.3 m/s. However, we can see one of the problems seen in the last chapter concerning the saturation of the thrusters. During the first 100 s the vehicle is pointing to the first waypoint correctly, but the big gain of the controller and the derivative action made the thrusters oscillate and saturate in two directions, which is not convenient. From 100 s to 220 s the vehicle uses the maximum power to turn to the second waypoint, and from 220 s on we see again the oscillations in the thrusters to maintain a specific yaw. Additionally, it is easy to see that the traveled path has a significant radius of curvature when it changes waypoint. As we saw in the last chapter (section 8.7.3), this is because of the outer loop.

The figure also shows a simulation of the traveled path. It is interesting to see the difference with respect to the experimental data, which we may attribute to currents.

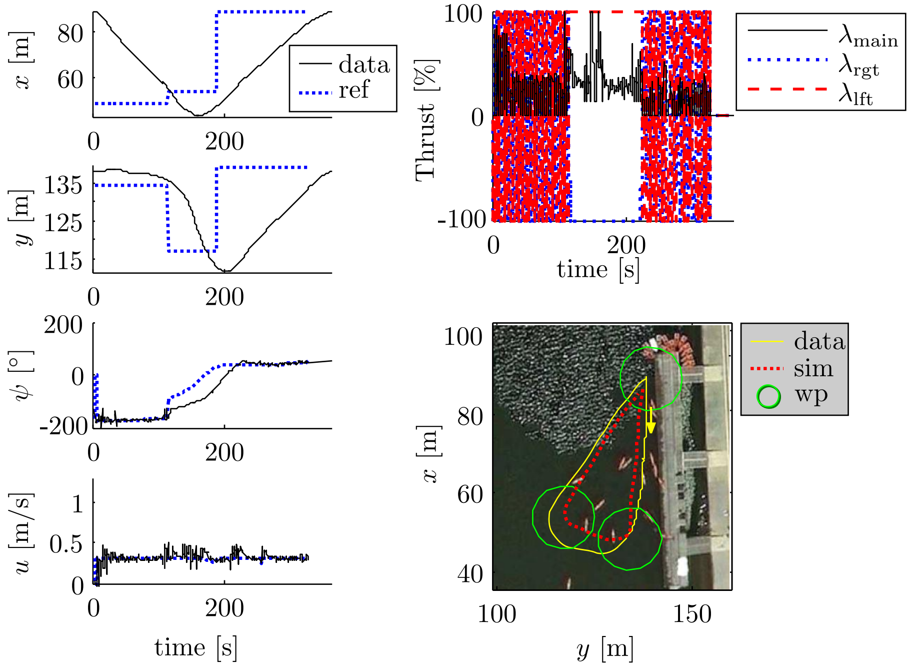


Figure 9.1: Field tests. Outer loop radius of curvature. Inner loop C_{Fuzzy1}. Path 1. $u = 0.3\text{m/s}$

Figure 9.2 shows test results when forward velocity for the mission is 0.6 m/s. In this test we see again the problems related to the power of the thrusters, where they oscillate and saturate in some periods, concretely, when the vehicle is pointing towards the waypoint. At the same time, the problem with the outer loop is more evident, because the radius to do the turn is even larger. The simulated trajectory is, in this case, very similar to the experimental data.

9.1.2 Inner loop: Comparative Fuzzy 1 — Gain Scheduled

In this section we present a comparative of the fuzzy controller 1 and the gain scheduling controller, *i.e.* a comparative of the controllers $C^{(u)}_{\text{Fuzzy1}}$ and $C^{(u)}_{\text{GainS}}$ for the forward velocity, and controllers $C^{(\psi)}_{\text{Fuzzy1}}$ and $C^{(\psi)}_{\text{GainS}}$ for the yaw (see sections 8.3 and 8.4). These tests were made using the path number 2 (with 4 waypoints).

Figure 9.3 shows the comparative when the forward velocity is 0.3 m/s and the radius of acceptance is 8 m. The differences are very small. We can see that velocity, thrusters, yaw, and path traveled are very similar for both controllers.

Figure 9.4 shows the comparative when the forward velocity is 0.6 m/s and the radius of acceptance is 3 m. The vehicle follows the waypoints for both controllers, but again the radius of curvature due to the outer loop is very large. Notice that the large radius provokes that the velocity reference u_{ref} decreases at some times. This figure also shows that the results of using the fuzzy and gain scheduling controllers are very close.

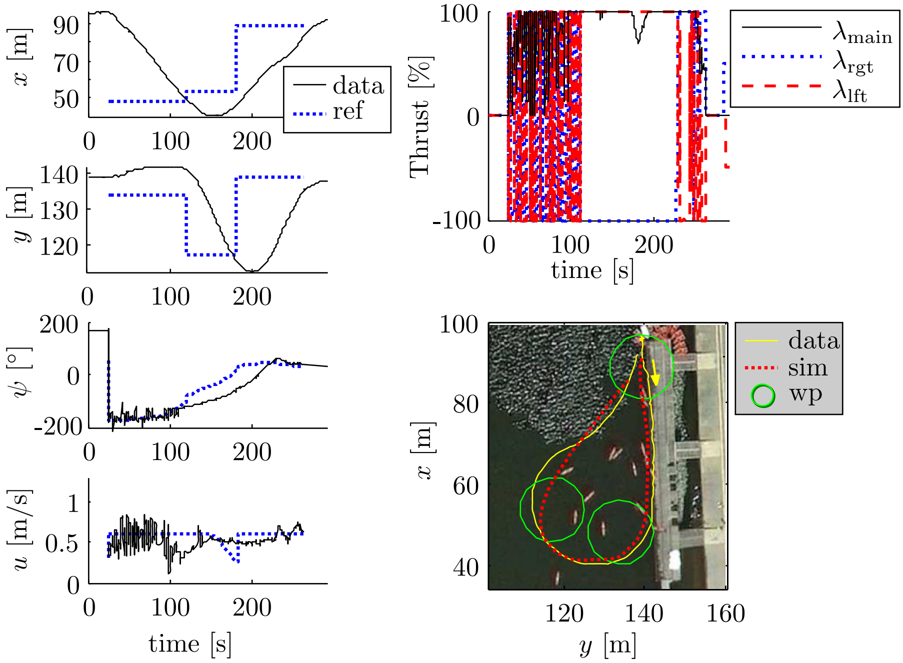


Figure 9.2: Field tests. Outer loop radius of curvature. Inner loop C_{Fuzzy1}. Path 1. $u = 0.6$m/s

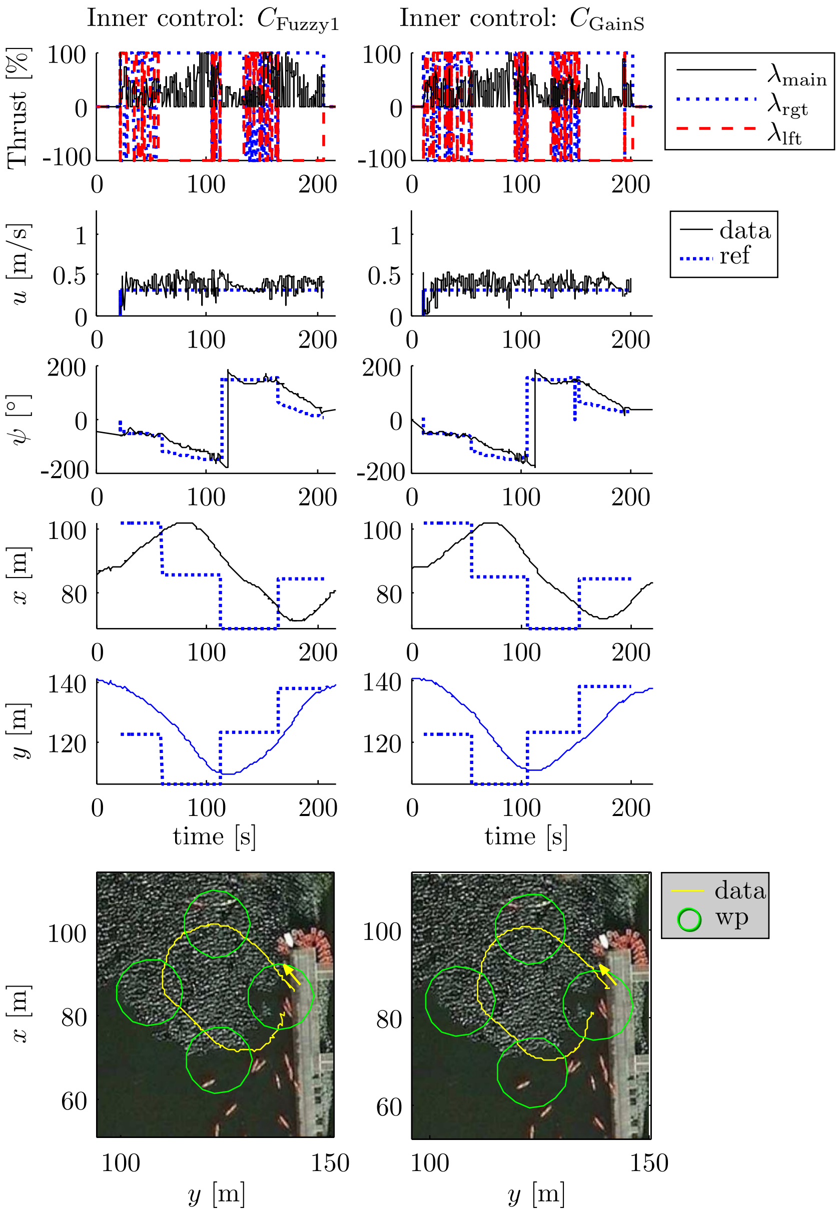


Figure 9.3: Field tests. Outer loop radius of curvature. Comparative of C_{Fuzzy1} with C_{GainS}. Path 2. $u = 0.3\text{m/s}$

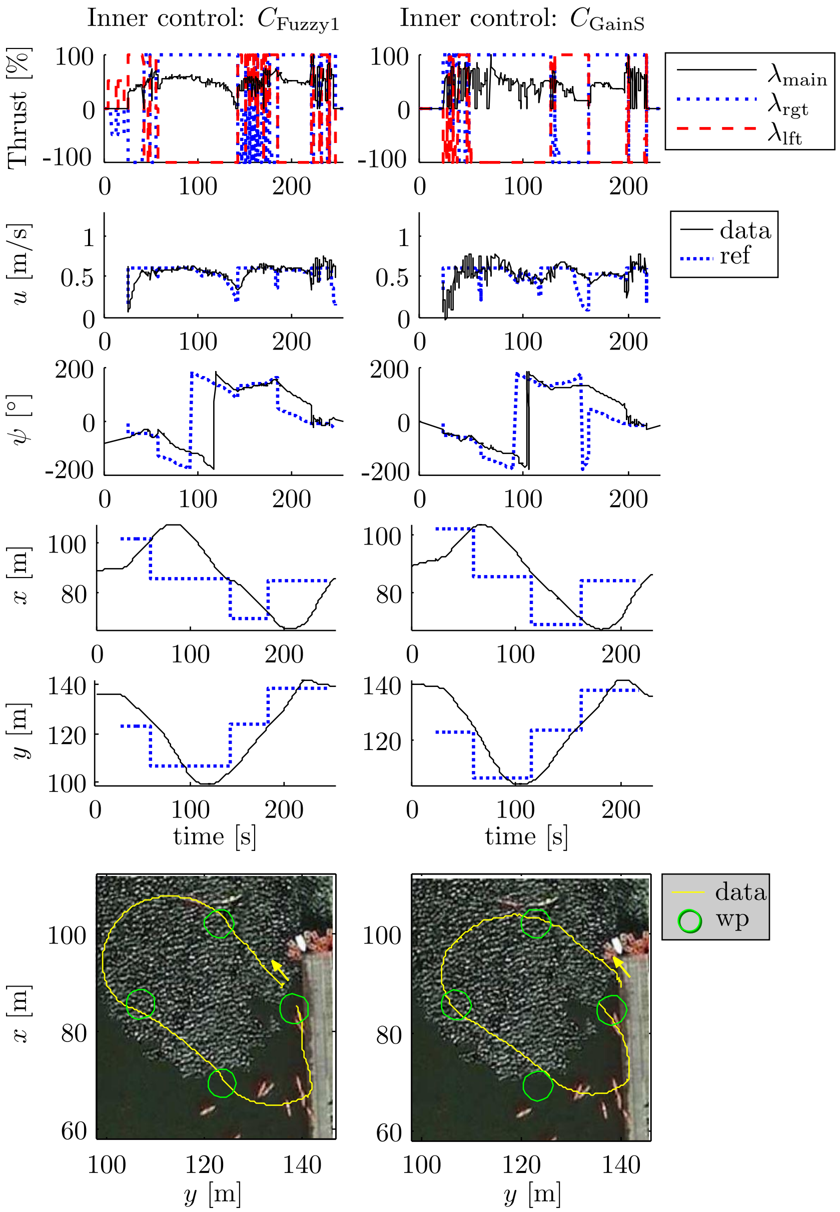


Figure 9.4: Field tests. Outer loop radius of curvature. Comparative of C_{Fuzzy1} with C_{GainS}. Path 2. $u = 0.6\text{m/s}$

9.1.3 Inner loop: Fuzzy control 2

We present next field tests with the velocity controller $C^{(u)}_{\text{Fuzzy1}}$, but using the fuzzy controller C_{Fuzzy2} for yaw control (see section 8.6). The radius of acceptance is 3 m and the path used is number 2.

Figure 9.5 shows a comparative using the 3 controllers for the yaw control: C_{Fuzzy2}, k_{p1} and k_{p3}. See section 8.6 for the description of these controllers. The velocity used in the three cases was 0.3 m/s. At first glance, we can see an improvement regarding the power used by the thrusters, because they do not oscillate as in the previous tests: the k_{p1} controller does not provoke large oscillations, and although oscillations increment a little using k_{p3} and C_{Fuzzy2}, their magnitude remains acceptable. However, looking at the path, we see that the outer loop with this radius of curvature causes some problems when trying to reach the waypoints. This figure displays also a simulation of the navigation using C_{Fuzzy2}. It shows some differences with respect to the experimental data, which again may be due to the water currents.

