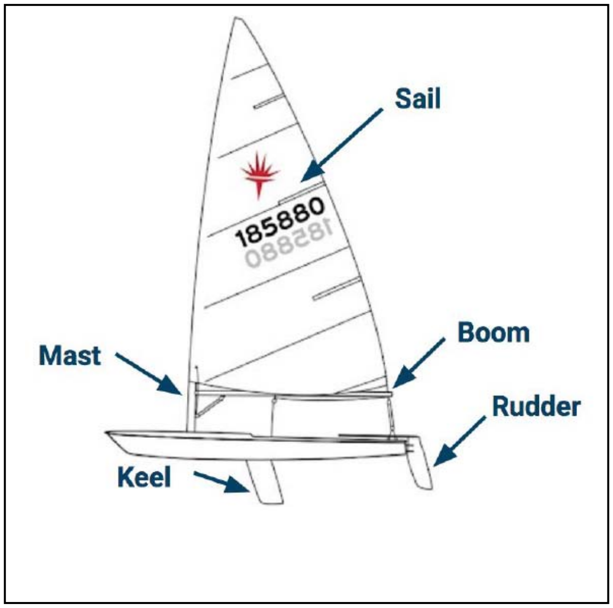
SMARTLAB Technical Report 2018-06.03

FINAL REPORT

Autonomous Hydrofoil Sailboat



by

V. Tran, B. Zhao, J. Leung,
M. Shi, and S. Bleveoet

Edited by: Drs. N. Sarigul-Klijn &Platzer

June 2018

smartlab@ucdavis.edu

SMARTLAB (Scaled Model Aerospace Research and Testing Laboratory)
Department of Mechanical and Aerospace Engineering
University of California, Davis
2132 Bainer Drive
Davis, California 95616

SMARTLAB is a research laboratory for scaled dynamic testing of new vehicles. This research laboratory was established by Professor N. Sarigul-Klijn in 1995 to conduct flight and ground testing of mobility systems.

For further information contact founding director Dr. Nesrin Sarigul-Klijn
via phone: (530)-752-0682, or email: nsarigulklijn@ucdavis.edu

How to cite this technical report as a source:

1. V. Tran, B. Zhao, J. Leung, M. Shi, and S. Bleveoet, Edited by: Drs. N. Sarigul-Klijn &Platzer, 2018, *Autonomous Hydrofoil Sailboat*, SMARTLAB Technical Report TR 2018.06.03, University of California at Davis, 2018.

TABLE OF CONTENTS

1. BACKGROUND

1.1 Introduction

With a growing international demand for clean energy, it is critical to explore the largely untapped resource of renewable wind energy over international waters. Over these large bodies of water, winds tend to be stronger, more consistent in magnitude, and hold greater potential for energy conversion. In response to this need for clean energy, the sponsors of this project have proposed an energy ship concept, which aims to incorporate a fleet of autonomous sailboats for harvesting oceanic wind energy via submerged hydrokinetic turbines.

Generally, manning a sailboat in fluctuating winds requires a high level of expertise and has significant risks and operational costs. Adding autonomous capabilities to a sailboat allows for these risks and expenses to be mitigated, but requires a robust control system and additional electronic and mechanical components. Therefore, this project focuses on the development of the autonomous control system and hardware design of a small-scale sailboat.

1.2 Sailing Terminology and Methodology

In the realm of sailing, there are various types of sailboats that have different geometries and functions. However, for this project, a roughly 1-meter long RC Laser sailboat was provided by the sponsors, and some of the essential components of the sailboat are labeled in Figure 1 below.

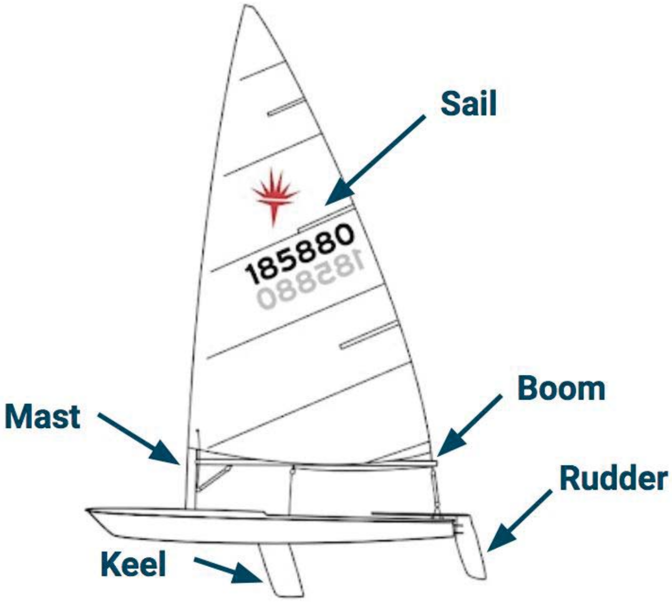


Figure 1. Overview of primary sailboat components.

Fundamentally, the sailboat's velocity and heading are controlled independently by two primary components: the rudder and sail. The rudder influences heading of the boat and the sail influences the boat's speed through a winch-driven servo, which collects and releases a tensioning cable in the presence of wind [1]. The keel is another crucial component, as it acts as a weighted ballast at the bottom of the boat, and generates a restoring moment to prevent the sailboat from capsizing.

It is important to note that for a given apparent wind angle, a boat's sail cable must be adjusted accordingly to achieve optimal velocity. In the case of the boat sailing directly into the wind, the boat is *in irons* or in *dead zone* as shown in Figure 2, and cannot gain any velocity [2]. Therefore, the sailboat must engage in an alternating zig-zag pattern (tacking) at an offset from the apparent wind to move upwind. Moreover, due to the contribution of the keel's restoring moment and apparent wind's interactions with the sail, a sailboat's maximum velocity is typically attained in the close reach configuration [3]. For designing a control system that intends to maximize velocity of the sailboat, it is integral to consider these sailing configurations.

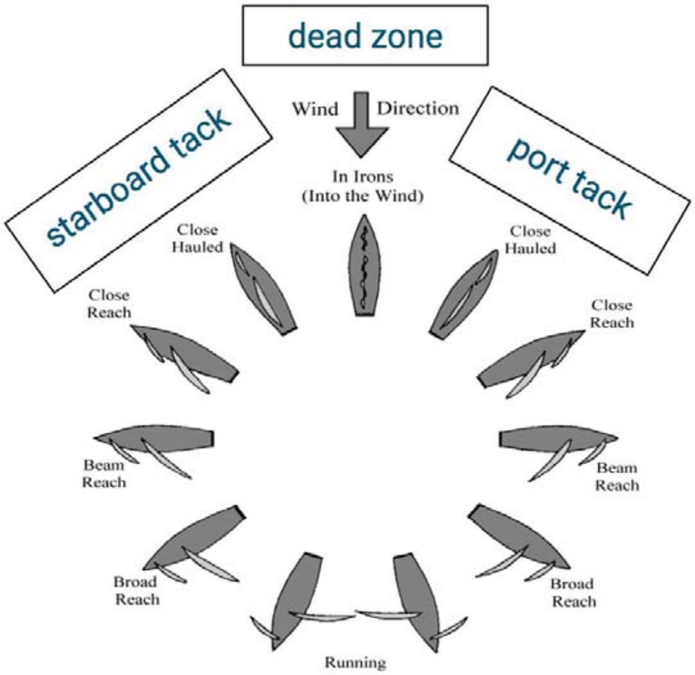


Figure 2. Various sailing positions for a given apparent angle of wind.

1.3 Design Requirements

For the actual design of the small-scale RC Laser sailboat's autonomous control system, the following requirements needed to be met:

1. Separate manual remote-control operation of the sail and rudder for safety
2. Automated sail control and rudder control to follow path planning scheme
3. Waypoint tracking with minimal deviation from desired path
4. Maximize boat speed with wind optimization
5. Data acquisition and storage of boat speed, boat position, and wind speed
6. Waterproof hardware enclosure to protect all components
7. Mechanically robust to handle wind speeds of 3 to 12 m/s

In order to satisfy the first requirement, the existing remote controller (transmitter) and the receiver were replaced with a new transmitter and receiver that have channel switching capabilities. With the use of a multiplexer, the autonomous mode and the manual mode can be switched by using the channel switch button on the transmitter. To perform path planning capabilities, the autonomous mode operates in waypoint navigation in order to to sail to pre-defined GPS coordinates. A proportional-integral (PI) rudder controller was implemented in order to minimize the error between the calculated heading and the actual heading. The autonomous mode also has capabilities to travel within a set boundary to optimize the wind and therefore maximize boat speed. A Raspberry Pi 3 was installed on the sailboat to store boat speed, boat position, and wind speed. Enclosures for electronics were designed to keep components safe from water damage. Finally, FEA analysis was performed on the Anemometer mount to ensure the prevention of mechanical failure. The repository of the code created for this project can be found on GitHub, which includes the scripts for waypoint tracking, wind optimization, sail control, and data acquisition, and also the CAD models for all hardware mounting equipment and enclosure [4]. An overview of the design layout is shown in the Appendix, Figure A.1 as well.

1.4 High-Level Design of a Robotic Sailboat

In small-scale autonomous sailboat projects that have been investigated by various research groups and university teams, there are typically five subsystems, which include: embedded intelligence, communication modules, actuators for the sail and rudder, a power supply, and a GPS or location provider for positional feedback [5]. This project follows a similar subsystem layout for designing the autonomous control system of the RC Laser boat. The following sections discuss the specific hardware utilized and the control algorithms implemented.

2. HARDWARE DESIGN

2.1 Hardware Components

The final hardware components were selected after reviewing commercially available hardware components and comparing a variety of factors including hardware performance, available documentation, ease of implementation, and cost. Table A below summarizes the selected hardware components and their functions. The modular architecture of the control system allowed for independent selection of the periphery components, such as the sensors and actuators. However, ease of integration with the onboard processors remained an important consideration. A majority of the GPS, anemometer, and multiplexer connections to the Arduino were soldered onto a protoshield, while the remaining connections were made through pins or screw terminals and secured with adhesives. The wiring diagram can be found in the Appendix, Figure A.2. Also, a comprehensive bill of materials can be found in the Appendix, section 6.3.

Table A. Selected hardware components and their functionalities.

COMPONENT	FUNCTION
Arduino Mega 2560	Process controls and interface with sensors and actuators
Raspberry Pi 3	Store data and provide future modularity
4-Channel Pololu Multiplexer	Switch between manual and autonomous modes
Davis Anemometer	Provide apparent wind speed and wind direction
LSM 303 Accelerometer/Magnetometer	Provide global heading data
Adafruit GPS Breakout Board	Provide global position and boat speed data
Non-Continuous Servo Motors	Actuate sail and rudder
Zippy 4000mAh 2S LiPo Battery	Supply power to all electronics

2.1.1 Onboard Processors

The onboard processors have two main purposes in the autonomous control system: data acquisition and real-time processing for navigation control. The Raspberry Pi 3 is utilized for data acquisition and storage, while the Arduino Mega 2560 is used for processing the control schemes.

More specifically, the Arduino Mega 2560 controls and interfaces with the sail and rudder servos, the sensors, and the multiplexer communication module which enables hardware-in-the-loop for safety operations. The Raspberry Pi 3 stores data that is printed in the serial monitor in CSV format and provides modularity for more computationally intensive operations that may be developed in future works, such as object avoidance with additional sensors or more optimized path planning schemes.

2.1.3 Sensors and Actuators

The autonomous control system has a number of sensors and actuators to inform the controller about the boat's environment and navigate through the water. The Adafruit Global Positioning System (GPS) breakout board provides latitude, longitude, and velocity of the boat. The LSM 303 magnetometer acts as a digital compass and provides the system with global heading, and also houses an accelerometer that may be utilized to verify boat velocity with more accuracy in future works. The Davis Anemometer provides both the apparent wind speed and apparent wind direction data via cupping arms and a mounted wind vane. The servo motors are rotary actuators that allow for control of the angular position of both the sail and the rudder. The existing servos on the RC Laser Sailboat provide adequate torque and power for the autonomous control system.

2.1.4 Power Supply

The power supply allows for appropriate power to be supplied to the Raspberry Pi 3 and the Arduino Mega, which powers all the other components. Lithium polymer (LiPo) batteries are popular for RC applications, but have relatively high costs and require maintenance. Furthermore, the LiPo battery requires a separate charger that balances the charge to each cell in the battery pack. However, LiPo batteries fulfill all of the desired performance requirements, including a large C-rate range of up to 90 C which easily enables a charge time under 1 hour.

Several factors were accounted for in the sizing of the power supply unit. The two primary ratings for sizing a LiPo battery were the batteries' voltage and capacity. Along with the LiPo battery, a UBEC voltage regulator was used to step the voltage down from 7.4 V to 5.25 V to power the electronics. Calculating the capacity was done by summing the total current draw from all hardware. This amount came out to be approximately 2865 mA. A table of the calculations can be found on the Appendix, section 6.4. To avoid any damages due to potential thermal expansion, a capacity larger than maximum current draw was desired (under worst-case electrical loads, a capacity of 4000 mAh can last for as long as 2 hours).

2.2. Mounted Hardware

All of the mounted hardware housing was designed in SolidWorks and fabricated using 3D printers. The solid models were first converted to STL (.stl) files in SolidWorks. Then the STL file were converted into G-code after being imported into Cura, a 3D processing software tool. The G-code was utilized by the 3D printer to print the model with a predefined resolution based on the nozzle size, print layer height and percent infill. The nozzles used for all of the prints ranged from 0.4 mm to 0.8 mm. The layer height varied between 0.1 mm and 0.2 mm. Lastly, the percent infill ranged from 20% to 90%. These percentages were dependent on the application, as waterproof prints used 90% infill, while strictly mounting parts used 20% infill.

