Tracking Fish Environments with an Autonomous Underwater Vehicle

E80: Section 4, Team 5 - May 1, 2018
Michael Guzman, Anant Kandoi, Pinky King, and Vicki Moran

Abstract—**This report details the creation of an autonomous underwater vehicle that can navigate to an underwater acoustic beacon and collect data about the water surrounding the beacon. The beacon and vehicle simulate fish tracking, with the beacon representing an acoustic tag on a fish and the vehicle representing a tracking device that collects data about the fish environment. The vehicle took temperature and turbidity measurements, as these are valuable indicators of the quality of a fish's environment. In a deployment at Dana Point, the vehicle failed to navigate autonomously towards the beacon due to problems with the microphones. The temperature data logged by the AUV was inconclusive, while turbidity varied from 200 to 600 NTU.**

I. Introduction

The objective of this project was to design, build, and deploy an autonomous underwater vehicle (AUV) with a budget of $75 for Harvey Mudd College's Experimental Engineering (E80) class. The vehicle had to employ at least three sensors to collect data, be able to navigate actively and autonomously, and return to the deployment location without harm. The specific data that the vehicle collected from the environment depended on the scientific and engineering goals of the project.

Underwater vehicles have previously been explored in Harvey Mudd College's Engineering Systems (E79) class. In the class, students created a tethered remotely operated vehicle, which used basic proportional control to collect depth, temperature and pressure data. In E80, this design was expanded upon to include other sensors and navigation techniques.

This paper describes the process of constructing and deploying an AUV. This AUV was designed to track an acoustic signal, simulating acoustic fish tracking. Thus, in addition to the AUV, a beacon was constructed with a speaker that constantly emitted a 10 kHz acoustic signal. The AUV employed microphones to listen for this signal and then attempted to navigate in the beacon's direction. As the AUV navigated toward the beacon, it collected data about its environment, specifically temperature and turbidity, while also logging its GPS location. This kind of navigation and data collection could be useful in determining what kinds of environments fish prefer and in understanding the effects of climate change and ocean warming.

II. Goals

In order to study the migratory behavior of fish, an AUV can follow a tagged fish and collect information on the surrounding environment. Collecting and analyzing this data can paint a picture of fish migration patterns and preferred environments. Long-term tracking of fish using acoustic tags and data collection using various sensors can fulfill this objective.

As climate change progresses, a major implication is the impact of ocean warming on fish livelihood and survival. Fish will have to change their migratory patterns and travel to different areas in order to survive. Tracking the temperatures at different locations will offer insight into the fish's ideal living conditions while dealing with climate change. In addition to temperature, turbidity is a valuable quantity to measure, as it is a strong indicator of water quality and pollution. High turbidity environments affect the survival of both vegetation and wildlife in an environment, with values exceeding 20,000 NTU killing certain species of fish in under 24 hours [2][3][12].

III. Sensor Selection

A. *Thermistor*: The Vishay NTCLE100E3104JB0 100 kΩ thermistor was selected to record temperature data. This thermistor has a response time of approximately 1.9 seconds [4]. The thermistor is easy to install and only requires a simple voltage divider circuit to output a temperature measurement. It is also simple to waterproof and can be done with heat shrink alone. A thermocouple would be more accurate and have a faster response time, but increases the complexity to install, and was thus not chosen for this design. The operating temperature for this thermistor ranges from -40ºC to +125ºC, more than sufficient for the temperatures expected at Dana Point [5].

B. *Turbidity Sensor*: The Gravity Analog Turbidity Sensor was selected for its cost and ease of use. The sensor cost just $10, fitting comfortably within the $75 budget. This sensor measures turbidity by measuring the transmittance and scattering of light between the two probes. The sensor can only accurately measure turbidity underwater, but only the probe portion of the sensor is waterproof, meaning the rest must be waterproofed. It has a response time of under 500 ms, according to its data sheet [6].

C. *Speaker:* The Dayton DAEX25W-8 Audio Exciter was selected for use as a speaker on the acoustic beacon because it can handle extreme outdoor environments and was previously used in E80 Lab 4. This waterproof speaker has a low impedance of 8 Ω and RMS power handling of 10W [7].

D. *Microphones*: The CME 102-2190-ND Microphones were used in E80 Lab 4, which made the circuit design easier to plan and implement. Each microphone cost less than $4, allowing for three in the $75 budget [8][9].

IV. Design

A. Mechanical Design

Acoustic tracking can be implemented by tagging a fish with an acoustic tag and picking up the emitted sound from microphones. A small beacon with a speaker simulated the fish and an autonomous underwater vehicle listened to the acoustic signal with microphones for navigation. To successfully implement this objective, the directionality of the microphones had to span a range such that the signal from the speaker can be heard at any heading. Three microphones are sufficient for spanning all 360º on the surface of the water. The vehicle compared the three signals from the microphones and dynamically found the largest signal in order to rotate and move in that direction.

Since there were three microphones, the robot was designed in a triangular shape. To create a triangle, three PVC pipes were connected with 60º elbows, which are not commercially available due to the sharp angle. Therefore, an adapter was custom-designed and 3D-printed using SolidWorks to fit inside the end of each PVC, allowing for a triangular frame. Another PVC pipe was attached to each joint and went straight down into the water, holding the motors and microphones at a certain depth (Figure 2). The joints had a hole for the microphone wires to go through, as depicted in Figure 1.

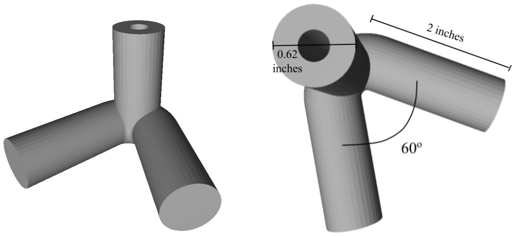


Figure 1: 3D-printed PVC joints.

