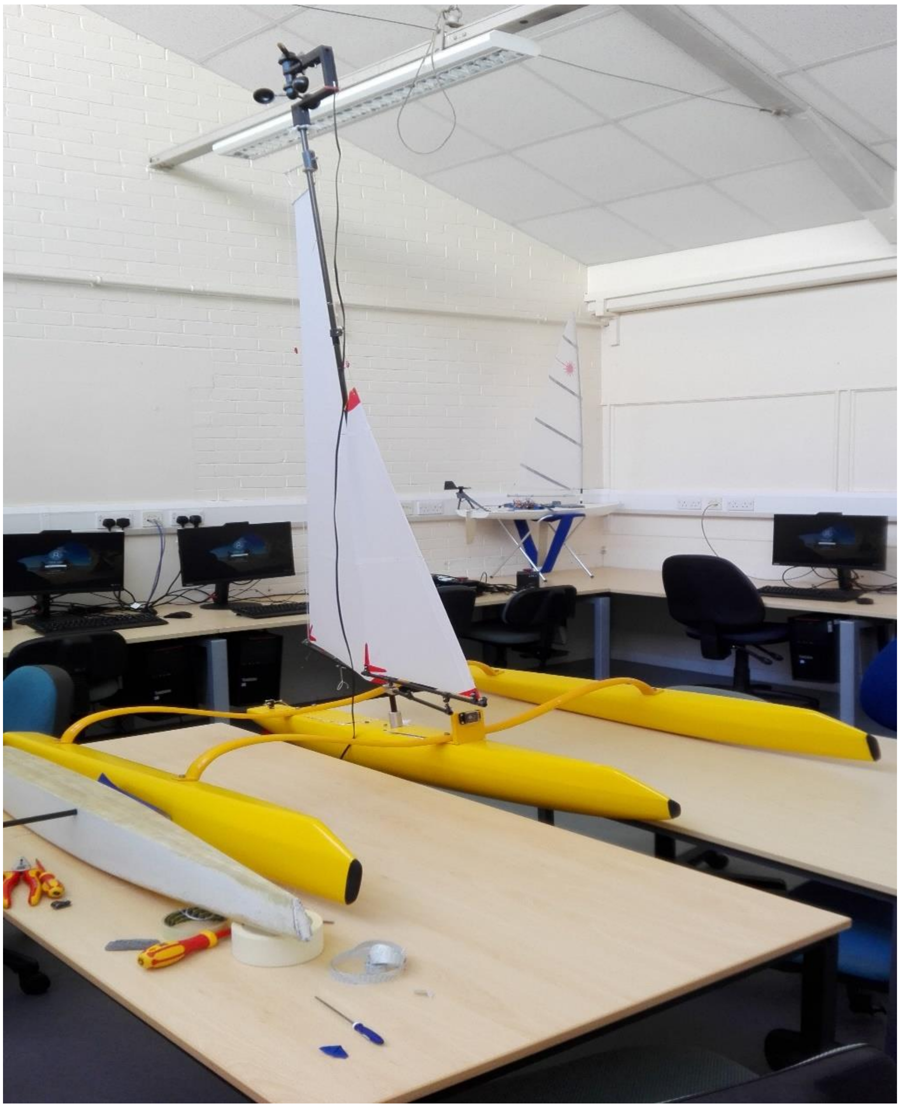


Internship report

Development of an autonomous trimaran for coastal monitoring and marine observation

Supervisor: Doctor Jian WAN, Lecturer, University of Plymouth - AMS Tutor: Professor Luc JAULIN, Professor in robotics, ENSTA Bretagne - LabSTICC	Internship performed at the University of Plymouth from 11/06/2018 to 02/09/2018 HOUDEVILLE Alexandre CI 2019 SPID ROB alexandre.houdeville@ensta-bretagne.org

Acknowledgement

I would like to thank Doctor Jian WAN profusely, who consented to take me on as a trainee student in the University of Plymouth and whose reactivity and reception made it possible to organise this internship in the easiest and the most efficient way. I also thank him for all the benefits he provided for me during this internship.

Similarly, I would like to thank Professor Luc JAULIN, who had offered me different places abroad to do this internship and put me through to Doctor Jian WAN. I would also highlight that the robotic courses he ensured during the year were very useful to perform some tasks in the project given for the internship. I also thank Ulysse Vautier and Christophe Viel for the precious help they gave me during this internship.

Résumé

Ce rapport a pour but de décrire le travail effectué lors de mon stage d'assistant ingénieur à l'Université de Plymouth dans la partie responsable des sciences marines et de l'ingénierie.

L'objectif du projet qui m'a été attribué pendant ce stage consistait en la conception d'un robot trimaran (voilier à trois coques) autonome miniature. Ce robot a été conçu dans le but futur de surveiller et de recueillir des informations marines le long des côtes au bord d'une étendue d'eau.

Ce projet fait appel à différentes compétences informatiques, électroniques et mécaniques. En effet, après avoir assemblé les éléments constituant la structure du trimaran, différents capteurs ont été utilisés afin d'attribuer au robot la capacité de percevoir son environnement. Une fois ces informations environnementales renvoyées au robot, un système de commande autonome a pu être mis en place par l'implémentation de micro contrôleurs afin de permettre au robot d'agir en conséquence et de se déplacer de façon autonome vers des objectifs fixés avant la mission.

Ce projet est basé sur un modèle mécanique de trimaran déjà existant. Le travail réalisé a donc majoritairement consisté en l'étude de capteurs et la fusion de leurs données à l'aide de la mise en place de micro contrôleurs ainsi qu'en l'élaboration d'algorithmes de contrôle autonome notamment permettant le suivi d'amers (points fixes devant être atteints par le bateau lors de la mission) ou l'évitement d'obstacles.

A la fin du stage, des tests en mer ont été effectués sur le prototype afin de valider ou non son comportement.

Mots-clés : voilier autonome, robotique, algorithme de contrôle, suivi de ligne, suivi d'amers, évitement d'obstacles, capteurs, micro contrôleurs.

Abstract

This report aims at describing the work accomplished during my engineer-assistant internship at the University of Plymouth, in particular related to the school of marine science and engineering.

The goal of the project which was assigned to me consisted in designing an autonomous miniature trimaran robot (sailing boat with three hulls). This robot was designed for coastal monitoring and marine observation.

This project made use of different computer, electronic and mechanical skills. Indeed, after having assembled the parts which form the trimaran structure, several sensors were implemented in order to allow the robot to perceive its environment. Once these environmental data are returned to the robot, an autonomous command system was implemented thanks to the use of micro controllers so as to enable the robot to behave consequently and to move in an autonomous way towards targets which should be defined before the mission.

This project was based on an existing mechanical model of trimaran. Thus, for the most part, the tasks accomplished consisted in the study of sensors and in the fusion of sensor data thanks to the arrangement of micro controllers as well as the writing of autonomous control algorithms in particular related to waypoint (GPS coordinates) following or obstacle avoidance.

At the end of the internship, practical tests were conducted with the prototype designed so as to assess its behaviour.

Key-Words: autonomous sailing boat, robotics, control algorithm, path following, waypoint following, obstacle avoidance, sensors, micro controllers.

Table of content

1. Introduction

1.1. Stakes of the internship related to my career objectives

The engineer-assistant internship aims at applying knowledge and skills learnt thanks to the engineering courses in relation to an engineering or research project.

To perform this internship, I wanted to go abroad for several reasons. First, I wanted to experience working abroad as interactions and working organisations are different from those in France. Then, I had never been more than two weeks alone in a foreign country whereas staying abroad for three months is very different, it allowed me to get time in order to immerse myself in the local culture.

Moreover, going in a foreign country means using the local language for work and daily life. Regarding languages in general, I can manage in written English or Spanish but before going to Plymouth, I had much difficulty with spoken English understanding and oral expression thus I had much relied on this internship in order to improve my English skills. Similarly, I wish I had gone to a Spanish speaking country in order to improve my Spanish skills but as English is more common and useful in the world in general, I first decided to focus on it.

Besides, I chose to do my internship in the University of Plymouth as it holds a branch in robotics which corresponds to my pathway at ENSTA Bretagne and since I knew what kind of internships are offered to the students because the students from ENSTA Bretagne had already come there before. Therefore I knew I could find a subject which could really interest me and even if the internship mainly consisted in undertaking a project similar to the projects which can be suggested in the working place, working in a University could also make a clarification for me about research fields as it is something common in the University and it could potentially raise my interest.

My supervisor Jian WAN had suggested different projects like the study of intervals or other projects consisting in developing other autonomous sailing boats. I had already studied intervals a little before coming there and I knew I had difficulty with the use of hardware in projects like implementing sensors so I wanted to learn much more about this practical part of computer science in order to feel more proficient in this branch for future projects that is why I chose to design the trimaran.

1.2. Hosting organisation: the University of Plymouth

The University of Plymouth [1] is an old organisation whose first part was founded in 1862. It is a huge structure hosting more than 23,000 students a year in many fields such as Arts and Humanities, Health and Human Sciences, Science and Engineering, Business, Medicine and Dentistry. The University is also a large research centre related to those different fields.

Figure 1 University of Plymouth

In this structure, I worked especially in the Brunel laboratory, a building related to the school of marine science and engineering. In this building, I essentially worked in a room with computers as it was an internship related to programming yet I also worked in technical areas so as to cut out and stick together some mechanical pieces, to solder electronic components, to drill and to print 3D elements.

During this internship, Doctor Jian WAN, a lecturer in control system engineering and a member of a research group working for Autonomous Marine Systems (AMS) was my supervisor.

2. The project

In this section, I will describe the project I was assigned and I will also explain how I tackled it in a technical way.

2.1. Project issue and requirements

The project on which I worked was entitled "Development of an autonomous trimaran for coastal monitoring and marine observation". Basically, it consisted in designing an autonomous miniature trimaran robot (sailing boat with three hulls). For now, I just had to design the robot so that it works autonomously, however the final goal of this robot will be coastal monitoring and marine observation for instance by collecting some data in the sea as measuring particle concentrations, depth, magnetic field or conductivity for example. The robot could also be used for harbour protection and military observation as it is rather discreet. Finally, a last obvious use of this robot would be acting as a relay to receive or send data to an underwater vehicle by following its trajectory, as current means to communicate underwater on a large scale, for instance from the land, are not much efficient.

To design the robot, main requirements had to be defined in order to abide by the customer's will. The first requirement was to define what "autonomous" means. According to Jian WAN, the robot had to act autonomously in such a way as to follow by itself a pathway characterised by a list of GPS coordinates representing waypoints. The boat also had to be able to avoid unexpected obstacles that it could meet during its trip.

Moreover, the boat also had to be autonomous regarding its movement, that is to say it had to be powered by a local and abundant source of energy: here it was wind power and a battery was used just to power the servomotors and the electronic components.

Finally, a last important requirement was to inform the other marine vehicles about its position and its current state by sending data through the radio channel dedicated.

Some prerequisites had already been completed before I began the project: the mechanical part of the project had already been done for the most part. As the main structure of the boat had already been achieved, I just needed to put the different parts together and to build the mast. Similarly, many electronic components and micro controllers had been already delivered like the IMU, GPS, anemometer, raspberry pi III, Arduino Mega but they still needed to be implemented on the boat.

2.2. Formalisation through system engineering

For any kind of project system engineering is very important since it allows to formalise the requirements given by the customer. In this way the customer can assess the understanding of its requirements by the engineer. System engineering is also a way to make engineers work according to the same base. Finally, it allows to break down the main system by assigning specific functions to the different subsystems.

Different methods can be used to apply system engineering in a project. I chose to use the APTE method as I am in the habit of using it and I think the tools offered by this method are very efficient. This method was created by the French company APTE.

2.2.1. Description of the functions assigned to the system

To describe the specifications of the system, the APTE Method provides two interesting diagrams which allow people to have a quick and intuitive knowledge of the system.

The horned beast diagram describes the goal of the system and its working environment.

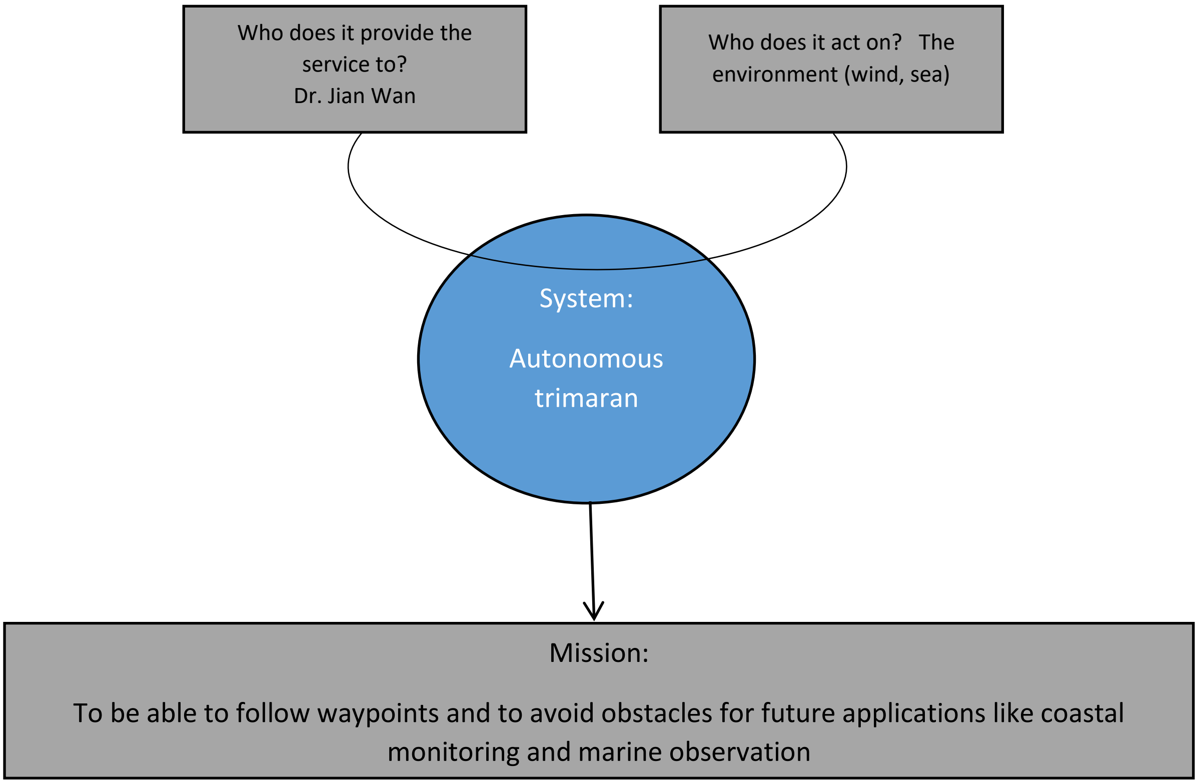

Who does it provide the service to?
Dr. Jian Wan
Who does it act on? The environment (wind, sea)
System:
Autonomous trimaran
Mission:
To be able to follow waypoints and to avoid obstacles for future applications like coastal monitoring and marine observation

Figure 2 Horned beast diagram

The “octopus diagram” raises the different interactions between the system and its environment. It is a clever way to summarise the different functions of the system.

