POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Development and Testing of a Biomimetic Ray Robot with Flexible Fins

Tesi di Laurea Magistrale in
Mechanical Engineering - Ingegneria Meccanica

Author: **Lorenzo Maffi**

Student ID: 968571
Advisor: Prof. Simone Cinquemani
Co-advisors: Giovanni Bianchi
Academic Year: 2021-22

Abstract

This thesis describes the development of a bioinspired autonomous underwater vehicle. Through evolution, nature adapted the swimming strategies of fishes to obtain a motion with high performances in terms of speed, agility, and efficiency. Bioinspired design exploits and mimics these techniques to develop novel solutions to substitute the screw propellers used until now eventually. The robot on which this work is about is inspired by the manta ray, which, thanks to its high maneuverability and efficiency of motion, is one of the most appealing to mimic.
This thesis is about the improvement of an already existing robot with flexible fins. The robot is completely revised, redesigning the external structure and changing the actuators to achieve better performance.
The software is upgraded, including motions to achieve maneuvers, real-time estimation of the robot orientation, and an improved user interface.
Experimental tests were performed to characterize the robot's behavior and understand the effect of each fin movement on its orientation and the influence of different values of the motion parameters, both during rectilinear swimming and maneuvers. These tests have been carried out in a real environment, a lake with unknown disturbances.
Finally, a control on the robot orientation is introduced, acting on pitch and yaw angles, which reduces the pitch oscillation and keeps the yaw constant during the motion.

Keywords: Bioinspired design; manta ray robot; autonomous underwater vehicle; flexible fins; sensor fusion; orientation control.

Abstract in lingua italiana

Questa tesi descrive lo sviluppo di un veicolo autonomo sottomarino bioispirato.
Attraverso l'evoluzione, la natura ha adattato le tecniche di nuoto dei pesci per ottenere un movimento con alte performance in velocità, agilità ed efficienza. Il design bioispirato sfrutta e imita queste tecniche per sviluppare nuove soluzioni che potranno sostituire le eliche usate fino ad oggi. Il robot su cui è basato questo lavoro prende ispirazione da una manta, che grazie alla sua alta manovrabilità ed efficienza di movimento è tra i più interessanti da imitare.
Questa tesi tratta di come è stato migliorato un robot con pinne flessibili già esistente. Il design del robot è stato completamente rivisitato, riprogettando la struttura esterna e apportando modifiche agli attuatori per avere performance migliori.
Il software è stato migliorato, includendo movimenti per fare manovre, una stima in tempo reale dell'orientazione e un'interfaccia utente migliorata.
Test sperimentali sono stati fatti per caratterizzare il comportamento del robot, capendo l'effetto del movimento delle pinne sulla sua orientazione e vedendo l'influenza di diversi valori dei parametri del moto, sia durante il nuoto in linea retta che nelle manovre. Questi test sono stati fatti in un ambiente reale, un lago con disturbi incogniti.
Infine, un controllo sull'orientazione viene introdotto, che, agendo sugli angoli di beccheggio e imbardata, permette di ridurre le oscillazioni presenti sul beccheggio e mantiene costante l'imbardata durante il moto.

Parole chiave: Design bioispirato; manta robot; veicolo subacqueo autonomo; pinne flessibili; sensor fusion; controllo dell'orientazione.

Contents

Introduction

The thesis presented here is about the realization and experimental testing of a biomimetic robot manta ray.
Bioinspiration is an innovative way to design objects that exploits what nature has perfected over millions of years. This design strategy is used in many environments, like materials, structures, and motion generation, where the last one is of major interest in this work. This kind of imitation of nature started because animals' motion is much more efficient than other types of motion ever created by mankind [23].
One of the fields in which natural locomotion strategies outperform current technology is underwater locomotion. Since the second half of the XIX century, boats, submarines, and every other kind of aquatic vehicle have been moved by propellers, which allow them to reach high velocities and cover big distances. On the other hand, they have very low efficiency, around 40-50% [26]. Moreover, the maneuverability and acceleration of these vehicles are limited [14], they are big, and produce a lot of noise that disturbs wildlife. Due to these characteristics, screw-propelled autonomous underwater vehicles (AUVs) are not suitable for backdrop exploring and marine life monitoring [15]. It is, therefore, of great interest to study and reproduce fish movements for AUV propulsion.
The uses of AUVs are multiple: they can be used for commercial operations, such as exploration of the seabed for oil and gas or locating ship and plane wrecks, for military applications, for example for submarine warfare or reconnaissance, and for research porpoises, like ocean mapping or water analysis [1].
The robot presented in this thesis has already been designed and partially built, but had many problems. Therefore, the objective of the work was to fix the problems of the robot and improve it so to make it work, then test it to have a system characterization, and lastly, start the development of a control system.
This thesis is organized as follows:
In Chapter 1, "State of the art", an analysis of the robot created in the last years is reported, and the way they replicate the manta ray motion is explained. In Chapter 2, "Initial conditions of the robot", an overview of the state of the robot is done to understand the starting point of this work better. Then, in Chapter 3, the changes and

upgrades are explained in addition to the technique used to waterproof the system. In Chapter 4, the method used to obtain the real-time orientation measure is explained. In Chapter 5 the software is described, and in Chapter 6 all the tests that were done are reported and analyzed. Finally, in Chapter 7, the conclusions are drawn and some further possible developments are described.

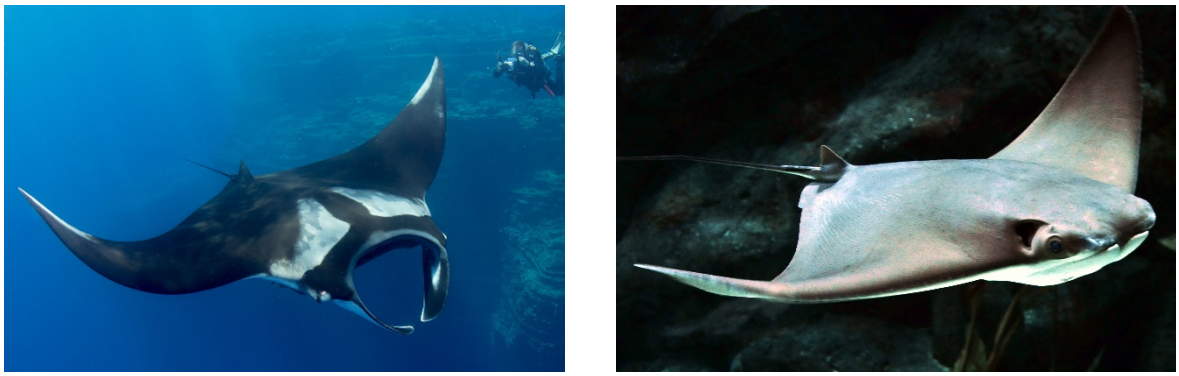
Figure 1: Final robot

1 | State of the art

In this Chapter, the movement of manta rays is analyzed accurately to better understand the reason of its high efficiency and maneuverability. Then, an analysis of the state of the art of bioinspired robots is done through a classification depending on how many degrees of freedom (DOFs) the fins use to generate the locomotion. The motion of fishes can be roughly divided into two groups, depending on how the motion is generated: the Body-Caudal Fin (BCF) and the Median-Paired Fin (MPF) [23, 24].
The first one is used by 85% of the fishes. In this one, the locomotion is generated by the bending of their body. There is a propulsive wave that moves backward, up to the caudal fin, generating thrust [23, 24].
The second one uses a combination of pectoral, pelvic, dorsal, and anal fins to produce thrust. This kind of locomotion allows for better maneuverability and higher efficiency but with a lower swimming velocity [23, 24].
Manta rays belong to the MPF category because they use their big triangular fins to move. This swimming strategy is considered to be one of the best for efficiency and maneuverability [10]. In these fishes, waves propagate in two directions: one in the chordwise direction, from head to tail, and the other in the spanwise direction, from the body to the fin tip. The main contribution to the propulsion is given by the first one, while the other supports the motion and improves efficiency and stability.
The motion of different species is classified depending on the ratio between body length L and wavelength λ of the chordwise wave. If the ratio L/λ is higher than 0.5, the movement is considered oscillatory, otherwise, it is considered undulatory [21]. The second is used by the smallest species living near the seabed, which have very high maneuverability without reaching high speeds [12]. On the other side, the largest species use the first type of motion, which generates higher top speed but with lower agility in movement [21]. The cownose ray is in the family of the *Batoidea*. These species are characterized by a ratio L/λ equal to 0.4, which is around the crossover point; therefore, they have a good compromise between speed and maneuverability. In particular, if they flap their fins around 1Hz they can achieve a cruising speed of 1.2m/s [22].

1.1. Manta ray shape and movement

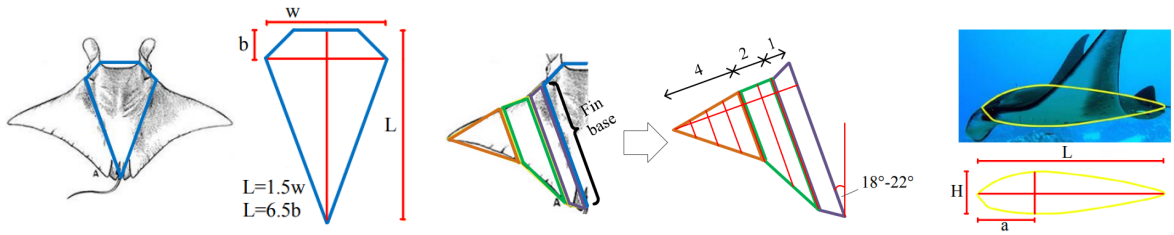
In literature, two species of Manta rays are considered, the *Manta Birostris* (Fig.1.1a) and the *Rhinoptera Bonasus* (also known as "*Cownose ray*", Fig.1.1b).



(a) Manta ray (b) Cownose ray

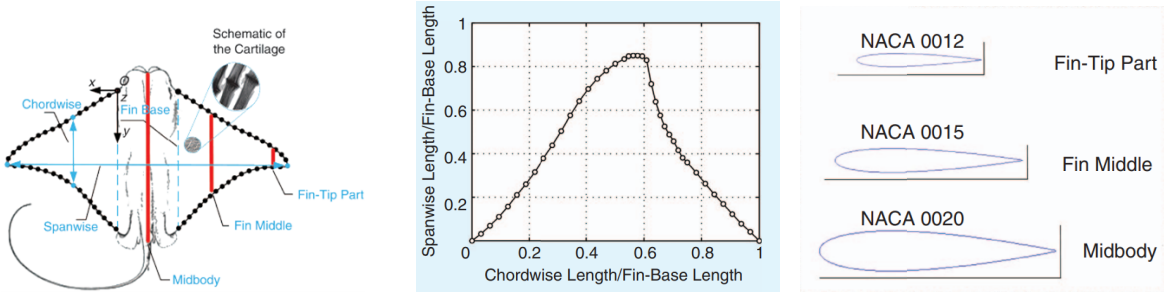
Figure 1.1: Types of fish studied

The body of a manta ray can be approximated as a diamond, as shown by Li et al. [15] in Fig.1.2. Instead, Cai et al. [2, 4] show that the cownose ray has a shape more similar to a rectangle due to its caudal fins being almost as large as the main body, as highlighted in Fig.1.3. The pectoral fins have the shape of triangles instead, and their section is similar to an airfoil profile, as shown in Fig.1.2c and Fig.1.3c.



(a) Body approximation (b) Fin approximation (c) Fin section

Figure 1.2: Shape of a manta ray approximated by Li et al. [15]



(a) Body approximation (b) Fin shape (c) Fin section

Figure 1.3: Shape of a cownose ray approximated by Cai et al. [2, 4]

After having seen how the shape is approximated, some nomenclature should be introduced, and this is done in Fig.1.4.

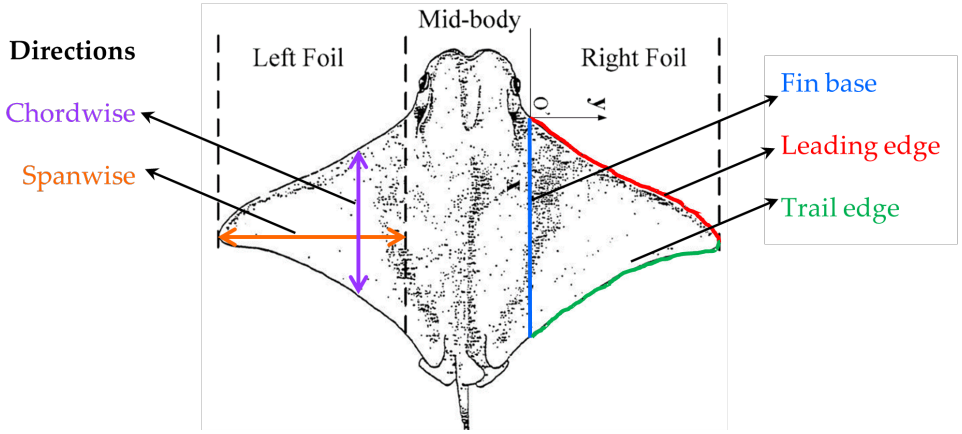


Figure 1.4: Manta ray significant nomenclature highlighted

The wave producing forward thrust propagates in two directions: the *chordwise* direction, parallel to the fin base (purple in Fig.1.4), and in the perpendicular *spanwise* direction (orange in Fig.1.4). Thanks to these two waves, four forces depicted in Fig.1.5 are generated, whose sum is the thrust force of the manta ray. The interaction between the fin and the water generates three forces, which are directed in the chordwise, spanwise and vertical directions. The vertical one is neglected in Fig.1.5 because it does not contribute to the thrust generation. Nevertheless, this force is a periodic component that is of key importance for the ray to keep its altitude while swimming.

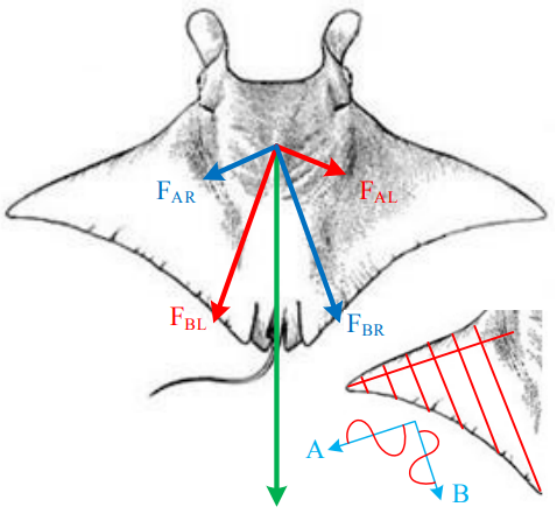


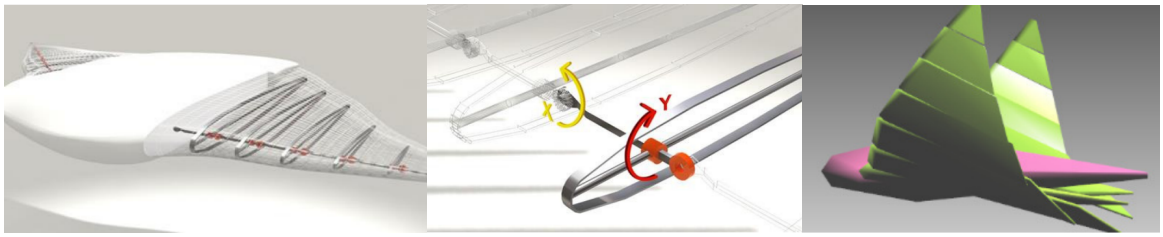
Figure 1.5: Manta ray forces generated by the propagating waves

1.2. Manta ray inspired robots

To mimic the flapping movement of the manta ray's pectoral fin, many different solutions have been found, all with different accuracy and complexity. The main difference between them depends on how many degrees of freedom ($DOFs$) the fin has. There are three main types of robots, with 1, 2, and 3 or more DOFs. The robot developed within this thesis work is of the 1-DOF type, but to have a complete overview of all the robots that were created in the last years, all three types are reported with some examples.

1.2.1. Three or more DOFs robots

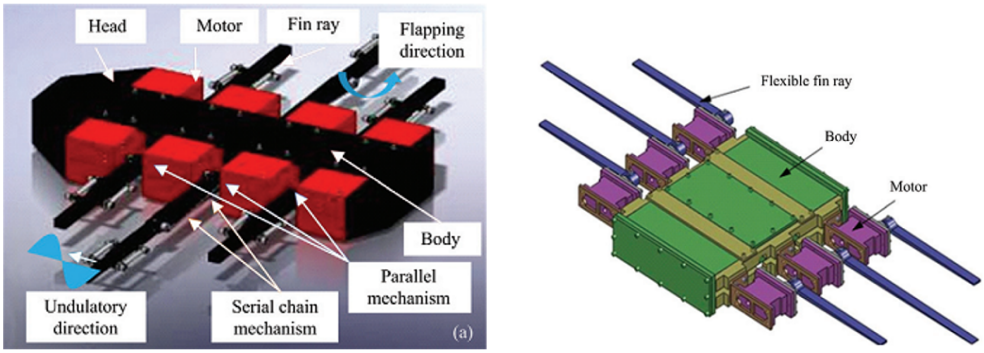
These kinds of robots use three or more DOFs to emulate the fin flapping. In particular, two categories are reported: one with 14 DOFs and one with 3 DOFs.
Li et al. [15] proposed the one with 14 DOFs. In this one, the pectoral fins are divided into seven segments, and each one can rotate around two axes, A and B, as shown in Fig.1.6b. Hence, the robot has 14 DOFs for each fin in order to generate the locomotion. Since this robot is only a concept that was only simulated on a PC, it is still not clear if they plan to actuate all of the 14 DOFs or if some of those will be passive.



