2nd YEAR INTERNSHIP REPORT 2023/05/01 - 2023/08/31

DEVELOPMENT AND CONTROL OF A FISH-SHAPED ROBOT

FISE 2024 – ROB

Hippolyte LEROY
hippolyte.leroy@ensta-bretagne.org

Supervised by: 橋本 卓弥 (Takuya HASHIMOTO)
tak@rs.tus.ac.jp

6-chōme-3-1 Niijuku, Katsushika-Ku, Tokyo 125-0051

Acknowledgements

I am incredibly grateful to Mr. 橋本 卓弥 (Takuya Hashimoto), tutor of the internship and guarantor in Japan, for accepting me at Tokyo University of Science. During this 4-month internship he integrated me very well into the group of students even though I did not speak Japanese at first and helped me a lot in my work, always giving me relevant advice to improve the work.

Furthermore, I also want to thank the other bachelor and master students in the class who warmly welcomed me in the lab and were always available to help me solving problems, in particular Mr. Stephane Weber, with whom I had the privilege to work on the fish robot and helped me lot to understand how the le lab works. I also want to thank Mr. Baptiste Lamothe, student and friend from ENSTA-Bretagne who was doing his internship in the lab next door, 元祐 昌廣 (Masahiro Motosuke) Lab, and with whom I had the pleasure of working on a sensor developed in his lab, a waterflow sensor.

In a more general way, I would like to thank all Japanese people with whom I could sympathize because they made me discover their rich culture and helped me to adapt in a country so different from France, as an example, my teammates in Tokyo University of Science's basket-ball team.

In addition to that I want to thank Mr. Luc JAULIN and Mr. Benoît ZERR (referent professors at ENSTA-Bretagne) for this year at the Ensta-Bretagne in autonomous robotics because I gained experience in this field and more particularly in marine robotics, which I enjoyed a lot.

Finally, I would like to thank my parents who also made this whole experience possible.

Sommar

Abstract

To finish my second year at ENSTA-Bretagne in autonomous robotics, I did an internship of about 18 weeks at Tokyo University of Science(東京理科大学) in Katsushika-city campus at the mechanical engineering department and more specifically in the Hashimoto Lab. I worked on the mechanical and electronic development of a fish-shaped robot with the aim of making it autonomous. The constraints on AUVs (autonomous underwater vehicles) of this kind are well known (waterproofing, communication via Wi-Fi impossible when changing environment, corrosion of materials). They make this project even more ambitious and instructive. The main objectives were to create a robot capable of behaving like a fish, while at the same time being able to control its depth and avoid obstacles. I also had the chance to work with a sensor that is currently under development at this university, which is a fluid velocity sensor using energy dissipation. For the first two months I was working alongside another Frenchman on this project, then on my own for the last two.

NB: All source codes and CAD available at:
https://gitlab.ensta-bretagne.fr/leroyhi/sakana_roboto

Introduction

I entered Hashimoto Lab at the beginning of May 2023. I was introduced in a group of Japanese students of about a twenty people, including bachelor and master students and a Frenchman who was doing his final graduation project. I was surprised to hear that it was the beginning of the semester as it starts in April in Japan. I think that this has been an advantage for my integration into the group. To compare to Ensta-Bretagne, it's as if I was doing my internship at the rob lab but at Tokyo University of Science.

This internship was my first experience abroad in terms of work It took me a while to understand the social norms that apply in Japan, but since the atmosphere in the lab was pleasant and friendly, I was able to make the most of it.

The way of working was as follows:

Firstly, students work autonomously on their respective projects, and they can purchase barely any item on the internet for their project in agreement with the teacher. Once a week they must make an oral presentation with slides alongside Professor Hashimoto to talk about the achievements of the previous week, to list the problems and discuss of possibles solutions. And on alternated Mondays they must make an oral presentation of their project in front of the whole group to show their achievements, eventually failures, projects status and future plans.

To give a few examples of other students' projects, there were two students working on robot-fish development projects like me, one of the fish had the particularity of having the shape and behavior of a stingray. Some other students were working on robotic personal assistance systems, such as detecting when someone is choking or falling out of bed. There was also a robot capable of autonomous welding, and systems to support athletes in their performance.

During my year at ENSTA-Bretagne, I really enjoyed working on marine and underwater robotics projects because of the challenges involved (watertightness, environmental constraints, positioning). I aim to work in the marine robotics field after graduating, so this is why developing an underwater robot during this internship fits in perfectly with my professional project.

As I said, the robot I worked on aims to look like a fish, which could have multiple applications today.

Under-water Vehicles are used by scientists to explore the seabed where it would be difficult and dangerous for humans. Those vehicles can be controlled through a cable or acoustic waves, or they can be autonomous. AUVs (Autonomous Underwater Vehicles) are the type of vehicles we're going to be looking at. But since thrusters make noises and are shaped very differently from their surroundings, they can disturb their environment.

In the context of unstable fauna and flora, ecosystems will need to adapt to their unfamiliar environment. A such robot will be useful to understand complexes phenomenon happening in seas or oceans, and to explore the very deep sea, which remains very unknown. As an example, we could imagine that a fish-like robot could carry out measurements in this kind of area with the appropriate sensors, actuators and embedded electronics. A robot of this size could perform dangerous tasks everywhere in the sea, and especially without disturbing its environment due to its shape.

Another possible approach possible is the military one. A fish-bot like this could easily vanish into its environment without being detected. Thus, it could be used to spy on the enemy or to directly attack him.

Finally, we can imagine introducing this kind of robot for recreation or education in aquariums to show people rare, endangered, or extinct species that are difficult to obtain for such an establishment.

The robot in this project is 3D-printed, we had 3D printers at our disposal, which is convenient for quickly obtaining the parts of the structure we wanted. The major advantage of having a 3D-printed robot is that we can send CAD files to the other side of the world, making it easy to duplicate the robot.

Of course, a 4-month internship is short to build such a complex technical object by myself. However, I had the pleasure to work on this project with Stephane WEBER, the French I talked about previously who was studying at Sea-Tech in Toulon. He finished his final graduation project at the end of June 2023. As a result of, I was alone on the project in July and August.

The division of tasks was as follows:

Stephane designed all the structural parts, with the computer aided design (CAD) associated while I was overseeing the electronic architecture to have all the necessary sensors and actuators to control the robot. Once this was done, we were able to create the control program for the robot to carry out its mission, which was to complete a course by avoiding obstacles.

Regarding the arrangement of the internal space, the mounting, and the waterproofing of the robot, we supervised it together by discussing the benefits and drawbacks of each solution.

To summarize, the first target was to finish the building of a fish-shaped robot that could go into shallow waters, considering the waterproofing problems. And then we had to control and guide the robot, by remotely sending code that it could easily interpret.

Once the robot is ready for use, we have a pool at our disposal for testing, which is shown in *figure 1* below.

Figure 1: Hashimoto Lab's pool, 2.2mx1.5mx0.6m

The size of the pool is sufficient to test the control of the robot and since it is not deep, we can easily recover the robot in case of a problem. The grey bars that we can see are metal bars that are not fixed and that we can place wherever we want to simulate obstacles.

So, in this report I'll expose first the mechanical design of the robot we created, its structure and actuators, to fully understand how they work. Then, in the second part, we'll look at the electronics, to see how the robot uses its actuators with input sensor data to guide itself. Finally, before concluding, I'll briefly outline possible improvements to this robot, whether for a future student working on this robot, or another prototype that would be similar.

1. Mechanical Structure

Before talking about the robot's autonomy, on-board electronics, and intelligence, I'll start by explaining the robot's mechanical design, so that we understand what's available to guide it.

1.1 Tail

When I first arrived at the beginning of May 2023, Stephane had finished the tail of the robot, the part that propels the robot, and the most mechanically difficult and complex part.

