ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Autonomous Sailboat

Team #2

RILEY BAKER
(rileymb32@illinois.edu)
LORENZO RODRIGUEZ
(lr12@illinois.edu)
ARTHUR LIANG
(chianl2@illinois.edu)

TA: Evan Widloski

May 4, 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

1 Introduction 1
1.1 Subsystem Overview 1

2 Design 3
2.1 FS-i6 Remote Configurations 3
2.2 Processing Sensor Output 4
2.2.1 GPS 4
2.2.2 eCompass 4
2.2.3 Shaft Encoder 5
2.2.4 Receiver and Servos 5
2.2.5 Telemetry Radio 6
2.3 Equations and simulations 8
2.4 Design Alternatives 9
2.4.1 Non-PID Rudder Control 9
2.4.2 Complete PID with derivative 9
2.5 Control Design Description 11
2.5.1 Rudder Control 11
2.5.2 Sail Winch Control 11
2.6 PCB 13

3 Requirements and Verifications 15
3.0.1 Results 16

4 Cost and Schedule 17

5 Ethical Considerations 17

6 Conclusions 18
6.1 Future Work/Alternatives 19

References 20

Appendix A Mathematical Model 21

Appendix B PCB Schematics 30

Appendix C Filter Alternative 33

Appendix D Schedule and Cost 34

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. 1 below for a block diagram overview of the entire system.

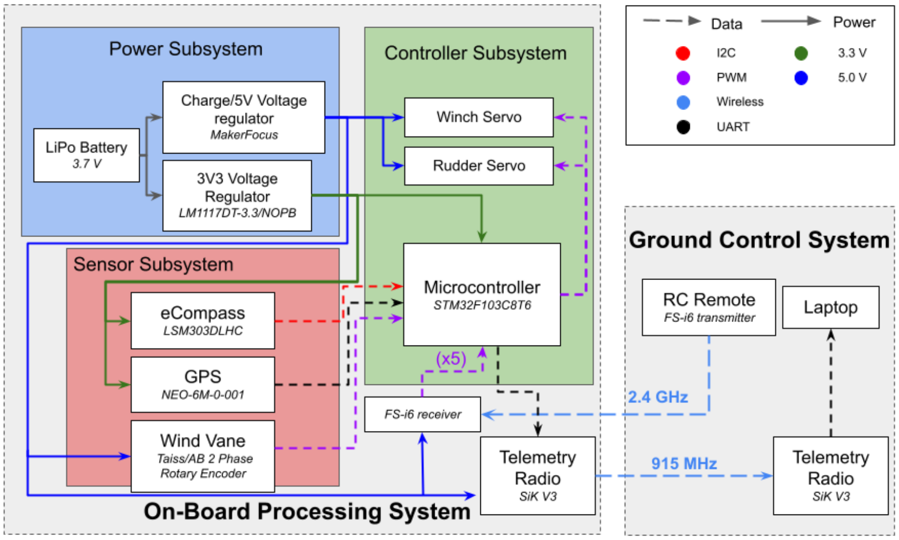


Figure 1: Block Diagram of the system

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.

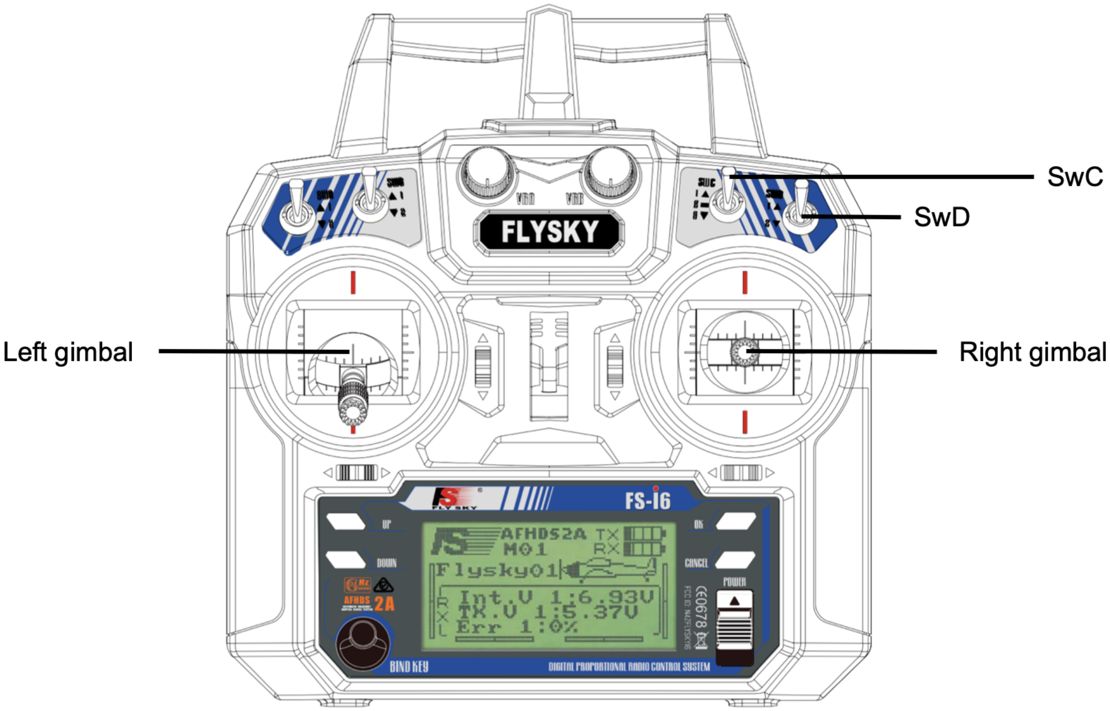


Figure 2: FlySky FS-i6 Remote

2 Design

2.1 FS-i6 Remote Configurations

The FlySky FS-i6 Remote allows the user to manually control the rudder and sail, trigger the autonomous mode and return to base functionality, and set the base position. The left gimbal differs from the right gimbal in that it maintains its position after moving it. This is particularly useful for controlling the rudder since the rudder maintains its position often. Furthermore, the position of the rudder is not visible to the user unlike the sail so the visibility of its position on the remote can be useful. The right gimbal is then used to tighten and loosen the sail.

As shown in Table 1, position 1 corresponds to when the user is in manual mode, position 2 enables autonomous mode and position 3 triggers the sailboat to return to base. The remote requires all switches to be set to their top position when powering on. This ensures the sensors have time to initialize and calibrate prior to any autonomous functionality executes. A visual of the gimbals and switches is shown in Fig. 2.

Table 1: Remote to Channel Configurations

Channel	Remote	Mapping	Use
1	Right Gimbal $\leftarrow, \rightarrow$	Winch Servo	$\leftarrow$: Tighten Sail $\rightarrow$: Loosen Sail
2	Right Gimbal $\uparrow, \downarrow$	N/A	N/A
3	Left Gimbal $\uparrow, \downarrow$	Rudder Servo	$\uparrow$: Rudder $\rightarrow$ Starboard $\downarrow$: Rudder $\rightarrow$ Port side
4	Left Gimbal $\leftarrow, \rightarrow$	N/A	N/A
5	SwC	Sailing Mode	1: Manual 2: Autonomous 3: Return to Base
6	SwD	Set Base Position	1 $\rightarrow$ 2: Sets Base Position to Current GPS Location

2.2 Processing Sensor Output

2.2.1 GPS

The GPS is used to offer the user a method of tracking the sailboat to map its course and allows for the return to base functionality. The NEO-6M GPS module outputs NMEA sentences that are parsed by the TinyGPS+ library [2]. The data used from the GPS includes the speed, heading and the GPS coordinates of the current position of the boat. It is important to note that the heading determined by the GPS is only used to initialize the desired heading within the return to base functionality. The GPS determines the heading through comparing its current location to its previous location the last time it was updated. Since the GPS is updated frequently to get an accurate speed and location, the heading is extremely inaccurate when the boat is not travelling 5 m between updates.

When the user initiates return to base mode, the desired course heading ϕ is calculated using [[3], eq. 1]. α_1, β_1 refers to the current latitude and longitude of the boat and α_2 and β_2 refers to the latitude and longitude of the base position respectively. All coordinates are assumed to be in radians. Furthermore, the distance to base Λ is provided to the user and is calculated using [[3], eq. 2].

$$\phi = atan(\frac{\sin(\beta_2 - \beta_1)\cos(\alpha_2)}{\cos(\alpha_1)\sin(\alpha_2) - \sin(\alpha_1)\cos(\alpha_2)\cos(\beta_2 - \beta_1)}) \tag{1}$$

$$\Lambda = atan(\frac{\sqrt{(\cos(\alpha_1)\sin(\alpha_2) - \sin(\alpha_1)\cos(\alpha_2)\cos(\beta_1 - \beta_2))^2 + (\alpha_2)^2}}{(\sin(\alpha_1) * \sin(\alpha_2)) + (\cos(\alpha_1) * \cos(\alpha_2) * \cos(\beta_1 - \beta_2))}) * 6372795 \tag{2}$$

2.2.2 eCompass

The LSM303 library is used with the eCompass is to acquire the current heading Φ and heeling angle ϕ_{xyz} [4]. The best positioning of the eCompass is within close proximity to the rudder and winch servos. The magnetic field of these servos produces hard iron distortion that was accounted for using hard iron calibration The calculation for hard iron offsets is shown in [[5] , eq. 3] where λ is one of x,y,z axes. The hard iron offsets are used to adjust the acceleration values in each axis, as shown in 4 where $G_{p\lambda}$ corresponds to the acceleration in the axis λ.

$$H_\lambda = \frac{1}{2}(\min(G_{p\lambda}) + \max(G_{p\lambda})) \tag{3}$$

$$G_{p\lambda} = G_{p\lambda} - H_\lambda \tag{4}$$

The heeling angle ϕ_{xyz} is approximated as the roll of the sailboat. This simple calculation is shown in Eq. 5 below.

