Exploring the Relationship Between Current Velocity and Depth

Henry Heathwood[1], Shreya Jampana[1] , Ian Osborne[1] and Delaney Pratt[1], *Team 22, Section 2, April 26th, 2024*

***Abstract*— This paper presents the design of an Autonomous Underwater Vehicle (AUV), which was designed, built, and tested as a part of the Experimental Engineering course at Harvey Mudd College. The AUV was equipped with sensors to measure the pressure, temperature, and fluid velocity of water. The AUV was deployed at two locations—pHake Lake at the Bernard Field Station and the beach at Dana Point—with the primary goal of measuring and comparing the velocity of current in the water at these locations. In order to model and calibrate the sensors, the velocity of a gantry crane in a freshwater tank was equated to the speed of seawater using similitude. While the team experienced technical difficulties during the deployments, a general understanding of current velocity in these two bodies of water was obtained. Deployment results at different locations in Dana Point led to the conclusion that water near the pier is relatively still, with a velocity detected by the flex sensor stationary around 0.17 m/s at varying depths. The flex sensor was functional, as acceleration of the AUV corresponded with velocity detection of the flex sensor. The effectiveness of the flex sensor in determining current velocity, focusing specifically on its sources of error, is also assessed in the paper. The information on current flow at various locations has important applications for navigation in the ocean and understanding marine life conditions.**

I. Introduction

The goal for this AUV was to create an accurate understanding and calculative approach to determine the fluid velocity under varying environmental conditions. The AUV measured pressure, temperature, and the speed of the fluid. Using the outputs gathered from the sensors, described in further detail later on, the speed of the current at different depths and temperatures could be determined. Calibration curves were necessary to calculate the depth from the pressure sensor and to calculate the velocity from the flex sensor. Both sensors output voltage, which needed to be converted into the desired final output.

To calculate the velocity of the current, the velocity of the AUV could be subtracted from the fluid velocity output by the flex sensor. The expected result was that as depth increased, the current velocity would decrease, meaning that the strongest current would be near the surface of the water. The temperature data can be related to the density of the water. The more dense the water is, the slower the current velocity will be.

II. Experimental Setup

A. Engineering Goal

During this project, the goal was to calculate the current velocity at different depths by utilizing a flex sensor. The flex sensor allowed for the use of fluid dynamics to relate the output of the flex sensor to the velocity of the fluid in which the AUV was moving. This provided useful information that could be important for navigation in the ocean, because knowing current flow can allow for modes of transportation to move with or without the current effectively to use less fuel. It also helps learn about marine life conditions in still bodies of water, such as Phake Lake, compared to active bodies of water, such as the ocean. With the results from the sensors, the team can also analyze the effectiveness of using a flex sensor to determine the current velocity. This is because confounding variables resulting from the team's experimental design, such as human-caused changes in the flex sensor, and from environmental conditions, such as large particles in the water influencing the flex sensor's motion, could affect the current velocity of data.

B. Sensor Selection

The four sensors that the team selected to work with were a pressure sensor, two temperature sensors, and a flex sensor.

The MPX5700 [1], which is a pressure sensor, was chosen to measure the depth of the water. The response time for the sensor is 1.0 ms [1], which was deemed appropriate for the experiment. This sensor allowed the AUV to relate output voltage to the depth in order to control the location of the AUV in the z-direction. This allowed for measurements to be taken at different depths, which were recorded and used to analyze fluid velocity at these different depths. More details on the pressure circuit design are provided below.

One of the temperature sensors, a Type E thermocouple [2], was chosen to measure the temperature of the water. The temperature is useful in determining other properties of the water that can be related to current velocity. Based on the relationship between temperature and salinity, the density of water can be found.

The second temperature sensor, an IC linear thermistor [11], was used to measure the interior temperature

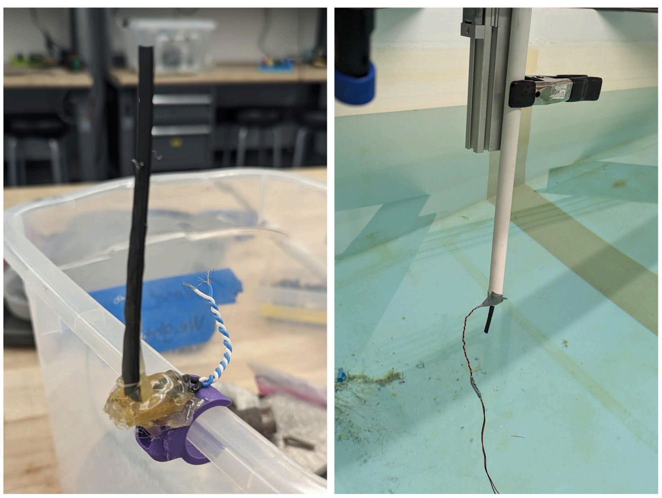
1 All authors are with the Department of Engineering, Harvey Mudd College, Claremont, CA 91711

of the waterproof box. This was necessary to conduct cold junction compensation (CJC) for the thermocouple sensor. This was required for measuring the ocean temperature, as the thermocouple output voltage is based on the difference between the temperatures measured at each end of the thermocouple wire.

The Spectra Flex Sensor [3] was chosen to measure the fluid velocity of the water as the AUV moved. Using the speed of the AUV and the fluid velocity, the current velocity can be calculated by finding the difference between the two. The flex sensor is particularly useful in detecting bending movements, indicating changes in direction or speed when navigating underwater currents.

C. *Fluid's Theory*

Figure 1. Flex Sensor Experimental Design



The flex sensor calibration curves were used to determine the freshwater velocity using the gantry crane from the tank room (Figure 1). From this calibration curve, output velocity could be related to the speed of the crane. The resulting velocity can be used to determine a Reynolds number (for freshwater) which is a dimensionless quantity that connects simulated studies and experimental ones in fluid dynamics.

