Rapport de Stage

ROB23

Laurent Droudun

Acknowledgements

Firstly, I would like to thank my internship tutor Jian Wan, Lecturer in Mechatronics and Robotics at Aston University, for his welcome and his time. He was always listening, willing to help and did his best to answer all the questions we could have.

Then I would like to express my sincere gratitude to my teacher in robotics, Professor Luc Jaulin, researcher at ENSTA Bretagne, for giving us the opportunity to get in contact with Mr. Jian Wan.

Also, thanks a lot to my colleagues and friends Augustin Morge, Virgile Pelle and Mathieu Drillon with whom I worked with pleasure during this internship.

Résumé

Ce document présente l'avancée de mon travail au cours de mon stage de deuxième année d'école d'ingénieur de 4 mois à Birmingham dans l'université d'Aston. Le sujet de ce stage était le contrôle d'un voilier autonome et était encadré par Mr. Jian Wan. Le recours à des voiliers autonomes présente beaucoup d'avantages, notamment d'un point de vue énergétique grâce à son fonctionnement basé sur la force du vent.

Ainsi le travail s'est décomposé en plusieurs parties. Dans un premier temps, il fallait prendre en main les capteurs et s'assurer d'acquérir toutes les données nécessaires au bon fonctionnement de nos programmes. Un bon réglage d'un filtre était alors nécessaire. Ensuite, une approche plus haut niveau était alors possible. Une bonne visualisation du comportement du voilier en temps réel permet d'ajuster certains paramètres avant de mettre le bateau à l'eau.

Abstract

This document presents the progress of my work during my 4 month second year engineering school internship in Birmingham at Aston University. The subject of this internship was the control of an autonomous sailing vessel and was supervised by Mr. Jian Wan. The use of autonomous sailboats has many advantages, particularly from an energy point of view thanks to its operation based on the force of the wind.

Thus the work was divided into several parts. Firstly, we had to get to grips with the sensors and ensure that we acquired all the data necessary for our programmes to work properly. A good setting of a filter was then necessary. Then, a higher level approach was possible. A good visualisation of the boat's behaviour in real time allows us to adjust certain parameters before putting the boat in the water.

Contents

List of Figures

Introduction

Before the industrial revolution, ships were mostly using sails to transport people and goods. With the arrival of this revolution, a growing interest about Unmanned Surface Vehicles (USV) has occured. In the maritime field, autonomy has already affected the military field, the oil and gas sector, as well as the scientific field, where maritime drones are already used for surveillance and oceanographic data collection. Technology will make it possible to replace human intervention by the crew, hence the use of the term "smart shipping". The ship has its own intelligence, enabling it to analyse the environment in which it is moving thanks to a series of sensors, to make its own decisions and to implement the actions that the on-board system will have determined on the basis of a number of factors. Because sailboats are less polluting and energy-consuming (as they are wind-driven) than steamboats, they're highly worthwile knowing the ecological crisis facing the world today. Those USVs can be remotely controlled or self-directing thanks to algorithm making it autonomous.

Controlling autonomous sailboat was the objective of this engineering internship at Aston University in Birmingham. Indeed, it aims to familiarize with various sensors such as IMU, windvane, camera, GPS receiver, LiDAR and others for sensor data collection, filtering and fusion. Thanks to this, our sailboat : the Regazza would be able to follow a line and to stay at a given position upstream. This document relates the work accomplished during those 4 months under the supervision of Jian Wan, Lecturer in Mechatronics and Robotics.

1. Sailboat

1. 1. General description

A sailboat is composed by 4 main parts :

- A hull : The hull represents all the elements and structures of the boat. It consists of a part above the waterline and a part underwater. It aims to optimise the boat's performance and ensure buoyancy and waterproofing.
- A keel : The keel contains the ballast which helps to increase and maintain the stability of the boat, counteracting the forces exerted on the sails: the more ballast there is, the more stable the boat will be, but also the heavier and slower.
- A sail (linked to a mast) : It allows the boat to move forward thanks to wind forces. Hence, depending on the wind direction and the desired heading of the boat, it will move relatively quickly. Its rotation must be between -$\pi/2$ and $\pi/2$.
- A rudder : It controls the heading of the boat. It is supposed to be able to rotate of an angle by $\pi/2$ whatever the direction of the rotation but actually, due to mechanical impossibility, it can only rotate by approximately $\pi/4$ on both direction.