$$\tan\phi_{xyz} = \left(\frac{G_{py}}{G_{pz}}\right) \tag{5}$$

2.2.3 Shaft Encoder

A wind vane is fixed to the top of a shaft encoder to determine the relative wind angle with respect to the bow. The encoder is attached to two interrupts that are triggered upon movement in both directions. The order that each interrupt is triggered corresponds to the direction the wind vane is turning. The RotaryEncoder library is used to get the position of the shaft encoder through timing the frequency and order of interrupts [6]. The position value from this library does not reset until it reaches overflow. This behaviour is accounted for in the mapping function between the position output from this library to a relative wind angle is shown in Eq. 6. θ_w is the relative wind angle, δ is the output from the encoder and ρ corresponds to the direction of rotation.

$$\theta_w = \left\{ \begin{array}{ll} 360 - map((\delta \mod 1190), 0, 1190, 0, 360), & \text{if } \rho < 0 \\ map((\delta \mod 1190), 0, 1190, 0, 360), & \text{else} \end{array} \right\} \tag{6}$$

2.2.4 Receiver and Servos

The 6-channel receiver outputs a pwm signal that is directly processed by the MCU. Unlike standard RC sailboats, there is no direct connection between the receiver and servos. The output 50 Hz waveform from the receiver ranges between a 2.3% and 11.9% duty cycle, and the range expected by the servos is between a 5% and 10% duty cycle. Each channel is attached to an interrupt pin that triggers whenever the pin changes value. The pulse width is calculated using the stored previous time that the interrupt was triggered. Since the rudder and sail movement are limited within the movement of the servo itself, the adjusted servo pulse width range is calibrated to the desired behaviour. These ranges are shown in Table 2. The movement range of the rudder and winch corresponds to the angle of the rudder and sail relative to when it is centered, respectively. No adjustments are needed for the switches (Channels 5 and 6), however, the range of accepted pulse width values for each mode and setting the base position is calibrated.

Table 2: Receiver and Servo Pulse Width Ranges

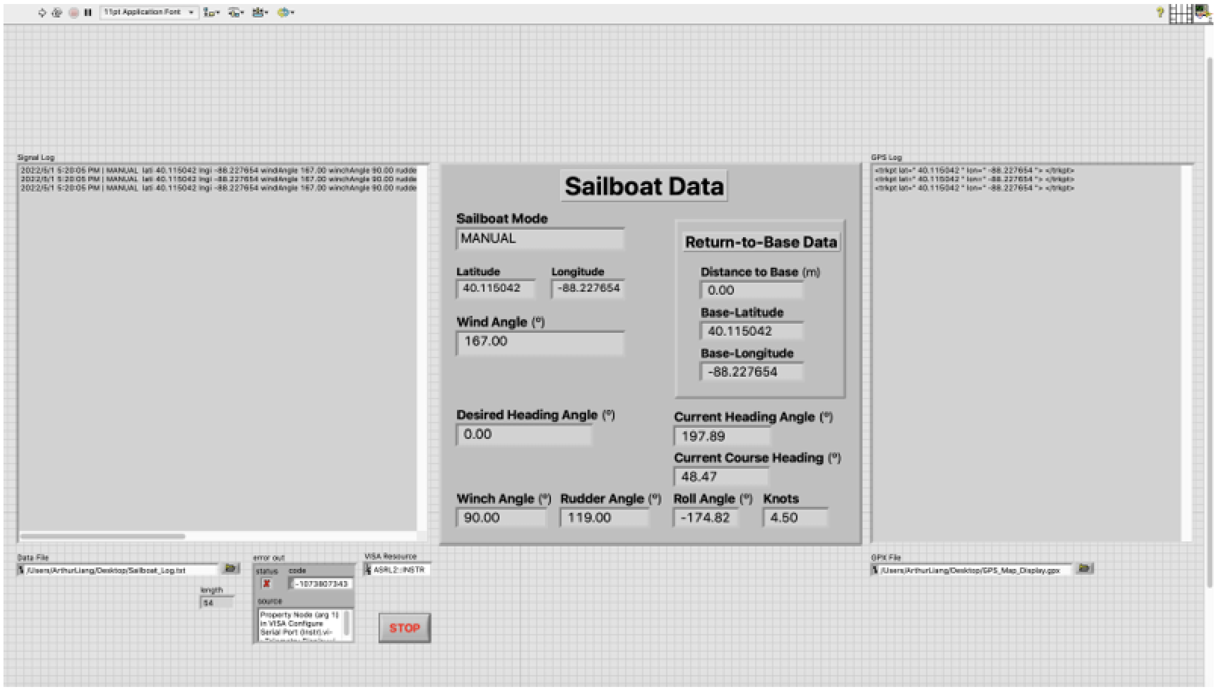
Channel	Use	Pulse Width Range Receiver (μs)	Pulse Width Range Servo (μs)	Movement Range (°)
1	Winch Servo	950 - 2050	1205 - 1916	-45 - 45
3	Rudder Servo	1000 - 2000	1400 - 1611	0 - 90

2.2.5 Telemetry Radio

The telemetry radio used in this project is a SiK Telemetry Radio Version 2, produced by mRobotics. Due to the design of the electronics, (check Fig.1), only single direction communication is required from the telemetry radio, hence the TX pin of the telemetry module on the sailboat is connected to the STM32 chip. In addition, the telemetry radio transfers data through serial communication, hence the module at base is connected via USB, which is then detected and read by the LabView program.

The LabView program consists of two sides, the user interface window and a coding window. The user interface window is designed so that the user can read off data from the boat, while providing the data log for quick debugging and choosing directory for saving the log files. Currently, the data displayed updates at a rate of 1 Hz, due to the Arduino code on board is programmed to send data every second (however, such rate can be increased when needed). In addition, a log for showing what is being entered into the GPX file is also displayed on the right of the user interface screen. This provide the user with a log of the GPS coordinates, but also shows when an error has occurred in the transferred data which affects the latitude and longitude values. The user interface of the ground control system in Labview can be seen in Fig.3.

The coding window in Fig. 4 shows all the coding for displaying the data and other functions like logging the data into a file (or reading data from serial port). The entirety of the code is placed inside a while loop, which keeps the program running until manually stopped. The entirety of the code can be separated into 3 sections. The blue section is for separating the received data storing the current values into their corresponding variables. Since there are 14 different data that are being sent through, many cuts in the received data are needed to separate them into each variable. The green section is for generating the GPX file. The code takes the latitude and longitude data (from the blue section) and insert them into each line of the file (at the corresponding locations) to generate the file. The red section creates the log, which takes in the raw data (before being cut) from the receiver, adds the date/time to the start (for easier tracking of data), and prints them in the new line.



Signal Log

```
2022/5/1 5:20:05 PM | MANUAL  lat: 40.115042 lngi -88.227654 windAngle 167.00 winchAngle 90.00 rudde
2022/5/1 5:20:05 PM | MANUAL  lat: 40.115042 lngi -88.227654 windAngle 167.00 winchAngle 90.00 rudde
2022/5/1 5:20:05 PM | MANUAL  lat: 40.115042 lngi -88.227654 windAngle 167.00 winchAngle 90.00 rudde
```

Sailboat Data

Sailboat Mode: MANUAL

Latitude: 40.115042

Longitude: -88.227654

Wind Angle (°): 167.00

Return-to-Base Data

Distance to Base (m): 0.00

Base-Latitude: 40.115042

Base-Longitude: -88.227654

Desired Heading Angle (°): 0.00

Current Heading Angle (°): 197.89

Current Course Heading (°): 48.47

Winch Angle (°): 90.00

Rudder Angle (°): 119.00

Roll Angle (°): -174.82

Knots: 4.50

GPS Log

```
<trkpt lat=" 40.115042 " lon=" -88.227654 "> </trkpt>
<trkpt lat=" 40.115042 " lon=" -88.227654 "> </trkpt>
<trkpt lat=" 40.115042 " lon=" -88.227654 "> </trkpt>
```

Data File: /Users/ArthurLiang/Desktop/Sailboat_Log.txt

length: 54

error out
status: ✗
code: -1073807343
source: Property Node (arg 1) in VISA Configure Serial Port (Instr).vi->

VISA Resource: ASRL2::INSTR

GPX File: /Users/ArthurLiang/Desktop/GPS_Map_Display.gpx

Figure 3: User interface of the ground control system in LabView

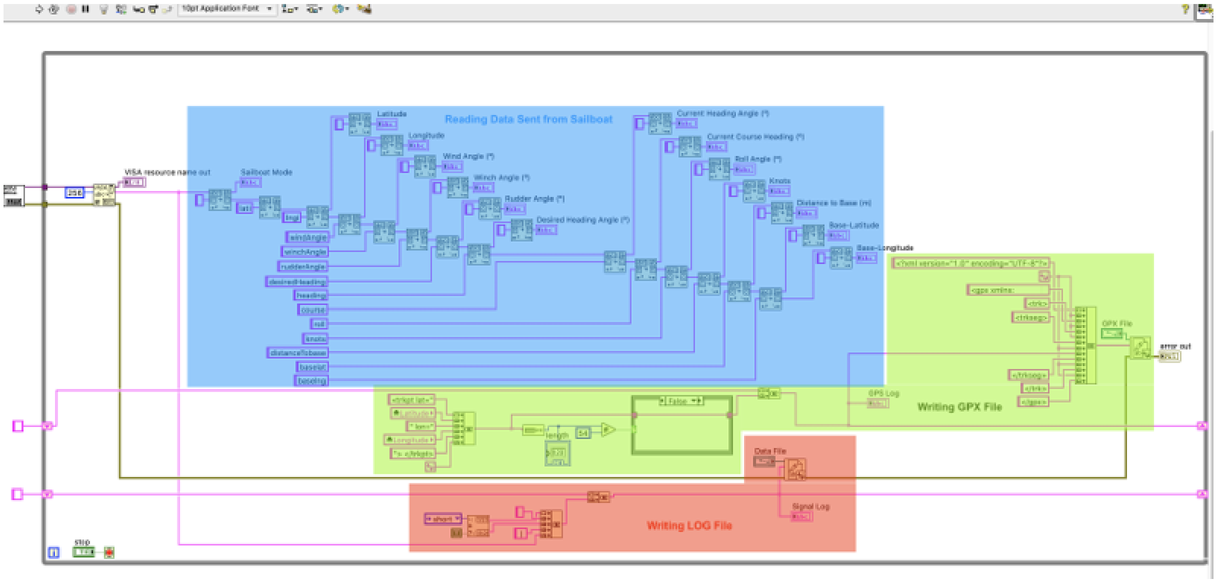


Figure 4: LabView Program Setup

2.3 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. 5 below.