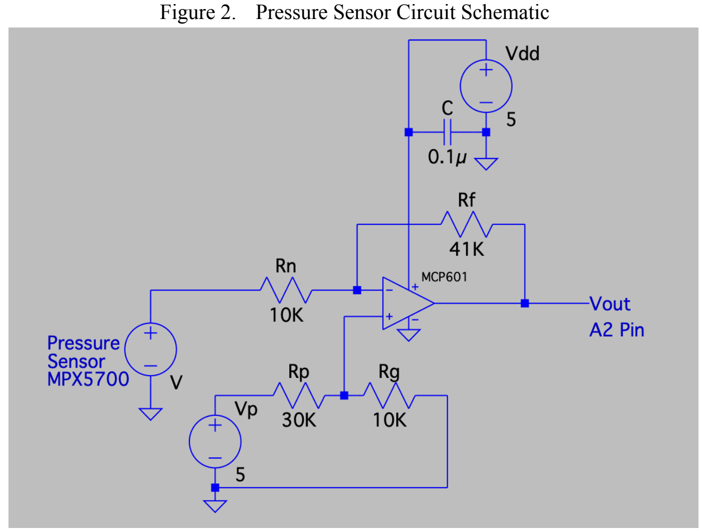
In order to analyze the output data using the Reynolds numbers, equation 1 is used:

$$Re = \frac{\rho U l}{u} \qquad (1)$$

where "ρ" is the density of the fluid, "U" is the characteristic velocity, "l" is the characteristic length of the flex sensor and "u" is dynamic viscosity [4]. Through the Reynolds number, the concept of similitude can be used to allow measurements made on one system to be used to describe the behavior of other similar systems, as a result comparisons between the calibrations with the gantry crane and the AUV sensor data can be made [4]. Doing so, equation 1 can be rearranged to find the velocity of the ocean water since the Reynolds numbers can be equated, and the necessary fluid property variables can be replaced.

D. *Pressure Sensor Circuit Design*

Figure 2. Pressure Sensor Circuit Schematic



An MPX5700 pressure sensor [1] was designed and implemented to monitor the AUV's depth. The pressure sensor circuit was designed to allow the AUV to submerge between 0 and 10 m. Using equation 2, which was provided in the MPX5700 datasheet, this range of depth corresponded to a pressure sensor output voltage range of 0.8514 V to 1.5 V. In equation 2, P denotes the pressure and Vs denotes the input voltage, which was chosen to be 5 V. This input voltage was supplied using a power supply.

$$V_{out} = V_s(0.0012858P + 0.04) \qquad (2)$$

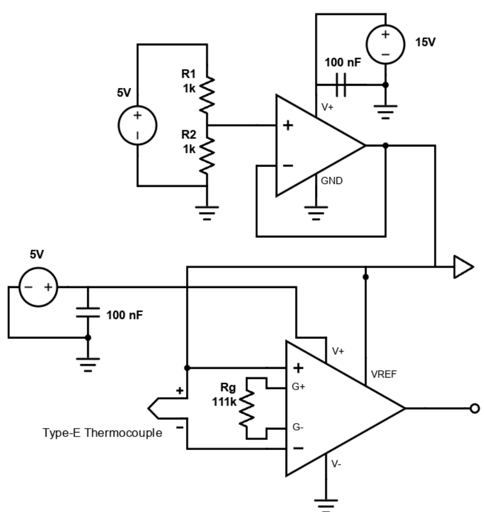
Signal conditioning was then employed to scale and shift this pressure sensor output voltage range to the input voltage range of the Teensy Microcontroller, which is 0 V to 3.3 V. An MCP601 Operational Amplifier [10], which is an inverting Op Amp, with a bias was used to reap the high input impedance and low output impedance benefits of Op Amps, as well as to scale and shift the pressure sensor output voltage range. The combined pressure sensor circuit, which was designed to be an offset amplifier, is shown in Figure 2.

$$V_{out} = \left(1 + \frac{R_f}{R_{n1}}\right)\left(\frac{R_g}{R_p + R_g}\right)V_p - \frac{R_f}{R_{n1}}V_{n1} \qquad (3)$$

Using the input voltage range from the pressure sensor described above, as well as the desired output voltage range of 0 V to 3.3 V, equation 3, which is the equation for the offset amplifier, was used to derive the necessary resistor values for the circuit. The resistors and their values are labeled in Figure 2. The final pressure sensor circuit was designed to have an output voltage range of 0.225 V to 2.88 V, which satisfied the necessary constraint that the output voltage range must be within 0 V and 3.3 V in order to be accepted by the Teensy Microcontroller.

E. Temperature Sensor Circuit Design

Figure 3. Thermocouple Sensor Circuit Schematic



For the Type E thermocouple circuit, an AD623 Instrumentation Amplifier [9] was used to amplify the narrow voltage output range of the thermocouple. Accounting for the reasonable temperature range of the water, the thermocouple had an output range of -0.725mV to -0.364mV. Using equation 4, which was obtained from the AD623 datasheet [9], the gain was set to 910 using a 111kΩ resistor to amplify the voltage range.

$$R_g = \frac{100k\Omega}{(G-1)} \qquad (4)$$

An MCP601 Operational Amplifier [10] was also used to offset the output range by +2.5V. This was accomplished by creating a non-inverting amplifier circuit with an input of 5V and a gain of 0.5. The voltage divider equation (equation 7) was used with values of 1kΩ for R_1 and R_2 to create this desired gain. This output of 2.5V was then connected to the reference junction of the AD623, creating the desired voltage offset. The final designed circuit is shown in Figure 3. This final circuit amplified the voltage output of the thermocouple to be 1.84-2.14V, which allowed it to be read by the Teensy Microcontroller, which has an input voltage range of 0.3-3.3V.

Converting from thermocouple output voltage to measured temperature required a couple of steps. Firstly, equation 5, which was provided in the IC linear thermistor datasheet [11], was used to convert the output voltage of the linear sensor to temperature in celsius.

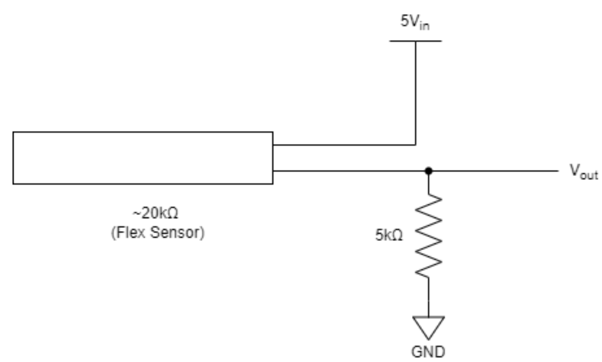
$$V_{out} = (0.0195)\,T_A + 0.4 \qquad (5)$$

Then, the measured temperature of the linear sensor was converted into thermocouple voltage using the NIST ITS-90 table [2]. Next, the gain and offset calculations were reversed from the thermocouple output voltage, and equation 6 was used to calculate the true voltage at the hot junction (V_{hot}) using the converted voltage from the linear sensor (V_{ref}) and the unscaled thermocouple output voltage. Finally, V_{hot} was converted back into degrees celsius using the NIST table.

$$V_{hot} = V_{ref} + V_{meas} \qquad (6)$$

F. Flex Sensor Circuit Design

Figure 4. Flex Sensor Circuit Schematic



To monitor the fluid velocity changes, a flex sensor was designed and implemented. According to the data sheet [3], a voltage divider was used to derive the output voltage values using equation 7:

$$V_{out} = V_{in}\frac{R_2}{R_1+R_2} \qquad (7)$$

Where 5kΩ corresponds to R_2 , ~20kΩ corresponds to R_1, and the V_{in} corresponding to the 5V input from the battery power to the protoboard (Figure 4). This provided a range of values around 0.3-1.6 V for V_{out}, depending on the amount of bend from the flex sensor. As the sensor bends backwards resistance increases, and as it bends forwards it decreases.

