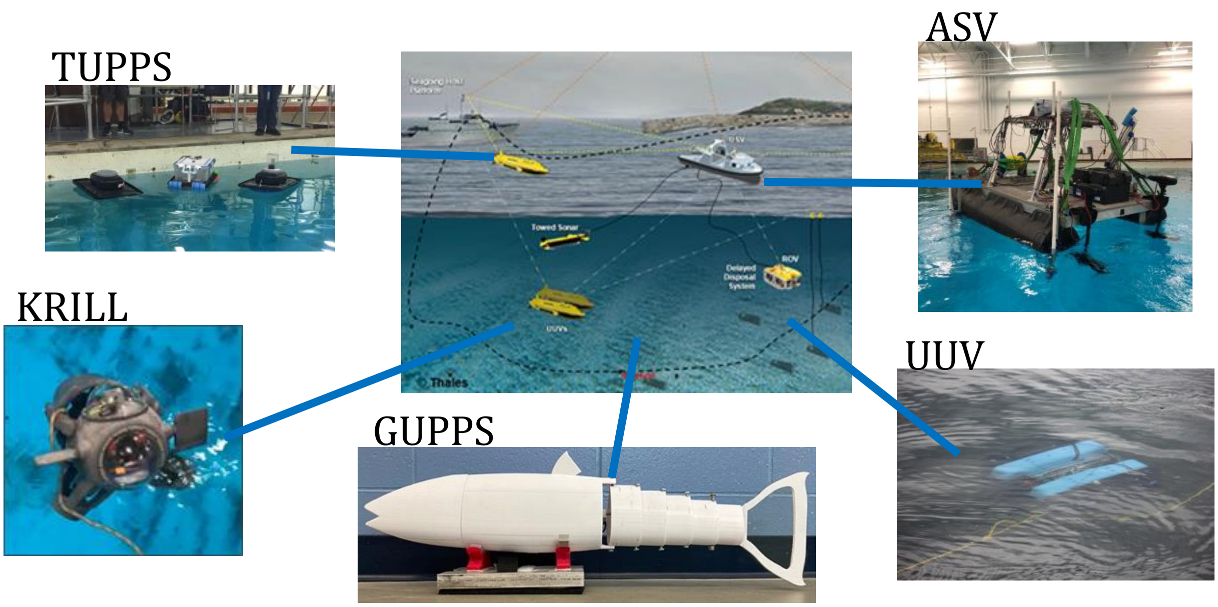
MANTA RAY

GUPPS: Ghost Unpiloted Performance Platform Submersible



Joshua Minard (ME), Stephanie Rota (OE), Margaret Enderle (OE), Thomas Oliver (ME),

Mia Seitz (Marine and Estuarine Biology), Eli Lemieux (ME), Ethan Danielson (ME),

Samuel Whitney (OE)

Project Advisors: Dr. May-Win Thein, Dr. Yuri Rzhanov, Dr. Elizabeth Fairchild

Abstract

As a part of the Marine and Naval Technological Advancements for Robotic AutonomY (MANTA RAY), the Ghost Unpiloted Performance Platform Submersible (GUPPS), is one member of a fleet of autonomous marine robots. The goal of GUPPS is to create an unobtrusive method to survey the environment through a variety of sensors using camouflage and biomimicry. Previous work was completed to achieve biomimicry in appearance and movement of the fish's body. This years team developed the previous year's design to create a robotic fish with the ability to propel itself through water. Working in parallel, two teams formed designs for propulsion, one using an internal propeller with a rudder, and the other imitating the natural motion of a fish's tail. In conjunction with the lateral motion, a buoyancy control device is being developed to enable the control of vertical motion. This year's work will enable further development of robotic autonomy of GUPPS to covertly monitor its surroundings and aid in the MANTA RAY mission.

Introduction

After much deliberation, the team came together and addressed three different problems: buoyancy, propulsion, and biomimicry. To solve these problems, the main team split up into three sub team: the buoyancy team, the propulsion team, and the tail team. The buoyancy focused on developing a system to enable GUPPS to move up and down the water column. They did this by changing the volume of a bag, fluctuating the buoyancy/ ballast in the fish. Secondly, the propulsion team set out to make a robotic fish using a Hobby Underwater Propellor. This would be controlled with a microcontroller that limits the speed of the propellor. Direction change would be controlled with a servo to act like a rudder. Lastly, the tail team's goal was to make a robotic fish as accurate as possible to a real-life steelhead trout. This includes using the sweeping motion of the tail to propel the fish forward. The team took inspiration from the previous year's and expanded on it. The three groups worked in parallel with the ultimate goal of potentially combining the three products into one.

Objectives

This year, the overall goal for the project was to have GUPPS propel itself 1 meter down in the water in a straight line. The mechanical goal of GUPPS is to have a system that allows the entire body to be propelled and controlled through the water. This system has to be small enough to fit within the shell of the fish with ample room for additional mechanical systems, such as sensors to collect data. The tail system needed to have a biomimetic component to propel GUPPS through the water using the tail's motion to create the driving force. The appendage needed to have enough surface area to displace water to achieve propulsion but also needed to have a strong enough motor to overcome the drag forces imposed by the water. Then the buoyancy team needed to create a system that would introduce ballast to lower it in the water or buoyant forces that would suspend the water higher toward the surface of the water.

Approach/Methods

To replicate the vertical movement of a Steelhead Trout in the water, a buoyancy device was made with the use of compressed air. After talking to graduate advisors for this project a system was created with 2 solenoids, an air bladder, compressed air (CO_2 cartridge), valve for CO_2 canister, inflation "tank" (PVC pipe), an Arduino Uno, a 7.4V LiPo battery, and connection tubes. The air was held back with 2 solenoids with the vent (#1 on the diagram of Figure 1 on the first one being blocked off to make sure the air would get built up first before entering the system.

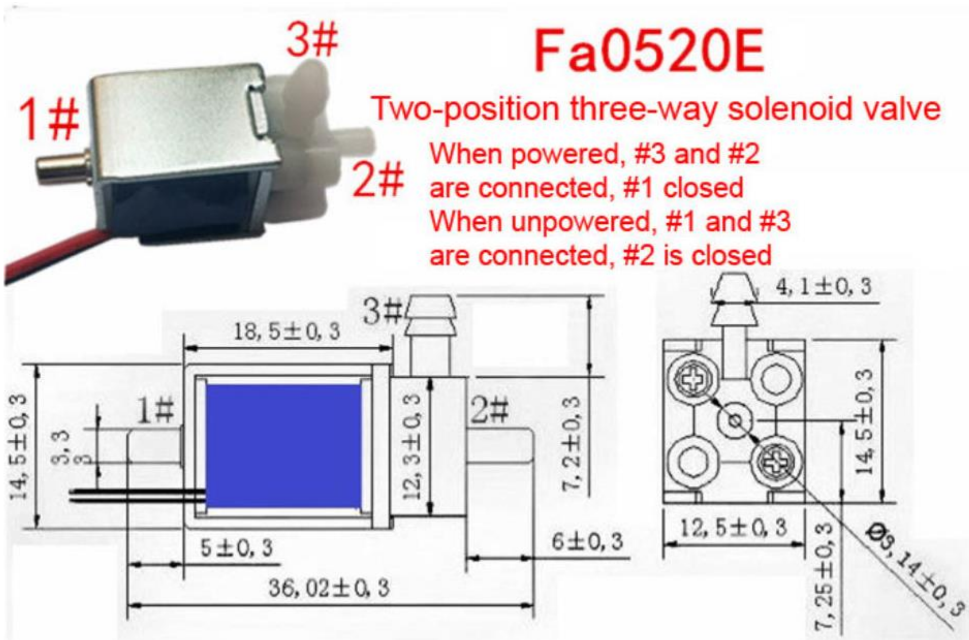


Figure 1: Diagram of the 6V solenoid air valve

With silicone tube connecting the whole system together the next solenoid coming off the compressed air controls the air going into the air bladder. This essentially acts like an on/off switch, when the solenoid is powered, the air is sent through to the air bladder. When powered the bag would fill up with air and drive the system vertically, and when unpowered the system would vent the air, decreasing volume in the bladder and sinking back down in the water. Below, is the system, including a regulator that steps down the pressure of the CO2 cartridge after going through a PVC pipe that also slightly steps down pressure introduced by the CO2 cartridge.

Another use of the system would also be for retrieval during autonomous testing. In the future, another team could implement the system and make a code to Figure out how much CO_2 would be needed to fill the buoyancy bag at the flip of a switch. This would ensure the safety of GUPPS and make sure it could always come up to the surface.

