Marine Vehicle

Team Members (from left-right)

Judas Taylor
Mark Buntsev
Jack Yarmas
Tyler Costa
Michael Schratz

Advisors

Professor May-Win Thein
Sital Khatiwada (Graduate Assistant)
John McCormack (Graduate Assistant)

Sea Grant
New Hampshire

Table of Contents

Table of Figures

MV

Acknowledgements

Special thanks to all who helped make this project possible including but not limited to our faculty advisor, Professor May-Win Thein, graduate assistants Sital Khatiwada and John McCormack, as well as many other faculty and staff in the Mechanical and Ocean Engineering departments through valuable inputs and tips. Kind donations from family and friends willing to extend their hand out to see us succeed. Additionally sponsored in part by the New Hampshire Sea Grant College Program and UNH Mechanical Engineering Department.

Marine Vehicle Overview

Today’s robotics/STEM world is vastly saturated with on-land and in-air vehicles that are becoming more and more user friendly and easier to create. A virtually unexplored area, in terms of STEM, is underwater. The purpose of the Marine Vehicle is to provide a platform for more advanced STEM education and research in high schools. Successful integration requires high levels of safety, a cost effective design optimized for effective manufacturing, and a fun platform for students to foster a love for marine robotics. The Marine Vehicle team wants to capture the attention of potential engineers with dynamic and highly functional controls, a high power to weight ratio, and a thruster placement design that emphasizes speed and agility.

The design was built to maximize functionality while remaining as simple as possible to allow for high school students to build with ease. High school students will gain experience in all aspects desired which includes but is not limited to casting and molding, electrical assembly, and potential to build off the pre-designed platform to be able to customize the vehicle in how they see fit. This will allow for all students in engineering to participate depending on their interests.

Research & Development

Initial Scope (SAV)

In the beginning, the senior design team aimed to be an innovator in the field of drones by bridging the gap bridge between air and water. The concept of Submersible Aerial Vehicles (SAVs) began as a business idea with Tyler Costa, Judas Taylor and Mark Buntsev during junior year of Mechanical Engineering. This idea soon became a senior design project once the group spoke with Professor May-Win Thein about potentially starting up a first year team called SAV. Consisting of 5 group members: Tyler Costa, Judas Taylor, Mark Buntsev, Mike Schratz and Jack Yarmas, SAV planned to create a drone capable of swimming under water and immediately taking off into the air. The applications for a drone with this type of multi-terrain capabilities are endless and include (but are not limited to) surveying underwater structures, studying marine biology, and maneuvering within difficult environments.

One of the huge potentials of SAV is to be used in a wide variety of purposes. Being able to have the same applications as an underwater Remotely Operated Vehicle (ROV) and the ability to liftoff from any terrain, SAV has a huge competitive advantage against an aerial drone or sole purposed underwater ROV. SAV would adhere to the desires of an average drone enthusiast, the requirements of bridge surveyors, and the needs of professional photographers such as Natural Geographic. Being able to eliminate the worry of losing a quadcopter to a mistake (such as falling into a river) is a huge incentive to choose SAV over any quadcopter currently on the market.

Interdisciplinary working is a key part to the success of the team. With such a diverse skillset, there is plenty of opportunity to connect high level skills and be able to work in conjunction with May-Win Thein’s other projects such as the Autonomous Surface Vehicle (ASV) and ROV to create top level projects. With four Mechanical Engineers on the team, there is a high emphasis on design to optimize cost vs. functionality. With an end goal to sell to consumers, SAV planned to make the project as user friendly as possible, avoiding the limitation of our vehicle to be used only by engineers and technically advanced consumers.

Utilizing modern tools disposable to everyone, SAV combined together elements taught to the group in Systems Modeling, Ocean Waves and Tides, Solid Modeling, and Scripting Languages. Using interdisciplinary skills, SAV strived to create a product that is both manufacturable and easily repeatable. With an end goal of patenting in mind, elementary instructions and repeatability were essential in the creation of SAV. The intention of the Capstone is to combine the knowledge that has been gained as University of New Hampshire students and apply these concepts to work in a professional environment. Through the creation of a new product that can be replicated in a manufacturing setting, the team will have proved that they learned several professional skills throughout our studies. SAV aimed to follow a timeline of milestones with an end goal of the Senior Capstone to create a working prototype that can be expanded on in the future, and these objectives are summarized in the Gantt chart below.

The biggest obstacle initially foreseen was the coding of SAV, however the addition of an IT major (Jack Yarmas) eliminated this concern. As the group leader, Tyler Costa interacted with

May-Win directly on the team’s behalf. Equally, Tyler worked in conjunction with Mark Buntsev on the aerial portion due to their experience gained over Summer 2017 working with Professor Thein. Mark Buntsev also took advantage of his business experience under the role of treasurer as he organized and directed funds properly throughout the course of the year (Appendix A1). Judas Taylor applied his extensive background in a marine setting to lead the underwater portion of the project. Michael Schratz worked with Judas on the marine aspects while he also acted as a general coworker to all other portions of the project.

Research & Testing

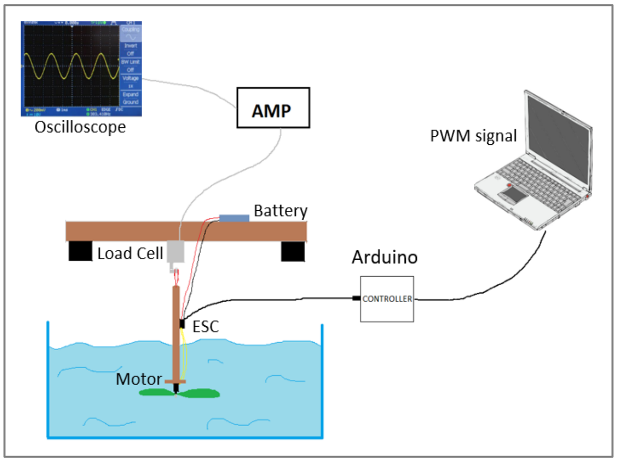


Figure 1: Schematic of aerial propeller tests underwater

For the first design of SAV, the team investigated the possibility of having aerial propellers assist in underwater propulsion. The driving force behind this decision was to minimize the cost of parts and total weight of the vehicle; if the aerial portion provided a sufficient force underwater, it would reduce the amount of underwater thrusters required for movement below the surface.

To test the capability of this solution, the team attached several propeller sizes to a 2500kV DC brushless motor which was fixed to a 10 pound load cell via a dowel. Once the motor and propeller were submerged in a water tank, the system was given pulse-width modulation (PWM) input signals ranging from 1000-2000ms through an Arduino controller. A basic 30 amp electronic speed controller (ESC) converted these readings to draw the appropriate power from a Turnigy LiPo battery (11.1V, 5000mAh) and sent them to the brushless motor. The resultant force on the load cell was provided a gain of 100 by sending the signal through an AD620 Instrumentation Amplifier, and the difference in output voltage was measured by an oscilloscope. A schematic of the underwater propeller tests is shown in *figure 1*.

In addition to testing the underwater performance of each propeller, all sizes were analyzed measuring the force they exerted in the air using the same methodology. The difference between the output voltages measured at rest versus the voltage at a given PWM input was converted to pounds-force in MATLAB. This was found using the sensitivity derived from the load cell calibration curve (1.1857 V/kg), and a portion of these results are included attached in appendix A2.

The naming convention for each propeller is as follows: the first two numbers refer to the length of diameter, while the last two define the pitch angle of the propeller. For example, Propeller 7038 has a 7 inch diameter with a 38° pitch.

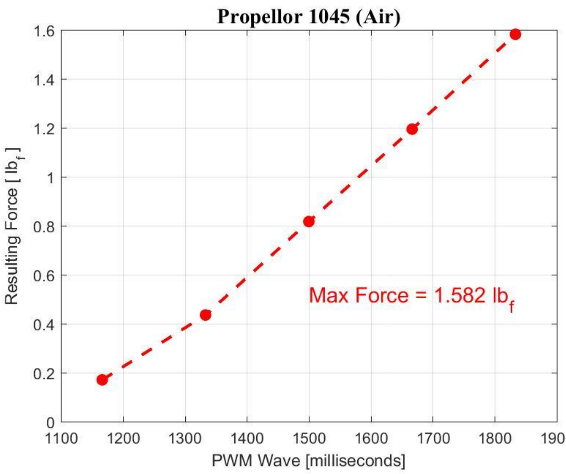


Figure 2: Propeller 1045: aerial results

In some instances, propellers provided no force readings underwater; several cases experienced an inconsistent jerking motion which was likely due to cavitation. Furthermore, propellers that were able to run in the water tank showed little to no correlation between the input signal and the magnitude of the resultant force, causing uncertainty when increasing the power draw. Finally, there was significant trade-off between the propellers' force exerted above and below the surface. For example, the most efficient propeller in air (1045) could not operate in water, and the optimal propeller for underwater propulsion (7038) gave insignificant lift force in the air compared to other options. From these results, the team concluded that there was no harmony between aerial lift and underwater propulsion using brushless motor propellers, therefore the aerial and underwater components of the SAV must be held independent of one another.

