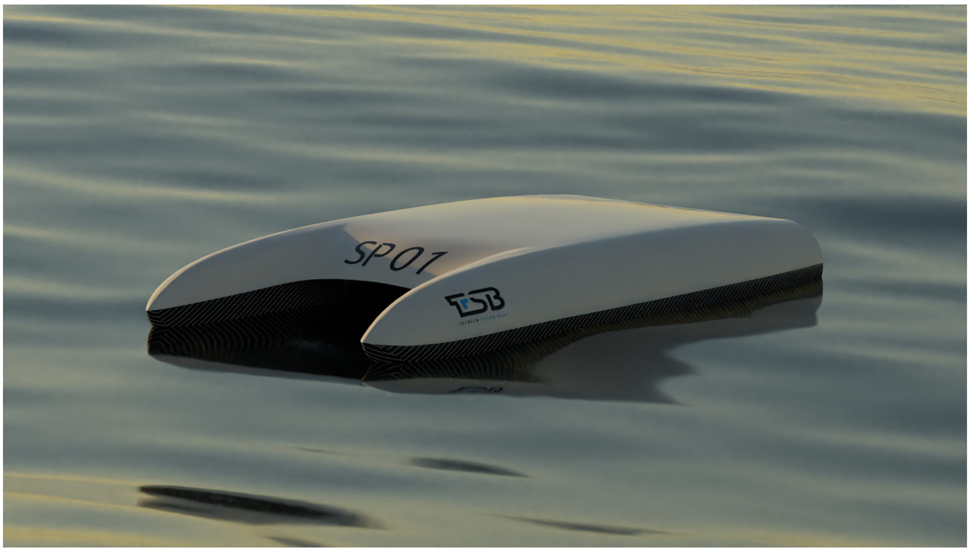
Técnico Solar Boat - SP01

Universidade de Lisboa - Instituto Superior Técnico

Technical Report - Njord Challenge

July 2022



Index

1 Introduction

Técnico Solar Boat is a university project consisting of cross-degree engineering students from Instituto Superior Técnico, the major engineering university in Portugal, that work together on the development of solar and hydrogen powered boats. Our main purpose is to participate in worldwide university competitions, such as Monaco Energy Boat Challenge, and promote Sustainability.

We count already with three solar powered boats (SR 01, SR 02 and SR 03)) which stands for São Rafael, one of the carracks that Vasco da Gama took in his fleet to India, and one Hydrogen powered boat (SM 01) which stands for São Miguel, also from Vasco da Gama's fleet. In 2019 SR 02 became second in the Monaco Solar & Energy Boat Challenge competition.

In the summer of 2021 we participated in two competitions and we had the second edition of our own event, *Odisseia TSB*, this time in the Madeira Archipelago. The first competition was in Portugal with other European teams, where we got a second place overall with our just unveiled SR03 and then we had the Monaco Energy Boat Challenge in Monaco. These international competitions tested our boat to the limit, challenging its performance in speed, maneuverability and endurance. SM01 on the other hand, only competed in Monaco.

After the development of three successful solar boats and one hydrogen-powered boat, we were looking for ways to innovate in our project, and having a functional autonomous boat has been in our sights for quite some time. Achieving full autonomy would definitely be a step in the right direction.

Njord Challenge seemed like a great opportunity to start getting familiar with everything that's needed to achieve this goal. We've all learned a lot, since none of us had experience with working in autonomous systems. We would like to thank the Njord Challenge Team for their efforts in the organization of this competition.

The Técnico Solar Boat project has been naming all our boats with names of old Portuguese Discoveries' Ships, in order to honor the history of Portugal, specifically Os Descobrimentos. Following that same logic, and after some research on historical facts that made sense for our autonomous boat, we came across Pedro Álvares Cabral's fleet, which consisted of both ships and caravels. As this vessel was relatively small, we concluded that it would be interesting to name it after a Caravel, because it was smaller than a ship. And so, the name São Pedro was born, a caravel from Pedro Álvares Cabral's fleet that discovered Brazil in the year of 1500.

The work done so far was made from a large group of multidisciplinary students, so the conception of the autonomous vessel and its technical aspects can be divided in two different areas: design and composites and electrical systems.

First we will give some details about the hull design, from the first iteration to its last and the thinking behind them. Then it will be shown the results of the construction of the vessel from the last few months and what was behind the choice of the material and the manufacturing process.

Then, with the purpose of fulfilling Njord's Challenge tasks, we design a system consisting of PID controllers and AI powered vision, as well as the path planner section that receives the cost-map from the SLAM algorithm and the information from the AI/LIDAR fusion, all connected through ROS, which will be explained in greater detail in the pages below. Finally there will also be a section for the hardware implementation, where it is explained the electrical schematic and the components needed.

2 Innovations

- One innovation of the project is the creation of our own vessel, designing it fulfilling the competition dimensions and improving the vessel's performance as much as possible, like reducing its drag.

- When projecting SP01, one of the established requirements was for it to meet the concern about the environment and make use of renewable energies. Thus, the implementation of solar panels was decided, creating a solar and battery powered vessel, contributing to the role of autonomy towards achieving zero-emission and for the energy transition that we have been witnessing in the recent years.