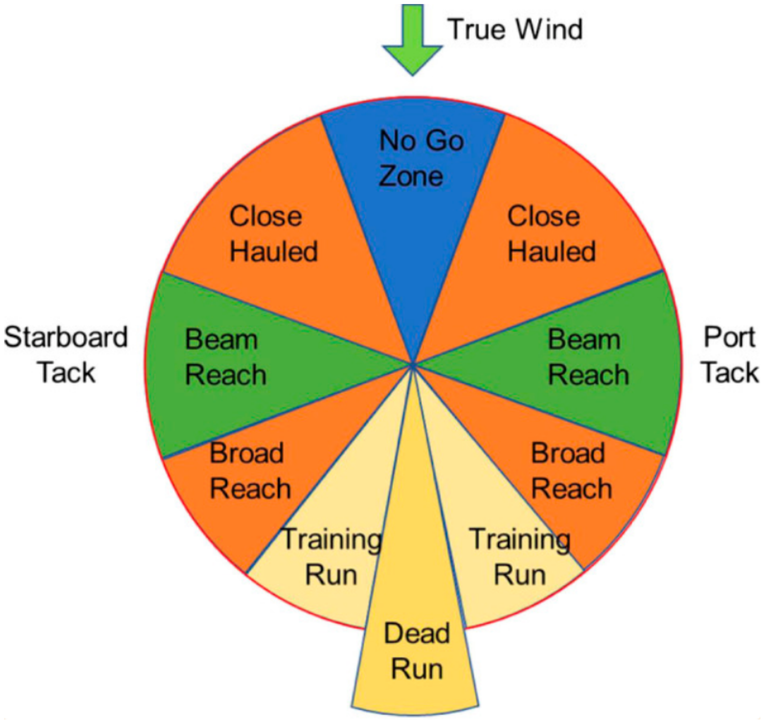


Figure 5: Points of sail [1]

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. See Appendix A for simulation, equations and results.

2.4 Design Alternatives

2.4.1 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. 6 below.

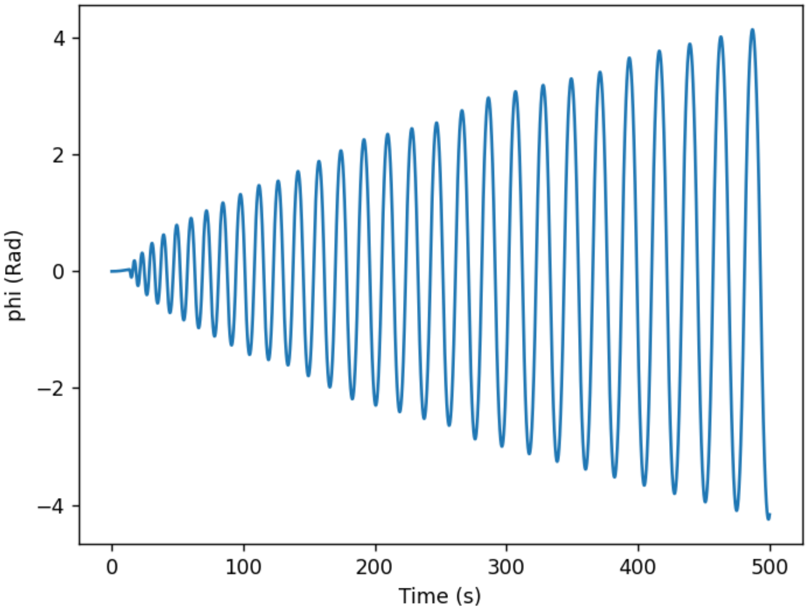


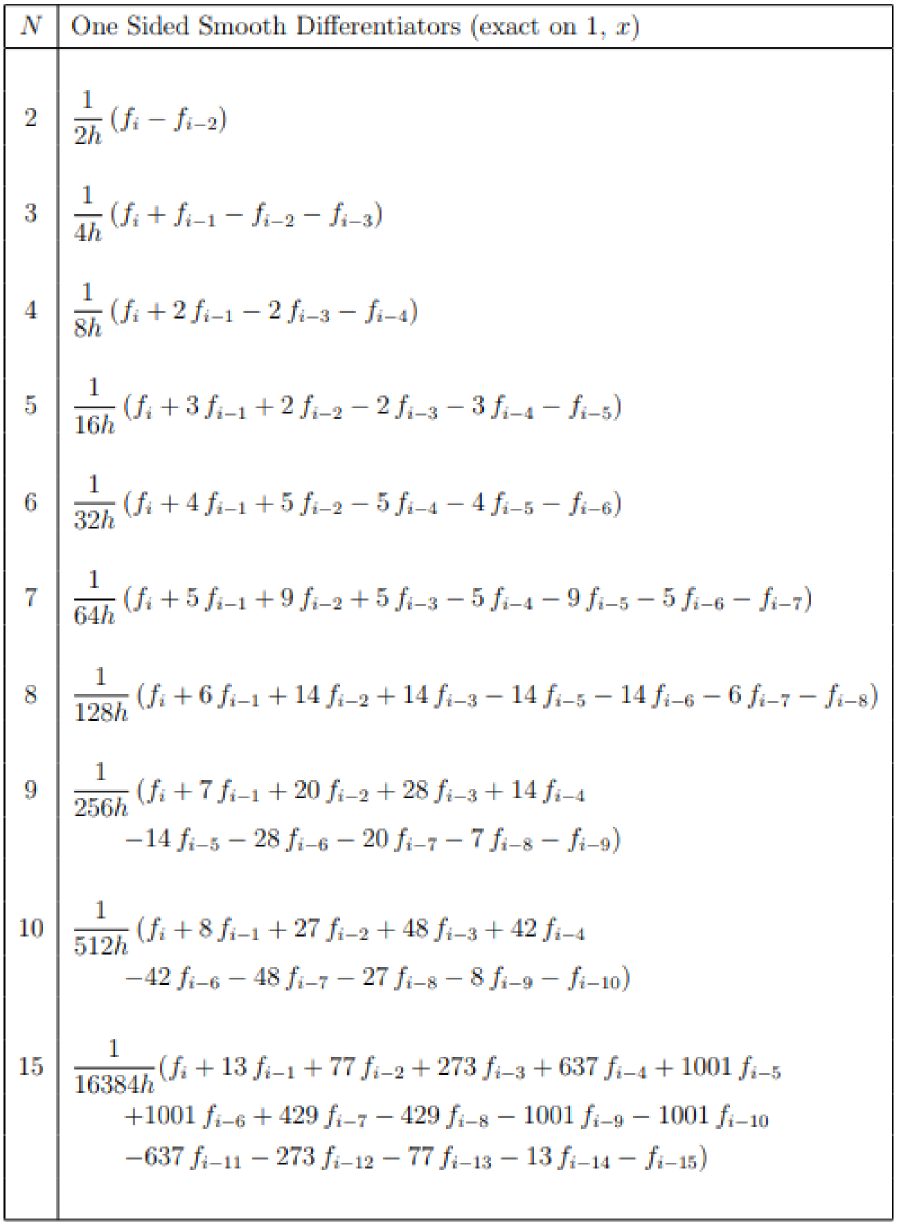
Figure 6: Phi (deviation angle) vs time for a non-PID control approach

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.

2.4.2 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. 7 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 7: One Sided Smooth Differentiators [7]

2.5 Control Design Description

2.5.1 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 8.

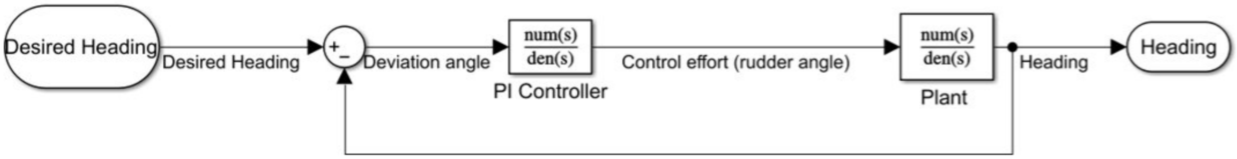


Figure 8: Block diagram of the rudder system

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.

2.5.2 Sail Winch Control

The sail winch was programmed using a Moore finite state machine. The finite state machine can be seen in figure 9.

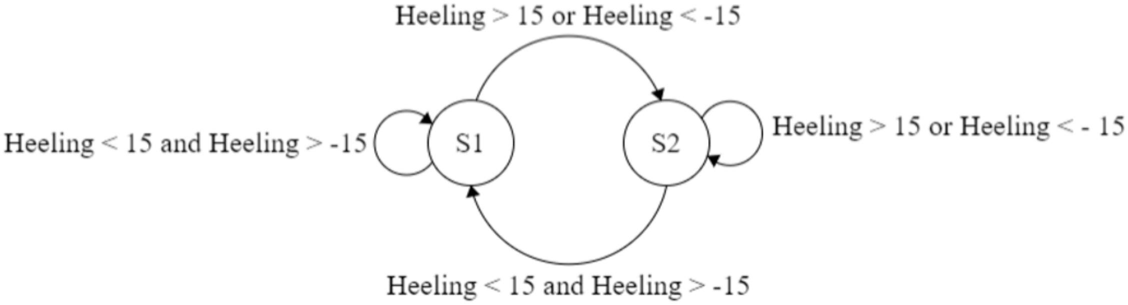

Heeling > 15 or Heeling < -15
Heeling < 15 and Heeling > -15
S1
S2
Heeling > 15 or Heeling < - 15
Heeling < 15 and Heeling > -15

Figure 9: Moore 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 3.