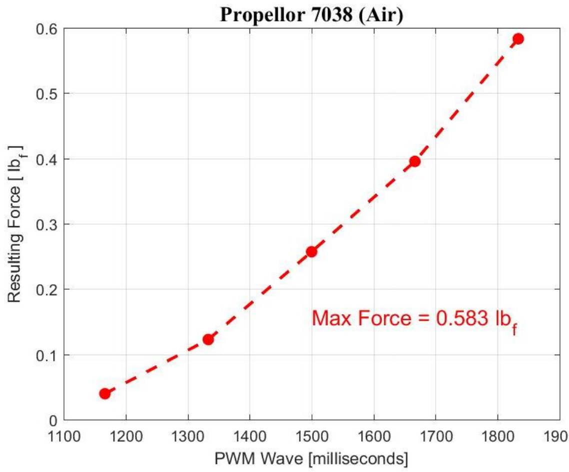


Figure 3: Propeller 7038: aerial results

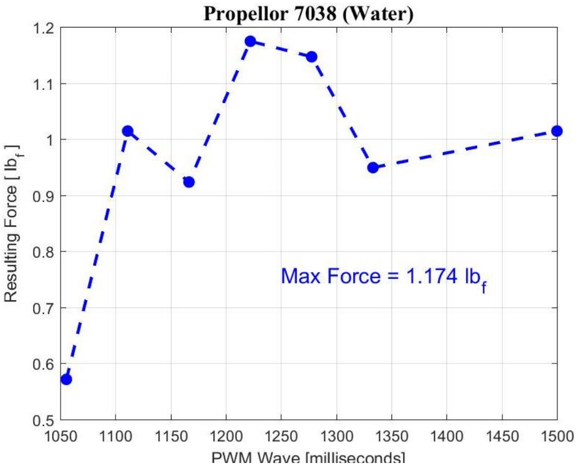


Figure 4: Propeller 7038: underwater results

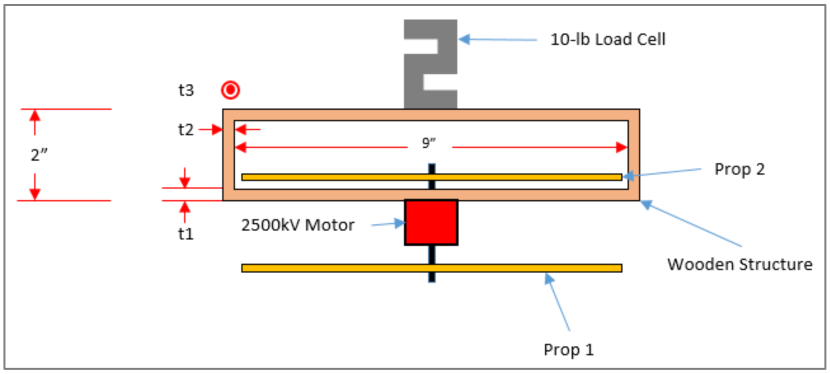


Figure 5: Schematic of aerial testing for duel propellers on a single motor

With this new information, a second design of SAV was constructed. Since the aerial team could focus solely on maximizing lift, further testing was performed to inspect whether an additional propeller on each motor would increase the resultant force. Though the wiring for both experiments was generally the same, the test section was slightly modified to include a second propeller (*Figure 5*). In order to closely replicate the stacked propellers actual functionality, the team minimized the thickness t_3 into the page to maximize the propeller's flow facing area. Many combinations of propellers were analyzed, some of which are outlined below.

As expected, the resulting lift force was proportional to the motor RPM for all propellers tested. *Figure 6* on the left outlines the effects of adding another propeller to the 2500kV DC motor while keeping the bottom propeller (8045) constant. The best combination between propellers was 8045/6030 which proved to have the largest average thrust over the RPM's measured. However, thrust was maximized when testing a single 8045 prop. Moreover, this trend continued to hold true when adding another propeller, regardless of the size or pitch angle (*Figure 7*). From these results, the team concluded that implementing a stacked propeller system would actually be a disadvantage to the overall functionality of the aerial design. Code for the propeller testing results are appended (Appendix A5).

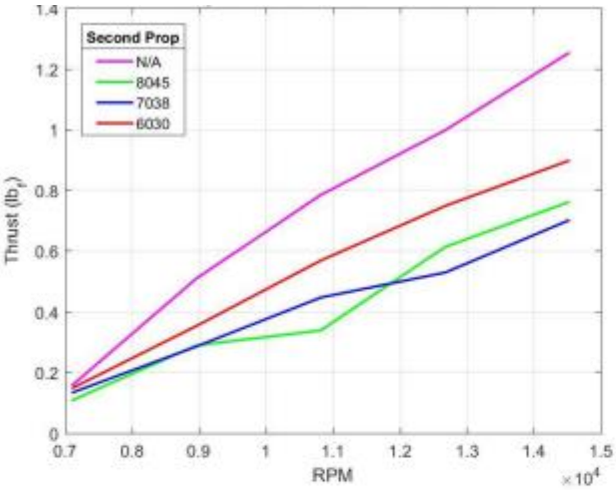


Figure 6: Propeller 8045: thrust comparison with varying propellers

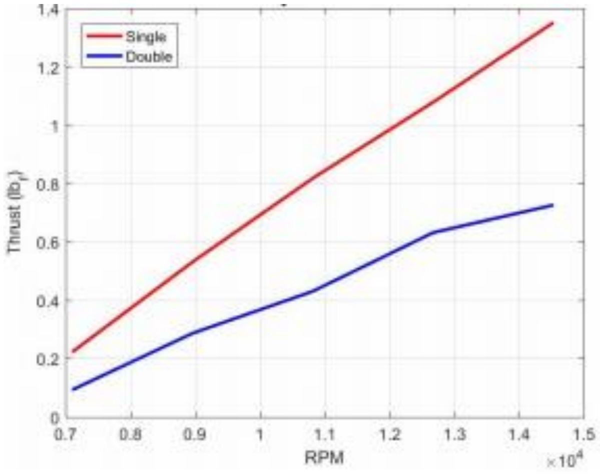


Figure 7: Propeller 7038: single vs. double thrust comparison

While the aerial team executed these experiments, the marine team composed a weight analysis of the entire SAV, striving to construct the lightest vehicle possible. To achieve maximum maneuverability underwater, 6 degrees of freedom (DOF) must be satisfied. These are outlined below and illustrated in *figure 8*. Though it uses an airplane as a model, the dynamics illustrated were similar to that used for the equations of motion in the code.

1. Move up & down
2. Move left & right
3. Move forward & backward
4. Swivel left & right about Z-axis (yaw)
5. Tilt forward & backward about X-axis (pitch)
6. Rotate side to side about Y-axis (roll)

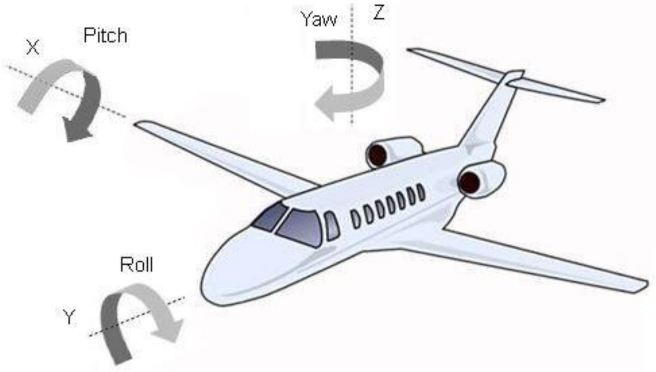


Figure 8: Diagram for X/Y/Z - Pitch/Roll/Yaw used

Utilizing knowledge gained from the ROV team, it was brought to MV's attention that specific configuration of 6 underwater thrusters was proven to ensure all types of motion are covered, which gave a starting point. The need for a waterproof electronics housing unit and a LiPo battery was evident, which added a substantial amount of weight to the vehicle. When estimating the expected masses of the aerial portion and SAV's hull, the total weight was quickly becoming an issue.

Finally, the largest contributor to the overall weight was the tether which provided direct communication between the submerged SAV and a topside interface. Typical quadcopters transmit signal wirelessly via Bluetooth connection. However, these types of signals can only travel a few inches in water before getting lost because water acts as an electrical conductor, dissipating any high frequency signals [1]. Very low frequency (VLF) waves in the range of 3-30 kHz are able to travel through water for depths of around 20 meters, but Bluetooth operates at 2400 MHz. Furthermore, if SAV were to use wireless low frequency signals, data transmission would take far too long to process, meaning live video streaming and constant changes of thrust demands would not be achievable. Therefore, the team decided it was necessary to incorporate a 25-meter, 2.4 pound tether which essentially acts as a long Ethernet cable, allowing for high frequency signal transmission.

Signal attenuation proved to be a major obstacle in the scope of SAV; although the aerial team maximized lift force and the marine team did everything possible to minimize the total weight, the addition of a heavy tether was unavoidable. Given the time constraint of senior design projects, the team reevaluated the plausibility of creating a marketable submersible aerial vehicle by the end of the year.

Revised Scope (MV)

Taking all research into consideration, the team decided to slightly refocus the scope of the project to engineer a manufacturable Marine Vehicle (MV). The team chose to isolate the underwater portion rather than aerial for one main reason: quadcopters are commonplace in today's society, whereas creating an underwater vehicle would breach a relatively new and competitive area within ocean engineering. MV still aimed to construct a user-friendly vehicle that can be replicated in a manufacturing setting. However, instead of adhering to all degrees of freedom underwater, MV strived to achieve a fast and sleek design by minimizing the amount of thrusters and therefore overall weight. Moreover, the end goal of MV was to provide a platform for more advanced STEM education and research within high schools.

Under the revised scope, the team investigated several methods of underwater propulsion. First, MV looked to repurpose bilge pumps which are used to remove water from the area around the outer surface of a ship's hull. Although there were numerous examples of bilge pumps modified as powerful thrusters [2], these were all too heavy for MV's requirements. Next, the team considered the possibility of creating in-house thrusters by combining M100 brushless DC motors with underwater propellers, but the process of fabricating a protective enclosure was too extensive.

The final solution was to integrate Blue Robotics' T100 thrusters [3] into the design of MV. This provided an optimal combination of cost, functionality, and ease of accessibility. Interactive charts on the Blue Robotics website allowed the team to analyze the T100's forward and reverse thrust based on the power draw [4]. Additional testing was performed on the T100 thrusters using CFD analysis through the program OpenFOAM in order to fully comprehend the flow characteristics (Appendix A3).

Moving forward, Mike Schratz lead the parts acquisition process, ensuring Jack Yarmas' requirements to code were satisfied. Judas Taylor and Mark Buntsev worked conjointly on the SolidWorks design of MV. Tyler Costa performed extensive research on the molding and casting process as well as preliminary testing of the various methods.

Design Process and Assembly

Initial Steps

The readjustment of the scope of the project left the Marine Vehicle team in an interesting situation that allowed room for a new wave of creativity while being confined in a fairly narrow spectrum to follow. The purpose of MV is to be a fully functioning underwater vehicle fulfilling the criteria of being easy to build, functional usage, fun, and most importantly safe. With these principles in mind, the design process had to begin from nothing. Inspiration was taken from existing models in place that could be modified to fit the criteria.