G. Sensor Modeling and Circuit Verification

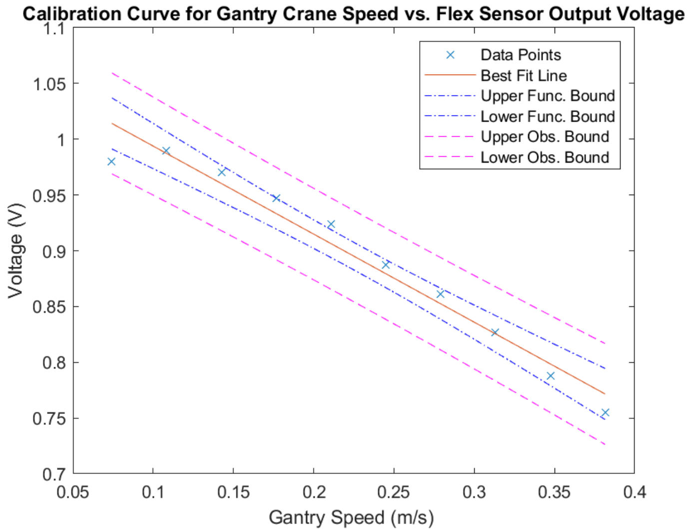


Figure 5. Flex Sensor Calibration Curve

Equation of the line:

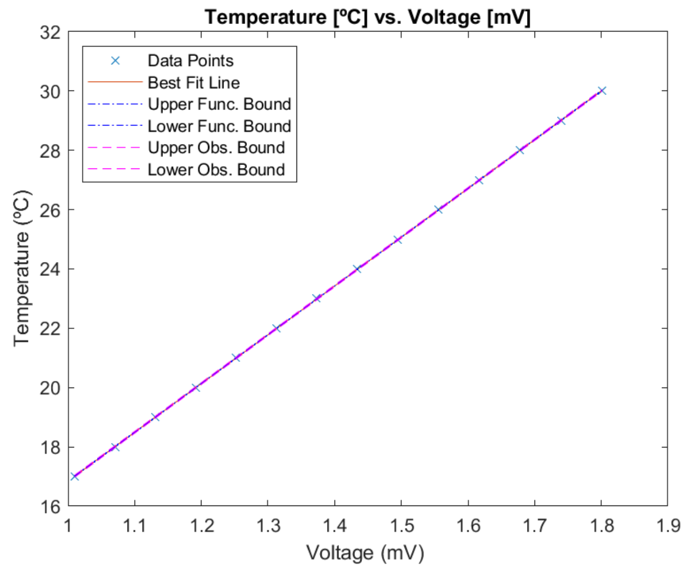
$$Flex\ Output = -0.79x + 1.07 \quad (8)$$

$$R^2 = 0.963$$

Calibration curves and expected sensor output models were created for the sensors in order to convert the voltage outputs read by the Teensy to relevant physical quantities and to verify results, seen in equation 8.

A linear relationship was found for the flex sensor circuit with an ideal R^2 ideal value close to one, reassuring the results were proper. The flex sensor was measured at gantry speeds ranging from 0.1 to 0.35 m/s (Figure 5). This calibration was repeated several times at different speeds with different experimental configurations (Figure 1). The rigidity and flex of the sensor did not return to its uniform state, as its bendiness began to deform after several tests. The original starting output voltage stayed within 15% of the starting output voltage of 1V, resulting in the error bounds seen in Figure 5. The curve had 95% confidence bounds with function bounds of ± 0.0312 and observational bounds of ± 0.1257. These are calculated based on the best fit line plus or minus the variability due to the uncertainty in the slope multiplied by the x-values. These bounds provide an estimation of where future observations are likely to fall, considering the uncertainty in the slope of the regression line. The function bounds indicate the range within where the true regression line may lie, given the current data and a confidence of 95%.

Figure 6. Flex Sensor Calibration Curve



Equation of the line:

$$Temperature = 16.4x + 0.402 \quad (9)$$

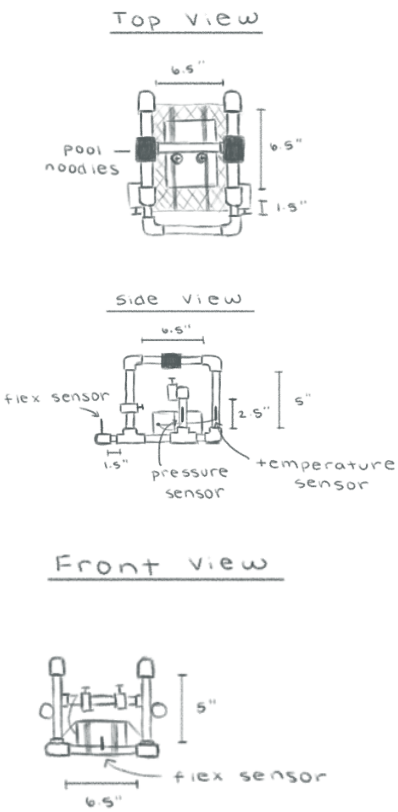
$$R^2 = 1$$

The thermocouple circuit measurements were converted to temperatures in celsius using the NIST ITS-90 thermocouple tables [2]. These tables only report the temperature in increments of 1ºC for given output voltages, as the relationship between voltage and temperature involves a high-degree nonlinear polynomial. However, as shown in Figure 6, the voltage and temperature values within the small relevant temperature range described a perfectly linear relationship that is represented by equation 9. Using this linear fit, all thermocouple circuit measurements were able to be converted to a precise temperature value in degrees celsius. Additionally, these measurements were compared against a ground truth taken at the same time and locations of deployments with a thermometer. This was used to verify the output from the Teensy was correct in terms of temperature measurements.