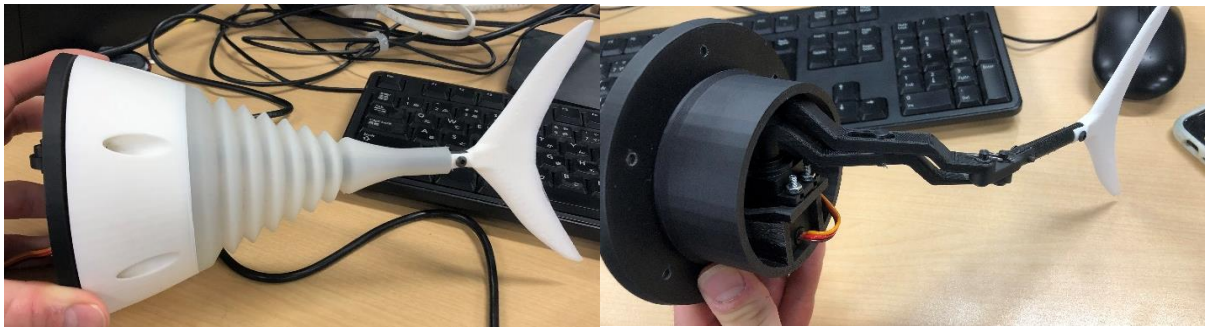

Figure 2: pictures of the fish-bot tail

Figure 2 is where I caught up on the project. The tail was done and mounted with a servomotor. I was able to control it with an Arduino uno board to make the tail move to imitate a fish-movement from the very first days. The inside of the tail is detailed in *figure 3.*

Except for the servomotors and the bolts, every part was 3D printed in the lab by Mr. Weber. The fin of the tail is inspired by a real tuna shape. The flexible accordion section that allows the tail to move is made of silicone and obtained from a 3D printed mold.

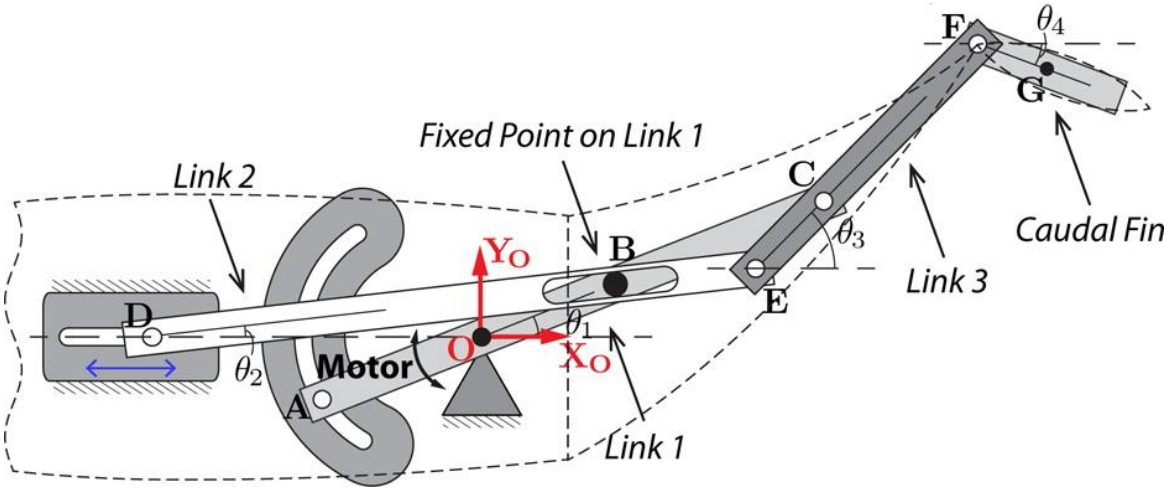

Figure 3: schematic of the robot tail ***[1]***

The technical solution of *figure 3* is inspired from Bio-Inspired Aquatic Robots, UC-Ika Series from University of Canterbury, New Zealand. **[1][2]** When the servomotor (fixed in A) moves, it gives a movement to the tail that looks like a fish.

Mechanically, a total range of 60° is available for the robot tail. The neutral position for the servo is 90°. So, to move forward, the idea is simple: we vary the tail position between 60° and 120° with a sine. To do this, we multiply by 30 the sine of the frequency at which we want the tail to move. We thus obtain the interval [-30;30] to which we add the 90° of the neutral position. This gives a sine varying between 60° and 120°.

Here is a quick demonstration of a such function:

https://www.youtube.com/shorts/MoKBBWCkwAY

https://youtube.com/shorts/WTRZjsFql3A?feature=share

To make the robot oscillate less, we can reduce the angle so that the tail makes smaller movements, thus increasing the frequency.

To make the robot turn on one side, the idea is to quickly move the tail to one end (60 or 120°) and then slowly bring it back to the center. **[3]**

Here's what it looks like on video:

https://www.youtube.com/shorts/9doFa13auB8

The interior part of the tail, i.e., the movable part is protected by a silicone structure that can be seen in *figure 2.* This silicone part was also made in the lab with the 3D printed mold we can see below.

Figure 4: mold for the silicone tail structure and silicone

To obtain the silicone, all we needed to do was prepare a mixture of half the two products listed above (*figure 4*) and pour it into the mold.

However, this has led to a problem that we had not suspected. The overpressure of the water when the robot is submerged, and the movement of the inside ballast causes the tail to deform (*figure 5*).

This would have made it more difficult to control depth, since the volume of water displaced by the robot and therefore Archimedes' thrust is reduced.

Figure 5: deformed silicone(left) and O-rings(right)

To solve this problem, we printed O-rings to insert into the slots in the silicone part (*figure 5*). This allowed us to gain strength while maintaining the flexibility required for the tail to move. This was not the primary purpose of these slots, but it worked very well.

The main goal of these slots was at first, to avoid large folds while maintaining good flexibility of the silicone. Because at the beginning the following tail shape was created (*figure 6*), using the same method of pouring silicone into a mold.

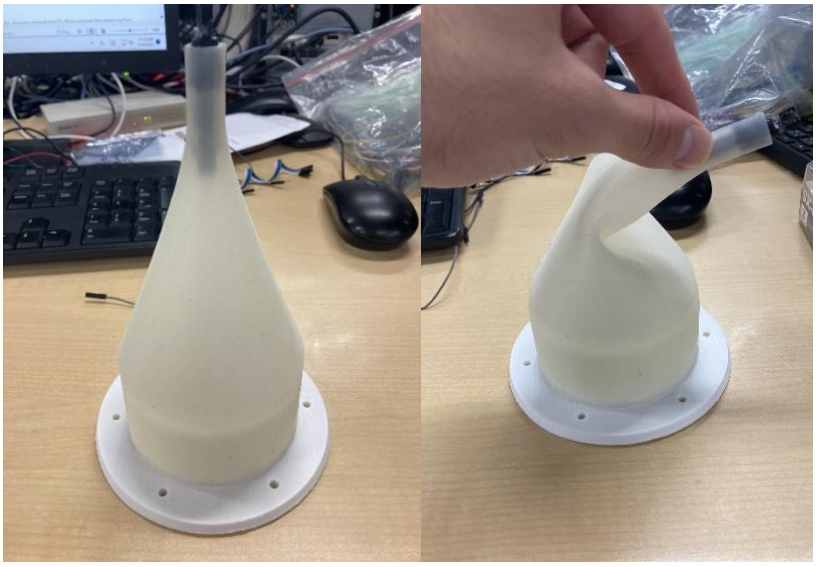

Figure 6: pictures of the first silicone tail

1.2 Fins

At the end of this tail, we can find the main fin, which provides the energy for the fluid to propel the robot. The first fin to be printed was inspired by the shape of a tuna tail. Then we decided to print four different kinds of fins to see which was the best *(figure 7).*

Figure 7: picture of the fins

From left to right, we have the regular fin, the tuna shaped one, then we have the same one but filled in the middle, after that we have a hammerhead-shark shaped fin and finally the normal fin but multiplied by 1.5. The second one and the last one gives some much power that their resistance to the fluid broke a part of the tail as the torque of the servomotor is around 35 kg.cm.