Inspiration was taken from multiple aspects of the "water" world. For example, the idea of fun came from the working of a jet ski and most boats. That is that continuous forward motion is needed for the vehicle to travel in any direction. In the case of a boat, a motor or engine in the back being the driving force of motion with left or right directions being guided by a rudder. Inspiration was also taken from nature for some aspects and initial considerations. Specifically, stingrays were analyzed and the idea of a flattened body that is able to glide along the ocean floor and other structures caught the attention of MV. "When they are inclined to move, most stingrays swim by undulating their bodies like a wave; others flap their sides like wings. The tail may also be used to maneuver in the water" [6].

The UNH Marine Vehicle team steered itself away from the idea of traditional Remotely Operated Vehicle (ROV) setups as they were seen to be too bulky, complicated, and expensive to fit the desired outcome. Something was needed with less components, less complexity, and more maneuverability. The intention was to not reinvent the wheel by using premade components proven to show functionality in the water. Research was done on deciding what sort of supplier to use to serve certain functions such as driving forces and a microcontroller to control the setup. Some initial thoughts were placed on using bilge pumps, computer pumps, or brushless motors with attached props, but ultimately, through extensive research ducted fans/ thrusters were decided upon as the pushing power. Cheap thrusters were found from all over including places such as Alibaba from non-reputable sources so those were thoughts were discarded. BlueRobotics was found to be the best supplier based on reputation, proven components, and as being a hub for all the required components needed to control the vehicle.

The first task before ordering parts was to come up with a design for the newfound concept. With the freedom of not having any bounds to fulfill, the team was able to allow their imaginations and creativity roam to create something unique and esthetically pleasing. Each Mechanical Engineer in the group took the task of creating a preliminary design to show to the rest of the group. These designs were created in SolidWorks with the goal in mind of bring together the best parts of each design into something that everyone agreed upon. With no real methodology to creating the designs aside from following the purpose of the project, each design had defining characteristics which exemplified the creativity in each group member. Some of the designs can be seen below with a small description of each.

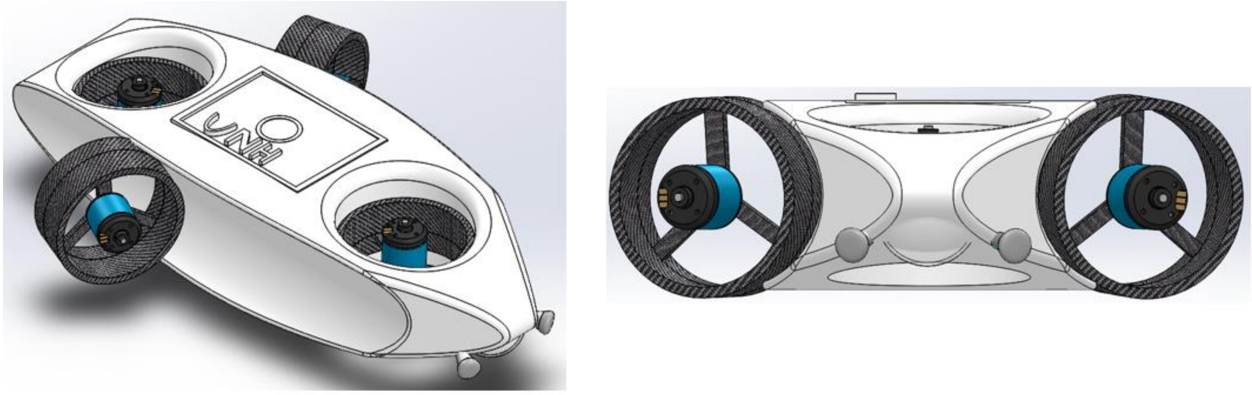


Figure 9: First preliminary design. Four thrusters with viewing dome bottom front.

Figure 9. Designed with the intent to be the most unique and differed heavily from all other designs. The internal components were to be stored under a hatch within the center hull.

Figure 10. Bulky initial design to allow for lights to be mounted within the enclosure, but deemed not optimal because of small field of view out of front.

Figure 11. Sleek design focusing on esthetic appeal, however, similar models exist with a thruster in front and two in back. Futuristic appeal to be modeled from Sci-Fi spaceship designs.

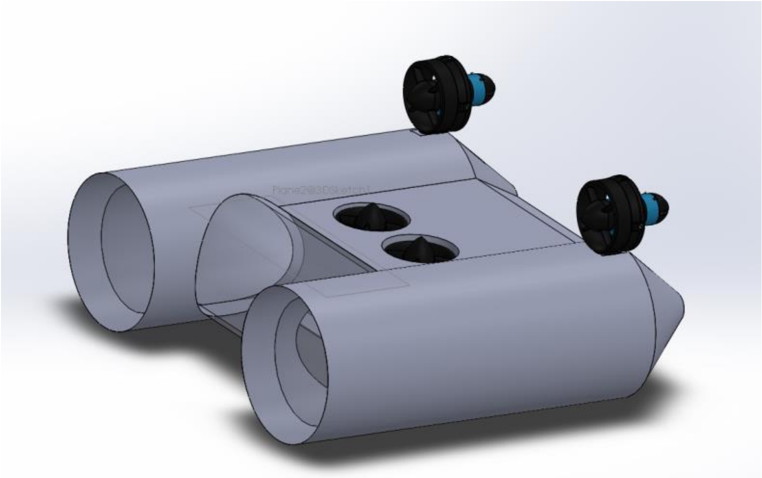


Figure 10: Bulkiest preliminary design.

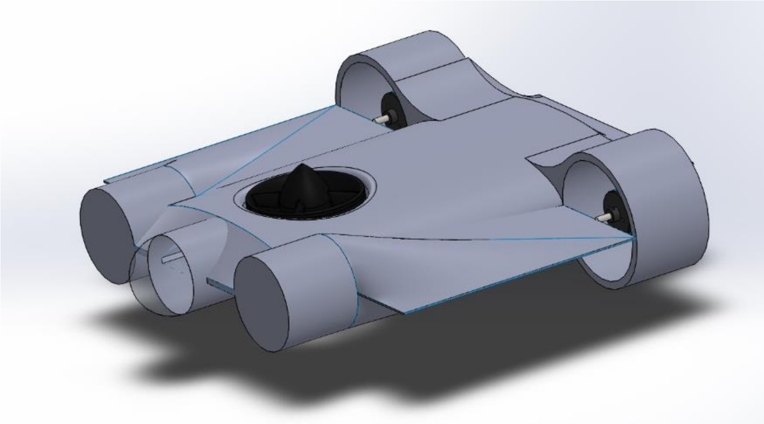


Figure 11: Sleek new design with two thrusters in back

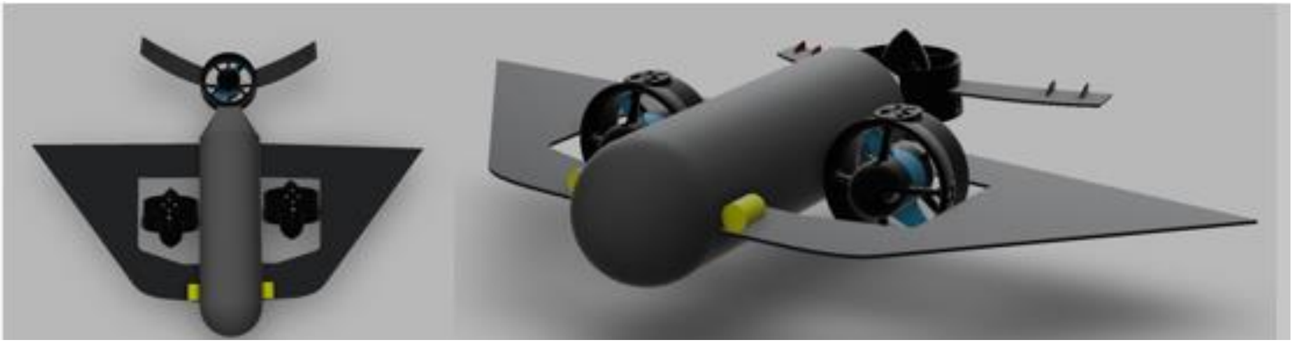


Figure 12: Code name “Batwing”. Two thrusters in front and one in back.

Figure 12. Focus was on having one motor in back and two in front. This was to maximize maneuverability and speed. The focus was to have a center hull with a wide angle lens at the front for maximum visibility.

Testing Model

Certain aspects are taken from each of the designs to come together with an initial testing model. Mark Buntsev and Judas Taylor took upon the responsibility to create the test model design. The models found in *figures 11* and *12* were the main designs used in the conception of the testing model. The idea was to have a streamlined body possessing hydrodynamic properties. The final decision was made to have two thrusters in the front and one in the back. As stated before, the concept was that the vehicle needs to be in forward motion to control direction. The two front motors would be the main driving force while the back motor controlled direction in the “y” axis. Some considerations that were explored and pondered over during the design process were the wire placements, enclosing the thrusters in case of impact, ease of combining all the parts and how waterproof the system would be. The design was made in SolidWorks, *Figure 13*, and 3D printed in Kingsbury Hall using black ABS plastic. This initial model had 4 total individually printed parts: two wings, a center hull, and a back piece.

Figure 13: Testing model 3D CAD design.

Figure 14: Watertight enclosure and electronics

The central hull contained a BlueRobotics 3” enclosure that safely kept all of the electronic components during underwater operations. A custom made electronics tray [5] was made out of acrylic and laser cut in an appropriate fashion. This electronic tray was built with screw holes coinciding with that of the end cap. This can be seen in *Figure 14* with all components that were placed inside.

The custom electronics tray was created with a piece of foam placed on the leading edge so the camera could be attached to extend out about an inch to get the full functionality of the wide angle lens on the selected camera. Velcro was used to keep the Raspberry Pi and bottom-side Fathom X in place during movement of the vehicle. It is important that all electric components stay in place to keep all connections solid without risk of tear out forces. The battery was found to fit snugly in the bottom portion of the enclosure.

Creating a test model first was imperative to having a physical understanding of what the final build would look like and how tight the tolerance could be. The wings were attached by having slots on either side of the outside of the center hull 3D printed piece. Along the inside of the center hull guides were created to allow for the wiring of the T100s to be fed though and wrap around in the back piece enclosure and through some penetrators within the watertight enclosure. Ample room was given in the back piece to allow for excess wire to be folded within. Through this initial testing model, it was seen that extra space needed to be added for the wiring to be fed through. Another modification that needed to be made was that the surrounding enclosures of the thrusters needed to be thicker in case of impact. Lastly, the wings were found to extend out too far from the center and needed to be adjust to minimize risk of breaking off in case of any sort of impact.