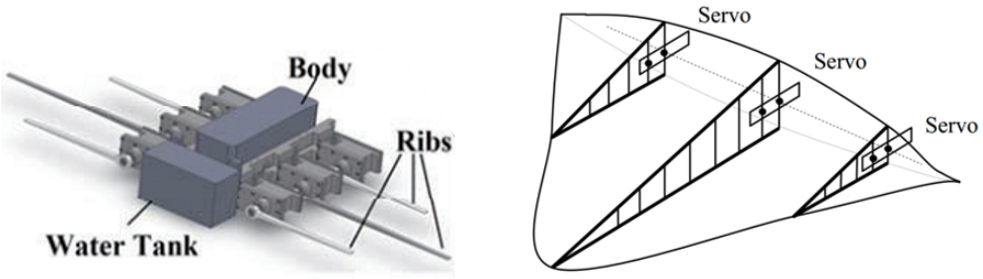
(a) Chassis (b) Mechanism (c) Simulated robot

Figure 1.6: Manta ray by Li et al. [15]

For what concerns the three DOFs robots, their fins are moved by three independent ribs, moved by a mechanism that allows to perform a motion that replicates the real fin curvature, while the phase gap between each rib creates the movement of the wave along the traveling direction.
Three examples of these robots are reported, the RoMan and two other prototypes. The RoMan, developed at the Nanyang Technological University of Singapore by the team of professor Low, was developed in 2009 and, since then, has had three upgraded versions. In RoMan-I, the ribs were divided into segments, and the bending was done through a serial chain mechanism. In RoMan-II [26] and III [16], the ribs were made of a flexible material; therefore, bending was performed in a passive manner; the amount of deformation depends on the stiffness of the chosen material. In RoMan-IV [8] instead, there is a compliance mechanism called Fin Ray Effect® to precisely control the ribs deformation.



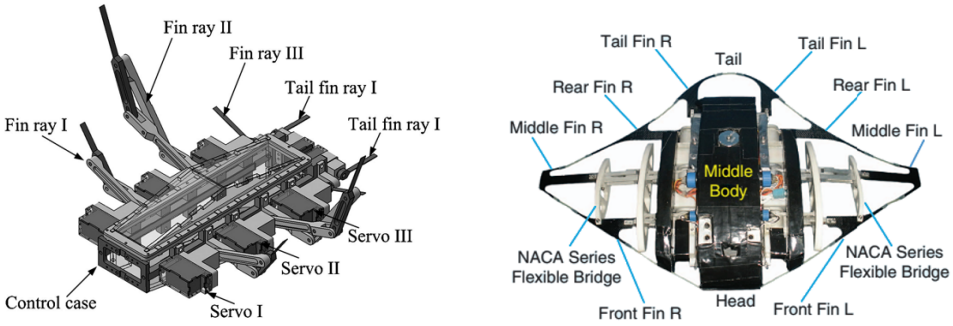
(a) RoMan I (b) Roman II



(c) RoMan III (d) RoMan IV, fin structure

Figure 1.7: RoMan series

Another prototype was made by Cai et al. [3]. at the Beihang University in Beijing. This robot has an innovative mechanism to perform the ribs bending and obtain a realistic fin bending. In 2018 an upgraded version of this robot was presented, which maintained the same mechanism.



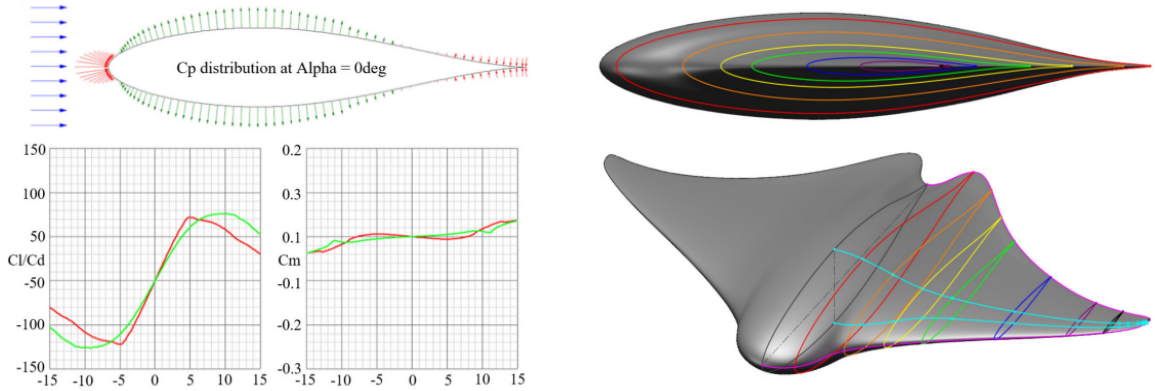
(a) Mechanism in 2012 (b) Improved robot in 2018

Figure 1.8: Manta ray of Cai et al. [3]

Polimi also developed a 3 DOF robot using the same mechanism as Cai et al. [3]. This robot is definitely bigger and more complex than the one on which this work is about, but since it is still of the Polimi, it is good to keep it as a comparison.

Figure 1.9: Polimi's 3 DOFs manta ray

The last prototype analyzed is also the most recent one, developed by Ghen et al. [5]. This introduces a novel solution that tries to combine a three-dimensional biomimetic shape that allows achieving higher speed with a multi-DOF pectoral fin, which allows better maneuverability. The manta ray shape was obtained by a combination of NACA63-20, 63-18, 0015, and 0012 profiles merged together to obtain the better match possible between the real fish and the robot (Fig.1.10).

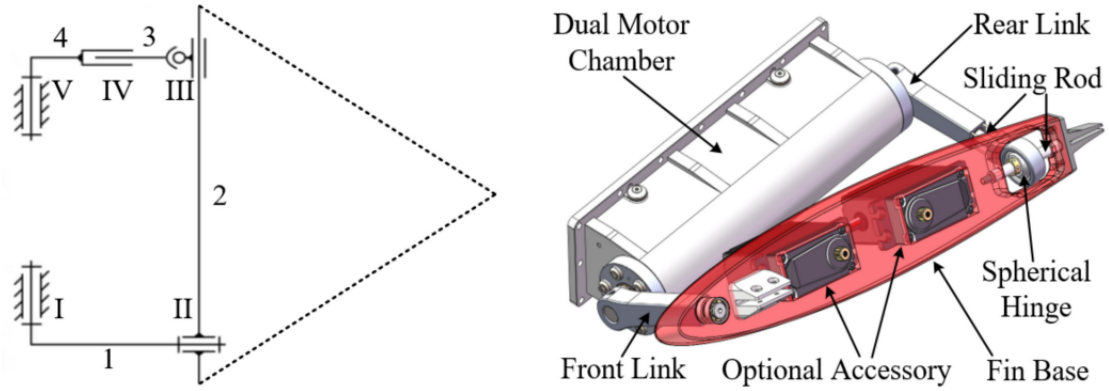


(a) 2D profile calculation and aerodynamic coefficients

(b) 3D surface created with multi-section lofting and spline control

Figure 1.10: Geometry of the robot by Ghen et al. [5]

The propelling mechanism was instead obtained through the union of 5 kinematic pairs actuated by two motors (Fig.1.11). The motors directly act on links 1 and 4. These two motors, in addition to the spherical hinge in 3, are crucial to obtain the biomimetic flapping of the fins.



(a) Mechanism scheme (b) 3D components

Figure 1.11: Multi-DOF bionic propelling mechanism by Ghen et al. [5]

1.2.2. Two DOFs robots

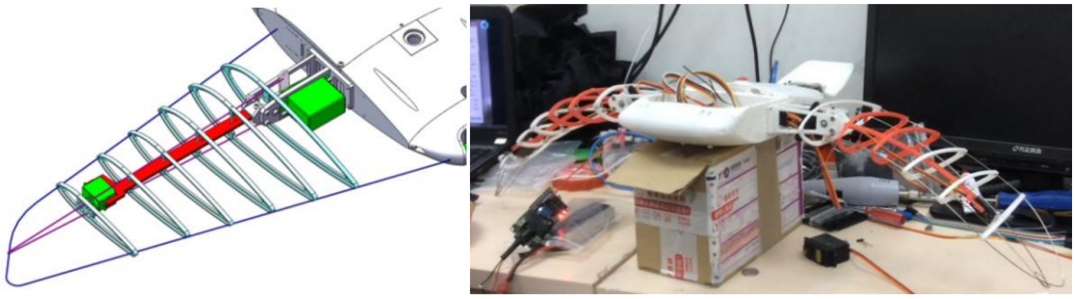
A simpler solution with respect to the 3 or more DOFs is this 2 DOFs configuration, which uses at most two servomotors per fin to obtain the fin deformation. Here, one motor is used to perform the flapping of the fin, and the other is used for rotation. This is possible because the wave number of manta ray fins is small, as said in the introduction.
This principle is used by the prototype created by Ma et al. [17]. In their studies, they discovered that if the rotation is applied on the fin tip, it is much more effective. Due to the limited space inside the fin, a servo acting directly in that position could not be placed; therefore, a flexible shaft was used to connect the motor in the central body and the fin tip. The flapping is instead given by a motor placed on the leading edge base.

(a) Fin scheme (b) Fin photo (c) Complete prototype

Figure 1.12: 2 DOFs robot by Ma et al. [17]

Another prototype was built by Zhang et al. [25], which has some interesting features. The twisting of the fin is provided by a servomotor (green in Fig.1.13a) placed at the end

of a flexible beam (red in Fig.1.13a), and the ribs are connected rigidly to this beam. The other servomotor in the central body pulls and releases two cables connected to the beam (in purple in Fig.1.13a), causing the deformation by the Fin Ray Effect®.



(a) Fin mechanism (b) Prototype

Figure 1.13: Robot by Zhang et al. [25]

The German company Festo [9] created a robot with the same principle but filled the flexible cover with helium, making it fly like a balloon. This means that the robot uses the flapping of the manta ray to fly and not to swim.

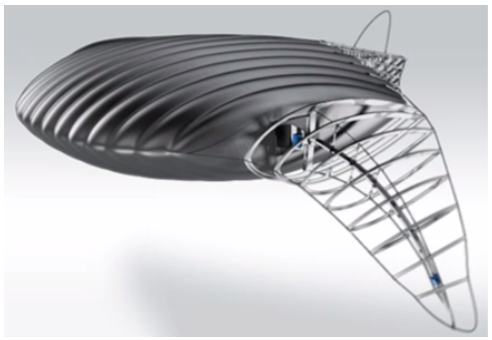
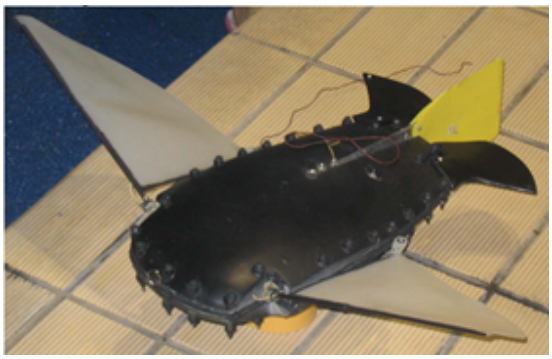


Figure 1.14: Flying manta ray robot by Festo [9]

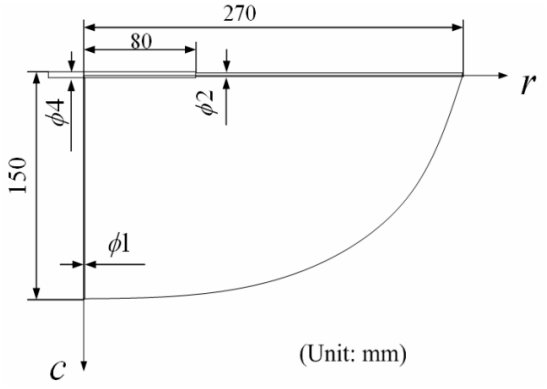
1.2.3. One DOF robots

The one DOF solution is the simplest of all because it uses only one actuator to move the fin, and the flapping is due to the passive movement of a flexible material. Two prototypes are reported here, one from the Beihang University in Beijing and the other made by the National University of Singapore (*NUS*). Gao et al. [11] created the first prototype of their robotic fish (Fig.1.15a) with the pectoral fin made of an L-shaped structure (Fig.1.15b) of carbon fiber on which a flexible structure made of silicone is attached. In this first prototype, only one motor is used, which moves both fins using a

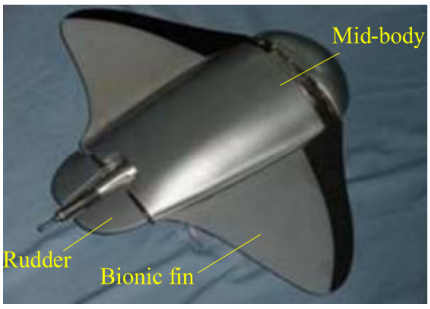
transmission system. This first prototype has been then improved [2]. The new shape is much similar to a real manta ray; the leading edge is made of fiberboard instead of carbon fiber and has a reduced width. These changes made to the fins were done to reduce the bending stiffness and obtain a more realistic deformation.



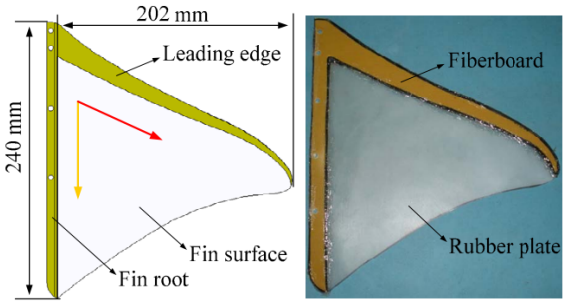
(a) First prototype.



(b) First prototype fin shape.



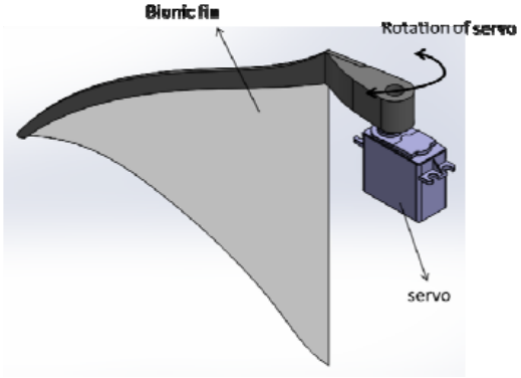
(c) Second prototype.



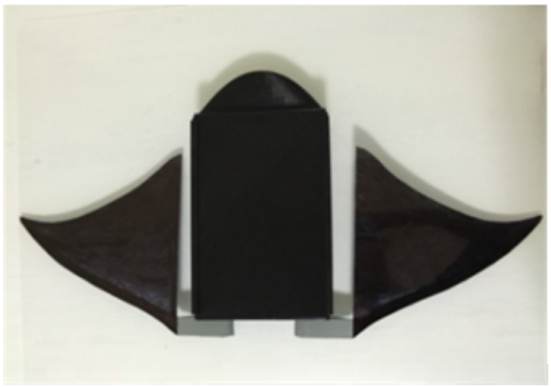
(d) Second prototype fin shape.

Figure 1.15: Robot created in the Beihang university

The other prototype made by *NUS* was created by Chew et al. [6]. Differently from the previous one, this has the fins root not connected to the body; therefore, they are completely free to oscillate. The leading edge is made of 3D-printed ABS, while the fin is made of PVC with uniform thickness. This research exploited fin thickness, oscillation amplitude, and frequency [6], fin shape and section [7]. The first prototype (Fig.1.16b) was afterward improved (Fig.1.16c) to obtain a speed up to twice its body length per second with an autonomy of 10 hours.



(a) Fin scheme.



(b) First prototype.

(c) second prototype.

Figure 1.16: NUS manta ray robot

2 | Initial condition of the robot

This chapter serves as an introduction to the state of the robot at the beginning of this work, highlighting all its main parts in detail so to better understand which changes/upgrades were done afterward.
This robot is composed of two main parts, the core, and the external structure. The core is the central box that contains all the electronics, on which some extensions have been mounted to allocate the camera, on/off button and connector that allows to charge the batteries and connect the Arduino board to the PC. Then, this core is mounted on a chassis made of aluminum, which was obtained from a sheet that was laser-cut and then bent. On this chassis, also the external structure is mounted, which is composed of 3D printed parts that give the robot the characteristic airfoil profile of manta rays [2, 4, 15]. The airfoil profile also extends to the caudal fins, while for the pectoral ones, the shape is different and explained in Section 2.3.