9.2 Outer loop: Pure pursuit — fuzzy control

This section shows the tests where the outer loop of pure pursuit with fuzzy control was used. As we saw in section 8.8, the idea of this controller is to improve the turns when the vehicle reaches a waypoint by further reducing the forward velocity. In regard of the inner loop, we present a comparative between the controllers for the yaw error.

9.2.1 Inner loop: Fuzzy control 2

We present several comparatives between three controllers to control the yaw: k_{p1}, k_{p3}, and C_{Fuzzy2}, which were defined in section 8.6. In all cases the velocity controller was $C^{(u)}_{\text{Fuzzy1}}$ (see section 8.3) and the radius of acceptance was 5 m.

Figure 9.6 shows a comparative using 0.3 m/s as the velocity of the mission, and using path number 1. The results are very good. We see in the three cases a great improvement when making the turns to go to the next waypoint. This is because the forward velocity was reduced when the vehicle reached a waypoint. With respect to the power used by the thrusters, we can see that both k_{p1} and C_{Fuzzy2} do not saturate the thrusters, and the fuzzy controller achieves the yaw reference faster. The k_{p3} controller almost saturates the thrusters. This figure also shows a simulation using C_{Fuzzy2}. The results are very similar to the experimental data.

Figure 9.7 shows the same comparative, but this time using a mission velocity of 0.6 m/s. The results are very similar, but notice the reduction in the velocity reference. The vehicle reaches 0.6 m/s only during two small time intervals, namely those corresponding to the two long straight lines.

Finally, figure 9.8 shows the same comparative but using more separated waypoints, and traveling at 1 m/s. Again the results are very similar. The velocity reference fluctuates between 0.2 and 1 m/s, increasing on the straight lines, and decreasing when performing the big turns. We should mention that, when the vehicle was under the action of the k_{p3} controller, it collided with a buoy (after reaching the first waypoint), but it successfully recovered its course. This can be seen in the XY-chart and yaw-chart. The simulation using the C_{Fuzzy2} controller shows a trajectory, in the first two segments, quite similar to the experimental data, while the following segments are smoother in the simulation and display some noticeable differences with respect to the experimental results.

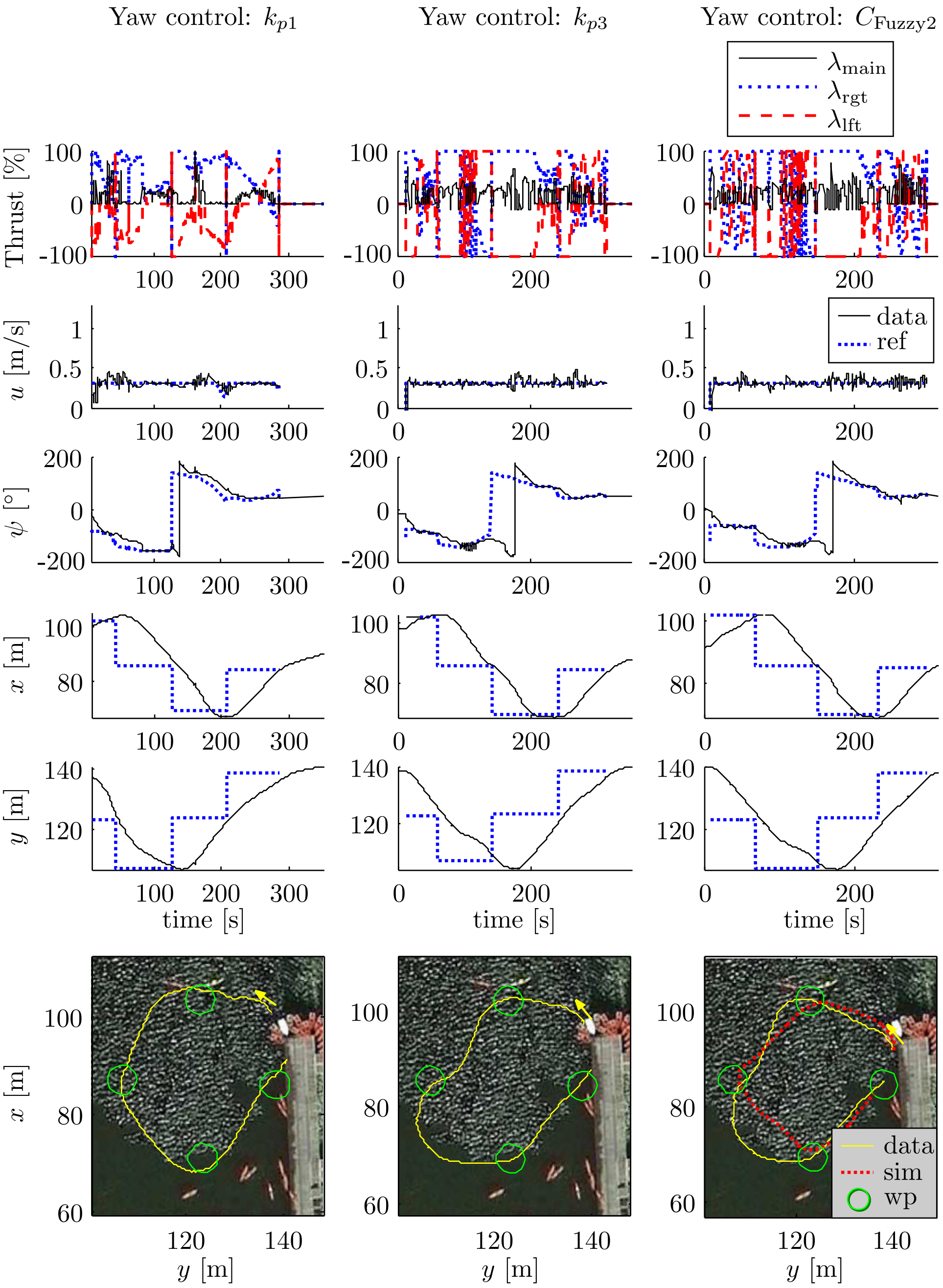


Figure 9.5: Field tests. Outer loop radius of curvature. Comparative of C_{Fuzzy2} with proportional controllers. Path 2. $u = 0.3$m/s

In conclusion, we see an improvement using both outer loop and inner loop controllers. The fuzzy control for the outer loop performs the turns in a better way, and reduces the forward velocity in order to achieve this. On the other hand, the fuzzy controller for the yaw does not introduce large oscillations or saturation of the thrusters, and reaches the yaw reference very well.

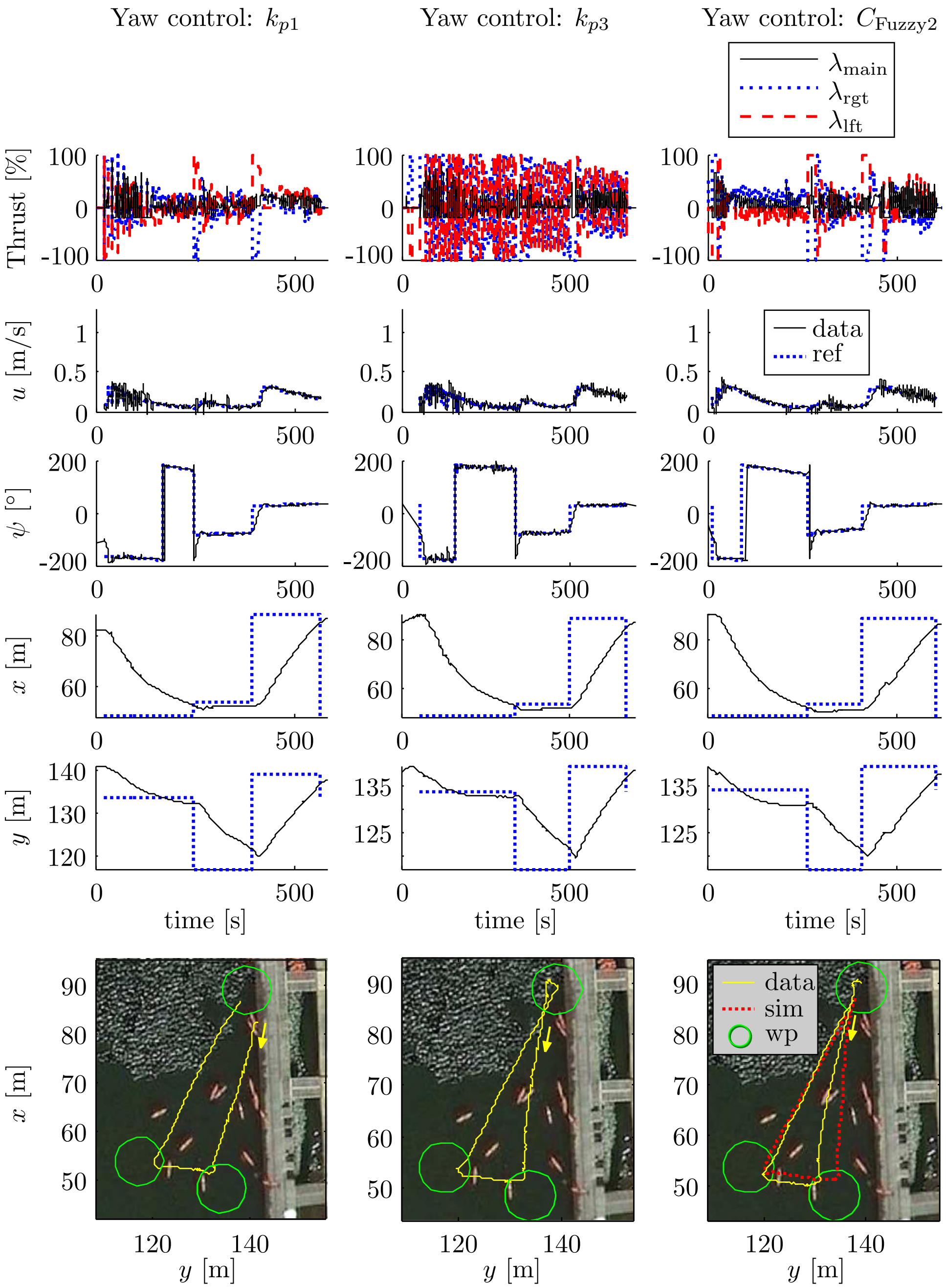


Figure 9.6: Field tests. Outer loop fuzzy control. Comparative of C_{Fuzzy2} with proportional controllers. Path 1. $u = 0.3$m/s

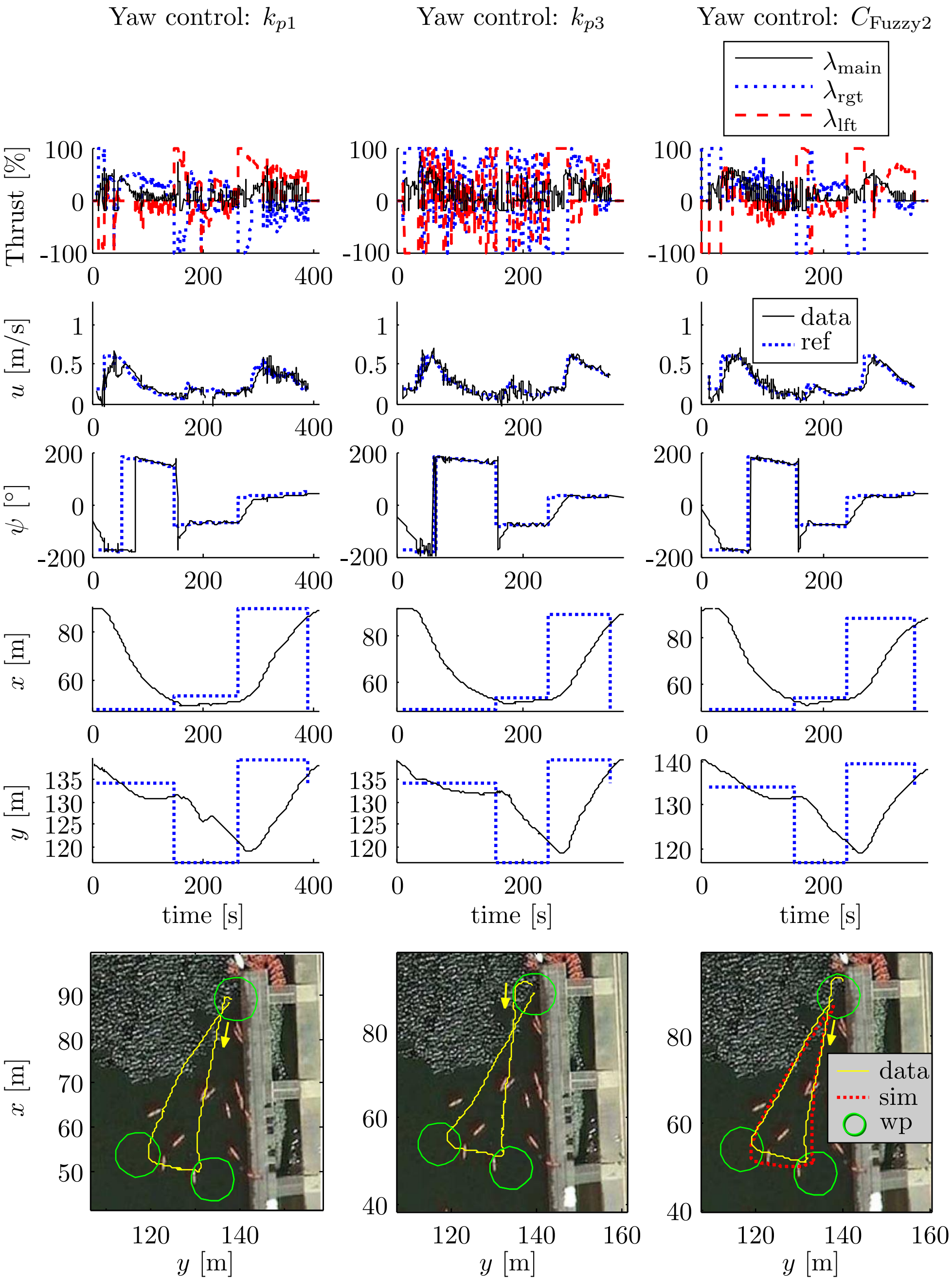


Figure 9.7: Field tests. Outer loop fuzzy control. Comparative of C_{Fuzzy2} with proportional controllers. Path 1. $u = 0.6\text{m/s}$