2.2.1. Hardware Enclosure

Protecting the hardware from water ingress is a critical design feature for ensuring security and longevity of the hardware. The final enclosure design, displayed in Figure A.3 in the Appendix, met the design requirement of protecting the onboard electronics through a three part assembly. The first part was the flexible O-ring, which was sized with an inner diameter (ID) of 4-¼ inches, to maximize compactness relative to the existing enclosure opening (with an outer diameter [OD] of 4 inches). The second part of the assembly included the fitting around the existing enclosure opening, which acts as a cap adapter. The dimensions of the fitting were determined iteratively to accommodate two critical clearances shown in Figures A.4 and A.5, which depicts the position of the O-ring and the static radial seal design based on AS568 Aerospace Size Standard for O-rings, respectively [6]. As shown in Figure A.6, the O-ring groove, which is red, was placed at the lower half of the fitting with female threading on the upper half. The third part is the cap with meshing male threading that secures the inner diameter surface compressing the outer diameter of the O-ring creating a proper seal between 20% and 40% compression, as demonstrated in Figure A.15. The thickness of the cap was increased to accommodate the cable gland, shown in Figure A.7, that requires a minimum top thickness for its 6 mounting screws to hold. The cap passthrough, displayed on the Top View of Figure A.16, was sized to be coincident with that of the cable gland.

The goal of the enclosure was to protect against worst case scenarios during sailboat testing. This included high winds, sharp turns, and drastic heeling. Waterproof testing based on the Ingress Protection Standards, was conducted to give an appropriate ingress rating [7]. The enclosure assembly waterproof testing results cleared for IPX7 rating, which matched the rating of the cable gland. Lastly, marine silicone was applied at any potential openings, which acted as both an adhesive and sealant. All sailboat tests on water with varying weather conditions have maintained zero water ingress.

2.2.2. Anemometer Mount

The mount that shipped with the anemometer did not fit the build for the model LASER sailboat, so a redesigned mount was necessary. This was accomplished by repurposing the guard for the front spool of the sail to accommodate the anemometer mount. The final design is displayed in Figure A.14. To ensure the mount was mechanically sound, two different FEA load cases were simulated to test against mechanical failure. Polylactic Acid flexural strength is typically between 48 and 110 MPa. The first simulation was by applying 3 times the

Anemometer vertical load, based on its mass, at the Anemometer attachment location. The maximum Von Mises stress was 21.24 MPa which is less than half the minimum PLA flexural strength. Those results are displayed in Figure A.10 and Figure A.11. The second simulation was done by assuming a uniform pressure load based on a 12 m/s constant wind applied horizontally and assuming a conservative drag coefficient of 1.0. The results are displayed in Figure A.12 and Figure A.13.

2.2.3. Magnetometer Mount

The design of the magnetometer mount consists of a mating plate and cubic container assembly. The height of the cubic container, shown in Side View of Figure A.20, accounts for secure wire connections to the magnetometer. The four holes feed the wires through the enclosure. A vertically extruded base plate allows the magnetometer board to rest flat with the surface of the boat. Marine silicone was applied to the mount where the wires feed through, between the mating surfaces of the box, and at the top to provide static waterproofing. These features can be observed in Figure A.20.

3. CONTROLLER DESIGN

3.1 Data Logging

For testing and research proposes, a data logging Python script can be run on the Raspberry Pi. This works by reading the serial port as the Arduino is running and saving the data to a comma-separated values (CSV) file. The advantage of running it on the Raspberry Pi is that there is an interface to look at the onboard data while testing or operating in more remote locations where wireless connection for data transfer is limited. With the SD memory card, the Raspberry Pi can hold a large data file for longer deployments.

3.2 Manual Mode

One of the design requirements necessary for this project was to have the sailboat operate manually, which gives the user complete control over the sailboat when necessary, such as in emergency cases or when launching the sailboat from onshore. Switching from autonomous to manual mode was done with the use of a 4 channel multiplexer. The multiplexer has two inputs (a slave and a master), one output, and one signal pin. The output draws from the slave source when the signal is turned on. Conversely, the output draws from the master source when the same signal is turned off. The signal pin is attached to channel 5 of the RC receiver, which is controlled by a button on the top right corner of the transmitter. This conveniently allows the user to switch modes with a press of a button. Using the multiplexer avoids the need for any software interrupts and allows for both the Arduino and the RC receiver to work simultaneously.

3.3 Waypoint Tracking

To demonstrate the autonomous capability of the sailboat, waypoint tracking was selected to assess whether the boat can follow a predetermined path with minimal deviation. The different waypoints were predetermined visually on Google Earth Pro for the sailboat follow continuously. Each waypoint was set at approximately 20 meters away from adjacent points to account for GPS error, a marginal error for the sailing algorithm, and physical constraints of the testing site. To account for possible error from GPS readings, which is roughly 3 meters [8], waypoints were given a margin of five meters. In other words, if the sailboat reached within five meters to the target waypoint, the system would have registered that the waypoint was reached. Figure 3 depicts the defined waypoints and defined margin of error.

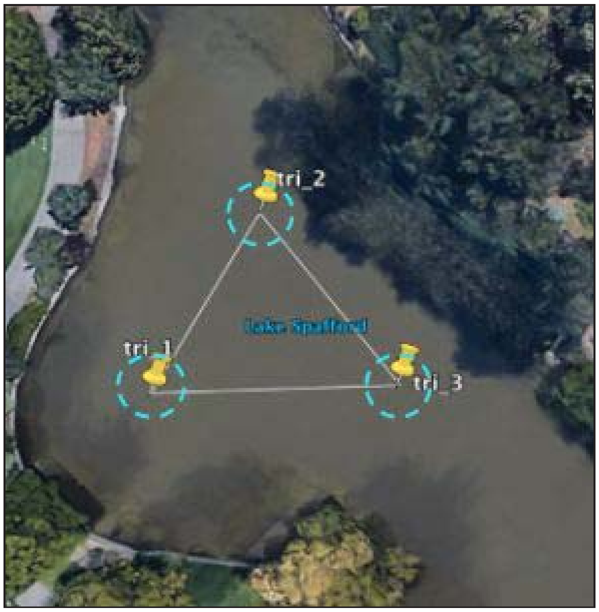


Figure 3. Predefined waypoints located at (Lake Spafford, UC Davis) mapped out on Google Earth Pro. The dotted circles around the waypoints represent the margin of error.

In waypoint tracking, the three waypoints is pre-defined and stored in a vector. The distance between the location of the sailboat and the target waypoint is calculated. If this distance is less than five meters, the next waypoint in the vector is set as the new target waypoint. If the sailboat has not reached the waypoint, then apparent wind direction, global position, and global heading (inputs from the anemometer, GPS, and magnetometer, respectively) are calculated. Both the rudder and sail controls are then applied and sent to their respective servos to control the sailboat. If the anemometer indicates that the sailboat is in the deadzone for more than three seconds, a tacking algorithm is then implemented. The tacking algorithm is depicted further in Figure 4. After completing all this, the distance between the sailboat and the target waypoint is calculated again, re-entering the algorithm. The waypoint tracking algorithm is shown and illustrated further in Figure 5.

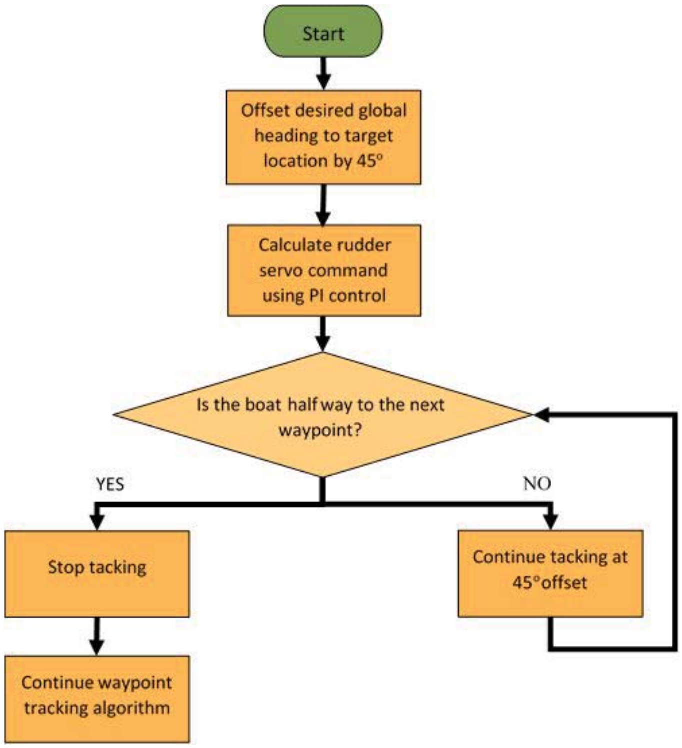

Start
Offset desired global heading to target location by 45º
Calculate rudder servo command using PI control
Is the boat half way to the next waypoint?
YES
NO
Stop tacking
Continue waypoint tracking algorithm
Continue tacking at 45°offset

Figure 4: Tacking Algorithm

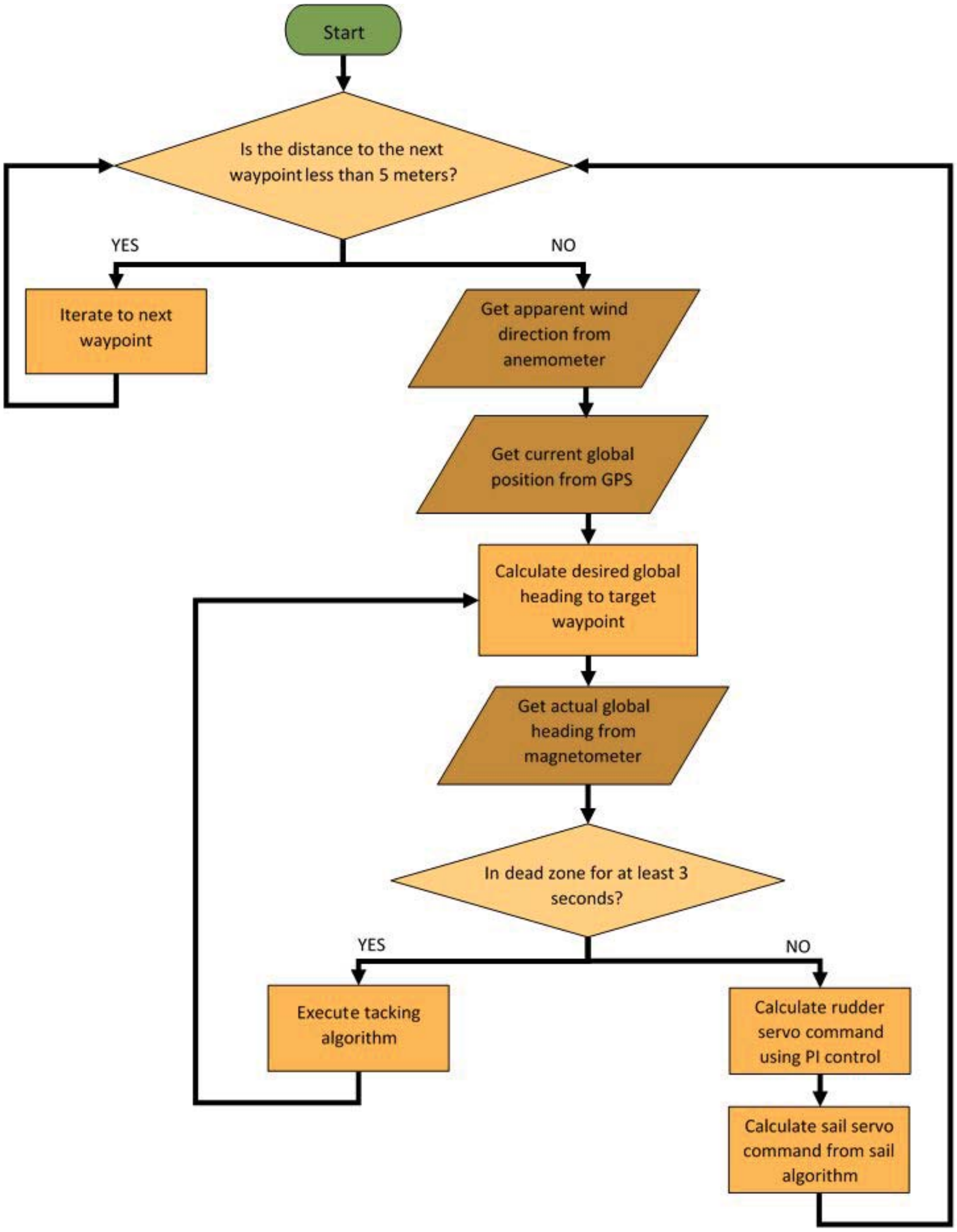

Start
Is the distance to the next waypoint less than 5 meters?
YES
NO
Iterate to next waypoint
Get apparent wind direction from anemometer
Get current global position from GPS
Calculate desired global heading to target waypoint
Get actual global heading from magnetometer
In dead zone for at least 3 seconds?
YES
NO
Execute tacking algorithm
Calculate rudder servo command using PI control
Calculate sail servo command from sail algorithm

Figure 5: Waypoint tracking algorithm.

3.4 Wind Optimization

The goal of the autonomous wind optimization mode is to maximize the boat speed with the purpose of harvesting wind energy in a specific area. This uses the fact that the maximum boat speed occurs when the boat travels into the wind at a 45^o angle and, when going with the wind, the maximum speed occurs when the boat is moving in the same direction as the wind. To take advantage of this while staying within a specified radius of the preset origin, the boat will travel in a “bowtie” path, going 45° against the wind, 0° with the wind, -45° against the wind, and, finally, 0° with the wind to complete one circuit.

