

Self-Sufficient Buoy Energy Exploration

Ian Smith, Abdullah Fattahi, Alex Robertson, Marina Bellido, and Omar Lolita Jr (AUV): Team 28, Section 2, April 29th, 2024

Abstract— This paper presents the design of and experiments conducted using an autonomous underwater vehicle for the final project of Harvey Mudd College’s Experimental Engineering course. The robot was designed in order to determine the ability to sustain a self-sufficient buoy using alternative forms of energy. To learn which options available in the ocean would sustain the robot, three sensors were used to predict potential power outputs from these energy sources; an anemometer, a flow sensor, and a pair of photodiodes. The goal of the robot was to autonomously traverse across a few set points in different locations away from the shore and report on the measurements from each sensor. When deploying the robot, the effectiveness of the GPS navigation system was too inaccurate to travel in the expected path correctly. However, the data collection process still illustrated what energy sources could possibly supply a buoy of certain parameters. In conclusion, it was found that wind is too inconsistent to provide sufficient power to the robot but using hydropower might be a viable option at greater speeds. Alternatively, the photodiode data displayed that a solar panel with enough surface area could provide enough power even if covered with water.

I. INTRODUCTION

The goal of this project was to explore the potential of using alternative forms of energy harvesting to sustain an AUV. To do so three sensors were used to predict potential power outputs from three possible energy sources. The sensors being used are; an anemometer to measure wind speeds, a flow sensor to measure the water speed, and a pair of photodiodes placed below and above water to compare the difference in light intensity. The motivation behind each of the sensors was rooted in the research of current solutions to alternative energy that are being used to power ocean vehicles [1]. The robot aimed to autonomously traverse across a few set points in different locations away from the shore and report on the measurements from each sensor. The following paper will further outline the motivation behind each of the sensors and how they contribute to the overall effectiveness of the AUV. Included is a discussion of the data collection process for each sensor and a description of how each set of data was processed to determine results from deployment.

II. EXPERIMENTAL SETUP

A. Sensor Selection

Three distinct sensors to measure relevant quantities in their experiments: an anemometer, a flow sensor, and a pair of photodiodes were used. The following section will highlight choices made in selecting each of these sensors and how they contribute to the overall goal of the project.

Photodiode

A pair of IR-blocking silicon photodiodes (VTP9812FH)[2] were used since the light intensity above and below water needs to be compared to understand if an underwater solar panel, or a solar panel that gets covered in water would still be viable as an energy source on a buoy. IR-blocking photodiodes were chosen because the spectral range peaked around a similar wavelength as the irradiance of sunlight as shown in Figures 1 and 2.

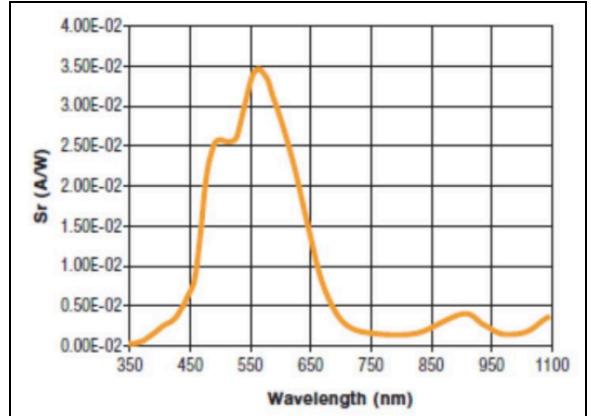


Figure 1: Wavelength spectral sensitivity at 25°C

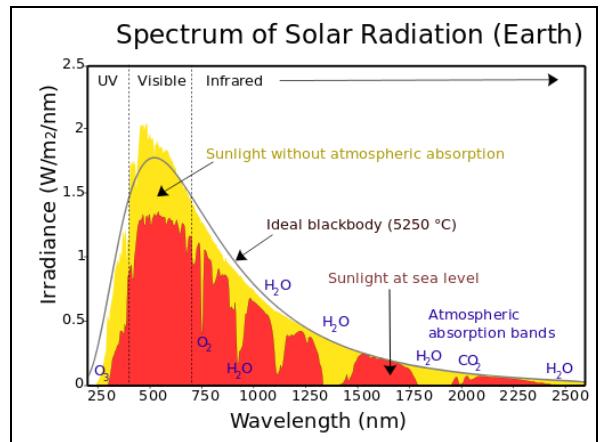


Figure 2: Peak Wavelengths of Sunlight

Flow Sensor

The YF-S201 flow sensor was selected as a way to measure flow for a potential source of hydropower. The YF-S201 flow sensor outputs a frequency. The YF-S201 flow

sensor was used in line with an LM2917 frequency-to-voltage converter. Creating a direct way to measure the voltage produced by the flow of water. These measurements would be compared to the power required by the AUV to see if hydropower is a suitable power source. However, the YF-S201 flow sensor was too resistant to changes in flow reproducible by the motors. The YF-S201 flow sensor was replaced by one made in the lab with a repurposed computer fan, a salvaged Hall effect sensor, and a ruler. The lab-made sensor was sensitive to flows producible by the motors attached to the AUV.

Anemometer

The anemometer is very similar to the flow sensor because it also uses a Hall effect sensor to determine wind speed. Essentially, the mechanical setup of the sensor consists of a magnet attached to the axle that the wings are fixtured to; see Figure 7 for a complete assembly of the anemometer. The hall effect sensor is held stationary and produces a low voltage when the magnet is detected (once per rotation). The sensor outputs a pulse train, digital output that relates to some frequency of the wings' rotation. This frequency can be used to determine the duty cycle of one rotation which is then correlated to some wind speed using a calibration curve that was experimentally found.

B. Circuits and Mechanical Design

Photodiode

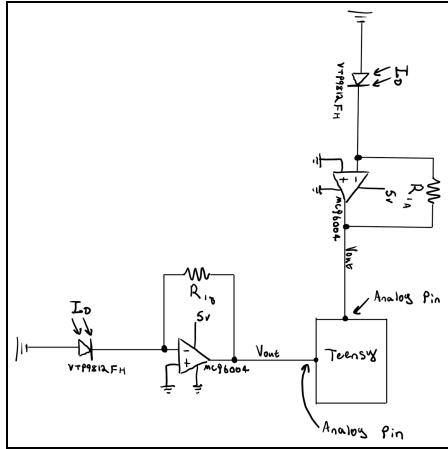


Figure 4: Photodiodes Circuit Schematic

This circuit shows both photodiodes connected to the teensy through trans-impedance amplifiers. These trans-impedance amplifiers transform the output current of the photodiodes into a voltage that is amplified to comply with the teensy range. Since the photodiodes produce a current, the only power needed is to supply the op-amp, which takes a single rail 5V input. This input comes from the battery through the voltage regulator on the motherboard. Using Equation 1,