  This also allows the vessel to be fully autonomous, meaning that not only it navigates autonomously but it also does not need to come to land in order to be charged.

- Another innovation in this year's guidance system is the use of a neural network for the detection and identification of the different sea markers used in the competition. By correctly identifying the sea markers, the boat is able to behave as it should in a real life scenario, being an important step towards full automation.

3 Design and composites

3.1 Hull design

In the beginning of the project, several options were considered for the hull design, being them the possibility of doing a monohull with flat bottom, a catamaran, and a trimaran. The monohull with flat bottom is a common type of hull for these small autonomous boats, but it is not an efficient shape and would not make the boat look good, so that hypothesis was disregarded. Then, the trimaran was also disregarded because of the size limitations. The existence of three hulls in such a small boat would limit the interior space for the several components. Then, the final choice was the catamaran, also known for its better stability, that would help the boat's performance.

The size of the boat was limited by the Njord competition rules (1,85 meters maximum length, 1,85 meters maximum beam and 1 meter maximum height). So, being it a catamaran, the dimensions of the boat have been established to 1,5 meters in length and 1 meter in beam, to keep a good length to beam ratio. The height was stablished to be around 25 cm, again in order to keep good ratios on the dimensions.

In terms of design, the autonomous vessel model has gone through several iterations. The initial boat design always included a raised deck (instead of a flat one), in order to keep the floaters at the same depth without sacrificing storage space for electrical and mechanical systems that would do the reconnaissance and take the boat from point A to point B. The initial design was more rounded with a deck that converged with the floats almost at the bow, as shown in the figure below.

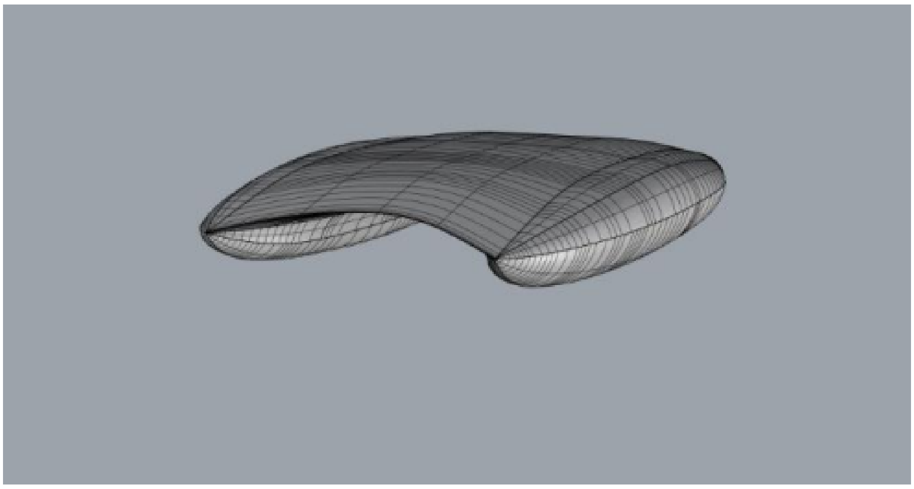


Figure 1: Initial design

This design proved to be problematic because at the maximum speeds we intended to reach, there was a tendency for the waves to gallop up onto the deck, using the floaters as a ramp, as it is possible to see in the picture.

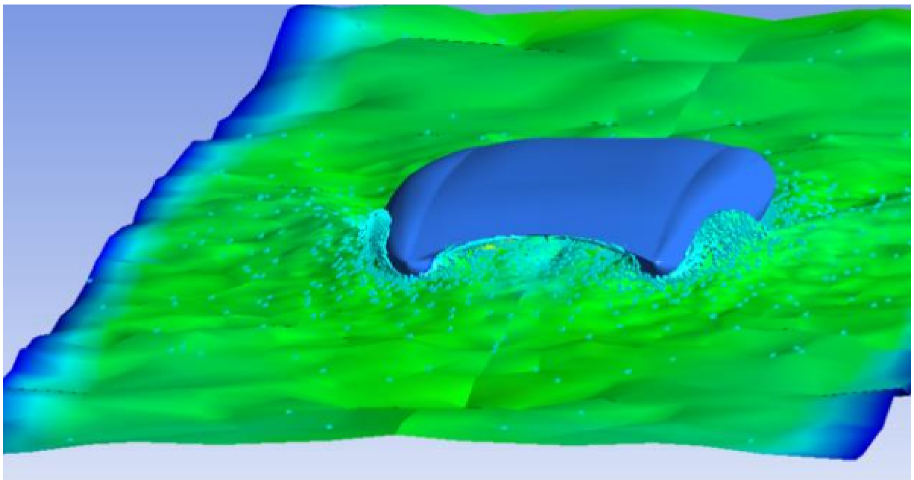


Figure 2: Wave testing

The deck was then pulled back, so that the waves go up and fall through the floaters, reducing drag and improving the vessel's performance. This second iteration is presented on the pictures below.