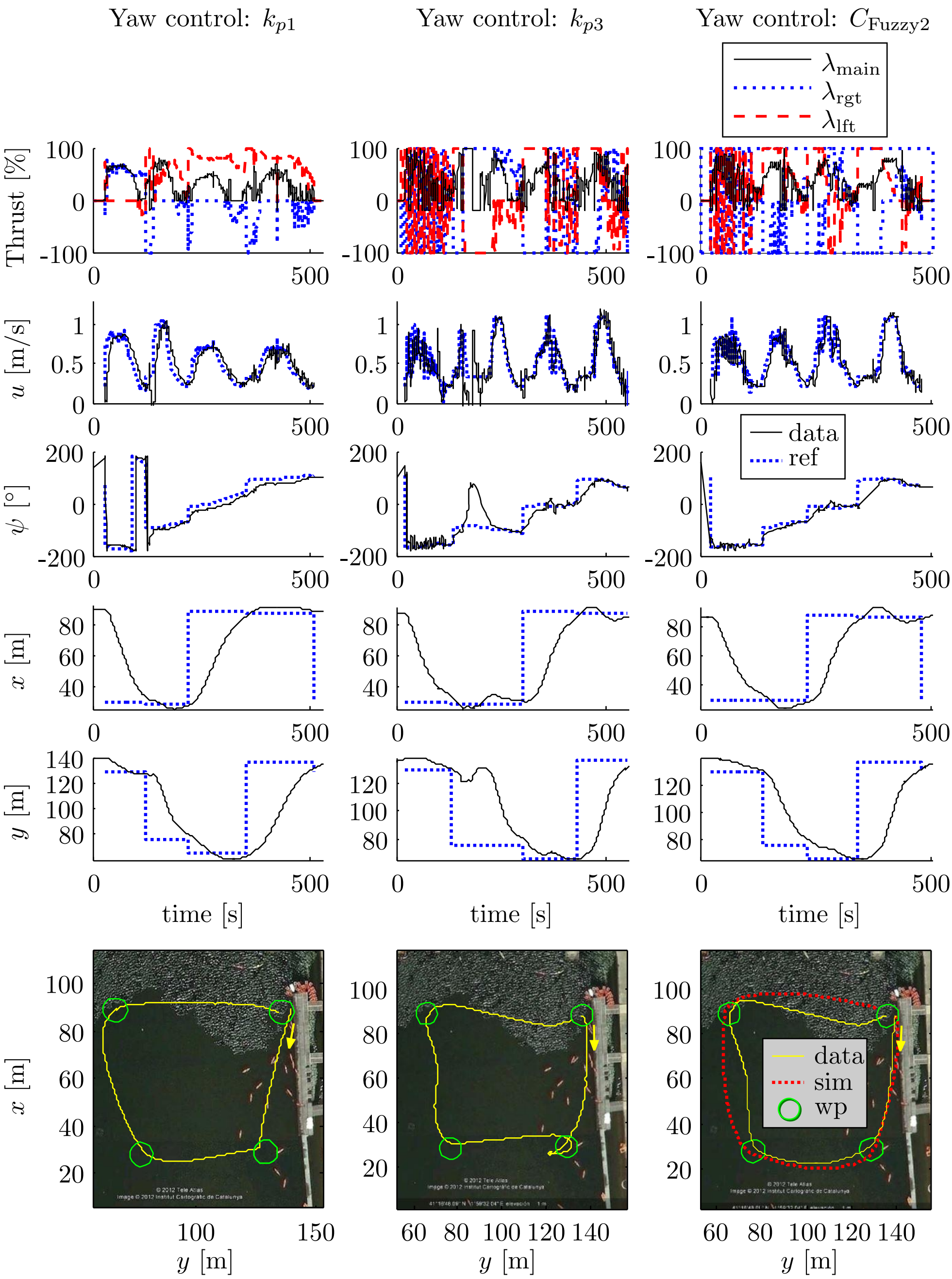


Figure 9.8: Field tests. Outer loop fuzzy control. Comparative of C_{Fuzzy2} with proportional controllers. Path 1. $u = 1\text{m/s}$

9.3 Outer loop: Path following

This section describes the results of a single test using the path following technique.

9.3.1 Inner loop: Fuzzy control 2

Figure 9.9 presents this test. The velocity of the mission is 0.3 m/s. The inner loop to control the velocity is $C^{(u)}_{\text{Fuzzy1}}$ (see section 8.3), and the inner loop to control the yaw is C_{Fuzzy2} (see section 8.6). The experimental results are close to those of the simulation. The vehicle makes a small oscillation while reaching the path, but then follows the path, although with an small distance offset.

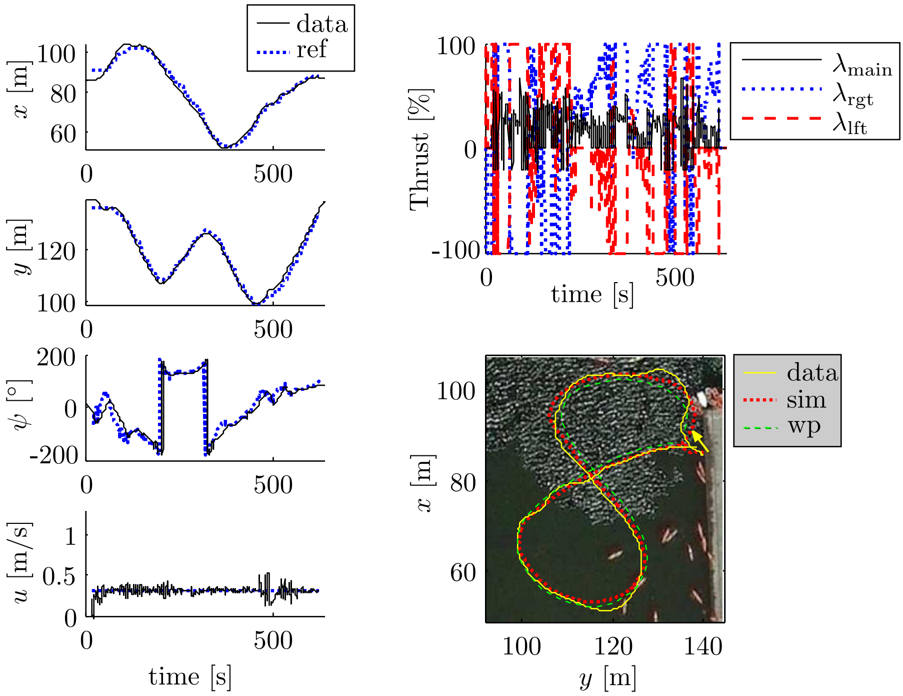


Figure 9.9: Field test using the path following technique. $u = 0.3$m/s

9.4 Final Test

We have made a final test but this time close to the port of Vilanova i La Geltrú. This location is very good because we can test the performance in presence of waves and we can program a very long path for the mission.

The mission consisted in following 13 waypoints using a pure pursuit guidance in a total travel of 3600 m. The radius of acceptance was 6 m. For the inner loop we used the controllers $C^{(u)}_{\text{Fuzzy1}}$ and C_{Fuzzy2} (see sections 8.3 and 8.6), and the outer loop used the fuzzy controller (see section 8.8). Additionally, we scheduled 3 vertical dives up to 5 m of depth, concretely, at waypoints 2, 6, and 10. Figure 9.10 shows the results of this test. It took about 70 minutes to complete the path, travelling at 1 m/s.

The vehicle accomplished the mission, performing all the waypoints and dives. However, the compass was a little bit out of calibration, and this caused some deviations in the path. Regarding the dives we can see in this figure three periods of time of about 2 minutes where the thrusters were stopped and the forward velocity was zero. During these intervals, only the cylinder acts to accomplish the vertical movement.

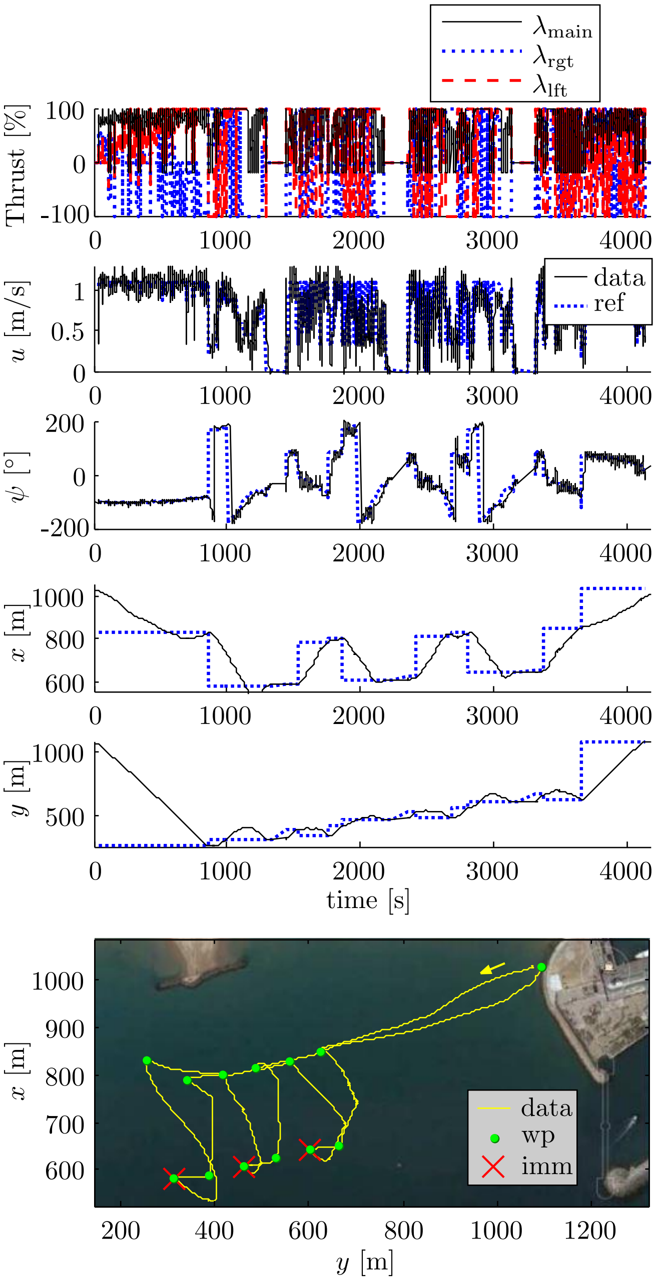


Figure 9.10: Final test. Navigation data

Figures 9.11 and 9.12 show the record of salinity and the temperature using the CTD. They show both the data at the surface and underwater at the dive points. The salinity varies from 37.68 to 37.97 PSU, and it also has variations underwater. The temperature varies from 18.47 to 19.16 °C. It is clear that the temperature decreases when descending.