$$V_{out} = I_D * R1 \quad (1)$$

where I_D is the current derived from the photodiode, the best resistor choice can be calculated. Using 3.3V as the maximum Voltage output, and extrapolating the data from Figure 9, an average $I_D = 100\mu A$ is expected which produces a resistor value of $33 k\Omega$. However, testing revealed that with this resistor in direct sunlight, the output voltage was 1.119V, which correlates to an $I_D = 33.91\mu A$. Based on this, the ideal resistor value can be calculated to produce 3.3V, the maximum teensy input voltage, in direct sunlight. The calculated resistance is $97,316\Omega$ so, $R1 = 97k\Omega$, and in testing, this resistor choice produced a maximum of 3.29V in direct sunlight. Since direct sunlight is expected at Dana Point this resistor value was used. The same resistor was selected for both the above and below-water photodiodes since preliminary testing displayed only a slight decrease in sensor output underwater, as the output voltage in direct sunlight dropped by only 0.3V. The minimum output voltage is 0V as the sensors are not capable of producing a negative current, so there are no concerns about the lower voltage limit. As for mechanical considerations, it is important that one sensor is always above the water and the other one below the surface, both always pointing directly upwards. This goal is accomplished by attaching one photodiode to the inner raised portion of the robot such that it cannot be submerged, and the other to an extended corner piece such that it is constantly submerged.

Flow Sensor

The YF-S201 flow sensor and LM2917 frequency to voltage converter system were used to determine the voltage produced by water flowing through the sensor. The YF-S201 flow sensor could handle a range of 1-30 Liters/min [3]. The LM2917 converter had no max frequency and required 5V to run [4]. Adjusting the LM2917 converters circuit to try and account for the YF-S201's resistance to flow resulted in the circuit pictured in Figure 5. Because the speed of the robot was not enough to move the flow sensor, a computer fan (E-waste from the stockroom) was used to create a new flow sensor for the robot.

Guide to assembling the computer fan flow sensor:

- The fan blade and motor were removed from the housing by removing the small plastic ring that holds the axle and fan.
- To remove the generator, the magnet from the outside of the motor and next to the outer metal motor are removed. So are the copper wires.
- The fan, small spring, and the two bearings are placed back into their original place.

- From the flow sensor bought, the small ring magnet is removed and then glued on top of the fan.
- Then an attach support for the Hall effect sensor is built. A piece of a ruler was cut to the length between the two holes and about a half-inch wide. Then, two holes were drilled so the bolts could slide into them. This creates the support beam for the hall effect sensor.
- For support, bolts that stick at least a half to $\frac{1}{4}$ inch out past the front of the housing through the holes in the corners of the fan housing were used.
- The Hall effect sensor is added from the previous flow sensor: the PCB and the 3 wires connected to it are removed. The entire assembly is waterproofed with hot glue (the electrical components). Once sealed, the Hall effect was attached to the support beam with more glue as shown in Figure 6.

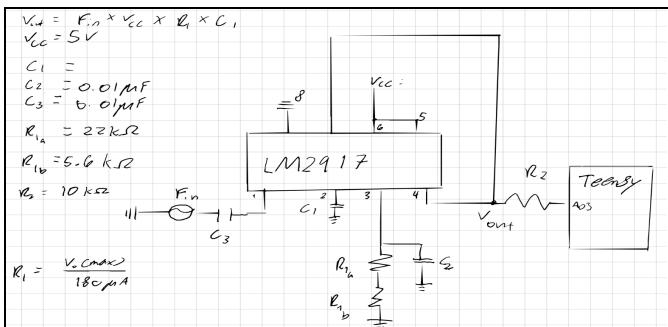


Figure 5: Adjusted YF-S201 Flow Sensor and LM2917 Converter Circuit Diagram.



Figure 6: New flow sensor constructed from a computer fan

Anemometer

Since the anemometer uses a sensor that outputs a digital signal, the signal did not need to be attenuated since the “high” reading out would just be clipped at 3.3 V. For the purposes of this sensor, that was reasonable enough as it then showed a pulse train regardless of the actual voltage. The peaks were then used to determine the time in which the magnet passes the sensor. When determining if a different program file was needed to ensure that there was no aliasing, the wind tunnel was used to spin the anemometer at speeds much higher than what was expected at Dana Point (~20 m/s). At this speed, the

frequency of the wings was still lower than 10 Hz, so no aliasing was observed. For this reason, the original data processing code sourced from *surface.ino* was used.

The circuitry for this sensor was relatively straightforward since there was no op-amp required for the hall effect sensor being used. Using the datasheet, it was determined that a $10\text{k}\Omega$ resistor was needed as a pull-up resistor and a 0.1\mu F capacitor was used to filter out any noise from the digital output as shown in Figure 8. Mechanically, the anemometer was largely built in reference to the anemometer process router linked to the E80 website [5]. A hall effect sensor was fixtured in place while a neodymium magnet was attached to a rotating axle that is used to count a single rotation of the anemometer. The mechanical assembly and fixturing of the anemometer proved to be a challenging step in the setup of this sensor since the sensitivity of the Hall Effect sensor was dependent on the strength of a magnetic field (related to the distance from the device).

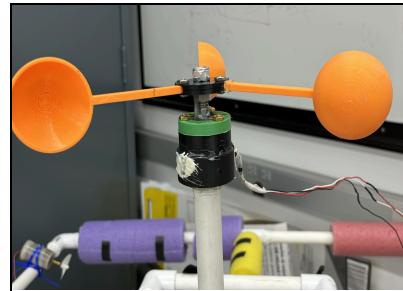


Figure 7: Fully assembled 3D printed anemometer

For that reason, many alterations were made from the reference design on the E80 website to ensure that the Hall Effect Sensor was able to detect the presence of a magnet while the wings rotated. Alterations were made to the piece that secured the magnet so that it was fixtured closer to the sensor. While this solution initially fixed the issue, in the launch it stopped reading the magnetic field. The anemometer was disassembled and an additional magnet was added, so the magnetic field would increase, and the inside layout was modified to better reach the sensor without hitting against the perimeters of the anemometer.

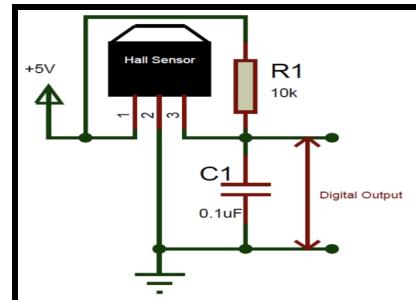


Figure 8: Hall Effect Sensor Electrical Schematic [5]

C. Sensor Modeling and Verification

Photodiode

Unobstructed sunlight is typically around 10,000 fc [6,7], so this value was used in preliminary testing to determine resistor values. Based on Figure 9, it can be concluded through extrapolation that the expected current output is around 100uA. Collected data displays that the average current above the water in direct sunlight is only 45uA, so the extrapolation of Figure 9 is likely, not accurate, and the linear trend likely does not continue. Because of this a new calibration curve was created with measured light intensities.

