COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

AUTONOMOUS SAILBOAT

Autor: Lorenzo Rodríguez Pérez

Director: Arne Fliflet

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título AUTONOMOUS SAILBOAT

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2021-2022 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Fecha: 18/ 05/ 2022

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Fecha: 19/ 05/ 2022

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

AUTONOMOUS SAILBOAT

Autor: Lorenzo Rodríguez Pérez

Director: Arne Fliflet

Acknowledgments

I want to thank my father for introducing me into the world of science and technology and my mother for supporting me in everything I do.

I would also like to acknowledge the contribution of the University of Illinois at Urbana-Champaign, specifically the ECE (Electrical and Computer Engineering) department. Finally, I would also like to acknowledge the Comillas Pontifical University, ICAI, for giving me the chance to study abroad for one year and complete this project during my exchange program.

Autonomous Sailboat

Autor: **Lorenzo Rodríguez Pérez**
Director: Arne Fliflet
Entidad Colaboradora: University of Illinois at Urbana-Champaign

Resumen del Proyecto

Introducción

WRSC (World Robotic Sailing Championship) es una competición de barcos de vela autónomos que tiene como objetivo estimular el desarrollo de la robótica marina autónoma. Para hacer que la navegación autónoma sea más accesible, algunos académicos han creado un diseño educativo genérico [1]. Sin embargo, estos modelos utilizan sistemas de piloto automático costosos, como el controlador de vuelo Pixhawk. La finalidad a largo plazo de este proyecto es desarrollar un barco de vela autónomo capaz de competir en el WRSC. Para ello es necesario comenzar con un diseño capaz de navegar de forma autónoma y dirigido por un control remoto (con capacidad limitada). Por lo tanto, se diseñará un barco de vela asequible que pueda usarse como un medio de aprendizaje de navegación autónoma a menor escala.

Objetivos

Los principales requisitos son los siguientes:

- Permitir al usuario cambiar entre navegar bajo radiocontrol y de forma autónoma. Cuando está en modo autónomo, el velero debería poder mantener el rumbo que tenía justo antes de activar este modo.
- El velero debería poder volver al punto donde se encendió de forma automática si el usuario lo desea (modo retorno a la base).
- El velero debería ser capaz de enviar los datos de los sensores y las posiciones de los servos al sistema de control en tierra.

Permitir cambiar entre modo manual y autónomo ofrece muchos beneficios a los usuarios aficionados, ya que el control de los veleros teledirigidos presenta una pronunciada curva de aprendizaje. Es menos probable que los usuarios pierdan su velero por el viento o las olas cuando están equipados con la funcionalidad autónoma y la función de regreso a la base. Además, la capacidad de monitorizar el estado de la embarcación y los datos de los sensores, pueden ofrecer al usuario un medio para evaluar su control. Esto último también puede dar indicaciones de los problemas y limitaciones del sistema.

Metodología

Diagrama de bloques

En la Figura.1 se puede observar el diagrama de bloques de todo el sistema.

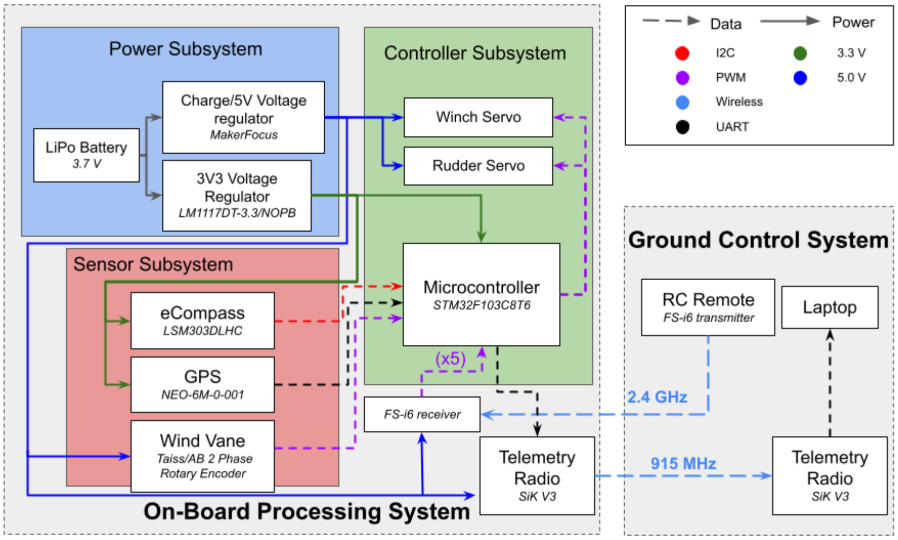


Figura 1: Diagrama de bloques del sistema

Sistema de control en tierra: El sistema de control en tierra envía las señales necesarias para el control y muestra los datos de los sensores y las posiciones de los servos en el portátil del usuario.

Subsistema de potencia: El subsistema de potencia proporciona el voltaje necesario a los diferentes componentes del sistema de procesamiento. La energía proviene de una batería Li-Po de 3V7 que está conectada a un convertidor boost de 5V y un regulador lineal de 3V3. La línea de 5V se conecta al encoder que mide la dirección del viento, al receptor y a la radio de telemetría. La línea 3V3 está conectada al microcontrolador, al compás electrónico y al GPS.

Subsistema de control: El subsistema de control calcula la posición de los servos del timón y de la vela. En modo manual, el microcontrolador emite señales PWM (pulse-width modulation) que dependen de la posición de los joysticks del control remoto (el control remoto utilizado se puede ver en la Fig.2). En los modos autónomo y de retorno a la base, las señales PWM dependen de las entradas de los sensores.

Sensores: El subsistema de sensores está constituido por 3 sensores: el compás electrónico, el GPS y el encoder pegado a la veleta. El compás electrónico emite la dirección actual, el GPS rastrea la posición del velero y el encoder emite la posición relativa al viento.

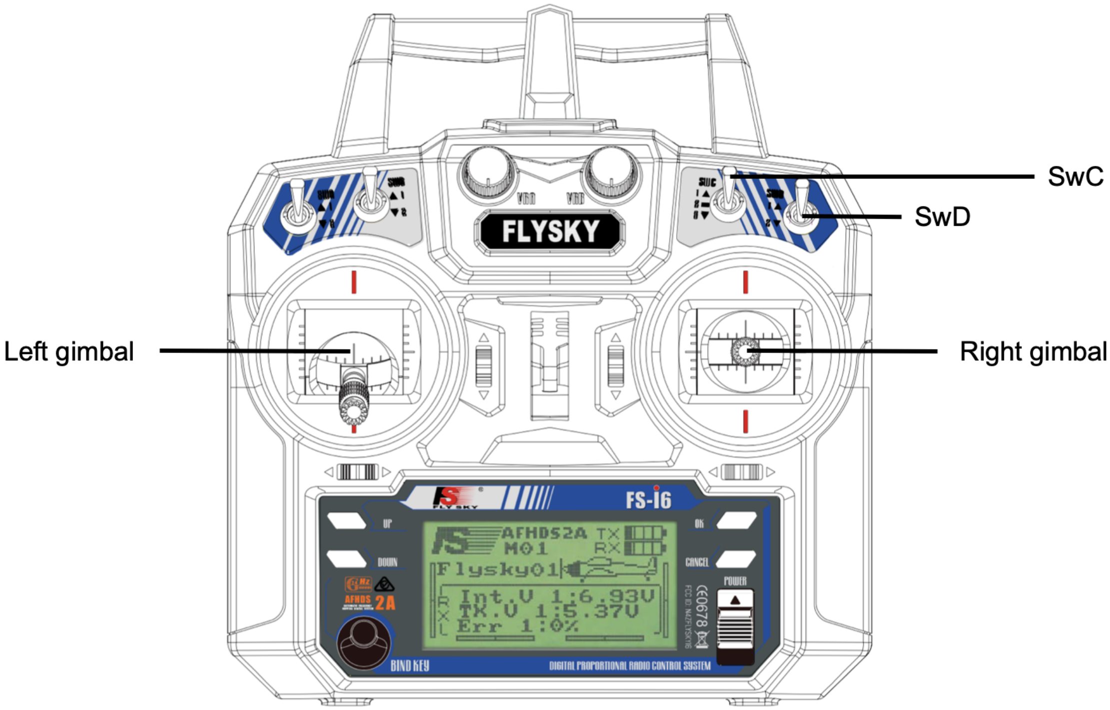


Figura 2: FlySky FS-i6 Remote

PCB

Se diseñó una placa de circuito impreso (PCB) para integrar los diferentes subsistemas del sistema de procesamiento. Son apreciables su esquema y disposición en las figuras 3 y 4 respectivamente.

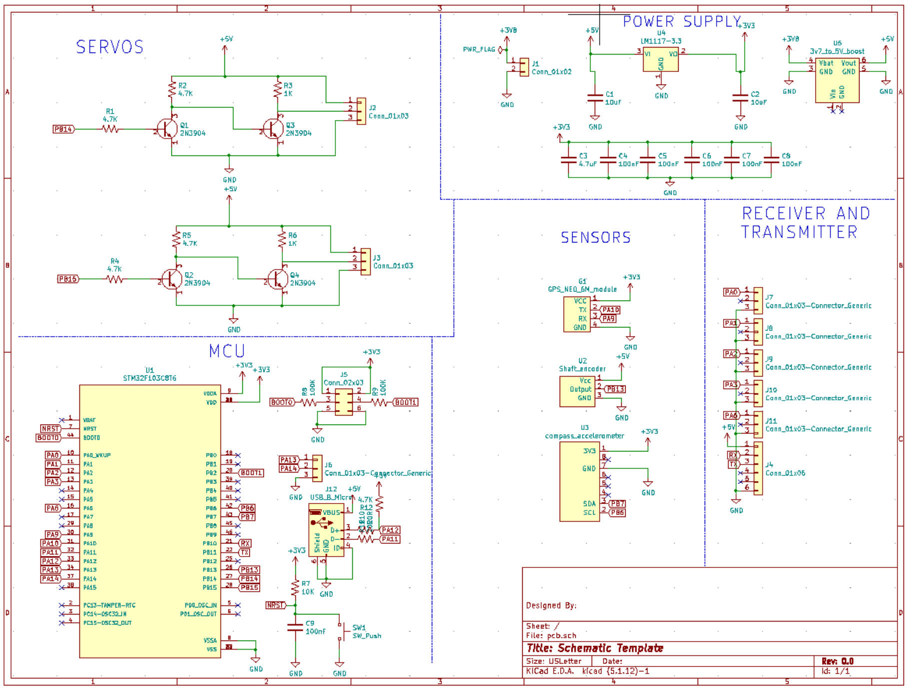


Figura 3: Esquema de la PCB

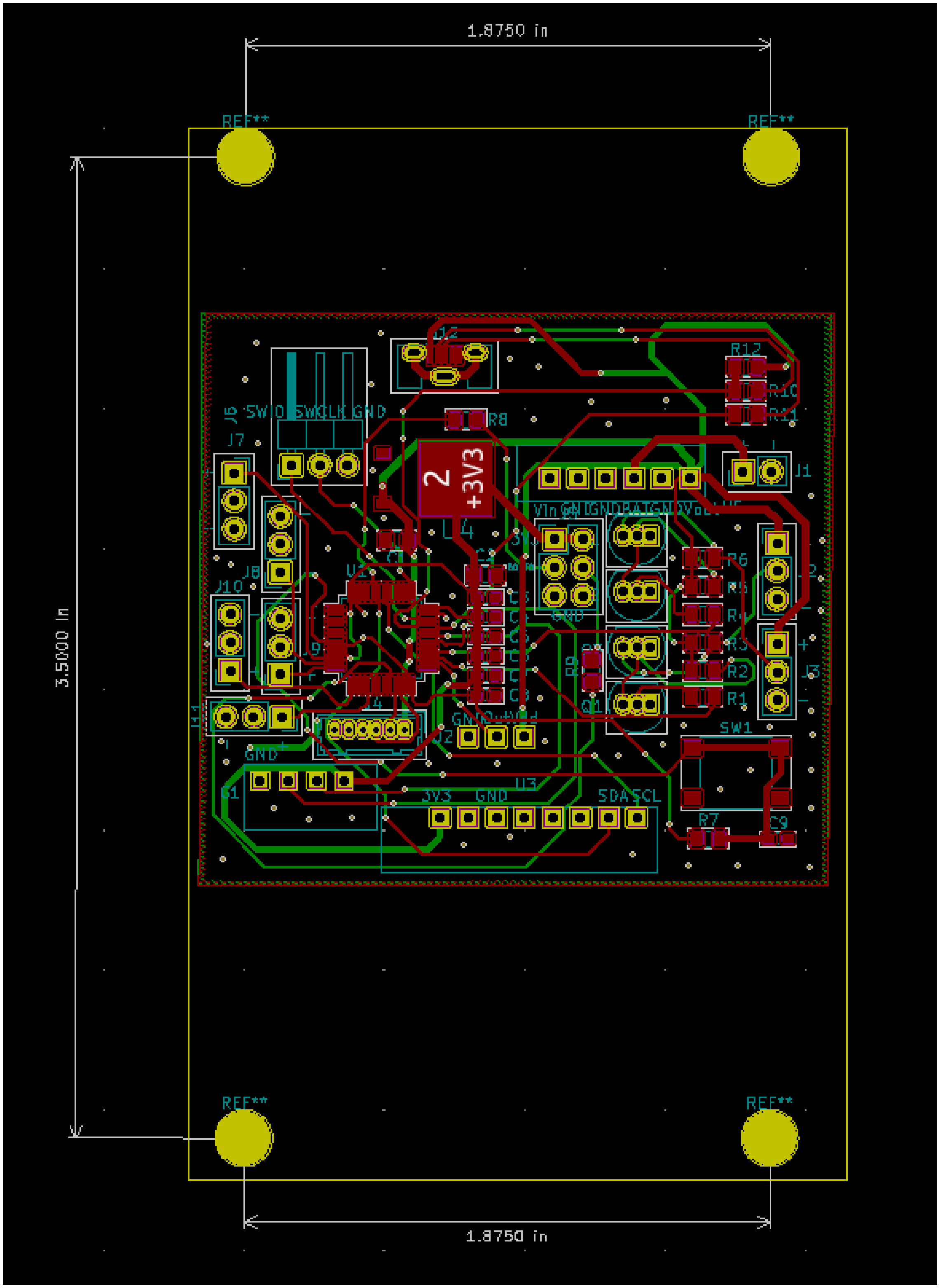


Figura 4: Disposición de la PCB

Diseño del Sistema de Control

Control del Timón El timón se controla mediante un controlador PI que tiene el ángulo de desviación como entrada y el ángulo del timón como salida. Un diagrama de bloques ilustrativo se puede ver en la Figura 5.

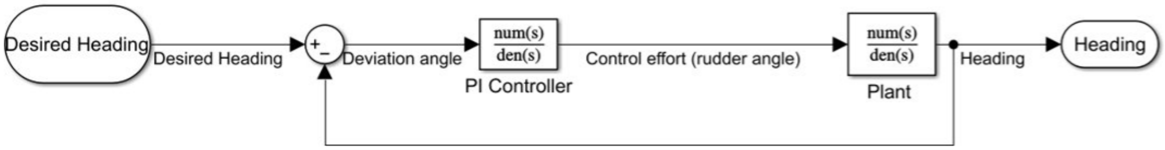


Figura 5: Diagrama de bloques del sistema del timón

Los parámetros asociados al contolador PI (k y k_i, las constantes que multiplican el ángulo de desviación y la integral del ángulo de desviación) se calcularon experimentalmente. El microcontrolador se programó con diferentes constantes hasta que el timón se movió lo suficientemente rápido y en la dirección correcta.