Table 3: Sail Angle Lookup Table

Apparent Wind Angle	Point of Sail	Sail Angle
$0 \leq \theta \leq 45 \vert\vert 315 \leq \theta \leq 360$	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.

2.6 PCB

PCB schematic figures can be seen in Appendix B.

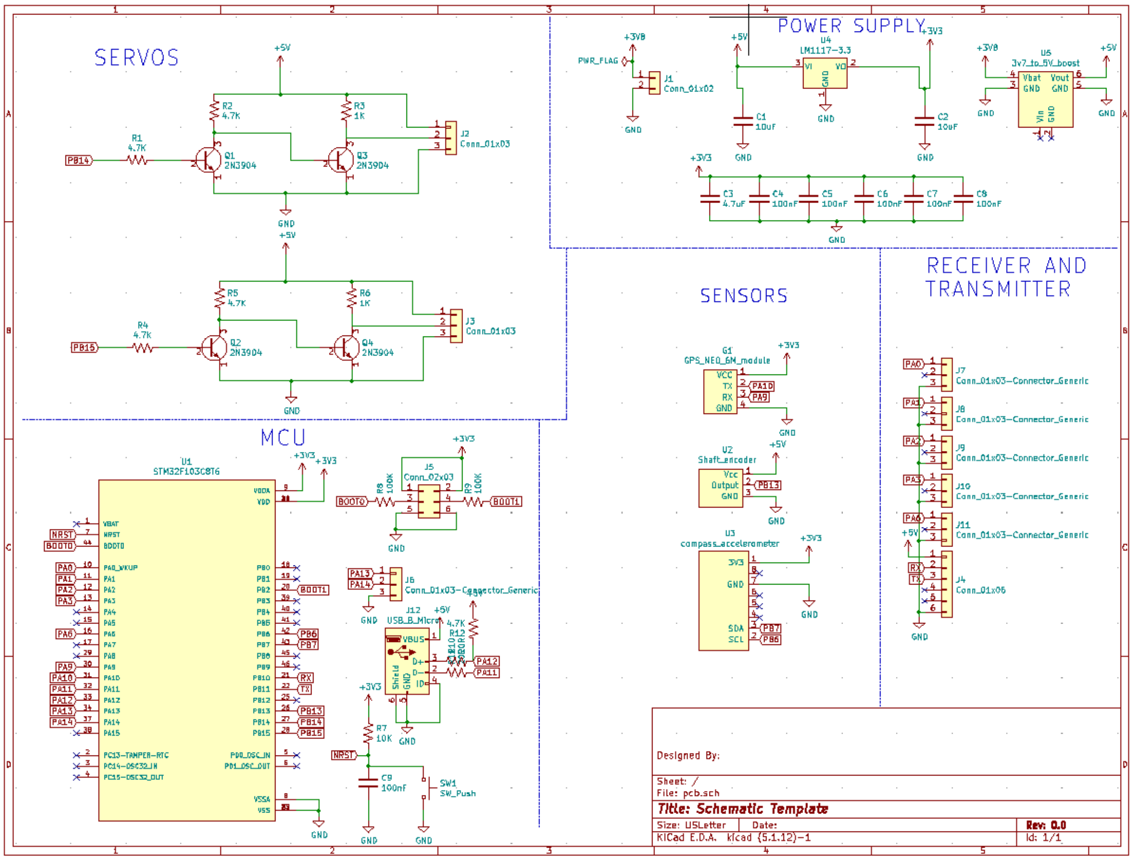


Figure 10: PCB Schematic

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. 26.

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.

Sensors: Footprints for the sensors were created for the sensors that the system uses (shatf encoder, GPS and compass). As you can see in the schematic below, 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.

MCU Module: The microcontroller module contains a reset button, the microcontroller and the connections needed to program the microcontroller.

Receiver and Transmitter Connections: All the connectors needed for communications can also be found in the PCB. The schematics can be seen in Fig. 30.

3 Requirements and Verifications

Table 4: 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 5: On-Board Processing System Requirements and Verification Pt 2.

• The processing system correctly adjusts the rudder angle and the sail angle according to Table 3.	• 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 3 • Rotate the sailboat to both sides and check that the angle of the rudder reaches its maximum after no more than 5 seconds.

Table 6: 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 3. 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 [8]. 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 [9]. Therefore, the total machine shop cost is 15hr x $36.65/hr = **$549.75**.

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

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

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

5 Ethical Considerations

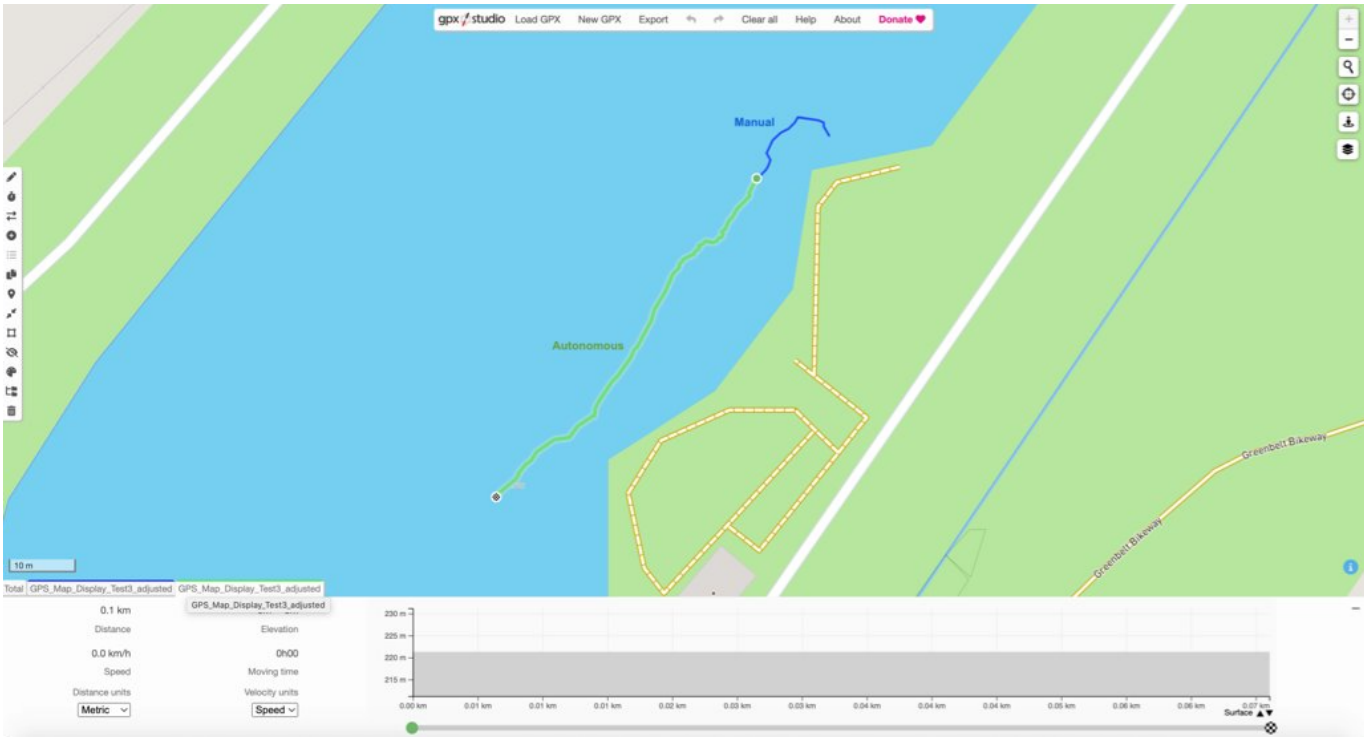
There are a few ethics policies that need to be taken into consideration with this project. Section 7.6 of the IEEE Code of Ethics I.5 states, "to seek, accept, and offer honest criticism of technical work... and to credit properly the contributions of others" [10]. As this project is not the first design 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" [10].

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 11.



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

Figure 11: Sailboat movement (green: autonomous mode, blue: manual mode)

As described in Section 2.4, 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

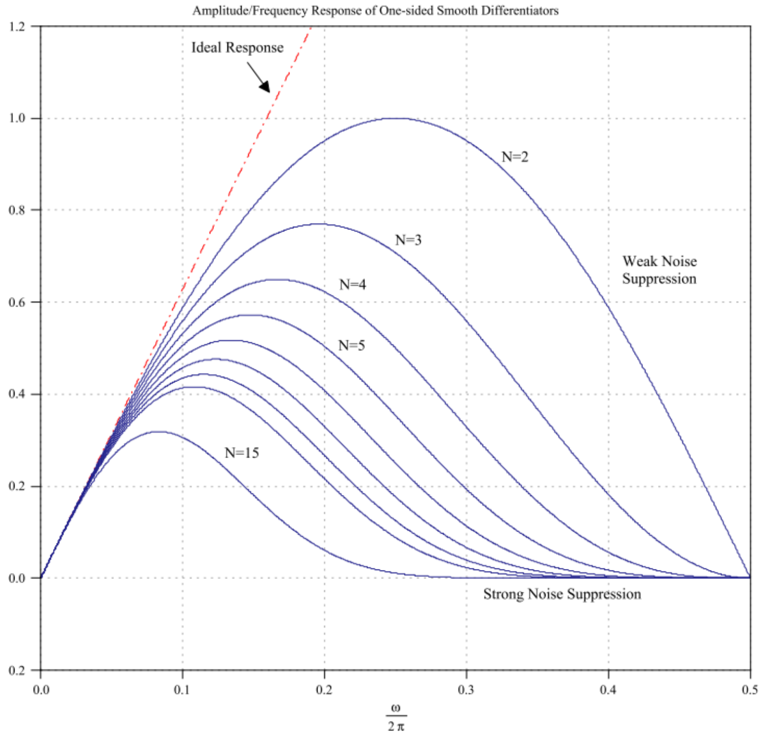


Figure 12: Amplitude/Frequency Response of One-sided Smooth Differentiators [7]

Filtered Derivative: A stronger filter than those shown in Figure 12 could be used to filter the noise out from the derivative and integrate the differential control. The filter would be designed to have a very low cutoff frequency and to mimic the behavior of a derivative for low frequency signals. The expression for this filter and its bode blot can be seen in Appendix C.