In wind optimization mode, the controller, as seen in Figure 6, determines when to turn by calculating if the current distance, or radius, from the origin is greater than the predefined maximum. The controller then determines in which direction relative to the wind the boat should turn based on which leg of the circuit the boat is currently running. While the boat is turning, the controller temporarily stops checking if the boat is out of bounds until after it is within the boundary again. This partially accounts for the boat’s turning radius and any additional drifting. The desired angle is then sent to the rudder controller to generate the rudder servo motor command. Because this is based on the apparent wind angle, the algorithm works when the wind direction changes, but is limited by physical factors as seen during testing.

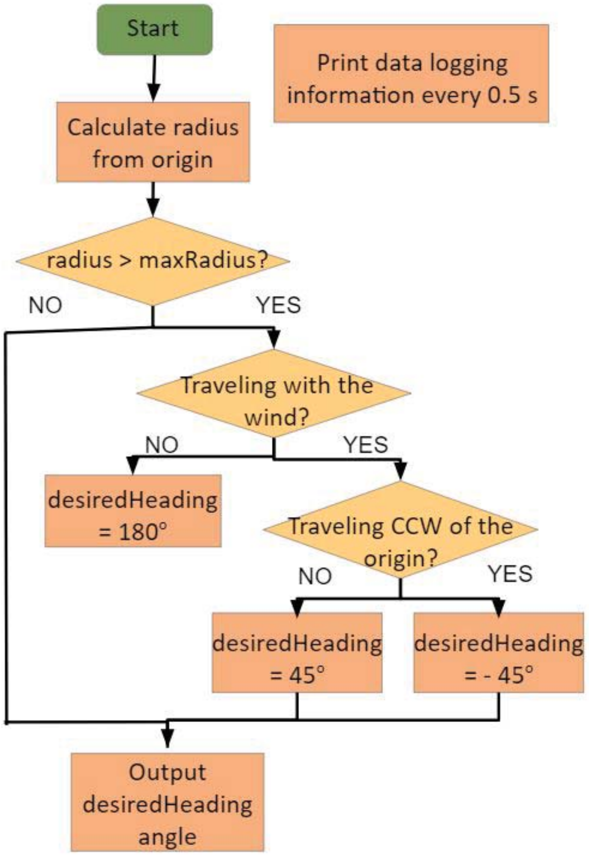


Figure 6. Flow diagram for Wind Optimization Mode.

3.5. Rudder Control

Since the rudder servo motor independently controls the orientation of the boat, while the sail servo motor controls the amount of rope slack given to influence velocity of the boat, both the rudder controller and sail controller were designed independently to feed different inputs to the microcontroller. For the rudder controller, PI control was implemented to control the boat's orientation by minimizing the error between the current heading and desired heading. The rudder angle was then saturated to ensure that the rudder rotated at a max angle of 45 degrees and -45 degrees. PI control was selected over other designs, such as PID, due to its simplicity and high levels of instability during sailing, which intensifies noise at high frequencies. Initial attempts to find the Kp and Ki constants were done through the use of a customizable nonlinear 4-DOF sailboat simulation tool from the University of Denmark [9]. Many assumptions were made during these simulations, including constant wind speeds and negligible damping from the waves. Ki and Kp values were further fine tuned through physical experiments (testing the sailboat on the water). A diagram of the PI controller algorithm is listed below in Figure 7**.**

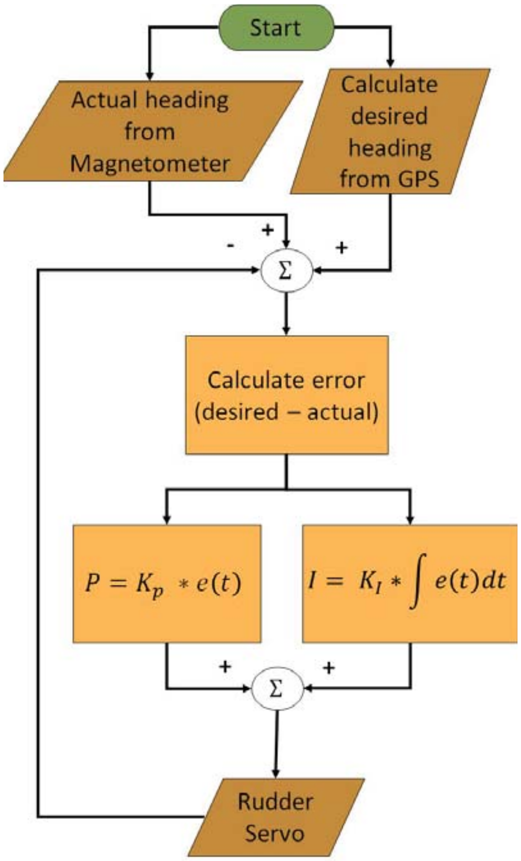

Start
Actual heading from Magnetometer
Calculate desired heading from GPS
+
-
+
Σ
Calculate error (desired – actual)
$P = K_p * e(t)$
$I = K_I * \int e(t)dt$
+
+
Σ
Rudder Servo

Figure 7. Rudder control algorithm implementing PI theory to reduce orientation error of the sailboat.

3.6 Sail Control

The angle of the sail determines the amount of wind captured by the sail and in turn, the sailboat velocity. Maximum wind capture was considered when designing the sail control algorithm, shown in Figure 8. Also, the sail control algorithm considers two possible types of motion: steady sailing and turns. The turning angle is calculated as the absolute difference between the previous and current heading of the sailboat, with the headings obtained from the rudder controller that is used in both wind optimization and waypoint tracking. In the sail control algorithm, the turning angle is defined as a difference of 45°. The second input into the sail control is the apparent wind direction sensed by the onboard anemometer. The raw potentiometer values from the wind vane of the anemometer sensor needed to be saturated to output angles between -179° and 180°, with 180° representing the dead zone. The curve fit function used to map the raw wind vane values to apparent wind angles is shown in the Appendix, Figure A.8. The curve-fit function will depend on the specific anemometer sensor used and the desired range of output angles.

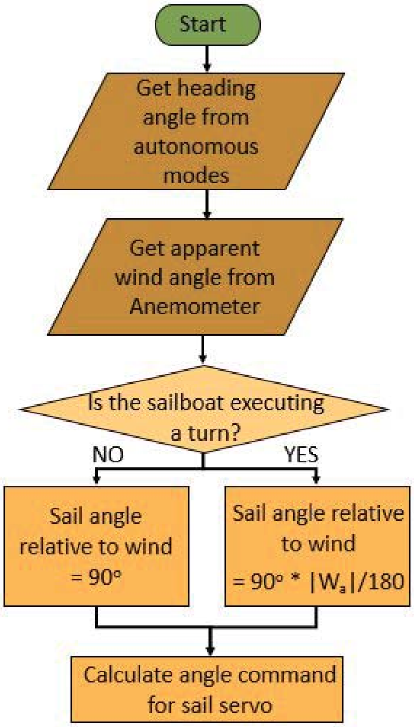

Start
Get heading angle from autonomous modes
Get apparent wind angle from Anemometer
Is the sailboat executing a turn?
NO
YES
Sail angle relative to wind = 90°
Sail angle relative to wind = 90° * |W$_a$|/180
Calculate angle command for sail servo

Figure 8. Algorithm for sail control to achieve maximum wind capture during steady sailing.

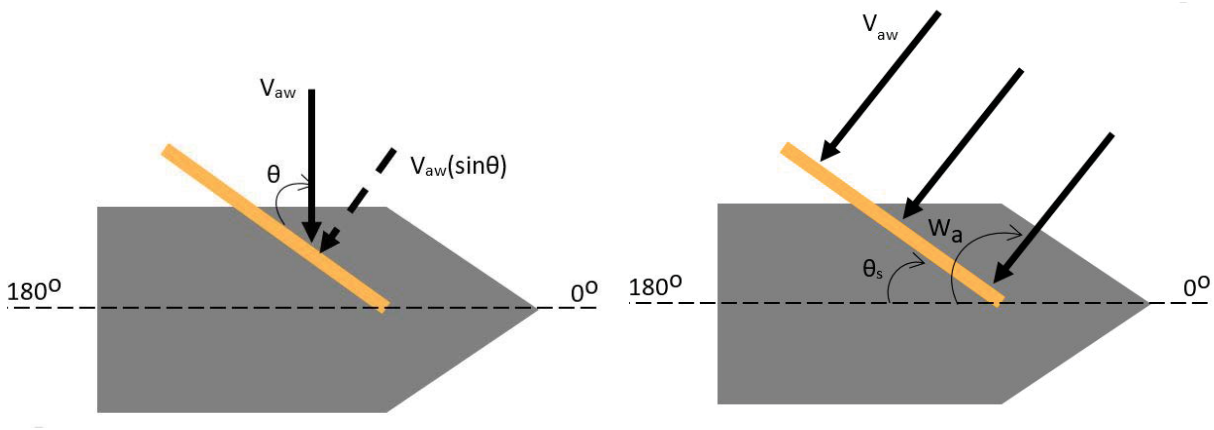


Figure 9. Sail captures perpendicular apparent wind component (Left).
Maximum sailboat velocity when sail set perpendicular to apparent wind direction (Right).

The wind captured by the sail is the perpendicular component of the apparent wind, V_{aw}, as shown in the left of Figure 9, with the angle between the sail and apparent wind denoted as θ. Therefore, the sail obtains maximum wind capture when the angle between the sail and apparent wind is equal to 90^o, as shown in the right of Figure 9. In the case of steady sailing when the calculated turn angle is less than 45^o, the sail control maintains this 90^o relative angle by calculating the desired sail angle, θ_s, using Equation 3.1. Only absolute values for the apparent wind direction, W_a, need to be considered because the wind naturally pushes the sail in the correct orientation. Then, the desired sail angle is calibrated to a sail servo command in degrees using another linear curve-fit function, shown in Appendix A.9.

$$\theta_s = \left| |W_a| - 90^o \right| \tag{3.1}$$

In the case of turns, when the calculated turning angle is greater than or equal to 45^o, the sail control calculates a desired sail angle that is less than 90^o and proportional to the apparent wind direction, shown in Equation 3.2. The relative sail angle needs to be less than 90^o during turns because the boat needs to slow down to prevent capsizing or instability. The sail control during turns is based off of the sail control law used in the Santos paper [1]. The sail control does not incorporate a direct feedback system since the servos do not provide feedback, i.e. they do not have built-in encoders.

$$\theta_s = 90^o * \frac{|W_a|}{180^o} \tag{3.2}$$

4. RESULTS

4.1 Waypoint Tracking

Several tests were conducted on Lake Spafford in Davis, CA. For each of the tests, the sailboat was initially launched in manual mode. Once the sailboat was away from shallow water, the system was switched to run autonomously. The first few tests ran unsuccessfully, with the system encountering many different problems, the first of which included faulty wire connections (specifically the power connections to the multiplexer), causing inconsistent servo movements. Another problem encountered was from the GPS, which often failed to get a reliable fix due to magnetic interference of neighboring components. This problem was solved by installing a waterproof antenna, which allowed the GPS to consistently obtain a fix in a shorter amount of time.

Several other issues encountered were related purely to software: errors in the code with memory access, logic errors, failed test cases, version control, etc. After many test runs in the water and several dry runs – with waypoints simply set on land – the sailboat successfully traveled to all three waypoints. The sailboat was first launched from shore, and then hit the waypoints in succession. Once waypoint three (final waypoint) was reached, the sailboat then returned to waypoint 1 to begin looping through the triangular path again. The trajectory of the sailboat can be viewed below in Figure 10. Overall, the sailboat had minimal deviation in the path and reached speeds up to 7.40 knots from waypoints 3 to 1, during wind speeds of 6-10 mph at Lake Spafford. However, it is important to note that the sailboat took a much longer time to reach waypoint 2, since the sailboat traveled head on into the wind (*dead zone*) reaching winds speeds as low as 0.07 mph. In response, a tacking algorithm was written and tested in attempts to obtain forward velocity for the boat when it reaches a dead zone, which was listed previously in Figure 4.

The tacking script did not successfully work for all cases of the wind direction, as it was written with an assumption that the wind direction would be relatively constant between two waypoints during a dead zone maneuver. However, it was observed that on certain test days, the wind exhibited highly non-stochastic behavior, with the wind direction and wind speed rapidly changing, as the wind vane even observed a full 180° over the course of 1 second.

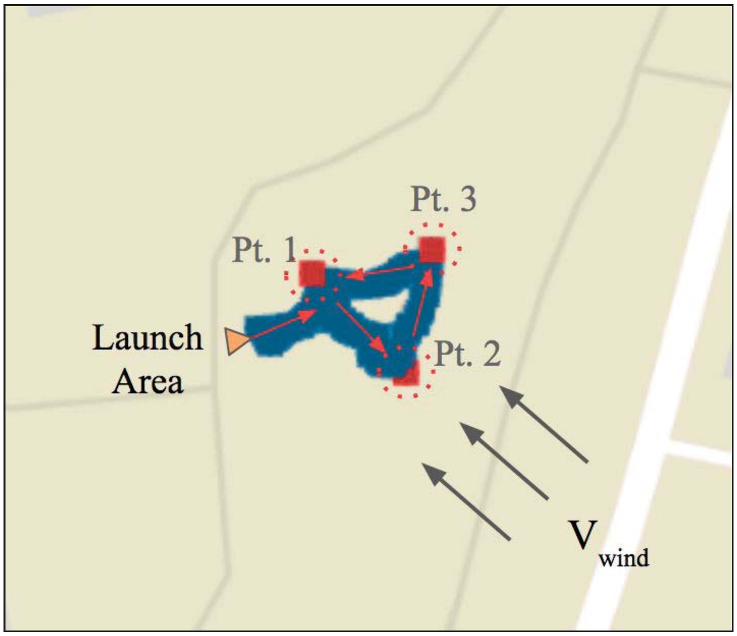


Figure 10. Visualization of GPS data from the triangular waypoint route, with average wind direction and sailing direction denoted.