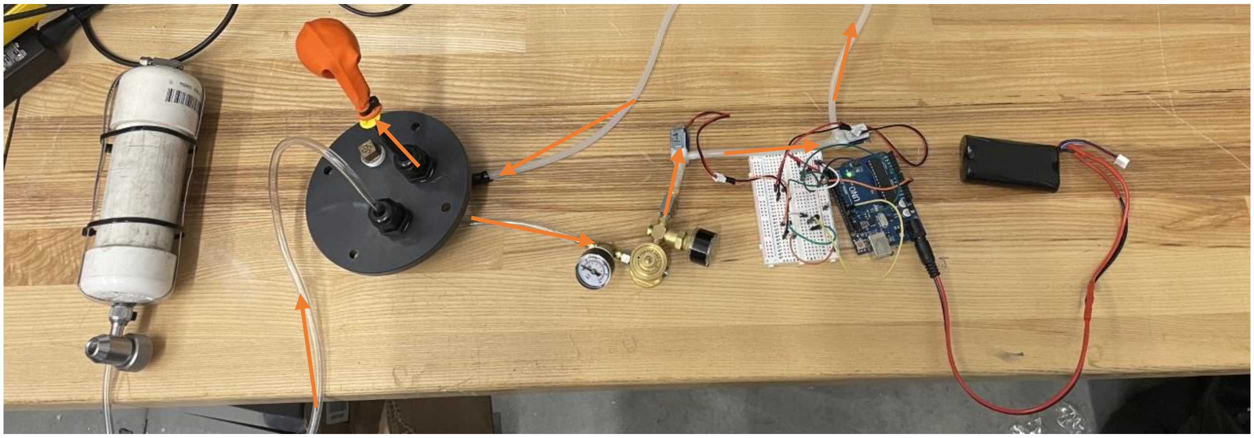
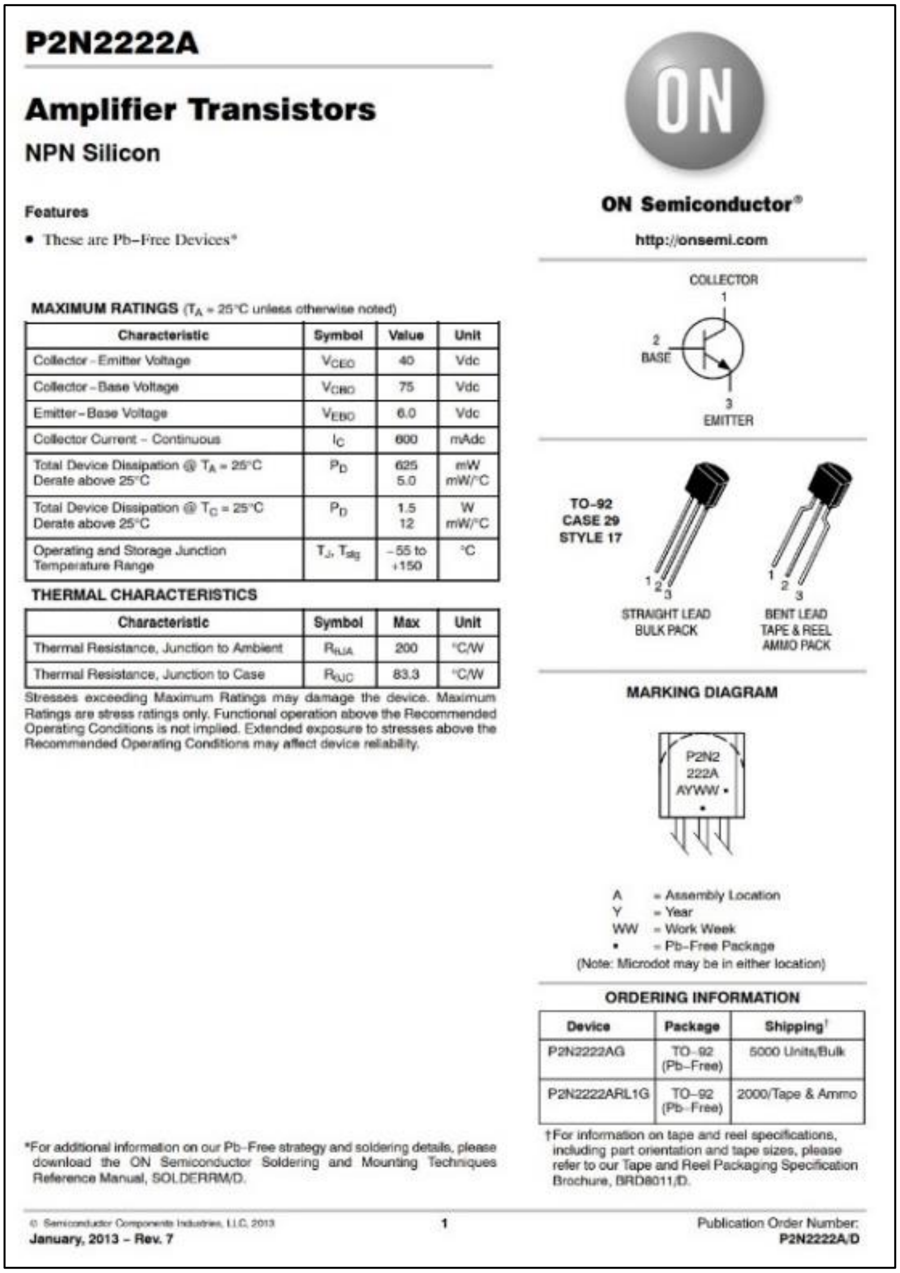


Figure 2: Layout of the buoyancy system. The orange arrows show the flow of air through the system. The CO_2 cartridge would be placed in the valve right under the PVC pipe.



P2N2222A

Amplifier Transistors

NPN Silicon

Features

- These are Pb−Free Devices*

MAXIMUM RATINGS ($T_A = 25°C$ unless otherwise noted)

Characteristic	Symbol	Value	Unit
Collector − Emitter Voltage	V_{CEO}	40	Vdc
Collector − Base Voltage	V_{CBO}	75	Vdc
Emitter − Base Voltage	V_{EBO}	6.0	Vdc
Collector Current − Continuous	I_C	600	mAdc
Total Device Dissipation @ $T_A = 25°C$ Derate above 25°C	P_D	625 5.0	mW mW/°C
Total Device Dissipation @ $T_C = 25°C$ Derate above 25°C	P_D	1.5 12	W mW/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	200	°C/W
Thermal Resistance, Junction to Case	$R_{\theta JC}$	83.3	°C/W

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

MARKING DIAGRAM

A = Assembly Location
Y = Year
WW = Work Week
• = Pb−Free Package
(Note: Microdot may be in either location)

ORDERING INFORMATION

Device	Package	Shipping†
P2N2222AG	TO−92 (Pb−Free)	5000 Units/Bulk
P2N2222ARL1G	TO−92 (Pb−Free)	2000/Tape & Ammo

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specification Brochure, BRD8011/D.

*For additional information on our Pb−Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

© Semiconductor Components Industries, LLC, 2013
January, 2013 − Rev. 7
1
Publication Order Number:
P2N2222A/D

Figure 3: Datasheet for P2N2222A transistor

The circuit, as seen in Figure 2, consisted of two P2N2222A transistors that used the 5V power output from the Arduino, the ground, and the signal from a digital write pin that transported our code to the solenoids.

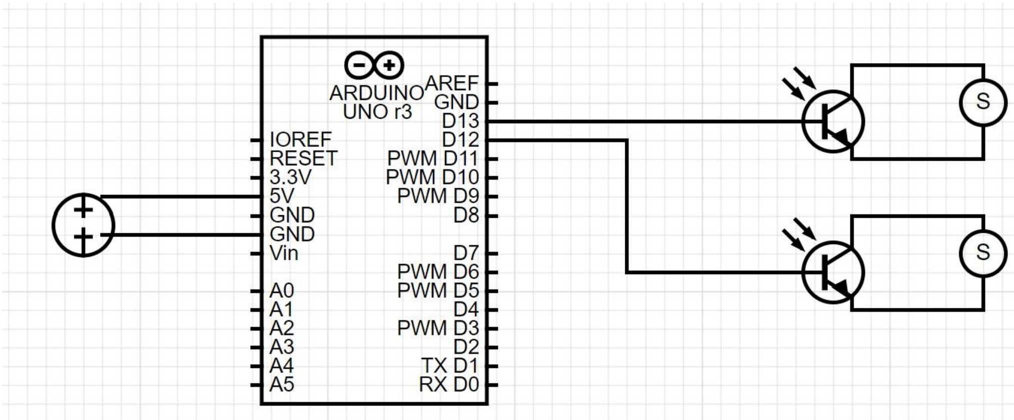
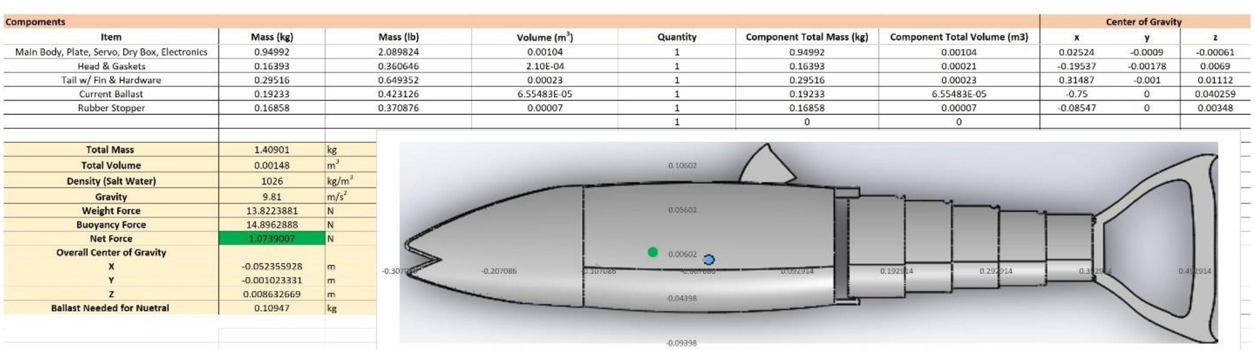


Figure 4: Circuit diagram of the electrical system. The source represents a 7.4V battery and the ‘S’ represents the solenoids extending off the transistors.