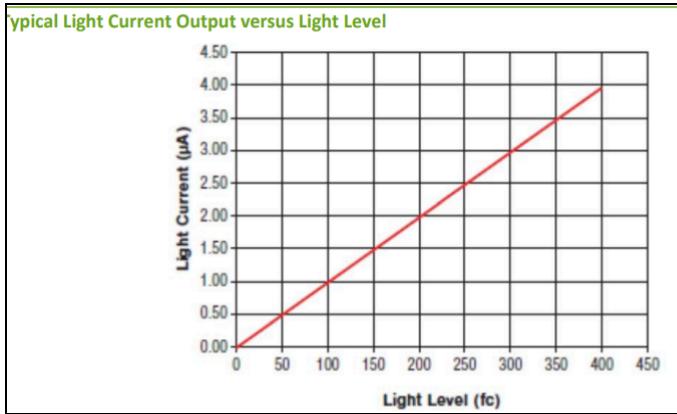
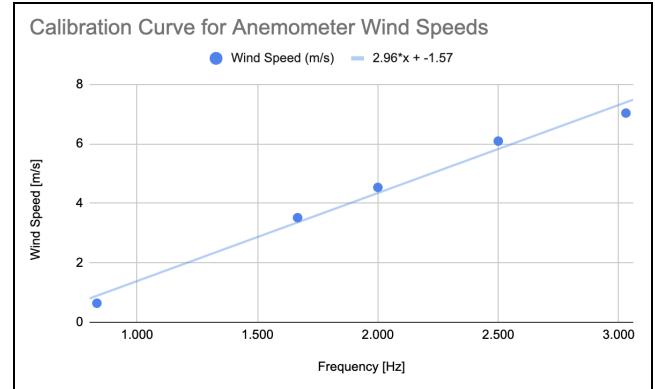


Figure 9: Light level vs. Light current from Datasheet

Anemometer

While direct modeling was not done for the anemometer, a calibration curve was experimentally measured in the HMC Wind Tunnel. This was used to determine what corresponding frequencies related to some wind speeds that were determined in the wind tunnel. Using the wind tunnel, Equation (2) was calculated using the data received from the wind tunnel and a curve that was fit using Excel. While the calibration curve seemed reasonably sound for many wind speeds at 5 m/s or greater, the calibration curve was experimentally tested at speeds much greater than the speeds that were actually collected at the deployment of Dana Point. For that reason, very slow wind speeds throughout the day were calibrated to be negative using Equation (2)

$$\text{Wind Speed} = 2.96(x) - 1.57 \quad (2)$$



Picture 10: Calibration Curve Relating Wind Speed to Frequency

Flow Sensor

No direct modeling was done for this sensor however a calibration curve was obtained for the lab-made flow sensor. This curve was used to determine the frequencies associated with different flow speeds and Equation (3) which relates the two. This calibration curve was tested for speeds of up to 30 cm/s as this was predicted to be higher than expected by the AUV.

$$\text{Flow Speed} = 0.0807x + 0.0326 \quad (3)$$

D. General Robot Mechanical Design

The intent was to create a surface self-sufficient buoy able to harvest energy while also having the ability to add more sensors in future iterations. For this reason, it was given a raft shape with high surface volume and symmetrical for more sturdiness. Its dimensions are big to prevent it from being tilted by waves and it is constructed with floaties so that it is buoyant. Though the robot had all motors on one side, due to the AUV's symmetry and floaty placement on the other side, it was evenly buoyant and the motors' placements did not affect the stability of the robot. The anemometer is placed in the middle to prevent it from getting hit by water. One photodiode is placed above the water surface and the other one is submerged below water on one of the sides. The flow sensor is placed in front of the AUV to prevent motor interference.

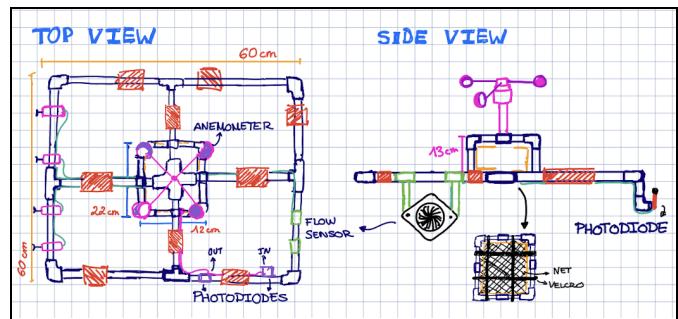


Figure 11: AUV Mechanical Design



Figure 12: Full AUV Assembly

III. DEPLOYMENT

A. Safety

Safety measures were taken to secure the safety of all members of the group, the ecosystem the robot interacted with, and the people near the experiment. Safety measures enacted are the use of gloves, and glasses, and the following of all security measures when building the frame and all sensors. The electrical box was also heavily waterproofed to prevent any type of battery malfunctioning or chemical reaction due to water exposure. All pieces were secured to the frame to prevent contaminating the ocean and the experiment was conducted at least 10 meters away from anyone at the beach.

B. Deployment

Before beginning the deployment, the magnetometer calibration to calibrate the heading is performed by programming the teensy with the magnetic calibration software: Mag_Calibration.ino. It is run while the motherboard is rotated 360 degrees for around 40 seconds, and Mag_Calibration.m is then used to extract the hard and soft iron correction coefficients, which are imputed into mag_offsets and mag_comp variables in SensorIMU.

The experimental protocol starts by attaching all motors to their specific h-bridges, connecting and checking all sensor cables, and tightening all of the bolts in the box to prevent leaking. Then the battery is connected, the motherboard is turned on, and the cable to the teensy from the computer to run *E80_Lab_07_surface.ino* is connected. All sensors will be tested for correct functionality during this first run by looking at the SD card data using Matlab. Once all sensors are checked and working, the GPS location of deployment is found using Google Maps and inputted into XYStateEstimator.h. Then *E80_Lab_07_surface.ino* is run again. After the Adafruit Ultimate GPS connects to 5 satellites, the serial monitor will

output the GPS position it believes to be located. This location will be imputed back into the XYStateEstimator.h for more precision. The E80_Lab_07_surface.ino is run again, and once it connects to 5 satellites, the motors should turn on and begin the Kp control system to find the waypoint dictated by E80_Lab_07_surface.ino. The robot is then disconnected from the cable, the box is closed and the AUV is placed in the water.

Once all waypoints are reached and the delay in each location is performed, it will have completed its path and can be taken out of the water for data collection. Its recovery is dependent on the end location you have set in the code. If you place the last waypoint as (0,0), then it should return to the GPS location inputted in the code. Once recovered, the SD card is read using logreader.m in Matlab. Each pin corresponds to a specific sensor and the latitude, longitude, and heading are all recorded in specific variables.

IV. DATA PROCESSING

Photodiode

After retrieving the data from the proper teensy pins, the output was converted from teensy units to voltage. Then using Equation (1) the voltage is converted to current since that is the output of the sensor. These currents can be utilized to determine the relationship between the above and below-water photodiodes. The data was also cleaned such that only data from when the robot was in the water was used since many of the runs began on the beach where the below-water photodiode was not submerged.

