Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Design and construction of the hull of an autonomous underwater vehicle with biomimetic profile and development of instrumentation and control system

A thesis presented by

Edisson Andrés Naula Duchi

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Sciences

Monterrey, Nuevo León, December, 2019

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

The committee members, hereby, certify that have read the thesis presented by Edisson Andrés Naula Duchi and that it is fully adequate in scope and quality as a partial requirement for the degree of Master of Science in Engineering Sciences.

Dr. Luis Eduardo Garza Castañón
ITESM
Principal Advisor

Dr. José Israel Martínez López
ITESM
Committee Member

Dr. Carlos Gustavo Sotelo Molina
ITESM
Committee Member

Dr. Rubén Morales Menéndez
Associate Dean of Graduate Studies
School of Engineering and Sciences

Monterrey, Nuevo León, December, 2019

Declaration of Authorship

I, Edisson Andrés Naula Duchi, declare that this thesis titled, ”Design and construction of the hull of an autonomous underwater vehicle with biomimetic profile and development of instrumentation and control system” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Edisson Andrés Naula Duchi
Monterrey, Nuevo León, December, 2019

©2019 by Edisson Andrés Naula Duchi
All Rights Reserved

Dedication

I dedicate this thesis to my parents for their advice, values and for being the fundamental pillar in everything I am today. To my brothers for having supported me at all times and always encouraging me to continue. To my teachers and friends, who without their help would never have been able to do this job.

Edisson Andrés

Acknowledgements

I want to thank my parents, siblings and all those who participated, directly or indirectly, in the elaboration of this work. To my thesis advisor, Dr. Luis Eduardo Garza, for his motivation and help in the problems that arose in the preparation of this work. To all members of the Robotics focus group that provided constructive criticism. And to my fellow students with whom I have shared good times. I also want to thank the Tecnológico de Monterrey for the opportunity provided and the support in tuition. Finally and without diminishing importance, I thank CONACYT for the financial support provided for living.

Edisson Andrés

Design and construction of the hull of an autonomous underwater vehicle with biomimetic profile and development of instrumentation and control system

by

Edisson Andrés Naula Duchi

Abstract

Seabed research has motivated the development of *Underwater Vehicle*s (UVs). Study of marine life can be performed using *Autonomous Underwater Vehicle*s (AUVs) but most of the development has been restricted to conventional propulsion systems based on jet or propeller thrust generation. These systems are limited in their deployment due to the excessive noise that tends to scare fishes and other forms of marine life during exploration. Moreover, the appearance of these vehicles can be disruptive to the marine environment. Nowadays, the relatively high-cost of fuel and rules of environmental protection have encouraged researchers to find more eco-friendly solutions. This work proposes a novel design of an AUV based on biomimicking the anatomy and swimming mechanics of tuna and dolphin. Simulation and construction of a prototype were performed to test and verify the successful integration of the components. Moreover, numerical simulation was developed to assess the effect of the propulsion system under hydrodynamic conditions to evaluate the feasibility of the design.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Underwater Vehicles

Three quarters of earth's surface are cover by water. Seabed research has motivated the development of Underwater Vehicles (UVs). These vehicles are classified according to their autonomy, type of mission and propulsion. A detail classification is shown in Figure 1.1.

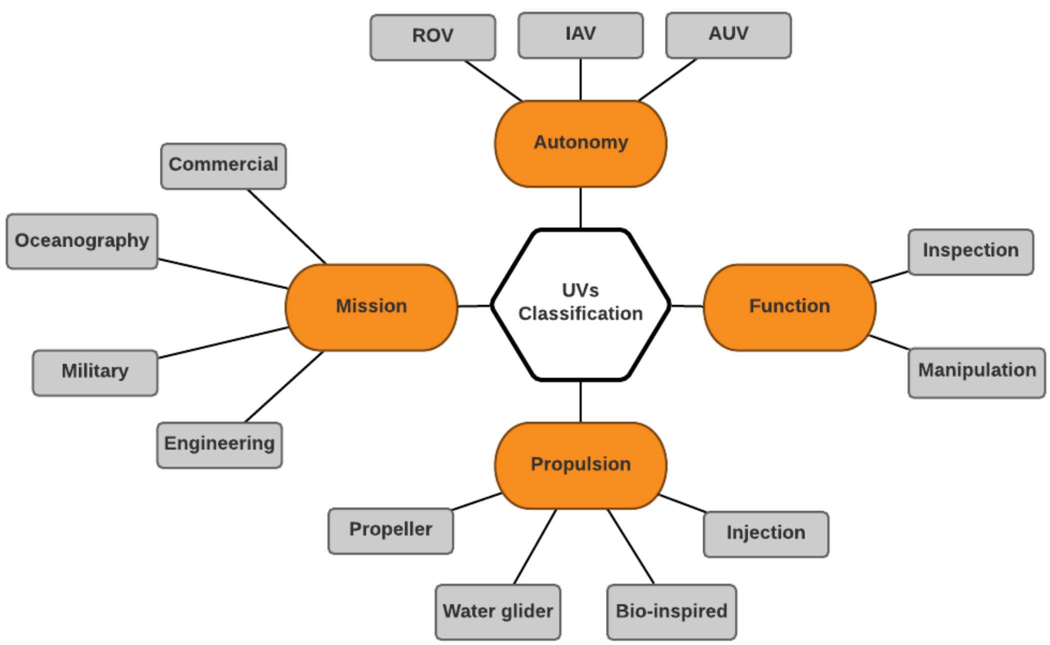


Figure 1.1. UNDERWATER VEHICLES CLASSIFICATION BASED ON [23]

According to their autonomy, underwater vehicles can be *Remote Operated Vehicle*s (ROVs), AUV those who are able to perform tasks without any human interaction and *Intervention Autonomous Underwater Vehicle* (IAUV) who are in an intermediate point between full autonomy and remotely operated. A ROV is operated by humans through a wired or wireless communication system, while an AUV does not require human intervention and is operated by an intelligent control algorithm. So ROVs and AUVs not only increase security but also provide certain benefits.

1.1.1 Autonomous Underwater Vehicles

The AUVs have an architecture that allows them to operate without an operator. In the work presented by [28], four categories are defined in which the AUVs are used:

- **Commercial**: industrial robots are used for underwater tasks such as: inspections, repairs, sample collection, etc.
- **Oceanography missions**: research and regulatory institutions use robots to obtain samples, estimation of the degree of pollution and to obtain health indices of marine life.
- **Military missions**: in this case robots are used to increase the surveillance and rescue capacity.
- **Engineering**: robots are used to test navigation systems, locations systems and control technologies.

1.2 Parts of an Underwater Vehicle

An UV's composition is determined by the type of vehicle, given a required task to be done underwater. Each task will required different types of tools. However, a general composition can be stablish as follows.

1. **Hull** The hull consists of the structure and coating of the UV, so here the main hardware is placed. The design depends on parameters such as: pressure, operating temperatures, corrosion, easy access to hardware, scalability and cost [33]. A main factor, for construction purposes, is the drag generated by the geometry of the hull, because of this 55% of UVs are torpedo shaped [2].
2. **Sensors** Sensors of an UV are used for navigation, measurement of the UV status and for specific tasks according the vehicle's mission.
3. **Propulsion system** The propulsion system allows the vehicle to move in an aquatic environment. Some of the main propulsion schemes, according [23], can be:
    - **Propellers**: this is the most common used in underwater robots and consist of an electric motor coupled to a helix that generates the thrust when displacing the fluid.
    - **Water glider**: this kind of vehicles move without a thrust generator, but they slide through the fluid instead. Usually, their missions consist in descend to a certain point and then ascend by changing buoyancy and the pitch.
    - **Bio-inspired**: here the propulsion system is inspired on fish's physiology and aim to simulate their way of displacement in water.

- **Injection**: this type of propulsion absorbs fluid from its surroundings to then eject it using a pump. The fluid comes out under pressure which generates the thrust to move the vehicle.
- **Magnetohydrodynamic**: this system circulates electric current through a magnetic field to generate movement. The advantage in this type of system is that there is no moving part on it.
- **Traction with the bottom of the sea**: this system has the same principle as a terrestrial vehicle, also has the advantage that the weight can be canceled by the buoyancy force.

4. **Robotic arms** These are necessary when the task involves the manipulation of objects. Robotic arms are usually remotely operated.

5. **Power source** The power source is a critical part to define the weight, time of operation and the volume of the UV. Batteries and fuel cells are widely used in small vehicles.

6. **Ballast System** According the work in [24], the ballast tanks can fit into two categories:

   - Main ballast tanks used for major adjustment of submarine mass.
   - Trim and compensation ballast system is used for minor adjustments in balance.

   These systems are usually located inside the main hull. In the work of [4], there is a review of different designs, which do not include a ballast system and are more efficient than those who use it.

1.3 Problem Definition

The use of AUVs can help with the study of marine life, but the classic propulsion systems are usually jet or propeller based which can generate turbulence and scare the fish ([16]). The study of marine life requires a non-disruptive way to observe the fish behavior. Also, the appearance of these vehicles can't fit in well with the marine environment. Nowadays the fuel price and the change in the rules of environmental protection ([4]), have rushed the development of energy efficiency and eco-friendly solutions.

1.4 Hypothesis

The design of a hull and pectoral fins with an efficient hydrodynamic profile, as well as the adequate electronic instrumentation, will allow the navigation of the *Biomimetic Autonomous Underwater Vehicle* (BAUV) to be carried out efficiently with minimum environmental impact.

In this study, the following research questions are posed:

- Is it possible to build in an economic manner the hull of a BAUV?
- Is it possible to integrate in an efficient way the components of the BAUV?

1.5 Objectives

1.5.1 General Objective

The research is focused on the design and construction of the hull of a biomimetic autonomous underwater vehicle (BAUV) with minimum drag and weight, as well as the design of the necessary instrumentation and control for navigation and communication with a terrestrial base.

1.5.2 Specific Objectives

- Design and build a hull for the BAUV with weight characteristics and hydrodynamic profile to minimize drag forces.
- Design and build the pectoral fins for the BAUV with hydrodynamic profile to minimize drag forces and maximize vehicle maneuverability.
- Design and implement the electronic, communication and control interfaces of the BAUV with efficiency characteristics to minimize energy consumption.

1.6 Methodology

The methodology to follow, for the thesis work, is shown in Figure1.2 and consists of the following stages:

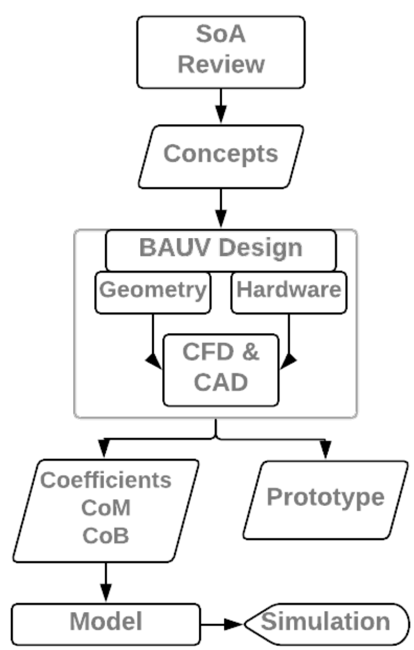


Figure 1.2. BLOCK DIAGRAM OF THE METHODOLOGY

1. **Review of the State of Art**:This stage is focused in the review and study of UVs. First, we search for a classic approach for submarine geometries. Then look for a biomimetic approach, but with special interest in thunniform propulsion. At last, similar works are reviewed. This is done to have basic concepts for the design of the BAUV.

2. **Design and Development**: The design starts with the basic concepts to be used. Then, we define a selection of material for the manufacture. This stage can be divided in:

   (a) Hull Design and Construction

   (b) Fin Design and Construction

   (c) Impermeability

   (d) Instrumentation

   (e) Software

3. **Model and Simulation**: A model of the motion of the vehicle is defined with the external forces influencing the motion. This model requires a hydrodynamic analysis of the geometry. Then we can use the model to have a first look of the BAUV's behavior.

4. **Results and Conclusions**: Finally, the results are a prototype and a mathematical model for the BAUV. Now an analysis of the results can be done and conclusions are given.

1.7 Thesis Outline

The next chapter will review the state of art on the development of UVs, but with a special focus on BAUVs. In chapter 3, the basic principles, for the development, are presented. In chapter 5, the development of the hardware and software is presented. The mathematical model is obtained in chapter 4 and complemented with a hydrodynamic study. Finally, in chapter 6, we review the result of the previous stages and make conclusions about them.

Chapter 2

State of the Art

2.1 Biomimetic Autonomous Underwater Vehicle

Biomimetic refers to the imitation of nature in systems created by man. It differs from bioinspiration in that it completely imitates natural systems and not only relies on them. The effectiveness of the biomimetic approach lies in the understanding of the mechanism to be imitated [25]. The abundance of life in the oceans offers different locomotion tactics, maneuver tactics, resistance to pressure, reduction of resistance, navigation systems and any other requirement for survival in an underwater environment.

Of interest are the locomotion tactics of the fish. For instance, the tuna can surpass any vehicle of the same size made by man, in speed and turning capacity [31]. The design of an AUV has a direct effect on factors such as speed, maneuverability, maximum implementation time, reliability and general robustness [25]. These factors are the basis for the development of BAUVs.

2.1.1 Locomotion

The locomotion of a fish is classified taking into account the thrust is generated: Many water specimens use *Body Caudal Fin* (BCF), to generate thrust by bending their bodies in a wave of propulsion backwards that extends to its tail fin. Other fish have developed alternative mechanisms of locomotion that involve the use of their medium and pectoral fins, called *Median Paired Fin* (MPF) [27]. Figure 2.1 shows the different types of locomotion. The thrust generated by BCF is used for propulsion while the one generated by MPF is used for maneuvers and stabilization. The locomotion can also be classified by undulations, when a wave crosses the body to the tail, or by oscillations, when the propulsive structure rotates on its base without exhibiting a wave formation.

Each type of locomotion has different advantages; They can be ecological, work with low frequency cycles and have a high efficiency. The combination of these types of locomotion imply better performance [8]. From all the mentioned above, the Thunniform locomotion mode is the most efficient [27]. It allows to maintain high cruising speeds for long periods of time. Also, significant lateral movements occur only in the caudal fin and in the area near the narrow peduncle. As a consequence, the body must be aerodynamic to significantly reduce the resistance to pressure.

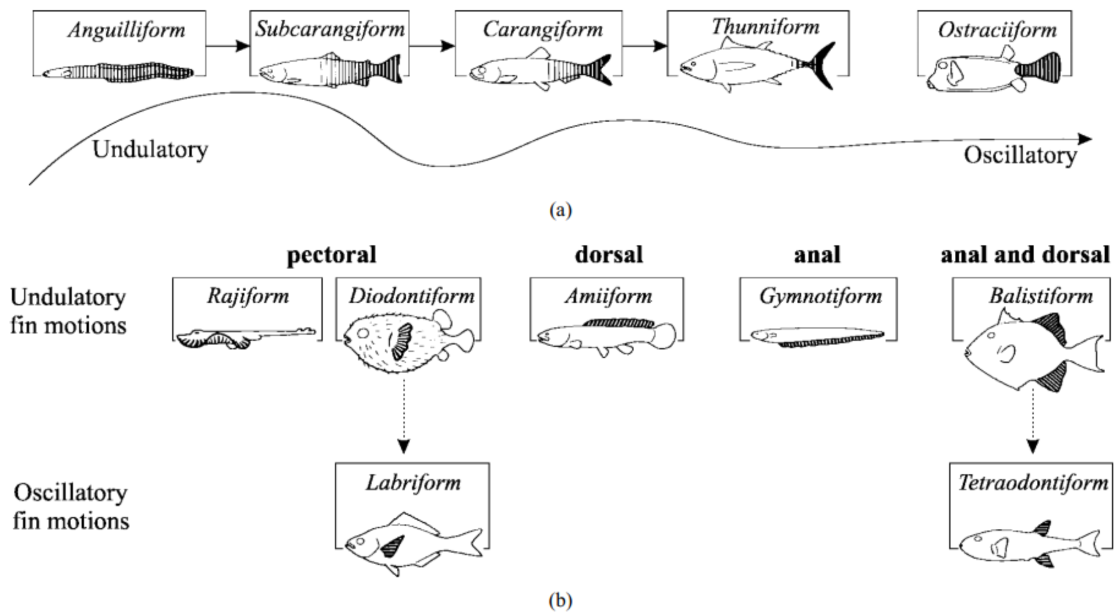


Figure 2.1. TYPES OF LOCOMOTION. A)BCF AND B)MPF BASED ON [27]

2.1.2 Body Shape

Submarine Geometry

Submarine vehicles are complex to design given the hydrodynamic analysis required. In the literature many different geometries have been developed. These geometries are unclassified but allow us to have benchmark results of their hydrodynamics. A summary of the main geometries is presented in [24]. Here, the geometries are parameterized with terms as: aft body, middle body, fore body, body length, cross-section diameter, tail and nose radius. Some of the mentioned geometries (Figure 2.2) are:

- **Series 58**

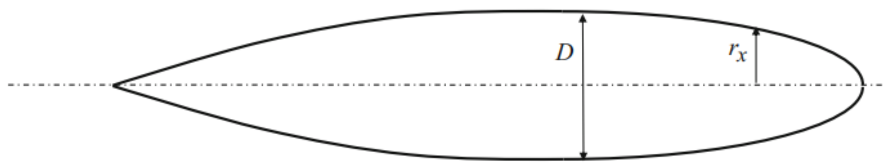
  This series compressed 24 mathematically related streamlined bodies of revolution. The resulting shape, shown in Figure 2.2a, is defined as a sixth-degree polynomial.

- **Myring**

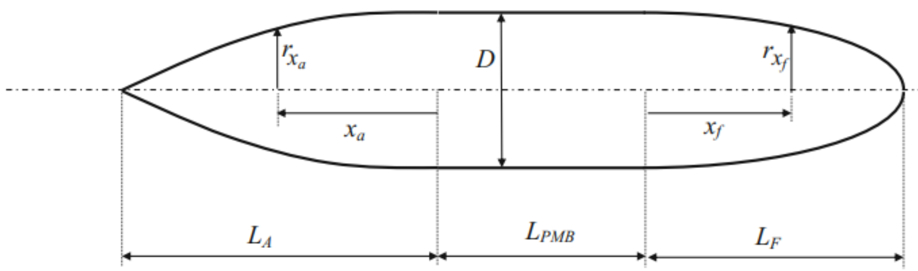
  The Myring shape, shown in Figure 2.2b, is suitable for hulls composed by three parts: an elliptical fore body, a parallel middle body and a parabolic aft body. The fore body radius is defined as $r_{x_f} = f(x_f, D, n_f, L_f)$, where x_f is the distance from the rearmost part of the fore body, n_f is the a coefficient that defines the fullness of the fore, D is the diameter of the middle body and L_f is the fore body length. The radius of the aft body is defined, like in the fore body, as $r_{x_a} = f(x_a, D, L_a, \alpha_t)$, in this case α is the half tail cone angle. The radius of the middle part is $r_m = D/2$.

- **Iranian Hydrodynamics Series of Submarines**

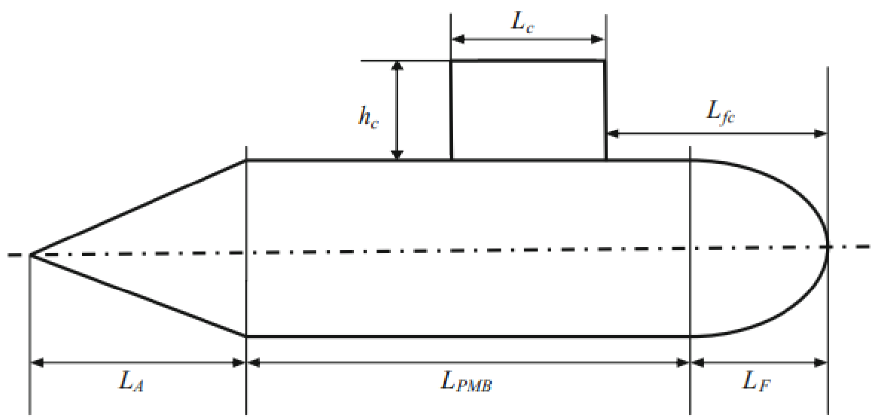
  The *Iranian Hydrodynamic Series of Submarines* (IHSS) are like the Myring shape, but these contain a sail at the top. The model was designed to systematically investigate the hydrodynamics. The name is specified as a 15-digit code where the first seven define the hull and the rest the sail. These digits are values of ratios between parameters of the shape. A diagram is shown in Figure 2.2c.



(a) PROFILE OF SERIES 58 SHAPE



(b) PROFILE OF MYRING SHAPE



(c) DEFINITION OF PARAMETERS IN IHSS

Figure 2.2. PROFILES AND PARAMETERS OF SUBMARINES BASED ON [27]

Biomimetic Shape

Despite the power of the main thrust, the shape of the body and the distribution of the mass ensure that the recoil forces are minimized effectively, and very little lateral slip is induced [27]. In an AUV, the helmet should ideally have the shape of a body suitable for the type of locomotion to mimic; but it must also be light and resistant. Another factor to consider is corrosion due to saltwater.

Spherical hulls offer the best structural integrity; however, the shape inhibits the efficient use of available space since most of the components and systems are rectangular. The cylindrical hulls provide the best alternative, which includes a high structural integrity and a suitable shape for the support of electronic components [7].

2.1.3 Rigid Body

The tuna was chosen as a source of inspiration because it is one of the fastest swimming fish in nature and can maintain long periods of locomotion at high speeds. Another characteristic of a tuna is that the different subspecies of tuna have similar morphologies despite the difference in size, this implies that any design would be easily scalable [25]. Also, Thunniform locomotion allows a large proportion of the body to remain rigid.

2.1.4 Pectoral Fins

In the Thunniform locomotion the displacement is like that of a glider. When the pectoral fins are fixed, a moving internal mass is necessary to adjust the sliding movement that can bring a slow response. In contrast, in the work of [36], the mobile pectoral fins allow to control the movements of yaw and pitch. In addition, the pectoral fins can also provide enough thrust for multimodal locomotion, such as forward swimming, spinning, descending and climbing.

2.1.5 Wireless Communications

The wireless communication devices depend on the bandwidth required and the distance at which it is desired to transmit. This, because the energy consumption, security, transmission distance and other parameters vary in the different available technologies. The table 2.1 shows a summary of existing technologies.

Table 2.1. BASIC WIRELESS COMMUNICATION TECHNOLOGIES

Technology	**Distance (m)**	**Capacity (kbps)**	**Frequencies (MHz)**	**Penetration in water (m)**
Bluetooth	10 to 100	1 to 3 $x10^3$	2400	10
Zigbee	500 to 1000	20, 40 and 250	868, 915 and 2400	10
WiFi	Up to 150	10 to 100 $x10^3$	2400 and 5000	10
Acoustic waves	100 to 460	100 to $1x10^3$	0.1 to 0.25	100 to 460

2.2 Latest developments

The mode of locomotion of a BAUV defines its type of movement and maneuverability. The hull, as previously mentioned, supports the vehicle's hardware, the propulsion device (tail), it also provides insulation and its geometry defines the drag force generated.

The best shape of an UV is when it completely mimics a fish shape, but this reduces the capacity of the vehicle's load. The corrugated robot FILOSE, proposed by Kruusmaa in [19], imitates a rainbow trout. This robot has a flexible body to adapt to the current flow. The main objective of its form is the exploitation of the same and the material properties of the body to create robots mechanically simple but energetically favorable and robust. The shape of this body does not allow to have a significant load, to perform different tasks, therefore another geometry must be used. The work of Katzschmann in [16], proposes a soft robotic fish that swims in three direction and records the aquatic life. This robot has an soft robotic actuator design to exhibits a lifelike undulating tail motion for potentially facilitate the integration in a marine environment. This fish navigates successfully at depths up to 18 meters.

Several designs agree that a cylindrical shape provides a better balance between higher load and less drag. In the robot developed by Aras in [7] a cylindrical hull with propulsion generated by a propeller is proposed. This work focuses on the need to optimize the use of the energy source. The *Iver3*, presented in [11] developed by Crowell, has a torpedo-shaped hull and is an improvement to its previous version in terms of energy, drag reduction and noise due to interference. The work of Hong-Jian in [12] uses a torpedo shape and analyzes the stability of the vehicle. The *ISiMI* vehicle developed by Jun in [15], also in the form of a torpedo, is used to test different algorithms and instruments that improve the performance of an AUV. In the work carried out by Alam in [3], an optimization of the physical components in a cylindrical helmet is performed. Likewise, the study of Singh in [29] proposes a model to design AUV that use a system of 4 helices for its displacement, this to minimize the drag. Finally, the publication of Wang ([33]) emphasizes the capabilities of cylindrical geometry to resist pressures and the ballast system is studied for depth control. In general, a cylindrical geometry is used because of its good characteristics, not the best ones, but that present an advantage in manufacturing. A review of the classification of biological and bioinspired aquatic systems is presented by Salazar in [26]. Some of these systems are shown in Table 2.2 and Table 2.3.

Table 2.2. SUBCARANGIFORM & CARANGIFORM FIN OSCILLATION ROBOTS

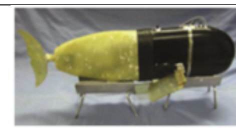
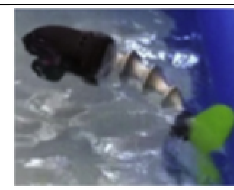
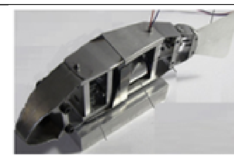
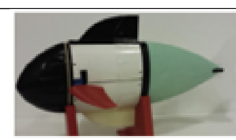
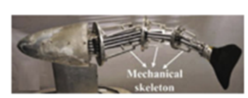
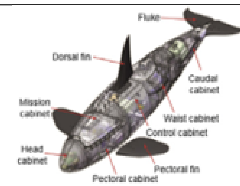
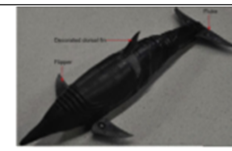
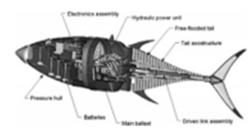
Robotic System	Description	Picture
Four-link Robotic Fish Large Pectoral Fin Control Surfaces [39]	-On board sensors and control surfaces -Pectoral fins (square surface area) can give backwards locomotion -First half of body is rigid -Four-link peduncle has a covering that is streamline with the body	
Four-link Carangiform Fish Robot [18]	-Four-linked actuation mechanism attached to small head unit -Small spinal skeleton for actuation -Tight skin over peduncle unit not streamline to body -Should be noted that this system did have to perform swimming missions	
iSplash-I [10]	-Whole body incorporates multiple links -Allows for head side to side movement -This is one of the first prototypes of this design -Components are made of rigid materials	
Wire-driven Shark Robot [20]	-Body undulation too large to be considered Thunniform -Simple cable-actuation mechanism -The control surfaces are semi-fixed -Peduncle does not have streamline skin	
Hydraulic Soft Robotic Fish [17]	-Hydraulic peduncle actuation, soft peduncle and tail design -Functioning control surfaces -Rigid body	
Mackerel Robot [35]	-Four-linked design used to study hydrodynamics -Belt translation of motion down the body and skeleton -Transmission shafts excite belts through vertical strut -Body is mimetic of the Mackerel -Flexible streamline skin	

Table 2.3. THUNNIFORM FIN OSCILLATION ROBOTS

Robotic System	Description	Picture
Gliding Robotic Dolphin [36]	-Three peduncle joints -Working pectoral fin control surfaces -Gliding implemented for increased endurance -Has rigid body compartment that holds the electronic components	
Dolphin Robot Capable of Leaping [38]	-This design incorporates dolphin head movement to realize greater tail thrust -Control surface pectoral fins and dorsal fin is rigid -Has sensors for pitch, yaw, and roll -Powerful DC motors contained in the main body and peduncle unit	
Vorticity Control Unmanned Undersea Vehicle(VCUUV) [5]	-A hydraulic robotic tuna -Main body is rigid -Peduncle section is comprised of different rigid sections to give flexibility -Heavy, multiple batteries, and large hydraulic actuator -Pectoral fins are rigid and only rotate for pitch control	

2.3 Proposed Design

With the previous review of features of biomimetic robotic systems, now a comparison can be established between the state of art and the proposed BAUV. Table 2.4 shows this comparison with features as body composition, control of surface with pectoral fins, autonomy of the vehicle, main power source and type of propulsion system.

The proposed robotic fish has general features as: a rigid body, to improve manufacture; pectoral fins, for control surface to change maneuverability; batteries as main power source; and propulsion system that allows a vectorized thruster. These features will allow the robot to integrate in a marine environment and provide an adecuate motion.

Table 2.4. FEATURE COMPARISON

	Features				
Robotic System	**Body Composition**	**Pectoral Fins**	**Autonomy**	**Power Source**	**Vectorized Thruster**
Robotic Dolphin	Rigid	Yes	AUV	Battery	No
VCUUV	Rigid	No	ROV	External	No
Sofi	Flexible	Yes	ROV	Battery	No
Filose	Flexible	No	ROV	External	No
Classic AUV	Rigid	No	IAUV	Fuel	No
Proposed	Rigid	Yes	AUV	Battery	Yes

Chapter 3

Mechanical Design

3.1 Basic Physics

3.1.1 Flotation

The principle of Archimedes states that if a body is partially or totally submerged in a fluid, it exerts an upward force on the body equal to the weight of the fluid displaced by the body. When submerging a body, the fluid exerts a total force upwards. This upward force is called the buoyant force and is given by equation (3.1). The line of action of the floating force passes through the center of gravity of the displaced fluid, which does not always coincide with the center of gravity of the body.

$$F_B = \rho g V_f \tag{3.1}$$

3.1.2 Drag Force

In a fluid-body interaction, a force action in opposition to the relative motion is called drag force. Unlike friction, this force depends on the speed of the movement as observed in the equation (3.2).

$$F_D = \frac{1}{2} C_D \rho u^2 A \tag{3.2}$$

where ρ is the density of the fluid, u the velocity, A is the cross section area of the object and C_D is the drag coefficient.

3.1.3 Stability

The stability of the fish ultimately depends on the creation of control forces to make corrections and to help resist the changes. Also, most fish behaviors involve maneuvers, which can be defined as a series of changes in direction and position for a specific purpose [34]. The state of the center of mass, in the frame of reference of the organism, are defined in three planes of translation and around three axes of rotation (Figure 3.1).