Overall the physical model a good transition into the final model because in SolidWorks the vehicle seemed to be thick enough all around, but upon further inspection the vehicle was found to be flimsy and at risk of breaking. Extra structural support was imperative to ensure that nothing would break in testing or if it were to be dropped.

Final Build

The final build, found in *figure 15*, was built in SolidWorks and implemented in design changes that added structural rigidity to the model, but also made the design more esthetically pleasing. This was done with the addition of curved edges to also improve hydrodynamic properties.

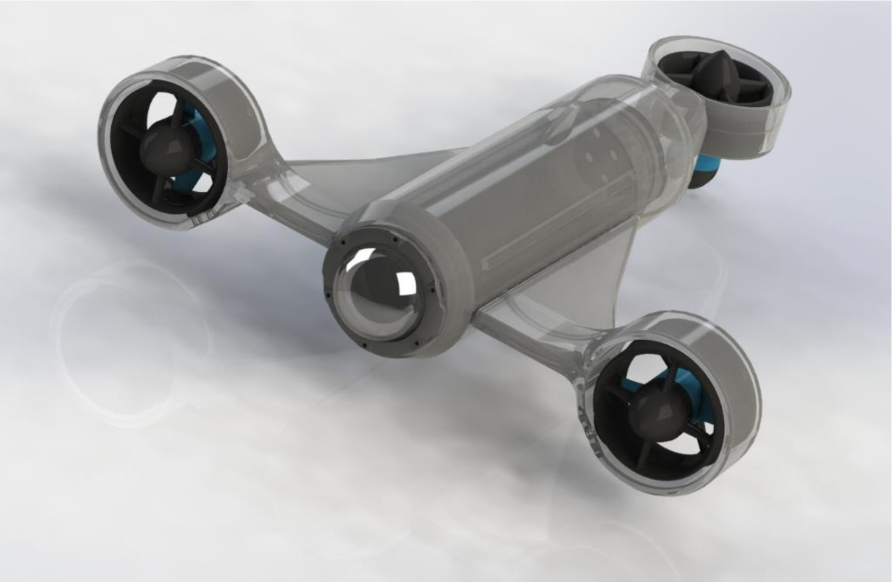


Figure 15: Final Rendering of design

Editing the dimensions was essential in creating a compact model that fit in all of the components properly without risk of wires falling out. Secondarily, the most important aspect of the project was to make it watertight. To do this, tighter tolerances were instilled into the design and the wires would be fed through the wings into the center hull and around the back. The ends were epoxied to guarantee that no water would spill in. A front cap was added to ensure a tight fit onto the center enclosure. A top tether relief was added onto the center hull to minimize the forces dragging onto the attached tether. Lastly a tether hole was added onto the back piece and epoxied around.

The final model design was split into 9 individual pieces for ease of mold and casting. Due to the complex design curves and edges, this was a necessary step to ensure smooth finishes without disconnects or unintentional bubbles to form. Each of the original 4 components were split in two along the horizontal axis. Each of these components were 3D printed using the 3D printer in Kingsbury Hall.

Wiring and Assembly

MV's design consisted of several components mounted to an acrylic tray and wired together in a cylindrical plastic, waterproof housing. This casing is mounted inside the hull of our vehicle and screwed into place to prevent component movement.

At the nose (front end) of the vehicle there is be a high-resolution 1080P Pi Camera mounted inside the cylinder with a clear port hole for maximum visibility (Appendix B1). The camera is wired directly to the Raspberry Pi and is attached to the front of the cylinder with Velcro for easy removal when necessary.

Figure 16: Assembled electronics tray (left)
Assembled electronics tray inside waterproof enclosure (right)

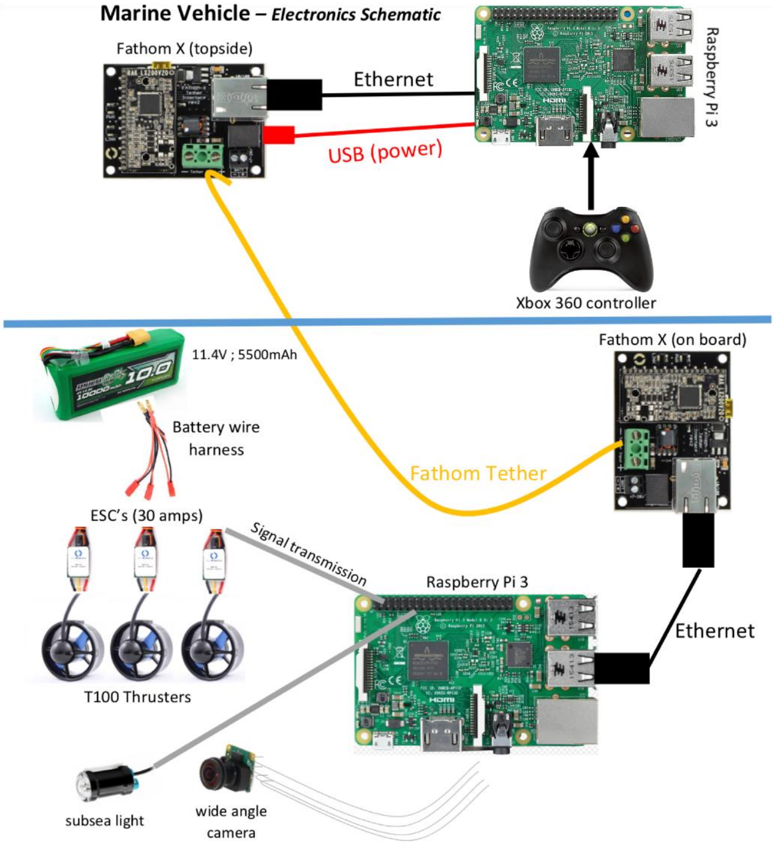


Figure 17: Schematic for electronics assembly

The Raspberry Pi on board the vehicle has an Adafruit 16-Channel PWM / Servo HAT mounted on top which allows the Pi to communicate with the Afro ESC's were are using to transmit PWM signals to our Blue Robotics T100 motors. On top of the Pi and HAT is the Fathom-X Tether Interface board which is connected via a standard Ethernet cable. The board transmits signals to and from the Pi up the tether cabling. At the top of the cable is another Fathom-X Tether interface board which connects to another Raspberry Pi via a standard Ethernet cable. The top side Raspberry Pi requires a monitor in order to display the camera forage from the vehicle underwater. The user then views the images transmitted from the Pi Camera and controls it with a wired Xbox 360 controller. All of the internal components are powered by an 11.1V, 5,500mAh battery which is housed internally in the vehicle's hull on the wiring tray.

Manufacturing

Overview

To effectively introduce the Marine Vehicle into schools, it needed to be cost effective, safe, durable, easy to assemble, and a platform for further exploration. The team implementing manufacturing skills its members learned in industry, was able to make design changes to meet all of these requirements. Through customer discovery and research, the cost to purchase and maintain is the first barrier for robotics programs. To adhere to a low cost budget, an investigation of the amount of material being used was conducted. The prototype was then simplified, removing aesthetic components, to reduce the material and build cost. Because much of the hardware is sourced from Blue Robotics, more than half the related costs of assembly are fixed. Material, time, and tools needed to assemble were aspects that could be re-engineered to increase the likelihood of bringing the Marine Vehicle into schools. With the amount of material reduced to the minimum required to hold structural integrity and perform, looking at the resources of the schools was the next step. Part of designing a product for manufacturability is understanding customer needs and their implementation. For school systems, teacher's time is expensive as well as the tools that may be needed to assembly a single unit. An assembly process was designed to reduce process time, and utilize common tools. Ideally, students will take part in the entire process of assembly and learn about many of the sub-systems and implementation of different fields of engineering in the process of prototyping a marine robotic vehicle.

Safety and reliability are important not only to the investment of the school, but to the students. The design utilizes a compartmentalized layout that separates each system. The electronics are built in with multiple layers of protection and catches for possible danger. When electronics and marine settings are combined, the risk of failure and dangerous accidents increase. The use of the inner waterproof housing not only protects the hardware, but is a barrier for students and participants to avoid any type of dangerous electrical interaction. By design, the thrusters are built into an enclosure that not only protects the blades, but inhibits students from hurting themselves while underway. Durability is a key aspect in protection of investment, when schools decide to utilize part of their budget, there needs to be a long term return of investment. MV designed housings for the T100 thrusters to protect against impacts, and used composite plastics to increase the yield stress of the structure.

The most difficult manufacturing engineering process was thinking in the future, analyzing the potential uses and applications of the Marine Vehicle, and developing a platform that can modularly adapt. Part of the process of developing interest in engineering is having the freedom is allowing a student to take an idea and implement it into something physical. That transformation alone is empowering and a necessary attribute that many STEM programs lack. The physical design and structure of the body features many flat surfaces that can be used to mount various instruments, lights and cameras. The tolerancing for all wiring through the internal channels of the body have been adjusted to account for future inputs. The hardware tray resembles a flow line construction, in that the final source is in the front (as access begins in the rear), and wiring works through critical components as it reaches the back and final destination.

Research

Though the 3D printed model was enough to showcase, the material properties did not suffice for underwater use. Research into manufacturing processes in the form of molding and casting needed to be done to essentially transition from the 3D printed model to a reliable and durable assembly.

Prior to finding the materials required for molding and casting, the understanding of how to both mold as well as cast was researched. This process if fairly simple: molding is the process of taking the part you want (positive) and creating a mold (negative). Depending on the geometry of each individual part, this process can get extremely complicated – some requiring a "two-part mold" where the part is split into two and molded individually where they are then reconnected and filled with the final casting material. The fundamentals of molding and casting can be found in figure 7 below which uses a symmetric and semi-complex geometry.