A variable buoyancy chart was also made to ensure that the fish would be neutrally buoyant when placed in the water. SolidWorks models used the assembly to find center of mass and center of gravity to relate relative position in the water. Figure 4 shows the chart on excel where the green dot represents center of mass/gravity, and the blue dot is the reference point for where each component was measured. This set up the work done by the other groups which is shown in the next sections.



Compoments							Center of Gravity		
Item	Mass (kg)	Mass (lb)	Volume (m^3)	Quantity	Component Total Mass (kg)	Component Total Volume (m3)	x	y	z
Main Body, Plate, Servo, Dry Box, Electronics	0.94992	2.089824	0.00104	1	0.94992	0.00104	0.02524	-0.0009	-0.00061
Head & Gaskets	0.16393	0.360646	2.10E-04	1	0.16393	0.00021	-0.19537	-0.00178	0.0069
Tail w/ Fin & Hardware	0.29516	0.649352	0.00023	1	0.29516	0.00023	0.31487	-0.001	0.01112
Current Ballast	0.19233	0.423126	6.55483E-05	1	0.19233	6.55483E-05	-0.75	0	0.040259
Rubber Stopper	0.16858	0.370876	0.00007	1	0.16858	0.00007	-0.08547	0	0.00348
				1	0	0			

Total Mass	1.40901	kg
Total Volume	0.00148	m^3
Density (Salt Water)	1026	kg/m^3
Gravity	9.81	m/s^2
Weight Force	13.8223881	N
Buoyancy Force	14.8962888	N
Net Force	1.0739007	N
Overall Center of Gravity		
X	-0.052355928	m
Y	-0.001023331	m
Z	0.008632669	m
Ballast Needed for Nuetral	0.10947	kg

Figure 5: Variable chart made on excel from information found on SolidWorks to ensure neutral buoyancy of GUPPS

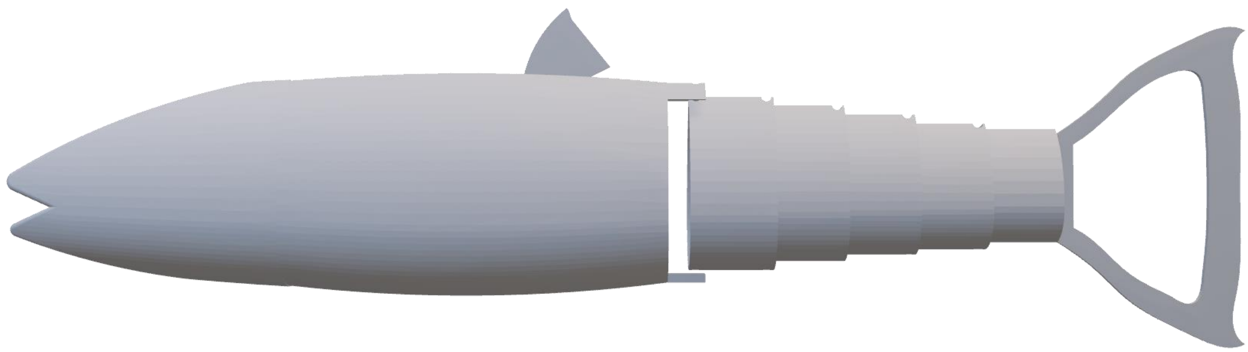


Figure 6: SolidWorks assembly of current GUPPS model

As seen above, the body was redesigned to be more biomimetic. The body is 0.125in thick, the body is 32 inches long, 8.5in tall, and 5in wide. The body has been lengthened and slimmed down from the previous year's model and painted for a full biomimetic effect. It is made from PETG plastic, designed in SolidWorks and printed on a Quidi 3-D printer. The body took about 50 hours (about 2 days) to print and had a 30% infill.

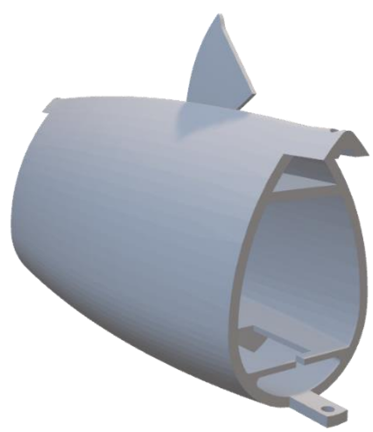


Figure 7: SolidWorks model of current GUPPS body

The shape was specifically created to mimic the shape of a steel head trout, while also providing space for internal components. The front of the body was made slightly larger than the back to allow for the 2.5in dry box. The body itself is 11in long. 8.5in tall, and 5 in wide. The body also has a dorsal fin featured on top, as seen in Figure 7.

The inside of the body houses two shelves, a top shelf about 1 in down from the top of the fish for buoyancy foam, and a bottom shelf, approximately 1 in up from the bottom of the fish, for ballast. Both shelves run through the fish lengthwise. It is important to place the buoyancy foam at the top of the fish so the body will orient itself upright in the water. The fish is designed to sit upright and level in the water.

The bottom shelf was designed as a platform for the safety box to sit on, house ballast, and securely attach the servo. To mount the servo, a cutout was made, it is the exact size of the servo, centered on the back of the shelf, and offset approximately 0.25in into to the fish create a 360-degree ring around the servo housing. A ~0.125in parameter was drawn around the hole and extruded to the bottom of the fish, to fully encase the bottom half of the servo. This was done to provide support which lessens the side-to-side canting when pushing a load, i.e. the tail. Underneath the front 2/3 of the bottom shelf is ballast. 0.19233kg of steel pellets housed in a plastic bag were used to make the fish neutrally buoyant and sit level in the water. Also, attached to the bottom shelf of the body is a bulkhead. The bulkhead consists of a wall that runs from the bottom shelf to the bottom of the fish. Its purpose is to stop water from going through the bottom of the shelf, it is also meant to separate the ballast and the servo.

Two extrusions for mounting the tail are seen on the back of the fish. They have a hole for a ¼-20 screw. They extrude approximately 1in and are the same thickness as the body.

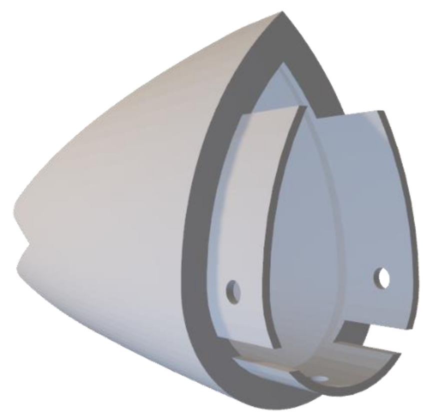


Figure 8: SolidWorks model of current GUPPS head

As shown above, the head of the fish was also designed in SolidWorks and 3-D printed out of PETG plastic. The head is 6.83in long, 5.47in high, and 3.48in wide. It features an indent mimicking the mouth and is hollow inside. The back of the head has three extrusions 0.125in thick and extrude about 1in, they mirror the shape of the body. They are offset in about 0.25in and rubber gaskets were added to the outer face. The purpose of these tabs is to attach the head to the body. The head was designed to be detachable to allow access to the safety box. Originally, the head was allowed to have back and forth movement, but that was found to counteract the movement of the tail and propel the fish backwards. Thus, the head was made to fit snugly into the body, so it acts as an extension of the body and allows the fish to propel itself forward.

When starting work this year, GUPPS had an exoskeleton tail that could achieve the motion of a fish tail. In doing preliminary tests with that design, there were a few key takeaways. The first was that the tail was not displacing water, which was why it could not propel through the water. The second was that the tension of the line connecting each piece played a major role in achieving the desired motion in conjunction with the displacement caused by the servo motor. Another notable aspect of the previous design was the time and hassle involved in the assembly of the segmented tail at the attachment points. The final take-away was that the servo was not waterproof and would need to be replaced before GUPPS was fully submerged. These initial findings became the basis of the tail redesign.