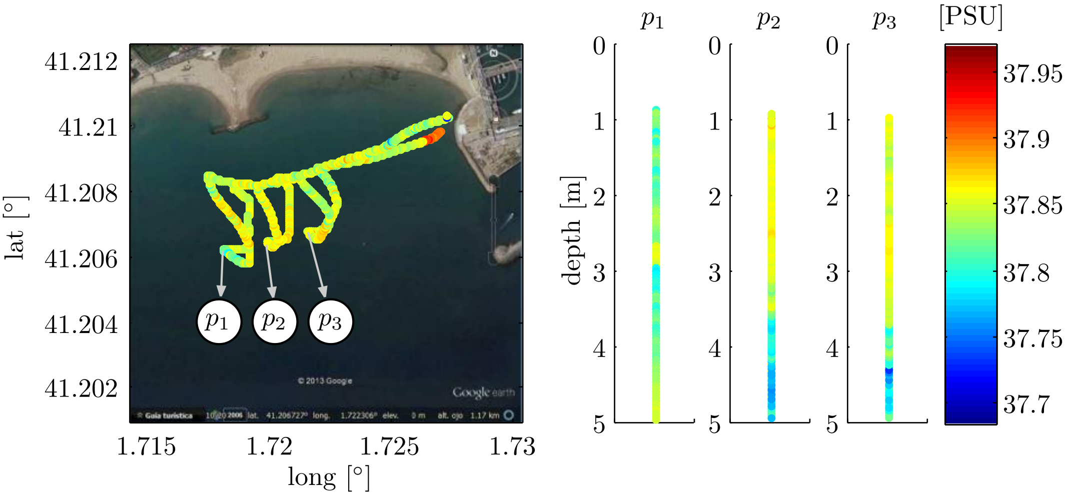


Figure 9.11: Final test. Salinity record

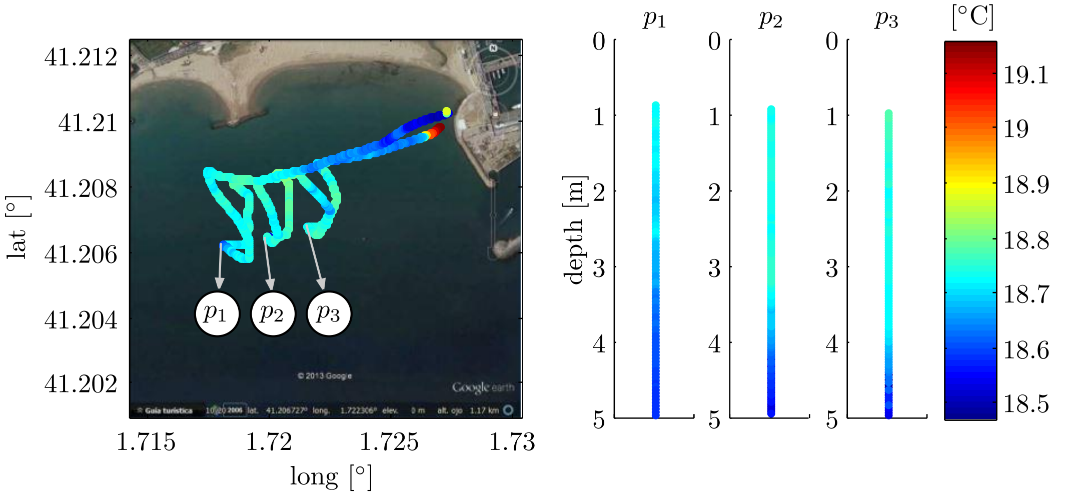


Figure 9.12: Final test. Temperature record

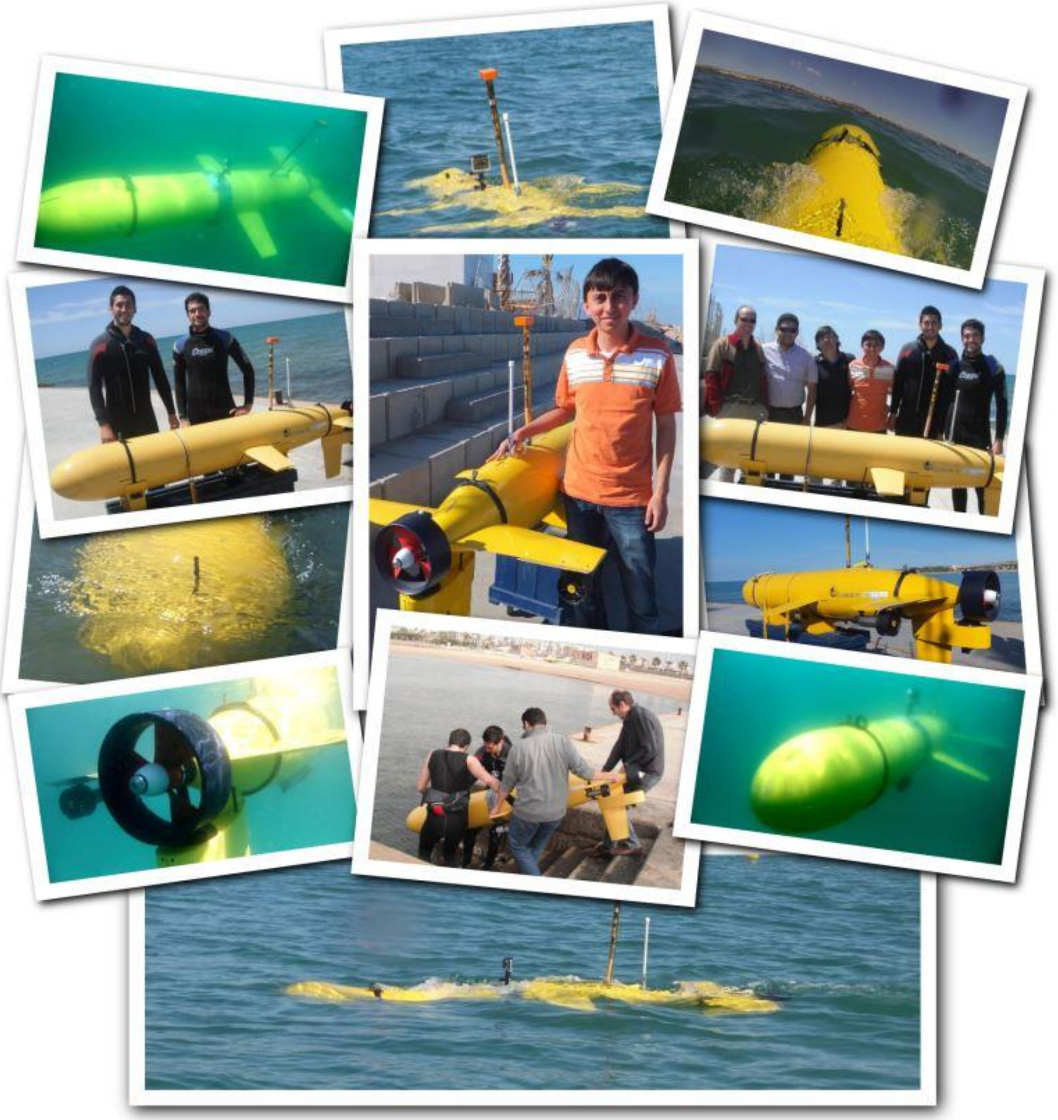


Figure 9.13: Photos during the final test off the coast of Vilanova i La Geltrú

9.5 Conclusions

In this chapter we have tested experimentally the performance of the controllers designed in the chapter 8 both for the inner and outer loop.

Several tests were performed using the pure pursuit guidance 1 (by radius of curvature) as outer loop. With respect to the inner loop, we used first the controller C_{Fuzzy1}, where a deficient use of the thrusters was observed, as they saturated and oscillated very quickly. The actual path had a large radius of curvature, since the outer loop tried to maintain the forward velocity constant. Next we showed some tests comparing the controllers C_{Fuzzy1} and $C^{(u)}_{\text{GainS}}$ for the inner loop. The performance of both was very similar, but again they exhibited saturation in the thrusters and large radius of curvature. Next, using the same outer loop controller, we showed a comparative of two linear controllers and the controller C_{Fuzzy2} to control the yaw (inner loop); here the importance of the fuzzy controller was manifested because it takes the best of the linear controllers. However, the paths still had large radius of curvature.

Tests continued using the pure pursuit guidance 2 (fuzzy control) as the outer loop. This controller has the advantage of setting the desired velocity in a better way. With respect to the inner loop we made again a comparative between two linear controllers and the controller C_{Fuzzy2}. In this case, we saw a great improvement when making the turns to go to the next waypoint. With respect to the power used by the thrusters, we saw that the C_{Fuzzy2} was the best option between the linear controllers due its tuning properties. The velocity and yaw references were followed very well.

Next we presented the results of the path following technique, using the controllers $C^{(u)}_{\text{Fuzzy1}}$ (for the forward velocity) and C_{Fuzzy2} (for the yaw) in the inner loop. The performance was very good, and the vehicle reached the path with small oscillations and then followed it. The experimental results were close to those of the simulation.

Finally, we made a test at the sea close to the port of Vilanova. This test consisted in to follow 13 waypoints and to make 3 immersions, getting a total travel of 3.6 km. For the outer loop we used the pure pursuit guidance 1, and for the inner loop we used the controllers $C^{(u)}_{\text{Fuzzy1}}$ and C_{Fuzzy2}. The results were very satisfactory. It took about 70 minutes to complete the path, navigating at 1 m/s.

With these results we conclude that the Guanay II AUV has a good controller to perform autonomous navigation. It can be configured for pure pursuit or path following, and the inner loop controls very well the yaw and the forward velocity.

Chapter 10

Conclusions and future work

This chapter contains the conclusions of the work presented throughout this thesis, which was focused on the development of the Guanay II AUV, as well as some suggestions about lines of improvement for the future.

First, it was necessary to devise a new driver for the lateral thrusters in order to enable them to thrust backwards. This driver consisted of a PIC16F1508 microcontroller to establish the communication with the PC104, and a DRV8432 motor drive to control the thrusters. Additionally, upgrades in the handling of the vehicle had to devised. In this respect, a wireless on/off system was presented to power the vehicle, and a WiFi connection was adapted to remotely manipulate the software of the vehicle. The contributions to the hardware continued with the immersion system, where the pressure sensor was characterized and an study about the use of a chamber was done. This study included the forces, electric currents and power involved in the immersion, and the maximum operating depth of the vehicle was established. Finally, the improvements in the hardware included the addition of some commercial devices, like an IMU to make measures related to navigation, a CTD to measure the salinity, and an acoustic localization system to locate the vehicle underwater.

The software had to be restructured and improved. First, some problems derived from previous works were debugged and the system was migrated to Linux, yielding a reduction of the starting time by 64%. Next, the system was restructured using multithreading, which provided robustness and modularity. Additional commands to configure and to control the vehicle form the base station were needed, and a robust protocol communication was implemented with the possibility to execute scripts. Finally, a new and more intuitive graphical user interface was designed.

Foams were added to the vehicle to improve buoyancy. These foams were designed tacking into account the vehicle's geometry, and a ballast system was also included for fine adjustment. This resulted in the possibility of a payload of 4.1 kg.

Several tests in the laboratory, a swimming pool, a channel, and at sea were performed in order to check the performance of the vehicle. The results were very satisfactory, and we concluded that the vehicle was fully operative, both in surface and under water, and that the operator can control, and get information from, the vehicle in a convenient and flexible way.

Design and implementation of navigation control techniques were needed in order to give autonomy to the vehicle, tacking into account its physical characteristics and operational requirements. This problem was addressed first with the calculation of an hydrodynamic model in 3DoF. An extensive study about the selection of the coefficients was performed, using a linearized model. The calculation of the coefficients was done using two approaches:

a geometric one and another one based on least squares techniques applied to experimental data obtained during sea trials. The least squares method gave satisfactory results and the simulations fitted the experimental data. The resulting hydrodynamic model was completed with the physical constraints of the actuators of the vehicle.

Solving the autonomous navigation problem required the design of controllers for both inner loop (dynamic) and the outer loop (kinematic). With respect to the inner loop, two solutions were presented for velocity control and yaw control based on type-1 TSK fuzzy control. These controllers were used to manage, at a higher level, different linear controllers designed for specific conditions. The hydrodynamic model played an important role in the design of one of these controllers. Additionally, a gain scheduled controller was designed in order to make comparisons. With respect to the outer loop, we presented two solutions for pure pursuit navigation and one for path following. The first one takes into account the radius of curvature of the vehicle and tries to preserve the forward velocity. The second one uses a fuzzy controller based on some variables of the path and the vehicle. The last method, used for path following, consisted in a proportional-integral controller for the yaw reference, and a fuzzy controller for the velocity reference.

All these controllers, both for inner and outer loop, were simulated and tested experimentally in a channel, obtaining satisfactory results. The fuzzy controller was the best option due to its tuning properties. With respect to the power used by the thrusters, we saw that the thrust did not saturate with high oscillations. With respect to the waypoints, the vehicle performed the turns very well.

Finally the autonomous navigation was tested off the coast of Vilanova i La Geltrú, where the vehicle was required to follow 13 waypoints in a travel totaling 3.6 km, including three vertical dives. The vehicle performed correctly, and salinity and temperature data were collected. In this order, we concluded that the Guanay II AUV has a good controller to perform autonomous navigation, and that it can be configured in several ways.

10.1 Future work

The development of a research project always brings about the discovery of new problems, as well as new interesting research projects. As future work there are some lines to continue the development of the vehicle.

At this moment, the Guanay II navigates on the surface and makes vertical immersion using the cylinder. However, this system does not allow to control the ballast or the pitch movement, making it impossible to navigate underwater. Some possible solutions for this type are:

- *Movable control surfaces.* These surfaces can generate drag and lift when the angle of attack is changed, and thus the pitch of the vehicle can be controlled. This is a popular actuator used in several AUVs, like Remus [10], Ecomapper [35], Maya [84], among others.

- *Vertical tunnel thrusters.* Tunnel thrusters increase the vehicle's resistance to forward motion through the water, but this can be mitigated through the tunnel aperture. The Delphin2 is an AUV that uses these actuators [133].

- *Vectored thrusters.* They have the ability to manipulate the direction of the thrust from the engine. The Bluefin-9 is an example of this type of thrust [17].

The most feasible option for the Guanay II is thrust vectoring, as we can include actuators for the lateral thrusters in order to change their direction in the vertical plane, and thus to control the pitch. The cylinder can still be used to perform vertical dives.

Autonomous navigation under water will require a new set of control techniques, and the calculation of the hydrodynamic model in 6DoF will be necessary. The global problem of navigation can be divided into two sub-problems, the autonomous navigation in the horizontal plane, and the autonomous navigation in the vertical plane. The first one is already solved, but the second one opens new research possibilities.

Furthermore, a new system to sense the position under water will be required, since the GPS only works over the surface. Solutions for this positioning could be:

- *Doppler velocity log (DVL).* This devise measures the velocity of the vehicle relative to the water using the Doppler effect for sound waves. Several works in the literature deal with the positioning using these devices [116] [66] [23].

- *GPS/DVL/IMU integration.* Although the present IMU has a considerable sensor noise, there are several integration possibilities with the GPS or the DVL that can be considered to reduce the error [63] [24] [47].

- *Acoustic positioning.* Acoustic signals are widely used in underwater environments due to the low attenuation of sound in water. Similarly to satellite navigation, it can be used to triangulate the position of a vehicle using acoustic signals from different nodes. The techniques more used in this respect include LBL, SBL, and USBL [101] [22] [97].

Finally, for a more secure system, the vehicle needs an obstacle avoidance module, which requires principally the adaptation of a sonar [87] [136]. This device is already purchased, and future developments will include the data acquisition and data processing systems.

10.2 Publications associated to the thesis

10.2.1 Journals

(RIAI'15) J. González, S. Gomáriz, C. Batlle. "Control difuso para el seguimiento de guiñada del AUV Cormorán". Revista Iberoamericana de Automática e Informática Industrial RIAI, 12(2):166-176. April 2015.

(IMEKO'15) I. Masmitjà, J. González, G. Masmitjà, S. Gomáriz, J. del Río Fernández. "Power system of the Guanay II AUV". Acta Imeko. 4(1):35-43. ISSN: 2221-870X. February 2015.

(JMST'14) S. Gomáriz, I. Masmitjà, J. González, G. Masmitjà, J. Prat. "GUANAY-II: an autonomous underwater vehicle for vertical/horizontal sampling". Journal of Marine Science and Technology. 20(1):81-93, Springer Japan 2015.

(IV'14) C. Galarza, I. Masmitjà, J. González, J. Prat, S. Gomáriz. "Configuración de los parámetros del sistema de detección de obstáculos del Guanay II". Instrumentation Viewpoint 17. ISSN 1886-4864. Winter 2014.