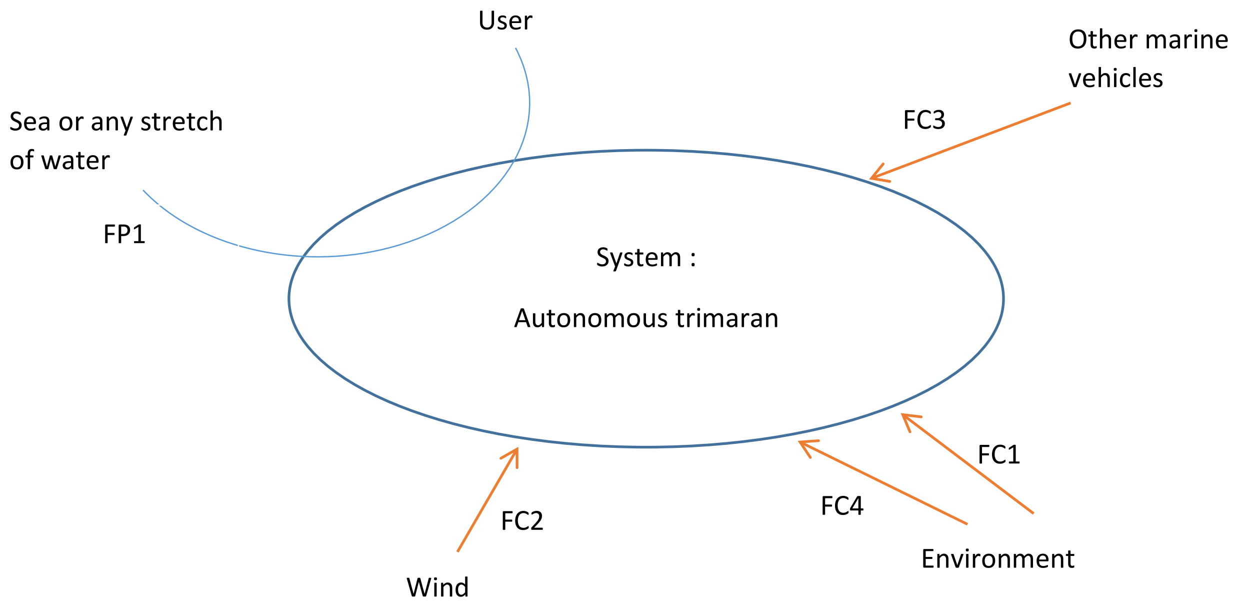


Figure 3 Octopus diagram

	Type	Expression	Significance
FP1	Function	The system must allow the user to collect data from the sea	
FC1	Constraint	The system must move from a point to another one without any human intervention	1
FC2	Constraint	The system must be mainly powered thanks to wind energy	2
FC3	Constraint	The system must inform other marine vehicles about its position and its current state	3
FC4	Constraint	The system must be eco-aware	4

Figure 4 Board summarising the functions and the constraints of the system

2.2.2. Functional architecture

Functional architecture is the step which consists in dividing the system into different subsystems according to the functions it should carry out, however this step does not require knowing exactly the components which will be used to build the system yet.

As the primary function of the system requires knowing the specific needs of the user regarding the analyses he wants to perform, I will mainly focus on the second most important function of the system: moving autonomously from one point to another one.

In order to design the autonomous trimaran technical architecture I leant on the following FAST diagram which provides an overview of the different subsystems which comprise the main system:

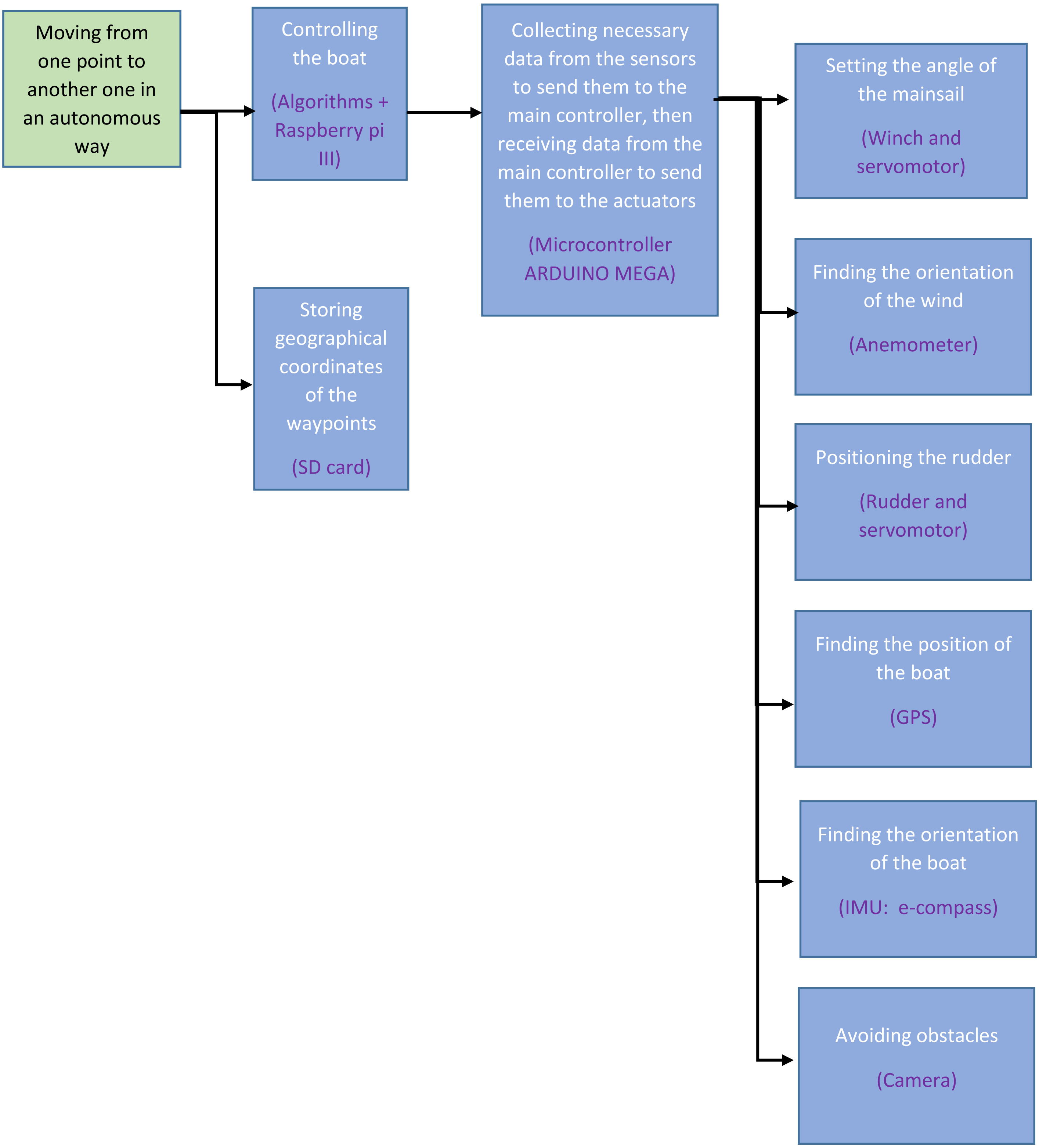

Moving from one point to another one in an autonomous way
Controlling the boat
(Algorithms + Raspberry pi III)
Storing geographical coordinates of the waypoints
(SD card)
Collecting necessary data from the sensors to send them to the main controller, then receiving data from the main controller to send them to the actuators
(Microcontroller ARDUINO MEGA)
Setting the angle of the mainsail
(Winch and servomotor)
Finding the orientation of the wind
(Anemometer)
Positioning the rudder
(Rudder and servomotor)
Finding the position of the boat
(GPS)
Finding the orientation of the boat
(IMU: e-compass)
Avoiding obstacles
(Camera)

Figure 5 FAST diagram

Thanks to this diagram, a flux diagram was introduced so as to highlight the interactions and the kind of flux between the different subsystems.

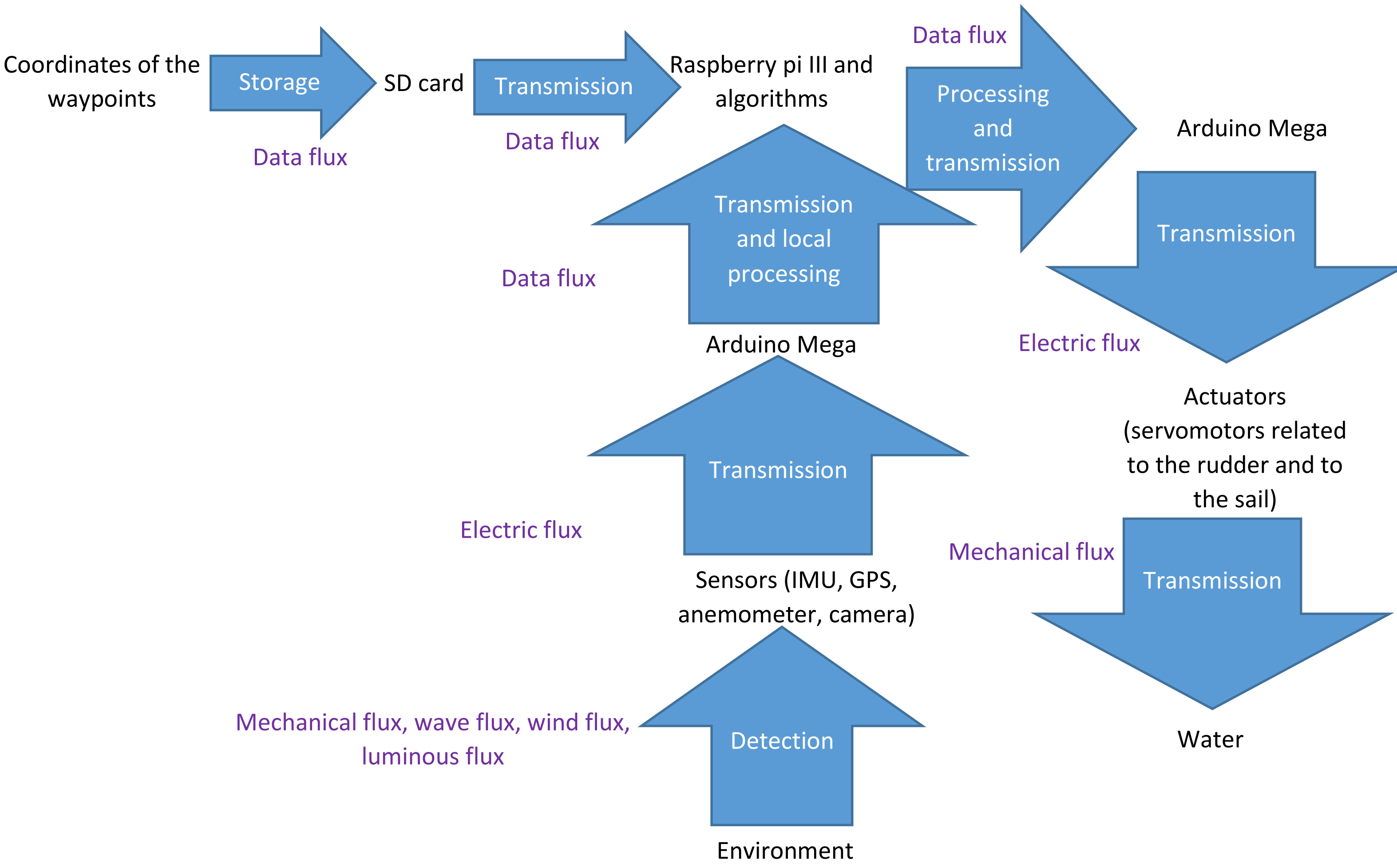

Coordinates of the waypoints
Storage
Data flux
SD card
Transmission
Data flux
Raspberry pi III and algorithms
Data flux
Processing and transmission
Arduino Mega
Transmission and local processing
Data flux
Transmission
Electric flux
Arduino Mega
Actuators (servomotors related to the rudder and to the sail)
Transmission
Electric flux
Mechanical flux
Transmission
Sensors (IMU, GPS, anemometer, camera)
Water
Mechanical flux, wave flux, wind flux, luminous flux
Detection
Environment

Figure 6 Flux diagram

2.3. Technical realisation

Now, as a functional architecture has been defined in the previous section, technical realisation can be tackled in order to build a prototype and to perform some tests so as to assess how it behaves in the sea.

2.3.1. Technical architecture

To build the prototype, a technical architecture was drawn up to summarise in concrete terms the plan of the different components which had to be used and the connections between each of them.

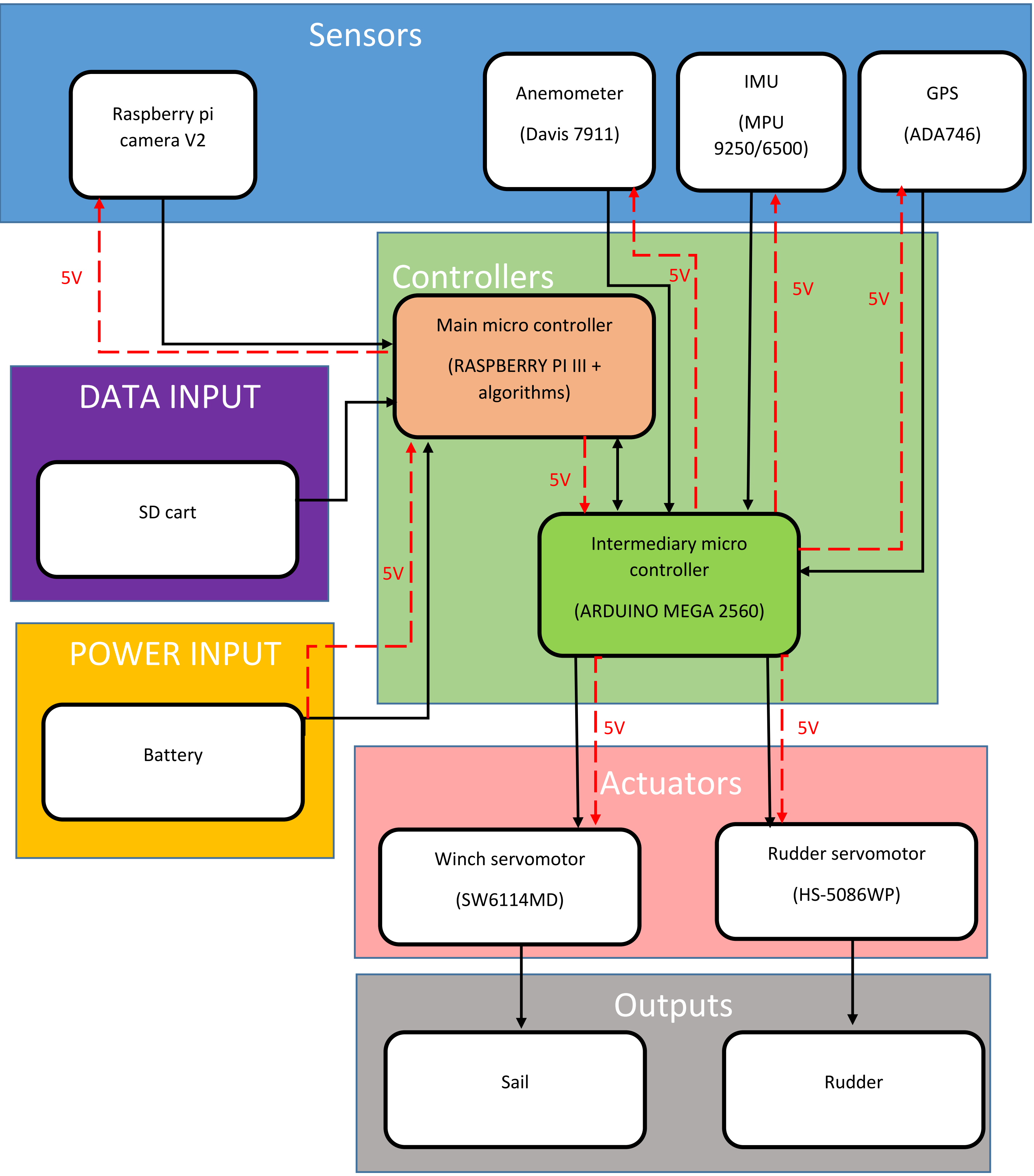

Sensors
Raspberry pi camera V2
Anemometer (Davis 7911)
IMU (MPU 9250/6500)
GPS (ADA746)
5V
Controllers
Main micro controller (RASPBERRY PI III + algorithms)
DATA INPUT
SD cart
Intermediary micro controller (ARDUINO MEGA 2560)
POWER INPUT
Battery
Actuators
Winch servomotor (SW6114MD)
Rudder servomotor (HS-5086WP)
Outputs
Sail
Rudder