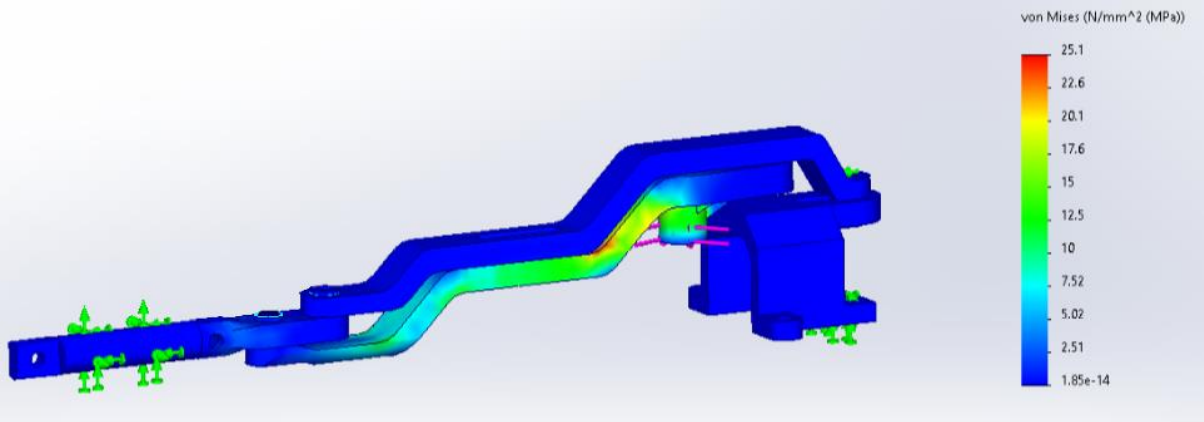

Figure 8: constraints of von mises in the tail

Figure 9: broken tail

As we can see in *figures 8 and 9*, the tail broke exactly at the maximum of constraints due to the servomotor torque and the fin fluid resistance. Following this small accident, we decided to use only the original tuna-inspired fin and the hammerhead shark-inspired fin. We also decided to reduce the speed of the tail and to print again to broken part.

Using the two remaining fins and filming from above the pool, we compared the performance of these two tails, which turned out to be similar *(figure 10)* even though they have a different shape. The only difference I noticed is that the standard tail gives stronger jolts to the water and makes the robot oscillate little bit more (in terms of heading), whereas the hammerhead tail realizes smoother movements.

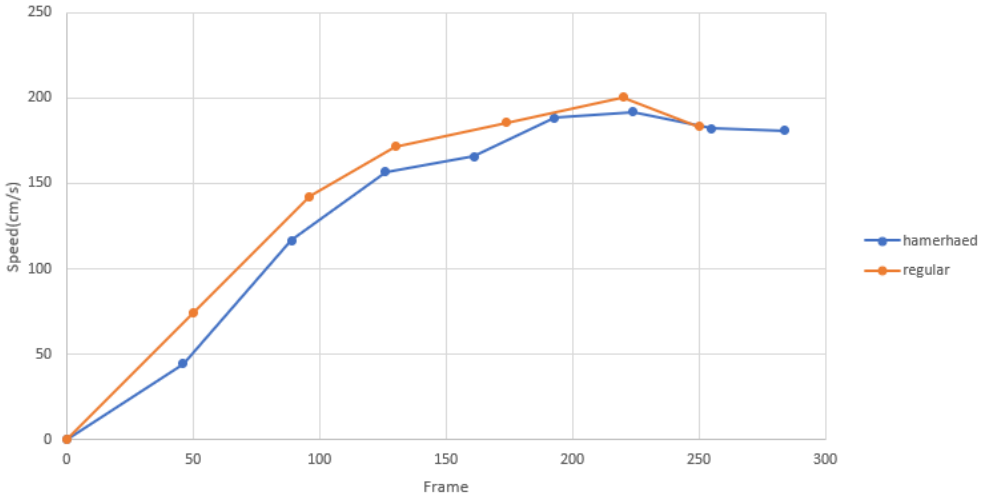

Figure 10: speed as a function of the frames of the camera

We can notice in *figure 10* that the maximum speed is about 20cm/s. As the camera frequency is 30ips, the maximum speed is reach in approximately 5-6s starting from a zero speed.

1.3 Mounting and waterproofing

The robot is made up of 3 main parts, the tail we saw earlier, the central part and the front part.

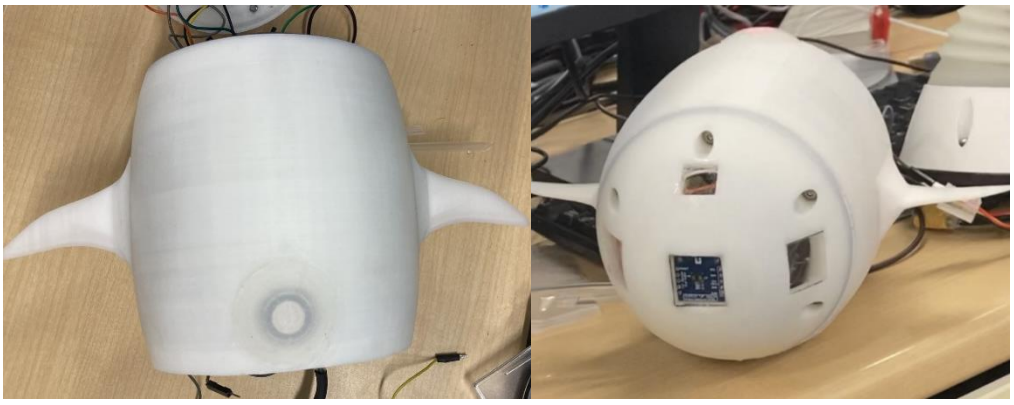

Figure 10: central (left) and front part(right)

These components are also manufactured using 3D printing technology. However, the fins in the central part do not function as actuators, their purpose is primarily to improve the robot's stability and aesthetics. The fins were printed separately, then assembled using superglue. This approach was adopted to minimize printing time and reduce the amount of material required for the supporting structures. Their major drawback is that they are very fragile in comparison to the robot mass, they broke several times and we had to glue them back together.

On the central part, we can see the ON/OFF button. At the time of printing, a hole had been provided for the button we had previously purchased. However, although the manufacturer claimed that this button was waterproof, we noticed leaks in the grey ring, which is an LED that lights up red when the robot is switched on (right of *figure 10*).

To solve this problem, we created the protection visible on top of the button. It was silicone molded with another mold we created. Then we glued this protection. The extensible properties of the silicone allowed us to press the button while keeping the inside area totally waterproof.

While the central part is used for the buoyancy control including a ballast, a pressure sensor and weights, the front part is used to store most of the embedded electronics and sensors we will be talking about later.

To assemble these three parts, there are a total of twelve bolts that need to be fitted (6 for each joint) as we can see in *figure 11*.

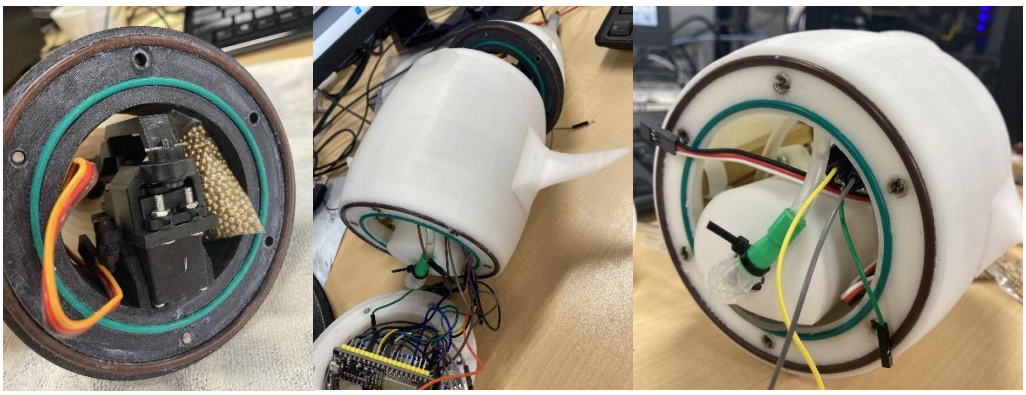

Figure 11: assembly of the three parts with joints and inserts

Between parts there are flexible O-rings for the waterproofing. The brown joints act as the main barrier. Since the 3D-printed parts of the structure have a few small defects, the brown joint may be in direct contact with water in certain places. After testing, and with the use of silicone grease (white traces in the photo on the left of *Figure 11 and on Figure 9*), these joints proved to be very effective, and enabled the green joints to block any water ingress due to the screws. The screws are placed in inserts, which have been glued into holes provided for this purpose in the middle structure.

