

Article

# Design and Implementation of a Control System for a Sailboat Robot ‡

Davi Santos <sup>1,†</sup>, Andouglas G. Silva Junior <sup>1,†</sup>, Alvaro Negreiros <sup>1,†</sup>, João Vilas Boas <sup>2,†</sup>, Justo Alvarez <sup>1,†</sup>, Andre Araujo <sup>3,†</sup>, Rafael Vidal Aroca <sup>4,†</sup> and Luiz M. G. Gonçalves <sup>1,\*</sup>

<sup>1</sup> Universidade Federal do Rio Grande do Norte, Natal-RN 59078-900, Brazil; davihenriqueds@gmail.com (D.S.); andouglasjr@gmail.com (A.J.); alvarodenegreiros@hotmail.com (A.N.); justo.alvarez@gmail.com (J.A.)

<sup>2</sup> Instituto Federal de Educação Tecnológica do Rio Grande do Norte, Natal-RN 59015-000, Brazil; joao.vilasboas@ifrn.edu.br

<sup>3</sup> Centro de Hidrografia da Marinha do Brasil, Rio de Janeiro-RJ 24048-900, Brazil; andrepaolodantas@gmail.com

<sup>4</sup> Universidade Federal de São Carlos, São Carlos-SP 13565-905, Brazil; aroca@ufscar.br

\* Correspondence: lmarcos@dca.ufrn.br; Tel.: +558-432-153-771; Fax: +558-432-153-738

† These authors contributed equally to this work.

‡ Junior, A.G.S.; Silva, M.V.A.; Araujo, A.P.; Aroca, R.V.; Gonçalves, L.M.G. N-BOAT: An Autonomous Robotic Sailboat. In Proceedings of IEEE Latin American Robotics Symposium and Competition (LARS/LARC), Arequipa, Peru, 21–27 October 2013; pp. 24–29.

Academic Editor: Huosheng Hu

Received: 14 November 2015; Accepted: 21 January 2016; Published: 15 February 2016

**Abstract:** This article discusses a control architecture for autonomous sailboat navigation and also presents a sailboat prototype built for experimental validation of the proposed architecture. The main goal is to allow long endurance autonomous missions, such as ocean monitoring. As the system propulsion relies on wind forces instead of motors, sailboat techniques are introduced and discussed, including the needed sensors, actuators and control laws. Mathematical modeling of the sailboat, as well as control strategies developed using PID and fuzzy controllers to control the sail and the rudder are also presented. Furthermore, we also present a study of the hardware architecture that enables the system overall performance to be increased. The sailboat movement can be planned through predetermined geographical way-points provided by a base station. Simulated and experimental results are presented to validate the control architecture, including tests performed on a lake. Underwater robotics can rely on such a platform by using it as a basis vessel, where autonomous charging of unmanned vehicles could be done or else as a relay surface base station for transmitting data.

**Keywords:** sailboat; robotics; autonomy; ocean monitoring; water quality

## 1. Introduction

Oceans play a key role in global climate control. Scientists, such as Lovelock [1,2], argue that the ocean is one of the most frequently overlooked environments; however, it is amongst the most severely affected, making scientists and the population each time more interested in monitoring it. In that way, long endurance autonomous monitoring of oceanic water is an important and difficult task due to several constraints imposed by the harsh environment of the seas. One of the biggest challenges is how to provide enough energy for an autonomous water vehicle for missions with an endurance of months or even years. Therefore, the use of wind-propelled boats is a good alternative to missions that require long duration tasks, since

the propulsion energy requirements are smaller when compared to boats propelled by motors. These boats are often called wind vane servomechanisms [3]. Besides their great potential for sustainability, another main advantage of using sailboats instead of boats with a screw-propeller is that while the last consumes the greater part of its energy with propulsion, a sailboat uses wind for propulsion, thus reducing the required electrical energy, which increases its potentiality for payload (applications). Such feature makes the use of sailboats an attractive approach, mainly in cases of long running missions, where velocity is not an issue. For example, it can serve as a base station for other short-range underwater vehicles or a UAV team operating in a sea area. Simply by using solar panels and a battery for turning on servos/actuators for guiding the rudder and sail in the right positions, one can make a sailboat navigate autonomously not only for hours, but also for days, weeks or even months without human intervention. The type of actuators necessary for this kind of operation consumes much less energy than the ones necessary in a normal helix boat.

On the other hand, while traditional propelled boats can be more easily controlled due to their simplicity (it is a straightforward control task), there are some considerable differences when developing the control laws of an autonomous sailboat. In order to control the displacement of a robotic sailboat from a given position to another position within a navigable space, several issues and conditions should be addressed, such as wind direction, wind speed, boat heading and sea currents. A control loop for such a kind of boat uses the above-mentioned information to position the sail and the rudder with respect to the wind so that the boat can head and displace to the desired position. In such a system, sensors are used to obtain the wind speed and direction, which are provided to the control loop. We remark that there are unpractical heading directions that cannot be taken by the boat, thus needing some special strategy to get to a given target in that direction.

This work discusses all of the issues related to this problem, proposing solutions for the control system, which we developed, including a prototype that is currently fully functional. In the next sections, this text presents the problem in more detail, the overall system architecture, including the hardware and software of a typical sailboat, its electronics, the control law's modeling and implementation. Finally, we present current results that have been simulated using *MATLAB* and further implemented and validated on a real sailboat prototype, which is named the N-Boat.

## 2. The Art of Sailing

Sailing was one of the first approaches developed by mankind in order to traverse the seas and transport cargo for long distances. A classic sailboat uses wind for propulsion, and its motion is possible thanks to two system components: the sail that serves to capture the wind, thus giving velocity to the boat; and the rudder that controls the direction of motion. In an analogy to a car, the sail would be the engine and the rudder would be the steering wheel. In that way, in order to develop a controller for a sailboat, both the sail and the rudder should be considered for the development of its propulsion system. The art of sailing consists of controlling the position/orientation of the sail and of the rudder in order to make the sailboat follow a given direction or target.

### 2.1. Parts of a Sailboat

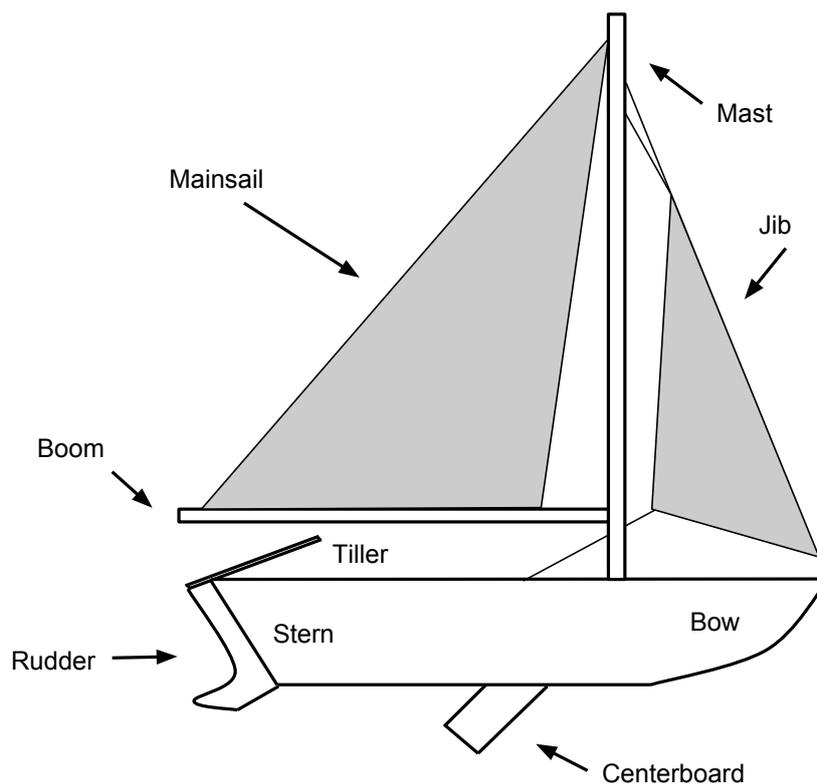
There are several types of sailboats with different shapes and functions. In general, a sailboat is composed of some essential parts shown in Figure 1:

- **Boom:** a rod situated in the bottom part of the sail that serves as the horizontal support for it.
- **Mainmast:** a central pole that supports the mainsail.
- **Mainsail:** responsible for taking the major part of the wind, being the biggest sail of the boat; it can be controlled by releasing or collecting a cable, which controls the aperture of the sail.
- **Bow:** the part located at the front of the boat.
- **Stern:** the part located at the back of the boat.
- **Rudder:** a device that allows changing the boat orientation in water.

- **Rudder stem:** a horizontal pole or rod used for controlling the rudder.
- **Keel:** a piece of lead or iron used to stabilize the boat.

Other parts, not essential for a human-piloted sailboat, but key for an autonomous one are:

- **Windsock:** a device used for measuring wind direction (gives an angular value).
- **Compass:** a device used for measuring the magnetic field of the Earth (gives an angular value related to the North Pole direction).
- **GPS:** a device used for measuring coordinates at a given position (gives latitude and longitude values related to the Earth center).



**Figure 1.** Overview of main sailboat parts (Adapted from [4]).

## 2.2. Sailing Maneuvers

It is worth noting that not all target directions can be followed by the boat. In fact, given the direction of the wind, it is impossible to go towards an angular region of directions against the wind, as shown in Figure 2. The boat cannot gain velocity when pointing towards this conic region called the dead area, which must be considered when developing the autonomous controller for the robotic sailboat. With that in mind, it is known that the design and implementation of controllers using classical control theory involve the determination of equations that model the system, which is not a trivial task for sailboats due to the highly nonlinear variables, such as water current and the direction of wind. Moreover, the drift caused by changes in wind direction and velocity is an unstable variable. According to Le Bars [5], there is no realistic state equations for sailboats. Besides, fortunately, the motion of a sailboat is in general not so fast, thus allowing certain restrictions and assumptions to be established. For instance, by using a control law that works based on minimization of errors to the target and assuming that the water current velocity is smaller than the wind velocity, in a general situation of motion, the water current effect can be neglected in the system variables.

In fact, it is transparently incorporated in the error model adopted. As our goal is to develop a simple, however effective, controller that can be applied to different types of sailboats, a pragmatic alternative is adopted. Basically, we start with the modeling of the boat. Then, we devise some controllers in the literature that are tested in *MATLAB*, and we come up with a final controller that is tested in a more realistic simulator and, finally, implemented on the sailboat prototype. For the control itself, there are not many restrictions to apply to the rudder; however, there are some for the sail mainly due to the above-cited dead region, with several cases that can be empirically established and tested. Basically, the following three situations of motion arise:

**Towards the wind direction**

Release all available cable to gather the maximum possible wind with the mainsail to get velocity.