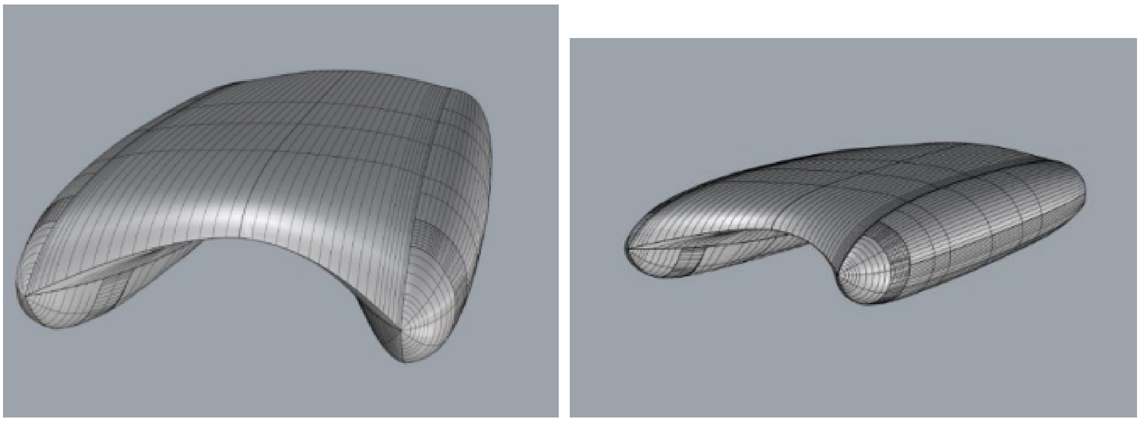


Figure 3: 2nd iteration design

Based on this design, we improved the lines so that the final appearance was rounded without sacrificing the hydrodynamics of the boat and in order to reduce as much as possible the double curvatures of both the hull and, especially, the deck, so that the intended installation of solar panels was not problematic. The double curvature of the deck would imply that when the panels were affixed, there would be the creation of micro-cracks in the photovoltaic cells that could be problematic for their integrity and performance. The final design is shown in the pictures below.

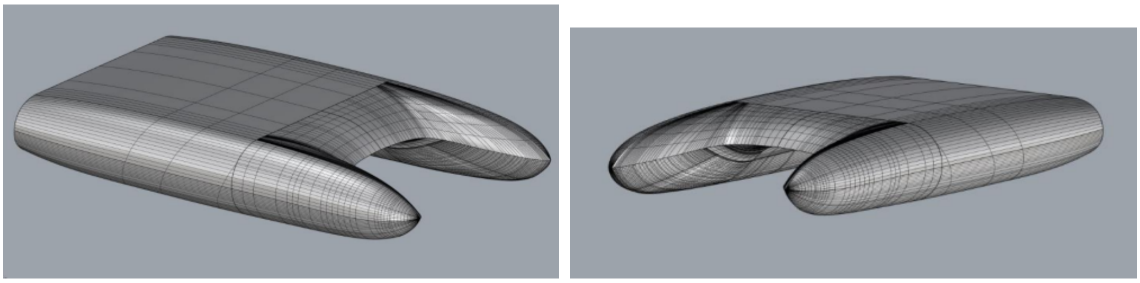


Figure 4: Final design

The work carried out also included the implementation of access hatches to the interior of the SP01, to enable the implementation, verification, and maintenance of the various systems.

3.2 Materials and construction

In terms of materials, we opted for a carbon fiber construction with epoxy resin matrix, and the implementation of sandwich laminate with a closed-cell PVC core where it was required, as these were the materials that were most favorable to our project, allowing a really light but stiff enough structure that could withstand the requirements of the test we proposed to. Most area of the floaters were chosen to be monolithic (only carbon fiber, no core), so a better heat transfer between the boat and the water is made, that way being a passive form of cooling the electrical and electronic components. A weight estimate of 10kg for the structure was done.

The main construction plan began with the CNC machining of two main molds in MDF, one for the deck and one for the hull. In terms of surface treatment, they were covered with resin, and then nicely faired, so that there would be no problems with the laminations for which they were used.

The production of the pieces (deck and hull being laminated individually) was done using the infusion process. It consists of the application of dry fibers and core in the mold (following the structural specifications required). This is followed by the application of the necessary consumables. Then, the mold is put onto an envelope shaped vacuum bag, and vacuum is applied using a pump adequate for this purpose. After that, it is possible to do the infusion of the resin, through resin intakes, which, being pulled by the vacuum effect around the perimeter of the piece, travel through the consumables, fibers, and core, impregnating the laminate. After giving enough time, the resin cures, forming a very intact and healthy layer of fibers and resin that allows us to obtain the two halves of the vessel.

After this process, the two halves were assembled using a high-strength epoxy structural adhesive (both to water and to the general wear and tear of the vessel) and the final touches of fairing and painting are applied. The goal is to have a clear coat on the hulls, so that the carbon laminate can be seen, and the deck painted in white, to avoid heating the interior compartment.