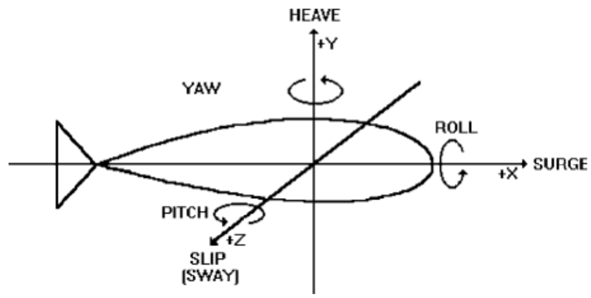


Figure 3.1. COORDINATE SYSTEM WITH SIX DEGREES OF FREEDOM [34]

Most disturbances, corrections and maneuvers involve changes of state. Changes of translation in the state are heave, slip, surge. The rotation changes are pitch, yaw, and roll. The disturbances and control forces used to achieve equilibrium and drive maneuvers can be of hydrostatic or hydrodynamic origin. Hydrostatic disturbances arise from the distribution of tissues. The body composition determines the location of the centers of mass and buoyancy (Figure 3.2).

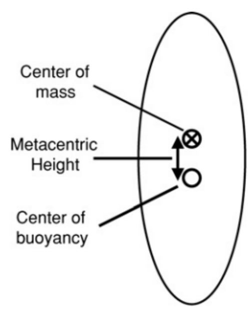


Figure 3.2. SCHEMATIC CROSS SECTION WITH THE RELATIVE LOCATION OF THE CENTERS OF MASS, BUOYANCY AND METACENTRIC HEIGHT [34]

Fish continuously experience disturbances that require control for correction. The alterations considered in the work of [34] are: the control of the orientation of the body (posture), the depth in the water column and the trajectory.

1. **Posture:** Posture control guarantees a stable base for sensory systems and minimizes energy costs during swimming by orienting the body to minimize resistance [34].

2. **Depth:** The control of depth is important depending on the activities to be performed. The density of tissues in a fish is greater than the density of water, so that it sinks. Gas is widely used to control density because it provides the greatest net thrust for a given volume, but as noted above, the volume of inclusions follows the laws of gases, which makes fish hydrostatically unstable in regulation of depth.

3. **Swim Speed:** Slow swimming is challenging for control systems [34]. In contrast, fast swimming presents different opportunities. One of them is the decrease in resistance induced by increasing speed. In addition, fish fold their pectoral fins at a higher swimming speed to eliminate a source of destabilization.

4. **Trajectory:** The caudal-body fin is used to swim at the highest speeds, and generates the greatest recoil forces, which causes sliding and skidding disorders [34]. To dampen this effect the fish, use the combined inertia and the hydrodynamic resistance to the sliding of the anterior part of the body.

3.2 Propulsion System

The propulsion system is based on a parallel mechanism with spherical motion and *3-UCU-1S* architecture. A more complete study is presented in the work of [6]. The mechanism is an assembly of three legs consisting on Universal-Cylindrical-Universal joints, from a base to a moving platform; and, a fourth leg fixing the base with the moving platform through a spherical joint on top. This is shown in Figure 3.3.

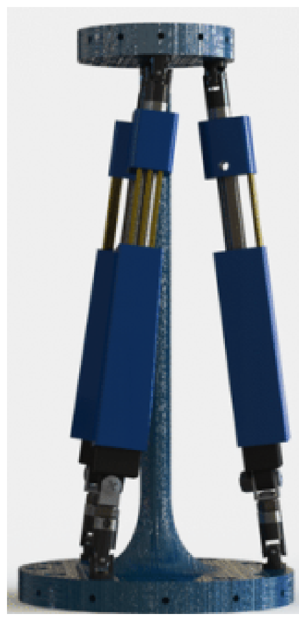


Figure 3.3. PROPULSION SYSTEM MECHANISM [6]

Parallel mechanisms configuration has advantages, over serial configurations, like mechanical simplicity and rigidity. AUVs with a parallel mechanism on the propulsion system can change its orientation along arbitrary space trajectories, producing a vectored thruster. This is shown in Figure 3.4a and Figure 3.4b. This mechanism allows us to mimic the thunniform swing.

3.3 Geometry of the hull

The hull for AUVs can be divided into two parts: the first is the nose that has a strong impact on the calculation of the drag force. The second is the body that contains the electronic elements of the BAUV and defines a part of the drag of the vehicle. The measures of a

(a) VERTICAL FLUTTERING

(b) HORIZONTAL FLUTTERING

Figure 3.4. POSITIONS OF THE CAUDAL FIN FOR MOTION

cylindrical hull can be optimized, in addition a conical nose can be defined to reduce the drag [3]. This design is observed in Figure 3.5.

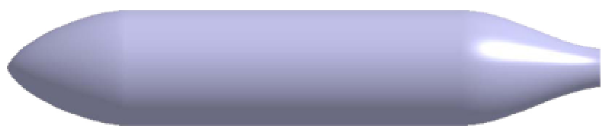


Figure 3.5. PRELIMINARY DESIGN OF A HULL [3]

Although the design of Figure 3.5 can be optimized to reduce resistance, this is not a design that imitates the body of a fish. This because it has a cylindrical cross section and the body of a fish has a more elliptical cross section; Then, the final design sought is like the previous works presented in chapter 2.

In general, a submarine hull has an axisymmetric body, this mean, it is perfectly symmetric around an axis. In the Figure 3.6, the body is symmetric around Z and X axis or symmetric respect the XZ plane. Given that the BAUV will have a thunniform propulsion, it is convenient to base the hull on the body of fish such as tuna, shark or dolphin.

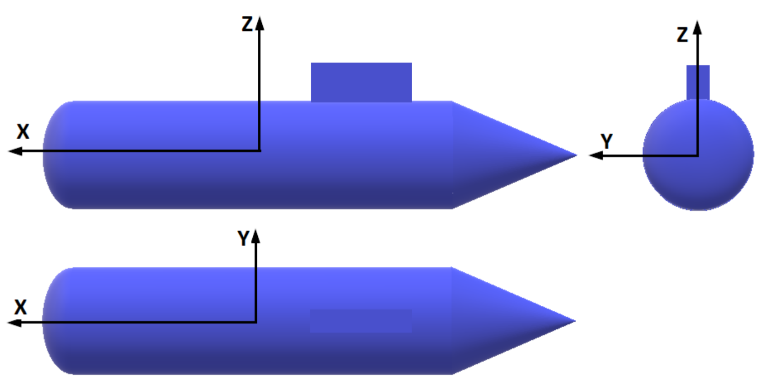


Figure 3.6. VIEWS OF THE HULL WITH AXIS

The Figure 3.7 shows the most important parts of the BAUV hull, such as: the elliptical front body, a parallel middle body and a parabolic shaped aft body. The front will contain a camera, the middle the main electronic components and the aft the propulsion mechanism. This implies that the BAUV is divided in the tail and the parts above, so the complexity of the impermeability and manufacture is based on these sections.

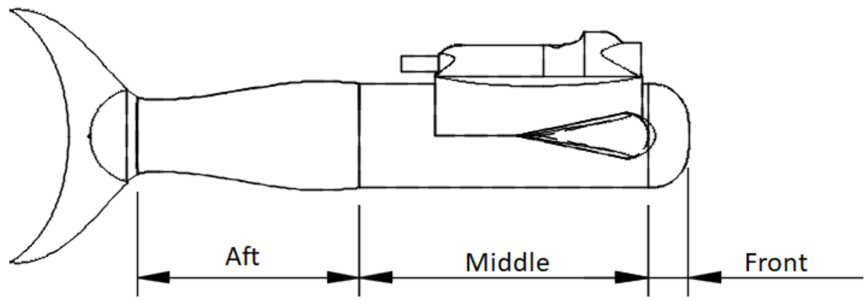


Figure 3.7. SCHEME OF THE BAUV PARTS

3.4 Pectoral Fins and Appendages

In thunniform locomotion the caudal fin causes the main impulse. In this case the pectoral fins are used to control the direction and inclination. These fins can be rigid and the variation of area, which cause to control the inclination of the same, allows to have more degrees of control.

Sharks are characterized by pectoral fins located ventral-lateral with an aspect ratio, in most of the species of 1.5-2.5, although some species of up to 5. These fins have effect on behaviors such as hold, rise and sink when swimming [21]. A schematic diagram of the orientation of the fin, during the retention, elevation and sinking behavior of leopard sharks is show in Figure 3.8. The angle of attack is measured between the string (broken line) and the flow (dotted line).

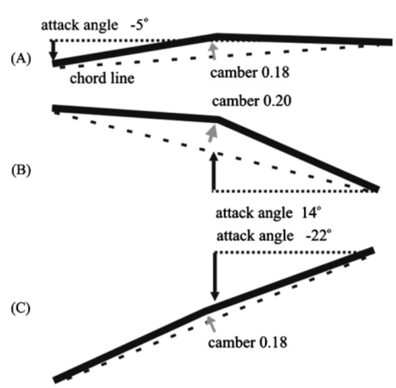


Figure 3.8. A) HOLD, B) RISE, C) SINK [21]

Fish also have dorsal and anal fins but, in our design, the dorsal fin is replaced by a sensor attached to the hull though a semicircular brace that will also help to contain the pectoral fins.

More details are given in chapter 5.

3.5 Impermeability and Manufacture

A problem commonly encountered in the manufacture of AUVs is to ensure impermeability between dry and wet sections. The wet sections depend on the buoyancy system. The term manufacturing complexity is defined as the number of impermeable walls required to seal the dry and wet sections [3]. Ideally, a small numbers is better for manufacturing. A scheme of the location of impermeable walls is shown in Figure 3.9.

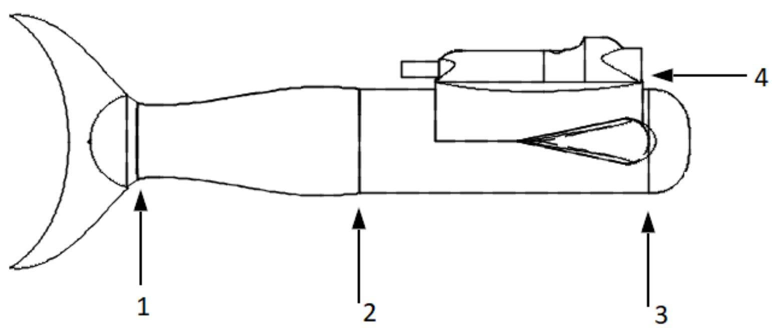


Figure 3.9. IMPERMEABLE WALLS REQUIRE FOR THE BAUV

Walls 2 and 3 have the same level of complexity for the size and location at both ends of the middle body. Wall number 4 is simpler given that is a small hole for wiring. The design of wall number 1 is not shown in this document. More details are presented in chapter5.

3.6 Design Optimization

The performance of an AUV can be measured based on its energy consumption, which implies the time of autonomy of the vehicle. This energy consumption is reflected in the power consumed by the electronic systems and the power used by the propulsion of the vehicle [9]. Therefore, the propulsion power is related to the drag force. The hull design directly influences the drag generated by the vehicle; this is seen in equation 1 where the drag coefficient is given by the shape of the hull. Hence, the design of the helmet has a great effect on the performance of an AUV.

3.6.1 Energy Consumption

The energy consumption can be measured in the consumption of the electric current per hour or also in required power. The desired performance is achieved by minimizing drag and improving propulsion efficiency [9]. Equation (3.4) shows the proportionality between propulsion power and drag.

$$P = P_{prop} + P_H \tag{3.3}$$

where

$$P_{prop} = \frac{1}{2}\frac{C_D A \rho u^3}{\eta} \tag{3.4}$$

where P_H represents the power consumed by all the subsystems such as communications, control and sensors. The power corresponding to the propulsion is a function of the drag coefficient of the vehicle (C_D), the transverse area (A), the density of the water (ρ), speed of the vehicle (u) and the efficiency of the propulsion system (η).

3.6.2 Drag Force

The drag can be estimated by numerical methods or using the drag coefficient of equation (3.2). Also according [1], the drag coefficient can be calculated by:

$$C_D = \frac{0.075}{(log_{10}R_n - 2)^2} \tag{3.5}$$

where R_n is the Reynolds number wich can be found as

$$R_n = \frac{\rho u L}{\mu} \tag{3.6}$$

where ρ is the density of the fluid, u is the velocity, l the overall length of the vehicle and μ is the dynamic viscosity of the fluid. For AUVs there are different ways to calculate the drag force based on the coefficient of viscous resistance C_V. In the work of Alam in [1], three alternatives for calculating C_V are proposed. Equations (3.7), (3.8) and (3.9) are based on the friction coefficient of the hull skin (C_D), which is a function of the Reynolds number. Therefore, three calculation methods of the coefficient are derived:

- The method of Gillmer and Johnson suggests calculating the coefficient using equation (3.7), where the maximum diameter of the hull (D) and the length of the vehicle (L) are involved.

$$C_V = C_D\left[1 + 0.5\left(\frac{D}{L}\right) + 3\left(\frac{D}{L}\right)^3\right] \tag{3.7}$$

- According to the Virginia Tech method (equation (3.8)), the previous method must be modified to consider the shape coefficients of the nose and tail of the vehicle (n_n and n_t respectively).

$$C_V = C_D\left[1 + 0.5\left(\frac{D}{L}\right) + 3\left(\frac{D}{L}\right)^{7-n_n-\frac{n_t}{2}}\right] \tag{3.8}$$

- Likewise, the MIT method considers the prismatic coefficient (C_p) and generates equation (3.9).

$$C_V = C_D\left[1 + 0.5\left(\frac{D}{L}\right) + 3\left(\frac{D}{L}\right)^3 + 0.002(C_p - 0.6)\right] \tag{3.9}$$

The vehicle drag is calculated as

$$D = \frac{1}{2}\rho u^2 C_V S \tag{3.10}$$

where S is the surface of the vehicle in contact with the water.

3.6.3 Hull Shape Optimization

The objective of optimizing the shape of a helmet is to reduce the resistance of the AUV to movement. This can be done for a specific design speed or several speeds as needed. There are several ways to choose target functions of an optimization problem. The method presented in equation (3.11), seeks to minimize resistance functions f_1, elevation f_2 and pitch f_3, where x is a vector with parameters dependent on the shape of the hull [13].

$$min\{f_1, f_2, f_3\}(x) \quad subject\ to \quad X \in S \subseteq R^d \tag{3.11}$$

In chapter 2, many shapes of a submarine hulls were shown. These shapes can be adapted for the BAUV geometry given the number of parts that form the hull, hence we can find a way to parameterize the geometry previously mentioned. Other approaches can be based on scale models and the principle of similitude theory.

Similitude Theory

Two systems, described by the same physical laws, that operate under different conditions are physically similar with respect to certain specified physical variables. The proportion of the variables magnitude between the two systems is the same everywhere. In [41], three concepts of similarity are presented:

- **Geometry Similarity:** The model, usually scaled, is the same shape as the application model.
- **Kinematic Similarity:** Fluid flow of both the model and the application must undergo similar time rates of change motions.
- **Dynamic Similarity:** Ratios of all forces acting on homologous points are a constant. These can be classified in cases:
  - The predominant force is that of gravity (similarity of Froude),
  - The body is submerged in a subsonic flow and the predominant force is that of viscosity (similarity of Reynolds).
  - The body is submerged in a supersonic flow and the predominant force is compressibility (similarity of Mach).
  - In very thin sheets of liquid the surface tension prevails (similarity of Weber).

An advantage provided by the dimensional theory is to predict the results of a project, based on those obtained by testing with a scale model.

Chapter 4

Model and Hydrodynamic Study

The simulation of the vehicle allows to estimate the behavior of the same before its construction. This is achieved with a mathematical model and software that allows to simulate mechanical systems [14, 32].

4.1 Mathematical Model

The movement of an AUV is described in two reference planes and the equations of forces that influence the vehicle.

4.1.1 Reference Frames

The frame of reference of an AUV has two coordinate systems: The inertial coordinate system $E - \xi\eta\zeta$ that is fixed to the Earth and the system of movement coordinates $O - xyz$ that is fixed in the vehicle. The AUV movement has six degrees of freedom (6-DOF) with three translational and three rotational movements. The movement parameters of an AUV include:

- Position and attitude (in $E - \xi\eta\zeta$)

$$R = [\xi, \eta, \zeta, \phi, \theta, \psi]^\top ;$$

- Linear and angular velocities (in $O - xyz$)

$$V = [u, v, w, p, q, r]^\top ;$$

- Force and moment parameters (in $O - xyz$)

$$G = [X, Y, Z, K, M, N]^\top ;$$

The transformation between [position-orientation] and velocities [linear-angular] is given by:

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} T_1 & 0_{3x3} \\ 0_{3x3} & T_2 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{4.1}$$

where T_1 and T_2 are

$$T_1 = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \quad (4.2)$$

$$T_2 = \begin{bmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sec\theta\sin\varphi & \sec\theta\cos\varphi \end{bmatrix} \quad (4.3)$$

4.1.2 Movement Equations

The general equations, of movement of the AUV, in six degrees of freedom are based on the kinetic theory of the rigid body and these are:

$$\begin{cases} X = m\left[(\dot{u} - vr + wq) - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})\right] \\ Y = m\left[(\dot{v} - wp + ur) - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})\right] \\ Z = m\left[(\dot{w} - uq + vp) - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})\right] \\ K = I_x\dot{p} + (I_z - I_y)qr + m\left[y_G(\dot{w} + pv - qu) - z_G(\dot{v} + ru - pw)\right] \\ M = I_y\dot{q} + (I_x - I_z)rp + m\left[z_G(\dot{u} + wq - vr) - x_G(\dot{w} + pv - uq)\right] \\ N = I_z\dot{r} + (I_y - I_z)pq + m\left[x_G(\dot{v} + ur - pw) - y_G(\dot{u} + qw - vr)\right] \end{cases} \quad (4.4)$$

where m is the vehicle mass, (x_G, y_G, z_G) are the coordinates of the center of gravity in the dynamic reference plane and (I_x, I_y, I_z) is the moment of inertia to the axes O_x, O_y and O_z. The external forces and moments acting on the vehicle, right part of equation (4.4), are described in vector form as:

$$F = F_G + F_B + F_H + F_T \quad (4.5)$$

The influencing forces are:

Gravity and Buoyancy Force

This is the hydrostatic force acting on the vehicle.

$$\begin{aligned} X_G + X_B &= -(P - B)\sin(\theta) \\ Y_G + Y_B &= (P - B)\cos(\theta)\sin(\phi) \\ Z_G + Z_B &= (P - B)\cos(\theta)\cos(\phi) \\ K_G + K_B &= (y_G P - y_B B)\cos(\phi)\cos(\theta) - (z_G P - z_B B)\sin(\phi)\cos(\theta) \\ M_G + M_B &= -(x_G P - x_B B)\cos(\phi)\cos(\theta) - (z_G P - z_B B)\sin(\theta) \\ N_G + N_B &= (x_G P - x_B B)\sin(\phi)\cos(\theta) - (y_G P - y_B B)\sin(\theta) \end{aligned} \quad (4.6)$$

where (x_B, y_B, z_B) are the coordinates of the center of gravity in the dynamic reference plane, plus the weight P and buoyancy F_B are

$$P = m * g, \quad F_B = \rho * g * V_{AUV}$$

Propulsion Force

Force provided by the propulsion system of the AUV. In equation (4.7) describes the thrust generated by a propeller propulsion system with rotation speed n, diameter D, thrust coefficient K_T and rotation coefficient K_Q.

$$\begin{aligned} X_T &= K_T \rho n^2 D_b^4 \\ Y_T &= 0 \\ Z_T &= 0 \\ K_T &= K_Q \rho n^2 D_b^5 \\ M_T &= 0 \\ N_T &= 0 \end{aligned} \tag{4.7}$$

Hydrodynamic Force

This force is related to the geometry of the vehicle. It includes the force of inertia and forces due to the viscosity of the fluid. The hydrodynamic forces and moments equations, in the $O - xyz$ coordinate system, are defined by:

$$\begin{aligned} X_H &= [X_{qq}q^2 + X_{rr}r^2 + X_{rp}rp] + [X_{\dot{u}}\dot{u} + X_{vr}vr + X_{wq}wq] + \left[X_{u|u|}u\,|u| + X_{vv}v^2 + X_{ww}w^2\right] \\ &+ [X_{\dot{w}}\dot{w} + X_{\dot{q}}\dot{q} + X_{qu}qu] \\ Y_H &= \left[Y_{\dot{r}}\dot{r} + Y_{qr}qr + Y_{\dot{p}}\dot{p} + Y_{pq}pq + Y_{p|p|}p\,|p|\right] + [Y_{\dot{v}}\dot{v} + Y_{vq}vq + Y_{wr}wr + Y_{wp}wp] \\ &+ \left[Y_r ur + Y_{v|r|}\tfrac{v}{|v|}\left|(v^2+w^2)^{1/2}\right||r| + Y_p up\right] + \left[Y_0 u^2 + Y_v uv + Y_{v|v|}v\left|(v^2+w^2)^{1/2}\right|\right] + Y_{vw}vw \\ Z_H &= [Z_{\dot{q}}\dot{q} + Z_{rr}r^2 + Z_{pp}p^2 + Z_{rp}rp] + [Z_{\dot{w}}\dot{w} + Z_{vr}vr + Z_{vp}vp] \\ &+ \left[Z_q uq + Z_{w|q|}\tfrac{w}{|w|}\left|(v^2+w^2)^{1/2}\right||q|\right] + \left[Z_0 u^2 + Z_w uw + Z_{w|w|}w\left|(v^2+w^2)^{1/2}\right|\right] \\ &+ \left[Z_{|w|}u\,|w| + Z_{ww}\left|w\,(v^2+w^2)^{1/2}\right|\right] + Z_{vv}v^2 \\ K_H &= \left[K_{\dot{p}}\dot{p} + K_{\dot{r}}\dot{r} + K_{qr}qr + K_{pq}pq + K_{p|p|}p\,|p|\right] + [K_p up + K_r ur + K_{\dot{v}}\dot{v}] + [K_{vq}vq + K_{wp}wp + K_{wr}wr] \\ &+ \left[K_0 u^2 + K_v uv + K_{v|v|}v\,|v|\left|(v^2+w^2)^{1/2}\right|\right] + K_{vw}vw \\ M_H &= \left[M_{\dot{q}}\dot{q} + M_{rr}r^2 + M_{q|q|}q\,|q| + M_{pp}p^2 + M_{rp}rp\right] + [M_{\dot{w}}\dot{w} + M_{vr}vr + M_{vp}vp] + \left[M_q uq + M_{|w|q}\left|(v^2+w^2)^{1/2}\right|q\right] \\ &+ \left[M_0 u^2 + M_w uw + M_{w|w|}w\left|(v^2+w^2)^{1/2}\right|\right] + \left[M_{|w|}u\,|w| + M_{ww}\left|w\,(v^2+w^2)^{1/2}\right|\right] + M_{vv}v^2 \\ N_H &= \left[N_{\dot{r}}\dot{r} + N_{qr}qr + N_{r|r|}r\,|r| + N_{\dot{p}}\dot{p} + N_{pq}pq\right] + [N_{\dot{v}}\dot{v} + N_{wr}wr + N_{vq}vq + N_{wp}wp] \\ &+ \left[N_r ur + N_{|v|r}\left|(v^2+w^2)^{1/2}\right|r + N_p up\right] + \left[N_0 u^2 + N_v uv + N_{v|v|}v\left|(v^2+w^2)^{1/2}\right|\right] + N_{vw}vw \end{aligned} \tag{4.8}$$

where u, v, w, p, q, r are the linear and angular velocities, and $[X_{ij}, Y_{ij}, Z_{ij}, K_{ij}, M_{ij}, N_{ij}]$ are the hydrodynamic coefficients of the effect of velocities ij in eah axis.

4.1.3 Fins forces

Fins works as rudders for the BAUV. When the fin has an angle of attack α (Figure 4.1a), the effects can be decomposed into lift force and drag force. $F_{L_{fin}}$ and $F_{D_{fin}}$ can be calculated as:

$$\begin{cases} F_{L_{fin}} = \frac{1}{2} C_L \rho A_f u^2 \\ F_{D_{fin}} = \frac{1}{2} C_D \rho A_f u^2 \end{cases} \tag{4.9}$$

where C_L and C_D are the lift and drag coefficients of the fin, ρ the density of the fluid, V the velocity of the fluid and A_f the lateral area of the fin. The coefficients are estimated through a *Computational Fluid Dynamics* (CFD) simulation for every angle. The result is shown in section 6.1.1.

4.2 Types of Test

To obtain the hydrodynamic coefficients for the model, the movements are restricted to different planes of motion. The planes of motion help to reduce the equation (4.4). In the ZX plane only surge, heave and pitch motion are present. In the XY plane only yaw and sway motion are present. The vehicle does not have roll motion.

First the movement of the BAUV is restricted to the ZX plane. This means that only surge, heave and pitch motion are consider ($v = r = p = 0$), so the equation of motion can be simplified as:

$$\begin{cases} X = m\left[(\dot{u} + wq) - x_G q^2 + z_G \dot{q}\right] \\ Z = m\left[(\dot{w} - uq) - z_G q^2 + x_G \dot{q}\right] \\ M = I_y \dot{q} + m\left[z_G(\dot{u} + wq) - x_G(\dot{w} - uq)\right] \end{cases} \tag{4.10}$$

Then, the hydrodynamic effects generated by the flow allow us to determine the heave and pitch moment by:

$$\begin{aligned} Z_H &= Z_{\dot{q}}\dot{q} + Z_{\dot{w}}\dot{w} + Z_q uq + Z_{w|q|} w\,|q| + Z_0 u^2 + Z_w uw + Z_{w|w|} w\,|w| \\ &\quad + Z_{|w|} u\,|w| + Z_{ww}\,|w^2| \\ M_H &= M_{\dot{q}}\dot{q} + M_{q|q|} q\,|q| + M_{\dot{w}}\dot{w} + M_q uq + M_{|w|q}\,|w|\,q \\ &\quad + M_0 u^2 + M_w uw + M_{w|w|} w\,|w| + M_{|w|} u\,|w| + M_{ww}\,|w^2| \end{aligned} \tag{4.11}$$

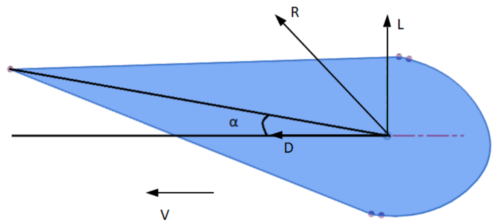
Then the movement of the BAUV is restricted to the XY plane. This means that only surge, sway and yaw motion are consider ($w = p = q = 0$), so the equation oF motion can be simplified as:

$$\begin{cases} X = m\left[(\dot{u} - vr) - x_G r^2 - y_G \dot{r}\right] \\ Y = m\left[(\dot{v} + ur) - y_G r^2 + x_G \dot{r}\right] \\ N = I_z \dot{r} + m\left[x_G(\dot{v} + ur) - y_G(\dot{u} - vr)\right] \end{cases} \tag{4.12}$$

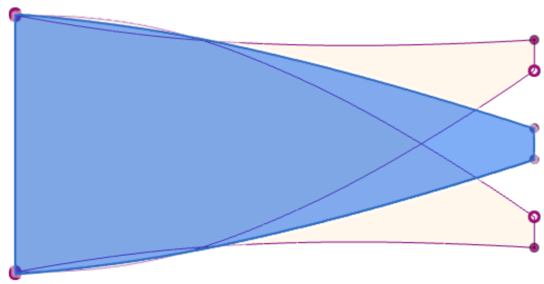
Now, the hydrodynamic effects generated by the flow allow us to determine the heave and pitch moment by:

$$\begin{aligned} Y_H &= Y_{\dot{v}}\dot{v} + \left[Y_0 u^2 + Y_v uv + Y_{v|v|} v\,|v|\right] \\ N_H &= N_{\dot{v}}\dot{v} + \left[N_0 u^2 + N_v uv + N_{v|v|} v\,|v|\right] \end{aligned} \tag{4.13}$$

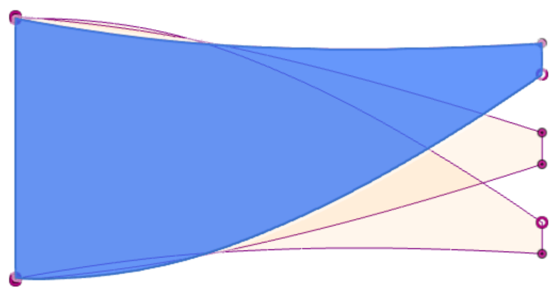
The coefficients of equation (4.11) are estimated by straight line , rotation arm and planar motion mechanism tests.



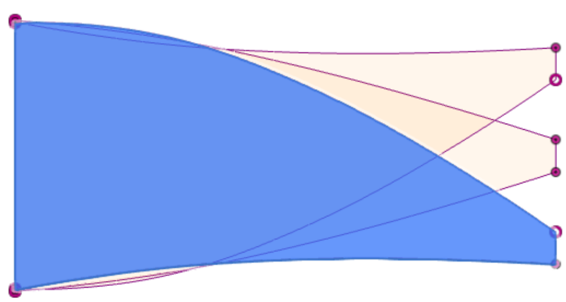
(a) RIGHT SIDE PROFILE



(b) FRONT PROFILE 0^{o}



(c) FRONT PROFILE 10^{o}



(d) FRONT PROFILE -10^{o}

Figure 4.1. PROFILES OF A PECTORAL FIN

4.2.1 Straight Line Test

The test simulates a tunnel where the AUV is placed and the flow has constant velocity. In this test the linear and angular acceleration are zero. Here, the vehicle is located under a given angle respect to the flow direction and there is no pitch movement ($q = 0$). Then, we can write equation (4.11) as follows:

$$\begin{aligned} Z'_H &= Z'_0 u^2 + Z'_{w'} u w' + Z'_{w'|w'|} w' \left|w'\right| + Z'_{|w'|} u \left|w'\right| + Z'_{w'w'} \left|w'^2\right| \\ M'_H &= M'_0 u^2 + M'_{w'} u w' + M'_{w'|w'|} w' \left|w'\right| + M'_{|w'|} u \left|w'\right| + M'_{w'w'} \left|w'^2\right| \end{aligned} \tag{4.14}$$

where $w' = sin\alpha = \frac{w}{U}$, $Z'_H = \frac{Z_H}{0.5\rho U^2 L^2}$ and $M'_H = \frac{Z_H}{0.5\rho U^2 L^3}$ are the non-dimensional components of hydrodynamic force and moment. Also, L is the AUV length, U its velocity and ρ the density of the flow. The resulting force its divided into Lift (F_L) and Drag (F_D) forces that are perpendicular and parallel to the flow velocity. The applied force perpendicular to the longitudinal axis is calculated by:

$$Z = F_D \sin\alpha + F_L \cos\alpha \tag{4.15}$$

4.2.2 Rotation Arm Test

The difference between the straight line test and this one relies on the geometry of the fluid domain (Figure 4.2). For this case, the tunnel is a semi-cylinder and the fluid has a constant angular velocity q.