Control de la Vela El control de la vela se implementó utilizando una máquina de estados finitos. La máquina de estados finitos se puede ver en la Figura 6.

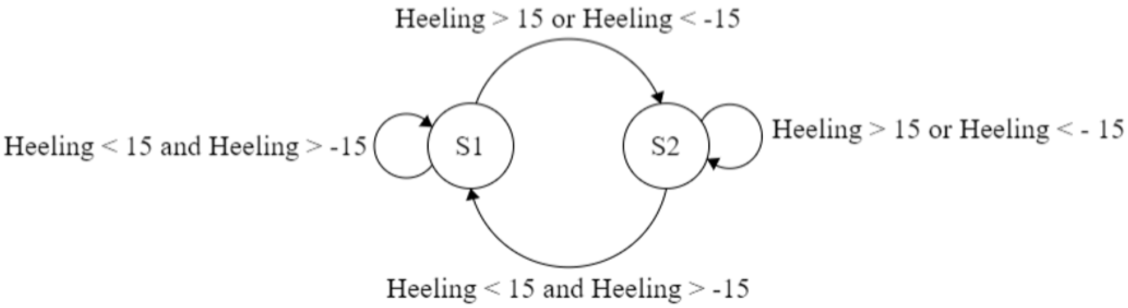


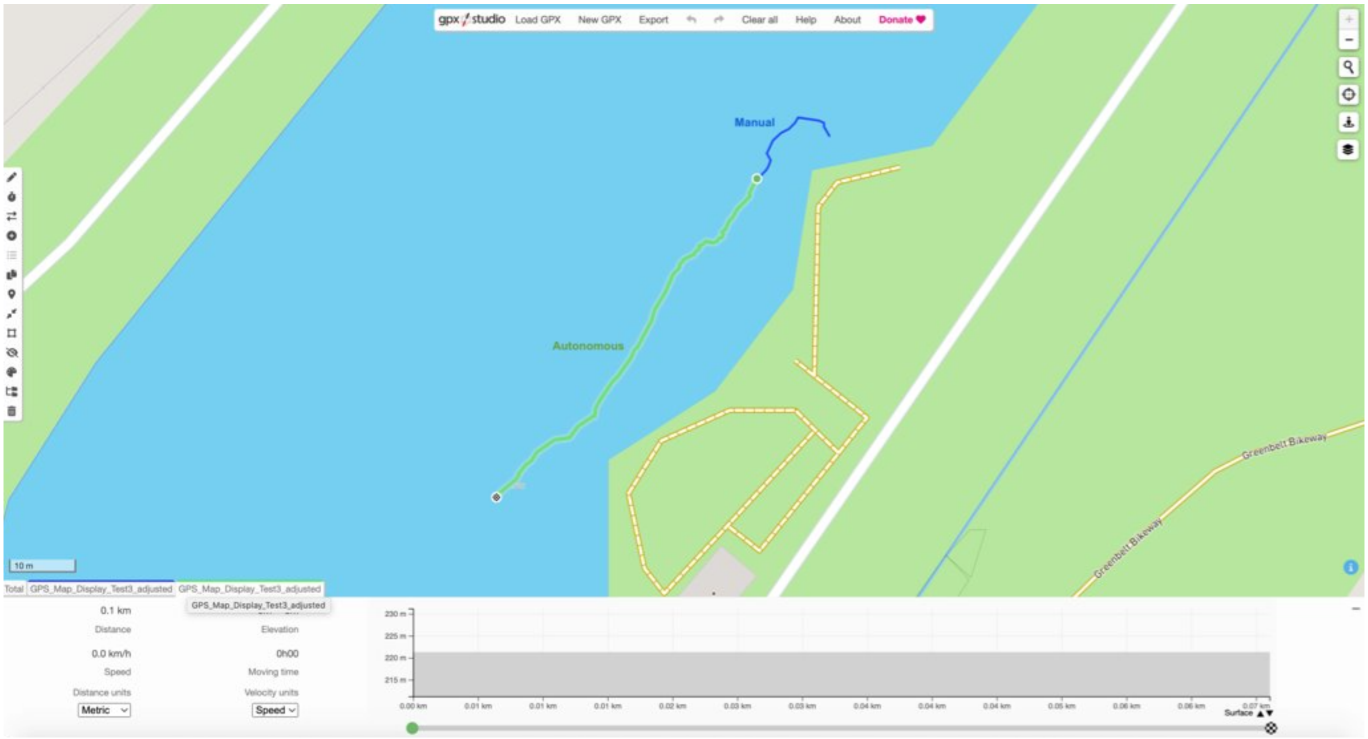
Figura 6: Máquina de estados usada para controlar la vela

En el estado 1, el control de la vela establece la posición del servo en función de la dirección relativa al viento. Utiliza una tabla que almacena el ángulo de vela óptimo en términos de la dirección relativa al viento.

En el estado 2, el controlador comienza a abrir la vela hasta que el ángulo de escora vuelve a un valor absoluto inferior a 15º. Después de eso, vuelve al estado 1.

Resultados

Se cumplieron todos los objetivos y el velero demostró ser capaz de mantener la dirección con precisión, como se puede observar en la Figura 7.



Distance: 0.1 km
Speed: 0.0 km/h
Moving time: 0h00
Distance units: Metric
Velocity units: Speed

Figura 7: Trayectoria del velero (verde: modo autónomo, azul: modo manual)

Conclusiones

Aunque se lograron todos los objetivos, el rendimiento del velero podría ser optimizado:

- El controlador no integraba una parte diferencial. Esto hace que el sistema no pueda tener una buena reacción frente a cambios bruscos en la dirección. Se podría diseñar un diferenciador resistente al ruido para implementar la parte diferencial del control. El sobremuestreo también podría ayudar a reducir el ruido en la entrada del compás electrónico.
- A pesar de que el velero es capaz de mantener su dirección, oscila bastante. Lo más probable es que esto se deba a que las constantes asociadas al control PID son excesivamente altas. Ahora que el sistema está funcionando, el método de Ziegler-Nichols podría usarse para calcular parámetros del control PID que ofrezcan un mejor rendimiento.

Referencias

[1] S. Yang, C. Liu, Y. Liu, J. An y X. Xiang. «Generic and Flexible Unmanned Sailboat for Innovative Education and World Robotic Sailing Championship». (2021), direccion: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7990777/#B18 (visitado 10-02-2022).

Autonomous Sailboat

Author: Lorenzo Rodríguez Pérez.
Director: Arne Fliflet.
Collaborating entity: University of Illinois at Urbana-Champaign.

Project Summary

Introduction

WRSC (World Robotic Sailing Championship) is an autonomous sailing competition that aims at stimulating the development of autonomous marine robotics. The long-term goal of this project is to develop an autonomous sailing boat capable of competing in WRSC. To accomplish that goal, it is necessary to start with a design capable of sailing autonomously and controlled by a remote controller (with limited capability).
In order to make autonomous sailing more accessible, some scholars have created a generic educational design [1]. However, these models utilize expensive and scarce autopilot systems such as the Pixhawk Flight controller. The goal of this project is to make an affordable, user-friendly RC sailboat that can be used as a means of learning autonomous sailing on a smaller scale.

Main Goals

The high level requirements of the sailboat are as follows:

- Dual-mode capability to allow the user to switch between sailing under radio control and autonomously. When in autonomous mode, the sailboat should be able to keep the compass heading it had before autonomous mode was triggered.
- Return to base functionality. The sailboat should be able to return to the point where it was turned on.
- Data communication to a ground control system to monitor the state of the boat.

The dual-mode capability offers many benefits to amateur users as the convoluted steering system on RC sailboats presents a steep learning curve. Users are less likely to potentially lose their sailboat to the wind or waves when equipped with the autonomous functionality and return to base feature. However, in the event that the autonomous system cannot navigate a difficult environment (harsh winds, etc.), the user must have the skill to manually control the boat back to base. Furthermore, the ability to monitor and track the state of the boat and data processed from on-board sensors can offer the user a means of assessing their control. It can also point to where there may be issues within the boat and its limitations.

Design

Block Diagram

See Fig. 8 below for a block diagram overview of the entire system.

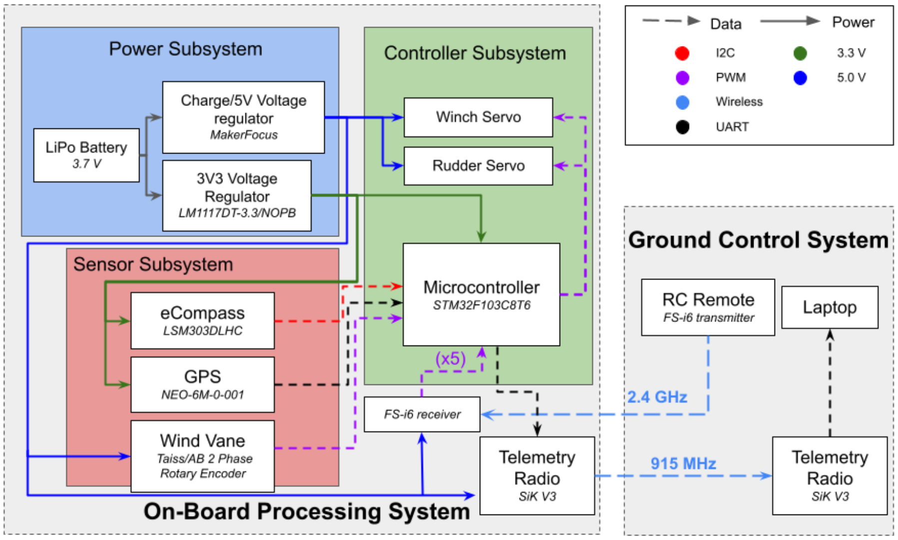


Figure 8: Block Diagram of the system

Ground Control System: The ground control system sends the necessary signals for control through the RC remote and displays the sensor, position and servo data on the user's laptop.

Power Subsystem: The Power subsystem provides the necessary voltage supply to the different components in the On-Board Processing System. The power comes from a 3V7 Li-Po battery that is connected to a 5V boost and a 3V3 linear regulator. The 5V line is connected to the wind vane encoder, the receiver and the telemetry radio. The 3V3 line is connected to the microcontroller, the eCompass and the GPS.

Controller Subsystem: The controller subsystem controls the position of the rudder servo and the sail winch. When in manual mode, the microcontroller outputs PWM signals that depend on the position of the RC joysticks (the remote controller used can be seen in Fig.9). In autonomous and return to base modes, the PWM signals depend on sensor inputs.

Sensor Subsystem: The sensor subsystem is constituted by 3 sensors: eCompass, GPS and wind vane encoder. The eCompass senses the current compass heading, the GPS tracks the sailboat position and the encoder outputs the relative position to the wind.

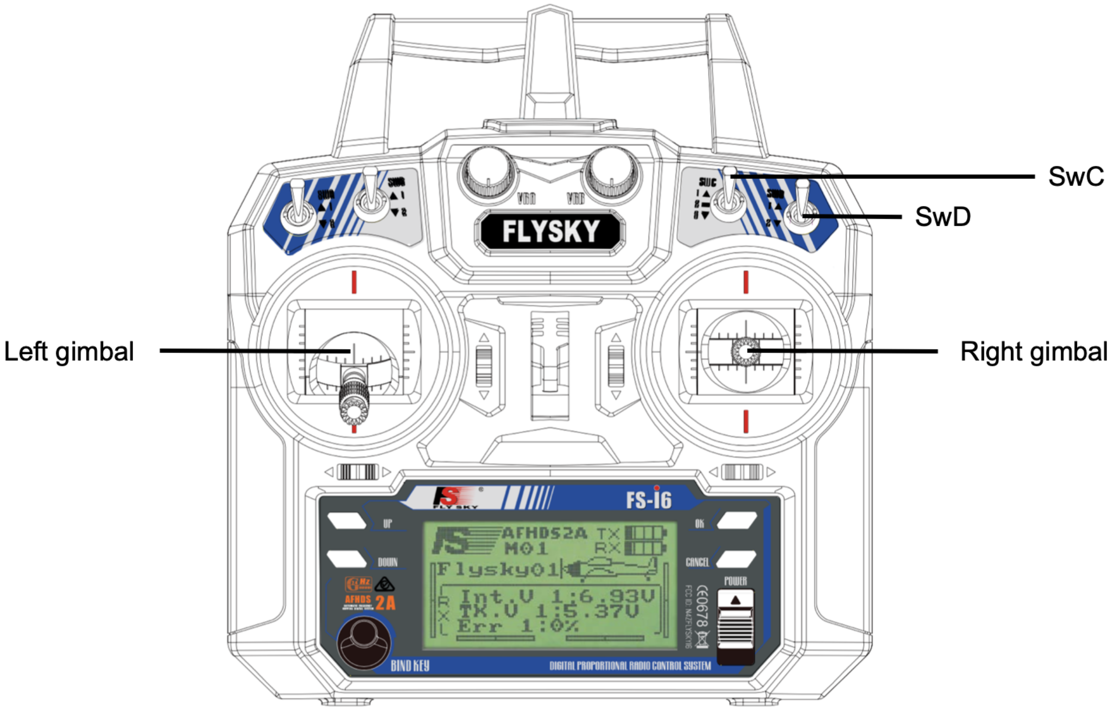


Figure 9: FlySky FS-i6 Remote

PCB

A printed circuit board (PCB) was designed to integrate the different subsystems in the On-Board processing system. You can see its schematic and layout in Fig.10 and Fig.11 respectively.