H. Mechanical Design

The AUV was designed to dive to different depths using P-control and travel autonomously. The AUV would dive to the desired depth, then move forward, repeating this process for different depths in the water. At each depth, the AUV would measure the fluid velocity and temperature which would later be related to current. The AUV would then return to the surface after all the data had been collected.

Figure 7. AUV Mechanical Drawings



As shown in Figure 7, the AUV was designed as a cube, with the bottom front extending outward to hold the flex sensor. The temperature sensor was secured to one side of the AUV and the tubing connected to the pressure sensor wrapped around the entire AUV, where it was then secured to the other side of the AUV opposite the temperature sensor. The waterproof box, containing the main electronics including a Teensy Arduino Microcontroller, an IMU, and a GPS, was secured to the bottom of the AUV with velcro and netting. The box contained three 0.5 inch holes, which had penetrator bolts leading out of them. Two of these bolts needed to be made for this project with marine epoxy. One penetrator bolt connected the thermocouple and the flex sensor wires, and the other was for the pressure tubing. The AUV used a total of four motors, two in the horizontal direction, and two in the vertical direction. The horizontal motors drew similar current values which ensured that the AUV would not veer in one direction. The same was true for the two vertical motors. Two motors in the vertical direction provided extra power to ensure that the AUV would be able to successfully reach the surface after diving.

The AUV was designed to be neutrally buoyant, ensuring that it could easily travel to the desired depths as well as return to the surface. To calculate the amount of ballast weight necessary, the following equation was used:

$$mg = \rho g V + 10N \qquad (10)$$

The expected buoyancy force of the AUV represented the 10N in equation 10, and the expected buoyancy force for the waterproof box was 19.92N, calculated given a density of 1000kg/m^3, and a volume of 0.00225m^3. This resulted in a total buoyancy force of 29.92N. Given that the AUV had a total mass of 1.66kg, and the mass needed to be neutrally buoyant was 3.05kg, there needed to be an additional 1.39kg of weight added with ballast. This accounted for the AUV being neutrally buoyant in fresh water, but additional ballast was needed to account for the increased density of the salt water during deployment at Dana Point. In both deployment in freshwater and saltwater, the AUV was slightly positively buoyant to ensure that the AUV would successfully be able to return to the surface. The center of mass of the AUV was located between the two vertical motors, which ensured that the AUV would not roll while diving.

The flex sensor mechanical design was reinforced with a layer of heat shrink in order to create more stiffness to remove some of the sensitivity from bending. The threads of the flex sensor were anchored onto a pcb which was then soldered to two external wires that connect back to the main motherboard PCB (Figure 1). In order to water the soldered connections, multiple layers of hot glue were used to seal any contact with external factors such as seawater. Additionally, in order to mount the sensor to the AUV, a ½ in 3D printed PVC mount from Thingiverse, created by Wiskers3, in order to support the placement of it at the front of the AUV (Figure 7).

I. Navigation Code

In addition to the software provided by the Experimental Engineering course, some changes were implemented to the AUV code to achieve the desired navigation. A proportional depth control algorithm was implemented for the AUV to travel to certain desired depths. In the range of desired depths for the AUV to dive down to, four waypoints were chosen at which the AUV paused momentarily to allow for data collection. At each of these desired depths, once the AUV paused, the two side motors were coded to turn on, allowing the AUV to travel forward using open loop navigation for 10 seconds before descending to the next depth waypoint. The AUV traveling forward allowed the flex sensor to bend more, thus allowing for the ease of obtaining fluid velocity measurements.

J. Experimental Procedure

The goal of the forward motion at each depth waypoint was to measure the underwater current using the flex sensor and find the difference between the measured current velocity and the internally recorded accelerometer velocity. This would allow the team to find the true velocity of the underwater current. After the AUV reached the final depth and performed its final forward thrust sequence, it would then reverse its vertical motor power and return to the surface.

The team attached a long tether to the AUV frame in order to prevent loss of the AUV in an emergency. All electronics were contained inside a waterproof box that must be sealed before deployment. These electronics included the power switch for the system which would automatically begin running the code when switched on. For this reason, there was also a ~40 second delay added at the beginning of the code to allow time for the team to seal and secure the waterproof box after turning on the switch before the motors started running.

For each round of data collection, the team would begin by inserting the micro SD card into the port on the motherboard. Then, the motherboard would get switched on, and the team would work together to seal the waterproof box and secure it to the AUV frame using velcro. One team member would then hold the AUV just below the surface of the water until the code delay finished and the motors began to run. Another team member would hold the AUV tether and slowly feed it into the water as the code sequence was performed. After the AUV finished its dive and returned (or was pulled up) to the surface, the waterproof box on the AUV was carefully wiped with a paper towel. This was to ensure that when the box opened, no water would fall into the box and damage the electrical components. After wiping the box down and opening it, a team member with dry hands would reach into the box and turn off the switch on the motherboard, ending the run. As a safety consideration, it was important that an individual with dry hands performed this last task so as to prevent an electrical shock.

III. Results and Analysis

A. Deployments Overview

Data collection was conducted at two different locations: pHake lake located at the Bernard Field Station (BFS) in Claremont, CA and at Dana Point, CA.

Deployment at pHake lake resulted in two successful tests, confirming that the AUV could successfully dive. The PWM of the motors had been set to 255, which is the maximum possible power, during deployment at pHake lake, however this was causing the motors to short, therefore the PWM was decreased to 200 for future deployments. After analyzing the data, it was determined the temperature and the pressure sensors were not reading properly, and the AUV stopped logging data after the 80 second delay, which was set for the pHake Lake deployment) due to the shorting problems in the motors.

Next, the AUV was deployed at Dana Point. The AUV was originally deployed in multiple locations from a kayak. These locations included midway through the channel entrance and in the waves near the beach. The AUV was placed in the water midway through the channel entrance and completed multiple runs successfully, as the team watched the AUV dive and move forward as intended, before returning to the surface. The AUV was then placed in the waves at the beach in order to gather flex sensor data from the waves.