Figure 5: Results from the construction until the beginning of July

Since the delivery of the intermediate report, we were able to finally conclude the production of the hull, where we can see the clear coat mentioned above, and the deck painted in white.

Figure 6: Final results of the hull

4 Electrical systems

4.1 Software implementation

So far we have explained the hull design needs for the Autonomous Surface Vessel. Now it is shown how the vessel will have the capacity to analyze its environment and decide its trajectory by its own. For this report the approach will be more conceptual showing some of the progress made, and at the final report there will be a more technical exposure of the systems mention below. First of all, here are the sensors used for the software implementation: a LIDAR, 3 cameras and a AHRS sensor.

These are the inputs of the system since they will allow the vessel to map its surroundings and selflocate (LIDAR and AHRS sensor), as well as identify certain types of objects (cameras). All of these actions will be explained further. The last input missing is the pose of the output. This corresponds to the linear and angular position that is desired that the vessel goes to. Since the body is at an aquatic environment, it can be represented as a 2D body, meaning that it only needs to be studied the position along the xy axis (longitudinal and transverse position), as well the heading angle (i.e. the direction in which the craft's bow is pointed, which may not necessarily be the direction that the vehicle actually travels).

Below it is represented the design behind the implementation, as well as a description of each subsystem:

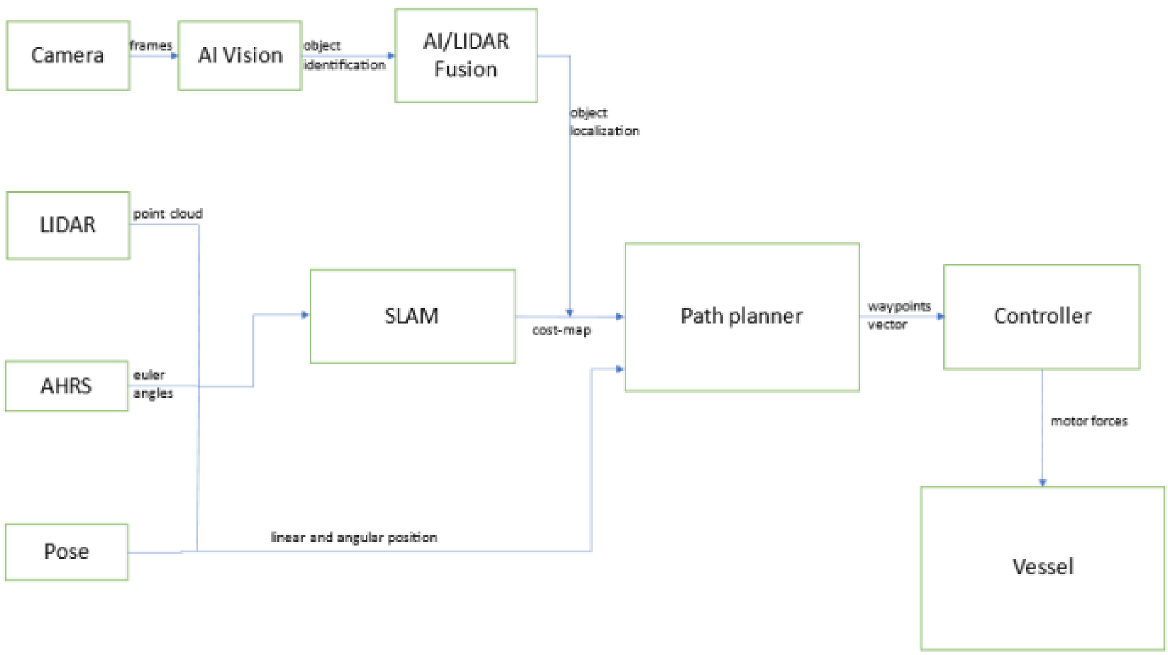


Figure 7: Softwares's design

- SLAM

  This is the first step towards automation and one of the main concepts. SLAM stands for Simultaneous Localization and Mapping, and it is the computational problem of constructing and updating a map of an unknown environment while simultaneously keeping track of the agent location within it. It works by using the information of the LIDAR that creates a point cloud of its environment, which enable the mapping of the surroundings of the vessel. At the same time, it uses the information of the AHRS sensor, that has an incorporated GPS, to allow the self-localization mentioned above.

For this implementation we used the Robot Operating System (ROS), which is a set of software libraries and tools that help build robot applications. For this case, we used the package gmapping, that uses the sensor information mention to create a 2D cost-map, which is a discrete map where each element is represented as free (white), occupied (black) or unknown (gray).