1.4 Balancing and sinking with weights and ballast

The volume of the robot is about 2.5litres, so for sinking we want its mass to be around 2.5kg. With the robot's structure and all its components, we reached a mass of around 1.4 kg. So, we had to find ways of adding weight. This first idea was to add some lead because its density is over 11, and it is very cheap. This means that for one unit of volume of this material added to the robot, we compensate the Archimedean thrust of around 11 units.

However, the university's policy is to abolish the use of this metal, as it is harmful to the environment. The metal we finally picked up was brass. It was more expensive and less efficient than lead would have been (density: 8.7) but it deserved well our purpose. We also could have used cast iron because this material is very cheap, but its effectiveness was even less good. It was also more difficult to find the bars and tiny balls we used.

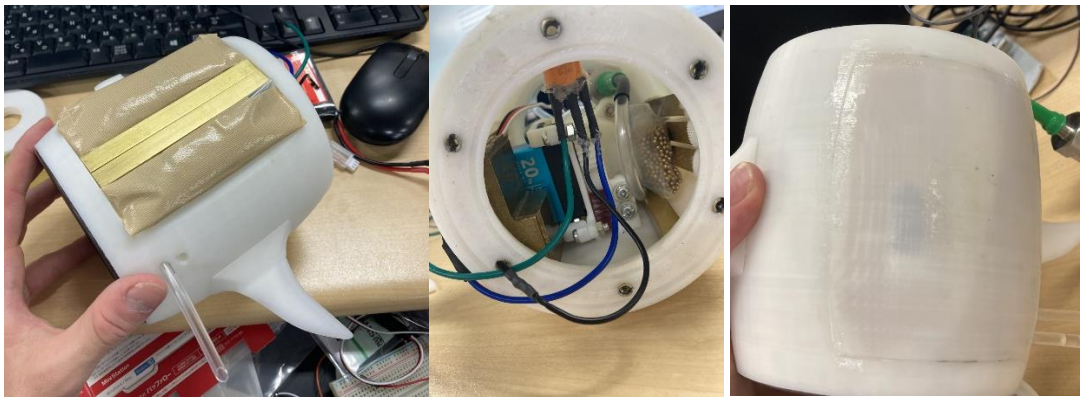

Figure 12: brass weights

The weights we added on the figure above are located at the bottom of the robot to have better stability. With a such tumble effect the roll angle remains null. In the picture on the left, the robot is upside down, this brass is found below the floor of the middle photo. Then we glued this stock of brass made with bars and little bags of brass balls with the appropriate shell (picture on the right).

With all these weight additions, the overall density of the robot manages to approach 1 while remaining below it, so as not to sink immediately.

Finally, to control buoyancy, we use the ballast shown in *figure 13*, which can also be partially seen on *figures 11 and 12*.

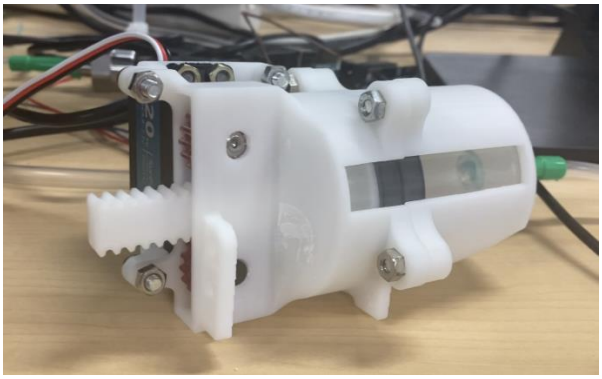

Figure 13: ballast

Basically, this ballast is made with a servomotor and a syringe. The white parts are made to guide the syringe and the gears attached to the servomotor. A rack is attached to the syringe, and it is powered by the movement of the gear attached to the servomotor. A plastic tube comes out of the ballast at the front (*figure 11 and 13*) and exits via one of the holes in *Figure 12(left)*.

This allows the robot to pump water in and out to add or remove weight. In this way, it can oscillate its overall density above and below 1 to reach a desired depth.

Here are two videos, in the first one we can see the ballast working separately from the robot and in the second one, the robot going up and down because of the ballast. In this video, the code sent to the robot was just to totally fill the ballast and after few seconds to empty it completely.

https://youtu.be/mdpfmBM87V0

https://youtube.com/shorts/p2riFoA6LqM?feature=share

We need to be careful when using the ballast because despite the diameter of the tube is small, we cannot pump water too fast, otherwise some of the gear teeth can skip out. This is due to aspiration resistance which may not be linear to the aspiration speed. The two extreme positions of the ballast actuator, in abutment on both sides of its support, are 0 and 175 degrees. 0 is when the ballast is full and 175 when its empty. Since the ballast servomotor can go from 0 to 270 degrees, if any teeth are skipped out, the servo's initial position is no longer the right one, and it can continue to force even though it's at the limit, at the risk of breaking the entire mechanism.

Furthermore, the ballast can extract a total of approx. 75 ml, i.e., 75 g. This is low compared to the 2.5 kg of the robot (3%). Since the ballast is slow for the reason given above. This implies that the robot has a lot of inertia, making it difficult to control its depth quickly. We will see how to do that in part 2.3.

2. Embedded electronics and sensors

In terms of electronics, we've already seen in part 1 that the robot has two servomotors as mechanical actuators. They are each used to perform one of the robot's two main tasks: obstacle avoidance and depth control. We're now going to look at how to use them with input from the sensors. Overall, we'll be looking at everything inside the robot that makes it autonomous.

2.1 On-board computer and communication

We chose the ESP32 board as the embedded computer because it already had a Wi-Fi module (*figure 14*, gray square), was small and operated under Arduino. Of course, we don't want to plug the card into the computer each time we want to send a new code or adjust some constant, because the mounting and unmounting of the robot is very constraining due to the lack of space and waterproofing constraints.

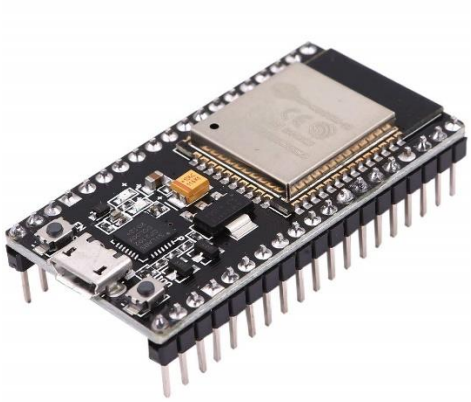

Figure 14 : ESP32 board, 5.5x2.8cm

We could have chosen a Raspberry Pi 4 and connected with SSH method, but it was too big to fit in and Arduino was easy to use, even if we cannot multi-thread.

Both ESP32 boards connect to the local network of the lab with the library Wifi. Using the library Webserver, we just use the IP address of each card to connect and upload code. To do this, we simply type the correct card IP address in the search bar of a standard web browser and enter the correct user and password, which are admin and admin.

```
IP address tail: 192.168.50.67
IP address ballast: 192.168.50.99
```

Then all we have to do is choose a compatible file to upload, .ino.bin format found in the build folder, which is created when the compiled binary is exported from the Arduino ide.

On my git, the Arduino codes corresponding to each board are test_buoyancy for the ballast and OTAWebUpdater2 for the tail.