Figure 7 Technical architecture

2.3.2. Boat structure

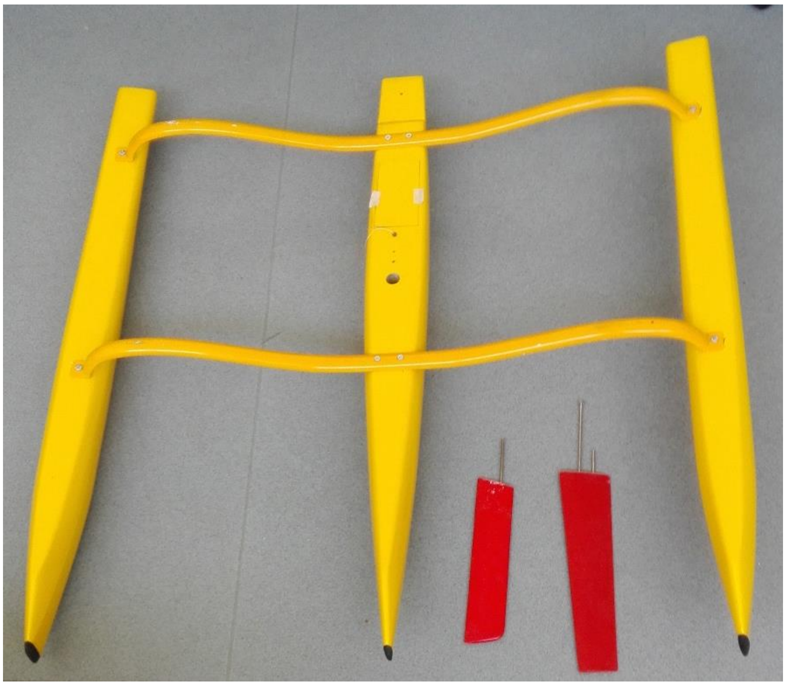


Figure 8 Boat's hulls + rudder + keel

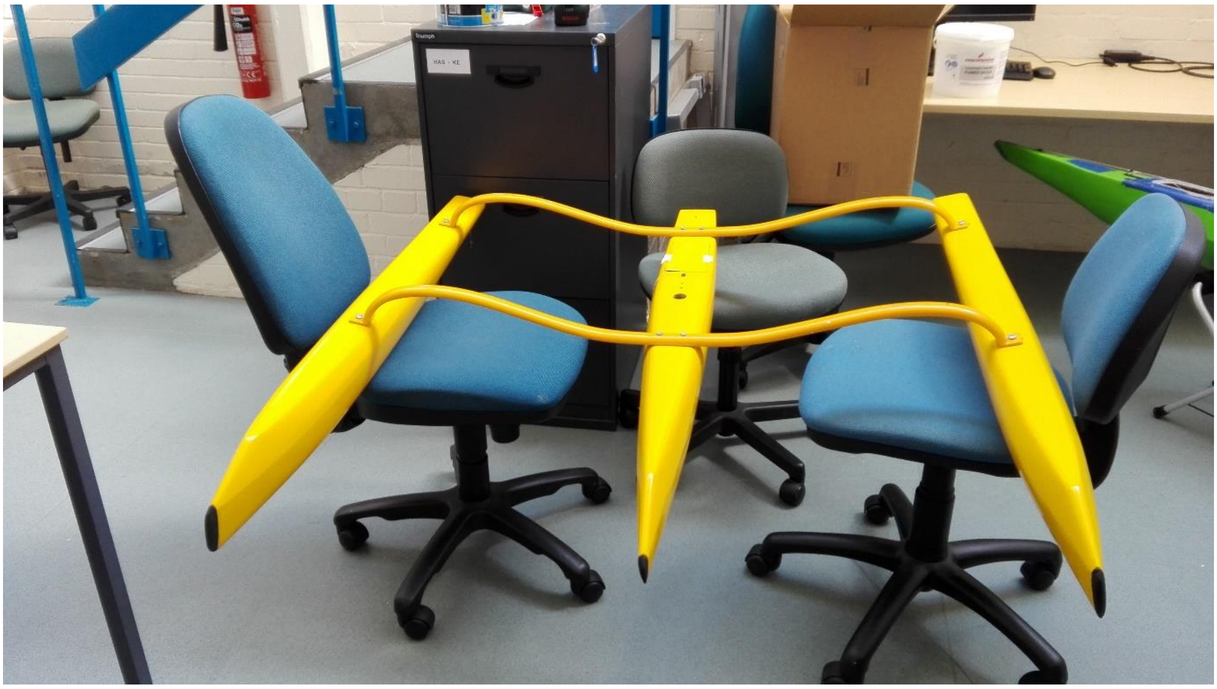


Figure 9 Notion of the dimensions

The boat structure comprises three hulls made of fibreglass and one mast made of carbon fibre. The hull in the middle is a little larger than the lateral hulls and it is hollow in order to contain all the hardware, the battery and the servomotors. There are also two sails, both made of rather tough fabric: the main sail and a smaller one. Two compartments were dug in the main hull so as to settle the electronic board and the servomotors inside.

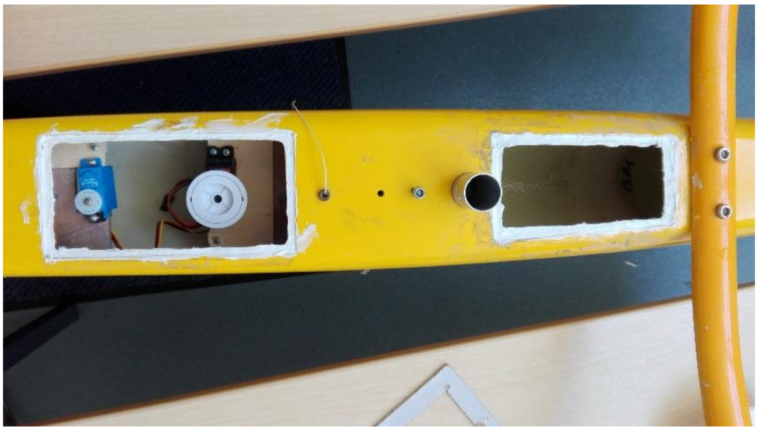


Figure 10 Compartments of the main hull

These two compartments were designed to be waterproof and accessible.

2.3.3. Hardware

In this section, I will describe the electronic components which have been used to make the trimaran work autonomously.

2.3.3.1 Sensors

The sensors allow the robot to perceive its environment (here: orientation of the wind, GPS position, orientation related to its environment, obstacle detection). They are essential for the boat to work autonomously since it needs a feedback from its environment so as to adjust the command which must be received by the actuators to reach a goal. If the boat only worked manually, it would not need such sensors. About exactness, on the one hand the accuracy of the sensors is important and must not be chosen randomly because the more accurate, the less mismeasurements. On the other hand, the accuracy of the sensors can considerably raise its price. Thus, choosing the accuracy of the sensors must depend on the use expected from the sensors. For instance, using a camera with high resolution would be necessary for long distance measurements but it would also be expansive whereas basic item recognition would only require cheap cameras.

GPS (ADA746)[2]

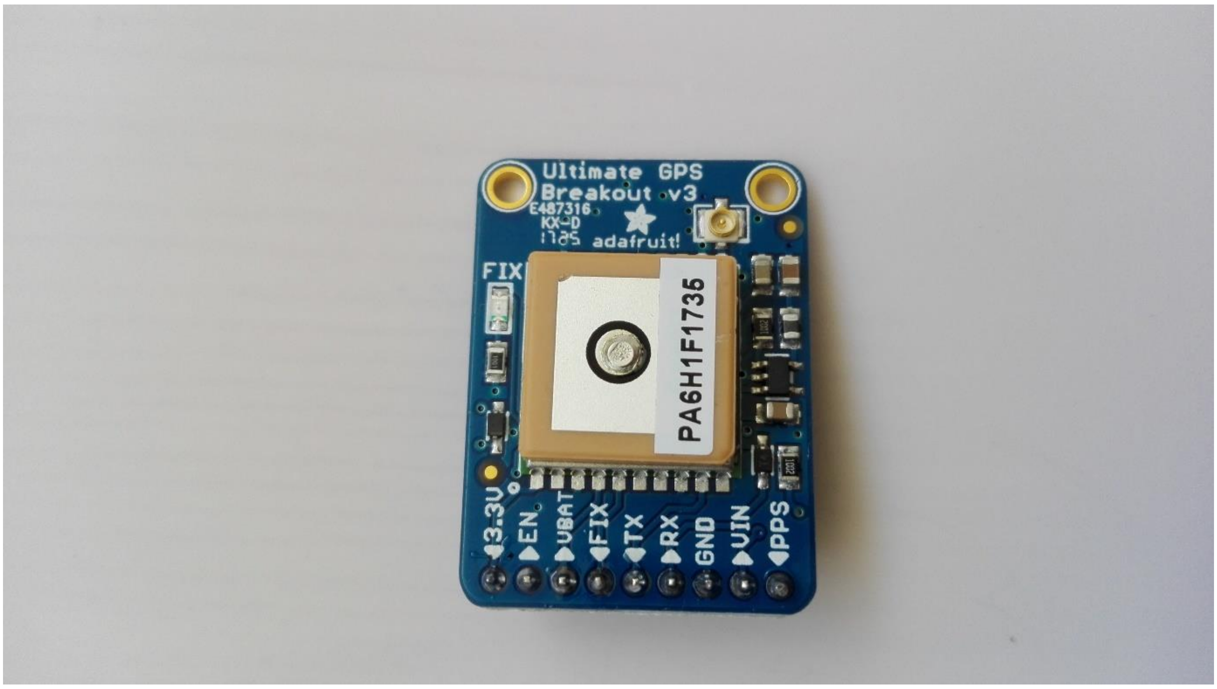


Figure 11 GPS (ADA746)

The GPS (Global Positioning System) aims at locating the robot on the sea. Data provided by the GPS and the coordinates of the waypoints must be compared in order to give new instructions to the boat so that it reaches the waypoints. This sensor can locate the robot with a +/- 3 meter-accuracy, which is sufficient for such a marine application. Data provided by the sensor are encrypted in the form of specific messages according to the NMEA (National Marine Electronics Association) 0A83 communication norm, such as GCA or RMC messages for instance. Such a message contains much unnecessary information for the system such as the number of satellites used to calculate the coordinates. Therefore, those kinds of messages need to be parsed so as to only provide the necessary information. With this aim in mind, an existing Arduino library has been used to receive data from the sensor and to parse it: “TinyGPSPlus”.

A second library has been used to turn the latitude and longitude data into the position of the boat in the local plane in the form of (X; Y). Such a conversion reduces the complexity of the upcoming computation, nevertheless working on the assumption that the robot stays within a 100 km delimited area during the mission so that the Earth is locally considered as flat [3].

This sensor communicates with the Arduino circuit board through the serial port. It can be powered by a 3.3 or 5.0 V voltage.

IMU (MPU 9250/6500)[4]

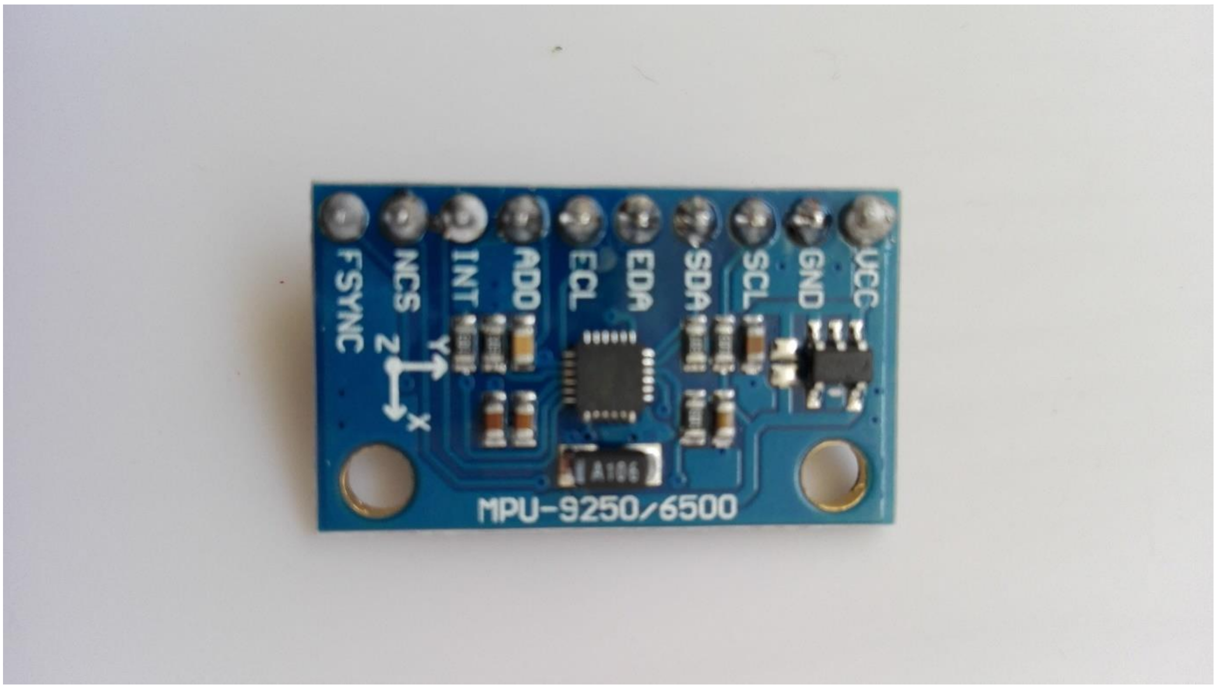


Figure 12 IMU (MPU 9250/6500)

The IMU (Inertial measurement unit) [5] is a sensor whose role is to measure a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers. Here, this sensor will be only a way to find the current heading of the boat in order to consequently adjust the yaw. To get the heading of the boat from the data provided by the gyroscopes and the accelerometers, the Arduino library "IMU" has been used because originally, gyroscopes and accelerometers are sensors which return angular and rectilinear accelerations but not directly the angle needed.

Roll angle determination from the IMU data have shown a Mean Absolute Error (MAE) [6] of 0.58◦ (trueness error: difference between the result and the "true" value) and a Root Mean Square Error (RMSE) of 0.77◦ (precision error: assesses whether the results are close to each other). That is to say, the measurements are on average the true value +/- 0.58° and consecutive measurements are on average 0.77° different.