Figure 8. PCB extracted from flooded AUV box

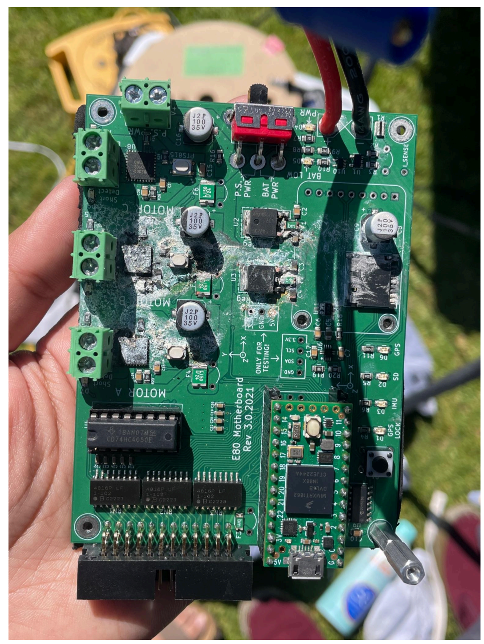
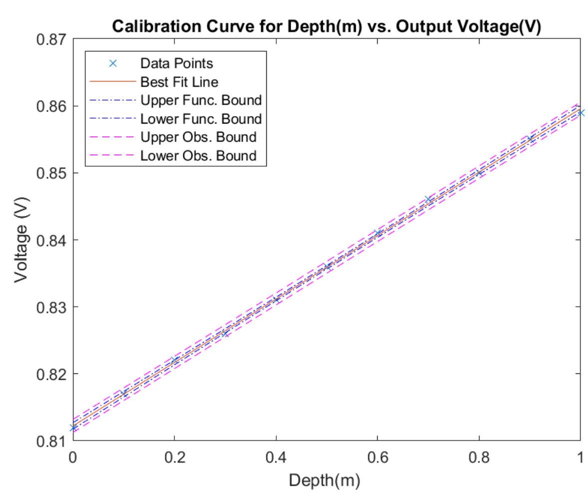


Figure 9. Pressure Sensor Updated Calibration Curve



Equation of the line:

$$Pressure\ Output\ =\ -\ 0.0474x\ +\ 0.812 \tag{11}$$

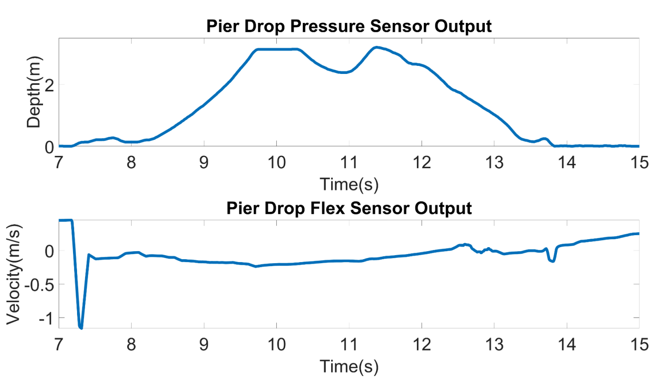
$$R^2\ =\ 0.999$$

After two hours spent gathering data on the kayak, the team returned to shore to analyze the data that had been collected, only to discover that their box had become flooded, damaging the motherboard, causing all the data to be lost. The damage to the motherboard, which is shown in Figure 8, made it unusable, so a new motherboard was needed in order to reattempt the deployment. The water damage also changed the resistances of the resistors within the pressure sensor circuit, which resulted in the need for a new calibration curve, using equation 11, once the team returned from Dana Point, where a similar approach from Figure 5 was used to determine the 95% confidence bounds (Figure 9). The changes in the resistor value meant that the P-control no longer worked properly, so the AUV could no longer dive to the desired depths.

By the time that the motherboard had been replaced and the AUV was ready for deployment again, the bus that was taking students back to Harvey Mudd College from Dana Point was about to leave. This meant that the team had very little time to gather data, which limited the time that was spent on each run. Without working P-control, the best solution was to gather data by first dragging the AUV through the waves at the beach, and then dropping the AUV off the pier. Both of these deployments resulted in good data from all three sensors; however depth needed to be recalibrated at a later time (Figure 9). Both of these deployments lasted for approximately 25 seconds due to the fact that the bus was getting ready to leave, however the AUV had proved that it could run successfully for more than a minute at both pHake lake and during the kayak deployment.

B. Sensor Results

Figure 10. Pressure and Flex Sensor Data Collected at Dana Point Harbor Pier



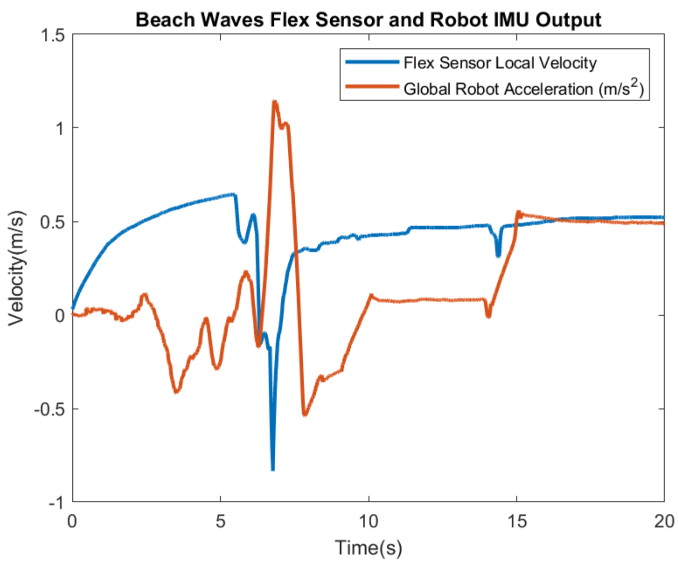
As previously described, due to the flood causing the P-control to stop functioning, the AUV was dropped off into the water at the pier to record data from the sensors, which is shown in Figure 10. As the pressure sensor output in Figure 10 shows, the AUV dove to a maximum depth of 3.2 m. Throughout the course of the dive, the velocity outputted by the flex sensor, also depicted in Figure 10, is relatively constant around 0.17 m/s. The initial spike in the flex sensor output can be attributed to the AUV being manually placed into the water after turning it on, closing the box, and dropping it in the water at the 7 second mark, causing the flex sensor to quickly bend. Around the 14 second mark is when the AUV was pulled out of the water, and dried off until being shut off several seconds later. As this data demonstrates, there is no significant change in the water current at varying depths near the pier. This low fluid flow, even at the surface of the pier, can be explained by the low boat activity near the pier.