Although fish realistically move in all directions, accounting for all three dimensions added a level of complexity that was deemed outside of the scope of this project. Thus, the vehicle was restricted to move only along the surface of the water. Since it was unable to dive, the temperature and turbidity were also only measured at the surface of the water. This limitation means that temperature measurements did not reflect the temperature beneath the surface, because the temperature drops significantly as distance from the surface increases. Therefore, the recorded data should only be taken as an approximation of the actual fish environment in the lower depths at which they swim. Turbidity is highly dependent on the surrounding light of the environment, and deeper waters have a lot less sunlight. This makes the turbidity measurements approximations as well.

The robot had to be fully buoyant to remain on the surface of the water. To do so, several pool floaties made from low density polyethylene foam surrounded the PVC pipes that formed the main frame to offset the weight of the robot. Because only a positive buoyancy was desired, the number of floaties needed was determined experimentally. A mesh was attached across the triangular frame to hold the orange waterproof box containing the circuit boards above water, ensuring GPS connectivity. The microphones and thermistor were connected to the motherboard via waterproof penetrators on the box. Another box containing only the turbidity sensor was placed below the water, allowing for the turbidity sensor to always collect accurate data. The final design is shown in Figure 2, with the turbidity sensor box located directly below the orange box. Each side of the triangle is 0.24 meters long to allow for sufficient separation between each microphone.

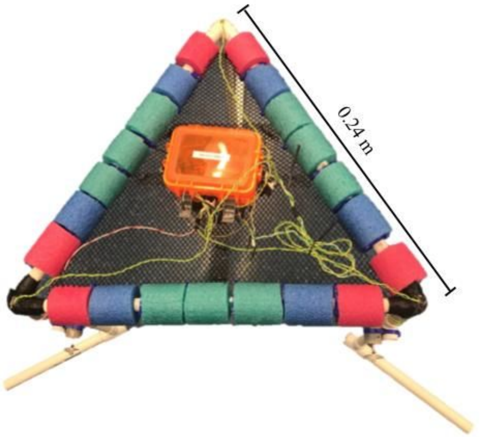


Figure 2: Top view of surface robot.

Each microphone was threaded through the legs of the AUV to a depth of 0.6 m. At this depth, PVC pipes with a diameter of ¾ inches extended outwards from the robot to make the microphones directional. These baffles had to be at least a wavelength long, which can be found as $\lambda = v / f = 1600 \text{ m/s} / 10 \text{ kHz} = 0.16 \text{ m}$.

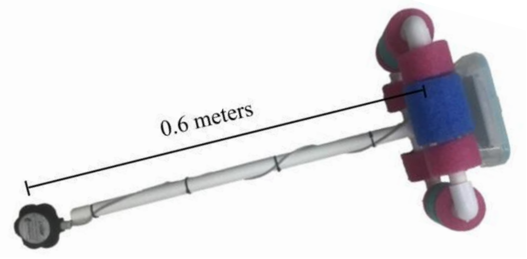


Figure 3: Side view of beacon.

The beacon was also designed to float at the surface of the water to hold the speaker at a constant depth of 0.6 m. The beacon's frame was built using PVC pipes, existing PVC joints, and pool floaties. The waterproof box was positioned above the surface of the water to minimize the chances of leakage. To keep the beacon from sinking, several pool floaties were attached to achieve full buoyancy. A PVC pipe extended into the water, onto which the speaker was attached.

B. Circuit Design

All three of the circuits on the main AUV were powered through a 5V regulator in order to stabilize the battery voltage and be able to use the MCP601 op-amps. The beacon circuit was powered using two 9V batteries in series.

I. Thermistor:

The thermistor acted as impedance between the input and output voltage in a voltage divider with a 100 kΩ resistor, so the voltage output and resistance were inversely related. The thermistor chosen for this circuit was a Vishay NTCLE100E3104JB0, which was used in E80 Lab 5. This thermistor has a negative temperature coefficient that decreases in resistance as outside temperature increases. At 25°C, the thermistor has a value of 100 kΩ. Given the temperatures expected at Dana Point, the voltage output was designed to range from 2-2.3 V, which is sufficient resolution and will not overload the analog Teensy inputs [5].

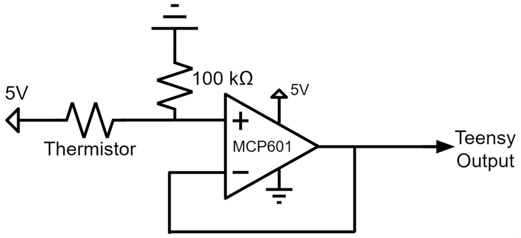


Figure 4: Thermistor voltage divider circuit schematic with a unity gain buffer.

II. Turbidity Sensor:

The Gravity Analog Turbidity Sensor is designed for easy integration with the Arduino platform. It uses a basic amplifying circuit, which allows for analog and digital outputs. This circuit was broken during early testing, so it was recreated from the manufacturer's schematics, and is shown in Figure 5.

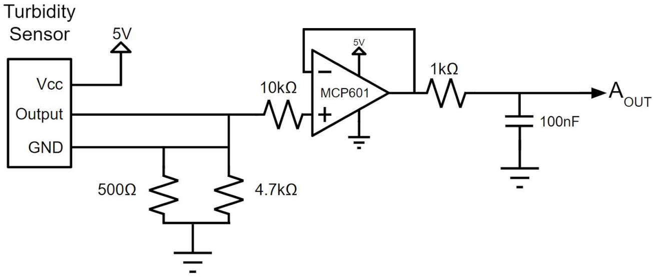


Figure 5: Recreated turbidity sensor circuit schematic, according to manufacturer's specifications.