2.2 C2 robot architecture

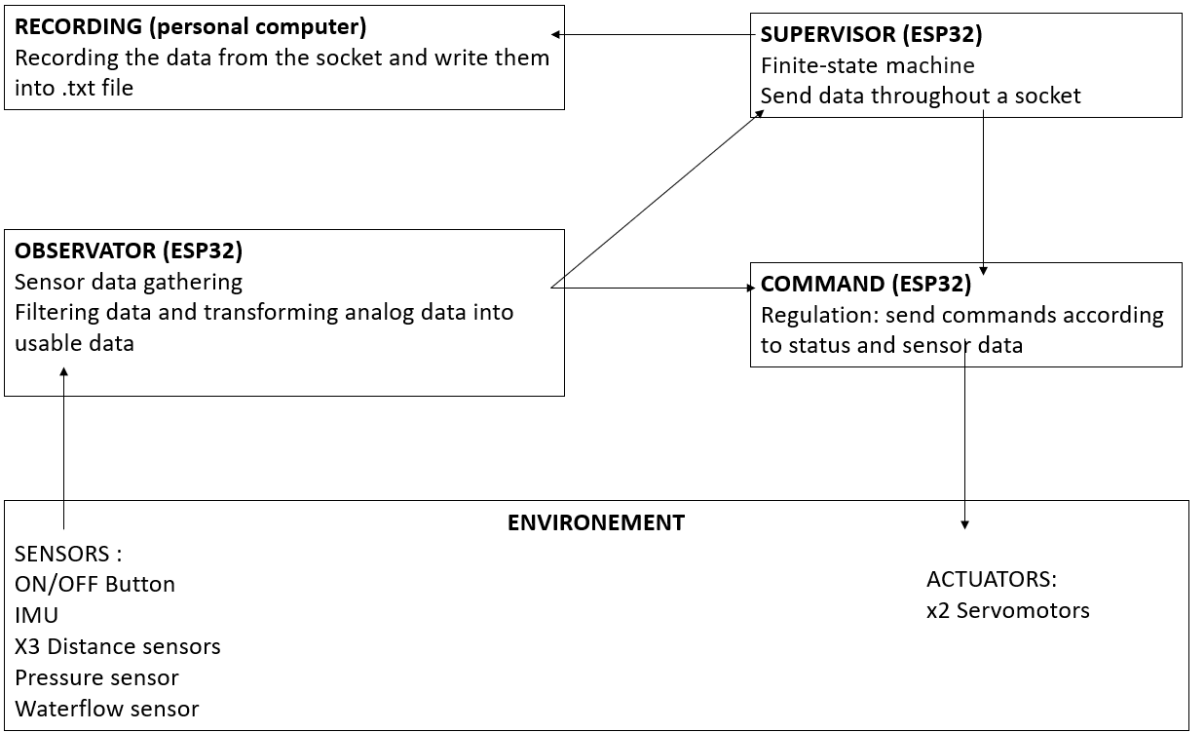

Figure 15: C2 robot architecture

The previous scheme is not entirely accurate, since there are two ESP32 cards communicating with the personal computer.

Originally, this robot was designed for a single board performing both main tasks and tests were carried out with only one task at a time. However, this type of board can't handle multi-threading, so when I wanted to combine the two different codes for ballast and obstacle avoidance, the robot was too slow and didn't carry out its actions properly. Indeed, without multi-threading, the robot waits until it has finished one action before starting another. So, you can't control the depth while trying to avoid an obstacle, and vice versa.

Returning to the C2 architecture, this doesn't change anything except that we have two IP addresses for the recording part, and we must power two boards instead of one. We can also see this as two architectures like the one shown in *figure 15*, but with a separate environment between the 2 schemes. Pressure sensor, ON/OFF button, and servomotor on one side, and everything else on the other, but with the ON/OFF button in common *(figure15)*.

Thus, the robot is controlled by two ESP32 boards, which are similar to Arduino nanos and allow wireless communication when the robot is at the surface. They are directly connected to the sensors and actuators as follows:

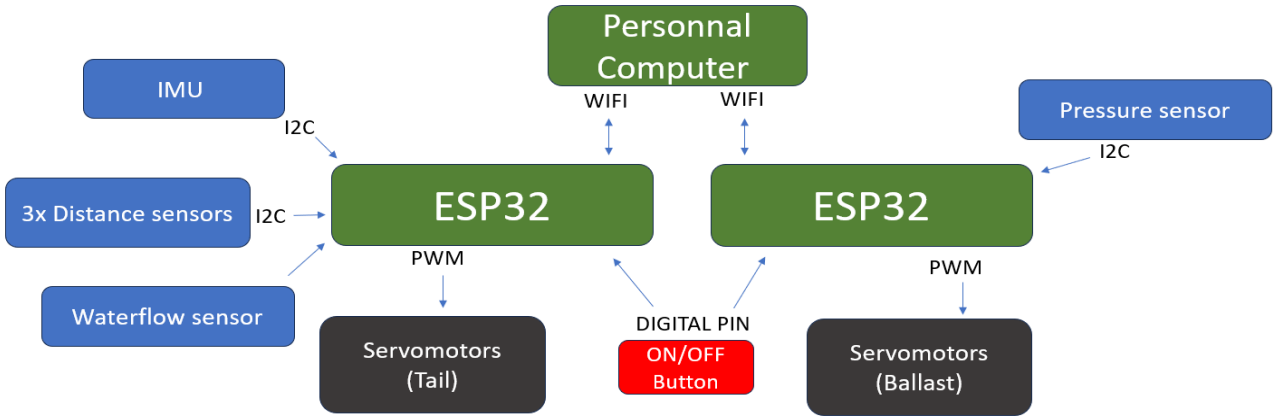

Figure 16: robot communication scheme

I haven't included information about the waterflow sensor, as its implementation has not yet been finalized and the type of communication is not fixed. (see 2.4)

In addition, this figure is not totally accurate, as there is in fact an I2C multiplexer *(figure 17)* between the I2C sensors of the first board controlling the tail. We added that component because there was a conflict between these sensors, particularly the 3 distance sensors, which all had the same I2C address. This meant that the values of only one of all these sensors could be obtained before multiplexer was implemented.

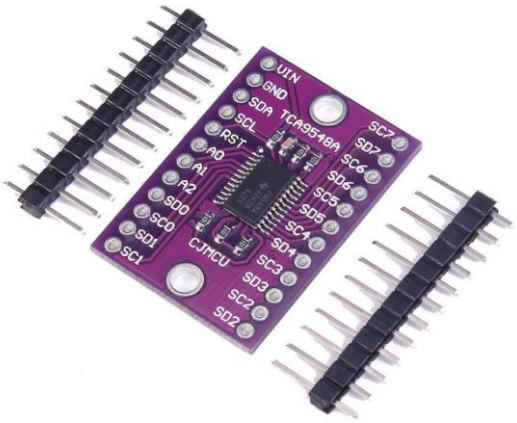

Figure 17: Multiplexer I2C TCA9548A (4x2cm)

This multiplexer is connected using I2C to the ESP32 board and can receive up to eight I2C inputs. All we had to do in the code was to choose the channel number you want to listen to, between 0 and 7.

2.3 Buoyancy control

We saw in 1.4 that the robot can go up and down using the ballast and the weights. But the main code does not only fill and empty the ballast tank, but it also controls the depth with a pressure sensor.

We used the Adafruit MPRLS sensor, which is very accurate. This sensor is also very small, so it fitted perfectly in the robot. We glued it to a plastic tube which we brought out through a hole in the side *(Figure 18).* We then sealed this hole by gluing the tube inside with superglue. The other tube is for pumping and releasing water with the ballast.

This sensor reacts with air. When the robot dives, there is an overpressure inside the tube equal to the water pressure.