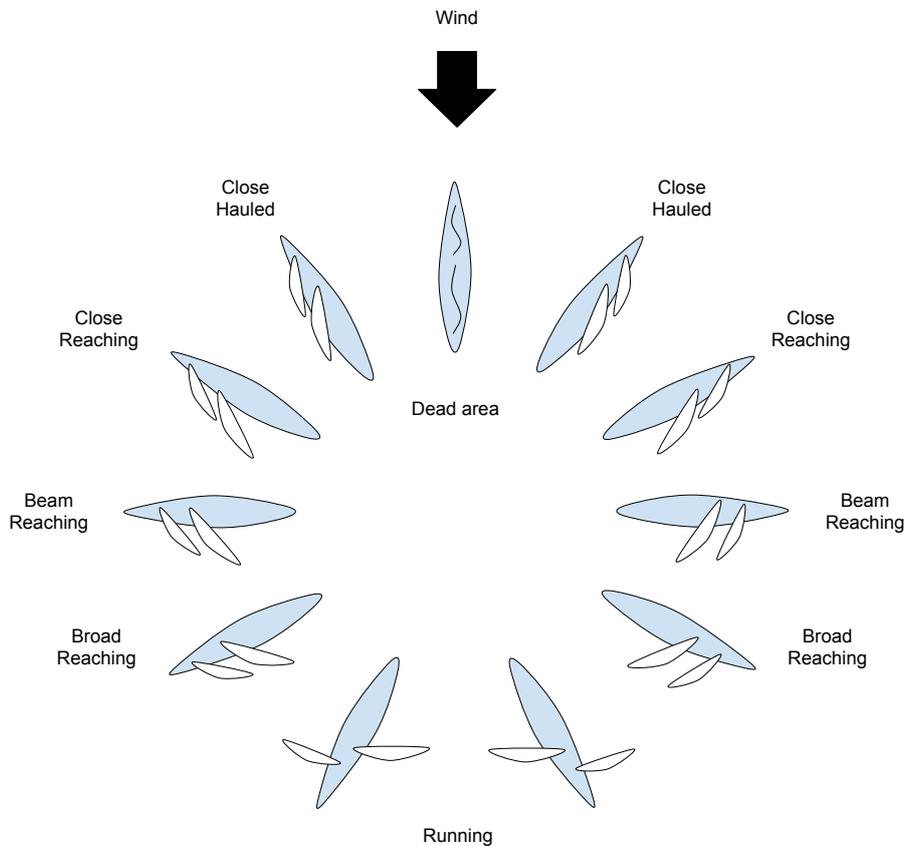
**Against the wind direction**

As shown in Figure 2, it is impossible to get velocity in this situation; thus, by tacking (alternating in lines on either side of the wind direction), the sailboat can go to the target, in a zigzag motion.

**Wind to port or to starboard**

The wind is lateral to the boat, so to get velocity, we just need to release about half of the cable size in such a way that the sail stays in about a quarter of the circumference (45 degrees) direction with respect to the wind direction.

The above actions are not the most effective; however, they solve the problem in a simple way, being enough for the situations depicted in this work context.

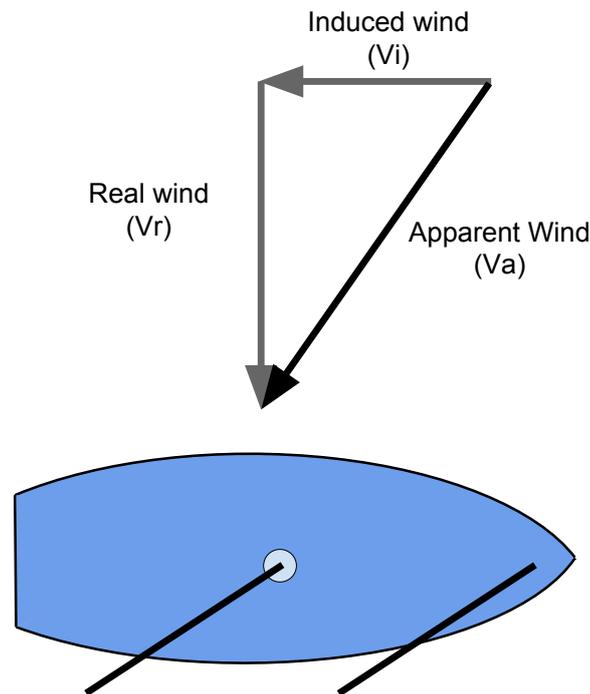


**Figure 2.** Wind circle showing positions of the sail according to the boat and wind directions (Adapted from [6]).

A last observation relates to the apparent and real wind. When the boat is moving, the sensed velocity of the wind is the apparent wind, which is a vector summation of the velocities of the true wind and of the headwind that the boat would have in still air. Note that the headwind velocity is inverse in relation to the boat velocity. In such a case, this can be understood as a vector subtraction, where the apparent wind is simply the wind velocity minus boat velocity, as is illustrated in Figure 3. Therefore, according to Figure 3, the apparent wind can be described by the Equation:

$$\vec{V}_a = \vec{V}_r + \vec{V}_i \quad (1)$$

where  $\vec{V}_a$  is the apparent wind,  $\vec{V}_r$  is the real wind and  $\vec{V}_i$  is the induced wind due to sailboat velocity. This Equation should be taken into account when developing the sail controllers.



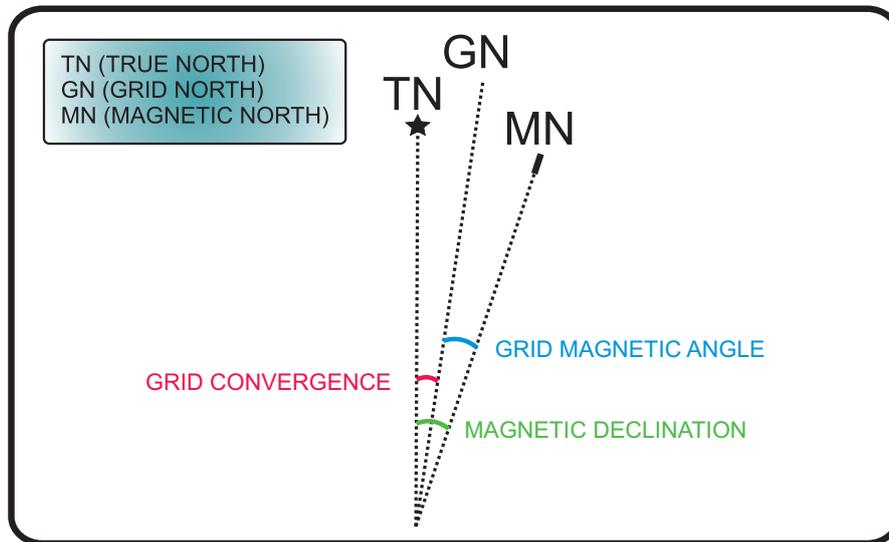
**Figure 3.** Representation of the apparent wind as a vector summation (Adapted from [7]).

### 2.3. Controllers

Controllers can be understood as system devices (or algorithms when implemented in software) that monitor the ambient conditions using sensors and change the operation of dynamical systems using actuators, based on these conditions, in order to get the desired behavior of the system. Adequate changes on the values of some maneuverable variables or parameters are done for that interference. For controlling the sailboat, we devise two parallel controllers that will be discussed in more detail later in this work.

Nonetheless, there are some other notions/terms adopted in cartography and/or geodesy used for the development of the control system that should be understood. One is the three different types of north used for orientation in geodesy: the magnetic north is the direction given by the compass; the geodetic or true north is the direction along the Earth's surface towards the geographic North Pole; and the grid north refers to the direction northwards along the grid lines of a map that (given by the grid map vertical lines). Figure 4 shows the north schematics. The correct definition and use of a north direction may influence the control, mainly in long-term running, in the calculations of routes for the sailboat, or in short running when using together the compass and GPS, for example. Of course, there are transformations that should be done between them to get more precision in the

directions. Their angular differences are known as the magnetic declination (difference between the true and magnetic north directions) and meridian convergence (difference between the true and grid north directions).



**Figure 4.** North schematic illustrating the three types of north and their differences (grid convergence and magnetic declination) (Adapted from [8]).

### 3. Related Works

There are several autonomous sailboat projects that present interesting performance and serve as a reference for the N-Boat design. Nonetheless, some of them are listed in Table 1. Next, we discuss a few of them in more detail, since they are closely related to our project. The Avalon [9], Fast [10] and Hyraii [11,12] follow the classical control theory model for the development of the sailboat controllers. The Avalon project [9] is developed to run on the *Microtransat* challenge [13] that requires a tough design to endure the harsh oceanic conditions. It is designed to be fast and precise and relies on a robust control software based on fuzzy logic both for the rudder and sail control. The IBoat project [14], also designed to run the *Microtransat* challenge, uses a fuzzy controller [15], both for the rudder and sail control. The Fast [10] project comes out with a robotic sailboat developed by the researchers of the Porto Engineering University. It presents similar characteristics to the other mentioned boats, with the difference that it uses reconfigurable computing (FPGAs) in its embedded digital control system. The HyRaii is a catamaran with T-shaped foils equipped with steerable flaps [11,12]. This sailboat uses sensors to get information about the attitude and height using devices, such as an inertial measurement unit and distance sensors [11]. Analogously, the rudder uses a flap to steer the boat. The HyRaii team found a potential to reduce the drag of operating foil flaps by less steering than with mechanically-controlled foils [12].

**Table 1.** Projects and used controllers.

Projects	Rudder Control	Sail Control
Iboat	Fuzzy	Fuzzy
Roboat	Fuzzy	Fuzzy
Fast	PI	Based on Angle of Attack
VAIMOS	Line Following	Human Behavior Observation
Avalon	PID	Based on Angle of Attack
Hyraii	PID	Based on Angle of Attack
N-Boat	PI	Based on the Wind Circle

Another project that can be considered as the first successful long running robotic sailboat is the Roboat [16,17]. It has also participated in the *Microtransat* and, besides, is also developed for monitoring a certain area of interest and to conduct marine research. It is part of a project developed by the Austrian Society for Innovative Computer Sciences (INNOC) researchers; in fact, Roboat is an enhancement of a simpler boat: the Roboat 1 [16].