Those two values allow to reckon an accuracy error of 0.97° (sqrt(MAE^2 + RMSE^2)), which means each measurement provided by the sensor is likely to belong to the set ["true" value – 0.97° ; "true" value + 0.97°]. This accuracy is sufficient for this marine application.

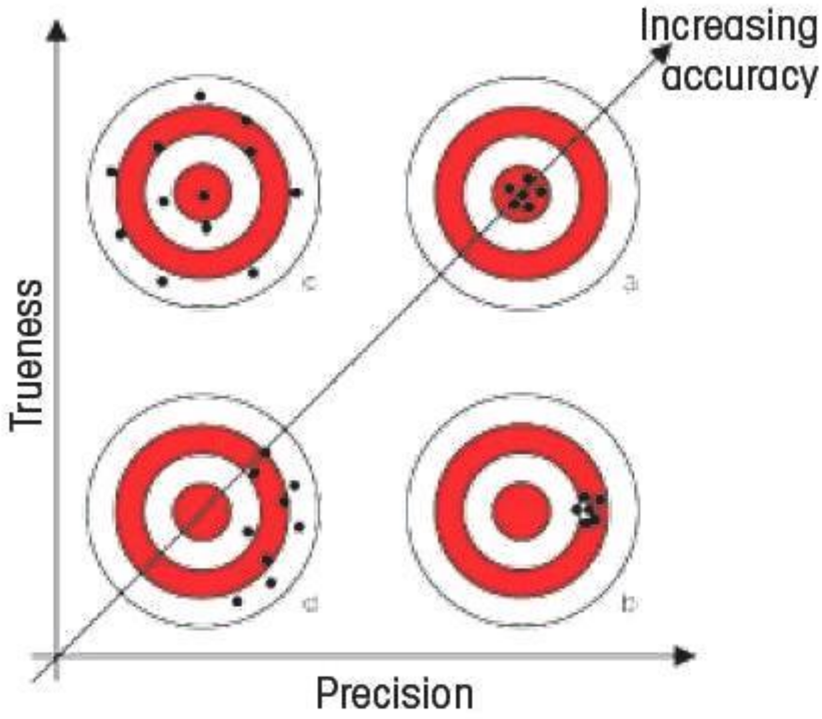


Figure 13 Trueness, precision and accuracy explanation [7]

This sensor communicates with the Arduino circuit board through the serial port. It can be powered by a 3.3 or 5.0 V voltage.

This IMU had been already ordered before my internship nevertheless, another IMU (Xsens MTi 1) [8] has been used instead of this one because this one did not have a compass, thus it could not return the heading of the boat.

Figure 14 IMU (Xsens MTi 1)

Anemometer (Davis 7911)[9]

Figure 15 Anemometer (Davis 7911)

This sensor aims at measuring the local speed and direction of the wind in relation to the boat. These data are processed so as to control the sail's inclination to power and guide the trimaran at best. A made-to-measure 3D piece designed with Solidworks, a CAD (Computer-Aided Design) system, was printed so as to attach the sensor at the top of the mast to provide the most reliable measurements.

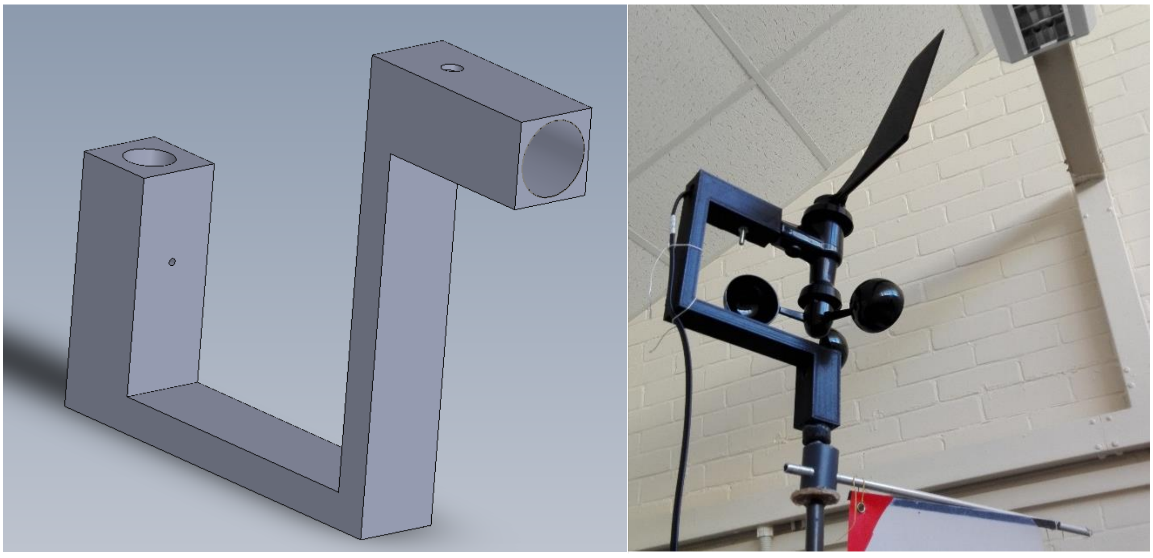


Figure 16 3D support for the mast

Raspberry pi camera V2 [10]

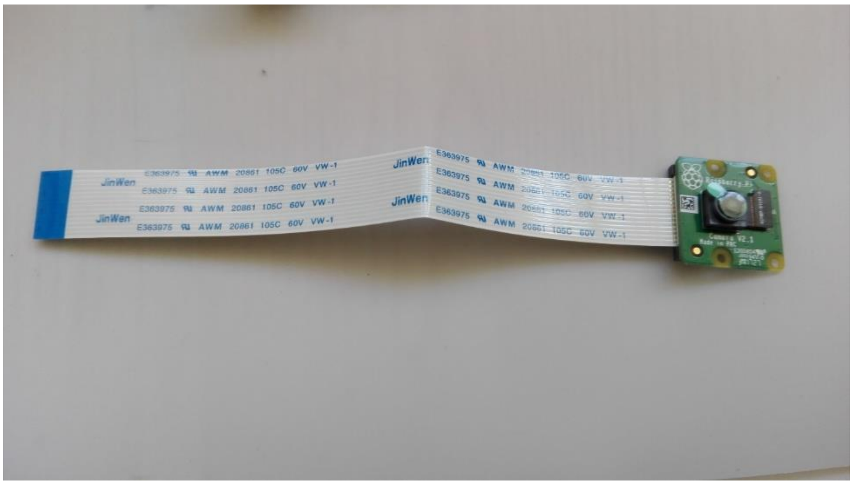


Figure 17 Raspberry pi camera V2

The Raspberry Pi Camera Module V2 is a digital camera module which has been used to detect obstacles so that the boat avoids them thanks to digital image processing. This sensor is directly compatible with the Raspberry pi III, that is to say unlike the other sensors connected to the Arduino circuit board, this sensor does not need the use of a library to work in itself and there is already a dedicated plug to connect the camera to the Raspberry pi board via a short ribbon cable. This sensor has an 8 mega pixel resolution and is able to take 3280 x 2464 pixel static images, 1080p30, 720p60 and 640x480p90 videos which is equivalent to mid-range phone features.

I designed and printed a 3D support so as to easily fix the camera to the boat.

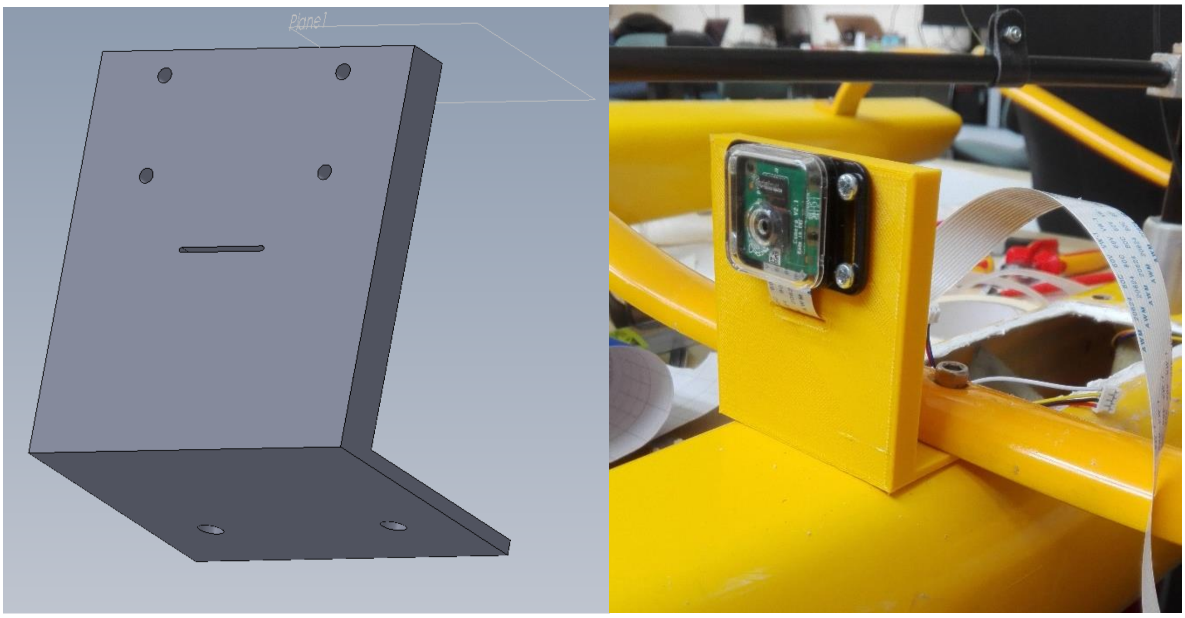


Figure 18 3D support for the camera

2.3.3.2 Actuators

The actuators embody the driving part of the system, they are responsible for the boat's motion. As regards the sailing robot, the rudder and the winch of the sail must be controlled by moving them more or less and in an accurate way. That is to say, the movement produced by the motor does not need to be uninterrupted and the system requires knowing the state of the motor all the time so as to produce an enslaved reply. The second interest of servomotors is the fact that they can resist an opposite effort which here is the stress of the wind on the sail and the stress of the water on the rudder.

Rudder servomotor (HS-5086WP)[11]

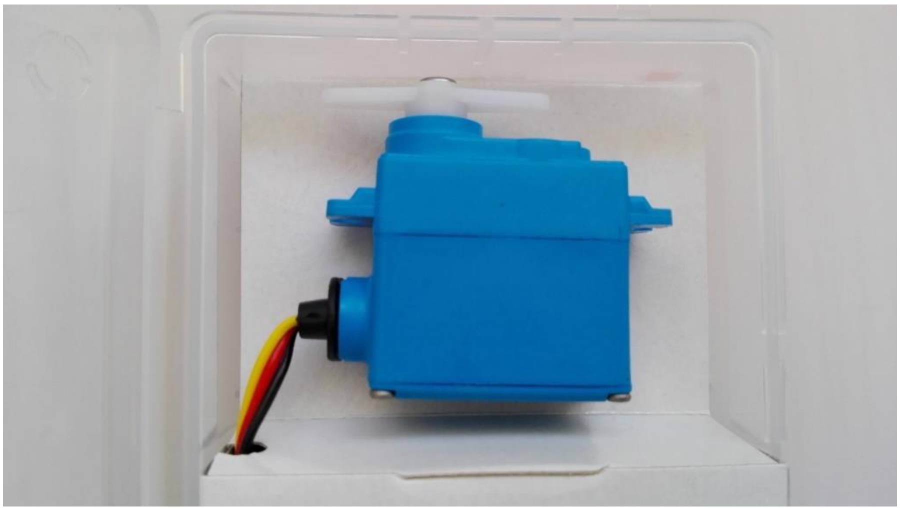


Figure 19 Rudder servomotor (HS-5086WP)

This servomotor aims at controlling the rudder of the boat. It is waterproof to avoid damaging itself if the boat capsizes.

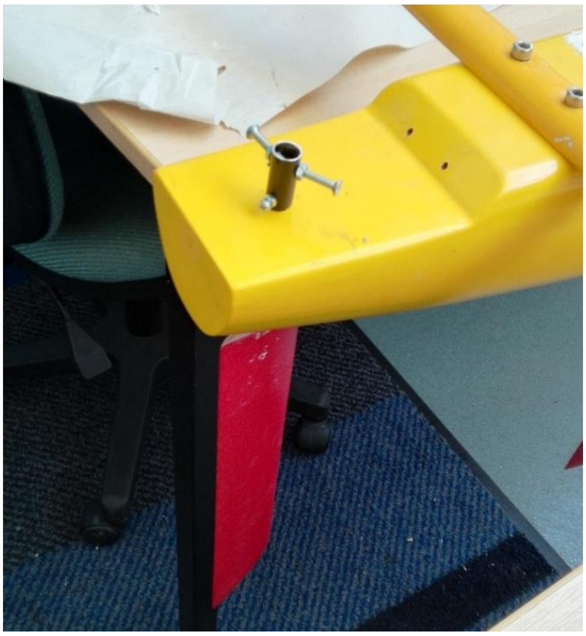


Figure 20 Rudder

Winch servomotor (SW6114MD)[12]

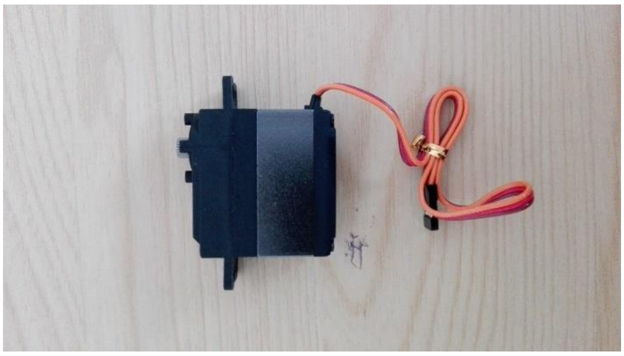


Figure 21 Winch servomotor (SW6114MD)

This servomotor aims at controlling the sail of the boat. Actually, this servomotor has been especially designed to stretch and relax the winch which is itself connected to the sail.

This component is located in the main hull of the boat to protect it from water. A tiny hole was drilled in the hull in order to let the sail rope link the servomotor and the sail.

2.3.3.3 Microcontrollers

Sensors and actuators mainly behave as energy converters: sensors usually turn energy from their environment like wind power, light energy etc. into electricity whereas actuators usually convert electricity into mechanical energy or light energy. Thanks to a local electronic circuit, energy is quantified and turned into data but it still needs something else to be processed, here is the role of microcontrollers. A microcontroller is like a small computer since it contains one or more processor cores, memory and electronic inputs to connect sensors and outputs to connect actuators. In this way, microcontrollers can process data, calculate and generate new variables to run algorithms.

They are also a way to power the actuators and the sensors which require energy to work. In this project, two microcontrollers have been used for different reasons. One microcontroller (Raspberry pi III) has been used as the main controller of the system as it is more powerful and convenient for programming. However, this controller is much more complicated than the other one (Arduino MEGA 2560) as it is a real microcomputer. The issue is that such kinds of microcontrollers are likely to face dysfunctional problems while they are running, however for safety issues and in order to avoid losing or damaging the prototype, a manual remote control was implemented and must keep working even if the main microcontroller does not work anymore. Therefore, to ensure such a function, the second controller which is more reliable must be able to handle manual remote control and then to control the actuators.

Last but not least, just extracting data from the sensors or sending data to the actuators consumes much power from the local microprocessor. Consequently in order to leave enough power to the main microcontroller devoted to calculate for the different algorithms which will be used, the second microcontroller (Arduino MEGA 2560) will be dedicated to collect data from the sensors and to control the actuators.

