ECE-498: Electrical and Computer Engineering Capstone Design Project II

The AFμS (Autonomous Flocking μ-Sub) Project

Technical Design Report

Written by Jacob Karaul and Xavier Quinn

Advisors: Walter Dixon and Professor Hedrick

November 27, 2019

Table of Contents

List of Tables and Figures

Tables

Figures

[1] J. Sticklus, P. A. Hoeher and R. Röttgers, "Optical Underwater Communication: The Potential of Using Converted Green LEDs in Coastal Waters," in IEEE Journal of Oceanic Engineering, vol. 44, no. 2, pp. 535-547, April 2019.
[2] Ainslie M. A., McColm J. G., "A simplified formula for viscous and chemical absorption in sea water", Journal of the Acoustical Society of America, **103**(3), 1671-1672, 1998.

Introduction

Underwater exploration is a field of study that involves investigating chemical and physical characteristics of large water bodies (typically oceans) and the organisms that reside in them. In total, humanity has explored about 8% of the world's water bodies, and only 5% of ocean basins[3]. Some areas of study in this field are oil spills, habitation patterns for aquatic species, residual radiation and other contaminants, turbidity levels, and general 3D mapping, all of which have significant health or monetary implications. A need for accurate underwater data can be found in the Great Lakes, a source of 21% of the world's freshwater supply[4], which suffers from harmful algal blooms (HAB's) in various locations that render the water undrinkable. More locally, HAB's are also present in Moreau Lake State Park in Saratoga

[3] https://oceanexplorer.noaa.gov/backmatter/whatisexploration.html
[4] https://coast.noaa.gov/states/fast-facts/great-lakes.html

County at such a debilitating quantity that several pets have died and parts of the state park have been shut down[5]. Underwater data collection would be useful in these instances to detect oxygen and toxicity levels where HAB's are present in order to characterize and document them.

Systems that can generate this data are vast; there are large research vessels outfitted with research teams and sensing equipment, stationary underwater sensor networks, and Remotely Operated Vehicles, or ROV's, that are controlled on the surface (just to name a few). All of these systems are limited by the fact that they require active governance by humans, therefore making these technologies for underwater data collection quite slow. This limitation spurred the emergence of Autonomous Underwater Vehicles, or AUV's, which are capable of performing some decision-making during their journeys. These vehicles can be "dropped" into a water body with mission parameters and, depending on the system, may run for days or weeks at a time with no human contact.

The goal for our project is to develop an autonomous underwater vehicle with reasonable accuracy at a much lower price point than is currently possible by leveraging recent emergences in low-cost Single Board Computers (SBC's), Inertial Measurement Units (IMU's), and high performance motors/motor drivers to do so. A single system will be affordable, but will still be accurate enough to be useful for a range of uses. This team also includes two mechanical engineers: Alexander Pradhan and Samuel Veith, who will be designing the hull, thrusters, and renewable charging solution for their senior capstone project. They will also construct and maintain all test systems. Their specific contributions are mentioned below when relevant.

Even with a team of four students, this project is a large endeavor to say the least. Two of the most limited resources are time and money, so it imperative that both be spent on worthy

[5] https://www.timesunion.com/7dayarchive/article/Moreau-State-Park-reports-algae-bloom-14341929.php

causes. Therefore, we will be purchasing off-the-shelf components whenever possible and focusing our efforts on implementing systems that require finer control and precision, and are large blocks in the scope of this project. Any non-mandatory systems that take too much time/money to implement will be either stashed away for future work or eliminated entirely. Finally, a clear schedule will provide deadlines from both computer/electrical and mechanical engineering teams to ensure that work continues in parallel. See *Implementation Schedule* for specifics.

Furthermore, with these constrictions, it is not feasible to design and build an AUV that can boast better performance than any of the systems already in existence. The *Design Requirements* section details the specifications that we believe can be accomplished, and are still respectable enough for this product to be useful.

This paper details the background of the field, overall design, testing plan, research performed, alternatives considered, schedule, and preliminary testing results for the Autonomous Flocking μ-Sub project.

Background

Though the development of AUVs was not possible until computation units were sufficiently advanced for real-time autonomous navigation, a wide range of AUVs have come out since this point. Based on the necessary size of of these computation units since their capabilities reached those necessary for autonomy, the initial generations were extremely large scale. Only in recent years have some AUVs shrunk below the size of human carrying submersibles. Currently, there exists a large variation in the size of available aquatic autonomous

vessels based on their desired functionality. The need for highly precise sensor data, locomotion, or long term deployment all result in a bulkier product, and there has even been an increased interest in highly use specific[6] AUV designs with their own physical requirements[7].

As developments in technology have allowed them to do so, an increase in micro-scale AUVs has occurred. These systems are roughly defined as being in the ~1 meter or less range, and inevitably have less precise sensors and reduced capabilities from their full scale counterparts. Examples of such AUVs are the Hydroid REMUS M3V[8], the Hydromea Exray[9] and the Riptide μUUV[10]. All three of these are small and light enough to be hand deployed and retrieved, making them drastically more useful to anyone who does not have access to hoist equipment. All of the companies that manufacture AUVs have attempted to maximize the precision and capability of their products; all manufacturers also maintain a price point high enough to limit who can reasonably afford to use their services.

The majority of the microscale AUVs on the market have been designed for very specific uses, and therefore contain a sensor suite perfectly suited to said task. An example of this is the Hydromea Exray, which is designed with the specific task of exploring and measuring the wall thickness of flooded confined spaces, such as shipping vessel ballast tanks. This means that this particular AUV does not have much use outside of this task, regardless of how well suited the rest of the system is for another application. The Riptide μUUV has the option for a sensor payload bay to be attached in the middle of the AUV that allows for water to flow through it.

[6] https://www.hakaimagazine.com/news/rangerbot-programmed-to-kill/
[7] https://spectrum.ieee.org/robotics/humanoids/meet-aquanaut-the-underwater-transformer
[8] https://www.hydroid.com/REMUS-M3V
[9] https://www.hydromea.com/exray-wireless-underwater-drone/
[10] https://riptideas.com/micro-uuv/

This is to allow the user to design and build their own completely seperate sensor collection system and place it inside the bay to record data.

As of now there is no AUV that fills the niche of a highly affordable system, with or without the drawbacks expected from less expensive equipment. As not all aquatic data collection needs high precision location or external sensor information, a system affordable enough for individual or university level research would greatly increase the amount of aquatic based monitoring and analysis, which is needed with today's climate conditions now more than ever.

While it may be desirable to make an AUV system as cheap as possible, there are some aspects which cannot be compromised when it comes to leaving something in a water source for extended periods of time. No corners can be cut when it comes to avoiding environmental impact, whether it be from materials leaching chemicals, the equipment interfering with local wildlife, or the entire system breaking under expected conditions and littering its parts into the water.

Design Requirements

This system is being designed with the aim of its use within a flock. While aspects of its functionality are chosen specifically to allow for this behavior, the specifications in the scope of this design report represent our expectations of not having a final product with full flocking capabilities within the timeline of this project. However, functionality is included that is not required for a solo system, and ideally the end result within this project scope is a functional proof of concept for an AUV with flocking capabilities.

In an attempt to ensure that our project is applicable to the sector it is being designed for, we held an ideation session at the start of the design process. This involved collecting and collating use cases for an AUV, then sorting them as a function of importance to the consumer and how widespread the specific use case is. These results were then analyzed, condensed, and repeatedly sorted through until a set of design parameters were generated with the goal of maximizing usefulness to the prospective customer. Further information on this process is available in Appendix A. The design requirements detailed in this report only cover the aspects that the computer and electrical team are directly involved in the design of.

The most important requirement of this system is that any environmental impact is mitigated. This means that the AUV must exhibit robustness and durability to avoid breakage and therefore contamination of the local environment. Most of the requirements to fulfill this need falls on the side of the mechanical engineering team, though ensuring that the AUV has the capability to keep itself out of dangerous situations is an interdepartmental effort. To be able to do this, the foremost requirement is that the AUV is able to translate itself in 3D space, which is a boolean requirement; it either can or cannot. A highly related requirement is the ability to translocate to a designated set of coordinates. The selected range for this to be considered successful is within 5 meters of the point. While this may seem like a wide region of error, it must be considered that this device will be acting on the scale of lakes, oceans and rivers. If the device can translocate to a point, it must also be able to hold its position within said range of that point for at least 5 minutes.

Another important factor for avoiding environmental impact is avoiding collisions between the AVU and other objects, not only to protect the AUV, but to protect whatever it has

the potential to collide with as well. To avoid objects, the AUV must be able to detect them at a minimum range of 1.5m. As the AUV may not be the object that is moving in this situation, it must be able to avoid an oncoming collision at a relative velocity of 1m/s from the time the AUV was detected. To be able to react quickly in situations like these, the AUV must be able to accelerate itself to its maximal velocity at at least 1 m/s^2 . For it to be able to handle and remove itself from currents, the AUV will need to be able to travel at least at 3*m/s.* If an obstacle is detected and avoided during point to point travel, the AUV must continue to the designated point.

For the AUV to be able to provide useful sensor data, it must be able to traverse a designated area, staying within the region, and passing over all parts of the area evenly e.g. not simply containing its exploration to a corner of the zone.

The chosen method of communication is wireless optical transmission/reception. Pertinent specifications for any communication system include Bit Error Rate (BER), minimum transmission distance, and data rate. Additionally, turbidity is included as optical communication capabilities can largely vary with clarity. These factors are all closely related, thus the requirement will include them all in a single scenario: At a turbidity of 10 NTU, the AUV must be able to communicate optically at a distance of 0.5 meters at a data rate of 62.5 bps with a BER of 10% or less.

A value of 10 NTU was found to the average during low-flow periods in rivers and lakes, where this sub will be tested experimentally[11]. Both BER and distance were estimated from a set

[11] https://www.fondriest.com/environmental-measurements/parameters/water-quality/turbidity-total-suspended-solids-water-clarity/

of tests run on an underwater optical communication system in various turbidity conditions[12]. Finally, the data rate was calculated from early tests run with optical components, described in the *Preliminary Testing Results* section, and a communication protocol. While the communication modulation technique is not yet set in stone, a clocking pulse-position modulation (PPM) scheme is being seriously considered and is used for this calculation.

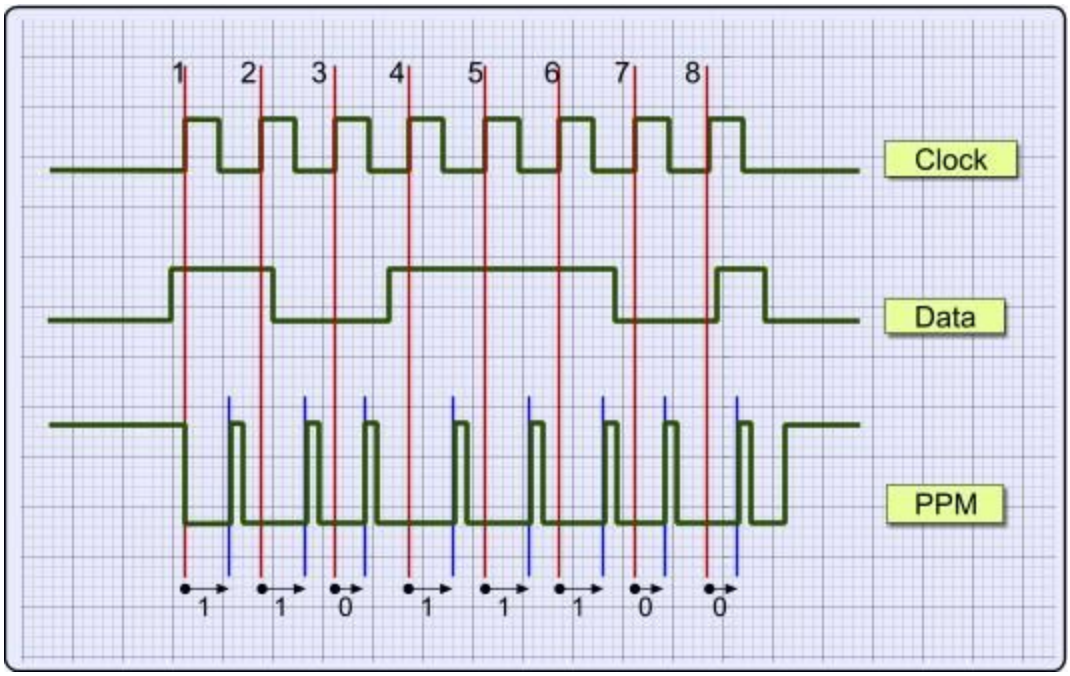


Fig. 1. Timing diagram of the PPM protocol

PPM is an encoding technique that modulates the time since the last clock pulse before transmitting a high pulse as a means of communicating data. For example, in a 15 ms period, a delay since the last clock pulse of 5 ms may indicate a '0' and a delay of 10 ms may indicate a '1'. This method allows for the detection of errors in a noisy channel by utilizing a fixed set of known locations a pulse must be at any given clock cycle. If pulses are not in the correct position

[12]M. E. G. Mital *et al*., "Characterization of underwater optical data transmission parameters under varying conditions of turbidity and water movement," *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, Malacca City, 2017, pp. 1-6.

within a clock cycle or are not the correct pulse width, then an error is detected. Except for the unique case in which an error both erases the original pulse and creates a pulse in another allowable pulse location, errors should be easily detectable.

Fig. 2. Illustration of PPM clocking

This protocol, however, does not provide a method for synchronization. That is, the clock at the receiving end may be out of phase, faster or slower than the clock at the transmitting end and thus the actual time since the last clock pulse is compromised. To account for this, clocking can be implemented by sending a pulse at the beginning of every clock cycle, as illustrated above. Blue pulses represent data, green dashed lines represent the start of clock cycles, and red pulses represent the pulse at the beginning of every clock cycle. The receiving end can use this periodic clocking pulse to establish the beginning and end of every cycle and stay synchronized with the transmitter.

Returning back to data rate, a single clock cycle of PPM consists of four block times for encoding a single bit per cycle: the time for a clock pulse, an intermediate period between pulses, the time for a data pulse indicating a bit of 1, and another intermediate period between pulses. In

the above figure, there is space for two data pulses and thus two bits can be transmitted every clock cycle. The block time used here is 4 ms, the minimum pulse period found for an early prototyping for a white light optical system as described in *Preliminary Testing Results*. Using a minimum pulse period and intermediary period between pulses of 4 ms, the data rate R can be computed as $R = (1\ bit)/(4\ blocks * 4 * 10^{-3}\ seconds) = 62.5\ bps$.

While many of the aspects of this requirement are not necessarily numerically quantifiable, one of the largest design requirements is that the AFμS system is as consumer facing as possible. This means making sure its features suit those that could be needed by the people of whom the specific functionality of the system is useful. These requirements are the addition of the modular sensor bay, ensuring that the final design is ergonomic enough to be easily deployed and retrieved, the ability to passively recharge itself, and that its price point is affordable for individual researchers, with the explicit goal of a sub-$500 product.

Design Alternatives

Communication

Besides waterproofing and pressurization, communication has historically been one of the largest hurdles to overcome when it comes to underwater vehicles. A variety of solutions have been implemented through the years in which active underwater exploration has been in vogue, though each has its own set of advantages and disadvantages.

The simplest of these solutions is running a physical wire between the transmitter and receiver. This has the benefit of being an extremely reliable method for data communication, as there is a provided transmission medium with favorable characteristics. The drawbacks of this system are due to having a physical connection between the transmitter and receiver. Not only does this limit the distance they can be apart, but it also adds significant mass to the system, and measures to ensure that the line does not get caught or tangled need to be taken.

Though radio frequencies are one of the most prominent methods of communication in the 21st century, these signals attenuate extremely rapidly in water as a factor of their frequency. Several AUV systems, such as the Hydromea Exray[13], have implemented radio communication systems. To accommodate the attenuation factor, these AUVs need to utilize high powered transmitters and high quality receivers to get a relatively low range signal (~10m). Further ranges have been achieved by using extremely low frequency signals, though this requires very large antennas and results in a drastically slowed bitrate compared to higher frequencies, as can be inferred via the relationship between frequency, period and wavelength in a known medium.

As sound travels better in water than it does in air, it is one of the more widely used aquatic communication methods, such that the U.S. Navy has a communication standard called JANUS[14]. This method by far has the best range characteristics as a ratio of form factor, though it comes at the cost of extremely low bitrates. The frequencies that do best in water are below 100kHz, and effective range only increases as the frequency goes down. This limitation means that even if the data being transmitted had no encoding, the highest possible data rate in this frequency range is 50kbps. Another factor is that because sound travels so well in water, the

[13] https://www.hydromea.com/exray-wireless-underwater-drone/
[14] http://www.januswiki.com/tiki-index.php?page=About+Janus

background level of noise is higher for this across a wider range of water than any other communication method. This means that it is often the case that complex encoding schemes or purposeful information redundancy is required to account for higher BERs (Bit Error Rates) in this communication channel, slowing the bit rate even further.

Similar to radio, light attenuates quite quickly in water, however, it has the advantages of not requiring hardware anywhere near as large or power intensive as radio transmitters to get a similar range. With the ubiquity of LEDs during this age, this is also a very frugal communication implementation. However, clear line of sight is a lot more important for optics than it is for radio, and impairments such as highly murky water or seaweed can affect light propagation to a much higher degree than radio waves. Similar to acoustics, there is a large amount of background noise in this medium, though the deeper the AUV goes, the less light from the surface there will be. Unlike acoustics, light travels very quickly, and the wavelength is nominally small, so very high bit rates can be achieved, albeit only over small distances.

As affordability is a key design requirement, and as communication is not vital for the AUVs navigation or base functionality, we decided to utilize optics, limiting our required communication range to 1.5m, but allowing for rapid transfer of information.

The “optimal” wavelength for this application should be established. Keeping in mind that the proposed audience for this product will be small research groups, local governments and colleges/universities, it is imperative that its widespread applicability be maintained. While lower wavelengths (~500nm, blue) attenuate less in ocean bodies, slightly higher wavelengths (~600nm) show less attenuation in more turbid conditions found in coastal waters[15], where this

[15] Johnson, L. J., Jasman, F., Green, R. J., & Leeson, M. S. (2014). Recent advances in underwater optical wireless communications. Underwater Technology: International Journal of the Society for Underwater, 32(3), 167–175. doi: 10.3723/ut.32.167

sub will be tested initially, and still boast relatively low attenuation is ocean bodies. The wavelength chosen for this application is 567nm (lime). Hence, range may be reduced but usability in a variety of environmental conditions is preserved.

As with any communication system, transmitting and receiving elements are both needed. For transmission, viable options include laser diodes and high power LEDs. As discussed in the *Design Requirements* section, omni-directional optical emission is desirable, as well as low power draw. Laser diodes provide high switching speeds, however they are sharply limited in range, temperature stability, and require specialized circuitry to compensate for environmental factors[16]. While LEDs cannot switch as fast as laser diodes, they offer greater beam divergence, resilience, and as mentioned before, come at a much lower cost. Therefore LEDs were chosen as the component for transmitting optical signals.

For optical reception, the choice is not so linear. Viable options include photoresistors, phototransistors, and photodiodes. Beginning with the most commonly known component, photoresistors exhibit different resistances over different light levels in a fairly linear relationship. However, photoresistors can take anywhere from a few milliseconds to a few seconds to return back to a dark state after being exposed to light, which is not reasonable for a communication link. Phototransistors are essentially typical bipolar or field-effect transistors with their base/gate exposed to the light source. Therefore they exhibit a current gain and collector-emitter voltage proportional to the light level they receive. Phototransistors offer high robustness in the presence of noise, and can switch at a moderately fast rate (=< 250 kHz). Photodiodes are components that convert light energy into electrical current. These components

[16] Brundage, H. (2010). Designing a wireless underwater optical communication system. Mechanical Engineering - Master's Degree. Retrieved from http://hdl.handle.net/1721.1/57699

are very fast, capable of switching in the MHz region, relatively hardy to ambient noise, and are very affordable. However, as this component produces current, extra circuitry is required to convert that current into a voltage level, such as a transimpedance amplifier, so it can be processed by the SBC.

A phototransistor was chosen for this application. While photodiodes display more desirable traits pertaining to performance, they are also capable of receiving optical signals from similarly powered LEDs over distances of 10 meters or more, which is beyond the requirements for this application. The design, testing and refinement of a transimpedance amplifier may also take a large amount of time to complete, which is not preferable when considering the time constraint of this project, and complicates the hardware requirements. Finally, as the expected data rate as described in the *Design Requirements* section is in the range of Hz as opposed to kHz, the good noise robustness and far simpler circuitry which the phototransistor offers makes it a suitable choice for this application.

Methods of signal modulation include using an ADC and extracting the data through software, and using a comparator circuit. Based on what ADCs exist within our price point, the selection may place another limit on the maximum data rate achievable, equal to half its maximum sampling rate, while delays in a comparator circuit are on the order of nanoseconds and are thus negligible. A comparator circuit will be implemented and tested first, followed by alternatives if necessary.

Power

The energy reserve of this system will physically vary in size according to the available space in the vessel. The battery chemistry is selected based on which chemistry type offers the greatest energy density. Lithium-ion (Li-ion) and Lithium-polymer (Li-Po) batteries lead in this respect, compared to options such as Nickel-Cadmium (Ni-Cd) and lead acid[17]. Due to widespread adoption of portable, rechargeable technological devices over the past several decades, these batteries are also available for a low price point.

The other consideration to incorporate is the number of cells to use in series and parallel. Increasing the number of batteries in a parallel configuration increases the maximum current draw of the array and the overall capacity. Again, this will be limited to the available space in the vessel. The number of cells for the series configuration modifies the efficiency of the buck-boost voltage regulators, and the speed that the motors can spin at as they are powered directly from the battery. Based on early PID tests described in the *Preliminary Testing Results*, a low RPM value is desirable for finer control and the greatest source of potential inefficiency in voltage regulation lies with the component that will be regulated continuously: the SBC. As SBC's are typically powered off of 5 V, the series configuration chosen was 2S, or a nominal voltage of 7.4 V.

[17] https://circuitdigest.com/article/different-types-of-batteries

Processing

For the computational requirements of this system, the final performance has yet to be seen. The ATS Mk. 1 incorporates the Raspberry Pi 3B, and this SBC will be used until the computation requirements exceed its specifications. This SBC has 1GB of RAM and a quad-core processor with clock speeds up to 1.2 GHz. As more real-time systems are developed and integrated, CPU performance tests will be run to ensure delays are not beyond reasonability. Raspberry Pi's have the benefit of low cost, a massive community base, a plethora of third-party software libraries, and backward-compatibility. These features will allow us to implement functionalities with existing libraries rather than creating them ourselves, which is highly useful considering the time constraints of this project. Hence, if the computing requirements of the 3B is not sufficient at greater offered load, the newly minted Raspberry Pi 4B (up to 4GB DDR4 RAM/quad-core processor up to 1.5 GHz)[18] may be a viable option, and if not then an SBC with superior computational characteristics such as the ASUS TinkerBoard S (2GB DDR3 RAM/quad-core processor up to 1.8 GHz/integrated graphics processor up to 650 MHz)[19] will also be considered.

[18]Raspberry Pi 4 Model B specifications – Raspberry Pi. (n.d.). Retrieved from https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/.
[19]Tinker Board S: Single Board Computer. (n.d.). Retrieved from https://www.asus.com/Single-Board-Computer/Tinker-Board-S/.

Sensing

Movement

The most common method of motion detection for autonomy applications is GPS. GPS will provide the devices specific coordinates via satellite triangulation as long as the device is not in a region where there is enough attenuation that communication with the required number of satellites is impossible. Due to this stipulation, GPS does not work particularly well under water, and can in fact only function directly below the surface or else the attenuation factor is too great to provide accurate data.

Another method of motion detection, which has recently started to gain traction and practicality, is called optical flow. This uses a camera to determine velocity based on the rate at which pixels travel across the optical sensor. This method only works if there is a non-monochrome surface for it to be pointed at, and enough light for the sensor to be able to pick up the surface. While this would likely work in very shallow clear water, it would provide little functionality in water deep enough that the optical sensor cannot detect the floor.

A sensor called an IMU, or Inertial Measurement Unit, can be used to get motion data as well. This sensor reads in acceleration and orientation data, which can be utilized to calculate the velocity and displacement of the device. Unfortunately, all but the extremely expensive IMUs[20] have intrinsic noise and error in their readings that accumulate with the calculations of velocity and displacement, making this a very imprecise solution.

[20] https://aerospace.honeywell.com/en/learn/products/sensors/hg1700-inertial-measurement-unit

A method especially suited for underwater use is having a set base station which can be used to reference distance from the station to the device. This usually works by having the base station emit a signal that the AUV can use to calculate its distance and position in relation to the station[21]. This adds the need for a station to be placed in every location the AUV will be utilized in, limiting effective range and use cases significantly.

A sensor called a DVL, or Doppler Velocity Log, is commonly[22] used on human scale vessels, though modern technology has allowed the size of these sensors to decrease over time, making it possible to have them on smaller systems. This sensor uses three or more transducers to generate acoustic waves while under motion, and measures the doppler shift in frequency upon return of the signal. These sensors remain quite large and expensive relative to the other options at the moment.

For this project, we determined to attempt to utilize both and IMU and GPS module. The GPS module will function while the AUV is on the surface, and the once the AUV submerges, it will utilize the less accurate method until it recalibrates on the surface.

The modern GPS modules available on the market in the <$75 price range, such as the SAM-M8Q, MT3339 and NEO-6M/V, all have similar sensitivity (~-162dBm) as well as similar power characteristics. The SAM-M8Q has the ability to connect to multiple constellations, and the Matek SAM-M8Q GPS Module breakout keeps a very low form factor and provides UART connection capabilities. This breakout has been selected to be the one used in our AUV.

---

[21] S. M. Smith and D. Kronen, "Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation," *Oceans '97. MTS/IEEE Conference Proceedings*, Halifax, NS, Canada, 1997, pp. 714-720 vol.1.
[22] J. Snyder, "Doppler Velocity Log (DVL) navigation for observation-class ROVs," *OCEANS 2010 MTS/IEEE SEATTLE*, Seattle, WA, 2010, pp. 1-9. doi: 10.1109/OCEANS.2010.5664561

Water Sensors

As there is a very large variety in the types of information researchers are interested in collecting[23] [24] [25], it was determined that the most accommodating solution would be to simply provide a bay in which water sensors to suit the individual researchers needs could be used.

Local Awareness

There are several options for detecting objects in the local of a device. The simplest to implement is a physical sensor. In the case of an AUV, this would be a system of switches or pressure sensors around the exterior of the hull, which would alert the AUV of any direct contact with external objects. This is very low level, and therefore has the advantage of being able to use minimal hardware and processing to determine collisions.

Acoustic rangefinders are a common method of not only detecting an object, but detecting how far away it is. These systems are often used both on land and in water, though the specifics of the hardware differ based on medium. For this system to work, the object being detected must reflect the acoustic signal instead of absorbing it.

A similar functionality exists with the use of infrared. IR range finders can detect how far away an object is from itself, though only as long as the object reflects infrared. Due to the attenuation of this wavelength in ocean water, as can be seen in figure 3, this would not be the ideal solution for underwater purposes (Infrared falls in the ≥700nm range).

---

[23] https://www.ysi.com/products/aquaculture-process-monitors-and-sensors

[24] http://news.mit.edu/2018/fundamental-equations-guide-marine-robots-optimal-sampling-sites-0510#separator-comments

[25] https://peerj.com/articles/1770/

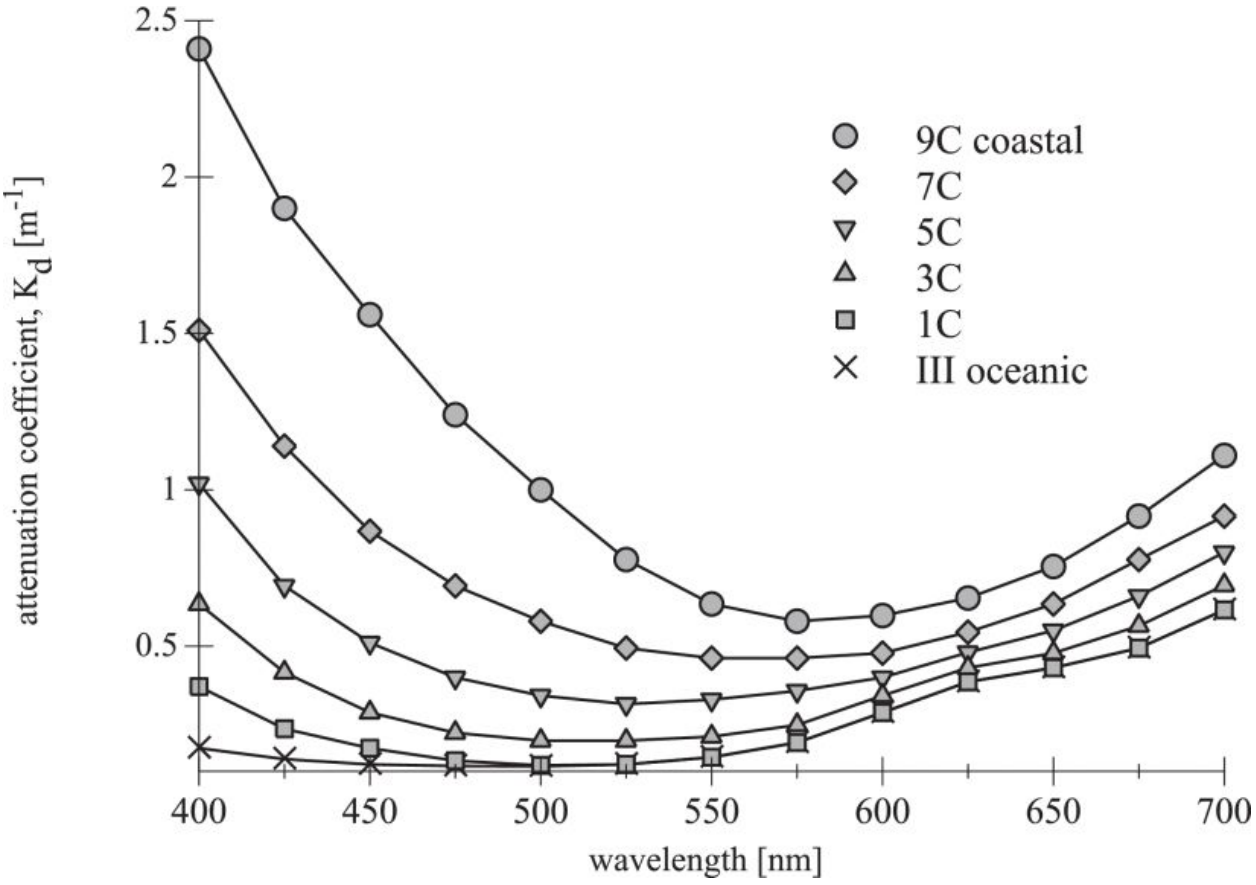


Fig. 3. Relationship between attenuation and wavelength in oceans.[26]

As minimization of environmental impact is one of the most important requirements on this project, any solution which involved direct contact with detected objects is not conducive to our specifications. Due to the attenuation factor in infrared sensing, acoustic sensing was determined to be the best option.