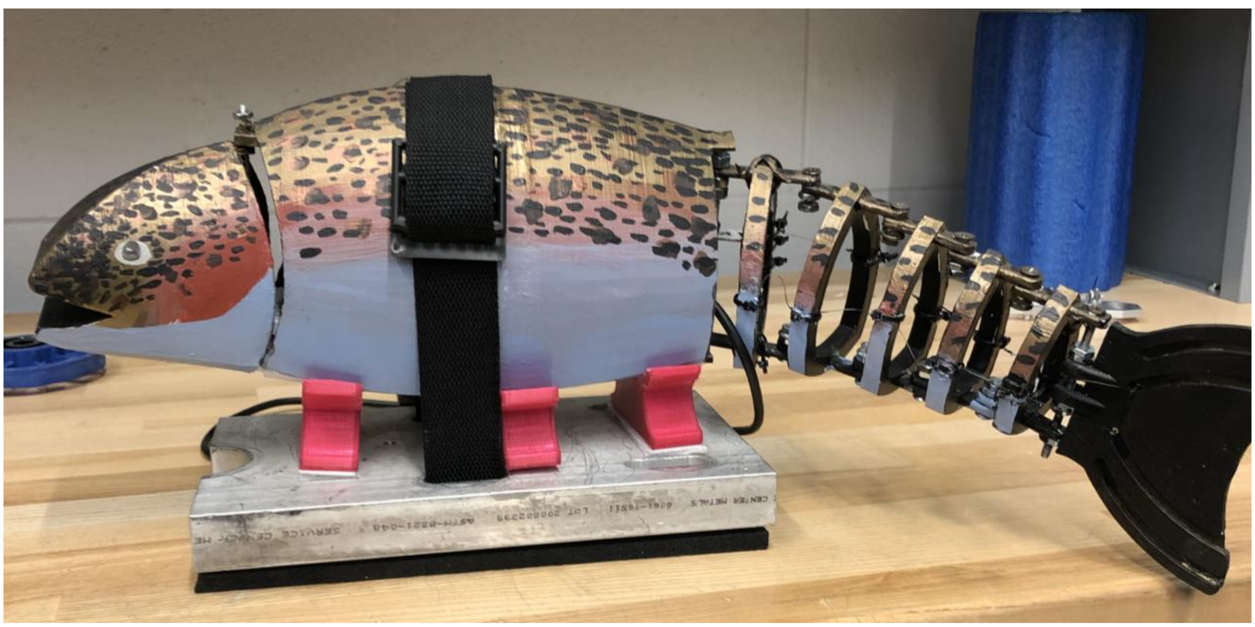


Figure 9: Photo of last year’s GUPPS

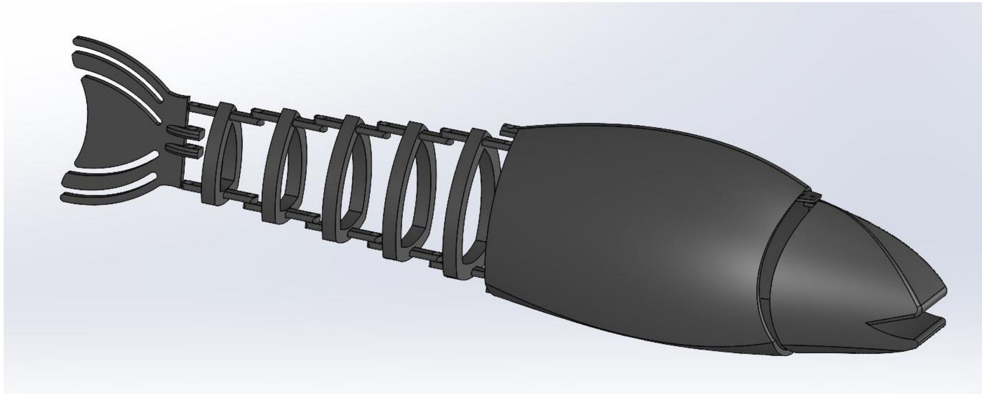


Figure 10: SolidWorks assembly of last year's GUPPS

The first take-away addressed was the lack of surface area of the tail pieces. It was determined that the tail pieces needed to be wider to create a larger interface with the water. It was clear that the previous team opted for a segmented tail design to achieve the curvature present in fish tails, so continuing with their design of 5 tail segments, the tail pieces were widened. Taking inspiration from other model fish toys, there were two ways to widen the tail pieces and still achieve the curvature, having cutouts to allow for maximum inward curvature in both directions, or nested pieces with gaps in between each segment to allow for maximum outward curvature. In order to make the most of the tail thrust, the most water displacement would need to occur, which prompted the desire for maximized surface area of the tail. The nested tail pieces design allowed for the most surface area, while still allowing the tail to have a natural curve as it was driven.

From the second take-away that the tension forces of the line in the tail were extremely helpful, that concept was expounded upon in the redesign. The tension in the line acted once the tail was fully in one extreme or the other and prevented it from rotating farther to each side. The concept of springs was assessed as springs provide the tension like the line, but additionally provide an active restoring force as the tail returns to the equilibrium position. The added restoring forces help to overcome the water resistance that increases as the surface area was increased. Extrusions were added to the inside of each tail piece to hook each of the springs to between the segments. To combat the issue of breakage in 3D prints, large fillets were added to the extrusions for added strength.

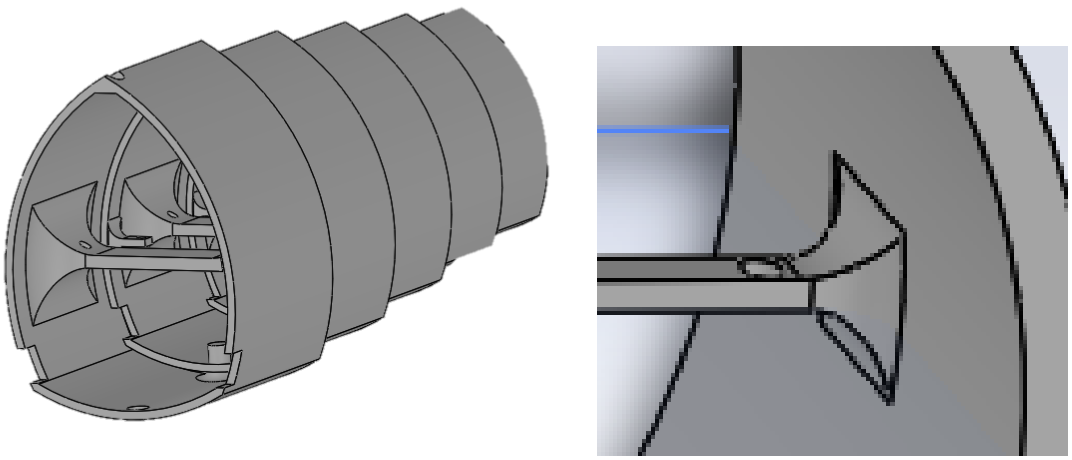


Figure 11: Overview and close-up view of spring mounts in tail assembly

When assembling and testing the tail, the springs were often excluded, which led to the conclusion that the nature of the nested segments provided enough restorative force to bring the tail back to equilibrium despite the increased water resistance. With this realization and the hassle the springs brought to the assembly, they were deemed unnecessary and the mounting extrusions were removed from the design.

With a redesign that included continued use of segmented tail pieces, the attachment points would continue to be a hassle going forward. This prompted the redesign of the connection points. Ideally the connection points would not have to require manual assembly, which sparked the idea of a “print and play” design which would involve all the tail pieces being printed at once and the connection method would be extrusions from the tail pieces with peg-hole connections.

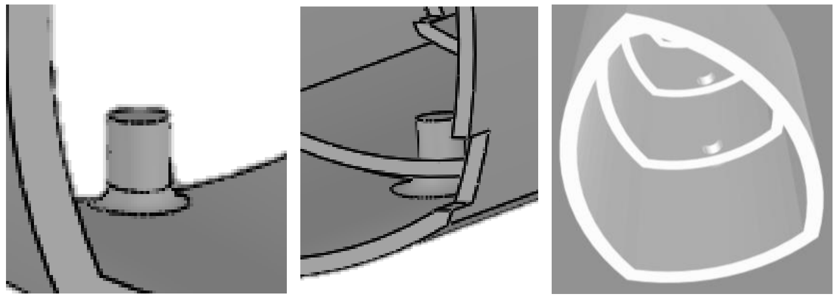


Figure 12: “Print & Play” connection points to attempt to eliminate manual assembly.

As printing and testing commenced, the pegs proved delicate and broke easily. To aid in their strength, larger and larger fillets were added until the fillets filled the space between the tail segments. Once the upper limit of fillet sizes was reached and the strength of the pegs had not significantly improved, the “print and play” design was abandoned and the previous connection method of nuts and bolts was reinstated. In addition to the delicacy of the pegs in the connection method, the “print and play” design also created challenges with accessibility to the spring mounts. This aided in the decision to return to the previous attachment design which allowed for access to each tail piece individually, which proved helpful in later design iteration.