Flow Sensor

The lab-made flow sensor is meant to output a square wave that jumps from high to low. The jumping indicates the presence of the magnet being read by the internal hall effect sensor. The data was read into with both axes in teensy units. To convert the x-axis into seconds the time vector in Teensy Unit was dot divided (./) in MATLAB. The low state of the flow sensor was 0 Teensy Units and the high state was 1023 Teensy Units. The data had 6 outliers that needed to be removed from the vectors. Due to the flow sensor's output only being high or low, these outliers are attributed to noise. The outliers were manually removed from the data by identifying their position within the vectors, both time and Teensy units, and removing that entry. To find the periods present in the graph the command *findpeaks* was used to make a vector of all of the leading edges producing figure 14. The cleaned data combined with the

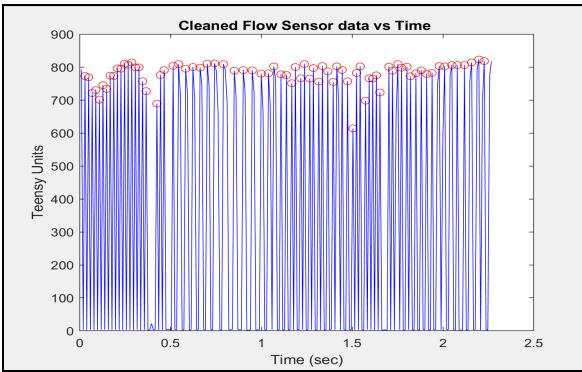


Figure 14: Cleaned Flow Sensor Data with the Leading Edges

identified leading edges. To find the period, a for loop was implemented that looped through the vector of rising edges and calculated the difference. Once the vector of periods was obtained, the *movmean* command was used to obtain a rolling average. The vector obtained by the rolling average was inverted using a dot divide to get a vector of the average frequencies. When plotted the vector of average frequencies results in the chart shown in figure 15. To obtain the power output that could be generated using this sensor, a water turbine was designed. The water turbine was created using the same fan model as the lab-made flow sensor and produces power as it spun. The initial fan has a generator inside that is broken when creating the fan sensor; however, this fan can be used to analyze the power output obtained at the speeds the sensor was spinning underwater. The inside of the fan was remodeled: all mount resistors and transistors were removed and the wires were directly connected to the induced current terminal wires [8]. The fan was spun in the wind tunnel to obtain the power output vs. velocity, and then similitude was used to be able to find the power generated in the water.

This way, the velocity measurements made in water with the fan sensor can be applied to the velocities obtained in the wind tunnel.

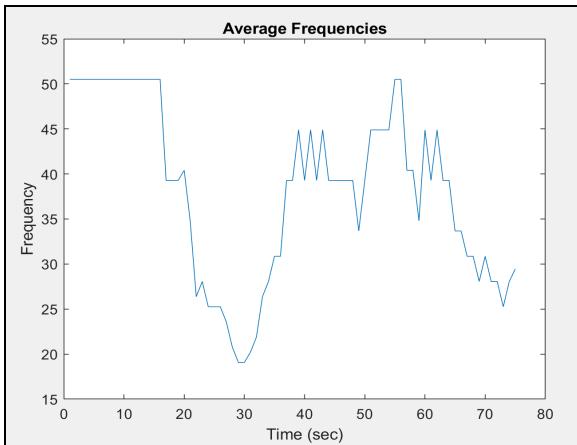


Figure 15: Frequencies obtained from Dana Point.

To have similitude, the Reynolds number (Re) in water and air are set equal (5). Re is the ratio of inertial force and viscous force of a fluid. It is a dimensionless parameter where V is upstream velocity [m/s], L is the characteristic length [m], and ν kinematic viscosity [m^2/s] (4). [9]

$$R_e = \frac{\rho V L}{\mu} = \frac{V L}{\nu} \text{ where } \nu = \frac{\rho}{\mu} \quad (4)$$

L is the same for both turbines in water and air, and through it, geometric similitude is also achieved. By equating the two Reynolds numbers kinematic similitude is obtained:

$$Re_a = Re_w \Rightarrow \frac{V_m * l_m}{\nu_{air}} = \frac{V_p * l_p}{\nu_{water}} \quad (5)$$

Since the models have geometric and kinematic similitude, there is dynamic similitude as quantified by the drag coefficient. Since the three types of similitude have been obtained, similitude can be enforced to compare the velocity in the air of the turbine to that of water. Setting the water speed values equal to the power calculated in air results in the calibration curve shown below in figure 16.

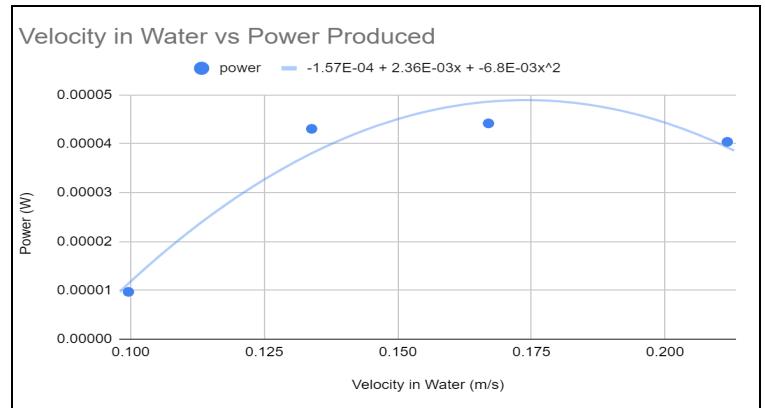


Figure 16: Water Speeds vs. Power Output from Turbine

Anemometer

As mentioned in previous sections, the output from the hall effect sensor was processed as a digital output of high and low voltages that illustrated a specific frequency through MATLAB. Using *logreader.m*, a plot of the frequency was plotted as a function of Teensy sampling rates which is equal to 0.1 seconds. With that data, a frequency for small intervals of time was calculated to determine the wind speed at those times. Using this data, further analysis was done to see whether the wind speeds at Dana Point would be enough to provide power to the rest of the robot.

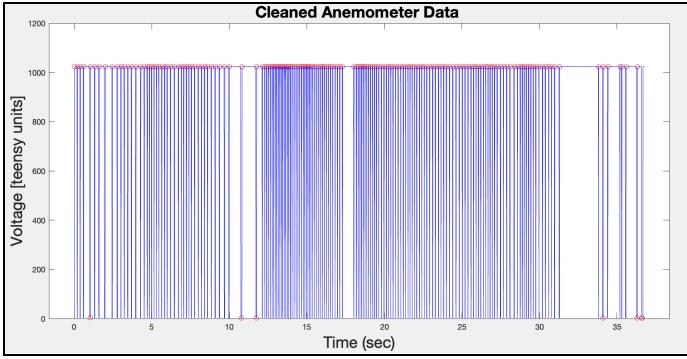


Figure 13: Cleaned Data from AUV Test Run

V. RESULTS AND ANALYSIS

Photodiode

The main goal was to determine if there was a significant difference in output currents between the above and below water sensors, and the data obtained agrees with the presumption that the above water photodiode produces a higher current, as displayed in figures 17.1-3. It is important to note that in Figure 17.3, the above water photodiode output is 3.3V, the maximum teensy input, which is why the line is flat, so it can be assumed that the true values are higher than displayed, but likely not by much. In figures 17.1 and 17.2, the erratic data at times 200-700 and 900 - 1300 respectively can be explained by the robot's motors being active during these times, which drew extra current, affecting the data, however, the general trend can still be examined.