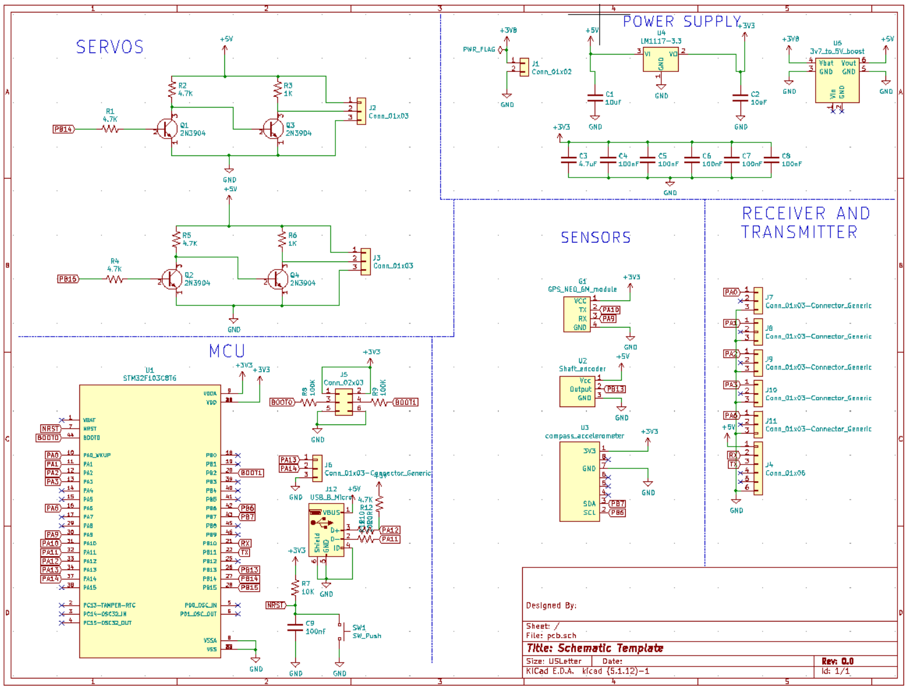


Figure 10: PCB Schematic

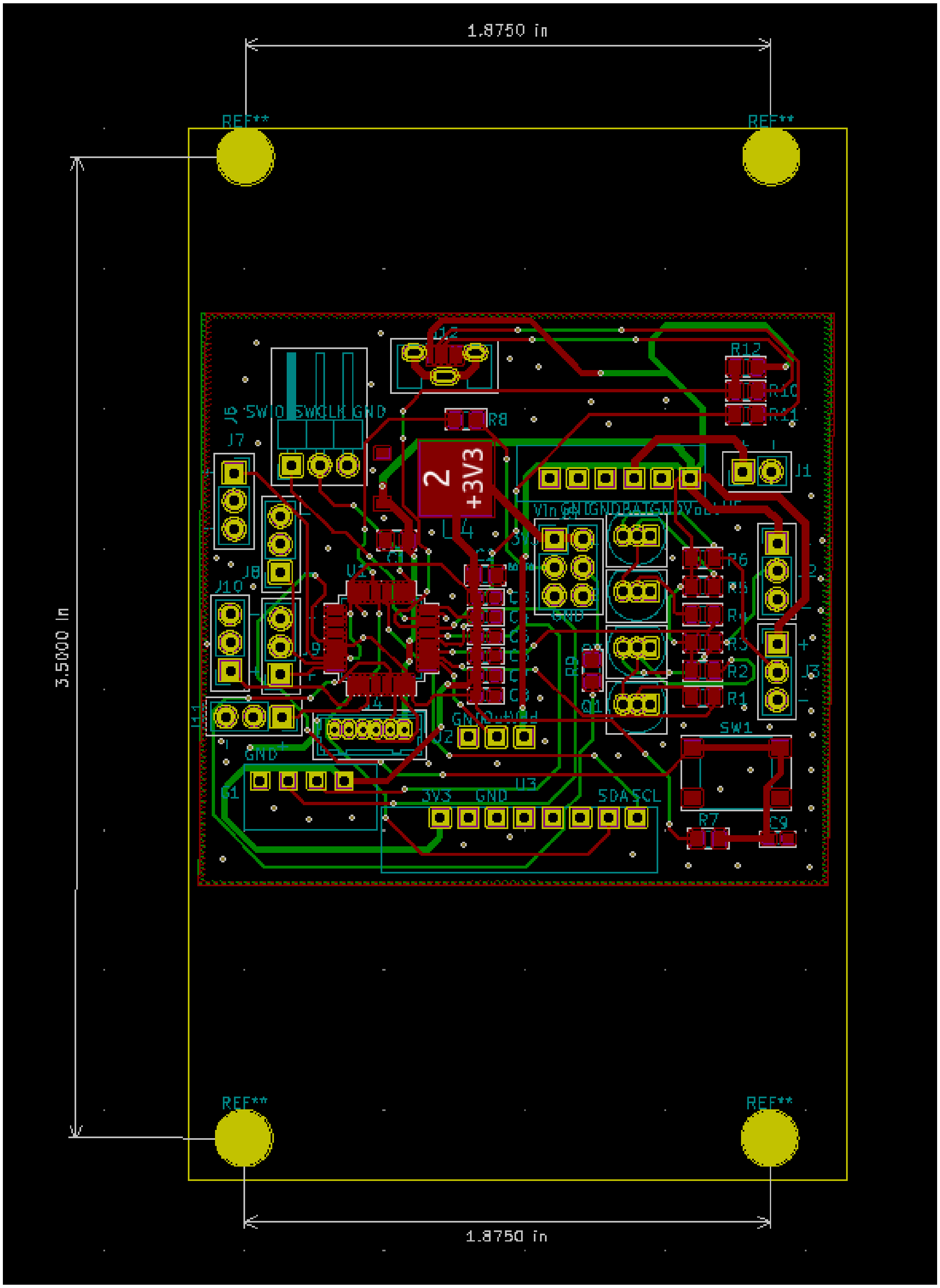


Figure 11: PCB Layout

Controller Design

Rudder Control The rudder is controlled using a PI controller that has the deviation angle as an input and the rudder angle as an output. An illustrative block diagram can be seen in Fig 12.

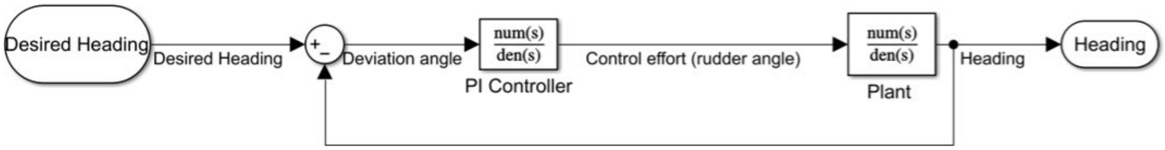


Figure 12: Block diagram of the rudder system

The PI parameters (k and k_i, the constants that multiply the deviation angle and the deviation angle integral) were computed experimentally. The microcontroller was programmed with different constants until the rudder moved quick enough and in the correct direction.

Sail Control The sail winch was programmed using a finite state machine. The finite state machine can be seen in figure 13.

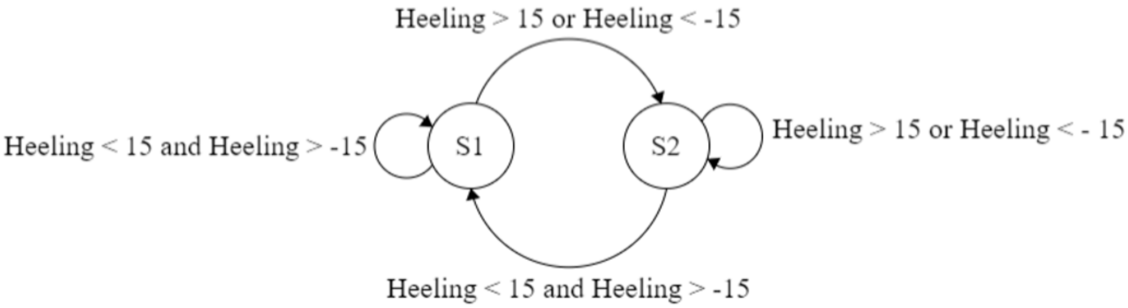


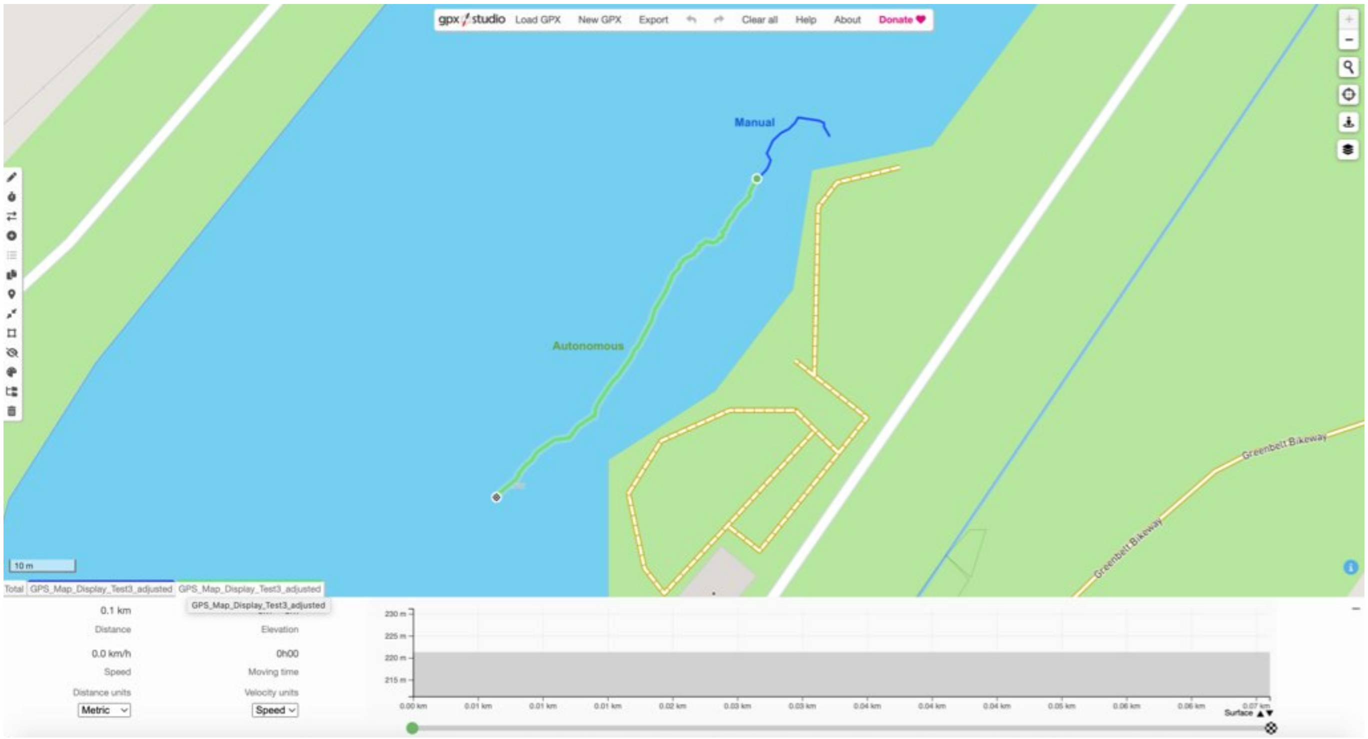
Figure 13: Finite state machine for the sail winch control

When in state 1, the sail winch control sets the servo position depending on the relative wind direction. It utilizes a lookup table that stores the optimal sail angle in terms of the relative wind direction.

When in state 2, the sail winch control starts opening the sail until heeling angle comes back to an absolute value smaller than 15. After that, it returns to state 1.

Results

All high level requirements were met and the sailboat proved to be able to maintain the compass heading accurately, as seen in Fig.14.



Distance: 0.1 km

Elevation

Speed: 0.0 km/h

Moving time: 0h00

Distance units: Metric

Velocity units: Speed

Figure 14: Sailboat movement (green: autonomous mode, blue: manual mode)

Conclusions

Although all high level requirements were achieved, the performance of the sailboat could be further improved.

The controller did not integrate a differential part. This makes the system not able to have a good reaction against compass heading changes that occur in small time frames. A noise robust differentiator could be derived to implement the derivative part of the control. Oversampling could also help reduce the noise in the eCompass input.

In addition, there were some significant oscillations when the sailboat was used. This is most likely due to the high PID constants. Now that the system is working, Ziegler-Nichols Method could be used to derive PID parameters that offer better system performance.

References

[1] S. Yang, C. Liu, Y. Liu, J. An, and X. Xiang. ((Generic and Flexible Unmanned Sailboat for Innovative Education and World Robotic Sailing Championship)). (2021), [On-line]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7990777/#B18 (visited on 02/10/2022).

Abstract

WRSC (World Robotic Sailing Championship) is an autonomous sailing competition that aims at stimulating the development of autonomous marine robotics. In order to make autonomous sailing more accessible, some scholars have created a generic educational design [1]. However, these models utilize expensive and scarce autopilot systems such as the Pixhawk Flight controller. The goal of this project is to make an affordable, user-friendly RC sailboat that can be used as a means of learning autonomous sailing on a smaller scale.

The high level requirements of the sailboat are as follows: dual-mode capability to allow the user to switch between sailing under radio control and autonomously, return to base functionality, data communication to a ground control system to monitor the state of the boat. The dual-mode capability offers many benefits to amateur users as the convoluted steering system on RC sailboats presents a steep learning curve. Users are less likely to potentially lose their sailboat to the wind or waves when equipped with the autonomous functionality and return to base feature. However, in the event that the autonomous system cannot navigate a difficult environment (harsh winds, etc.), the user must have the skill to manually control the boat back to base. Furthermore, the ability to monitor and track the state of the boat and data processed from on-board sensors can offer the user a means of assessing their control. It can also point to where there may be issues within the boat and its limitations.

Contents

List of Figures

List of Tables

1. Introduction

As mentioned previously, this design relies on low-cost sensors that are easy to set up and install into any standard RC sailboat. The affordability of the design contributes to its marketability among amateur users who are not willing to spend the extra money on alternative systems. Thus, the sailboat offers a "return to base" feature that diminishes concern of an amateur losing his/her boat via lack of steering ability.

Users are also able to monitor real-time sensor data that is critical in tuning the autonomous steering feature of the sailboat and potentially useful in understanding manual control. This ground control system receives data from the on-board MCU via a telemetry transceiver. The only communication the operator maintains with the sailboat is via the RC remote control. The remote control features two joysticks for manually steering the boat and 3 buttons for toggling autonomous mode, returning the sailboat to base, and setting a new base position.

The sailboat has 3 modes of operation: manual mode, autonomous mode and return to base. In manual mode, the rudder and sail winch positions are controlled by the operator. In manual mode, using the RC controller, the operator is able to move two servos that control the rudder and sail winch position. In autonomous mode, the sailboat is able to maintain its compass heading upon activation. Finally, in return to base mode, the sailboat navigates to a base position defined by GPS coordinates. This base position is initially set to the location that the boat was first turned on, however, this position may be changed in manual or autonomous mode by the user via a switch on the RC remote.

1.1. Subsystem Overview

See Fig. 15 below for a block diagram overview of the entire system.

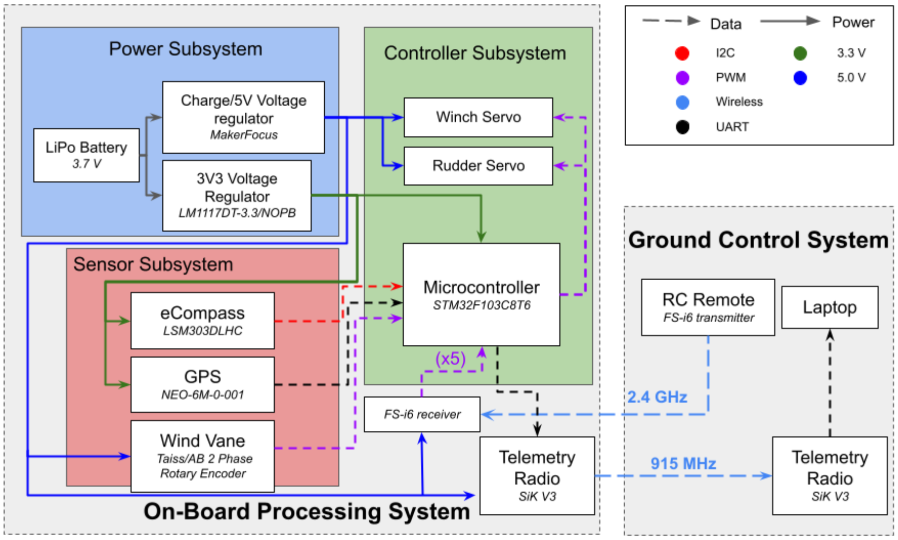


Figure 15: Block Diagram of the system

Ground Control System: The ground control system sends the necessary signals for control through the RC remote and displays the sensor, position and servo data on the user's laptop.