Figure 1: Sailboat nammed 'Regazza' and the different main parts numbered

1. 2. Points of sail

As said before, the sailboat does not move forward with the help of a motor, but thanks to its sail for which the wind is the fuel. That's why the direction of the wind will have a huge impact on its speed, and its heading. Indeed, as the range of rotation of the sail is π, the key is to orientate the sail perpendicular to the wind direction. Here arise various more or less problematic situations which are described on the figure after.

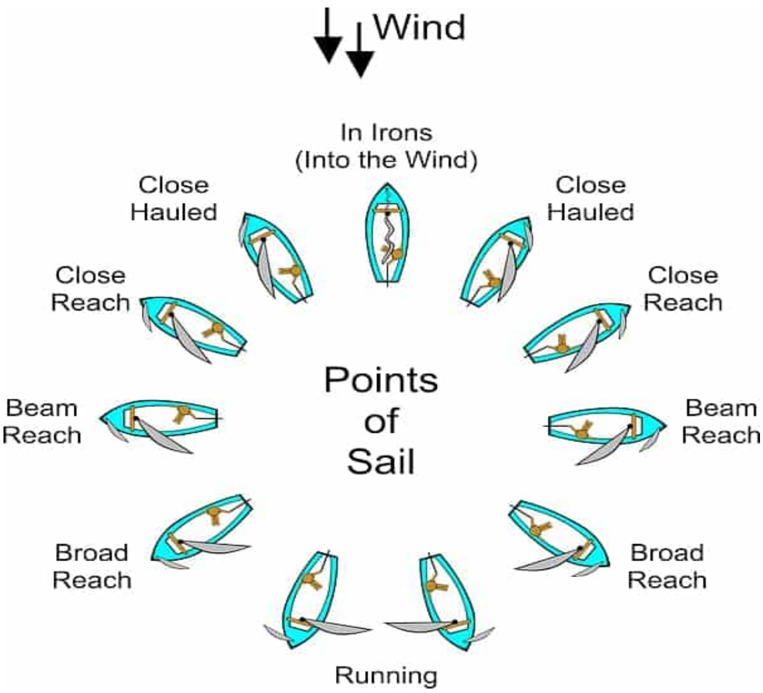

Figure 2: The 6 points of sail.

- **Running**
  That is the situation where the sail acts a lot like a parachute. Indeed, the boat is sailing downwind so the force of the wind is directly propelling the sailboat forward.

- **Broad reach, Beam reach, Close Reach and Close Hauled**
  Those situations do not vary that much. Obviously, it will have an impact on some caracteristics of the boat like its speed and its heading, but it is possible to navigate on these zones. The close hauled zone is the limit between navigable areas and the "no-go zone".

- **Into the wind** or **no-go zone**
  It corresponds to the most complex situation. To better optimise the speed of the boat, you need the wind to come through the inside of the sail. In contrast, when going upwind, it can not be possible. The no-go zone is the range of directions into which a sailboat can not sail. The solution for this is to sail close hauled in one way until the boat is too far from the line to follow (the radius is defined by the user) and then sail close hauled in the other way as described by the figure.

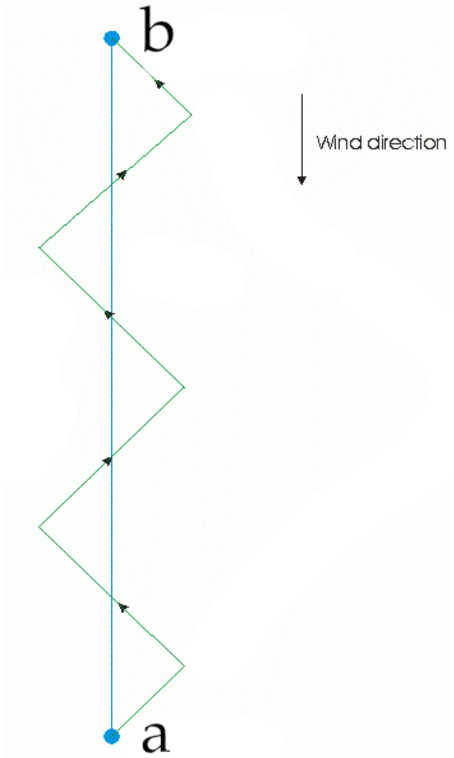

Figure 3: No-go zone situation

1. 3. True and apparent wind

Apparent wind

The wind we actually feel or the vessel feels when moving. It is a combination of the true wind and the effective wind created by our motion. If our speed is zero, the apparent wind is the same as the true wind. The apparent wind is described by the apparent wind speed and the apparent wind angle.