(IV'12) J. González, I. Masmitjà, S. Gomáriz. "Ballast system for Guanay II AUV". Instrumentation Viewpoint 13. Winter 2012.

(IV'11) J. González, I. Masmitjà, J. del Río, S. Gomáriz. "Salinity studies in Mar Menor using Guanay II AUV". Instrumentation Viewpoint 12. Winter 2011.

10.2.2 Conferences

(NGCUV'15) J. González, S. Gomáriz, C. Batlle, C. Galarza. "Fuzzy controller for the yaw and velocity control of the Guanay II AUV". IFAC Workshop on Navigation Guidance and Control of Underwater Vehicles. Girona, Spain. April 2015.

(OCEANS'14) I. Masmitjà, J. González, S. Gomáriz. "Buoyancy model for Guanay II AUV". Oceans MTS/IEEE. Taipei, China. April 2014.

(MARTECH'13a) J. González, I. Masmitjà, S. Gomáriz. "Linear control of the yaw and rudder limitations for Cormoran AUV". 5th MARTECH International Workshop On Marine Technology. Girona, Spain. October 2013.

(MARTECH'13b) I. Masmitjà, J. González, S. Gomáriz. "Latest contributions to Guanay II". 5th MARTECH International Workshop On Marine Technology. Girona, Spain. October 2013.

(IMEKO'13) I. Masmitjà, J. González, S. Gomáriz. "Measuring system and power management of the Guanay II AUV". 19th Imeko TC4 and 17th TC4 workshop IWADC. Barcelona, Spain. July 2013.

(OCEANS'13) J. González, I. Masmitjà, S. Gomáriz, C. Batlle, D. Sarriá. "Mathematical model of the Guanay II AUV". Oceans 2013 MTS/IEEE. Bergen, Austria. June 2013.

(AUTOMAR'12) I. Masmitjà, S. Gomáriz, J. González. "Guanay II: Aportaciones incorporadas al AUV". V Jornadas de automática marítima - Automar 2012. Girona, Spain. December 2012.

(I2MTC'12) J. González, A. Benezra, S. Gomáriz, D. Sarriá. "Limitations of linear control for Cormoran - AUV". International Instrumentation and Measurement Technology Conference - I2MTC 2012 IEEE. ISBN: 978-1-4577-1771-0 pags. 1726 - 1729. Graz, Austria. May 2012.

(NGCUV'12) J. González, I. Masmitjà, S. Gomáriz, E. Molino, J. del Río, A. Mànuel, J. Busquets, A. Guerrero, F. López, M. Carreras, D. Ribas, A. Carrera, C. Candela, P. Ridao, J. Sousa, P. Calado, J. Pinto, A. Sousa, R. Martins, D. Borrajo, A. Olaya, B. Garau, I. González, S. Torres, K. Rajan, M. McCann, J. Gilabert. "AUV based multi-vehicle collaboration: Salinity studies in Mar Menor coastal lagoon". IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles. Porto, Portugal. April 2012.

(ROBOT'11) J. González, I. Masmitjà, G. Masmitjà, J. Prat, S. Gomáriz. "Desarrollo de un vehículo de observación oceanográfica autónomo". Robot 2011. ISBN: 978-84-615-6787-4 pags. 295 - 301. Sevilla, Spain. November 2011.

(MARTECH'11) J. González, A. Benezra, S. Gomáriz, A. García. "Hydrodynamic model, simulation and linear control for Cormoran - AUV". Martech 2011. Cadiz, Spain. September 2011.

(OCEANS'11) S. Gomáriz, J. González, A. Arbos, I. Masmitjà, G. Masmitjà, J. Prat. "Design and construction of the Guanay-II Autonomous Underwater Vehicle". Oceans 2011 IEEE/OES. Santander, Spain. June 2011.

(AUTOMAR'10a) J. González, A. Benezra, S. Gomáriz, A. Lázaro. "Linear control design for a path planning of AUV-Cormoran". IV Jornadas de automática marítima - Automar 2010. Cartagena, Murcia. October 2010.

(AUTOMAR'10b) I.Masmitjà, G. Masmitjà, J. González, S. Shariat-Panahi, S. Gomáriz. "Desarrollo de un sistema electrónico de control para el AUV-Guanay II". IV Jornadas de automática marítima - Automar 2010. Cartagena, Murcia. October 2010.

(AUV'10) I.Masmitjà, G. Masmitjà, J. González, S. Shariat-Panahi, S. Gomáriz. "Development of a Control System for an Autonomous Underwater Vehicle". Autonomous Underwater Vehicles - AUV2010. Monterey, California, EEUU. September 2010.

Appendices

Appendix A

PIC code

This appendix describes the code used for the PIC16F1508. It has three files: "*functions.h*" where some pins and functions names are defined, "*functions.c*" which contains the functions, including the pin configuration for the use of PWM and UART standards, and the main program, "*main.c*", which sets the PWMs for the thrusters and defines the communication protocol. All this code was programmed and compiled using MPLABX.

A.1 main.c

```
#include <pic16f1508.h>
#include <htc.h>
#include "functions.h"

__CONFIG(WDTE_OFF);

double dutty;
int dutty2;

// variables for uart
int indexBufferRX;
int isReceivedMessage;
char bufferRX[SIZE_STRING];
char message[SIZE_STRING];

void main(void) {
    initPIC();
    setPWM1(0.5);
    setPWM2(0.5);
    while(1){
        if(readMessage("M1",&dutty2)==1){
            dutty = (double)(dutty2+100.0)/200;
            setPWM1(dutty);
            sendOK();
        }
        if(readMessage("M2",&dutty2)==1){
            dutty = (double)(dutty2+100.0)/200;
            setPWM2(dutty);
            sendOK();
        }
    }
}

void interrupt ist(){
    char aux;
    if (RCIF==1){
        aux = RCREG;
        if(aux==13 || aux==10){
            if(indexBufferRX>0) isReceivedMessage = 1;
            indexBufferRX = 0;
        }else{
            isReceivedMessage = 0;
            bufferRX[indexBufferRX] = aux;
            indexBufferRX++;
            if (indexBufferRX==SIZE_STRING) indexBufferRX--;
            bufferRX[indexBufferRX] = 0x00;
        }
    }
}
```

A.2 functions.c

```
#include "functions.h"
#include <pic16f1508.h>
#include <stdlib.h>
#include <string.h>

/* Extern variables
 */
extern int indexBufferRX;
extern int isReceivedMessage;
extern char bufferRX[SIZE_STRING];
extern char message[SIZE_STRING];

void initPIC(void){
    // Enable pins
    PIN_ENABLE_PWM1 = 0;
    PIN_ENABLE_PWM2 = 0;
    TRIS_PIN_ENABLE_PWM1 = 0; //pin as output
    TRIS_PIN_ENABLE_PWM2 = 0; //pin as output
    ANSC6 = 0; //not analog

    // Internal oscillator frequency: 8MHz = 1110(pic16f1508)
    IRCF3 = 1;
    IRCF2 = 1;
    IRCF1 = 1;
    IRCF0 = 0;

    // configuration of PWMs (freq:50kHz)
    PR2 = 39; //4*(PR2+1) = posible values of PWMxDC
    TMR2IF = 0;
    T2CKPS0 = 0; //preescaler 1:1
    T2CKPS1 = 0;
    TMR2ON = 1; //timer 2 on
    PWM1OE = 1; //enable pwm1 pin
    PWM2OE = 1; //enable pwm2 pin
    TRISC5 = 0; //pin as output for pwm1
    TRISC3 = 0; //pin as output for pwm2
    PWM1EN = 1; //enable pwm1
    PWM2EN = 1; //enable pwm2

    // configuration UART
    ANSB5=0;    //pin RX must be digital I/O
    TRISB5=1;   //set RX as input
    TRISB7=0;   //set TX as output
    BRG16=0;    //9600baud
    SPBRGH=0;   //9600baud
    SPBRGL=51;  //9600baud
    BRGH=1;     //9600baud
    SYNC=0;     //Asynchronous
    TXEN=1;     //Enable TX
    CREN=1;     //Enable RX
    SPEN=1;     //Serial Port Enable
    GIE=1;      //Enable global interruptions
    PEIE=1;     //Enable peripheral interrupts
    RCIE=1;     //Enable receive interrupt
}

void setPWM1(double dutty){
    int regPWMdutty = (int)(dutty*4*(PR2+1));
    PWM1DCH = regPWMdutty>>2;
    PWM1DCL = (regPWMdutty & 0x03)<<6;

    if(dutty == 0.5){
        PIN_ENABLE_PWM1 = 0;
    }else{
        PIN_ENABLE_PWM1 = 1;
    }
}

void setPWM2(double dutty){
    int regPWMdutty = (int)(dutty*4*(PR2+1));
    PWM2DCH = regPWMdutty>>2;
    PWM2DCL = (regPWMdutty & 0x03)<<6;
    if(dutty == 0.5){
        PIN_ENABLE_PWM2 = 0;
    }else{
        PIN_ENABLE_PWM2 = 1;
    }
}

int readMessage(const char *header, int *val){
    int error=0;
    int i;
    if(isReceivedMessage==0) return 0;

    strcpy(message,bufferRX);
    if(strcmp(strtok(message,","),header) == 0){
        strcpy(message,strtok(NULL,","));
        //checking if was received a number
        if((message[0]>=48 && message[0]<=57)||message[0]==45){
            for(i=1;i<SIZE_STRING && message[i]!=0;i++){
                if(message[i]<48 || message[i]>57) error=1;
            }
            if(i==1 && message[0]==45) error=1;
        }
        //number reading
        if(error==0){
            *val = atoi(message);
            if(*val>=-100 && *val<=100){
                indexBufferRX = 0;
                bufferRX[0] = 0x00;
                isReceivedMessage = 0;
                return 1;
            }
        }
    }
    return 0;
}

void sendAscii(int a){
    while(TXIF==0);
    TXREG = a;
    TXIF = 0;
    while(TRMT==0);
}

void sendMessage(const char *message){
   for(int j=0;j<SIZE_STRING;j++){
       if(*(message+j)==0) break;
       sendAscii(*(message+j));
   }
```

```
}

void sendOK(){
    sendMessage("OK");
    sendAscii(13);
    sendAscii(10);
}
```

A.3 functions.h

```
#define SIZE_STRING 20
#define PIN_ENABLE_PWM1 RC4
#define PIN_ENABLE_PWM2 RC6
#define TRIS_PIN_ENABLE_PWM1 TRISC4
#define TRIS_PIN_ENABLE_PWM2 TRISC6

/* initPIC()
 * It configures PWMs and UART
 */
void initPIC(void);

/* setPWM1(double dutty)
 * set the PWM1, where dutty is a float number between 0 and 1
 */
void setPWM1(double dutty);

/* setPWM2(float dutty)
 * set the PWM2, where dutty is a float number between 0 and 1
 */
void setPWM2(double dutty);

int readMessage(const char *header, int *val);
void sendAscii(int a);
void sendMessage(const char *message);
void sendOK();
```

Appendix B

Communication protocol

The communication protocol consists of a series of commands used to configure and control the vehicle. The format of the frames is as follows:

Type of command	:	Instruction	EOL
3 bytes	1 byte	n bytes	2 bytes

where the type of command can be `SET`, `GET`, `MAN`, `IMM`, `AUT`, `RUN`, or `CMD`. The commands finish with EOL (end of line), which consists of a CR (`0x0D`) and a LF (`0x0A`).

All commands must be used with uppercase letters, and all of them provide a response. Generally the response consists of the type of command with the word `ACK`. For instance, the response for `AUT:START` is `AUTACK`. A the description of all commands is provided next.

B.1 Manual control

Manual control includes the manual operation of the thrusters and cylinder.

`MAN:%f_%f_%f`. This command sets the propulsion of the three thrusters. The first argument sets the main thruster, the second sets the right thruster, and the third sets the left thruster. All values must be in the range $[-100, 100]$. The response consists of a frame with the main sensors data:

MANAUV-x_1//x_2//SEN_x_3_x_4_x_5//VEL x_6//BAT x_7//IMM x_8

where

x_1: Compass frame
x_2: GPS frame
x_3: Pressure
x_4: Humidity
x_5: Max depth reached
x_6: Forward velocity
x_7: Battery frame
x_8: Immersion Boolean values

The immersion Boolean values are 3 bytes long. The first shows the state of the immersion. The second and the third show the state of the magnetic sensors for full cylinder and empty cylinder respectively.

`IMM:%f_%f`. This command starts a new immersion. The first argument is the target depth (in meters), and the second is the maximum time underwater. When one of these values is reached the vehicle goes up. The response is `IMMACK`.

`SET:POWERC=%f`. This command sets the power of the cylinder. The value must be in the range $[0, 100]$, but a greater value than 70% is recommended. By default it is 100%. The response is `SETACK`.

`GET:POWERC`. This command returns the value of `POWERC`.

`GET:COMPASS`. This command returns the frame generated by the compass.

`GET:GPS`. This command returns the frame generated by the GPS.

`GET:MAXDEPTHREACHED`. This command returns, in meters, the depth reached during the last immersion.

`GET:CYLINDER`. This command returns the position of the cylinder: "0" means that it is empty, "1" means that it is full, and "0.5" means that none of the two magnetic sensors is true, so it is in the middle.

`GET:PRESSURE`. This command returns, in bars, the measure of the pressure sensor.

B.2 Automatic control

This set of commands provides all the tools to set out an automatic movement.

`SET:INNERCONTROL=%d`. This command sets the type of controller for the inner loop, that is, to control the forward velocity and the yaw. The response is `SETACK`. The following table presents the possible configurations and the pages to consult further details.