Power Subsystem: The Power subsystem provides the necessary voltage supply to the different components in the On-Board Processing System. The power comes from a 3V7 Li-Po battery that is connected to a 5V boost and a 3V3 linear regulator. The 5V line is connected to the wind vane encoder, the receiver and the telemetry radio. The 3V3 line is connected to the microcontroller, the eCompass and the GPS.

Controller Subsystem: The controller subsystem controls the position of the rudder servo and the sail winch. When in manual mode, the microcontroller outputs PWM signals that depend on the position of the RC joysticks. In autonomous and return to base mode, the PWM signals depend on sensor inputs.

Sensor Subsystem: The sensor subsystem is constituted by 3 sensors: eCompass, GPS and wind vane encoder. The eCompass senses the current compass heading, the GPS tracks the sailboat position and the encoder outputs the relative position to the wind.

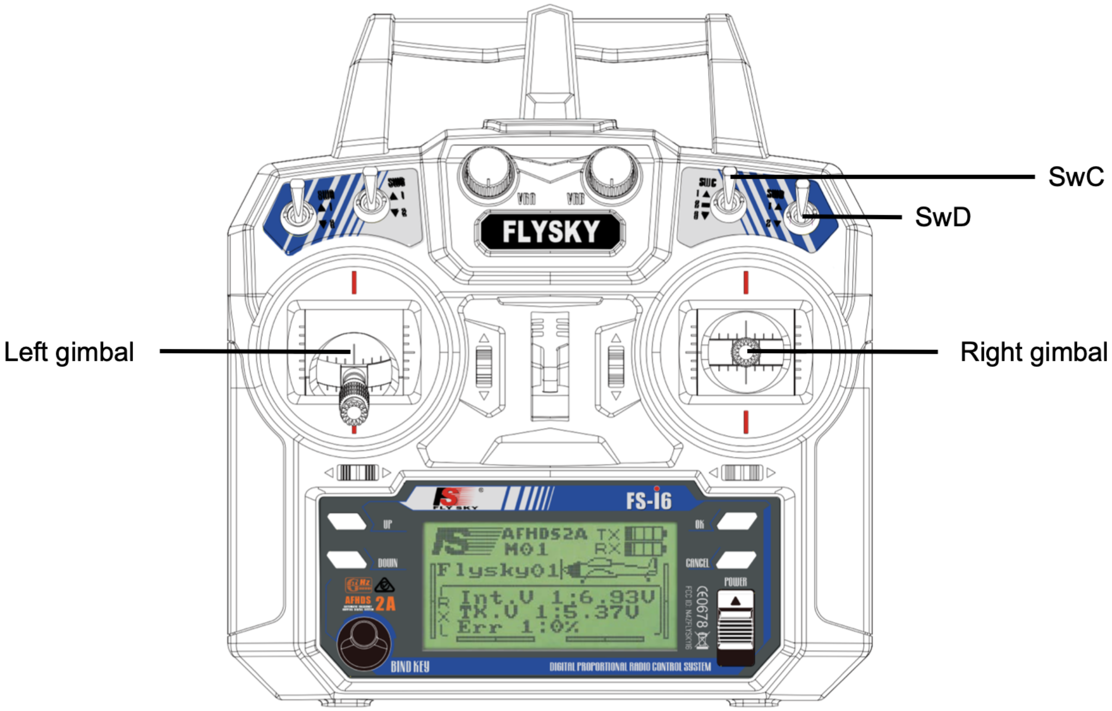


Figure 16: FlySky FS-i6 Remote

2. Design

2.1. Equations and simulations

A simple mathematical model is used to ensure the control can make the sailboat stay within the desired compass heading range. The model relies on the following assumptions:

- Waves are ignored.
- The modulus of the lift force in the sail is approximately constant.
- The modulus of the force in the rudder is approximately constant.
- The roll and pitch angles are approximately 0.
- Yaw angle variation does not affect the forces, as its variation is approximately 0.
- The movement of the sail barely changes the center of mass of the system, G.

A sailboat has different points of sail depending on its relative direction to the wind. These points of sail can be seen in Fig. 17 below.

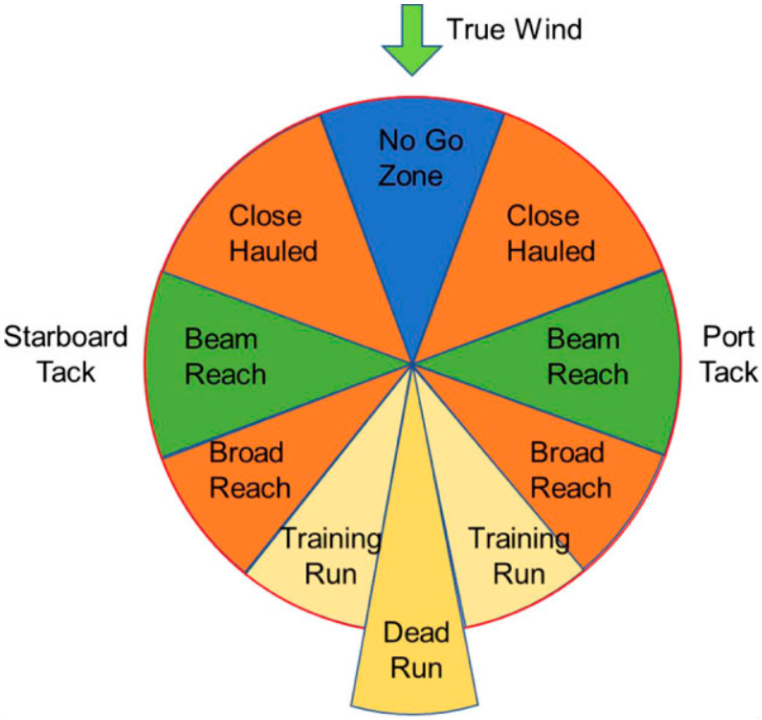


Figure 17: Points of sail [1]

Simulations were carried for every point of sail assuming the sail was correctly placed and using a PID controller in the rudder to keep the desired compass heading.

This is a free body diagram of the sailboat taking into account the assumptions 2.1:

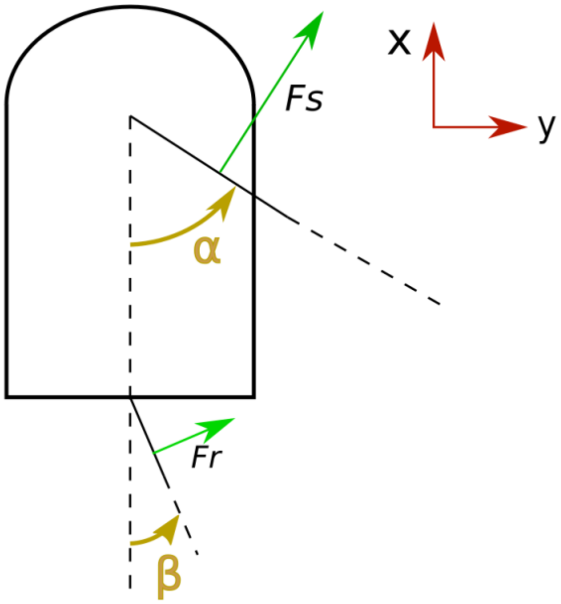


Figure 18: Free body diagram of the sailboat

With this body diagram, a mathematical model for the sailboat can be easily derived using Newton's laws for rotation and movement as shown below 2.1 (note the directions of forces F_r and F_s change when $\beta < 0$ and $\alpha < 0$ respectively).

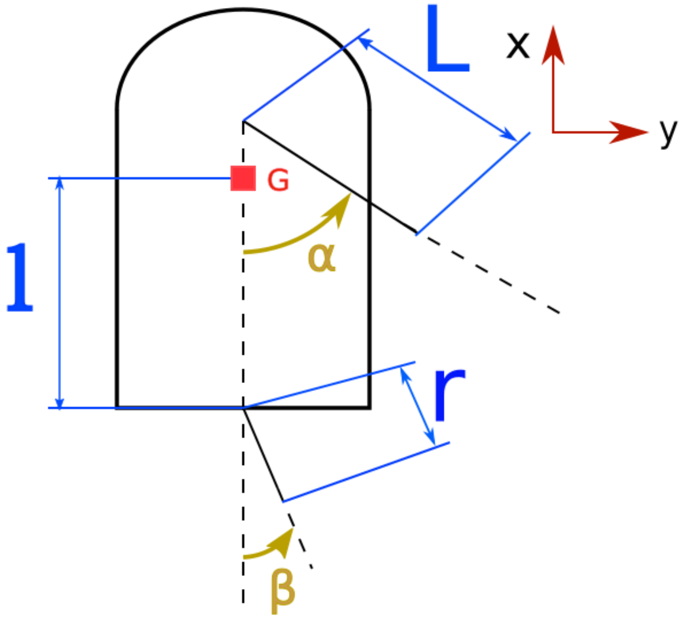


Figure 19: Depiction of variables L, l and r

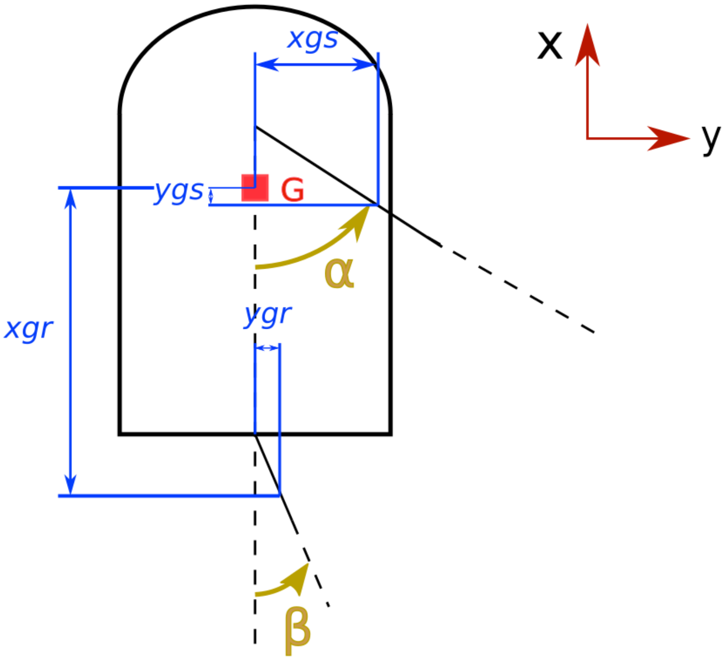


Figure 20: Depiction of variables $x_g r$, $x_g s$, $y_g r$ and $y_g s$

$$
\begin{aligned}
if\,(\alpha, \beta > 0) : x &:= F_s|\sin\alpha| - F_r|\sin\beta| = m\frac{d^2x}{dt^2} \\
y &:= F_s|\cos\alpha| - F_r|\cos\beta| = m\frac{d^2y}{dt^2} \\
F_s|\cos\alpha|x_{gs} + F_s|\sin\alpha|y_{gs} - F_r|\cos\beta|x_{gr} - F_r|\sin\alpha|y_{gr} &= M\frac{d^2\phi}{dt^2} \\
if\,(\alpha < 0, \beta > 0) : x &:= F_s|\sin\alpha| - F_r|\sin\beta| = m\frac{d^2x}{dt^2} \\
y &:= -F_s|\cos\alpha| - F_r|\cos\beta| = m\frac{d^2y}{dt^2} \\
-F_s|\cos\alpha|x_{gs} - F_s|\sin\alpha|y_{gs} - F_r|\cos\beta|x_{gr} - F_r|\sin\alpha|y_{gr} &= M\frac{d^2\phi}{dt^2} \\
if\,(\alpha < 0, \beta < 0) : x &:= F_s|\sin\alpha| - F_r|\sin\beta| = m\frac{d^2x}{dt^2} \\
y &:= -F_s|\cos\alpha| + F_r|\cos\beta| = m\frac{d^2y}{dt^2} \\
-F_s|\cos\alpha|x_{gs} - F_s|\sin\alpha|y_{gs} + F_r|\cos\beta|x_{gr} + F_r|\sin\alpha|y_{gr} &= M\frac{d^2\phi}{dt^2} \\
if\,(\alpha > 0, \beta < 0) : x &:= F_s|\sin\alpha| - F_r|\sin\beta| = m\frac{d^2x}{dt^2} \\
y &:= F_s|\cos\alpha| + F_r|\cos\beta| = m\frac{d^2y}{dt^2} \\
F_s|\cos\alpha|x_{gs} + F_s|\sin\alpha|y_{gs} + F_r|\cos\beta|x_{gr} + F_r|\sin\alpha|y_{gr} &= M\frac{d^2\phi}{dt^2}
\end{aligned}
$$

Where m is the mass of the sailboat, M is the moment of inertia of the sailboat, F_s and F_r are the forces in the sail and in the rudder, α is the sail angle, β is the rudder angle, x_gs is the distance between the center of mass and the point of application of F_s in the x-axis, y_gs is the distance between the center of mass and the point of application of F_s in the y-axis, x_gr is the distance between the center of mass and the point of application of F_r in the x-axis, y_gr is the distance between the center of mass and the point of application of F_r in the y-axis, L is the length of the sail, l is the distance between the center of mass and the rudder and r is the length of the rudder and ϕ is the compass heading deviation angle ($\Phi_{initial} - \Phi_{actual}$).

$$
\begin{aligned}
x_{gs} &:= \frac{L}{2}|\cos\alpha| \\
y_{gs} &:= \frac{L}{2}|\sin\alpha| \\
x_{gr} &:= l + \frac{r}{2}|\cos\beta| \\
y_{gr} &:= \frac{r}{2}|\sin\beta|
\end{aligned}
$$

To come up with an approximate solution for the differential equations, the equations 1 - 6 were used.

$$F = m\frac{d^2x}{d^2t} \tag{1}$$

$$v(t) = \int_0^t \frac{F(t)dt}{m} \approx \int_0^{t-t_0} \frac{F(t-t_0)dt}{m} + \frac{F(t)t_0}{m} \tag{2}$$

$$x(t) = \int_0^t v_x(t)dt \approx \int_0^{t-t_0} v_x(t)dt + v_x(t-t_0)t_0 \tag{3}$$

$$y(t) = \int_0^t v_y(t)dt \approx \int_0^{t-t_0} v_y(t)dt + v_y(t-t_0)t_0 \tag{4}$$

$$w(t) = \frac{\int_0^t T(t)dt}{M} \approx \frac{\int_0^{t-t_0} T(t)dt}{M} + \frac{T(t-t_0)t_0}{M} \tag{5}$$

$$\phi(t) = \int_0^t w(t)dt \approx \int_0^{t-t_0} w(t)dt + w(t-t_0)t_0 \tag{6}$$