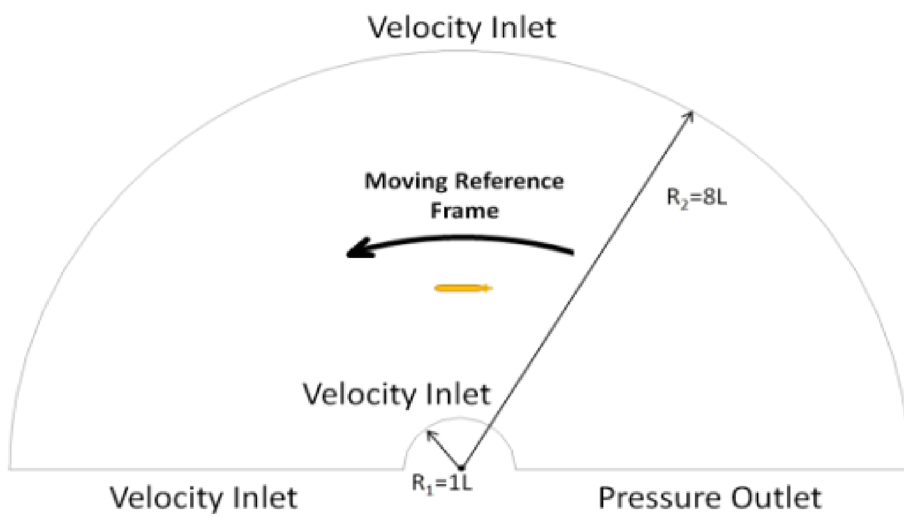


Figure 4.2. FLUID DOMAIN CONDITIONS FOR ROTATION ARM TEST [42]

4.2.3 Planar Motion Mechanism Test

Planar Motion Mechanism (PMM) maneuvers consist in different types of motion. In our first case, only heave and pitch are considered. Here the AUV is subjected to a harmonic motion with constant amplitude and frequency.

Pure Heave

The pure heave motion consists in a harmonic oscillation along z axis and a forward velocity U. The pitch angle θ, angular velocity q and acceleration $\dot{q}$ would be equal to zero in the trajectory. Then for heave motion, the next equations are defined:

$$\begin{aligned} z &= z_0 \sin \omega t \\ w &= \dot{z} = z_0 \, \omega \cos \omega t \\ \dot{w} &= \ddot{w} = -z_0 \, \omega^2 \sin \omega t \end{aligned} \tag{4.16}$$

here z is the vertical displacement , w is the vertical velocity, $\dot{w}$ is the vertical acceleration w, z_0 is the amplitude and ω is the frequency for pure heave motion. The equations (4.11) can be written as:

$$\begin{aligned} Z_H &= Z_{\dot{w}}\dot{w} + Z_0 u^2 + Z_w uw + Z_{w|w|} w\,|w| + Z_{|w|} u\,|w| + Z_{ww}\,|w^2| \\ M_H &= M_{\dot{w}}\dot{w} + M_0 u^2 + M_w uw + M_{w|w|} w\,|w| + M_{|w|} u\,|w| + M_{ww}\,|w^2| \end{aligned} \tag{4.17}$$

Pure Pitch

The pure pitch motion consists in a movement along x axis, a forward velocity U and a harmonic change of θ. The linear velocity w and acceleration $\dot{w}$ would be equal to zero in the trajectory. Then for pitch motion, the next equations are defined:

$$\begin{aligned} \theta &= \theta_0 \sin \omega t \\ q &= \dot{\theta} = \theta_0 \, \omega \cos \omega t \\ \dot{q} &= \ddot{\theta} = -\theta_0 \, \omega^2 \sin \omega t \end{aligned} \tag{4.18}$$

here θ_0 is the amplitude and ω is the frequency for pure pitch motion. The equations (4.11) can be written as:

$$\begin{aligned} Z_H &= Z_{\dot{q}}\dot{q} + Z_q uq + Z_0 u^2 \\ M_H &= M_{\dot{q}}\dot{q} + M_{q|q|} q\,|q| + M_q uq + M_0 u^2 \end{aligned} \tag{4.19}$$

Pure Yaw

The pure yaw motion consists in a rotation movement along y axis, a forward velocity U and a harmonic change of ψ. The linear velocity v and acceleration $\dot{v}$ would be equal to zero in the trajectory. Then for yaw motion, the next equations are defined:

$$\begin{aligned} \psi &= \psi_0 \sin \omega t \\ r &= \dot{\psi} = \psi_0 \, \omega \cos \omega t \\ \dot{r} &= \ddot{\psi} = -\psi_0 \, \omega^2 \sin \omega t \end{aligned} \tag{4.20}$$

here ψ_0 is the amplitude and ω is the frequency for pure pitch motion. The equations (4.13) can be written as:

$$\begin{aligned} Y_H &= Y_{\dot{r}}\dot{r} + Y_r ur + Y_0 u^2 \\ N_H &= N_{\dot{r}}\dot{r} + N_{r|r|}r\,|r| + N_r ur + N_0 u^2 \end{aligned} \tag{4.21}$$

Pure Sway

The pure sway motion consist in a harmonic oscillation along y axis and a forward velocity U. The yaw angle ψ, angular velocity r and acceleration $\dot{r}$ would be equal to zero in the trajectory. Then for sway motion, the next equations are defined:

$$\begin{aligned} y &= y_0 \, \sin \omega t \\ v &= \dot{y} = y_0 \, \omega \cos \omega t \\ \dot{v} &= \ddot{v} = -y_0 \, \omega^2 \sin \omega t \end{aligned} \tag{4.22}$$

here y is the horizontal displacement , v is the linear velocity, $\dot{v}$ is the acceleration, y_0 is the amplitude and ω is the frequency for pure sway motion. The equations (4.13) can be written as:

$$\begin{aligned} Y_H &= Y_{\dot{v}}\dot{v} + Y_0 u^2 + Y_v uv + Y_{v|v|}v\,|v| \\ N_H &= N_{\dot{v}}\dot{v} + N_0 u^2 + N_v uv + N_{v|v|}v\,|v| \end{aligned} \tag{4.23}$$

All these coefficients can be calculated dimensionless as in straight line test.

4.3 Computational Fluid Dynamics

Computational fluid dynamics CFD is a branch of fluid mechanics. They are numerical methods and data structures used to simulate the interaction of a fluid and a surface defined by boundary conditions. With the constant development of computers, CFD applications for the marine industry have become feasible and important [40]. Most models are based on the *Reynolds Averaged Navier-Stokes* (RANS) equations. If it is assumed that the fluid is incompressible, the average flow field is governed by the RANS equations:

$$\frac{\partial U_j}{\partial x_j} = 0 \tag{4.24}$$

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j}\left(U_i U_j\right) = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left(\Gamma_{ij} - \rho\overline{u}_i^{'}\overline{u}_j^{'}\right) \tag{4.25}$$

where ρ is the density of the fluid and Reynolds stress tensor is

$$-\rho\overline{u}_i^{'}\overline{u}_j^{'} = \begin{bmatrix} -\rho\overline{u}_1^{'2} & -\rho\overline{u}_1^{'}\overline{u}_2^{'} & -\rho\overline{u}_1^{'}\overline{u}_3^{'} \\ -\rho\overline{u}_2^{'}\overline{u}_1^{'} & -\rho\overline{u}_2^{'2} & -\rho\overline{u}_2^{'}\overline{u}_3^{'} \\ -\rho\overline{u}_3^{'}\overline{u}_1^{'} & -\rho\overline{u}_3^{'}\overline{u}_2^{'} & -\rho\overline{u}_3^{'2} \end{bmatrix} \tag{4.26}$$

4.4 Computational Simulation

A CFD simulation was implemented to obtain estimate the hydrodynamic parameters of the model. There are different types of test, each has its own assumptions and define certain coefficients. In summary the steps to perform a CFD simulation are:

1. **Domain and Geometry Definition:** This part consists of defining the domain on which the calculations are going to be made. The geometry of the space is defined by the element or surface with which the fluid interacts. The calculation domain is defined by a mesh and the size of this defines the precision and calculation time. Figure 4.3 shows how to use two sizes of meshes to optimize the time in which calculations are made without losing accuracy near the surface of an AUV.

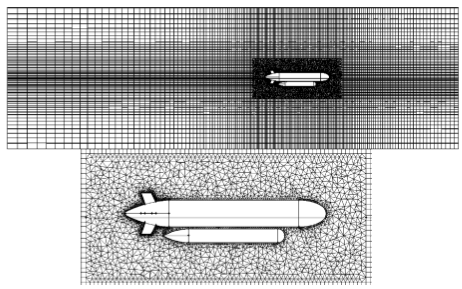


Figure 4.3. MESH AND DOMAIN FOR CALCULATIONS [14]

2. **Simulation Parameters Definition:** this consists of defining parameters of the fluid model, simulation parameters and indicating the boundary conditions of the surface.

3. **Numerical Results Calculations:** In this step, numerical methods are applied to obtain results, which depend on the simulation parameters configured previously.

4. **Results Analysis:** finally, the data are interpreted, either visually (Figure 4.4) or by summary tables, for the pressure and velocity of the fluid in the calculation domain. This part can also include the estimation of the hydrodynamic coefficients.

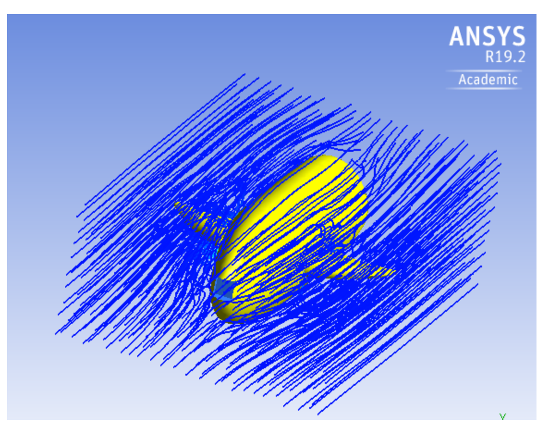


Figure 4.4. FLUID BEHAVIOR IN THE CALCULATION DOMAIN

There are several CFD applications to perform these simulations. The steps, mentioned above, can be performed separately or together. For this work, *ANSYS Fluent* was chosen to perform the simulation.

4.4.1 Geometry Definition

As mentioned before, the geometry defines the domain of the fluid and the surface of the vehicle. Three different tests were defined in section 4.2 and the geometry allow to perform these tests. *Space Claim* and *Meshing*, from *ANSYS Fluent*, were used to create the different geometries.

Straight Line Test

Here, the geometry is a tunnel surrounding the AUV. Two areas were created to have different resolutions in the domain. The result for this test is shown in Figure 4.5.

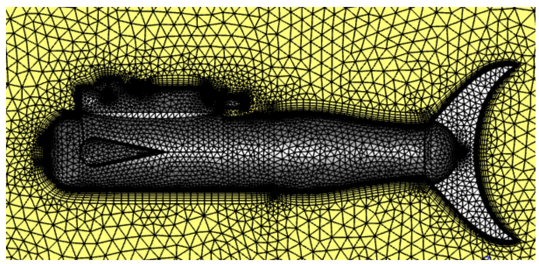


Figure 4.5. DOMAIN FOR STRAIGHT LINE TEST

Rotation Arm Test

Here, the geometry is a tunnel surrounding the AUV. Two areas were created to have different resolutions in the domain. The result for this test is shown in Figure 4.6.

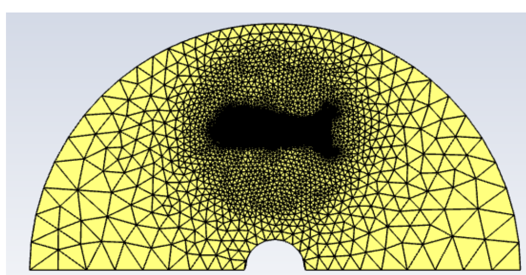
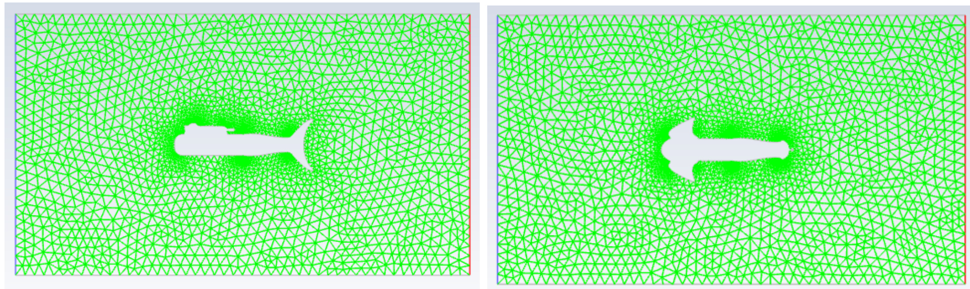


Figure 4.6. DOMAIN FOR ROTATION ARM TEST

PMM Test

Here, the geometry is a tunnel surrounding the AUV. Two areas were created to have different resolutions in the domain. The result for this test is shown in Figure 4.7.



(a) ZX PLANE **(b)** XY PLANE

Figure 4.7. DOMAINS FOR PMM TESTS

4.4.2 Fluid Parameters Definition

A way to simulate turbulence , as simplest and complete as possible, is with a two-equations model to determine the turbulent velocity and length scales independently. The $K - \epsilon$ model, presented in [22], is a semi-empirical model, based on transport equations for the turbulence kinetic energy (k) and its dissipation rate (ϵ). Ansys Fluent uses an improvement presented by [37]. In Figure 4.8 are shown the default values of the $K - \epsilon$ model for the simulation.

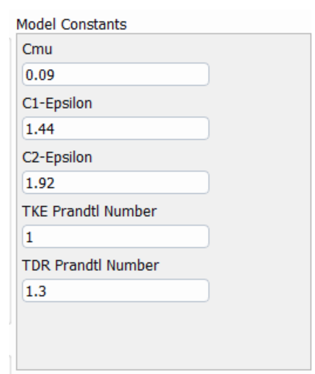
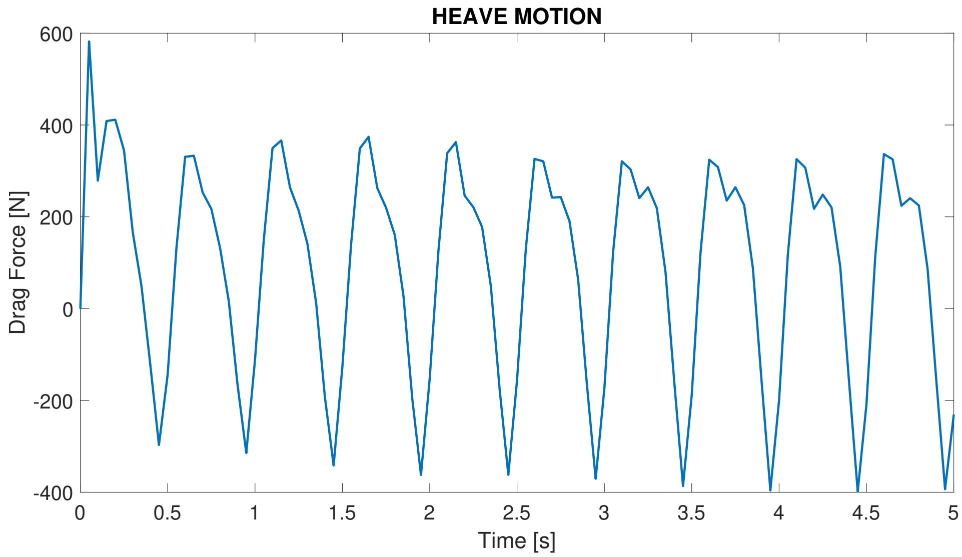


Figure 4.8. DEFAULT VALUES OF $K - \epsilon$ MODEL IN FLUENT

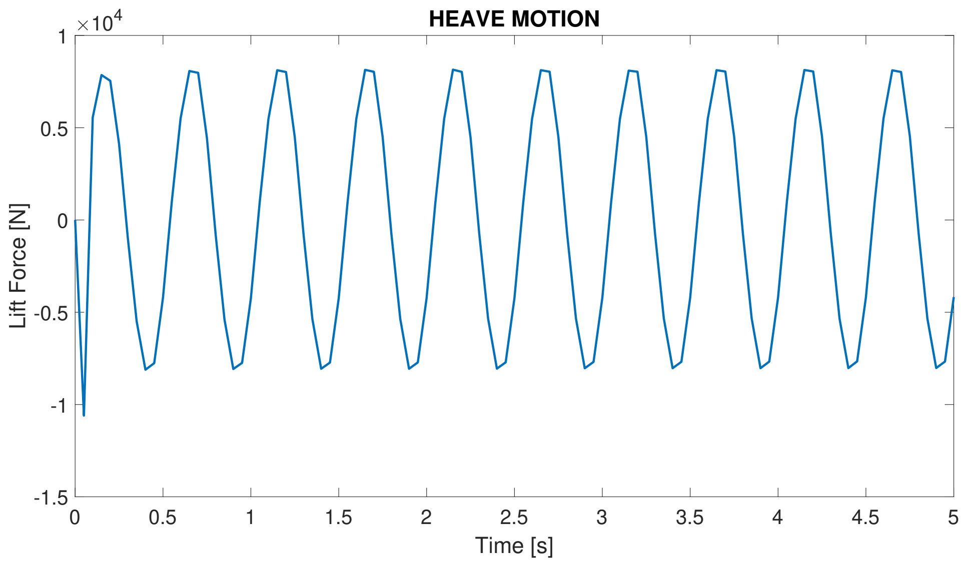
4.4.3 Numerical Results Calculation

Once the tests, previously mentioned, are simulated in a CFD software, the output data depends on the software used. Ansys Fluent provides an option to generate an output file with values of variables as: drag force and coefficient, lift force and coefficient, moments, flow velocity and so on. For the straight line test, the output data is variable calculated in each iteration. The PMM test generates a time-series data for each variable. The frequency report depends on the settings on the simulation and report interval.

In figure4.9, the results of calculation of drag and lift force are shown as a time series. These forces act on the surface of the BAUV. The value obtained is the resulting force in the corresponding axis. The moments calculated depend on the type of motio; the moment around Y axis for pitch and heave, and the moment around Z axis for yaw and sway. More graphics of the results, for each test, are presented in appendix A.



(a) DRAG FORCE FOR HEAVE MOTION



(b) LIFT FORCE FOR HEAVE MOTION

Figure 4.9. RESULTS OF 10 SECONDS SIMULATION

4.5 Mechanical Simulation in Simscape Multibody

The model can be used to apply a control algorithm. MATLAB® is a programming language that allow us to simulate the model as state equation. The BAUV is represented as a system with multiple inputs and multiple output. The system outputs are obtained by integration of equations of motion at a determine time. This required to know the initial conditions and values of the inputs at the time step. The function *ODE* can be used to solve the system.

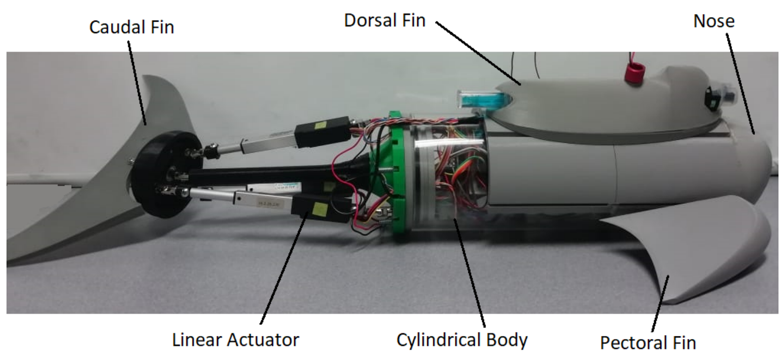
The code to simulate consist in functions that are called by a main program to simulate the time steps. The functions calculate values of buoyancy, propulsion (basic propeller), translation among frames and the main model. The results can be seen in section 6.1.1.

The system can also be simulated as a block for Simulink. Simscape Multibody™ (previously called SimMechanics™) provides a multibody simulation environment for 3D mechanical systems for MATLAB®. This software can model multibody systems using blocks that represent bodies, unions, constraints, force elements and sensors. Simscape Multibody formulates and solves the equations of motion for the complete mechanical system. The results are shown in section 6.1.1. There Figure 6.3 shows the BAUV model in Simscape Multibody. Also, a system can be provided to simulate a mechanism

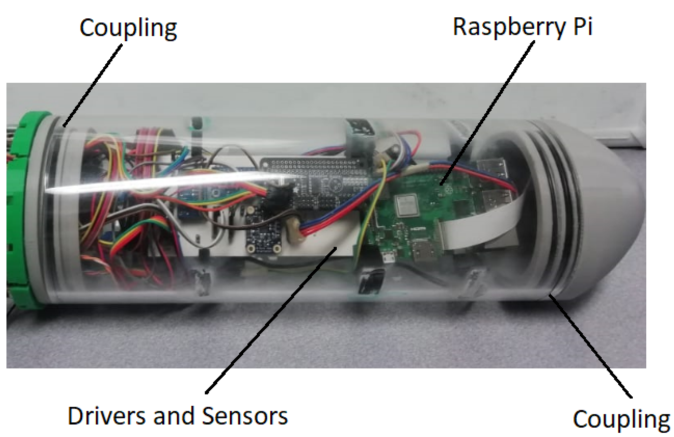
Chapter 5

Materials and Methods

The developments of the prototype involve a CAD, the manufacturing of the parts, the instrumentation of electronic components and the implementation of software to command the vehicle. Figure 5.1 shows the final prototype and the respective components on it. The design process, selection of components and implementation of software is detailed in the following sections.



(a) EXTERNAL COMPONENTS OF THE BAUV



(b) INTERNAL COMPONENTS OF THE BAUV

Figure 5.1. PROTOTYPE DEVELOPED AND COMPONENTS

5.1 Computer-Aided Design

A *Computer-Aided Design* (CAD) is the use of several graphic processing tools to develop a 3D draw from 2D images. Currently, there is a CAD design of the BAUV (Figure 5.2) developed in Fusion 360 software. This model is used in CFD simulations and allows a first view of the dimensions of the vehicle to be built. In appendix C, drawings of the design are presented with more detail.

Figure 5.2. BAUV CAD DESIGN DEVELOPED IN FUSION 360

5.1.1 Dorsal Fins

The dorsal fins will help to stabilize the BAUV. They have circular movement and are perpendicular to the hull side (XZ plane). Their rotation is generated by a servomotor and it is limited to $\pm 5°$. An assemble of a fin and the servomotor is show in Figure 5.3

Due to the necessity to accommodate the servomotors, the pectoral fins are thicker than the fins of a fish with thunniform propulsion. The fin was created by sweeping a profile along a path composed of two parabolic curves (Figure 5.4). The profile consists in a semi circumference join with an isosceles triangle at its base. Then a hole was implemented in the fin for the servomotor.

5.1.2 Hull Design

As mentioned in chapter 3, the hull is made up by 3 main sections: the nose is the front body, the middle body and the aft part that is part of the propulsion system.

Propulsion System

The propulsion system is a mechanism based on 3 linear actuator. Figure 5.5 shows the CAD design of the mechanism. The details and the development of the mechanism are presented on other work.

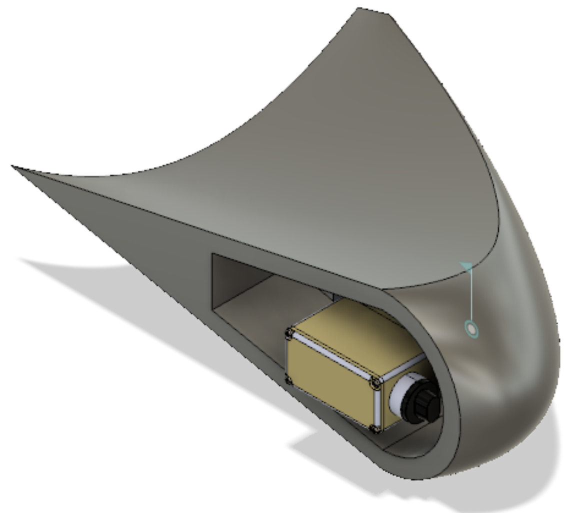


Figure 5.3. CAD ASSEMBLE OF THE DORSAL FIN

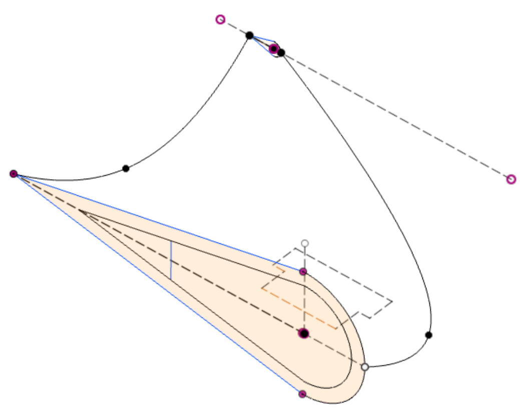


Figure 5.4. CAD SKETCH OF THE DORSAL FIN PROFILE

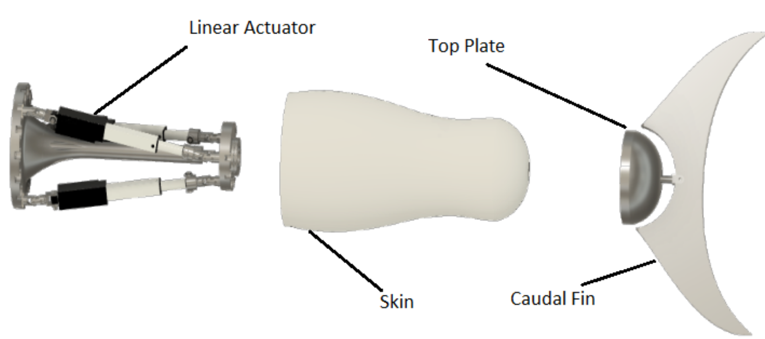


Figure 5.5. EXPLOID SCHEME OF THE PROPULSION SYSTEM

Most of the parts of the physical prototype are 3D printed. The cover for the parallel mechanism is designed to be a polydimethylsiloxane (PDMS), which is a flexible silicon that is waterproof and can stretch. The shape is achieved using a mold.

Middle Body

The parallel middle body consist in an acrylic cylinder (Figure 5.6) of $L = 240\,mm$ of length, $R_1 = 50\,mm$ of internal diameter and $R_2 = 57.15\,mm$ of external diameter. This will contain the main electronic components for control and power.

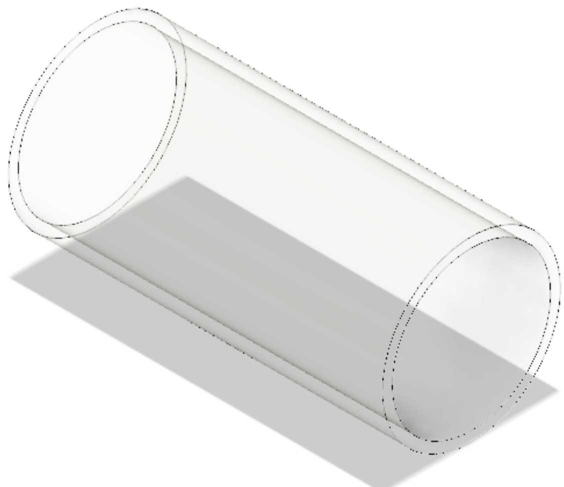


Figure 5.6. CAD DESIGN OF THE HULL

Nose Design

The nose (front body) was design considering the abilities to contain a camera and seal one of the ends of the middle body. The water impermeability is achieved with hydraulic seals of ring shape as shown in Figure 5.7.

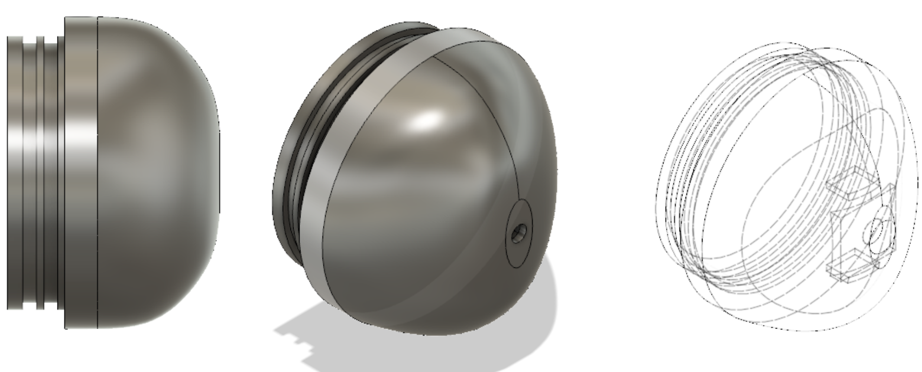


Figure 5.7. CAD DESIGN OF THE NOSE

Sensor Holder

This part (Figure 5.8) holds the two sensors of pH and turbidity, but also holds the dorsal fins though the servomotors. The design is based on a brace for the middle body and an upper appendage that holds the sensors. For the appendage, most of the faces are rounds to minimize the impact in the hydrodynamics of the hull.

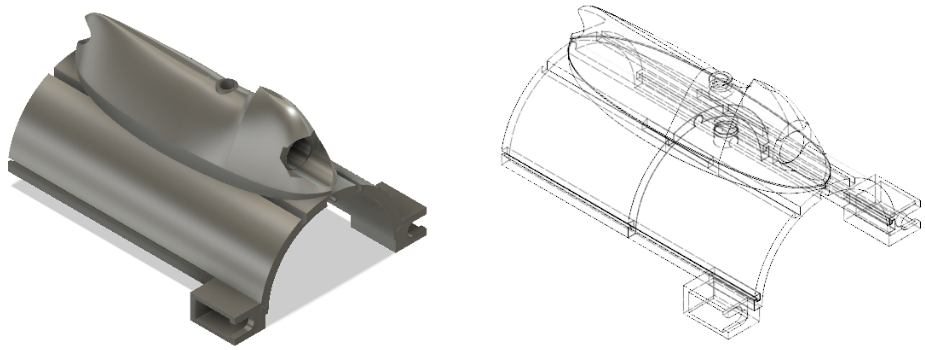


Figure 5.8. CAD ASSEMBLE OF THE SENSOR HOLDER

5.1.3 Couplings

Couplings were consider for the design of the BAUV. The most important coupling connects he middle body with the propulsion mechanism (aft body) and seals the union (Figure 5.9). Also, a similar design is embedded in the nose with the same objective.

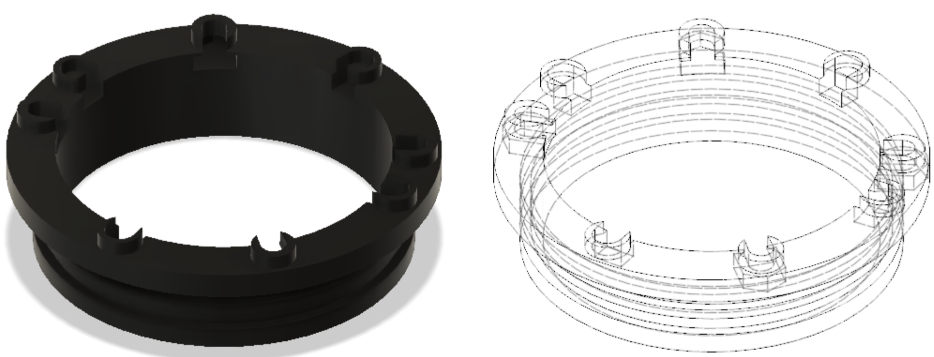


Figure 5.9. CAD DESIGN OF THE BACK COUPLING

The coupling is created as a revolution body from a sketch, as in Figure 5.10, around the axis of the middle body. This profile matches the inner walls of the cylinder and has two slots. The slots objective is containing two rubber bands that will put pressure against the inner walls, sealing the union.