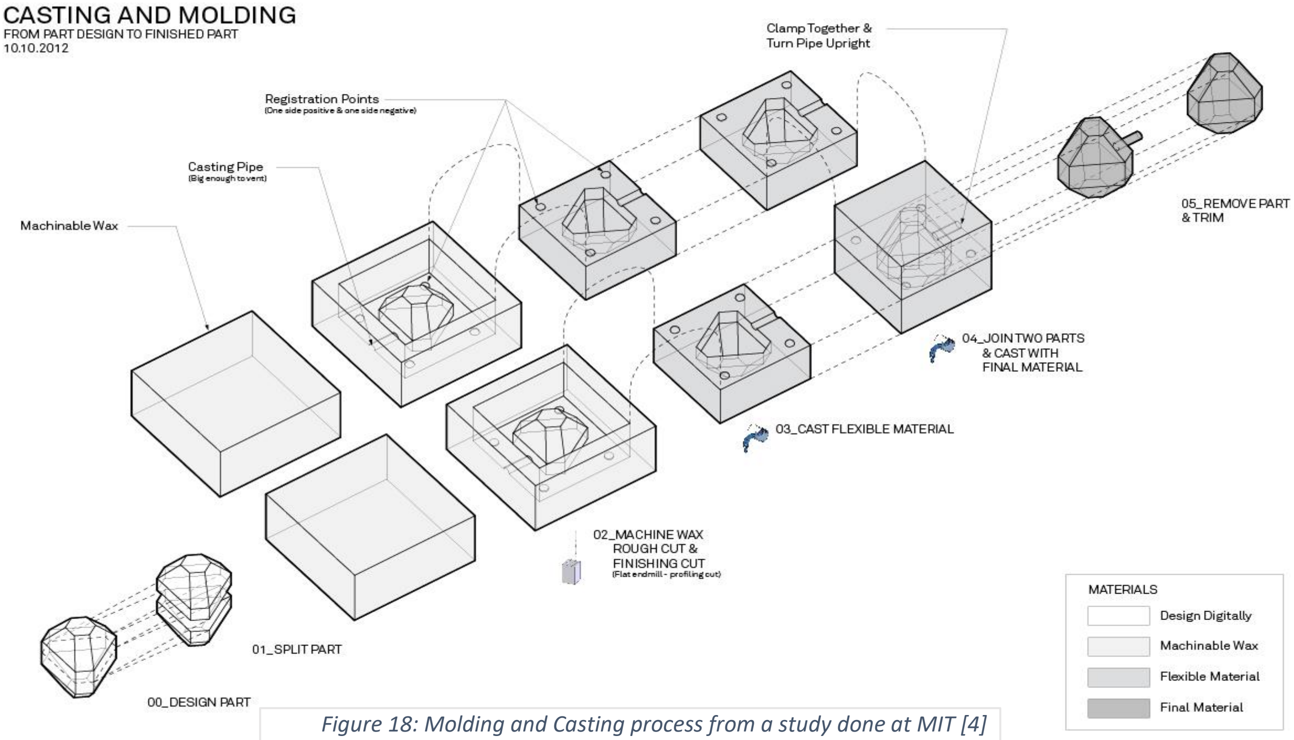


Figure 18: Molding and Casting process from a study done at MIT [4]

With a full understanding of the molding and casting process, the team then needed to follow up with selecting materials in which to mold and cast. Selecting a molding material required a few considerations to be well-thought-out. The biggest factor that required the most consideration in this portion of the project was the cost of the materials and trying to keep it low without sacrificing strength and durability. It is important to note that of the two necessary procedures, the molding process requires much more material in order to fill the volume when covering the part and is therefore, the more expensive of the two processes per volume.

Molding Selection

Thorough research on molding techniques gave rise to two options: two-part and brush-on molding. Due to the complex geometries and number parts, it was important to have a working plan using the selected technique, especially when using the two-part molding technique. The team had to pick a high quality, flexible and easy-to-use molding material which came down to two different silicon rubbers: Polymer Planet's RTV silicone rubber [7] & SRC's Cast-A-Mold Platinum [8].

The factor that helped decide on which molding material to use between the two mentioned above was cost. SRC is known for how easy and effective their products are but they're cost per volume was roughly 130% greater than Polymer Planet. As mentioned previous, the molding process was the most expensive in terms of manufacturing the body of the marine vehicle and for that reason, it was concluded that the two-part molding material was to be Polymer Planet's RTV silicone rubber. This silicone had a pot life (working time) of 30 minutes and a cure time of 16 to 24 hours.

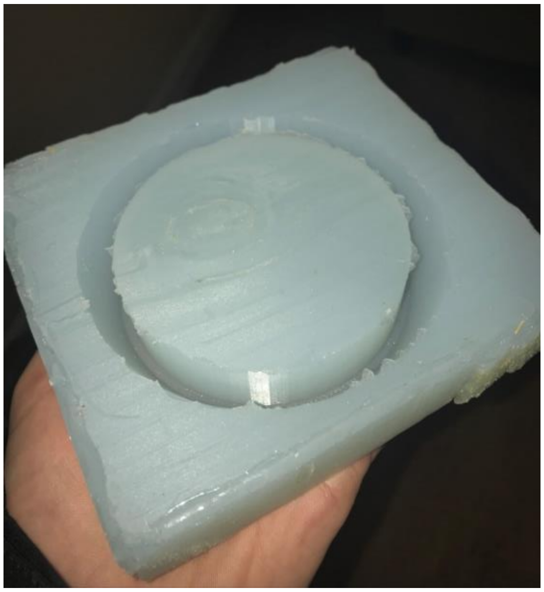


Figure 19: Finished mold of front adapter piece

Figure 19 (right) illustrates the finished mold used in making the adapter which goes in between the center hull and the front end of the waterproof enclosure. Note that in order to successfully mold the vehicle, it was necessary to split the rest of the 3D printed parts into two – avoiding the need for making two molds per part – whereas the adapter was a simple one-part mold.

Figure 20: Application of the brush-on silicon mold

MV attempted to mold the rest of the parts using the two-part method but had difficulty with some of the geometries, specifically the back piece which holds the rear thruster in place as well as closes off the rear end of the marine vehicle. This issue led the team to begin working with the brush-on silicon rubber mold from Smooth On, specifically Oomoo 25. This method involves brushing layers of molding material onto each individual piece without the need for filling/pouring. Not only was this method simple and required a simple 1:1 ratio of part A to part B, but it also had a shorter pot and cure time of 15 minutes and 6 hours, respectively. The downsides to this method were the relatively higher viscosity making it challenging to stick and stay on vertical surfaces as well the need for multiple (up to 8) layers. Adding thickener to the mixture lowered the viscosity of the silicon, thus correcting this issue. The application of this process can be seen in figure 20 (left).

Casting Selection

Casting selection was the easier side to this process as typical casting resins are durable and have high tensile strength. Prior to making this selection, the team looked at finding a casting resin that followed the same criteria as the molding material in terms of non-material properties (cost and ease of use) with a focus on making the internals visible from the outside in case of failure; in essence, safety.

Figure 21: Finalized adapter using casting resin

It was concluded that the most cost effective casting resin was SRC’s Color Pro Semi-Clear casting resin due to its high tensile strength of 225 kpi – Rockwell hardness of 75-D. Not only did these material properties meet the team’s criteria, but it was priced at a very reasonably. The work time for this product was 2 minutes while the cure time was a quick 15 minutes. This short cure time allowed for testing of the material – the finished adapter using the casting resin can be seen in figure 21 (left). Though it was recommended degassing the material in a vacuum chamber, the team concluded that it was unnecessary for this application after the first completed cast.

Looking Ahead

Parts which have been fully molded and casted include the adapter, the center hull, and the wings. MV is attempting to find a simple way to mold and cast the rear piece and are currently leaning towards reprinting the part but splitting it differently. There was also an issue fitting the tether through the rear of the design – this was due to the watertight enclosure being pushed too far back. To fix this, MV will extend the adapter in order to push the face in contact with the watertight enclosure further from the center hull.

Coding Integration

Communication

Both of the Raspberry Pi devices used for our vehicle were setup nearly identically. The only difference between the two computers are their static IP addresses. Static IP addresses were set so that they could communicate to each other over Ethernet through the tether interfaces. Outline of each Pi's /etc/network/interfaces file can be found in appendix C1.

After both Pi devices have their respective /etc/network/interfaces file written correctly, a restart is required for each. Once fully connected through the Fathom X interfaces, tether, and Ethernet cables, an SSH session is established from the top side Pi to the submerged Pi for full control over the submerged Pi from the surface.

Streaming Images

Video streaming was done with Gstreamer – a tool for manipulating video streams. The video stream was forwarded from the submerged Pi to the top-side Pi with the use of the Fathom X interface boards which provide the ability to communicate through the tether. The command for downloading the Gstreamer library and code are located in appendix C2.

Two bash scripts (appendix C3) were written in order to execute and receive the stream, one for each respective Pi. The submerged Pi executed the Gstreamer script which then output the stream topside.

Once both scripts are executed, the video feed was automatically display on the user's screen.

Controller

Control inputs were handled by an open source python library xboxdrv located in a GitHub repository [4]. The command for installing xboxdrv can be located in appendix C4.

The controller driver handled inputs from the controller and translated them into decimal numbers assigned to specific variables. For example, when the left trigger is pressed, a value of 1.000 is outputted to the left trigger assigned variable. The values of the triggers and joysticks are modular, so they can have values which are not whole numbers. The values of the joysticks range from -1.000 to 1.000, and the values of the triggers range from 0.0 to 1.0. When a trigger is half pressed down, it outputs a value of 0.500. A quarter press is 0.250, and so on.

The xboxdrv was used in a python script to handle controller inputs and convert the scale to the required for BlueRobotics T100 thruster inputs.

Raspberry Pi Setup (ESC's)

The Adafruit python library was used to communicate with the ESCs which controlled the T100 BlueRobotics thrusters. Commands for setting up of the library on the Raspberry Pi can be found in Appendix C5.

There was a python code written to run the vehicle (see appendix C6) which had many different functions. First, it initialized the Xbox 360 controller followed by the PWM Servo Hat pins to ensure the Hat is setup correctly. The ESC frequency was set to 48 as a factory default. The center, calculated with a series of equations, was set and used to determine what value is assigned to "stopped" for the T100 thrusters. Generally, it is around 300 to 310. This meant that anything in the 200-range signals the thruster to run in reverse, and anything above the 300-310 range signals the thruster to run forwards. The greater the magnitude, the faster the thruster will spin in its corresponding direction. Each thruster was then initialized based and set to its corresponding pin on the Pi Hat. Pins 0 and 1 were to the left and right thruster, respectively; pin 3 was set to the rear thruster. Finally, the program accepted inputs from the Xbox 360 controller in the -1.000 to 1.000-range. It converted the controller range to the 300-range suitable for the T100 thrusters. Use of a gear variable set the overall scale of the thruster speed with the default set to 20 meaning the thrusters were allowed to run from 280 (reverse by 20) up to 320 (forward by 20). The gear variable can be set higher or lower, depending on how quickly the operator would like the vehicle to move.