True wind versus apparent wind

We need to keep in mind that the Calypso weathervane shows the direction of the apparent wind as distinguished from the true wind. The difference between the directions of the true wind and the apparent wind depends on how fast you are moving relative to the true wind speed. For boat speeds less than 10 or 20 percent of the true wind speed, the difference can be neglected, and you can read the true wind direction directly. When you are moving, the direction of the true wind is always abaft of the apparent wind. If the apparent wind is on the beam, you must face this apparent wind and turn abaft to be looking in the direction the true wind comes from. This is true regardless of your point of sail. If the apparent wind is 45° on the bow, the true wind is closer to the beam. If the apparent wind is on the quarter, the true wind is closer to the stern. The exact number of degrees the true wind is abaft of the apparent wind depends on your speed relative to the wind and on your point of sail. At any relative speed, the difference between the two is largest when you are sailing with the apparent wind on the beam. The difference is typically somewhere between 10° and 40°, where, generally speaking, the higher the performance of the sailboat,

the bigger the shift can be. Apparent and true wind directions will not always be different enough to matter, but in order to read the wind - for maneuvering or for weather analysis - we must keep their potential differences in mind.
Let assume that the wind vector W_{ap} has only two components, corresponding to the wind along the x-axis and the wind along the y-axis. Then, the true wind will be :

$$W = \begin{pmatrix} ||W_{ap}|| * cos(angle(W_{ap}) + \theta) - v * cos(\gamma + \pi) \\ ||W_{ap}|| * sin(angle(W_{ap}) + \theta) - v * sin(\gamma + \pi) \end{pmatrix} \quad (1)$$

$$\psi = angle(W)$$

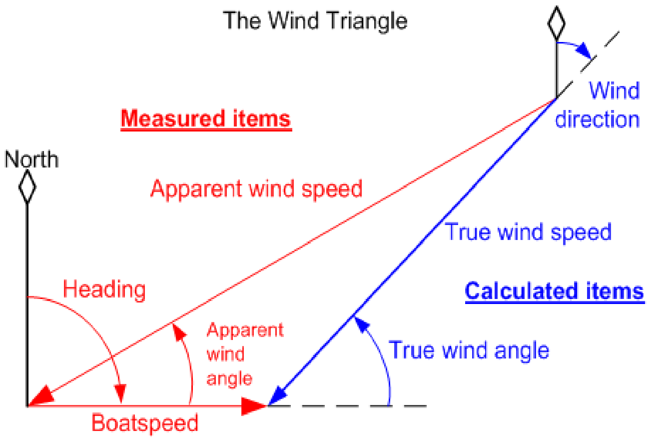

Figure 4: Computing true wind with apparent wind and boat's speed

1. 4. Modelisation

Modeling the system and its environment forces to take into consideration various constraints linked to sailboat properties. In order to keep the system simple, one has to make some initial assumptions.

- Influence from waves and currents will be neglected. As the sailboat was used on the lake, this assumption is completely accurate.
- Coriolis and Centripetal forces are neglected. The speed of the boat is not enough high to consider these forces.
- The mainsail and the forsail are supposed combined into one sail. A little horizontal gap between the mainsail and the forsail is let in order to maximise the speed of the boat.

The different variables are :

- (x, y) : 2-dimensional coordinates of the sailboat position
- v : velocity
- θ : the heading of the boat
- ω : rotational speed
- ψ_{ap} : apparent wind direction
- a_{ap} : apparent wind force
- δ_s and δ_r : respectively the angle of the sail and the angle of the boat

$X = (x, y, \theta, v, \omega)$ is the state vector that will be used for the sailboat. The inputs are δ_s and δ_r.