For acoustic signal generation, a transducer is required, which is a component that converts electrical signals to physical motion. Due to how well they are suited to the task, piezoelectrics[27] seem to be the only standard method of making acoustic transducers. The method

[26] J. Sticklus, P. A. Hoeher and R. Röttgers, "Optical Underwater Communication: The Potential of Using Converted Green LEDs in Coastal Waters," in IEEE Journal of Oceanic Engineering, vol. 44, no. 2, pp. 535-547, April 2019.
[27] https://en.wikipedia.org/wiki/Piezoelectricity

for unidirectional sonar purposes uses flat piezo elements sandwiched between two layers. One layer is the impedance matching layer, which should have an acoustic impedance half way between the piezoelectric's and transmission medium's impedances. The optimal thickness of this layer is debated to either be one quarter wavelength of the resonant frequency of the piezo[28], or half the thickness of the piezo itself[29]. Tests are planned to compare the performance of each of these methods. The other layer is required to be "immovable". That is, when the piezo is run at a frequency, making it contract and expand, the impedance matching layer should be producing the maximal amount of motion, while the other side is as relatively stationary as possible. As previously noted, acoustic waves travel furthest in water under values of around 100kHz, with the lower the frequency the better. A graph of this phenomena can be seen in

[28] (n.d.). Retrieved from https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/EquipmentTrans/characteristicspt.htm.
[29] Butler, J. L., & Sherman, C. H. (2018). Transducers and Arrays for Underwater Sound. Cham: Springer International Publishing.

figure 4.

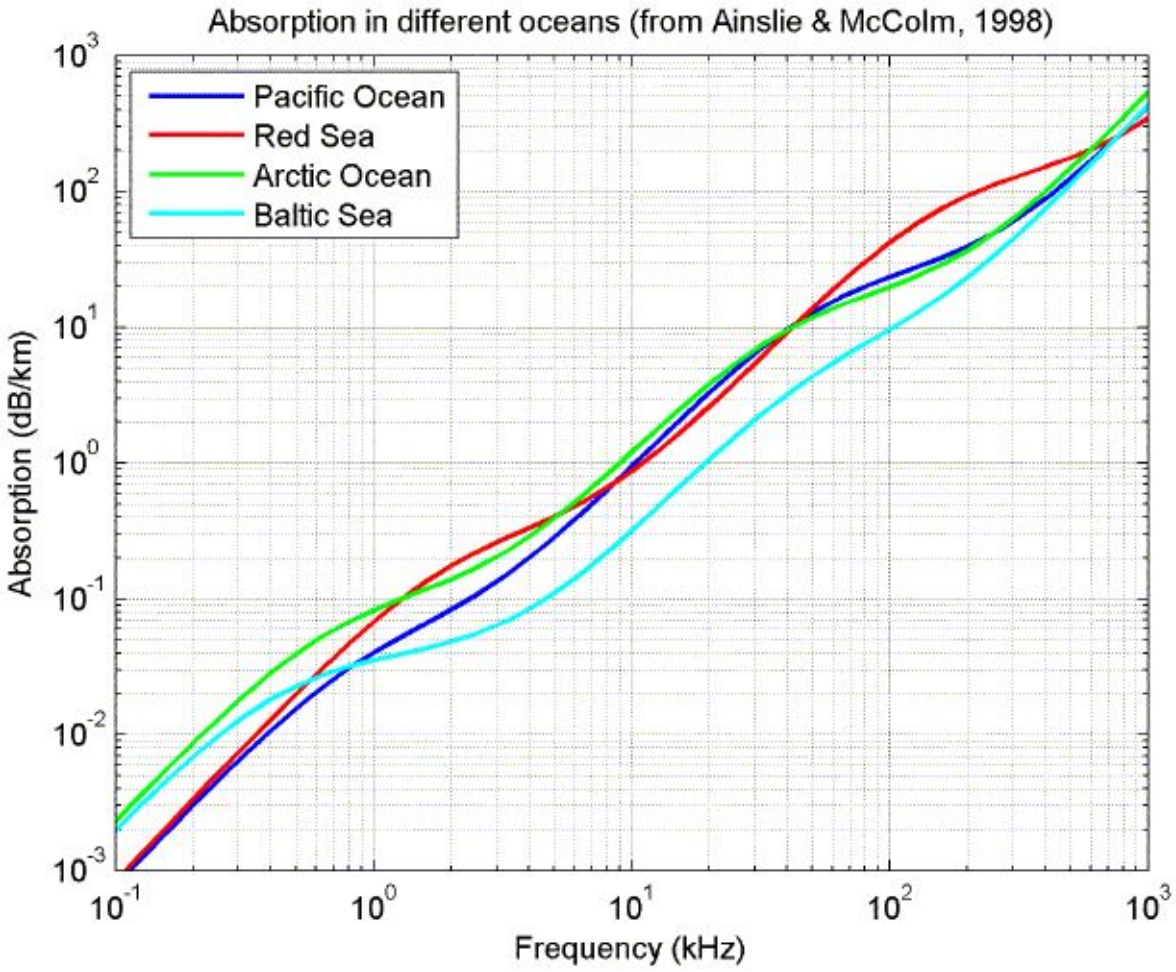


Fig. 4. Relationship between attenuation and frequency in oceans.[30]

This provides a wide range of options for the frequency selection of the AUVs transducers. We settled on 4.6kHz as this falls on the lower end of the frequency range, and has a wavelength of 7.17cm in water, which aligns with our sonar needs. Anything in this range would have worked, but 4.6kHz is a commonly available piezo frequency. As piezoelectric devices produce significantly greater amplitudes at their resonant frequency, this frequency will be used for the

[30] Ainslie M. A., McColm J. G., "A simplified formula for viscous and chemical absorption in sea water", Journal of the Acoustical Society of America, **103**(3), 1671-1672, 1998.

transmission of pulses. To transmit acoustic signals, the piezo devices produce higher amplitude waveforms at higher voltages. As our transducers are custom made, testing will need to be done to determine the voltage at which these will be driven.

There are several options for processing the received acoustic signals. The raw data will be analog AC voltage, which contains all vibrations imparted on the piezoelectric, not just the ones which contain information from the other AUVs. The first option is to run the signal through an ADC and perform the processing on the SBC. This has the disadvantage of increasing SBC load, but it also means that there is minimal hardware required to read in this data.

If the data is to be processed beforehand, a filter is required. Though the signal may experience some frequency distortion, it will still approximately be the same as the piezo's resonant frequency. To remove background noise, a bandpass filter, which allows a specific frequency range to pass through, will be required. This signal, which would now ideally only contain the amplitudes of signals received in the exact frequency range of the transmitted waves, could then be fed through an ADC to the SBC for further processing. To be able to select a bandpass filter IC, we need to know how much the frequencies shift during transmission, which will not be measured until the acoustic testing begins. This method has the advantage of removing a portion of the processing requirements while maintaining information about how strong the received signal was. However, in the situation where there is noise within this frequency range, there would still be a good amount of processing required to detect and extract the features of the actual data.

An alternative to this would be to replace the ADC in the previous configuration with a comparator, such as the LM111, which would have a set threshold, and if the received signal is

above said threshold, a digital 1 would be output to the SBC. This would effectively directly convert from the received AC signal to digital data in hardware, removing all significant processing from the SBC. This has the drawback of losing amplitude information, which could be used to extrapolate location information or to algorithmically reject certain types of data based on its characteristics.

There are a lot of ways that sound waves can be distorted in water, and a primary concern is multipath, where the signal ends up bouncing off of various surfaces, concluding in the same transmission reaching the receiver across a range of times. The signal can also bend, shift frequency, or heavily attenuate, and we have yet to perform the testing which will inform us of how these problems will need to be approached, which guide how to optimally decide the hardware configuration.

Movement

Several propulsion methods were explored by the mechanical engineering team, and as the control of thrusters is an interdepartmental effort, the choice of methodology affected the electrical design.

The options were determined to fall within two categories; Biomimicry or propellor based systems. Biomimicry, or the copying of methods used by organisms, has been a subject of interest in the development of AUVs[31] [32] . A large amount of this research shows that these

[31] Fish, F.E.; Schreiber, C.M.; Moored, K.W.; Liu, G.; Dong, H.; Bart-Smith, H. Hydrodynamic Performance of Aquatic Flapping: Efficiency of Underwater Flight in the Manta. *Aerospace* **2016**, *3*, 20.
[32] Font D, Tresanchez M, Siegentahler C, et al. Design and implementation of a biomimetic turtle hydrofoil for an autonomous underwater vehicle. *Sensors (Basel)*. 2011;11(12):11168–11187. doi:10.3390/s111211168

propulsion methods are highly efficient, which suits the needs of this project. However, these methods require a large number of complex moving parts, which reduces the robustness of the system. The thrust vectors produced by biomimicry options are also significantly less controlled than traditional propellor based propulsion methods, and due to these limitations, the mechanical engineering team chose to pursue the propellor based path in the interest of having a functional product within the designated time frame.

Traditional propellers, with the blades extending outwards from a central axel, have a significant amount of data collected on their characteristics, and therefore designing them to match a performance is relatively simple. However, when it comes to their actual usage for an AUV of this scale, there are a large number of drawbacks that show up. The foremost is the tendency of this prop type to get caught up in seaweed, line, or any other fibrous free floating material. Having the blades externally facing also provides the opportunity for the harm of wildlife[33], which violates one of our design requirements.

To solve this, the Alex Pradhan, a member of the mechanical engineering team, designed a hubless rim driven propellor with hydro lubrication. This has the advantage of being significantly more difficult to tangle, and not having potentially harmful elements directly exposed. Alex worked with Xavier to write an optimization algorithm for the blade characteristics, and production and testing of the thruster will be performed in 2020. A CAD model of the design can be seen below in figure 5.

[33] https://www.nationalgeographic.com/animals/2019/07/north-atlantic-right-whales-mass-mortality/

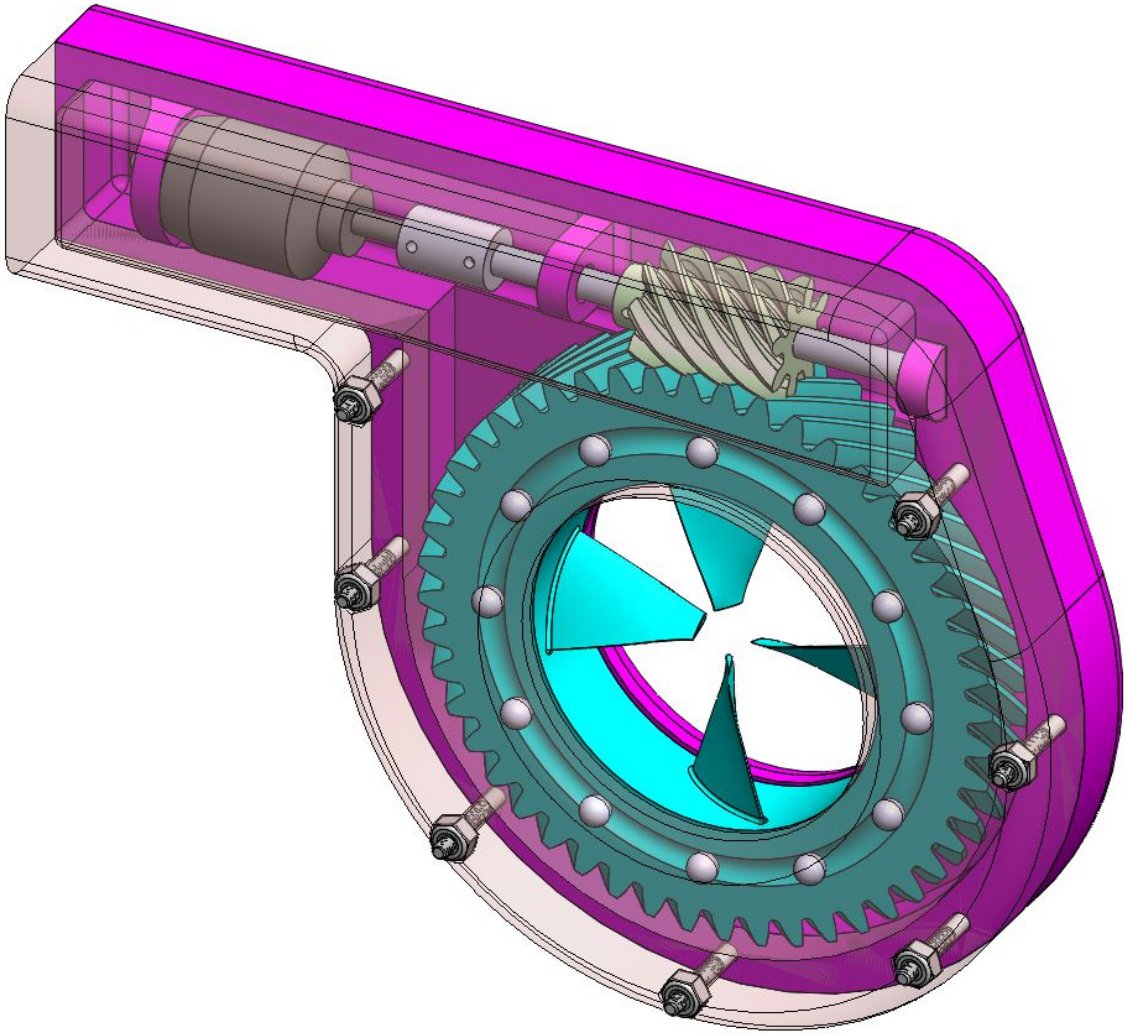


Fig. 5. CAD model of hubless rim driven thruster

While the required characteristics of the motor are not yet known, the motor selection can be narrowed down based on suitability for our requirements. The types of motors that are easily accessible within this size range and are able to provide speeds high enough to fall in the predicted range of thruster requirements are DC brushed motors or brushless outrunners.

Brushed motors utilize brushes to generate and alternating current in the rotor, where permanent magnets induce electromagnetic force on the rotor, causing it to rotate at a velocity

proportional to the input voltage. Due to the constant contact between the spinning rotor and the brushes, brushed motors are extremely prone to failure due to wear. As the brushes are live contacts, any introduction of water to the interior of the motor would result in immediate termination of its functionality,

Brushless motors utilize 3 coils provided with a 120° phase offset AC voltage to induce a rotational velocity proportional to the input frequency from the magnets positioned on the rotor. As there is no direct contact between the stator and the rotor anywhere besides the bearings, there is very little wear on these motors over time. As the coils are insulated, these motors are also able to function while submerged as long as the connection to the ESC is waterproofed.

Due to their increased lifespan and suitability for aquatic usage, we have determined brushless motors will be used for the final thruster. Depending on the predictability of the thruster characteristics, an encoder may be included with the motor to get direct feedback on rotational velocity to ensure synchronization of thrust. This will be determined based on the results of preliminary thruster testing.

Design

As there are so many interlinking aspects that will need to function in conjunction for the final system to work as designed, a minimum viable product[34] approach is being used for the development of the system. This means that versions of the system which are able to represent minimal subsets of the finals systems functionality to allow for testing before implementation into the next iteration. The scope of this project has been divided into three different iterations,

[34] https://en.wikipedia.org/wiki/Minimum_viable_product

referred to as as the MKI through MKIII. The MKI is designed to exhibit 2D motion in water, and provides a testbed for motion sensing and navigation code to be testing. The MKII has the addition of communication and 3D navigation capabilities, allowing for testing of a significantly greater functionality. The final iteration will implement all aspects and components as designed by the mechanical engineering and computer/electrical engineering teams.

The software and hardware aspects of design for this project will be examined in two separate sections.

Hardware

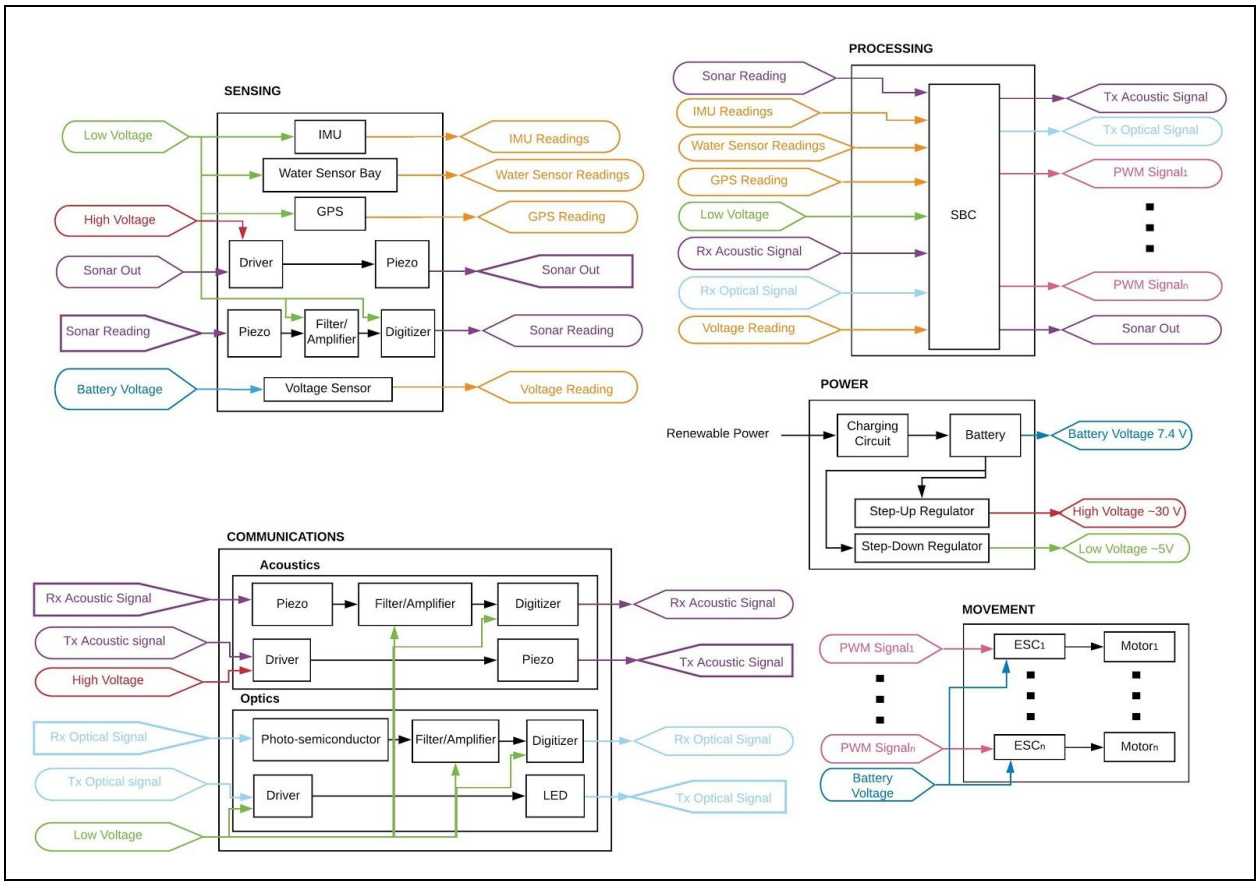


Fig. 6. Hardware Block IO Diagram

There are five main blocks in the hardware design: Communications, Movement, Power, Processing, and Sensing. Each of these blocks are described in detail below.

Communication block

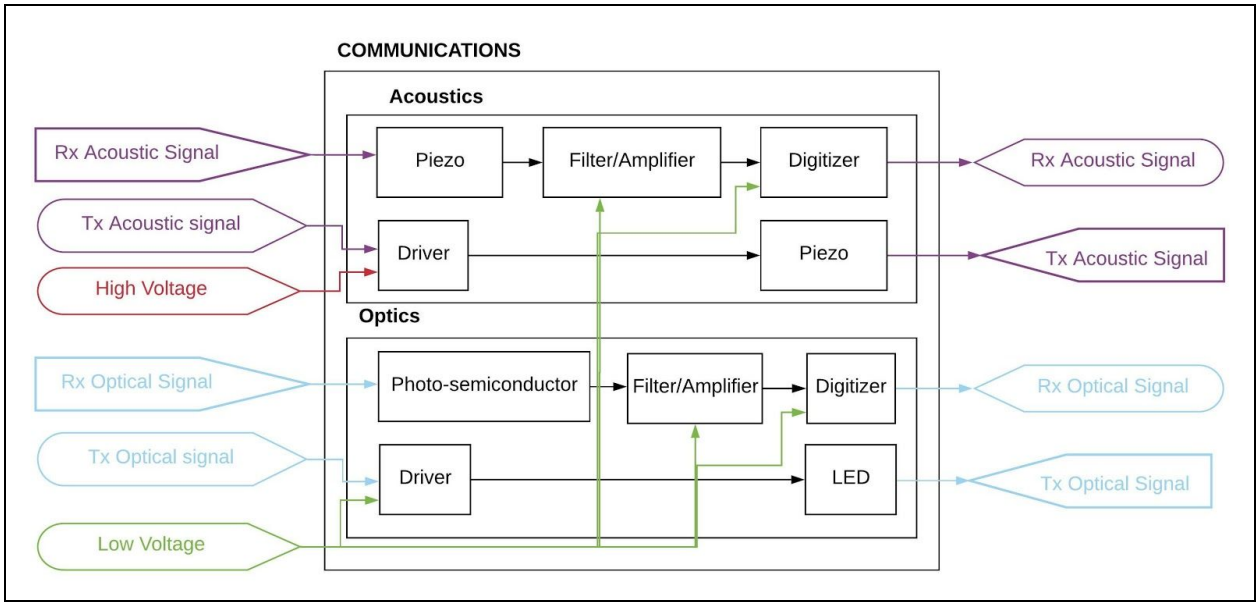


Fig. 7. Hardware IO diagram for Communications block

The Communications block is responsible for sending and receiving all inter-sub messages. While this block is not expected to form these messages, and in fact is expected to treat all messages agnostically, it handles all physical transmission and reception of both optical and acoustic signals to the degree that it can be digitally processed by the SBC. This section is divided into two subsections: Acoustics and Optics, which details the internal structure of the two methods for communication.

For transmitting optical pulses, an n-channel MOSFET transistor circuit will be used in a switching application to drive a high power LED. The high power LEDs used here are 3 W Power LEDs with a viewing angle of 120 degrees, a maximum current draw of $I = 1A$ and a forward voltage of $V_f = 3V$. A thorough justification of this implementation can be found in the *Design Alternatives* section. A diagram for this circuit is shown below.

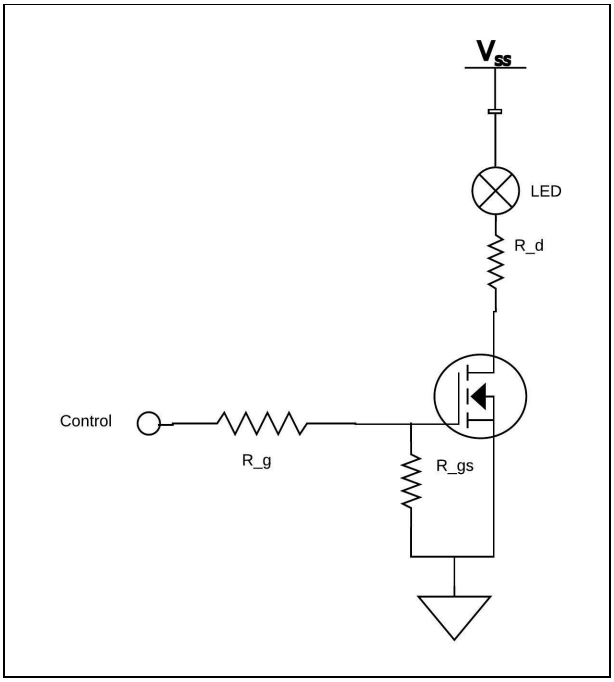


Fig. 8. Optical emitter transistor switching circuit

The transistor used in this application will be a power MOSFET with a high continuous drain current, made necessary by the draw characteristics of the LEDs. As mentioned in the *Design Requirements* section, omni-directional optical emission during transmission is desirable, so a single MOSFET with a maximum continuous drain current greater than the current draw of multiple power LEDs running simultaneously would simplify circuitry and keep the price low. As this MOSFET will be driven by a GPIO pin of the SBC (3.3 V), a single MOSFET would also prevent potential draw issues at the gate, as opposed to multiple MOSFET gates connected to a single GPIO pin. The transistor chosen for this application is an IRLB8721PbF MOSFET, which has a continuous drain current $I_D = 62$ A, max drain-to-source voltage $V_{DS} = 30$ V, gate threshold voltage $V_{GS,th} = 1.8$ V, drain-to-source resistance $R_{DS,on} = 8.7$ mΩ, and switching delay times in the tens of nanoseconds.

In *Fig. 8*, R_g is implemented to reduce buzzing at rising/falling edges during switching. The value of R_g will be kept small in order to prevent the consequential RC delay (from R_gate and C_gate-source) from limiting the overall switching speed of the system.

R_gs is implemented to ensure that noise and stray internal capacitances, often known as "Miller Capacitance", do not accidentally turn the gate on. This is especially necessary in high frequency switching applications.

R_d will be used to limit current draw from V_ss. Due to the LEDs low forward voltage of V_f = 3 V, V_ss can be the low voltage of 5 V, as discussed in the Power block section. Therefore, the optimum R_d value for I_d = 1A is $R_d = \frac{5V-3V}{1\,A} - 8.7m\Omega = 1.99\Omega$.

For receiving optical pulses, a 570nm phototransistor will be utilized. A full justification for this choice can be found in the *Design Alternatives* section. A diagram for this circuit is shown below.

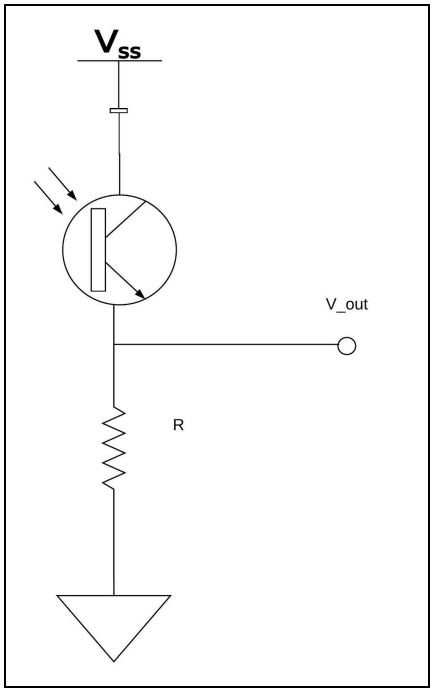


Fig. 9. Optical receiver circuit

The common-collector circuit consists of a phototransistor and a resistor in series, with V_out = V_R. When light is present and the phototransistor is saturated, it has a very low voltage across the collector and emitter. When light is absent, it has a higher voltage. V_out will feed into a comparator circuit, which will output a logical 1 for the microcontroller when a strong enough signal is detected by the phototransistor. Through empirical testing, the optimal threshold voltage will be determined and implemented into the comparator circuit

The value of R will be selected according to how sensitive the circuit must be to different light levels. The phototransistor used in this application is an SFH 3310 570nm NPN phototransistor, with V_CE_max = 5.5 V and I_CC_max = 20 mA. With V_ss = 5 V, the minimum resistor value is $R_{min} = \frac{5V}{20mA} = 250\Omega$. However, this condition is only reached at full saturation, which may not ever be reached in this application. Therefore the final value of R will reflect the maximum saturation that can be reached.

Power block

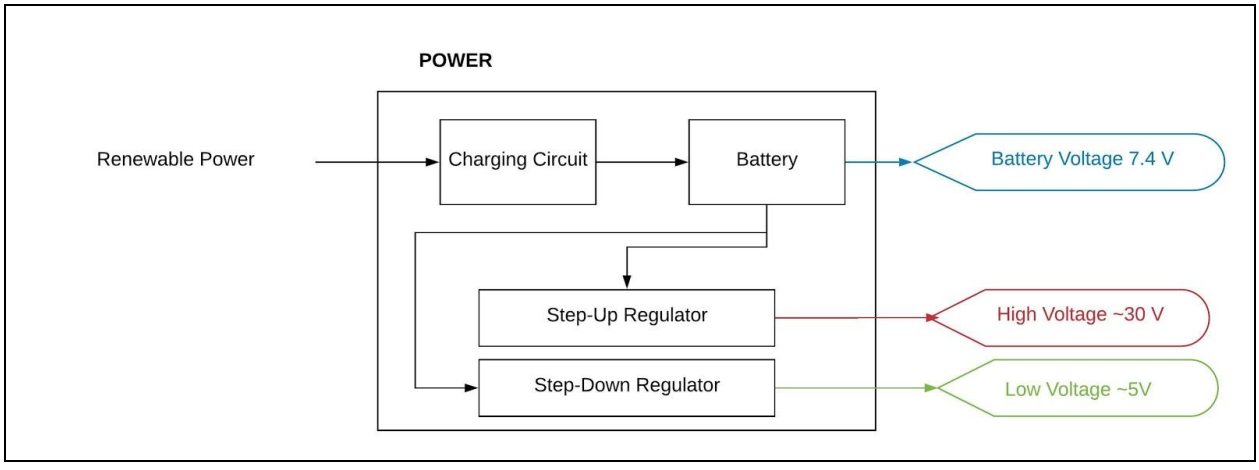


Fig. 10. Hardware IO Diagram for Power block

The Power block is responsible for powering all active components in the system, and for generating power renewably while away from a charging station. The power reserve will be some configuration of common off-the-shelf batteries with a single cell voltage of around 3.7 V, therefore requiring that step-up and/or step-down regulators be used to provide at least two distinct voltage levels: one of a relatively low voltage (~5V) for powering the numerous low-power components, and one of a relatively high voltage (~30V) for driving piezoelectric transducers.