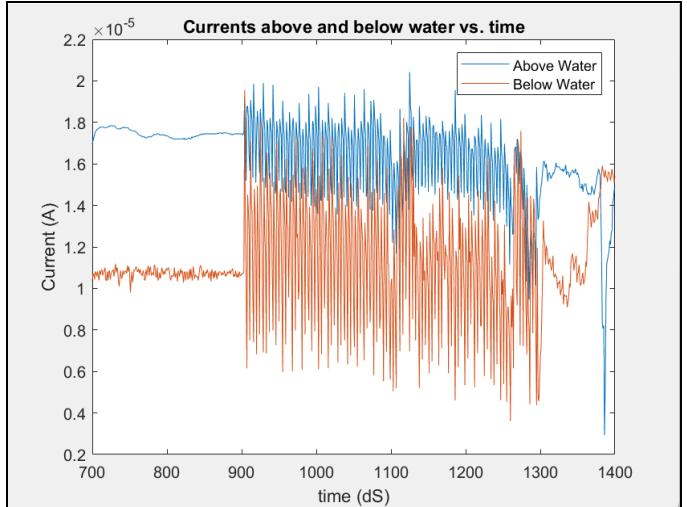


Figure 17.2: Photodiode current readings at 10:36 am

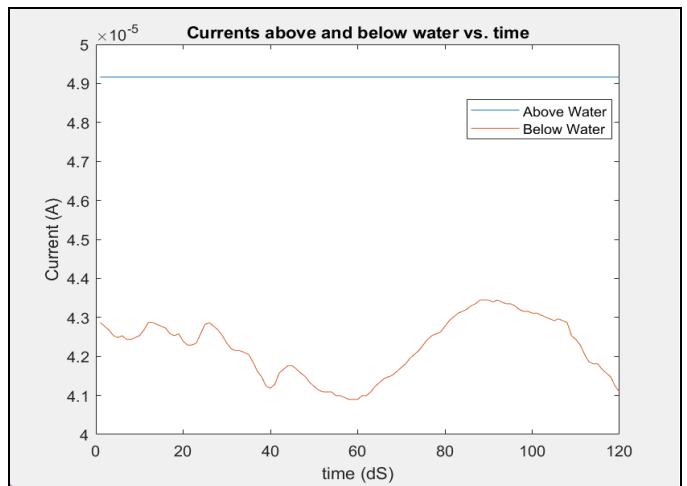


Figure 17.3: Photodiode current readings at 12:35 pm

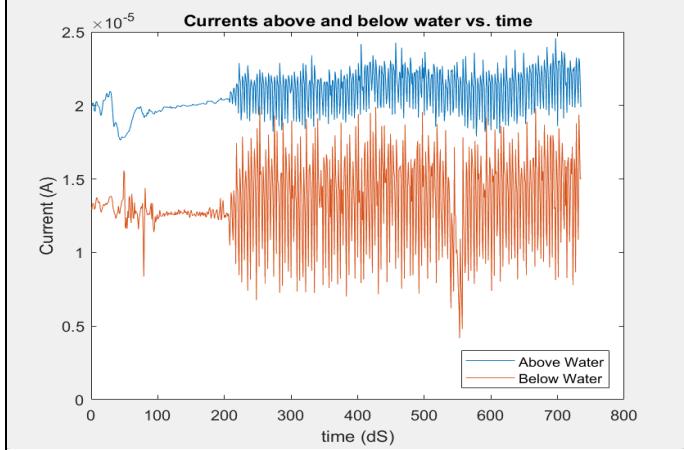


Figure 17.1: Photodiode current readings at 10:20 am

While this data demonstrates that there is a difference between the photodiode currents, the underwater photodiode still produces a relatively high current, as on average the currents produced were $2.9\mu\text{A}$ less than the currents produced by the above-water sensor. Based on the final run data, when the sun was brightest, there was a drop of 3403 fc below water displayed in figures 17.3 and 18. Given a proportional relationship between solar panel output and light intensity [10], this correlates to a 9.5% drop in output power below water. At this lower current solar panels would be less effective, but still produce power and be a viable option for a self-sufficient buoy system. Since the datasheet graph for light intensity vs. current datasheet cannot be extrapolated reasonably, a light measuring app was used during the runs to create our own curve. Using the camera on a phone, the light intensity in footcandles was measured during each of the runs.

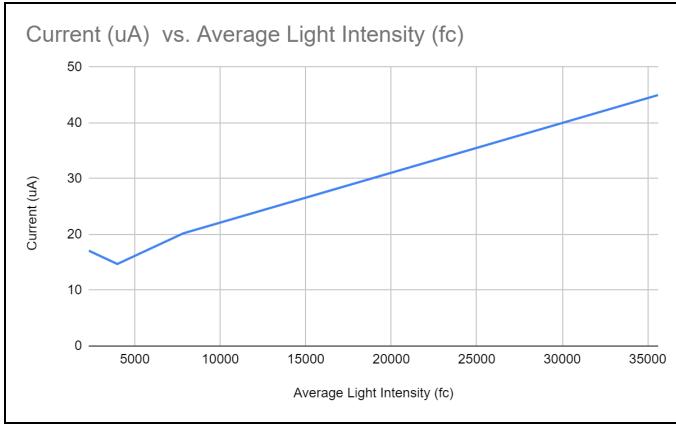


Figure 18: Measured light intensity (fc) vs. Current of above water

This data reveals that the average light intensity in fc of direct sunlight is 35,613fc on average, contrary to the previous 10,000fc assumption.

Flow Sensor

To find the power that would be produced by the data obtained, the calculated velocities were put into the equation obtained from the calibration curve. A graph of the resulting power values is shown below in Figure 19.