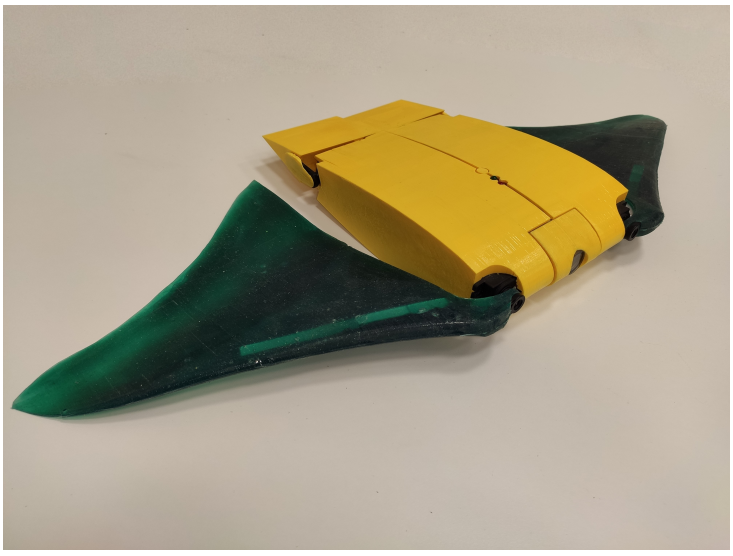


Figure 2.1: Old design of the manta ray

2.1. Mechanical design

In this section, the mechanical design is analyzed and explained in detail, starting from the core with all its extensions and how they were connected, then going on to the chassis and closing with the external structure and how it was chosen.

2.1.1. Core

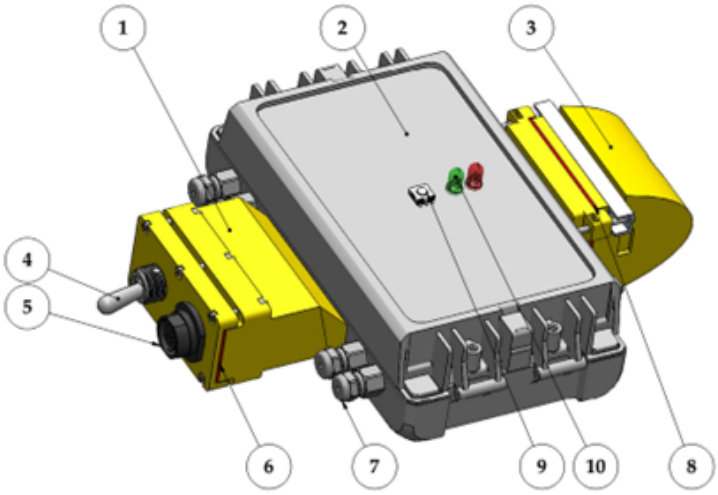


Figure 2.2: Core of the robot

1	Rear extension
2	Central box
3	Front extension
4	Power switch
5	Connector
6	Rear gasket
7	IP68 cable gland
8	Front gasket
9	Button
10	LEDs

Table 2.1: Components marked Fig.2.2

The core of the manta ray is composed of a central box (2), rear (1), and front (3) extensions. The central box contains the main electronics, like Arduino DUE, the batteries, and the accelerometer. The rear and front extensions were introduced to have more space

to place all the remaining electronic components and to better exploit the space available inside the external structure.

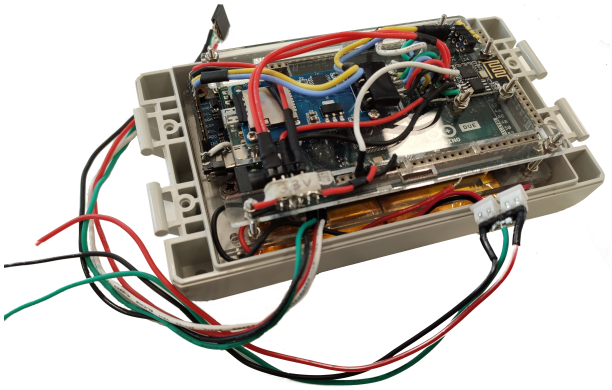


Figure 2.3: Central box containing the electronics of the robot

The central box is made of ABS plastic and is certified to be IP68, with its cover tightened by four screws. In this box, some holes were made, more precisely three on the back, one on the front, and one on top of it. These holes are there to make possible the cable passage towards the two extensions and to the motors; the one on top serves for the LEDs and button.

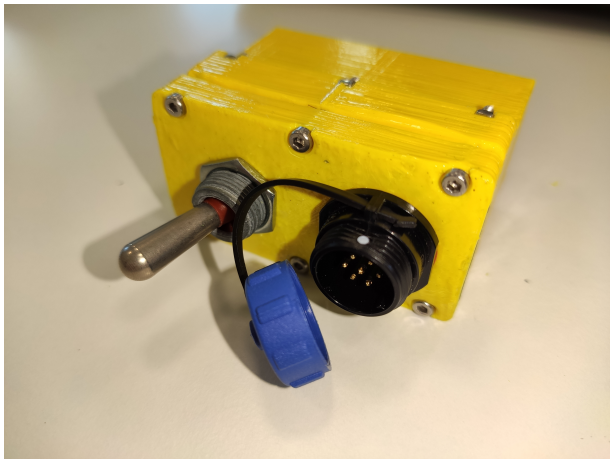


Figure 2.4: Rear part of the core

Moving on, we have the rear part, which has the switch to turn on/off the power supply and an IP68 connector that allows to charge the batteries and connect the PC to Arduino through a USB cable.

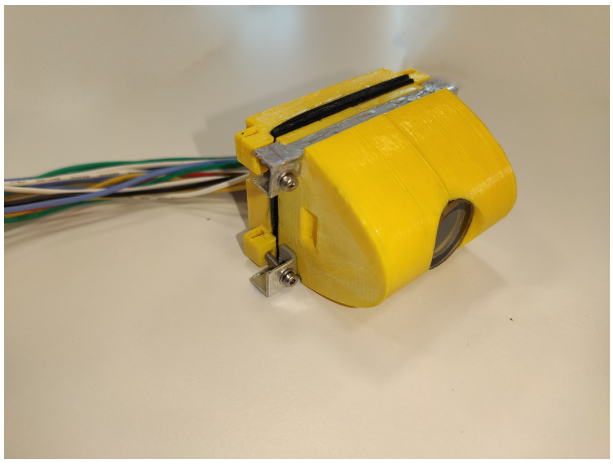


Figure 2.5: Front part of the core

The last part of the core is the front one, where the camera is found. This component is not yet connected to Arduino since it is not a fundamental part for the initial tests.
These two extensions were attached to the central box through a waterproof sealant glue.

2.1.2. Chassis

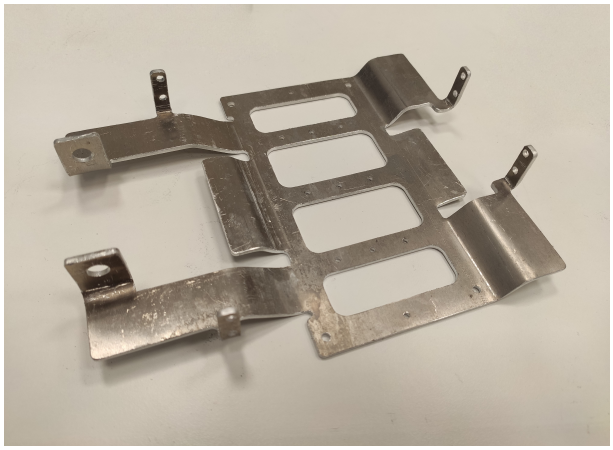


Figure 2.6: Aluminium chassis

The chassis is a fundamental part of the robot, on which both the central box, the external structure, and the servomotors are screwed. This chassis is obtained from a laser-cut plate, which was then bent by hand. Hand bending implies some defects. Nonetheless, it still does its functions well enough, even with those little imperfections.

2.1.3. External structure

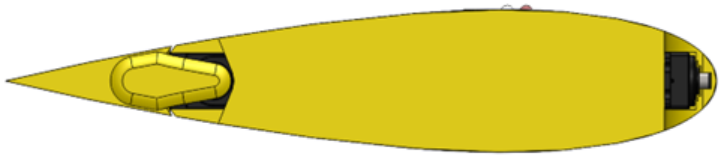


Figure 2.7: External body

The external structure was 3D printed in ABS plastic. It has a NACA0020 airfoil shape, which was the smallest possible airfoil profile that contained the central box. Also, the rear fins are a continuation of that profile, as can be seen in Fig.2.7.
The profile is obtained through the following formula:

$$y_t = 5 \cdot t \cdot [0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4] \tag{2.1}$$

Where:

- x is the position along the chord from 0 to 1 (0 to 100%);
- y_t is the half thickness at a given value of x (centerline of the surface)
- t is the maximum thickness as a fraction of the chord, so t gives the last two digits in the NACA 4-digit denomination.

This formula does not consider the chord length CL, which in this case was chosen at $260mm$, selected by looking at the chassis and the box dimensions. Considering also the chord length, the formula becomes:

$$y_t = CL \cdot 5 \cdot t \cdot [0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4] \tag{2.2}$$

This airfoil profile has a maximum thickness of $52mm$, which is enough to contain the box of $40mm$ height.
This external shell is composed of two parts connected to the chassis through screws, while between them, there are some little plastic bars that serve as a connection by shape. These parts were a little too wide, so to reinforce those and increase the resistance, some brass wire was used in both parts.

2.2. Electronic components

Inside the central box, the microcontroller, the batteries, and other electronic components are contained. Its electronic scheme of it can be seen in Fig.2.8.

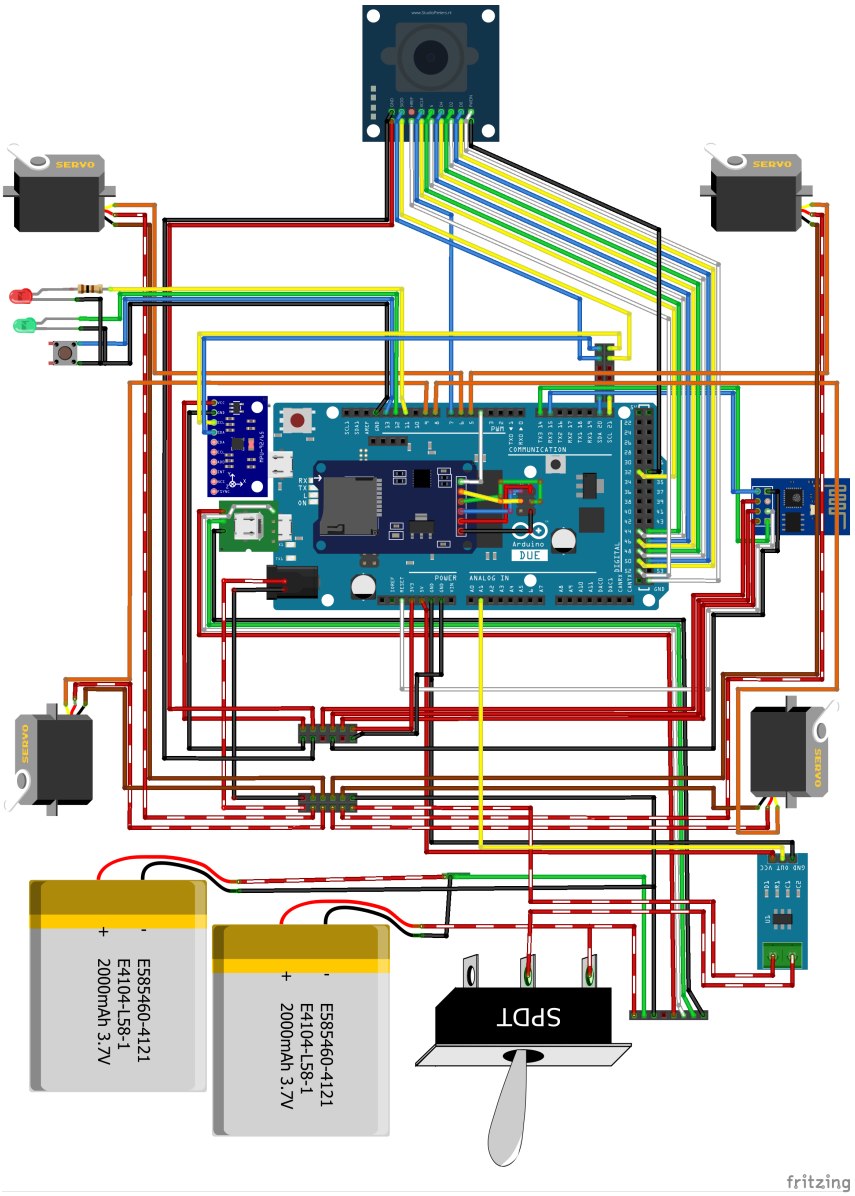


Figure 2.8: Electric scheme of the manta ray

From the scheme, it is possible to understand how every component is connected to the Arduino board. In particular, the red and white cables are the 7.4V ones; they start from the battery, pass inside an ammeter and go to feed the motors and the board; the brown ones are the ground of the motors, while the ground of all other components is drawn

in black; the red cables are the 5V and the 3.3V that are provided by Arduino to power the ammeter (5V), the Wi-Fi module and the MPU (3.3V). All the other cable colors are used for signals. The ammeter is in the bottom right of the figure, the Wi-Fi module is the blue chip on the right, the SD card reader is on top of Arduino, and the MPU is on the top left of it, the camera is on the top part of the figure.

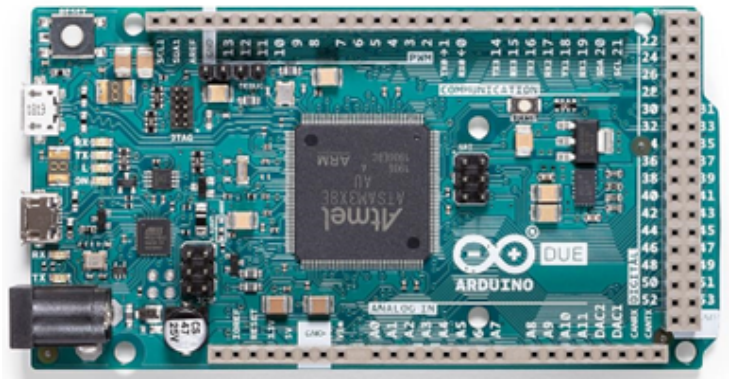


Figure 2.9: Arduino due microcontroller

As can be seen in Fig.2.9, the microcontroller chosen is an Arduino DUE, because it has enough ports to govern everything and has 524288 bytes of flash memory, which allows to host a complex program. In particular, this has the highest computational power among all the Arduino boards and can be programmed through a USB port. The only problem with this board is the dimensions because this Manta Ray was created to be as small as possible, so having a big controller does not help.

Figure 2.10: Batteries

The robot is powered by the LiPo batteries in Fig.2.10. They are connected in series to reach 7.4V of voltage, which is needed to power the motors and the board. In reality, those batteries were not enough to power all four motors, therefore the rear fins did not move at the beginning of the work and the batteries had to be changed.

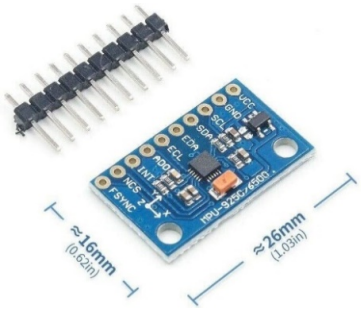


Figure 2.11: GY-MPU9250

The next electronic component analyzed is the MPU, which is a sensor that contains an accelerometer, a gyroscope, and a magnetometer, which allows to measure respectively linear accelerations, angular velocities, and magnetic field. This is of fundamental importance when data need to be obtained during the tests.

Figure 2.12: OV7670 camera module

Then, there is the camera module, which is placed in the front extension of the core. This camera is a cheap sensor with a bad resolution, so this is the reason why it was chosen not to attach it to the Arduino in the first place; the second reason is that it was also thought to introduce a way to set an autonomous navigation system in the future, but it is something very advanced and not feasible in a starting phase.

Figure 2.13: Micro SD reader

The micro-SD reader is used to store the data arriving from the MPU sensor and the ammeter during the tests and save the videos and images from the camera. Clearly, the videos were not present because the camera was not connected, but when it will be possible they can be stored here.

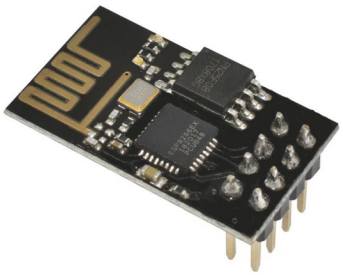


Figure 2.14: WI-FI module ESP8266-01

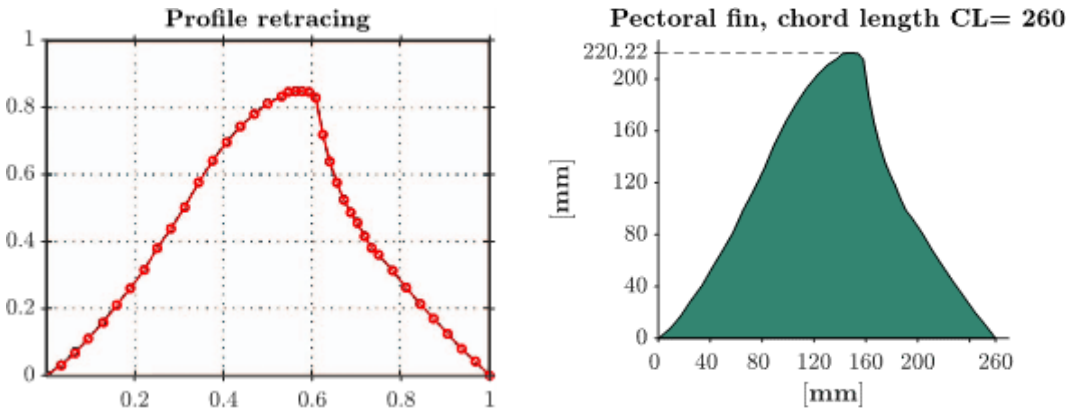
The WI-FI module was installed to allow the communication PC-Arduino DUE without the cable. By connecting to this WI-FI network and going on its page, some instructions can be delivered to Arduino such as parameter change, printing files, etc...
It is not possible to upload the program or communicate in real-time with it during the experiments since underwater, the Wi-Fi has a very low range and cannot reach the coast even if it is just a couple of meters away. For these reasons, the SD card was an obliged addition.