The battery will have a nominal voltage of 7.4 V, also known as a 2S configuration. The SBC will need to be continually powered during any given mission, so selecting a nominal battery voltage close to the actively regulating voltage will minimize power loss. Using a lower nominal voltage will also lessen the power consumption from the motors. The mechanical engineering team working on this project have implemented a helical gear train calculator in their ongoing thruster design with a modifiable gear ratio, meaning that thruster speed and torque can be tuned mechanically. With driving high performance motors, active voltage regulation and periodic high current draw from communicative components, the battery must boast a high maximum continuous current draw and decent capacity. Therefore the battery will include several individual batteries in parallel, creating a 2S*x*P configuration. The final value of *x* will be determined by the available space in a single sub. Please refer to the Movement block section for more information regarding motors and the Communications block section for more information regarding communicative components.

The step-up regulator will provide a high voltage level for the piezoelectric transducers to be driven off of. Generally speaking, AC waves produce greater displacement in a piezoelectric

transducer when higher voltages are utilized, meaning that a high driving voltage is needed to send acoustic waves over large distances. For more information about transducers, please refer to the Communications block section. The implementation for this regulator will be a boost switching converter, ideally with high efficiency ratings and low latency. As this regulator has a single relatively-stable load, large continuous current ratings are not required.

The step-down regulator will provide a low voltage level for the SBC, IMU module, GPS module, optical driving circuit, and all active filters/amplifiers. The regulation for this voltage will be continuous as discussed before, however the load will largely vary; driving circuits will draw power periodically and in bursts, and filters/amplifiers will be similarly pulsed. Therefore, it is important that this regulator has a high current rating and handles under non-linear conditions. The exact current draw will be dependent upon the total number of LEDs used and other particulars of the system, but may very well be up to 10 amps. The implementation for this regulator will be a buck switching converter. This use case, being the conversion of some battery voltage to 5 V, is a common issue in RC circuits. While primary motors are driven at the battery voltage, other components such as servo motors and the receiver are typically powered off of 5 V. In order to avoid needing to use a second battery for these components alone, a Battery Eliminator Circuit, or BEC, is used to step down the battery voltage to 5 V for only these components. Therefore, regulators that can handle nonlinear supply voltages and loads are relatively cheap, tried, and true.

Finally, the charging circuit will provide a means for some renewable power to be harvested and recharge the battery. The implementation of this component will be handled by the

mechanical engineers on this project. The only requirements for this circuit is that it recharge the battery entirely, and does so completely renewably.

In an effort to gauge what the power draw might be, calculations of consumption for each component to be included in this design is performed and provided below. These figures are for a worst-case scenario, the details and justifications of which are provided in the following paragraph.

Component	**Notes**	**Final Power Consumption (Wh)**
SBC - Raspberry Pi 4B	*Overclocked @ max utilization*[35]	8
Piezoelectric transducers	With 6 transducers, $Z_{resonant} = 300\Omega$, $V = 30V \rightarrow$ $P = \frac{V^2}{Z_{resonant}} * 6\ transducers = 18W$, assuming 50% on time $\rightarrow$ $E = 9Wh$	9 Wh
LED driving circuit	With 6 LEDs and FET characteristics of $V_{cc} = 5V$, $I_c = 1A$, $R_{ds(on)} = 8.7m\Omega \rightarrow$ $P = (6\ LEDs * I_c)^2 * R_{ds(on)} = 0.31W$, assuming constant transmission $\rightarrow$ $E = P * \frac{on\ time}{off\ time} = 0.31W * \frac{3}{8} = 0.12Wh$	0.12
LEDs	With 6 LEDs, $P = 3W$, assuming constant	6.75

[35] https://www.pidramble.com/wiki/benchmarks/power-consumption

	transmission $\rightarrow E = 3W * 6\ LEDs * \frac{3}{8} = 6.75Wh$	
IMU	With IMU characteristics of $V_{DD} = 3.3V,\ I_{DD} = 12.3mA \rightarrow P = 0.041W,$ assuming constantly on $\rightarrow E = 0.041Wh$	0.041
GPS	With $I_{continuous} = 29mA,\ V = 3V \rightarrow P = 0.087Wh$	0.087
Filters/ Amplifiers	With 6 LEDs, 6 transducers, $V = 5V,\ I = 7mA \rightarrow P = 12 * 5V * 7mA = 0.45W$, assuming 50% on time for transducers and constant transmission for LED $\rightarrow$ $E = (\frac{0.45W}{2} * 0.5) + (\frac{0.45W}{2} * \frac{3}{8}) = 0.197Wh$	0.197
Digitizers	With 6 LEDs, 6 transducers, $V_{cc} = 3.3V,$ $I_{in} = 7.5mA \rightarrow P = 12 * 3.3V * 7.5mA = 0.297W,$ assuming 50% on time for transducers and constant transmission for LED $\rightarrow$ $E = (\frac{0.297W}{2} * 0.5) + (\frac{0.297W}{2} * \frac{3}{8}) = 0.13Wh$	0.13
ESCs/Motors	With $V_{nominal} = 7.4V,\ I = 15A$, 5 motors, and 50% on time $\rightarrow P = 111W$, assuming 5 motors and 50% on time $\rightarrow E = 111W * 5 * 0.5 = 277.5Wh$	277.5
Phototransistor	With phototransistor characteristics of	0.04

circuit	$V_{ce,\,max} = 5.5V$, $I_{c,max} = 0.02A \rightarrow P = 0.11W$, assuming constant transmission → $E = 0.11W * \frac{3}{8} = 0.04Wh$	
Piezoelectric transducer driving circuit	With 6 transducers, $I = \frac{30V}{300\Omega} = 0.1A$ and FET characteristics of $V_{cc} = 30V$, $R_{ds(on)} = 8.7m\Omega \rightarrow$ $P = (6\ transducers * I)^2 * R_{ds(on)} = 0.003W$, assuming 50% on time → $E = 0.0015Wh$	0.0015
	TOTAL:	**301.8665 Wh**

Table. 1. Power consumption breakdown of all hardware components

For the piezoelectric transducers, 6 transducers are used in this calculation. This provides sonar readings in all directions of all axes. The possibility of obstacles in all axes is a concern, but as LED placement is also omnidirectional, this will allow a sub to prevent collisions with other subs as local awareness is also omnidirectional. The on-time to off-time ratio used here is is 0.5, as described in the Sensing block.

For LED driving circuit, 6 LEDs are used in this calculation. As stated in the Communication block, the power LEDs used in this design have a max current draw of 1 A and a viewing angle of 120 degrees. Therefore, 6 LEDs allow for omnidirectional visibility. For the ratio of on-time to off-time in the same category, the communication protocol as described in the design requirements section (PPM) describes the ratio of on-time to off-time for sending a bit '0' or a bit '1'. For '0', the clock pulse is the only on-time, so this ratio is ¼. For '1' both the clock pulse and the data pulse make up the on-time, so this ratio is ½. Therefore, the average is on-time

to off-time ratio is ⅜. The FET used here is the one selected for this design as described in the Communication block.

For filters/amplifiers, the exact part selection has yet to be completed for reasons described in *Design Alternatives*. For calculations performed above, a universal active filter similar in functionality to our use case, the UAF42, is used for calculations with specifications listed in the above table.

For digitizers, a differential comparator as described in *Design Alternatives* is used for calculations. This comparator is used for digitization of both optical signals and sonar readings due to its wide operating range.

For ESCs/motors, the voltage figure is the nominal voltage for the battery configuration this design will utilize and the current draw is based on the maximum current rating for the bidirectional ESCs currently used. The mechanical engineering team was consulted for the greatest number of motors that may be incorporated in the design, for which the answer was 4. In this configuration, 2 motors would be used to propel the AUV forward as a tank-drive propulsion system, and 2 motors would be used to surface and dive the AUV. The average on time assumes that two motors on the AUV will always be spinning; either the two motors for forward motion, or the two motors for vertical motion.

Processing block

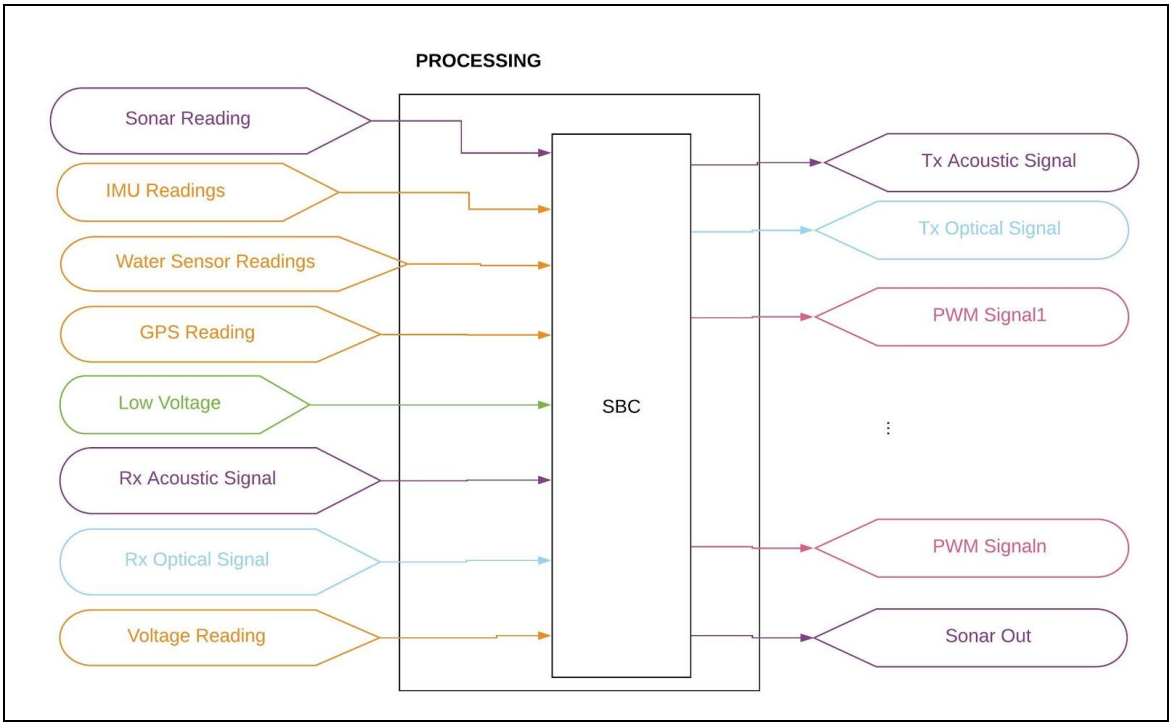


Fig. 11. Hardware IO Diagram for Processing block

The Processing block is responsible for all computational requirements. Therefore its inputs are all readings from the Sensing block, all digitized incoming communication signals, and low voltage for powering. It's outputs are digitized outgoing communication signals and PWM signals for all motors/control surfaces. For more information about the communication signals, please refer to the Communications block.

The processing block itself does not have any internal hardware implementations aside from the SBC. The SBC essentially contains the entire software design, making for a short conversation in a hardware IO breakdown. A full examination of the IO requirements for this SBC is provided below, along with a list of all hardware components that the SBC will interface

with and their IO. These requirements are based on a worst-case scenario that has been established in the Power block.

Block	Component	I/O Required
Sensing	Piezoelectric transducer driver circuit	GPIO pin
Sensing	Piezoelectric transducer digitizer	GPIO pin
Communication	LED driver circuit	GPIO pin
Communication	Phototransistor circuit	GPIO pin
Sensing	Battery voltage sensor	I2C
Sensing	IMU	I2C/UART
Sensing	GPS	UART
Sensing	Sonar out	GPIO pin
Sensing	Sonar reading	GPIO pin
Sensing	Water sensor bay	I2C
Movement	Motor drivers	GPIO pin*5

Table. 2. IO breakdown of all hardware components that interface with the SBC

As can be seen above, the overall IO requirements for the SBC include: 11 GPIO pins, two I2C ports, a UART port, and one more I2C/UART port. As described in the *Design*

Alternatives section, the current SBC being used is the Raspberry Pi 3B. This SBC has 2x I2C, 2x UART, 2x SPI, and 14x dedicated GPIO pins (that is, not also for I2C/SPI/UART). It should be mentioned that one of UARTs on the Raspberry Pi is tied to its wireless/bluetooth module which is used for interfacing with the pi during tests, effectively making this 1x UART. As I2C can be performed by chaining multiple devices, the water sensor bay I2C channel will be chained to one of the two I2C ports and properly addressed to ensure that there is no address collision between I2C peripherals. Therefore, this SBC still has the necessary GPIO to house all peripherals stated in this design.

For process-based design, please refer to the Software section for more information.

Sensing block

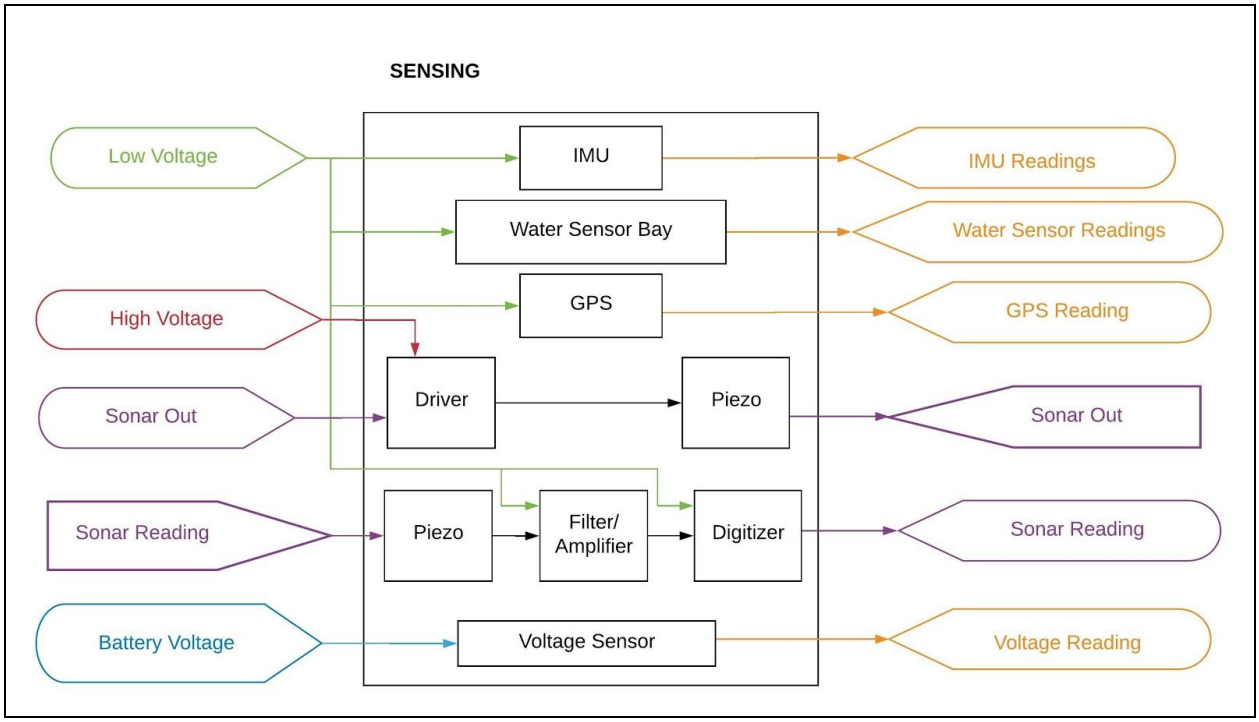


Fig. 12. Hardware IO diagram for Sensing block

The Sensing block is responsible for handling all sensory input. This does not include communication between individual systems. There are five sensing modules involved in this block: an IMU module, a GPS module, a water sensor module (a bay containing multiple water sensors), a sonar module, and a battery voltage sensor module. The first four sensing modules will require power at a low voltage, and the last sensor module will require battery voltage.

The purpose of an IMU module is to provide estimations for velocity and displacement underwater, necessary for dead-reckoning. A desirable IMU for this application is a 9-DOF sensor, which includes an accelerometer, capable of providing instantaneous acceleration readings, a gyroscope, capable of providing orientation and angular velocity, and a magnetometer, capable of gauging magnetic fields (most commonly, Earth's magnetic field). This reports absolute orientation in 3-space and motion estimations in all axes.

Calculating displacement from instantaneous acceleration results in unbounded error accumulation, spurring the need for some post-processing and filtering on the collected data. This work is explored under *Preliminary Testing Results*. After comparing similarly priced IMU's, the BNO055 absolute orientation sensor was chosen due to its relatively high accuracy, configurable sampling rates and onboard sensor fusion and processing capabilities. The second point was especially poignant, as sensor fusion is a complicated field that would take significant time to develop and refine (potentially a senior capstone project on its own), and onboard processing means those computations can be offloaded on the SBC- a desirable trait because these readings will happen continuously during a mission and the possibility of software delays due to absolute orientation calculation is abated.

The purpose of the GPS module is to reference the sub to a global coordinate frame and realign all estimated positions from dead-reckoning. This allows for course correction if the sub accidentally strays from its projected path. For point-to-point travel, GPS will also provide a means to definitively start and conclude a mission. The SAM-M8Q chipset, selected based on the criteria outlined in *Design Alternatives*, was released in 2017.

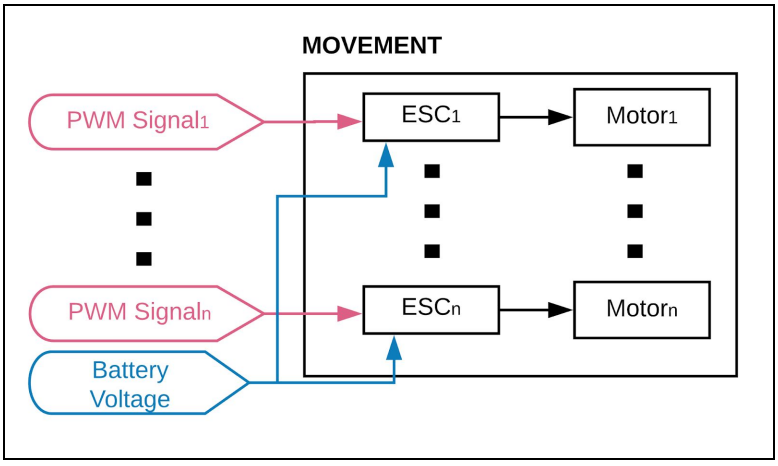
The purpose of the water sensor module is to generate the underwater data that the user is interested in, allowing them to choose what sensors best suit their purpose. With the water sensors that the user decides to equip their sub with, readings will be taken periodically and timestamped for later post-processing. The water sensor bay will be modular, enabling any water sensor with I2C capabilities to be attached to the bay. This will allow the user to decide what factors they care about and implement sensors accordingly. Additionally, this keeps the overall price lower because expensive sensors won't be implemented as a default option which might not be needed at all. In the instance in which a user wants to execute point-to-point travel, they may choose to omit water sensors altogether.

The purpose of the sonar module is to provide basic information about local surroundings to the sub. This implementation will be as selected in design alternatives, though the specifics of the hardware will not be confirmed until preliminary tests are performed. Sonar is a necessary implementation to ensure that the sub does not contact any walls, floors, or ceilings while underwater because position estimations may not be fully accurate. To match our requirement of being able to detect obstacles at a distance of 1.5 meters, the transmission power will be selected based on the results of our initial acoustics testing. It takes sound $\frac{3m}{330m/s} = 0.009s$ to travel a distance of 3m. To meet the requirement of being able to avoid an obstacle with a relative

velocity of 1m/s, the AUV would have $2s - 0.009s = 1.991$s of time to negate its relative velocity. Based on the design requirement of being able to exhibit at least 1m/s of acceleration, this gives the AUV an overhead of 0.991s to achieve this, as it would reach a negating velocity after 1s. Sensors will be positioned top, bottom, fore, starboard and port, with the potential for multiple sensors dependant on the final length of the AUV as determined by the mechanical engineering team.

The purpose of the battery voltage sensor module is to detect when the battery must be recharged. This is strictly vital, as a battery voltage below the operating range of the step-down regulator will result in shutdown of the SBC, at which point the sub will be lost. Once a low battery voltage is detected, with a factor of safety integrated to ensure the sub can resurface before shutting off, the sub will do so and recharge. This will be accomplished with the 2S fuel gauge MAX17044 with the sparkfun breakout board[36] which uses I2Cto interface. When each cell reaches the suggested minimum threshold for the selected battery chemistry.

Movement block



[36] https://www.sparkfun.com/products/10617

Fig. 13. Hardware IO diagram for Movement block

The movement block will simply consist of the selected motors and and a reversible ESC for each motor. Each ESC will require a single PWM channel. The thruster design and layout, along with the motor selection and number of motors, falls under the jurisdiction of the mechanical engineers. Once the parameters of the motors have been chosen to match the relevant requirements, ESCs will be selected to match these characteristics. Thruster flow testing and iteration through rapid prototyping will occur at the beginning of 2020, and once the speed to thrust relationship is determined, the motor requirements will be finalized.

Software

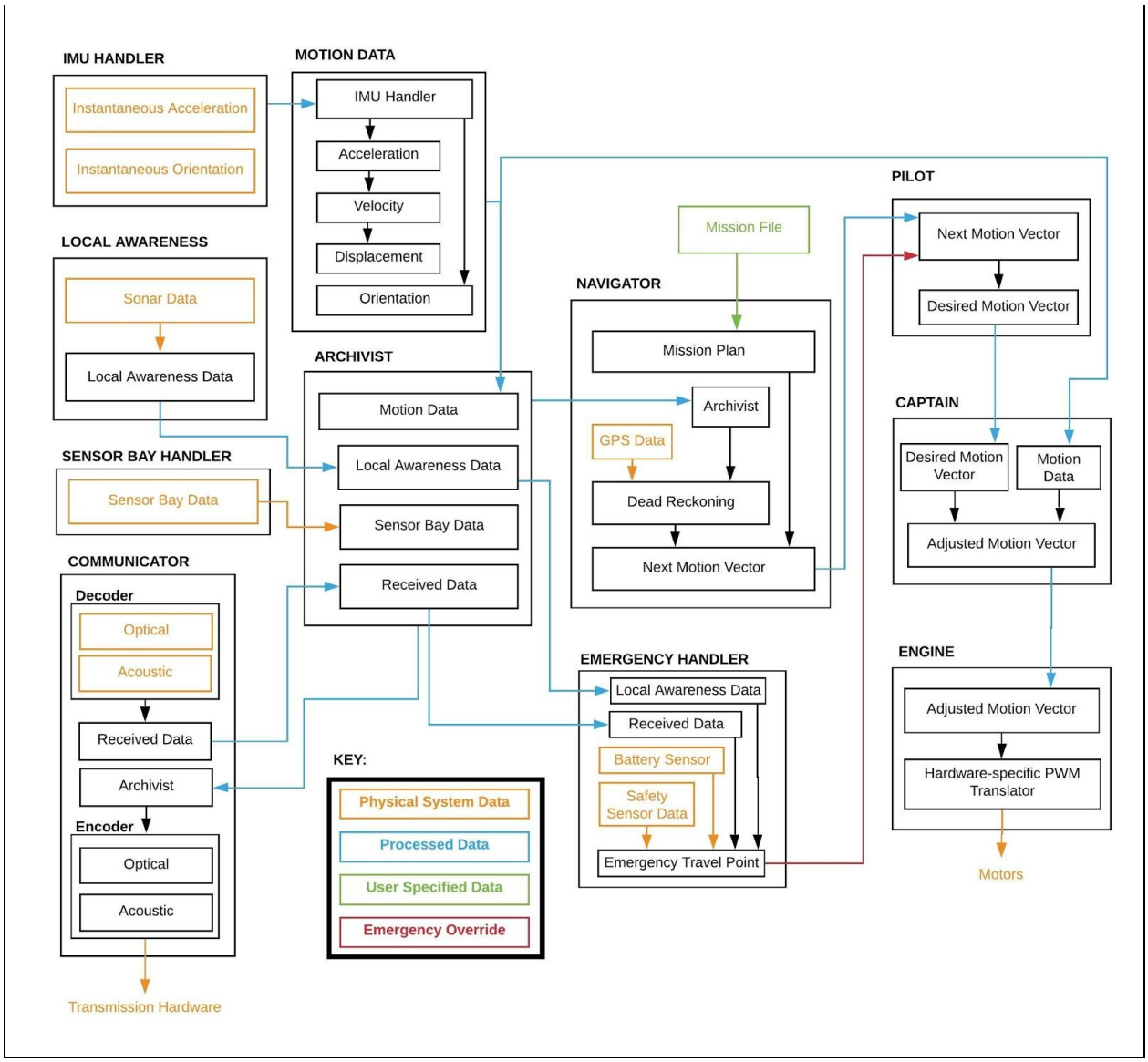


Fig. 14. Software Diagram

There are eleven main sections of code for the system, each of which will be described below in detail. Figure 14 contains a representation of the connections between these systems.

IMU Handler

The IMU Handler is responsible for initializing, calibrating and starting the IMU's measurement. It has a function that returns the most recent acceleration and orientation data when called. The BNO055 interfaces using I2C, and is being accessed via the adafruit unified sensor library. When this is initialized, calibration data is loaded in from the previous successful calibration, and continues running until all values have reached their maximum calibration level. The IMU Handler reads in the linear acceleration vector and the euler vector via the library.

Local Awareness

Depending on the finalized results from the acoustic testing, the precise handling of incoming data will be different based on the finalized selection of hardware, however, the main functionality of this block remains the same. The local awareness block will periodically instruct the acoustic system, described in the *Hardware* section of *Design* to broadcast a signal encoded with a distinct pattern, and then listen for the same encoded signals return. The time gap between sending and receiving will be utilized in conjunction with the speed of sound in water to calculate how far away the object that the signal bounced off of was. The transmission hardware should be perfectly functional as receiving hardware, though testing to confirm that this is the case remains to be done. At the required awareness range of 1.5m, sending and receiving a pulse would take 4.54ms to respond. The encoding is necessary to ensure that the received signal was not broadcast from another sonar sensor. The distances detected for each transducer can then be logged by Archivist.

Sensor Bay Handler

The Sensor Bay Handler will simply interface with the modular sensor bay connections. A protocol will be designated that these sensors will be required to use, and therefore any sensor connected in accordance with this will be able to have its data read. This data will be collected at a user specified period, and the resulting data will be collected by Archivist.

Communicator

The Communicator will be the interface with the communication hardware. Transmission will entail determining the transmission method based on content, encoding the data to be transmitted, waiting for its turn to transmit, and transmitting said data.

The communicator will also consistently listening for incoming transmissions. Upon reception, and processing depending on the state of the received data, the information will be decoded and ready to be archived.

Motion Data

Motion Data starts a daemon process which queries IMU Handler for the instantaneous acceleration and orientation values. Acceleration and orientation values are collected and added to logs as they are measured. Once the number of new acceleration values reaches a predetermined range, R, the most recent R values are passed into the filter function. Because the data is being filtered in relation to frequency, the filter is affected by the size of R. The double integration drastically amplifies even the smallest of errors, so a lot of filtering and processing is

required to ensure that the displacement data even resembles reality. The R values of filtered acceleration are appended to its own log, and from it the integrated velocity and displacement ranges are generated and appended to their own logs. Further development and testing is required to determine if this method is capable of permitting our design requirements without a functional GPS connection, and the current progress of this testing is available in *Preliminary Test Results.* It is important to note that the functionality of the dead-reckoning system is not a direct indicator of the AUVs translocational capabilities, as it will only serve as an input to the navigation system.

Archivist

Archivist is the section of code responsible for collecting, logging and providing all important information. It takes in the collected information from Local Awareness, the Sensor Bay Handler, Motion Data, and Communication, which contains the information received from other AUVs. Archivist then ensures that all data that is frequently used by other modules is held in memory, while all else is stored in a database and removed from memory.

Navigator

The Navigator is a high-level mission control block that performs several functions. First, it generates a mission plan from a user-provided mission file. This includes the mission type (i.e. exploration) and coordinates of the pick-up location at a minimum. Second, it retrieves the most recent motion data from the Archivist and uses this data to estimate its position via dead-reckoning. This estimated position coupled with the waypoint in the mission plan produce a

next motion vector, which is sent to the Pilot. When GPS is used (i.e. during recharging and initial deployment), actual coordinates will be received and the estimated position at that time will be discarded in favor of the newly attained reference position. GPS will be periodically used to prevent unbounded error accumulation from getting to an unreasonable degree.

Emergency Handler

The Emergency Handler is responsible for detecting unsafe states and signalling the pilot accordingly. Emergencies can full under one of three types: Collision, Man-down, and Low battery.

Collision emergencies occur when the sub is at risk of colliding with some physical object. At this point, the sub will immediately stop if need be and take steps to ensure its own safety. This type of emergency is found from local awareness data.

Low battery emergencies occur when the sub has detected a low battery level and must resurface and recharge. This type of emergency is found from battery sensor data.

Pilot

The Pilot is responsible for passing high level motion instructions to the Captain that factor in high priority safety requirements. Thus, it acts as more of an intermediary class between the Navigator and the Captain, but with a vital function. The Pilot will be blind to any physical implementation of the system and only give instructions pertaining to desired motion vectors.

These motion instructions can include a heading change, distance to move, a combination of these two, an emergency stop or an emergency redirect. Heading and distance changes are

routine and should happen repeatedly, but emergency overrides will only occur during situations where the wellness/safety of a sub is compromised, and is discussed in the Emergency Handler section.

Captain

The Captain is responsible for housing logic to exhibit repeatable and modular movement, such as heading changes and linear distance changes. Internally, the Captain adjusts the motion vector instruction received by the Pilot by any deviation produced from the environment, which comes from the most recent motion data. These deviations can include small heading shifts and overall displacements from waves. Continuous motion data is also required to provide feedback for closed-loop control systems.

For the ATS Mk. 1, heading changes are performed by a PID controller. Work describing the implementation and testing of the controller can be found under *Preliminary Testing Results*. This section will describe how the output of the PID is translated into motor output. There are two types of heading changes; stationary and mobile.

While stationary, the goal is to keep a net displacement of zero, and thus motors are spun at equal but opposite speeds. This allows the ATS to modify its heading without compromising it's position drastically. In this case, the PID output o is linearly scaled to a speed change s_c with a programmable PID output ceiling o_{max} and maximum speed change $s_{c,max}$, so that if $o \geq o_{max}$ then $s_c = s_{c,max}$. Otherwise, $s_c = (o/o_{max}) * s_{c,max}$. This speed change is duplicated and sent to the Engine as left/right motor values, polarized according to the intended direction of rotation.

While mobile, the motors are already spinning at some speed s, whereas this was assumed to be 0 for the stationary case shown above. As the captain is the only module that can directly control the motors, it is safe to assume that these motors are being driven at the same speed value as the only time when they should not be is when a heading change is underway. The process for converting PID output to changes in motor speed is very similar to that for a stationary heading change: s_c, $s_{c,max}$, o, and o_{max} terms are still used in the same manner as described above, but the final motor speeds are not sent to the Engine in the same manner. Once s_c is computed, it is subtracted from the current speed s and sent to one of the two motors in order to slow that motor down. For example, if the sub is moving forward and needs to make a slight left turn, the speed change will be subtracted from the left motor's speed value to yaw the sub to the left. Generally, the updated motor speed can be expressed as $s = s - s_c$.