From the above projects, one that is most related to ours is the Voilier Autonome Instrumente pour Mesures Oceanographiques de Surface (VAIMOS) boat [5,18]. It implements a simple, however robust, controller that allows the system to be energetically sustainable and that can be easily adaptable to other sailboats. For controlling the rudder, the VAIMOS uses a simple line-following technique, and for the sail control, it uses data observed in (or inspired by) human behavior. The high-level control of VAIMOS also has autonomous behaviors for obstacle avoidance.

#### 4. Mathematical Modeling

We develop here a simple mathematical modeling for N-Boat, in order to determine the system variables and, consequently, the equations to be used on the control. The main advantage of having this model established is that we could create a simulation environment for the sailboat. This speeds up the design of a control law by avoiding doing several tests in the real sailboat, thus improving logistics. An initial control law is projected with the basis on this simulator that is enhanced in the real one. We remark about the existence of works with more complex mathematical modeling in the literature [9,14,19,20]. The foundations for our modeling are based on the Works of Erckens [9] and Alves [19]. Basically, such modeling is divided into three main modules: coordinate system definition, kinematics and the dynamics of the model.

##### 4.1. Choosing a Reference Frame

A system with six independent coordinates (see Figure 5) is used to model and study the boat's kinematics positioning and guidance. Such a robotic system uses location based on reference points [9]. When controlling an underwater vehicle, it is important to obtain its position and velocity with respect to an inertial reference frame fixed to the Earth, but as the dynamics of the vehicle is modeled in reference to a moving frame, transformations are needed, which depend on isolated coordinates to another reference frame. The following nomenclature was proposed by the Society of Naval Architects and Marine Engineers (SNAME) [21]; these notations for the frames are shown in Table 2.

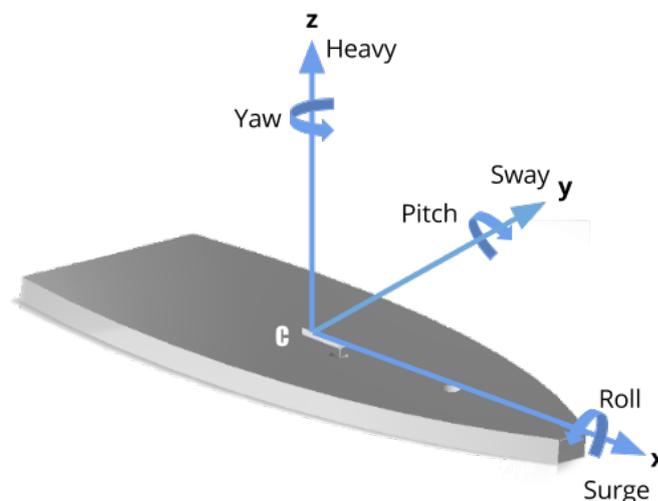


Figure 5. Adopted coordinate system.

**Table 2.** Notation used in marine vehicles by the Society of Naval Architects and Marine Engineers (SNAME).

N1	N2	V1	V2	$\tau_1$	$\tau_2$
$X, Y, Z$	$\Phi, \Theta, \Psi$	$u, v, w$	$P, Q, R$	$x, y, z$	$K, M, N$

Thus, a system is designed using two Cartesian references: the Earth inertial frame ( $X, Y, Z$ ), where the source of reference is located in the plane that is tangent to the area of interest, and the body referential frame ( $X_b, Y_b, Z_b$ ), which undergoes translational and rotational movements of the body depending on the source and generally coincides with the center of mass (CM) of the vehicle. Such frames are summarized in Table 3.

In Table 3, N1 and N2 are, respectively, the positions and orientations of the reference inertial frame fixed to the body and relative to the Earth; V1 and V2 are the linear and angular velocities in the reference frame fixed to the body; and  $t_1$  and  $t_2$  are the forces and moments exerted on the body.

**Table 3.** Nomenclature for rotations.

Motion/Rotation	Position/Euler Angle	Linear Speed/Angular	Force/Moment
X Motion (Surge)	X	$u$	$x$
Y Motion (Sway)	Y	$v$	$y$
Z Motion (Heave)	Z	$w$	$z$
X Rotation (Roll)	$\Phi$	$P$	$K$
Y Rotation (Pitch)	$\Theta$	$Q$	$M$
Z Rotation (Yaw)	$\Psi$	$R$	$N$

#### 4.2. Kinematics

Given the orientation of the boat in relation to the inertial frame of the Earth, it is possible to get any of the guidelines by means of three elementary rotations:

$$R_{x,\Phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix} \tag{2}$$

$$R_{y,\Theta} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix} \tag{3}$$

$$R_{z,\Psi} = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

#### 4.3. Dynamics

In brief (more details can be found in the related literature [9,19]), the dynamics has been divided into calculations of hydrodynamics, moment of inertia, viscous damping, lift, effort sail and center of mass. After calculating all of these parts of the dynamic force and the moment, the following expression is obtained:

$$(M_{RB} + M_A) \dot{v} + (C_{RB}(v) + C_A(v)) v + (D_k(v) + D_h(V)) V + g(\eta) = \tau_s + \tau_r \tag{5}$$

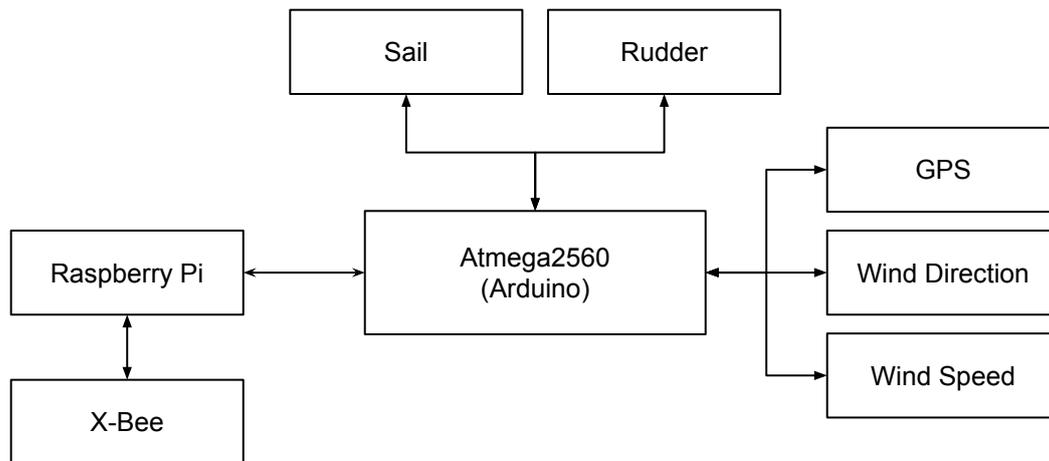
where  $M_A$  refers to the term hydrodynamics,  $C_{RB}$  is dynamics (including the center of mass),  $C_A$  comes from hydrodynamics related to Coriolis and centripetal forces,  $D_k$  is lift moments generated by

the keel and  $D_h$  are lift forces acting on the hull. Manipulating Equation (5) brings the establishment of the next control law, which, for simplicity, is the equation that governs our sailboat control:

$$(M_{RB} + M_A)\dot{v} = \tau_s + \tau_r - (C_{RB}(v) + C_A(v))v - (D_k(v) + D_h(V))V - g(\eta) \quad (6)$$

## 5. Design and Implementation of the N-Boat Control System

To obtain a more flexible system, the N-Boat system architecture is distributed within only two processing units. Low level control decisions that should be executed in real time are performed by a dedicated micro-controller, an *AtMega2560*, which is built in an *Arduino* module. This *Arduino* module is used as the low level controller of N-Boat, as seen in Figure 6, linking actuators straight to the sensors through control laws.

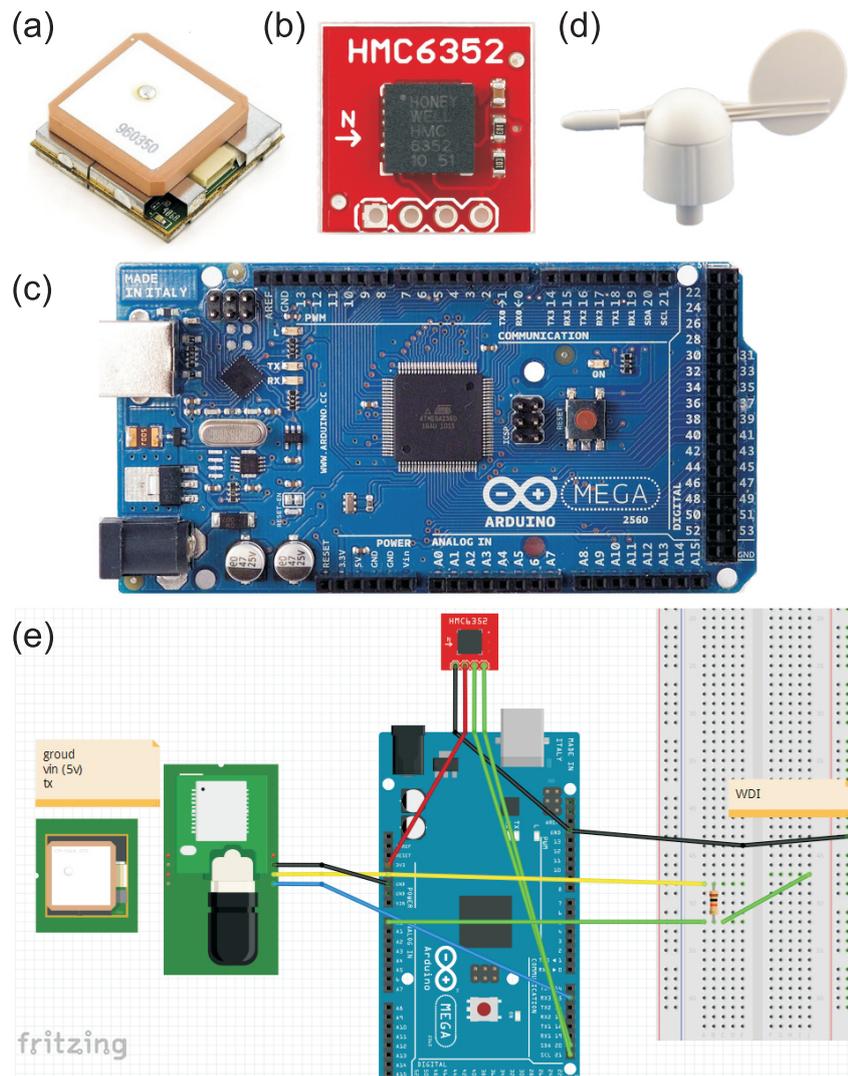


**Figure 6.** Current high-level hardware architecture of N-Boat, including communication.

The *Arduino* performs communication with the various sensors and actuators on the system, with robust control laws that would work in a stable manner. This module also interconnects sensors and the high level processing unit, a Raspberry Pi in this version of N-Boat, which is responsible for high-level navigation decisions, planning and external communication. Figure 6 shows an overview of the higher-level architecture, while Figure 7 shows the basic electronic components in a schematic view used on the N-Boat. The *Arduino* is shown as (c) in Figure 7; the other parts will be detailed further.