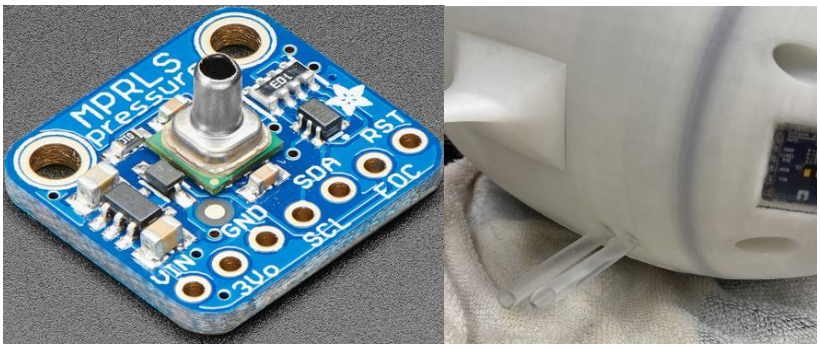

Figure 18: Adafruit MPRLS (1.7x1.7cm) and tubes coming out of the robot.

So, with this pressure information, we can deduce the depth. This allows us to control buoyancy.

The first idea was to set a desired depth, pump in water with the ballast when the actual depth was shallower and empty it of water when the desired depth was deeper. And thus, oscillate around this desired depth.

However, with the robot inertia mentioned earlier, the result was equivalent to what can be seen on the last YouTube link when filling and emptying the ballast without control.

Then the second idea was to look at the pressure derivative to see whether the robot was sinking or not. With this information, we used the following finite-state machine:

Actual depth	Sinking	Not Sinking
Desired depth – [0,50cm]	Do nothing	Pump water
Desired depth + [0,50cm]	Release water	Do nothing
Desired depth – [50cm, ∞]	Pump water	Pump water
Desired depth + [50cm, ∞]	Release water	Release water

Figure 19: finite state machine

Nb: The z axis points downwards, so depths are positive distances.

In the code, the Boolean Sink is only true for significant pressure variations indicating that the robot is sinking or going up. For small pressure variations, such as at the start of a mission when the robot is quite stable, but a little bit moving at the surface, this Boolean is false and does not stop the robot from pumping water.

So, when the robot starts to sink, it stops pumping water until it passes the limit of the desired depth, and then rejects water. On the contrary when the robot is going up, it waits until it reaches the limit before pumping water again.

However, this was not enough to precisely control the depth. The robot still touched the bottom and the surface when going up and down as in the video.

The depth of the pool was 60 cm but there was 40cm of water, and the desired depth was 20cm. This means that with this approach, the robot oscillates more than 20 cm around the given depth on either side. Considering how much time the robot stays at the bottom and at the surface before going up or down again, I estimate that the robot would oscillate in an approximate 1m range area (50cm each side).

The final answer is an improvement on the previous approach. When we're close enough to the desired depth, the moment we detect that we're sinking or rising, we pump or dump water in the other direction. For example, if the robot is close enough to the desired depth and starts to sink, it'll reject a little water (raise the servo a few degrees), even if the desired depth is not reached, and see if he is still sinking. So, we can descend by increments while checking the desired depth. We also do this the other way, and here's what it looks like on video：

https://youtu.be/2F84bx-p3y8

https://youtu.be/SoO2fsS2jeY

2.4 Waterflow sensor

To know the speed of the robot in real time, we used a waterflow sensor.

This sensor is developed in our neighboring laboratory, Motosuke lab. This sensor is a complex network of resistors heated with Joule effect. Since there is dissipation of energy with the water in contact, the more energy the sensor dissipates, the more current it must absorb to heat up. **[4]** The higher the speed of the fluid is, the more energy is dissipated. So, using the heat transfer laws of energy dissipation by convection, we can link the current consumed for heating to the speed of the fluid, and therefore to that of the robot.

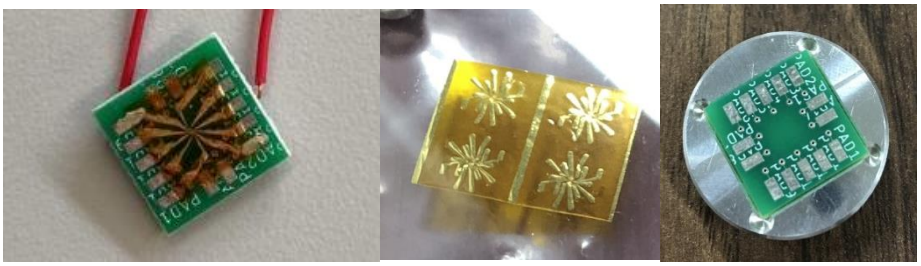

Figure 20: waterflow sensor(1.5x1.5cm)

The resistor network is shown in the middle photo of *figure 20,* it is made with gold, and it is attached to the green electronic part on the right, with titanium and polyimide, using different processes to obtain the final sensor on the left. The problem is that it's not a sensor in its own right because the two red wires you can see on the left picture are the + and – poles. In other words, no information is sent back to any card. This system needs to be paired with another to have an output.

The first approach we took was to measure the current consumed by the sensor to determine the amount of energy dissipated. So, we tried to apply different water velocities to the sensor, manually and consistently while measuring the current. We assumed that when the current doesn't change, the speed is constant.

Here's how we took the measurements *(figure 21):*

Figure 21: current measurements

After connecting the amp meter to the sensor and powering the system, we attached the sensor to a plastic tube. All of this was mounted on a rack that can be moved along a metal bar *(figure 21, right)*. So, by measuring the time it takes to go from point A to point B, trying to keep the current constant, we can deduce the average speed which we take to be constant. In this way, we took a series of measurements at different speeds.

The power balance shows that the power consumed by the Joule effect is equal to the sum of the accumulation power, which corresponds to the free-running energy, and the power dissipated by heat exchange. Given that power dissipated by radiation and power dissipated by conduction are negligible compared to power dissipated by convection, we have:

$$RI^2 = Cte + h * A * \Delta T \quad (1)$$

Taking the temperature of the water and of the sensor as constant (note that this is not true for the resistor network, especially for the central point, which heats up a lot), we took ΔT as a constant and obtain :

$$RI^2 \approx Cte1 + Cte2 * h \quad (2)$$

By taking *h* proportional to the square root of the Reynolds number for laminar flow **[4]**, which we can do here with a fish moving at speeds of less than 20cm/s in water and with a to 10^{-1} metre characteristic dimension. Since Reynolds number and velocity are proportinal, we finally obtain :

$$RI^2 \approx Cte1 + Cte3 * \sqrt{Re} \quad \Rightarrow \quad \boxed{I \approx b + a * \sqrt[4]{V}} \quad (3)$$

R : Global restitance of the resitance network in Ω.

I : Current in A

Re : Reynolds number

h : Thermal convection coefficient in $W.m^{-2}.K^{-1}$

A : Surface of the resistance network in m^2

ΔT : temperature difference between fluid and sensor in K

V : Speed in $m.s^{-1}$

Cte1,Cte2,Cte3,a,b $\in \mathbb{R}$.

Considering that and with the measures as follow on *figure 22*, we used the Python library scipy.optimize to obtain the trend curve (source code on the Gitlab).

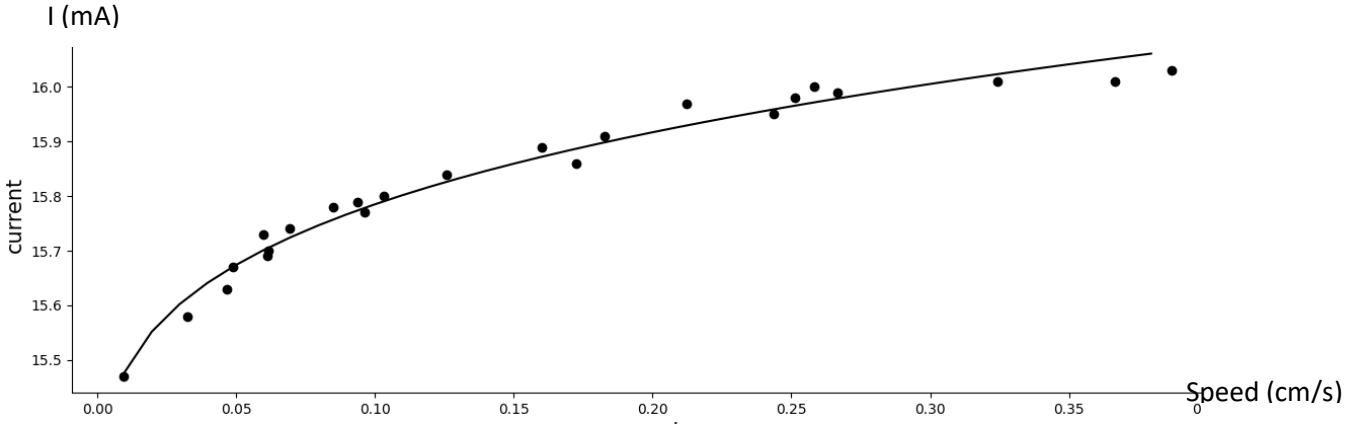

Figure 22: measures of current as a function of speed and trend curve

Finally, we got $b \approx 15.4$ and $a \approx 1.24$.