Possible sources of error can be posed upon analysis of the experimental and collected data. The small fluctuations in the flex sensor output line up with the fluctuations in the pressure sensor output, as seen around 14 seconds. Therefore, these fluctuations are not representative of true changes in fluid flow velocity; rather they are a product of experimental errors and inaccuracies. The flex sensor is highly sensitive, with even a small bend causing a stark change in output voltage. Therefore, due to the AUV being manually placed into and pulled out of the water at the pier, human error, such as uneven tugs in the tether, could correspond to drastic changes in the flex sensor output. Due to this, only general fluid flow velocity trends—such as the water being relatively still at the pier—can be observed from the data. External environmental sources that cause changes in the flex sensor output were not carefully eliminated, so although the flex sensor's sensitivity might be beneficial in measuring minute changes in current, its sensitivity coupled with the data collection method limited its ability to do so.

Another source of error is the inaccuracies in similitude, which is a concept relating the behavior of an object in one flow field to its behavior in another flow field. More specifically, the dynamic viscosity is used within the Reynolds number to establish similitude. It represents the internal friction of a fluid, measuring how much resistance it offers against deformation when subjected to external shear stress (pressure) [5]. It helps in predicting the flow behavior and rate of the freshwater, or any fluid. This was affected within the gantry crane calibration process, where the relationship to the dynamic viscosity could not be measured. An online assumption was used for this value based on local temperatures and densities in the water. It is affected by varying temperatures and pressures in a fluid, where the rate of change of velocity is proportional to the shear stress, when the direction of the fluid is perpendicular to the flow [5]. If the dynamic viscosity is not matched with the actual conditions within a fluid, such as with the gantry crane or the ocean, then it could lead to many inaccuracies within the final results.

For the thermocouple results, a ground truth of 17.8ºC was obtained using a thermometer at the pier. The initial goal was to measure the temperature at different depths, but since the temperature was found to be constant across the entire range of depths, the average value was taken to compare to the ground truth. Equation 9 was used to calculate a temperature value of 19.88ºC, which has a percent difference of 11% with the ground truth measurement.

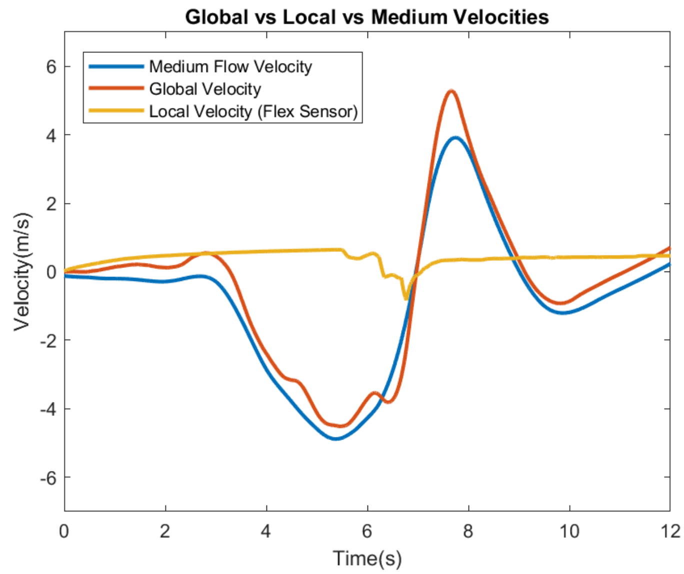
Figure 11. AUV X-Axis Movement and Flex Sensor Data at Dana Point Beach



At the Dana Point beach launch, the flex sensor peaked at 0.64 m/s around 5 seconds and had an average velocity around 0.42 m/s, as shown in Figure 11. Between 0-5 seconds, the AUV was placed into the wave flow and moved against the water coming at it. Since the motherboard flooded, the AUV motion was manually replicated. An initial flood current hit the AUV at 5 seconds and was followed by an ebb current. The flood current occurs when the waves are coming onto the shore, and the ebb current occurs when the seawater is flowing back out to the Ocean [6]. The incline in the sand at the beach produces a stronger ebb current than flood current as it flows back into the main body of water backwards. The IMU had manufacturing errors built into it, causing the acceleration output to be much larger than expected. IMUs generally are manufactured with compensation in order to increase the stability of the measurements [7].

In theory, to determine the medium flow velocity (current flow velocity), the global velocity (IMU) would be subtracted from local velocity (flex sensor). The medium is the value between the two extremes that were being measured through the IMU and the flex sensor. These velocities are shown below in Figure 12.

Figure 12. The Subtracted Difference Between the IMU and Flex Sensor Data at the Beach



To find the global velocity from the IMU global acceleration data, equation 12 was used:

$$V = Vo + a * dt \qquad (12)$$

where V is the global velocity, Vo is the initial, a is the acceleration, and dt is the change in time. Due to the significant error in the global acceleration data propagation, the global velocity was unrealistic in terms of the AUV's actual movement, approaching speeds above 4m/s. Since the error of IMU was integrated, it resulted in unrealistic results.

When integrating a value that has a significant amount of error, the cumulative effect of these errors tends to amplify over the course of the integration. This happens because integration sums up values over an interval. Each small error in the measurement contributed to an increasingly larger deviation from the true value as the integration proceeded. This is especially prevalent in the demonstration of the AUV sensor measurement systems where errors are correlated to the IMU. These do not cancel out but rather accumulate, leading to increased errors in the integrated result.

IV. Conclusion

A. Main Results

Despite a number of setbacks, the team was able to draw successful conclusions from the data that was collected. The current velocity that was calculated from the flex sensor output may not be accurate for this deployment, but provided that future work can reduce the amount of error, useful data can be gathered about the current velocity in the ocean.

The flex sensor performed well in both the deployment at the beach and the deployment at the pier, performing as expected. At the beach, the flex sensor experienced positive velocity which corresponded with the flood current, and the sharp decline in velocity corresponded with the ebb current of the waves. This ebb current caused an increase in the acceleration of the robot, and once the flood current returned, both the acceleration of the AUV and the velocity detected by the flex sensor restabilized. The flex sensor velocity at the pier remained at zero, which means there was no current flowing in this area of the ocean. The error from the IMU caused the largest error in the current velocity, meaning that if this error is reduced, the current velocity can be more accurately calculated using the same methods.

While the flood in the box caused the P-control to stop working, the pressure sensor still successfully reported the depth at which the AUV gathered data at. The pressure sensor allowed for depth to be related to the flex sensor output, even if there was no control over the depth.

The temperature sensor reported an average ocean temperature of 19.88ºC. Compared to the ground truth temperature measurement of 17.8ºC, this meant there was an 11% difference between the temperature. The temperature of the ocean found with the thermocouple was used to calculate the density of the ocean water, which provided information on current velocity.