5.1.4 Supports

The supports consist in the parts that holds the inner circuit boards. These are the single-board computer raspberry pi and main connectivity board holders (Figure 5.11a) and the voltage regulators holder (Figure 5.11b). All these pieces were 3D printed with PLA filament.

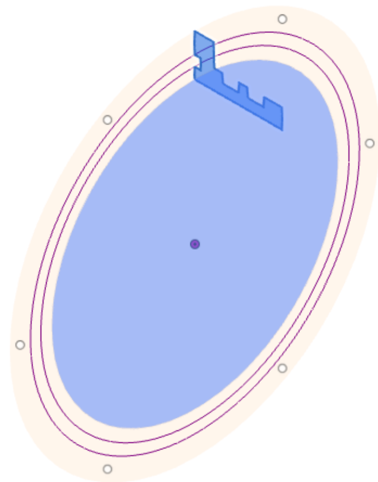
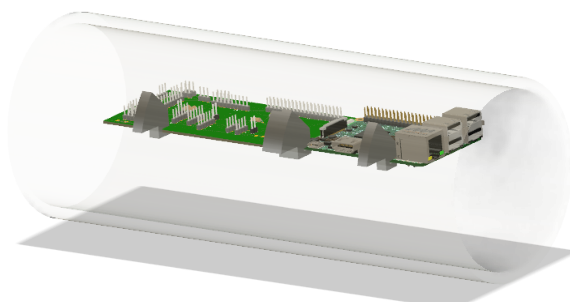
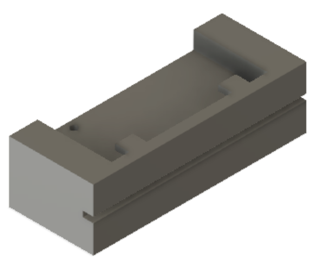


Figure 5.10. PROFILE OF THE COUPLING MECHANISM



(a) CIRCUITS BOARDS HOLDERS



(b) VOLTAGE REGULATOR HOLDER

Figure 5.11. SUPPORTS DESIGNS

5.1.5 Battery

The battery can be used as a mobile weight to change the center of mass. This allows to have an adaptable buoyancy effect. This is achieved with a rail under the battery. A first approach is presented in Figure 5.11a.

The mechanism has no electric part, so the mobile center of mass is set when the BAUV is assembled. All the support components are glued to the cylinder walls to avoid drilling the structure and making more impermeability sections.

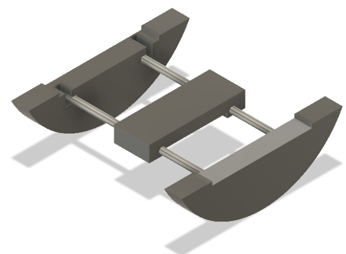


Figure 5.12. MECHANISM TO MOVE THE BATTERY

5.2 Instrumentation

The electrical system provides energy from the vehicle's battery to the propulsion system, the sensors, the control system and other actuators. Also, the main controller should be selected to minimize energy consumption, provide the necessary calculation capacity and integrate with other components. A diagram of the hardware architecture is shown in Figure 5.13.

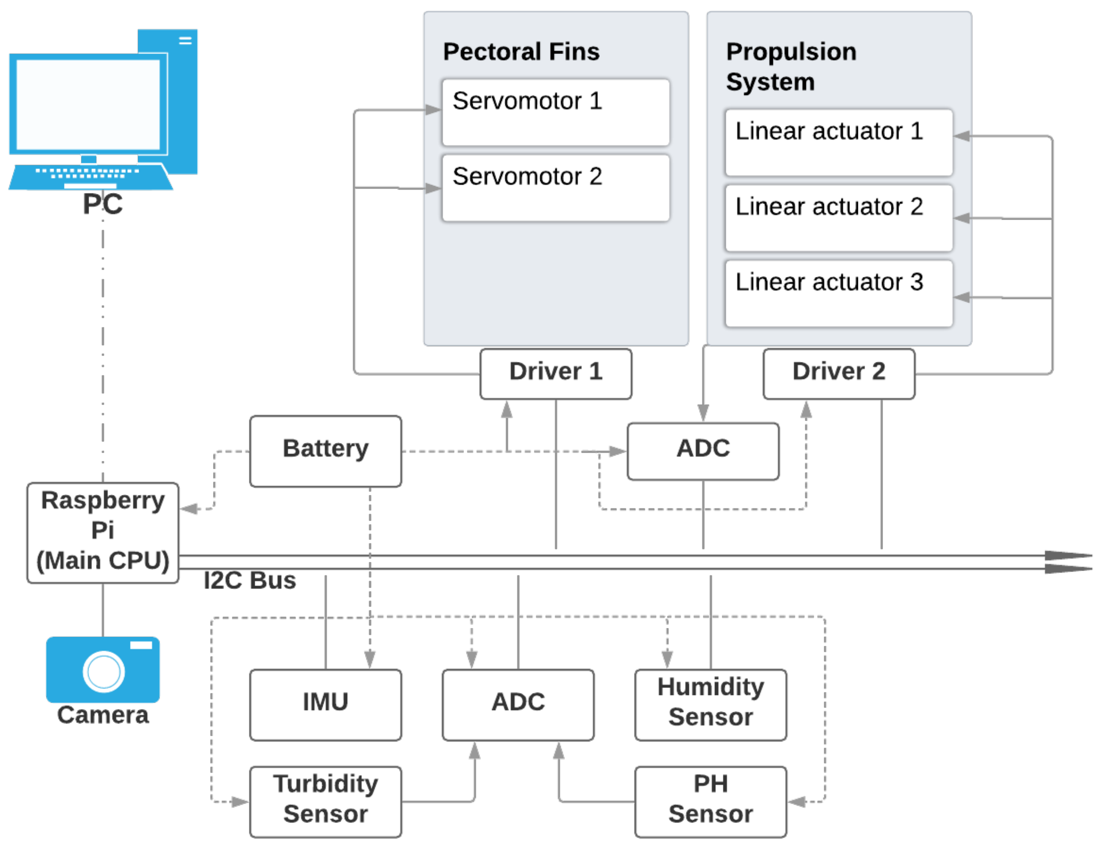


Figure 5.13. BAUV ARCHITECTURE DIAGRAM

5.2.1 Energy Source

The energy sources used for submarine robots include batteries, accumulators, fuel cells, among others. The batteries are composed of electro-chemical cells that store energy; they are also classified in primary and secondary, being the last ones, more used in AUVs [23].

Table 5.1. COMPARISON OF THE DIFFERENT TYPES OF BATTERIES [23]

Element	**Energy density (Whr/Kg)**	**Charge cycles**
Lead-Acid	31.5	~ 100
Ni-Cd	33	~ 100
Ni-Zn	58.5	~ 500
Li-Ion	144	~ 500
Li-Polymer	193	~ 500

5.2.2 Central Processing

The central processing will eventually be handled by an on-board computer responsible for the main control processes of the system. Table 5.2 shows the options for devices to perform central processing.

Table 5.2. CPUS SUMMARY FOR CENTRAL PROCESSING

Platform	**Model**	**CPU**	**Ports I/O**	**Ports A/D**	**OS**
Arduino	Mega	ATmega2560	54	10 input	No
	Zero	ATSAMD21G18 32-Bit ARM	20	6 input 1 output	No
Raspberry	B+	Broadcom BCM2837B0 64-bit ARMv8	40 GPIO	0	Raspbian
	Zero W	1GHz, single-core CPU	40 GPIO	0	Raspbian

The two platforms have important differences: the first is that Arduino has analog ports, which are practical for the use of sensors, but in the case of Raspberry requires additional devices to fulfill this function. The second difference is the Raspbian operating system, which is a version of Linux, which provides greater versatility in communications, programming and greater ease of connection to other types of plug and play devices. The selection of the device for the main processing depends on the calculation capacity required for the control and the type of wireless communications to be implemented.

5.2.3 Actuators

The actuators for the BAUV are the propulsion system (caudal fin) and the steering/tilting system provided by the pectoral fins.

1. **Propulsion**

   The propulsion is through the oscillation of the tail fin, this is achieved by a parallel mechanism of linear actuators (Figure 5.5). The base of operation are 3 *Miniature Linear Motion Series L16* (Figure 5.14), which are coordinated to produce the desired oscillations. These motos can push and pull along their full stroke length. Also, the model P has feedback of the position. They can work with an input voltage up to 15 VDC.

Figure 5.14. LINEAR ACTUATOR L16

2. **steering/tilting**

   These movements are achieved through the control of the pectoral fins and certain movements of the caudal fin. Then rigid fins are proposed whose inclination is controlled by servomotors.

5.2.4 Sensors and Signal acquisition

The sensors of the BAUV depend on the activities to be carried out. Since the main objective is the study of the dynamics, energy consumption and variables related to the mission of the BAUV. Other devices as *Analog to Digital Converter*s (ADCs) and *Digital to Analogue Converter*s (DACs) are used to acquire or generated signals.

Inertial Measurement Unit

The *Inertial Measurement Unit* (IMU) is an electronic device that measures velocity, orientation and gravitational variables. This usually integrate an accelerometer, gyroscope and magnetometer. The *BNO055* is a *System in Package* (SiP), integrating the previously mentioned devices. Each device has different resolution: 14 bits on the accelerometer and 16 bits for the gyroscope and magnetometer. The devices are triaxial and are integrated with a 32-bit Cortex-MO+ microcontroller. The Adafruit BNO055 Absolute Orientation Sensor, shown in Figure 5.15, uses the *BNO055* SiP and allows a easy handling of the measured data.

The *BNO055* is integrated with bidirectional I^2C and *UART* interfaces for integration with other systems. This device has different types of operation modes, ranges of functionality, basic low pass filters and different power modes. The manufacturer suggests its use in applications such as: navigation, robotics, augmented reality and fitness and well-being.

Humidity Sensor

Relative humidity is the ratio of partial pressure of water vapor to the equilibrium vapor pressure of water at a given temperature. The *AM2320* (Figure 5.16) provides measures of temperature and relative humidity. The communication is through I^2C protocol.

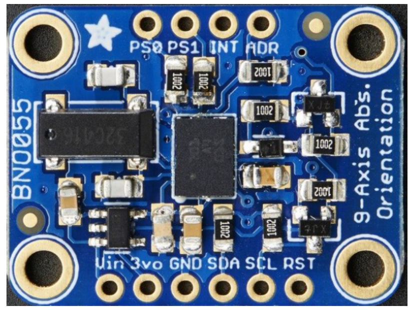


Figure 5.15. ADAFRUIT BNO055 ABSOLUTE ORIENTATION SENSOR

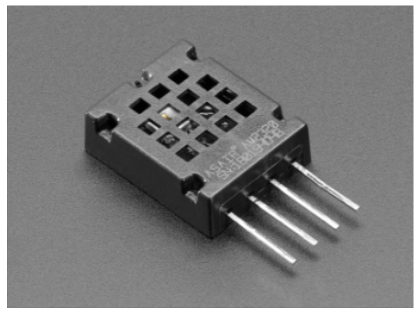


Figure 5.16. DIGITAL TEMPERATURE AND HUMIDITY SENSOR

PH Meter

In chemistry, pH is a scale to indicate the acidity and alkalinity of water-based solution. The Figure 5.17 shows the *SEN0161*, an analog pH sensor with a voltage regulator and a filter at the output. The objective of the sensor is testing the quality of water in water bodies, rivers and lakes.

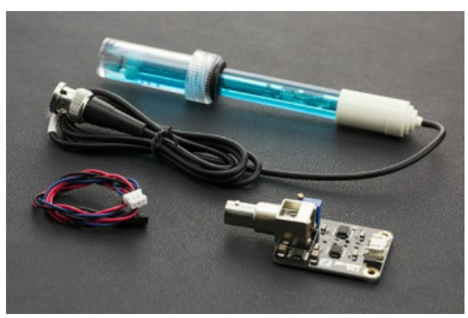


Figure 5.17. ANALOG PH SENSOR/METER KIT

The sensor requires a two points standard calibration, this are (4.0 and 7.0). The power supply is required to be between 3.3 and 5.5 V. The output varies from 0 to 3.0 V and has an accuracy of ± 0.1 at 25ºC.

Turbidity Sensor

The $TDS - 10$ module (Figure 5.18) measures the number of suspended particles (turbidity) in water. This is an optical sensor that measures turbidity using the refraction of wavelength between photo transistor and diode. The principle of operation of the sensor is that when the amount passed through a sample of water depends on the amount of soil. Hence the amount of light is inversely proportional to the amount of soil in water.

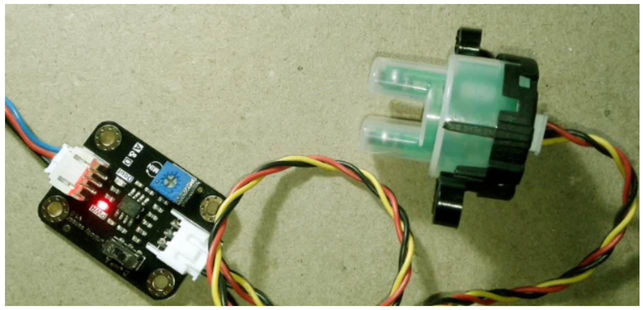


Figure 5.18. TURBIDITY SENSOR TSD-10

Current Sensor

Power can be used to determine the time of operation for a limited power source. An sensor Allegro™ACS712 provides a way to sense the current for AC or DC circuits. This sensor requires a 5V power supply and has a sensitivity of 66 to 185 mV/A at the output. The sensing can be improved with the use of the filter pin. The sensor has electric isolation between the conductive path and the signal leads; hence the sensor can be used in applications that usually requires opto-electric isolation.

Analog to Digital Converter (ADC)

An ADC is a device that converts an analog signal to digital so it can be processed. The BAUV has different signals that required conversion to be handled. The ADS1115, show in Figure 5.19, is a precision ADC with 16 bits of resolution and 4 channels in a small package. It has an onboard reference and oscillator. The data from its channels is transmitted through I^2C with four selectable slave addresses to work along other ADS1115 devices. The channels can be use as single ended or differential. It requires a power supply of 2.0V to 5.5V. It can acquire data at 8 programmable data rates going from 8 *Samples per Second* (SPS) up to 860 SPS.

5.2.5 Drivers

A driver controls the operation of different devices from a control signal provided by a main CPU. This device can handle different scales of current according the requirement of the actuator. This featured is useful to isolate a control circuit from a power circuit. The actuators of the BAUV are motors, so the driver helps with the handling of the high currents needed. In the Figure 5.20

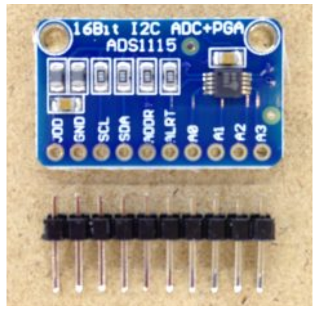


Figure 5.19. ADAFRUIT ADS115 ADC

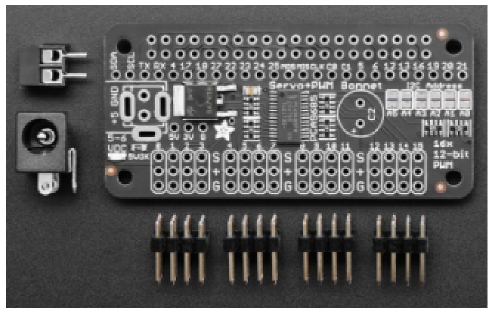


Figure 5.20. ADAFRUIT 16-CHANNEL PWM/SERVO HAT

In the Figure 5.20 can be seen the Adafruit 16-Channel PWM/Servo HAT. This module has 16 channels to handle servomotors with a 12-bit precision. The driver can handle motors powered by 5V and 3.3V logic signals. This driver is based on the PCA9685 chip controller. It can generate a *Pulse Width Modulated* (PWM) signal up to 1.6 kHz, all running independently. The logic signal is provided through a I^2C interface and it has 62 different slave addresses.

5.2.6 Communications

The type of communication used varies according to the development phase of the vehicle, some options were revied in Table 2.1. First for the testing phase, communication with a WiFi and serial interface is used for dry and water tests respectively. The communication interfaces define the communication protocol, either TCP/IP for WiFi or asynchronous communication for serial. In future work, an interface will be sought that adapts to the resources of the project and allows to communicate with the vehicle in the water.

5.2.7 Printed Circuit Boards

In order to integrate the different actuators and sensor that the BAUV will have, a circuit board was designed to provide connectivity and power distribution. The design was done on Eagle. It has a schematic and a PCB design. The schematic shows the interconnection between boards.

The PCB design shows the parts distribution and the routing of each connection. The routing is done according to the amount of current that should be handled. The main power lines must be wider than the lines that handle communications, but the lines that provide

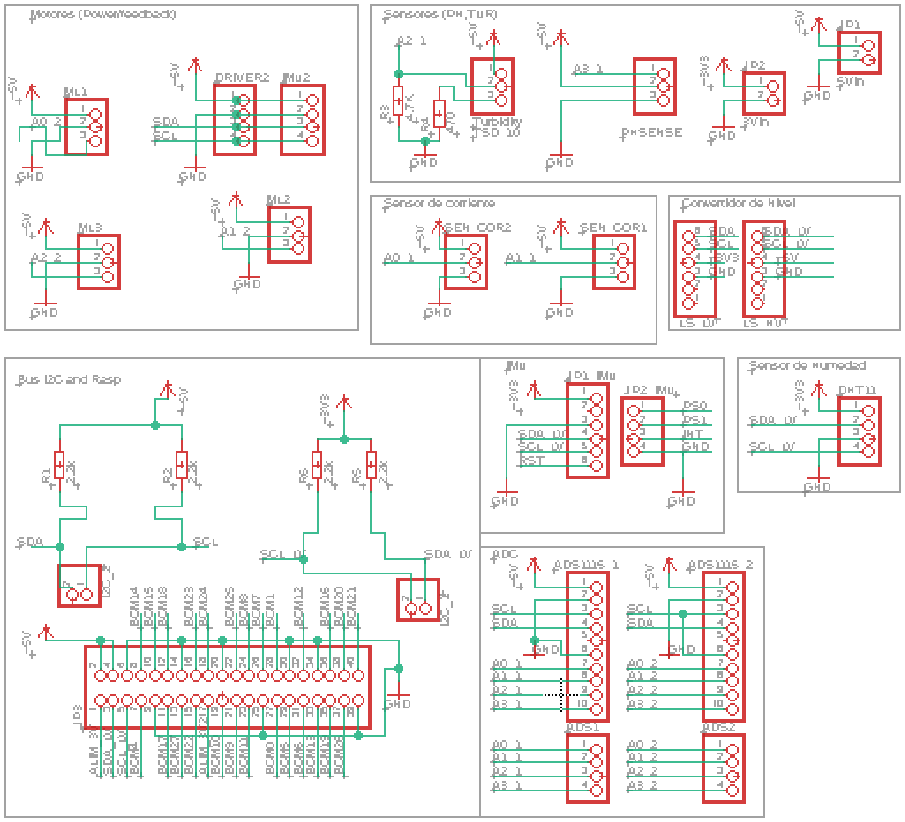


Figure 5.21. SCHEMATIC DESIGN OF ELECTRONICS SYSTEMS IN BAUV

power to motors should be capable of handle at least 2 amperes. The distribution of the parts has no special reason other than reduce the amount of vias of the board.

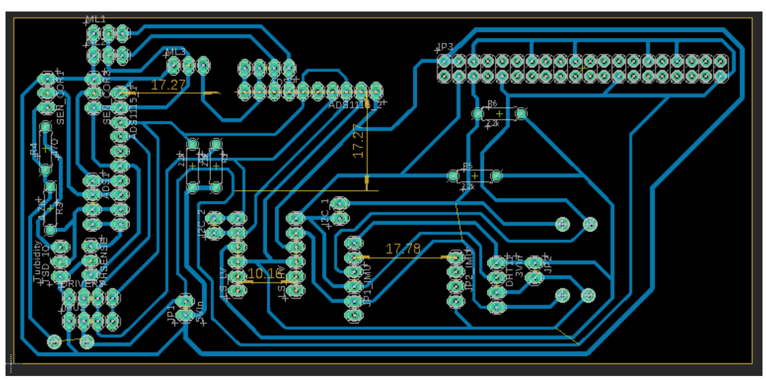


Figure 5.22. BOARD DESIGN TO CONNECT THE ELECTRONIC COMPONENTS

5.3 Software

The software development is performed by using Python language, given that the main board is Raspberry pi 3B+. Python is an open source programming language of high-level. This language is defined as multi-paradigm, this mean it suits to different programming styles.

The software will have a server-client architecture (Figure 5.23), with the BAUV single-board computer acting as the and a laptop computer (station) being the client. This architecture is used during the testing period, but in the future, it will change to allow the BAUV to be more autonomous. In the testing period a computer is used to monitor the data from the different sensors that allow to know the BAUV status. For this period the communication between the BAUV and the computer is performed by using WiFi and serial communication for some of the tests.

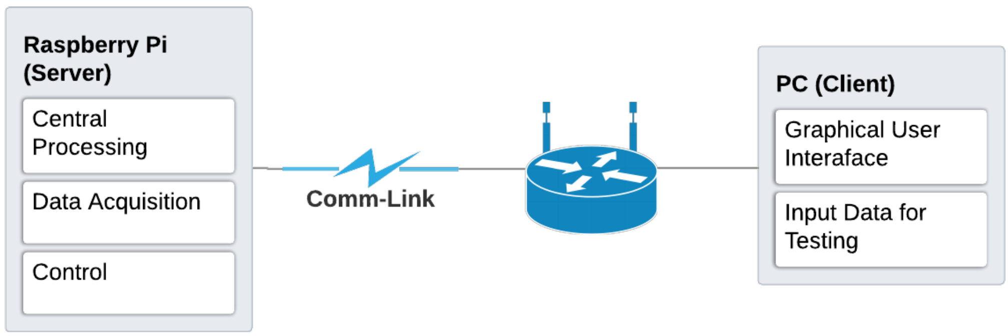


Figure 5.23. ARCHITECTURE DIAGRAM OF THE BAUV SOFTWARE

The software design of the system includes two scripts. The first script will run in the Raspberry pi on board the BAUV, and will work as the server. The second will run in the computer on land as a client. The main processing is done in the Raspberry pi; the client allows to monitor and change the values of the main actuators.

A graphical user interface is implemented on the client. This interface shows the values of the sensors in real time, also allows to change the values of the actuators. Later, this interface will change depending on the type of control that will be implemented in the BAUV. A more complete explanation of the software, implemented in Python, is shown in appendix B.

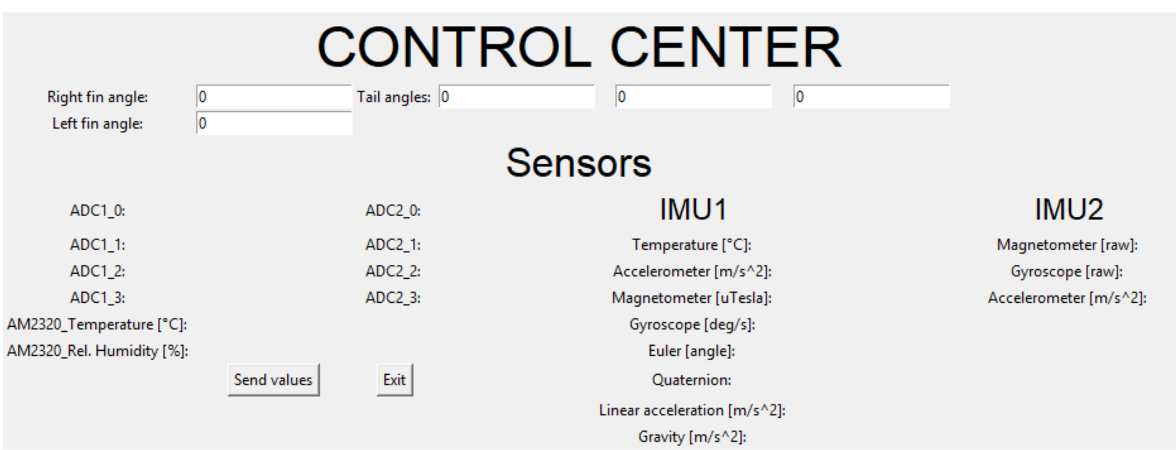


Figure 5.24. GRAPHICAL INTERFACE ON THE CLIENT SCRIPT

Chapter 6

Results and Conclusions

6.1 Simulation Results

6.1.1 Model and Hydrodynamics

With the data from the CFD simulations, some of the normalize coefficients are calculated. These are show in table 6.1. The coefficients are calculated by regression analysis using equations (4.14) for straight line test, normalized version of equations (4.17), (4.19), (4.21) and (4.23). These coefficients are estimated for a viscose laminar fluid. Other estimations can be done for other scenarios.

Table 6.1. ESTIMATED HYDRODYNAMIC COEFFICIENTS

Coefficient	Estimated value	Coefficient	Estimated value
$Z'_{\dot{w}}$	-0.0522E04	$M'_{\dot{w}}$	-0.0746E04
Z'_{w}	-0.9044E04	M'_{w}	1.292E04
$Z'_{w\|w\|}$	2.7662E04	$M'_{w\|w\|}$	3.9517E04
$Z'_{\|w\|}$	-0.0899E04	$M'_{\|w\|}$	-0.1284E04
Z'_{ww}	3.1957E04	M'_{ww}	4.5653E04
Z'_{q}	-0.2373	$M'_{\dot{q}}$	-0.3390
$Z'_{\dot{q}}$	-16.2668	$M'_{q\|q\|}$	-0.9451
$Y'_{\dot{v}}$	-0.3553E03	M'_{q}	-19.6976
Y'_{v}	-5.8686E03	$N'_{\dot{v}}$	-0.5076E03
$Y'_{v\|v\|}$	-3.1556E03	N'_{v}	-8.3837E03
Y'_{r}	0.1675	$N'_{v\|v\|}$	-4.5079E03
$Y'_{\dot{r}}$	-3.7102	$N'_{\dot{r}}$	0.2393
$N'_{r\|r\|}$	-0.2581	N'_{r}	-4.3336

Also, a curve for the behavior of the drag and lift coefficients of the fin was obtained (Figure 6.1). These values are used to calculate the lift and drag effect of the fins using equations (4.9). The values of the lateral area for equations (4.9) are obtained from the CAD and the value is $9.032223 \times 10^{-3}\ m^2$.

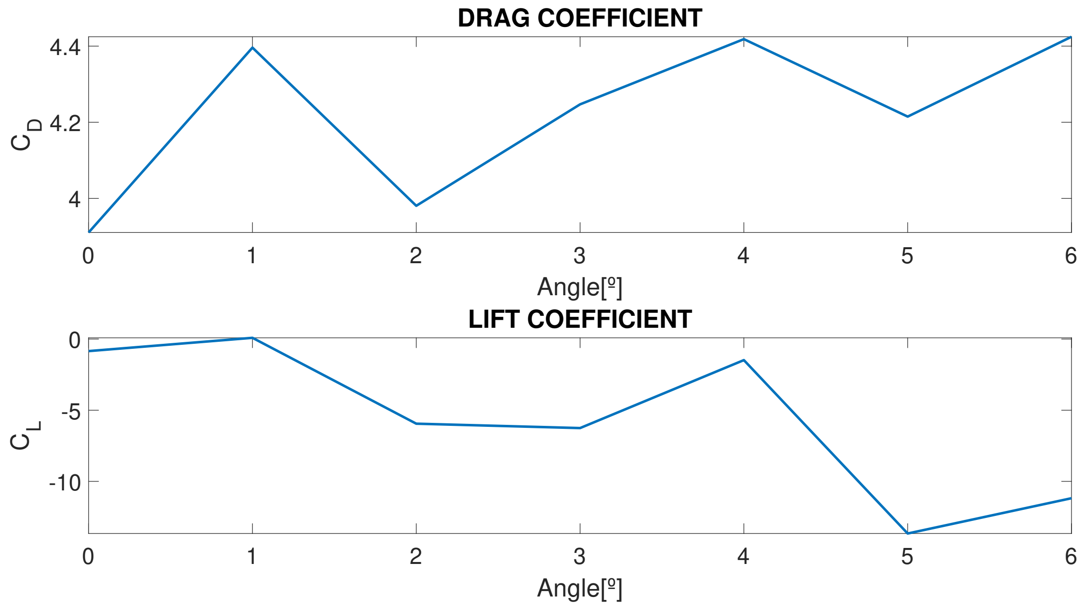


Figure 6.1. DRAG AND LIFT COEFFICIENTS FOR THE FIN

At the same time, a mathematical model was developed based on the general equations for AUVs's motion (4.4). This model requires a hydrodynamic analysis for the body geometry effects in the motion.

With the equations of motion, now we can simulate the behavior of the BAUV. This model is solved as state equations to which we apply an integrator. The results of a 10 seconds simulation can be seen in Figure 6.2. This simulation has:

- 0.5 seconds of sample time,
- No propulsion force applied,
- centers of mass and buoyancy aligned with a separation of 5 mm in Z axis.

The data shown in Figure 6.2 correspond to the three traslational degrees of freedom (ζ, η, ξ) at the left and the three rotational degrees of freedom (ψ, Φ, θ) at the right. Several cases were test, shown in Appendix D, where the center of mass is displaced to the left or right in the X axis by a range of 2 cm; the behaviour on the traslational components shows that the BAUV sinks in a range of 10^{-3} and had a displacement in X withing the same range. Only the pitch angle ϕ is affected in the rotational components in both cases. When the center of mass is aligned with the center of buoyancy the vehicle only sinks. These results are shown in Figures D.1, D.2 and D.3.

With a propulsion force is provided, the behaviour of the vehicle change drastically. First, when the centers of mass and buoyancy are aligned in the Z axis, the vehicle moves in the X axis as expected but has slight displacements in the other axes (Figure 6.2). Only a roll motion is observed in the rotational components. The other rotational components are affected when the centers are not aligned, and these cases are shown in Figures D.5 and D.6.