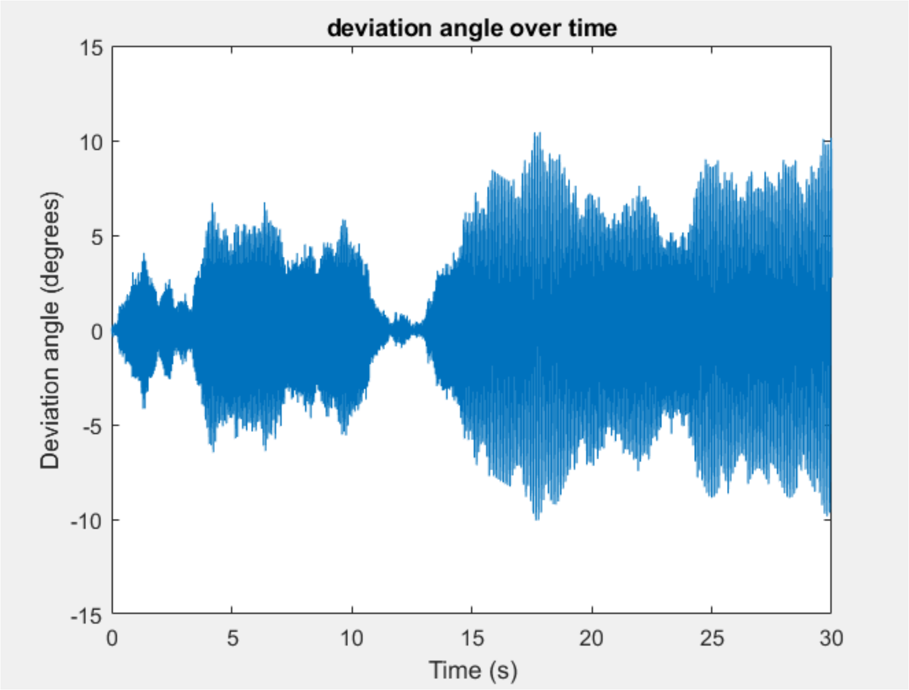


Figure 21: Deviation angle over time while running dead run

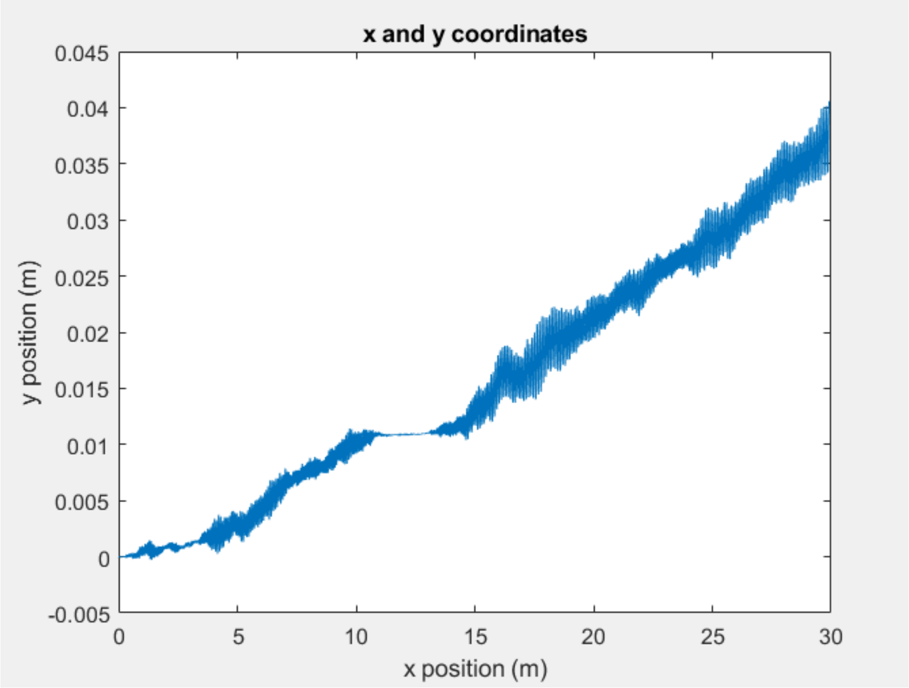


Figure 22: Boat position while running dead run

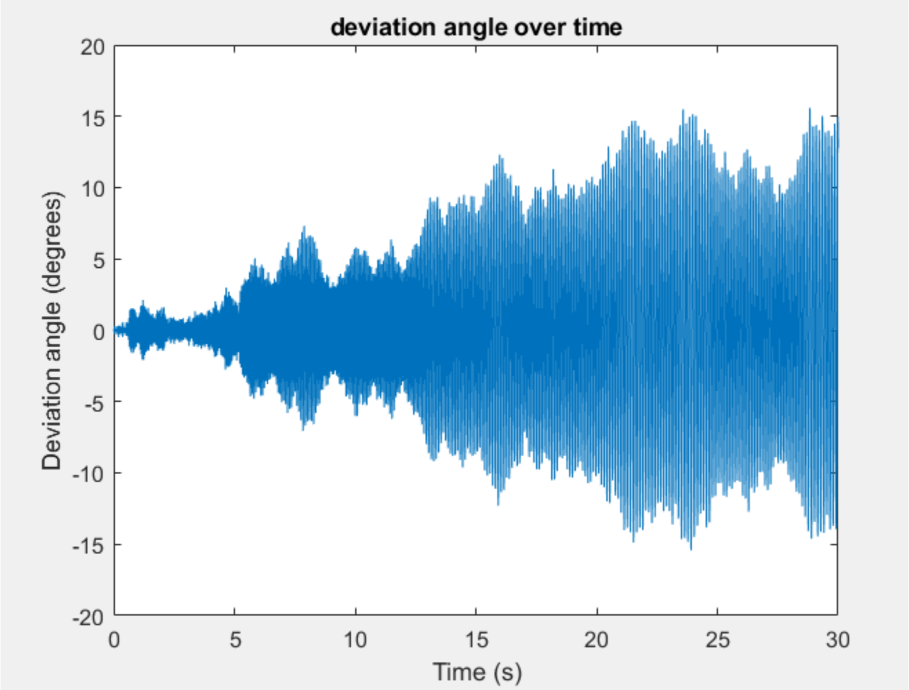


Figure 23: Deviation angle over time while running training run

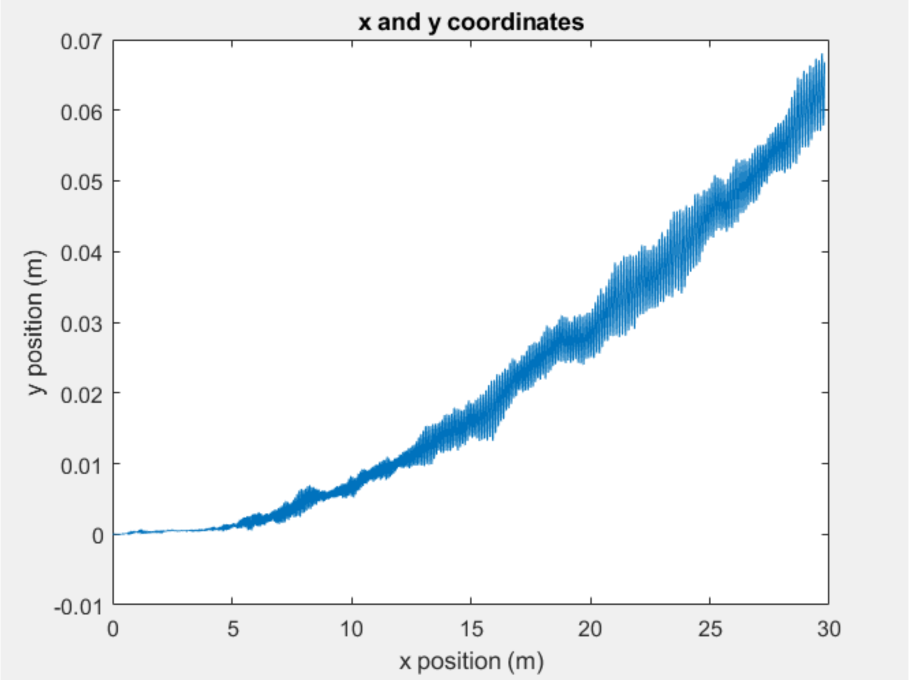


Figure 24: Boat position while running training run

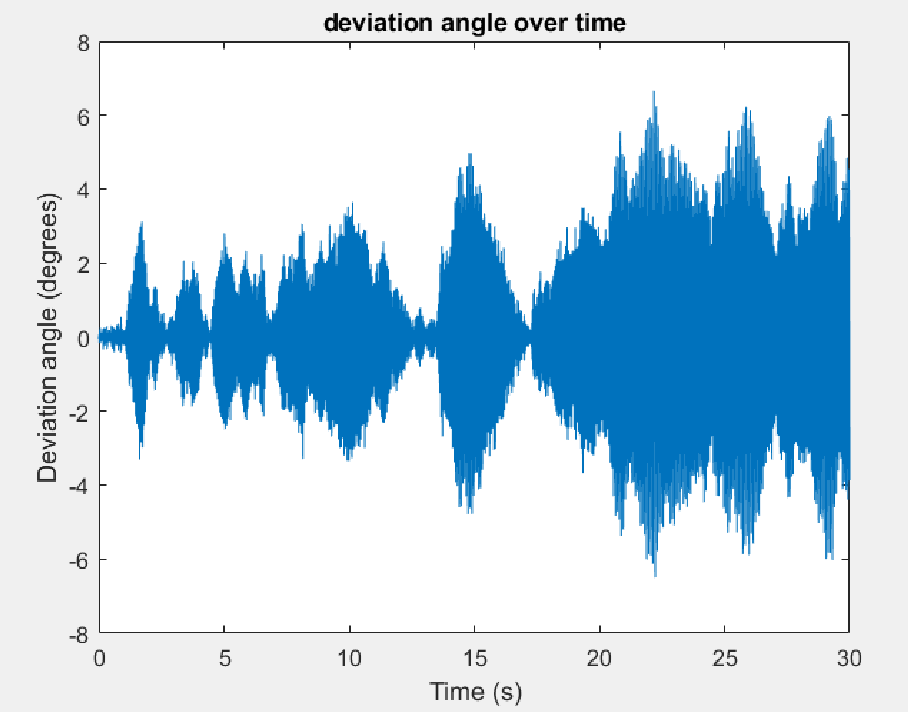


Figure 25: Deviation angle over time while running broad reach

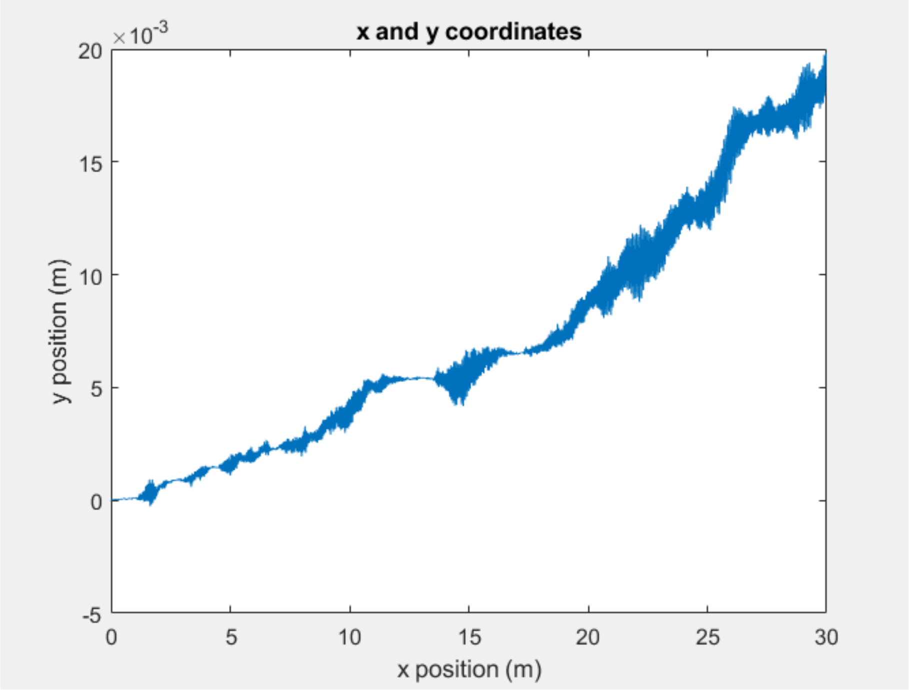


Figure 26: Boat position while running broad reach

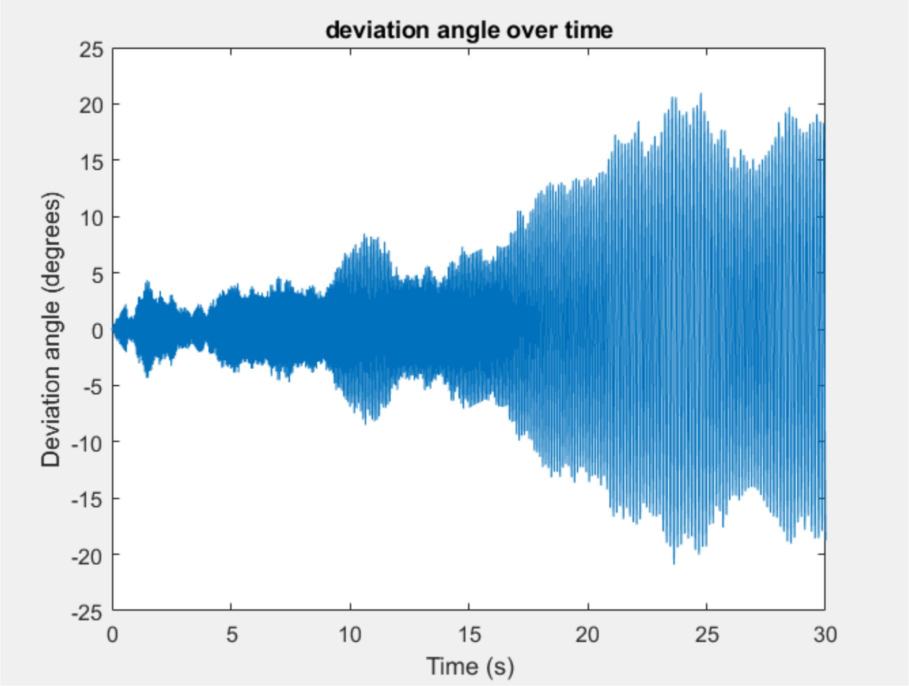


Figure 27: Deviation angle over time while running beam reach

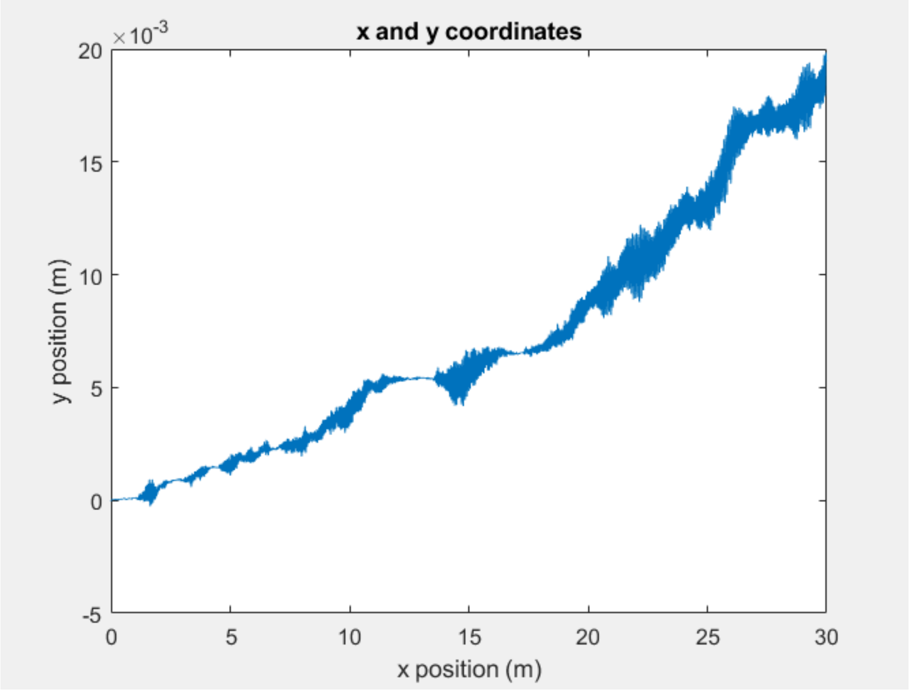


Figure 28: Boat position while running beam reach

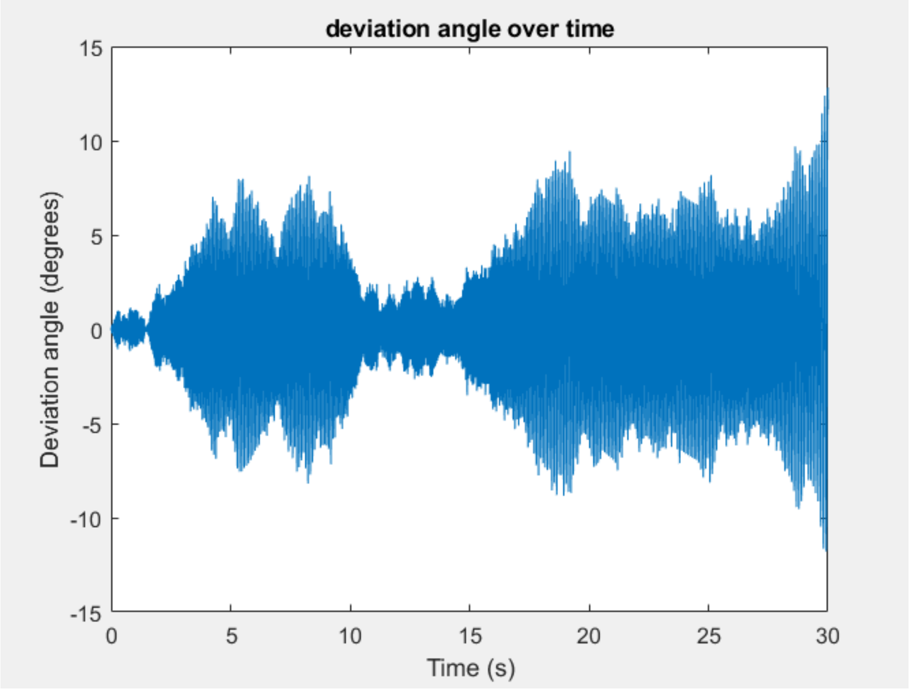


Figure 29: Deviation angle over time while running closed hauled

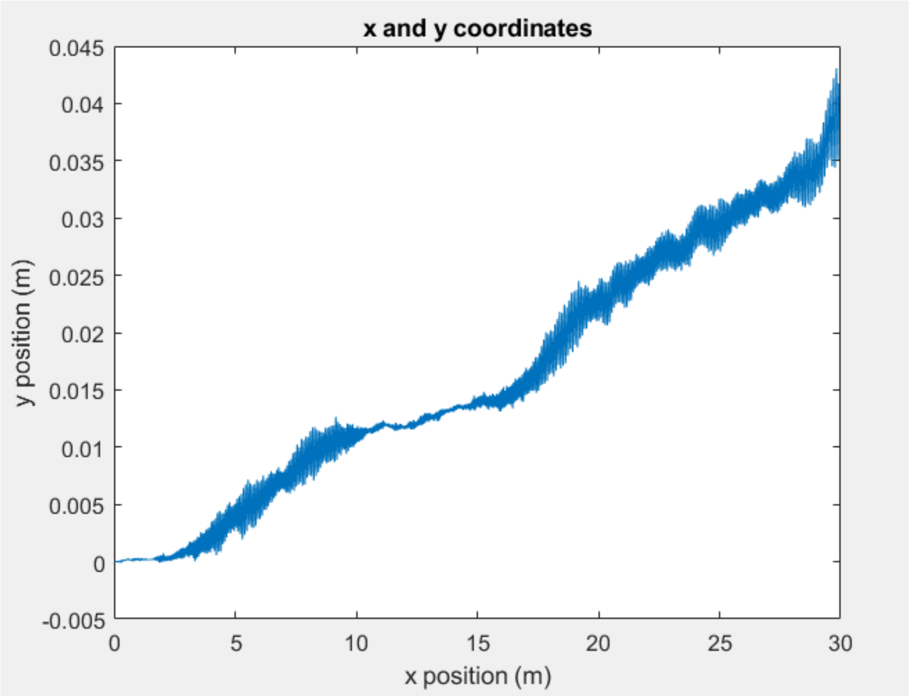


Figure 30: Boat position while running closed hauled

In all simulations, the deviation angle and the y axis movement is small. Therefore, the control can make the sailboat stay within the desired compass heading. However, in the final design a PI controller is used instead of a PID controller due to excess noise within the eCompass signal. This is further explained in section 2.2.

2.2. Design Alternatives

2.2.1. Non-PID Rudder Control

Various control methods for the rudder were discussed and tested throughout the duration of the project. The first rudimentary approach consisted of changing the position of the rudder to its maximum or minimum angle if the absolute value of the deviation angle was bigger than a given tolerance value. After running some simulations for this control method, it was determined to be unstable as shown in Fig. 31 below.

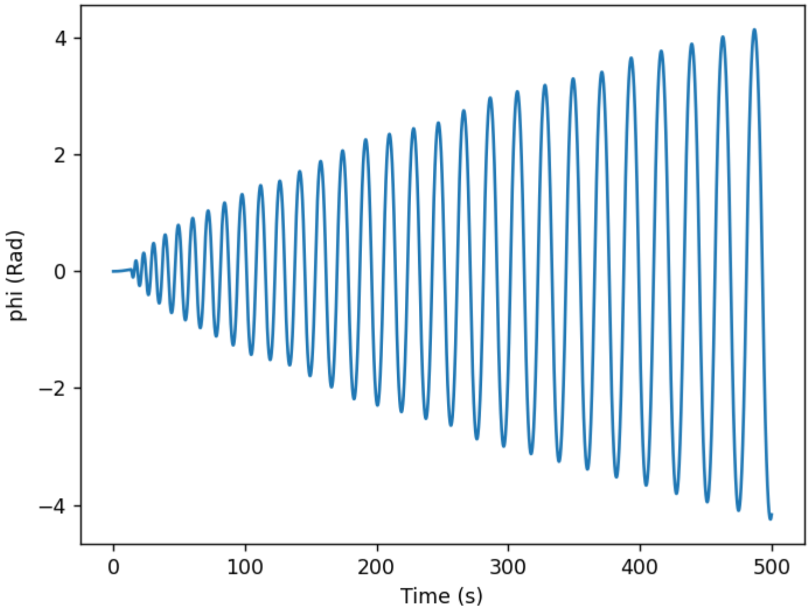


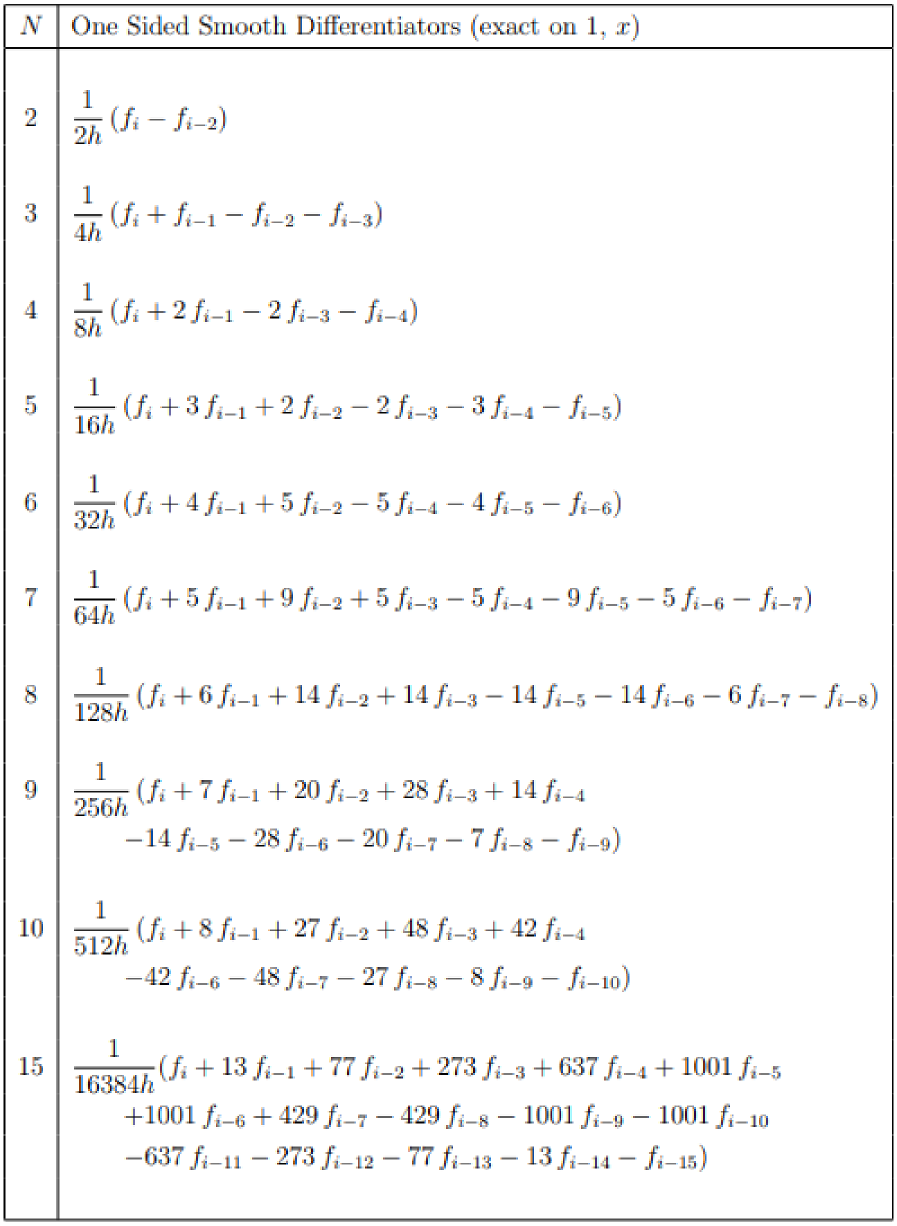
Figure 31: Phi (deviation angle) vs time for a non-PID control approach

As the control reacts to deviations it successfully turns the rudder such that the sailboat comes back to the desired compass heading. However, the deviations explode and the system is unable to stabilize. Note the model is only valid for small ϕ deviations, which means that this plot is only realistic at the beginning. Once ϕ becomes too large the model is not valid and the system is unstable.

2.2.2. Complete PID with derivative

Noisy signals pose a problem for control methods that include a derivative. The derivative of high frequency signals is higher than the derivative of low frequency ones, consequently, derivatives highly amplify noise. The amplification of this noise can make the control unstable.