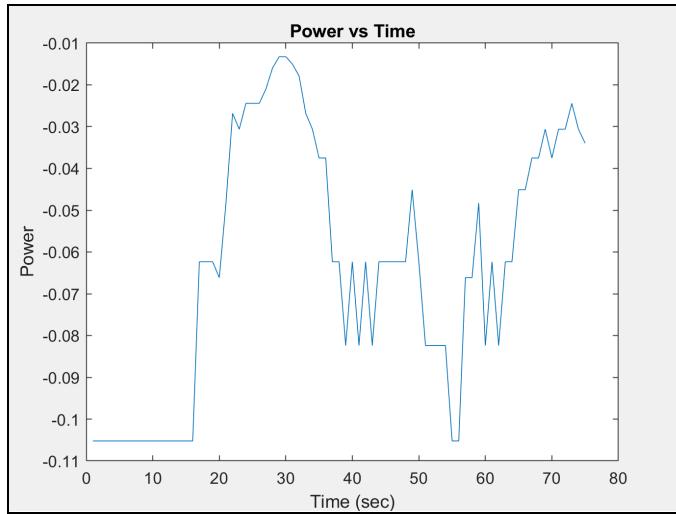


Figure 19: Plot of the Calculated Power Values

Since the velocities calculated from the data ranged from 0.1671 to 0.8779 cm/s and the values that calibrated the turbine ranged from 0.0996 to 0.2117 m/s the calculated power values should be considered negligible or 0. The calibration curve is set for values a factor of 100 higher than the values calculated so the calibration curve is not an accurate representation of the power that would be produced. Since the flow sensor was spinning at the lower speeds some amount of power would be generated. The amount of power would likely be close to 0 but not negative since negative power can't be produced.

Anemometer

The main goal in data collection at the Dana Point deployment for the anemometer was to determine if the wind speeds were high enough to sustain a self-sufficient buoy to some extent. After running a few deployments, one run provided relatively clear data that shows the wind speeds along a continuous path that the robot took. To come to a conclusion, the data from the hall effect sensor was first processed to create a plot of the average wind speeds for one run as shown in Figure X.

This plot shows that the peak wind speeds during the run were at around 2.5 m/s while the average speed was closer to 1.5 m/s. After further analysis of the robot's course, it was concluded that the non-zero wind speeds were at times when the robot was moving and therefore inducing some wind speed due to its own motion. While this data does not inherently show that there is or is not enough fast-moving wind to power the robot, further analysis can be taken to conclude how much power a theoretical vertical axis wind turbine (VAWT) could generate at the wind speeds collected during this run.

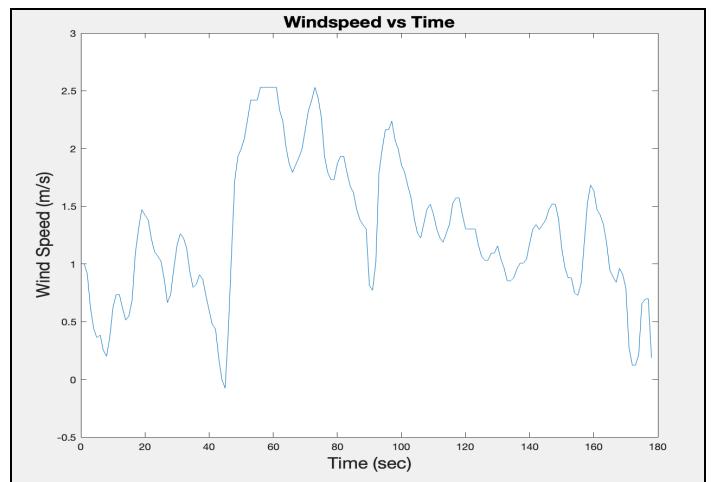


Figure 20: Plot of Wind Speeds over Time

Using an online calculator [12] for a VAWT that had wings with a cross-section of 2 ft² and a wind speed of 2.5 m/s it was determined that the maximum power at full efficiency that could be generated was 1.77 W. From the datasheet [12] of a motor similar to the one used for this project's AUV, it noted that the motor requires 1.5 W of power. From this, it can be concluded that even at maximum wind speeds collected at Dana Point (which only were sustained for ~10 seconds), wind energy would only be able to power one motor which would not prove effective for the goals of the project.

VI. SOURCES OF ERROR

Flow Sensor

Similitude was used to determine the output power of the fan in water. However, the fan is not static but rather spinning, so it will experience a lateral drag from the water pushing against the blades when it spins in the water. Likely, the lateral drag is not taken into account when calculating similitude, possibly becoming a source of error in the data. Due to time constraints, no further studies could be done. However, in future iterations, further simulation should be performed using COMSOL. Mechanical sources of error are that the fan with the generator has a larger initial static force that the wind must overcome before it starts spinning. This initial static force is due to the magnets inside of it that induce a magnetic field opposing the motion. This could be the reason why it is observed in water that the sensor starts spinning at lower velocities than in the wind tunnel.

Moreover, the flow sensor should produce a square wave when collecting data, however, Figure 14 shows a signal with single peaks. The shape of the data was originally believed to be a problem with aliasing and folding. To test this an oscilloscope was attached to the flow sensor output, and a shop vacuum blew into the flow sensor. The level of flow produced by the shop vacuum was incapable of producing aliasing and folding. This level of flow was much greater than what could be produced by the water at Dana Point. A plot of the current draw over the output from the flow sensor highlights the issues in Figure 21. The shape of the flow sensor output is due to the current loading from the motors.

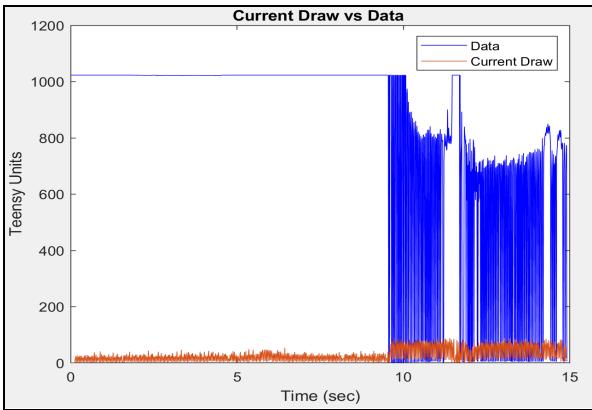


Figure 21 : Data & Current Draw Plotted as Function of Time

Surface Navigation

The AUV used K_p control for autonomous navigation. Given an origin GPS coordinate it would move to a chosen waypoint and become stationary to be able to collect data at that location for 20 seconds.

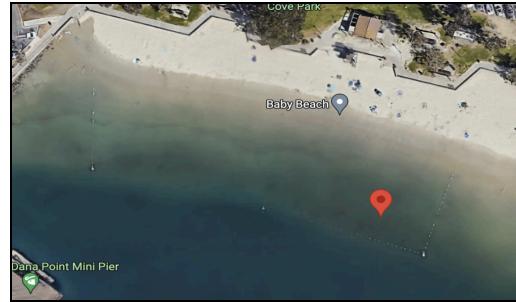


Figure 22: Aerial view of Dana Point and origin point:
Lat = 33.4624023; Lon = -117.7047195

the origin location's direction towards the Dana Point Mini Pier. Several groups experienced inaccuracy in their GPS readings, especially in the x-coordinate with a 16-meter error margin, so the waypoint selection was done in y, changing therefore only the latitude: { 0,-8, 0,-5 }. Nevertheless, after implementing those changes the AUV was still moving in the longitude plane towards the pier. Since the primary focus was the collection of sensor data, it was decided to run the code turning all motors off and manually moving it and stopping it at a specific point like the surface navigation would've done.