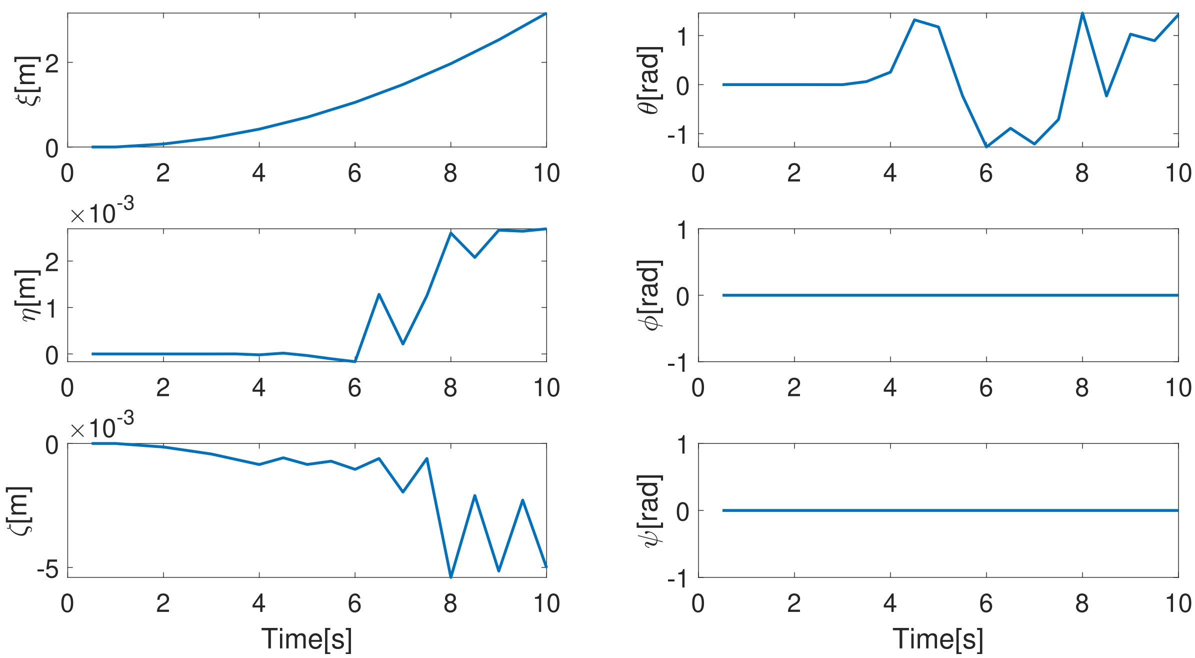


Figure 6.2. MODEL POSITION AND ATTITUDE THROUGH 10 S

With the data generated previously, now we can visualize the behavior of the BAUV in a simulation on Simscape (Figure 6.3). This simulation allowsallows to observe more graphically the movement of the vehicle.

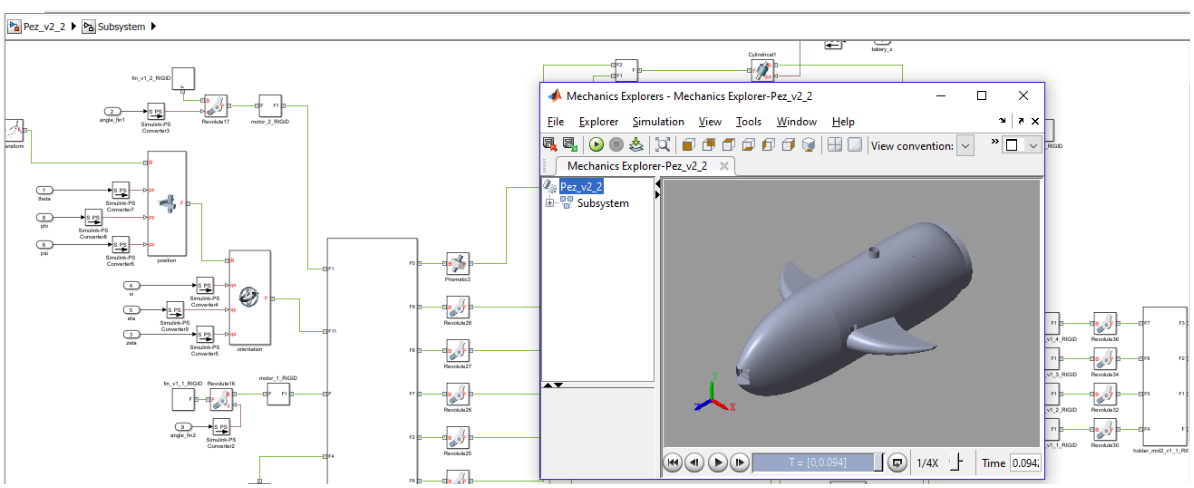


Figure 6.3. MECHANICAL MODEL OF A BAUV IN SIMSCAPE

6.1.2 Prototype

The software used to design the vehicle and its parts was Fusion 360. This software was easy to use, and a license was provided by the university. This part presents no complications but has no easy integration with other software and that can be troubling for the optimization process. However, the software assists in calculations such as the center of mass and buoyancy (Figure 6.4).

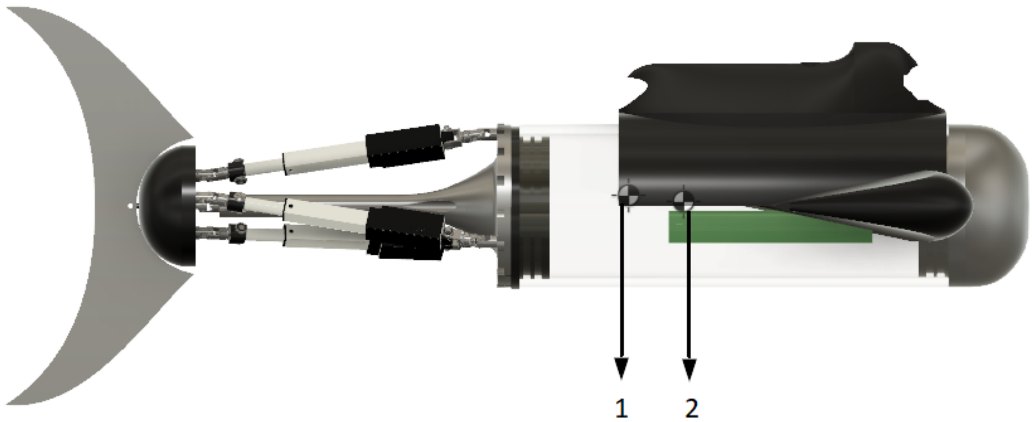


Figure 6.4. ESTIMATION OF CENTERS OF MASS (1) AND BUOYANCY (2)

The center of mass is calculated with the complete assembly, but for the center of buoyancy the displaced volume is required. This was achieved by creating a solid with the external surface of the BAUV. In Figure 6.4, the two centers are shown. As said in section 3.1, it is required the alignment of the two points along the Z axis with the center of mass below the center of buoyancy. This tell us that additional weight is required or that other battery can added. Other tests proved that a displacement of the battery along X axis produce a displacement in the center of mass by a factor of $1/5$. All these calculations are merely an estimation and can vary with the results of the final prototype. However, this estimation helps to know a priori the effect of the buoyancy. The final prototype is shown in Figure 6.5.

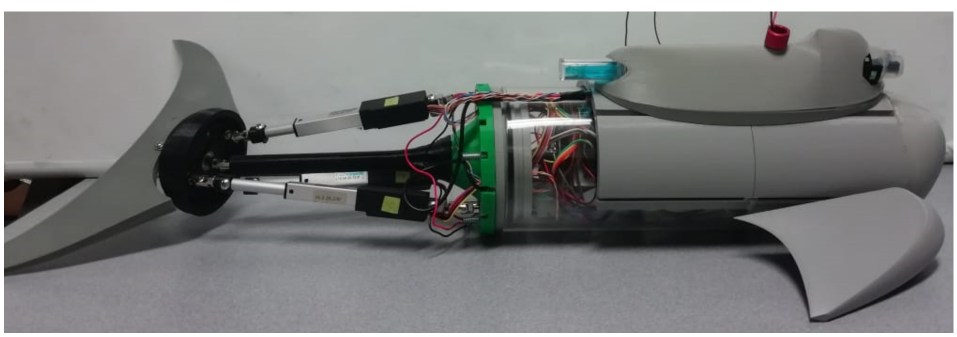


Figure 6.5. FINAL BAUV PROTOTYPE

As mentioned in Chapter 2 and shown in Table 6.2, the BAUV takes features from the classic AUVs. For example, the cylindrical hull facilitates the manufacturing process. Also, some features, as the fins, add benefits like maneuverability and integration with a marine environment. Finally, the vectorized thruster is a new way of propulsion for BAUV.

Table 6.2. FEATURE COMPARISON WITH CLASSIC UVS

	Features					
AUVs	**Body**	**Pectoral Fins**	**Autonomy**	**Power Source**	**Vectorized Thruster**	**Integration with environment**
Classic	Rigid	No	IAUV	Fuel and Batteries	No	No
Proposed	Rigid	Yes	AUV	Battery	Yes	Yes

6.2 Conclusions

6.2.1 Mechanical Design and Development

The design and development of the BAUV follows the methodology mentioned in 1.6. First, we design the hull, fins, and internal components. Then we implement the designs and make corrections. Each stage had its own complications and different assumptions had to be made. Some of these are:

- Changes in the geometry due to sensors and manufacture issues.
- Selection of components with low energy consumption.
- Limit the motions that the BAUV will have.
- Limit the hydrodynamic analysis for the PMM tests due to the size of meshing and computational calculation required.

Geometry

The geometry its intended to mimic a fish body, but the resources can limit how faithful the vehicle mimics the fish geometry. This means that actuators and sensors also define the geometry. For this work, a tuna was the fish to mimic, so the body had to be stiff, had elliptical cross section and relatively thin fins.

At the end, the main body was a cylinder with an elliptical fore section and a parabolic aft section. A cylindrical body was selected for its easy fabrication; This is because its 3D printing, in one piece, required a higher capacity printer or therefore dissecting the cylinder into smaller parts. A greater number of parts involve more waterproof walls that must be sealed. The actuators and sensor change this geometry adding a dorsal fin and making the pectoral fin thicker. This change the ideal drag that the vehicle will have. This modification was made due to the need for sensors to be in contact with water.

Appendages

Appendages are an important for UVs, because as its mentioned in [24], they can help with stability but also, they add drag to the model. Ideally the tuna fins had little effect in the model but given the required space for motor and the sensors, they can be considered as appendages.

The pectoral fins can be used to add maneuverability to the model, but for future works the dorsal fins must be thinner: This implies the use of other type of sensors or have a better way to embed them in the geometry with less increase of the drag. The drag coefficient has a value of 1.15 for the complete design and a value of 0.8 without the appendages. This means that indeed an increase of drag occurs when we add the appendages.

Impermeability

The complexity in the manufacture involves the numbers of impermeable walls for the vehicle. As mentioned in 3.5, this vehicle has 4 impermeable walls. Two of them are in the unions of the body sections and they are more important because the internal components need to be dry.

The method selected to seal these walls eliminates the need of nuts and bolts but add complexity in the disassembled of the vehicle. The main factor that influence the selection of this method was the easy manufacturing of the parts involved.

Optimization

The optimization of a process can be done by modifying the different parameters of the design. This optimization can aim to reduce an effect (drag) depending on variables as dimension and the cross section of a vehicle. In sections 2.1 and 3.6 are presented ways to parameterize and optimize the design of an AUV based on widely studied models of geometries for submarines. The problem with BAUVs is the complexity in the geometry, however it can be adapted on one of these models mentioned.

In this work, the biggest challenge was the hydrodynamic study that provides an estimation of the drag coefficient of the design. The software used does not allow to implement an efficient way to systematize this process. The optimization process seeks for parameters that minimize the drag coefficient, but this is done by brute force. A solution, for future works, it's to use a software with low level programming language. This comes in hand with a good parameterization of the BAUV's geometry, also a software that can be integrated with others is required. Three types of software can be involved: the first to design the geometry (Fusion, SolidWorks), other for the hydrodynamic analysis (Ansys package) and the last one to integrate the others and run an optimization algorithm (Matlab, Python).

Instrumentation

All the electronic components need a main processor for control and communication with each other. The raspberry pi 3 $B+$ has this task, but only has digital I/O ports and requires additional ADCs. The internal components instrumentation can be classified in three categories: power source, control circuits and power circuits.

A Lipo battery was selected as power source given the required autonomy of the vehicle. The battery features depend on the required current for the other electric components and the time of autonomy required for a task. The total charge of the components, provided by the datasheets, can give an estimation of the time that the battery will handle. A 5000 mah battery can handle 3 small linear servomotors, 2 micro-servos, an IMU, 2 ADCs, 2 drivers, 3 sensor and a raspberry pi for about 15 minutes. Also, for a testing stage, the power source can be external to the BAUV.

The control circuits are the main processor, the drivers and sensors involved. They usually have a low current drain and can be set with a low-power mode. The drivers can handle high currents, but their consumption is low. These components use I^2C protocol and for communication with an external computer they can use either TCP/IP or serial protocol. For the testing stage, serial protocol is used to communicate and control from another computer.

The actuators are the main current draw in the BAUV. Three linear servomotors are used for the propulsion system, with a consumption of 1.2A at maximum load each. The fins current draw is 1.8A at stall. The last motors had to be waterproof given the position where they are.

In overall, the selection of electronic components follows the time requirements for the BAUV's task. They can vary on its features, but they can be also limited as the power source. it is necessary to reach a common agreement between the time of autonomy and the weight required for the batteries.

Software

The software of the vehicle controls its behavior. Given the platform used as main processor, the programming language used was Python. Section 5.3 details the software developed. The language used has a lot of support online, so the software development does not represent a complication for the overall design of the vehicle.

The result is shown in Figure 6.5. This prototype was tested on dry environment, but the sealing mechanism was previously tested so it's expected to work properly. Most of the components were 3D printed with an Ender 3 printer, so the main cost relies on the electronic components. A problem with the material is that it can bend during the printing and the quality relies on the printers features and calibration

6.2.2 Simulations Analisys

The behavior of the BAUV seems to be the adequate. Now a control algorithm can be applied, and further analysis can be obtained. Also, the effect of the propulsion mechanism needs to be integrated to the system and a more complete model can be achieved. From the results shown in D, we can see the effect of moving the center of mass, hence we can use this to control how fast the BAUV sinks without using a propulsion force. The hydrodynamic study of the geometry allows us to have a personalized model for the BAUV and only required a new estimation of parameters if the design is changed.

6.3 Future Works

The entire development of the BAUV was divided by stages. Each stage has its own complications, some of them can be easily fix but other limited the efficiency of the development. To avoid this, the following recommendations should be taken into account for future work:

- The geometry of the body could be based on submarine body models more widely studied. This would affect how much the BAUV's body mimic a fish body, but for these shapes can be more easily parameterized. This also means having less appendages with high drag coefficient.

- At the beginning the weight was a problem given the relation it has with the propulsion system, but a CAD can be effective for weight estimation. For this works, the body was cylindrical due to manufacturing limitations related to the capacity of the 3D printer. This limitations and ease of assembly caused the 3D printing to occur in parts. In addition, certain parts such as the brace and the sensors can be integrated in the middle body to avoid creating appendages. The problem with dividing the parts for 3D printing relies in the creation of new impermeable walls that need to be sealed. A solution could be use bigger printers or the reduction of the size of the hull.

- The CAD is very powerful tool. Allows to know a priori certain variables as weight, length, width, depth, center of mass and buoyancy. A problem with the software used is that does not allow the integration with other software and this is a problem to execute an optimization process. The only solution is either change the CAD software or the main software used for optimization.

- The model can be improved when needed. A new hydrodynamic analysis its recommended to include roll. Also, a 3D hydrodynamic analysis would give a wider perspective of the geometry effects in the motion.

Acronyms

ADC *Analog to Digital Converter*. 1, 43, 45, 54, 71

AUV *Autonomous Underwater Vehicle*. v, 1–3, 6, 8, 10, 16, 19–24, 26, 27, 30, 31, 42, 50, 52, 54

BAUV *Biomimetic Autonomous Underwater Vehicle*. 1, 3–6, 9, 12, 16–18, 21, 24, 25, 32–34, 36, 39, 40, 42, 43, 45, 46, 48, 50–56, 69, 77, 79

BCF *Body Caudal Fin*. 1, 6

CAD *Computer-Aided Design*. 1, 36, 56

CFD *Computational Fluid Dynamics*. 1, 25, 29–32, 36, 49, 61

CoB *Center of buoyancy*. 1

CoM *Center of mass*. 1

DAC *Digital to Analogue Converter*. 1, 43

IAUV *Intervention Autonomous Underwater Vehicle*. 1

IHSS *Iranian Hydrodynamic Series of Submarines*. 1, 7

IMU *Inertial Measurement Unit*. 1, 43

MPF *Median Paired Fin*. 1, 6

PMM *Planar Motion Mechanism*. 1, 27, 32, 53

PWM *Pulse Width Modulated*. 1, 46

RANS *Reynolds Averaged Navier-Stokes*. 1, 29

ROV *Remote Operated Vehicle*. 1, 2

SiP *System in Package*. 1, 43

SPS *Samples per Second*. 1, 45

UV *Underwater Vehicle*. v, 1–3, 5, 9, 53

Symbols

A Cross-section area.

A_f Lateral area of the fin.

α_t Coefficient defining the fullnes of the fore body.

$z_0, \theta_0, y_0, \psi_0$ Amplitude of oscillation of pure heave, pitch, sway and yaw motion.

$[\phi, \theta, \psi]$ Roll, Pitch and Yaw angles.

C_D, C_L Drag and lift coefficients.

C_p Prismatic coefficient.

C_V Coefficient of viscous resistance.

D Middle body diameter.

D_b Diameter of the propeller blades.

$E - \xi\eta\zeta$ Inertial coordinates system fixed to the Earth.

$O - xyz$ Inertial coordinates system fixed to the vehicle.

η Efficiency of propulsion system.

$F_{d,b}$ Drag and buoyancy forces.

F_{d_f} Drag force produced by a fin.

F_{l_f} Lift force produced by a fin.

$[X, Y, Z, K, M, N]$ Forces and moments on the movement coordinate system O-xyz.

g Gravity of Earth 9.81 m/s^2.

I_x, I_y, I_z Moments of inertia respect the x, y and z axis respectively.

K_{tg} Thrust and rotation coefficients of a propeller.

L Overall length of the vehicle.

l_a Length of the aft body.

l_f Length of the fore body.

m Mass of the vehicle.

μ Dynamic viscosity of the fluid.

n_f Coefficient defining the fullnes of the fore body.

n_n Shape coefficient of the nose.

n_t Shape coefficient of the tail.

$[X', Y', Z', K', M', N']$ Non-dimensional forces and moments on the movement coordinate system $O - xyz$.

ω Frequency for pure heave, pitch, sway and yaw motion.

P Weight of the vehicle outside of the water.

P_H Power consumed by the hardware.

P_p Power of propulsion system.

$[\xi, \eta, \zeta, \phi, \theta, \psi]$ Position and attitude on fixed coordinates $E - \xi\eta\zeta$.

R_e Reynolds number.

r_m Middle body radius.

r_{x_a} Aft body radius.

r_{x_f} Fore Body Radius.

ρ Density of water 973 kg/m^3.

S Surface of the vehicle in contact with water.

$T_{1,2}$ Transformation matrices.

V_f Displaced volume of the fluid by the solid.

v Velocity.

V_{AUV} Volume of the AUV.

$[u, v, w, p, q, r]$ Linear and angular velocities on the dynamic coordinate system O-xyz.

x_a Distance from the rearmostpart of the aft body.

x_f Distance from the rearmostpart of the fore body.

x_B, y_B, z_B Coordinates of the center of buoyancy on the dynamic frame.

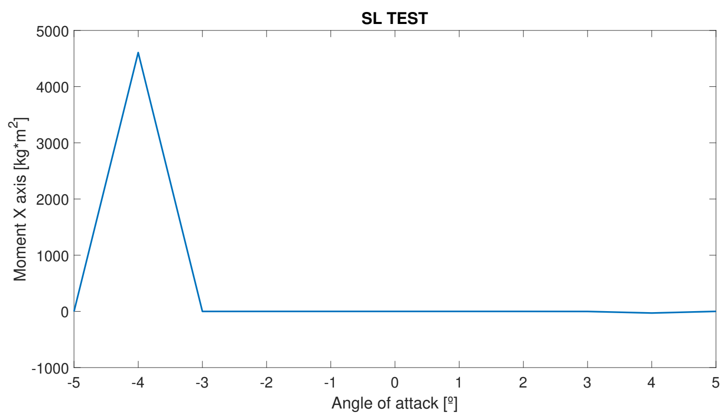
x_G, y_G, z_G Coordinates of the center of mass on the dynamic frame.

Appendix A

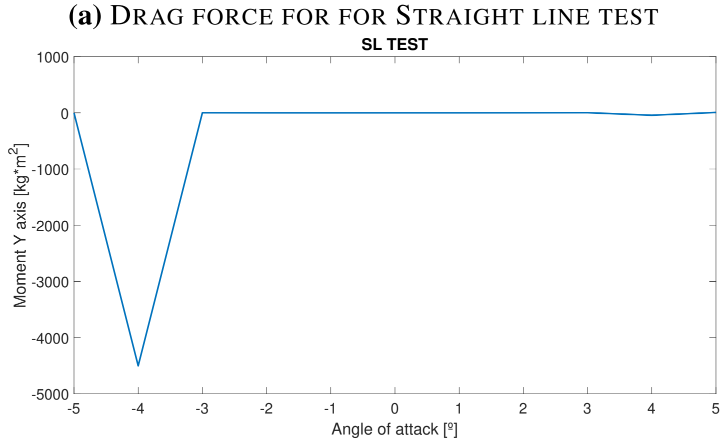
Ansys Fluent Simulations

Here the results of each test, shown in 4.2, of CFD analysis are presented in the plots below.

A.1 Calculation Results

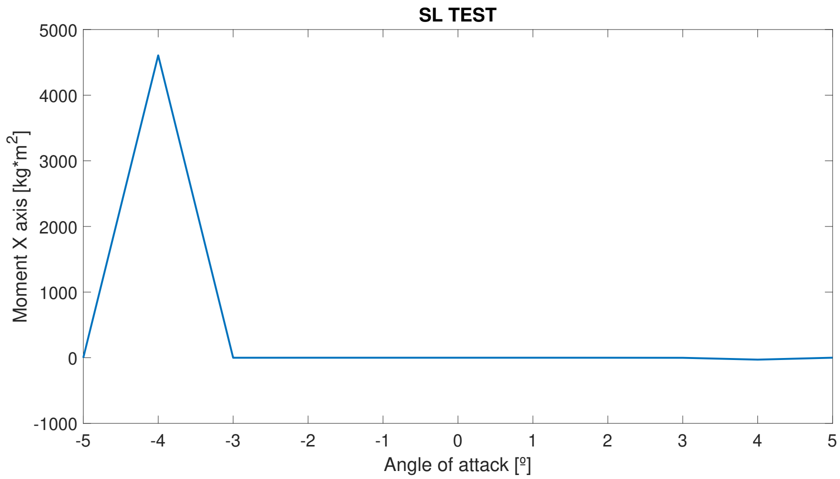


(a) DRAG FORCE FOR FOR STRAIGHT LINE TEST

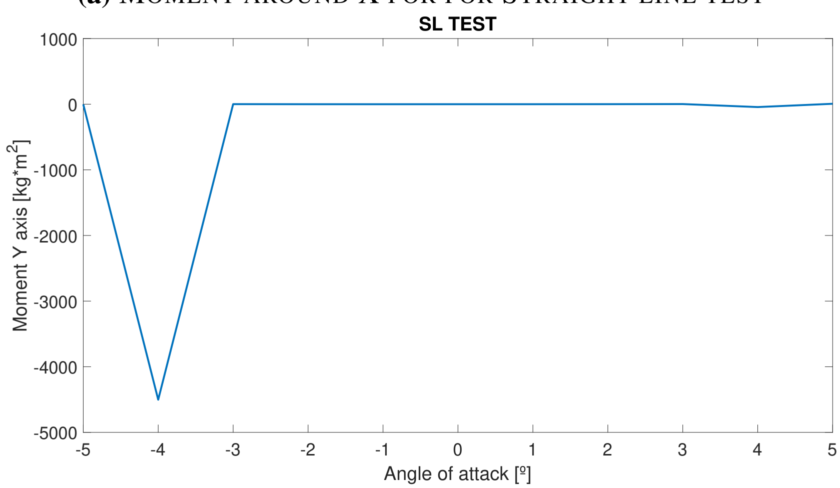


(b) LIFT FORCE FOR FOR STRAIGHT LINE TEST

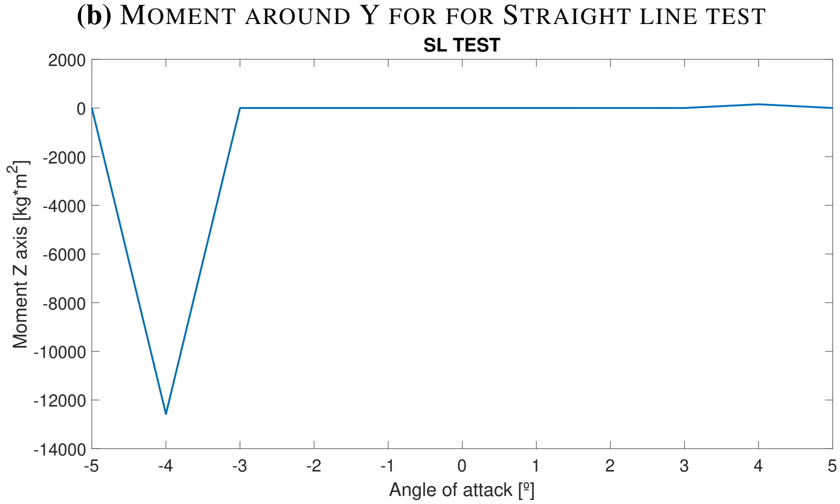
Figure A.1. RESULTS OF SIMULATION WITH DIFFERENT ANGLES OF ATTACK



(a) Moment around X for for Straight line test

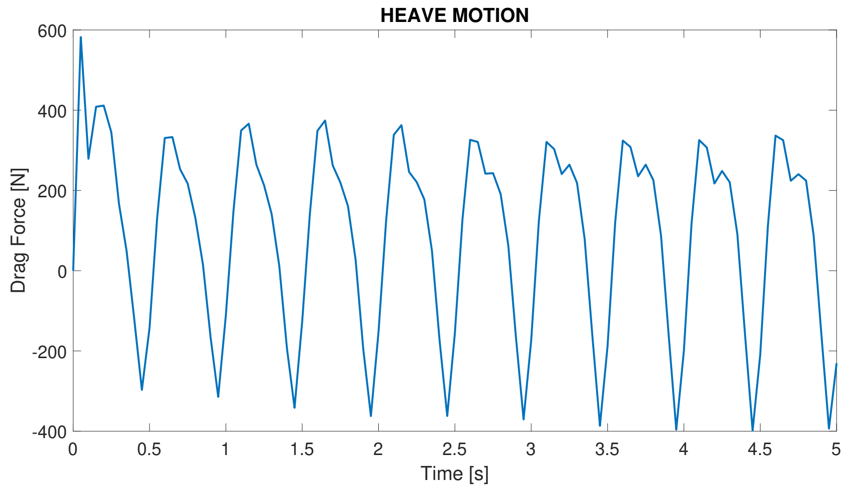


(b) Moment around Y for for Straight line test

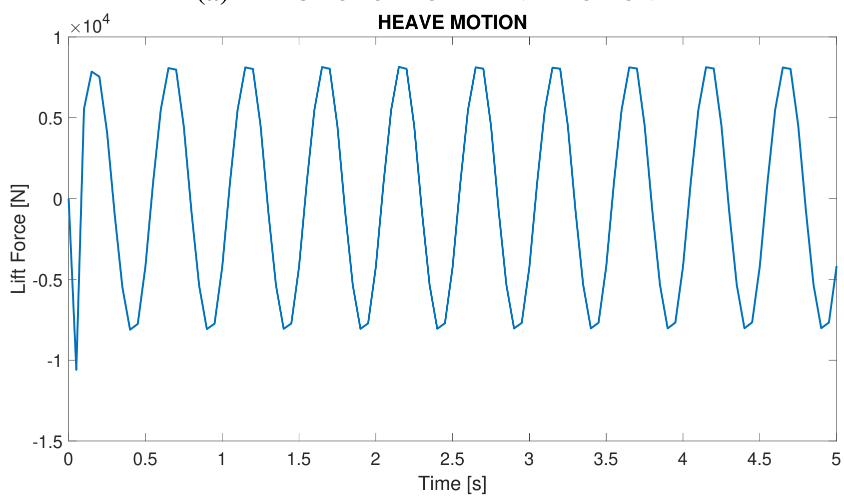


(c) Moment around Z for Straight line test

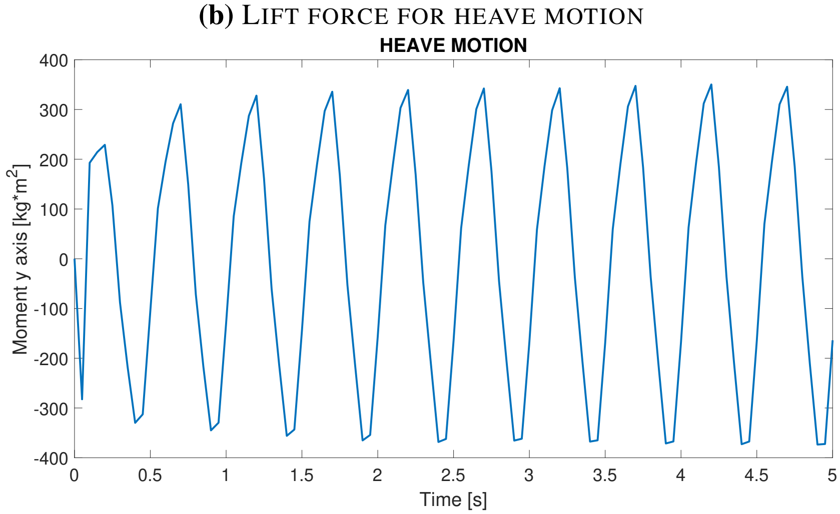
Figure A.2. Results of simulation with different angles of attack