III. Beacon:

The beacon circuit was composed of a LM555 Timer and an LM384 5W Audio Power Amplifier. The LM384 was used in E80 Lab 4 to power the speaker, so the same circuit was reused. An Astable 555 Timer Oscillator was used to generate a constant 10 kHz square wave that was input into the speaker circuit. This frequency was chosen because most humans cannot hear frequencies exceeding 20 kHz, which would make testing difficult, and lower frequencies could be mistaken for background noise. The following equation determines the oscillator circuit frequency:

$$f = \frac{1}{T} = \frac{1.44}{(R_1 + 2R_2) \cdot C}$$

To generate the 10kHz frequency, R_1 = 1 kΩ, R_2 = 220 Ω and C = .1 μF. The values of the resistors affected the duty cycle, but the duty cycle did not affect the microphones' ability to pick up the signal. Nonetheless, it is calculated according to the following equation:

$$\text{Duty Cycle} = \frac{R_1+R_2}{(R_1+2R_2)} = 84.7\%$$

Since the beacon circuit was separate from the main robot, it was powered with a separate power source. The LM555 has a voltage range of 0-18V and the LM348 has a voltage range of 12-26V, allowing for an overall safe operating range of 12-18V. In order to achieve this voltage, two 9V batteries were placed in series to reach 18V [13][14][18].

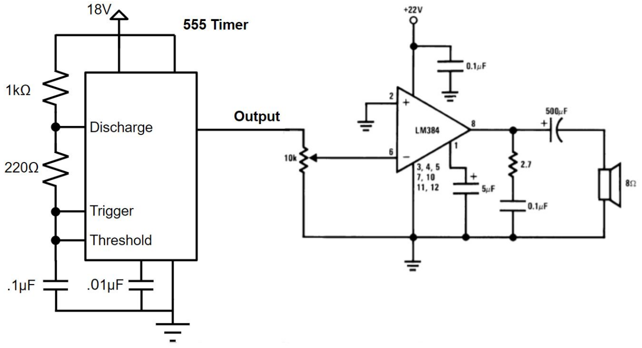


Figure 6: Beacon circuit schematic, powered by the 9V batteries in series.

IV. Microphones:

In order to determine the proximity of the robot to the speaker, the signal from the speaker had to be filtered, amplified, and enveloped. First, the signal had to be filtered to eliminate background noise, which could tamper with the microphone's output. Next, the signal had to be amplified because the output of the microphone was on the order of 10 mV. To use the Teensy analog pins, a voltage swing from 0 to 3.3 V was required. Finally, the signal had to be enveloped to convert it from an AC to a DC signal, allowing for a simpler control system, as explained in the Autonomous Control System section. Thus, the microphone circuits consisted of three components: a high pass filter, two cascaded operational amplifiers, and an envelope detector.

First, the high-pass filter attenuated low-frequency signals from the output of the microphone. A high-pass filter was sufficient because high-frequency signals naturally attenuate more rapidly than low-frequency signals in more viscous substances, like water [19]. Thus, the main concern was filtering out frequencies less than 10 kHz, which was accomplished by high-pass filter with a cutoff frequency of about 5 kHz.

After the high-pass filter attenuated undesired signals, two cascaded single-rail op-amps amplified the desired signal. The MCP601 op-amps were in a non-inverting configuration and each one had a gain of 30, resulting in a total gain of 900. The single-rail op-amps were not offset, which meant that the negative component of the signal was cut off. This loss of data was acceptable because it did not hinder the ability of the circuit to detect changes in signal amplitude.

After the op-amps amplified the AC signal, the envelope detector converted it to a DC signal that changed in amplitude as the volume of the 10 kHz acoustic signal changed. The envelope detector was a simple demodulator circuit composed of a diode, capacitor, and resistor. The values of the capacitor and resistor influenced the response time of the envelope detector relative to the signal. To determine appropriate values for resistance and capacitance, different values were tested in an HSPICE simulation.

The filtering, amplifying, and enveloping components of the circuit worked together so that if a microphone detected an increase in the amplitude of the 10 kHz beacon signal, the output of the circuit, which went to a Teensy analog pin, would increase.

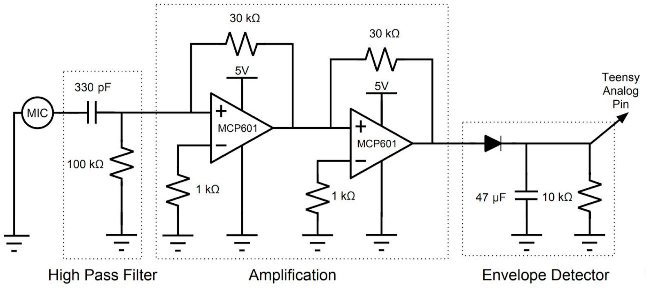


Figure 7: Cascading-amplification microphone circuit schematic.

V. Autonomous Control System

In order to navigate toward the beacon, the robot compared the outputs of the three microphone circuits. The three microphones were designated as the front mic, left mic, and right mic. The two motors were labeled left and right, and corresponded to the left and right microphones, shown in Figure 8. The robot's control system worked to ensure that the point of the triangular frame with the front mic would always point toward the beacon.

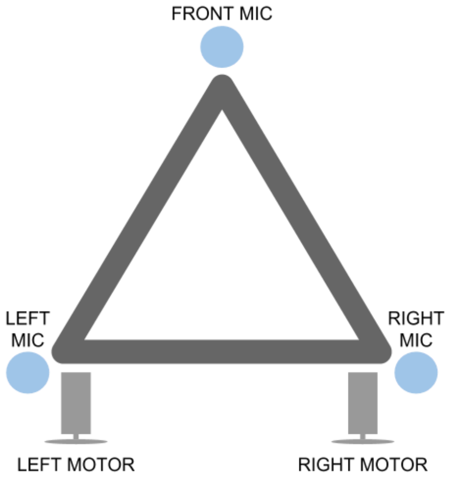


Figure 8: Robot frame with configuration of motors and microphones.

The output of each microphone's circuit went to analog pins on the Teensy. The Teensy code read these microphone output values and used them to determine the amount of power to drive to the motors.