`INNERCONTROL`	Velocity control		Yaw control	
0	$C^{(u)}_{\mathrm{GainS}}$	(p. 137)	$C^{(\psi)}_{\mathrm{GainS}}$	(p. 137)
1	$C^{(u)}_{\mathrm{Fuzzy1}}$	(p. 133)	k_{p1}	(p. 140)
2	$C^{(u)}_{\mathrm{Fuzzy1}}$	(p. 133)	k_{p2}	(p. 140)
3	$C^{(u)}_{\mathrm{Fuzzy1}}$	(p. 133)	k_{p3}	(p. 140)
4	$C^{(u)}_{\mathrm{Fuzzy1}}$	(p. 133)	C_{Fuzzy2}	(p. 140)

`GET:INNERCONTROL`. This command returns the type of controllers for the inner loop.

`SET:OUTERCONTROL=%d`. This command sets the type of controller for the outer loop, that is, to control the vehicle position. The response is `SETACK`. The following table presents the possible configurations and the pages to obtain further details.

`OUTERCONTROL`	Controller	
0	fuzzy pure pursuit	(p. 149)
1	path following	(p. 155)

`GET:OUTERCONTROL`. This command returns the type of controllers for the outer loop.

`SET:MAXU=%f`. This command sets the velocity of the mission in m/s, that is, the maximum velocity allowed. By default it is 0.4 m/s. The response is `SETACK`.

`GET:MAXU`. This command returns the velocity of the mission.

`SET:RADIOWP=%f`. This command sets the radius of acceptance for pure pursuit. By default it is 3 m. The response is `SETACK`.

`GET:RADIOWP`. This command returns the radius of acceptance for pure pursuit.

`SET:PFK1=%f`. This command sets K_1 for path following (see equation 8.69). By default it is 0.09. The response is `SETACK`.

`GET:PFK1`. This command returns the value of K_1 for path following.

`SET:PFK2=%f`. This command sets K_2 for path following (see equation 8.69). By default it is 0.001. The response is `SETACK`.

`GET:PFK2`. This command returns the value of K_2 for path following.

`SET:WPFILE=%s`. This command sets the file name where there the waypoints for automatic movement are stored. The system responses `SETACK` when the assignation is correct. By contrast, it responses `SET:FILENOTFOUND` if the file does not exist. In this case the system keeps the last file name.

The file of waypoints has the desired position (longitude and latitude), and the target depth and the time underwater for the submersion after reaching the waypoint. The values are separated by tabulations. The format is as follows:

$$\begin{array}{cccc}
\text{long}_1 & \text{lat}_1 & \text{target\_depth}_1 & \text{time\_underwater}_1 \\
\text{long}_2 & \text{lat}_2 & \text{target\_depth}_2 & \text{time\_underwater}_2 \\
\vdots & \vdots & \vdots & \vdots \\
\text{long}_n & \text{lat}_n & \text{target\_depth}_n & \text{time\_underwater}_n
\end{array}$$

`GET:WPFILE`. This command returns the file name for the waypoints for the automatic mode.

`SET:HOMEFILE=%s`. This command sets the file name where the waypoints to come back home are stored. The system responses `SETACK` when the assignation is correct. By contrast, it responses `SET:FILENOTFOUND` if the file does not exist. In this case the system keeps the last file name. The format is as in `WPFILE`.

`GET:HOMEFILE`. This command returns the file name for the waypoints for come back to home.

`AUT:START`. This command starts the automatic mode. It uses the above parameters and the file `WPFILE` to set the waypoints. The response is `AUTACK`. When the task finishes the vehicle changes to manual mode.

`AUT:STOP`. This command stops the automatic mode and it changes the operation mode to manual. The response is `AUTACK`.

`AUT:GOHOME`. This command starts the automatic mode to come back to home. It uses the file `HOMEFILE` to establish the waypoints. This command is useful after aborting a mission. The response is `AUTACK`.

`GET:GOHOME`. This command returns the state of `GOHOME` using a Boolean value.

`AUT:GET`. This command returns a frame with the principal sensors data:

$$\texttt{AUTAUV-}x_1\texttt{//}x_2\texttt{//SEN\_}x_3\texttt{\_}x_4\texttt{\_}x_5\texttt{//VEL }x_6\texttt{//BAT }x_7\texttt{//IMM }x_8$$

where

x_1: Compass frame
x_2: GPS frame
x_3: Pressure
x_4: Humidity
x_5: Max depth reached
x_6: Forward velocity
x_7: Battery frame
x_8: Immersion Boolean values

The immersion Boolean values are 3 bytes. The first shows the state of the immersion. The second and the third show the state of the magnetic sensors for full cylinder and empty cylinder respectively.

`AUT:INFO`. This command returns a frame with data about the errors, as follows:

$$x_1, x_2, x_3, x_4, x_5$$

where

x_1: Sensor error
x_2: GPS error
x_3: Compass error
x_4: Base station error
x_5: Index of waypoint

`GET:NWP`. This command returns the total number of waypoints stored in the actual file for automatic mode.

`GET:WPID`. This command returns the ID of the present waypoint. If the vehicle is in manual mode the command returns -1.

B.3 Safety configuration

This set of commands sets important parameters regarding the stopping of the thrusters or their limitation.

`SET:DEPTHSURFACE=%f`. This command sets the depth (in meters) where the vehicle is still considered to be at the surface. If the vehicle is at a depth higher than this value the thrusters are stopped for safety reasons. The response is `SETACK`. By default it is 0.5 m.

`GET:DEPTHSURFACE`. This command returns the maximum depth at which the vehicle is considered to still be at the surface.

`SET:MAXMTH=%f`. This command sets the maximum value for the main thruster, which must be a value in the range $[-100, 100]$. By default it is 100%. Changing this value affects the upper limit of the thruster both in manual and automatic mode. The response is `SETACK`.

`GET:MAXMTH`. This command returns the maximum value for the main thruster.

`SET:MINMTH=%f`. This command sets the minimum value for the main thruster, which must be a value in the range $[-100, 100]$. By default it is -20%. Changing this value affects the lower limit of the thruster both in manual and automatic mode. The response is `SETACK`.

`GET:MINMTH`. This command returns the minimum value for the main thruster.

`SET:MAXTAUTGET=%d`. When the vehicle is in automatic mode the operator can send the command `AUT:GET` to ask for the actual position and state of the vehicle. As a safety feature it is established that if the vehicle does not receive `AUT:GET` the thrusters will be stopped because it implies that there is no communication. The value `MAXTAUTGET` defines, in seconds, how long to wait before stopping the thrusters. By default it is 300 s. The response is `SETACK`.

If the mission (in automatic mode) is one hundred meters away from the base station `MAXTAUTGET` must be set at a very large value in order to not interfere with the mission when the RF communication fails.

`GET:MAXTAUTGET`. This command returns the time before stopping the thrusters when the vehicle does not receive the command `AUT:GET` from the base station in automatic mode.

`SET:MAXTCOMPASS=%d`. This command sets the time in seconds before stopping the thrusters when there is an error with the compass. This is a safety feature and it works both in manual and automatic mode. By default it is 8 s. The response is `SETACK`.

`GET:MAXTCOMPASS`. This command returns the time before stopping the thrusters when there is an error with the compass.

`SET:MAXTGPS=%d`. This command sets the time in seconds before stopping the thrusters when there is an error with the GPS. This is a safety feature and it works both in manual and automatic mode. By default it is 8 s. The response is `SETACK`.

`GET:MAXTGPS`. This command returns the time before stopping the thrusters when there is an error with the GPS.

`SET:MAXTMOT=%d`. This command sets the time in seconds before stopping the thrusters when there is no communication with the base station. This is a safety feature which works only in manual mode. By default it is 15 s. The response is `SETACK`.

`GET:MAXTMOT`. This command returns the time before stopping the thrusters when the vehicle does not receive the command `MAN:%f_%f_%f` from the base station in manual mode.

`SET:NOP=%d`. As a safety feature every t seconds the vehicle sends the word `NOP`. The reception of this word means that the vehicle is on, that it started the software in LabVIEW and that the communication works correctly. By default $t = 3$ s, and using this command the operator can change the frequency of transmission. The response is `SETACK`.

`GET:NOP`. This command returns the period for transmission of the word `NOP`.

`SET:TCONAUTGET=%d`. This configuration is similar to the above. There is a command in automatic mode, `AUT:GET`, to ask for the actual state of the vehicle (position, heading, etc). To avoid having to ask this every time the `TCONAUTGET` parameter allows the sending of these data every t seconds continuously. The response is `SETACK`.

GET:TCONAUTGET. This command returns the period of transmission of the sensors data in automatic mode.

GET:BATTERY. This command returns the frame generated by the battery module.

GET:HUMIDITY. This command returns the internal humidity of the vehicle.

GET:TEMPERATURE. This command returns the internal temperature of the vehicle.

GET:VOLTAGE. This command returns the voltage of the power module.

GET:CURRENT. This command returns the current of the power module.

CMD:%s. This is an special command to communicate directly with the console of the operating system. This is very useful when the WiFi connection fails (maybe due to distance) and we want to get some information or to do some configuration. For instance, `CMD: cat file.txt` returns the data stored in a file.

B.4 Scripts

The scripts are very useful tools which simplify the work of the operator. The idea is to have a set of instructions saved in a file in the vehicle. The operator only has to run the file and the commands are then executed sequentially.

SET:SCRIPTFILE=%s. This command sets the name of the file where the instructions of the script are stored. The system responses SETACK when the assignation is correct. By contrast, it responses SET:FILENOTFOUND if the file does not exist. In this case the system keeps the last file name.

GET:SCRIPTFILE. This command returns the name of the script file.

SET:SLEEP=%d. This commands sleeps the script thread during a time specified in seconds. This is useful to wait before the execution of the next instruction. The response is SETACK.

GET:SLEEP. This command returns the remainder time to finish the SLEEP instruction.

SET:WAITAUT=%d. Similar to the previous, this commands sleeps the script thread during an specific time, but it is aborted when an automatic mode finishes. The response is SETACK. This command must be used in conjunction with the automatic mode, for instance:

```
...               //configuration
AUT:START         //start the mission
SET:WAITAUT=3600 //wait at least 1 hour
AUT:GOHOME        //go to home
```

GET:WAITAUT. This command returns the remainder time to finish the WAITAUT instruction.

SET:WAITIMM=%d. Similar to the above, this command sleeps the script thread during a specific time, but it is aborted when a immersion finishes. The response is SETACK. This command must be used in conjunction with the immersion command, for instance:

```
IMM:20_200        //diving to 20m
SET:WAITIMM=200  //wait at least 200s
IMM:10_200        //diving to 10m
```

For this example notice the importance of `WAITIMM`. The absence of the second command would mean that the vehicle was set to dive to 20 m, but the target would be immediately changed to 10 m.

`GET:WAITIMM`. This command returns the remainder time to finish the `WAITIMM` instruction.

`RUN:START`. This command starts the script. The response is `RUNACK`. Additionally, it is possible to define the file name of the script and run it, for instance: `RUN:my_script.txt`.

`RUN:KILL`. This command stops the actual script. The response is `RUNACK`.

`GET:IDSCRIPT`. This command returns the number of the actual instruction of the script. If the vehicle is not in script mode the command returns -1.

B.5 Emulation

The emulation consists in imitating of the behavior of the sensors. This is used for debugging. For instance, it is possible to imitate an immersion of 20 m to test some control or script without making a real dive.

`SET:EYAW=%f`. This command sets the yaw angle (in degrees) of the vehicle in emulation mode. The response is `SETACK`.

`GET:EYAW`. This command returns the yaw angle in emulation mode.

`SET:ELAT=%f`. This command sets the latitude (in degrees) of the vehicle in emulation mode. The response is `SETACK`.

`GET:ELAT`. This command returns the latitude in emulation mode.

`SET:ELONG=%f`. This command sets the longitude (in degrees) of the vehicle in emulation mode. The response is `SETACK`.

`GET:ELONG`. This command returns the longitude in emulation mode.

`SET:EDEPTH=%f`. This command sets the vehicle depth in emulation mode, in meters. The response is `SETACK`.

`GET:EDEPTH`. This command returns the depth in emulation mode.

`SET:EEMPTY=%d`. This command sets the magnetic sensor of the cylinder in emulation mode in order to be empty. The argument is Boolean, 0 or 1. The response is `SETACK`.

`GET:EEMPTY`. This command returns the magnetic sensor for empty in emulation mode.

`SET:EFULL=%d`. This command sets the magnetic sensor of the cylinder in emulation mode in order to be full. The argument is Boolean, 0 or 1. The response is `SETACK`.

`GET:EFULL`. This command returns the magnetic sensor for full in emulation mode.

`SET:EMUL=%d`. This command starts or stops the emulation mode. The argument is Boolean, 0 or 1. The response is `SETACK`. When this mode is on four sensors are emulated, and therefore the real sensors are not taken into account:

-Compass: It generates a frame with yaw angle `EYAW`, roll 0°, pitch 0°.

-GPS: It generates a frame with latitude ELAT, longitude ELONG, and 12 satellites.

-Depth: The depth is taken from EDEPTH.

-Cylinder: The ends of the cylinder are taken from EEMPTY and EFULL.

GET:EMUL. This command returns the state of the emulation mode.

Bibliography

[1] M. A. Abkowitz. Lectures on ship hydrodynamics - steering and manoeuvrability. 1964.

[2] A.P. Aguiar and J.P. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *Automatic Control, IEEE Transactions on*, 52(8):1362 –1379, aug. 2007.

[3] A Alessandri, M. Caccia, G. Indiveri, and G. Veruggio. Application of ls and ekf techniques to the identification of underwater vehicles. In *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1084–1088 vol.2, Sep 1998.

[4] A. Alvarez and G. Olivar. Cormorán, development of a new mobile and autonomous ocean observation platform. *Instrument Viewpoint*, (2):4, Spring 2004.

[5] J. Alves, P. Oliveira, R. Oliveira, A. Pascoal, M. Rufino, L. Sebastiao, and C. Silvestre. Vehicle and mission control of the delfim autonomous surface craft. In *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on*, pages 1–6, Jun 2006.

[6] G. Antonelli. An adaptive law for guidance and control of remotely operated vehicles. In *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on*, pages 1–6, June 2006.

[7] G. Antonelli, F. Caccavale, S. Chiaverini, and G. Fusco. A novel adaptive control law for underwater vehicles. *Control Systems Technology, IEEE Transactions on*, 11(2):221–232, Mar 2003.