The evolution equations that define the system are the following :

$$\begin{cases} \dot{x} &= v\cos\theta + p_1 a\cos\psi \\ \dot{y} &= v\sin\theta + p_1 a\sin\psi \\ \dot{\theta} &= \omega \\ \dot{v} &= \frac{f_s\sin\delta_s - f_r\sin u_1 - p_2 v^2}{p_9} \\ \dot{\omega} &= \frac{f_s(p_6 - p_7\cos\delta_s) - p_8 f_r\cos u_1 - p_3\omega v}{p_{10}} \\ f_s &= p_4||W_{ap}||\sin\delta_s - \psi_{ap} \\ f_r &= p_5 v\sin u_1 \\ \sigma &= \cos\psi_{ap} + \cos u_2 \\ \delta_s &= \begin{cases} \pi + \psi_{ap} & \text{if } \sigma \leq 0 \\ -\text{sign}(\sin\psi_{ap})\cdot u_2 & \text{otherwise} \end{cases} \\ W_{ap} &= \begin{pmatrix} a\cos(\psi-\theta) - v \\ a\sin\psi - \theta \end{pmatrix} \\ \psi_{ap} &= \text{angle } W_{ap} \end{cases}$$

Figure 5: Model of the sailboat

(The values of the coefficients (p1 , p2 , ..., p10) are design parameters of the sailboat.)

2. System description

Now that we know what we need to make the system evolve, we need to collect the different data in order to make the sailboat move as we want. For this, the boat is made up of different sensors, actuators and communication components.

2. 1. Controling the system

Main unit

The Raspberry Pi 4 Model B is the latest addition to the Raspberry Pi range of nano-computers. It brings revolutionary developments to the Raspberry Pi range.

Figure 6: Raspberry Pi 4 Model B

Navio2 Board

The Navio2 board allows to turn the Raspberry into a controller for our system. It is composed of a GNSS receiver, where we will plug the GPS antenna, a dual IMU which will help for orientation and motion sensing thanks to accelerometers, gyroscopes and magnetometers.

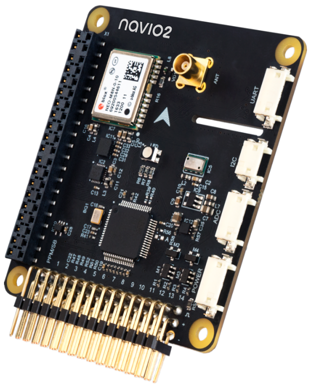

Figure 7: Navio 2 board

2. 2. Sensors

GPS

The GPS allows us to get the latitude, the longitude and the speed of the boat. By speed, it must be understand the norm and the angle. This angle can be different of the heading of the boat in the event that the boat is not going straight forward but is also drifting.

Figure 8: GPS antenna

IMU

Tampering with the data of Accelerometer, Gyroscope and Magnitometer, we can get the orientation of the boat in space. As the sailboat is navigating on a lake for our tests, we can think that it will stay in a 2D plan. Nevertheless, the orientation of the boat is highly impacted by the wind, that's why the rotation around x-axis can not be neglected. We implement a tilt compensation as follows in order to get the real orientation of the boat.

Anenometer and Weather Vane

This new product incorporates an innovative energy system. The assembly integrates a solar panel that feeds an internal battery. The Ultrasonic Portable has been designed to avoid any mechanical parts to maximize reliability and minimize maintenance. The transducers communicate between themselves two by two using ultrasonic range waves. Each couple of transductors calculate the signal delay and get information about both, wind direction and wind speed. It is important to put the sensor under the sunlight upstream in order to test it afterwards. We had some issue with the Calypso battery that's why it was sometimes difficult to test our programs with the Bluetooth sensor activated.

Figure 9: Calypso ultrasonic anenometer and weather vane.

2. 3. Actuators

The actuators are two servomotors that will steer the rudder and the sail. The one that controls the sail will rotate of an angle betwen -$\pi/2$ and $\pi/2$ while the other one will rotate of an angle between -$\pi/3$ and $\pi/3$. The signal sent by the controller is a Pulse Width Modulation signal.

Figure 10: Servomotors guiding the sail and the rudder.

2. 4. Communication components

XBEE

These modules make it easier and cheaper to deploy wireless technology in electronic devices, and to control in distance the boat and visualise it.

Figure 11: XBEE antenna

Controller

In the case the algorithm doesn't work well, we have to be able to get the boat back on the coast. It serves to ensure the safety.

Figure 12: Radio controller

2. 5. Calibration of the sensors

The principal sensor to calibrate is the IMU and more particularly the magnetometer. There are two types of distorsion possible :

- Hard iron distorsion : Hard-iron distortion is produced by materials that exhibit a constant, additive field to the earth's magnetic field, thereby generating a constant additive value to the output of each of the magnetometer axes. A hard-iron distorsion is creating an offset on the origin of the ideal magnetic circle, that is to say at the point (0,0,0).

- Soft iron distorsion : Unlike hard-iron distortion where the magnetic field is additive to the earth's field, soft-iron distortion is the result of material that influences, or distorts, a magnetic field but does not necessarily generate a magnetic field itself, and is therefore not additive.