First, the code checked if the output from the front microphone was greater than the outputs of the left and right microphones. If the value from the front microphone was the smallest, just one motor would turn on, causing the robot to rotate until the front microphone was again receiving the most power. If the front microphone was receiving the most power, then both motors turned on, according to the following equations:

$$\textit{left motor power} = \frac{255 \times \textit{right mic voltage}}{\textit{right mic voltage} + \textit{left mic voltage}}$$

$$\textit{right motor power} = \frac{255 \times \textit{left mic voltage}}{\textit{right mic voltage} + \textit{left mic voltage}}$$

For example, if the right mic received more power than the left mic, indicating that the right mic was closer to the beacon, then the left motor would turn on more than the right motor, causing the robot to move forward and to the right. The factor of

$$\frac{255}{\textit{right mic voltage} + \textit{left mic voltage}}$$

in the equation serves to normalize the microphone voltages, where 255 is the maximum motor power. This normalization ensures that the AUV will behave similarly whether it is close to or far from the robot. The absolute value of the microphone voltages does not influence navigation; the robot navigates by comparing the different microphone voltages.

VI. Modeling

A. Sensor Calibrations

I. Thermistor:

The Teensy reads the voltage across the thermistor rather than the temperature directly. Therefore, a calibration curve was necessary to find the relationship between temperature and voltage. The calibration was created by recording the voltage across the thermistor when immersed in water baths of known temperatures. Specifically, the temperatures of the baths ranged from 10ºC to 30ºC to account for all temperatures expected during the deployment. The average surface temperature of the water at Dana Point in April ranges from 14ºC to 18°C, while the average temperature outside of the water ranges from 15ºC to 25°C. The calibration curve was linear as expected, and matched the curve shown on the data sheet [15]. The resulting relationship and equation relating temperature and voltage are shown in Figure 9.

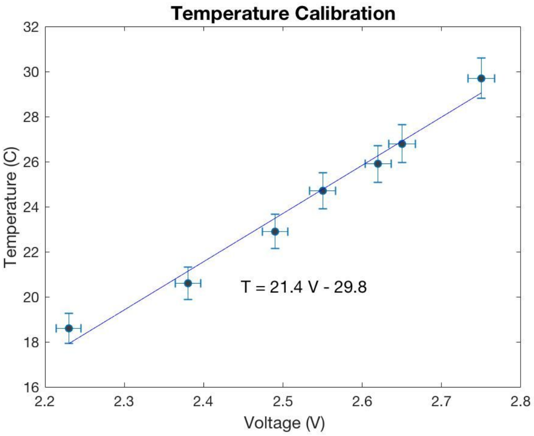


Figure 9: Thermistor calibration curve.

II. Turbidity Sensor:

Because the turbidity sensor circuit outputs a voltage, a calibration curve comparing voltage to turbidity was necessary. The standard turbidity unit is Nephelometric Turbidity Units (NTU). The experimental procedure for this calibration was the same as in E80 Lab 3, where a set of solutions of known turbidities were created using evaporated milk and water and were measured using a TU-2016 calibrated laboratory turbidity meter. The calibration trendline was second order, which matched the trendline found in the data sheet, verifying its accuracy [6].

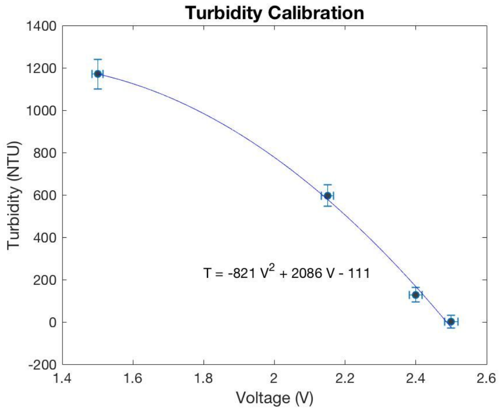


Figure 10: Turbidity sensor calibration curve.

B. *Predicted Performance*

The expected temperature at noon on April 21st, the time of the deployment, was 22 ºC outside of the water and 16ºC at the surface of the water [15][17]. Although some variance was expected, the temperatures measured by the thermistor were expected to stay within a range of ±3ºC from these values.

Expected turbidity was more difficult to quantify, but the turbidity of the water in the ocean might range from 1 NTU to 500 NTU, which falls within the values on the calibration curve in Figure 10.

Finally, the voltages picked up by each microphone were expected to range from 0 V to 3.5 V, as shown in Figure 11.

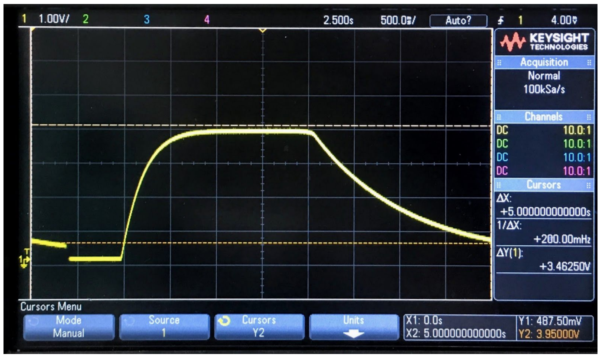


Figure 11: Successful microphone circuit response curve.

The increase in voltage over time as recorded by the oscilloscope follows a first order response. Figure 11 illustrates the expected behavior of the output of the microphone circuit as the signal from the speaker increases in amplitude and subsequently decreases.

VII. BFS Deployment

The Bernard Field Station deployment aimed to check the directionality of the microphones as well as to find a reasonable distance range between the robot and the beacon. A ½ inch PVC pipe was initially used as a baffle for the microphones, but the range was too limited. Because the directionality is dependent on the diameter of the pipe, a ¾ inch PVC pipe replaced the original pipe to test for a wider range. The microphone then picked up the signal from the speaker within the ideal conical area, with an aperture of about 130º. With this PVC sound baffle, the microphone could distinguish the signal within a distance of about two meters. The directionality was tested experimentally, and the microphone circuit output a voltage of about 0.8 V at the greatest when pointed directly at the speaker. When pointing 90º away, the circuit did not output more than 0.3 V. The range was also tested experimentally by increasing the distance between the microphone and speaker until the voltage neared zero. This distance was determined to be approximately 2 meters.