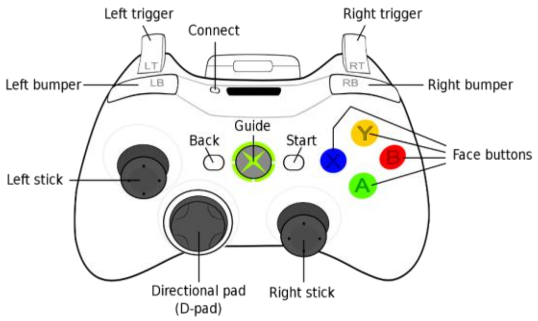


Figure 22: Xbox 360 Button Layout

The left sticks X axis of movement controlled the vehicles yaw (left or right). The right sticks Y axis of movement controlled the vehicles pitch (rotate up or down about the vehicles X axis). Exiting the program was set two different ways: by pressing the back button on the Xbox 360 controller or as a failsafe method by pressing the ESCAPE key on the keyboard linked to the top-side Raspberry Pi. Refer to *figure 9* for the Xbox 360 button-layout.

Issues & Solutions

There were countless hours spent writing code in different languages, and a majority of the code written was done so in a "trial and error" fashion. With that being said, listing every single problem encountered would be repetitive and difficult to accomplish. Two of the largest problems faced involved controller integration and camera streaming.

The first road block was forwarding the controller inputs from the surface Pi to the submerged Pi. A normal SSH session from the surface to the vehicle does not carry the Xbox 360 controller's USB inputs to the Python xboxdrv library. The program written to handle control inputs and outputs to the thrusters would not recognize that there was a controller connected to the surface Pi. The solution was to use a piece of software called VirtualHere which created a USB Server on the submerged Pi and enables remote access to USB devices over a network. This allowed the submerged Pi to recognize the Xbox 360 controller's inputs through the surface Pi's SSH connection as if it were directly wired to the submerged Pi.

The surface Pi's startup script was edited in order to automatically start the VirtualHere software on its initial boot (Appendix C7).

The second roadblock was with the camera stream. The original thought process was to use any computer with any operating system for the surface control center instead of the surface Pi. The main problem with this was that Raspberry Pi devices run on Raspbian, a Linux operating system variation, which communicates best with devices on the same OS. It was anticipated that Mac OS X would communicate well with the submerged Pi, but this proved more difficult than anticipated. The solution was to use a second Pi (now the surface Pi) to receive the stream from the submerged Pi.

This solution did allow for streaming capabilities, but it was not perfect. The Raspberry Pi's processing power was not sufficient enough resulting in the Raspberry Pi reaching maximum capacity. The Pi camera does not have a built-in processor, so the submerged Pi was used to process the HD images. The submerged Pi was also tasked with forwarding the raw data up the tether and to the surface Pi – on top of handling inputs from the surface Pi and outputting them to the thrusters.

This large combination of tasks, which are performed simultaneously, caused two problems. The first problem involved the submerged Pi device's streaming framerate to drop slightly. The second problem was that the submerged Pi would get very hot when these tasks were performed for an extended period of time. A solution currently being worked on is to implement the use of a different camera with a built-in processor. The idea is that the camera's processor will handle all of the video processing, so the submerged Pi will only be tasked with forwarding the already processed images up the tether to the surface Pi. The camera with would most likely be the Pixy CMUcam5 Sensor (figure 23).

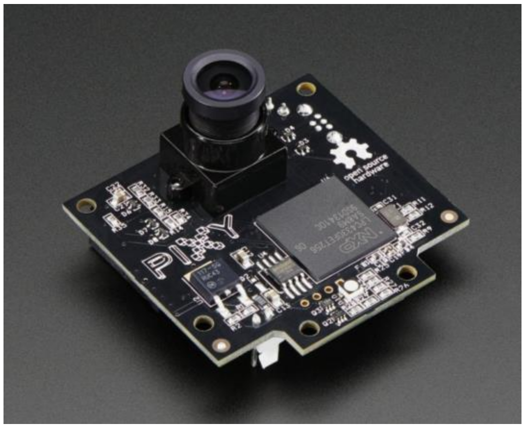


Figure 23: HD camera with a built-in processor

Conclusion

The initial focus of SAV was to pursue a design and protect its intellectual property rights. Gaining experience in bringing a product to market not only develops skills in design and implementation, but testing, and most importantly the legal and business side of engineering. SAV was a platform for interdisciplinary study and exploration. As described earlier, pursuing a revolutionary platform that can both perform in the air and water did not align with the timeline left in the academic year. Changing scope to pursue the Marine Vehicle not only allowed for a successful prototype, but still was an applicable design to pursue IP rights. The market for marine vehicles with an emphasis on speed and maneuverability (non-ROV platform), is still new. There is one product available in the market, with two soon to be introduced. The new competition fostered an entrepreneurial setting throughout the year and pushed the team to prototype a function design. At the production cost, the Marine Vehicle will soon become a product at a price that wasn't otherwise available.

During the research process, the MV team reached out to local robotics programs, marine robotics programs, and small companies that could double as a customer discovery process and an invaluable resource of knowledge. Throughout this process an available market niche became apparent; school systems with integrated robotics programs could only offer their students basic educational tools. Programs like SeaPerch offer ROV experience at a middle school level, and students that master that platform do not have access to more complicated systems. The availability of educational programs in high schools that introduce young adults to engineering will be key in the future of STEM based learning in the United States. MV began to modify their prototype to become a viable option for students interested in exploring robotics to get excited about the future and their education in engineering. There are a handful of key attributes of the Marine Vehicle that will captivate potential students, the most important, it being fun. High school students need to not only be able to explore the engineering behind the product, but to have fun.

The team reached out to local high schools and programs to learn more about how the integration process would work, and to assess interest in the Marine Vehicle. Most robotics programs expressed high levels of interest, the US Albacore Program felt it was a fantastic platform to catch the students that "burn out" after programs that hook them in the years before college. In the last weeks of the academic year, MV intends on travelling to schools and demonstrating the capabilities of the prototype and how it would empower students. An important part of the integration is utilizing the talents of the teachers' native to the programs, and showing them the prototype is reproducible and effective.

The Marine Vehicle is a platform for furthering education in all facets of engineering, design, and industry. Students can not only assemble the vehicle, but use it as a base structure to integrate further functions such as more complex control dynamics, sensors, autonomy, and design changes to improve speed and agility. Those interested in mechanical engineering and manufacturing can explore the assembly, necessary CAD designing, material analysis, and most importantly the connection between electrical, mechanical, and software. Those interested in robotics and electrical engineering can not only assemble the internal hardware, but tackle all the needed improvements to make the prototype unique. Students with a passion or a desire to learn software applications and coding can integrate more complex control dynamics and program the additions of features and adjustment of attributes.

MV

The various core engineering focusses are all being explored in a marine environment, making man tasks more complex, but exemplify the range and application of robotic engineering. An important aspect of STEM education, and a key contributor to students desire to complete degrees in the different fields of engineering.

References

[1] “Communication with Submarines.” *Wikipedia*, Wikimedia Foundation, 10 Apr. 2018, www.en.wikipedia.org/wiki/Communication_with_submarines.

[2] TyrantBB. “ROV Thruster - Modified Bilge Pump.” *YouTube*, YouTube, 23 Mar. 2013, www.youtube.com/watch?v=DBNDEQjZI84

[3] “T100 Thruster.” *Blue Robotics*, 2018. www.bluerobotics.com/store/thrusters/t100-thruster/.

[4] “Thrusters & Motors.” *T100 Thruster Documentation*, 2018. www.docs.bluerobotics.com/thrusters/#performance-charts.

[5] Thingiverse.com. “Electronics Tray for Blue Robotics 3’ Enclosure by Dreadnaught52.” *By Dreadnaught52 - Thingiverse*, www.thingiverse.com/thing:2718527.

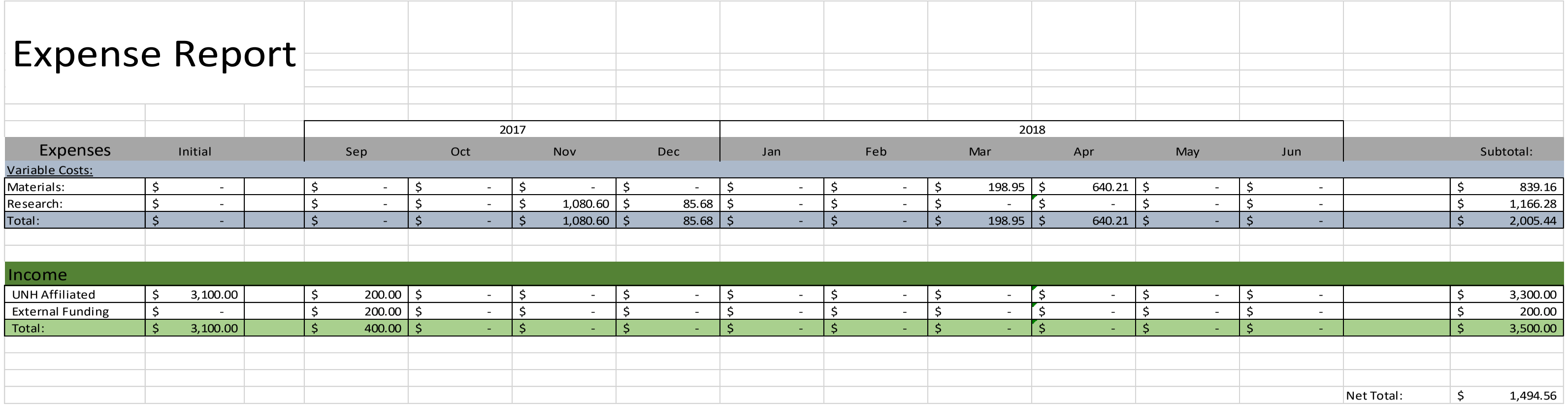
[6] “Stingrays.” *National Geographic*, 26 June 2017, 2015-2018. www.nationalgeographic.com/animals/fish/group/stingrays/.