In order to calibrate the magnetometer, the values of the hard iron, which is an offsetting value **h** and soft iron distorsion in every direction, so it is represented by the matrix **S** must be found. The calibrated value of magnetometer h_{cal} can be found with the following equation :

$mag_{cal} = S^{-1} * (mag - h)$

To calculate A and b, we had to rotate the Raspberry around every axis, tilting it in order to have enough points around each axis and have a better rotation matrix. After getting all this values, the python module scipy and more particularly the function optimize gave the offsets on every axis by calculating A and b.

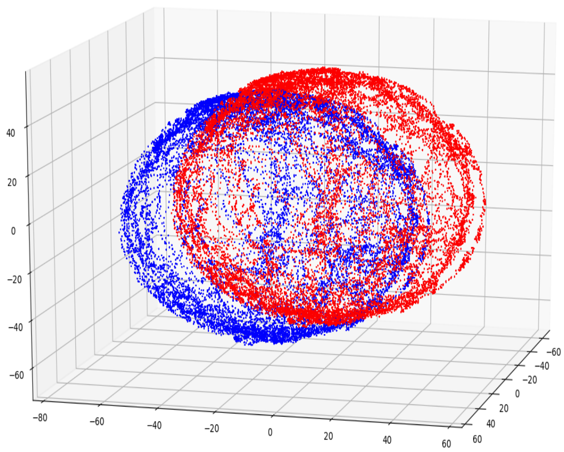

Figure 13: Magnetometer calibration

2. 6. Filtering the data

To get the right roll, pitch and yaw, we had to filter the raw data from the IMU because it showed some inconcistencies.

Kalman Filter

The Kalman Filter is an algorithm that uses a series of measurements observed over time, including statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone. The issue with a basic Kalman filter is the non linearity of angles. An Extended Kalman filter is chosen to fix this issue. The extended Kalman Filter has two parts : the predicting part, and the updating part. For more details, you can refer to the Kalman Filter course of ENSTA Bretagne.

The wind sensor and the GPS did not have many errors but sometimes it was confusing for the sailboat to have some drops on the data because it sometimes considered that a checkpoint - that was not already reached - was reached. That's why it tried to go to the next checkpoint and the wanted path was not followed.
To cancel these errors, we tried different solutions. It did not especially require a Kalman filter so the option we chosed was to implement a Low Pass Filter.

Low Pass Filter

A Low-Pass Filter is a filter that will weighten the actual value and the past value in order to avoid the data from droping due to noise in the sensor. As drops of data were the issue, we had to remove high frequency, that is why it was the best choice. The formula involves a gain α to regulate the drops :

$$v = \alpha * v_{prev} + (1 - \alpha) * v_{new}$$

3. Control, Simulate and Results

In order to get all the data as fast as possible and to have non-blocking algorithm, we use threads to gather all the data in one state vector.

3. 1. Getting the attitude

The sensors will allow us to get the heading of the boat θ thanks to the IMU, the position m will be given by the GPS, the force and the angle of the wind ψ will be given by the Calypso anemometer. As the drift of the boat can not be neglected, the direction of the speed vector is not the same as the heading, this vector is also given by the GPS.
At first, we can try to calculate the yaw, pitch and roll angles thanks to the magnetometer. However, depending on the wind force, the sailboat will tilt, that is why we have to compensate this tilt and use the accelerometer instead of the magnetometer. To do so, we compute the pitch and roll angles as follow:

$$\text{roll}=\arctan\left(\frac{A_x}{\sqrt{A_y^2+A_z^2}}\right)$$
$$\text{pitch}=\arctan\left(\frac{A_y}{\sqrt{A_x^2+A_z^2}}\right)$$

Then, we compute two auxiliar components X_h and Y_h thanks to the roll and the pitch angles that will help us to get the real heading. Finally, we compute the compensated yaw angle as follow:

$$X_h = X_m * cos(pitch) + Z_m * sin(pitch)$$
$$Y_h = X_m * sin(roll) * sin(pitch) + Y_m * cos(roll) + Z_m * sin(roll) * cos(pitch)$$
$$\text{yaw} = \text{arctan2}(Y_h, X_h)$$

3. 2. Controlling the sailboat

Line Following

In order to control the sailboat, we used the line following algorithm from the paper "A simple controller for line following of sailboats" written by Luc Jaulin and Fabrice Le Bars.