Due to a lack of Teensy functionality at this time, data could not be saved. Nonetheless, the microphone output values were skewed, as measured by an oscilloscope and multimeter, due to interference by a large 20 kHz signal. This was likely caused by the nearby generator that overshadowed the 10 kHz signal emitted by the speaker on the beacon.

VIII. Dana Point Deployment

A. *Plan*

There were two scheduled deployments off the dock of Dana Point. The purpose of the first deployment was to check for basic functionality of the robot in a new environment. More specifically, the data would be analyzed to check if the robot recorded data with the Teensy, output appropriate values for the thermistor and turbidity sensor, and picked up sound through the microphones. The collected data could then be compared with expected values from modeling predictions to check for functionality.

At the second deployment, the beacon would be placed at a reasonable distance away from the dock and the robot would ideally move towards the beacon while collecting data. The beacon distance would begin at 2 meters, as previously tested, and increase until unable to function. The objective of this deployment was to gather temperature and turbidity data that could describe the environment of a fish by the dock.

B. *Preparation*

All sensors and motors were connected to the motherboard and PCB using labeled jumper cables. The batteries were then connected and the Arduino code, as given in Appendix 1, was uploaded onto the Teensy.

In order to reduce time of repairs in the event of a component breaking, an emergency repair kit was prepared. This kit contained components that were known to cause issues in the past, specifically the MCP601 op-amp, 5V regulator, and LM555 timer. Spare sets of male-female and female-female adapters were stored here, since they were used to connect the motors and sensors to the motherboard. All of the resistors and capacitors used in the protoboard circuit were kept in the kit as well, in the event of a larger repair being needed.

C. *Troubleshooting*

I. *Teensy:* Before the first deployment, the Teensy Arduino stopped functioning, likely due to excess voltage applied to an analogue pin. The Teensy has four 5 V-tolerant pins: A0, A1, A2, and A3. These pins can tolerate up to 5 V without breaking the Teensy, but can only read voltages from 0V to 3.3 V. The AUV needed five pins: three for the three microphones, one for the thermistor, and one for the turbidity sensor. So, in addition to the 5 V-tolerant pins, pin A11, which is 3.3 V-tolerant, was used for one of the microphones. When the microphone circuit behaved defectively, it output 4.5 V to pin A11, breaking the Teensy. This problem was resolved by wiring the thermistor output to pin A11 instead, because the thermistor circuit was more consistent and never output more than 3.3 V.

II. *Thermistor:* After the first deployment, the output of the thermistor circuit began to output more than 3.3 V. Immediately after taking the AUV out of the water, the resistance of the thermistor was measured at 10 kΩ, far below the 100 kΩ expected at 25°C. A new thermistor was installed onto the robot to attempt to fix this. An in-depth explanation as to what might have occurred with the first thermistor is found in the Improvements section.

III. *Turbidity:* Initially, the turbidity sensor functioned properly, giving values that were realistic for the environment. By the second deployment, the sensor output a voltage higher than 5 V, which maxed out the Teensy pin. A value that high would suggest a turbidity less than zero, which is unrealistic. Most likely, the turbidity sensor broke due to insufficient waterproofing.

IX. Results and Analysis

A. *Temperature*

Outside of the water, the calibrated output of the thermistor circuit generated temperatures consistent with the air temperature in Dana Point that day, about 22°C [15]. However, when the AUV was placed in the water, there was a large drop in voltage, which, according to the calibration, corresponded with sub-zero temperatures. These data were obviously wrong, as the water was not at all icy. This phenomenon is most evident in temperature data from the first deployment. In Figure 12, the temperature rapidly drops to unrealistically low levels when the AUV is placed in the water, and rises again to the air temperature as the AUV is taken out of the water.

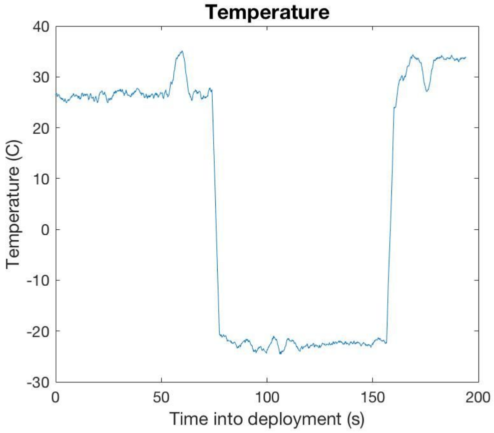


Figure 12: First deployment temperature data.

In a laboratory setting, thermistor voltages and temperatures were consistent with the calibration in Figure 9, even when the thermistor was moved from air to water and back again. These tests were conducted in the lab with both tap water and salt water. Furthermore, the same exact thermistor was used inside and outside of the lab, and functioned as expected inside of lab while displaying peculiar behavior outside of lab. Since the thermistor functioned as expected in the lab, it was probably not defective.

The biggest difference between thermistor use at Dana Point and in the lab was the time spent in the water. In the lab, the thermistor was kept in the water for a couple seconds, just long enough to record a reading on the multimeter. At Dana Point, on the other hand, the thermistor was kept in the water for several minutes at a time, as evidenced by Figures 13 and 14. It is possible that the thermistor wires were not adequately waterproofed, and that this only became an issue after extended time in the water. If this were the case, the temperature values in the beginning of the deployment (at about 60 seconds) could indicate the true temperature of the water, about 19°C.