Raspberry pi III B- [13]

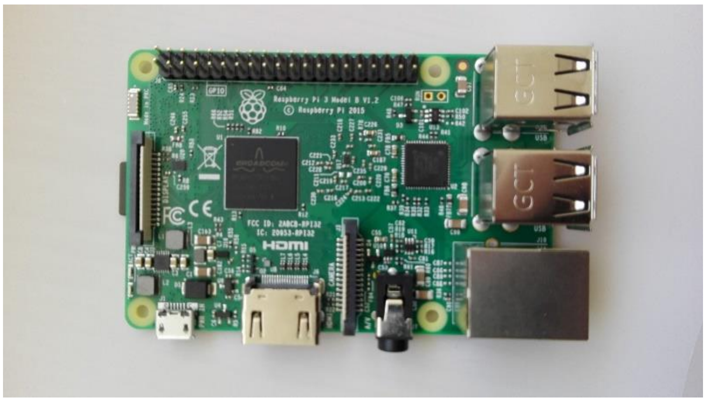


Figure 22 Raspberry pi III B-

This controller has been used as the main controller of the system. It aims at co-ordinating all the electronic components on the whole and it runs the different algorithms that the boat needs to work as expected.

This controller has a quad-core processor and the clock speed of each core is 1.2 GHz. That is to say, as each core is independent, the controller can carry out up to 4.8 billion instructions per second.

This device is equipped with a wireless router which allows to control it remotely at short range thanks to a computer.

The micro SD card slot has been used to store data and the algorithm scripts.

The HDMI and USB ports are very convenient as they allow to handle the circuit board as if it was a common computer by connecting a screen, a mouse and a keyboard.

Features

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot (now push-pull rather than push-push)
- VideoCore IV 3D graphics core

Power

- Can be powered via a +5.1V micro USB supply

Arduino MEGA 2560 [14]

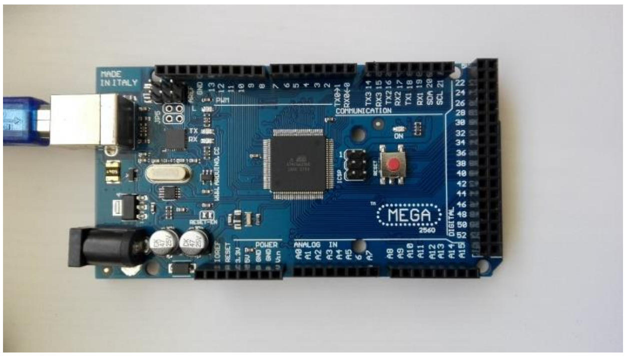


Figure 23 Arduino MEGA 2560

This controller has been used as a transitional interface between the Raspberry pi III and most of the sensors.

It is an open-source microcontroller compatible with the Arduino software (IDE) which was used to deliver scripts to the circuit board.

The power managing system of the circuit board can provide a 5V or a 3.3V power supply which matches the input voltage of the sensors and actuators previously mentioned.

The clock speed of this controller is 16 MHz, that is to say it can carry out up to 16 million instructions per second, which is less than the Raspberry pi III.

Features

- ATmega2560 Microcontroller (MCU)
- 5V operating voltage
- 7-12V input voltage (recommended)
- 6-20V input voltage (limit)
- 54 digital input/output (I/O) pins
  - 15 of which provide PWM output
- 16 analogue input pins
- 20mA DC current per I/O pin
- 50mA DC current for 3.3V pin
- 256KB Flash memory
  - 8KB used by bootloader
- 8KB SRAM
- 4KB EEPROM
- 16MHz clock speed
- 101.52mm x 53.3mm (Length x Width)
- 37g weight

Power

- Can be powered via USB connection or with external power supply
  - Power source selected automatically

2.3.4. Software

An operating system, software programs and different programming languages have been used to develop the project.

Operating system

Raspberry pi III was equipped with Linux as operating system. An Ubuntu image made by Ubiquity Robotics has been used as Linux distribution because it entails many advantages for projects in robotics. Indeed, not only comprises this distribution already many software programs which have been used for the project but also it includes its own WIFI hotspot, which allows to control the raspberry from another connected computer without using any additional screen.

Software programs

Two main software programs have been used to get the system working. Firstly, the Arduino software (IDE) has been used to deliver the Arduino scripts to the Arduino circuit board. This software program is the easiest way to quickly deliver such scripts and it allowed to perform some quick tests to assess the behaviour of the sensors.

Secondly, Robot Operating System (ROS), a robotics middleware has been used so as to connect the software of the different subsystems together. Indeed, in computer science, a middleware is a software which creates a data network between two or more software applications so as to make them exchanging data. The use of this software is very interesting since it allows to connect the subsystems together but keeping their own structure independent individually. That is to say, several subsystems can be developed individually and then quickly gathered thanks to the setting up of a basic link or "node". The subsystems are just expected to send data or "messages" with a certain type, for example "Float64" through the link so as to ensure data compatibility. Consequently, this software program is interesting to perform teamwork and to test each function of the system without necessary implementing all the functions yet. The ROS distribution release which has been used in the project is ROS Kinetic Kame.

Programming languages

In this project, different programming languages have been used: C++ (.cpp files), Python (.py files), Arduino language which is merely close to C/C++ (.ino files). Arduino language is necessary to implement scripts through an Arduino circuit board. Python is less efficient as regards calculating than C or C++, however it provides more intuitive tools and with more possibilities thus both languages have been used and it does not bring new compatibility issues since ROS processes data independently and can handle scripts written in C/C++ or python.

Algorithms

Different algorithms have been developed in this project, so as to make the boat autonomous. In this part, I will briefly explain the role of each of them and I will give more details about the algorithm for which I was responsible.

As detailed below thanks to the class diagram, each element of the software structure has been described thanks to object-oriented classes through Object-oriented programming. This kind of structure allows to easily make the connection with the hardware structure of the boat no matter how it is. Basically, this software structure can be easily applied for boats with one or several sails or with one or several hulls as well. Similarly, it does not depend on the sensors or actuators in themselves.

According to the class diagram, each sensor and actuator is represented by its own class. Basically, this allows to manage each of them easily thanks to Arduino libraries related to this electronic elements. The “Sailboat” class is the class which digitally embodies the boat, this class centralises data related to the sensors, transmits them to the “Controller” class and receives data from the “Controller” class dedicated to the actuators.

The “Controller” class aims at controlling the actuators in accordance with the data received from the sensors. This class contains all the algorithms devoted to make the boat autonomous. Four main algorithms have been implemented so as to give the boat four different control means.

- ## Go home mode :

  In this mode, the boat has to focus on the first GPS coordinates that it retained. The main goal of this mode is to get back the boat in case the Raspberry pi III does not work anymore.

- ## Waypoint follower mode :

  Thanks to this mode, the boat can autonomously reach a GPS position. In order to do this, the controller calculates the bearing between the current GPS point and the next waypoint and applies a basic P control on the heading.

- ## Line follower mode :

This mode aims at following a line between two waypoints. In order to do this, the controller calculates the closest distance to the line between two waypoints and tries to converge to the line [15].

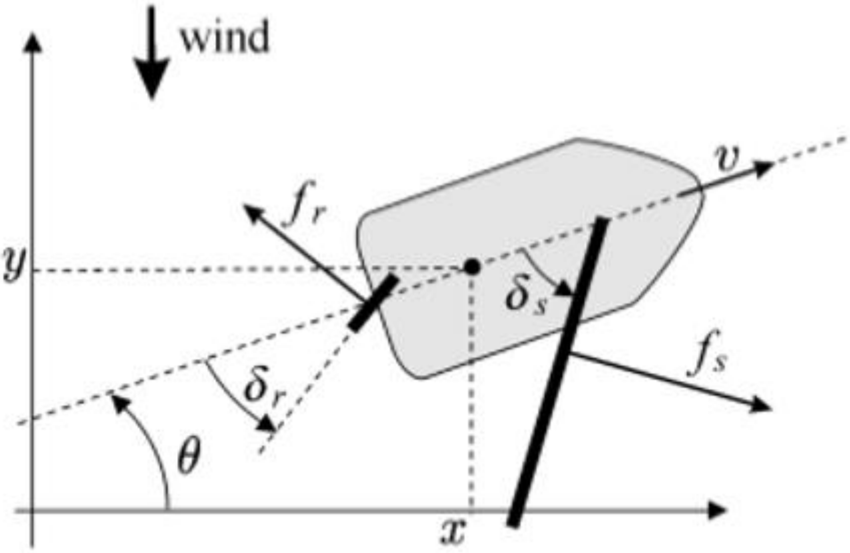
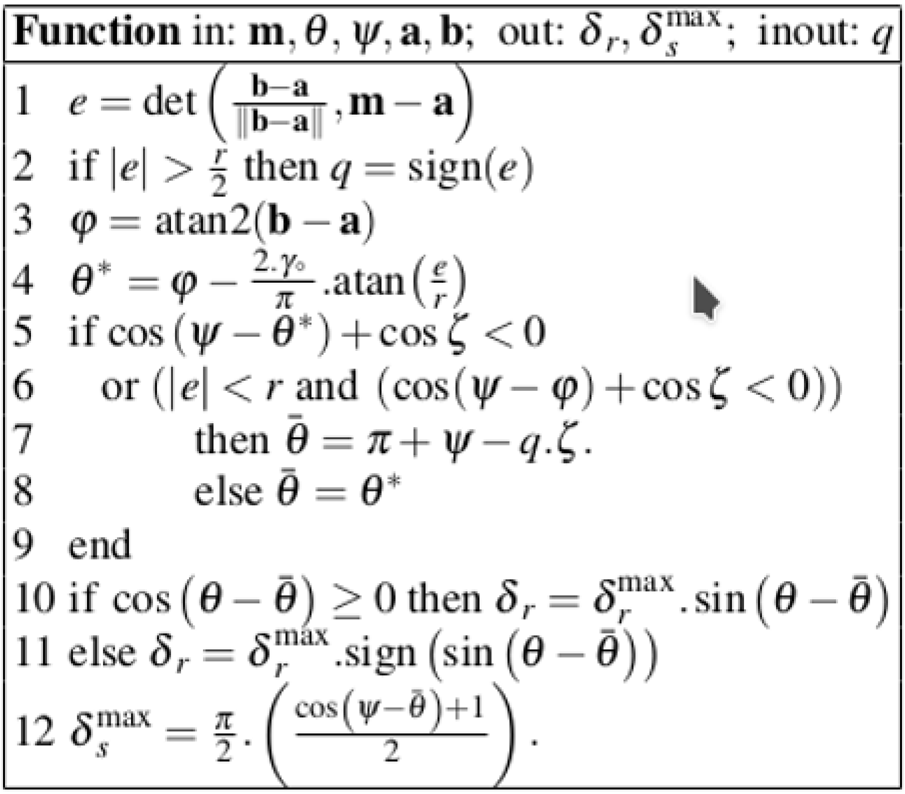


Figure 24 Drawing explaining the line follower algorithm



```
Function in: $\mathbf{m}, \theta, \psi, \mathbf{a}, \mathbf{b}$;  out: $\delta_r, \delta_s^{\max}$;  inout: $q$
$e = \det\left(\frac{\mathbf{b}-\mathbf{a}}{\|\mathbf{b}-\mathbf{a}\|}, \mathbf{m}-\mathbf{a}\right)$
if $|e| > \frac{r}{2}$ then $q = \mathrm{sign}(e)$
$\varphi = \mathrm{atan2}(\mathbf{b}-\mathbf{a})$
$\theta^* = \varphi - \frac{2.\gamma_\infty}{\pi}.\mathrm{atan}\left(\frac{e}{r}\right)$
if $\cos(\psi-\theta^*) + \cos\zeta < 0$
   or $(|e| < r$ and $(\cos(\psi-\varphi) + \cos\zeta < 0))$
         then $\bar{\theta} = \pi + \psi - q.\zeta$.
         else $\bar{\theta} = \theta^*$
end
if $\cos(\theta-\bar{\theta}) \geq 0$ then $\delta_r = \delta_r^{\max}.\sin(\theta-\bar{\theta})$
else $\delta_r = \delta_r^{\max}.\mathrm{sign}(\sin(\theta-\bar{\theta}))$
$\delta_s^{\max} = \frac{\pi}{2}.\left(\frac{\cos(\psi-\bar{\theta})+1}{2}\right)$.
```

Figure 25 Line follower algorithm

Heading of the boat: θ

Angle of the wind: ψ

Position of the boat measured by the GPS: m

Angle of the rudder: δr

Angle of the sail: δs

Maximum angle of the sail: δ_s^{max} ($|\delta s| \leq \delta_s^{max}$)

Hysteresis used for close hauled sailing: $q \in \{-1, 1\}$

Maximum rudder angle: δ_r^{max}

Cutting distance: r (we would like the distance to the line to be always smaller than r)

Close haul angle: ζ

Angle of the sail in crosswind: β

Details of the algorithm:

- 1 - The sign of "e" indicates the side of the line, on which we are: e < 0 i.e. right side and e > 0 i.e. left side.
- 2 - If this step is validated, the boat has to tack. "q" indicates the side of the tack.
- 3 - ϕ is the angle of the aim (line between "a" and "b"), in the geographical reference, so the angle is with respect to the East.
- 4 - Correction of the trajectory, with the attraction of the line.
- 5-7 - If this is activated, the command has to be corrected.
- 8 - No correction.
- 10 - Controlling the rudder: soft command of it.
- 11 - Bang bang command.
- 12 - Finding the exact command of the sail.

• Standby mode :

This mode will be used to manually control the boat thanks to a remote control.

Below, the software and node structure of the boat.