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 $q = \text{sign}(e)$
$\varphi = \text{atan2}(\mathbf{b}-\mathbf{a})$
$\theta^* = \varphi - \frac{2.\gamma_\infty}{\pi}.\text{atan}\left(\frac{e}{r}\right)$
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 $\delta_r = \delta_r^{\max}.\text{sign}(\sin(\theta - \bar{\theta}))$
$\delta_s^{\max} = \frac{\pi}{2}.\left(\frac{\cos(\psi - \bar{\theta}) + 1}{2}\right)$.

Figure 14: Line Following algorithm

Where $\mathbf{m}$ is the position of the boat, θ the heading of the boat, ψ the angle of the wind and where $\mathbf{a}$ and $\mathbf{b}$ are the two points of the line.
The farest the sailboat is from the line it must follow, the more it will be attracted. This algorithm also underlines the situation where the wind is coming from above, so the boat has to adapt.

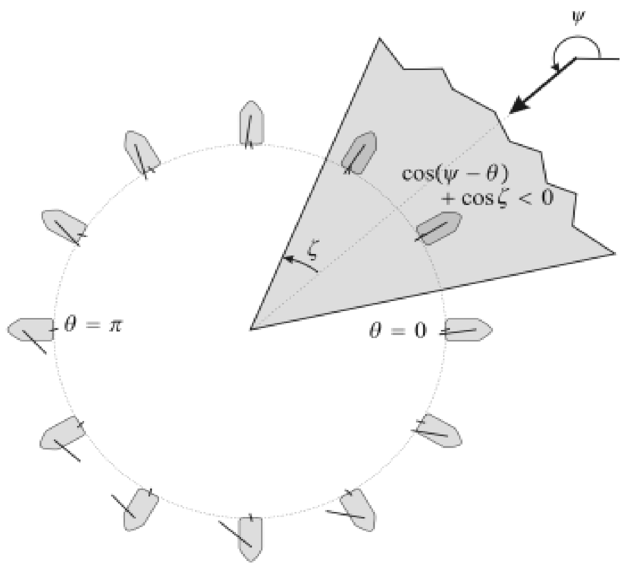

Figure 15: No-go zone situation

3. 3. Simulation

In order to check if the system well reacts according to the wind, we firstly had to program a simulator. I choosed to consider VPython as advised by my internship tutor. I created a basic representation of the sailboat as a rectangular part which represents the hull. I put another rectangular part to draw the rudder in white on the picture that follows. The last part is a combination of the sail and the mast. The wind is depicted by the white arrow and the line to follow is shown in red. To get all the data from the sensors, I implemented a server and client communication. The server was the Raspberry that gathered all the data in order to send it to the client which was my computed. This communication system based on wifi was then replaced by a communication with XBee component.

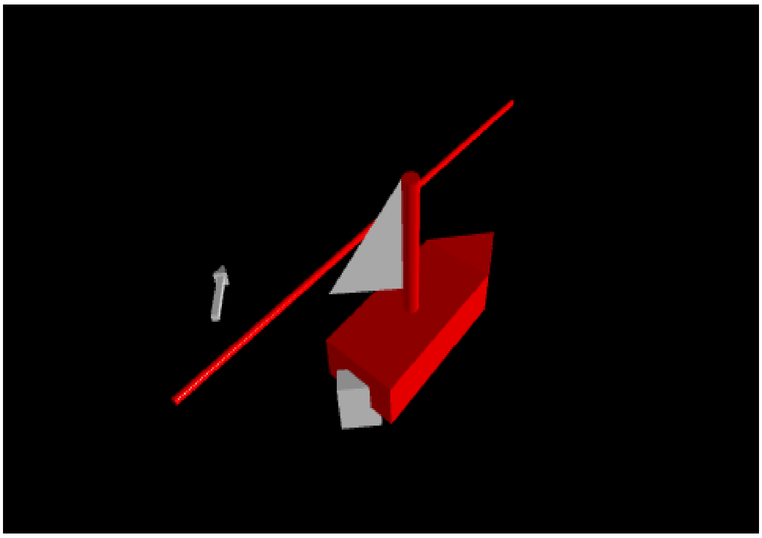

Figure 16: Simulation with VPython

3. 4. Results

At first, we tried the algorithms on a river next to the University, but the GPS was not working well as there were many buildins around it. That is why we started trying the programs on a lake at the south of Birmingham.