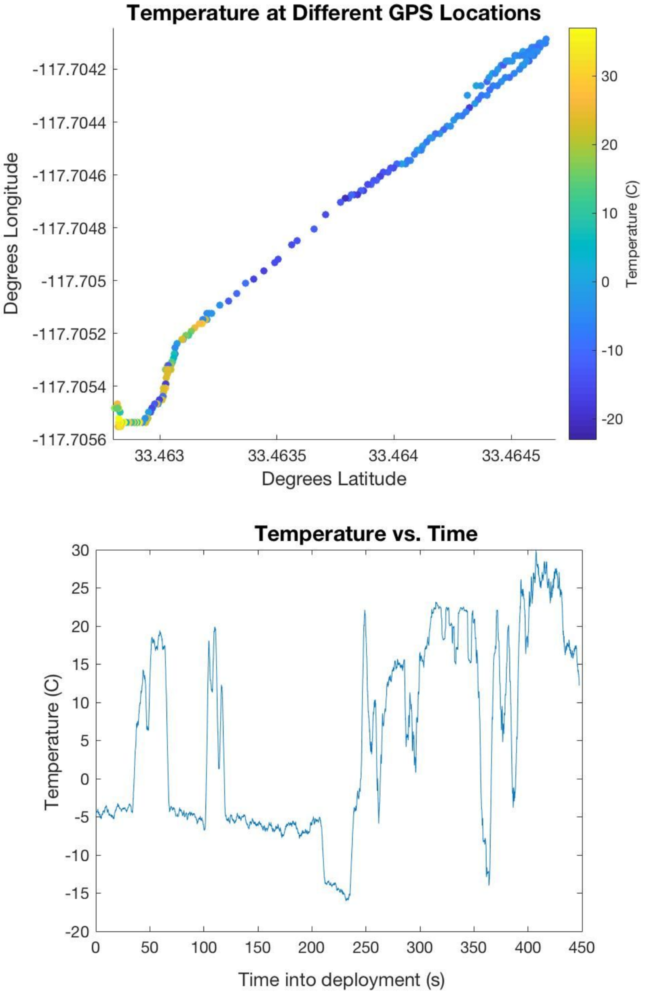
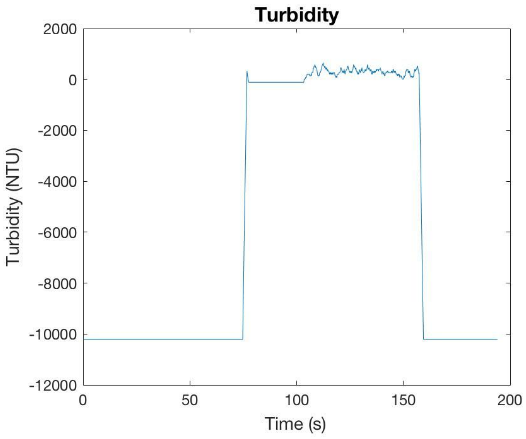
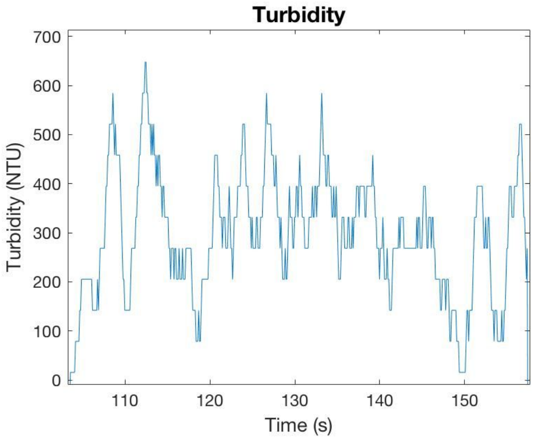


Figure 13-14: Temperature vs. GPS location (top), and the same temperature data vs. time into deployment (bottom). The data in both figures comes from the third deployment.

B. Turbidity

The turbidity sensor did not work outside of the water, which is reflected in Figure 15. Before the AUV was placed in the water, the turbidity was, according to the calibration, an unrealistic value of -10,000 NTU. When the AUV was placed in the water, between 75 and 160 seconds, turbidity values jumped to more reasonable levels (Figure 16).





Figures 15-16: Turbidity data in air and in water (top) and just in water (bottom).

Figure 16 shows that once the AUV was in the water, turbidity mostly fluctuated between 200 NTU and 600 NTU. These are reasonable turbidity values; water samples analyzed in the lab varied from 0 NTU to 1200 NTU. Figure 16 suggests that turbidity varied greatly with from location to location, which is possible if the number of particles and disrupted sediment varied from one location to another. Additionally, strong currents can make water more turbid, so this variability could be explained by the presence of currents in certain areas.

Ideally, turbidity would be plotted with GPS location to show how turbidity changes with position. However, in the first deployment, the GPS data did not log correctly. In subsequent deployments, the turbidity sensor stopped working, most likely due to a waterproofing issue; parts of the turbidity sensor are sensitive to moisture, and a slight leak could have broken it. Thus, with the data logged at Dana Point, it is not possible to plot turbidity with GPS location.

C. Acoustics

In the lab before the Dana Point deployment, the microphones worked inconsistently. Different microphone circuits started and stopped functioning at different times without any changes to the circuits; occasionally they would all work at the same time.

At Dana Point, two of the microphones, the left and right microphones, functioned, as shown in Figure 17. The front microphone circuit output a constant 0 V. With just two microphones working, the control did not fully work; the Teensy detected that the front microphone circuit was always outputting a lower voltage than the left and right microphones, which triggered just one motor to turn on so that the AUV would turn around.

The data in Figure 17 show the left and right microphone voltages when the AUV was in the water. Evidently, the left microphone had a 1V offset, as its voltage was at 1V even when the AUV was far from the beacon and the left microphone should have had a voltage of 0V. When the beacon was brought close to the AUV at about 250s the microphones appeared to work as expected. The spikes in right mic voltage indicate that the beacon was brought closest to the right mic, while spikes in left mic voltage indicate that the beacon was brought closest to the left mic.

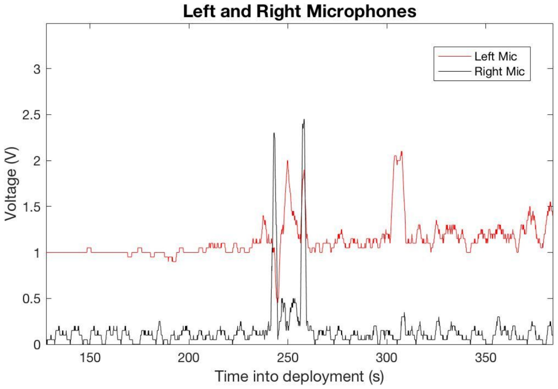


Figure 17: Voltages of left and right microphone circuits in third deployment.