4.2 Wind Optimization

Wind Optimization tests were also conducted on Lake Spafford at UC Davis. Similarly, the sailboat was pushed from shore in the direction of the origin and manually navigated to within the specified maximum radius. Then, the sailboat was released into autonomous mode and would orient its heading at a predetermined angle relative to the wind, in this case, 45^{o}. The tests were relatively successful in that the sailboat was able to maintain a constant angle with the wind for maximizing speed. From onshore observations, the boat was maintaining a significantly faster boat speed than what was observed with just manual mode; however, turning became an issue when the boat would reached the end of the boundary. This was due to many issues including faulty wiring and wind vane errors. Most hardware issues were similar to those of Waypoint Tracking, but additional wind vane errors came into play. One issue with the wind vane occured when wind speeds were too low for the wind vane to register the wind direction. Another issue was inaccurate and/ or erratic wind direction readings when the boat itself was turning, so that the anemometer could not properly gauge how far the boat has turned or needs to turn to navigate itself back inside the boundary.

One potential solution is to use the magnetometer and ensure that the boat turns through a certain angle from the boat's initial turning heading, which is the measurement taken at the start of the turning process when the boat is at or beyond the set maximum radius of the boundary. After, the the system can continue using the wind direction data to correct the boat's trajectory and optimize for speed again. Another potential solution is incorporating Waypoint Tracking directly to navigate the boat back into the boundary before releasing it to Wind Optimization mode again. At this point, these potential solutions require more testing and debugging, but regardless the control system can still maintain a constant angle with the wind to achieve maximum boat speed, just not within a specified boundary. In testing, the boat was turned around manually to avoid obstacles and to return the boat within the boundary.

From the May 28, 2018 test day at Lake Spafford, the anemometer measured true wind speeds between 0 and 1.96 knots, with an average of about 1.076 knots. Because the system does not have a separate onshore anemometer, this was measured using the mounted anemometer before launching the boat, which currently has a resolution of 0.65 knots. On the water, the average apparent wind speed was measured to be between 0 and 6.52 knots with an average of 1.173 knots, while the boat speed ranged from 0 to 20.91 knots with the average boat speed was measured to be about 1.215 knots. The relatively high recorded boat speed may be due to GPS location error from which the speed is derived from or perhaps the boat was actually turning or moving at that high speed, which seems improbable based from qualitative, onshore observations. Together, the data would indicate a 9.08% increase in average apparent wind compared to average true wind. However, between the average apparent wind speed and boat speed, there is a discrepancy of at least 0.0416 knots, since the boat speed should not exceed the apparent wind speed, even though it can optimally exceed the true wind speed. Given these evaluations, the data is inconclusive, and the system will require further testing with higher winds and perhaps more accurate sensors. It is finally noted that for testing both the waypoint tracking and wind optimization schemes, the sailboat's hardware proved to be fully waterproof, and the hardware mounts did not have mechanical failure.

Table B: Wind Optimization Testing Data Summary

	Minimum Speed (knots)	Maximum Speed (knots)	Average Speed (knots)
True Wind	0.0	1.96	1.076
Apparent Wind	0.0	6.52	1.194
Boat	0.0	20.91	1.215

5. FUTURE WORKS

5.1 Small-scale Sailboat Improvements

If given more time to further develop the small-scale sailboat, a few points would be worth pursuing. First, designing a PCB board for the existing electronics would make the wiring safer and much easier to manage. Second, improvements to the tacking algorithm can be made to account for all different scenarios of the varying wind angles. Third, object avoidance could be added to the sailboat, giving more flexibility to operate the sailboat wherever, without any concern that the sailboat would collide into anything. Lastly, more testing could be done at the Sacramento Aquatic Center located in Sacramento, CA, where longer trajectories with more waypoints can be tested. Refer to Figure 11 below for a map of the waypoints.

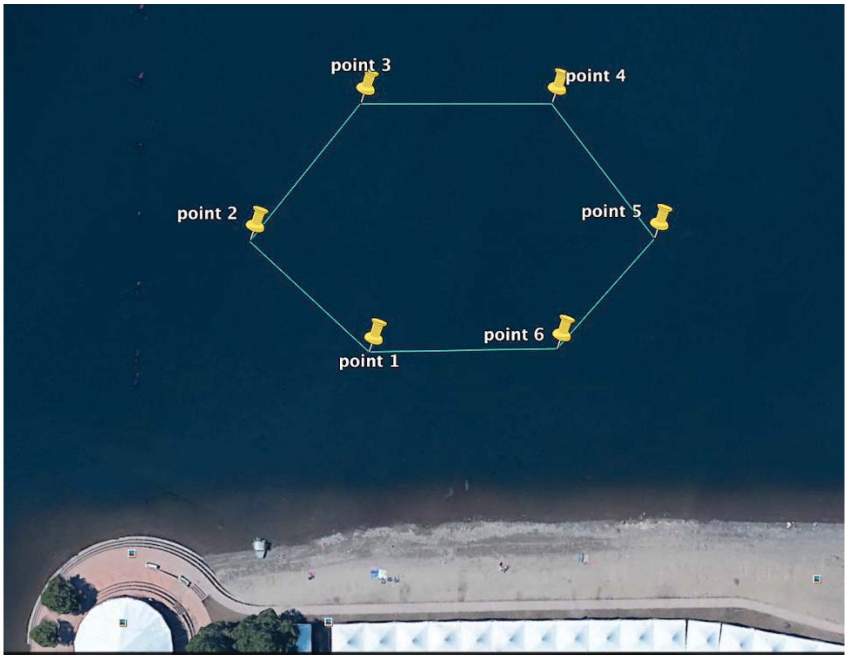


Figure 11. Predefined waypoints located at Sacramento Aquatic Center mapped out on Google Earth Pro.

5.2 Nonlinear Simulation for Controller Design

The development of an autonomous controls system for a full-scale sailboat will require simulations with a mathematical model of sailboat dynamics rather than a trial-and-error approach with on-water testing. A nonlinear four degree of freedom (surge, sway, roll, and yaw) Simulink simulator is readily available for developing sail and rudder controllers with greater complexity [10]. The article by Xiao and Jouffroy of the University of Southern Denmark explains the physics and assumptions of the sailboat dynamics simulator [9]. Currently, the simulator parameters are set for a 12 meter class yacht. However, these parameters can be changed to match the parameters of other sailboats. Parameters that must be changed include the sailboat dimensions, moments and products of inertia, added mass coefficients, and lift and drag coefficients for the sail, rudder, and keel airfoils [9].

5.3 Theoretical Full-Scale Hardware

The hardware conversion to a full-scale Laser sailboat is purely conceptual and has not been analyzed, tested, or validated. A few major differences in sailboat anatomy between small-scale and full-scale include the physical properties, inertial matrices, and dimensions, which inevitably alter the dynamics of the system. In particular, the full-scale Laser sailboat also lacks a keel. The keel is a deadweight below the small-scale sailboat, which gives a restoring moment when the sailboat encounters heavy roll, and needs to be replaced by designing an automated ballast or modifying the full-scale Laser sailboat to install a fabricated full-scale keel as well as integration with a hydrokinetic turbine. Highly durable actuators would be installed with mounting hardware to control the sail and rudder in similar fashion to the final small-scale sailboat configuration. The existing sensors for the small-scale sailboat including anemometer and GPS could serve as a starting point as the anemometer is shipped with a full-scale sailboat mounting hardware, and the GPS error would have relatively less impact on position accuracy in relation to waypoint tracking. For the magnetometer, upgrading to an IMU would be more effective as surrounding metals are significantly less obtrusive compared to the extremely limiting enclosure space of the small-scale. The Arduino Mega and Raspberry Pi, collectively, provide enough processing power and data acquisition. Adding a larger SD card for storage would help for lengthier test cycles. A custom PCB would help with wire management, especially with troubleshooting soldered connections. Depending on the physical hardware setup of the full-scale sailboat, the control system could generally be adapted to the full-scale configuration.

6. APPENDIX

6.1 FIGURES

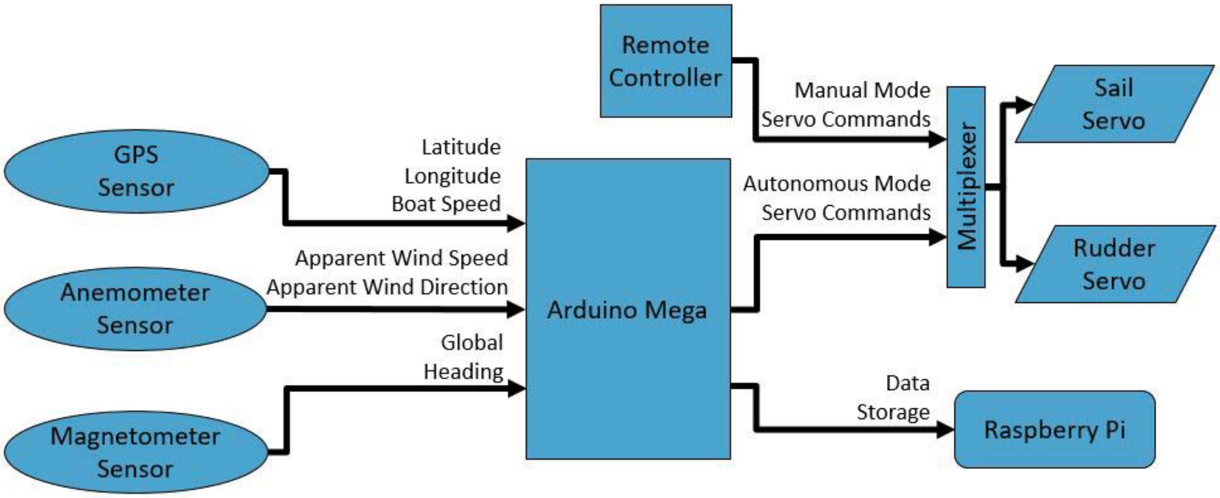


Figure A.1: System overview of hardware and functionalities.

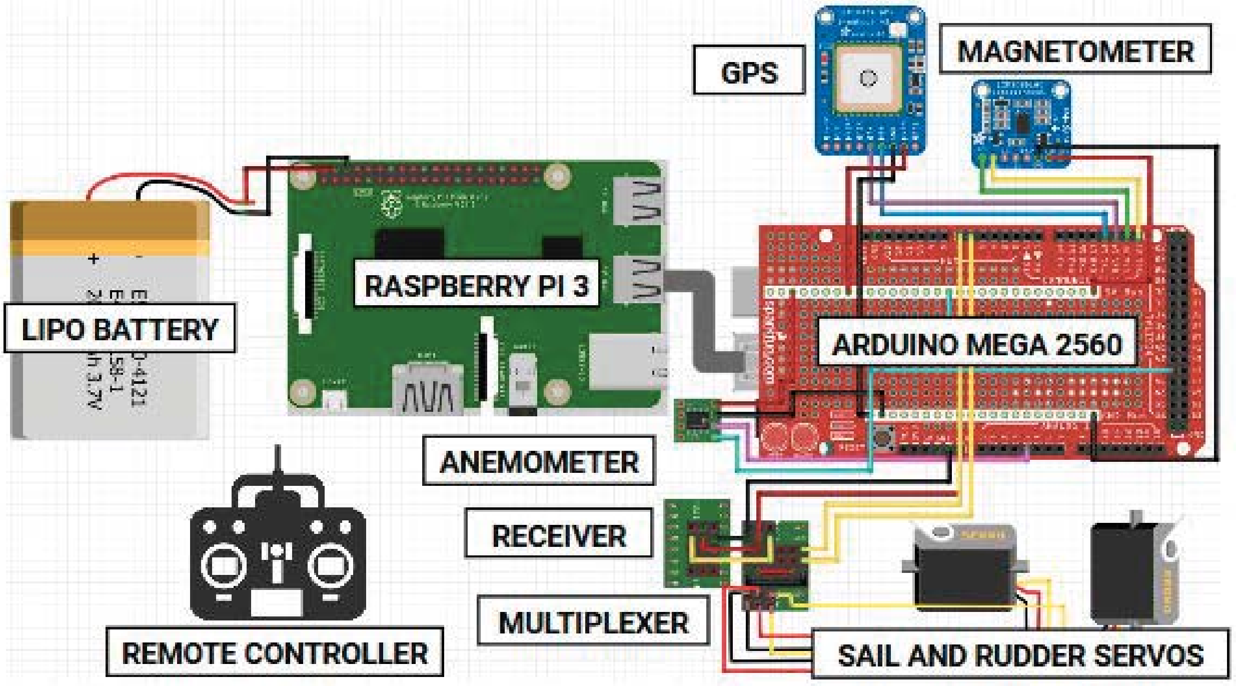


Figure A.2: Fritzing wiring diagram for autonomous controls system.