When trying the algorithm of line following, the sailboat was sometimes considering that it was in the situation where it must go against the wind while it was not true.

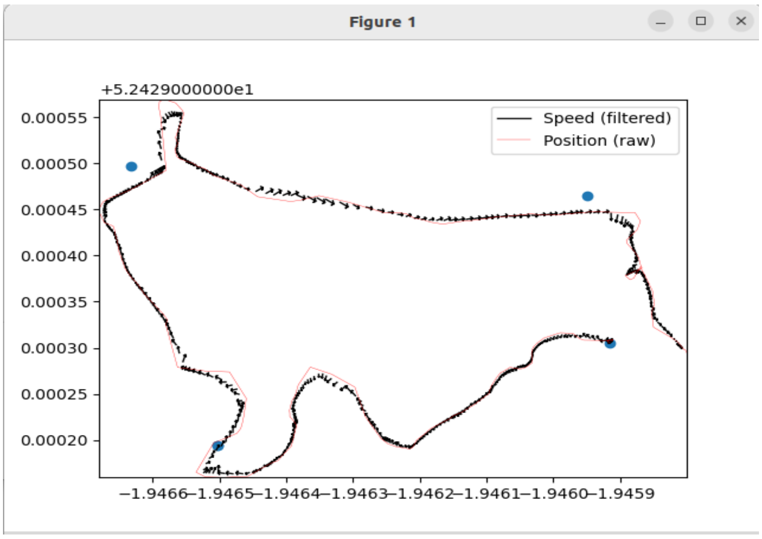

Figure 17: Triangle

This may be because the wind sensor was placed on the hull. Indeed, the sail can create some disruption and impact on the data captured by the Calypso sensor. A better place for this sensor would be on the top of the mast so that the sail would not disturb the real values of wind force and direction. Moreover, we considered that the calypso was giving the wind vector in an horizontal plane. That is true when the sail is not tilting, but when it starts to tilt, a rotation would be necessary.

Nevertheless, we can still see that the sailboat well arrive to the waypoints, drawing the triangle we asked for.

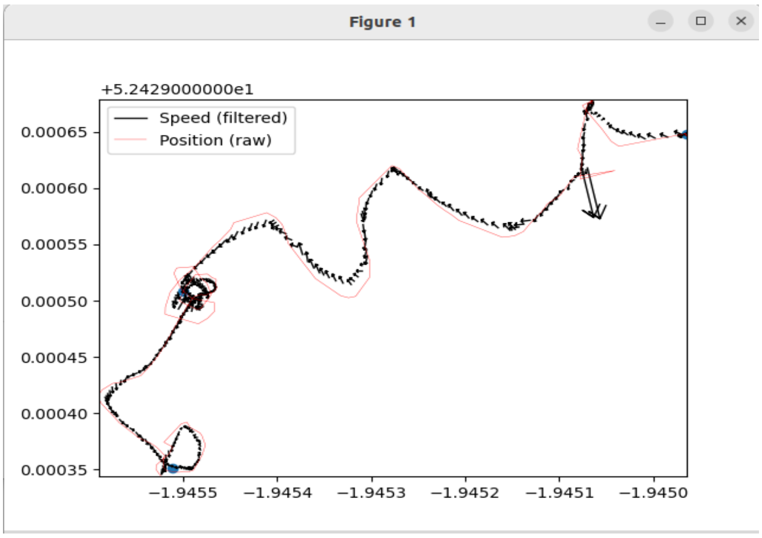

Figure 18: Station keeping

4. Human Machine Interface (HMI)

A Human-Machine Interface (HMI) is a user interface or dashboard that connects a person to a machine, system, or device. While the term can technically be applied to any screen that allows a user to interact with a device, HMI is most commonly used in the context of an industrial process. On this project, the idea was to develop a clean interface to make it easier for everyone to launch a mission. Indeed, we must currently add many arguments to inform the program that we want to log the data for example. Moreover, it could be easier to check the sensors, the calibration and to change the waypoints when we want to change it. This HMI is not ended yet, it can be something to improve in the future.

All the windows are created and we can display it but they are not linked. On the mainpage, it is possible to select whether you want to use some sensors or not. A map linked with Google Maps can show the lake where the tests were carried out and depending on the waypoints, it would appear on the map. At the top right hand corner, we can decide to check some of the sensors and then we can launch the mission.