Once all of these iterations were implemented, new challenges arose once GUPPS was tested in the water. Continuing from the previous year’s design, the tail was driven by a servo connected to the first tail section closest to the body. As in the original design, this worked well in air, however, when tested in water, the power of the servo and the reaction forces between each tail piece were not enough to overcome the water resistance, so the first tail piece moved as expected while the rest of the tail remained stationary. To combat the water resistance, the moment arm of the servo was extended so that the tail would be driven at the middle segment. To achieve this, a vertical extrusion was added to the center tail piece and a threaded rod was extended from the servo arm to this segment.

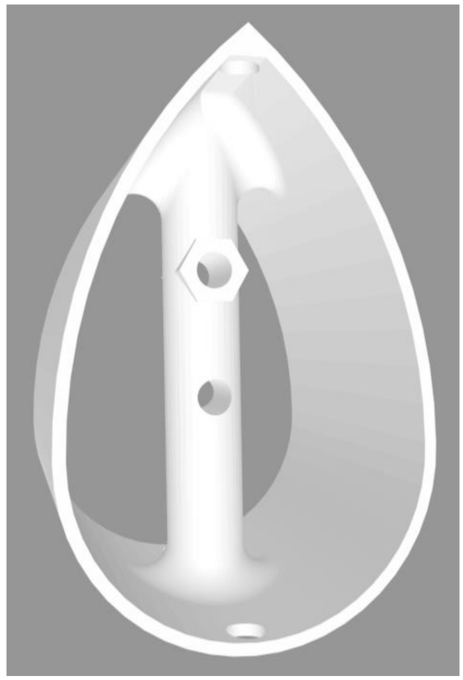


Figure 13: Vertical member with holes to connect with threaded rod to drive tail from center tail section.

To mount the threaded rod to the servo arm, a 3D modeled piece was designed and printed to connect the threaded rod and arm mounting at a 90º angle. Due to the limited strength of PET-G, and the limitations of adding threads to 3D prints, the 3D modeled design was replaced with a custom made acrylic mounting block. The block has a ¼-20 threaded hole centered on one side, and 3 M3 0.5 threaded holes on the adjacent face that align with the threaded holes in the servo arm. This piece allows the rotation of the servo to be transferred laterally to the center tail section to achieve displacement.

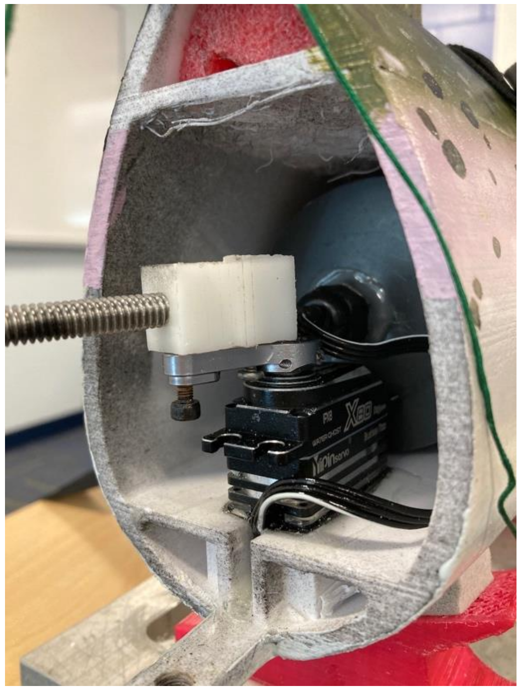


Figure 14: Acrylic mounting block to connect threaded rod to servo arm.

The first step to making the mechanical, moving system of GUPPS was to ensure that every component involved was waterproof. Working with the original dry box and servo, struggles immediately occurred once placed in the water. The first came with the servo. The original servo, when placed in the water, seized up due to the fact that it was not waterproof as it was advertised. A new servo was ordered as a replacement and to upgrade to a 60 kg servo that could produce more torque to move the tail in the water. After multiple rounds of testing this servo also seized due to water shortening out the motor control within the servo. Further research pointed to the need for a servo with an IPx68 rating to be able to withstand continuous submersion. This servo was also another upgrade to an 80 kg servo to improve movement within the water for the increase in size, weight, and drag of the tail while in the water.

The next struggle was working with the dry box to house the electronics. The original dry box leaked along where the wires entered the box. This caused the Arduino board controlling the servo to short out and fry. Multiple attempts were made to make the already existing dry box completely waterproof, adding grease over the wires and adding electrical tape to make a tighter seal on the gasket. Eventually, a redesign was necessary due to the constant leaks and shortening of Arduino boards after a two-minute swim. The solution was to cap the end of a 2-inch PVC pipe with silicon and to drill a hole in the cap to feed the wire through and to silicone the wires in place to produce a watertight seal. The other end of the pipe is sealed with a rubber stopper that, when screwed, expands the rubber to produce a watertight seal against the pipe edges. This produces a quick and easy system to access the electronics of the fish vs the original box where there were four screws that had to be loosened to achieve access to the box.

The next issue came from power limits. Originally the fish was powered by an external power source that one would use to recharge a phone, but this did not produce enough current or volts to power the servo. The solution was to use the power jack on the Arduino UNO boards and power the board with a 9-volt battery. Adding the 9-volt battery supplied the system with enough power and current to function. The only drawback was that the batteries had a very short life span due to the heavy draw from the servo causing each battery to have a single use depending on the length of run time. The next choice was to upgrade the battery to a LiPo battery that would give a longer battery life and off the ability to be recharged. The battery of choice was a 2 cell 7400mAh LiPo battery. The only issue was that the battery connections did not match that of the power jack on the Arduino board. Power jack adapters were ordered and soldered to the battery's leads in replacement of its original connection to allow the battery to power the board.

The final headache of the entire mechanical system was the code that controlled the servo motor. The original code was a bang-bang control with no form of adjustment. After an accidental loss of the code, it was decided to start from scratch and write a new code for the control of the servo motor. The new code was written with a center start point, a degree of sway (how open the servo arm will swing), and a delay for the speed at which the arm moves. This code was then copied so that one code would control the motion of the tail and the other would have the tail stay at a set degree. This code was called the centering code because it allowed the option to adjust the center location degree of the servo without having the entire tail flail about.

To enhance the propulsion, the rear fin design was examined. The previous year's GUPPS had a tail that was connected via two aligned points, allowing for the tail to pivot as it swam. With the ability to pivot, the fin was at the mercy of the water and followed the tail motion instead of acting with the tail motion. To enhance this, the rear fin was extruded directly from the final tail piece to force the fin to move with the rest of the tail. When originally designed, the rear fin was solid with the outline of a steelhead trout fin. When tested, the solid tail fin created too much drag, which prevented the tail to move with the power of the current servo. To combat this, a cutout was added to the center of the fin whose outline mimicked the overall fin outline. To preserve biomimicry, mesh fabric was added to the cutout to create the illusion of a solid tail. In the future, the cutout can be modified to be more biomimetic, or a stronger servo can be implemented to have the strength to overcome the drag created by the solid tail.

When the tail achieved the desired motion and propulsion, focus then shifted to biomimicry. Through consult with marine biologists, it was determined that the cross section of GUPPS needed reshaping. The original cross section had a flatter bottom, while steelhead trout have a very round underbelly.

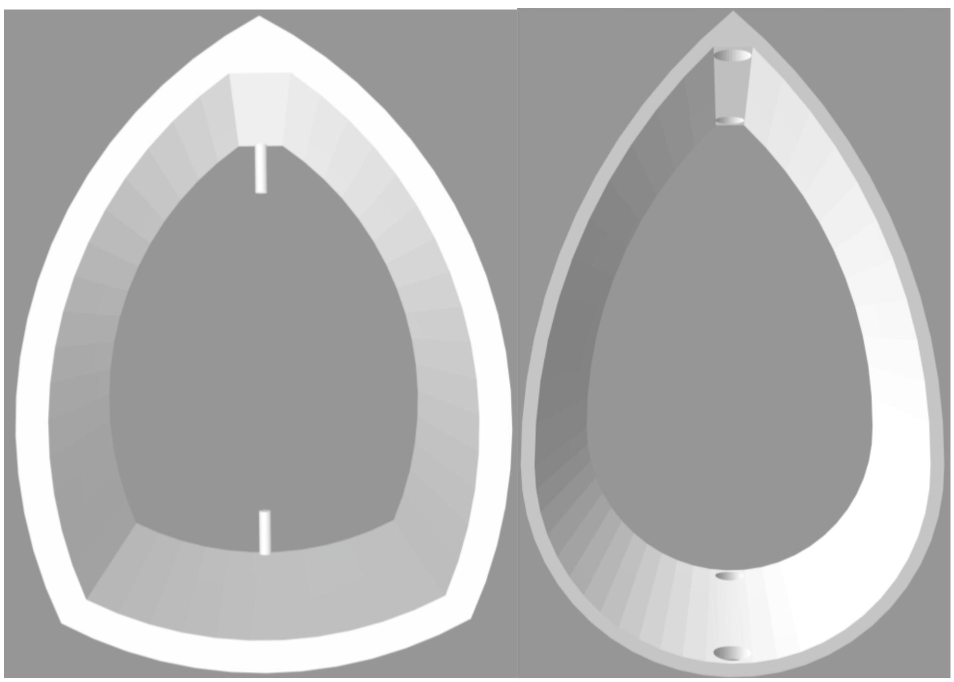


Figure 15: GUPPS cross-section profiles before and after redesign for biomimicry

The new cross section more closely reflects the shape of the trout while also maintaining the surface area necessary for propulsion.