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] M. Hart. "TinyGPSPlus." (2022), [Online]. Available: https://github.com/mikalhart/TinyGPSPlusv (visited on 03/21/2022).

[3] C. Veness. "Calculate distance, bearing and more between Latitude/Longitude points." (2022), [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html (visited on 03/22/2022).

[4] Pololu. "LSM303 Library." (2016), [Online]. Available: https://github.com/pololu/lsm303-arduino (visited on 03/20/2022).

[5] T. Ozyagcilar. "Calibrating an eCompass in the Presence of Hard- and Soft-Iron Interference." (2013), [Online]. Available: https://www.nxp.com/docs/en/application-note/AN4246.pdf (visited on 03/18/2022).

[6] M. Hertel. "RotaryEncoder Library." (2021), [Online]. Available: https://github.com/mathertel/RotaryEncoder (visited on 03/16/2022).

[7] P. Holoborodko. "One-Sided Differentiators." (2009), [Online]. Available: http://www.holoborodko.com/pavel/wp-content/uploads/OneSidedNoiseRobustDifferentiators.pdf (visited on 04/30/2022).

[8] G. C. of Engineering. "Salary Averages." (), [Online]. Available: https://ece.illinois.edu%20/admissions/why-ece/salary-averages (visited on 02/24/2022).

[9] S. of Chemical Sciences at Illinois. "Machine Shop." (), [Online]. Available: https://scs.illinois.edu%20/resources/cores-scs-service-facilities/machine-shop (visited on 02/24/2022).

[10] 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 Mathematical Model

This is a free body diagram of the sailboat taking into account the assumptions 2.3:

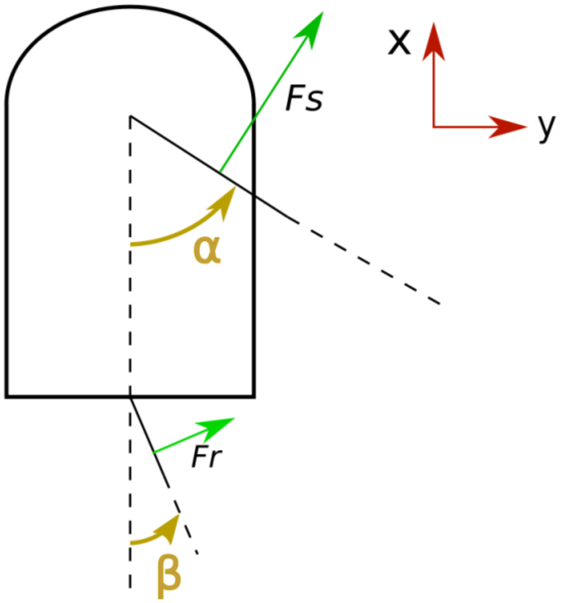


Figure 13: Free body diagram of the sailboat

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 A (note the directions of forces F_r and F_s change when $\beta < 0$ and $\alpha < 0$ respectively).

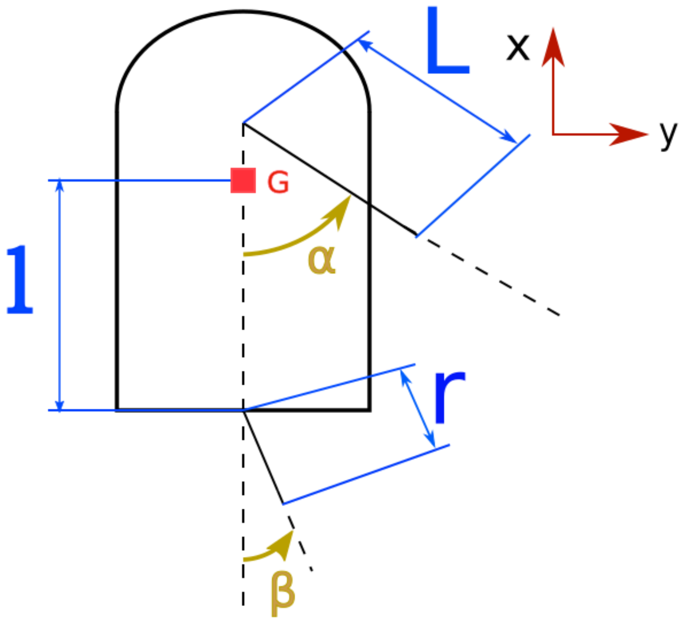


Figure 14: Depiction of variables L, l and r

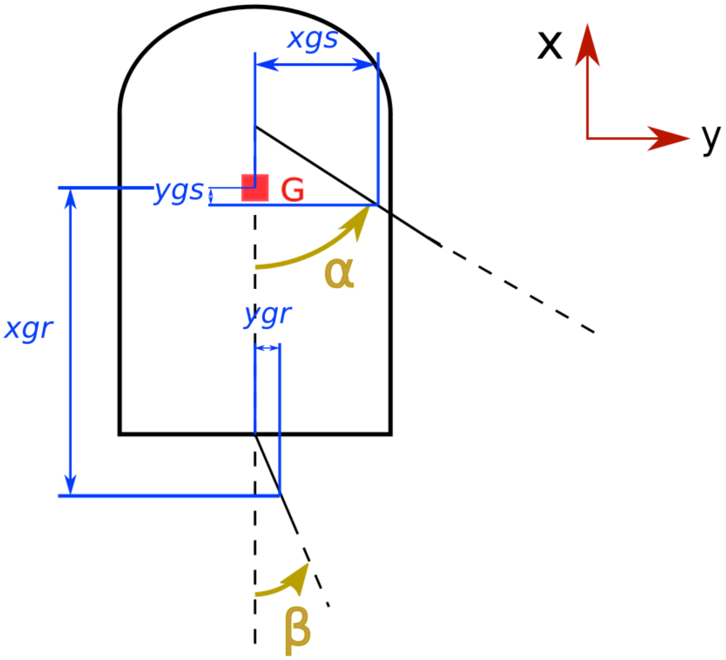


Figure 15: Depiction of variables $x_g r$, $x_g s$, $y_g r$ and $y_g s$

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

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 7 - 12 were used.

$$[h]F = m\frac{d^2x}{d^2t} \tag{7}$$

$$v(t) = \int_0^t \frac{F(t)dt}{m} \approx \int_0^{t-t_0} \frac{F(t-t_0)dt}{m} + \frac{F(t)t_0}{m} \tag{8}$$

$$x(t) = \int_0^t v_x(t)dt \approx \int_0^{t-t_0} v_x(t)dt + v_x(t-t_0)t_0 \tag{9}$$

$$y(t) = \int_0^t v_y(t)dt \approx \int_0^{t-t_0} v_y(t)dt + v_y(t-t_0)t_0 \tag{10}$$

$$w(t) = \frac{\int_0^t T(t)dt}{M} \approx \frac{\int_0^{t-t_0} T(t)dt}{M} + \frac{T(t-t_0)t_0}{M} \tag{11}$$

$$\phi(t) = \int_0^t w(t)dt \approx \int_0^{t-t_0} w(t)dt + w(t-t_0)t_0 \tag{12}$$