### 5.1. Preprocessing Module

The processing module is the central part of the robot. It is distributed in the *AtMega* microcontroller (*Arduino*), responsible for low-level control decisions and a high-level controller based on an ARM processor on a Raspberry Pi board. Raspberry Pi is a complete computer running the Linux operation system, which gives great flexibility for the system, such as the possibility of easily adding cameras, WiFi for remote access and telemetry and several other different devices. Although the system can run on speeds up to 700 MHz or more, the ARM processor is adjusted to decrease its speed to save energy when no processing is needed. This module is connected to the *Arduino* via an RS-232 serial port. This computer is also connected to a base station using an Xbee radio link of up to one mile.



**Figure 7.** Schematics of the electronic components of the N-Boat hardware: (a) GPS Receiver; (b) Digital Compass; (c) Arduino Mega; (d) Wind sensor; (e) GPS Receiver and Xbee radio link.

### 5.2. Sensing Module

The sensing and actuation module, as the name suggests, consists of all sensors and actuators of the sailboat. The used sensors are the GPS, the wind sensors (two) and the compass. The used GPS is an EM-406 SiRF III module, which provides date, time, latitude, longitude, speed, altitude and heading angle. It can be seen in part (a) of Figure 7. The system also includes a three-axis digital compass shown in Figure 7b. The wind direction is obtained through a simple windsock, or weather vane, seen in Figure 7d. Basically, the direction is given from the voltage value provided by the sensor, which has an internal potentiometer.

## 6. Devising a Preliminary Control in MATLAB

From the mathematical model obtained above and analyzing the intrinsic limitations for sailboats, two controllers were initially developed: a PID controller for the movements of the rudder, to correct the angles, and a fuzzy controller, which controls the sail (in *MATLAB*). This chosen strategy makes the control project faster, since the search space for empirically determining the parameters is so big by using brute force. Of course, the model does not truly represent the sailboat. This fact increases the amount of tuning necessary for applying the control in practice. Nonetheless,

this approach consumes less time when compared to a traditional way of adjusting the parameters. Of course, non-linear approaches could be used here; however, we prefer to follow this straight approach. It is unfortunate to note that control parameters are tuned in a straightforward way from the ones devised in the simulation (in the model). Therefore, this is the link between the control in simulation (model) and on the prototype. One could be done previously on the other, making easier the tuning of the parameters on the real one. Tests were made with these controllers in an open and closed loop.

One of the tests performed in an open loop (Figure 8) starts with the body at rest, with an orientation of  $\pi$  rad and with  $\tau_w = [-5, 0]T$ . Furthermore, we adjust the value of the rudder angle to 0 rad, and varying the angle of the sail, it was possible to analyze the behavior of the path made by the sailboat. Figure 9 shows that path. Finally, it was deduced empirically that the value of the rudder angle would be for a sail angle of  $\frac{\pi}{6}$  rad, in order to maintain this orientation of  $-\pi$  rad.

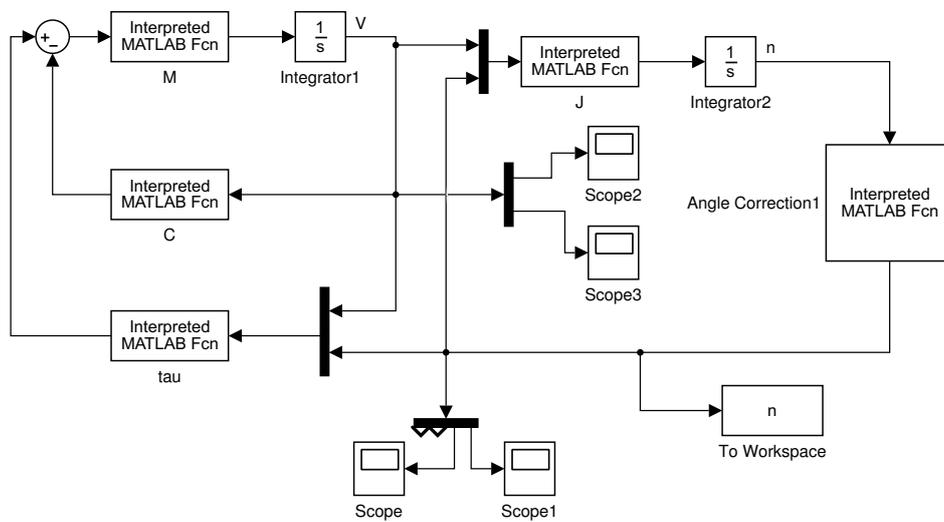


Figure 8. Open loop controller block diagram.

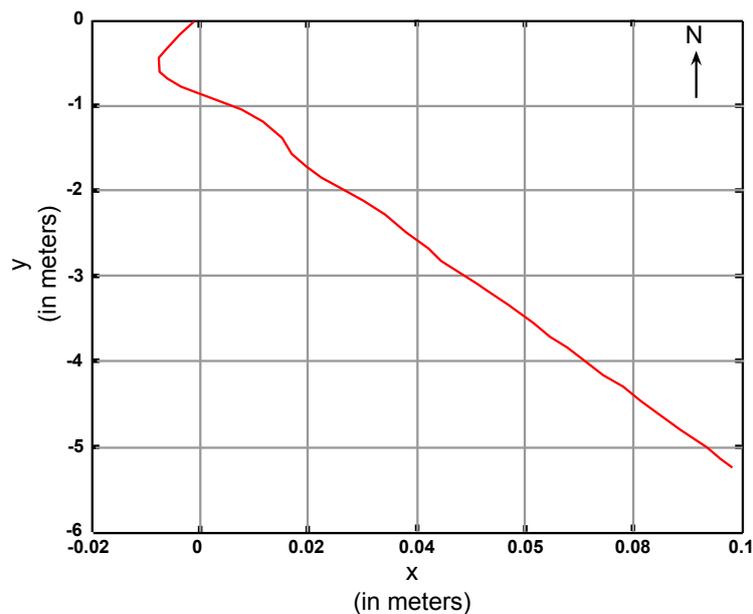


Figure 9. Simulation output with the resulting displacement in the X and Y directions.

Rudder control is firstly designed as a PID controller (Figure 10), where the variable to be integrated is the orientation error (the desired orientation less current). We also used the three components to calculate the rudder angle (output). The values of the gains are calculated from an empirical analysis of the behavior of the boat, with tuning from *Matlab* and with a snapshot of 10 s (due to the non-linearity of the model, it is necessary to indicate the length of the linearization for *MATLAB*). The values of the gains are  $K_p = 0.68$ ,  $K_i = 1.125$  and  $K_d = -0.130$ . Aiming to control the sail, we initially studied the behavior of the robot movements toward a particular orientation, to later build a controller based on fuzzy logic for representing and controlling these movements. This controller has two inputs, the orientation error and its derivative. The output is the angle of the sail, which can be mapped to how tight the sail cable must be. A set of four rules implemented in the controller is developed below:

- Case 1:** If  $\Theta = A_1$  and  $\dot{\Theta} = B_1$ , then  $\rho_S = \rho_{S1}$
- Case 2:** If  $\Theta = A_1$  and  $\dot{\Theta} = B_2$ , then  $\rho_S = \rho_{S2}$
- Case 3:** If  $\Theta = A_2$  and  $\dot{\Theta} = B_1$ , then  $\rho_S = \rho_{S3}$
- Case 4:** If  $\Theta = A_2$  and  $\dot{\Theta} = B_2$ , then  $\rho_S = \rho_{S4}$

The controller output is comprised of four different equations that make the defuzzification. These equations are of the type  $A\Phi + B\Delta\Psi + C$ , where A is a coefficient that will multiply the error, B will multiply the error derivative and C is a constant. In this initial experimental model, simulated in *MATLAB*, the control law matches the initial expectations. As seen in Figures 9–13, the initial desired objectives are achieved. Each graphic presented shows clearly both the route that the boat should follow according to the entries, as the angle for each axis in the coordinate system. For an experimental setup, sensor data can be obtained and the proposed system used to send commands to the sail motor and rudder, according to the end position desired. Therefore, from now on, we go to the boat construction and practical implementation of the control.

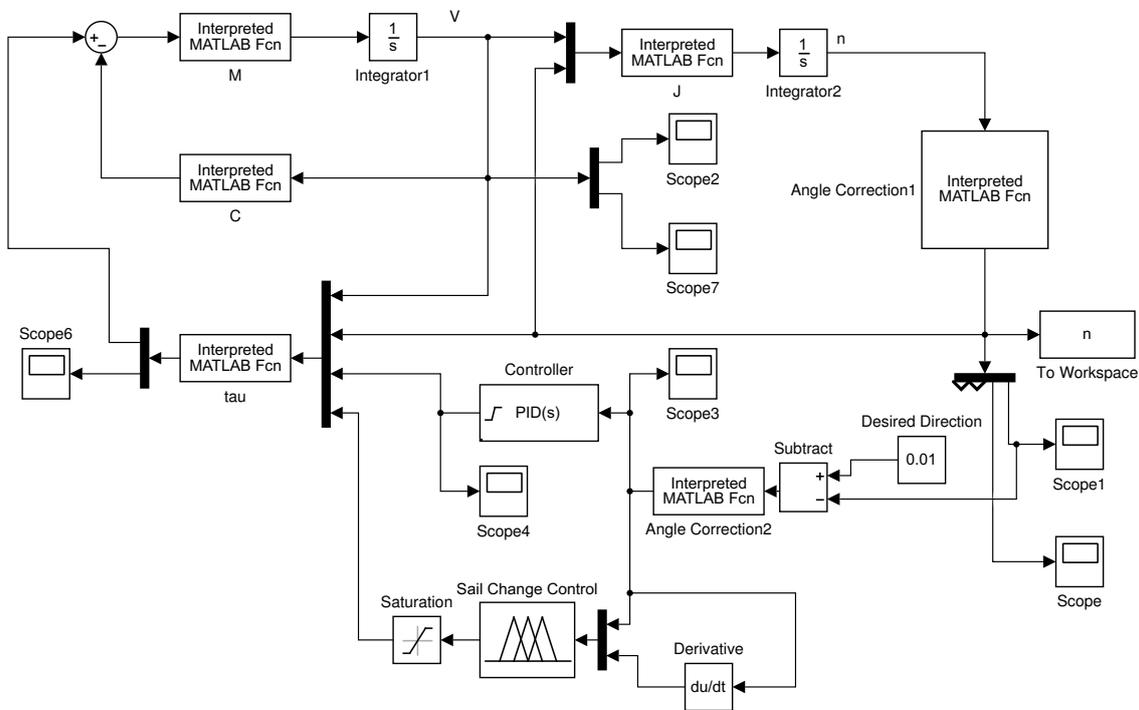


Figure 10. Closed loop system.