One sided smooth differentiators can be used to filter out the noise from a derivative. When adjusting the PID controller, even the highest order filter in Fig. 32 was not successful at filtering the noise from our eCompass heading. Although the control was stable, the rudder continually oscillated when the deviation angle was minimal. This is unfavorable as it overdrives the servo and can potentially cause small deviations from the desired control when in water. The PI controller was performing just as efficiently as the PID controller and without the unwanted oscillation, so it was decided to use this instead in the final design.



N	One Sided Smooth Differentiators (exact on 1, x)
2	$\frac{1}{2h}(f_i - f_{i-2})$
3	$\frac{1}{4h}(f_i + f_{i-1} - f_{i-2} - f_{i-3})$
4	$\frac{1}{8h}(f_i + 2\,f_{i-1} - 2\,f_{i-3} - f_{i-4})$
5	$\frac{1}{16h}(f_i + 3\,f_{i-1} + 2\,f_{i-2} - 2\,f_{i-3} - 3\,f_{i-4} - f_{i-5})$
6	$\frac{1}{32h}(f_i + 4\,f_{i-1} + 5\,f_{i-2} - 5\,f_{i-4} - 4\,f_{i-5} - f_{i-6})$
7	$\frac{1}{64h}(f_i + 5\,f_{i-1} + 9\,f_{i-2} + 5\,f_{i-3} - 5\,f_{i-4} - 9\,f_{i-5} - 5\,f_{i-6} - f_{i-7})$
8	$\frac{1}{128h}(f_i + 6\,f_{i-1} + 14\,f_{i-2} + 14\,f_{i-3} - 14\,f_{i-5} - 14\,f_{i-6} - 6\,f_{i-7} - f_{i-8})$
9	$\frac{1}{256h}(f_i + 7\,f_{i-1} + 20\,f_{i-2} + 28\,f_{i-3} + 14\,f_{i-4}$ $-14\,f_{i-5} - 28\,f_{i-6} - 20\,f_{i-7} - 7\,f_{i-8} - f_{i-9})$
10	$\frac{1}{512h}(f_i + 8\,f_{i-1} + 27\,f_{i-2} + 48\,f_{i-3} + 42\,f_{i-4}$ $-42\,f_{i-6} - 48\,f_{i-7} - 27\,f_{i-8} - 8\,f_{i-9} - f_{i-10})$
15	$\frac{1}{16384h}(f_i + 13\,f_{i-1} + 77\,f_{i-2} + 273\,f_{i-3} + 637\,f_{i-4} + 1001\,f_{i-5}$ $+1001\,f_{i-6} + 429\,f_{i-7} - 429\,f_{i-8} - 1001\,f_{i-9} - 1001\,f_{i-10}$ $-637\,f_{i-11} - 273\,f_{i-12} - 77\,f_{i-13} - 13\,f_{i-14} - f_{i-15})$

Figure 32: One Sided Smooth Differentiators [2]

2.3. Control Design Description

2.3.1. Rudder Control

The rudder is controlled using a PI controller that has the deviation angle as an input and the rudder angle as an output. An illustrative block diagram can be seen in Fig 33.

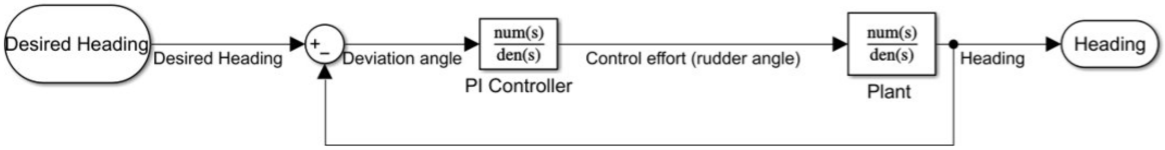


Figure 33: Block diagram of the rudder system

The PI parameters (k and ki, the constants that multiply the deviation angle and the deviation angle integral) were computed experimentally. The microcontroller was programmed with different constants until the rudder moved quick enough and in the correct direction.

2.3.2. Sail Winch Control

The sail winch was programmed using a finite state machine. The finite state machine can be seen in figure 34.