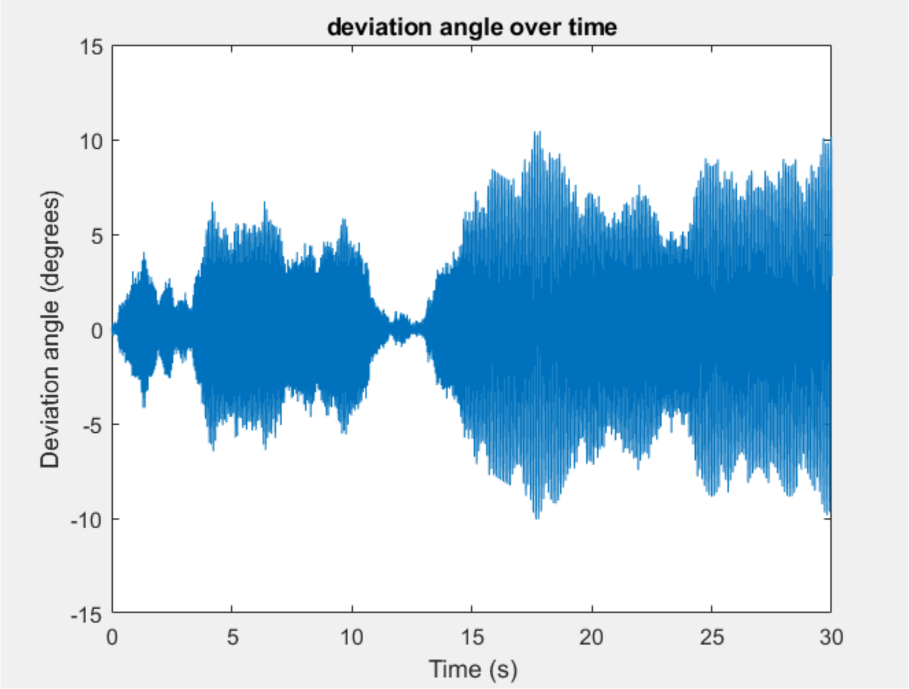


Figure 16: Deviation angle over time while running dead run

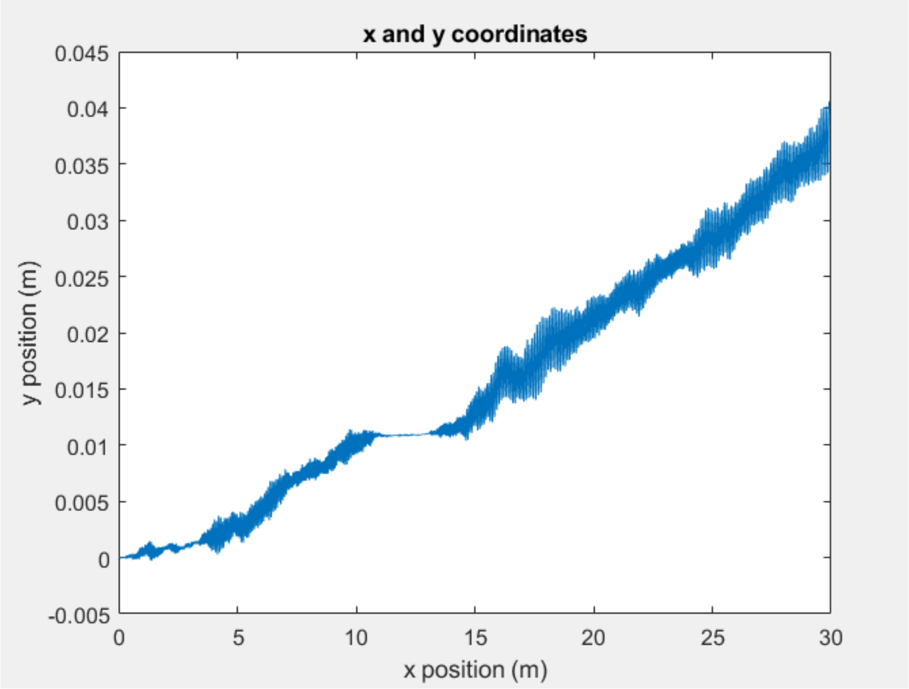


Figure 17: Boat position while running dead run

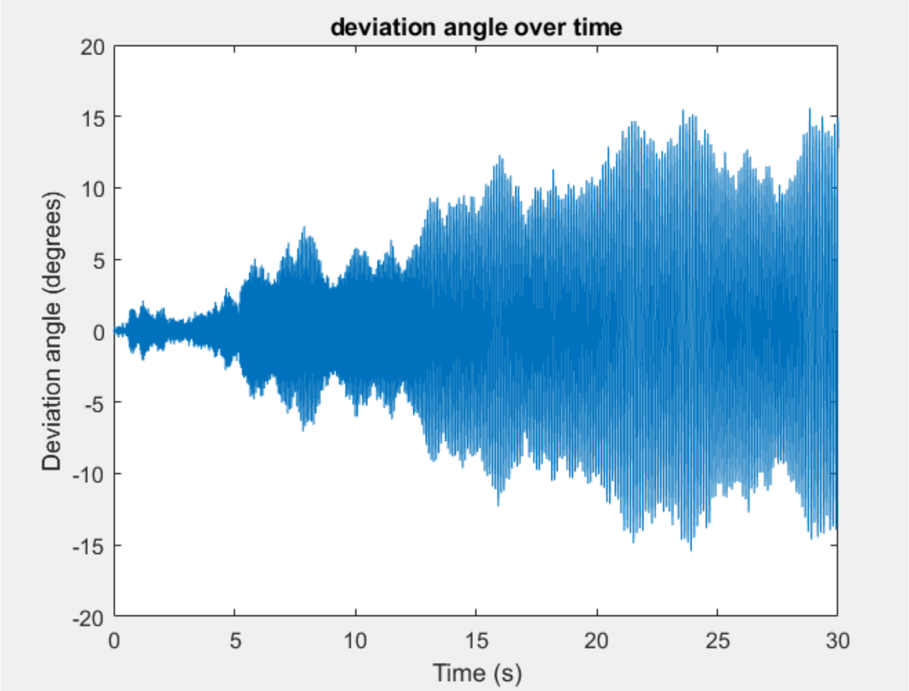


Figure 18: Deviation angle over time while running training run

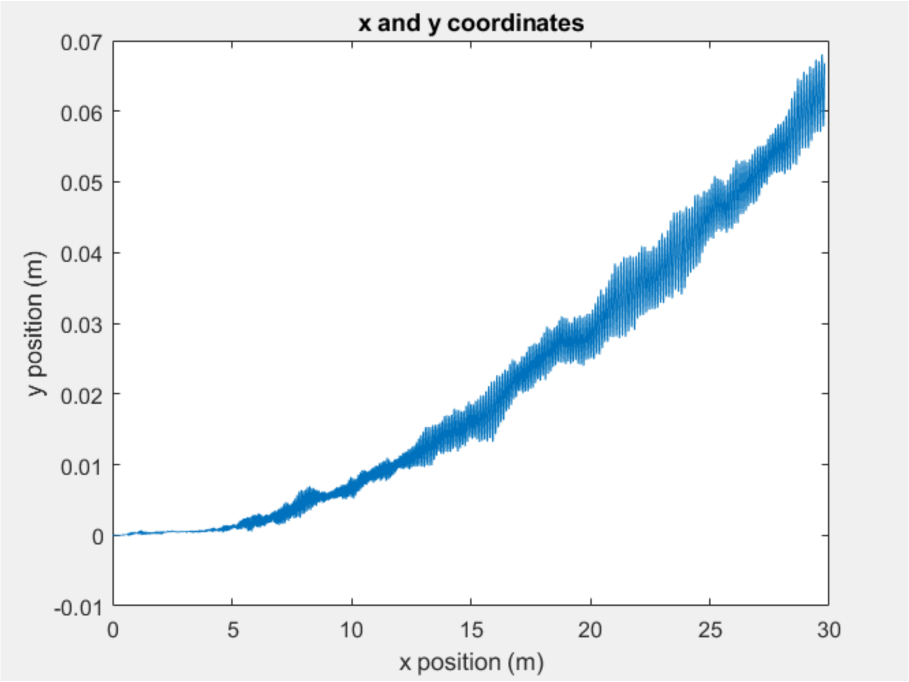


Figure 19: Boat position while running training run

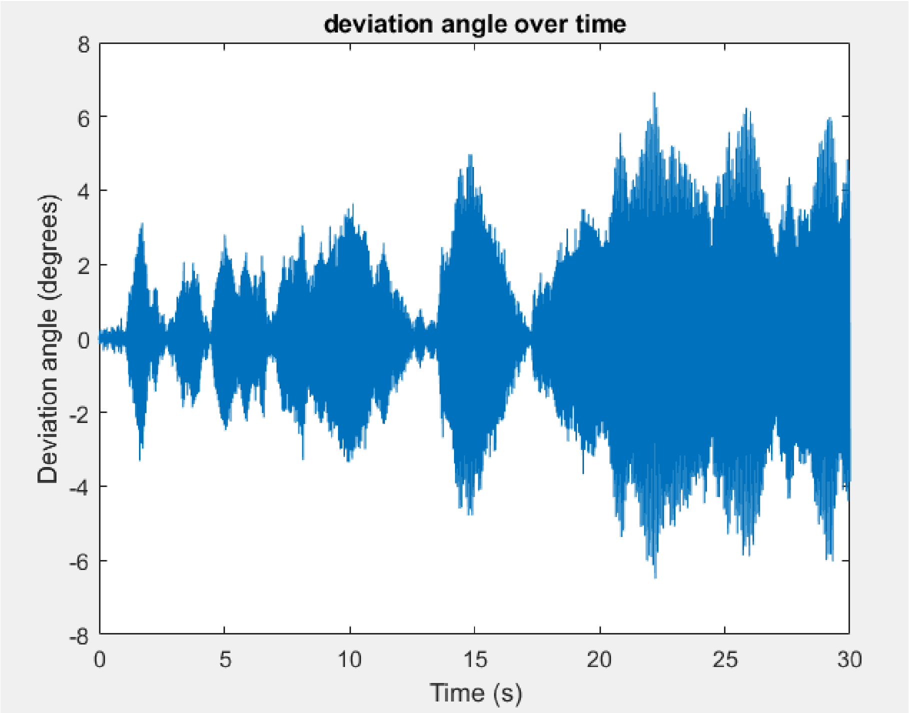


Figure 20: Deviation angle over time while running broad reach

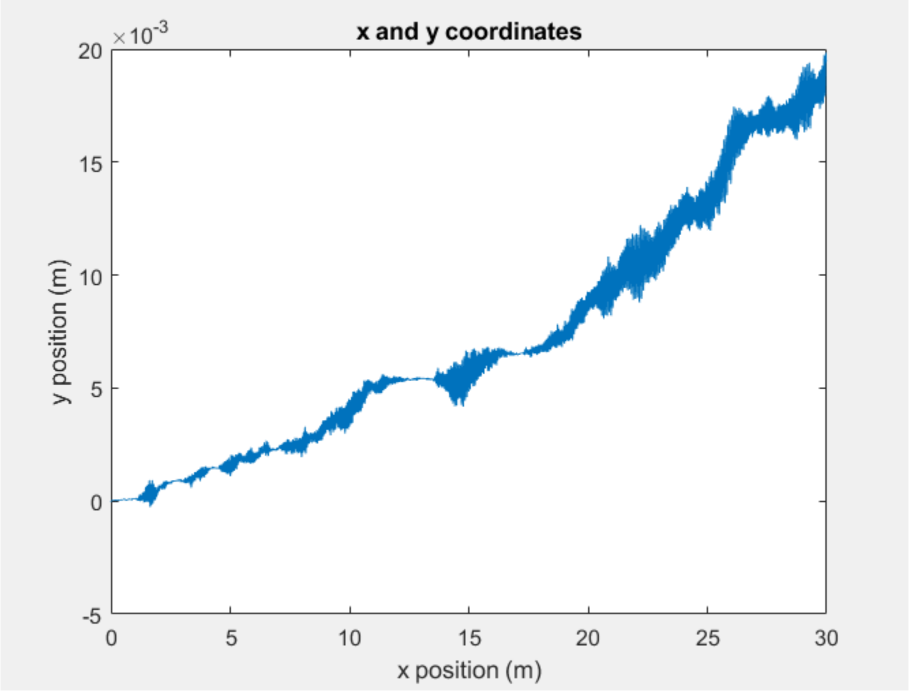