Figure 2.15: TD-8320MG servomotor

The last component to be analyzed is the servomotor. These are waterproof components with an internal microcontroller that controls their rotation. They deliver 1.96 $N \cdot m$ without load, with a working frequency of 333 Hz.

2.3. Pectoral fins design

The last section of the mechanical design is about the fins. The profile of the fins is obtained using the studies made by Cai et al. [2, 4]. In particular, the profile used is shown in Fig.2.16.



(a) Fin profile obtained in MATLAB (b) Fin profile considering the chord length

Figure 2.16: Both the dimensionless profile and the one with real dimensions are shown

For what, instead, concerns the cross-section, it was calculated by dividing the former profile into 10 sections and defining a thickness for each of them, then connecting these ten

profiles. In the following pictures, this process is illustrated: in Fig.2.17, the dimensionless section used by Riggs et al. [20] is shown. This profile was then multiplied by the length of the section to obtain the 10 different sections which were interpolated afterward.

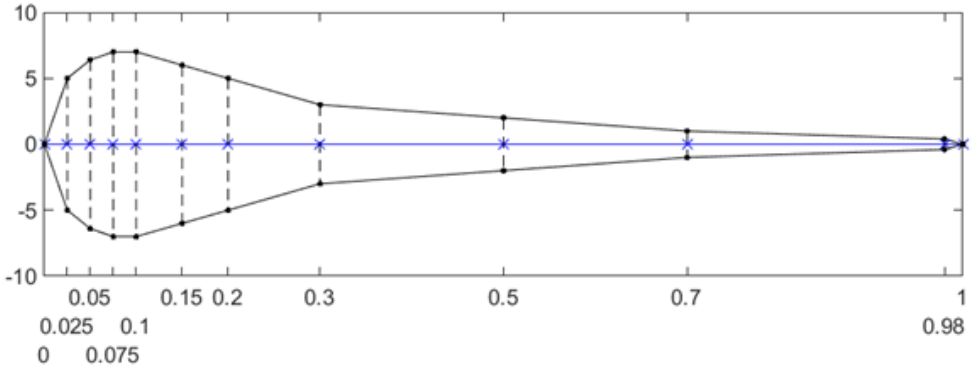
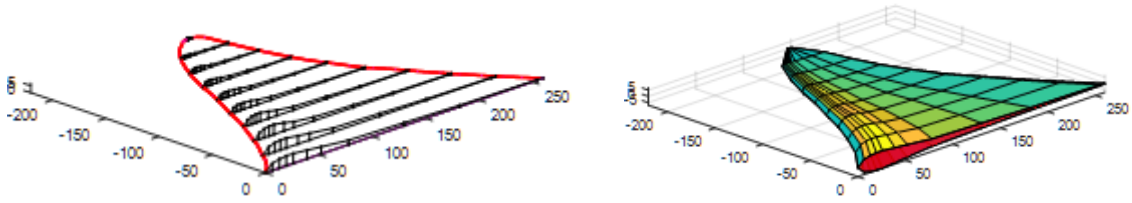


Figure 2.17: Dimensionless profile thickness



(a) Fin profile divided in the 10 sections

(b) Final fin profile

Figure 2.18: Final shape of the fins, in both profile and thickness

These fins were realized in silicone rubber, to be flexible and exert thrust through passive motion when moved underwater. To be actuated, they have a stick that is inserted in the leading edge of the fin that connects it to the servomotor. This also keeps the thickest part straight, resulting in a motion able to give enough thrust to the Manta Ray when underwater.

3 | Upgrades

Before testing, some changes had to be made. Starting from the batteries, which were not enough to power all four servomotors; then, the box with the electronics had to be substituted with a bigger one due to some waterproofing problems, and consequently, also the external structure had to be modified; afterward, the core and the servomotors had to be completely waterproofed to avoid short circuits and malfunctioning generated by the water leaking in. Lastly, the robot had to be hydrostatically balanced through the use of some ballast in the correct positions to make it float underwater.

3.1. Batteries

The first problem to be addressed was that the batteries used before were not enough to power all four servomotors due to a low discharge rate, therefore, a new set of batteries was necessary.
The already fitted-in batteries were the MIKROE-1120; in Table 3.1, the technical data are reported. As already said in Section 2.2, these are two 3.7V LiPo cells connected in series, so to reach the 7.4V needed to feed the motors and the Arduino board.
The new batteries were bought from "Grepow", and, in particular, the ones selected are the GRP6134060. The specs of these batteries can be seen in Table 3.1.

	MIKROE-1120	**GRP6234060**
Rated voltage	$3.7V$	$3.7V$
Rated capacity	$2000mAh$	$1200mAh$
Max discharge current	$2A$ $(1C)$	$18A$ $(15C)$
Size (H*W*L)	$7{\cdot}63{\cdot}43.5mm$	$6.1{\cdot}34.5{\cdot}62.5mm$
Weight	$45g$	$24g$

Table 3.1: Technical data of the MIKROE-1120 and GRP6134060

These new battery cells are still LiPo, with a lower capacity but a much higher discharge

rate. The old batteries could not provide enough current to feed all the motors even though the voltage was correct. These batteries are, instead, oversized since the current needed by the motors is roughly 4A. The choice was taken in anticipation of a possible change of motors since, thanks to their 18A of maximum discharge current, will have no trouble feeding also more powerful actuators. This problem was solved with just a little loss in capacity, which may become a problem when a long working time has to be reached. For now, the robot does not have to cover long distances because it is still in the testing phase, and many things can change from now to when it will be finished. For these reasons, capacity was a parameter that could be sacrificed in favor of other more suitable for the type of work of this thesis. The new batteries can be seen in Fig.3.1.

Figure 3.1: GRP6134060

Other important parameters that were considered while choosing these batteries were their dimensions. As shown in Fig.3.2, the space occupied by the batteries is a bit less. This was important because the space inside the core is not much; therefore, these batteries allow to keep the occupied space almost constant. Moreover, these new ones are lighter than the old ones.

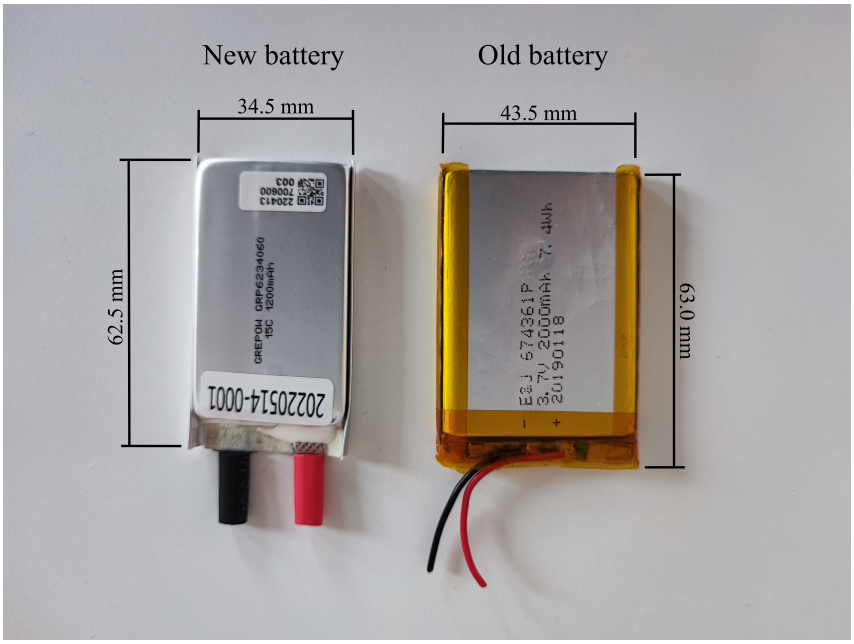


Figure 3.2: Batteries comparison

3.2. Hardware changes

As said in Section 2.1.1, the core is the part that contains all the main electronic components; it is, therefore, a very important part that must be waterproofed. The box itself was certified IP68 waterproof, but due to its extensions and cables that exit, holes were done in it, compromising the original water resistance. The tests to see if the waterproofing was successful were performed in a small plastic water tank in the laboratory.
Several tests were performed, and it was clear that the box had lost its original capability of being waterproof because the water was entering from a split between the top and the bottom part of the box. This happened because there were too many cables inside of it, therefore the top part was bent in the middle. This was confirmed after dismantling it because with the box completely closed and with nothing inside, the rear part was clearly not sealed with respect to the front part; obviously, if this was happening with the empty box, it must have been worse with all the electronics and cables inside.

3.2.1. New box selection

The choice of the new box had to consider that both the chassis and the external structure were custom-made for the old one. The external structure was 3D printed, so changing those components was not a big deal; moreover, since the new box must be bigger, they had

to be changed in any case. For what concerns the chassis, instead, it was not something that could be easily replaced, because the one already present had a three months lead-time and then had to be bent by hand. Therefore, the chassis remained the same. For this reason, the obvious choice was to buy the exact same box but taller.
The original one was a "Bopla B 140804 PC-V0 7035", with dimensions 80x151x40mm, the new one is instead a "Bopla B 140806 PC-V0 7035", of the same family with the same base dimensions but 20mm taller, which was enough to keep everything inside without any problem.
The new box had to be modified as the old one; therefore, holes had to be made to allow the passage of cables and to fit the front and back extensions. While doing this, also a restyling of the back and front parts was done to fit the new external structure.

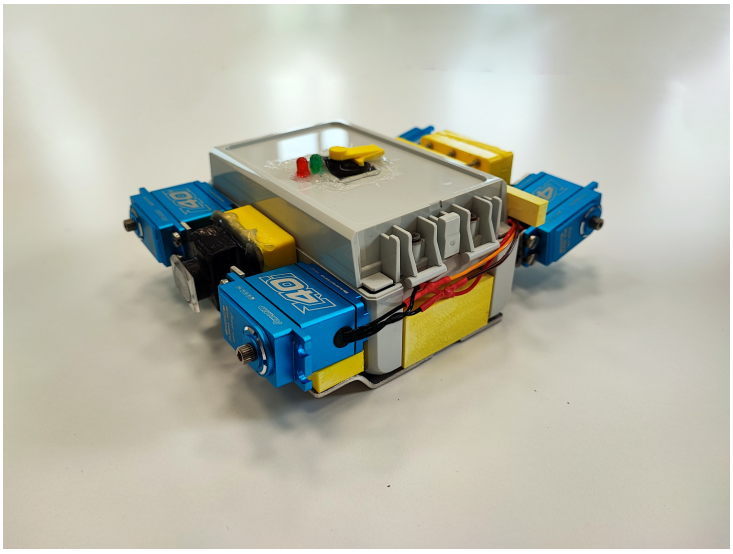
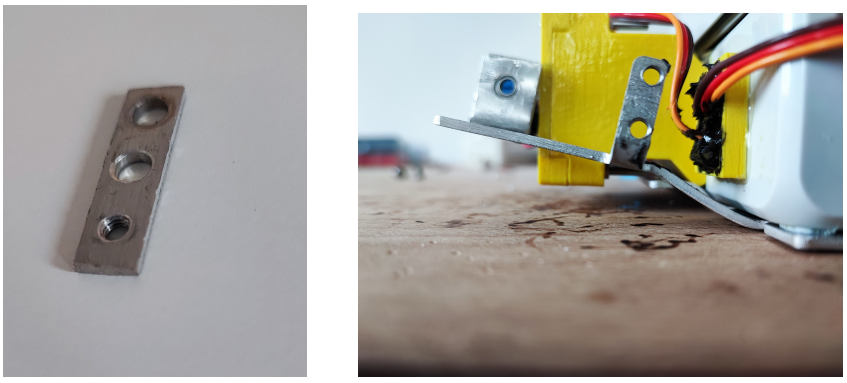


Figure 3.3: Final configuration of the core

3.2.2. External structure modifications

With a new box, the external structure was revised so to fit the new box. By using Equation 2.2 already explained in Section 2.1.3, it is possible to change the thickness, maintaining the same chord length of 260mm, so to fit the new box keeping the old chassis.
The objective was still to have a robot as slim as possible; therefore, optimization of the internal spaces had to be done. The old box was exactly symmetrical; therefore, with the symmetrical profile, everything was fine. Now, the box is 20mm taller, but the chassis is the same; therefore, when it is fixed on it, the symmetry is lost. For this reason, the symmetrical profile placed with the same middle axis as before created a lot of empty space

under the chassis. Two options were possible, the first one was to adopt an asymmetrical profile, and the second was to slightly modify the chassis to raise the motor position to the new midline, which was at 30mm from the box base and no more at 20mm.
The first option was more complex because it implied another problem characteristic of the asymmetric airfoil profile, which is a non-zero lift coefficient for 0 angle of attack. Since the profile is asymmetric, the top and bottom parts are different, and this generates a rotational moment that should be compensated with the fins' motion. This is obviously possible but limits the possible freedom of movement that the robot could have, which was the reason why it was avoided.
The chassis modification was therefore chosen. The servomotors needed to be raised to the new midline, which is 10 mm higher than before. To do so, the front motors were raised by using two little aluminum plates, while the back ones were raised by bending the chassis. Doing so, the new position of the servos, and, consequently, of the center of rotation of the fins, was again perfectly at half of the height, avoiding any rotational moment generation due to some asymmetry.



(a) Aluminium plate used to raise the front motors

(b) Rear part of the chassis that was bent to be no longer horizontal

Figure 3.4: Changes to the motor positions to match the new design

As said before, using the formula for the NACA symmetric airfoil profile with a higher thickness (30% of the chord length instead of 20%) was enough to create the new exterior of the robot. Then, instead of only two parts, it was better to do it in three. This was done because the old one had to be reinforced with some brass wires on the width to avoid excessive bending. The three parts are then joined together through the same guides used in the old configuration and then screwed onto the chassis.

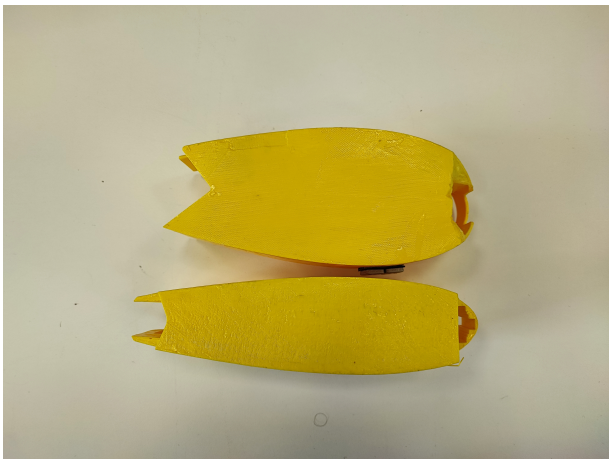


Figure 3.5: The new and bigger section (top) near the old and thinner one (bottom)

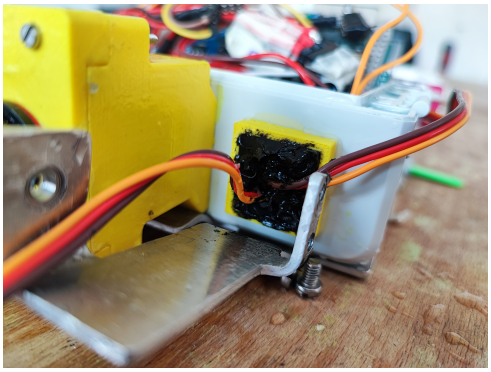
3.3. Waterproofing

The waterproofing was one of the most important parts of this work and also the most time-consuming because the trials done to reach the final configuration were several and all the products used to close the entrances of the core needed time to solidify. In the end, the waterproofing was successful, completely avoiding water leaking.

3.3.1. Core waterproofing

To completely avoid the entrance of water inside the core, two products were used. The first one is the "Kanglibang glue", a glue that kept its resistance also underwater and was used also to attach the extensions to the central box; and the second is the "Ray Tech magic gel", a dielectric gel that is made to waterproof electronic components. Also, silicone was tried but was unnecessary, since glue and gel were enough. More than one try had to be done to completely seal the box, hereafter are reported the main two; the first is the one used in the first phases of testing since it seemed to be working at first, then the last one that solved every problem, and it is the definitive one.
The first technique used was the following: the glue was used to seal the hole on the top, around the LEDs, and on the back, around the motor cables. While the dielectric gel was poured inside both the front and back extensions to completely fill them. Moreover, these extensions, made of 3D-printed plastic, were first treated with acetone to smooth out the surfaces and improve their intrinsic water resistance. To have some additional protection around the cables of the motors, a little plastic part that served as a container for the dielectric gel was placed; so to be able to pour a small quantity of it also there.