[7] “RTV SILICONE LIQUID RUBBER MOLD MAKING KIT - HIGH STRENGTH.” *Polymer Planet*, 2018. www.polymerplanet.com/products/rtv-silicone-liquid-rubber-rtv-mold-making-quart-kit-high-strength?variant=6044853993507.

[8] “Color-Pro Semi-Clear – Specialty Resin & Chemical - Epoxy, Silicone Rubber.” *Specialty Resin & Chemical - Epoxy, Silicone Rubber*, 2017. www.specialtyresin.com/product/color-pro/.

Appendix

(A1) Expense Report

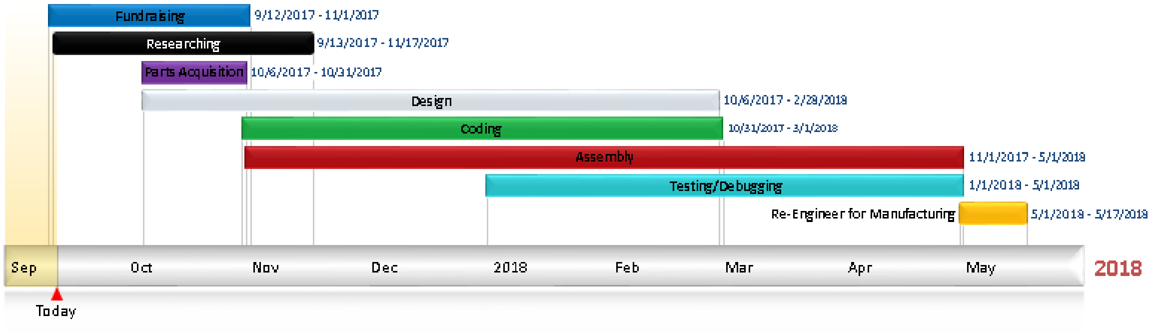


Expense Report

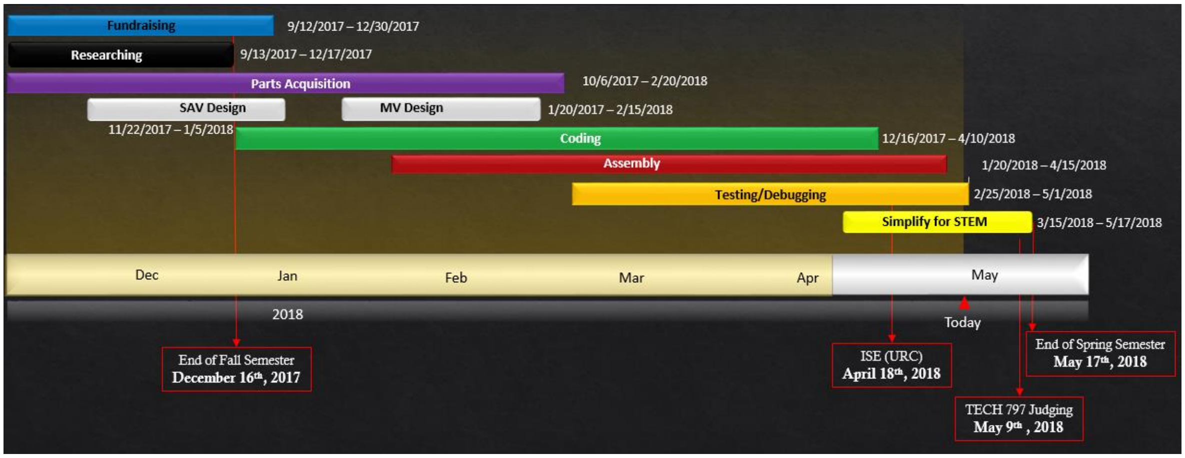
			2017				2018							
Expenses	Initial		Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun		Subtotal:
Variable Costs:														
Materials:	$ -		$ -	$ -	$ -	$ -	$ -	$ -	$ 198.95	$ 640.21	$ -	$ -		$ 839.16
Research:	$ -		$ -	$ -	$ 1,080.60	$ 85.68	$ -	$ -	$ -	$ -	$ -	$ -		$ 1,166.28
Total:	$ -		$ -	$ -	$ 1,080.60	$ 85.68	$ -	$ -	$ 198.95	$ 640.21	$ -	$ -		$ 2,005.44
Income														
UNH Affiliated	$ 3,100.00		$ 200.00	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -		$ 3,300.00
External Funding	$ -		$ 200.00	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -		$ 200.00
Total:	$ 3,100.00		$ 400.00	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -	$ -		$ 3,500.00
													Net Total:	$ 1,494.56

Appendix Figures 1: Expense Report

(A2) Updated Gantt Charts



Appendix Figures 2: Original SAV Gantt Chart



Appendix Figures 3: Most recent MV Gantt Chart

(A3) Flow Simulation using OpenFOAM

Fluid Analysis Simulation of T100 Underwater Thruster

Mark Buntsev
University of New Hampshire
Mechanical Engineering
Department
Durham, NH USA

Objectives and Background:

The T100 Thruster created by BlueRobotics is an essential element to the S.A.V. (Submersible Aerial Vehicle) Team at the University of New Hampshire. It is the main driving force of the prototype underwater drone and is essential in the success of a properly working model. When constructing the design of the apparatus, the fluid properties of a forward moving prototype is essential in the revision processes from initial design. The prototype was designed to travel an anticipated 2 m/s, however was simulated at 3 m/s to better visualize the effects of fluid flow at a heightened speed (worst case scenario).

Through analysis of the T100 thruster within OpenFOAM, a firmer understanding of fluid direction can be seen and necessary revisions can be made to the prototype design of S.A.V.s drone. The aerodynamics underwater are essential in determining the functionality and expected path of the apparatus underwater.

Simulation Setup:

The T100 thruster CAD design was taken from the BlueRobotics website and imported into Solidworks. A design can be seen in *Figure 1*.

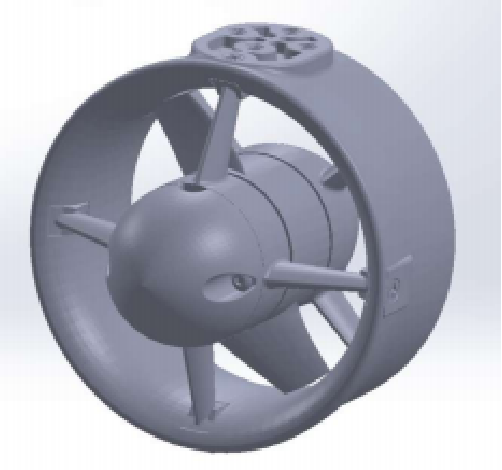


Figure 1. T100 Thruster with Blue ESC

The file was taken and exported out as a very finely meshed ".STL" file. To ensure there would be no holes in the mesh, the model was then taken, imported into an opensource code software known as "Blender" and once again exported out again as a ".stl" ASCII file.

Once a finely meshed file was taken in, it would be brought into OpenFoam. In OpenFoam, an outside box needed to be created to house the T100 thruster which was set to a cube bounded in the X, Y and Z coordinated of -100 (mm) to 200 (mm). Once these parameters were set in the *BlockmeshDict*, blockmesh function was run and the outside box was set.

To import in the Thruster geometry, *SnappyHexMesh* was used. Initially to have all the necessary files, *SurfacefeatureExtract* needed to be used to create all appropriate "trisurface" files. *SnappyHexMesh* function was then used to import in the geometry within the bounds initially set. The refinement level was set to level 2 within 1 mm of the intersecting geometry. This could be refined further down, however, with the number of cells being analyzed and created within this complex geometry, level 2 was deemed accurate enough for preliminary analysis of fluid flow. The entire thruster was set as a wall boundary, assuming that there would be no fluid properties emanating out from the solid body. While the thruster would be operating underwater, the fluid properties were just set to that of air (also a Newtonian fluid) just for visualization of the flow. To solve for the cell arrays, a SIMPLEfoam solver was implemented using a *k*- ω SST model. The model went through 50 iterations to bring the system close to convergence without unreasonably long computing times.

To analyze the flow patterns, a few different tools were implemented for visual representation. To visualize the image as a whole, the simulation was imported into ParaView and set to a Wireframe visualization setting, and the velocity and pressure fields were viewed, the slice feature gave accurate representations of the velocity profile at any 2D plane, and the streamline feature visually represented the direction of flow.

Results and Discussion:

OpenFoam provided good visual and numerical representation of the flow direction with respect to the T100 Thrsuter. This could be used to better design a prototype and have a firm understand as to which direction the drone would travel with no forward forces and no movement of the thrusters. Initially, the model was seen in a wireframe view, *figure 2*. Here it can be seen that the flow speeds up along the boundaries of the inside of the thruster. This was good to see because this signifies that more waterflow is going through the enclosed thruster than if there was no outer ring encasing it.

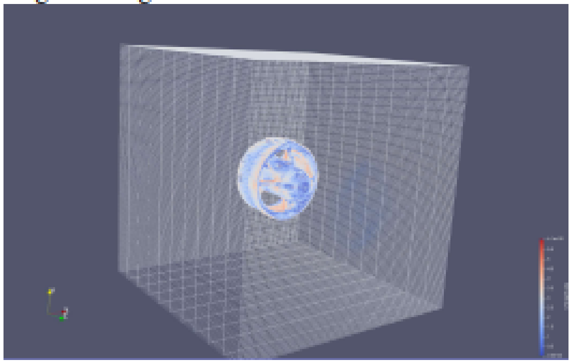


Figure 2. Wireframe image of T100 Thruster

With this visualized analysis, some design modifications that could be implemented are a conical design to the inlet of the thruster. This would maximize water intake to give the thruster more "pushing" power by giving it more of a substance to push. This would require additional design elements and further simulation to numerically justify how beneficial this modification would be.

The slice feature was then used to visualize the velocity profile before and after the fluid reaches the thruster. This can be seen in *figures 3 and 4*, before and after respectively. The "before" section signifies that there is no fully developed flow going into the system. This could be modified by placing a meshed/honeycombed inlet into the thruster. This would allow for more unified flow to in turn allow for more simplified linear movement.