Once this year's overall design was finalized, the final step toward biomimicry was taken by painting GUPPS to look like a steelhead trout.

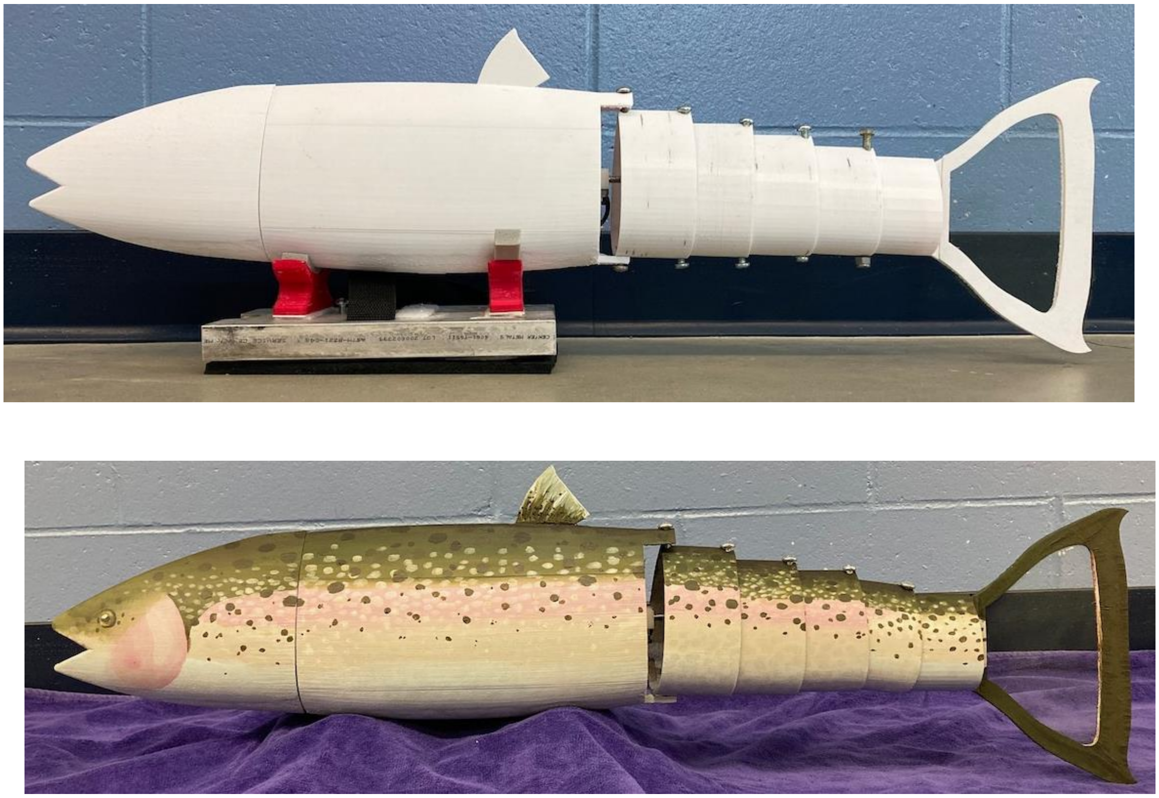


Figure 16: GUPPS before and after painting for biomimicry

Inspiration for the paint job was taken from photos on the internet of trout of the same species, most notably a photo from NOAA Fisheries as seen below.

Figure 17: Reference photo of steelhead trout for the basis of biomimicry

Setting out with the goal to add control to GUPPS via a propellor and a rutter, the first step taken was electronics. The team used a Hawk Hobby underwater propeller and a 60-kilogram digital servo and connected them to an Arduino Uno. A corresponding code for each was found in the Arduino Library, and minor adjustments were made to better fit our goals. These changes included the range of the servo, only allowing it to travel 50 degrees so as to not break the tail and changing the propeller to only rotate in one direction, rather than being able to go forwards and backwards. Code was also changed to allow for the use of potentiometers that were connected that allowed us to control the servo and propeller. The system was powered via a 7.4V LiPo battery, the Arduino was powered via a 9V battery, and an Electronic Speed Controller (ESC) was used to bridge the connection between the controller and the propeller. The wiring diagram can be seen below in Figure 18.

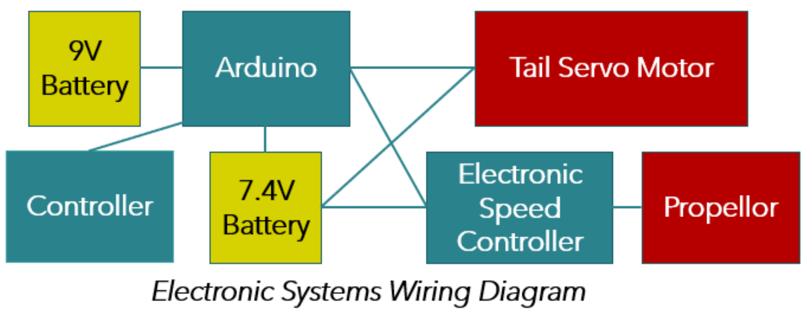


Figure 18: Propellor wiring diagram

The batteries, ESC, and Arduino were placed inside an Otterbox 3500 dry box that was equipped with port holes for the respective wires.

With the electronics working, the next step was to design the body of the fish. Solidworks was used to design a new, bigger body, largely based on the previous year's design, but tweaked to better resemble a steel head trout by elongating the body and head and removing the loft of the body. The tail was also changed to better allow for it to operate as a rutter. In its current design, it is simply a planar tail, but from the side it is meant to resemble the correct shape. This can be seen below in Figure 19.

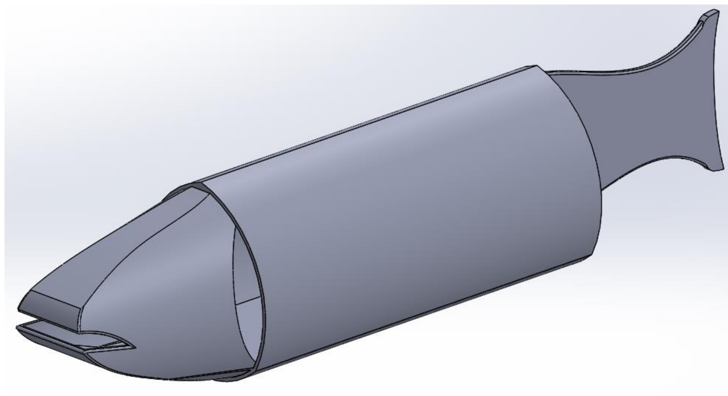


Figure 19: SOLIDWORKS model of propellor GUPPS

The head, tail, and both halves of the body were then 3D printed. The two halves of the body, printed separately due to lack of space on the print bed, were attached using clear J-B Weld. Holes were drilled in the body, head, and tail in preplanned places to attach the servo and propeller, and to connect each piece together.

With the electronics in place, the dry box inserted, and the body assembled, the final step was to balance the buoyancy. Large pieces of steel, roughly four inches squared and a centimeter thick, were needed in the center of the body and in the head to offset the positive buoyancy of the dry box. Then, smaller pieces of steel, such as bolt washers, were used to fine tune the buoyancy until it became neutrally buoyant. The final step was to paint the fish in order to really attain biomimicry, as can be seen below in Figure 20.

Figure 20: Propellor full body picture

Results

The buoyancy team successfully built a working prototype that fluctuated water depths by introducing air into the system, having code that made the solenoids hold the desired amount of air and then release on a timer.

The tail team constructed a biomimetic fish that swam in a straight line using the motion of the tail, that was able to swim uninterrupted for several hours. The mechanical system is more of a proof mechanical/electrical system that if set up right will perform without fail. The upgraded battery has been able to power the fish for over two hours on a full charge.

At the URC, the propeller team presented a biomimetic steelhead trout that was neutrally buoyant, could swim on the surface of the water, and was able to be controlled by changing its speed and direction.

Discussion

Finalizing, the buoyancy system needs work with graduate students to implement a depth/ pressure sensor that is correctly calibrated to work with the system. Then it needs computer science students to create a code to enable reactions from the sensor to the system, opening and closing the air flow. Overall, the system is too large to fit in the body of GUPPS and needs to have an overhaul with components that would allow for seamless integration. Team members have suggested aquarium components since they double as reliable and related to flow mitigation. Teams in the future may also want to go down the path of water systems, which allow water from outside the body into the system to navigate the water column. This may make it easier to control since there is an infinite amount of water to be let in and extracted while there is only so much CO2 that can be used in one run.