(a) Detail of the top of the box



(b) Plastic component to contain gel or silicone

Figure 3.6: Details of the sealing done

As said, this technique was successful at first, but after some usage, it was compromised and water leaked in. In addition, some holes on the bottom of the box were found, created by the screws that hold the Arduino board in place inside the box.
To completely solve this problem, the two holes on the bottom were sealed with the Kanglibang glue, while the dielectric gel was poured inside the box to fill half of it so to have the Arduino board and the batteries totally submerged, and every hole was closed by this gel, exception made for the one on the top.

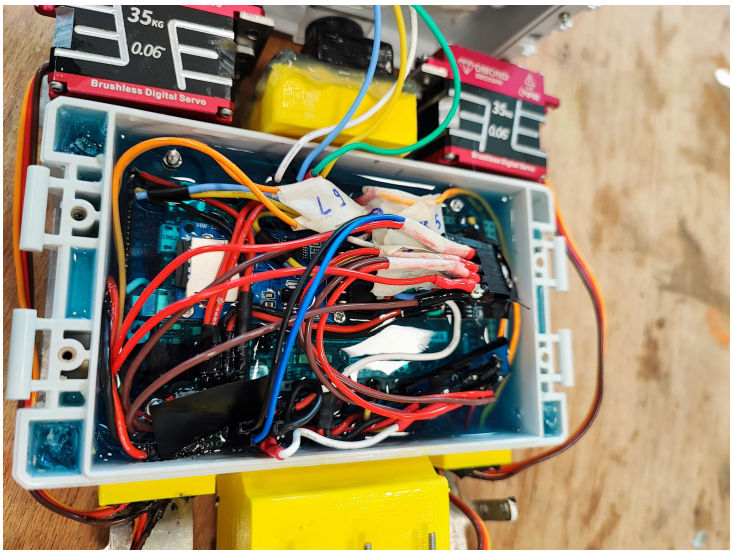


Figure 3.7: Central core filled with gel

This second method solved every problem, and the water never leaked again inside the core, leaving Arduino and every other component safe.

3.3.2. Servomotor waterproofing

After the first testing phase, all the original servos that were considered to be waterproof broke up. This happened because some water went inside of them, precisely inside the potentiometer which allowed them to understand in which position they were, causing a lot of damage and in some cases also short circuits. Therefore, these motors were not suited for this application. After some research and tests, the motors that were ultimately chosen are the Power-HD WH-40KG.

(a) TD-8320mg (b) POWER HD WH-40KG

Figure 3.8: New (right) and old (left) motors

The specs of these new servos are reported inside Table 3.2.

Working tension	6.0-8.4V
Working speed at 6V	0.20s/60°
Working speed at 7.4V	0.18s/60°
Working speed at 8.4V	0.17s/60°
Stall torque at 6V	2.84 $N{\cdot}m$
Stall torque at 7.4V	3.43 $N{\cdot}m$
Stall torque at 8.4V	3.92 $N{\cdot}m$
Dimensions	40.7·20.5·38.5mm
Weight	82g

Table 3.2: Technical data of the Power-HD WH-40

These servos were chosen because easy to waterproof, which means it is simple to pour

the dielectric gel inside of them. Even though it is said that they are waterproof, they are not made to be fully submerged in the water, therefore, the electronics have been covered with the same dielectric gel that was used in Section 3.3.1 to waterproof the core. This solution was considered the best one.
Between the original and the final motors, an intermediate model was tried. They were the DMOND DB3584V, a servomotor that exerted at maximum voltage (8.4V) 3.42Nm of torque. This model was tried because of their IP68 waterproof capability. In reality, just after a couple of immersions, they broke up due to some water leaking into the potentiometer. This component is used by the motor controller to understand the position of the shaft. The water that leaked inside made the controller unstable and, therefore, started to vibrate around the given position. This was a really important problem since the initial vibration was destined to worsen. Before abandoning these actuators, a waterproofing try was done, as can be seen in Fig.3.9. Even with the gel on the board, it was still not enough to keep the water outside. Since the problem could not be solved, this model was left in favor of the final one.
The waterproofing technique shown in Fig.3.9 is also valid for the servos that were used at last, because it was done by pouring some dielectric gel onto the control board on both. During the characterization phase of testing (Section 6.2), the POWER-HD motors were actually tried outside the laboratory. After these tests, one motor broke up, but an analysis done on all of them showed that this failure was due to a defective waterproofing of that particular servo, since the others had not a single drop of water inside.

Figure 3.9: Waterproofing of the servomotors with the dielectric gel

3.4. Balancing

Before testing, the robot had to be hydrostatically balanced. This balancing is of fundamental importance since it allows the Manta Ray to float in water without emerging or sinking. Due to the nature of this robot, neutral buoyancy is necessary for the correct operational condition. To balance the robot, some iron ballasts were placed onto the exterior cover.
Many different balancing configurations were done before having the final one, due to the continuously changing conditions of the robot. At first, it was lighter, therefore more ballast was needed, while at the end, due to the presence of the dielectric gel inside the core, it became heavier of around $100g$, therefore a lower amount of weights were used.
In addition, some 3D printed components were introduced around the core (Fig.3.10a), printed with low infill (10%), working as an air pocket. This was done because while balancing, it is always possible to add some weight, but it is not that easy to remove it.
To know where to place the ballast, a first immersion without any weight applied is performed. From there, it is possible to understand where the robot is lighter or heavier and act consequently. The rear fins were at first 3D printed with a 10% filling, therefore with a lot of air inside; this caused a very fast emersion of the robot and a shift of the center of gravity towards the front. To avoid excessive use of ballast, these caudal fins were printed again but as a solid object (100% filling), which made them weight $83g$ more and removed the air. By doing this, the robot's weight was well distributed and emerged remaining almost horizontal.
To finish the balancing, a total of four masses were applied on the bottom of the fish (Fig.3.10b), for a total of 20g of additional weight. The final configuration weighted 1.86kg.

(a) Core "floaties" (b) Ballast on the external structure

Figure 3.10: Balancing of the robot

4 | Orientation estimation

To design the orientation and position measurement in real-time, it was decided first to create it by post-processing the data obtained in the tests of Section 6.1 and comparing the results with the videos. Then, after having the method to calculate the orientation and position, it was just a matter of introducing it into the code so that also the controller could be designed. The most important controller to be implemented, due to the problem seen in Section 6.2, is an orientation controller. This is very much needed due to the difficulties of the robot to keep moving in a straight line. Then another type of control that could be implemented is on acceleration, velocity, or position, but for that one, many problems occurred.
First of all, the objective was to use the data to obtain the orientation and position over time of the robot. The procedure to obtain those values was:

- Obtain the initial Euler angles using the initial acceleration and the reference one (gravitational);
- Use those angles to obtain the orientation in the x-y-z space rotating the reference vector [0;1;0];
- Start an iteration procedure integrating the gyroscope measurements to obtain the angles at every time instant;
- Use the angles to rotate the acceleration measurement of the accelerometer to obtain it in the global reference frame;
- Remove the gravitational acceleration and integrate two times to obtain the velocity and position.

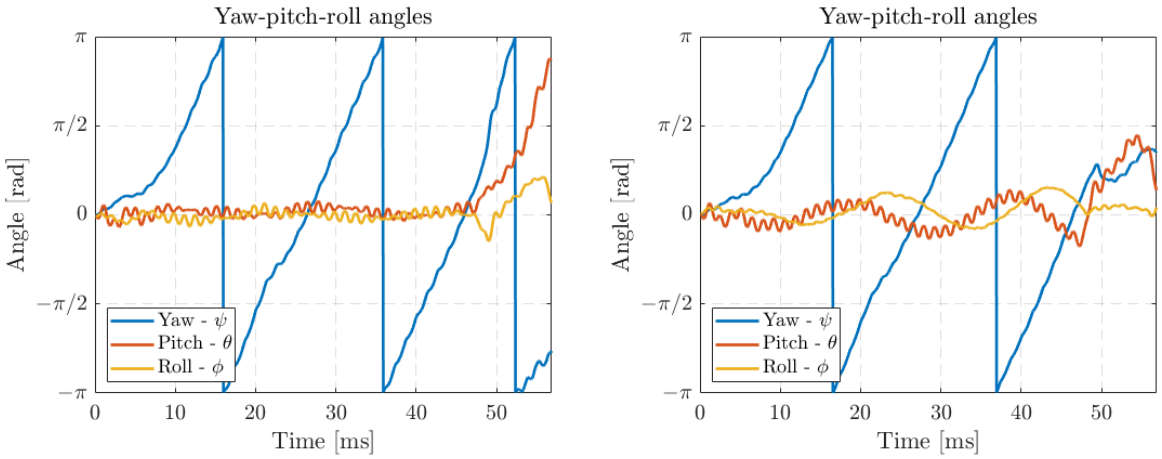
This procedure is very linear and easy, but sadly unfeasible due to the inertial navigation systems (INS) intrinsic problem. This is generated due to the sensors error that sums over time when integrating. In particular, acceleration is the worse one regarding this aspect due to the double integration.
This is a common problem while working with INS; therefore, some solutions were already available. The main ones are heavy filtering of the data, less accurate, or using the sensor

fusion, which is definitely more accurate. With heavy filtering, it is indicated the procedure where both highpass and lowpass filters are introduced on the raw data, substantially modifying them. Both were tried, but only the sensor fusion provided acceptable results.

Sensor fusion consists in using more than one sensor data, preferably of different types, to obtain a better estimation of the quantity of interest. This fusion is done through the use of a Kalman filter, either normal or extended depending on the application, or a similar one, like the Mahony. Through the use of an accelerometer, a gyroscope, a magnetometer, and a GPS it is possible to have a very good estimation of the time history of position and orientation of the object in question. Unfortunately, the manta ray robot only has an accelerometer and a gyroscope. The magnetometer is present but is very inaccurate, due to the presence of both hard and soft iron effects that deviated the values from the real ones. The first one is caused by the presence of other magnetic fields other than the Earth one, in this particular case, the current generated one. The second type of distortion is created by the presence of other metals in proximity of the magnetometer. The values that a magnetometer should read belong to a sphere centered in the origin; the hard iron effects rigidly shift this sphere from the origin, while the soft iron one modifies its shape to an ellipsoid. These effects can be calibrated, but even if a calibration was done, when the motors are on, the current that is absorbed by them generates an intense magnetic field that makes the readings of the magnetometer unreliable due to a very big variance. Performing an extended Kalman filter (EKF) with just two sensors to obtain acceleration, velocity, and position is not better than heavy filtering the data and then integrating it. Therefore, a precise position estimation was not possible; on the other hand, the orientation estimation could be performed.

The goodness of the results obtained with the heavy filtering method was estimated thanks to the MATLAB function "imufilter" [18]. This function allows to fuse the readings of the accelerometer and gyroscope through a Kalman filter to obtain the orientation. The results obtained with this procedure were very accurate since they were cross-checked with the videos of the experiments. By looking at those recordings it was possible to hypothesize the values of the angles and then see if the MATLAB plots were similar. In Fig.4.1, the comparison between the angles obtained from the rotation matrices created by integrating the rotational speed and the one created using "imufilter" can be seen. During this test, the robot performed two and a half circles around z while keeping the altitude almost constant and without having a lot of roll/pitch effect. Therefore, the expectation was to have a final yaw (rotation around z, also called ψ) of about 5π and

almost constant roll (rotation around y, also called θ) and pitch (rotation around x, also called ϕ). In the last ten seconds, there are a lot of variations that happen suddenly, that is because the manta ray was attached to a fishing wire that allowed to always have the possibility to bring it back if something happens or if it is going too far; this means that the final part is not what was measured while the robot was moving freely but while it was recovered. While recovering it, in this case, it was almost upside down, so the pitch should increase by almost π in the last ten seconds.



(a) Kalman filter Euler angles

(b) Heavy filtered data Euler angles

Figure 4.1: Methods comparison

It is very clear from Fig.4.1 that the trend obtained with the Kalman filter resembles perfectly the behavior described above, since the value of the yaw increases of 5π, which are two and a half rotations, the pitch is stable at 0 and then increases in the last ten seconds to almost π, while the roll stays almost constant at zero; this is exactly what happened and it is proved by the recordings. On the other hand, the one obtained from the integration process is clearly different; first of all, the yaw increases by only 4π, therefore half rotation was missed in the process, then the roll and pitch values have a low-frequency oscillation that goes from -0.5 to 0.5 radians, which in degrees are $\pm 28.65°$, that is obviously too much because from the videos the pitch and roll could be at most of 10/15 degrees and just for little intervals of time. An example of what has just been said is shown in Fig.4.2.

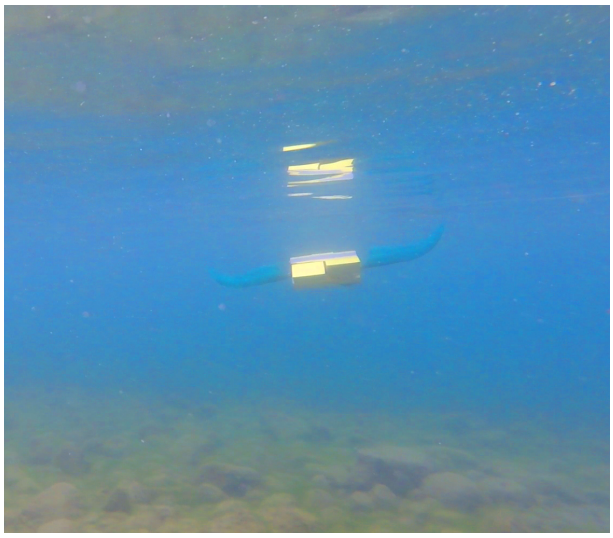


Figure 4.2: Maximum roll filmed during the motion

For the reasons that were just highlighted, if a real-time orientation controller has to be introduced, it must be done with a sensor fusion approach through a Kalman filter, otherwise, there will always be an error that grows as time goes on.
Since the Kalman filtering of data gave acceptable results, the "imufilter" function was introduced also on the Arduino board so to obtain the Euler angles in real time for the control system. The main difference between Arduino and MATLAB functions is that the first one uses a modified Mahony filter instead of an indirect Kalman filter, but the results are the same.
A problem that can occur is that for some angular values the multiplicity of the Euler angles can create big discontinuities in the yaw, pitch, and roll values. In particular, this happens for values higher/lower than $\pm 90°$. This problem can be solved by using quaternions instead of Euler angles, but since the controller that will be introduced has to maintain the orientation, angles that large should never be reached. For this reason, Euler angles were considered an acceptable solution.

5 | Code development

In this Chapter, the upgrades and changes made to the code are highlighted. In Section 5.1, the existing code is explained; in Section 5.2, the upgrades done to the functionality already present are explained in detail, in Section 5.3 is explained how the orientation measurement was introduced and, lastly, in Section 5.4 the orientation control was developed.
The control and the measurement were done only on the orientation and not also in position because the second one, at the actual state of the robot, cannot be measured, as it will be explained in Chapter 4.

5.1. Previous code

The already present code allowed to generate a sinusoidal motion on each motor and to connect to the robot via Wi-Fi, introducing a way to change the motion parameters. In addition, it also saved the data from the MPU on the SD card, which could be afterward downloaded on the PC, and it had the possibility to check those files using Wi-Fi without physically removing the SD card from the robot.
The servomotors of the pectoral fins were moved using the "Servo" library of Arduino, using the "writeMicroseconds" function. The value that the function provided to the motors was calculated using a sine wave with amplitude and frequency typically at 20° and 1Hz, but, as said before, they can be changed. The maximum angle that the old structure allowed was ±30°, while with the new one, the limit was increased to ±45°.
The rear fins instead have a different kind of motion since there is no sinusoidal motion for them, but just a reference value. To impose this value to the motors, both the "write" and "writeMicroseconds" functions could be used. The difference between those is that the second one is more precise since instead of 90, it has 1000 numbers for the same 90° range. The limit values with the old structure were +30° to -70°, the new one made this range symmetric around 0, and the limit values became ±45°.

5.2. Code upgrades

The upgrades that had to be performed were the introduction of new types of motions and a new way to name the files. Another problem the old code had was that the file printing via Wi-Fi was not working. Since it was not of primary importance, it was decided to avoid using it. Moreover, the box should always be opened after the tests to check if some water leaked in; since it has to be opened anyway, the Wi-Fi transmission of the data is not helpful in this phase.

The first and more important change performed on the code was the possibility of having different types of motions. This change was already introduced for the preliminary test phase in order to try it before the characterization tests, which were definitely more important.
These motions, explained in Section 6.2, were generated at first using different programs, which were uploaded right before the test using the cable connection. This connection had a problem: with the new design of the robot, it is impossible to connect it without removing the right caudal fin. Therefore, it can happen that the fin is mounted in the wrong way or maybe a little bit shifted every time, introducing errors. Another problem seen on that first test day was that the pectoral fins were not in a perfectly horizontal position, so a way to change the starting angles of all motors had to be introduced.
Firstly, a solution to the continuous need to upload a new program was found by creating a unique code containing every kind of motion and making it changeable via Wi-Fi. All the different motions were included inside a "switch case" function, then, the possibility of modifying the starting angle of every motor, in addition, to two different parameters for the amplitudes of the right ad left pectoral fins were introduced, which could be changed via Wi-Fi.