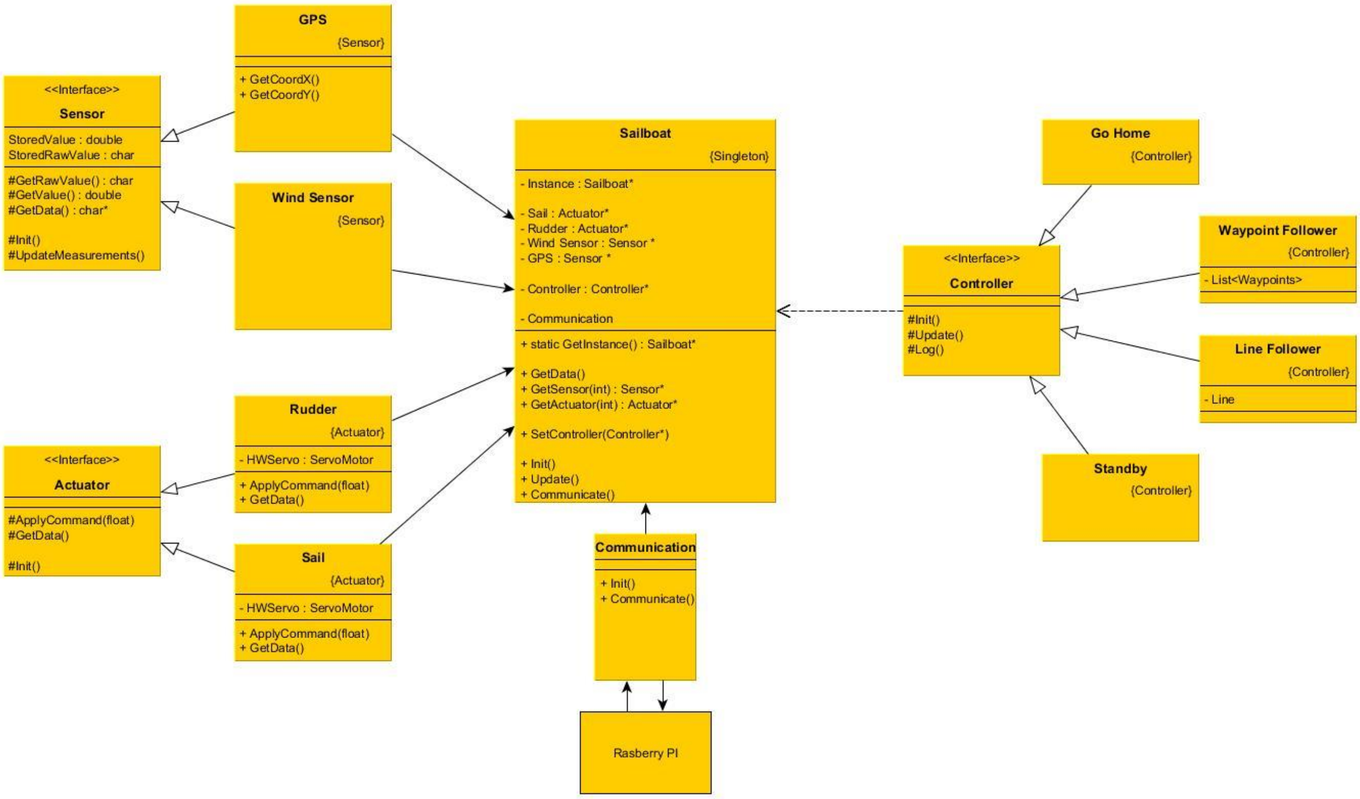


Figure 26 Class diagram

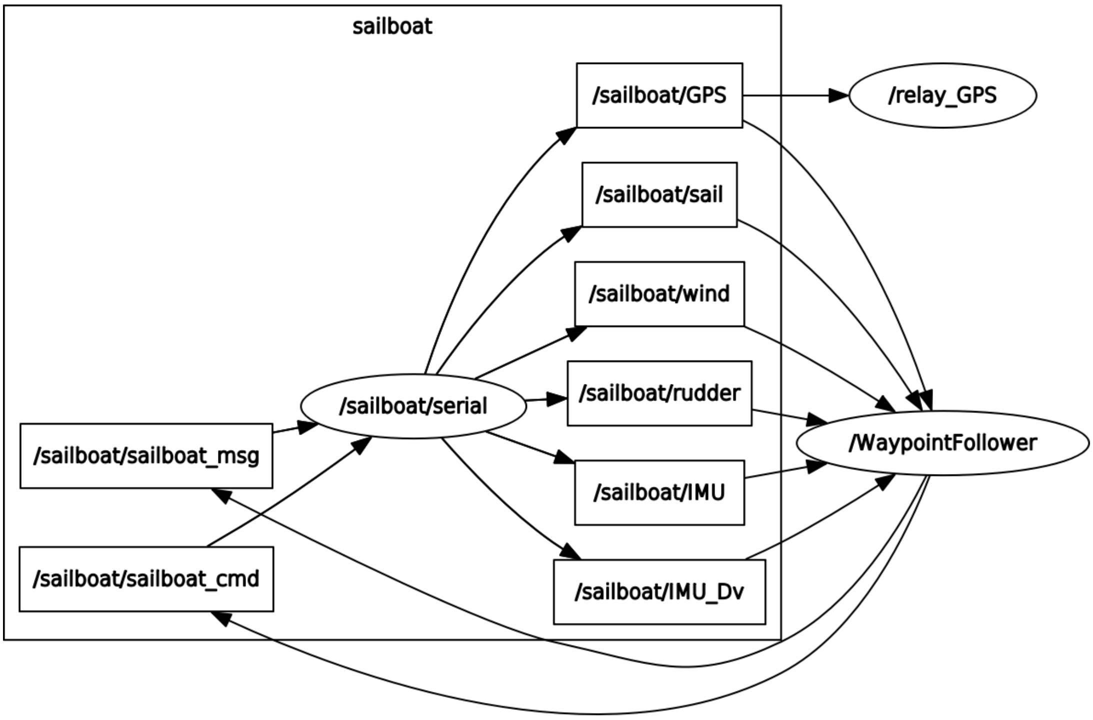


Figure 27 ROS RQT node diagram

Obstacle avoidance by digital image processing:

Not only is asked the boat to head autonomously for GPS positions but it is also required to avoid obstacles autonomously.

In order to run the boat in accordance with aims and obstacles, a potential field approach has been considered. Basically, this approach aims at describing the sailing area as a vector field in which each vector embodies the moving tendency of the boat if it has to behave in the local zone. According to this approach, the GPS positions which must be reached by the boat beget vectors tending to attract the boat whereas obstacles tend to repulse the boat. This approach is briefly explained by the drawing below in which the pink disc is the target and the green disc is the obstacle, for more details, see https://www.ensta-bretagne.fr/jaulin/ensi_isterobV2.pdf pages 114 & 115.

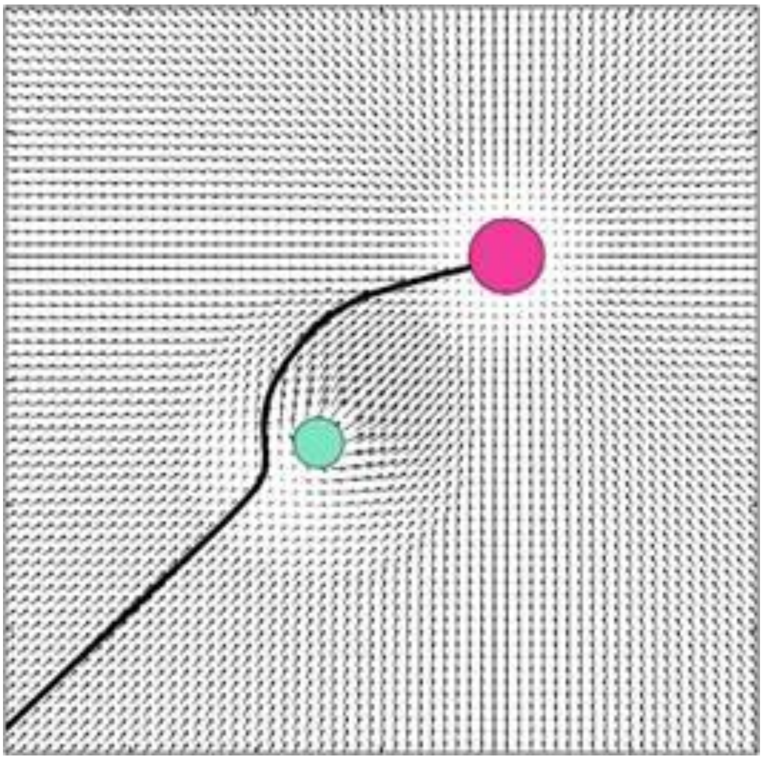


Figure 28 Potential field approach

One of my roles as regards the algorithmic part of this project was to implement an algorithm able to detect obstacles so that the potential field algorithm interprets them like that. I was asked to use a camera in order to detect obstacles, indeed for instance using ultrasonic sensors would not have been a relevant solution as the boat is supposed to detect obstacles on the surface and quite far from it (several tens meters) whereas ultrasounds are known to badly spread in the air.

Moreover, on the scale of the project, only buoys whose colour and size can be known beforehand were considered as obstacles because there is no easy way to measure the distance between only one stationary camera and an obstacle without knowing anything about the obstacle. I made use of ROS and CV2 library, compatible with the programming language Python in order to design my script and to link it with the main script of the boat.

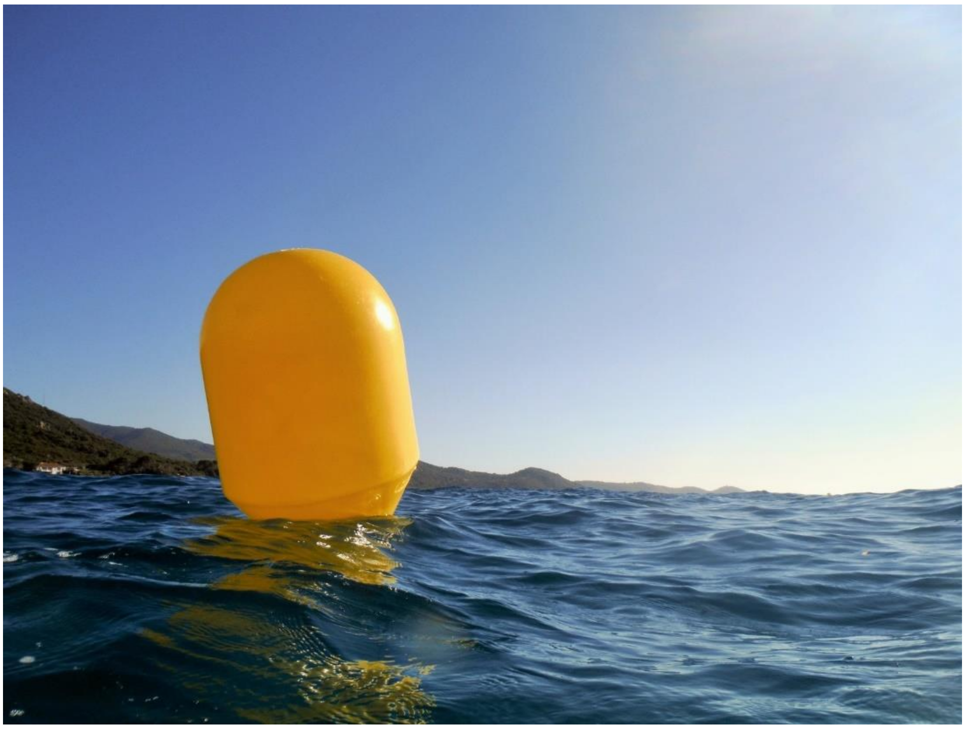


Figure 29 Kind of obstacles which must be detected by the camera

The main steps of the algorithm are explained by the diagram below:

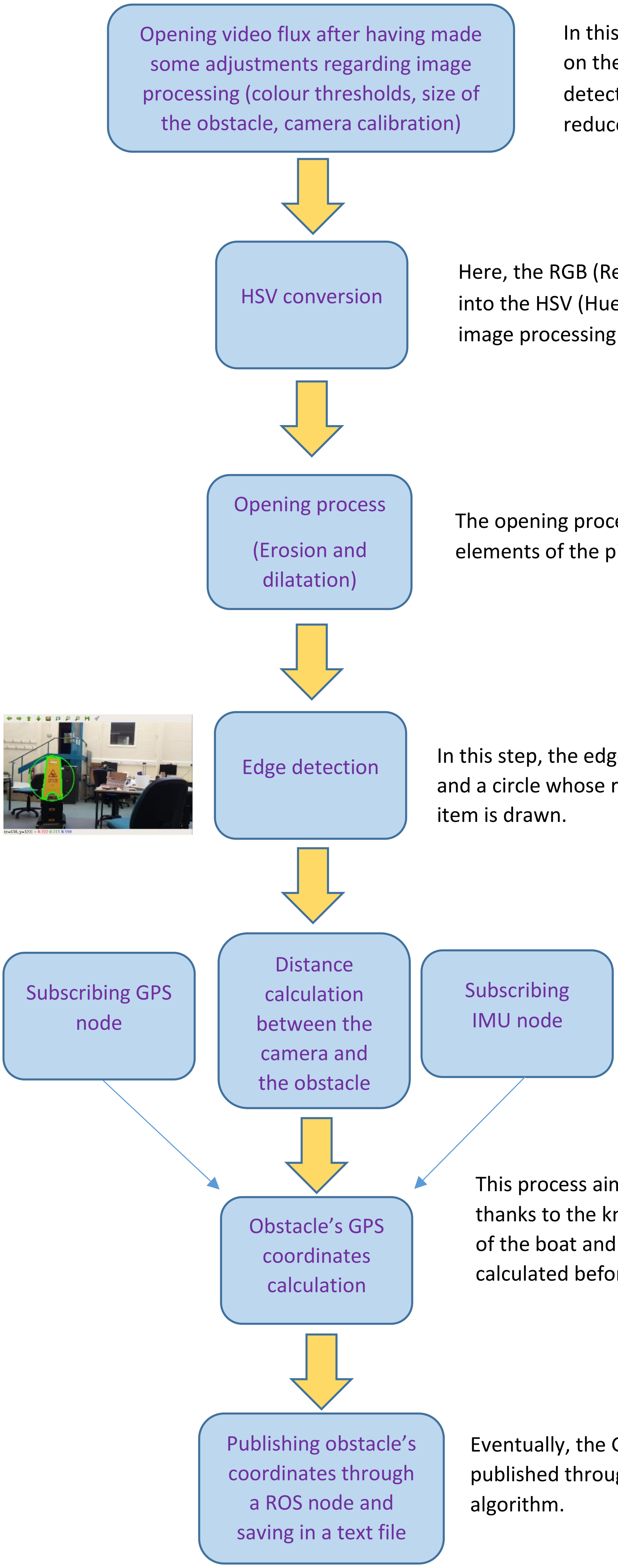

Opening video flux after having made some adjustments regarding image processing (colour thresholds, size of the obstacle, camera calibration)
HSV conversion
Opening process
(Erosion and dilatation)
Edge detection
Subscribing GPS node
Distance calculation between the camera and the obstacle
Subscribing IMU node
Obstacle's GPS coordinates calculation
Publishing obstacle's coordinates through a ROS node and saving in a text file

Figure 30 Diagram of the obstacle coordinates algorithm

In this step, the user has to set the parameters depending on the obstacles so as to allow and optimise their detection. The camera must also be calibrated [16] so as to reduce the distortion of the picture.

Here, the RGB (Red Green Blue) format of the picture is converted into the HSV (Hue Saturation Value) [17] format in order to make the image processing independent of the current lighting.

The opening process [18] aims at refining the edges of the different elements of the picture.

In this step, the edge of the item which must be detected is identified and a circle whose radius corresponds to the longest length of the item is drawn.

Here, the distance between the camera and the obstacle [19] is calculated thanks to the radius of the circle drawn and the knowledge of the parameters set before.

This process aims at calculating the GPS coordinates of the obstacle [20] thanks to the knowledge of the GPS position of the boat, the heading of the boat and the distance between the boat and the obstacle calculated before.

Eventually, the GPS position of the obstacle is written in a text file and published through a ROS node so as to be used by the potential field algorithm.

2.3.5. Interfaces between subsystems

To connect the Raspberry pi III, the Arduino circuit board, the sensors and the actuators together, an intermediary circuit board was designed and produced by a specialised company. Different electronic components were soldered to make it operational.