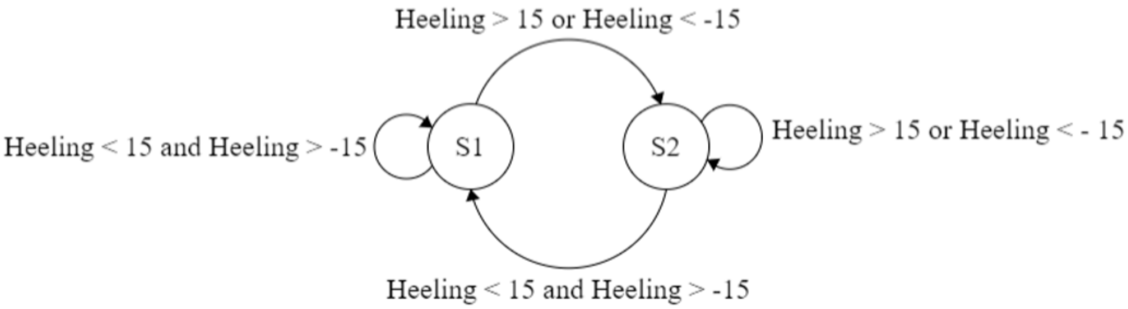


Figure 34: Finite state machine for the sail winch control

State 1: When in state 1, the sail winch control sets the servo position depending on the relative wind direction. It utilizes the lookup table that can be seen in Table 1.

Table 1: Sail Angle Lookup Table

Apparent Wind Angle	Point of Sail	Sail Angle
$0 \leq \theta \leq 45 \| \| 315 \leq \theta \leq 360$	No-Go Zone	0°
$45 \leq \theta \leq 75$	Close-Hauled	15°
$75 \leq \theta \leq 105$	Beam Reach	-45°
$105 \leq \theta \leq 135$	Broad Reach	-60°
$135 \leq \theta \leq 225$	Running	±90°
$225 \leq \theta \leq 255$	Broad-Reach	60°
$255 \leq \theta \leq 285$	Beam Reach	45°
$285 \leq \theta \leq 315$	Close-Hauled	15°

The control stays in state 1 unless the absolute value of heeling angle is bigger than 15°.

State 2: When in state 2, the sail winch control starts opening the sail until heeling angle comes back to an absolute value smaller than 15. After that, it returns to state 1.

2.4. PCB

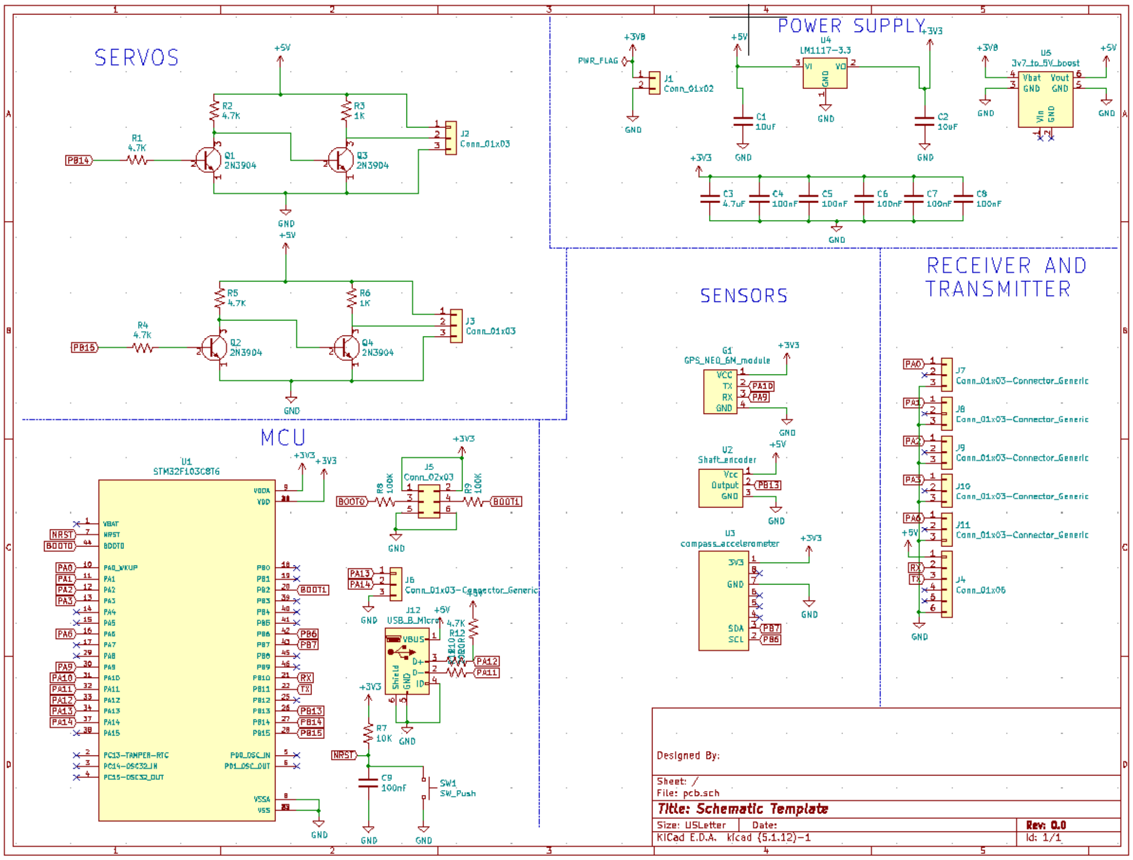


Figure 35: PCB Schematic

Level Shifters: The microcontroller outputs are 3V3 signals, and the servos work with 5V. To fix this issue, two level shifter circuits were added in the PCB. These boost the signal from 3V3 to 5V. The level shifters are shown in Fig. 36.

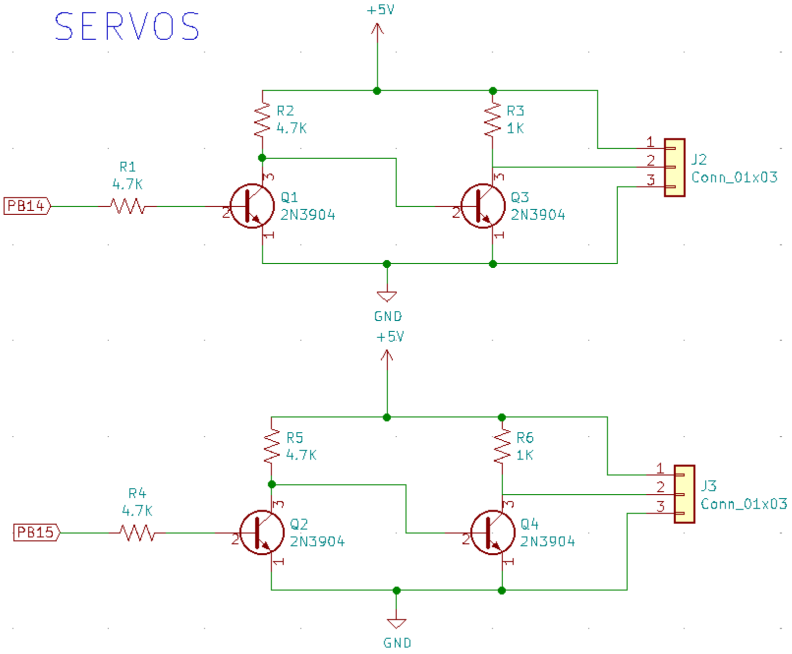


Figure 36: Level Shifters Schematic

Power Supply: As mentioned in the introduction, the power supply uses a linear regulator and a boost converter to change the input voltage to 3V3 and 5V. There are 10uF capacitors placed in the input and output of the 3V3 linear regulator because the datasheet recommended putting them. The capacitors between the 3V3 net and ground are placed to ensure a stable input voltage for the microcontroller. The Power Supply schematic is shown in Fig. 37.

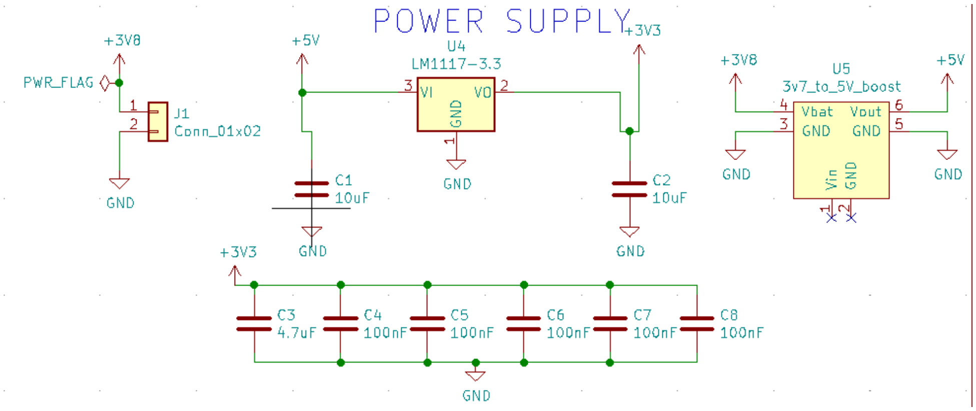


Figure 37: Power Module Schematic

Sensors: Footprints for the sensors were created for the sensors that the system uses (shatf encoder, GPS and compass). As you can see in Fig. 38, the shaft encoder works at

5V while the GPS and the compass work at 3V3, which is one of the reasons why both 3V3 and 5V regulators are needed.

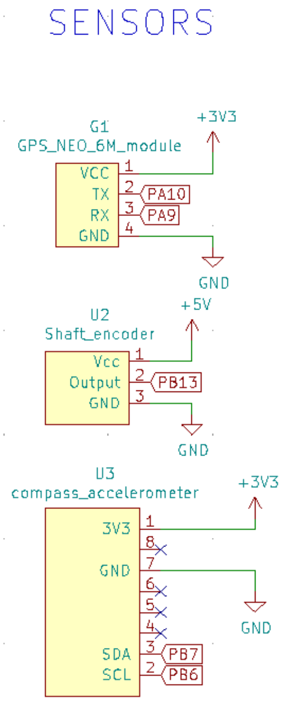


Figure 38: Sensors schematic

MCU Module: The microcontroller module contains a reset button, the microcontroller and the connections needed to program the microcontroller. The microcontroller module schematic is shown in Fig. 39.

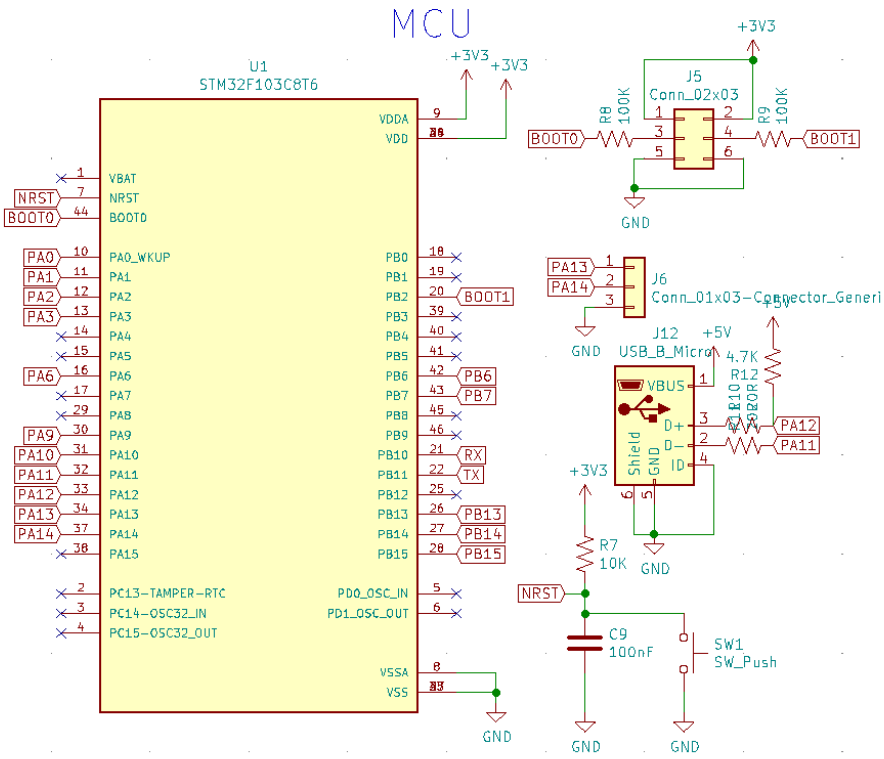


Figure 39: MCU Module Schematic

Receiver and Transmitter Connections: All the connectors needed for communications can also be found in the PCB. The schemati can be seen in Fig. 40.

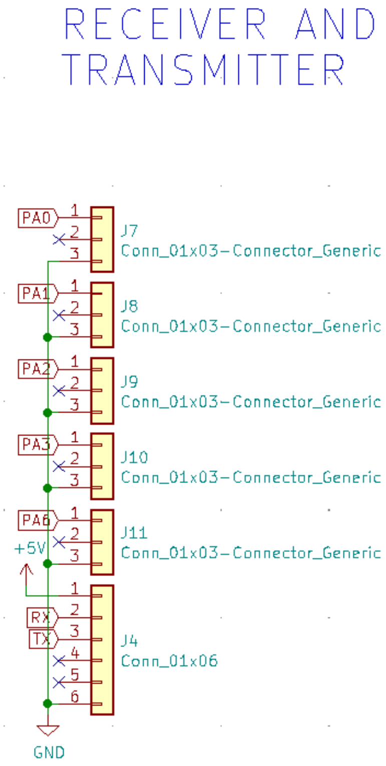


Figure 40: Receiver and Transmitter Connections Schematic

3. Requirements and Verifications

Table 2: On-Board Processing System Requirements and Verification Pt 1.

Requirements	Verification
▪ The processing system must prevent the user from activating the return to base feature when it is not in autonomous mode. Furthermore, the processing system must prevent the operator from controlling the rudder and winch servos when in autonomous mode.	▪ Place the sailboat on the lazy susan, turn on the box fan simulating the wind. Turn on the sailboat and flip the autonomous mode switch ON. Move the two joysticks to control the rudder and winch servo. Verify that the servos do not respond to the movement of the joysticks.

Table 3: On-Board Processing System Requirements and Verification Pt 2.

▪ The processing system correctly adjusts the rudder angle and the sail angle according to Table 1.	▪ Turn on the sailboat and activate autonomous mode. Place a box fan directly in front of the lazy susan such that the wind direction is directly ”upwind” of the sailboat. Check the sail angle matches the table. Repeat this for the different sailing points in Table 1 ▪ Rotate the sailboat to both sides and check that the angle of the rudder reaches its maximum after no more than 5 seconds.

Table 4: Ground Control Subsystem Requirements and Verification

Requirements	Verification
▪ The ground control system displays updated sensor data with a feedback delay ≤ 1 s.	▪ Turn on sailboat and note base position on laptop display. Turn off sailboat and repeat at a location approximately 10 m away from starting location. Then, turn on sailboat and walk back to starting position; verify the GPS coordinates match initial measurement within 1 s. ▪ Place sailboat on lazy susan such that its heading lines up with 0°. Rotate the sailboat 90°, 180°, and -90°. Verify that the compass heading reads 90°, 180° and -90° respectively with a 5° tolerance.

3.0.1. Results

It was checked that the sail servo adjusted its angle according to Table 1. It was also measured that the rudder was able to reach saturation in 4.5 s.