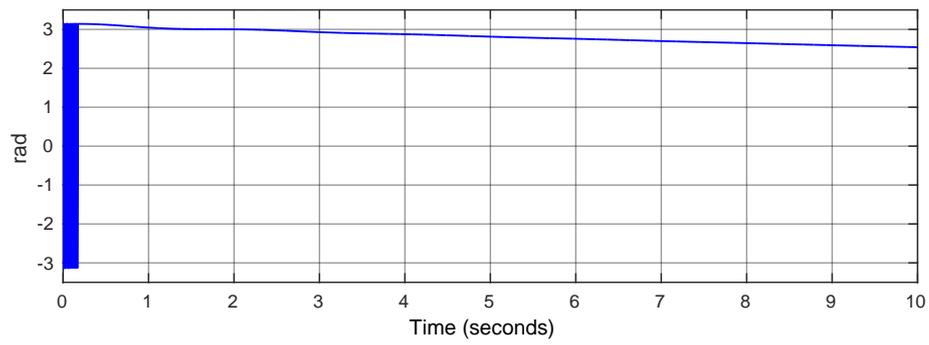


Figure 11. Simulation output with the resulting control response for the yaw angle.

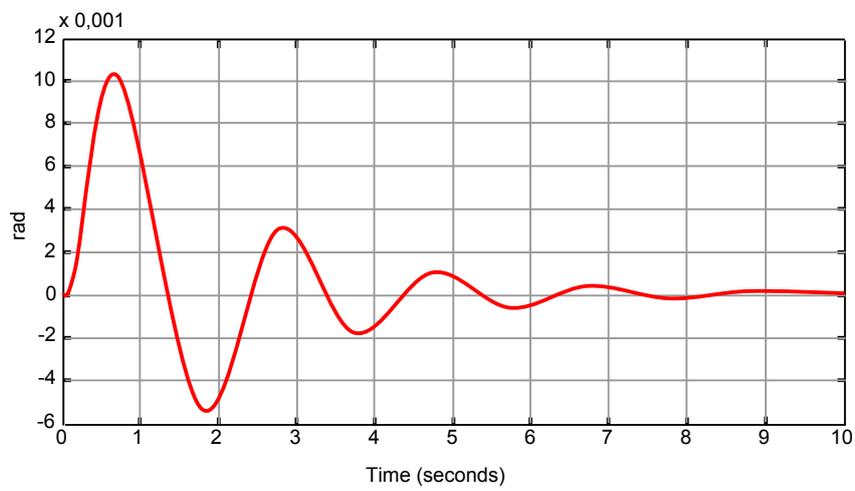


Figure 12. Simulation output with the resulting control response for the roll angle.

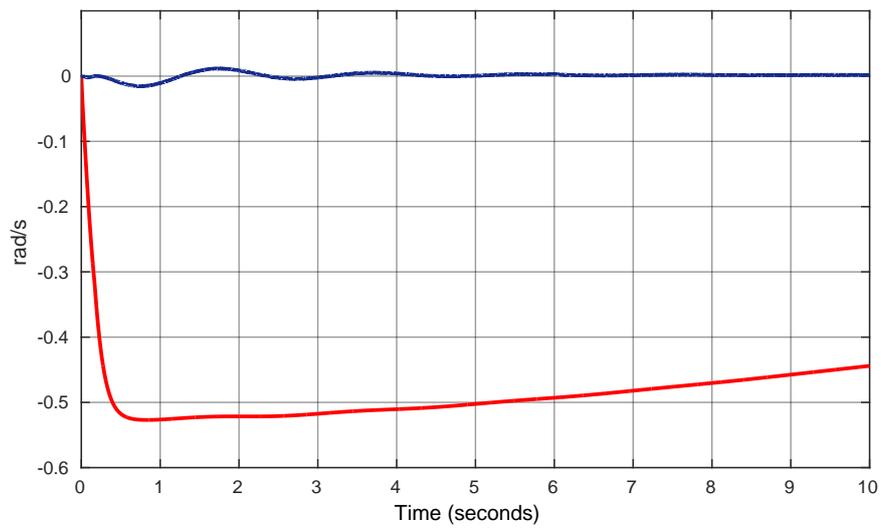


Figure 13. Simulation output with the resulting velocity profile.

### 7. Developing a Simple, Effective Controller

The problem in focus is to develop a low-level controller to autonomously get the sailboat from an initial point to a target in acceptable time and that can be reproducible in another sailboat types. Therefore, the process to be controlled should be better understood, as well as the maneuverable variables and parameters identified that we want to control. With the preliminary study about the art of sailing and the initial implemented controller in *MATLAB*, we could perceive that, for simplicity, the only variables to be controlled are the velocity and angular displacement of the boat direction in relation to the target direction. That means modifying the incidence angle of the wind at the sail and the rudder orientation that are mapped at the final angles of the rudder and sail servo-motors. Following this discovery, the current version of the proposed control system is organized as shown in Figure 14, with components that are better detailed in the following. Equation (5) is adapted here to work with the practical system devised. We noticed that the parameter D can be disregarded in the prototype control, mainly because of instability, which is generated due to the intensification of high-frequency noise caused by the derivative. Then, we came up with a simpler PI controller. We remark that the influence of the water current variable can be neglected considering that current velocity is smaller than wind velocity and that the displacement and the angle to the target are chosen as parameters to be modified in the control. Its effect to put the boat out of the target direction gets corrected in the control. The final control algorithm depicted in Figure 15 shows the boat position being controlled independently of the water current. This means that the water current influence does not matter, because the control system, at each iteration, corrects the course of the sailboat.

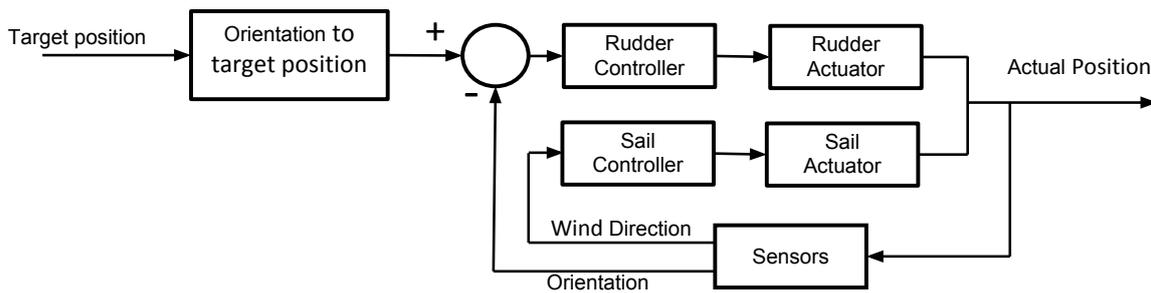


Figure 14. Complete control diagram.

#### 7.1. Sail Controller

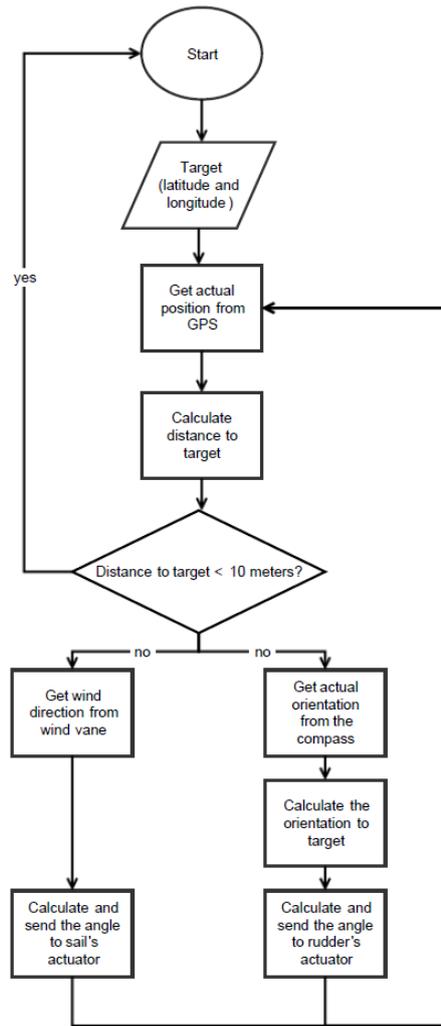
The sail controller uses information obtained from the windsock sensor and works based on the above study on the maneuverability of the sailboat. Basically, its mathematical expression can be given by the following equation:

$$S = S^- + (S^+ - S^-) \times \frac{|W_d|}{180} \tag{7}$$

where  $S^-$  is the input angle for the servo when the cable is totally retracted,  $S^+$  is the angle of the servo when the sail is totally opened and  $W_d$  is the direction of the wind given by the windsock.

For an example situation with wind in the same direction of the boat, let us say wind equals 180 degrees, we have that  $\frac{|W_d|}{180} = 1$ ; thus, the sail servo will receive  $S^+$ , turning the mainsail to totally opened, giving some propulsion to the boat.

There are situations where the direction given by the windsock is negative (southwest, west and northwest). In this case, by observing Figure 16, one can perceive that the amount of cable liberated is the same, for example to  $90^\circ$  and  $-90^\circ$ , thus explaining the use of the absolute value in the above Equation.



**Figure 15.** Algorithm used in the N-Boat control system. The inner loop is depicted in detail in the figure. In an outer loop, not shown here, the system takes waypoints (targets) to follow through (this can be set by the base station or planner).

### 7.2. Rudder Controller

The rudder controller is responsible for adjusting the boat orientation with respect to the defined trajectory. For this, the directions of the compass and the desired orientation are compared (of course transformed to the magnetic azimuth). The last is calculated using the actual boat position and the target (desired) position. This generates an error that should be reduced at each iteration of the algorithm due to the action of a PI controller. The control action  $u(k)$  is calculated with the following equations:

$$u(k) = P + I(k) \tag{8}$$

$$P = K_p E_r \tag{9}$$

$$I(k) = I(k - 1) + K_i E_r t_p \tag{10}$$

$$E_r = \theta_d - \theta_c \tag{11}$$

where  $P$ ,  $I$ ,  $K_p$  and  $K_i$  are the proportional action, integral action, proportional gain and integral gain, respectively,  $E_r$  is the error between the desired angle ( $\theta_d$ ) and the actual angle ( $\theta_c$ ) in relation to the target position and  $t_p$  is the sample time of the system.