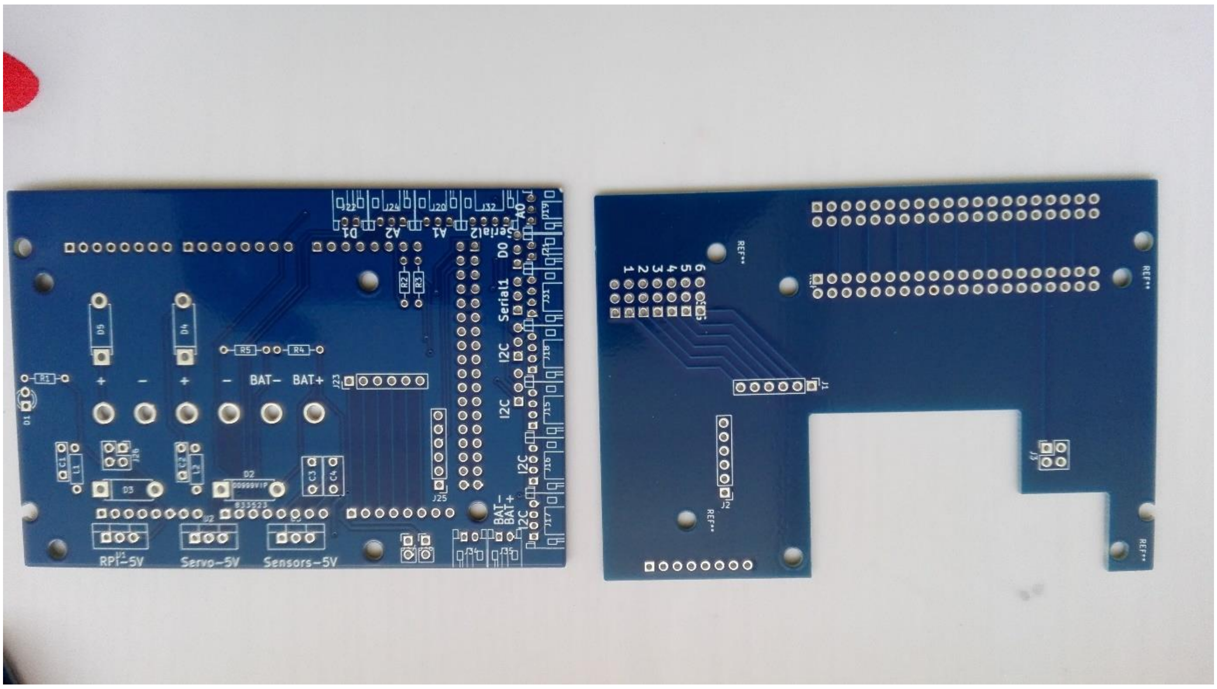


Figure 31 Printed circuit boards connecting Raspberry pi III, Arduino MEGA, sensors and actuators

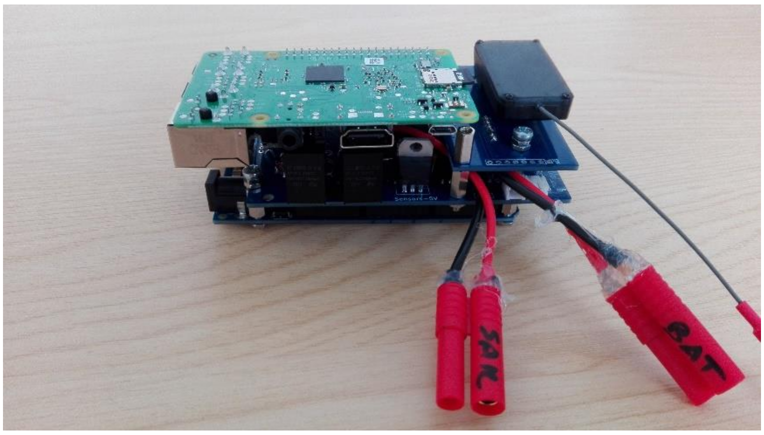


Figure 32 Circuit board entirely assembled

2.3.6. Miscellaneous

Remote control: Hobby King 2.4Ghz 4Ch Tx & Rx V2 [21]

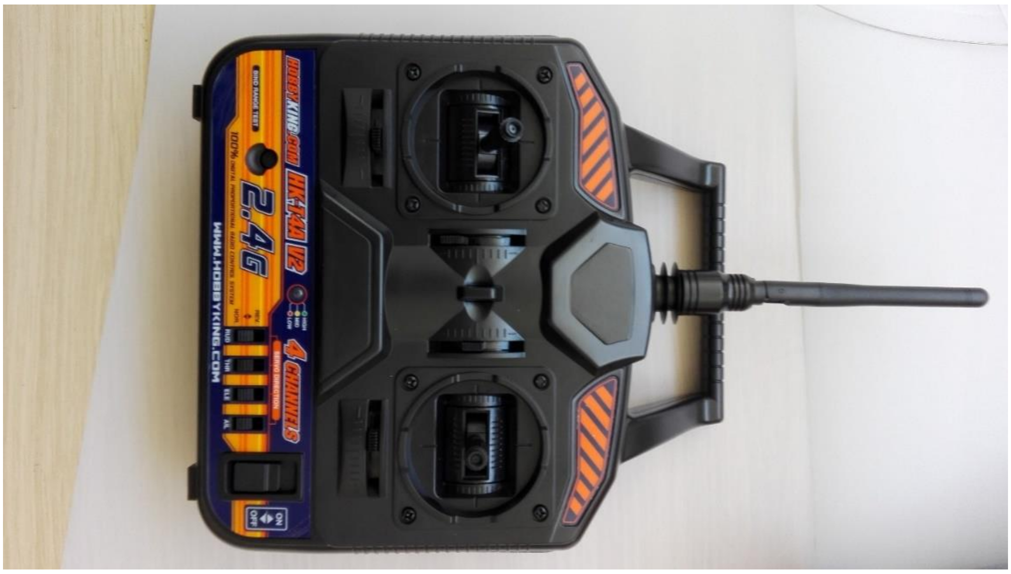


Figure 33 Remote control: Hobby King 2.4Ghz 4Ch Tx & Rx V2

This remote control is dedicated to be used to manually control the boat, in case of emergency, for example if the autonomous mode does not work anymore.

Radio receiver: FlySky FS-R6B 2.4Ghz 6CH RC AFHDS FS R6B Receiver [22]

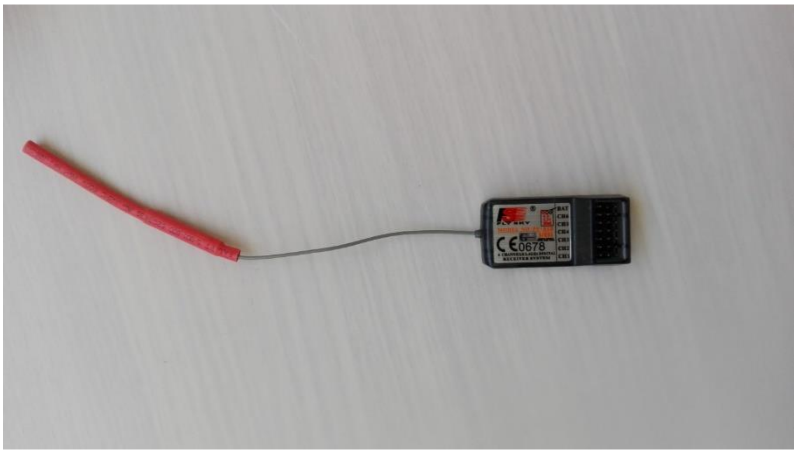


Figure 34 Radio receiver: FlySky FS-R6B 2.4Ghz 6CH RC AFHDS FS R6B Receiver

This device is used to allow the boat to receive instructions from the remote control.

3. Project management

3.1. Schedule and organisation

One of the main features of this project was independent work. That is to say, to carry out this project, I had to make my own timetable. To be as efficient as possible, I had mainly planned two different kinds of tasks at the same time since many tasks required to request specialised technicians but they were not in the laboratory anytime so to keep working when I needed to wait for them, I did another task simultaneously. Overall, the table below summarises my schedule during this project.

TASK	BEGINNING	END
Assembly of the hulls	12/06	14/06
Preparation and software installation related to Raspberry pi III circuit board	13/06	19/06
Getting familiar with the existing software and algorithms	18/06	28/06
Programming the algorithm dedicated to obstacle avoidance through digital image processing	28/06	13/07
Arranging the sails and the anemometer on the mast	13/07	27/07
Soldering components on the circuit boards	16/07	18/07
Designing and printing the support for the anemometer	19/07	23/07
Designing the waterproof compartments to store electronic components in the main hull	19/07	06/08
Modifying the mast for some issues with the anemometer	02/08	14/08
Inserting electronic components in the main hull	06/08	09/08
Designing the support of the camera	10/08	13/08
Testing image processing algorithm	14/08	17/08

3.2. Teamwork

A large part of this project was carried out as a teamwork as we were five people working on four different boats but all the boats overall shared the same features, that is to say if someone tackled a task for his boat, this work done could be reused for the other boats. In this way, the different tasks were shared to work efficiently and we shared programs and technical skills so as to help each other. Everyone used the same software structure as Ulysse Vautier and Christophe Viel had already implemented the software architecture for their boat before I come to Plymouth. This architecture can be found through the GitHub link https://plymouth-sailboat.github.io

Teams	Boat assigned
Ulysse Vautier and Christophe Viel	Small and large single-hull sailing boats
Charlie Goutorbe and Evann Clavier	Catamaran
Alexandre Houdeville	Trimaran

4. Tests and results

Image processing algorithm test:

In order to test my image processing algorithm, I used the SSH (Secure Shell) [23] [24] network protocol. This protocol allowed me to control the Raspberry pi from a laptop and then to conveniently analyse the results as I needed to perform outside experimentations with the Raspberry but I did not have screen to watch the camera window. As explained before in this report, the algorithm dedicated to get the coordinates of the obstacle thanks to the camera is divided in two main parts.

The first part aims at finding out the distance between the obstacle and the camera. To try out this part, I tried to measure the distance between the yellow sign that you can see below and the camera after having adapted the colour and the size parameters to the sign in the algorithm. I chose this item because it was the only item whose colour was very different from the colour of the other things there were in the room. After having gotten a sequence of different measurements for a constant distance, I deduced that the algorithm could measure a 5-meter distance between the camera and the obstacle with the following accuracy: +/- 20 cm, which is satisfactory.

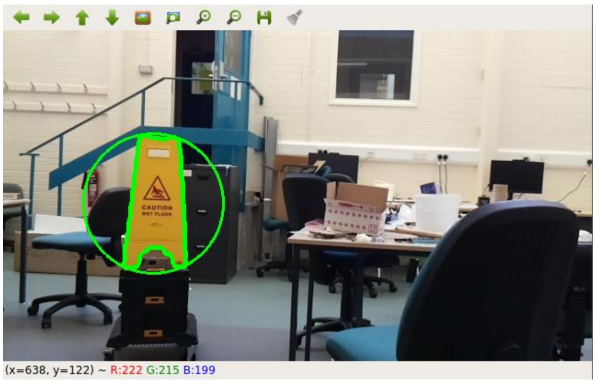
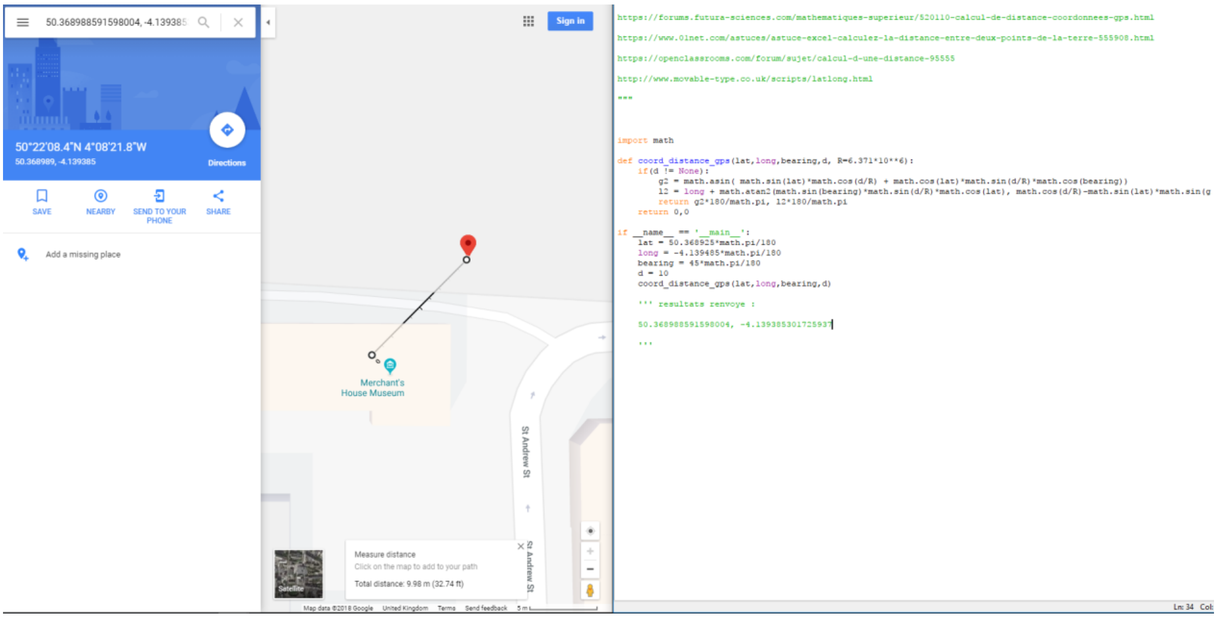


Figure 35 Test: distance measurement with the camera

The second part of the algorithm aims at calculating and writing in a text file the coordinates of the obstacle after having obtained the distance between the camera and the obstacle. To test this part independently, I considered I had the GPS coordinates and the heading of the boat and the distance between the boat and the obstacle. These simulated data allowed me to get a result from the algorithm. Then thanks to Google Maps, I drew the GPS position of the boat and the position of the obstacle that the algorithm returned. Finally, I tried to measure the distance between the two points and the inclination related to the line between them to make a correspondence between the simulated data (distance and heading). The results have been satisfactory as well.



Measure distance

Click on the map to add to your path

Total distance: 9.98 m (32.74 ft)

```
https://forums.futura-sciences.com/mathematiques-superieur/520110-calcul-de-distance-coordonnees-gps.html

https://www.01net.com/astuces/astuce-excel-calculez-la-distance-entre-deux-points-de-la-terre-555908.html

https://openclassrooms.com/forum/sujet/calcul-d-une-distance-95555

http://www.movable-type.co.uk/scripts/latlong.html

"""



import math

def coord_distance_gps(lat,long,bearing,d, R=6.371*10**6):
    if(d != None):
        g2 = math.asin( math.sin(lat)*math.cos(d/R) + math.cos(lat)*math.sin(d/R)*math.cos(bearing))
        l2 = long + math.atan2(math.sin(bearing)*math.sin(d/R)*math.cos(lat), math.cos(d/R)-math.sin(lat)*math.sin(g
        return g2*180/math.pi, l2*180/math.pi
    return 0,0

if __name__ == '__main__':
    lat = 50.368925*math.pi/180
    long = -4.139485*math.pi/180
    bearing = 45*math.pi/180
    d = 10
    coord_distance_gps(lat,long,bearing,d)

    ''' resultats renvoye :

    50.368988591598004, -4.139385301725937

    '''
```

Figure 36 Obstacle avoidance test: GPS conversion from camera distance

Finally, both parts of the algorithm have been inserted into the main software and I made sure that the whole algorithm could return consistent results by testing the algorithm outside with the yellow sign. Overall, I drew the conclusion that the algorithm could locate an obstacle on average 10 meters far from the boat with a 1-meter accuracy.

Global test:

I did not have to test the other software parts of the boat because they had already been tested by Ulysse and Christophe before I come to Plymouth, nevertheless I had to test the structure of the boat in the water and whether the different sensors and actuators had been correctly implemented on the boat. With this aim in mind, Jian Wan drove me to Roadford lake, a lake not too far from Plymouth so as to try out the boat. During this trial period I faced many problems, that is why finally I have not tested the boat, indeed firstly someone who was with us during the trial period fell on the boat while we were loading it in the van, thus the main hull was damaged but we finally could temporarily fix it with adhesive tape.