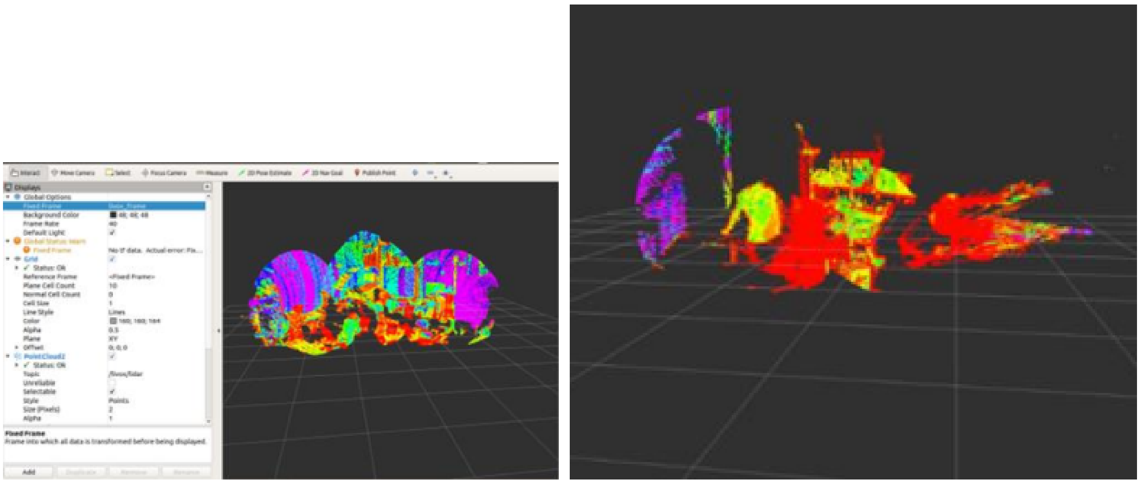


Figure 8: Point cloud obtained using the data from a LIDAR

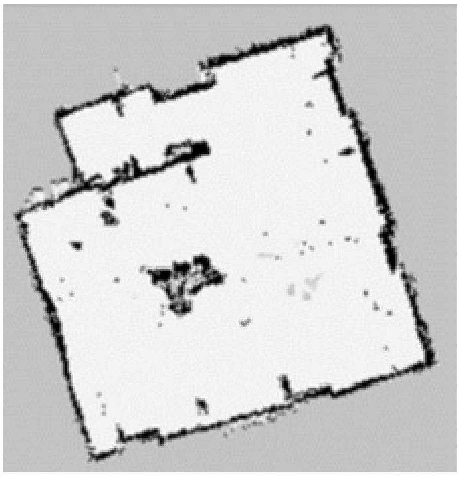


Figure 9: Example of a cost-map from gmapping

- AI Vision

  This is the subsystem where we use state of the art technology when it comes to computer vision. Although we have explained how to detect the surroundings of the vessel and its distance to it, with this process it is still not possible to identify the objects found.

  This is extremely useful for any intelligent vehicle, since the type of the object around can influence its behavior. This system was tested in the simulation we used along the way to implement our algorithms. For instance, we had identify four types of aquatic markers ahead of the boat where each one would change the trajectory in a certain way thanks to its meaning.

In order to fulfill this, we used the YOLO algorithm, which is an object detection, deep learning convolutional neural network. The first step for the implementation is to get the data that will be used to train the neural network. We used hundreds of frames of a recorded video from the simulator, where the boat was passing through many markers while travelling a path. In each one of these frames, it was manually identified the bounding box of each market represented and its identification. Without getting into the technical details, the neural network will go through several iterations in order to find features that are associated with each class of objects that we want to identify, and the output will be the knowledge to recognize from new data where the object is (using bounding boxes like represented below) and uses class probability to determine what the object is.

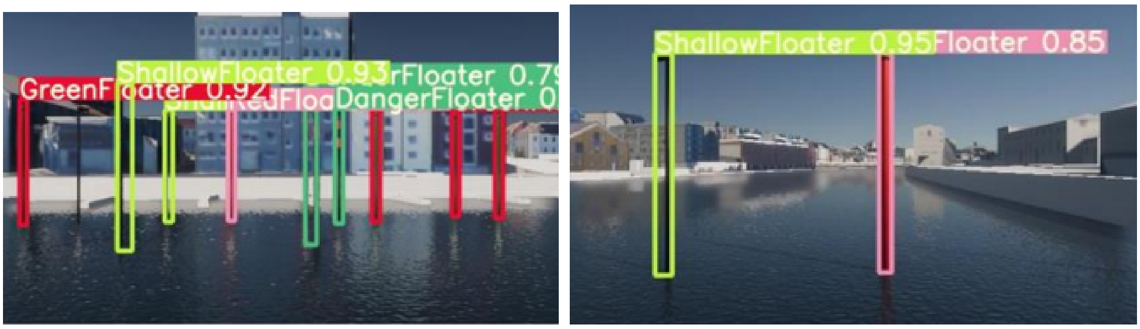


Figure 10: Demonstration of the results from the neural network YOLO implemented

- AI/Lidar Fusion

  This subsystem focuses on the correspondence between the information from de AI and from the LIDAR. The object detection from the neural network indicates which pixel from each frame matches the centre of the identified bodies. Knowing the cameras parameters, such as the FOV ("field of view") it is possible to calculate the angles between each pixel and the camera axis.

  In order to associate the detected object to a measurement from the LIDAR it is necessary to filter the point cloud to get only the points that are along the angle previously obtained. Since the Lidar and cameras are not in the same point in space, a transformation needs to be applied to the PointCloud to map it in the camera referential. Then, for each point it is computed the vertical and horizontal angle relative to the camera referential, so that it is possible to select the sample pretended.