Figure 21: Boat position while running broad reach

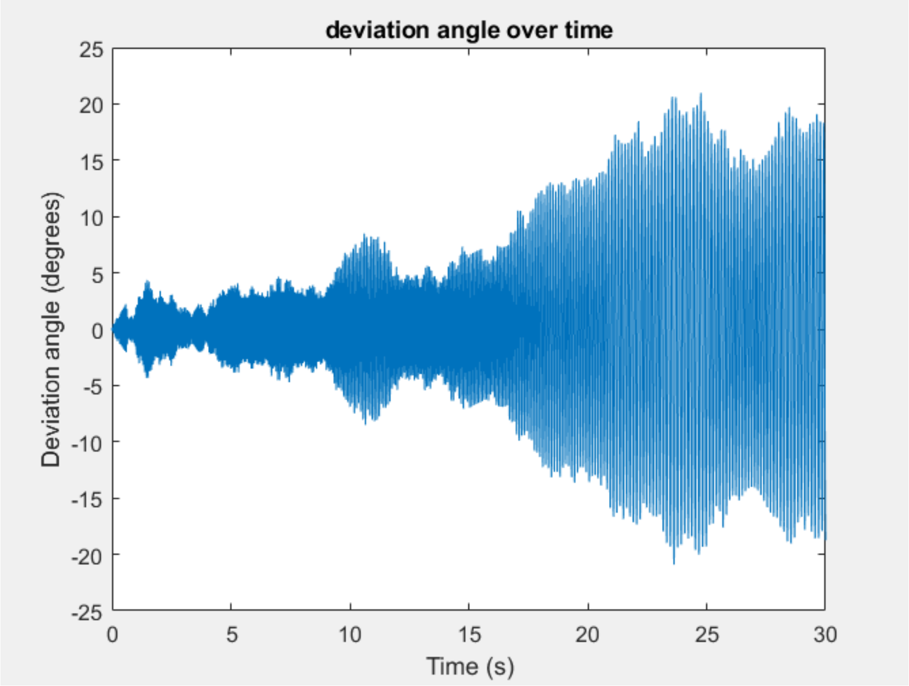


Figure 22: Deviation angle over time while running beam reach

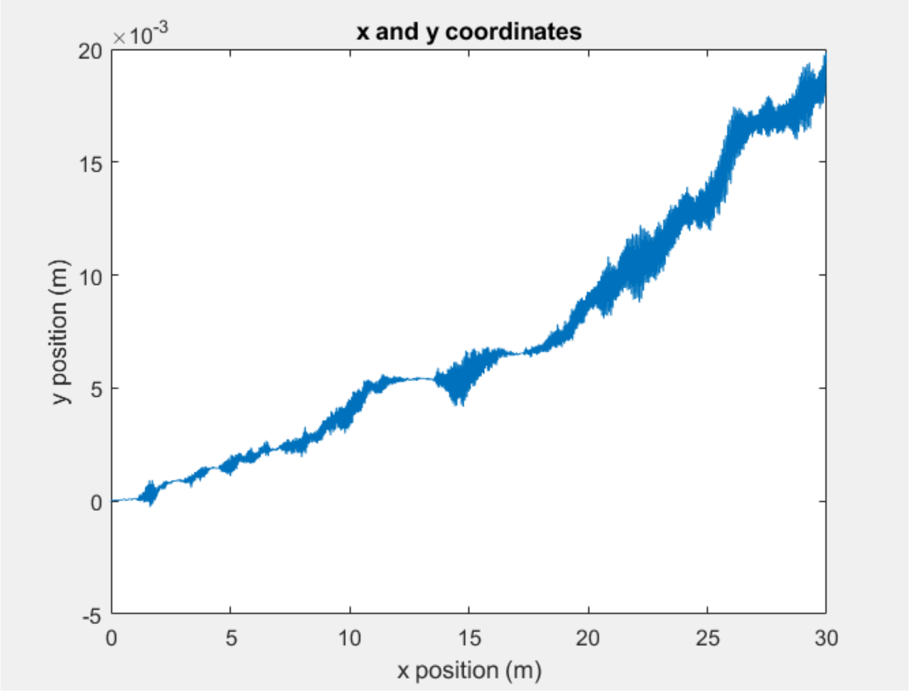


Figure 23: Boat position while running beam reach

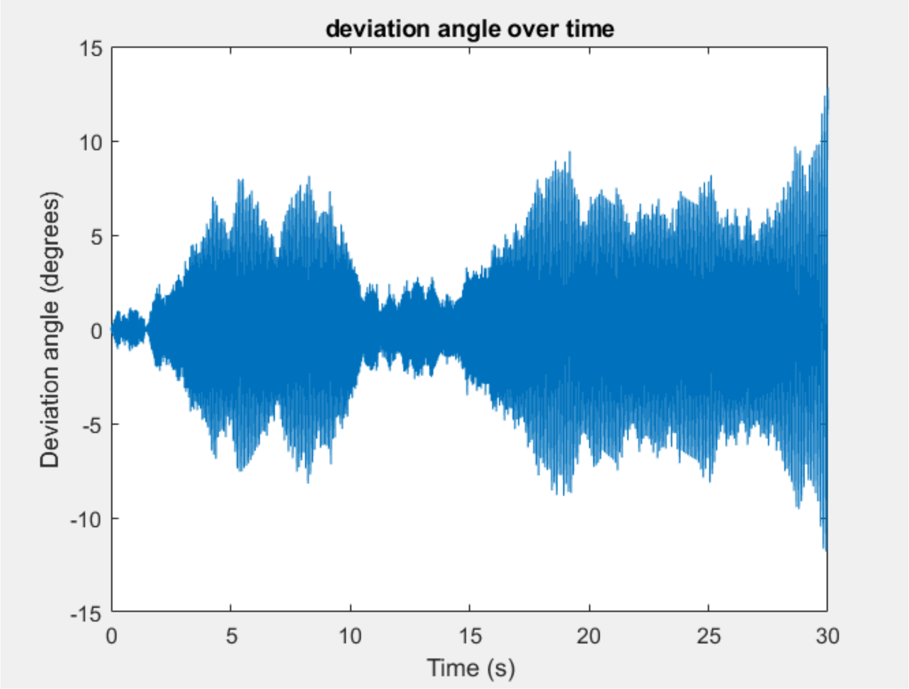


Figure 24: Deviation angle over time while running closed hauled

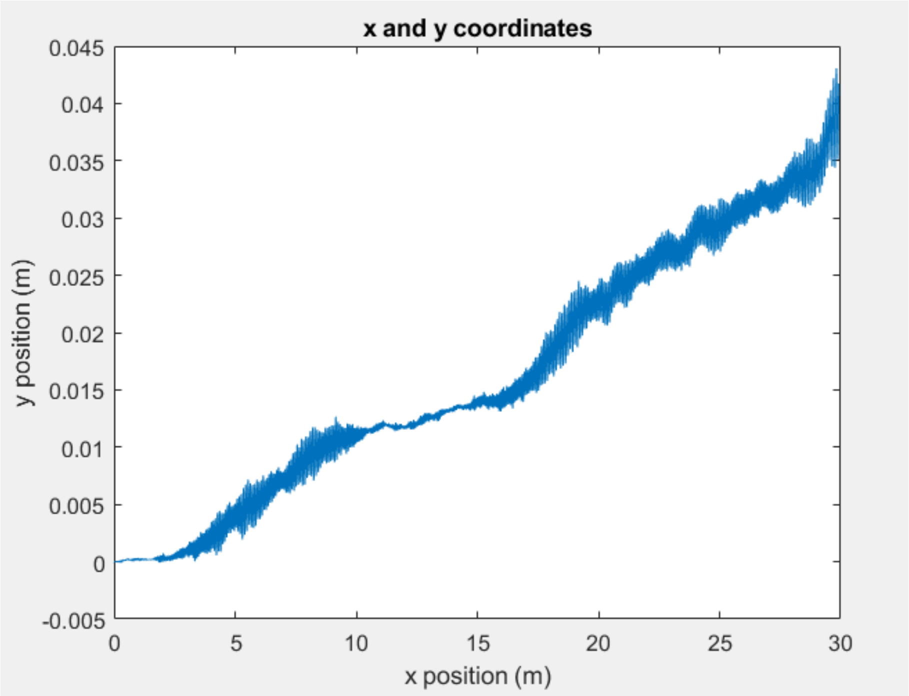


Figure 25: Boat position while running closed hauled

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 2.4.