The hypothesis was that the GPS module was inaccurate and that was the reason for the inability of the robot to find its origin nor the waypoints. However, during post-processing, the GPS was plotted and it was extremely accurate with the location where the trials were run. As observed in Figure 23, the GPS lat and lon plotting shows the precise robot's path. This isn't coherent with the observation of inaccuracy at the beach. Since it was moving in the opposite direction to the waypoint it was thought there could be a problem with the heading, so further tests were done to find the cause of this issue. The heading was plotted and compared to specific points during the trial to observe where the robot thought it was heading.

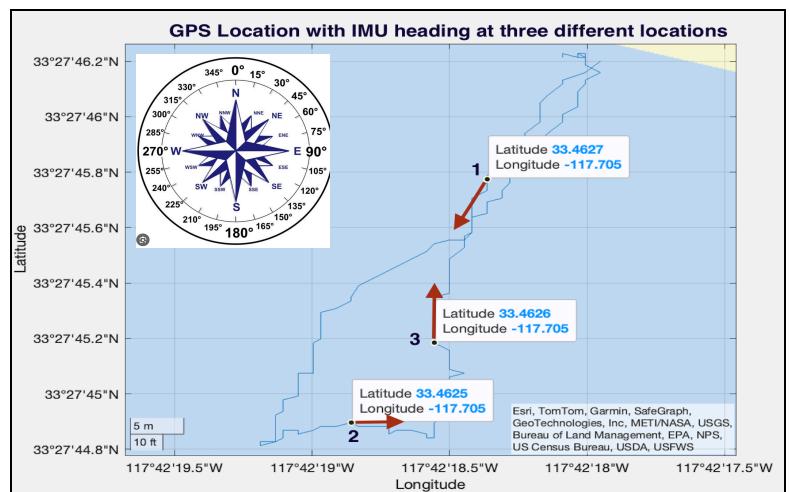


Figure 23: Path taken and arrows showing the AUV's heading

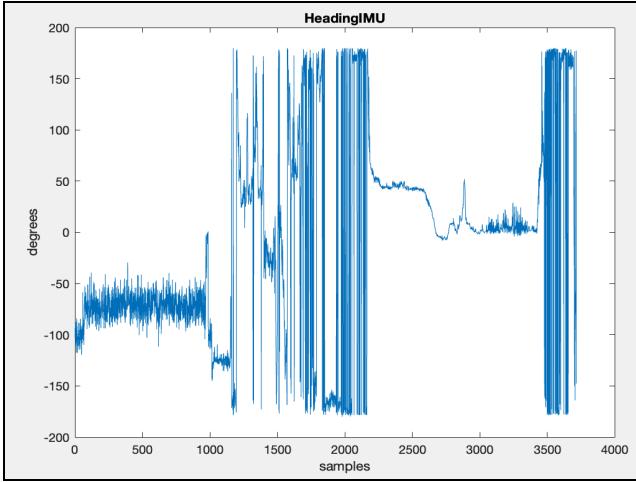


Figure 24: Plot of Heading IMU

Plotting the heading IMU at the 3 selected positions: Location 1 is between samples 1250-1500, Location 2 is between samples 2200-2900, and Location 3 is between samples 2900-3400. The robot is heading in the same direction it is moving, so the error does not seem to come from headingIMU. Therefore, further analysis should be conducted to observe other possible sources or errors, such as GPS location delay when on site. Satellite connection can also be studied, as using more satellites will most likely make the gps data more accurate.

VII. CONCLUSION

A. Overall Results

Based on the collected data, the photodiodes reveal that solar panels could act as a power source for a self-sufficient buoy, even if they get wet or are fully submerged in the water as there is only a 9.5% drop in power output in direct sunlight. The anemometer data reveals that there was not enough wind speed at Dana Point to create power for the buoy, however in areas with higher wind speeds it may work as a power source. Further analysis at other sites over the course of more days and time would help show if different wind conditions would be viable as an energy source. The flow sensor turbine is not designed to spin via the flow of water, and as such could be optimized for future implementations. Since the turbine uses the same fan blade as the flow sensor it has the same problem of not being designed to catch the flow of water which if fixed, could improve power output. In future iterations, it would be important to consider whether or not to implement a water turbine instead of a flow sensor. The water turbine used, is designed with a baseline resistor of $1M\Omega$ and thus the power obtained is an underestimation from what it could output if the optimal resistor is found. In future iterations, a variable load

resistance should be used to find the maximum power output. Furthermore, calculations of power should also be adjusted as they currently don't account for lateral drag along the fan blades of the water turbine. When implementing the flow sensor, future groups should test for current loading from the motors. Finally, the GPS used was very inaccurate while testing. Further testing should be done to observe whether connecting to more satellites makes the GPS data more accurate. If the increased satellites are still inaccurate a new GPS would be the next step.

VIII. ACKNOWLEDGMENTS

Special thanks to Corey Hickson, for building the new flow sensor using a Computer fan and ideating along Xavier Walter and Ian Smith. Special thanks to team 27 for helping test and calibrate the new flow sensor.

IX. REFERENCES

- [1] D. Zhu, W. Yan, W. Guang, Z. Wang, and R. Tao, "Influence of guide vane opening on the runaway stability of a pump-turbine used for hydropower and
- [2] "IR-Bloc TM Ambient Light Sensor IR-Blocking Silicon Photodiode." Accessed: April 30th, 2024. [Online]. Available:https://mm.digikey.com/Volume0/opasdata/d220001/me_dias/docus/2212/VTP9812FH.pdf
- [3] Texas Instruments, "Special Function Amplifier / Frequency Converters," LM2917-N datasheet, June 2000 [Revised Dec. 2016].
- [4] Foshan Shunde Zhongjiang Energy Saving Electronics Co. "C898+datasheet," YF-S201 datasheet, [Accessed Mar. 2024].
- [5] Spencer, Brake "Lab Manual". Harvey Mudd College. [Accessed April 2024]
- [6] R. Barkalow, "Natural Sunlight Intensity > First Rays LLC," Jul. 08, 2019. <https://firstrays.com/natural-sunlight-intensity/>
- [7] A. Dodson, "Commercial Lighting 101: What is a Foot-Candle?," FSG Electric & Lighting, Dec. 09, 2019. <https://fsg.com/what-is-a-foot-candle/#:~:text=Here%20are%20some%20typical%20foot>(accessed May 01, 2024).
- [8] Ahmededebed, "Old PC Fan- Wind Turbine in 10 Minutes", 2013. <https://www.instructables.com/Old-PC-Fan-Wind-Turbine-in-10-Minutes/> (accessed, March 12)
- [9] RGlenn, "The Reynolds number". Nasa: Glenn Research Center. [Accessed April 2024]
- [10] "CALIFORNIA STATE SCIENCE FAIR 2017 PROJECT SUMMARY." Accessed: May 01, 2024. [Online]. Available:<https://csef.usc.edu/History/2017/Projects/J2106.pdf>
- [11] B. Szyk, "Wind turbine calculator [HAWT and VAWT]," Omni Calculator, <https://www.omnicalculator.com/ecology/wind-turbine> (accessed Apr. 25, 2024). %20the%20future%20of%201%2C050
- [12] "DC-Motors 12VDC 0.2a," Transmotec, [https://www.transmotec.com/product/l3fn-14280-cvc/?c=usd&ga_d_source=1&gclid=\(accessed Apr. 28, 2024\).](https://www.transmotec.com/product/l3fn-14280-cvc/?c=usd&ga_d_source=1&gclid=(accessed Apr. 28, 2024).)