The tail team is very satisfied with the final product. It was quite a process getting to that point but, well worth it. The research on all the parts and their level of water resistance and waterproof levels would have saved a lot of time and headache. Using the right components, the first time would have saved a lot of hassle with trying to figure out points of failure and how to modify pre-existing conditions to fit the addition of new parts. There was also a lot of failed prints along the way and it would have been a lot easier if the buying process was not so long. Despite this the head, body, and tail work well in unison and were able to swim for extended periods of time without fail.

With this year's final product of the propellor design of the GUPPS fish, having neutral buoyancy is a very important of a working biomimetic fish. It will allow GUPPS to move through the water steadily instead of pitching and rolling unexpectedly. GUPPS being able to be controlled was an important goal of this project as having this fish being uncontrolled wouldn't be nearly as productive. Having the control through all of the components in our dry box through a rudimentary breadboard controller into the servo and propellor proved difficult a task due to needing a completely dry space for all these electronics to be housed. The team overcame this challenge with wire adapters that could screw onto the dry box and secure themselves with rubber fittings to the wires. The servomotor had a 3 wire which is incompatible with these fittings so hot glue was used on this wire which was not ideal but worked for the time being.

Conclusion/ Next Steps

The goal of this year's GUPPS was to swim straight 1 meter below the surface. Both propulsion designs were able to achieve that goal and steps for further buoyancy control were explored.

The continued development of GUPPS will include additional control elements with the ultimate goal of having a completely autonomous robotic trout. In the immediate future, the focus will be on flushing out the design of the buoyancy control and integrating that into one or both of the propulsion designs to enable control of water column positioning. In conjuncture with the buoyancy system, the lateral motion of the fish can be advanced in both propulsion systems. The propeller system currently has the ability to control the turning, so the next steps would be to enhance the biomimicry of the tail and convert to an autonomously controlled system so the tether can be removed. Throughout this research, the most difficult part was achieving completely waterproof components (i.e.

box, servomotor). Secondly, was achieving neutral buoyancy and having a fish that would swim in a straight line instead of immediately pitching forward due to a moment caused by the propellor. The function of an Arduino with the electronic components is a good idea that seems like it will work well but if any other motors for side fin control is added then an Arduino mega will most likely be necessary. If a controller is wanted to be added to the fish, it will most likely need to be through another controller like a Pixhawk.

The buoyancy team needs to create a smaller version of the design that was made throughout the semester. The Arduino for the servo and the solenoids can get combined to create more space for sensor housings. Additionally, the system could use beefier solenoid that can handle larger pressures from the CO2 canister. During testing the solenoids had connections that would pop off because there would be a large release of pressure or there would be leakage out of the system through the vent because of this pressure buildup. Code and sensor implementation would be the final thing to get added to the system to make it autonomous. This would work by controlling a valve that introduces air or vents out air based off what the sensor would read for depth to keep it at a desired height in the water.

The design aspect of the tail of the fish will be crucial for next year. With this propeller design, the team had all working components but didn't have a completely biomimetic trout. The current open tail design was mostly for proof of concept since it does not look very biomimetic yet. From the current design of how it operates, it was imagined it would be integrated into a design like what the current tail team has. If the oscillation could be as such, the tail would oscillate with a small degree of oscillation around a middle oscillation if going straight go between -5 degrees to 5 degrees. If it should turn, then it would move the tail to 30 degrees but move between 35 degrees and 25 degrees. As if the tail was not oscillating it will be at location 0 moving straight and 30 degrees if turning. It would then have an added constant moving +- 5-degree oscillation around those values so it is able to still have the biomimetics of the fish while having the ability to turn and not only relying on the tail for motion.

The tail propulsion system needs to be developed to turn left and right by adjusting the speed and displacement angle, so the tail will displace further on one side than the other. Looking further, pitch control will be developed with the addition of pectoral fins and overall autonomous control can be achieved through the integration of all of these systems. With the integration of all of these systems, a downsize of the dry-box will be required in both propulsion systems to make room for the additional components, like buoyancy. To accommodate the downsize, a smaller microcontroller will need to be considered as the current Arduino UNO is dictating the current dry-box size. In the end, this year's goal was to swim straight 1 meter below the surface. Both propulsion designs were able to achieve that goal. Overall, the GUPPS team is proud of this year's accomplishments and excited to see how future teams advance it.

Acknowledgements

This work was funded in part by New Hampshire Sea Grant's Workforce Development Project E/WFD-2, pursuant to National Oceanic and Atmospheric Administration Award No. NA22OAR4170124. Additional funding was supplied by NEEC grants #N00174-17-1-0002 & #N00174-20-1-0006.

Special thanks to our CS consultation, Clifton Sullivan, Zachary DiCicco, Harrison Ursitti, Aiden Boucher and Christina Karagianis, our artistic talent, Anne Burg, UNH graduate student ChanLing Beswick, and UNH staff members, John Ahern, Wendy Goldstein, Jane Miller and Sheldon Parent.

Citations

Berlinsky, David (04/2023), Buoyancy and Swimming

Industries, Adafruit. "6V Air Valve with 2-Pin JST PH Connector." Adafruit Industries Blog RSS, Adafruit Industries, 26 Aug. 2020, https://www.adafruit.com/product/4663.

NOAA Fisheries. "Steelhead Trout" NOAA, 9 Feb. 2023, https://www.fisheries.noaa.gov/species/steelhead-trout

Appendix

Buoyancy Unit Operation Instructions

Things you will need:

1. Air bladder
2. 2 Solenoids
3. 2 cell 7200 mAh battery (7.4V)
4. Connecting tubes
5. Air valve
6. Compressed air with regulator (CO2 Cylinder)

Start up:

1. Ensure the system is setup correctly

2. The compressed air fills the bag very quickly if it is released too quickly, *CAREFULLY* fill the transfer devise with the compressed air (CO2)
3. The one-way valve will limit the amount of air escaping when the code is not running
4. Start the code by plugging in the battery to the Arduino

Code for the buoyancy unit:

1. Changing the HIGH delays on this code for testing purposes allows different amounts of air in the air bladder changing volume of the bag, changing buoyancy
2. Changing the LOW delays on this code for testing purposes allows different amounts of air in the air bladder out, making the fish sink

The system works like so (starting from left of the image below):

- CO2 gas in a small canister sheathed in protective rubber (they get VERY cold while being used) is fed to the system using a canister connected via the road bike CO2 valve to the Presta valve on the PVC gas accumulator
    - Alternatively, gas is pumped into the system using the hand pump connected to the Presta valve on the PVC gas accumulator
- When the road bike CO2 valve is rotated towards the "Inflate" position, gas flows through the accumulator into the system where it passes through an inlet pressure gauge on its way into the Beswick Engineering pressure regulator
- The Beswick Engineering pressure regulator drops the pressure of the gas to be more manageable for the small solenoid valves, (the pressure drop of the regulator is changed by tightening/loosening the small screw stem on the top)

- The gas passes through an outlet pressure gauge and a mini ball valve (turned parallel for on and perpendicular for off) on its way to solenoid 1
- Solenoid 1 is blocked at the silver vent outlet, so a command of LOW to the Arduino Uno pin it's connected to will direct air towards the blocked vent and effectively block the airflow), a command of HIGH will direct the air towards the plastic barbed outlet and on to solenoid 2
- Solenoid 2 either directs air towards the balloon with a command of HIGH or vents gas with a command of LOW, which also deflates the balloon

Currently, the solenoid valves can only handle about 6 psi, whereas the CO2 canisters we use come in at about 850 psi of interior pressure. Thus, we have not fully managed to prevent all leakage of gas in our system when the CO2 is flowing. It is best to have solenoid 2 directed towards the silver vent with a command of LOW while gas is flowing, so any leakage will be directed away from the balloon and will not cause any unwanted flotation. For the future, we recommend playing with new solenoids if you would like to better prevent gas leakage. It should also be noted that the PVC gas accumulator, created to view the effects of releasing less pressure into the system, may not be needed if the solenoids are still too weak. What's more, it should be noted that the pressure regulator may perhaps only work when gas is flowing through it and will not aid in fully stopping gas as we intended. Overall, it is possible that preventing all leakage is virtually impossible with these components, and there will always be some small loss of unused gas.

Tail System Assembly and Operation Instructions

Tail System Assembly:

Things you'll need:

1. Patience
2. A very small flat head screwdriver
3. A 7/16" wrench
4. A Philips head screwdriver
5. A towel

Notes:

- GUPPS (Denise) is made up of 3 sections: head, body, and tail
- *DO NOT REMOVE DRY BOX FROM BODY*, wires are permanently fixed from Arduino board to servo, servo is permanently mounted to body, any excessive pulling on the dry box can cause damage to internals

Head Removal/Attachment:

- The head of GUPPS is held onto the body by three tabs with a tight fit, carefully pull on the head to remove from body (aka decapitate GUPPS). Push the head back onto the body with the three tabs inside to re-attach.