(a) DRAG FORCE FOR HEAVE MOTION

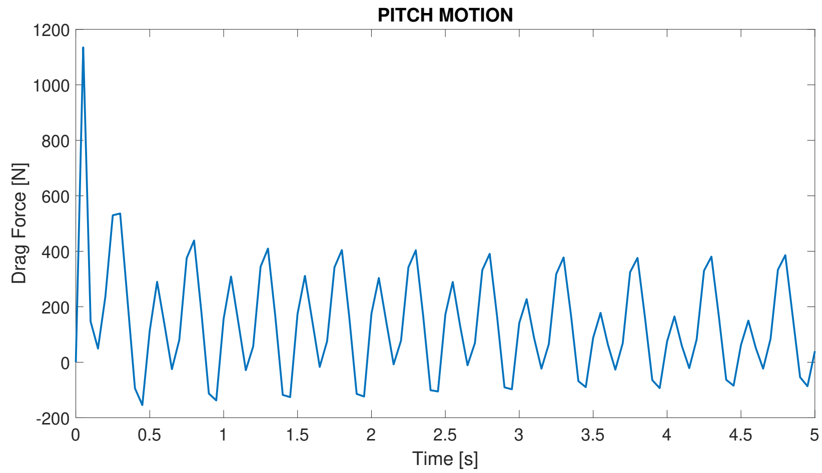


(b) LIFT FORCE FOR HEAVE MOTION

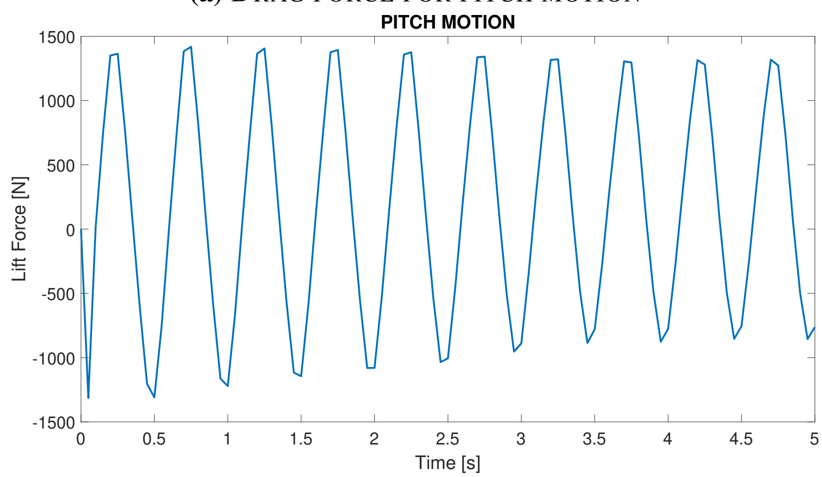


(c) MOMENT FOR HEAVE MOTION

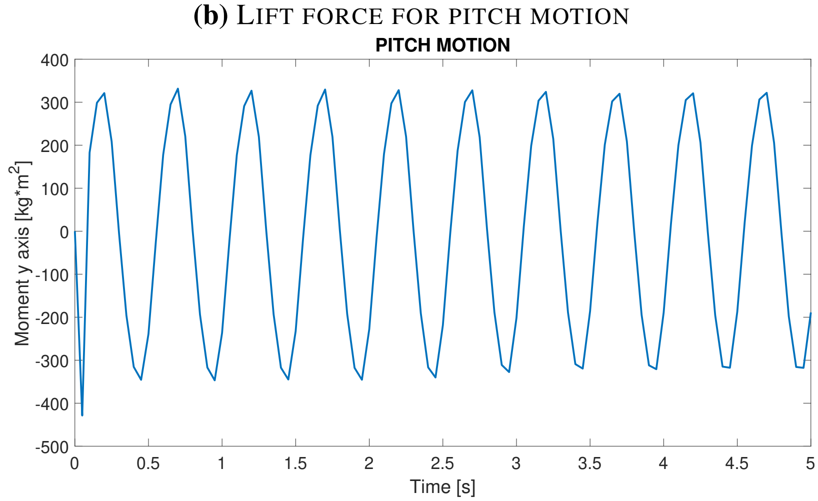
Figure A.3. RESULTS OF 10 SECONDS SIMULATION



(a) DRAG FORCE FOR PITCH MOTION

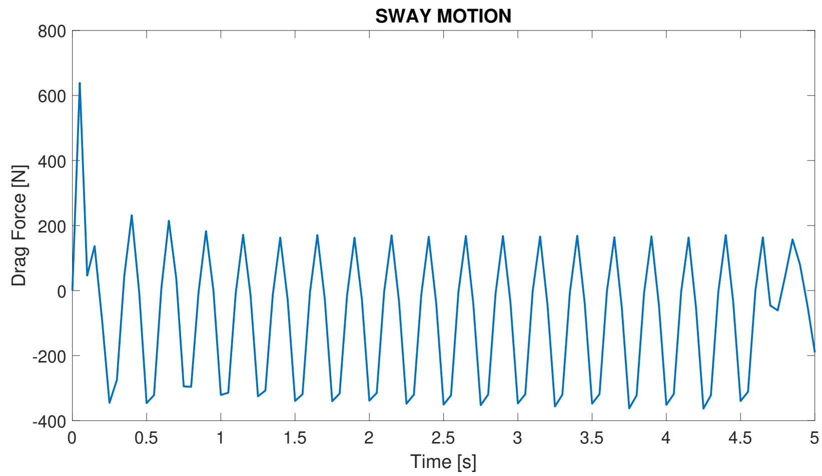


(b) LIFT FORCE FOR PITCH MOTION

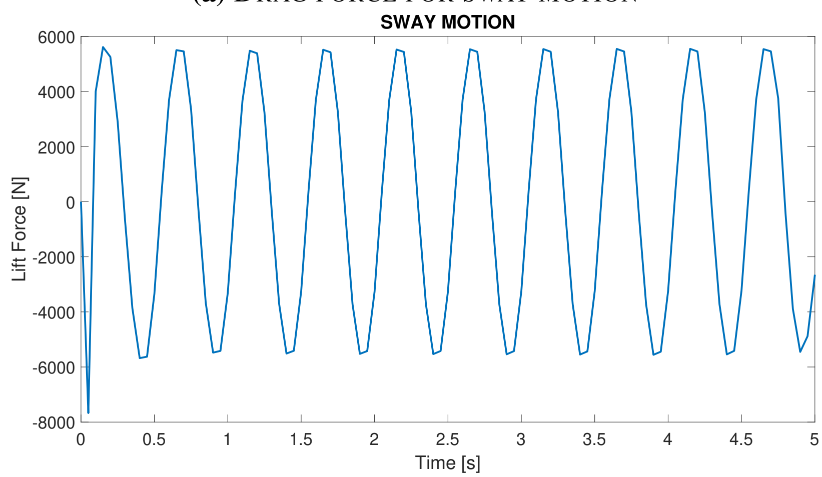


(c) MOMENT FOR PITCH MOTION

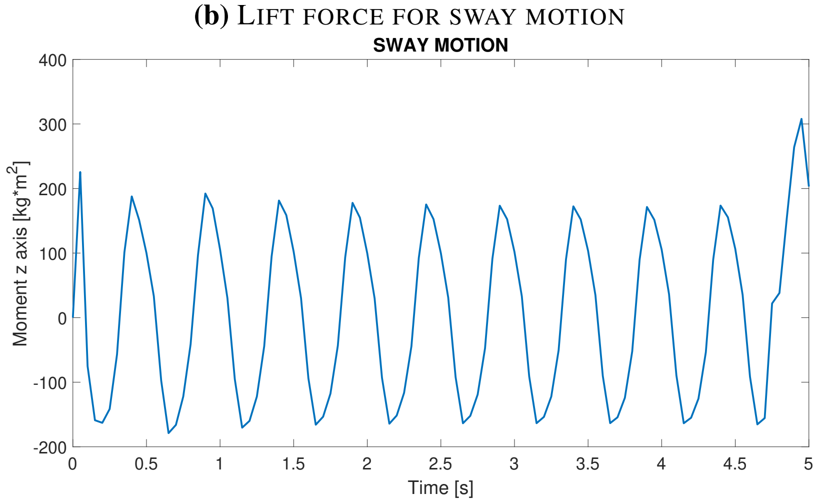
Figure A.4. RESULTS OF 10 SECONDS SIMULATION



(a) DRAG FORCE FOR SWAY MOTION

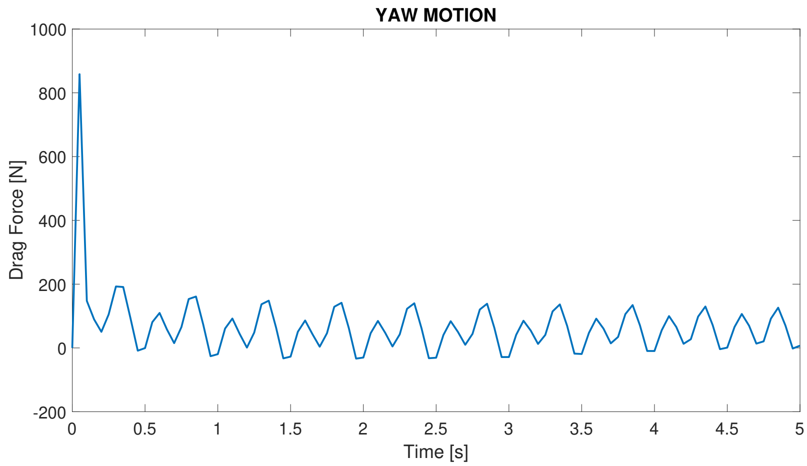


(b) LIFT FORCE FOR SWAY MOTION

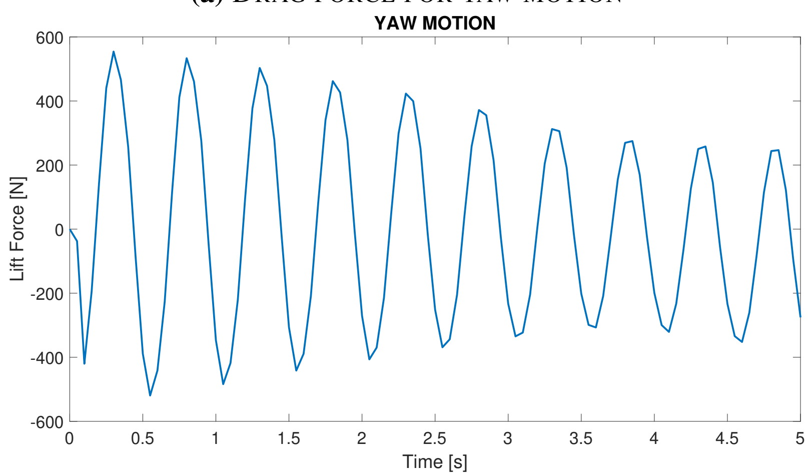


(c) MOMENT FOR SWAY MOTION

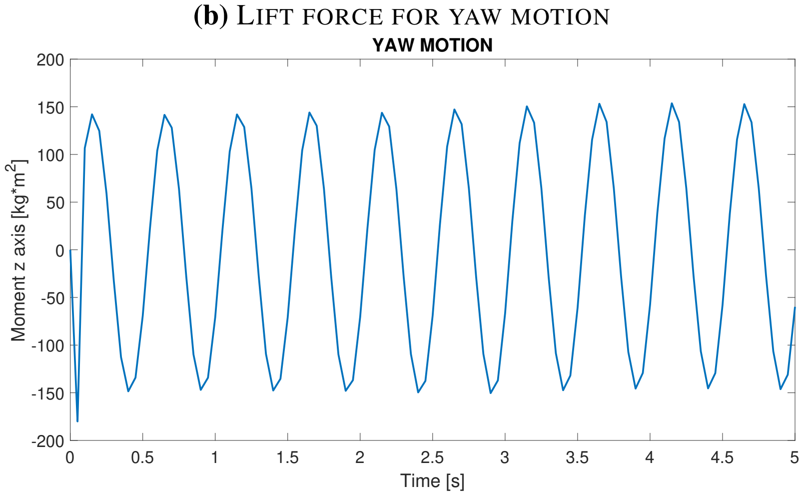
Figure A.5. RESULTS OF 10 SECONDS SIMULATION



(a) Drag force for yaw motion



(b) Lift force for yaw motion



(c) Moment for yaw motion

Figure A.6. Results of 10 seconds simulation

Appendix B

Software Codes

B.1 Client Script

First some libraries are imported and variables are initialized.

```python
# -*- coding: utf-8 -*-
import tkinter                          # library for GUI interfaces
from tkinter import messagebox          # pop up messages
import threading                        # library for threads
import socket                           # library for ethernet sockets
import time
import sys
```

```python
ip = "raspberrypi"  # default ip address
# defaul values for the motors
motor_1 = [90]
motor_2 = [90]
motor_3 = [90, 20, 0.5, 0.1, 20]
motor_4 = [90, 20, 0.5, 0.1, 20]
motor_5 = [90, 20, 0.5, 0.1, 20]
flag_main = True
```

Various functions are created to do task as call a thread to read the data in the server or to close everything,

```python
def insert_callback(txt):
    """ Callback for sending data through a socket to the server.

    :param txt: text data to be send. Example: txt = b"w
        ,0;1;2;3;4;5,0;0;0;0;0"
    """
    messagebox.showinfo("Thread", "Sending data....")
    client_write = ClientBAUV(txt, 2, ip)       # creates an instance of a
        client for the BAUV
    client_write.start()                        # Start the thread
```

```
def exit_callback():
    """ Function to close everything before exiting

    """
    global flag_main
    messagebox.showinfo("Bye_Python", "Finishing...")
    time.sleep(1)
    client_read.finish()
    flag_main = False
    sys.exit()
```

A class to connect with the server is defined as a thread to run in parallel with the main program.

```
class ClientBAUV(threading.Thread):
    """Class to create a client to communicate with the server as a thread
        .

    """
    def __init__(self, message, type_client, ip_server, *inter):
        """ constructor for the client.

        :param message: text to be send.
        :param type_client: 1 to read information, 2 to write values on
            the server
        :param ip_server: ip address of the server
        :param inter: Interface to write the read data. Only when
            type_client=1.
        """
        # Call to initialize the Thread class
        threading.Thread.__init__(self)
        # save the parameters and set initial values
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.msg = message
        self.type = type_client
        self.ip = ip_server
        self.app = inter[0]
        self.sep = [0]

    def finish(self):
        msg_send = b"done"
        self.socket.send(msg_send)
        self.socket.close()
```

The most important method connects with the server and does the task assigned.

```
    def run(self):
        """ Method to read or write data of the server through a socket.
            When data is read,
         this data is display in a GUI every second.
        Stops when the GUI  is closed.
```

```
        When data is written in the server, only works one time then it
            closes itself.
        """
        print("Process: " + str(self.type))
        self.socket.connect((ip, 8000))     # connect to the server
        if self.type == 1:                  # select the type of client
            while True:
                try:
                    msg_send = b"hola"      # establish first contact
                    self.socket.send(msg_send)
                    msg_send = self.msg     # send request to read
                    self.socket.send(msg_send)
                    request = self.socket.recv(1000)    # read answer
                except socket.error:
                    print("Conection error")
                    break
                print(request.decode("utf-8"))
                self.sep = request.decode("utf-8").split(",")   # decode
                    and interpret data
                adc_data = self.sep[0:8]
                sen_hum = self.sep[8:10]
                sen_imu = self.sep[10:33]
                thread_modify = Modify(self.app, self.sep)      # modify
                    the GUI
                thread_modify.start()
                time.sleep(1)       # wait a second
        else:
            msg_send = b"hola"          # establish first contact
            self.socket.send(msg_send)      # send request to write
            data_send = "w," + ';'.join(list(map(str, motor_1))) + "," + '
                ;'.join(list(map(str, motor_2)))\
                        + "," + ';'.join(list(map(str, motor_3))) + "," +
                            ';'.join(list(map(str, motor_4)))\
                        + "," + ';'.join(list(map(str, motor_5)))
            self.msg = data_send        # create data with specific format
            print("Send: " + data_send)
            self.socket.send(self.msg.encode())     # semd data
            time.sleep(1)
            self.finish()       # call a method to close the socket
```

Other class is defined to create an interface to visualize data and send new values to the motors of the BAUV. Its based on *Tkinter* and threads. It has its constructor, a method to take data from the interface, a method to close everything and the main method that creates the interface objects and displays them.

```
class MyTkApp(threading.Thread):
    """ Class to create a GUI to visualize data and send data to the
        server

    """
    def __init__(self):
        # input values
        self.in1 = 0; self.in2 = 0; self.in4 = 0
```

```
        self.in31 = 0; self.in32 = 0; self.in33 = 0
        # ADC values
        self.in51 = 0; self.in61 = 0; self.in71 = 0; self.in81 = 0
        self.in52 = 0; self.in62 = 0; self.in72 = 0; self.in82 = 0
        # Humidity
        self.in9 = 0
        self.in10 = 0
        threading.Thread.__init__(self)
        self.start()

    def callback(self):
        self.root.quit()

    def take_data(self):
        motors = (self.in1.get(), self.in2.get(), self.in31.get(), self.
            in32.get(), self.in33.get())
        data_send = 'w,'+','.join(("1; 2; 3; 4; 5", ';'.join(motors)))
        insert_callback(data_send)

    def run(self):
        global x0d
        self.root = tkinter.Tk()
        self.root.protocol("WM_DELETE_WINDOW", self.callback)
        # ===============Title of  GUI=================
        title = tkinter.Label(self.root, text="CONTROL CENTER", font=("
            Helvetica", 34))
        title.grid(row=0, columnspan=10)
        lbl11 = tkinter.Label(self.root, text="IMU1", font=("Helvetica",
            18))
        lbl11.grid(row=4, column=4)
        # ===============Labels and Entries==============
        lbl1 = tkinter.Label(self.root, text="Right fin angle: ")
        lbl1.grid(row=1, column=0)
        self.in1 = tkinter.Entry(self.root)
        self.in1.insert(0, "0")
        self.in1.grid(row=1, column=1)
        # ============= Sensors ========================
        lbl5 = tkinter.Label(self.root, text="ADC1_0: ")
        lbl5.grid(row=4, column=0)
        self.in51 = tkinter.Label(self.root, text=" ")
        self.in51.grid(row=4, column=1)
        # ============ buttons==========================
        btn1 = tkinter.Button(self.root, text="Send values", command=self.
            take_data)
        btn1.grid(row=10, column=1)
        btn2 = tkinter.Button(self.root, text="Exit", command=
            exit_callback)
        btn2.grid(row=10, column=2)
        self.root.mainloop()
```

The last class has the task of modify the data displayed on the interface. As before, it uses threads and the previously created interface.

```
class Modify(threading.Thread):
    def __init__(self, inter, dat):
        """

        :param inter: GUI where to apply the changes.
        :param dat:  data to modify the GUI.
        """
        # Call to initialize the Thread class
        threading.Thread.__init__(self)
        self.tk_interface = inter
        print(inter)
        if len(dat) >= 1:
            self.adc = list(map(float, dat[0:8]))
            self.hum = list(map(float, dat[8:10]))
            self.imu1 = list(map(float, dat[10:33]))
            self.imu2 = list(map(float, dat[33:42]))
        else:
            self.adc = [0, 0, 0, 0, 0, 0, 0, 0]
            self.hum = [0, 0]
            self.imu1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0]
            self.imu2 = [0, 0, 0, 0, 0, 0, 0, 0, 0]

    def run(self) -> None:
        self.tk_interface.in51.config(text=str("{0:.1f}".format(self.adc
            [0])))
        self.tk_interface.in61.config(text=str("{0:.1f}".format(self.adc
            [1])))
        print("Modified")
```

The main program instances the interface class and a client to start reading the data. Then its kept in a loop until the program closes with other thread.

```
if __name__ == '__main__':
    # Start the main GUI
    gui_app = MyTkApp()
    msg_txt = b"r,0;0;0,1;1;1"      # message to read data
    # Start connection thread
    client_read = ClientBAUV(msg_txt, 1, ip, gui_app)
    client_read.start()
    while flag_main:
        print("Main loop")
        time.sleep(5)
```

B.2 Server Script

First some libraries are imported and variables are initialized. Some libraries, of drivers, ADCs and sensors, are open source and were obtained in the web.

```
# -*- coding: utf-8 -*-
import threading
```

```
import socket
import board
import busio
import adafruit_ads1x15.ads1115 as ads
from adafruit_ads1x15.analog_in import AnalogIn
import math
from adafruit_servokit import ServoKit
import adafruit_am2320
import adafruit_bno055
import Adafruit_ADXL345
import hmc5883l as gy85mag
import ITG3205 as gy85gyro
import time
import sys
```

```
channels = [0, 0, 0, 0, 0, 0, 0, 0]
adc_hum_imu = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
kit1 = 0
kit2 = 0
i2c = busio.I2C(board.SCL, board.SDA)
am = 0
sensor = 0
compass = 0
gyro = 0
gyro_sensitivity = 14.373  # LSBs per degree/seg
accel = 0
old_angles = [0, 0, 0, 0, 0]
```

Various functions are created to do task as: initialize the sensors, change values, calculated conversion and get positions. A first function its defined to initialize each sensor.

```
def initialize(chips):
    """ Initialize the sensor to be used by the BAUV.

    :param chips: logical values that indicate which sensor is to be used
    """
    global sensor, compass, gyro, accel, am, kit1, kit2
    counts = 0
    for i in chips:     # initiate each sensor
        counts = counts + 1
        if counts == 1 and i == '1':
            sensor = adafruit_bno055.BNO055(i2c)
        elif counts == 2 and i == '1':
            compass = gy85mag.Hmc5883l(gauss=4.7, declination=(-2, 5))
            gyro = gy85gyro.ITG3200()
            accel = Adafruit_ADXL345.ADXL345()
        elif counts == 3 and i == '1':
            am = adafruit_am2320.AM2320(i2c)
        elif counts == 4 and i == '1':
            kit1 = ServoKit(channels=16)
        elif counts == 5 and i == '1':
```

```
            kit2 = ServoKit(channels=16, address=0x41)
        elif counts == 6 and i == '1':
            ads1 = ads.ADS1115(i2c, address=0x48)
            channels[0] = AnalogIn(ads1, ads.P0)
            channels[1] = AnalogIn(ads1, ads.P1)
            channels[2] = AnalogIn(ads1, ads.P2)
            channels[3] = AnalogIn(ads1, ads.P3)
        elif counts == 7 and i == '1':
            ads2 = ads.ADS1115(i2c, address=0x4b)
            channels[4] = AnalogIn(ads2, ads.P0)
            channels[5] = AnalogIn(ads2, ads.P1)
            channels[6] = AnalogIn(ads2, ads.P2)
            channels[7] = AnalogIn(ads2, ads.P3)
```

Other function is created to read the sensors.

```
def read(chips) -> list:
    """
    :param chips: logical values of chips working  example: chips
       ="1010111"
    :return: data read as a list
    """
    print("Reading sensors ")
    counts = 0
    for i in chips:
        counts = counts + 1
        if counts == 1 and i == '1':
            adc_hum_imu[10] = sensor.temperature
            adc_hum_imu[11:14] = sensor.accelerometer
            adc_hum_imu[14:17] = sensor.magnetometer
            adc_hum_imu[17:20] = sensor.gyroscope
            adc_hum_imu[20:23] = sensor.euler
            adc_hum_imu[23:27] = sensor.quaternion
            adc_hum_imu[27:30] = sensor.linear_acceleration
            adc_hum_imu[30:33] = sensor.gravity
        elif counts == 1 and i == '0':
            adc_hum_imu[10:33] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif counts == 2 and i == '1':
            adc_hum_imu[33:36] = compass.axes()
            adc_hum_imu[36:39] = gyro.read_data()
            adc_hum_imu[36:39] = map(convert_raw_to_dps, adc_hum_imu
                [36:39])
            adc_hum_imu[39:42] = accel.read()
            adc_hum_imu[39:42] = map(convert_g_to_mps2, adc_hum_imu
                [39:42])
        elif counts == 2 and i == '0':
            adc_hum_imu[33:42] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif counts == 3 and i == '1':
            adc_hum_imu[8] = am.relative_humidity
            adc_hum_imu[9] = am.temperature
        elif counts == 4 and i == '1':
            print("No read")
        elif counts == 5 and i == '1':
```

```
            print("No read")
        elif counts == 6 and i == '1':
            adc_hum_imu[0] = channels[0].value
            adc_hum_imu[1] = channels[1].value
            adc_hum_imu[2] = channels[2].value
            adc_hum_imu[3] = channels[3].value
        elif counts == 7 and i == '1':
            adc_hum_imu[4] = channels[4].value
            adc_hum_imu[5] = channels[5].value
            adc_hum_imu[6] = channels[6].value
            adc_hum_imu[7] = channels[7].value
    return list(map(str, adc_hum_imu))
```

The we have functions to change the values of the actuator and calculated conversion.

```
def change_values(number, new_angle):
    """ changes the values of the angle on the servomotors

    :param number: number of servomotor 1 to 5
    :param new_angle: angle in degrees to be set
    """
    global old_angles
    print("The servo: " + str(number) + " change angle to " + str(
        new_angle))
    for i in range(len(number)):
        if old_angles[i] != new_angle[i]:
            kit1.servo[i].angle = new_angle[i]


def oscillation(freq, amp, t_actual, r0):
    """


    :param freq: frequency of oscillation
    :param amp: amplitud of oscillation
    :param t_actual: actual time step
    :param r0: initial position
    :return: values of a sinusoidal movement
    """
    return int(r0 + amp*math.sin(2*math.pi*freq*t_actual))
```

```
def convert_g_to_mps2(val):
    """ covert values of gravity to meter per second square

    :param val: value of gravity relative to a 1g
    :return: value meter per second square
    """
    multiplier = 4 * 2 ** (int(accel.get_range())) / (2 ** 10)
    return val*multiplier*9.81


def convert_raw_to_dps(val):
    """ conversion of the gyroscope
```

```
    :param val: value in integers
    :return: value in degrees per second
    """
    return val/gyro_sensitivity
```

A class to handle the actuator is defined as a thread to run in parallel with the main program.

```
class Servomotor(threading.Thread):
    def __init__(self, motor_type, address, values):
        """

        :param motor_type: motor of the dorsal fins or the propulsion
            system
        :param address: hexadecimal address of the driver for th I2C bus
        :param values: values to be set
        """
        self.type = motor_type
        self.id = address
        if self.type == 1:
            self.angle = values[0]
        else:
            self.r0 = values[0]
            self.amplitude = values[1]
            self.freq = values[2]
            self.t_sample = values[3]
            self.t_sim = values[4]

    def run(self) -> None:
        if self.type == 1:
            kit1.servo[self.id].angle = self.angle
        else:
            t_cum = 0
            t_end = self.t_sim
            t_start = time.time()
            print("start: " + str(t_start))
            while t_cum <= t_end:
                if time.time() - t_start >= self.t_sample:
                    t_cum = t_cum + self.t_sample
                    t_start = time.time()
                    new_angle = oscillation(self.freq, self.amplitude,
                        t_cum, 0)
                    kit1.servo[self.id].angle = new_angle
```

Every request of the client needs to be attended, so a class is created to handle every connection in parallel. There are two ways to handle requests, first as a readers and writers. The read request requires the server to stay connected with the client and the write request only needs a temporary connection.

```
class ClientBAUV(threading.Thread):
    def __init__(self, client_socket, info_client, chips):
        """
```

```
        :param client_socket: object to handle communication with the
            client
        :param info_client: information of the client
        :param chips: logical values for the available chips
        """
        # Call to initialize the Thread class
        threading.Thread.__init__(self)
        # save the parameters
        self.socket = client_socket
        self.data = info_client
        self.ch = chips

    def run(self):
        """The class handle in two ways the requests.
        """
        while True:
            print("Waiting request from: ")
            try:
                request = self.socket.recv(4)
                print(request.decode("utf-8"))
            except:
                print("Connection error")
                request = b"done"
                break
            if request.decode("utf-8") == "hola":
                while self.is_alive():
                    try:
                        request = self.socket.recv(100)
                        print("Receive: " + request.decode("utf-8"))
                    except:
                        print("Connection error")
                        break
                    if request.decode("utf-8") == "":
                        print(request.decode("utf-8"))
                        break
                    sep = request.decode("utf-8").split(",")
                    if sep[0] == 'w':
                        print(sep[1])
                        print(sep[2])
                        nums = list(map(int, sep[1].split(';')))
                        angles = list(map(int, sep[2].split(';')))
                        change_values(nums, angles)
                    elif sep[0] == 'r':
                        data_send = ','.join(read(self.ch))
                        try:
                            self.socket.send(data_send.encode())
                        except:
                            print("Connection error")
                        # self.socket.send(str(y).encode())
                    break   # break for alive
            elif request.decode("utf-8") == "done":
                self.socket.close()
                break
```

The main program determines which chips will be available and then initialize them. After that creates a server to attends request on port 8000. Finally a loop listen to request of a client and creates new threads to handle them.

```
if __name__ == '__main__':
    print('Number of arguments:', len(sys.argv), 'arguments.')
    if len(sys.argv) > 1:
        chip = str(sys.argv[1])
    else:
        chip = '1111111'
    initialize(chip)
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(("", 8000))
    while 1:
        server.listen(1)
        # Waiting for a client
        print("Waiting for a client")
        socket_client, data_client = server.accept()
        socket_client.settimeout(5)
        # Client information
        print("connect to " + str(data_client))
        thread = ClientBAUV(socket_client, data_client, chip)
        thread.start()
```

B.3 Model system equations

Main function on MATLAB to simulate the BAUV's behavior. This system states are the linear and angular accelerations of the model, then the traslation matrices are applied and the the position is calculated.

```
function x_out = AUV_model(t,x,F,I,cg)
    syms ud vd wd pd qd rd
    m=1.;    u=x(1);    v=x(2);    w=x(3);    p=x(4);    q=x(5);    r=x(6)
        ;
    X(ud,rd,qd) = m*((ud-v*r+w*q)-cg(1)*(q^2+r^2)+cg(2)*(p*q-rd)+cg(3)*(p*
        r+qd));
    Y(vd,pd,rd) = m*((vd-w*p+u*r)-cg(2)*(r^2+p^2)+cg(3)*(q*r-pd)+cg(1)*(q*
        p+rd));
    Z(wd,qd,pd) = m*((wd-u*q+v*p)-cg(3)*(p^2+q^2)+cg(1)*(r*p-rd)+cg(2)*(r*
        q+pd));
    K(pd,wd,vd) = I(1)*pd+(I(3)-I(2))*q*r+m*(cg(2)*(wd+p*v-q*u)-cg(3)*(vd+
        r*u-p*w));
    M(qd,ud,wd) = I(2)*qd+(I(1)-I(3))*r*p+m*(cg(3)*(ud+q*w-r*v)-cg(1)*(wd+
        p*v-q*u));
    N(rd,vd,ud) = I(3)*rd+(I(2)-I(1))*p*q+m*(cg(1)*(vd+r*u-p*w)-cg(2)*(ud+
        q*w-r*v));
    h = solve(X==F(1),Y==F(2),Z==F(3),K==F(4),M==F(5),N==F(6), ud, vd, wd,
         pd, qd, rd);
    x_out=[double(h.ud(1)),double(h.vd(1)),double(h.wd(1)) ,...
```

```
            double(h.pd(1)),double(h.qd(1)),double(h.rd(1))]';
end
```

```
function Rd=position(t,x,V)
    xi=x(1);eta=x(2);zeta=x(3);
    phi=x(4);theta=x(5);psi=x(6);
    T_1=[cos(psi)*cos(theta) cos(psi)*sin(theta)*sin(phi)-sin(psi)*cos(phi
       ) cos(psi)*sin(theta)*cos(phi)+sin(psi)*sin(phi);...
         sin(psi)*cos(theta) sin(psi)*sin(theta)*sin(phi)+cos(psi)*cos(phi
           ) sin(psi)*sin(theta)*cos(phi)-cos(psi)*sin(phi);...
         sin(theta) cos(theta)*sin(phi) cos(theta)*cos(phi)];
    T_2=[1 tan(theta)*sin(phi) tan(theta)*cos(phi);...
         0 cos(phi) -sin(phi);...
         0 sin(phi)*sec(theta) cos(phi)*sec(theta)];
    Rd=[T_1 zeros(3);zeros(3) T_2]*V;
end
```

Other important function is for the propeller effect. The model was obtained from [30] and has and effect on the displacement on X axis and in the roll. It has parameters as diameter of the propeller, revolutions per second and constants obtained by open water testing.