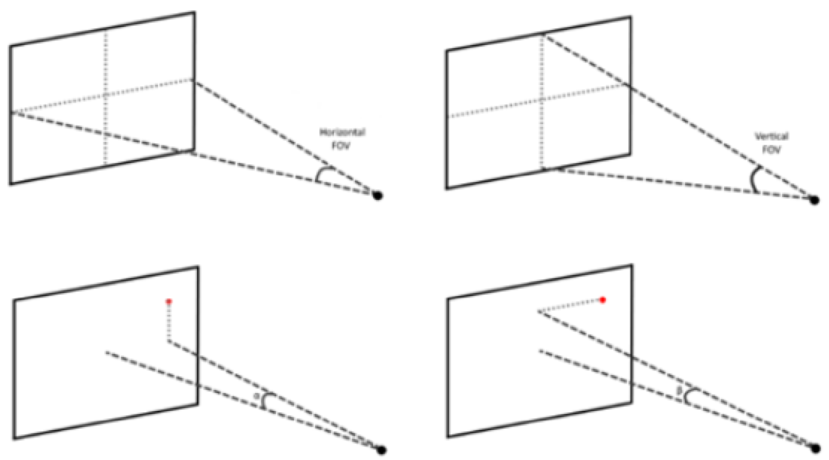


Figure 11: Representation of the vertical and horizontal angles of each point in the 2D camera referential

Finally, after analyzing the distance from each point of the subset and filter the noise, it is possible to estimate the distance and direction of the object. This new information will head to the path planning subsystem so it will have in consideration the type of objects identified in the AI Vision section.

- Path Planner

  The path planner is composed of the global planner and the local planner.

  The global planner takes in a set of GPS coordinates as a goal to move the robot to and provides an approximate trajectory for the boat to follow in order to reach its destination.

  This works by representing the world as a graph and performing a path-finding algorithm on it. The limitation of this module is that it does not take into account things such as obstacles and small details of the environment.

  The Local planner uses the map and location provided by the SLAM module. Knowing the location of elements such as buoys and other boats is critical to make a decision of where to tell the control system to move the boat to. Depending on the task set to the boat it can interpret the buoys and docks to navigate and dock autonomously.

  The output of this module is a set of waypoints that describe intermediate goals that the boat must follow in order to safely reach its destination. The waypoints are then given to the control system in order for it to follow them.

- Controller

  The objective of the autonomous boat controller is the accurate tracking of checkpoints of the desired trajectory. For this, it is necessary that the control is very precise. Given the simplicity of the dynamics of the boat in question, the control is carried out through 2 PIDs and logic blocks.

  The boat has 2 rotors/motors in the slurry, the only 2 system inputs being the power that each of the engines generates. This way, depending on the force that each rotor generates, a force (oriented longitudinally) and a resultant moment (in z) are generated. To simplify the dynamics of the boat in order to use the theory behind linear control, it was considered that the net force of the system is the only responsible for the acceleration and speed in x of the boat and the resulting moment the is only responsible for the angular acceleration (in z) , and, consequently, by the yaw angle.

The control receives the position error (difference between the current position and the desired position), calculating the yaw angle and the reference speed. The yaw error is multiplied by the gains of one of the PIDs, giving rise to the moment that needs to be applied to the boat, while the speed error is multiplied by the gains of the other PID, giving rise to the required force. Knowing the force and the necessary moment, the force of each one of the motors is determined. When the boat is in a radius close enough to a checkpoint, the logic block updates the reference checkpoint to the next point on the trajectory.