There is the case where $s_c \geq s$; that is, the speed change is greater than the current motor speed. In this case, $s - s_c$ will either stop the motor or cause it to spin in the opposite direction. This is undesirable in real-time navigation, as although logically this is a speed difference of only a small percentage, spinning motors in opposite directions will cause the rate of heading change to increase very quickly and make stability harder to achieve. Therefore, if $s_c \geq s$, the motor to be slowed down will change its speed to 1%, and the other motor will change its speed to the value of the speed change + 1%. This ensures that both motors keep spinning in the same direction and that the difference between the motor speeds is always equal to s_c.

```
/* For yawing to the left */
if PID output <= max PID output; then
    speed change = max speed change
else; then
    speed change = (PID output/max PID output)*max speed change

if  speed > speed change; then
    left motor speed =  speed - speed change
    right motor speed =  speed
else; then
    left motor speed = 1
    right motor speed = speed change + 1
```

Fig. 15. Pseudocode for PID output to motor change logic

Examples have been given for forward motion, but the same logic is implemented for backward motion. This can be useful if a sub needs to backtrack after reaching a dead end in a convoluted environment.

A final implementation detail to mention is the translation of heading values. Heading, commonly read on a compass, ranges from 0 degrees (inclusive) to 360 degrees (exclusive). Consider the situation in which the target heading h_t is equal to 355 degrees and the current heading h is equal to 2 degrees. In this instance, the PID will see a subtraction of 353 degrees as the only way to get to h_t; however, yawing in the other direction would only require a 7 degree heading change. Another example if $h_t = 220$ degrees and $h = 40$ degrees. In this case, the heading difference is 180 degrees whether the sub moves clockwise or counter-clockwise. This caps the maximum necessary turn radius to 180 degrees, which is both power- and time-efficient. Therefore, the following logic is implemented: If the initial heading h_0 is within 180 degrees of h_t, do nothing. If $h_t - h_0 > 180$ degrees, then $h_t = h_t - 360$ degrees. If $h_0 - h_t < 180$ degrees, then $h_t = h_t + 360$ degrees. While the target heading isn't between 0 and 360 anymore, the

controller will now see the shorter path between h_t and h_0. Proper framing is performed to ensure that h is always within the proper reference window for the PID controller.

Engine

The Engine is responsible for driving all motors and control surfaces. This is layout-dependent, however motor drivers and servo motors alike will use PWM as modulation scheme so a uniform interface for controlling these actuators will be implemented, called a PWM translator.

The interface for driving a motor is a single integer that corresponds to a speed and direction. Specifically, this integer is in the range of [-100, 100]. Sending a value of +100 spins the motor at full speed in one direction (corresponding to forward motion), sending a value of 0 stops the motor, and sending a value of -100 spins the motor at full speed in the other direction (backward). The ATS Mk. 1 has only two motors, so spinning clockwise (for example) simply consists of sending a positive left motor value and a negative right motor value, and vice-versa for counter-clockwise.

In implementation, the Electronic Speed Controllers (ESCs) used here are bidirectional and map out the range of motor values to the standard range of duty cycles for servo motors and ESCs. This duty cycle range is 5% - 10% of a 50 Hz signal, meaning that the pulse width ranges from 1ms to 2ms. Therefore a motor value of -100 corresponds to a duty cycle of 5%, 0 to 7.5%, 50 to 8.75%, and so on.

With the assistance of the mechanical engineering team, a tachometer was used to ensure a common minimum speed (i.e. speed = 1 or -1) across all motors. This motor value was

hardcoded into the PWM translator to ensure that at base speeds, motors spin at the same rate. Undesirable outcomes that this prevented include passive drift over time when movement is supposed to be linear.

Preliminary Testing Results

Displacement (performed by Xavier)

As previously noted, generating accurate displacement values from an IMU is very difficult due to the fact that any source of error is exponentially increased during the calculation. Not only does this mean that accuracy in determining how far the AUV has moved if not processed correctly, but it could also falsely generate information saying it has moved when it has been effectively stationary. With the selected BNO055 IMU sensor, these are the steps which have been implemented in an attempt to get usable displacement values.

To have a starting place for this process, information about the measured values from the IMU needed to be collected. To make sure these values were as close to possible as what would be measured in one of the AUV iterations, the internal skeleton of the AFμS MK1, was 3D printed and all tests were performed with the IMU mounted on the center axis. This had the added benefit of encapsulating the Raspberry Pi, which became useful further down the line in testing. The skeleton was moved by hand in a variety of directions at a rate predicted to fall

within the range of likely motion. As the goal of this test was to get a broad indication of the values that will be experienced, precise motion was not necessary.

A fourier transform of the acceleration data was taken, which shows the frequencies experienced along with their amplitudes.

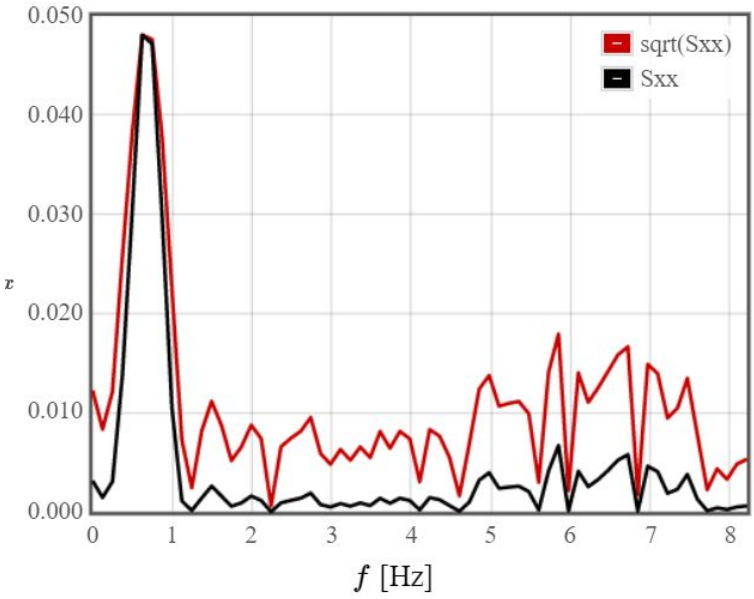


Fig. 16. Fourier transform of predicted motion

As can be seen in figure 16 above, the purposefully generated motion falls in the range of 0-~2Hz, while the lower amplitude data falls into higher frequencies.

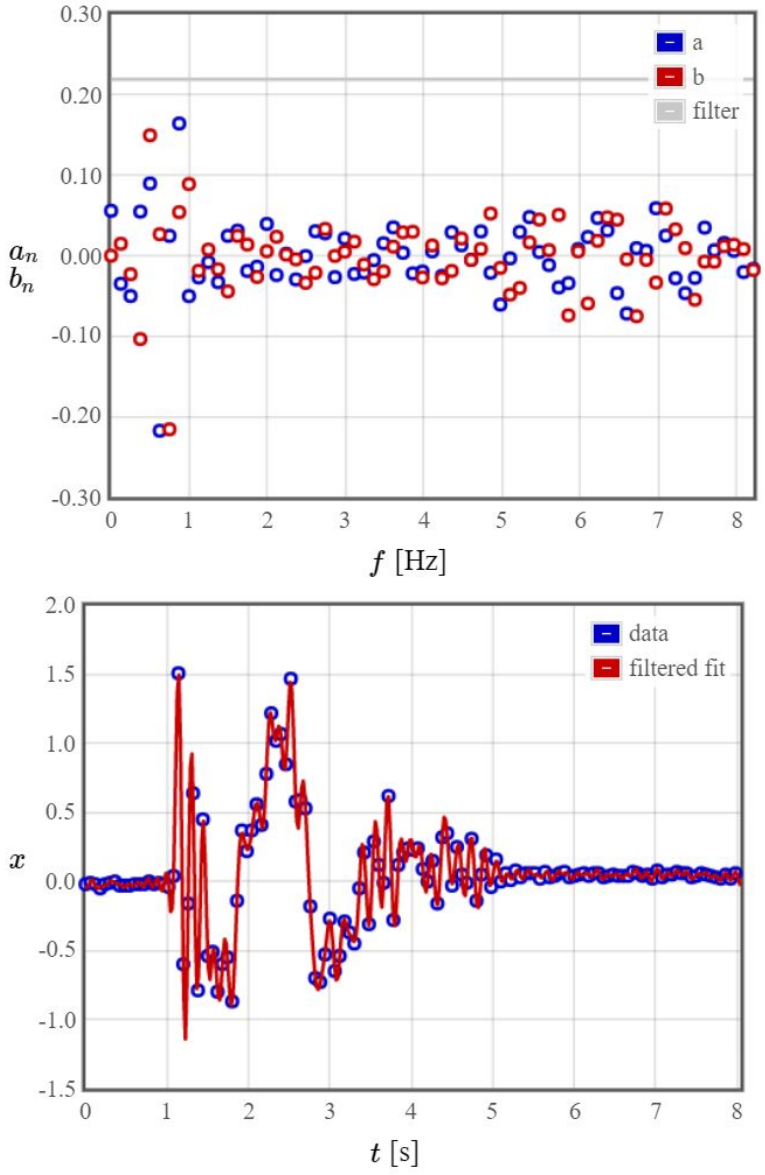


Fig. 17. Unfiltered predicted motion data

Figure 17 shows the collected data run through a filter with no parameters. Knowing the frequency range of collected motion from the initial fourier transform, the data was passed through this function again with a lowpass filter of 1.5Hz.

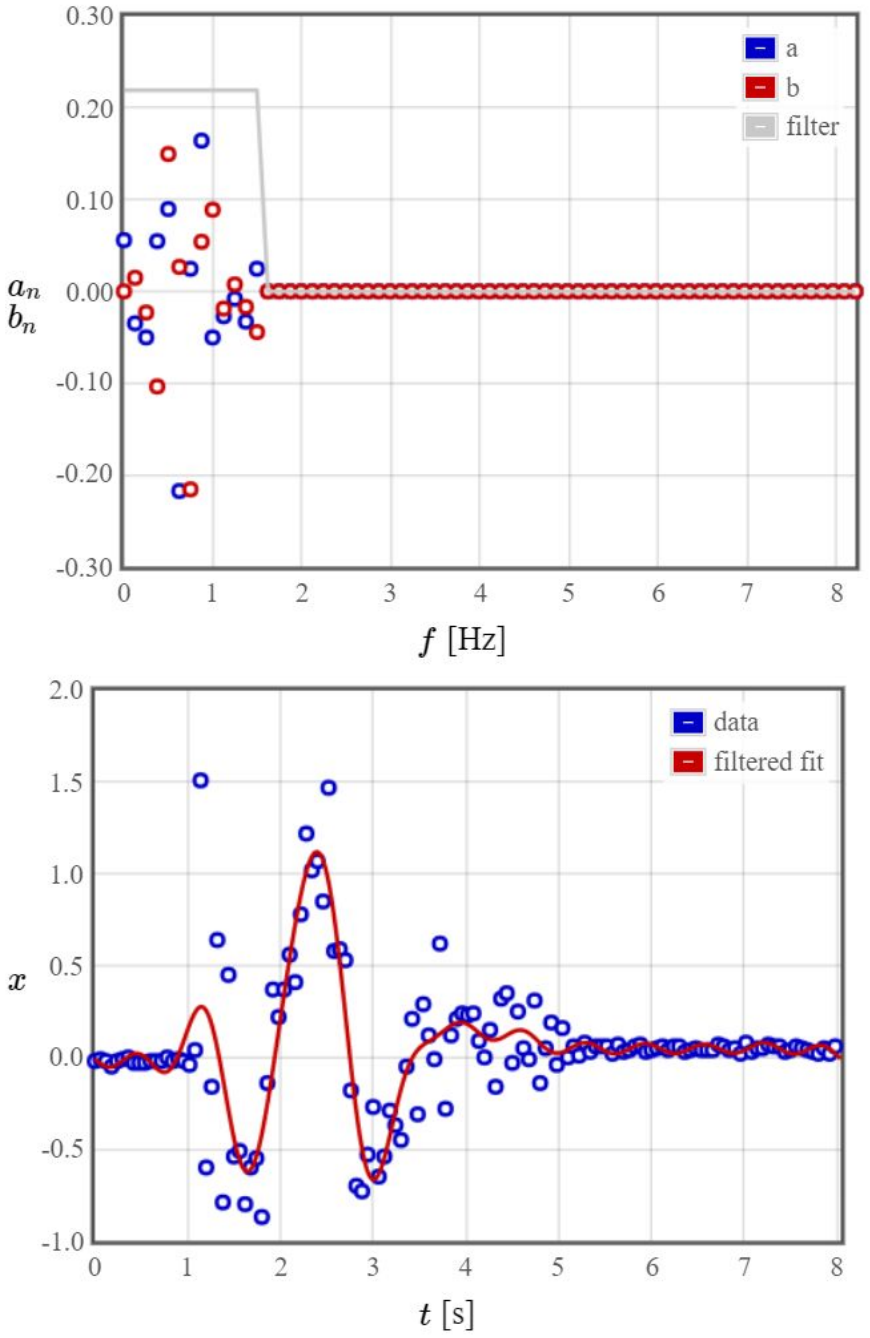


Fig. 18. Filtered predicted motion data

Figure 18 shows the same data with the filter applied to it. A comparison of this to figure **n** shows a significantly cleaner signal, which is at least visually indicative of expected motion.

While a cutoff value of 1.5 certainly produces better data than no filter, this value will be tuned as more data is collected.

The results of this test were implemented into filter functions using the scipy.filtfilt()[37] function, which results in a phase offset of 0.

The next step was to verify that the IMU was reading acceleration data correctly, or at least a range approaching correctly. To do this, a simple drop test was performed, and the acceleration data was recorded.

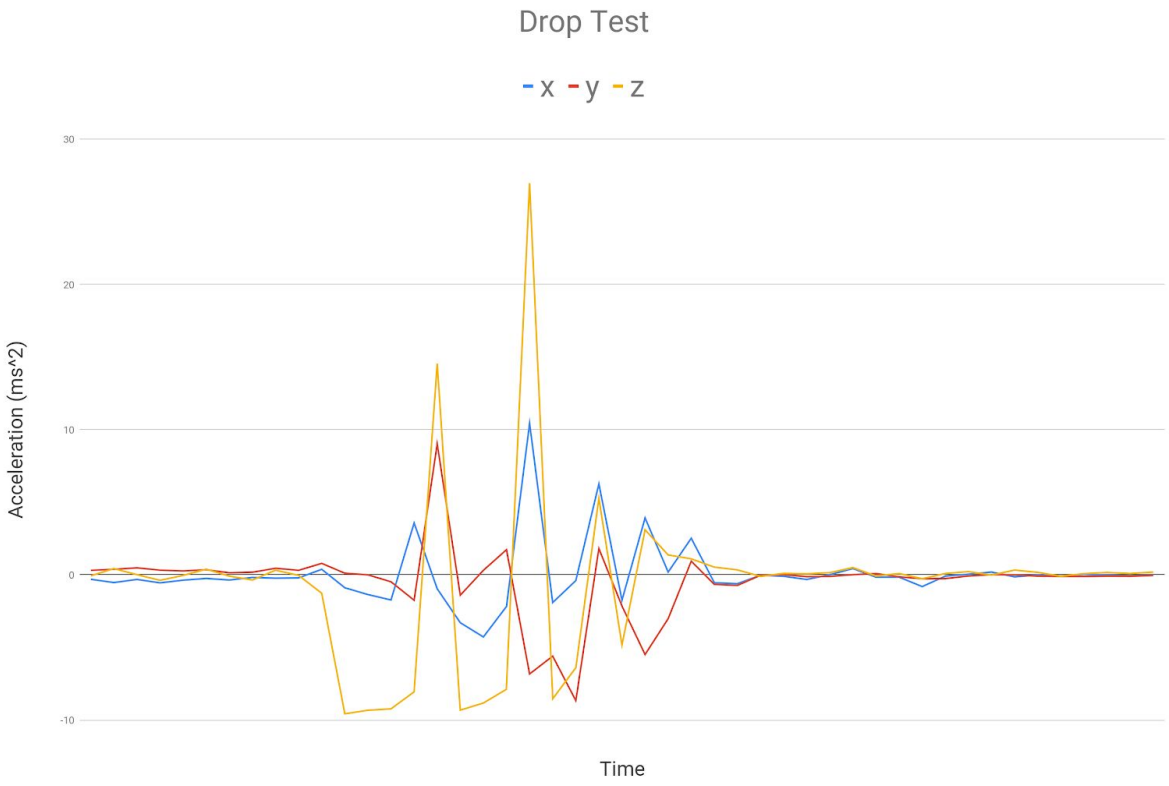


Fig. 19. Drop test acceleration data

[37] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html

The peak negative value shown in figure 19 is -9.57 m/s^2, which is close enough to earth's gravity of 9.8 m/s^2 that the discrepancy can be explained by the orientation not remaining completely flat during the fall. Now that the values read from the IMU are verified to fall within reasonable bounds of reality for large measurements, edge case information needed to be gathered. This test entails gathering stationary acceleration data to see what affect the application of the lowpass filter has on the background acceleration noise across a range of cutoffs. While stationary noise will be easy to filter out using a threshold, this noise will be present in all measurements, and therefore analysing it in isolated conditions is important for the overall design of the filter system.

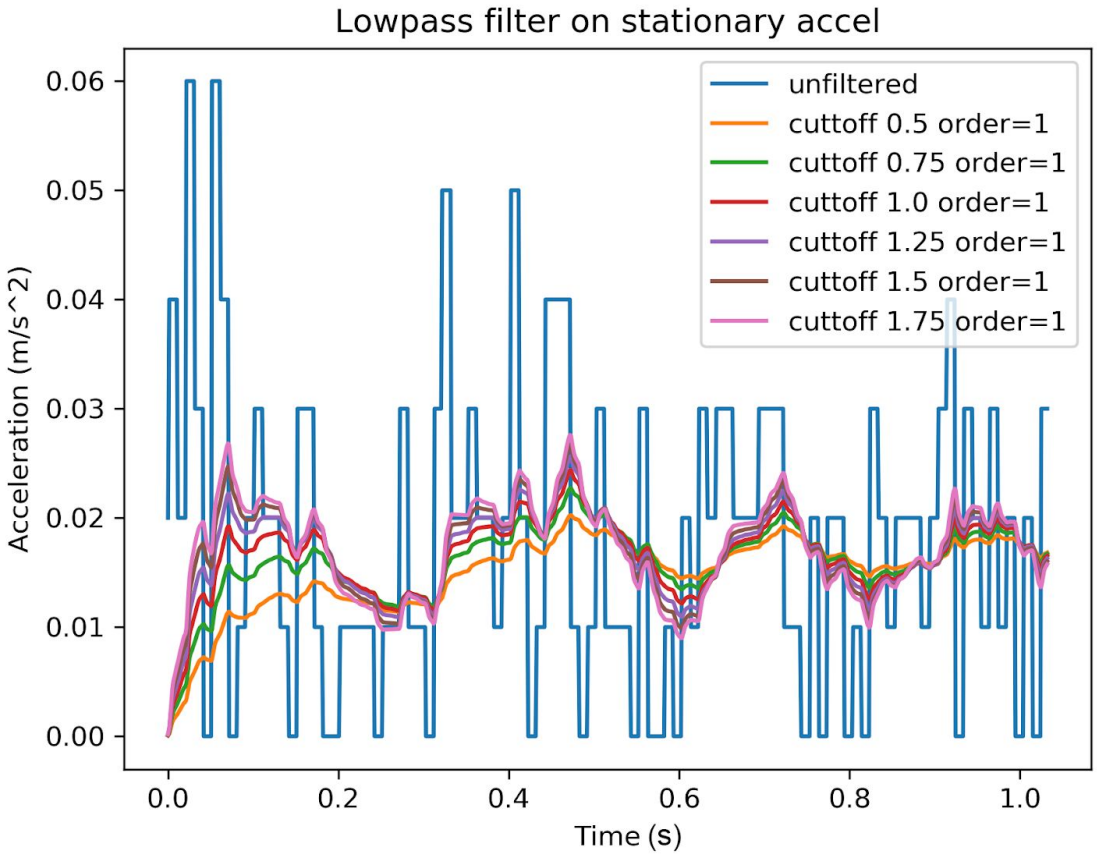


Fig. 20. Stationary Acceleration data with a range of filters applied, order 1

Though there is variance in the results shown in figure 20, none of it is drastic. This test was repeated for a higher order, and a range of orders.

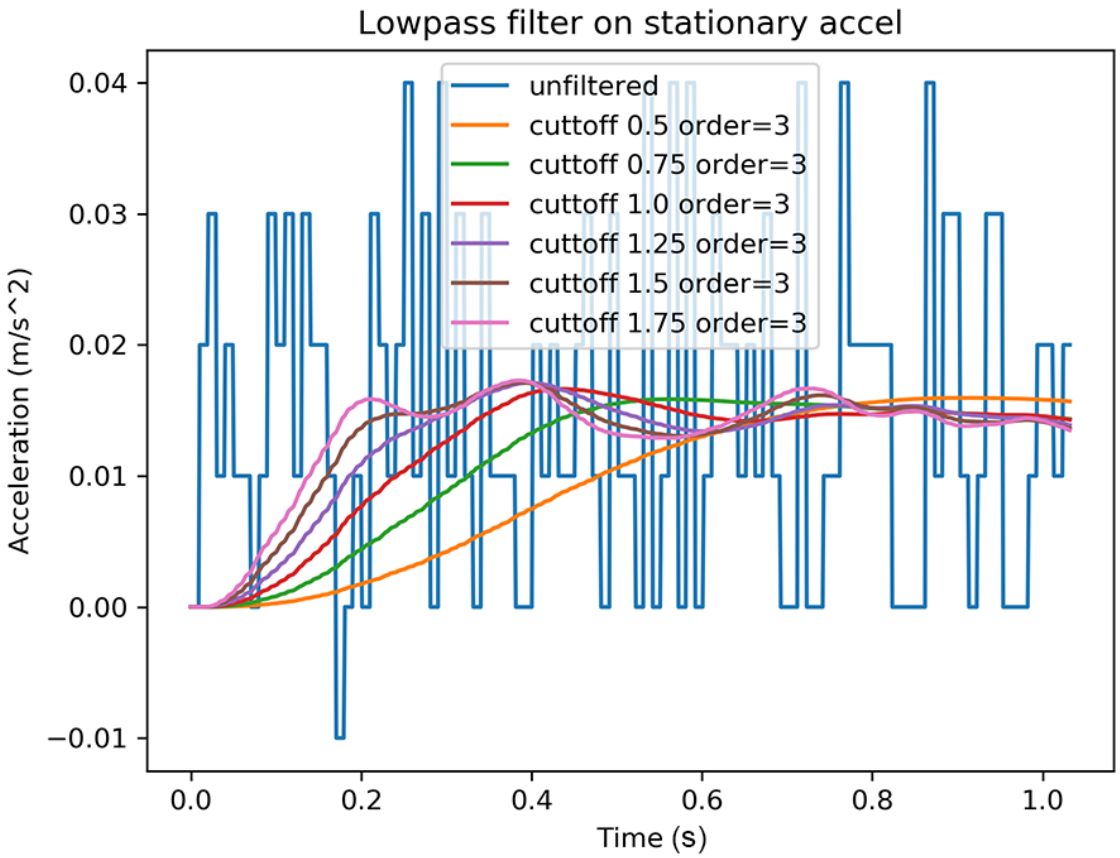


Fig. 21. Stationary Acceleration data with a range of filters applied, order 3

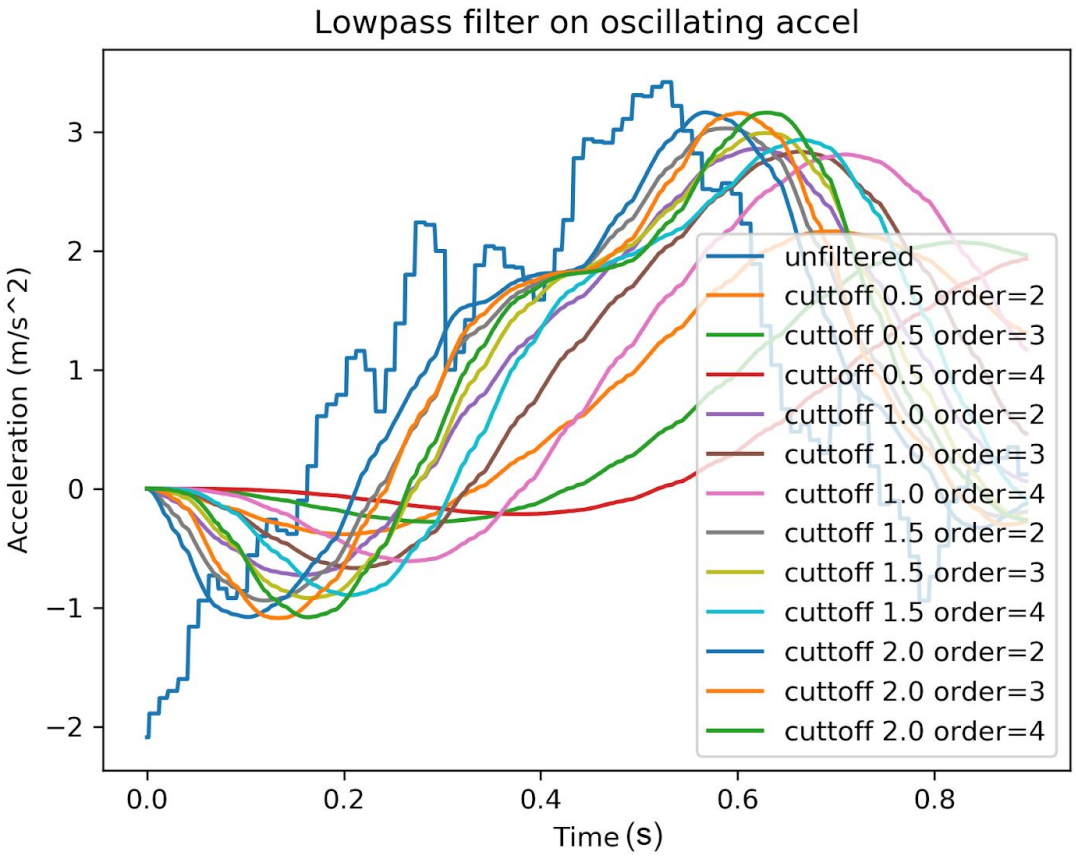


Fig. 22. Acceleration data with a range of filters applied, under motion

From the data shown in figures 20 through 22, the relationship between order, delay, and signal smoothness comes through. In figure 22, the acceleration data comes from the system under motion, which brings to light the phase offset resultant of higher order filters.

With this information gathered, the next step was to develop the code to collect and generate this data in the manor that will be used in the AUV itself. As the output of the IMU is instantaneous acceleration, and because integration over a single value is not possible, the calculation of displacement has to be handled in chunks. Hypothetically, integration could be performed over the entire range of acceleration every time there is a new value added, though

this would be very computationally inefficient and result in unneeded time resolution. The final system is as described in the design section under *Motion Data*.

To test this system, simulated sinusoidal acceleration values were fed into Motion Data as though they were coming from IMU handler.

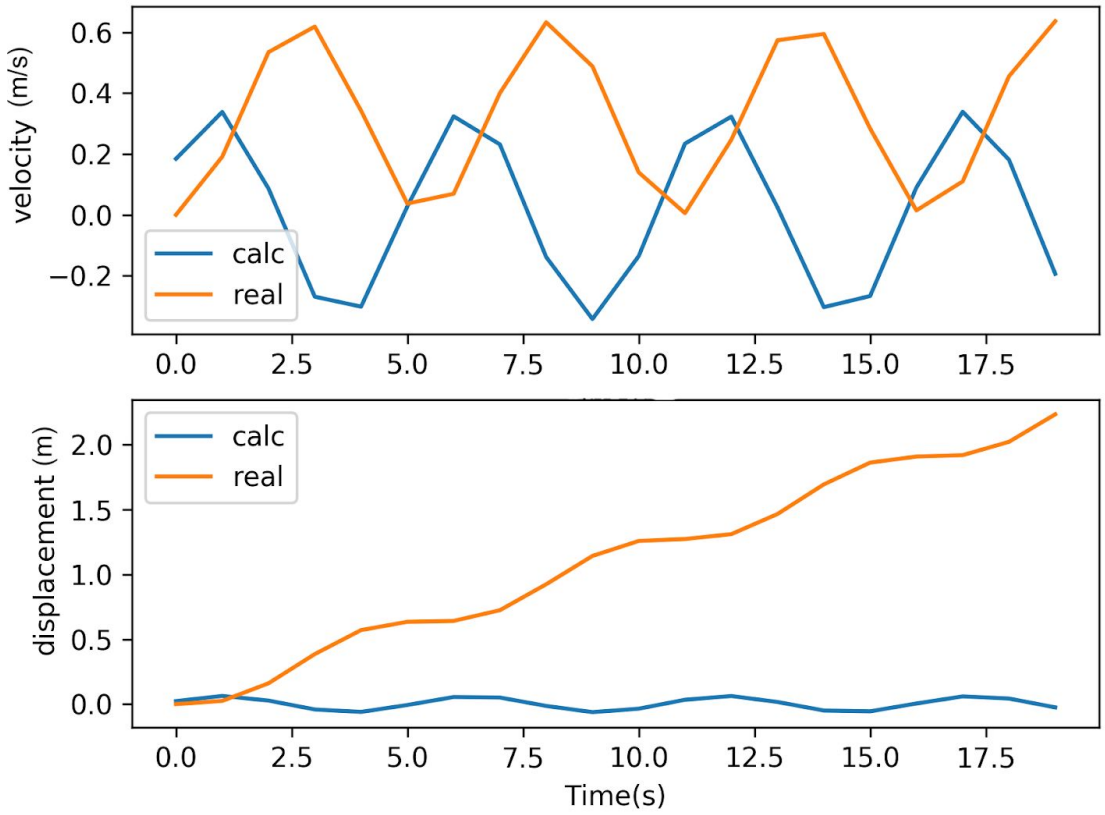


Fig. 23. Velocity and displacement from simulated acceleration data

As can clearly be seen in figure 23, the first results of this system were incorrect. After significant debugging and comparison of results with the "real" values, a much improved output was achieved, which can be seen below in figure 24.