X. Improvements

All of the AUV systems saw major issues, from the navigation system to the data collection system.

The initial plan for constructing the frame for the robot involved simply connecting the PVC pipes using the 3D printed joints. However, due to inaccuracies in the printing and the weakness of the material, the joints could not handle the stress of being pressed into the PVC pipes and broke a day before the deployment. Under the time constraints, plastic reinforcements made from PVC were used as a temporary fix. They were attached to the existing joints using electrical tape and duct tape, and hot glue was used to waterproof the tape edges. Although this solution maintained the structure of the robot through the deployment, a more permanent solution involves using a stronger plastic, such as ABS, to 3D print the joints.

The microphone circuits were inconsistent both in the lab and at Dana Point. The root of this problem was not determined within the limited time of the project. All three microphone circuits sporadically started and stopped working, often separately. It was most likely not a problem with connectivity or op-amps, as these problems were routinely checked. The connectivity was verified using the multimeter and the op-amps were still functional in an external circuit. When the microphone circuits did not work, their response to the speaker was a flat line with many narrow spikes, which increased in magnitude as the speaker volume increased. These spikes were frequent, but did not appear to have a consistent frequency.

One possible problem with the microphone circuit was the envelope detector. The input to the detector should first be filtered with a band-pass around the desired signal to prevent the demodulation of multiple signals [16]. The microphone circuit had a high-pass filter rather than a band-pass filter. Because the signal was a square wave, it contained frequencies higher than 10 kHz. It is possible that the envelope detector was attempting to demodulate these many high frequency signals, in addition to extraneous noise, which may have caused some errors. However, this explanation does not directly explain why microphone circuits would start and stop working.

The thermistors showed a negative temperature from the deployment at Dana Point. The calibration curve created in class was linear, as expected, so it was not a likely source of error. Upon further inspection of the thermistor, it became clear that the heat shrink insulation was not as waterproof as previously thought, and some of the metal wire was exposed and slightly corroded.

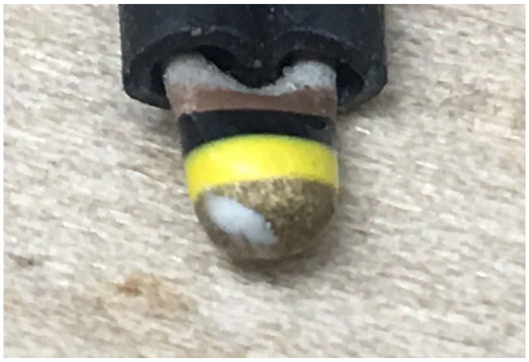
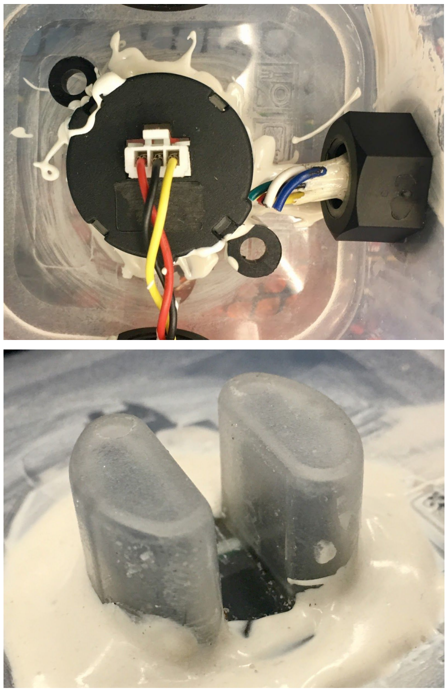


Figure 18: Thermistor with exposed metal wire near heat shrink.

This suggested that the unexpected temperatures may be related to saltwater exposure at Dana Point. When the thermistor was placed into a freshwater solution at about 20°C, the thermistor accurately measured about 110 kΩ. Per contra, when the thermistor was placed into a saltwater solution (of 35 parts per thousand or 35 grams per kilogram, like that of ocean water), the resistance slowly dropped until it appeared to stabilize at about 50 kΩ [20]. This suggested that the issue with the thermistor was related to saltwater exposure.

However, according to the thermistor data sheet, a resistance of 50 kΩ corresponds to an outside temperature of about 40°C [5]. This is the opposite of the data collected, where the Teensy measured a voltage of about 0.2 V, corresponding to a temperature of -20°C in the ocean. According to the voltage divider equation, a decrease in resistance, as measured in lab, would result in an increase in voltage, which is not what was measured at Dana Point. This suggests it may be an issue related to the higher conductivity of saltwater compared to that of freshwater. Unfortunately this could not be explored within the time frame of this project, but could offer potential for future exploration.

The turbidity sensor first gave reasonable values, but began to output more than 3.3V after the first deployment. This was most likely an issue of waterproofing. Due to these high voltage outputs (the Teensy peaks at 3.3V), the data could not be analyzed once the turbidity sensor started to malfunction. Moreover, the sunlight may have interfered with calibration curves, since they were created indoors. Better waterproofing of this sensor, as well as testing in conditions similar to the deployment conditions would lead to more reliable data collection.



Figures 19-20: Inside and outside of turbidity sensor box, showing the waterproofing using marine epoxy.

XI. Conclusion

The objective of this project was to develop an autonomous underwater vehicle that could autonomously navigate while collecting relevant data at Dana Point. The process of constructing and testing various sensors emphasized the importance of verifying the functionality of each subcircuit at every step. Initially the design seemed straightforward, but the sensors acted erratically and their behavior at Dana Point could not be comprehensively explained.

Behavior of each sensor at the deployment was modeled and predicted beforehand, but the data collected during the experiment largely contradicted expectations. In an unfamiliar environment, new problems and challenges are anticipated. Much of the learning process involved debugging issues on site and figuring out the source of errors within a narrow time frame.