Moreover, there was slack in the articulations between the mast and the anemometer as well as between the mast and the main hull whereas the mast is not allowed to spin so as to keep the anemometer well-oriented, consequently I spent much time trying to stick these elements together with adhesive tape. Besides, when I tried to insert the circuit board into the hull of the boat, a weld between the winch servomotor and the circuit board came apart and I did not have the means to fix it on the spot. Finally, I could only have claimed that the floatability was good, however I could not have tried the different dynamic modes of the boat since we did not have time to try it before I leave.

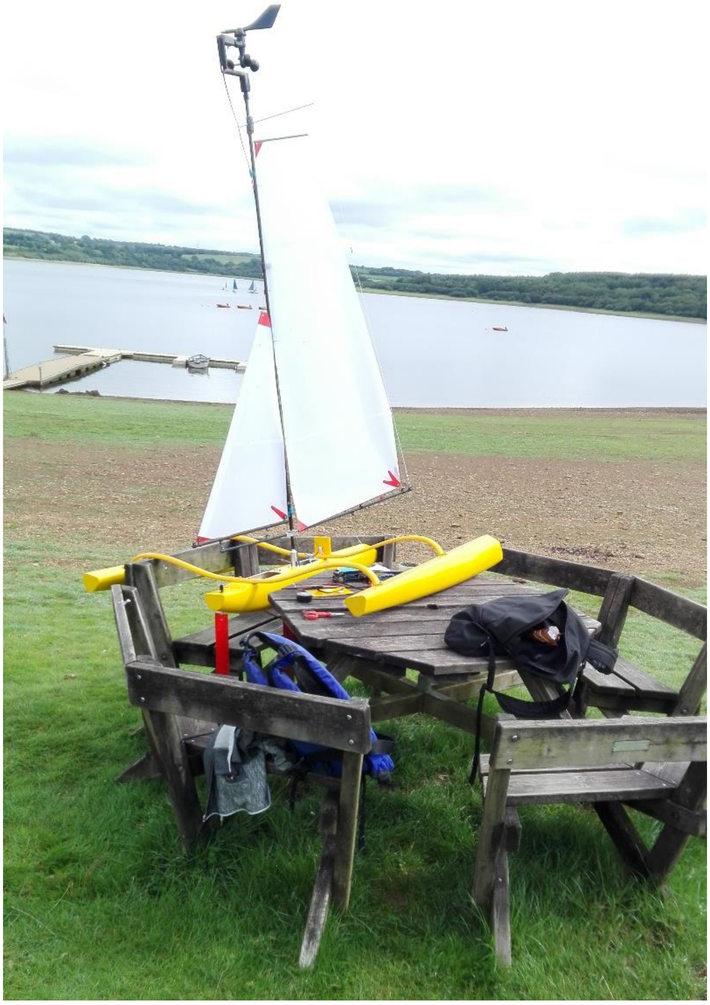


Figure 37 Roadford Lake

Eventually, before I leave Plymouth, Jian Wan invited me to the World Robotics Sailing Championship (WRSC)[25]. This is a robotic competition which consists in developing an autonomous sailing boat and improving the way to sail with it. I came there to support Ulysse Vautier and Christophe Viel's boat and to be aware of the stakes of such a competition.

Conclusion

Eventually, the initial goal of the project has been partially completed indeed the boat has been successfully designed with the different algorithms, sensors and actuators required, nevertheless, despite the fact that the obstacle avoidance algorithm have been tested, the boat could not have been tried out with the other algorithms running, therefore the efficiency of the boat is for now still unknown. Such problems faced during the trial period have shown that the way to carry out the tests could have been greatly improved. For instance, on the one hand the trial period was at the end of the internship thus if I had faced some problems with the algorithms, I would not have had time to fix them for another trial period anyway. On the other hand, the boat was not completely ready before starting the trial period because the obstacle avoidance algorithm was not working yet and I faced some troubles with the mast fixing. Furthermore, I could not use electricity to solder the broken weld during the trial period thus I had better have checked the reliability of the welds before going to the lake.

Overall, this 3-month internship was very enriching since I could improve in relation to many different fields. On the one hand, working on the hardware and on the boat structure allowed me to feel more confident with the use of tools that I was not used to handling. For instance, I have learnt how to solder and put up a seal. As regards the software part of the project, it allowed me to learn more about ROS, C language and Github. I have also understood why an Object-oriented programming approach is almost essential in such a project. Working autonomously was also one of the main features of this internship, thus I have learnt how to organise my timetable by myself and to coordinate tasks so as to avoid wasting time.

On the other hand, this period spent abroad was not only rewarding by the internship in itself. I also indeed made the most of the local culture by visiting local important places or spending my spare time with British people. Actually, I was living in a shared house with only British people and foreigners mainly coming from Spain, thus I nearly always spoke English or Spanish in my house. Now I feel fluent in English as well as in Spanish thanks to the time spent with my housemates and I know much more about the culture of other countries as other foreigners that I have met there made many comparisons between the United Kingdom and their own country.

To conclude, it has been the real first time I worked on such an engineering project. In this way, I have now a clearer opinion about what is being an engineer in robotics. Furthermore, working in a research laboratory was for me an opportunity to know more about engineering research as I had not had the chance before this, to have a look at what is research related to engineering. My technical skills have also improved much thanks to this project. Eventually, I have also really enjoyed carrying out this project abroad since it has provided much positive experience for me, therefore I have planned to do my next internship abroad, probably in a Spanish-speaking country.

List of figures

Bibliography

[1] Plymouth University website https://www.plymouth.ac.uk (Visited on 13/06/2018)

[2] Adafruit. Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates - Version 3 https://www.adafruit.com/product/746 (Visited on 22/06/2018)

[3] Luc JAULIN. Mobile robotics. Pages 127 & 128. 2016. https://www.ensta-bretagne.fr/jaulin/ensi_isterobV2.pdf (Visited on 22/06/2018)

[4] THAOYU ELECTRONICS. 9-DOF IMU Module With MPU-9250. http://www.hotmcu.com/9dof-imu-module-with-mpu9250-p-172.html (Visited on 22/06/2018)

[5] Wikipedia. Inertial measurement unit. https://en.wikipedia.org/wiki/Inertial_measurement_unit (Visited on 22/06/2018)

[6] Benoit ZERR. CAPTEURS ET SYSTEMES DE MESURE. 2017. https://moodle.ensta-bretagne.fr/pluginfile.php/44259/mod_resource/content/10/compound_ing1_2017_2018.pdf (Visited on 22/06/2018)

[7] AZO Materials. Thermal Analysis – Precision, Trueness, Accuracy and Errors. Jul 4 2011. https://www.azom.com/article.aspx?ArticleID=5744 (Visited on 22/06/2018)

[8] Mouser electronics. Xsens MTi 1-Series Development Kit https://eu.mouser.com/new/xsens/xsens-mti-1/ (Visited on 22/06/2018)

[9] DX Engineering. Davis Instruments Anemometers 7911. https://www.dxengineering.com/parts/dvi-7911 (Visited on 23/06/2018)

[10] www.raspberrypi.org . CAMERA MODULE V2. https://www.raspberrypi.org/products/camera-module-v2/ (Visited on 23/06/2018)

[11] Hitec Multiplex. HS-5086WP Metal Gear, Micro Digital Waterproof Servo. http://hitecrcd.com/products/servos/waterproof-servos-2/hs-5086wp-digital-waterproof-micro-servo/product (Visited on 23/06/2018)

[12] King Max. SW6114MD--61g 14kg.cm,digital,metal gears sail winch servo. http://www.kingmaxhobby.com/pd.jsp?id=71 (Visited on 23/06/2018)

[13] www.raspberrypi.org . RASPBERRY PI 3 MODEL B. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ (Visited on 15/06/2018)

[14] Mouser Electronics. Arduino Mega 2560 Microcontroller Board. https://www.mouser.fr/new/arduino/arduino-mega2560/ (Visited on 17/06/2018)

[15] Luc JAULIN. Mobile robotics. Pages 85 - 94. 2016. https://www.ensta-bretagne.fr/jaulin/ensi_isterobV2.pdf (Visited on 22/06/2018)

[16] docs.opencv.org . Camera Calibration. https://docs.opencv.org/3.1.0/dc/dbb/tutorial_py_calibration.html (Visited on 11/07/2018)

[17] Wikipedia. HSL and HSV. https://en.wikipedia.org/wiki/HSL_and_HSV (Visited on 29/06/2018)

[18] Wikipedia. Opening (morphology). https://en.wikipedia.org/wiki/Opening_(morphology) (Visited on 03/07/2018)

[19] Taha EMARA. Emaraic. Real-time Distance Measurement Using Single Image. 2018-03-11. http://emaraic.com/blog/distance-measurement (Visited on 29/06/2018)

[20] Movable Type Scripts. Calculate distance, bearing and more between Latitude/Longitude points. http://www.movable-type.co.uk/scripts/latlong.html (Visited on 29/06/2018)

[21] HobbyKing. HobbyKing 2.4Ghz 4Ch Tx & Rx V2 (Mode 2). https://hobbyking.com/en_us/hobby-king-2-4ghz-4ch-tx-rx-v2-mode-2.html (Visited on 21/07/2018)

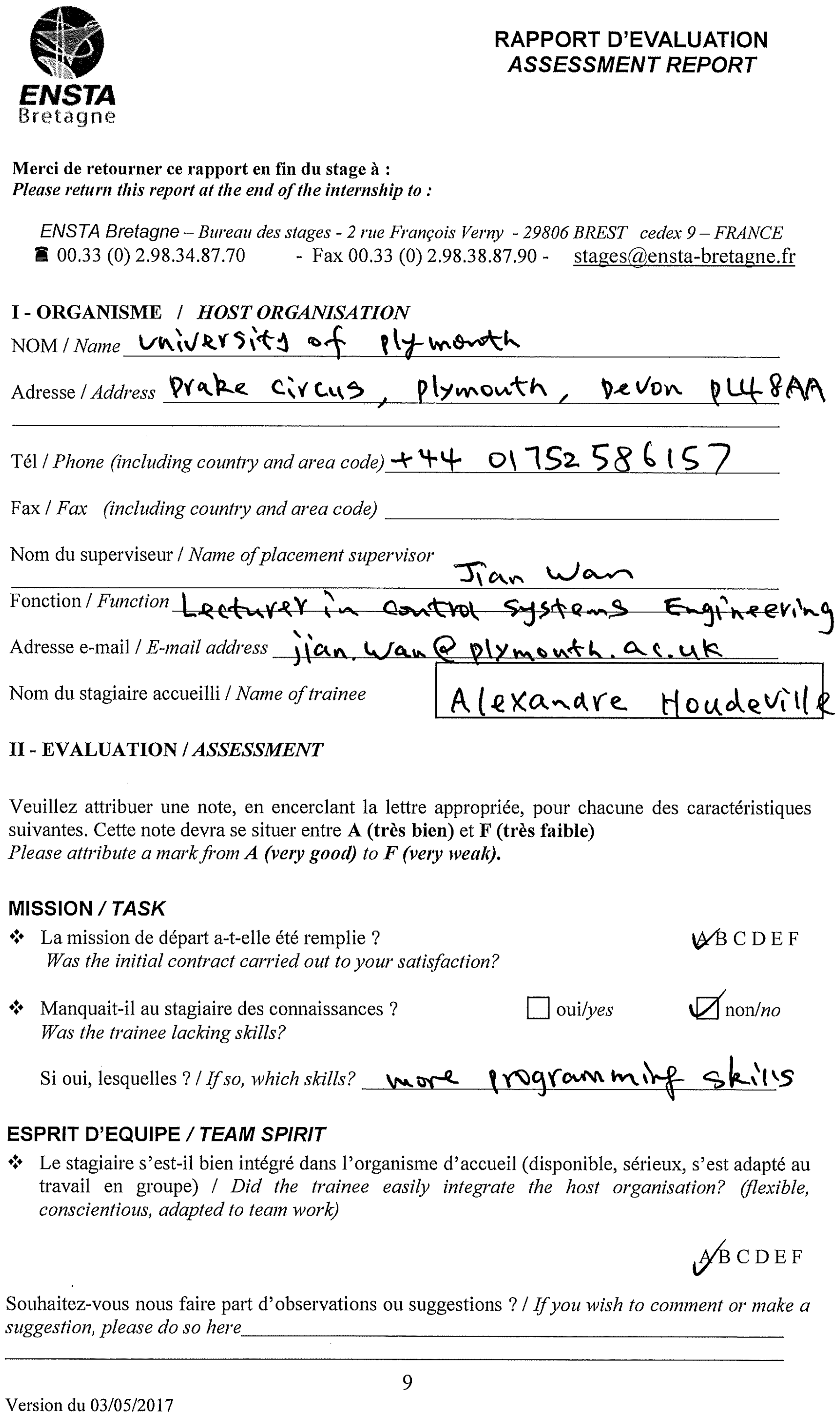
[22] ALIEXPRESS. FlySky FS-R6B 2.4Ghz 6CH RC AFHDS FS R6B Receiver for i6 i10 T6 CT6B TH9x Transmitter Remote Control Parts. https://www.aliexpress.com/item/FlySky-FS-R6B-2-4Ghz-6CH-RC-AFHDS-FS-R6B-Receiver-for-i6-i10-T6-CT6B/32669652306.html (Visited on 21/07/2018)

[23] www.raspberrypi.org . SSH (SECURE SHELL). https://www.raspberrypi.org/documentation/remote-access/ssh/ (Visited on 14/08/2018)

[24] Wikipedia. Secure Shell. https://en.wikipedia.org/wiki/Secure_Shell (Visited on 14/08/2018)

[25] WRSC. World robotic sailing championship. https://www.wrsc2017.com (Visited on 22/08/2018)

Annexe 1: assessment report



ENSTA Bretagne

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport en fin du stage à :
Please return this report at the end of the internship to :

ENSTA Bretagne – *Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE*
☎ 00.33 (0) 2.98.34.87.70 - Fax 00.33 (0) 2.98.38.87.90 - stages@ensta-bretagne.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name* university of plymouth

Adresse / *Address* Drake circus, plymouth, Devon PL4 8AA

Tél / *Phone (including country and area code)* +44 01752 586157

Fax / *Fax (including country and area code)* ______

Nom du superviseur / *Name of placement supervisor* Jian Wan

Fonction / *Function* Lecturer in control systems Engineering

Adresse e-mail / *E-mail address* jian.wan@plymouth.ac.uk

Nom du stagiaire accueilli / *Name of trainee* Alexandre Houdeville

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*** *(very good) to* ***F*** *(very weak).*

MISSION / *TASK*

- La mission de départ a-t-elle été remplie ?
  Was the initial contract carried out to your satisfaction? [✓ A] B C D E F

- Manquait-il au stagiaire des connaissances ? ☐ oui/*yes* ☑ non/*no*
  Was the trainee lacking skills?

  Si oui, lesquelles ? / *If so, which skills?* more programming 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 trainee 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* ______

9

Version du 03/05/2017

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) ?
Did the trainee 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* Can be more focused and better time management / organization

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 trainee 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 trainee 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
  Evaluate the technical skills of the trainee:

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

- Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?
  Would you be willing to host another trainee next year? ☑ oui/*yes* □ non/*no*

Fait à ______________________, le
In Plymouth, on 23/08/2018

Signature Entreprise ______________________ Signature stagiaire
Company stamp ______________________ *Trainee's signature*

School of Engineering
University of Plymouth
Drake Circus
Plymouth
Devon
PL4 8AA

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