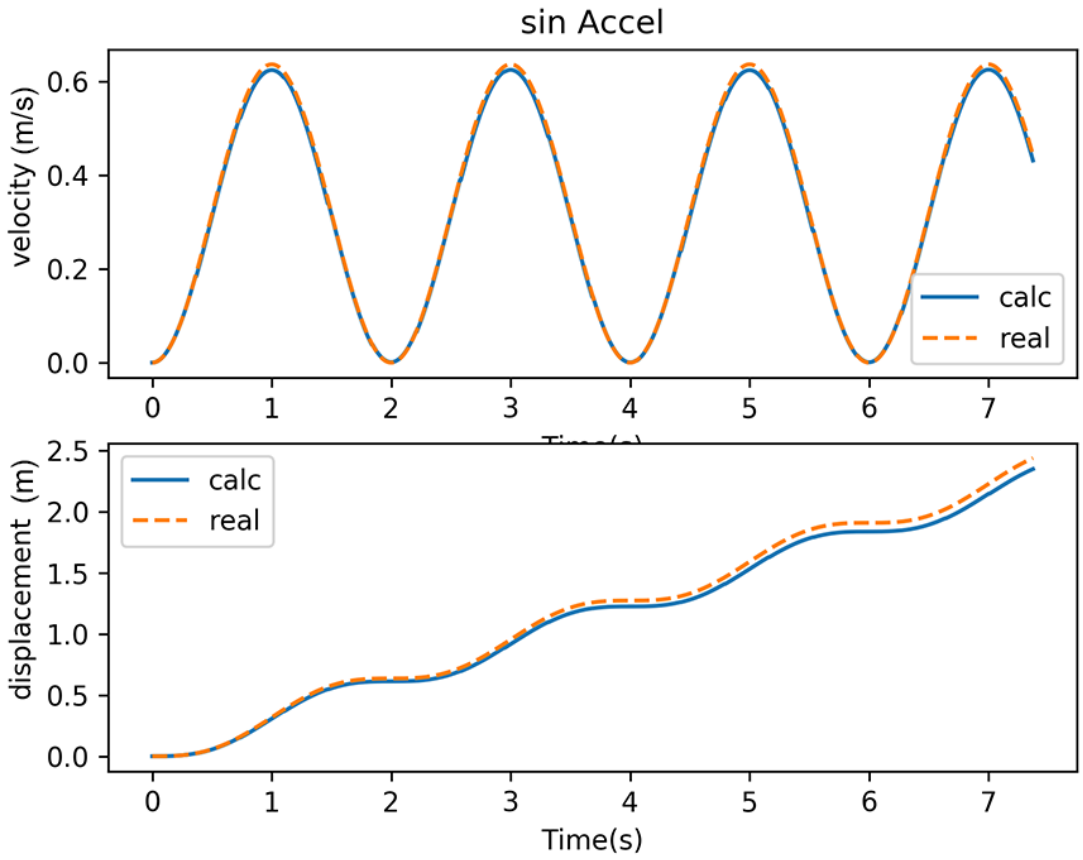


Fig. 24. Velocity and displacement from simulated acceleration data

This initially seemed correct, but close inspection shows that the values stray further from each other as time progresses. The source of this error was difficult to track down, but it turned out to be because acceleration needs to be passed along with the the immediately most recent value to have velocity align with the previous calculation.

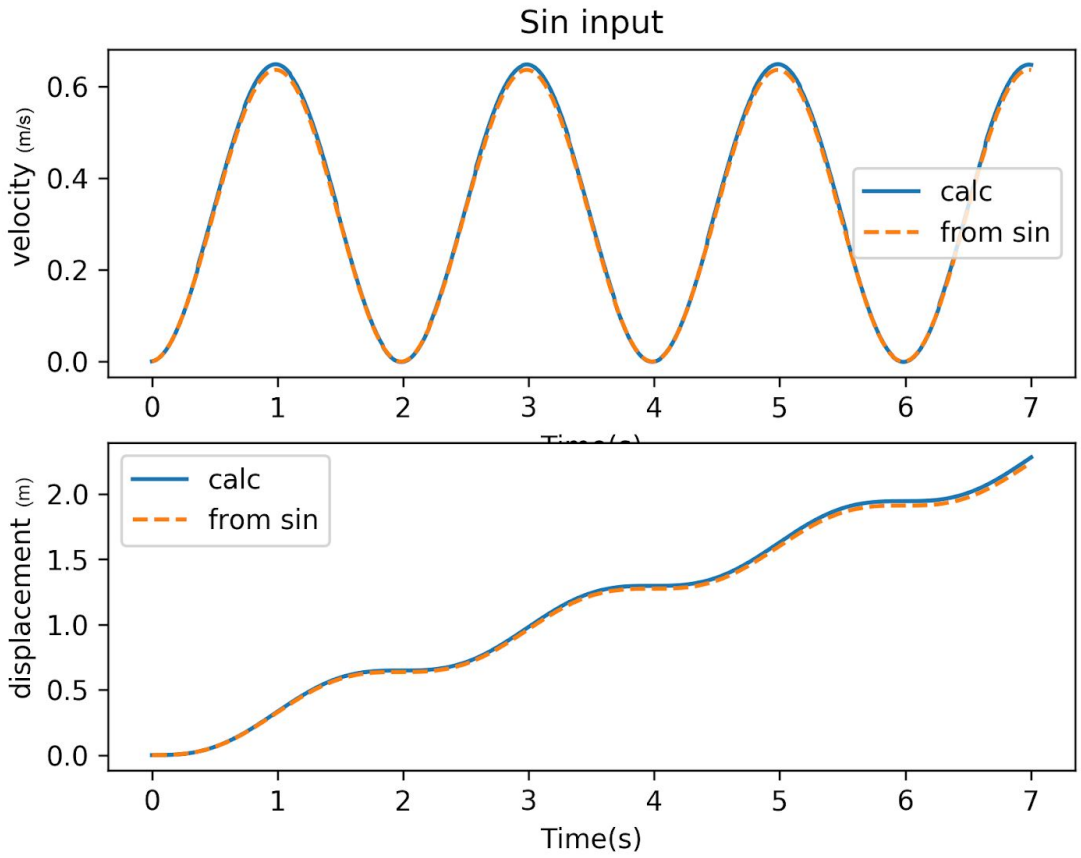


Fig. 25. Velocity and displacement from simulated acceleration data

Though the calculated values in figure 25 do not exactly match the the base values, further inspection showed that this discrepancy is due to the difference in integration methods used for the integration calculation.

Now that the algorithm was confirmed to be functioning correctly, it was time to work with real data. The simplest way to gather data was to move the system by hand while calculating the input, then graphing out the resultant displacement to compare with the

performed motion. As this data is coming from the IMU, it is in 3 dimensions, therefore the results of these tests are in 3D graphs.

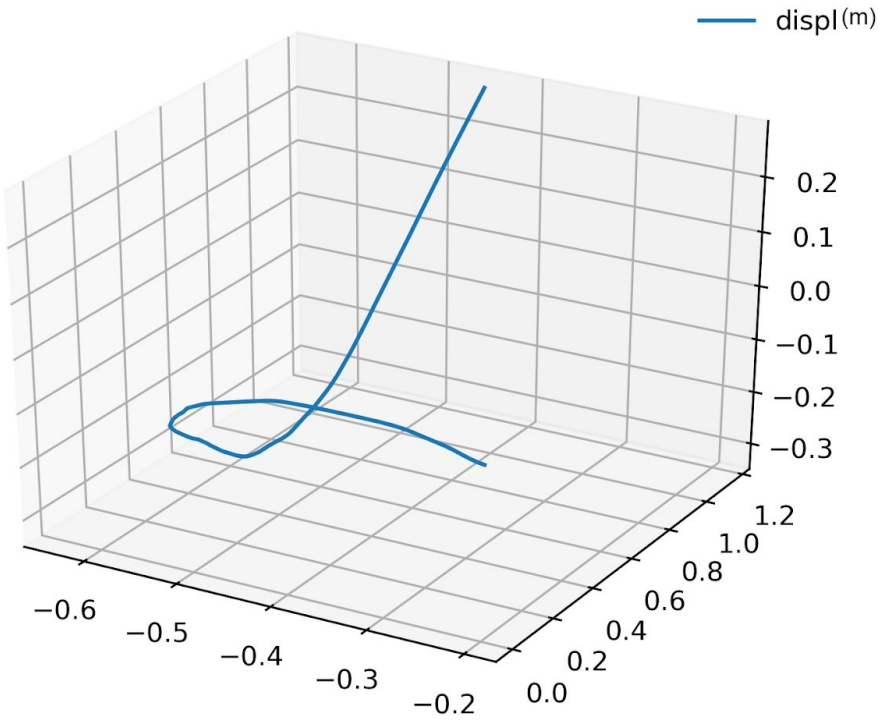


Fig. 26. Displacement XYZ (cm), vertical motion test

The test displayed in figure 26 is the resulting data from directly moving the system upwards and then terminating the test. Though this does display direction indicative of this motion, observation of the axis show that the displayed figure is misleading, and little to no aspect of the real motion was recorded. Due to limitations of the 3D graphing suite, mainly the inability to have defined axis dimensions, pursuit of this presentation method was rapidly discontinued,

though after going through several hand controlled motions, it was quickly determined that significant work on the filter would be required.

To evaluate how well the filter is functioning in order to improve it, repeatable known motion of the system were required. To enable this, the mechanical engineering team was asked to design and build a test rig, which would hold and spin the IMU+Raspberry Pi system smoothly, controlled by a continuous rotation servo.

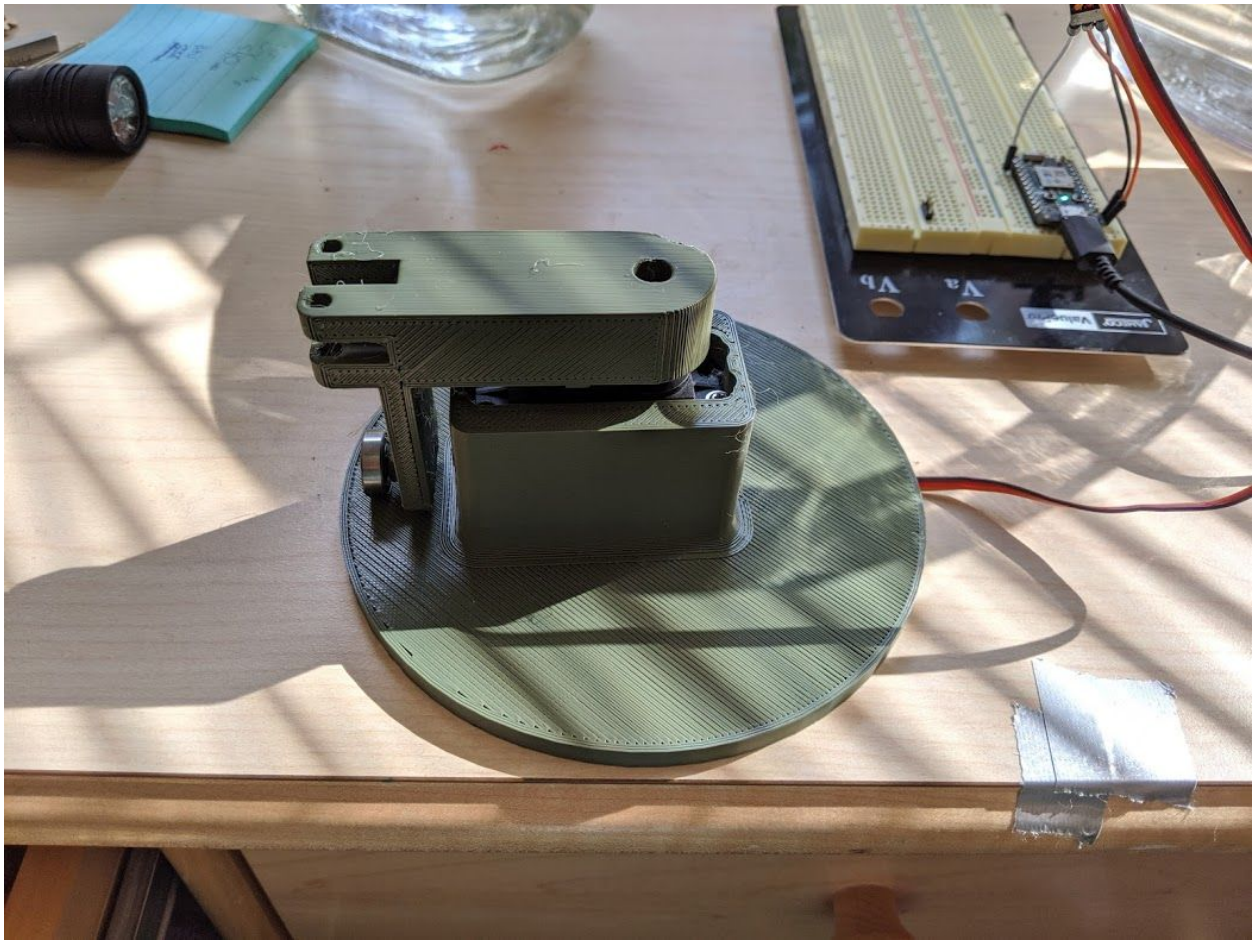


Fig. 27. Motion generating test rig

With the continuous rotation servo set at a known PWM value, the RPMs were measured, and from this the tangential velocity of 0.713m/s was calculated. From $A_n = \frac{Vt}{r}$ a normal acceleration of 5.09 m/s^2 was determined. This is displayed in the IMUs measurements as well.

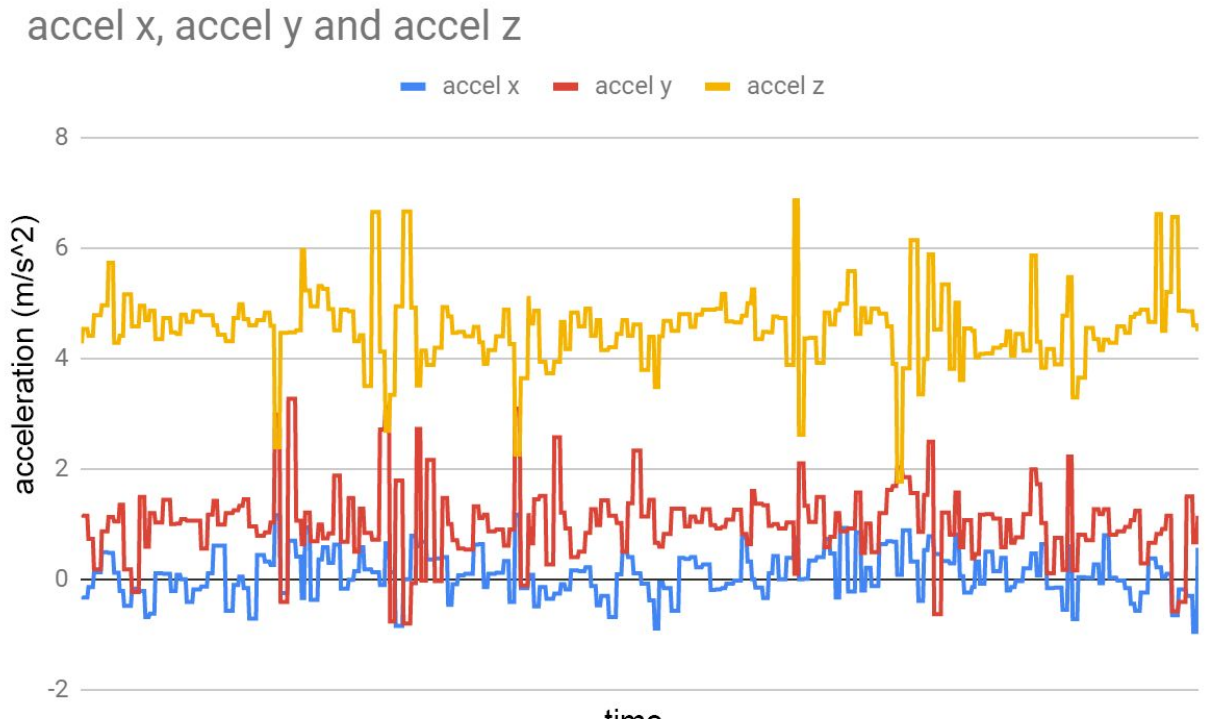


Fig. 28. Spin test acceleration values

Now that there was a method for generating values to compare against which included the IMUs errors, the next step was to use this comparison to optimize the filter parameters. It was decided to use regression algorithms to hone in on the ideal parameters. The way this works is the regression algorithm is provided with a function to pass parameters to, and which will return a score. The algorithm attempts to find the input parameters which produce the lowest returned score from within a designated range. The function that this algorithm is given initializes Motion Data with the parameters it has been given, and gathers data for a set period of time while under

motion in the test rig. Once data has been collected for said period of time, it sums the RMSE[38] score over each value in the gathered and real datasets. This summation is then returned the the regression function. The parameters that this regression function is tuning are range, cutoff frequency, and filter order. The range that the algorithm is allowed to vary over was informed by the initial filter testing, and narrowed down over time based on the results of previous regression runs.

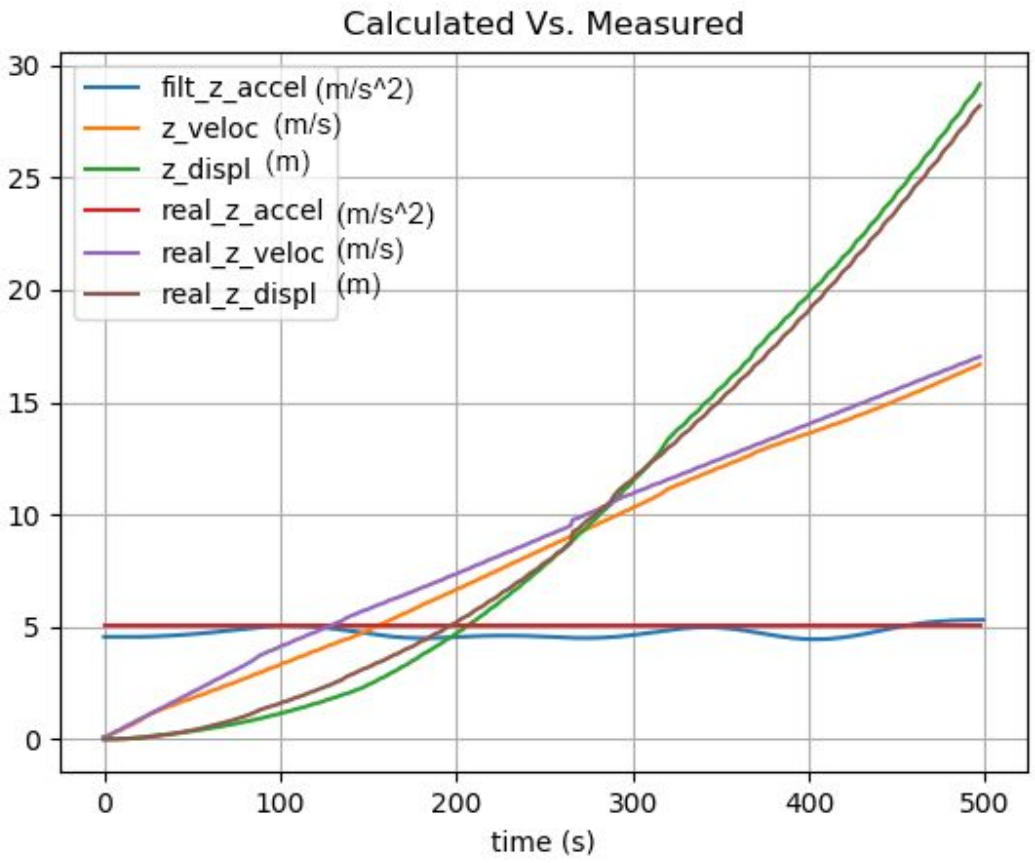


Fig. 29. Spin regression results

[38] https://www.statisticshowto.datasciencecentral.com/rmse/

Figure 29 shows the gathered data, run through a filter with the regressed parameters, graphed on top of the real data. These results appear very promising, however this is for linear acceleration, which is significantly simpler than nonlinear.

The test rig was next modified to generate a normal acceleration, which upon integration would produce $sin(t) + 1$. This is analogous to velocity, though because it is generated solely from the normal component of acceleration, it is not a real velocity, even though the IMU will not be able to differentiate. This function will be referred to as velocity to simplify matters, and from it, virtual displacement is calculated, which will just be referred to as displacement.

"Real" values for each of these were generated, and can be seen in the following three figures.

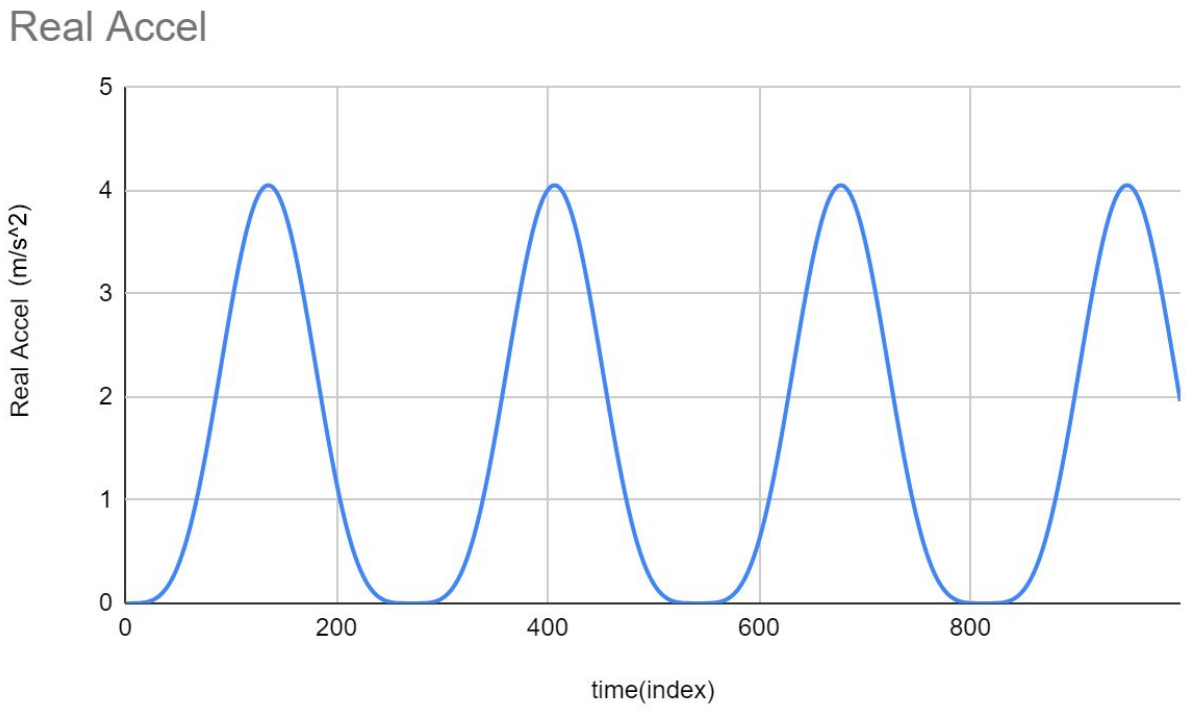


Fig. 30. Calculated acceleration value for scoring

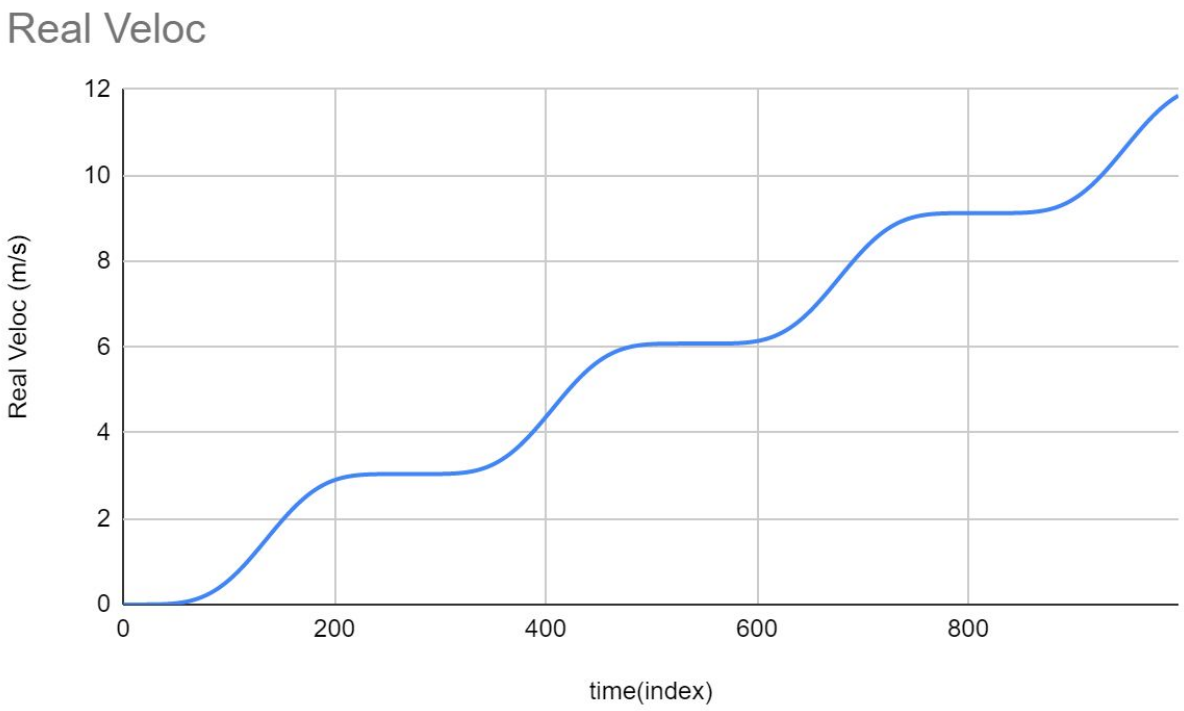


Fig. 31. Calculated velocity value for scoring

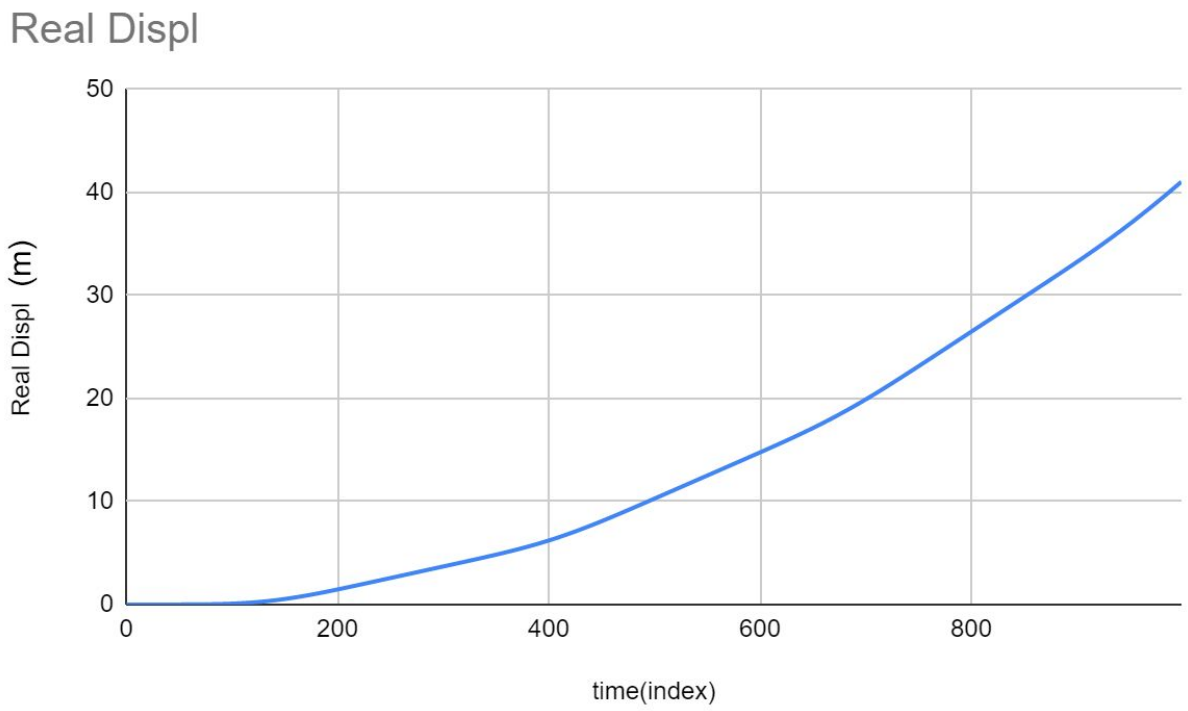


Fig. 32. Calculated displacement value for scoring

Using the filter parameters provided from the linear acceleration regression, an initial graph of measured and filtered acceleration was generated.

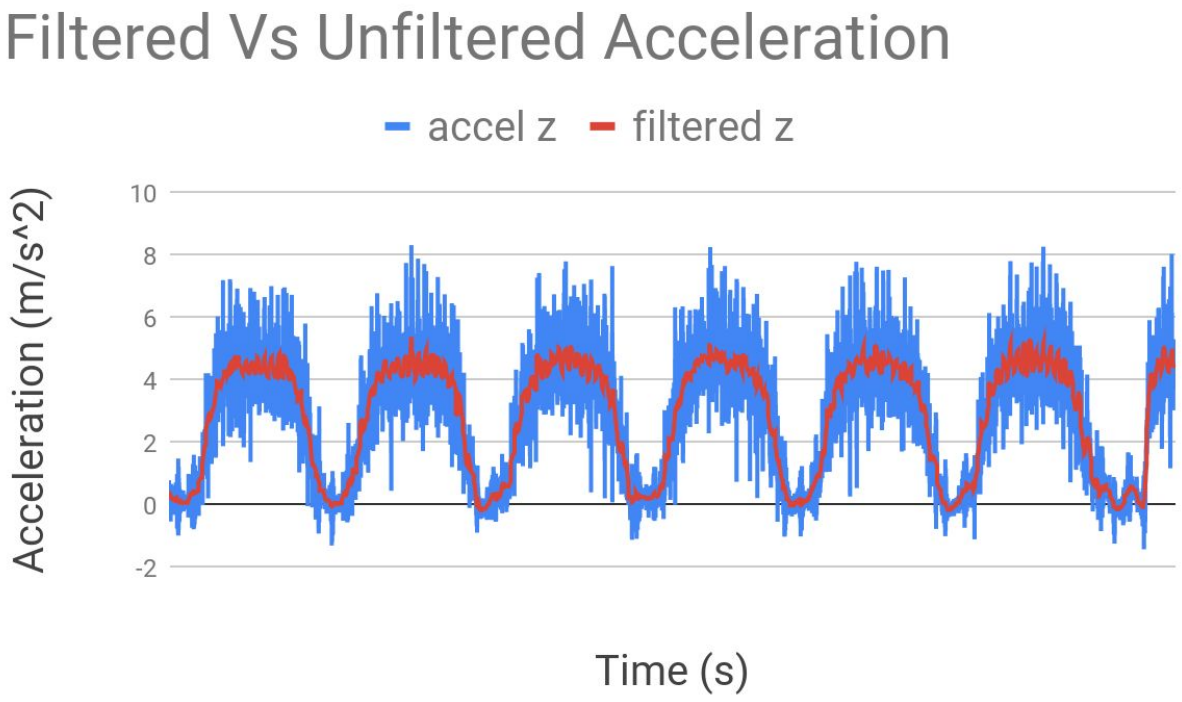


Fig. 33. Nonlinear spin acceleration

While the filtered acceleration in figure 33 is certainly not an ideal representation of what the acceleration data should look like, it is a lot better than the unfiltered values.