We could observe that the PI controller works well in the simulation with the parameters that are generated by the model. However, it is unfortunate to mention that with these same parameters, the PI does not work well in the sailboat prototype. In fact, these values are given by the simulation model to be further used as an initial estimate in the real model, over which a fine tuning is performed in order to get the final values for  $K_p$  and  $K_i$ , such that the sailboat behaves as desired. After some tests, we found the values of  $K_p = 2.3$  and  $K_i = 0.8$ .

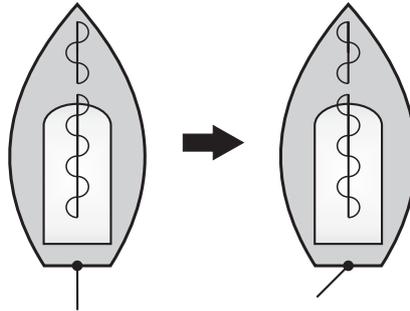


Figure 16. Negative section: from  $\frac{R^+}{2}$  to  $R^-$ .

The rudder working region is divided into two regions: negative and positive. The negative section (Figure 16) is responsible for turning the boat in the counterclockwise direction. It corresponds to the interval  $\frac{R^+}{2}$  to  $R^-$  degrees in the rudder servo angle and is given by:

$$R = \frac{R^+}{2} + (R^- - \frac{R^+}{2}) \times \frac{u(k)}{180} \tag{12}$$

where  $R^+$  is the limit angle of the rudder servo-motor in the clockwise sense and  $R^-$  is the limit angle of the rudder servo in the counterclockwise sense.

The positive section (Figure 17) is responsible for turning the boat in the clockwise sense, which corresponds to the interval  $\frac{R^+}{2}$  to  $R^+$  degrees in the rudder servo-motor. Its equation is given by:

$$R = \frac{R^+}{2} + \left( R^+ - \frac{R^+}{2} \right) \times \frac{u(k)}{180} \tag{13}$$

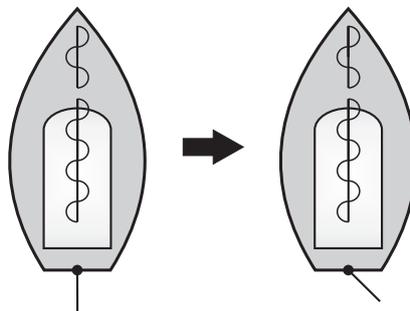


Figure 17. Positive section: from  $\frac{R^+}{2}$  to  $R^+$ .

Analyzing Figure 18, we can better note how the rudder controller works. Supposing that  $R^+ = 100^\circ$  in the initial situation and that the boat is pointing to the grid north, thus  $\theta_c = 0$  and  $R = \frac{R^+}{2} = 50^\circ$ . Defining the target at 90 degrees from the grid north, that is  $\theta_d = 90^\circ$ , the error is calculated according to Equation (11).

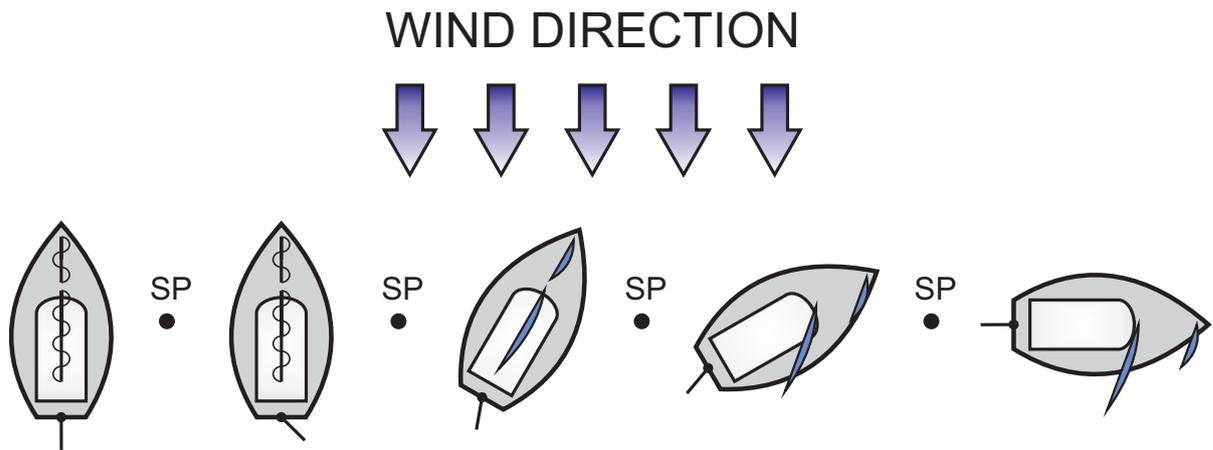


Figure 18. Example of the behavior of the sailboat.

In case of positive error, this means that the sailboat should turn in the clockwise sense in order to reach the target. Additionally and consequently, the rudder should go to the positive region, which implies the calculation of the rudder position according to Equation (14), resulting in  $R = 75$ .

$$R = \frac{R^+}{2} \times \left( 1 + \frac{u(k)}{180} \right) \tag{14}$$

When the sailboat starts its rotation, the angle in relation to the grid north starts changing. In the third situation of Figure 18, the sailboat is at  $45^\circ$  from the grid north. As a consequence, the error will be smaller ( $E_r = 90 - 45 = 45$ ). With error reduction, the value of  $R$  goes close to  $\frac{R^+}{2}$ , a situation in which the boat does not rotate anymore.

## 8. Hardware and Software

In the presented version of the N-Boat, the GPS, the compass and the windssock are used as the main sensors to obtain the necessary information for the navigation controller.

### 8.1. The Hardware

Figure 7 shows the way the hardware components of the system are disposed, as a whole. We remark that some calibration of the sensors are needed in order for the system to work properly.

#### 8.1.1. Calibrating the Windssock

As explained previously, the windssock works as an endless potentiometer. Thus, in order to measure the wind direction, it is necessary to map the analog voltage present on the output of the potentiometer to all of the wind directions. As the cardinal and collateral points of the windssock have different resistances, their positions can be measured by the proposed system using an analog to digital converter.

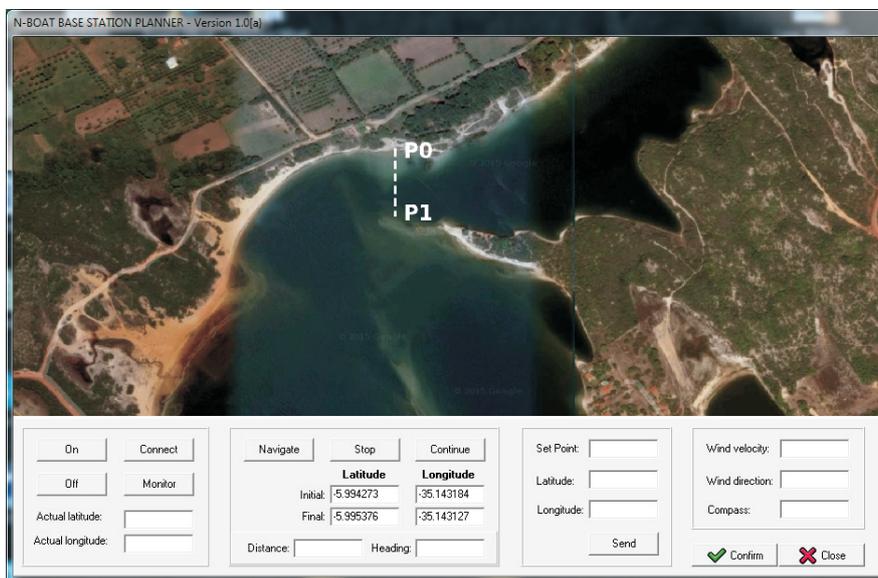
#### 8.1.2. Calibrating the Compass

The calibration of the compass is straightforward and consists of measuring its north according to the grid (or it could be geodetic) north. In our case, we used the grid north. For that, a point of reference is selected, and with the boat pointing to it, the value of the compass is measured. This value is then subtracted for all of the next measures of the compass. In this way, when the compass gives  $90^\circ$ , the direction is also  $90^\circ$  from the grid north. Of course, for huge distances, there is a considerable change in the magnetic north due to magnetic declination, as seen above, which does modify the functioning of the compass and could affect this kind of approach. However, at smaller distances,

there are few changes in the magnetic declination, thus allowing this strategy to work properly. In practice, there are maps of the magnetic declination available that could be used for that.

## 8.2. Software

In the whole functioning of the sailing system, there are two functions (loops), an external loop that is responsible for high-level control, or navigation, and the inner control loop that effectively provides the control (which will be better depicted here). In the external loop, which is run by the base station or by a planner, the waypoints are provided. Other interruptions could also be provided by this function, as changing directions or such. We have this outer loop running already, based on a supervisory system where the waypoints can be set (Figure 19). The current inner part of the system is implemented in C language, and its basic control loop is organized according to Figure 15. Basically, in the algorithm, the sailboat takes a waypoint executing the upper iteration steps shown in Figure 15 until a distance of less than 10 meters from the target is reached.



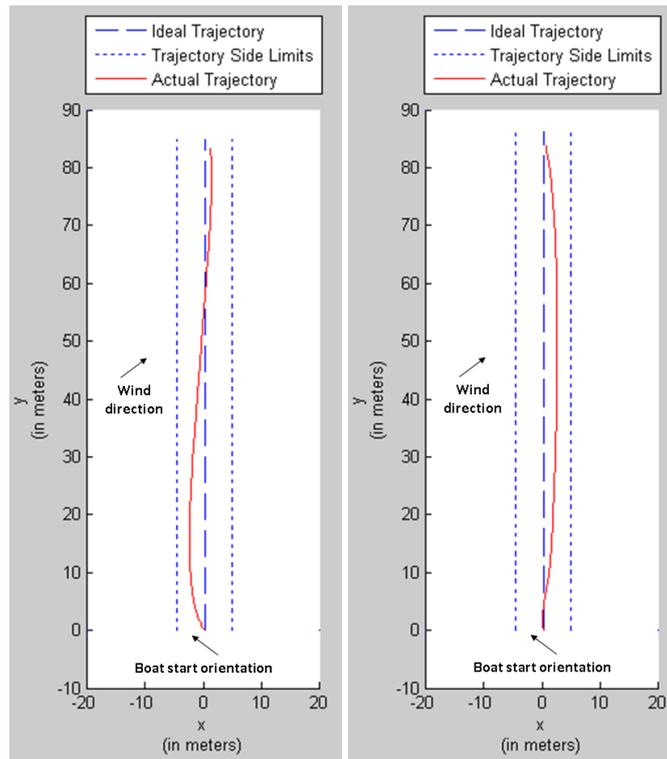
**Figure 19.** Base station user interface for the boat's waypoint configuration and trajectory visualization.