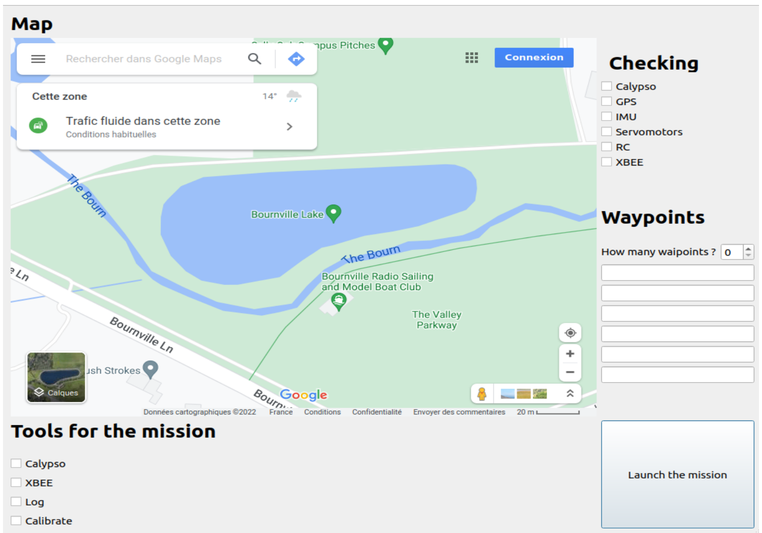

Figure 19: Main page of the HMI

When clicking on the checking part, every box to check will open a new window like the one on the picture. On this window, we can decide on how many values we want to check if the GPS is working well or not. Then it will display the different values and according to these ones, we chose to "Retry" or to check the sensor. If all the sensors are checked, the mission can be launch with better chance to succeed.

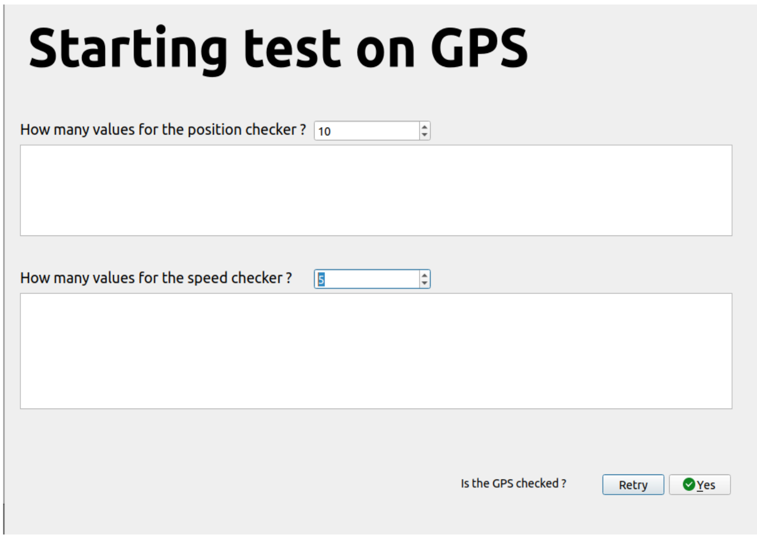

Figure 20: GPS and Speed checker window

5. Conclusion

This internship gave me the opportunity to discover more professionally speaking the research field in robotics and to work in United Kingdom. It was also fulfilling to work on this type of project as it rewarded knowledge in hardware and software. Familiarizing with different components, calibrating and filtering it, programming control algorithm and simulating it allowed me to discover which field of robotics I prefered.

Although some particularities must be taken in account to improve the success consistency, most of the original objectives were met. Moreover, . The simulation part was important before testing the system on the lake because it allowed to check any little mistakes on the program that we do not see at first.

In the future, it would be interesting to adjust the program take in account the situations said before, that is to say when the boat is tilting and the wind is not well described by the vector. It would also be interesting to test it on the sea, because there was no wave on the lake and it can disturb the system if there are some as we neglected the movements around y-axis. Finally, the HMI also needs to be improve for everyone to access more easily.

References

Luc JAULIN. *Mobile robotic*, KalMOOC. 2019.

Luc JAULIN. *Mobile robotic*, RobMOOC. 2019.

Luc JAULIN and Fabrice LE BARS. *A simple controller for line following of sailboats*.

Roland STELZER. *Autonomous Sailboat Navigation*, 2012

Jon Melin. *Modeling, control and state-estimation for an autonomous sailboat*, 2015

Ulysse Vautier. *Model Identification, Navigation, Guidance and Control of Autonomous Sailboats*, 2021