In the future, it would be important to repeatedly test the sensors and circuits in environments similar to those found in Dana Point. Creating calibration curves for the directionality and range of the microphone would also help solve several issues with the microphones.

XII. Acknowledgements

The team would like to thank the Harvey Mudd Engineering professors, especially E80 Section 4 professors Christopher Clark and Matthew Spencer for all their help both inside and outside of lab. The team would also like to thank Sam Abdelmuati for all his help in the stockroom.

In addition, the team would like to thank Richard Zhang's team for helping out with the turbidity circuit.

References

[1] "Project: Field Deployment of an Autonomous Robot." *E80 Experimental Engineering*, Harvey Mudd College.

[2] "Turbidity: Description, Impact on Water Quality, Sources, Measures." Minnesota Pollution Control Agency, Mar. 2008.

[3] Trebitz, Anett S., et al. "Turbidity Tolerances of Great Lakes Coastal Wetland Fishes." *North American Journal of Fisheries Management*, vol. 27, no. 2, 2007, pp. 619–633., doi:10.1577/m05-219.1.

[4] "Lab 5 - Temperature Sensors." *Vicki Moran and Anant Kandoi*, Harvey Mudd College.

[5] "NTCLE100E3 Data Sheet." *Vishay*, Vishay BCcomponents.

[6] "Turbidity Sensor SKU: SEN0189." *DFRobot Wiki*, DFRobot, 25 May 2017.

[7] "DAEX25W-8 Audio Exciter Spec Sheet." *Dayton Audio*. 2014.

[8] "Lab 4: Underwater Acoustics and Fourier Analysis." *E80 Experimental Engineering*, Harvey Mudd College.

[9] "CME-1538-100LB." *DigiKey*.

[10] "Lab 3: Operational Amplifiers and Marine Sensors." *E80 Experimental Engineering*, Harvey Mudd College.

[11] "Turbidity sensor SKU: SEN0189." *DF Robot.* 2017.

[12] Rowe, D. K. "Lethal Turbidity Levels for Common Fish and Invertebrates in Auckland Streams." *Auckland Regional Council*, 2002.

[13] "555 Timer Astable Circuit." *All About Circuits,* Web.

[14] "LM555 Timer" *Texas Instruments,* Web.

[15] "Dana Point, CA Monthly Weather Forecast." *The Weather Channel*, 1 May 2018. *Surf Forecast Surf Report*.

[16] "Envelope Detector." *Wikipedia.* 2012.

[17] "Dana Point Sea Temperature." *World Sea Temperatures*, 21 April 2018.

[18] "LM384 5W Audio Power Amplifier" *Texas Instruments*, Web.

[19] M. Scutaru, "High Frequency Signal Attenuation Through Materials", *Semantics Scholar*, Web.

[20] "Variations in a Salty Ocean." *Exploring Earth*, Web.

APPENDIX

Appendix 1: Arduino Motor Control Code

```
// Control.h

#ifndef __Control_H__
#define __Control_H__

#include <Arduino.h>
#include "MotorDriver.h"
#include "StateEstimator.h"
extern MotorDriver motorDriver;

class Control {
public:

 Control(void);

 void init();

 // sets the motor speeds using BangBang-Control
 void calculateControl();

 int lastExecutionTime = -1;
 bool stopMoving = false;

 String printMicVals(void);

 String printString(void);

 void setMicVals(int mL, int mR, int mF);
 double uL;              // left motor effort
 double uR;              // right motor effort

private:
 int micL;
 int micR;
 int micF;

 void calcMotorVals();
};

#endif
```

```
// Control.cpp

#include "Control.h"
#include "Printer.h"
extern Printer printer;

Control::Control(void) {
}

void Control::init() {
}

void Control::setMicVals(int mL, int mR, int mF) {
 micL = mL;
 micR = mR;
 micF = mF;
}

void Control::calculateControl() {
 calcMotorVals();
 if (micF > 1000) {
   stopMoving = true;
   uL = 0;
   uR = 0;
 }
}

void Control::calcMotorVals() {
 // if the front mic is getting a weaker signal than left
    and right mics, turn around
 if (micR > micF && micL > micF) {
   // right mic is getting a stronger signal, so turn on
      left motor
   if (micR > micL) {
     uL = 200.0;
     uR = 0.0;
   }
   // left mic is getting a stronger signal, so turn on
       right motor
   else {
     uR = 200.0;
     uL = 0.0;
   }
 }
 // if neither left or right is getting a signal, go
    straight
 else if (micR == 0.0 && micL == 0.0) {
   uR = 200.0;
   uL = 200.0;
 }
 // if the front mic is getting a stronger signal than left
    and right, go straight
 else {
   double total = (double) micR + (double) micL;
   double conversion = 255.0/total;
   uL = micR * conversion;
   uR = micL * conversion;
 }
}

String Control::printMicVals(void) {
 String printString = "Mics: Left Mic: " + String(micL)
   + ", Right Mic: " + String(micR) + ", Front Mic: "
   + String(micF);

 return printString;
}

// prints out how fast each motor is spinning
String Control::printString(void) {
 String printString = "Control: Left Motor: " + String(uL)
   + ", Right Motor: " + String(uR);

 return printString;
}
```

Appendix 2: Parts List with Costs

Part	**Cost (USD)**
Analog Turbidity Sensor For Arduino	10.20
Dayton DAEX25W-8 Audio Exciter	26.83
CME 102-2190-ND Microphones, 3x	6.32
9V Batteries, 4x	9.38
Foam Insulation	2.43
1/2"-3/4" Tee Joint Adapter, 3x	2.94
1/2" PVC Slip/Thread Adapter, 3x	1.14
1/2" Sch. 40 PVC, 10 Feet	2.09
3/4" Sch. 40 PVC, 10 Feet	2.57
Zip Ties	0
3D Printed PVC Joints, 3x	0
Pool Noodles	0
Plastic Mesh	0
1/2" PVC	0
Tupperware Containers, 2x	0
Outdoor Products Large Watertight Box	0
Total (with tax)	70.29