## 9. Results and Discussion

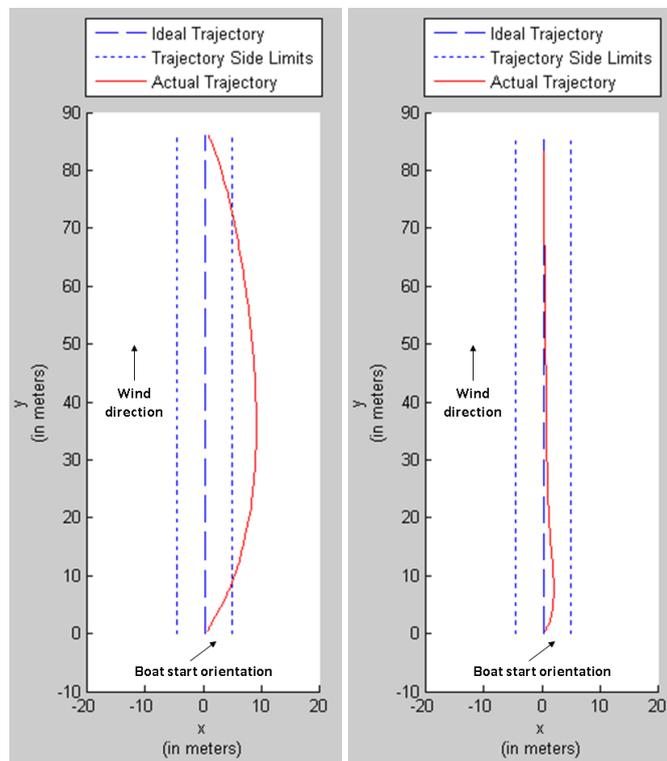
In order to assess the method devised in this work, besides the initial verification of the controller done in *MATLAB*, we also have done other experiments and tests for enhancing the model, in the simulation, and we also show practical experiments in the prototype. We start with further simulation done to enhance the model.

### 9.1. Enhancing the Controller in Simulation

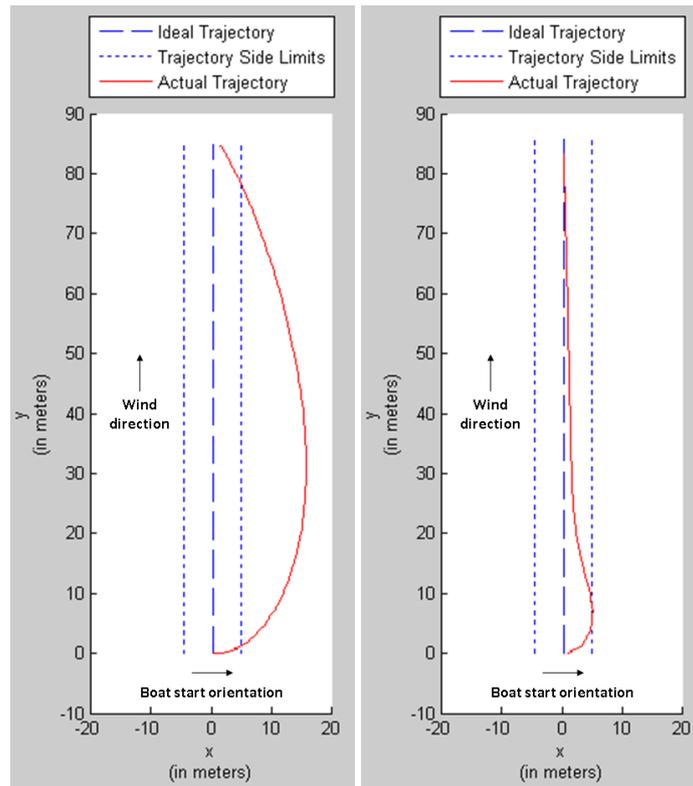
The controller should bring the boat from an initial point to a target defined by the user in the base station. The only restriction for the controller is that it should keep the boat inside a region of 10 meters away from the ideal trajectory (GPS precision). Figure 20 shows results for the first experiment with a wind direction of 45 degrees and the sailboat with an azimuth of 270 degrees. The subsequent figures show resulting trajectories for the sailboat at 45 degrees and wind at 180 degrees (Figure 21), wind at 180 and sailboat at 90 degrees (Figure 22), wind at 180 and sailboat at 135 degrees (Figure 23), wind at 90 degrees and sailboat at 180 degrees (Figure 24) and, finally, wind at 90 degrees and sailboat at 200 degrees (Figure 25). In all of the figures, left is for the P controller and right is for the PI controller.



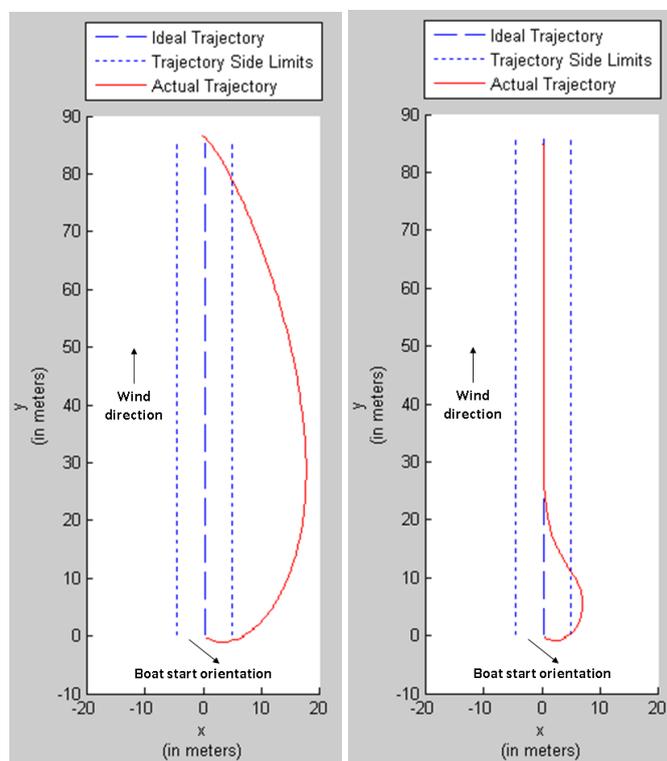
**Figure 20.** Simulation experiments for a path followed by the prototype (shown further) with wind at 45 degrees and the boat heading at 300 degrees from the true north. Left is a simple P controller, and right is PI.



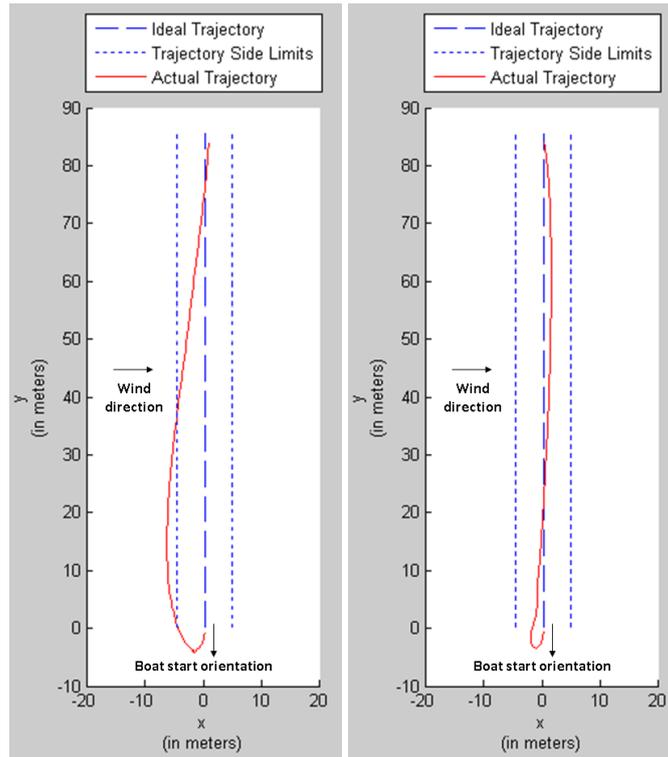
**Figure 21.** Simulation run for wind at 180 degrees and the boat heading at 45 degrees from the true north (left is P controller, and right is PI).



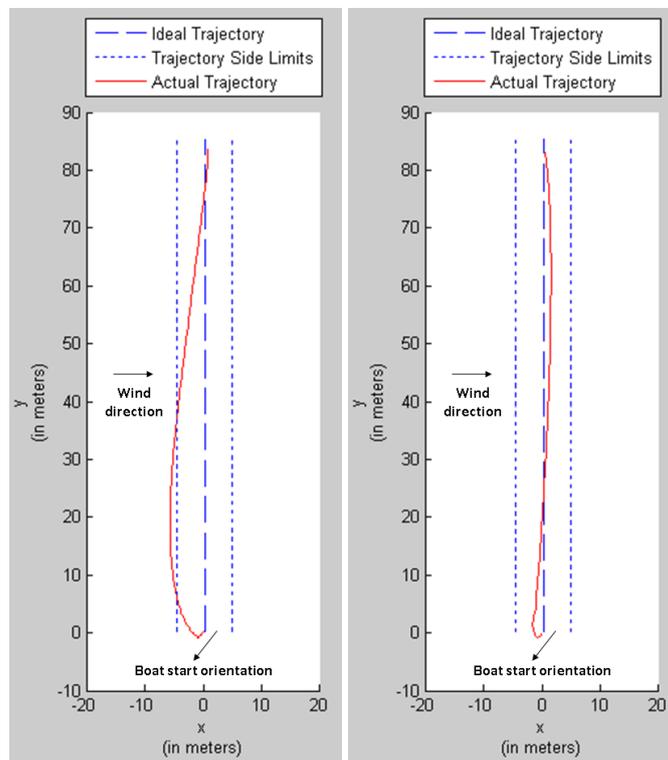
**Figure 22.** Simulation run for wind at 180 degrees and the boat heading at 90 degrees from the true north (left is P controller, and right is PI).