X. APPENDIX

Code Appendix 2: LOGREADER.M FUNCTION FOR ANALYZING DATA COLLECTED BY THE ANEMOMETER

Code Appendix 1: LOGREADER.M FUNCTION FOR ANALYZING DATA COLLECTED BY THE FLOW SENSOR

```

78 % gets t1 vector in time instead of teensy units
79 t3 = t1./10;
80
81 %makes the data into a double so we can find the peaks
82 data = double(A02);
83
84 % puts the red circles on the leading edge peaks
85 [tilde, locs] = findpeaks(data);
86
87 %makes a new vector of the locations of the edges
88 x_values = t3(locs);
89
90
91 x_values(19) = [];
92 locs(19) = [];
93
94
95 x_values(22) = [];
96 locs(22) = [];
97
98
99 x_values(59) = [];
100 locs(59) = [];
101
102 cleanedxval = x_values;
103
104 figure(1)
105 %plots clean data
106 % plot(cleanedxval, data(locs), 'ro')
107 % hold on
108 plot(t3, A02, "blue")
109 title("Current Draw vs Data ")
110 xlabel("Time (sec)")
111 ylabel("Teensy Units")
112 hold on
113 plot(t3, Current_Sense, "-")
114 legend("Data","Current Draw")
115 hold off
116 % Dont put in original data since it still has noise
117 % "After removing noies ... "
118
119 %Getting period
120 period = zeros(1,length(x_values) - 1);
121 for i = 1:(length(x_values) - 1)
122     Period(i) = x_values(i + 1) - x_values(i);
123 end
124
125 %Getting Frequencies
126 freq = 1 ./ period
127
128 %averages the frequencies
129 averagefreq = movmean(freq, 3);
130
131 figure(2)
132 plot(averagefreq)
133 title("Average Frequencies")
134 xlabel("Time (sec)")
135 ylabel("Frequency")
136
137 % 3rd degree polynomial curve (harder to argue values)
138 % freqtospeed = @(x) -6.63*(x^3)+(3.82*(x^2))+(-0.179*x)+0.0608;
139
140 % 2nd degree polynomial curve (hard to argue values)
141 % freqtospeed = @(x) 0.0223+(0.934*x)-0.303*(x^2);
142
143 % linear (easy to argue values)
144 freqtospeed = @(x) (0.0807*x)+0.0326
145
146 speed = [];
147 for i = 1:length(averagefreq)
148     speed = [ speed freqtospeed(averagefreq(i))];
149 end
150
151 %Polynomial 3rd degree
152 %speedtopower = @(x) ((5.76*(10^-8))*(x^3)) + (-1.26*(10^-4))*(x^2)+(0.09*x)-18.
153
154 %Polynomial 2nd degree
155 %speedtopower = @(x) -5.2*(10^-4) + (0.01*x) -0.0583*(x^2) + 0.111*(x^3)
156 speedtopower = @(x) -1.57*(10^-4) +(2.36*(10^-3))*x + (-6.8*(10^-3))*(x^2)
157
158 power = [];
159 for i = 1:length(speed)
160     power = [ power speedtopower(speed(i))];
161 end
162
163 figure(3)

```

```

%% read column-by-column from datafile
fid = fopen(datafile,'rb');
for i=1:numel(varTypes)
    %# seek to the first field of the first record
    fseek(fid, sum(varLengths(1:i-1)), 'bof');

    %# % read column with specified format, skipping required number of bytes
    R{i} = fread(fid, Inf, [* varTypes{i}], colLength-varLengths(i));
    eval(strcat(varNames{i}, '=' , 'R{', num2str(i), '}'));
end
fclose(fid);

%% Process your data here
% yaw_error = (yaw - yaw_des);

t1 = [1:size(u,1)].*0.099;

%t2 = [1:size(yaw_error,1)].*0.099;

% plot(t2, yaw_error)
% title("Yaw Error versus Time ")
% xlabel("Time (sec)")
% ylabel("Yaw Error (degrees)")
% hold on

% gets t1 vector in time instead of teensy units
t3 = t1./10;

%plots the hall effect output
figure(1)
plot(t3, A03)
title("Flow Sensor Output versus Time ")
xlabel("Time (sec)")
ylabel("Voltage [teensy units]")
hold off

%data cleaning (remove false peaks from the data)
x_values(5) = []; % 1.2096 peak
locs(5) = [];

x_values(45) = [] %10.7712
locs(45) = [];

x_values(46) = [] %11.7414
locs(46) = [];

x_values(174) = [] %34.0857
locs(174) = [];

x_values(179) = [] %36.2934
locs(179) = [];

x_values(180) = [] % 36.6003
locs(180) = [];

x_values(180) = [] % 36.6201
locs(180) = [];

%plots clean data
plot(x_values, data(locs), 'ro')
hold on
plot(t3, A03, "blue")
title("Cleaned Anemometer Data ",FontSize= 15)
xlabel("Time (sec)",FontSize= 15)
ylabel("Voltage [teensy units]",FontSize= 15)

% Dont put in original data since it still has noise
% "After removing noies ... "

%Getting period
period = zeros(1,length(x_values) - 1);
for i = 1:(length(x_values) - 1)
    period(i) = x_values(i + 1) - x_values(i);
end

%Getting Frequencies
freq = 1 ./ period

```

Code Appendix 3: SURFACECONTROL.CPP. The code was changed to run a for loop while doing the delay (increased the delay to 20 seconds) to make sure that the AUV stays within a radius of 2 meters of the waypoint selected.

```
if ((dist < SUCCESS_RADIUS && currentWayPoint <
totalWayPoints) || delayed) {
    String changingWPMMessage = "";
    int cwpmTime = 20;

    // navigateDelay
    if (delayStartTime == 0) delayStartTime =
currentTime;
    if (currentTime < delayStartTime + navigateDelay) {
        if (currentTime < delayStartTime + 10000)
//new line
        delayed = 0;
    }
    else{
        delayed = 1;
    }
    //delayed = 1;
    changingWPMMessage = "Got to surface
waypoint " + String(currentWayPoint)
+ ", waiting until delay is over";
}
else {
    delayed = 0;
    delayStartTime = 0;
    changingWPMMessage = "Got to surface
waypoint " + String(currentWayPoint)
+ ", now directing to next point";
    atPoint = 1;
    currentWayPoint++;
}
```