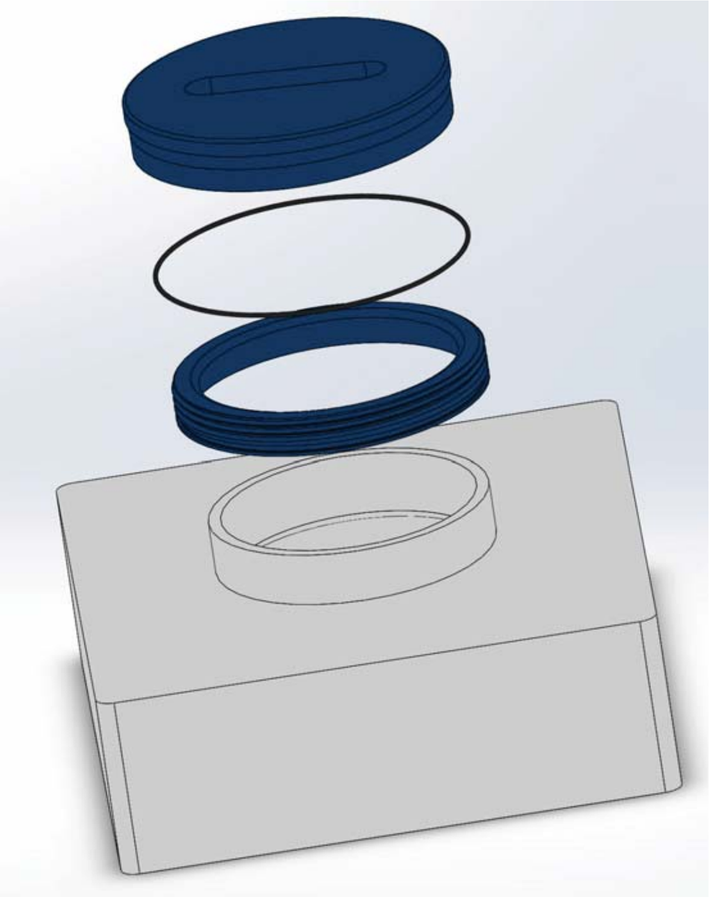


Figure A.3: Solidworks screen capture of exploded view for the enclosure assembly.

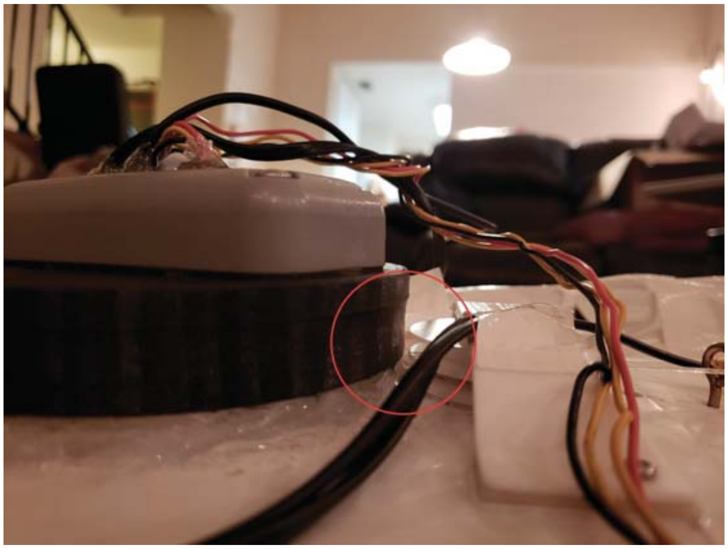


Figure A.4: Snapshot showing clearance of rear sail spool.

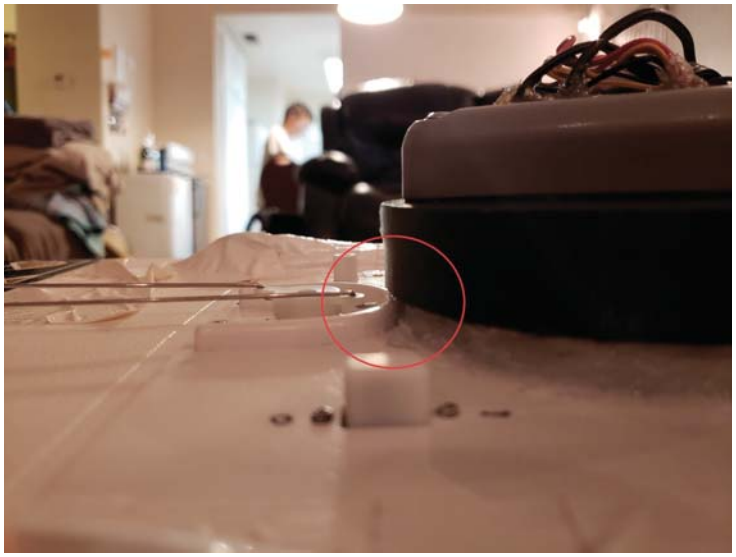


Figure A.5: Snapshot showing clearance of rudder guard.

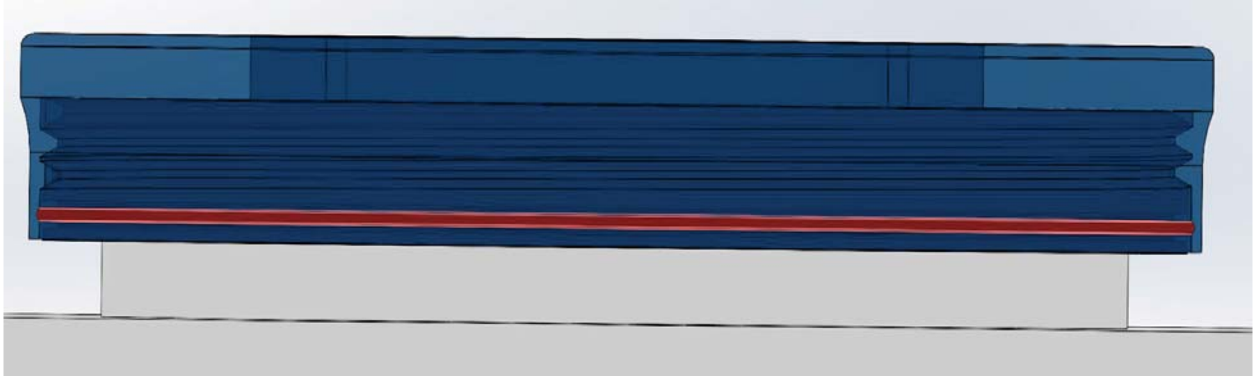


Figure A.6: Solidworks screen capture highlighting intended part interference between O-Ring and Cap.

Figure A.7: Stock image of cable gland for multiple wires.

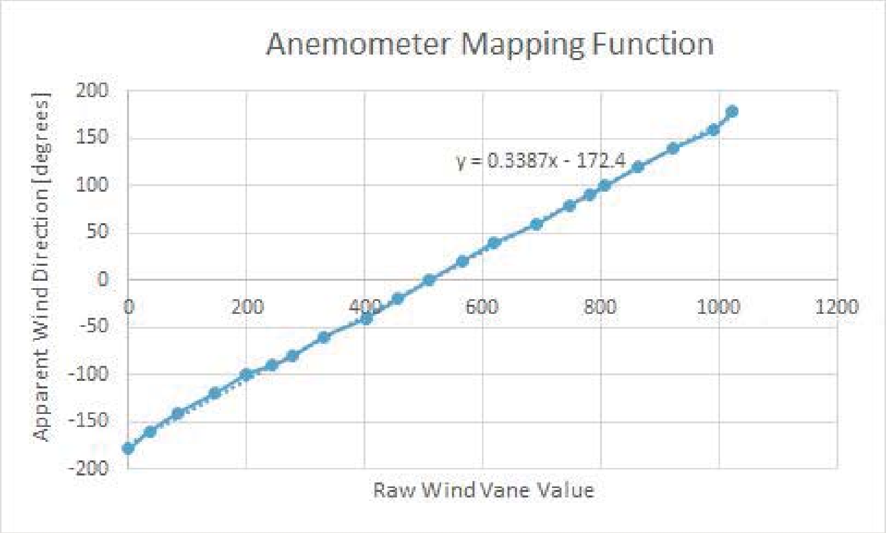


Figure A.8: Linear curve-fit function used to map raw potentiometer values from the anemometer wind vane to apparent wind direction angles in degrees.

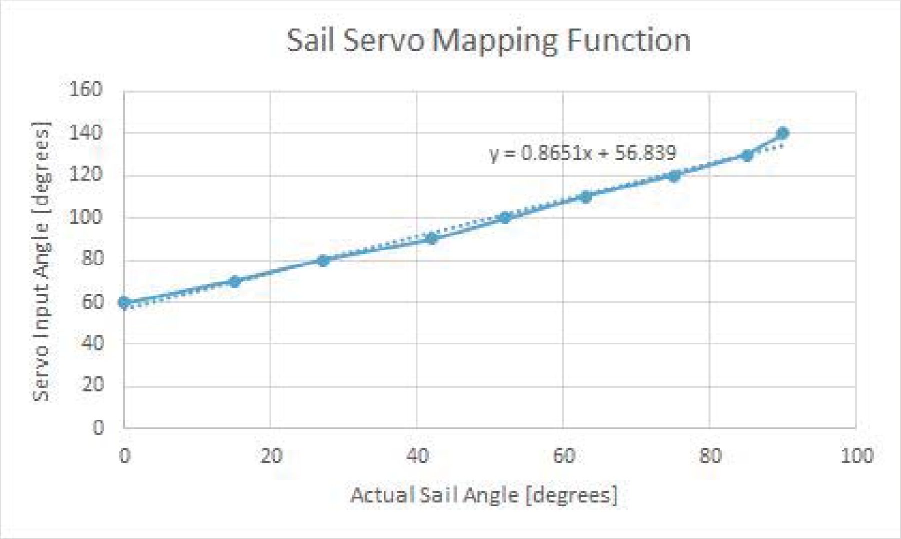


Figure A.9: Linear curve-fit function used to map actual sail angles to servo input angles.

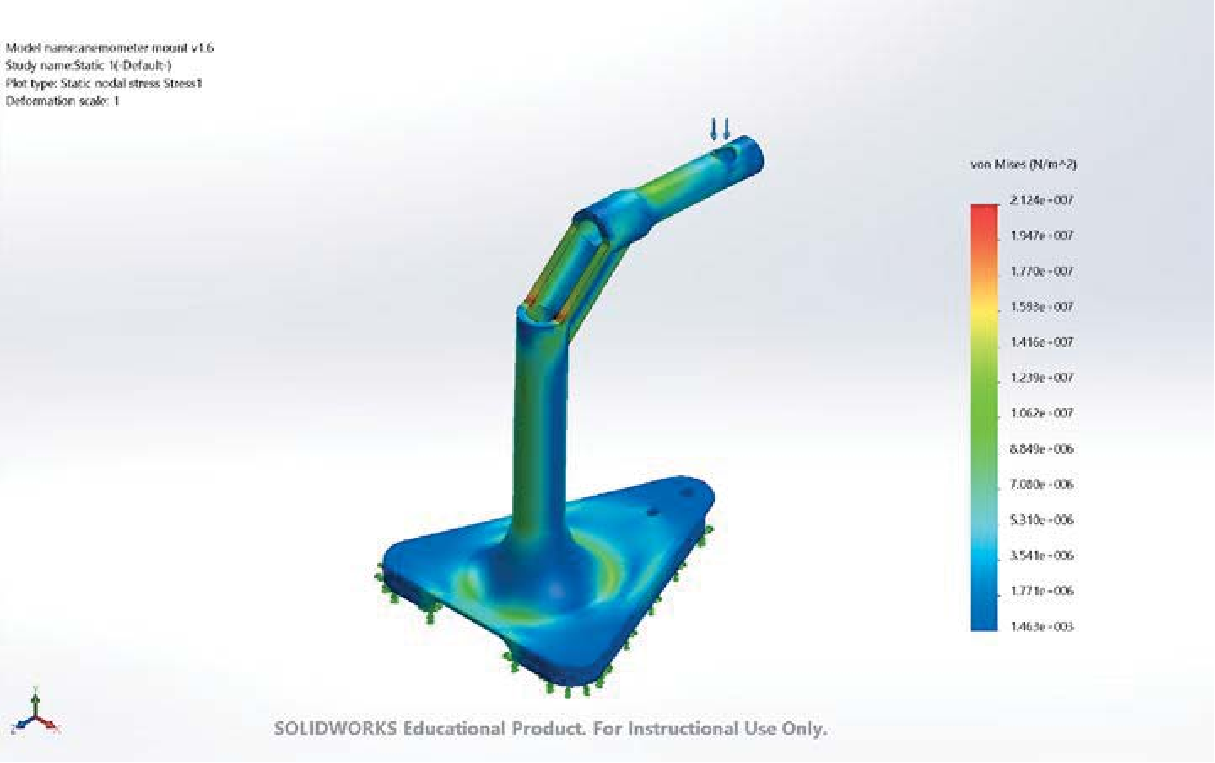


Figure A.10: FEA showing Von Mises Stress due to 3 times anemometer load case on the hole for mounting the anemometer.