The regression code was modified to generate a graph for each epoch, and code was added to start the nonlinear spinning at a specific time to synchronize the measured values with the starting point of the calculated values. The graphs allow for human analysis of the results of each test, regardless of the score, and insights made from this were used to tune the range of parameters the algorithm optimized over. For this human analysis it was much easier to compare

values using index as the x axis, so each x value corresponds to 0.007s. Initially results appeared promising.

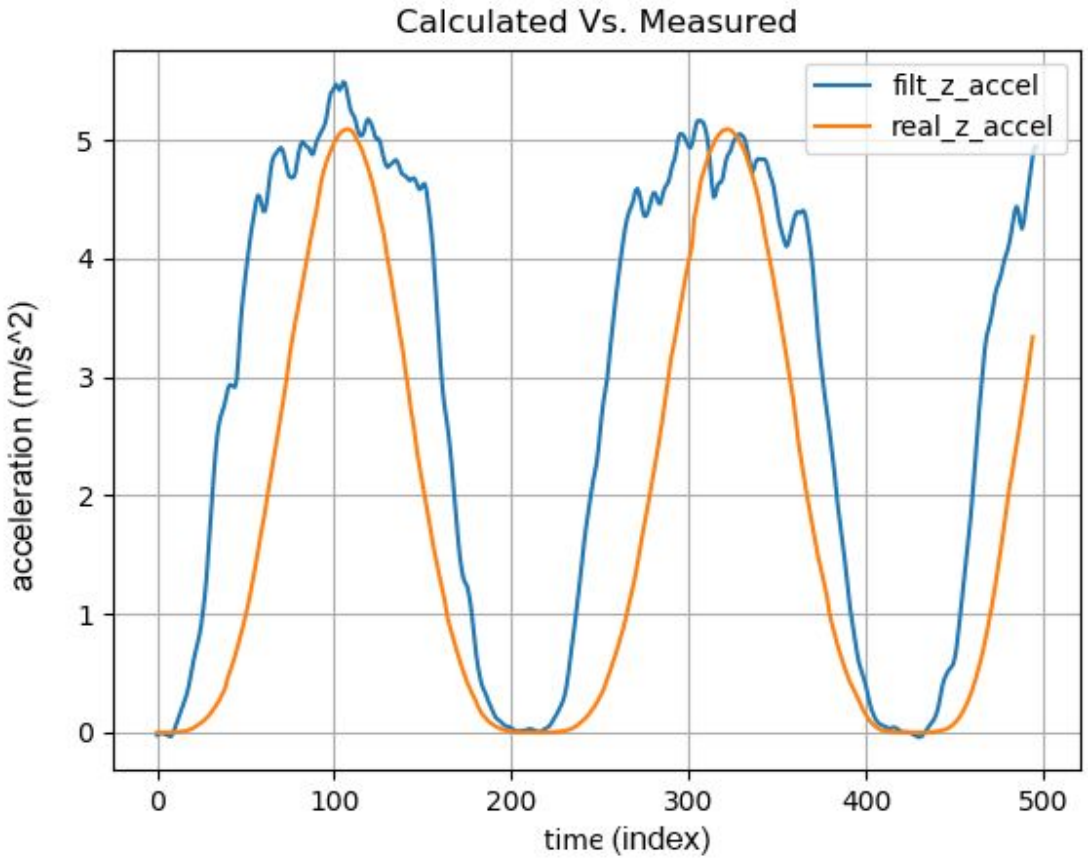


Fig. 34. Calculated Vs. measured & filtered acceleration

However, as testing progressed, issues with synchronization began to show up.

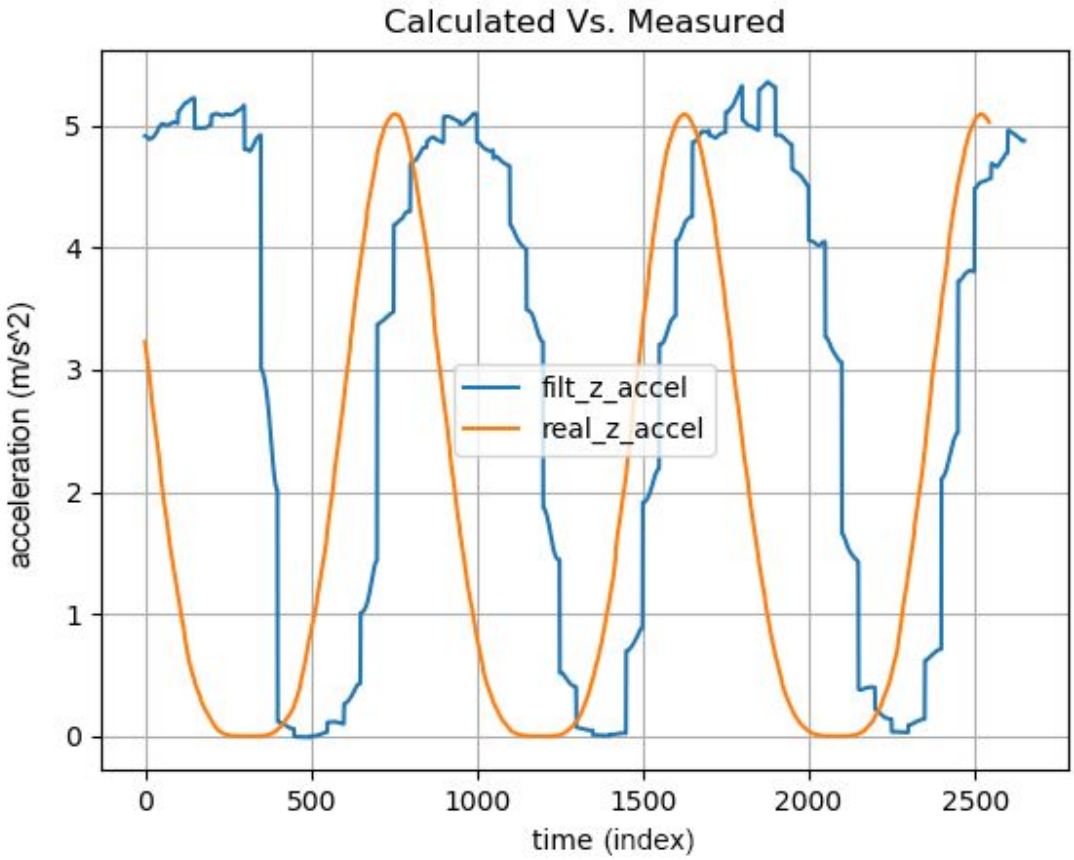


Fig. 35. Calculated and measured acceleration, synchronization issue

While the synchronization distance in figure 35 may not seem particularly large, due to the scoring method used, slight differences compound greatly. Further testing and investigation lead to the conclusion that a lagging network connection was causing discrepancies in the start time of the test rig.

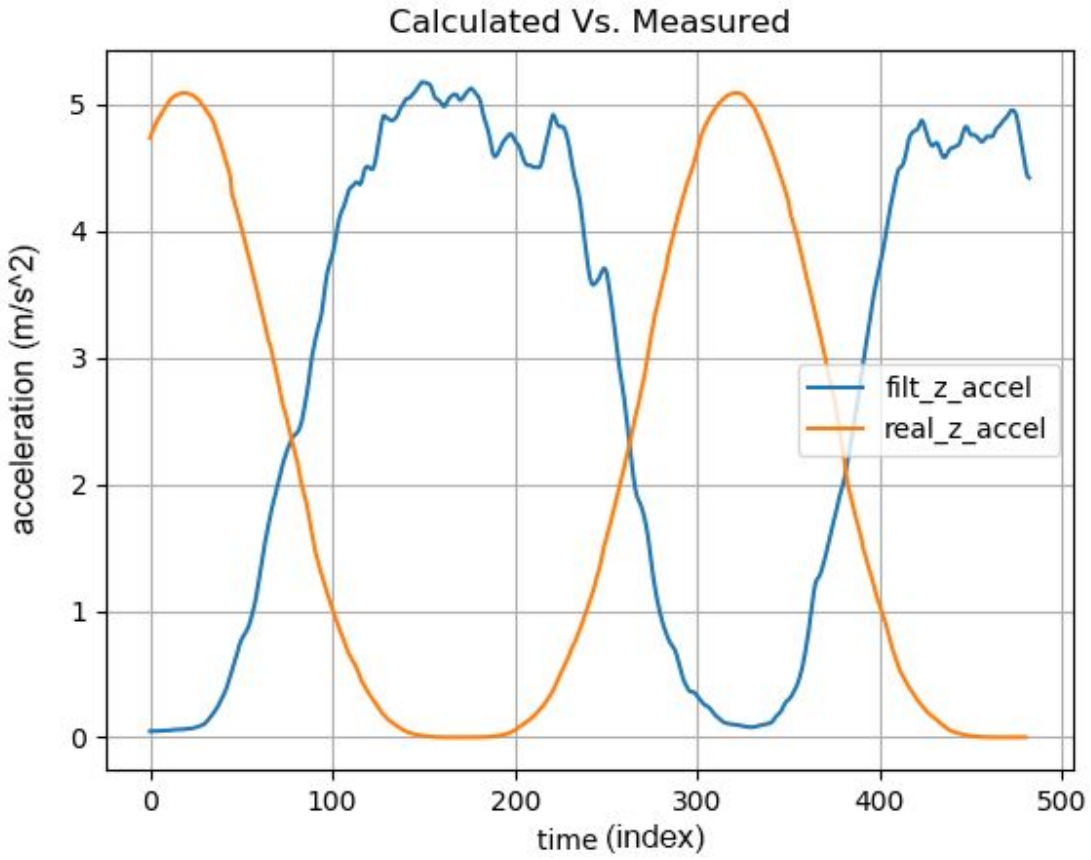


Fig. 36. Calculated and measured acceleration, significant synchronization issue

Said lag was sometimes great enough to put the signals completely out of phase with each other, as can be seen in figure 36. This produces a horrible score, but as the algorithm is only aware of its parameters, it ruins the regression. As the start signal requires the connect to an external server, and direct interfacing between the Pi and servo controller is impossible due to the Pis spinning, there was no way to concretely eliminate the lag itself.

The solution found for this issue was to make the servo function continuous, so there was no initialization signal, and have some onboard method of synchronizing the signals after they are collected, but before they are scored.

To do this, a function to detect the valleys in the filtered acceleration was implemented, and used to synch to the valleys of the real acceleration.

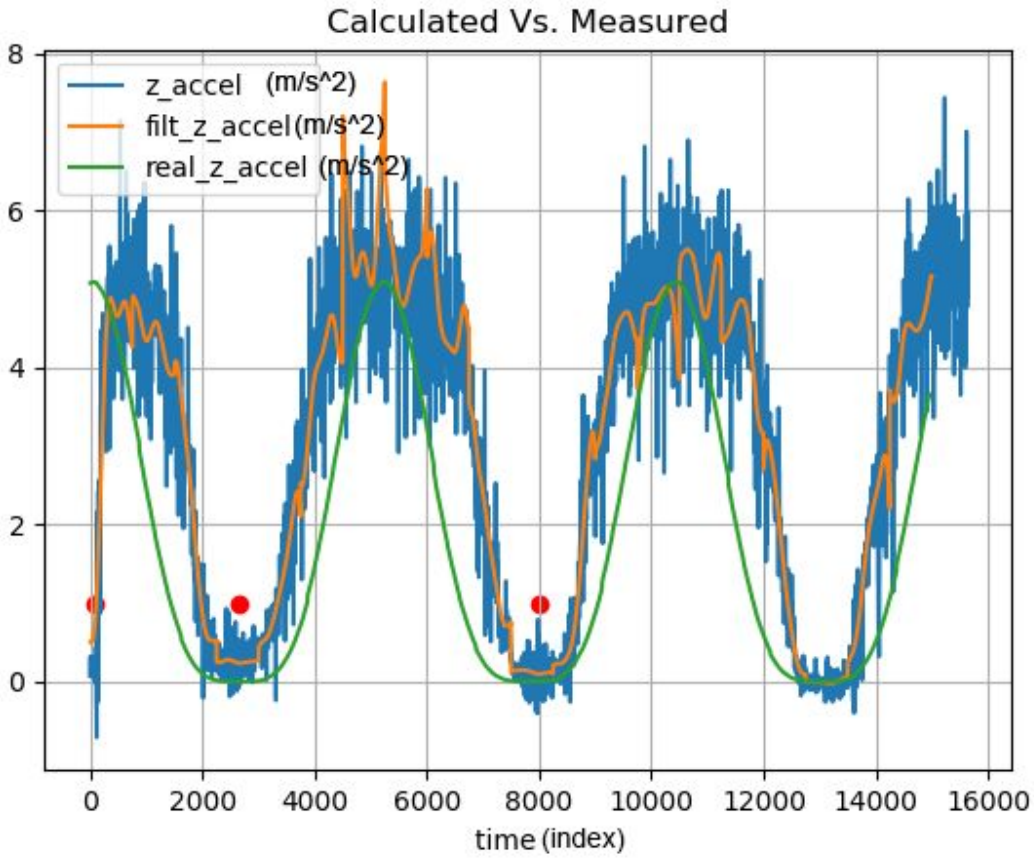


Fig. 37. Calculated, measured and filtered acceleration, valley based synchronization

The red dots in figure 37 represent the location of the located valleys, and it can be seen that the real acceleration signal is adequately synchronized with the filtered one. This figure also displays large spikes in the filtered data, which signifies that a cutoff value of 1.5 is too high, allowing

insight which can be applied to the parameter ranges. Unfortunately, the valley detection algorithm was not as robust as desired, and resulted in situations where no valleys where detected for no discernable reason.

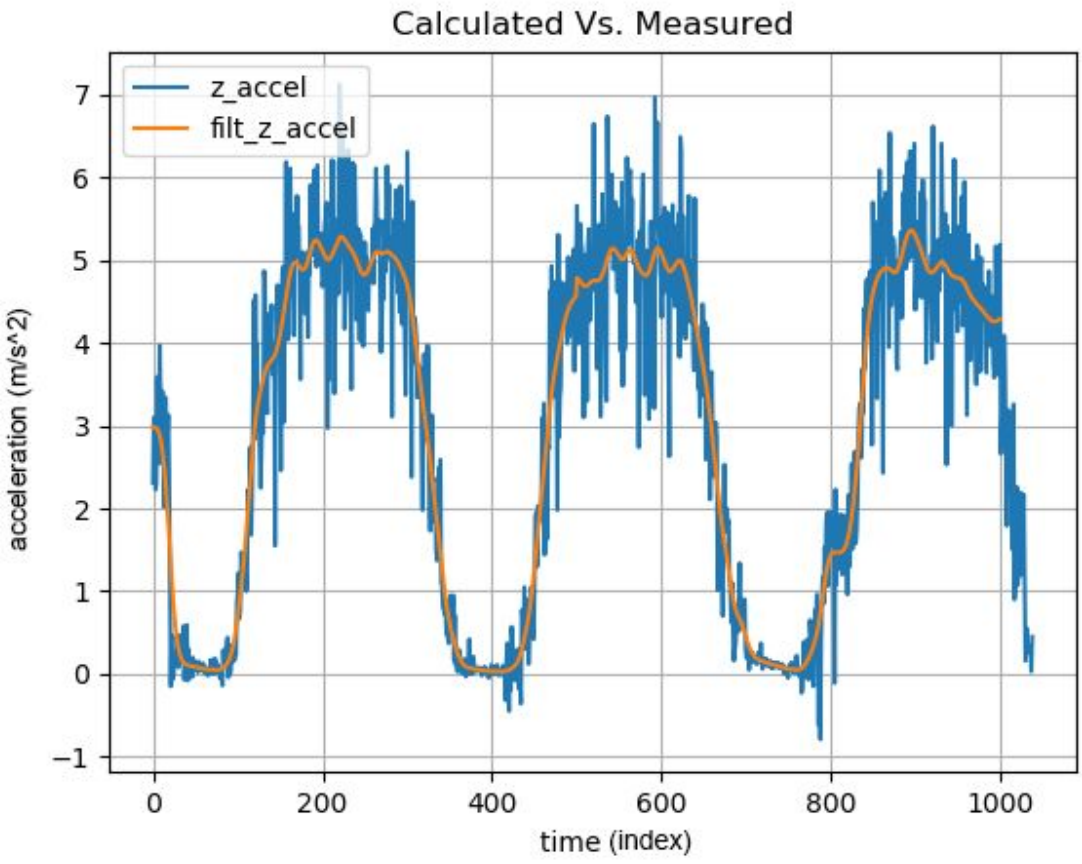


Fig. 38. Calculated and measured acceleration, valley detection issue

No valleys were detected in the filtered acceleration signal shown in figure 38, even though the data appears to be clean enough for the detection to work flawlessly. This error was too difficult

to reproduce for it to be efficient to try to directly debug it, so in the case where no valleys were found, the smallest acceleration value would be assumed to be a valley, and then used to synchronize the calculated real acceleration.

After running the optimization algorithm and honing of the parameter ranges over the course of a week, the resulting filtered values began to consistently display desirable results.

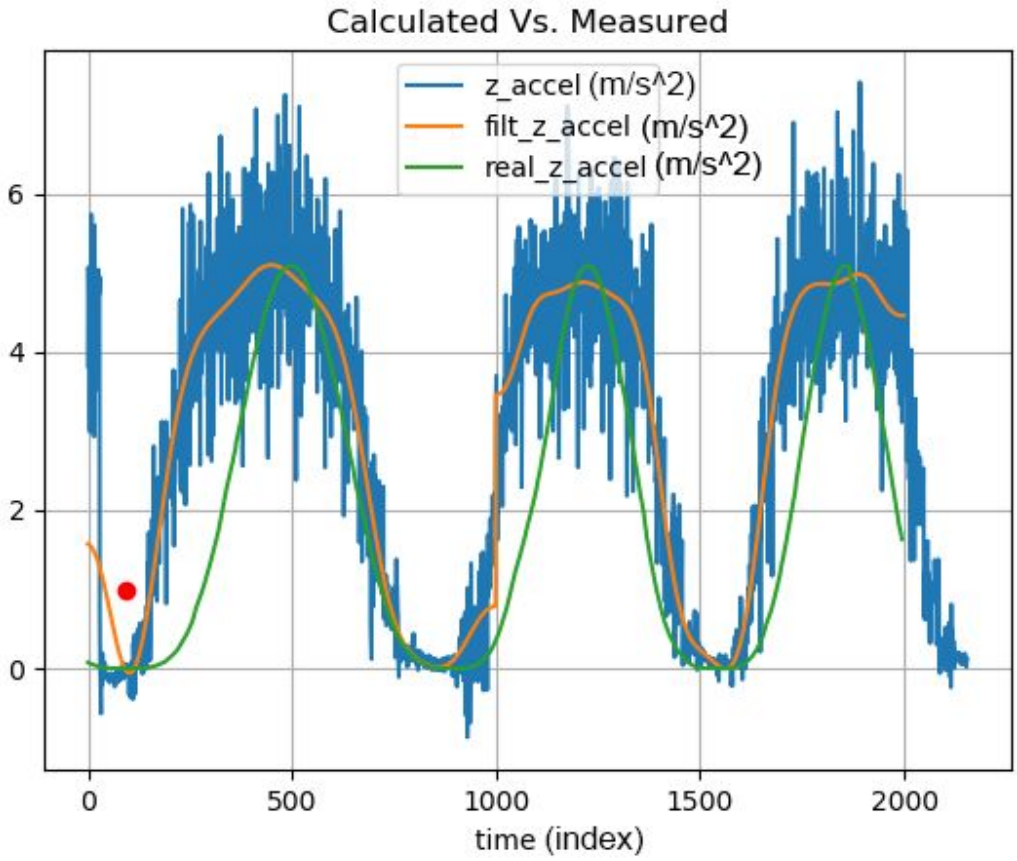


Fig. 39. Calculated, measured and filtered acceleration, good looking results

Once the selected parameters were constantly coming up with low scores, the velocity and displacement values were added to the graphs.

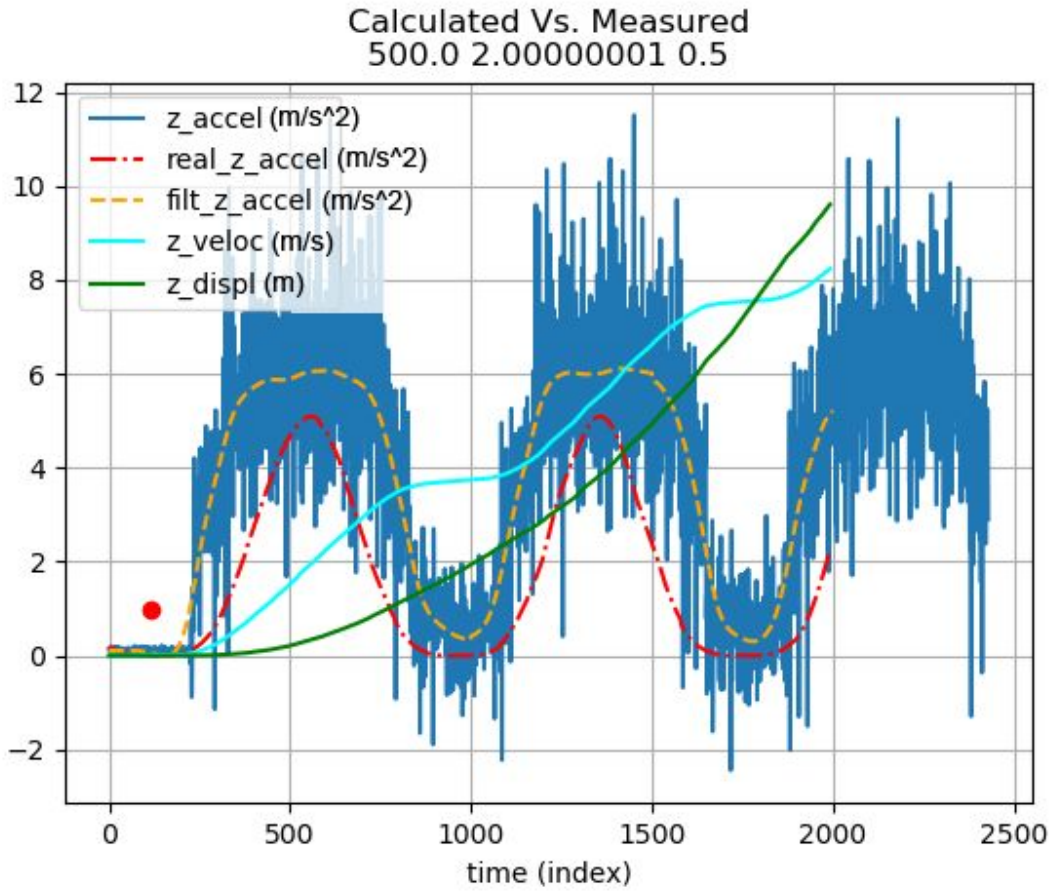


Fig. 40. Measured, real and calculated values from filter

If you compare the calculated values in figure 40 with figures 30 31 and 32, they all appear to match fairly well, which was taken as an extremely promising sign. The filter parameters were set to range=500, order=2, and cutoff=0.5, and the system was set to be stationary while it gathered data.

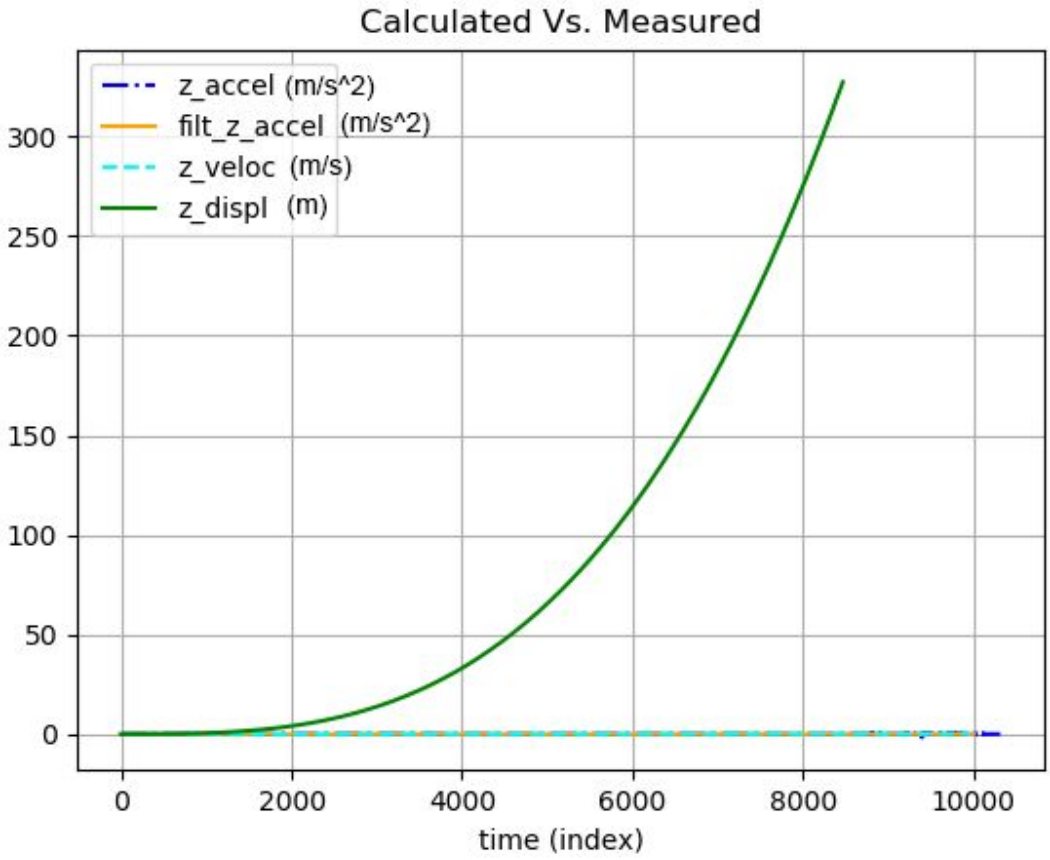


Fig. 41. Measured, real and calculated values, stationary test

As can be seen in figure 41, the displacement value increases exponentially fast, which is not desirable behavior. This implies that the driftrate/noise level in the IMU is high enough to cause this level of imprecision. To alleviate this problem, the maximum acceleration value from completely stationary data collection, 1 m/s^2 , was implemented as a threshold, so any acceleration value below that would not be registered.

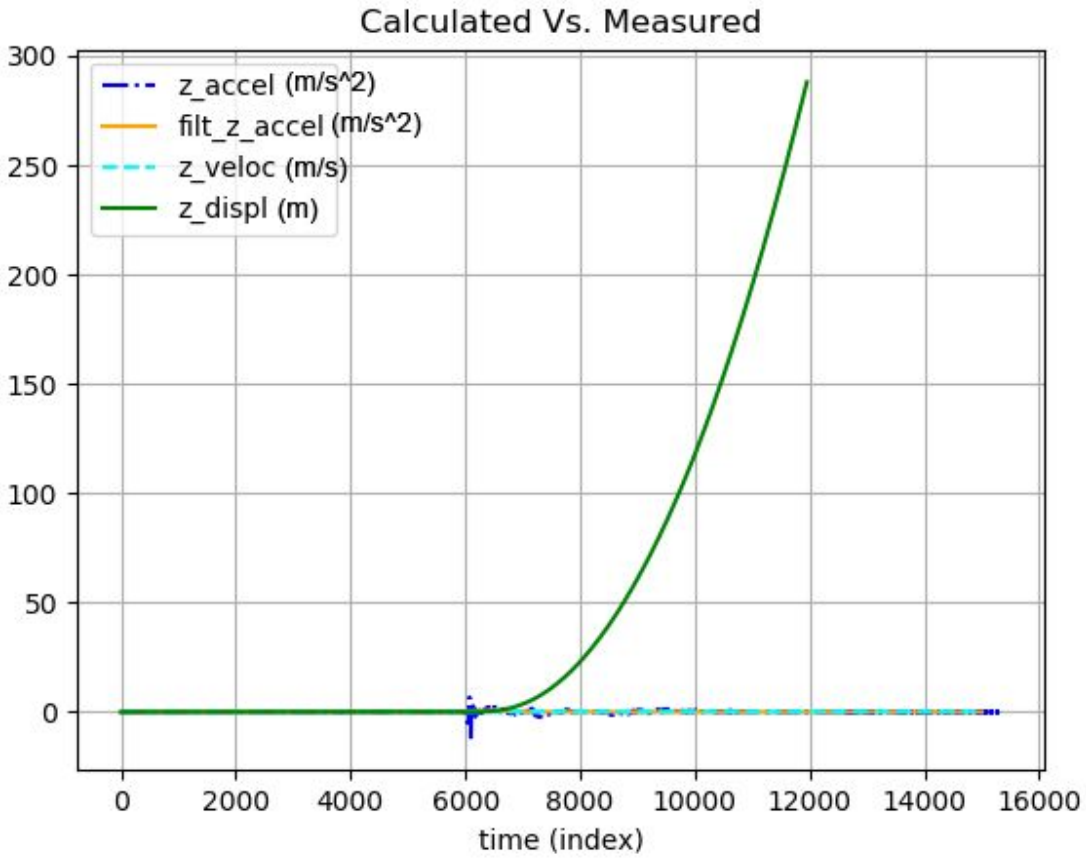


Fig. 42. Measured, real and calculated values. Stationary, then moved, test

Figure 42 shows a test where the system was left static for a period of time, then picked up. As can be seen, there is no drift until the system is moved, at which point the displacement values begin to exponentially increase.