The difference between the measurements and the trend curve remains small, so we can say that these results are satisfying, considering the precision with which the measurements were made.

However, those measurements were made with a precise amp meter which wouldn't fit into the robot. We tried with several hall-effect current sensors, Arduino compatible, but we did not manage to find one precise enough. The best we found had a resolution close to 1 mA, but this is not sufficient to detect small variations between 15.4 mA and 16.0 mA.

After discussions with Motosuke Lab, the conclusion was that measuring current for this sensor is not the best way to have the speed, and that another prototype will be developed to measure the energy dissipated by measuring the temperature of the central point of the resistor network.

2.5 Positioning with IMU (Inertial Measurement Unit) and Waterflow Sensor

As soon as the robot is completely submerged, it is no longer possible to communicate with it via electromagnetic waves. The use of GNSS (GPS) is therefore impossible. Here, we're only looking for the position in *(x, y) since* the position in *z* is obtained with the pressure sensor. To do this, we use the initial position, which we integrate using Euler's method **[5].**

$$x_t = x_{t-1} + v_x dt$$
$$y_t = y_{t-1} - v_y dt \quad (4)$$

We therefore need speed in (x, y). To get it, the idea was to retrieve the Euler angles from the IMU using quaternions and use them as follows with the initial heading ψ_D :

$$v_x = \|\boldsymbol{v}\|\cos(\psi - \psi_D)$$
$$v_y = \|\boldsymbol{v}\|\sin(\psi - \psi_D) \quad (5)$$

Where ‖v‖ is the velocity norm, which we would normally have measured with the waterflow sensor. ψ the heading obtained with the q quaternions of the IMU *(figure 20)* as follows **[5]:**

$$\phi = \begin{cases} \arctan\left(-\frac{2q_yq_z - 2q_xq_w}{2q_w^2 + 2q_z^2 - 1}\right) & (\cos\theta \neq 0) \\ \arctan\left(\frac{2q_yq_z + 2q_zq_w}{2q_w^2 + 2q_y^2 - 1}\right) & (otherwise) \end{cases}$$
$$\theta = \arcsin(2q_xq_z + 2q_yq_w) \quad (6)$$
$$\psi = \begin{cases} \arctan\left(-\frac{2q_xq_y - 2q_zq_w}{2q_w^2 + 2q_z^2 - 1}\right) & (\cos\theta \neq 0) \\ 0 & (otherwise) \end{cases}$$

※製作例

Figure 23: IMU, BNO55 sensor, 1x2cm

2.6 Distance sensors and obstacles avoidance

To identify obstacles, we use distance sensors that work with infrared lasers. The sensor emits infrared waves and, by reflecting these waves against an obstacle, deduces the distance to that obstacle. It has a range of 27 degrees and 4m in air. However, in the pool, no matter where the robot was, I didn't measure any distance greater than 1.5m, which means it's blind beyond 1.5m. The robot has 3 of these sensors, all on the robot head, 1 frontal and 2 lateral.

These sensors can be seen in *figure 10* (right) *24*:

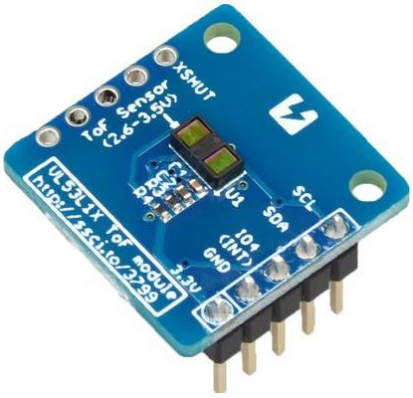

Figure 24: distance sensor VL53L1X, 2.5x2.5cm

Due to the infrared rays that may be present from the sun or other sources, measurements may be incorrect in the air. In addition to that, because the windows of the robots (*figure 10*) are made of a material like plexiglass to protect against water, there are problems of reflection/absorption of some wavelengths which can influence the measurements. Sometimes in the air the robots sensed an obstacle at 40-60 cm when there was nothing closer to a few meters. In spite of that, there don't seem to be any problems when the sensor is submerged.

The robot is then guided as follows: if an obstacle is too close to the front sensor, it turns towards the side where the lateral distance is greatest, using the turning function of part 1.1. Similarly, if an obstacle is too close to one of the side sensors, the robot will turn away from it. Otherwise, it swims straight ahead. Given the dimensions of the pool and the obstacles used, the minimum distance for the robot to swim straight ahead is 50 cm for the front sensor and 10 cm for the side sensors. Of course, if the robot were to be used in a different environment, these values would have to be adjusted.

Here's an example of what it looks like on the surface, not combined with buoyancy control:

www.youtube.com/watch?v=UrIDJrSAj24

In this video, we can see that the robot's lateral fins can be a problem when close to obstacles. Since the balance is already assured by the weights, they can be cut off.

This is how it look like when I added the other ESP32 board to control the depth at the same time:

https://youtu.be/IagdZVsx55c?si=vLsMt1raCqti5NA_

2.7 Power supply

This whole system is powered by a Lipo 2S battery with a claimed performance of 11.1 Wh *(figure 25)*.

Figure 25: battery Lipo 2S de 11.1Wh, 6.8 x 3.5 x 2.8 cm

After calculating that the robot will consume a maximum of 15W when the two 5W servomotors are active, we can deduce that the minimum autonomy is around 44 minutes. In addition to that both servomotors are not always active so we can deduce that the average autonomy is more than that, probably up to one hour.

This battery powers the robot following the next figure:

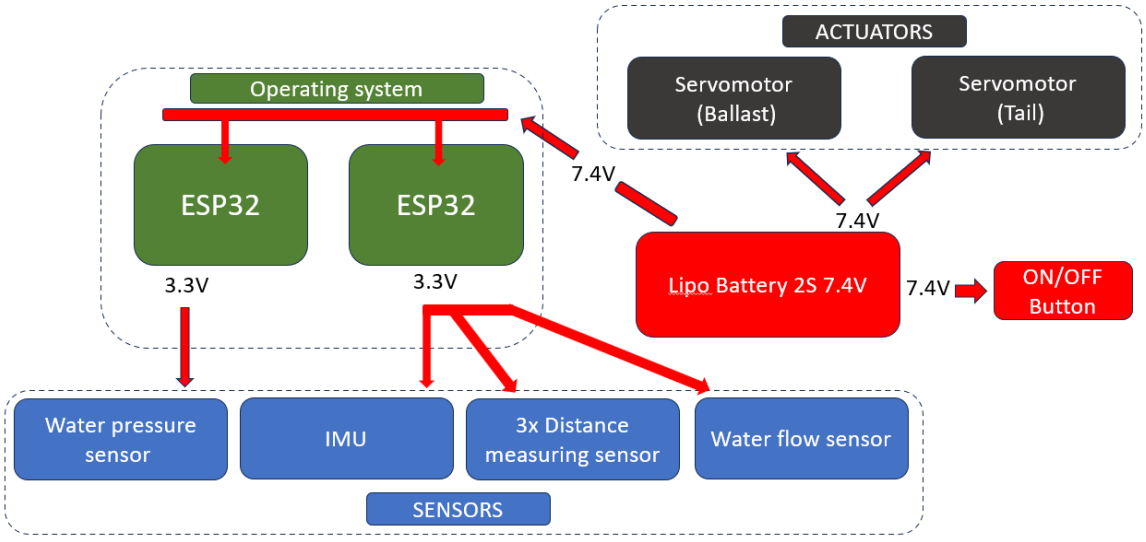

Figure 26: robot power supply scheme

Note that the ON/OFF is not a real switch, it doesn't cut the power or anything, it can be seen as a sensor who sends rising edges to both ESP32 cards. These cards run their code when the button is pressed once and do nothing when pressed another time.