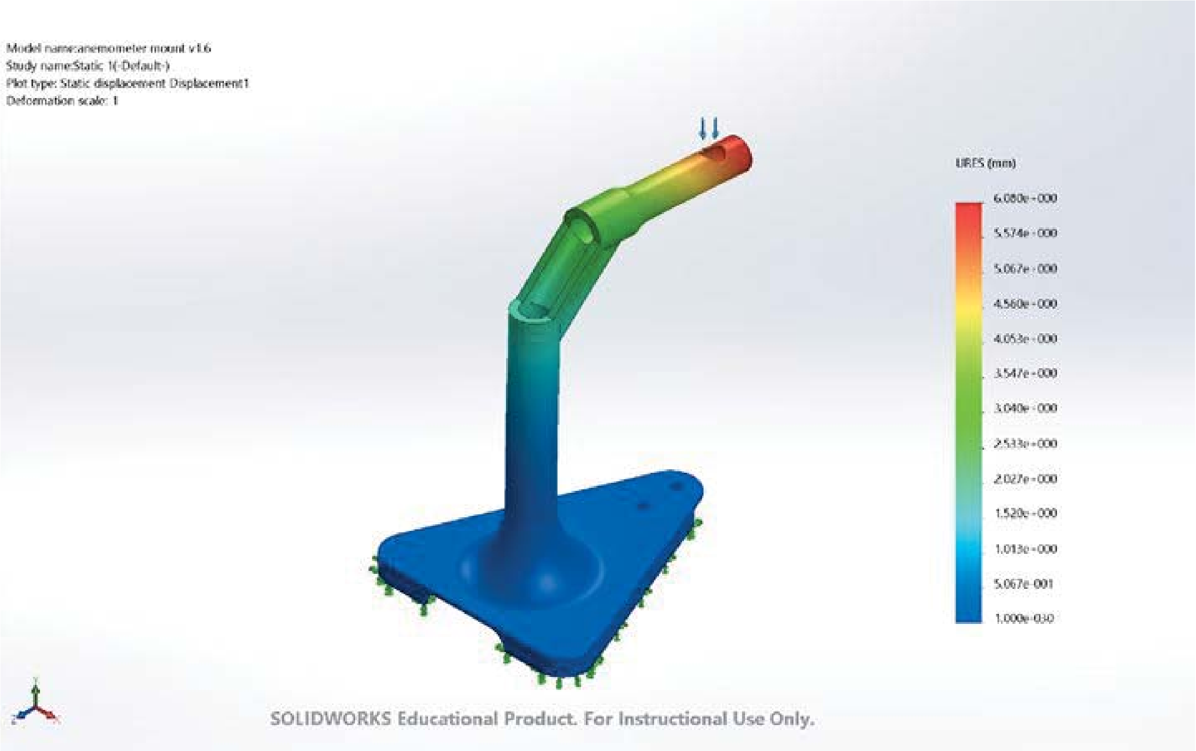

Model name:anemometer mount v1.6
Study name:Static 1(-Default-)
Plot type: Static displacement Displacement1
Deformation scale: 1
URES (mm)
6.080e+000
5.574e+000
5.067e+000
4.560e+000
4.053e+000
3.547e+000
3.040e+000
2.533e+000
2.027e+000
1.520e+000
1.013e+000
5.067e-001
1.000e-030
SOLIDWORKS Educational Product. For Instructional Use Only.

Figure A.11: FEA showing Displacement due to 3 times anemometer load case.

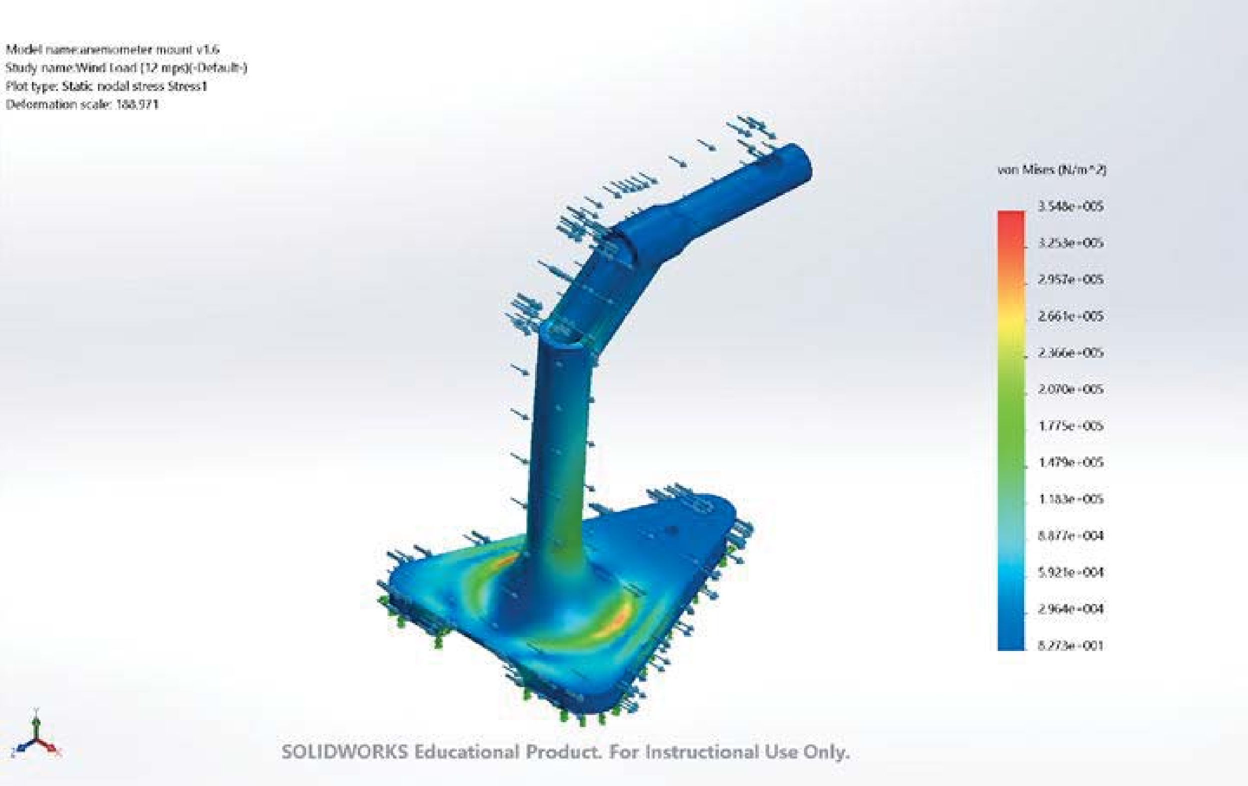

Model name:anemometer mount v1.6
Study name:Wind Load (12 mps)(-Default-)
Plot type: Static nodal stress Stress1
Deformation scale: 188.971
von Mises (N/m^2)
3.548e+005
3.253e+005
2.957e+005
2.661e+005
2.366e+005
2.070e+005
1.775e+005
1.479e+005
1.183e+005
8.877e+004
5.921e+004
2.964e+004
8.273e+001
SOLIDWORKS Educational Product. For Instructional Use Only.

Figure A.12: FEA showing Von Mises Stress due to constant wind load of 12 m/s.

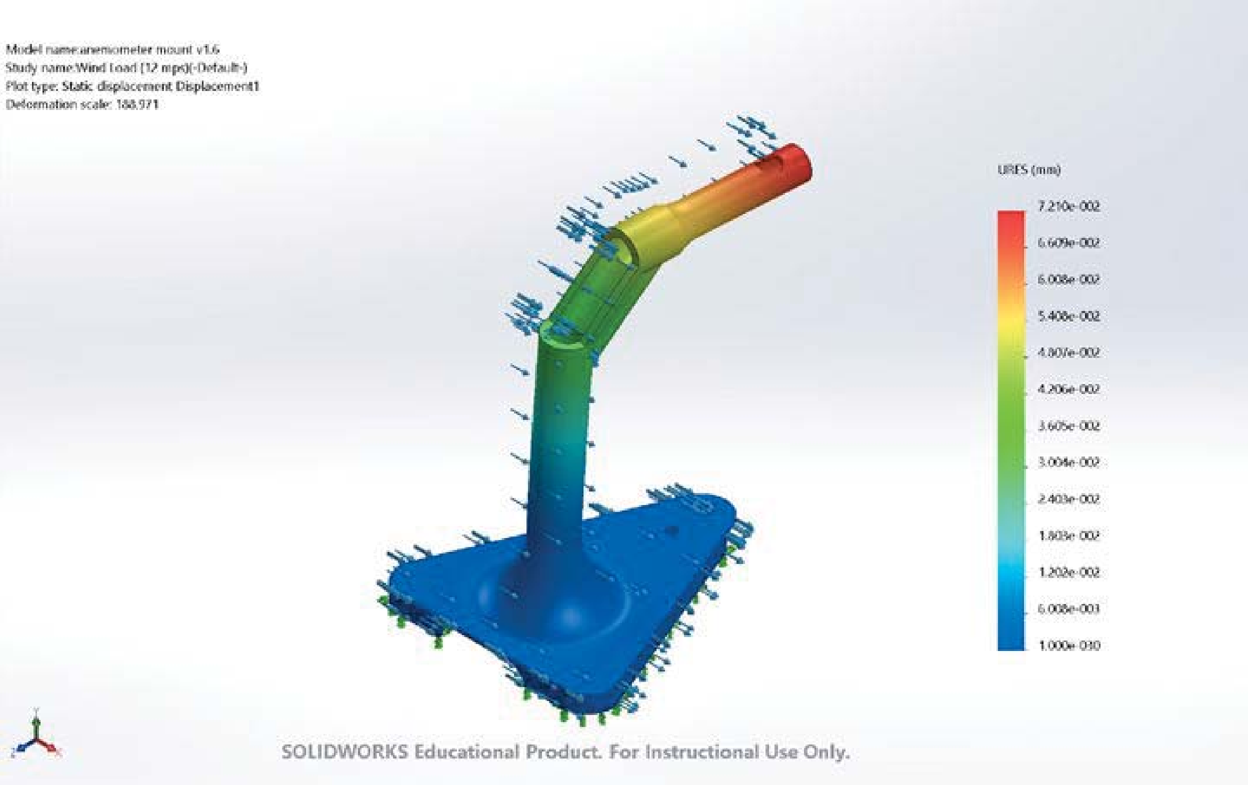

Model name:anemometer mount v1.6
Study name:Wind Load (12 mps)(-Default-)
Plot type: Static displacement Displacement1
Deformation scale: 188.971
URES (mm)
7.210e-002
6.609e-002
6.008e-002
5.408e-002
4.807e-002
4.206e-002
3.605e-002
3.004e-002
2.403e-002
1.803e-002
1.202e-002
6.008e-003
1.000e-030
SOLIDWORKS Educational Product. For Instructional Use Only.

Figure A.13: FEA showing displacement due to static wind load case.

6.2 LAYOUT DRAWINGS

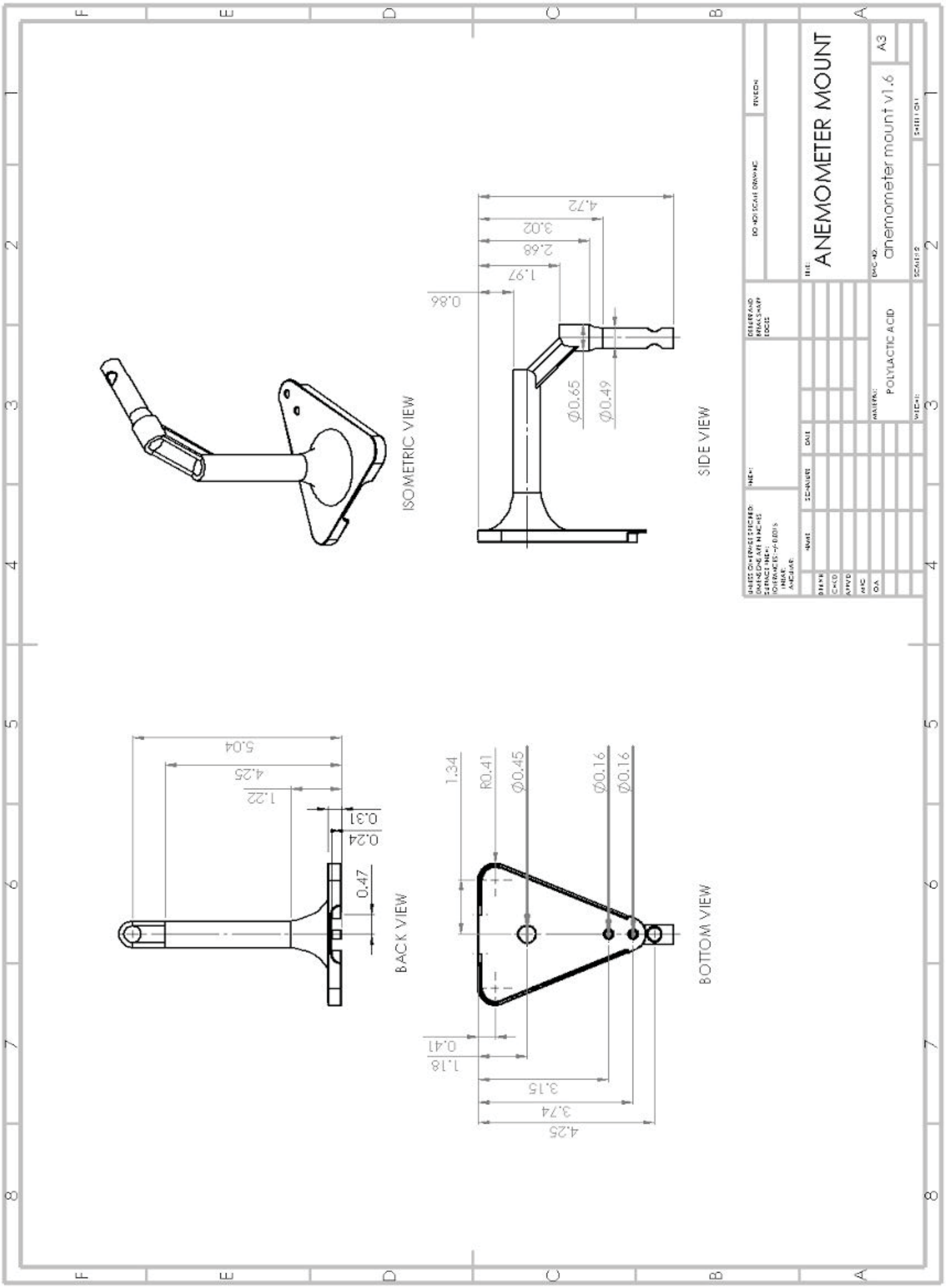


Figure A.14: Anemometer mount.