[8] G. Antonelli, S. Chiaverini, N. Sarkar, and M. West. Adaptive control of an autonomous underwater vehicle: experimental results on odin. *Control Systems Technology, IEEE Transactions on*, 9(5):756 –765, sep 2001.

[9] G. Antonelli, T. I. Fossen, and D. R. Yoerger. Underwater robotics. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 987–1008. Springer Berlin Heidelberg, 2008.

[10] Autonomous underwater vehicle - remus 100. Taken from Kongsberg Maritime: http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/D241A2C835DF40B0C12574AB003EA6AB?OpenDocument. Accessed June 2015.

[11] Pedro T. M. Batista. *Sensor-based Navigation and Control of Autonomous Vehicles.* PhD thesis, Jun 2010.

[12] María Andreína Benezra. Diseño de estrategias de guía y control de navegación de un vehículo autónomo submarino. Bachelor's thesis, Universidad Simón Bolívar, Nov 2010.

[13] M Berger, A Camps, J Font, Y Kerr, J Miller, J Johannessen, J Boutin, MR Drinkwater, Niels Skou, N Floury, et al. Measuring ocean salinity with esa's smos mission. *ESA Bull*, 111(113f):113–121, 2002.

[14] Volker Bertram. *Skibsteknisk Selskab, Copenhagen, Denmark*, 2008.

[15] Robert D. Blevins. *Formulas for Natural Frequency and Mode Shape*. Kreiger Publishing, 1979.

[16] D. Richard Blidberg. The development of autonomous underwater vehicles (auv): A brief summary. *Woods Hole Oceanographic Institution*, 2003.

[17] Bluefin-9 auv. Taken from Bluefin Robotics: http://www.bluefinrobotics.com/products/bluefin-9/. Accessed June 2015.

[18] Morten Breivik and Thor I. Fossen. Principles of guidance-based path following in 2d and 3d. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 627 – 634, dec. 2005.

[19] Morten Breivik and Thor I. Fossen. A unified concept for controlling a marine surface vessel through the entire speed envelope. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 1518 –1523, june 2005.

[20] Morten Breivik and Thor I. Fossen. Guidance laws for autonomous underwater vehicles. In Alexander V. Inzartsev, editor, *Underwater vehicles*. 2009.

[21] R. W. Brockett. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*.

[22] Andrea Caiti, Francesco Di Corato, Davide Fenucci, Benedetto Allotta, Riccardo Costanzi, Niccolo Monni, Luca Pugi, and Alessandro Ridolfi. Experimental results with a mixed usbl/lbl system for auv navigation. In *Underwater Communications and Networking (UComms), 2014*, pages 1–4. IEEE, 2014.

[23] Jianhua Cheng, Daidai Chen, Rene Landry Jr, Lin Zhao, and Dongxue Guan. Research on wavelet singularity detection based fault-tolerant federated filtering algorithm for ins/gps/dvl integrated navigation system. *Journal of Applied Mathematics*, 2014, 2014.

[24] Won-Suck Choi, Nhat-Minh Hoang, Jae-Hoon Jung, and Jang-Myung Lee. Navigation system development of the underwater vehicles using the gps/ins sensor fusion. In *Intelligent Robotics and Applications*, pages 491–497. Springer, 2014.

[25] Robert D Christ and Robert L. Wernli Sr. *The ROV Manual: A User Guide for Observation Class Remotely Operated Vehicles*. Butterworth-Heinemann, 2007.

[26] Juan C. Cutipa and Décio Donha. Auv identification and robust control. In *18th IFAC World Congress*, 2011.

[27] K. S. M. Davidson and L. I. Schiff. Turning and course keeping qualities. Technical report, 1946.

[28] Dg14 receiver - advanced, high-precision gps technology for highly demanding applications. http://www.navtechgps.com/ashtech_dg14_oem_receiver/. Accessed June 2015.

[29] K.D. Do and J. Pan. Robust and adaptive path following for underactuated autonomous underwater vehicles. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 1994 – 1999 vol.3, june 2003.

[30] Khac Duc Do and Jie Pan. *Control of Ships and Underwater Vehicles*. Springer, 2009.

[31] Tao Dong, X.H. Liao, R. Zhang, Zhao Sun, and Y.D. Song. Path tracking and obstacles avoidance of uavs - fuzzy logic approach. In *Fuzzy Systems, 2005. FUZZ '05. The 14th IEEE International Conference on*, pages 43 –48, may 2005.

[32] Dimiter Driankov, Hans Hellendoorn, and Michael Reinfrank. *An Introduction to Fuzzy Control*. Springer, 1996.

[33] Drv-8432 12a dual brushed dc or single bipolar stepper motor driver (pwm ctrl). http://www.ti.com/product/drv8432. Accessed June 2015.

[34] Dsor lab - dynamical systems and ocean robotics lab. https://plus.google.com/118043347454060065364/posts/65jUHqzxHKx, 2014. Accessed June 2015.

[35] Ecomapper auv. Taken from YSI Systems: http://www.ysisystems.com/media/pdfs/E50-EcoMapper-AUV-Specs.pdf. Accessed June 2015.

[36] P. Encarnacao and A. Pascoal. 3d path following for autonomous underwater vehicle. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 3, pages 2977 –2982 vol.3, 2000.

[37] P. Encarnacao and A. Pascoal. Combined trajectory tracking and path following: an application to the coordinated control of autonomous marine craft. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 964 –969 vol.1, 2001.

[38] Paula Espinal-García, Alan Giraldo, Mario Londoño-Mesa, and Luz Marina Mejía-Ladino. Variabilidad en la abundancia de larvas de crustáceos y poliquetos en bahía málaga, pacífico colombiano (enero-junio de 2010). *Boletín de Investigaciones Marinas y Costeras-INVEMAR*, 41(2):355–373, 2012.

[39] K.M. Fauske, F. Gustafsson, and O. Hegrenaes. Estimation of auv dynamics for sensor fusion. In *Information Fusion, 2007 10th International Conference on*, pages 1–6, July 2007.

[40] K. K. Fedayevsky and G. V. Sobolev. *Control and stability in ship design*. State Union Shipbuilding Publishing House, USSR, 1963.

[41] R. Fierro and F.L. Lewis. Control of a nonholonomic mobile robot using neural networks. *Neural Networks, IEEE Transactions on*, 9(4):589–600, Jul 1998.

[42] Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons Ltd., England, 1994.

[43] Thor I. Fossen. *Marine Control Systems Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, Trondheim, Norway, 2002.

[44] Thor I. Fossen, Morten Breivik, and Roger Skjetne. Line-of-sight path following of underactuated marine craft. In *6th IFAC MCMC*, pages 244–249, Girona, Spain, 2003.

[45] Ctd profiler / conductivity/ temperature / depth xr-420/620. http://www.nauticexpo.com/prod/rbr-uk/product-50402-392935.html. Accessed June 2015.

[46] Antonio Galo. Diseño de un sistema de control autónomo para un vehículo submarino auv. Bachelor's thesis, Universitat Politècnica de Catalunya, June 2009.

[47] Yanrui Geng, Ricardo Martins, and Joao Sousa. Accuracy analysis of dvl/imu/magnetometer integrated navigation system using different imus in auv. In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pages 516–521. IEEE, 2010.

[48] S. Gomáriz, J. Prat, A. Arbos, O. Pallares, and C. Violo. Autonomous vehicle development for vertical submarine observation. *Instrument Viewpoint*, (8):39–40, Autumn 2009.

[49] S. Gomáriz, J. Prat, A.G. Ruiz, J. Sole, P. Gaya, and J. Del Rio. Development of a low-cost autonomous oceanographic observation vehicle. In *OCEANS 2009 - EUROPE*, may 2009.

[50] S. Gomáriz, J. Soler, P. Gayá, and J. Prat. Development of an autonomous oceanographic observation platform. *Instrument Viewpoint - Martech'07*, (6):55–56, Autumn 2007.

[51] Spartacus Gomáriz, Jordi Prat, P. Gaya, and J. Sole. An autonomous vehicle development for submarine observation. *Journal of Maritime Research*, 6(2):23–36, 2009.

[52] M. Greytak and F. Hover. Underactuated point stabilization using predictive models with application to marine vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3756–3761, Sept 2008.

[53] Hani Hagras. Type-2 fuzzy logic controllers: Towards a new approach for handling uncertainties in real world environments, Oct 2008. IEEE Expert Now multi-media educational course, sponsored by the IEEE Computational Intelligence Society, October, 2008, ISBN: 1-4244-1451-2, Run Time: 1:00:00.

[54] Hir6-433 - rf receiver decoder hybrid. http://www.rfsolutions.co.uk/acatalog/DS010-7%20HIRK.pdf. Accessed June 2015.

[55] Sighard F. Hoerner. *Fluid Dynamic Drag*. Published by author, 1965.

[56] Tiemin Hu, Simon X. Yang, Fangju Wang, and G.S. Mittal. A neural network controller for a nonholonomic mobile robot with unknown robot parameters. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 4, pages 3540–3545 vol.4, 2002.

[57] Humidity sensors - hih-4000. http://www.honeywellscportal.com//product-page?pr_id=53944. Accessed June 2015.

[58] Ig-500a: miniature attitude and heading reference system (ahrs). http://www.sbg-systems.com/products/ig-500a. Accessed June 2015.

[59] Igus drylin w - linear guide for almost unlimited leeway in design. http://www.igus.eu/wpck/1994/drylin_w. Accessed June 2015.

[60] Tor Marius Jensen. Waypoint-following guidance based on feasibility algorithms. Master of science in engineering cybernetics, Norwegian University of Science and Technology, Jun 2011.

[61] Robert John and Simon Coupland. Extensions to type-1 fuzzy logic: Type-2 fuzzy logic and uncertainty. *Computational Intelligence: Principles and Practice*, pages 89–102, 2006.

[62] Sung Woo Jun, Do Wan Kim, and Ho Jae Lee. Design of t-s fuzzy-model-based diving control of autonomous underwater vehicles: Line of sight guidance approach. In *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*, pages 2071–2073, Oct 2012.

[63] Tashfeen Karamat. Improved land vehicle navigation and gps integer ambiguity resolution using enhanced reduced-imu/gps integration. 2014.

[64] Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *J. Field Robot.*, 29(2):315–378, March 2012.

[65] T.W. Kim and J. Yuh. A novel neuro-fuzzy controller for autonomous underwater vehicles. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2350 – 2355 vol.3, 2001.

[66] Nak Yong Ko and Seokki Jeong. Attitude estimation and dvl based navigation using low-cost mems ahrs for uuvs. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*, pages 605–607. IEEE, 2014.

[67] Nikolai Kornev. *Lectures on ship manoeuvrability.* Faculty of Mechanical Engineering and Marine Technology, Universitat Rostock, 2013.

[68] Thomas R. Krogstad. *Attitude synchronization in spacecraft formations: Theoretical and experimental results.* PhD thesis, Nov 2009.

[69] Rudolf Kruse, Joan E. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[70] Labview system design software. http://www.ni.com/labview/. Accessed June 2015.

[71] L. Lapierre, D. Soetanto, and A. Pascoal. Nonlinear path following with applications to the control of autonomous underwater vehicles. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 1256 – 1261 Vol.2, dec. 2003.

[72] Legacy tcm - 3 axis compass module. http://www.pnicorp.com/products/tcm-legacy. Accessed June 2015.

[73] Library for comedi. libcomedi0 package. https://packages.debian.org/sid/libcomedi0. Accessed June 2015.

[74] Linear actuators - atl series. http://servomech.it/main/linear-actuators.htm. Accessed June 2015.

[75] Shifei Liu, Yanhui Wei, and Yanbin Gao. 3d path planning for auv using fuzzy logic. In *Computer Science and Information Processing (CSIP), 2012 International Conference on*, pages 599–603, Aug 2012.

[76] J.E. Manley. Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, pages 1–4, Sept 2008.

[77] Mantrak manual tracking kit. http://sonotronics.com/?page_id=1070. Accessed June 2015.

[78] Andreu D. Marchese, Cagdas D. Onal, and Daniela Rus. Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. *Soft Robotics*, 1(1), 2014.

[79] Alejandro Martínez. Baiona recibe el robot prodigioso. *La voz de Galicia*, 2009. 10 de diciembre.

[80] V Martinez-Alvarez, B Gallego-Elvira, JF Maestre-Valero, and M Tanguy. Simultaneous solution for water, heat and salt balances in a mediterranean coastal lagoon (mar menor, spain). *Estuarine, Coastal and Shelf Science*, 91(2):250–261, 2011.

[81] Ivan Masmitjà. Aportacions realitzades al vehicle guanay ii auv. Master's thesis, Universitat Politècnica de Catalunya, 2013.

[82] Ivan Masmitjà and Gerard Masmitjà. Desenvolupament d'un sistema de control per a un vehicle autònom submarí. Bachelor's thesis, Universitat Politècnica de Catalunya, July 2010.

[83] Pramod Maurya, Pedro Aguiar, and Antonio Pascoal. Marine vehicle path following using inner-outer loop control. In *Manoeuvring and Control Marine Craft. Proceedings of the 8th IFAC Conference on*, Sep 2009.

[84] Pramod Maurya, E. Desa, A. Pascoal, G. Navelkar, R. Madhan, A. Mascarenhas, S. Prabhudesai, S. Afzulpurkar, A. Gouveai, S. Naroji, and L. Sebastiao. Control of the maya auv in the vertical and horizontal planes: Theory and practical results. In *7th IFAC Conference MCMC2006 (Maneuvering and Control of Marine Craft)*, September 2006.

[85] C. McGann, F. Py, K. Rajan, H. Thomas, R. Henthorn, and R. McEwen. A deliberative architecture for auv control. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1049–1054, May 2008.

[86] A. L. Meyrowitz, D. R. Blidberg, and R. C. Michelson. Autonomous vehicles. In *Proceedings of the IEEE*, volume 84, pages 1147–1164, april 1996.