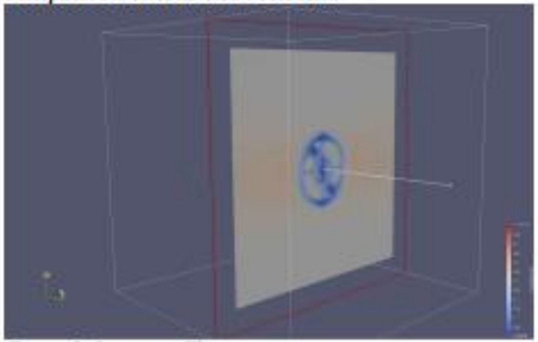


Figure 3. Incoming Flow

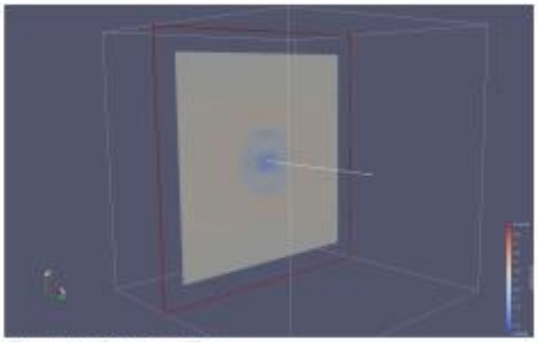


Figure 4. Outgoing Flow

The quickened flow extends out to the sides in the X direction, while the flow slows down to the cross sectional shape of the thruster. After the outlet of the thruster, the flow becomes more uniform along the z axis. This means that the outwards facing flow is easily controllable. This could be improved in later prototypes by building flaps in front of the thruster, controlled by servos, to better direct a drone. With minimal turbulence of the back, the front needs to conform to the desired direction and the outlet will behave as the front directs.

A streamline plot represents the direction of the streamline traces about an object. *Figure 5* represents the flow in the X direction and how it spreads out when the flow met with the thruster.

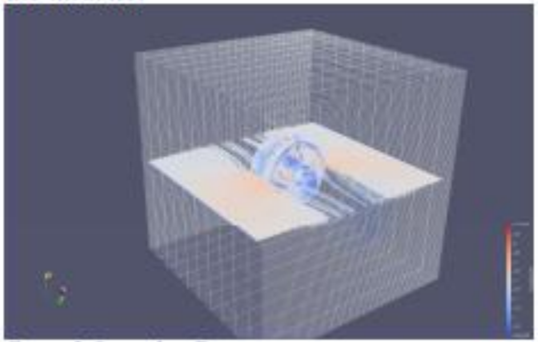


Figure 5. Streamline Tracers

As stated before, possible design modifications to the thruster would be to implement a conical inlet design to force more fluid in through the powered thruster. The simulation proves that a majority of the fluid is forced outside causing the surrounding fluids to speed up.

Conclusion:

The simulation proved to be a success in analyzing the effects of constant flow into a T100 Thruster. Further research is required in optimizing the design of the thruster, however, enough background knowledge is acquired to further modify the prototype design of the S.A.V. Some modifications that could be made are a conical inlet into the thruster to maximize the effect of the powered propeller and to created a honeycomb inlet of the thruster to provide more uniform fully developed flow. Again, additional research will be required to make an optimal design, but preliminary steps were taken as a basis for future design modifications.

(A4) Testing of Aerial Propellers

Appendix Figures 4: Testing of Aerial Propellers

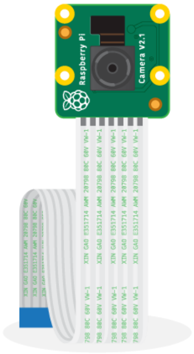
(A5) Propeller Testing (Code)

```
% SAV - Propeller Testing (Air and Water)

clear all; close all;

Mass = [.1076 .2076 .3076 .4076 .5076 .6076 1.0076];
V = -[7.9846 7.7608 7.6489 7.5385 7.4275 7.3152 6.8694];
Vout = V+7.9932;
Newton = 9.81*Mass;

CON = Vout./Newton;
CONVERSION = .1137;

% air test 1045 %
A_1045 = -([7.861 7.727 7.534 7.343 7.147]-7.9471);
A_lbf_2 = (A_1045)./(4.44822*CONVERSION);
A_input_2 = [30 60 90 120 150];
A_pwm_2 = ((A_input_2./180)*1000)+1000;

% air test 7038 %
A_7038 = -([7.89 7.848 7.78 7.71 7.615]-7.91);
A_lbf_3 = (A_7038)./(4.44822*CONVERSION);
A_input_3 = A_input_2;
A_pwm_3 = ((A_input_3./180)*1000)+1000;

% water test 7038 %
W_7038 = 7.968-[7.679 7.455 7.501 7.374 7.388 7.488 7.455];
W_lbf_3 = (W_7038)./(4.44822*CONVERSION);
W_input_3 = [10 20 30 40 50 60 90];
```

```
W_pwm_3 = ((W_input_3./180)*1000)+1000;


% PLOTS %

figure(1);
plot(A_pwm_2,A_lbf_2,'r*','Markersize',7); grid on;
xlabel('PWM Wave [milliseconds]')
ylabel('Resulting Force [ lb_f ]')
title('Propellor 1045 (Air)')

figure(2);
plot(A_pwm_3,A_lbf_3,'r*','Markersize',7); grid on;
xlabel('PWM Wave [milliseconds]')
ylabel('Resulting Force [ lb_f ]')
title('Propellor 7038 (Air)')

figure(3);
plot(W_pwm_3,W_lbf_3,'b*','Markersize',7); grid on;
xlabel('PWM Wave [milliseconds]')
ylabel('Resulting Force [ lb_f ]')
title('Propellor 7038 (Water)')
```

Design Appendix Section

(B1) HD Raspberry Pi Camera



Appendix Figures 5: Raspberry Pi Camera

Coding Appendix Section

(C1) Setup for Raspberry Pi's Network Interface Files

Topside Pi

```
auto eth0
     allow-hotplug eth0
     iface eth0 inet static
     address 192.168.0.10
     netmask 255.240.0.0
```

Submerged Pi

```
auto eth0
     allow-hotplug eth0
     iface eth0 inet static
     address 192.168.0.12
     netmask 255.240.0.0
```

(C2) Gstreamer Library & Code Download and Setup

```
sudo apt-get install gstreamer1.0-tools
sudo apt-get install gstreamer1.0-plugins-good gstreamer1.0-plugins-bad
gstreamer1.0-libav
```

(C3) Bash Scripts for Streaming

Submerged Raspberry Pi

```
#!/bin/bash
pkill -INT tee
pkill -INT raspivid
pkill -INT gst-launch-1.0

# Wait for the process to clean up and terminate
sleep 1

raspivid -t 0 -w 1080 -h 720 -fps 25 -hf -vf -awb off -awbg 1.9,1.6 -drc    high
-b 2000000 -o - |
gst-launch-1.0 -v fdsrc ! h264parse ! rtph264pay config-interval=1 pt=96 !
gdppay ! tcpserversink host=192.168.0.12 port=5000 &
```

Topside Raspberry Pi

```
#!/bin/bash
until 1; do
      gst-launch-1.0 -v tcpclientsrc host=192.168.0.12 port=5000 ! gdpdepay !
   rtph264depay ! avdec_h264 ! videoconvert ! autovideosink sync=false
      echo "Restarting process in 2 seconds..."
      sleep 2
done
```

MV

(C4) Installing xboxdrv Library

```
sudo apt-get install xboxdrv
```

(C5) Setting Up Library on Raspberry Pi

```
sudo apt-get install git build-essential python-dev
cd ~
git clone https://github.com/adafruit/Adafruit_Python_PCA9685.git
cd Adafruit_Python_PCA9685
sudo python setup.py install
```

(C6) Submerged Pi Running Code

```
#!/usr/bin/python
from Adafruit_PWM_Servo_Driver import PWM
import time     # may not be necessary for final build
import xbox


joy = xbox.Joystick()

# Define the hat over the I2C connection pins
hat = PWM(0x40)

# Set the desired frequency for the ESCs
f = 48
hat.setPWMFreq(f)

# ================== CONTROLL THRUSTERS ============================

# Deadzone is 1500 microseconds. Calculate the tic number and store as center
f =48.00000          # decimals force precision

center = 1500        # microseconds

pulsetime = 1/f      # length of pulse (s)
pulsetime *= 1000    # length of pulse (ms)

tictime = pulsetime/4096    # time of each tic (ms)
tictime *= 1000             # time of each tic (microseconds)

center /= tictime           # set center to tic of 1100 microseconds

center = int(center)
#center = 310

# Define the thruster channels
thruster1 = 0
thruster2 = 1
thruster3 = 2

# Initilize
print "Initilizing ..."
hat.setPWM(thruster1, 0, center)
hat.setPWM(thruster2, 0, center)
hat.setPWM(thruster3, 0, center)
t1curSpeed = center
```

```
t2curSpeed = center
t3curSpeed = center
gear = 20

if joy.connected():      #tests connection to Xbox controller
    print("Joy Connected")
else:
    print("Joy Disconnected")

try:
    while not joy.Back():          #cancels program with select button

        if abs(joy.leftY()):               #handles horizontal motion

            hat.setPWM(thruster1, 0, center + gear*joy.leftY())
            hat.setPWM(thruster2, 0, center + gear*joy.leftY())
            t1curSpeed = center + gear*joy.leftY()
            t2curSpeed = center + gear*joy.leftY()
            if joy.leftX() < 0:
                hat.setPWM(thruster2, 0, t2curSpeed + abs(gear*joy.leftX()))
            if joy.leftX() > 0:
                hat.setPWM(thruster1, 0, t1curSpeed + abs(gear*joy.leftX()))

     if abs(joy.rightY()):         #handles vertical motion

            hat.setPWM(thruster3, 0, center + gear*joy.rightY())
except KeyboardInterrupt:          #cancels program with ESC key
    hat.setPWM(thruster1, 0, center)
    hat.setPWM(thruster2, 0, center)
    hat.setPWM(thruster3, 0, center)
```

(C7) Start Virtualhere Software

```
wget https://www.virtualhere.com/sites/default/files/usbserver/vhusbdarm
sudo chmod +x ./vhusbdarm
sudo mv vhusbdarm /usr/sbin
wget http://www.virtualhere.com/sites/default/files/usbserver/scripts/vhusbdpin
sudo chmod +x ./vhusbdpin
sudo mv vhusbdpin /etc/init.d
sudo update-rc.d vhusbdpin defaults
```