4. Cost and Schedule

Cost: The average salary of an ECE graduate is $79,714/year [3]. The average person works 2080 hours per year. $79,714/2080 = $38.32/hour. Each member of the group will work an average of 10 hours per week in the project. 10 hours/week * 10 weeks = 100 hours. Therefore, the total labor cost of this project comes to a total of **$11,496**.

This project will take the machine shop about 15 hours. According to UIUC's machine shop website, the average pay is $36.65/hr plus materials [4]. Therefore, the total machine shop cost is 15hr x $36.65/hr = **$549.75**.

The total parts cost comes to **$251.83** (See Appendix 6.1).

Total project cost = Parts + Machine Shop + Labor = $251.83 + $549.75 + $11,496 = **$12297.58**.

Schedule: Refer to Appendix 6.1 for the devised schedule for all team members.

5. Ethical Considerations

There are a few ethics policies that need to be taken into consideration with this project. Section 7.6 of the IEEE Code of Ethics I.5 states, "to seek, accept, and offer honest criticism of technical work... and to credit properly the contributions of others" [5]. As this project is not the first design for an autonomous sailboat, the team has credited sources from previous projects and credited resources used in the design [1]. This project is a challenging assignment for the members of the team; the team made use constructive criticism along the way.

Furthermore, there are a few safety concerns needed to address. As the sailboat has an on-board power supply, it was ensured that the casing of the power subsystem was completely waterproof and did not pose any risk for electrical shock. It was also ensured that wire connections from the servos to the waterproof casing are robustly secured to resist vibration and rolling as the sailboat may face on-board water exposure. Finally, ground control system application allows users to monitor the sensor data from the sailboat. Such data as the GPS coordinates of the boat, and hence user, poses a risk to their privacy. We ensured that this application protects and does not monitor the user's data. Through ensuring safety we abide to uphold IEEE standards I.1; "to hold paramount, the safety, health, and welfare of the public... and to protect the privacy of others" [5].

Finally, the team ensures to follow Lab Safety guidelines in testing circuits and sensors. The team also followed COVID-19 CDC recommended safety guidelines when meeting in person to work in the project.

6. Conclusions

All high level requirements were met and the sailboat proved to be able to maintain the compass heading accurately, as seen in Figure 41.

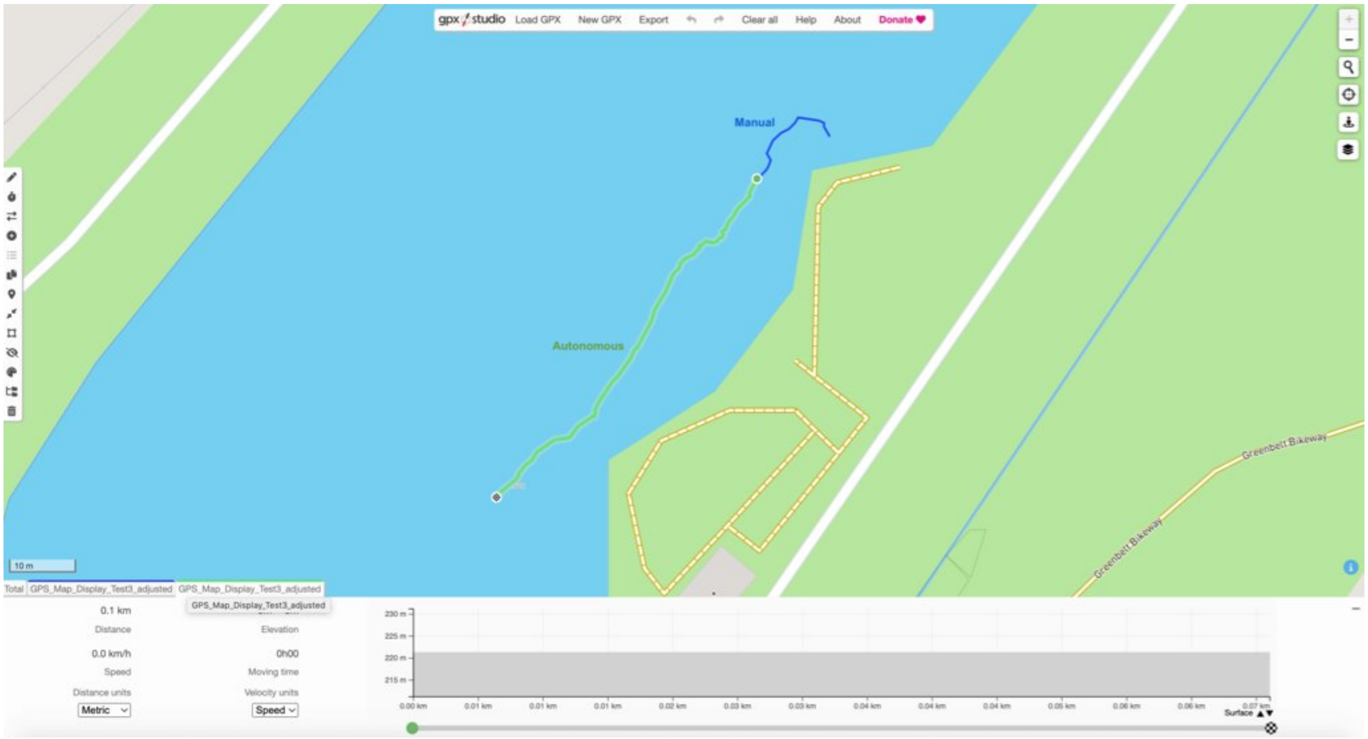


Figure 41: Sailboat movement (green: autonomous mode, blue: manual mode)

As described in Section 2.2, the controller did not integrate a differential part. This makes the system not able to have a good reaction against compass heading changes that occur in small time frames. Additionally, the system was able to keep the compass heading quite accurately (maximal deviation of 12°), despite some significant oscillations.

6.1. Future Work/Alternatives

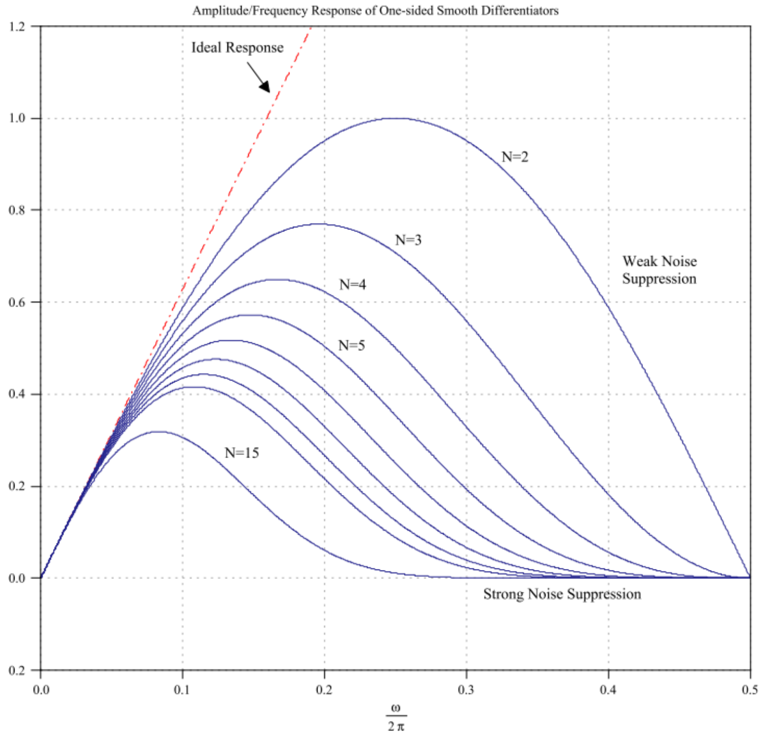


Figure 42: Amplitude/Frequency Response of One-sided Smooth Differentiators [2]

Filtered Derivative: A stronger filter than those shown in Figure 42 could be used to filter the noise out from the derivative and integrate the differential control. A possible expression for the filter, with two poles in $-w_p$, three poles in $-10 * w_p$ and four poles in $-100 * w_p$ can be seen in Equation 7.

$$\frac{s}{(1+\frac{s}{w_p})^2 * (1+\frac{s}{10*w_p})^3 * (1+\frac{s}{100*w_p})^4} \tag{7}$$

As seen in Fig. 43 the filter mimics the behavior of a derivative until it reaches the frequency w_p, after that, it suppresses noise. Using a low enough w_p, this filter could help integrate the differential part into the controller. An equivalent transfer function in the z domain could be computed using the zero order hold method. After that, a difference equation to compute the filtered derivative could be easily derived.

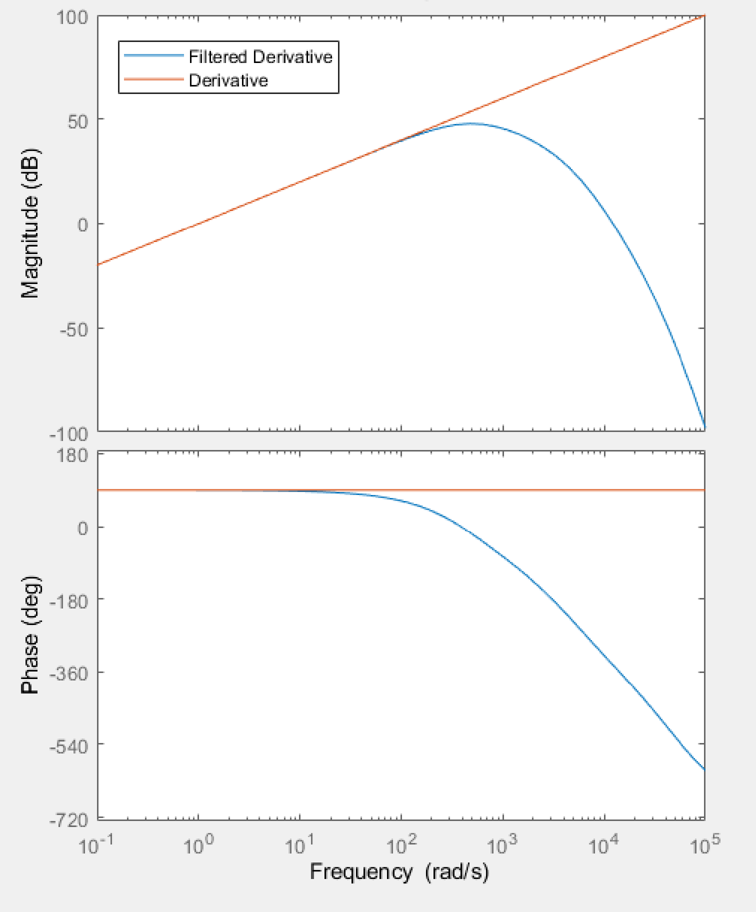


Figure 43: Bode plot for the filter and pure derivative (wp = 500 rad/s)

Oversampling: An alternative method to reduce noise from the eCompass is oversampling. By reducing the period, a mean of different values provided by the eCompass could be computed. This mean value would have less noise, since the average of gaussian noise is 0.

Ziegler-Nichols Method: The sailboat is able to keep its compass heading when in autonomous mode, but it oscillated substantially. This is most likely due to the high PID constants. Now that the system is working, Ziegler-Nichols Method could be used to derive PID parameters that offer better system performance.

References

[1] S. Yang, C. Liu, Y. Liu, J. An, and X. Xiang. «Generic and Flexible Unmanned Sailboat for Innovative Education and World Robotic Sailing Championship». (2021), [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7990777/#B18 (visited on 02/10/2022).

[2] P. Holoborodko. «One-Sided Differentiators». (2009), [Online]. Available: http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/ (visited on 04/30/2022).

[3] G. C. of Engineering. «Salary Averages». (), [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages (visited on 02/24/2022).

[4] S. of Chemical Sciences at Illinois. «Machine Shop». (), [Online]. Available: https://scs.illinois.edu/resources/cores-scs-service-facilities/machine-shop (visited on 02/24/2022).

[5] IEEE. «IEEE Code of Ethics». (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

Appendix A Schedule and Cost

Table 5: Weekly Schedule

Date	Tasks Overview	Riley	Lorenzo	Arthur
2/21	Design Document Check, Order Parts	STM32 research, Design Document	Circuit Schematic, PCB Design	Physical Design, Design Document
2/28	Design review, PCB board reviews	Sensor module unit testing, PCB design	Power module unit testing, PCB design	Controller module unit testing
3/14	Individual autonomous software design	STM32Cube setup, HAL library	Servo control with BluePill	Sensor input with BluePill
3/21	Finish design using BluePill dev board	Sensors and Servo STM32Cube Libraries	Sensors and Servo STM32Cube Libraries	Sensors and Servo STM32Cube Libraries
3/28	Implement return to base feature with BluePill dev board	STM32Cube Autonomous mode libraries	Configuring servo module with BluePill	Configuring controller module with BluePill
4/4	Transfer implementation to PCB	Programming PCB	Constructing boat with PCB	Soldering PCB
4/11	Adjust sensors and algorithms	Optimize servo adjustment algorithms	Optimize efficiency of servos	Optimize efficiency of controller
4/11- 5/2	Mock demo, demonstration, presentation, final paper	Final adjustments, Final Paper	Final adjustments, Final Paper	Final adjustments, Final Paper

Table 6: Parts Cost

Part	Manufacturer	Part Number	Quantity	Extended Cost
Microcontroller	STMicroelectronics	STM32F103C8T6	1	$7.00
Sail Winch Servo	Joysway Hobby	880545	1	$25.95
Rudder Servo	Joysway Hobby	881504	1	$9.95
BJT Transistor	Micro Commercial Co	2N3904-AP	4	$1.36
4.7 KΩ Resistor	YAGEO	RC1206FR-104K7L	4	$0.40
1 KΩ Resistor	YAGEO	RC1206FR-101KL	2	$0.20
3 Pin Male Header	Molex	0022284036	4	$0.74
5 V Converter	MakerFocus	B07PZT3ZW2	1	$2.35
3V3 Regulator	Texas Instruments	LM1117DT-3.3/NOPB-ND	1	$1.89
2 Pin Male Header	Molex	0022284028	3	$0.87
10 µF Capacitor	Smasung Electro-Mechanics	CL21B106KPQNFNE	2	$0.58
4.7 µF Capacitor	Smasung Electro-Mechanics	CL10A475KQ8NNWC	1	$0.10
100 nF Capacitor	KEMET	C0603C104K8PAC7867	5	$1.00
GPS Module	Hiletgo	GY-NEO6MV2	1	$17.49
Wind Vane Encoder	US Digital	MA3-P10-125-B	1	$60.22
eCompass	HiLetgo	GY-511 LSM303DLHC	1	$7.99
ARM 10 Pin Connector	Amphenol CS	G821EU210AGM00Y	1	$0.75
Telemetry Radio	Holybro	SiK V3 17012	1	$56.00
RC Controller	FlySky	FS-i6 6CH	1	$38.00
Receiver	FlySky	FS-iA6	1	$18.99

Appendix B Sustainable Development Goals

This project contributes to the goal 13 (climate action) of the sustainable development goals. 40.4 million metric tons of carbon dioxide were emitted by ships and boats in the USA in 2019. Sailboats do not emit carbon dioxide because they are powered by the wind. In addition, the electronic system that controls autonomous sailboats could be powered by solar cells, giving them infinite endurance without emitting carbon dioxide [1].