**Figure 23.** Simulation run for wind at 180 degrees and the boat heading at 135 degrees from the true north (left is P controller, and right is PI).



**Figure 24.** Simulation run for wind at 90 degrees and the boat heading at 180 degrees from the true north (left is P controller, and right is PI).



**Figure 25.** Simulation run for wind at 90 degrees and the boat heading at 200 degrees from the true north (left is P controller, and right is PI).

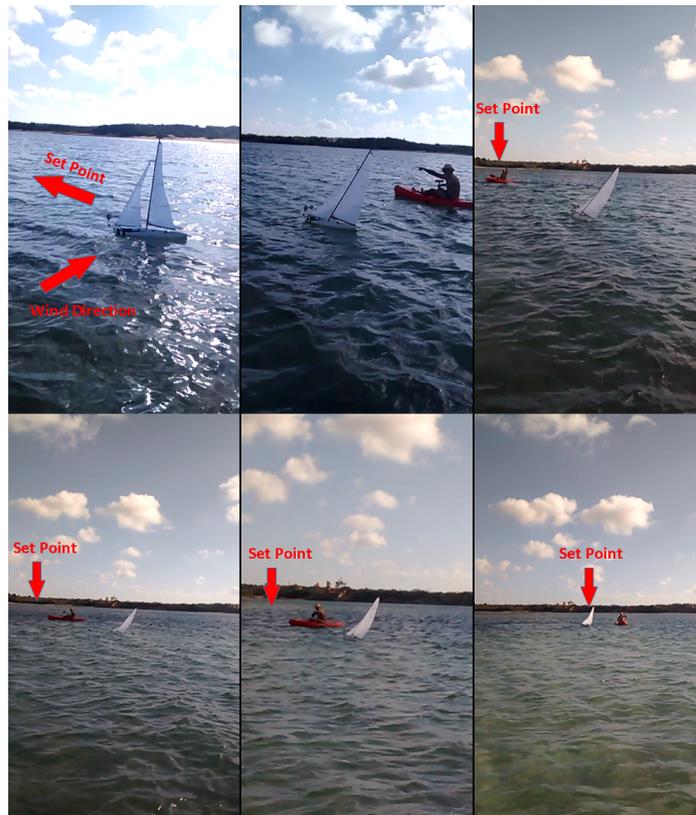
It is worth mentioning that in all experiments conducted, the sailboat always reached the specified goal. Furthermore, it can be observed in the figures that an accentuated catenary at the starting of the trajectory occurs. Therefore, based on these experiments, it is possible to devise another controller that diminishes the curvature of the initial catenary, which is the currently running controller that is finally implemented in the prototype.

### 9.2. Practical Tests in the Sailboat Prototype

The sailboat used in the experiments consists of a radio-controlled sailboat model, of 90 cm in length. We changed the initial radio control to be autonomously controlled with the on-board hardware as described earlier. The electronic controller has been developed and embedded onto the vessel. Some modifications on its structure have been also done in order to avoid water from entering the boat. After all implementations, including the development of several on-board circuits, the N-Boat prototype was ready for the tests realized in a lake (Alçaçuz) in the Northeast of Brazil (Natal).

For this practical test (illustrated in Figure 26), the sailboat is positioned at a point on the water with an orientation perpendicular to the target point in order to test the practical behavior of the controller. The controller could correctly adjust the orientation and reach the goal. In this first experiment, the selected target is a destination 85 m away from the initial position of the boat, which correctly changed its direction and took approximately 2 min to reach the goal. The mean velocity is 0.7 m per second. Figure 27 shows the path followed by the N-Boat in the first run.

In the second run, the sailboat is positioned such that it points to a direction that is opposite the target. The boat could make the turn and initiate its motion towards the goal. With these practical results, we consider that the current version of the control system is successfully implemented.



**Figure 26.** Practical test on the prototype. The boat takes about two minutes to reach the goal. A video of this experiment can be watched on YouTube [22].



**Figure 27.** Trajectory of the sailboat in the practical experiment.

## 10. Conclusions

This work has introduced a system architecture and a basic control paradigm for autonomous sailboats. The initial proposal is to design and develop a control system able to control the sailboat displacement from a position to another with enough generality, in such a way that it can be implemented in other sailboats. Therefore, the main contribution is the control system developed that is able, in a robust way, to bring the sailboat to its target using very simple sensors, such as a compass, a GPS and a windsock. Three types of controllers were tested, with fuzzy, PID and a very simple strategy that is currently implemented in the prototype and working in a robust way.

The presented results were validated with a simulator and on a prototype of a sailboat and show that the control laws match the expected performance for displacement of the boat in a particular region. Another contribution is that from now on, it allows the validation of other techniques developed by the group, with the generation of trajectory, tacking techniques, among others, improving research in this area. As applications, we envision that the sailboat can work as a surface vessel in support of autonomous underwater vehicles, for surface telemetry, recharging and other payloads.

Ongoing future work includes the development of a sailboat in a larger scale. In fact, a second version of our sailboat, with 2.4 m in length, has been already constructed. It will receive all of the hardware already developed in this smaller version plus other hardware that we have already acquired/developed. Two high definition depth sensing cameras (Stereolabs ZED 2K Stereo Camera) and a GPU-based image processing board (Jetson TK) will provide more information to the sailboat, allowing obstacle detection and avoidance. Other sensors would be used, such as water velocity in relation to the boat, and telemetry will be improved with a satellite transmission-based system. A kit for measuring water quality is already working and will also be used in this bigger sailboat, which has also solar panels in order to be more sustainable. Thus, such a system shall be applied in water monitoring and other long-running tasks on the ocean.

**Acknowledgments:** The authors acknowledge the Brazilian sponsoring agencies CAPES and CNPq for the grants to Luiz Gonçalves, Justo Alvarez, Andouglas Junior and Davi Henrique Souza.

**Author Contributions:** All of the authors have equally contributed to this work, either in the implementations or in the writing of this article. The N-Boat project has financial support from CNPq and the PI (and Coordinator) is Luiz M. G. Gonçalves.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lovelock, J. *Gaia: A New Look at Life on Earth*, 3rd ed.; Oxford University Press: Oxford, UK, 2000.
2. Lovelock, J. *The Vanishing Face of Gaia. A Final Warning*; Basic Books: New York, NY, USA, 2009.
3. Belcher, B. *Wind-Vane Self-Steering: How to Plan and Make Your Own*; International Marine Publishing Company: New York, NY, USA, 1982.
4. How to Build a Sail Boat that is Much Cheaper Than Retail Ones. Available online: <http://www.instructables.com/id/Build-a-sail-boat-that-is-much-cheaper-than-retail/?ALLSTEPS> (accessed on 4 February 2016).
5. Le Bars, F.; Jaulin, L. An Experimental Validation of a Robust Controller with the VAIMOS Autonomous Sailboat. In Proceedings of the 5th International Robotic Sailing Conference, Cardiff, Wales, UK, 17–21 September 2012.
6. Sailing Glossary and More. Available online: <http://1-36.com/terms.php?menu=11> (accessed on 4 February 2016).
7. School of Sailing: True & Apparent Wind. Available online: <http://www.schoolofsailing.net/true-and-apparent-wind.html> (accessed on 4 February 2016).
8. Vicmap Topographic Maps Online: Magnetic Declination. Available online: <http://services.land.vic.gov.au/maps/imf/search/Topo30Information.jsp> (accessed on 4 February 2016).
9. Erckens, H.; Busser, G.; Pradalier, C.; Siegwart, R.Y. Avalon—Navigation Strategy and Trajectory Following Controller for an Autonomous Sailing Vessel. *IEEE Rob. Autom. Mag.* **2010**, *17*, 45–54.
10. Alvesma, J.; Cruz, N. FAST—An Autonomous Sailing Platform for Oceanographic Missions. In Proceedings of the MTS-IEEE Conference Oceans'2008, Quebec City, QC, Canada, 15–18 September 2008.
11. Furrer, F.; Gohl, P.J. Control of an Autonomous Sailing Catamaran on the Concept of Hydrofoils. Available online: [http://control.ee.ethz.ch/~richters/hyrai\\_report.pdf](http://control.ee.ethz.ch/~richters/hyrai_report.pdf) (accessed on 30 January 2016).
12. Furrer, F. Developing a Simulation Model of a Catamaran Using the Concept of Hydrofoils. Available online: [http://students.asl.ethz.ch/upl\\_pdf/253-report.pdf](http://students.asl.ethz.ch/upl_pdf/253-report.pdf) (accessed on 4 February 2016).
13. The Microtransat Challenge. Available online: <http://www.microtransat.org/> (accessed on 4 February 2016).
14. Briere, Y. IBoat: An unmanned sailing robot for the Microtransat challenge and ocean monitoring. In Proceedings of the TAROS-2007: Towards Autonomous Robotics Systems Conference, Aberystwyth, UK, 3–5 September 2007.
15. Briere, Y. Iboat: An autonomous robot for long-term offshore operation. In Proceeding of The 14th IEEE Mediterranean Electrotechnical Conference, Ajaccio, France, 5–7 May 2008; pp. 323–329.
16. Stelzer, R.; Proll, T.; John, R.I. Fuzzy Logic Control System for Autonomous Sailboats. In Proceedings of the IEEE International Conference on Fuzzy Systems, London, UK, 23–26 July 2007; pp. 97–102.
17. Stelzer, R.; Jafarmadar, K. The Robotic Sailing Boat ASV Roboat as a Maritime Research Platform. In Proceedings of the 22nd International HISWA Symposium on Yacht Design and Yacht Construction, Amsterdam, The Netherlands, 12–13 November 2012.
18. Le Bars, F.; Jaulin, L. A simple controller for line following of sailboats. In Proceeding of the 5th International Robotic Sailing Conference, Cardiff, UK, 17–21 September 2012.
19. Carlos, M.M.A. Sailboat: Autonomous Marine Robot of Eolic Propulsion. Master's Thesis, Universidade Técnica de Lisboa, Lisboa, Portugal, 2010.
20. Lindegaard, K.W. Acceleration Feedback in Dynamic Positioning. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2003.
21. Nomenclature for Treating the Motion of Submerged Body Through a Fluid (SNAME Technical and Research Bulletin No. 1–5). Available online: <http://www.itk.ntnu.no/fag/gnc/papers/Sname1950.PDF> (accessed on 4 February 2016).
22. Teste Agua. Available online: <https://youtu.be/G-6RhZYpKJA> (accessed on 4 February 2016).