Another improvement done to the code is on the way the files are saved on the SD card. Before, every file was saved as "RUNXXX", from number 001 to 999. The problem with this kind of naming is that the algorithm used to choose the number was pretty inefficient. Its procedure is:

- The number 1 is assigned to the filename
- A check in the memory card is done to see if the name is already present
- If it is, increase the number by one and check again, otherwise end the procedure and start the motion

For small numbers, it was still pretty fast, but as soon as the number reached 2/3 hundred, the motion start took around 15 to 20 seconds. In order to solve this problem and make more recognizable the kind of motion in action when the file was saved, the file name containing the data from the MPU changed with the motion selected so to be self-explanatory. The naming inside Arduino is of the 8.3 type, which means an eight-character name with a three-character extension, in this case *.txt*. The names were all set to be like "XXXXXX00"; therefore, six letters and two numbers were assigned in the same way as in the original code. Even if the number assignment is not very efficient, the maximum number is now 99 and not 999. Moreover, it is very rare to do more than 10/15 tests with the same type of motion; therefore, it's hard to reach a number that generates a substantial slowdown in the starting phase of the robot.

5.3. Real-time orientation measurement

Having finished the problem-solving part of the code, some additional functionalities were introduced; the first one is about the orientation measurement. As said in Chapter 4, sensor fusion is of fundamental importance to obtain the orientation of the robot in every time instant. The objective of this procedure is to obtain the Euler angles of the robot. The three angles, yaw, pitch, and roll, are also called ψ, θ, and ϕ; these are respectively the rotation around z, around x, and around y. The reference system is of the east-north-up (ENU) type; therefore, the x-axis is positive to the right of the robot, the y-axis is positive forward, and the z-axis is positive upwards. All rotations are positive if anticlockwise.
To obtain the angles, the library "imuFilter" by RCmags [19] was used. The introduction of this tool allowed to fuse the sensor readings coming from both accelerometer and gyroscope so to correct the measurements and obtain the Euler angles at every time instant, which is of fundamental importance when a controller is added. This library uses a gyroscope model where the measured angular velocity ω is composed of three elements, which are the latent ideal angular velocity $\widehat{\omega}$, the gyro bias which changes with time and other factors b_g and the white gaussian noise n_g.

$$\omega = \widehat{\omega} + b_g + n_g \tag{5.1}$$

The accelerometer model is similar, the measured acceleration a is composed of three elements also here, but it is a little bit more complex.

$$a = R^T \cdot (\widehat{a} - g) + b_a + n_a \tag{5.2}$$

In that equation, R^T is the matrix that tells the orientation of the sensor with respect to the world frame, $\widehat{a}$ is the ideal angular velocity, and g is the gravitational acceleration in world's coordinates. Also, in this model, the bias b_a and the white noise n_a are present. Due to the presence of bias and white noise, sensor fusion is crucial to reduce measurement errors. The Mahony filter used in this library is done using a P controller and an integrating process. α is the angle that becomes the setpoint, while the rotation rate ω is considered as a disturbance.

$$\theta = K_p \frac{1}{s}(a - \theta) + \frac{1}{s}\omega \tag{5.3}$$

Where $a-\theta$ is the error input for the proportional controller. After having rearranged equation 5.3 and then separating it in error $\epsilon = a-\theta$ and angle $\theta = \frac{1}{s}(\mathrm{K}_p\epsilon+\omega)$, the result is

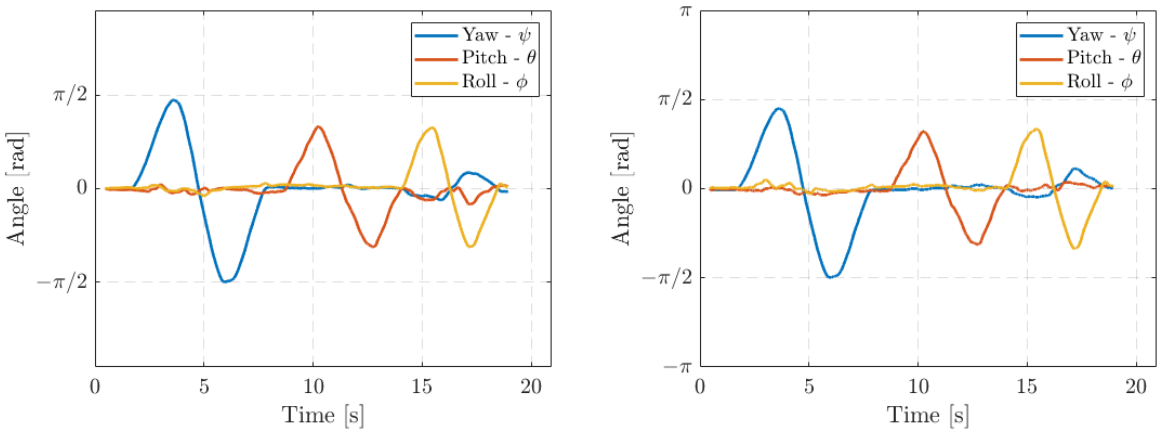
$$\epsilon_k = a_k - \theta_{k-1} \tag{5.4}$$

$$\theta_k = \theta_{k-1} + (K_p\epsilon_k + \omega_k)\Delta t \tag{5.5}$$

Therefore, the complete update law is in equation 5.6.

$$\theta_k = \alpha\theta_{k-1} + (1 - \alpha)a_k + \omega_k\Delta t \tag{5.6}$$

With α=1-$\mathrm{K}_p\Delta$t.
Moreover, the gain of the proportional controller, called "fusion gain", can be changed between 0 and 1. To set this gain, a test run where the yaw, pitch, and roll angles were moved by hand one at a time was performed for different gain values, which were 0.25, 0.5, 0.75, and 1. To judge which was the best, the results obtained from the tests were compared with the ones coming from the MATLAB function "imufilter". This function through the use of a Kalman filter does the same orientation measurement of the Arduino library described above. The comparison was done by measuring the maximum and mean differences between the two methods. The gain equal to 1 was considered the best and the results are reported in Fig.5.1.



(a) Euler angles from MATLAB imufilter function

(b) Euler angles from Mahony filter

Figure 5.1: Comparison between the real-time results and the post-processed one

Apart from some minor differences, the trend of the angles and their magnitude are very similar. Maximum and mean differences between the real-time and post-processed data are reported in Table 5.1.

	Max difference	Mean difference
Yaw	1.10°	0.19°
Pitch	1.63°	0.26°
Roll	1.33°	0.20°

Table 5.1: Differences between real-time and post-processed Euler angles

It is possible to say that both mean and maximum differences are acceptable in magnitude, therefore, the method is valid and can be used for the control system.

5.4. Orientation control

The last upgrade done to the code was the introduction of a control system on the robot orientation. The state of the art for what concerns bioinspired AUV control has been explained by He et al. [13], highlighting the most used techniques. From their analysis, they concluded that since underwater robot systems are highly non-linear, model development is complex and time-consuming, therefore, some control strategies that are not dependent on a model are used, such as fuzzy control. The main problem of this method is the dependency on the human experience, which modifies the effectiveness of

the control. Other kinds of controllers used are disturbance suppression and sliding mode control, which have complex algorithms and depend on model accuracy. The controller He et al. [13] developed had the objective of being functional with low reliance on the human factor and to be of simple implementation. To obtain this, the depth control was based on the S-plane control method, while the heading control design was created by combining fuzzy control and the desired heading transition value function.

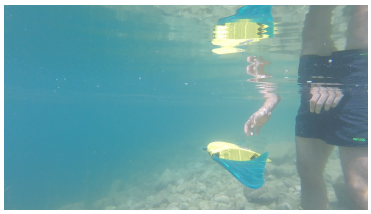
The controller just highlighted is a very advanced one, since the robot of this thesis had no control, before creating complex control strategies, it is better to start with a simpler one, such as the PID, in particular, only the proportional action was used. The control actions introduced were designed after the experimental results from Section 6.2, where the system was characterized. The reason behind this choice is due to the complexity of the system. In fact, this robot has flexible fins attached to the body only through the servomotor. Therefore, all the fin deformation is passive and depends on the fluid in which the robot is. To obtain a precise model of this robot both finite element analysis (FEA) and computational fluid dynamics (CFD) simulations must be combined, which would be demanding in both computational power and time.
More than only one motion was possible, depending on which parameter was modified by the control action. The tests done, which correspond to the motion created in the code, are reported in Section 6.3. The control action on the pitch was done by acting only on the angle of the rear fins, while for what concerns the yaw correction, the proportional action was different in each motion; it could act on the angle of the rear fins just as the pitch controller or on some parameter of the front fins, either the amplitude or the mean angle. The amplitude creates a shift in the fins of double the value of the control action calculated, by reducing the amplitude of the "internal" fin and increasing the one of the "external" fin of the same amount. Internal and external are considered with respect to the center of the curve that has to be performed. For example, if the robot has to turn right, the left fin is external while the right one is internal. The other kind of turn performed through the front fins is by moving the mean angle of the sine wave. This shift introduces some roll in the manta ray motion allowing the robot to perform the turn.
To allow this control action to run at 100Hz, a system interrupt was introduced. This was of fundamental importance to have a fast controller since the whole code was running at around 25 to 30Hz, which is a bit slow.
All the gains and the parameters on which the control acted were made changeable through Wi-Fi so that the tuning could be performed easily. The results of these controlled motions are reported in Section 6.3.

6 | Testing

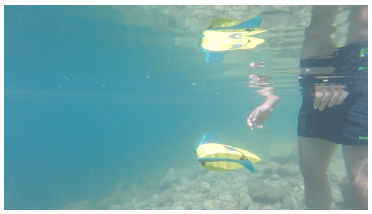
The testing phase could not be performed in the laboratory due to the lack of a swimming pool or of a big enough water tank. Therefore, to be tested in open water, the robot was brought to Lake Como.
The tests were performed in three main phases: the preliminary phase, which consisted of a first approach to the water motion; the characterization phase, where the objective was to understand the effects of each fin movement and start collecting data to be analyzed; and the last one, where a simple PID to control the robot's orientation was tested.

6.1. Preliminary test phase

This first phase of testing had as its main objective the understanding of the swimming capability of the robot and see if the waterproofing also resisted during the motion. To accomplish this objective, the robot ran for about half an hour, not continuously, with some different motions to see the behavior of the fins in different conditions. Also, the parameters of the motion were changed, modifying the frequency and the amplitude of the sine wave of the pectoral fins.

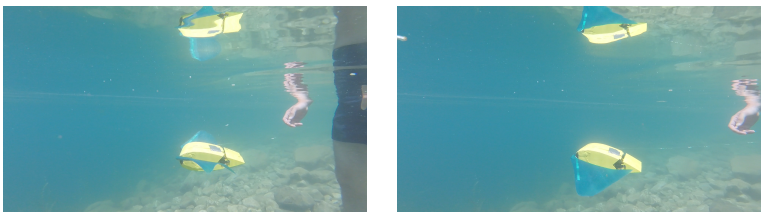


(a)



(b)

(c)



(d) (e)

Figure 6.1: Preliminary motion test

In conclusion, the first testing phase was promising since the robot swam well, and the fin behavior was good, providing a good thrust force. Even if this was true, some problems were found.
The first problem was the tendency of the robot to steer right when a symmetric motion was applied to the pectoral fins and with the caudal ones horizontal; it has, therefore, some asymmetry in the robot motion or in the external structure.
The second problem was that the different motions were placed on different codes; therefore, to change it, the robot had to be connected to the PC. The connector on which the cable is attached is behind the right rear fin, and to reach it, that fin must be removed. Clearly, this could generate errors when mounting back on the fin; for example, some misalignments may be introduced. The solution for this can be found in Chapter 5, where every type of motion created was unified in a single code.
The third and last problem, which was actually the main one, regards the waterproof capacity of the motors (Section 3.3.2). This problem was solved as said in Section 3.3.2, where the motors were substituted and waterproofed using a dielectric gel.
The last two were solved before the characterization phase.

6.2. Characterization test phase

Having solved the problems found in Section 6.1, it was time for some proper testing, data collection, and analysis.
Many more different motions and combinations were tested this time, with both pectoral and caudal fins. Then, every test performed was filmed, and all the data coming from the IMU, which were accelerations on the three axes, angular velocities around the three axes, and temperature inside the box, were saved on the SD card.
The tests done were, at first, an attempt to find the correct fin angles for the caudal ones and different motion amplitude of the pectoral ones to make the robot proceed in a straight manner without the presence of the control to compensate for eventual asymmetry. It

seemed that by giving a small angle to the rear fins and a little amplitude difference to the front ones, the behavior improved, but ultimately, it was clear that the factors that influenced its trajectory were also on the "initial conditions" of the motion and the environment disturbances. The first is created by the user upon release, which could introduce little deviations from a straight motion that the robot would not recover, while the second is due to the currents and waves present while testing since they happened in a lake and not in a pool. For these reasons, the correction through angles was left behind, and it was preferred to solve this problem through a control system.
Then, other motions were tested, starting with different combinations with the front fins and then different positions of the rear fins. The motions performed with the pectoral fins are the following:

- Symmetric motion of the fins but with different amplitudes and frequencies;
- Motion with different fins amplitude to test the turning capability;
- Motion with fins having opposite phase;
- Motion with a different mean value of the sine wave.

During all those motions, the rear fins were left still in a horizontal position. Then, the characterization moved on to the rear fins:

- Symmetric shift of the fins upward or downward;
- Asymmetric fins position.

In these motions, the pectoral fins moved with 20° amplitude and 1Hz frequency.

6.2.1. Symmetric pectoral fins motion

The first motion tested is the rectilinear one, where different parameters of the motion were tested. The amplitudes values were 20, 30, and 40°, while the frequency used were 0.5, 1, and 1.5Hz. Most of the possible combinations were tested, and it is possible to say that the speed and acceleration of the robot increased when increasing either amplitude or frequency. The fastest configuration is, therefore, 45° of amplitude and 1.5Hz of frequency to the front fins, but sadly it is not possible to quantify the maximum speed value from the analytic data, as explained in Chapter 4 for the reasons that were explained. Moreover, the test environment was not a controlled one; therefore, even external sensors could not be mounted. To obtain an estimation of the speed value, knots were made on the cable that allowed to recover the robot, each at 1m distance from the other. Thanks to this and the videos that allowed to measure time, the speed estimation was obtained.

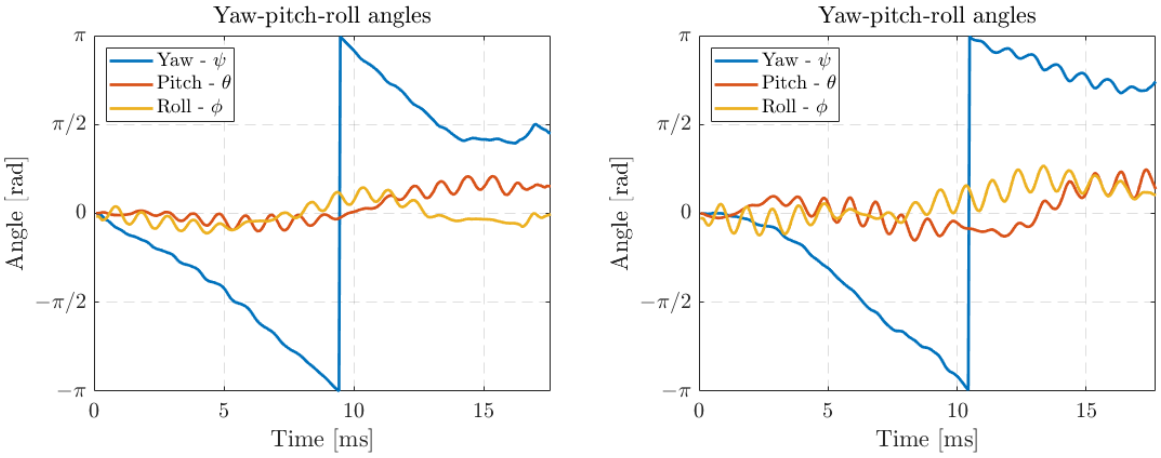
The value was between 0.7 and 1m/s, and since the robot is a little less than 0.3 meters long, it is more than twice its body length.

(a) (b) (c)

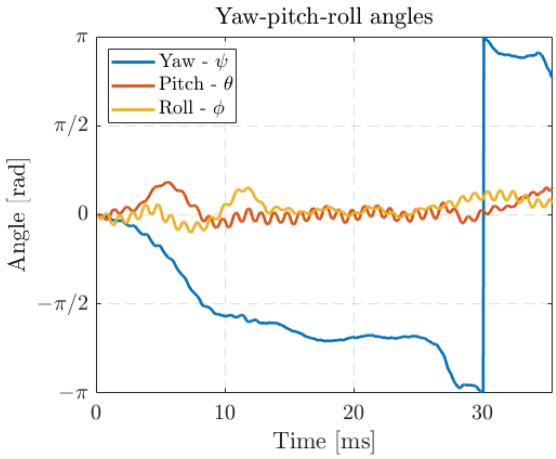
Figure 6.2: Symmetric pectoral fin motion, f=1Hz, A=20°

6.2.2. Pectoral fins turning motions

The second and third points of the list in the introduction to this section will be discussed together since the effect of those two motions is the same. Three main tests were conducted, the first with only the left fin moving, the second with both fins moving but the right one had half amplitude and the last with the counter phase fins.
By analyzing the data, as expected, the slower is the second one, while the other two are similar, but the single fins motion has the edge over the other.