3. Going further

Although we have gone far with this project, the robot isn't perfect and would benefit from a few improvements:

Firstly, even though it's not cheap, it would be very useful to make the structure with a more or less heavy metal. This would have the huge advantage of not having to bother with all the brass weights inside, which require a lot of space and make wire connections difficult.

Secondly, it would also be good to use a Raspberry Pi4, as this board is more powerful and allows multi-threading. We could also easily attach a camera to this board, which would be very great for obstacle detection using image processing software/libraries. By using a camera, we can differentiate the type of obstacles and make the robot react appropriately.

Thirdly, there is still work to do on the waterflow sensor to determine properly the robot speed.

In addition to that, we've seen that pectoral fins can be an obstacle to the robot's progress, and since they're not very useful, we could think about a design without them.

Moreover, since this robot is heavy compared to the water that can be pumped, we could create a robot with a less rounded shape, but more oval, oriented upwards, saving space on the sides.

Finally, the turning method described in 1.1 *(figure X, mode C)*, is not the only way to do this **[3]**. I chose this method because it allows me to turn abruptly, since the dimension of the pool implies that I had little room for maneuver.

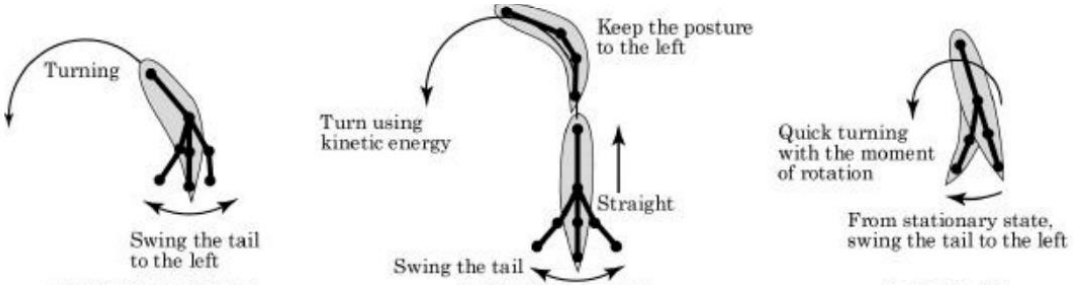

*Figure 27: methods for turning **[3]**, mode A, B and C from left to right*

In fact, if the robot is in a larger expanse of water, a lake for instance, and we want it to follow a longer course, all we must do is follow the same technique as for swimming straight ahead, but with an offset of a certain number of degrees on the robot's tail *(Figure X mode A)*, depending on where you want to go. It would be smoother.

On the other hand, if the robot already has sufficient speed and you want to slow down as well as turn, you can simply block the tail on one side or the other to act in the same way as a rudder *(Figure 27 mode B).*

Conclusion

To conclude, we saw that this robot has two servomotors as actuators, controlled by two different ESP32 boards which give instructions depending on the sensor inputs (pressure sensor, ON/OFF button, distance sensors, IMU, waterflow sensor).

This enables the robot to control depth through a ballast system, which has been designed with a syringe, a servomotor. It can also reach a maximum speed of 20cm/s with the other servomotor, which is the tail accelerator, while avoiding obstacles using the distance sensors at the front and positioning itself thanks to the IMU and waterflow sensor.

I also showed how to improve this prototype for the future, in terms of structure, electronics and ways of turning.

Working on this project allowed me to solidify some of the knowledge I had acquired during my studies at ENSTA-Bretagne, but also to apply it to electronics and mechanical design, sensors, Arduino, and network communication between different computers.

It also taught me to learn how to better choose sensors and actuators on the Internet, which communication methods are suitable for each and to be more discerning when reading datasheets in response to the needs of a project.

I'm glad that I was almost able to complete this project, given the ambit of the objectives, and would like to thank Professor Hashimoto once again for this fabulous experience in Japan, both human and professional.

References

[1] Sayyed Farideddin Masoomi, Stefanie Gutschmidt, XiaoQi Chen, Mathieu Sellier, 2015 "The Kinematics and Dynamics of Undulatory Motion of a Tuna-Mimetic Robot"

https://journals.sagepub.com/doi/10.5772/60059

[2] "BIO-INSPIRED AQUATIC ROBOTS.UC-IKA SERIES"
https://www.canterbury.ac.nz/engineering/schools/mechanical/research/gutschmidt/underwater-robotics/

[3] Koichi HIRATA, Tadanori TAKIMOTO and Kenkichi TAMURA, "STUDY ON TURNING PERFORMANCE OF A FISH ROBOT"

https://www.nmri.go.jp/archives/eng/khirata/list/fish/isamec2000.pdf

[4] Baptiste LAMOTHE, 2023, Motosuke LAB, Tokyo University of Science "development of a speed MEMS sensor"

[5] Hirotaka Tomi, Taichi Araya, Keisuke Kitano, Member, IEEE, Kenta Matsumoto, *Member, IEEE*, Hiroshi Kobayashi, *Member, IEEE*, and Takuya Hashimoto, *Member, IEEE "*Route Estimation with Obstacle Avoidance toward Autonomous Swimming of Fish-type Robot"

Appendix

ENSTA Bretagne

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :
At the end of the internship, please return this report via mail or email to:

ENSTA Bretagne – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE
00.33 (0) 2.98.34.87.70 / stages@ensta-bretagne.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name* Hashimoto Lab., Tokyo University of Science

Adresse / *Address* 6-3-1 Niijuku, Katsushika-ku, Tokyo, 125-8585, Japan

Tél / *Phone (including country and area code)*

Nom du superviseur / *Name of internship supervisor*
Takuya Hashimoto
Fonction / *Function* Associate Professor

Adresse e-mail / *E-mail address* tak@rs.tus.ac.jp

Nom du stagiaire accueilli / *Name of intern* hippolyte LEROY

II - EVALUATION / *ASSESSMENT*

Veuillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
*Please attribute a mark from **A (excellent)** to **F (very weak).***

MISSION / *TASK*

- La mission de départ a-t-elle été remplie ? A B C D E F
  Was the initial contract carried out to your satisfaction?
- Manquait-il au stagiaire des connaissances ? ☐ oui/*yes* ☑ non/*no*
  Was the intern lacking skills?

  Si oui, lesquelles ? / *If so, which skills?*

ESPRIT D'EQUIPE / *TEAM SPIRIT*

- Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / *Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)*

  A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here*

COMPORTEMENT AU TRAVAIL / *BEHAVIOUR TOWARDS WORK*

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?

7

Version du 05/04/2019

Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?

A B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

INITIATIVE – AUTONOMIE / *INITIATIVE – AUTONOMY*

Le stagiaire s'est –il rapidement adapté à de nouvelles situations ? A B C D E F
(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

Did the intern adapt well to new situations? A B C D E F
(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

CULTUREL – COMMUNICATION / *CULTURAL – COMMUNICATION*

Le stagiaire était-il ouvert, d'une manière générale, à la communication ? A B C D E F
Was the intern open to listening and expressing himself /herself?

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

OPINION GLOBALE / *OVERALL ASSESSMENT*

❖ La valeur technique du stagiaire était : A B C D E F
Please evaluate the technical skills of the intern:

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?

Would you be willing to host another intern next year? ☑ oui/*yes* ☐ non/*no*

Fait à _____, le _____
In _____, on _____

Signature Entreprise _____ Signature stagiaire

Company stamp 椅本卓弥 _____ *Intern's signature*

29th Aug. 2023

Merci pour votre coopération
We thank you very much for your cooperation

8

Version du 05/04/2019