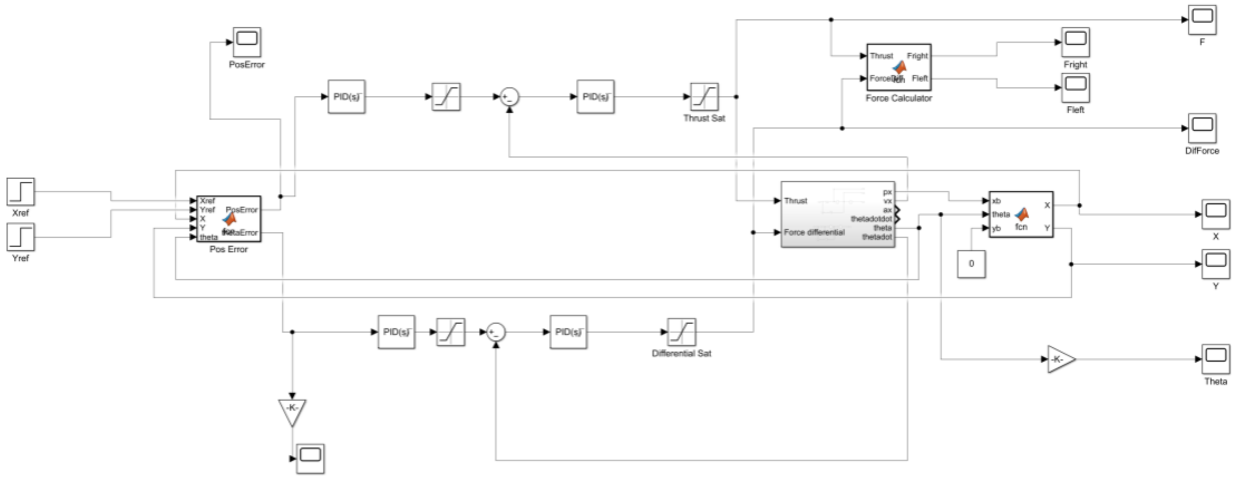


Figure 12: Implementation of the controller simulation using MATLAB

4.2 Hardware Implementation

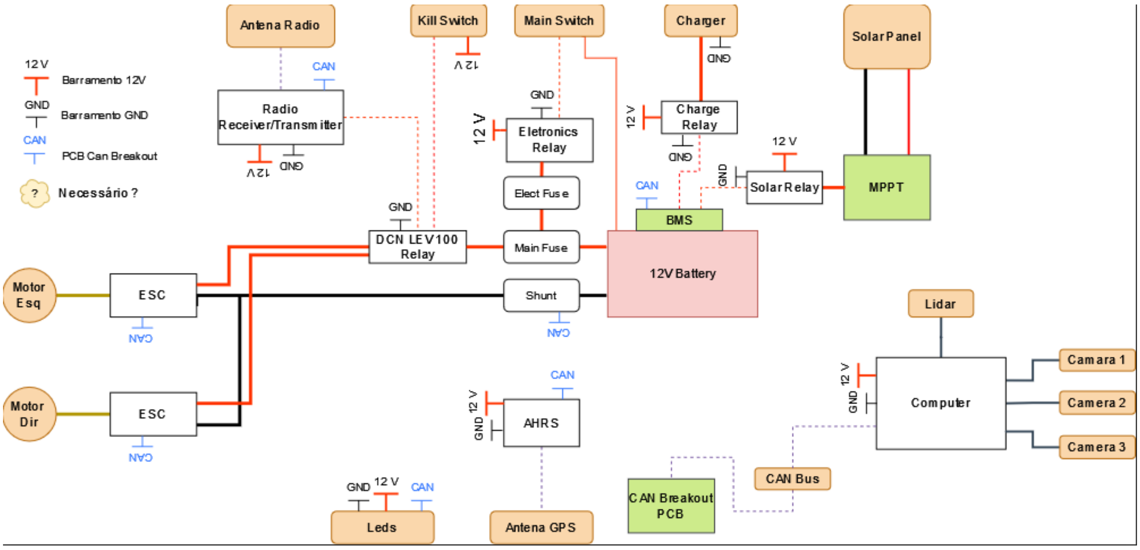


Figure 13: Electrical system schematic

We designed a theoretical schematic of what our system would look like. Firstly, we decided to use a 12V battery to power the whole boat. Perhaps we will need a DC-DC (direct current to direct current converter, to lower or increase voltage) converter to power some components, but 12V is a fit value for our motors, and the majority of the desired components.

Associated with the battery there is a BMS (battery management system), that monitors the battery cells voltage, their temperature, among other things, to ensure correct functioning. It also has a lot of safety features, preventing overcharge, over discharge, overcurrent and short circuit.

Following the battery, there will be a fuse, which ensures that there are no current peaks coming from the battery that could damage the boat's components.

We have decided to use two electric motors (each with 390W power), which should be more than enough for the boat to operate at the desired velocity. They are controlled through a ESC (Electronic Speed Controller), which is setting the motors rotations per minute and switching frequency.

The communication around the boat is done via CAN (Control Area Network) bus, which allows for a centralized communication in both ways, between several nodes. This is used to send messages to the motor controllers, and the LED switch, for example. It is also via CAN bus that the sensors data is sent to the computer to be processed. The CAN is capable of sending 1000 kbits/s.

For safety reasons, we have also implemented several switches in our boat, which allow us to shut down some systems. We have the main switch that shuts down all the boat (disconnecting everything from the battery), and also the kill switch, that shuts down the motors connection to the battery.

We also would like to install a solar panel, that will charge our battery, and its associated MPPT (maximum power point tracker), which ensures the solar panel is operating at its point of maximum efficiency.

There will be an antenna, with a receiver associated, used for the telemetry of the boat and for remote control operation. We also would like to implemented a safety feature in case we lost radio signal, we would shut down the motors shortly after.

We also will have an LED light, that displays the boat state of operation, indicating if it's operating autonomously (green light), through remote control (yellow light) or if it's shut down (red light).

Finally, we also would like to install relays between the battery sources of power (charger and solar panels), that allows us to cut off the sources of energy.

5 Navigation System for the simulator

5.1 Guidance system Implementation

For the online competition we implemented ROS nodes to carry out specific parts that make up the guidance system. Unfortunately, we were not able to implement all the features we had initially planned.