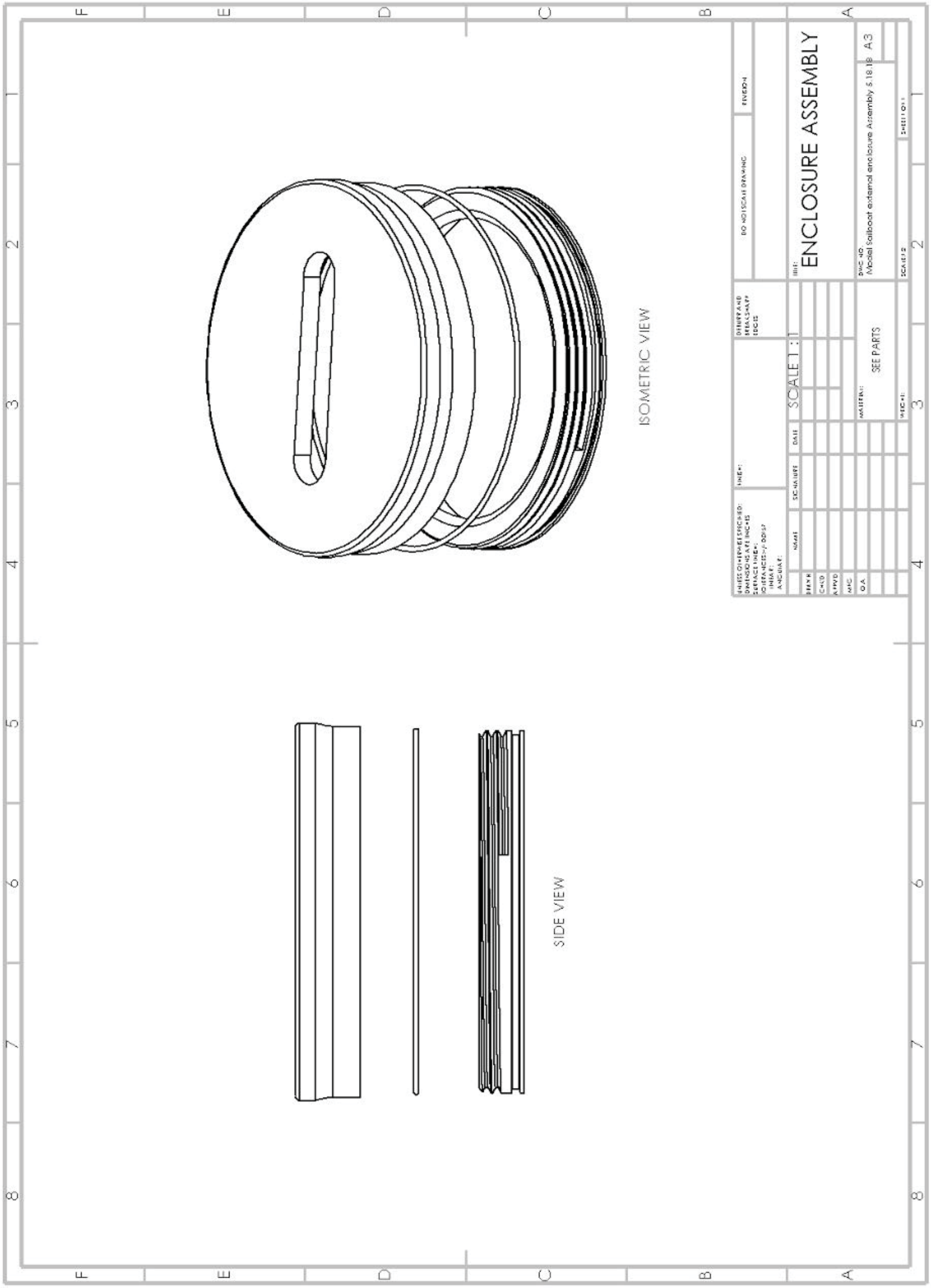


Figure A.15: Enclosure assembly.

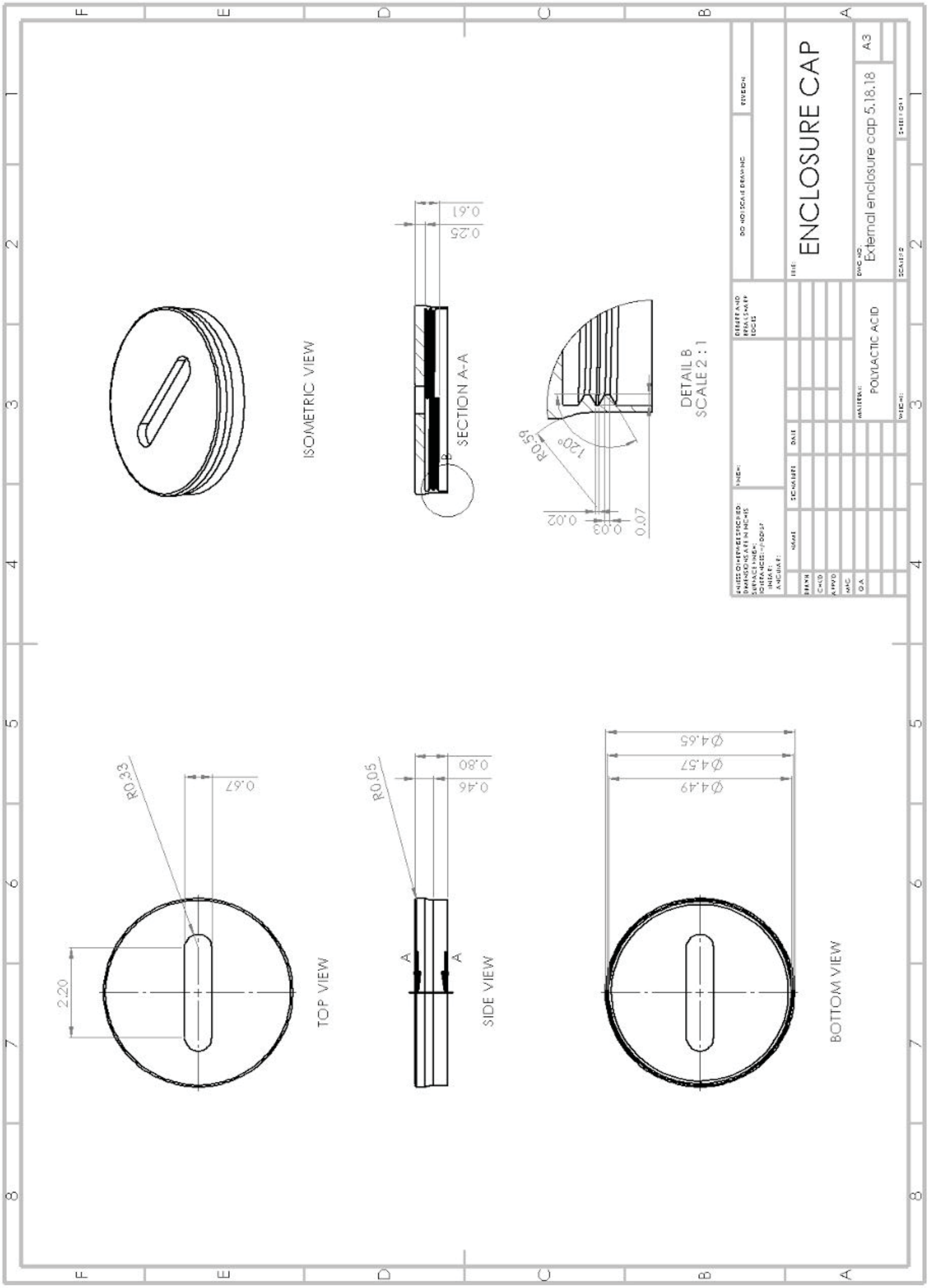


Figure A.16: Enclosure cap

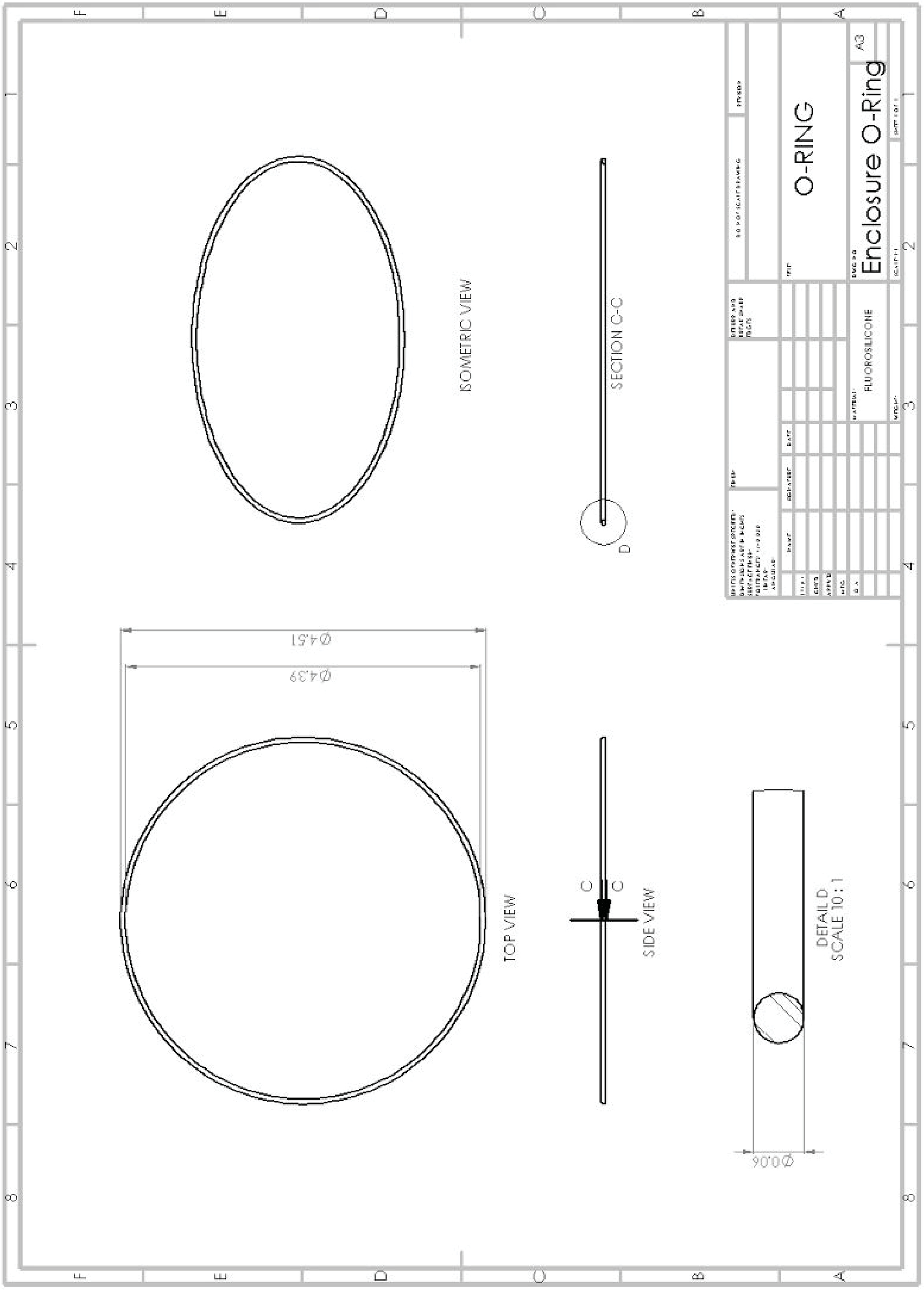


Figure A.17: O-ring.

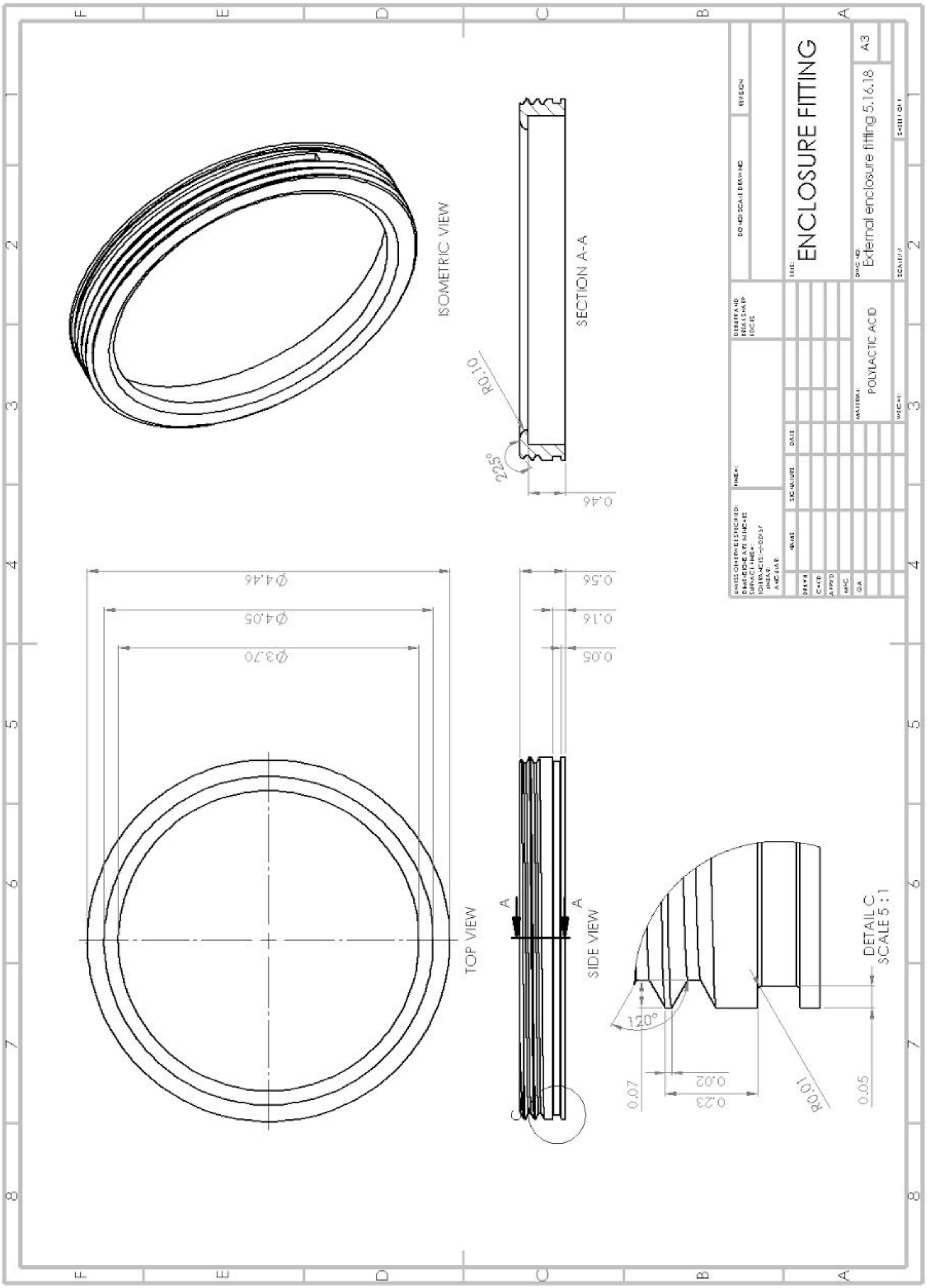


Figure A.18: Enclosure fitting.