(a) Right fin stopped, left fin amplitude=20° (b) Counterphase fins, amplitude=20°



(c) Right fin amplitude=10°, left fin amplitude=20°

Figure 6.3: Yaw-pitch-roll angles for the three turning methods

From Fig.6.3, it is possible to extract the rotation speeds of every motion. The one in Fig.6.3c is the slowest, with just 0.11 rad/s of rotational speed. The other two are 0.33 rad/s for Fig.6.3a and 0.3 rad/s for Fig.6.3b. These are almost equal in performance, but the main difference is that in the counter phase case has many more oscillations in both pitch and roll, making the one fin rotation look a lot more stable and, therefore, easier to fix through a control action. These oscillations of the counter phase motion can also be seen in Fig.6.4, where it is clear that during the motion, the roll is not negligible.

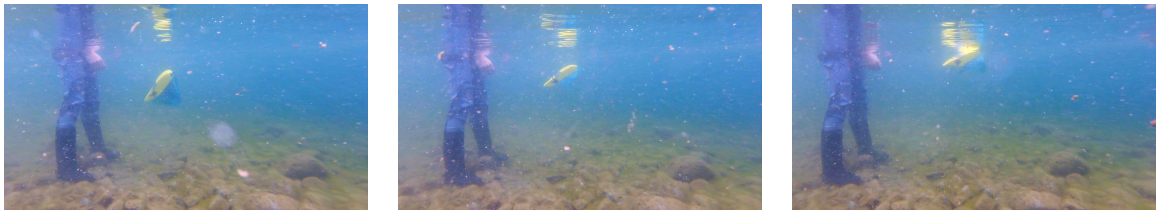
(a) (b) (c)

Figure 6.4: Counter phase motion

6.2.3. Ascending and descending motions with pectoral fins

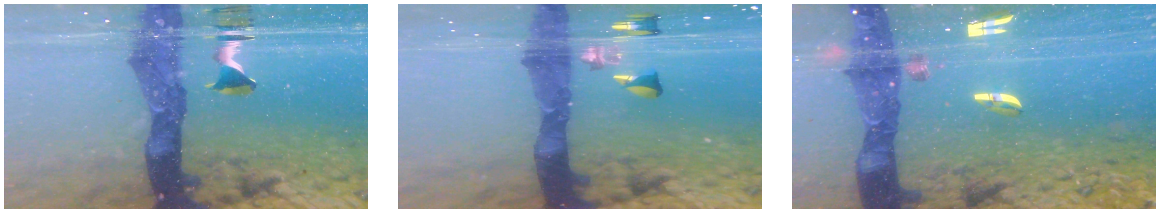
The last tests done on the front fins were with a different mean angle of the sine wave. The range of motion of the pectoral fins is ±45°, so the motion was set with an amplitude of 20° and a shift of 22.5°, half of the maximum value. The shift was applied in both positive and negative directions, and the resulting motions are the ones reported in Fig.6.5 and Fig.6.6.

(a) (b) (c)

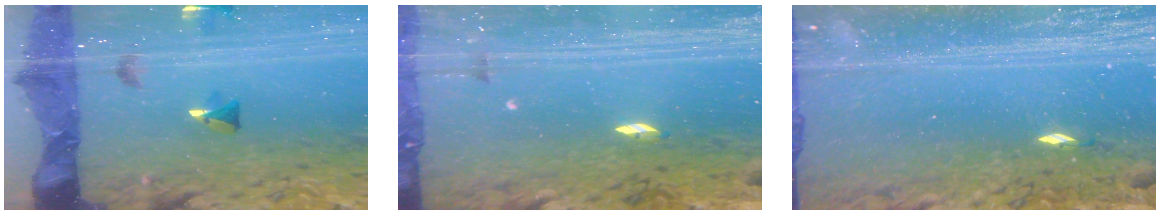


(d) (e) (f)

Figure 6.5: Motion with a shift of -22.5°



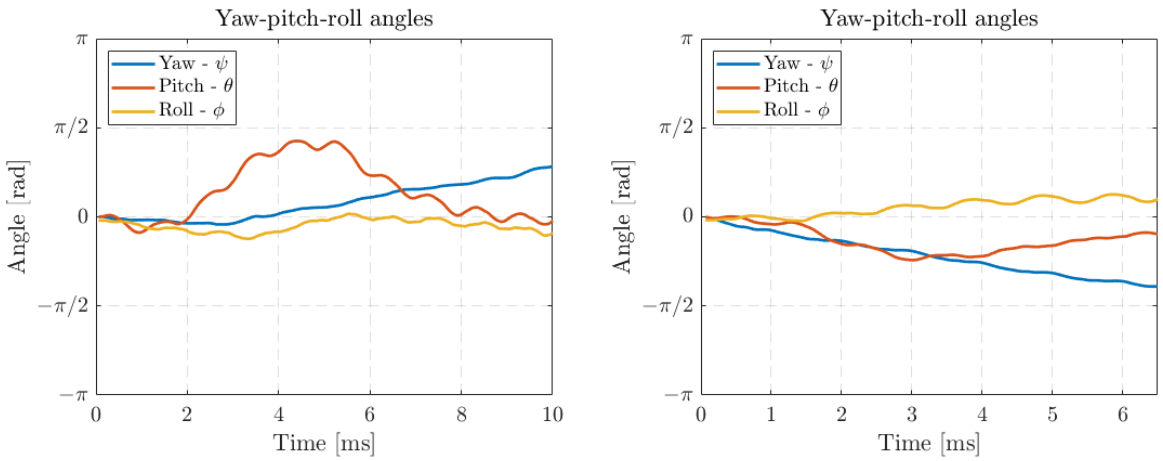
(a) (b) (c)



(d) (e) (f)

Figure 6.6: Motion with a shift of +22.5°

From the pictures, it is possible to say that a positive shift makes the manta ray descend while having a negative shift makes it ascend. Analyzing the data, it is possible to know the value of the maximum pitch angle reached during this ascension/descent.



(a) Ascending motion (b) Descending motion

Figure 6.7: Ascending and descending motion

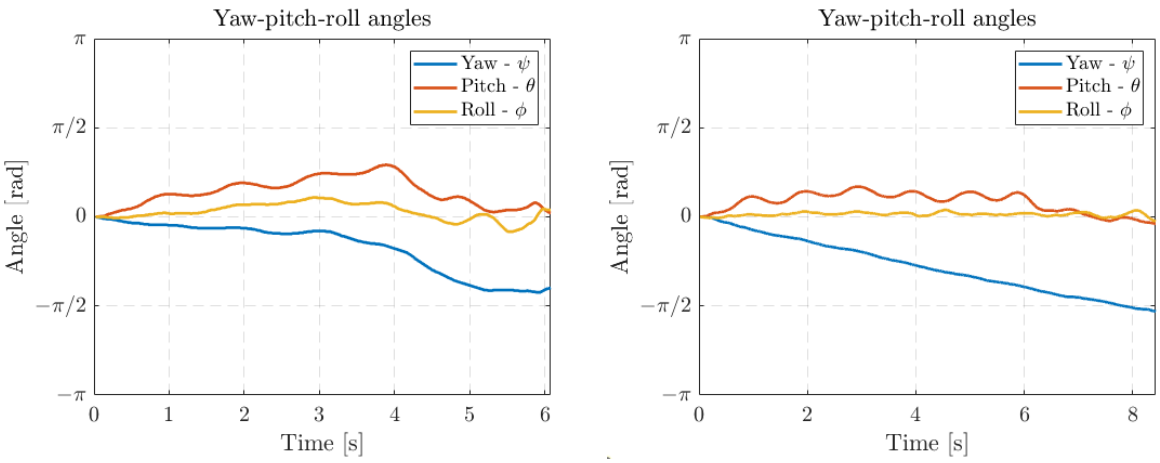
During the descending motion, the maximum pitch registered is 43.7°, while during the ascending one was 76.6°. Also, from Fig.6.5 is clear that the pitch was definitely higher than the one in Fig.6.6. This is probably due to the tendency of the robot to surface and maybe also to some initial forces or angles applied to it upon release. Even if this was the case, 45-50° of pitch angle for ascending/descending maneuvers is still a good achievement.

6.2.4. Ascending and descending motions with caudal fins

The first tests performed were done with the fins in symmetric positions, therefore both up or down. Of this kind of test, four were performed: fins all the way up, fins halfway up, fins all the way down, and fins halfway down.

(a) (b) (c)

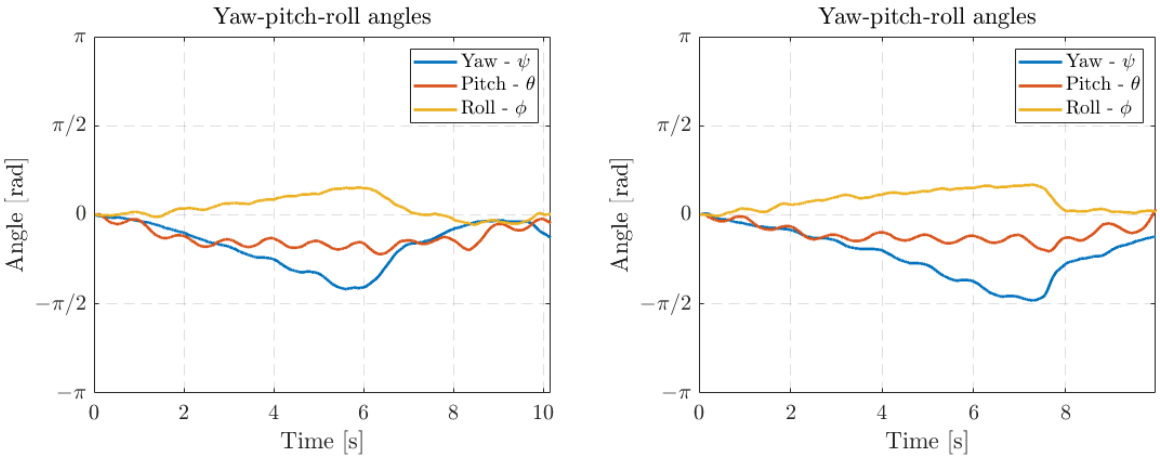
Figure 6.8: Ascending maneuver done with the rear fins at 45°



(a) Fins all the way up (b) Fins halfway up

Figure 6.9: Ascending maneuver done with the rear fins

In Fig.6.9, the rear fins were used to perform an ascending maneuver by placing them in an upward position, as can be seen from Fig.6.8. Two tests were performed, one with the fins at their limit angle (45°) and the other at half of it (22.5°). The maximum pitch obtained during these motions is 52.7° for the first motion and 30.37° for the second.



(a) Fins all the way down (b) Fins halfway down

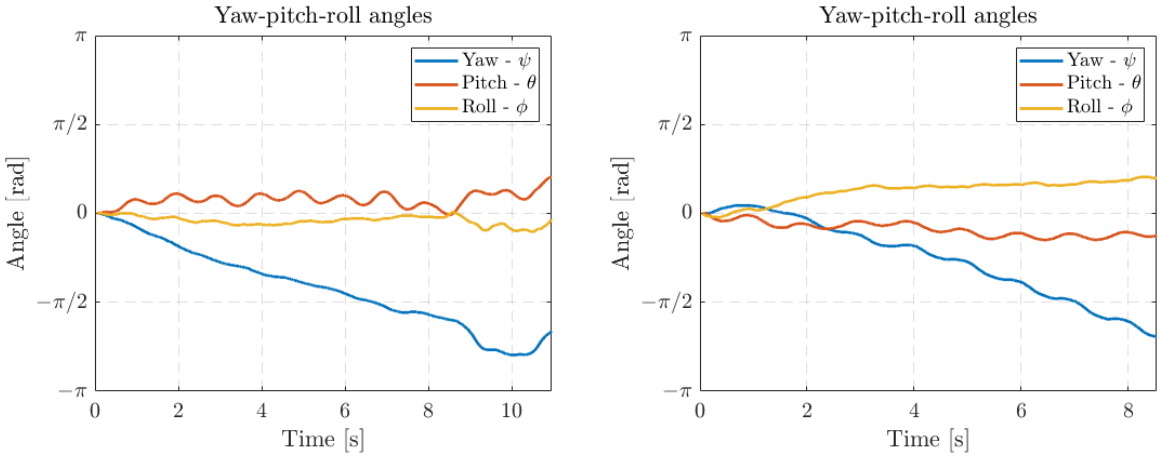
Figure 6.10: Descending maneuver done with the rear fins

Then, the same tests that were done for the ascension were also performed for the descent. In this case, the minimum pitch values were 39.53° for the maximum angle case and 36.67° for the middle angle one. These values are both similar to the ones obtained when the descent was performed using only the front fins. Having seen that also in this

case the ascend obtains a bigger angle than the descend, it is possible to say that probably is the structure of the robot that creates this difference since it floated before the balancing and, therefore, it is possible that this tendency also remained after being balanced.

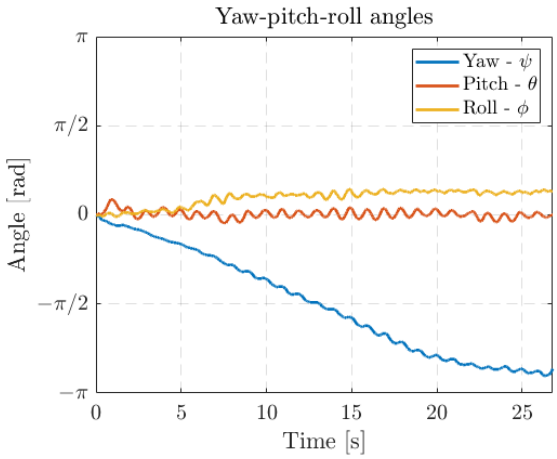
6.2.5. Turning with caudal fins

The next set of tests that were performed were with asymmetric positions of the rear fins, in particular, three main tests were done: one fin all the way up and the other in horizontal position, one fin all the way down and the other in horizontal position, one fin all the way up and the other all the way down. These three configurations all introduced a roll angle during the motion, making the robot steer. These tests were done only for one fin and not both because the system is considered to be symmetric.



(a) One fin up

(b) One fin down



(c) One fin up and one down

Figure 6.11: Turning maneuver performed with the rear fins

The main result that these tests produced was the turning speed, which was 0.24rad/s with the "one fin up" motion and 0.22rad/s for the "one fin down". These speeds are similar to the one obtained with the front fins turning, which was around 0.3rad/s. Therefore, also this method can be used, especially when also a different altitude has to be reached in the same motion. The last one instead has a much lower impact, only 0.12rad/s; therefore, this last method can be used for turns with big radii.

Having seen every possible way to maneuver the robot, some comparative graphs can be made so to have a direct comparison between all methods. In Fig.6.12, all the turning methods presented before are grouped inside a single plot. From there, it is clear that the one fin rotation, motion with only one fin and the other is stopped, has the edge over every other method, while the less performing is the opposite position of the rear fins, where the fins are one in the top position and the other in the bottom one. This last is on the same level of the half amplitude rotation, the one where both front fins are moving but one with half the amplitude of the other. Every other method sits in the middle between these two extremes. The curves have all different lengths due to the different duration of each test.

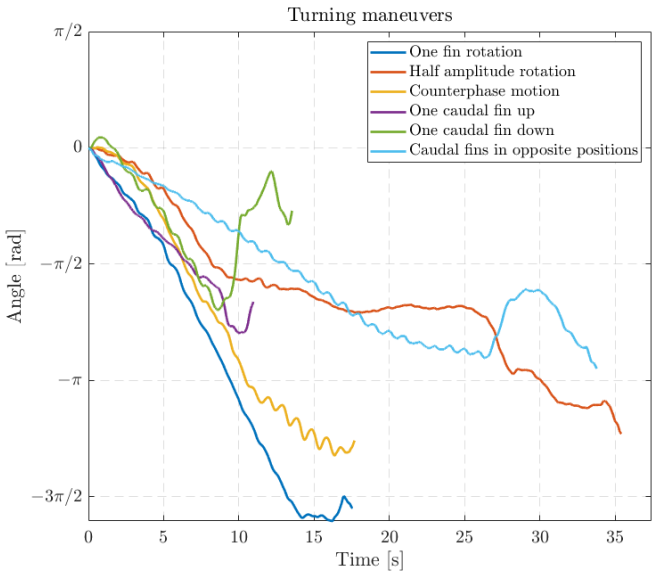
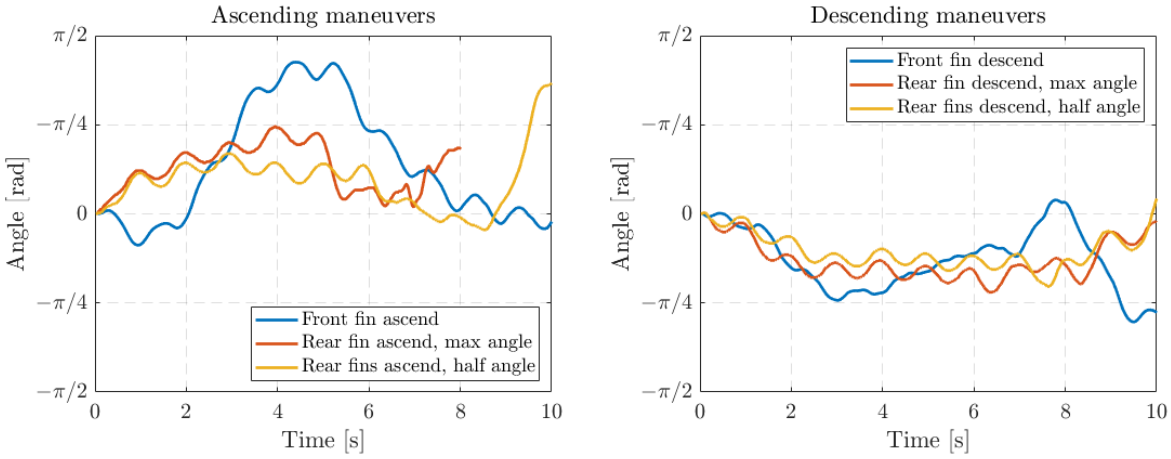


Figure 6.12: All possible turning maneuvers data

The same thing can be done also for the ascending and descending maneuvers.