We used a image classification model called yoloV4 which is a neural network specialized in detecting objects. We trained it to recognize the sea markers and it performed quite well, being able to detect objects reliably up to 60 metres away in the simulator.

In order to implement the neural network in a ROS node we used the OpenCV Python library, and the CV_bridge library for converting between ROS messages and OpenCV images. The lidar sensor is then used to find out the exact location of each detected sea marker as described in the section above. This node publishes an array of measurements in a custom message called "RBarray" in which RB stands for Range and Bearing. The range is the distance from the marker to the center of the camera and the bearing is the angle that the marker describes from the axis of the camera. Each element in the array contains the range and bearing measurements to a marker as well as an "id" field that specifies what type of marker it is (green, red, pointed pole and shallow waters).

Then, there is a node that implements the fastSLAM algorithm which basically consists of a set of Kalman Filters that are used to estimate the location of the sea markers in a global reference frame, as well as the coordinates of the boat in that same global reference frame.

As of now, the object detection and position controller (which will be further explained in the section below) are fully working but we weren't able to successfully implement the local planner which would connect the information from the detection and SLAM nodes and grant the guidance system a higher level of autonomy. In the future we intend to convert this information into an occupancy grid map that will be later used by a local path planner node. In order to make the guidance system safer we also implemented a verification for the boat to slow down and stop in case there is an object in its way. It may not be able to find a path around it but in the very least it avoids collision.

5.2 Controller Architecture

The current position controller, designed and tested for the online simulator receives as inputs a certain position goal(given in xy coordinates), the boat's current position and also it's pose (in a quaternion).

Our controller starts by calculating the respective euler angles given the quaternion. Then we calculate the respective error in the x and y coordinate. We use this errors to calculate the needed yaw to reach the position goal.

With this, we now have the x position error, the y position error and the yaw error. This three errors are now used as inputs for two different logical functions to determine the desired speed and to correct the yaw error. So we then feed two different PIDs, one for the desired yaw, and the other for the desired speed.

The speed reference calculation and yaw error test functions are used to implement the following features:

- whenever we receive a new waypoint we start by correcting the yaw to an error of less than 30 degrees, and only then we give the boat a certain speed reference different than zero;

- to ensure that the yaw error is in the -180 degrees to the 180 degrees range;
- to slow down the boat when we're close to reaching the position goal.

Below, we can see the architecture of the controller in MATLAB, which was then further translated to ROS, and it can be seen on the force control node.

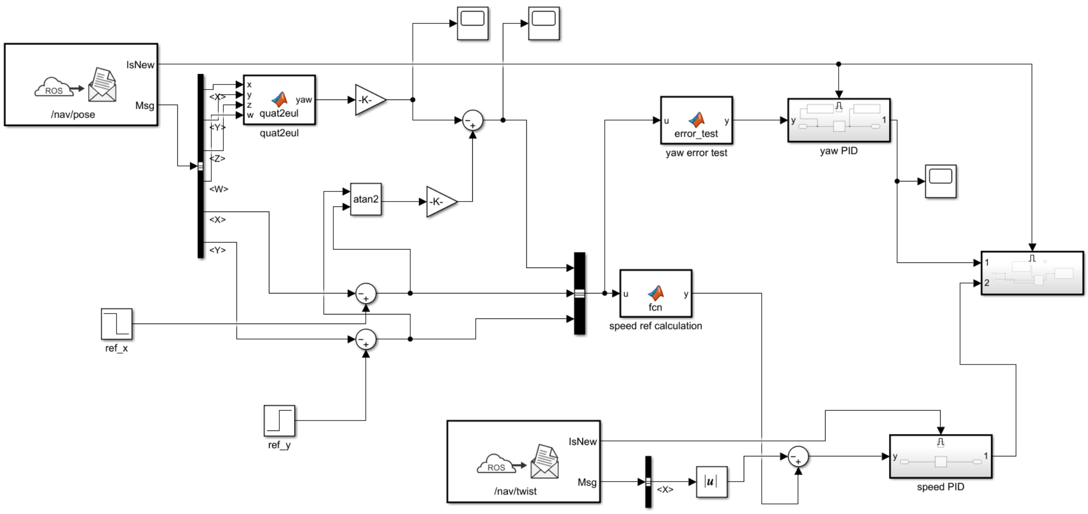


Figure 14: Controller architecture for the simulator in MATLAB

6 Conclusion

To sum up, participating in the Njord Challenge has been a great opportunity to learn what it takes to develop software for an autonomous boat as well as encourage ourselves to desgin a new vessel.

For the construction of the vessel we all worked for a very long time every week these past months in order to have the good results shown so far, and to finally have the vessel ready.

When it comes to the software implementation, we had for a long time problems with the simulator, and now we are looking for some needed hardware, like the LIDAR and the cameras, as well as new buoys so we can develop a neural network in a real environment. On the other hand, we are very confident with the yolo algorithm and the controller is already working in simulation. Also we are testing other systems like the path planner or the SLAM, with increasing success.

In conclusion, over the year we have made several progress towards developing our autonomous prototype, and hopefully, with another year of hard work on our already produced vessel, we can attend the physical competition in 2023!