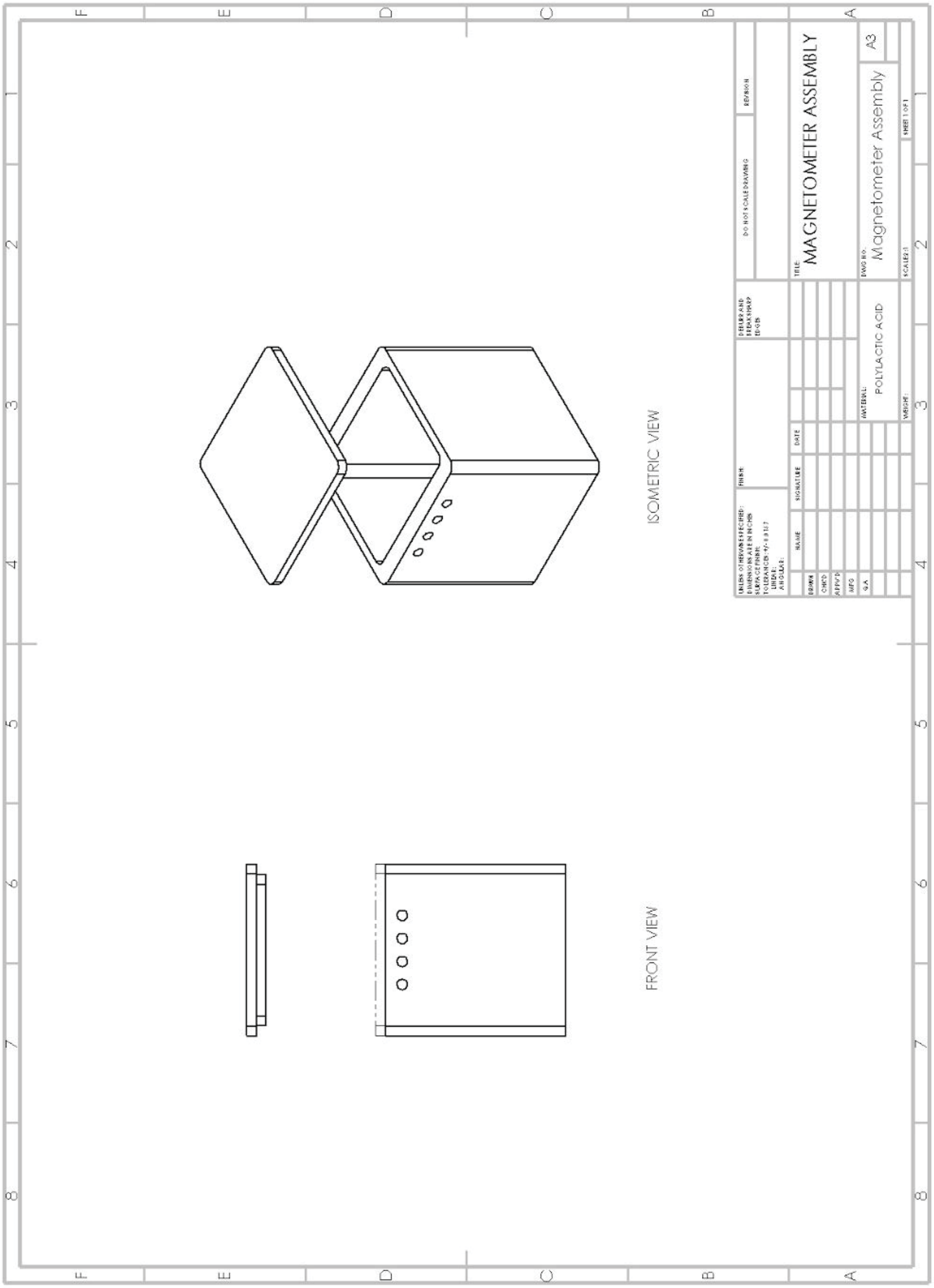


Figure A.19: Magnetometer enclosure assembly.

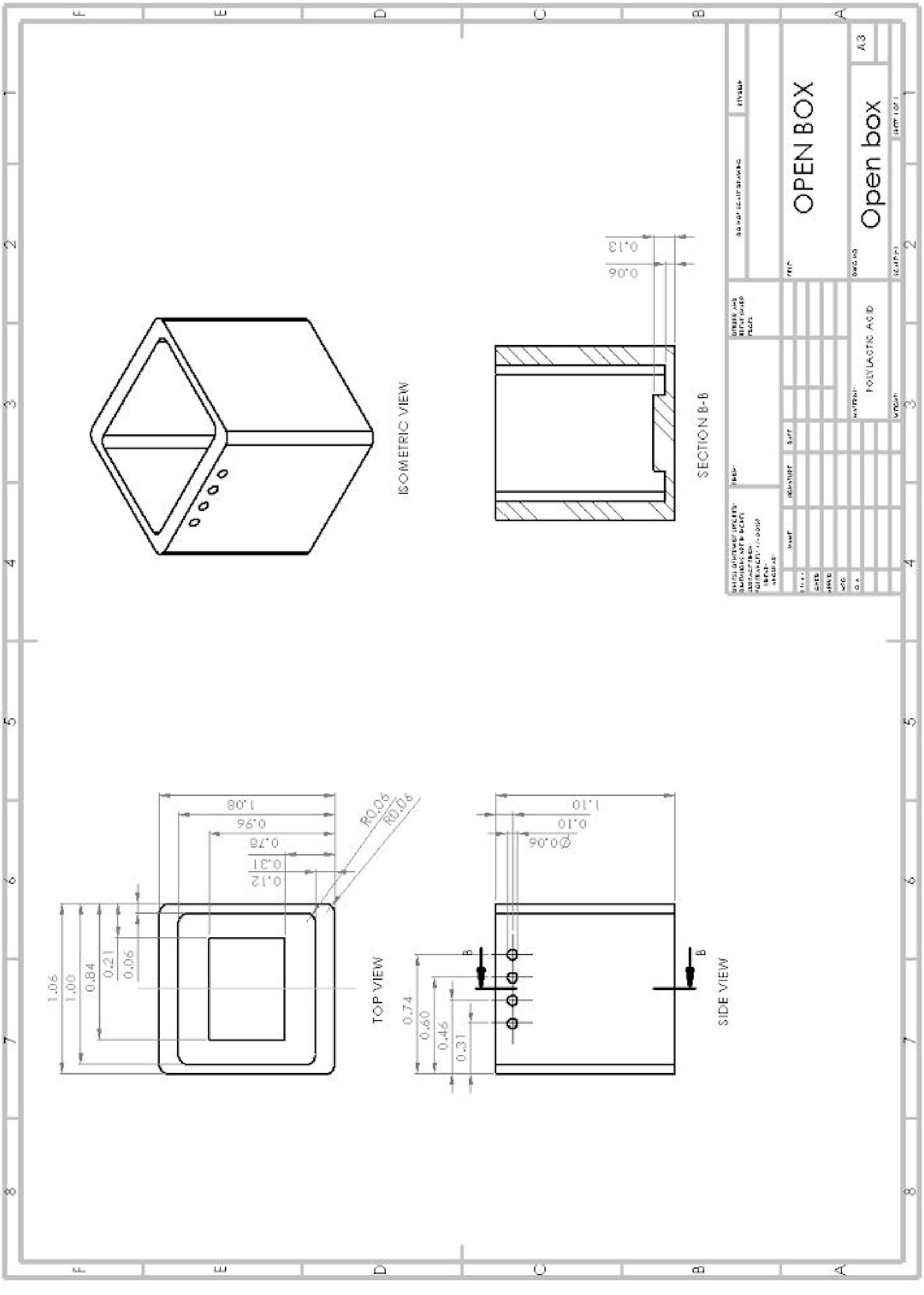


Figure A.20: Magnetometer assembly open box.

6.3 BILL OF MATERIALS

Component	Item Name	Quantity	Price
SAILBOAT	RC Laser Sailboat (Provided)	1	--
POWER SUPPLY	Zippy Compact 4000mAh 2S1P 25C L iPo	1	$16.92
	Hobbywing 5V/6V 3A Switch-mode UBEC, Max 5A Lowest RF	1	$8.09
	Keenstone 10A 100W AC/DC 1S-6S Digital Battery Pack Balance Charger/Discharger for LiPo	1	$51.99
	Turnigy E3 Compact 2S/3S LiPo Charger 100-240v (US Plug)	1	$25.99
	COLCASE Fireproof Explosionproof LiPo Safe Bag for Lipo Battery Storage	1	$12.99
	Power Splitter Female to 1 Male and 1 Female Barrel Jacks Standard DC Plugs	1	$7.52
SENSORS	Davis Instruments Anemometer	1	$127.50
	RJ11 Connector Breakout Board Module to Screw Terminals	1	$8.79
	Adafruit Ultimate GPS Breakout Board	1	$37.99
	GPS External Antenna 5m SMA	1	$15.36
	Mini PCI UFL to SMA Female Adapter	1	$1.66
	LSM303DLHC Magnetometer and Accelerometer	1	$14.95
	Arduino Protoshield	1	$15.61
PROCESSORS	Raspberry Pi 3.0 Model B Kit	1	$54.99
	Arduino Mega 2560	1	$14.99

	6” USB 2.0 to USB 3.0 Right Angle Cable	1	$5.99
COMMUNICATION	HiTec Optic 5 4.2GHz Transmitter	1	$45.99
	Minima 6E Receiver	1	$44.99
	Pololu 4-Channel RC Servo Multiplexer	1	$9.95
	3’ Ethernet Cable	1	$4.49
HARDWARE	Anemometer Mount (3D Print, Part)	1	$10.00
	Magnetometer Mount (3D Print, Assembly)	1	$10.00
	Enclosure Assembly (3D Print, Assembly)	1	$10.00
	046 Fluorosilicone O-Ring, 70A Durometer, Round, Blue, 4-¼” ID, 4-⅜” OD, 1/16” Width	1	$5.19
	Seaview Waterproof ABS Plastic Cable Gland for Multiple Cables 2 to 17 mm Diameter Wires, Grey	1	$36.15
MISC.	Wires (MM, MF, FF)		
	HXT 4mm Connectors	1	$2.24
	Permatex 80050 Clear RTV Silicone Adhesive Sealant, 3 oz	1	$5.49
	Duct Tape	1 Roll	$5.50
	Electrical Tape	1 Roll	$3.98
TOTAL EXPENSES	--	--	$615.30
ALLOTTED BUDGET	--	--	$1,500.00

6.4 CALCULATIONS

Table A.A: Initial Power Supply Calculations

Component	Supply Voltage (V)	Current Draw (mA)
Arduino Mega	7-12	25
Raspberry Pi 3 Model B	5	2,500
Adafruit GPS Breakout	5	20
Adafruit IMU	3.3-5	50
XBee Pro and Adapter*	3-5	250
Davis Anemometer	5	20
HS322HD Rudder Steering Servo	4.8-6	180
HS785HB Sail Winch Servo	3-5	285
	Total	2,865

*****This was replaced by a multiplexer and the original RC transmitter and receiver.

7. REFERENCES

[1] D. Santos, et. al., “Design and Implementation of a Control System for a Sailboat Robot,” *Robotics*, vol. 5, no. 5, Feb. 2016.

[2] M. Cronin, “Getting in and out of irons,” *US Sailing*. [Online]. Available: <https://www.ussailing.org/news/getting-in-and-out-of-irons/>.

[3] The University of New South Wales, Sydney, Australia, “The physics of sailing,” *unsw.edu.au*, [Online]. Available: <http://newt.phys.unsw.edu.au/~jw/sailing.html>**.**

[4] S. Blaevoet, J. Leung, M. Shi, V. Tran, and B. Zhao, “ucdsailboat/autonomous-sailboat,” [Online]. Available: <https://github.com/ucdsailboat/autonomous-sailboat>

[5] C. Alt and N. Wittinghofer, "Autonomous Sailing Boats," University of Salzburg, Department of Computer Sciences, May 30, 2011.

[6] Apple Rubber Products, “Seal Design Guide,” *Apple Rubber Products*, [Online]. Available: <http://www.applerubber.com/src/pdf/seal-design-guide.pdf>

[7] DSM&T CO., INC., “IP Rating,” *DSM&T CO., INC.*, [Online]. Available: <http://www.dsmt.com/pdf/resources/iprating.pdf>

[8] Lady Ada, "Adafruit Ultimate GPS", Adafruit Learning System, Feb. 2018.

[9] L. Xiao and J. Jouffroy, “ Modeling and Nonlinear Heading Control of Sailing Yachts,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2., pp. 256-268, 2014.

[10] J. Jouffroy, “Sailing Yacht Model (Version 1.0),” *MathWorks*, Oct. 2, 2014. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/47981-sailingyachtmodel-zip>.