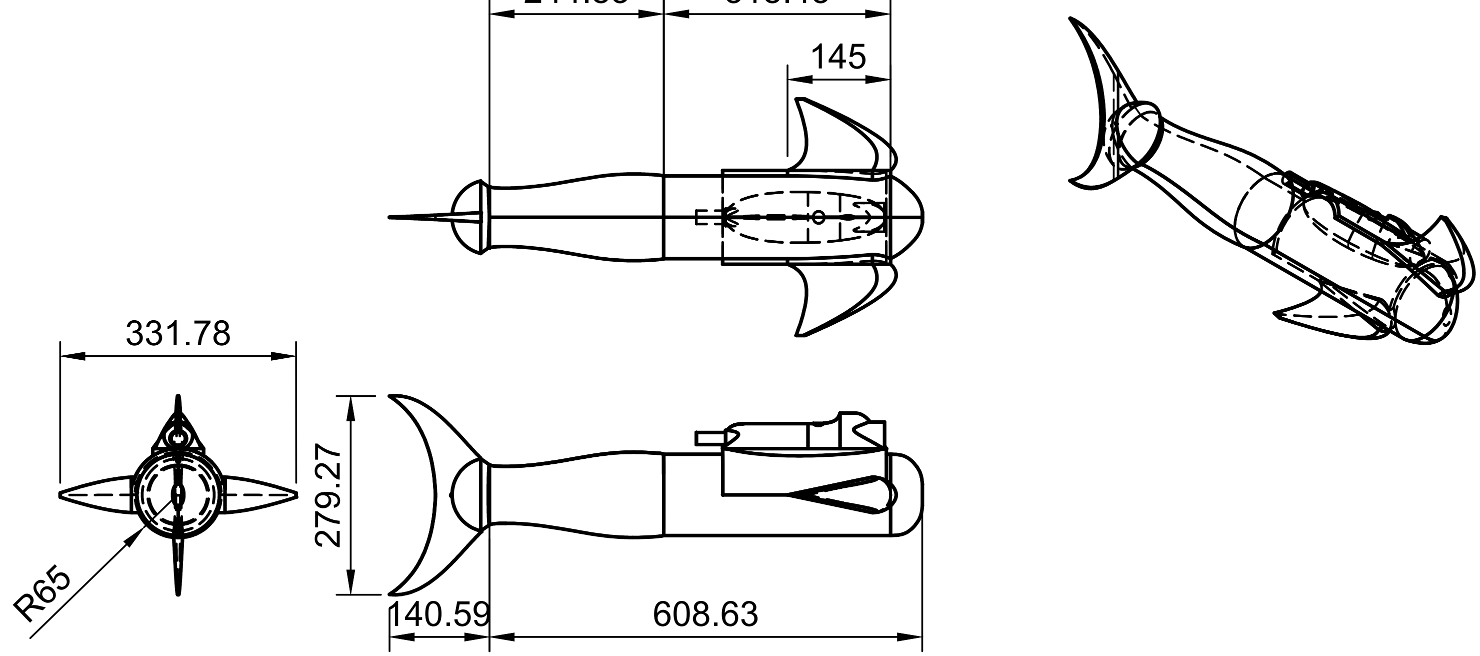
```
function Fp=propeller(n,kt,kq)
rho=1000;
D=0.15;

Fp=[kt*rho*(D^4)*n^2 0 0 kq*rho*(D^5)*n^2 0 0]';
end
```

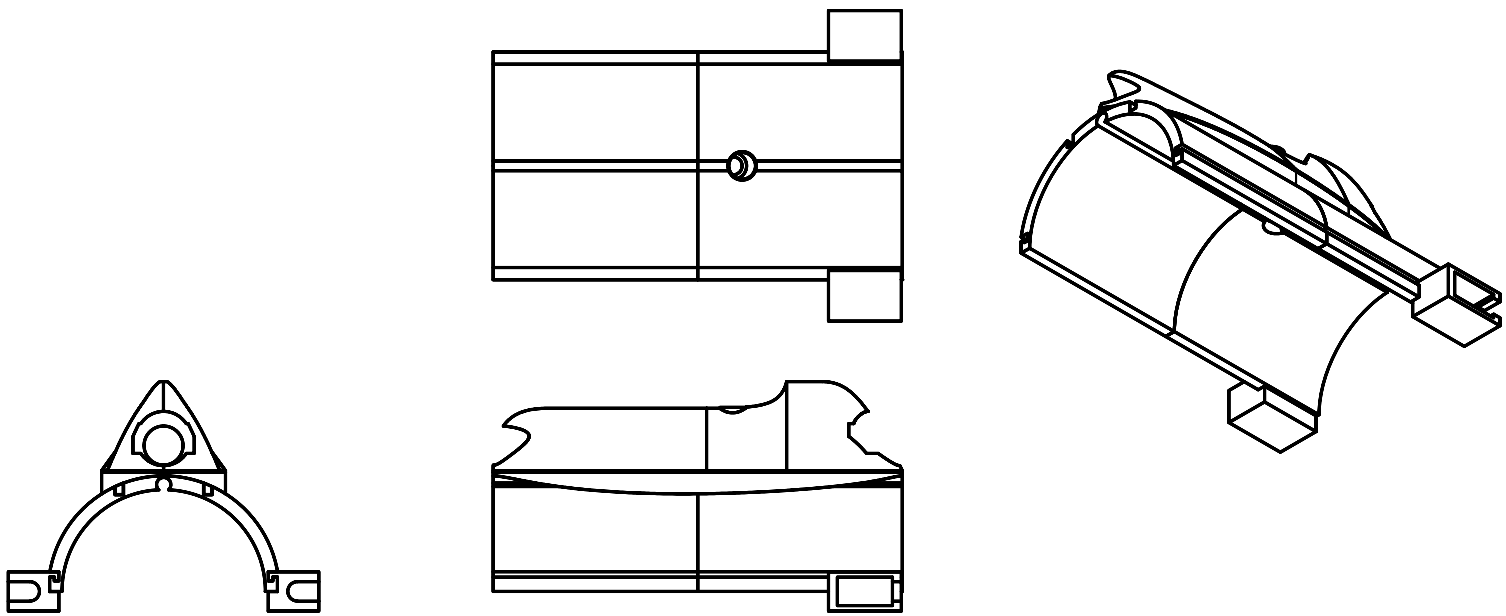
Appendix C

Drawings of the hull

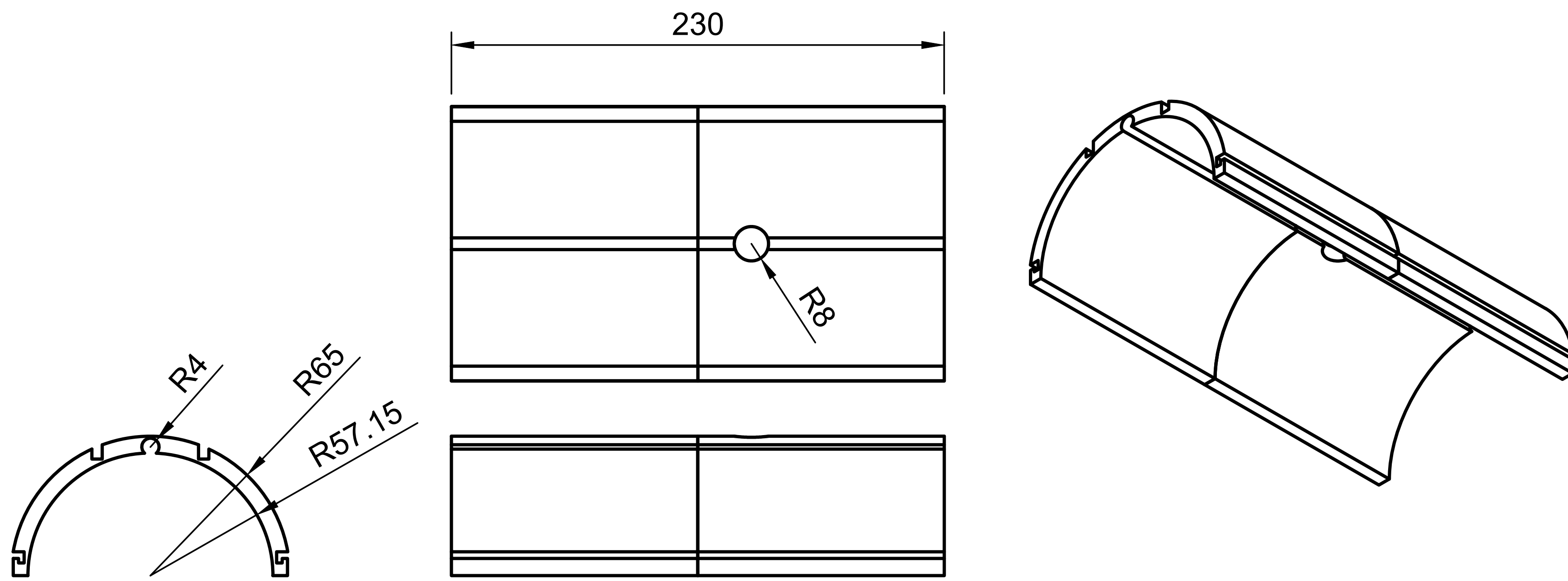
Here are included drawings of the designs presented in 5.1, only general dimension are displayed to give a general idea of the BAUV size.



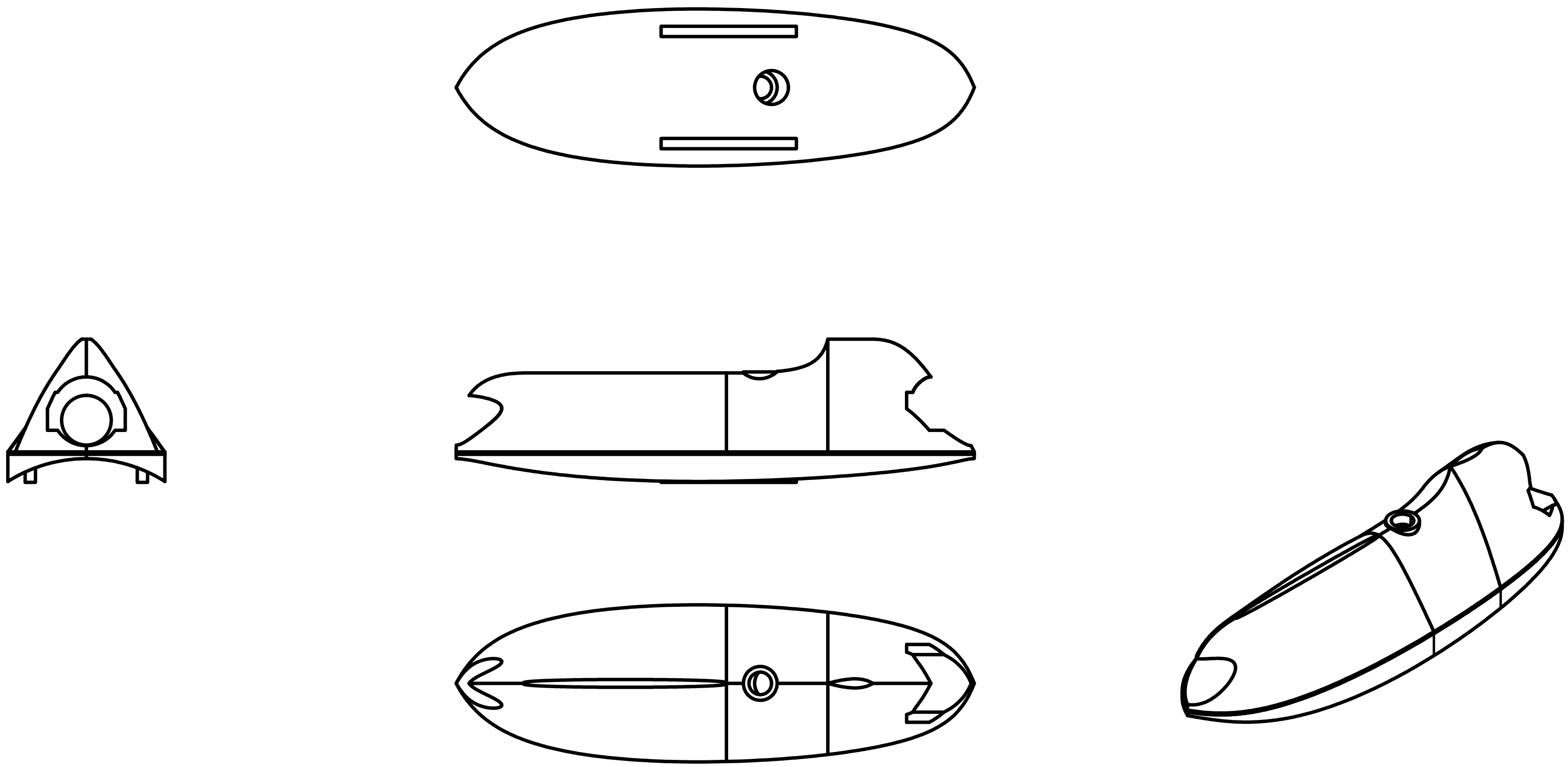
Dept.	Technical reference	Created by	Approved by
		Edisson Naula 06-Nov-19	
		Document type	Document status
		Title: BAUV Complete	DWG No.
			Rev. / Date of issue / Sheet 1/1



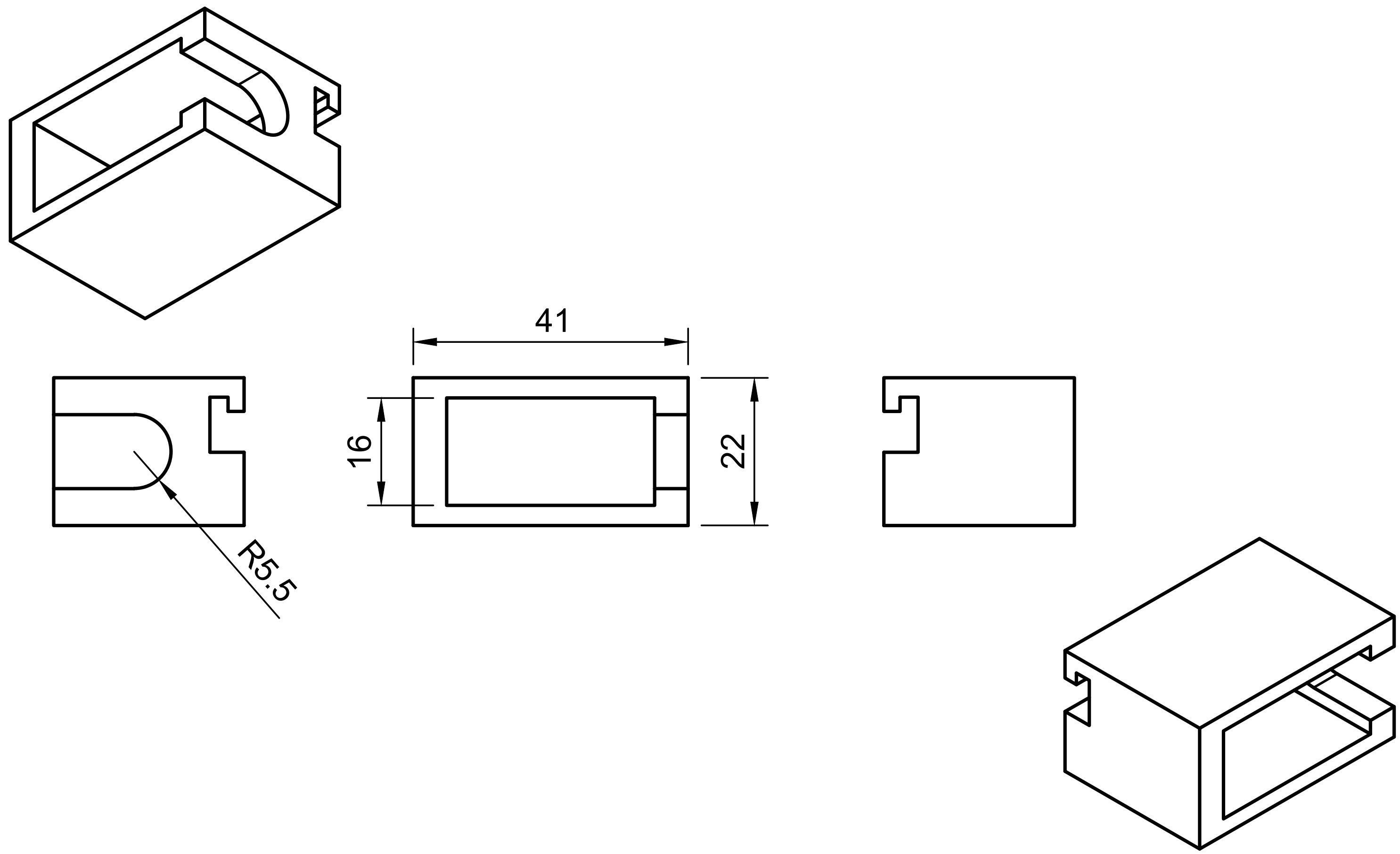
Dept.	Technical reference	Created by Edisson Naula 06-Nov-19	Approved by		
		Document type	Document status		
		Title Brace complete	DWG No.		
			Rev.	Date of issue	Sheet 1/4



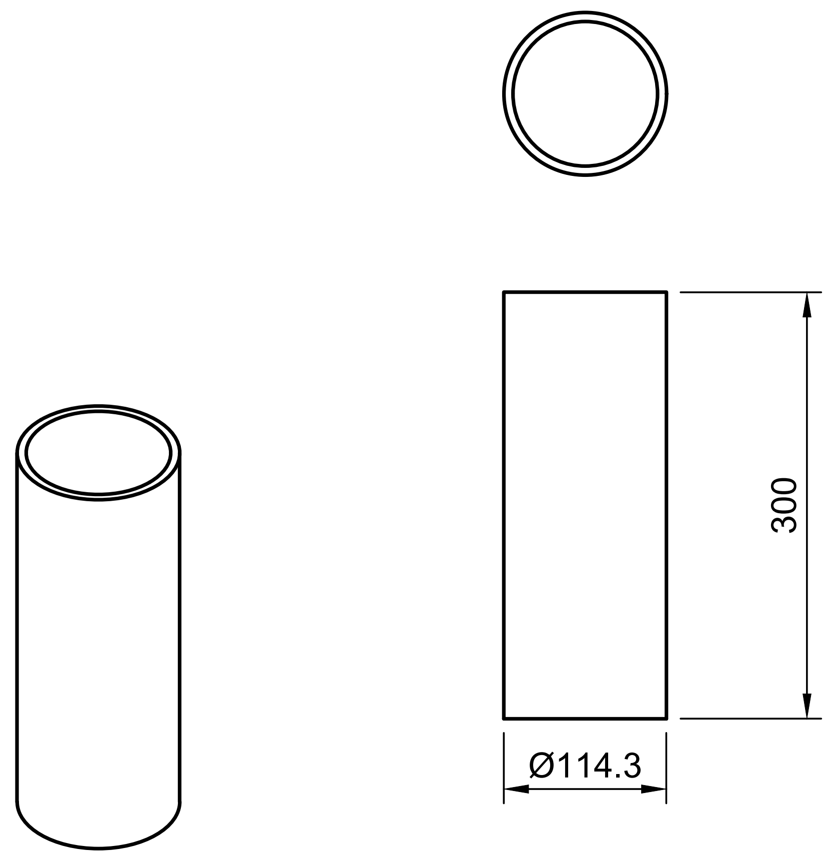
Dept.	Technical reference	Created by	Approved by
		Edisson Naula 06-Nov-19	
		Document type	Document status
		Title: Brace	DWG No.
			Rev. / Date of issue / Sheet 2/4



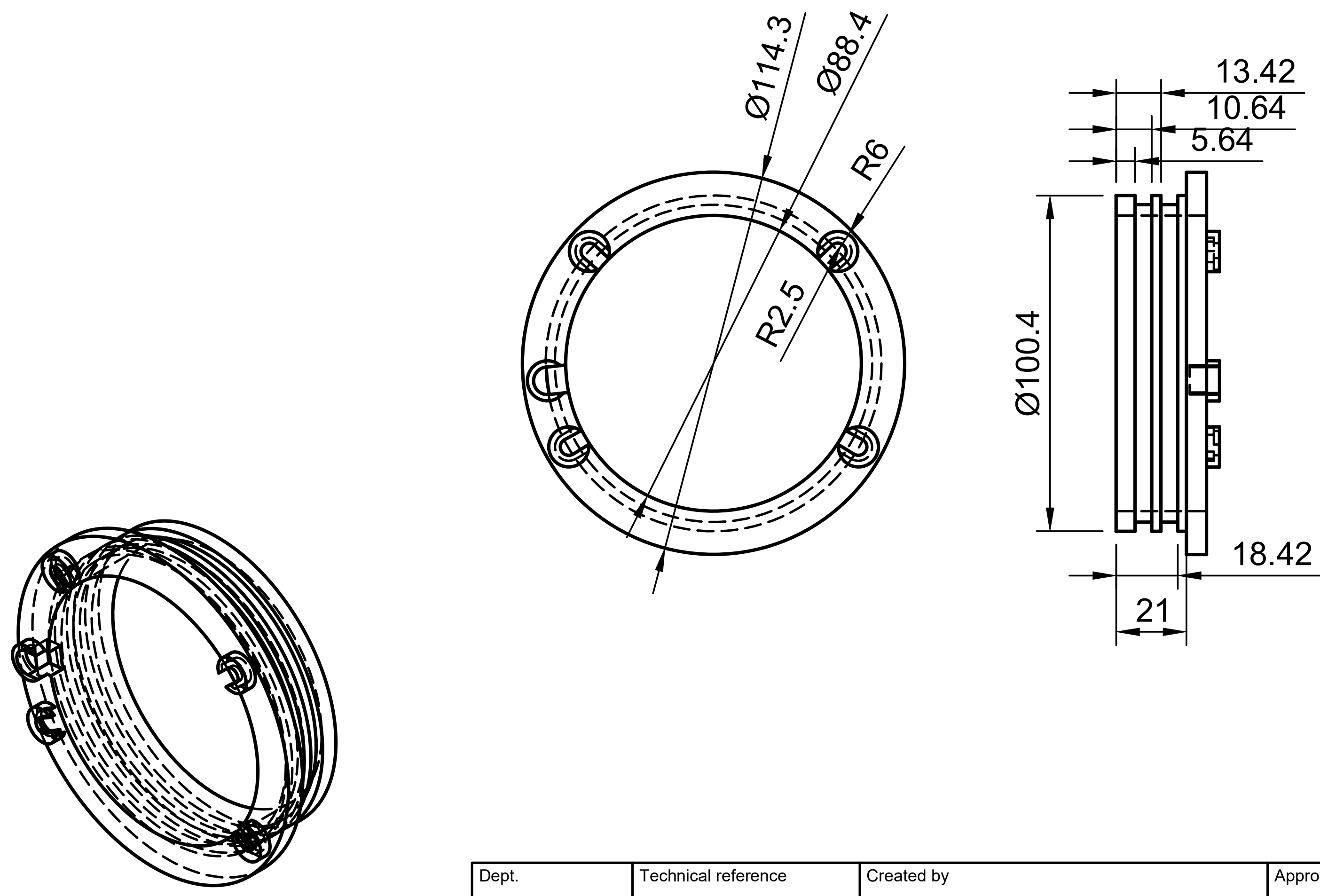
Dept.	Technical reference	Created by Edisson Naula 06-Nov-19	Approved by		
		Document type	Document status		
		Title Sensor holder	DWG No.		
			Rev.	Date of issue	Sheet 3/4



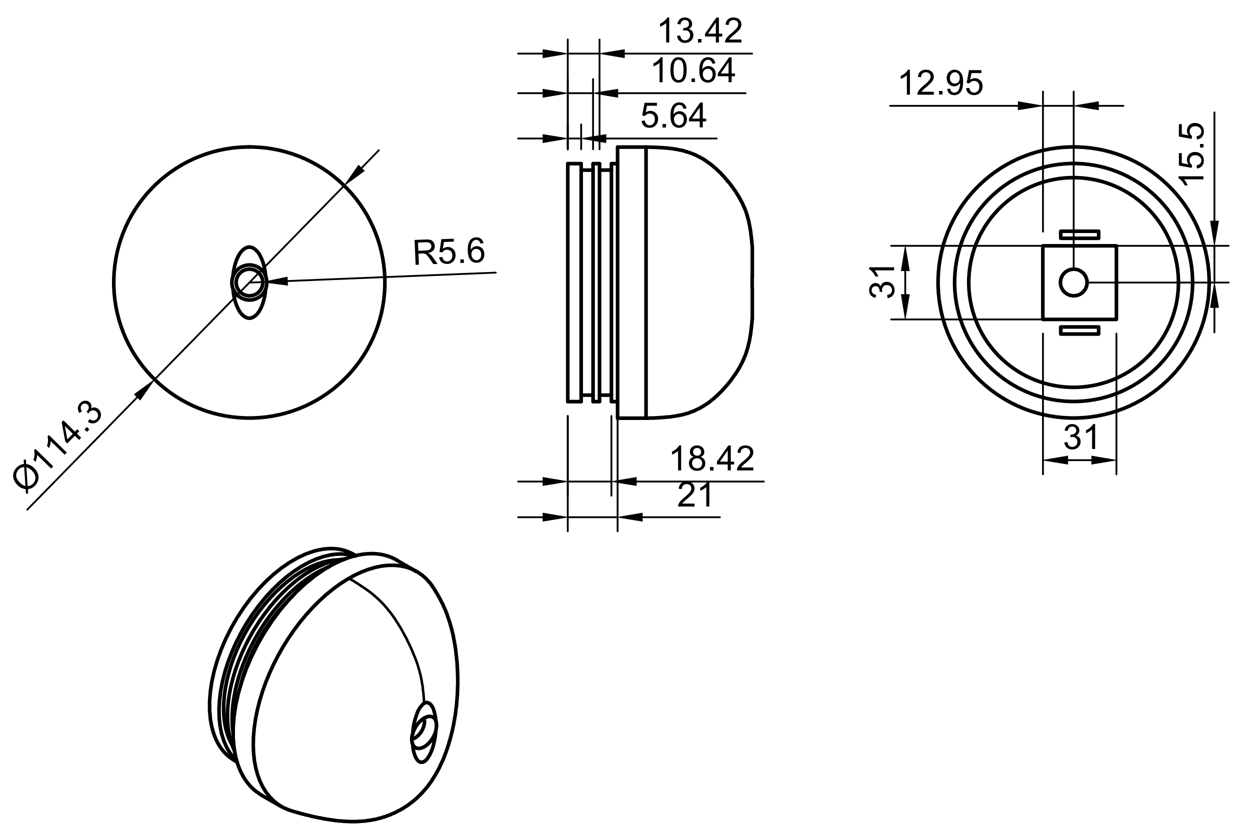
Dept.	Technical reference	Created by	Approved by		
		Edisson Naula 06-Nov-19			
		Document type	Document status		
		Title	DWG No.		
		Motor holder	Rev.	Date of issue	Sheet 4/4



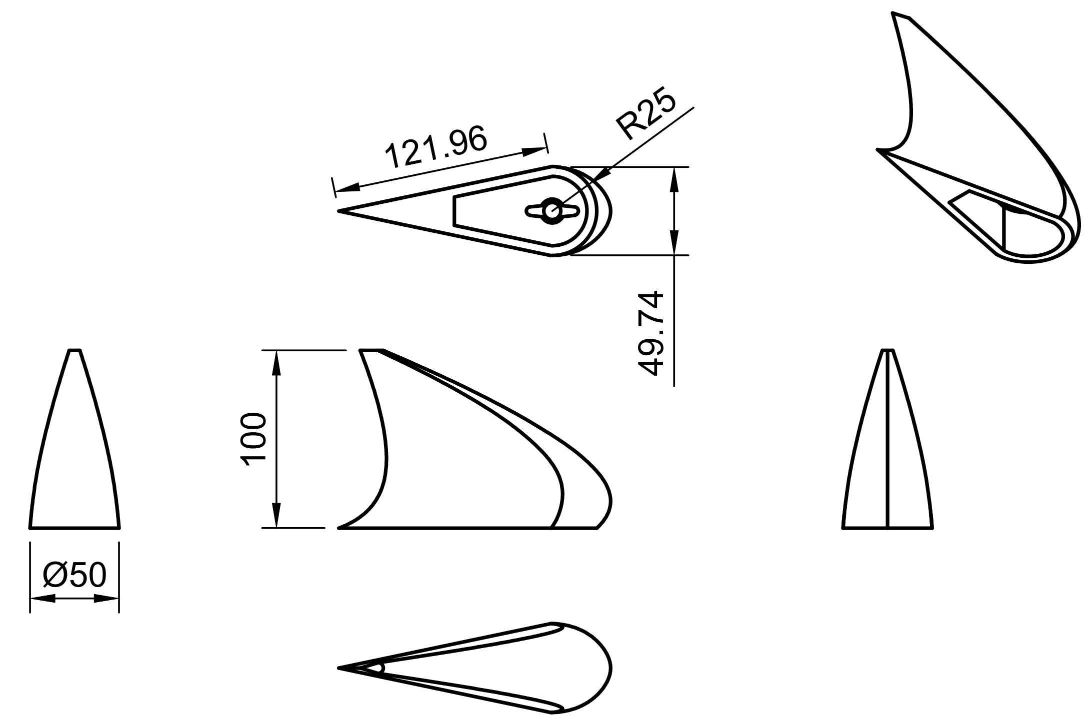
Dept.	Technical reference	Created by Edisson Naula 06-Nov-19	Approved by		
		Document type	Document status		
		Title Cilinder	DWG No.		
			Rev.	Date of issue	Sheet 1/1