(a) Ascending motions (b) Descending motions

Figure 6.13: Height varying motions

In both Fig.6.13a and Fig.6.13b, the front fin motion is better than the rear one, but performing these maneuvers with the pectoral fins reduces the thrust force that they could provide, slowing down the robot. It is, therefore, better to perform these maneuvers with the caudal fins unless a very fast variation of the height is needed; in that case, a combination of both fins can be used to achieve an even higher pitch angle.

As a conclusion for these tests, it can be said that the steering problem was not solved even by changing the angles of the back motors and setting different amplitudes to the front fins for the reasons already explained in Section 6.2.1. It is, therefore, mandatory to correct this problem through an orientation controller.
Another thing that can be seen in every graph that must be corrected through the controller is the periodic pitch oscillation. The cause of this is the pectoral fin movement that generates a periodic vertical force. This effect neglects itself after a complete flap of the fins, but to reduce the amplitude of those oscillations, a pitch control must be introduced. The reduction of these oscillations should be done for two main reasons; the first one is to have a more stable motion, and the other is to have better videos when a camera is installed. The first is self-explanatory, a more stable motion means that also maintaining a certain height or orientation is easier; the second one is instead a more advanced problem since it is not something that concerns the robot as is but is a possible future inconvenience which, if solved before it has the possibility to happen is certainly better.

6.3. Control test phase

The last tests performed were done to try an orientation PID controller. The control action used is only the proportional one. Moreover, the angles on which this controller was applied were only the yaw and the pitch, while the roll was kept free.
The choice of not introducing a control action also on the roll was taken relying on the results obtained from Section 6.2, where it was clear that the pitch had a lot of fast variations introduced by the pectoral fins flapping, the yaw had to be controlled to avoid unwanted steering, while the roll had slow variations and of low magnitude. For these reasons, the controller was simplified by not accounting for the roll.
The pitch correction was performed using only the rear fins, while the yaw could be corrected by both front and back fins. The final tests done are:

- Pitch correction changing the rear fins angle;
- Yaw correction changing the rear fins angle;
- Yaw correction changing the front fins amplitude;
- Yaw correction changing the front fins sine wave mean value;
- Using all of the effects written above all together;

The control strategy was developed based on the characterization test result. The pitch correction exploits the caudal fins to introduce a moment that is opposite to the one that is generated from the front fins, so to counter that effect and reduce the oscillations.
The theory explained above had to be validated through tests. As said before, a proportional action was introduced to reduce the pitch oscillation when moving. The controller introduced is far from state of the art presented by He et al. [13], but still had a positive effect on the system. Several gains were tried, and the best one was 800. The results for this gain are in Fig.6.14.

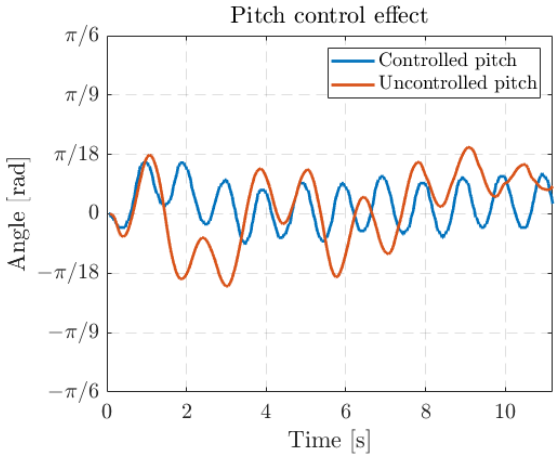


Figure 6.14: Controlled vs uncontrolled pitch during a rectilinear motion

The uncontrolled pitch has a value of ±14°, while the controlled one is ±6.8°. Therefore, even with this simple controller, the pitch oscillation has been reduced by half, and the stability of the values is increased. This is an encouraging result for when a more advanced controller will be introduced.

Then, for the yaw correction, more than one strategy was used. These could be done with an angle on the caudal fins, with a different amplitude or mean angle on the pectoral fins. In Section 5.3, the problem of the multiplicity of Euler angles was introduced. Before testing, the belief was to have small angles, and therefore this problem could be ignored for now. In reality, this was not true, also because the lake in which the tests were performed was rough. Therefore, all the yaw control actions had problems following the reference. Nevertheless, when the lake was calmer, a test was conducted, and the results were encouraging since the yaw was kept constant.

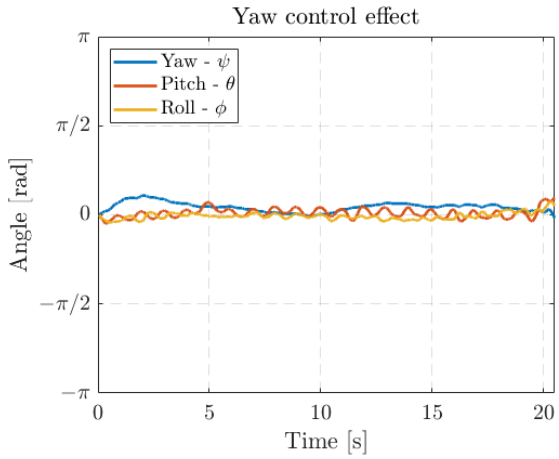


Figure 6.15: Yaw control performed using every strategy at the same time

The main problem that this controller has is that even if the orientation is kept still. In fact, during the test, the robot did not proceed in a straight line but drifted carried by the lake waves, as can be seen from Fig.6.16. Here, the robot orientation is always the same, but it is progressively closer to the camera. Therefore, the controller works, but it is incomplete since it needs a position reference to reject all the remaining disturbances.

(a) (b) (c)

Figure 6.16: Yaw control test

Having now seen all the results of the tests, it can be said that the robot responded as expected to all the characterization tests, which were crucial in the development of the control. The possibilities for further improvements in this area are several, especially when a position measurement is available. When there is the capability to insert a velocity/position controller, this robot will be able to move, perform sharp accelerations and keep a cruising speed just like a real manta ray.

7 | Conclusion and future developments

This thesis work was about the improvement and testing of a bioinspired manta ray robot, which final configuration can be seen in Fig.7.1.
The upgrades performed to the robot started from the batteries, changed with a more performing LiPo model to have a current supply that was enough for every component inside.
The waterproofing was the most demanding operation since it was also the most important. To reach this objective, the central box of the core had to be changed with a bigger one, the servomotors were substituted with an easier-to-waterproof model, and the external structure design was revised completely.
The code has been developed accordingly to the demand of the tests that were performed. A big step forward was done also in this field since now a lot more types of motion can be performed, and it is very simple to add a new one if necessary.
A PID control system was introduced on the orientation so to have the possibility to reduce the pitch oscillation generated by the flapping of the pectoral fins and to correct the orientation by keeping the initial yaw angle as a reference for the motion. Moreover, the correction of the orientation was done in three different ways to see the effect of each one individually and then merged all together. Even though this did not work in the test phase, the code can be used in future when the quaternions will substitute the Euler angles.
The data obtained from the tests were manipulated to obtain the Euler angles that were reported. In addition, the method to obtain those angles was at first done there in a post-processing phase and then introduced in real-time.

The future developments that are possible are mainly regarding the improvement of the controller. First of all, a new sensor has to be introduced to obtain also the velocity and position measurement.
With regard to the controller, the PID that was created for the pitch had a positive in-

fluence on the system. The yaw controller was instead incomplete; therefore, this could be fixed after the position measurement is available.
Before thinking of more complex controllers, Euler angles must be substituted by quaternions to have a unique value for each angular position of the robot. After that, other control strategies can be explored and developed to obtain better performances.
Through the measure of velocity and acceleration, it is possible to optimize the frequency and amplitude of the movement and also explore new fin designs that could further improve the performances.

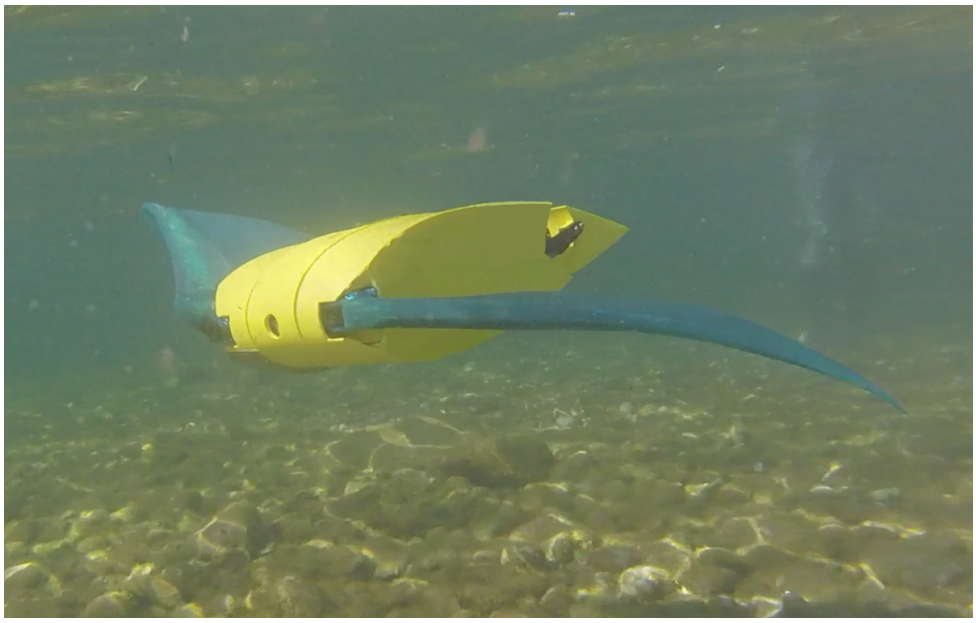


Figure 7.1: Frame of the robot while swimming

Bibliography

[1] D. Alderton and S. A. Elias. *Ecyclopedia of Geology*, volume 6. Academic Pr, 2 edition, 12 2021. ISBN 9780081029091.

[2] Y. Cai, S. Bi, and L. Zhang. Design and implication of a bionic pectoral fin imitating of a cow-nosed ray. In *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 10 2010.

[3] Y. Cai, S. Bi, and L. Zheng. Design Optimization of a Bionic Fish with Multi-Joint Fin Rays. *Advanced Robotics*, 26:177–196, 2012.

[4] Y. Cai, S. Bi, G. Li, H. P. Hildre, and H. Zhang. From natural complexity to biometric simplification: therealization of a bionic fish inspired by the cownose ray. *IEEE Robotics and Automation Magazine*, 2019.

[5] L. Chen, S. Bi, Y. Cai, Y. Cao, and G. Pan. Design and Experimental Research on a Bionic Robot Fish with Tri-Dimensional Soft Pectoral Fins Inspired by Cownose Ray. *Journal of Marine Science and Engineering*, 10(4), 4 2022.

[6] C. Chew, Q. Lim, and K. Yeo. Development of Propulsion Mechanism for Robot Manta Ray. In *2015 IEEE Conference on Robotics and Biomimetics*, Zhuhai, China, 12 2015.

[7] C. Chew, S. Arastehfar, G. Gunawan, and K. Yeo. Study of Sweep Angle Effect on Thrust Generation of Oscillatory Pectoral Fins. In *2017 IEEE/RS International Conference on Intelligent Robots and Systems (IROS)*, Vanvouver, BC, Canada, 09 2017.

[8] W. Chi and K. H. Low. Review and Fin Structure Design for Robotic Manta Ray (RoMan IV). *Journal of Robotics and Mechatronics*, 24(4):620–628, 2012.

[9] FestoAG and Go.KG. Air-ray, Festo Corporate, 2010. URL `https://www.festo.com/us/en/e/about-festo/research-and-development/bionic-learning-network/highlights-from-2006-to-2009/air-ray-id_33851/`.

[10] F. Fish, C. Schreiber, K. Moored, G. Liu, H. Dong, and H. Bart-Smith. Hydrodinamic Performance of Acquatic Flapping: Efficiency of Underwater Flight in the Manta. *Aerospace*, 3, 2016.

[11] J. Gao, S. Bi, and C. Liu. Development and Design of a Robotic Manta Ray Featuring Flexible Pectoral Fins. In *2007 IEEE Conference on Robotics and Biomimetics*, Sanya, China, 12 2007.

[12] K. Hall, P. Hundt, J. Swenson, A. Summers, and K. Crow. The Evolution of Underwater Flight: the Redistribution of Pectoral Fin Rays in Manta Rays and their Relatives (Myliobatidae). *Journal of Morphology*, 279:1155–1170, 2018.

[13] Y. He, Y. Xie, Y. Pan, Guang Cao, Q. Huang, S. Ma, D. Zhang, and Y. Cao. Depth and Heading Control of a Manta Robot Based on S-Plane Control. *Journal of Marine Science and Engineering*, 10, 11 2022.

[14] M. Ikeda, S. Hikasa, K. Watanabe, and I. Nagai. A pectoral fin analysis for diving rajiform-type fish robots by fluid dynamics. *Artif Life Robotics*, 19:136–141, 2014.

[15] G. Li, Y. Deng, O. Osen, S. Bi, and H. Zhang. A Bio-inspired Swimming Robot for Marine Aquaculture Applications: from Concept-design to Simulation. In *OCEANS 2016 - Shangai*, April 2016.

[16] K. Low, C. Zhou, G. Seet, S. Bi, and Y. Cai. Improvement and Testing of a Robotic Manta Ray (RoMan III). In *2011 IEEE International Conference on Robotics and Biomimetrics*, Phuket,Thailand, December 2011.

[17] H. Ma, Y. Cai, Y. Wang, S. Bi, and Z. Gong. A biomimetric cownose ray robot fish with oscillating and chordwise twisting flexible pectoral fins. *Industrial Robot: An International Journal*, 42:214–221, 2015.

[18] MathWorks. imufilter: Orientation from accelerometer and gyroscope readings, 2018. URL `https://it.mathworks.com/help/nav/ref/imufilter-system-object.html`.

[19] RCmags. imuFilter, 2021. URL `https://github.com/RCmags/imuFilter`.

[20] P. Riggs, A. Bowyer, and J. Vincent. Advantages of a Biomimetric Stiffness Profile in Pitching Flexible Fin Propulsion. *Journal of Bionic Engineering*, 7:113–119, 2010.

[21] L. Rosemberger. Pectoral Fin Locomotion in Batoid Fishes: Undulation versus Oscillation. *The Journal of Experimental Biology*, 204:379–394, 2001.

[22] R. Russo, S. Blemker, F. Fish, and H. Bart-Smith. Biomechanical model of ba-

toid (skate and rays) pectoral fins predicts the influence of skeletal structure on fin kinematics: implications for bio-inspired design. *Bioinspired & Biomimetrics*, 10, 2015.

[23] R. Salazar, V. Fuentes, and A. Abdelkefi. Classification of biological and bioinspired acquatic systems: A review. *Ocean Engineering*, 148:75–114, 2018.

[24] M. Sfakiotatis, D. Lane, and J. Davies. Review of Fish Swimming Modes for Acquatic Locomotion. *IEEE Journal of Oceanic Engineering*, 24:237–252, 1999.

[25] Y. Zhang, S. Wang, X. Wang, and Y. Geng. Design and Control of Bionic Manta Ray Robot With Flexible Pectoral Fin. In *14th International Conference on Control and Automation (ICCA)*, Anchorage, Alaska, USA, June 2018.

[26] C. Zhou and H.-H. Low. Better Endurance and Load Capacity: An Improved Design of Manta Ray Robot (RoMan II). *Journal of Bionic Engineering*, 7:137–144, 2010.

List of Figures

List of Tables

Acknowledgements

Con oggi, termina il mio percorso di studi durato 5 anni, nei quali ho potuto incontrare persone stupende con le quali ho condiviso momenti indimenticabili. Voglio ringraziare in primis i miei genitori, Lorena e Roberto, e mia nonna Rosanna, che mi sono sempre stati vicini, supportando nelle mie scelte e assecondando i miei capricci. Un ringraziamento speciale va a tutti i miei compagni di corso, specialmente a coloro che sono stati i miei coinquilini nell'ultimo anno, grazie a voi questi anni mi sembrano volati e mi hanno fatto maturare parecchio sotto ogni punto di vista. Voglio ringraziare anche i miei amici che sono al di fuori dall'università, grazie a loro ho potuto passare serate in leggerezza anche nei periodi peggiori.
Vi voglio bene.

Infine, un grazie è doveroso anche al professor Cinquemani e a Giovanni, dato che il periodo in cui ho lavorato a questa tesi sarà sicuramente tra i ricordi più belli dell'università.