B. Significance of Results

The main goal of the project was to understand the current velocity in the ocean, which can provide useful information about how currents are moving underwater. Though the results from the sensors were largely unsuccessful in reporting the actual velocity of the current, other methods can be explored that may be more successful.

Ocean currents are important to understand because of the ways in which they contribute to different environmental factors. Ocean currents are important in controlling the climate, as well as in understanding marine life. [8] The current velocity can provide information on when currents are changing, to understand what is causing any environmental changes. In addition, the temperature and pressure sensors can show how the temperature and depth of different currents can change over time, provided that data is consistently gathered over long periods of time.

C. Recommendations for Future Work

As previously discussed, flooding of the box housing all of the electrical components contributed to the majority of the errors in the experiments. To prevent this in the future, more precautionary waterproofing must be done. A possible reason for the flood could be that due to a high amount of pressure at very large depths, the penetrator bolt and epoxy combination was not strong enough to hold the waterproofing. Two possible solutions for this could be exploring and experimenting with other waterproofing techniques and diving to shallower depths to avoid the high pressure. Upon multiple experiments, it was discovered that water was leaking in through the thermocouple from outside to inside the box. Although the issue was quickly bandaged with parafilm, more sustainable waterproofing for the thermocouple could have prevented extra water from entering the box.

In order to avoid some of the sources of error described above, better calibration methods for the sensors could have been employed. The pressure sensor was originally designed for depths ranging from 0 to 10 meters, which is a range far beyond the diving range of the AUV. Therefore, redesigning and calibrating the pressure sensor for a smaller range of depths could lead to more sensitive readings from the pressure sensor at small variations in depth.

Exploring calibration methods for modeling sea water rather than fresh water with the gantry crane could remove error between fluid environments by removing the density variable changes from freshwater. Additionally, to create better similitude relationships, tools like viscometers could be used to better approximate the dynamic viscosity. Using multiple methods and simulation should be used in the future using software, like COMSOL, to compare against wind tunnel, gantry crane, and sea water tank experiments.

One of the main goals of this project was to determine the velocity of the ocean water at varying depths. Due to human error and the high sensitivity of the flex sensor, in the future, exploring other mechanisms, such as a flow sensor, to test current velocity might be useful. It might also be beneficial to get data from multiple flow sensors and take that average, as this could lead to a higher accuracy in produced results.

V. Acknowledgements

The team would like to thank Team 15 for their help in lending us their motherboard when the AUV flooded in order to gather last minute data at Dana Point. We'd also like

to thank Xavier Walter, Lynn Kim, and the E80 Professors for their continued support throughout the course!

VI. References

[1] Freescale Semiconductor, “Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated.” MPX5700 Datasheet, 2007-2009 [Revised Oct. 2012]

[2] National Institute of Standards and Technology, *NIST ITS-90 Thermocouple Database*. U.S. Secretary of Commerce, 2000 [Revised Apr. 2008]

[3] "SpectraFlex Flex Sensor Specifications and Usage Guide," SpectraSymbol, Salt Lake City, UT, USA, Rev. A1, pp. 1-2. Available: spectrasymbol.com. Accessed: Apr. 29, 2024.

[4] A. Dato. ENGR 80. Class Lecture, Topic: “Fluid Measurements” Engineering, Harvey Mudd College, Claremont, CA, Feb. 21, 2024.

[5] "Dynamic Viscosity," Vaia, 2024. [Online]. Available: https://www.vaia.com/en-us/explanations/engineering/engineering-fluid-mechanics/dynamic-viscosity/. [Accessed: Apr. 29, 2024].

[6] "Introduction to Tides," National Oceanic and Atmospheric Administration, U.S. Department of Commerce, 2024. [Online]. Available:https://oceanservice.noaa.gov/education/tutorial_tides/tides01_intro.html. [Accessed: Apr. 29, 2024].

[7] "IMU Errors and Their Effects," APN-064, Rev. A, Feb. 21, 2014, pp. 1-6.

[8] “Ocean Currents,” *education.nationalgeographic.org*. https://education.nationalgeographic.org/resource/ocean-currents/#

[9] Analog Devices, “Single and Dual-Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier,” AD623 datasheet, Aug. 1999 [Revised Aug. 2020].

[10] Microchip, “MCP601/1R/2/3/4” MCP601 Datasheet, 2007

[11] Microchip, “Low-Power Linear Active Thermistor ICs,” MCP9701A datasheet, July. 2015.

CODE APPENDIX 1: Calibration_Curve.m FUNCTION FOR CREATING A CALIBRATION CURVE

```
% The x, r, and y below are for testing. Comment them out and add your own
% data to the sample data at the end of the comments, and uncomment your
% data
x = [];% X axis data points input
r = normrnd(0,1,[1,length(x)]);
y = []; % Y axis data points input
% Script for the linear fit of data. The independent values are
% in the x array and the matched dependent values are in the y array. This
% script does not use MATLAB's built-in fitting functions, but uses the
% formulas from the videos/class notes.
% You also need to enter the confidence level, typically 95%. The values
% that are calculated and displayed are:
% 1. Beta_hat_1 (the best-fit slope)
% 2. Beta_hat_0 (the best-fit y-intercept)
% 3. The Root Mean Square Residual, Se
% 4. The Standard Error for beta0, Sbeta0
% 5. The Standard Error for beta1, Sbeta1
% 6. The confidence intervals for beta1 and beta0.
% After calculating these quantities, the script plots the original data,
% the best fit line, and the upper and lower bounds for the confidence
% interval on the best fit line.
% x = [7.5 4 6 5 8]; % Uncomment and add your own data
% y = [3.70 3.10 3.32 2.98 3.68]; % Uncomment and add your own data
confLev = 0.95; % The confidence level
N = length(y); % The number of data points
xbar = mean(x);
ybar = mean(y);
Sxx = dot((x-xbar),(x-xbar));
%Sxx = (x-xbar)*transpose(x-xbar);
% beta1 is the estimated best slope of the best-fit line
beta1 = dot((x-xbar),(y-ybar))/Sxx
% beta1 = ((x-xbar)*transpose(y-ybar))/Sxx
% beta0 is the estimated best-fit y-intercept of the best fit line
beta0 = ybar - beta1*xbar
yfit = beta0 + beta1*x;
SSE = dot((y - yfit),(y - yfit)) % Sum of the squared residuals
% SSE = (y - yfit)*transpose(y - yfit) % Sum of the squared residuals
Se = sqrt(SSE/(N-2)) % The Root Mean Square Residual
Sbeta0 = Se*sqrt(1/N + xbar^2/Sxx)
Sbeta1 = Se/sqrt(Sxx)
% tinv defaults to 1-sided test. We need 2-sises, hence:(1-0.5*(1-confLev))
StdT = tinv((1-0.5*(1-confLev)),N-2) % The Student's t factor
lambdaBeta1 = StdT*Sbeta1 % The 1/2 confidence interval on beta1
lambdaBeta0 = StdT*Sbeta0 % The 1/2 confidence interval on beta0
range = max(x) - min(x);
xplot = min(x):range/30:max(x); % Generate array for plotting
yplot = beta0 + beta1*xplot; % Generate array for plotting
Syhat = Se*sqrt(1/N + (xplot - xbar).*(xplot - xbar)/Sxx);
lambdayhat = StdT*Syhat;
Sy = Se*sqrt(1+1/N + (xplot - xbar).*(xplot - xbar)/Sxx);
lambday = StdT*Sy;
figure(1)
plot(x,y,'x')
hold on
plot(xplot,yplot)
plot(xplot,yplot+lambdayhat,'-.b',xplot,yplot-lambdayhat,'-.b')
plot(xplot,yplot+lambday,'--m',xplot,yplot-lambday,'--m')
title('X vs. Y') % X vs. Y
xlabel('X Axis ')
ylabel('Y Axis')
if beta1 > 0 % Fix this
    location = 'northwest';
else
    location = 'northeast';
end
legend('Data Points','Best Fit Line','Upper Func. Bound',...
    'Lower Func. Bound', 'Upper Obs. Bound', 'Lower Obs. Bound',...
    'Location', location)
hold off
```