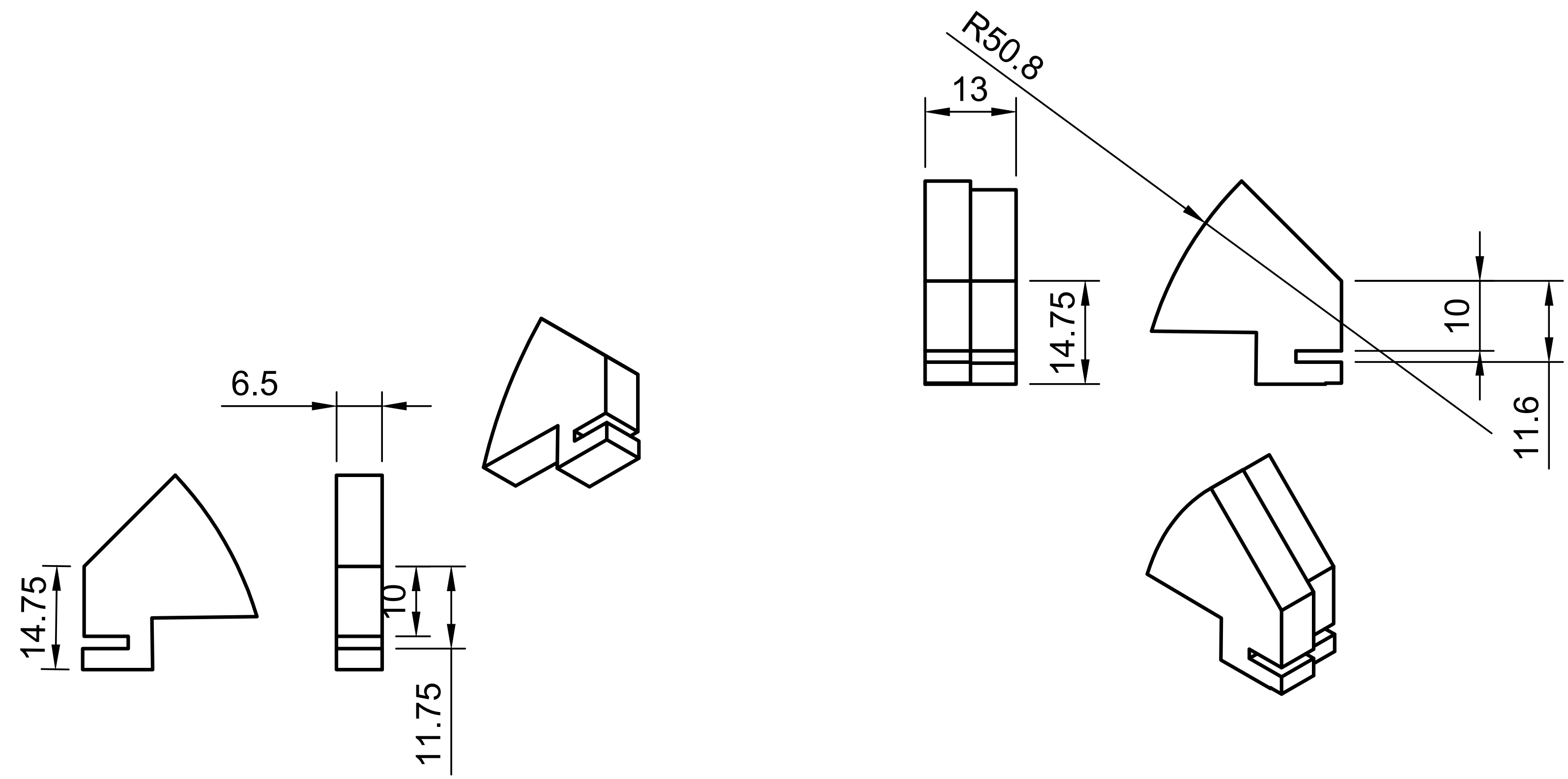
Dept.	Technical reference	Created by	Approved by		
		Edisson Naula 06-Nov-19			
		Document type	Document status		
		Title: Back Flange	DWG No.		
			Rev.	Date of issue	Sheet 1/1



Dept.	Technical reference	Created by	Approved by
		Edisson Naula 06-Nov-19	
		Document type	Document status
		Title: Nose	DWG No.
			Rev. / Date of issue / Sheet 1/1



Dept.	Technical reference	Created by	Approved by
		Edisson Naula 06-Nov-19	
		Document type	Document status
		Title: Pectoral Fin	DWG No.
			Rev. / Date of issue / Sheet 1/1



Dept.	Technical reference	Created by	Approved by
		Edisson Naula 06-Nov-19	
		Document type	Document status
		Title: Internal Supports	DWG No.
			Rev. / Date of issue / Sheet 1/2

Appendix D

Simulation of the model

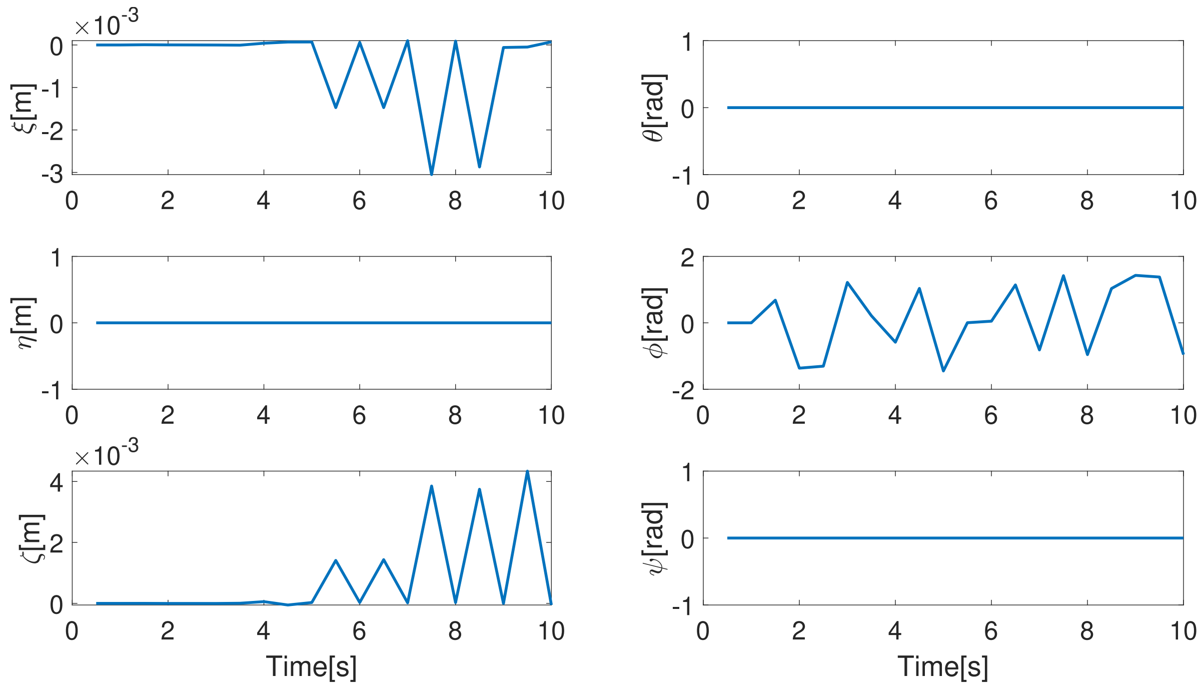


Figure D.1. MODEL STATES WITH CENTER OF MASS TO THE RIGHT

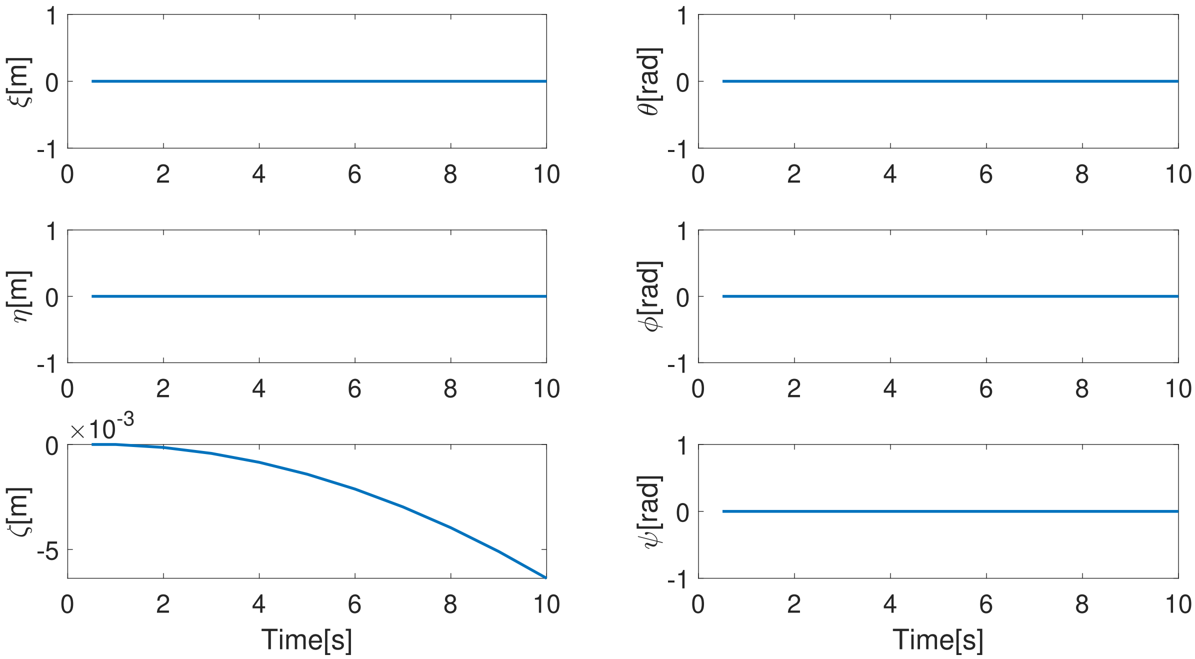


Figure D.2. MODEL STATES WITH CENTER OF MASS ALIGNED

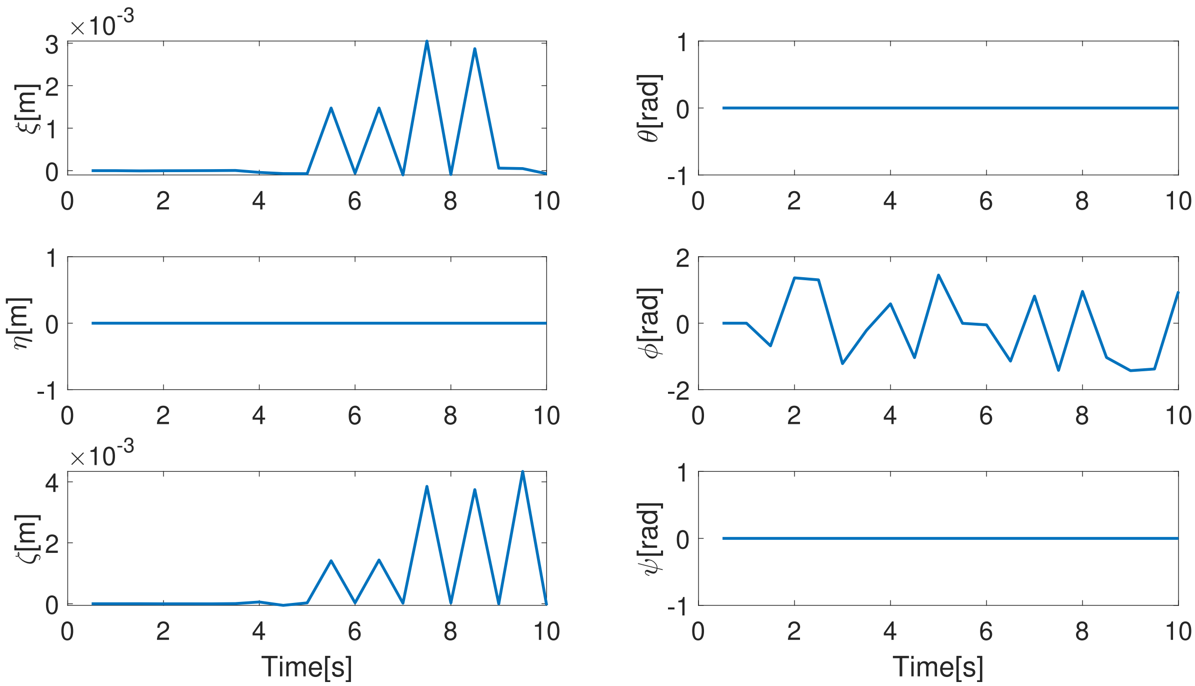


Figure D.3. MODEL STATES WITH CENTER OF MASS TO THE LEFT

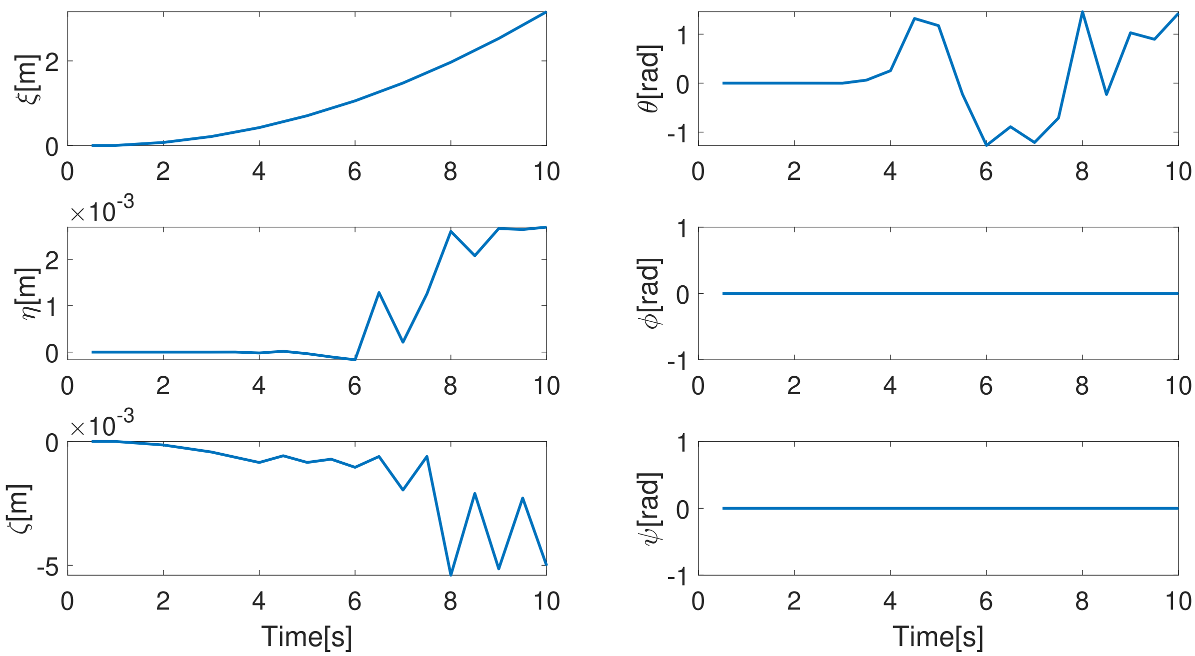


Figure D.4. MODEL STATES WITH PROPULSION FORCE

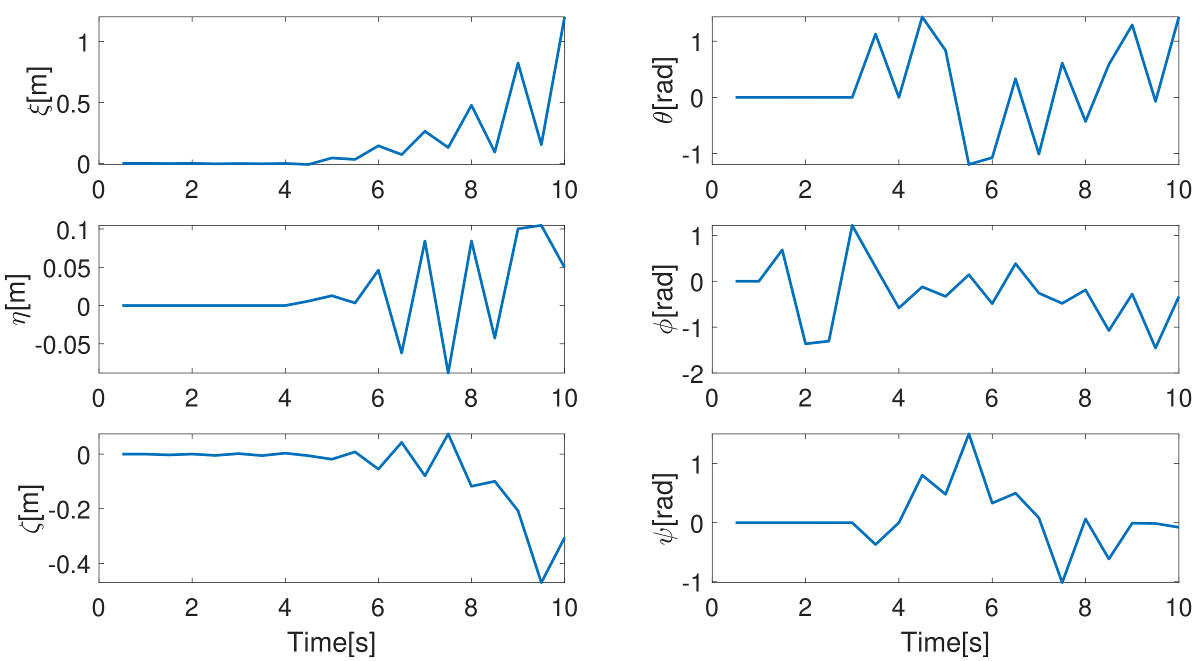


Figure D.5. MODEL STATES WITH CENTER OF MASS TO THE RIGHT AND PROPULSION FORCE

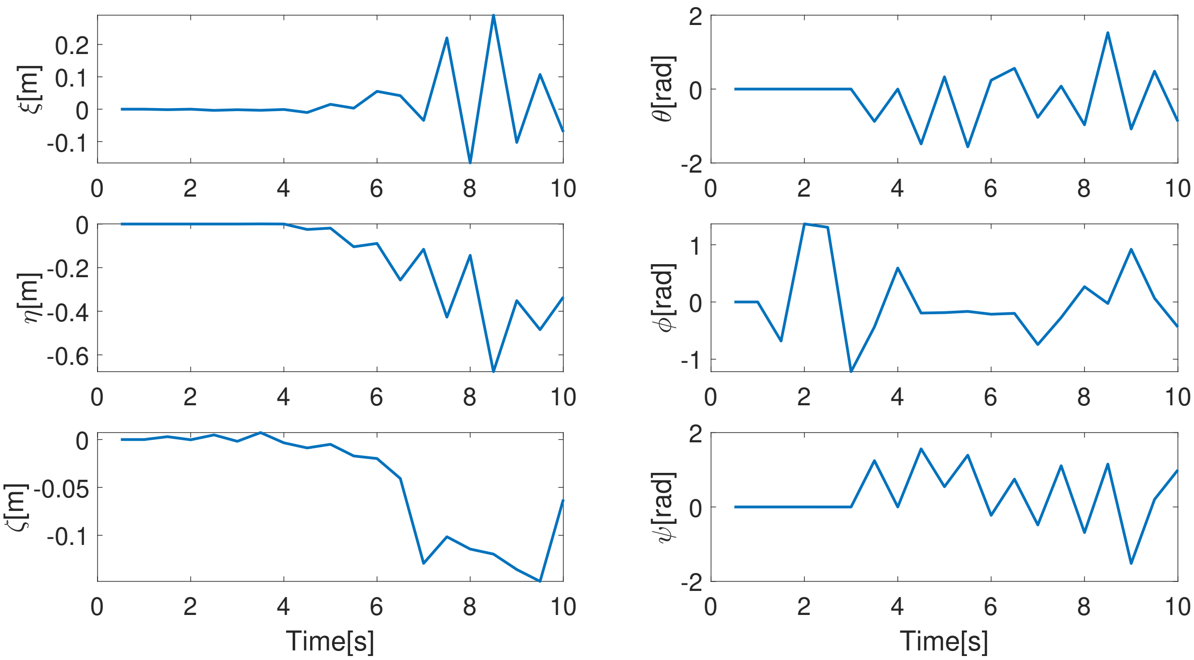


Figure D.6. MODEL STATES WITH CENTER OF MASS TO THE LEFT AND PROPULSION FORCE

Bibliography

[1] ALAM, K., RAY, T., AND ANAVATTI, S. G. A new robust design optimization approach for unmanned underwater vehicle design. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 226*, 3 (2012), 235–249.

[2] ALAM, K., RAY, T., AND ANAVATTI, S. G. A brief taxonomy of autonomous underwater vehicle design literature. *Ocean Engineering 88* (Sept. 2014), 627–630.

[3] ALAM, K., RAY, T., AND ANAVATTI, S. G. Design and construction of an autonomous underwater vehicle. *Neurocomputing 142* (2014), 16–29.

[4] ALLAL, A. A., MANSOURI, K., YOUSSFI, M., AND QBADOU, M. Toward a review of innovative solutions in the ship design and performance management for energy-saving and environmental protection. In *2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON)* (2018), IEEE, pp. 115–118.

[5] ANDERSON, J. M., AND CHHABRA, N. K. Maneuvering and Stability Performance of a Robotic Tuna1. *Integrative and Comparative Biology 42*, 1 (02 2002), 118–126.

[6] APARICIO, C. Development of a Biomimetic Propulsion System for an Autonomous Underwater Vehicle based on Parallel Mechanisms. Advances report 271119, Instituto Tecnológico Y de Estudios Superiores de Monterrey, Nov. 2019.

[7] ARAS, M. S. M., KASDIRIN, H., JAMALUDDIN, M. H., BASAR, M. F., AND ELEKTRIK, U. Design and development of an autonomous underwater vehicle (AUV-FKEUTeM). In *Proceedings of MUCEET2009 Malaysian Technical Universities Conference on Engineering and Technology, MUCEET2009, MS Garden, Kuantan, Pahang, Malaysia* (2009).

[8] BABU, M. N. P., KRISHNANKUTTY, P., AND MALLIKARJUNA, J. M. Experimental study of flapping foil propulsion system for ships and underwater vehicles and PIV study of caudal fin propulsors. IEEE, pp. 1–7.

[9] BELLINGHAM, J. G., ZHANG, Y., KERWIN, J. E., ERIKSON, J., HOBSON, B., KIEFT, B., GODIN, M., MCEWEN, R., HOOVER, T., PAUL, J., HAMILTON, A., FRANKLIN, J., AND BANKA, A. Efficient propulsion for the Tethys long-range autonomous underwater vehicle. IEEE, pp. 1–7.

[10] CLAPHAM, R. J., AND HU, H. isplash-i: High performance swimming motion of a carangiform robotic fish with full-body coordination. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), IEEE, pp. 322–327.

[11] CROWELL, J. Design challenges of a next generation small AUV. In *Oceans-San Diego, 2013* (2013), IEEE, pp. 1–5.

[12] HONG-JIAN, W., LONG, X., JUAN, L., AND HUI-NAN, Z. Design, construction of a small unmanned underwater vehicle. In *OCEANS-Bergen, 2013 MTS/IEEE* (2013), IEEE, pp. 1–6.

[13] HUANG, F., WANG, L., YANG, C., AND OTHERS. Hull form optimization for reduced drag and improved seakeeping using a surrogate-based method. In *The Twenty-fifth International Ocean and Polar Engineering Conference* (2015), International Society of Offshore and Polar Engineers.

[14] JINXIN, Z., YUMIN, S., LEI, J., AND JIAN, C. Hydrodynamic performance calculation and motion simulation of an AUV with appendages. In *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on* (2011), vol. 2, IEEE, pp. 657–660.

[15] JUN, B.-H., PARK, J.-Y., LEE, F.-Y., LEE, P.-M., LEE, C.-M., KIM, K., LIM, Y.-K., AND OH, J.-H. Development of the AUV 'ISiMI' and a free running test in an Ocean Engineering Basin. *Ocean Engineering 36*, 1 (Jan. 2009), 2–14.

[16] KATZSCHMANN, R. K., DELPRETO, J., MACCURDY, R., AND RUS, D. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics 3*, 16 (2018), eaar3449.

[17] KATZSCHMANN, R. K., MARCHESE, A. D., AND RUS, D. Hydraulic autonomous soft robotic fish for 3d swimming. In *Experimental Robotics* (2016), Springer, pp. 405–420.

[18] KOCA, G. O., KORKMAZ, D., BAL, C., AKPOLAT, Z. H., AND AY, M. Implementations of the route planning scenarios for the autonomous robotic fish with the optimized propulsion mechanism. *Measurement 93* (2016), 232–242.

[19] KRUUSMAA, M., FIORINI, P., MEGILL, W., VITTORIO, M. D., AKANYETI, O., VISENTIN, F., CHAMBERS, L., DAOU, H. E., FIAZZA, M. C., JEŽOV, J., LISTAK, M., ROSSI, L., SALUMAE, T., TOMING, G., VENTURELLI, R., JUNG, D. S., BROWN, J., RIZZI, F., QUALTIERI, A., MAUD, J. L., AND LISZEWSKI, A. FILOSE for Svenning: A Flow Sensing Bioinspired Robot. *IEEE Robotics Automation Magazine 21*, 3 (Sept. 2014), 51–62.

[20] LAU, W. P., ZHONG, Y., DU, R., AND LI, Z. Bladderless swaying wire-driven robot shark. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)* (2015), IEEE, pp. 155–160.

[21] LAUDER, G., AND DRUCKER, E. Morphology and Experimental Hydrodynamics of Fish Fin Control Surfaces. *IEEE Journal of Oceanic Engineering 29*, 3 (July 2004), 556–571.

[22] LAUNDER, B. E., AND SPALDING, D. B. *Lectures in mathematical models of turbulence [by] B. E. Launder and D. B. Spalding*. Academic Press London, New York, 1972.

[23] MORENO, H. A., SALTARÉN, R., PUGLISI, L., CARRERA, I., CÁRDENAS, P., AND ÁLVAREZ, C. Robótica Submarina: Conceptos, Elementos, Modelado y Control. *Revista Iberoamericana de Automática e Informática Industrial RIAI 11*, 1 (Jan. 2014), 3–19.

[24] RENILSON, M. *Introduction*. Springer International Publishing, Cham, 2018, pp. 1–11.

[25] ROPER, D. T., SHARMA, S., SUTTON, R., AND CULVERHOUSE, P. A review of developments towards biologically inspired propulsion systems for autonomous underwater vehicles. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 225*, 2 (May 2011), 77–96.

[26] SALAZAR, R., FUENTES, V., AND ABDELKEFI, A. Classification of biological and bioinspired aquatic systems: A review. *Ocean Engineering 148* (Jan. 2018), 75–114.

[27] SFAKIOTAKIS, M., LANE, D., AND DAVIES, J. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering 24*, 2 (Apr. 1999), 237–252.

[28] SHUKLA, A., AND KARKI, H. Application of robotics in offshore oil and gas industry—A review Part II. *Robotics and Autonomous Systems 75* (Jan. 2016), 508–524.

[29] SINGH, J., GANDHI, D., SANGHANI, M., ROBI, P., AND DWIVEDY, S. Design and development of underwater robot. In *Robotics, Automation, Control and Embedded Systems (RACE), 2015 International Conference on* (2015), IEEE, pp. 1–7.

[30] SU, Y., ZHAO, J., CAO, J., AND ZHANG, G. Dynamics modeling and simulation of autonomous underwater vehicles with appendages. *Journal of Marine Science and Application 12*, 1 (2013), 45–51.

[31] TRIANTAFYLLOU, M. S., AND TRIANTAFYLLOU, G. S. An efficient swimming machine. *Scientific american 272*, 3 (1995), 64–70.

[32] WANG, B., WAN, L., XU, Y.-R., AND QIN, Z.-B. Modeling and simulation of a mini AUV in spatial motion. *Journal of Marine Science and Application 8*, 1 (2009), 7–12.

[33] WANG, W. H., CHEN, X. Q., MARBURG, A., CHASE, J. G., AND HANN, C. E. A Low-Cost Unmanned Underwater Vehicle Prototype for Shallow Water Tasks. In *2008 IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications* (Oct. 2008), pp. 204–209.

[34] WEBB, P. W. Stability and Maneuverability. In *Fish Physiology*, vol. 23 of *Fish Biomechanics*. Academic Press, Jan. 2005, pp. 281–332.

[35] WEN, L., WANG, T., WU, G., AND LIANG, J. Quantitative thrust efficiency of a self-propulsive robotic fish: Experimental method and hydrodynamic investigation. *IEEE/ASME Transactions on Mechatronics 18*, 3 (2012), 1027–1038.

[36] WU, Z., YU, J., YUAN, J., TAN, M., AND ZHANG, J. Mechatronic design and implementation of a novel gliding robotic dolphin. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Dec. 2015), pp. 267–272.

[37] YAKHOT, V., AND ORSZAG, S. A. Renormalization group analysis of turbulence. i. basic theory. *Journal of scientific computing 1*, 1 (1986), 3–51.

[38] YU, J., SU, Z., WU, Z., AND TAN, M. Development of a fast-swimming dolphin robot capable of leaping. *IEEE/ASME Transactions on Mechatronics 21*, 5 (2016), 2307–2316.

[39] YU, J., WANG, K., TAN, M., AND ZHANG, J. Design and control of an embedded vision guided robotic fish with multiple control surfaces. *The Scientific World Journal 2014* (2014).

[40] ZHANG, H., XU, Y.-R., AND CAI, H.-P. Using CFD software to calculate hydrodynamic coefficients. *Journal of Marine Science and Application 9*, 2 (2010), 149–155.

[41] ZOHURI, B. *Similitude Theory and Applications*. Springer International Publishing, 2015, p. 93–193.

[42] İNEKÇI, M., ARSLAN, S., GÜLAY, E., AND ŞEHIT YÜZBAŞI ADEM KUTLU SOK. Numerical simulations of hydrodynamic tests for underwater vehicles.

Edisson Andrés Naula Duchi

Currículum Vitae

PERSONAL INFORMATION

DOB:	August 21, 1993
Address:	Puerto Marquez 1121, Narvarte, Monterrey, Nuevo León
Status:	Single
Phone:	8127668300
Email:	edisson.naulad@gmail.com

AREAS OF INTEREST

Engineering Design, Industrial Maintenance and Control, Systems and Telecommunications, Process Improvement, R&D.

EDUCATION

Master Degree in Engineering Sciences 2017-2019
Instituto Tecnológico y de Estudios Superiores de Monterrey
Monterrey, Nuevo León

Bachelor in Electronics and Telecommunications Engineer 2011-2017
Universidad de Cuenca
Cuenca, Ecuador

Seminar on "5G Cellular Communications" 2016
"New transmission schemes for the next generation of 5G cellular communications: Mass MIMO and Heterogeneous Networks"
Escuela Politécnica Nacional
Cuenca, Ecuador

WORK EXPERIENCE

Apprenticeship 2016
Red Sísmica del Austro(RSA), Cuenca.
Prototype development:

- Charge Controller for batteries in RSA monitoring stations;
- Smart Multiplexer for repeater stations of the RSA monitoring system.