Further testing showed that the exponential incrementation could be influenced, at least temporarily, by initiating motion in one direction, and then changing to the opposite direction, but instead of returning to 0 or representing a reasonable displacement value, the exponential change continues. This effect can be seen in figure 43 below.

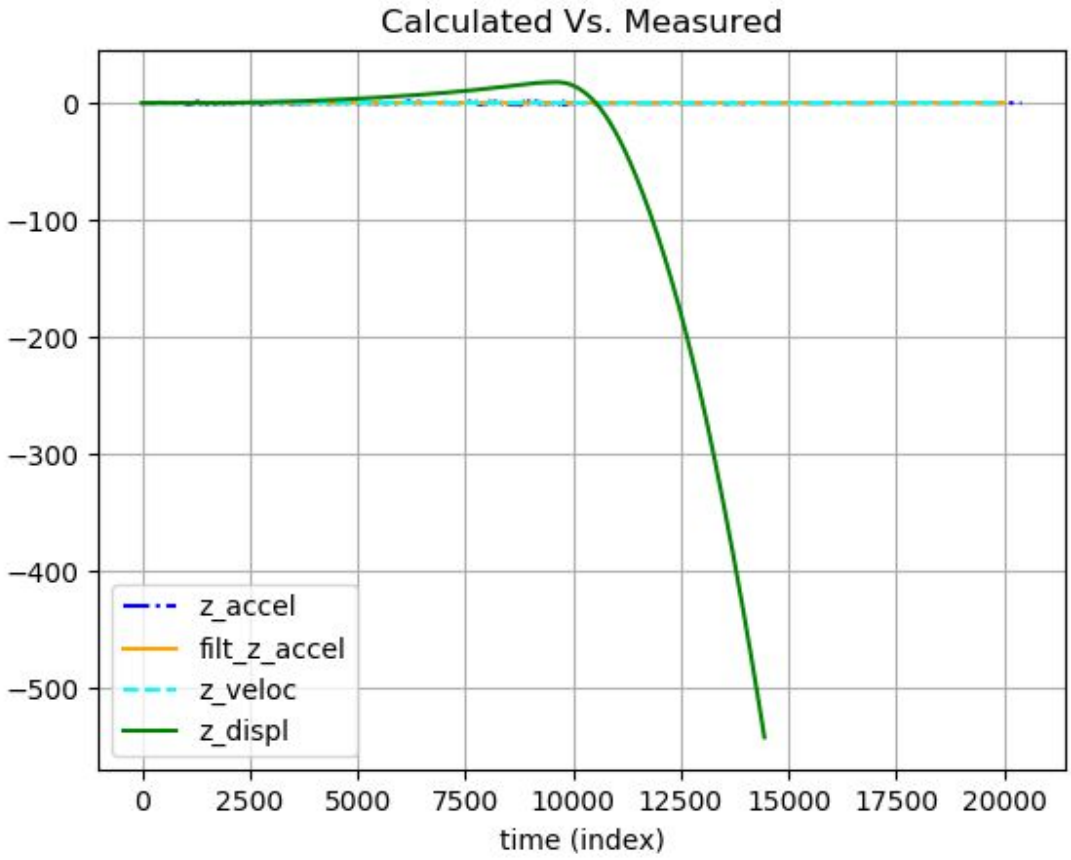


Fig. 43. Measured, real and calculated values. Reversed motion test

This behavior indicates sensor drift, which, according to preliminary research, is nearly impossible to correct without a baseline to reset to.

Further testing and development on this will continue for the duration of 2019, and if the results are promising, into 2020. However, if it is deemed not feasible to implement this system with the timeframe of this project, alternative solutions will be implemented.

Heading Change Control System (performed by Jacob)

One of the most basic behaviors this sub must be able exhibit is changing it's heading to a target heading. Therefore, some control system must be capable of deciding what PWM values to send to the actuators in the system to achieve this heading. The optimal control system for this application would include an environmental model and possibly some fuzzy logic to handle the continuously changing system dynamics. The design of a control system of this nature, however, is beyond the scope of this project and thus an alternative controller is tested; specifically, a PID controller.

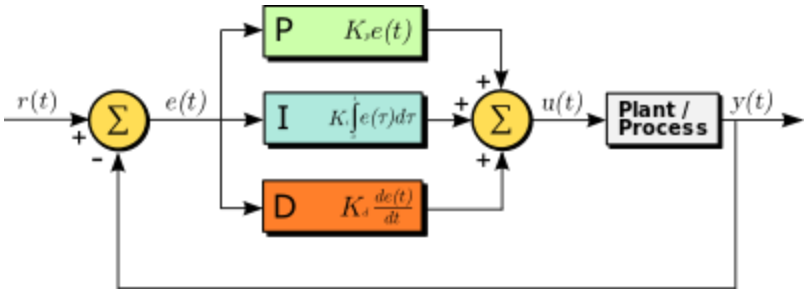


Fig. 44. PID controller diagram

PID controllers are relatively simple to understand and easy to implement. The output of the controller is a function of the deviation from a process variable (PV) and a desired setpoint (SP), called $e(t)$ in the above figure. Using weighted proportional, integrative, and derivative factors of this function, the output is computed by summing these three respective terms. The output of the PID controller, defined as $u(t)$, is expressed mathematically below.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

Fig. 45. PID controller output equation

In the above figure, K_p represents the weight of the proportional term, K_i represents the weight of the integral term, K_d represents the weight of the derivative term, and $e(t)$ represents the error function once again, equal to $SP - PV(t)$. The proportional gain linearly scales the error function, which largely determines how reactive the controller is and how often/severely it overshoots. The integral gain sums the instantaneous error over time, and functionally accelerates motion towards the setpoint. Finally, the derivative gain is computed through the slope of the error function over time and predicts system behavior. Due to unpredictable environmental disturbances such as waves and strong winds, the derivative gain may not be useful in practice.

While PID controllers are not ideal for complex physical systems, they have been used in real-time, fluid dynamic environments with success; specifically, quadcopter auto stabilization is often implemented as a PID controller. With this in mind, a PID was implemented as a control system for heading change on the ATS Mk. 1, with the target heading as the SP and the current heading as the PV.

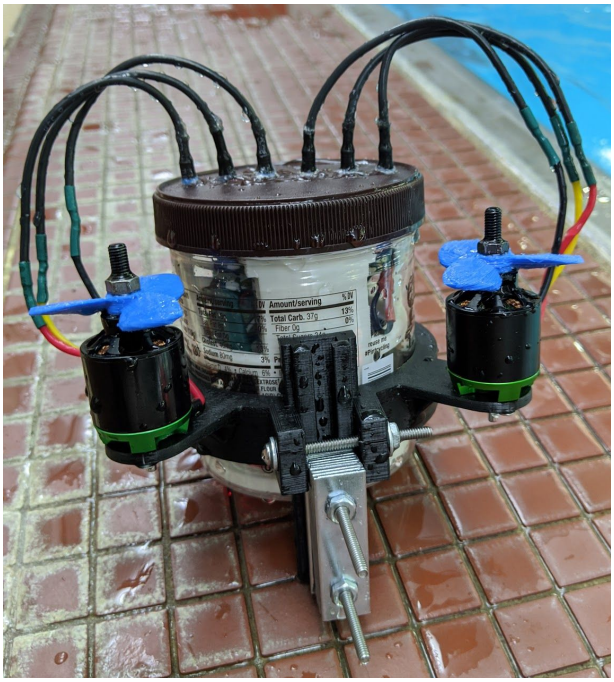


Fig. 46. Photo of the ATS Mk. 1

As described in the *Design* section, Engine subsection, each motor is driven from a value in the range of [-100, 100]. It should also be mentioned that the motors used for this application are high performance, high RPM motors that can draw up to 35 A each. This is overpowered for testing with the ATS Mk. 1 which is a small and light submersible, but were the only available motors at the time. For the stationary tests examined in this section, motor value/speed range was constricted to [-2, 2] (i.e. 2% speed) for general stability reasons as a speed over 2% caused overshooting in all cases. This sharply limits the range of speeds that the motor can be driven at by the PID controller, and future tests and test systems will include hand-selected motors that are appropriately powered.

Once the PID was implemented and confirmed as working correctly, the next step was to tune the controller. Several heuristics exist for tuning PID controllers; manual tuning and the Ziegler-Nichols method are two prominent choices. Manual tuning follows a set experimental

plan[39]: Set the I and D terms to 0. Increase P until the system oscillates, then halve that value- this is you K_p. Increase the I term until the system corrects itself within a reasonable period of time, however not too much as that will call instability- this is you K_i. Finally, increase the D term until the system corrects itself within a reasonable time frame- this is you K_d. This method guarantees a result that is suitable for the use case, however can be quite time consuming. The Ziegler-Nichols method produces K_p, K_I, and K_d by using the proportional gain term at the point of oscillation K_c and the period of that oscillation T_c as variables for the three respective terms. For a PID controller (methods exist for P and PI controllers as well), these equations are $K_p = 0.6K_c$, $K_i = 2.0/T_c$, and $K_d = T_c/8.0$[40]. This method is quick to implement, however has several limitations. The loop is tuned for quarter-amplitude damping, meaning that absolute oscillations above the setpoint have amplitudes that are ¼ of the previous amplitude. Inherently this means the loop is built to oscillate, which is not desirable. Instability is another issue, especially in lag-dominant processes such as this one (that is, slow reaction time and drift due to momentum on turns). Therefore, as the loop time for this process is fairly short, manual tuning was used.

[39]Bucz, Š., & Kozáková, A. (2018). Advanced Methods of PID Controller Tuning for Specified Performance. PID Control for Industrial Processes. doi: 10.5772/intechopen.76069

[40]Ziegler, J. G., & Nichols, N. B. (1993). Optimum Settings for Automatic Controllers. Journal of Dynamic Systems, Measurement, and Control, 115(2B), 220–222. doi: 10.1115/1.2899060

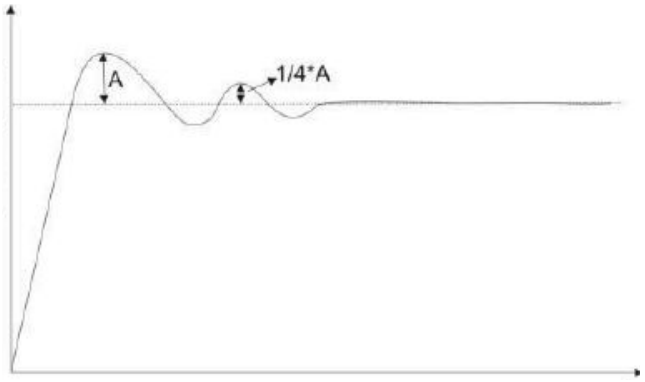


Fig. 47. Illustration of quarter amplitude decay

Early testing results are shown below, performed in the Union College swimming pool. While one of the main goals of these tests were to determine the optimal K_p, K_i, and K_d terms, the other was to gain some insight into how large of an impact waves and other transient motions in a body of water would impact the stability of the system. Specifically, how much did waves/buoyancy impact heading vs. the PID controller. Therefore, qualitative observations are provided at various phases of testing to offer a potential explanation for any unidentified behaviors, which are obtained by watching footage obtained of every test.

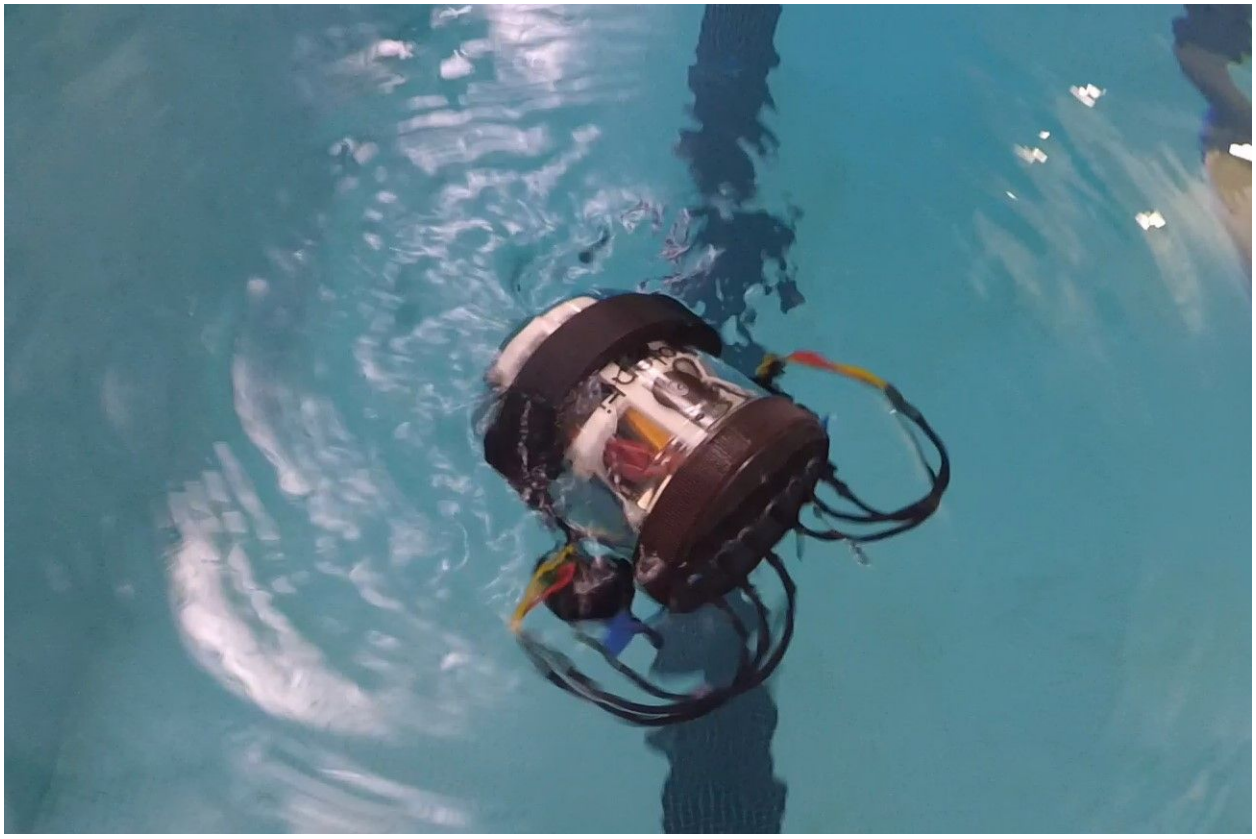
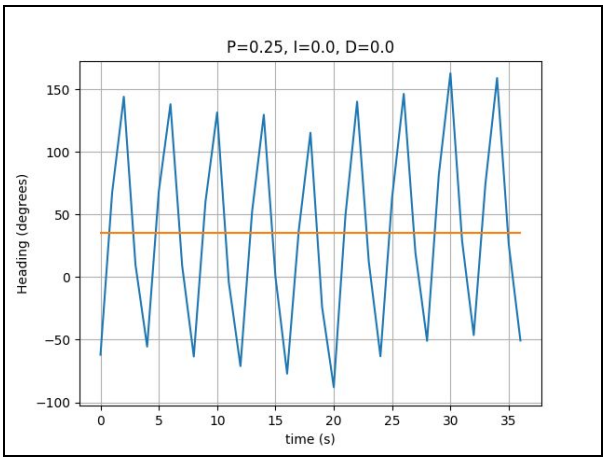


Fig. 48. ATS Mk. 1 in action

Once the system was placed in water and confirmed as ready, the I and D terms were set to 0, the P term was increased until oscillations formed. Below is an example of severe oscillation with P=0.25.



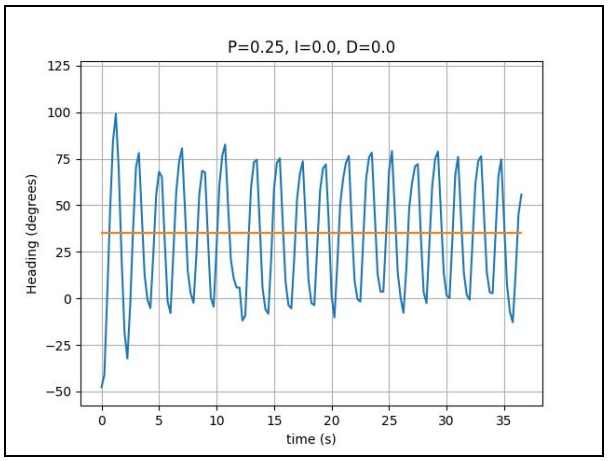
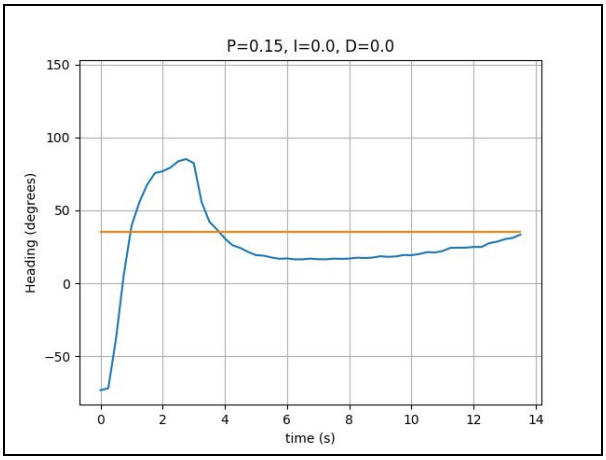


Fig. 49. PID oscillation at P=0.25, low frequency

Fig. 50. PID oscillation at P=0.25, high frequency

During these tests, almost all motion was generated by the motors on the sub and waves played a very small role. Although both tests were run with the same PID parameters, the period of the oscillation is significantly different between the two, along with amplitude. This points to severe overshooting in both directions and appears to create an endless loop; a large heading difference spurs a strong motor response, which overshoots the setpoint largely creating another large heading difference, hence a strong motor response, and so on. The P term was scaled back to 0.15.



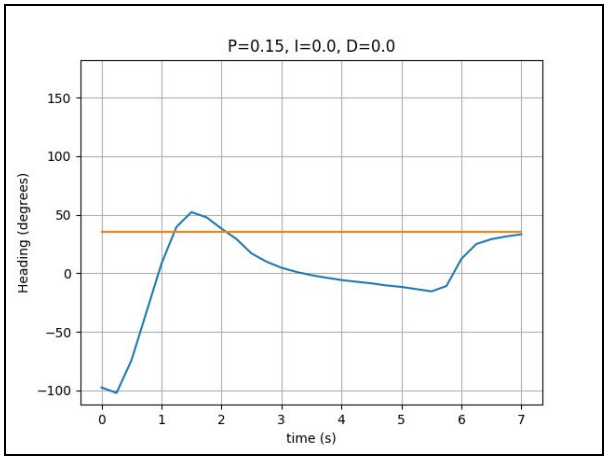


Fig. 51. PID performance at P=0.15, t=14s

Fig. 52. PID performance at P=0.15, t=7s

To ensure that the oscillations from the previous tests (P=0.25) were addressed, the sub was positioned at the same starting heading, roughly -50 degrees. As can be seen above, the oscillation is completely gone, which is an overcorrection as by definition the K_c term requires fixed oscillation. Hence, $K_c = 0.2$ was determined.

For the left figure above, waves placed a large role in the controller's reaching of the setpoint. There was a counter-clockwise motion induced by the motors at $t = 3\ s$ but all motion afterwards was purely due to forces in the environment acting on the sub. As any motor pulse typically creates more disturbance than in corrects, reaching a space generally close to the setpoint was understood as a large factor for continuous stability.

The next step was to divide this value by two and set it as the K_p term, then start on determining the optimal I term. Several I terms were used, with some of the best results being at I=0.0 and I=0.05.

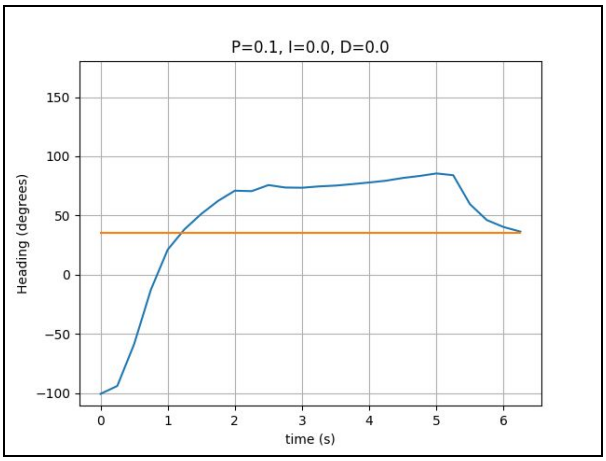


Fig. 53. PID performance at P=0.1, I=0.0

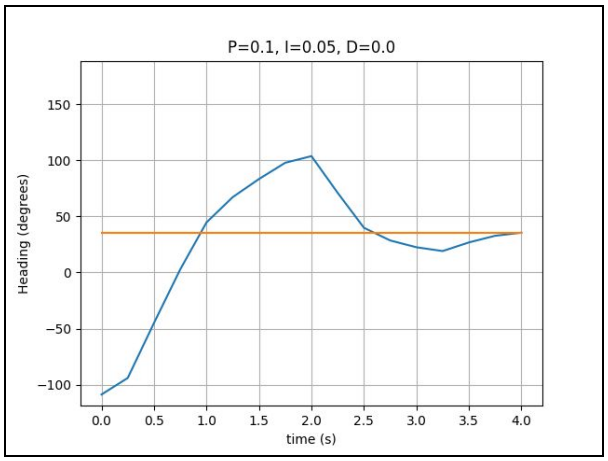


Fig. 54. PID performance at P=0.15, I=0.05

Results begin to look promising, and at the very least the controller closes in on the setpoint within a more reasonable time period. The right figure above points to better stability and loop control, as the slope's tangent line towards the final headings approaches is closer to the horizontal. Once again, for this figure, environmental forces caused the inflection at $t = 3.25\ s$, and the only motor actions were at times $t = 0\ s$ and $t = 2\ s$.

Increasing the I term to 0.075 caused oscillation, as seen below and as expected at some point when taking the integral term. Therefore, $K_i = 0.05$ was determined.

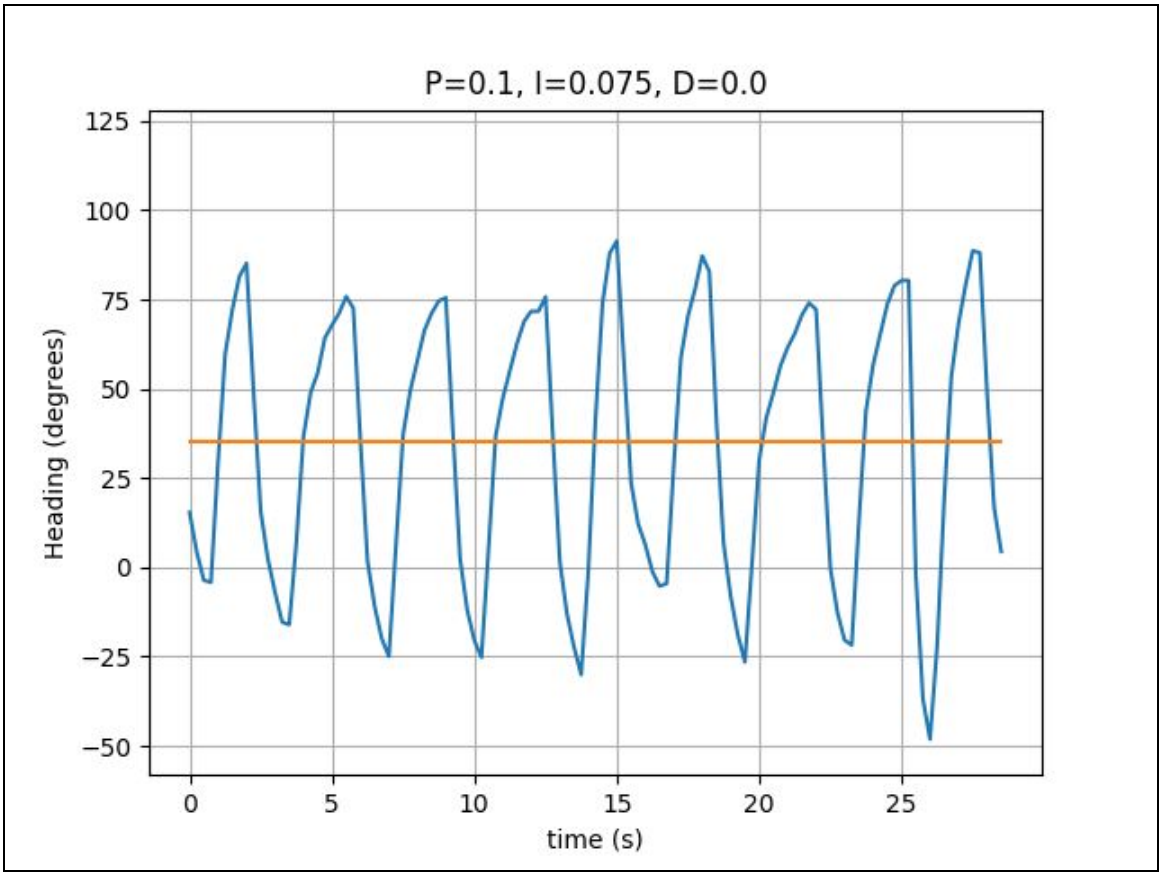


Fig. 55 PID performance at P=0.1, I=0.075

Finally, the D term was examined. As mentioned before, the D term may not be useful in this application, especially when factoring in how disturbing spinning motors can be to the relatively tranquil state of the environment. A good visual example of this hindrance is displayed below, when the D term was set to 0.1.

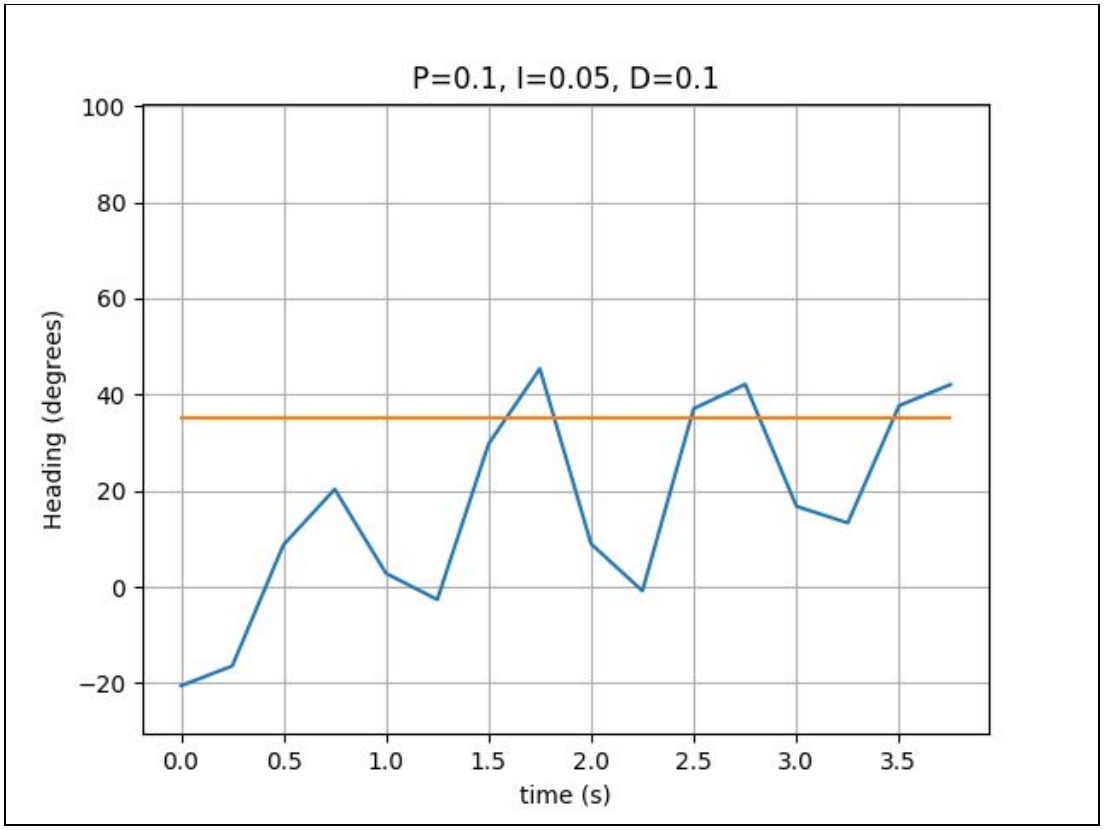


Fig. 56. PID performance at P=0.1, I=0.05, and D=0.1

While the controller was able to close in on the setpoint fairly quickly, the rapid motion exhibited by the sub caused drift and a relatively steep tangent line at the finishing point. There is also an example of preemptive action at $t = 0.75\ s$, where the sub backed away from the setpoint line as it was heading towards it. Therefore while more motion may allow the PID to reach its target quicker, and can even be orchestrated in a way that doesn't cause infinite oscillation, stability is still problematic.

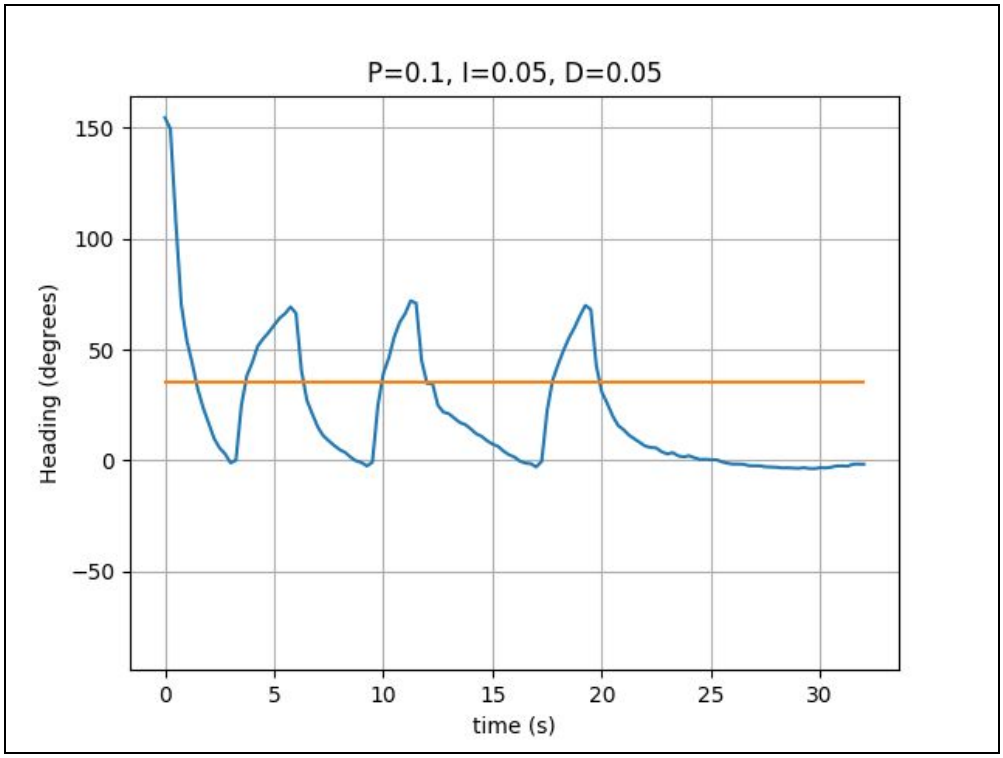


Fig. 57. PID performance at P=0.1, I=0.05, D=0.05

Using D=0.05 similarly produced undesirable results. Oscillations occurred once more, but at higher amplitudes and significantly longer periods than tests with D=0.075. After the third oscillation, the controller stops pulsing the motors entirely (effectively deciding that even a speed of 1 is too great), thus allowing drift and environmental forces move the sub. This test also began with a heading difference of more than 100 degrees, whereas the previous test had a heading difference of only 50 degrees or so.