Appendix B PCB Schematics

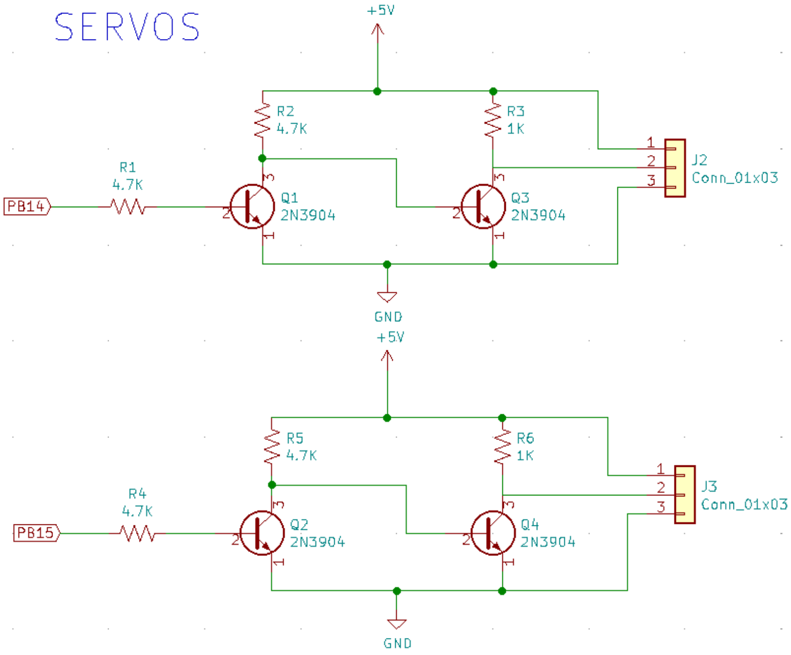


Figure 26: Level Shifters Schematic

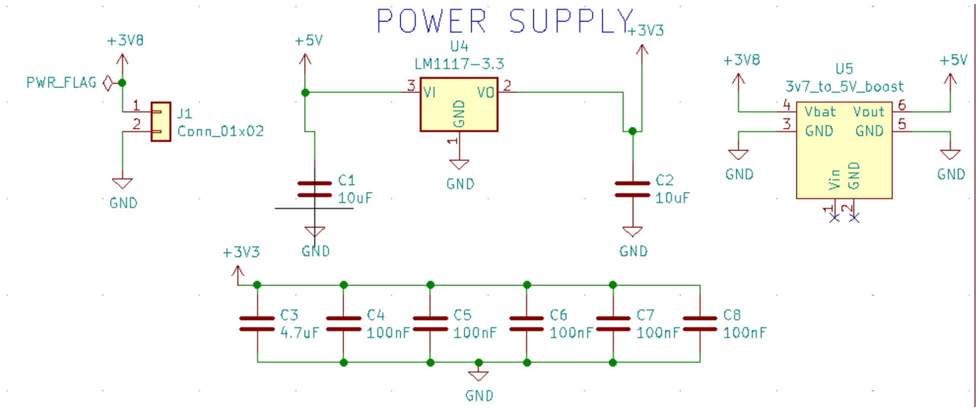


Figure 27: Power Module Schematic

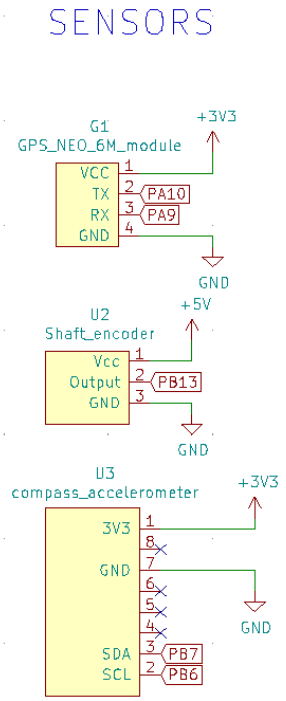


Figure 28: Sensors schematic

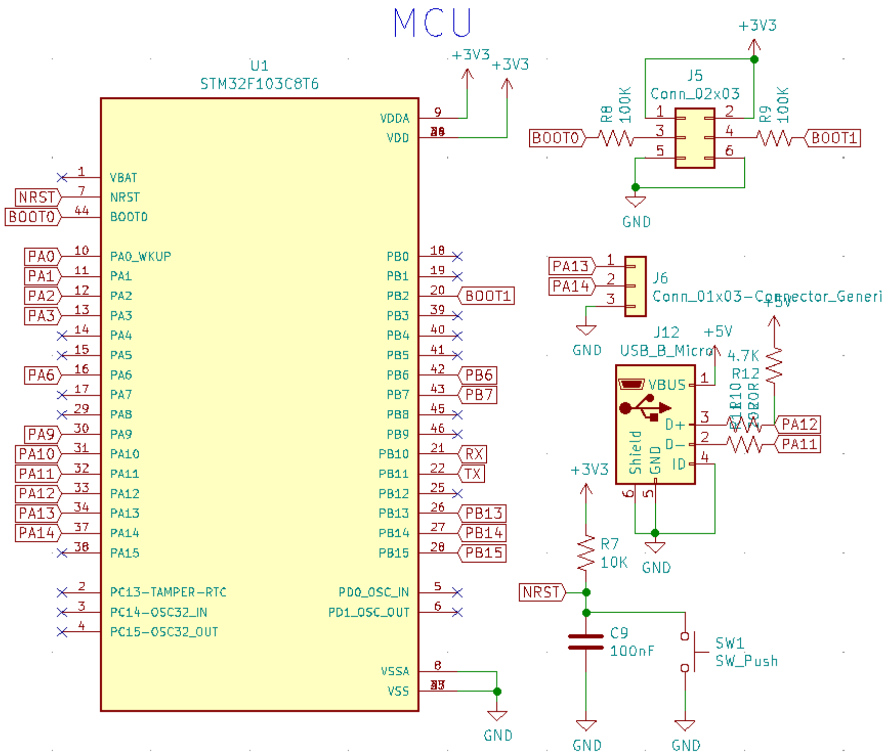


Figure 29: MCU Module Schematic

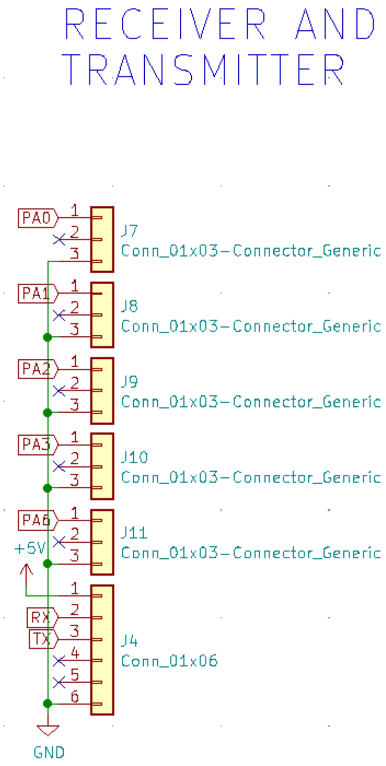

RECEIVER AND
TRANSMITTER
PA0
PA1
PA2
PA3
PA6
+5V
RX
TX
1
2
3
4
5
6
J7
Conn_01x03-Connector_Generic
J8
Conn_01x03-Connector_Generic
J9
Conn_01x03-Connector_Generic
J10
Conn_01x03-Connector_Generic
J11
Conn_01x03-Connector_Generic
J4
Conn_01x06
GND

Figure 30: Receiver and Transmitter Connections Schematic

Appendix C Filter Alternative

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 13.

$$\frac{s}{(1+\frac{s}{w_p})^2 * (1+\frac{s}{10*w_p})^3 * (1+\frac{s}{100*w_p})^4} \tag{13}$$

As seen in Fig. 31 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.

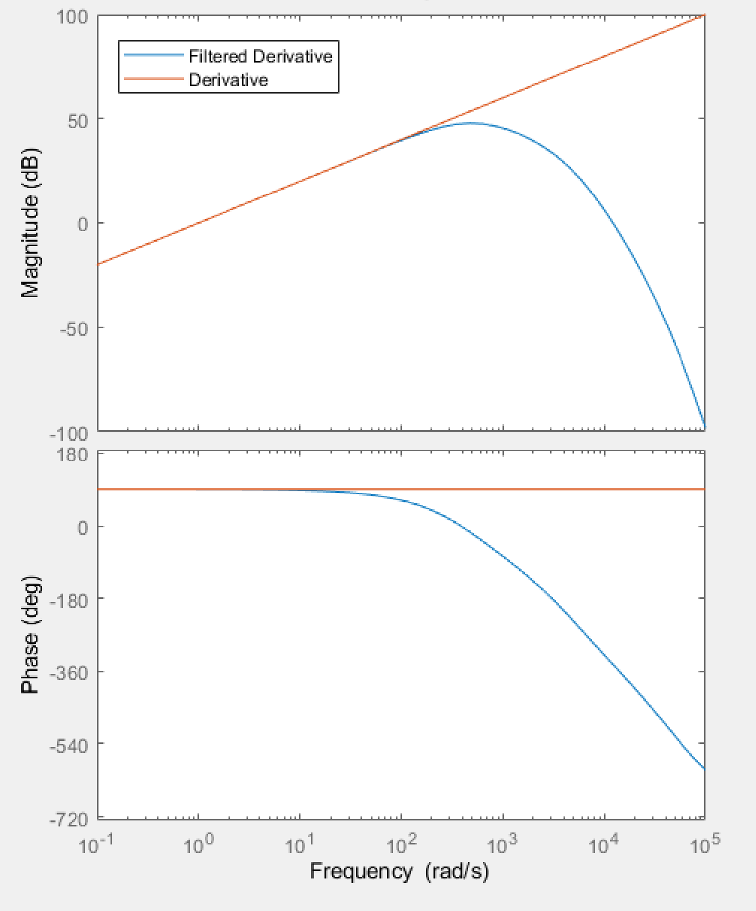


Figure 31: Bode plot for the filter and pure derivative (wp = 500 rad/s)

Appendix D Schedule and Cost

Table 7: 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 8: 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