CODE APPENDIX 2: logreader.m FUNCTION FOR ANALYZING DATA COLLECTED BY THE AUV

```
% logreader.m
% Use this script to read data from your micro SD card
clear;
%clf;
filenum = '023'; % file number for the data you want to read
infofile = strcat('INF', filenum, '.TXT');
datafile = strcat('LOG', filenum, '.BIN');
%% map from datatype to length in bytes
dataSizes.('float') = 4;
dataSizes.('ulong') = 4;
dataSizes.('int') = 4;
dataSizes.('int32') = 4;
dataSizes.('uint8') = 1;
dataSizes.('uint16') = 2;
dataSizes.('char') = 1;
dataSizes.('bool') = 1;
%% read from info file to get log file structure
fileID = fopen(infofile);
items = textscan(fileID,'%s','Delimiter',',','EndOfLine','\r\n');
fclose(fileID);
[ncols,~] = size(items{1});
```

```
ncols = ncols/2;
varNames = items{1}(1:ncols)';
varTypes = items{1}(ncols+1:end)';
varLengths = zeros(size(varTypes));
colLength = 256;
for i = 1:numel(varTypes)
   varLengths(i) = dataSizes.(varTypes{i});
end
R = cell(1,numel(varNames));
%% 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{',num2st
   r(i),'};'));
end
fclose(fid);
%% Process your data here
%Conversions to acceleration units from
   teensy units
NewZ=(accelZ)*.01;
NewY=(accelY)*.01;
NewX=(accelX)*.01;
samplingFreq = 10;
numSamples = length(accelZ);
samplingPeriod = 1/samplingFreq; % s
totalTime = numSamples*samplingPeriod; % s
t = linspace(0,totalTime,numSamples);
%teensy logger data, converted from teensy
   units to voltage
aflexsensor= (double(A00)*3.3)/1023;
athermocouple  = (double(A01)*3.3)/1023;
apressuresensor = (double(A02)*3.3)/1023;
alinearsensor = (double(A03)*3.3)/1023;
subplot(5,1,1)
plot(t,athermocouple)
title('thermocouple hot junction')
xlabel('time(s)')
ylabel('i dont know')
subplot(5,1,2)
plot(t,apressuresensor)
title('pressure sensor')
xlabel('time(s)')
ylabel('voltage')
subplot(5,1,3)
plot(t,aflexsensor)
title('flex sensor')
xlabel('time(s)')
ylabel('voltage')
subplot(5,1,4)
plot(t,alinearsensor)
subplot(5,1,5)
plot(t,Current_Sense)
%%
%%
% Plots generated here of x, y, and z
   accelerations
figure(2)
plot(NewZ,"r", LineWidth=2)
hold on
plot(NewX,"b", LineWidth=2)
plot(NewY,"g", LineWidth=2)
title("Acceleration Data X,Y,And Z")
xlabel("Number of Samples")
ylabel("Acceleration [m/s^2]")
legend("Z","X","Y")
%xlim([0,250]); %cuts off sides of graph to
   remove empty space
hold off
```

CODE APPENDIX 3: DataValueConversions.m FUNCTION FOR CALIBRATING DATA COLLECTED BY THE AUV

```
depth = (apressuresensor-0.858)/0.0474;
   %pressure calibration curve adjusted
   for sea level
depth1 = (apressuresensor-0.812)/0.0474;
   %pressure calibration curve original
   (not adjusted)
FlexVelocity =
   ((aflexsensor-1)-1.07)/-0.79; %Flex
   sensor Velocity Extraction from
   Calibration Curve
rhoA = 1.293; %density of air
rhoW = 1025; %density of seawater
   %%%%%%%%%%%%%%CHANGE THIS ACCORDING TO
   TEMPERATURE FROM XAVIER CHART
rhoO = 997; %density of water
l = 0.1125; % characteristic length
Uw = 1.0*10^-3; %kinematic viscosity water
Uo = 1.2*10^-3;%kinematic viscosity ocean
   %%%%%%%%%%%%%%CHANGE THIS ACCORDING TO
   TEMPERATURE using google
ReW = (FlexVelocity*l*rhoW)/(Uw);%reynold's
   number water
OceanFlexVelocity =
   (Uo*ReW)/(l*rhoO);%reynolds number
   ocean
IMUa=movmean(NewX,100)-1.091666+.00723
   %acceleration of IMU in X direction
Vimu = zeros(length(IMUa)); %Integration of
   global acceleration from IMU to find
   global velocity from IMU
for i = 1:length(Vimu)
   Vimu(i+1) = Vimu(i) + IMUa(i)*0.1;
end
bestflex = movmean(OceanFlexVelocity,10); %
   Creating a rolling average to provide
   smoother data
bestIMU = Vimu; % Creating a rolling
   average to provide smoother data
currentspeed =
   movmean((bestIMU-bestflex),100);
```