D was reduced to 0.025. Results for these parameters are shown below.

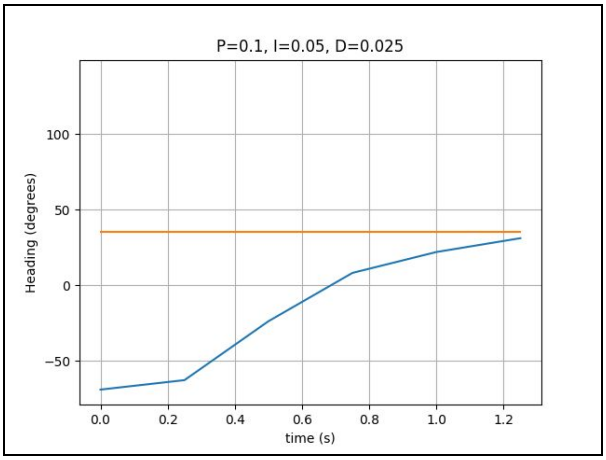


Fig. 58. PID performance at P=0.1, I=0.05, D=0.025, test #1

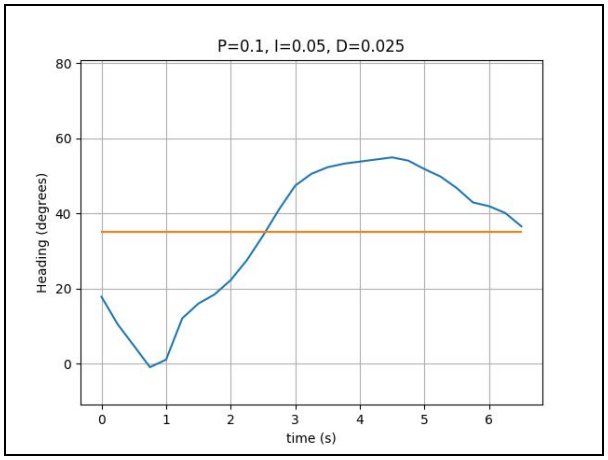


Fig. 59. PID performance at P=0.1, I=0.05, D=0.025, test #2

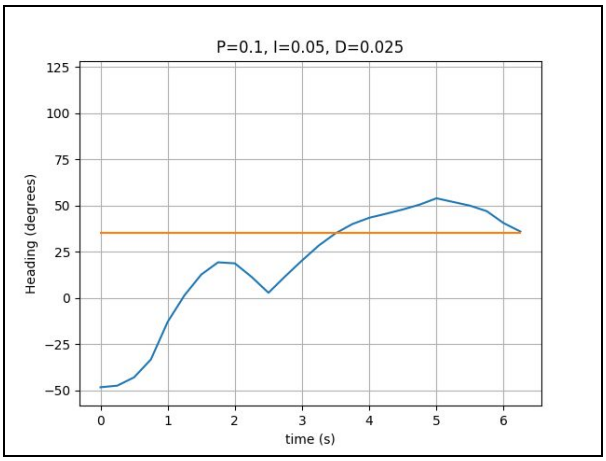


Fig. 60. PID performance at P=0.1, I=0.05, D=0.025, test #3

Test #	1	2	3	Avg.
Completion time (s)	1.3	6.5	6.2	4.667

Table. 3. Average completion times with P=0.1, I=0.05, D=0.025

For all tests, the controller needs no more than a single motor pulse and the final tangent line is relatively close to horizontal. Additionally, average completion time took less than 5 seconds overall, as can be seen in the above table. This provides the best parameters encountered thus far for achieving system stability. The top left figure above is especially encouraging, as the

heading difference was roughly 100 degrees and a single motor pulse was sufficient to achieve stability in a small period of time.

This is not an indication that these parameters are optimal, however they are a good starting point for future tests. Due to limited time to run tests after waiting several weeks to get access to the pool, the number of tests run thus far is small. The ATS Mk. 1 was able to hone in on target headings in many cases, but the class of motors used makes actuation rather blunt at this point- future tests may demand slower motors.

Future testing will include testing more PI configurations as opposed to PID, to see if a derivative term is needed. Beyond that, tuning the controller will include more intermediate values, instead of blocks of 0.025 used to increment/decrement parameters. Finally, testing of Ziegler-Nichols method will also be performed using $K_c = 0.2$ and an averaged T_c value at many different starting headings, as it has been shown that the oscillation period can vary greatly.

Testing performed thus far has also been solely for stationary heading changes, although the majority of heading changes in a mission will be mobile. This grants more granular control of yaw change over time and thus should allow for an increase of allowable speed differences between motors. Testing for this controller will continue into 2020.

Optical Component Performance Results (performed by Jacob)

This subsection details early work performed in gauging performance of optical communication components. Specifically an SFH 310 900nm phototransistor and a 3W white power LED are used, as described in the *Design* section, Communication block. These

components do not operate in the wavelength necessary for underwater wireless optical communication, but were used for general prototyping. Neither of the datasheets for these products include an average rise/fall time, so the maximum pulse rate at which the phototransistor can still receive full saturation could not be computed mathematically. Therefore, a small test was run using a signal generator, oscilloscope, DC power supply, an LED driving circuit and a phototransistor circuit (see the *Design* section for a schematic of these circuits). LEDs were pulsed at various frequencies and the voltage across the phototransistor was read out.

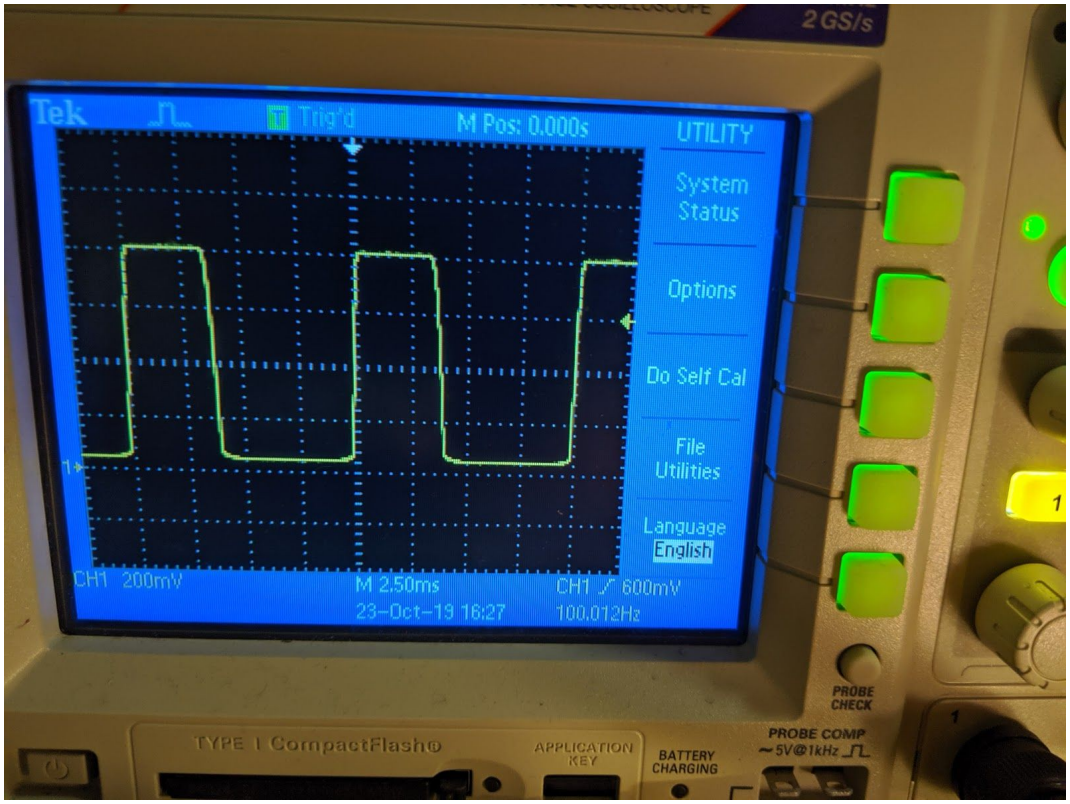


Fig. 61. Optical pulsing at 100 Hz

At 100 Hz, the square wave is read out without issue and the waveform is relatively rectangular. Rounding appears to be more of an issue on the falling edge, which indicates that the

LED fall time may be the hindering factor as LEDs typically have quicker rise times than fall times[41].

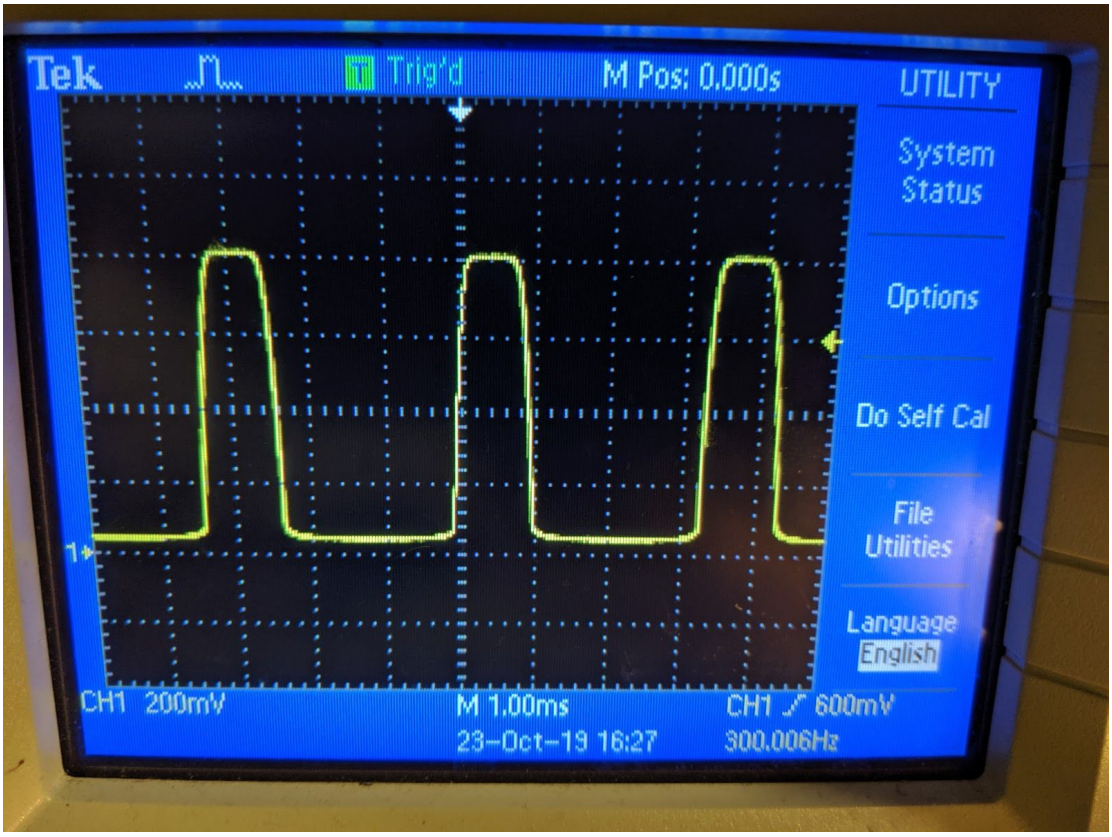


Fig. 62. Optical pulsing at 300 Hz

At 300 Hz, rounding is noticeable on both rising and falling edges, and the time delay is roughly even for both (0.4 µs). Potentially, the phototransistors' rise time is becoming more visible.

[41] High-speed switching of IR-LEDs — Background and data sheet definition

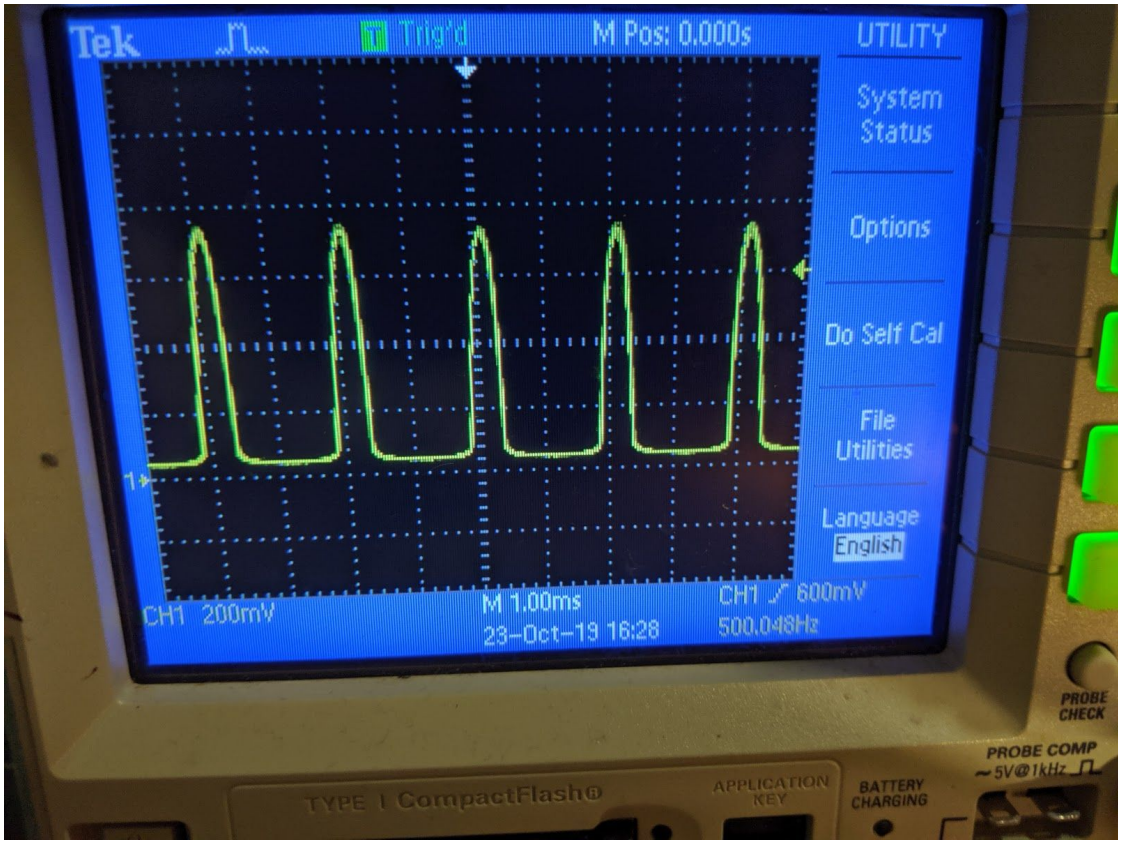


Fig. 63. Optical pulsing at 500 Hz

At 500 Hz, the pulses are almost completely rounded at maximum voltage. After 500 Hz, the saturation voltage cannot be achieved by the phototransistor. While this is not strictly necessary for communication, the high level of optical signal attenuation makes it especially important to keep the maximum voltage achievable at the receiving end as high as possible. Therefore, for the white light system, a minimum period between pulses of 4 ms is defined.

Tests were also performed to determine what the phototransistor voltage was at various distances to the LED. This is shown below.

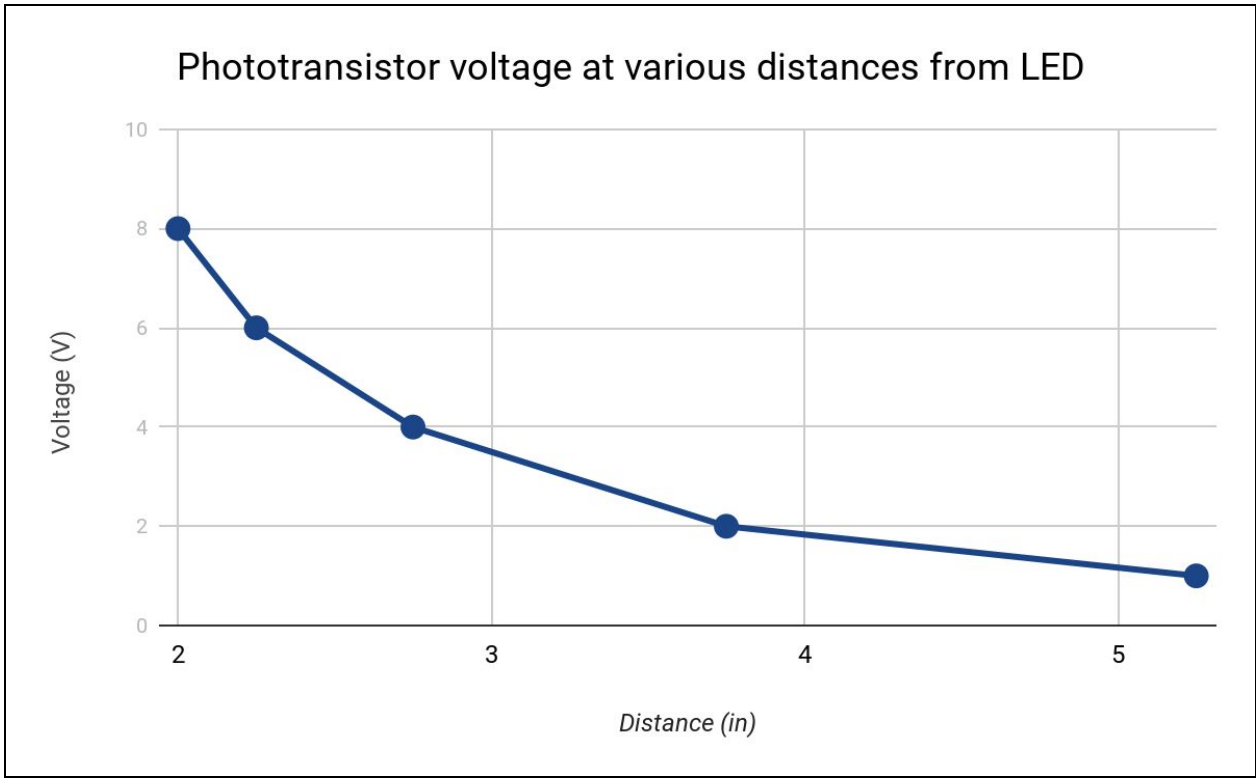


Fig. 64. Phototransistor voltages vs. distance for white light system

As can be seen, the signal strength is roughly 1/8th of maximum at a little of 5 inches away. It must be mentioned that the wavelengths of the LED and phototransistor are not very compatible; the peak wavelength of sensitivity of the phototransistor corresponds to about 0.6 at maximum for the relative spectral power distribution at the LED, and is largely scattered. For the components selected for optical communication, this number is at least 0.95 with a much smaller bandwidth and the LED draws approximately four times as much current.

These components have arrived but have yet to be extensively tested. Early indications show that a minimum pulse width of 2 ms is achievable, half of that for the white light system.

Software PWM Library under load (performed by Jacob)

For the implementation of PWM in the *Design* section, Engine block, a software PWM library is used to drive all ESCs. To ensure that the PWM library would be able to keep up with real time duty cycle changes and not oscillate uncontrollably, a load test was performed to determine the performance of this library.

PWM was performed on five GPIO pins. One of the five pins was randomly selected to act as the reference pin. The reference pin was sent a 5% duty cycle at 50 Hz (as would be experienced experimentally) and the maximum jitter of the signal was recorded before any other pins were turned on. Here, "jitter" refers to the deviation of the pulse period from the expected value. For a 5% duty cycle at 50 Hz, the expected pulse period is $0.05 * (1/50\ Hz) = 1ms$. However, deviations (or jitters) of up to 3 μs were observed, or 0.33% of the expected pulse width. The other four pins were then randomly and continuously assigned duty cycles between 5 and 10% at 50 Hz while the reference pin's output signal was examined. It was found that the maximum jitter was approximately 3.2 μs. This is a negligible difference and thus it can be concluded that the PWM library is capable of performing PWM on multiple pins simultaneously without issue.

Implementation Schedule

MKI phase

Testing relating to the further development of Motion Data and Captain will be performed in the remaining weeks of 2019, before the completion of the MK II. Simultaneously, testing of the optical communication system and the local awareness sensors will be performed on their own test beds to ensure they are ready for implementation into the MKII. During this phase, the evaluation of using an IMU based displacement sensor will be finalized, and alternative designs will be selected if deemed unfeasible.

MKII phase

The completion of the MKII, scheduled for the first two weeks of 2020 bar any mechanical setbacks, will include 3-space navigation capabilities as well as communication and local awareness hardware. Testing, integration and additional development of the code designed in the previous phase will be completed over a couple weeks following the systems completion. The remainder of the time before the completion of the MKIII will be dedicated to building up the environment in which all code sections communicate and evaluation of the final hardware component selection.

MKIII phase

The MKIII, which should exhibit all described design aspects, is scheduled to be completed two weeks before the end of winter term. As the MKII will have the same motion and sensor capabilities, minimal refactoring will be required for this transition. Testing, modification, verification and documentation will occur during these final weeks.

Future work

This project is planned to continue outside of the defined timescope, and further testing and development will be performed during the spring term. Depending on the actualized capabilities exhibited by the MKIII, this will be focused on the development of functionality within a flock. At least one more MKIII will be constructed, and the development and testing of group dynamics will proceed until graduation in June 2020.

References

M. E. G. Mital *et al*., "Characterization of underwater optical data transmission parameters under varying conditions of turbidity and water movement," *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, Malacca City, 2017, pp. 1-6.

Johnson, L. J., Jasman, F., Green, R. J., & Leeson, M. S. (2014). Recent advances in underwater optical wireless communications. Underwater Technology: International Journal of the Society for Underwater, 32(3), 167–175. doi: 10.3723/ut.32.167

Brundage, H. (2010). Designing a wireless underwater optical communication system. Mechanical Engineering - Master's Degree. Retrieved from http://hdl.handle.net/1721.1/57699

Tinker Board S: Single Board Computer. (n.d.). Retrieved from https://www.asus.com/Single-Board-Computer/Tinker-Board-S/.

S. M. Smith and D. Kronen, "Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation," Oceans '97. MTS/IEEE Conference Proceedings, Halifax, NS, Canada, 1997, pp. 714-720 vol.1.

J. Snyder, "Doppler Velocity Log (DVL) navigation for observation-class ROVs," OCEANS 2010 MTS/IEEE SEATTLE, Seattle, WA, 2010, pp. 1-9. doi: 10.1109/OCEANS.2010.5664561

J. Sticklus, P. A. Hoeher and R. Röttgers, "Optical Underwater Communication: The Potential of Using Converted Green LEDs in Coastal Waters," in IEEE Journal of Oceanic Engineering, vol. 44, no. 2, pp. 535-547, April 2019.

(n.d.). Retrieved from https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/EquipmentTrans/characteristicspt.htm.

Butler, J. L., & Sherman, C. H. (2018). Transducers and Arrays for Underwater Sound. Cham: Springer International Publishing.

Ainslie M. A., McColm J. G., "A simplified formula for viscous and chemical absorption in sea water", Journal of the Acoustical Society of America, 103(3), 1671-1672, 1998.

Fish, F.E.; Schreiber, C.M.; Moored, K.W.; Liu, G.; Dong, H.; Bart-Smith, H. Hydrodynamic Performance of Aquatic Flapping: Efficiency of Underwater Flight in the Manta. Aerospace 2016, 3, 20.

Font D, Tresanchez M, Siegentahler C, et al. Design and implementation of a biomimetic turtle hydrofoil for an autonomous underwater vehicle. Sensors (Basel). 2011;11(12):11168–11187. doi:10.3390/s111211168

Bucz, Š., & Kozáková, A. (2018). Advanced Methods of PID Controller Tuning for Specified Performance. PID Control for Industrial Processes. doi: 10.5772/intechopen.76069

Ziegler, J. G., & Nichols, N. B. (1993). Optimum Settings for Automatic Controllers. Journal of Dynamic Systems, Measurement, and Control, 115(2B), 220–222. doi: 10.1115/1.2899060

Appendices

A) Team dynamics

While the parallelization of work is a significant advantage that stems from working in a team, a great deal of effort must be exerted to ensure that the team can function as an efficient unit. As the scope of the course is not designed to accommodate or teach methodologies conducive to this situation, significant research was performed at the start of the project with the aim of proactively avoiding future problems. As all members of the teams are students, it is not feasible for any individual to spend the majority of their time on this project, unlike the majority of workplace environments. This meant that all research methodologies needed to be adapted to our situation, and certain aspects would need to be developed or adapted to match the environment.

The first implementation of research methodologies was a group ideation session where the critical design features were determined by analysis of current aquatic research efforts, currently available AUV solutions, and aspects judged to be important to customers. Many of the

relevant methodologies from here[42] and here[43] were adapted to apply towards the domain of this project. This had the dual benefit of resulting in a well thought out project in regards to what is available in the field, as well as ensuring that all team members were on the same page regarding the project and its core features, which would remain an important factor for the duration of its development. Photos of this session are available in *Appendix B*.

To keep everyone on the same page, we decided to adopt the idea of weekly sprint meetings where we would inform each other of what we have accomplished since the previous meeting, and what we planned to accomplish before the next meeting. To further this goal, it was mandated that all research and documentation is stored in a shared cloud based environment, allowing all members to quickly catch themselves up on the progress of other team members, and the easy sharing of documents and ideas.

Having two distinctive groups from different departments provided more challenges than initially anticipated. The most evident of these problems is that this effectively creates two separate projects that need to be meticulously synchronized in their development. To assist in this, a shared document was generated where each team could designate a project for the other team to work, and then fill out needs and wishes. There is then a section where the other team can leave notes on each of these aspects, either to reference or to request clarification. This was to supplement rather than replace discussion during the weekly meetings.

The unforeseen issues stemmed from the way members of each department have been trained to think. This has shown to be beneficial in some cases, for example there have been design issues that one team was struggling to solve and the other was able to come up with a

[42] https://medium.com/the-creative-founder/needfinding-for-disruptive-innovation-71d8532f2cf3
[43] https://medium.com/the-creative-founder/ideation-sprints-for-new-products-services-74f925190b4f

solution rapidly due to a different perspective. However, it has been the case multiple times that there was a disconnect between teams due to assumptions of common knowledge. For example initial issues on the MKI and the spinning test rig were due to the mechanical team overlooking the physical dimensions of wires, something that the other team took for granted as a constraint. All members have learned to be careful when communicating with the other team to ensure that optimal comprehension is achieved.

To further the efficiency of working in groups, we have made sure that as many aspects of the AUVs components can be duplicated, allowing parallel development and testing by different members of the team. An example of this is that the MKI will be duplicated, in full or in part, so both members of the Computer/Electrical team can continue work over winter break.

B) Media

Uncurated live updated photo gallery of work to date:

https://photos.app.goo.gl/6BxMWJuVAo57jb7g8

Media from pool tests:

https://drive.google.com/drive/folders/18NZGOvMQ7kgKeahAzZrMWY-nTwBu4vXU?usp=sharing

C) Meeting Log

An attempt was made to document all team meetings, though inevitably there were several situations where documentation was forgotten. Below is an informal meeting log.

When	What	Who	Notes
5/14/2019	Ideation Session	Alex, Jacob, Sam, Xavier	See Photos
9/12/2019	Initial Piezo Testing	Jacob, Xavier	See Photos
9/13/2019	Piezo testing with driver and hydrophone prototypes	Jacob, Xavier	See Photos
9/18/2019	Group progress and planning meeting	Alex, Jacob, Sam, Xavier	
9/19/2019	CpE specification meeting	Jacob, Xavier	Discovered transducer housing development issues
9/19/2019	Call with Scott about Transducers	Jacob, Sam, (Scott), Xavier	Relevant document here

9/20/2019	Setting up a pi environment with needed software and file management. Connecting the BNO055 IMU to test base functionality	Jacob, Xavier	Pi chosen, raspbian installed, scripts added to path, IMU packages installed. There may be communication limitations on the pi with I2C, so we shall see
9/22/2019	Discussing timeline, Developed Gantt Chart	Alex, Jacob, Xavier	Sam was unable to make it, which resulted in some drawbacks in our overview of planning
9/24/2019	SRG writing meeting	Alex, Jacob, Sam, Xavier	Also debriefed on Walt Dixons insights on the Gantt chart and the rest of the

			project
10/1/2019	Meeting with ME professors	Alex and Sam	
10/4/2019	Full team meeting	Alex, Jacob, Sam, Xavier	Caught up all team members on cross departmental needs
10/6/2019	IO and specifications meeting	Jacob, Xavier	
10/7/2019	Discussion of optical receiver requirements and potential component selection	Jacob, Xavier	
10/15/2019	Team meeting, green grant proposal	Alex, Jacon, Sam, Xavier	
10/26/2019	Sonar, delegation, general procedure meeting	Alex, Sam, Xavier	
10/27/2019	Sched_deadline and interrupt meeting	Jacob, Xavier	
10/30/2019	Prop optimization meeting		

10/31/2019	Poster planning meeting	Jacob, Xavier	
11/5/2019	Total power draw estimation meeting	Jacob, Xavier	
11/7/2019	Team meeting, planning out the rest of the term, ensuring MK2 will have required features	Alex, Jacob, Sam, Xavier	
11/12/2019	First pool test	Jacob, Xavier	
11/13/2019	Meeting to discuss the results and issues encountered during the first pool test	Alex, Jacob, Sam, Xavier	
11/18/2019	Second pool test	Jacob, Sam, Xavier	
11/20/2019	Third pool test	Alex, Jacob, Xavier	
11/22/2019	Fourth pool test	Alex, Jacob	
11/23/2019	Design report meeting	Jacob, Xavier	