[87] Gina Millar. An obstacle avoidance system for autonomous underwater vehicles: A reflexive vector field approach utilizing obstacle localization. In *Autonomous Underwater Vehicles (AUV), 2014 IEEE/OES*, pages 1–4. IEEE, 2014.

[88] A. Mittal, P. Jain, T.K. Saxena, and A. Kapoor. Genetic algorithm based neuro fuzzy tuning of pid controller for a nonlinear temperature water bath with feedforward control. In *Advanced Electronic Systems (ICAES), 2013 International Conference on*, pages 34–38, Sept 2013.

[89] Héctor A. Moreno, Roque Saltarén, Lisandro Puglisi, Isela Carrera, Pedro Cárdenas, and César Álvarez. Robótica submarina: Conceptos, elementos, modelado y control. *Revista Iberoamericana de Automática e Informática Industrial {RIAI}*, 11(1):3 – 19, 2014.

[90] Msmx104+ - microspace pc/104 peripheral. http://www.kontron.com/products/boards-and-standard-form-factors/pc-104/microspace-r-pc-104-peripherals/msmx104.html. Accessed June 2015.

[91] D. F. Myring. A theoretical study of body drag in subcritical axisymmetric flow. *Aeronautical Quarterly*, pages 186–194, Agosto 1976.

[92] J. N. Newman. *Marine Hydrodynamics*. MIT, Massachusetts, 1977.

[93] Nickel-cadmium vr 7 fl - saft rechargeable battery systems. http://www.houseofbatteries.com/documents/VR_7_FL.pdf. Accessed June 2015.

[94] M. Nogueras, J. Del Rio, J. Cadena, J. Sorribas, C. Artero, J. Danobeitia, and A. Manuel. Obsea an oceanographic seafloor observatory. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pages 488–492, July 2010.

[95] N. Norrbin. Theory and observations on the use of a mathematical model for ship manoeuvring in deep and conner water. Technical report, Swedish state shipbuilding experimental tank, Gothenburg, 1971.

[96] L. Paull, S. Saeedi, M. Seto, and H. Li. Auv navigation and localization: A review. *Oceanic Engineering, IEEE Journal of*, 39(1):131–149, Jan 2014.

[97] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. Auv navigation and localization: A review. *Oceanic Engineering, IEEE Journal of*, 39(1):131–149, 2014.

[98] Pc104-das16jr/12 - 125 ks/s analog input board with 16 se or 8 di 12-bit inputs. http://www.mccdaq.com/pc104-data-acquisition/PC104-DAS16JR-12.aspx. Accessed June 2015.

[99] Marc Carreras Pérez. *A proposal of a behavior-based control architecture with reinforcement learning for an autonomous underwater robot*. PhD thesis, May 2003.

[100] Tristan Perez. *Ship Motion Control: Course Keeping and Roll Stabilization Using Rudder and Fins*. Springer, London, 2005.

[101] Alex Alcocer Peñ. *Positioning and navigation systems for robotic underwater vehicles*. PhD thesis, Oct 2009.

[102] Pic16f1508 microcontroller. http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en553471. Accessed June 2015.

[103] Pm6100 - pc/104+ with amd geode lx800 processor, vga/ttl, 2 lan, 2 com, lpt, ide, 4 usb. http://www.aewin.com.tw/en/products/p/176/PM-6100. Accessed June 2015.

[104] J. Prat, J. Sol, and P. Gayà. A vehicle design for submarine observation. *Instrument Viewpoint*, (4):20–21, Autumn 2005.

[105] Timothy Prestero. Verification of a six degree of freedom simulation model for the remus autonomous underwater vehicle. Master's thesis, University of California at Davis, September 2001.

[106] Proximity sensors - round design. http://www.festo.com/cat/en-us_us/products__33419. Accessed June 2015.

[107] Quantum - workclass rov. Taken from SMD: http://www.smd.co.uk/products/work-class-rovs/quantum.htm. Accessed June 2015.

[108] G.J.S. Rae and S.M. Smith. A fuzzy rule based docking procedure for autonomous underwater vehicles. In *OCEANS '92. Mastering the Oceans Through Technology. Proceedings.*, volume 2, pages 539–546, Oct 1992.

[109] B.A. Reddy, Y. Srinivas, and E. VenkataRamesh. Analytical structures of gain-scheduled fuzzy pi controllers. In *Industrial Electronics, Control Robotics (IECR), 2010 International Conference on*, pages 122–128, Dec 2010.

[110] D. Ribas, P. Ridao, L. Magi, N. Palomeras, and M. Carreras. The girona 500, a multipurpose autonomous underwater vehicle. In *OCEANS, 2011 IEEE - Spain*, pages 1 –5, june 2011.

[111] David Ribas, Pere Ridao, Juan Domingo Tards, and Jos Neira. Underwater slam in man-made structured environments. *Journal of Field Robotics*, 25(11-12):898–921, 2008.

[112] P. Ridao, A. Tiano, A. El-Fakdi, M. Carreras, and A. Zirilli. On the identification of non-linear models of unmanned underwater vehicles. *Control Engineering Practice*, 12(12):1483 – 1499, 2004. Guidance and control of underwater vehicles.

[113] P. Ridao, J. Yuh, J. Batlle, and K. Sugihara. On auv control architecture. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 2, pages 855–860 vol.2, 2000.

[114] D. Roig, M. Martínez, B. Garau, A. Alvárez, and J. Tintoré. A low-cost autonomous vehicles for coastal sea monitoring. *Instrument Viewpoint*, (4):16–17, Autumn 2005.

[115] M. J. Ross. *Polar Pioneers: John Ross and James Clark Ross.* Mcgill Queens Univ Pr, 1994.

[116] A Rossi, M Pasquali, and M Pastore. Performance analysis of an inertial navigation algorithm with dvl auto-calibration for underwater vehicle. In *Inertial Sensors and Systems Symposium (ISS), 2014 DGON*, pages 1–19. IEEE, 2014.

[117] Router wifi tp-link, model tl-mr3220. http://www.tp-link.com.au/products/details/?model=TL-MR3220. Accessed June 2015.

[118] Rov categories summary. Taken from ROV Marine Technology Society: http://www.rov.org/rov_categories.cfm. Accessed June 2015.

[119] Wilson J. Rugh and Jeff S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401 – 1425, 2000.

[120] A. Sakalli, T. Kumbasar, E. Yesil, and H. Hagras. Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy pid controllers on the magnetic levitation system. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 1859–1866, July 2014.

[121] L.L. Scharf, W.P. Harthill, and P.H. Moose. A comparison of expected flight times for intercept and pure pursuit missiles. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-5(4):672–673, July 1969.

[122] Sensit hps-a series voltage output pressure transducer. http://www.roxspur.com/Product/Sensit-HPS-A-Series-Voltage-Output-Pressure-Transducer. Accessed June 2015.

[123] Xiaocheng Shi, Jiajia Zhou, Xinqian Bian, and Juan Li. Fuzzy sliding-mode controller for the motion of autonomous underwater vehicle. In *Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on*, pages 466 –470, aug. 2008.

[124] Alison Sider. Big oil's new toy: Robotic solar surfboards, prowling the ocean. wave glider. http://blogs.wsj.com/corporate-intelligence/2014/05/09/big-oils-new-toy-robotic-solar-surfboards-prowling-the-ocean/, 2014. Accessed June 2015.

[125] C. Silvestre and A. Pascoal. Depth control for the infante auv using gain-scheduled reduced order output feedback. *Science Direct*, pages 883–895, 2007.

[126] Kwanchai Sinthipsomboon, Issaree Hunsacharoonroj, Josept Khedari, Watcharin Pongaen, and Pornjit Pratumsuwan. A hybrid of fuzzy and fuzzy self-tuning pid controller for servo electro-hydraulic system. In *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pages 220–225. IEEE, 2011.

[127] Sirene project - dsor lab. http://dsor.isr.ist.utl.pt/Projects/Sirene/index.html. Accessed July 2014.

[128] S.M. Smith, G.J.S. Rae, and D.T. Anderson. Applications of fuzzy logic to the control of an autonomous underwater vehicle. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1099 –1106 vol.2, 1993.

[129] The society of Naval Architects and Marine Engineers, 29 West 39th Street, New York 18, N. Y. *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*, Apr 1950.

[130] Feijun Song and S.M. Smith. Design of sliding mode fuzzy controllers for an autonomous underwater vehicle without system model. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 2, pages 835 –840 vol.2, 2000.

[131] Ssc32-servo controller - 32 channels, by lynxmotion. http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx. Accessed June 2015.

[132] Stainless steel cylinders crdng. http://www.festo.com/cat/en-us_us/products_CRZYL. Accessed June 2015.

[133] Leo Steenson, Alexander Phillips, Eric Rogers, Maaten Furlong, and Stephen Turnock. The performance of vertical tunnel thrusters on an autonomous underwater vehicle operating near the free surface in waves. In *Second International Symposium on Marine Propulsors*. Hamburg University of Technology, June 2011.

[134] K.J. Åströ and C.G. Källström. Identification of ship steering dynamics. *Automatica*, 12(1):9 – 22, 1976.

[135] Subconn - underwater pluggable electrical connectors. http://macartney.com/subconn. Accessed June 2015.

[136] Bing Sun, Daqi Zhu, Lisha Jiang, and Simon X Yang. A novel fuzzy control algorithm for three-dimensional auv path planning based on sonar model. *Journal of Intelligent and Fuzzy Systems*, 26(6):2913–2926, 2014.

[137] Robert Sutton and Geoff Roberts. *Advances in Unmanned Marine Vehicles (IEE Control Series)*. Institution of Engineering and Technology, 2006.

[138] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on SMC*, 15(1):116–132, 1985.

[139] The scarlet knight's trans-atlantic challenge. Taken from Rutgers: http://rucool.marine.rutgers.edu/atlantic/. Accessed June 2015.

[140] The science: sounding - hms challenger. Taken from UC San Diego: http://aquarium.ucsd.edu/Education/Learning_Resources/Challenger/science2a.php. Accessed June 2015.

[141] Thruster si-mct01-b. http://www.seaeye.com/thrusters.html. Accessed June 2015.

[142] Thruster btd-150. http://www.seabotix.com/products/auv_thrusters.htm. Accessed June 2015.

[143] A. Tiano, F. Pizzocchero, and P. Venini. A global optimization approach to nonlinear system identification. In *Control and Automation 1999. Proceedings of the 7th Mediterranean Conference on*, Jun 1999.

[144] A. Tiano, R. Sutton, A. Lozowicki, and W. Naeem. Observer kalman filter identification of an autonomous underwater vehicle. *Control Engineering Practice*, 15(6):727 – 739, 2007. Special Section on Control Applications in Marine Systems CAMS2004 Control Applications in Marine Systems.

[145] J Timothy Pennington and Francisco P Chavez. Seasonal fluctuations of temperature, salinity, nitrate, chlorophyll and primary production at station h3/m1 over 1989–1996 in monterey bay, california. *Deep Sea Research Part II: Topical Studies in Oceanography*, 47(5):947–973, 2000.

[146] Tmod c48+ series wireless modem. http://web3.farell-i.com/en/product/2. Accessed June 2015.

[147] Michael S. Triantafyllou and Franz S. Hover. *Maneuvering and control of marine vehicles*. Department of Ocean Engineering. Massachusetts Institute of Technology, Cambridge, Massachusetts USA, 2003.

[148] Deepak Trivedi, Christopher D. Rahn, William M. Kier, and Ian D. Walker. Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, 5(3):99–117, september 2008.

[149] Kuang-Hsuan Tu and J.S. Shamma. Nonlinear gain-scheduled control design using set-valued methods. In *American Control Conference, 1998. Proceedings of the 1998*, volume 2, pages 1195 –1199 vol.2, jun 1998.

[150] F. Valenciaga, P.F. Puleston, O. Calvo, and G.G. Acosta. Trajectory tracking of the cormoran auv based on a pi-mimo approach. In *OCEANS 2007 - Europe*, pages 1 –6, june 2007.

[151] Very low noise mobile gps antenna - 3910. http://www.antenna.com/apg_products.cgi?id_num=11222. Accessed June 2015.

[152] J. Vidal, S. Gomáriz, and A. Mànuel. Autonomous underwater vehicle control. *Instrument Viewpoint*, (4):22–24, Autumn 2005.

[153] Carlos Viñolo and Oriol Pallarés. Disseny d'un sistema electrònic pel control de rumb i de posicionament d'un auv. Bachelor's thesis, Universitat Politècnica de Catalunya, July 2009.

[154] Christopher von Alt. Autonomous underwater vehicles. *Woods Hole Oceanographic Institution*, 2003.

[155] Vrb d-30w series - single output dc-dc converter. http://mornsun-power.com/product/ProductModel.aspx?channelid=18&Seriesnumber=1066. Accessed November 2014.

[156] Sabiha Wadoo and Pushkin Kachroo. *Autonomous Underwater Vehicles: Modeling, control design, and simulation.* Taylor & Francis Group, LLC, USA, 2011.

[157] Web del canal olimpic de catalunya. http://www.canalolimpic.com/. Accessed June 2015.

[158] Wireless usb-wifi tp-link model tl-wn727n. http://www.tp-link.es/products/details/?model=TL-WN727N. Accessed June 2015.

[159] Fredrik Wulff and Robert E Ulanowicz. A comparative anatomy of the baltic sea and chesapeake bay ecosystems. *Coastal and estuarine studies*, 32:232–256, 1989.

[160] D.R. Yoerger, J.G. Cooke, and J.-J.E. Slotine. The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. *Oceanic Engineering, IEEE Journal of*, 15(3):167–178, Jul 1990.

[161] D.R. Yoerger, J.-J.E. Slotine, J. Newman, and H. Schempf. Robust trajectory control of underwater vehicles. In *Unmanned Untethered Submersible Technology, Proceedings of the 1985 4th International Symposium on*, volume 4, pages 184–197, Jun 1985.

[162] Lofti A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[163] Wei Zhang, Hongtao Wang, Xinqian Bian, Zheping Yan, and Guoqing Xia. The application of self-tuning fuzzy pid control method to recovering auv. In *Oceans, 2012*, pages 1–5, Oct 2012.