Tail Removal/Attachment:

- The tail is attached to the body by two bolts that mount to two tabs on the rear of the body.
- Unbolt the two bolts on top and bottom, then slide the tail off the threaded rod carefully.

- Reverse to put the tail back on, remember to slide servo arm back through the upper hole in the center tail piece support.

Tail Break Down/Set up:

- The tail is made up of 5 sections. The first 3 sections have a set of bolts, top and bottom, affixing the sections together. The final section is attached with a singular long bolt through top and bottom. Unbolt to dissemble the tail.
- When reassembling the tail, it's best to start from the center tail piece and build out. This makes for getting the wrench on the inside easier.
- Alternatively, the tail can be built starting by attaching the first tail piece to the body and then working outward.

Servo Arm set up:

- The servo arm connects to the servo and an attachment piece to connect to the threaded rod.
- Connect the attachment piece to the servo arm and threaded rod by screwing into the pre-tapped holes in the two sides of the mount.
- Place the modified servo arm onto the servo when the zeroing code has been uploaded to the Arduino, centering the arm.
- Secure the servo arm to the servo using a small flat head screwdriver while the circuit is unplugged to rotate the servo arm to access the set screw. (it's is a royal pain to do this and is ill advised, so it's best to not remove the servo arm when possible)

Tail System Operation:

Things you'll need:

1. GUPPS (Denise)
2. Laptop (with Arduino software)
    a. Centering Code
    b. Motion Code
3. Arduino USB Cable
4. 2 cell 7200 mAh battery
5. String or twine
6. Towel (not needed but highly recommended if water testing)
7. The listed supplies from dismantle/assembly (helps to have for adjustment)

Start Up:

1. Remove the head from GUPPS (Denise)
2. *Gently* pull dry box forward to access the cap
3. Unscrew the wing nut on the rubber cap to remove the cap from the dry box (turn the screw counterclockwise) and set cap aside
4. *Gently* pull the Arduino board out of dry box enough to access the ports on Arduino.
5. Fire up the laptop and have the two codes open (Centering Code and Motion Code)
6. Plug the Arduino USB cable into both laptop and Arduino and load the Centering code onto the Arduino board.

7. Ensure that the tail is relatively centered and does not excessively point to one side or the other (its hard to say dead center because some of the tail sections can still be moved slightly even if the tail is stationary)
8. Plug the LiPo battery into the Arduino
9. Tie string or rope to the top Bolt that attaches the tail (this allows for recovering GUPPS from the tank)

**Note Steps 10 -13 must be done relatively quick to put together GUPPS before the delay for motion is up (makes life easier than trying to put together GUPPS while it's flopping like a fish)*

10. Load the Motion Code onto the Arduino and unplug USB cable from Arduino board
11. Put the Arduino back into the dry box along with the battery
12. Place the cap back on the dry box and tighten (turn clock-wise) the wing nut to produce air tight seal
13. Push dry box as far back into body as possible (ensure no foam pieces are obstructing the dry box)
14. Reattach Head
15. Take GUPPS for a Walk

Break Down:

1. Follow Steps 1-4 of "Start Up"
2. Unplug and remove the battery
3. Dry off GUPPS (if water testing)
4. Place Arduino back in dry box

5. Place cap back on dry box
6. Place head back on body

TUNNING (working with the code):

-Centering (ensure that the tail will always return to center when center code is run):

1. Remove the tail of GUPPS completely, unbolting the two bolts that hold the tail onto the body and slide the tail off the servo arm
2. Follow steps 1-6 of “Start Up”
3. Ensure that the servo arm is perpendicular to the cross section of the body (based off visual approximation, if uncentered, repeat steps 1 and 2 in reverse, putting the tail back on and bolting to the body again, otherwise continue
4. Adjust the “int posi” value on the centering code until the arm reaches the desired center location (again by visual approximation). This value is the number of degrees from the servos built in zero mark (this number must always be a value from (0-360). It is recommended to increase or decrease in increments of 1.
5. After adjustment, save the code to save the new initial, “zero” position, and copy that value over to the “int zero” on the Motion code for later use.
6. Rebuild GUPPS following steps 1-2, putting the tail back on and tightening the bolts that attach the tail to the body and ensuring that the servo arm is slid through the top hole in the support of the middle tail section

Motion (working with the motion code):

1. “int zero” is the initial, or starting point of the servo when it begins its loop

2. “int pos” is taking the initial zero value and turning it into a position value
3. “int time” is the time it takes in mili-seconds for the tail to move from one extreme to the next (though it’s a lot slower when actually happening for real due to drag and resistance)
4. “int degree” is the angle the tail makes between both extremes (it is recommend to make this value an even number)
5. “int up” and “int down” are the upper and lower extremes or bounds of the servo and is made to be a function with the input to these values being the “int degree” (this is where the servo will stop and change direction)
6. In the set up, “delay(3000)” is a 30 second delay before the loop starts to perform, this is mean to allow time to plug in and assemble GUPPS before the tail starts moving

Note: The loop is a bang-bang control that has the servo constantly going from upper bound to lower bound and back again.

<u>Propeller System Operation Instructions</u>

1. Start with body of fish in three separate pieces (head, body, and tail)
2. Ensure propellor is mounted to top of body with pointy end facing the front using M3 screws (the front is the side without 4 holes on the mounting platform). Those holes will be for the servo.
3. Wire the electronics accruing to Figures 18 above and the written instructions below. Due to lack of wire colors, there isn’t much reasoning to Arduino wire

coloring but each component stays the same like the propellor, servo and controller wires. For the controller wiring, pay attention to the four colors of wires in the long cord, (red, brown, yellow, orange) not the Arduino wires coming off those (white, purple, blue, orange).

a. From Arduino, ground goes into mini strip of breadboard.
b. From Mini breadboard ground, one ground wire goes into the blue robotics ESC, and one goes into the servomotor ground in the top of box. Last goes into the controller wire color brown.
c. From Arduino 5V wire goes to mini breadboard. From the breadboard one wire goes to the servomotor red. One wire goes into the controller wire red.
d. From Arduino analog, A0 goes to orange wire of the controller. A1 goes to the yellow wire of the controller.
e. From Arduino digital, 9 goes to the ESC signal input which is white. 10 goes to signal for the servomotor.
f. From the ESC, white is voltage and connects to red of the propellor wire, green is signal and connects to yellow of the propellor. Blue is ground and connects to black of the propellor. Can also follow the wiring from the other side of the ESC which follow each other down the ESC.
g. Plug in the large LIPO battery to the terminal that is connected to the ESC. Plug in a 9V battery to the Arduino.

In summary:

SERVO: Analog A1 to controller yellow. Digital 10 to servo signal white.

LARGE SERVO: If you want to use a larger servo motor that needs higher voltage on which isn't necessary but could be done. Can wire the voltage and ground to the 7.4v LIPO through the terminal, but signal still comes from Arduino digital 10.

PROPELLOR and ESC: The ESC is a Blue robotics electronic speed controller which is needed to use the propellor. 7.4v comes from the LIPO battery into the terminal to the ESC to the Propellor. Analog A0 goes to the controller wire orange. Digital 9 goes to the white wire of the ESC.

Controller: There are four wires in the controller wire. Red: Voltage, Brown: Ground, Orange: Signal 1 Propellor, Yellow: Signal 2 Servo motor. Each potentiometer has 3 prongs, the sides don't matter if it's voltage or ground but they need both. The middle is always signal. Which potentiometer to which signal doesn't matter either it's more of preference but we were usually doing orange (propellor) to the black potentiometer and yellow (servo) to the small blue potentiometer.

4. Close the dry box with the electronics inside (be sure the servo, propellor and controller are coming out of their tether ports and tight)
5. Put the closed dry box through the front of the fish and pull the servo and tether wire out of the back. Place front of box inline with the line drawn in body so buoyancy is correct.
6. Mount the servo using the four long black bolts and corresponding nuts (ensure that the arm of the servo is pointed to the right as you are looking at it from the back)
7. Attach the tail to the body using two of the long silver bolts (again, ensure that the arm is pointing to the right to be able to connect to the servo arm)

8. Use an M3 screw to connect the servo arm to the swivel mounting arm that goes between the servo arm and tail arm.
    a. If the tail does no line up straight out of the body once mounted, the screws on the arm of the servo can be loosened and the arm can be removed and rotated into correct position to allow for the proper range of motion which is set in the code of the Arduino.
9. Use three more of the long silver screws to mount the head to the front of the body.
    a. The fish is now ready to be put in the water (With the fish in the water, it may seem at first like it is not neutrally buoyant, but we have accounted for the water that enters through the 3D printed body, so it takes about a minute to balance out. This problem can be fixed by having the head and tail printed on a higher quality printer such as the J-Lab printer). Keep the body in the water until the bubbles aren't as large.
10. Once the fish is floating neutrally in the water, it can then be operated.
    a. On the breadboard, there are two potentiometers. One controls the speed of the propeller, and the other controls the direction of the tail.
    b. Use these dials to move the fish around in the water.