The Agile^3 ROV Final Design Report

Submitted April 18, 2017

Team Agile^3

1931 Duffield Street

Ann Arbor, MI 48109

John Cooney
Christopher Geng
Nancy Hernandez
Prakash Kumar
Celine Schlueter
Product Engineers

Submitted to Gulf of Mexico Special Projects: Remotely Operated Vehicle Design
Document: 1000-35-PM-SW-000600-F

Table of Contents

Executive Summary

Agile^3 developed an ROV for BP, in response to the Gulf of Mexico Special Project: Remotely Operated Vehicle Design, document number 1000-35-PM-SW-000600-F. This request calls for the construction of a prototype ROV to work in the oil and gas industry that is easy to deploy and transport. This process was divided into two distinct phases: Phase 1, which focuses on the design and construction of the ROV, and Phase 2, which focuses on the testing of the ROV's performance. Agile^3 responded to this request by constructing a versatile and robust ROV that can easily be used by BP. The purpose of this report is to communicate the design, construction, and scaling of the ROV.

Per the design requirements stated by BP, the mass of the prototype ROV must not exceed 8.0kg, and it must not exceed dimensions of 58 x 40 x 30 cm. For control, the ROV must be controllable forward, backward, upward, downward, left and right, while being controlled by a control box from land. Additionally, the ROV must be able to support the payload and tether supplied by BP. Finally, a mechanical arm must be mounted to the ROV for underwater manipulations.

For the design, the team decided on a rectangular 25.0 x 35.0 x 45.0 cm, floodable and cemented frame. This frame allows for a variety of open space which enables BP to add new devices and components to allow for increased usability. Also, open space on the frame provides the room for two adjustable horizontal thruster, in case of center of buoyancy and gravity changes throughout different uses and configurations. Two vertical thrusters provide vertical control, while a custom control box allows for simple and intuitive control of the thrust, turn, pitch and heave of the ROV. Permanent buoys and tape-wrapped foam, together with adjustable nets to hold tape-wrapped foam have been added for buoyancy. The camera and arm are mounted to the front of the ROV in a visible, but protected location. Based on preliminary thruster testing, the estimated top speed for the prototype is 0.4711 m/s. The ROV is designed for stability, with the center of buoyancy closely above the center of gravity. After thruster testing, the hydrodynamic efficiency of the prototype was determined to be 8.7%.

During testing at the GFL and MHL, Agile^3 finalized the design and conducted speed testing, that showed a maximum top speed of 0.4260 m/s. At the newly changed competition, the team competed to deliver as many dive rings as possible to a drop zone, and finished after delivering 2 rings successfully.

For the full scale ROV, the team scaled all known prototype values by a factor of $\lambda = 2.5$, as specified by BP. Thus, the full size ROV will be 112.5 x 87.5 x 62.5 cm, with a top speed of 0.2244 m/s. In order to make the ROV long-lasting and environmentally friendly, the team decided to build it out of anodized aluminum which will provide a strong, resilient frame for low maintenance. The overall estimated cost of construction is $1791.73.

Introduction

The offshore oil and gas industries face unpredictable challenges and often require quick thinking and changes to plans in order to be successful. However, typical ROV systems used by companies in this industry require a large amount of resources and time to deploy and manage. Because of this, BP has requested a prototype of a small, fast-deploy ROV that can be transported by helicopter or boat in order to mitigate many of the issues current ROV systems have. In their request, they specify many requirements the ROV must have. A few of the key ones are described below:

- Must not exceed a total mass of 8.0kg
- Must be storable in a 58cm x 40cm x 30cm high container.
- Must be able to go forward, backward, upward, downward, left and right
- Must support BP's supplied cylindrical payload and tether
- Must use up to four 12V thrusters that are secured to the ROV
- Must use a control box with switches or buttons
- Must have a protected camera
- Must have a mechanical arm for underwater manipulations

Agile^3 has designed and built an ROV prototype in response to BP's request, and fulfilled all of the criteria described above.

Design

In this section the reasoning, execution and evaluation of the ROV design will be explained. This includes the frame, thrusters, buoyancy, detachable arm, camera, and the control box, together with the velocity estimation, stability analysis, and efficiency calculation.

Design Overview

For the design concepts the team decided to focus on versatility, maneuverability, and stability. A versatile design was needed in order to allow BP to adjust the ROV based on their needs. This called for an open frame with space to add new components and tools, in addition to an adjustable propulsion system to account for varying centers of buoyancy and gravity.

For maneuverability, the ROV needs to be controllable in all directions in a simple and easy-to-use manner. Thus, the team decided to mount thrusters both horizontally and vertically, and pair them with a custom, simple-to-use control box.

Finally, in terms of stability, a design that would remain upright under water was wanted. As a result, the team added buoyancy at the top of the ROV, to balance out the mass of the other components and keep the ROV from rolling. The CAD model shown below in Figure 1 was produced to conceptualize this design.

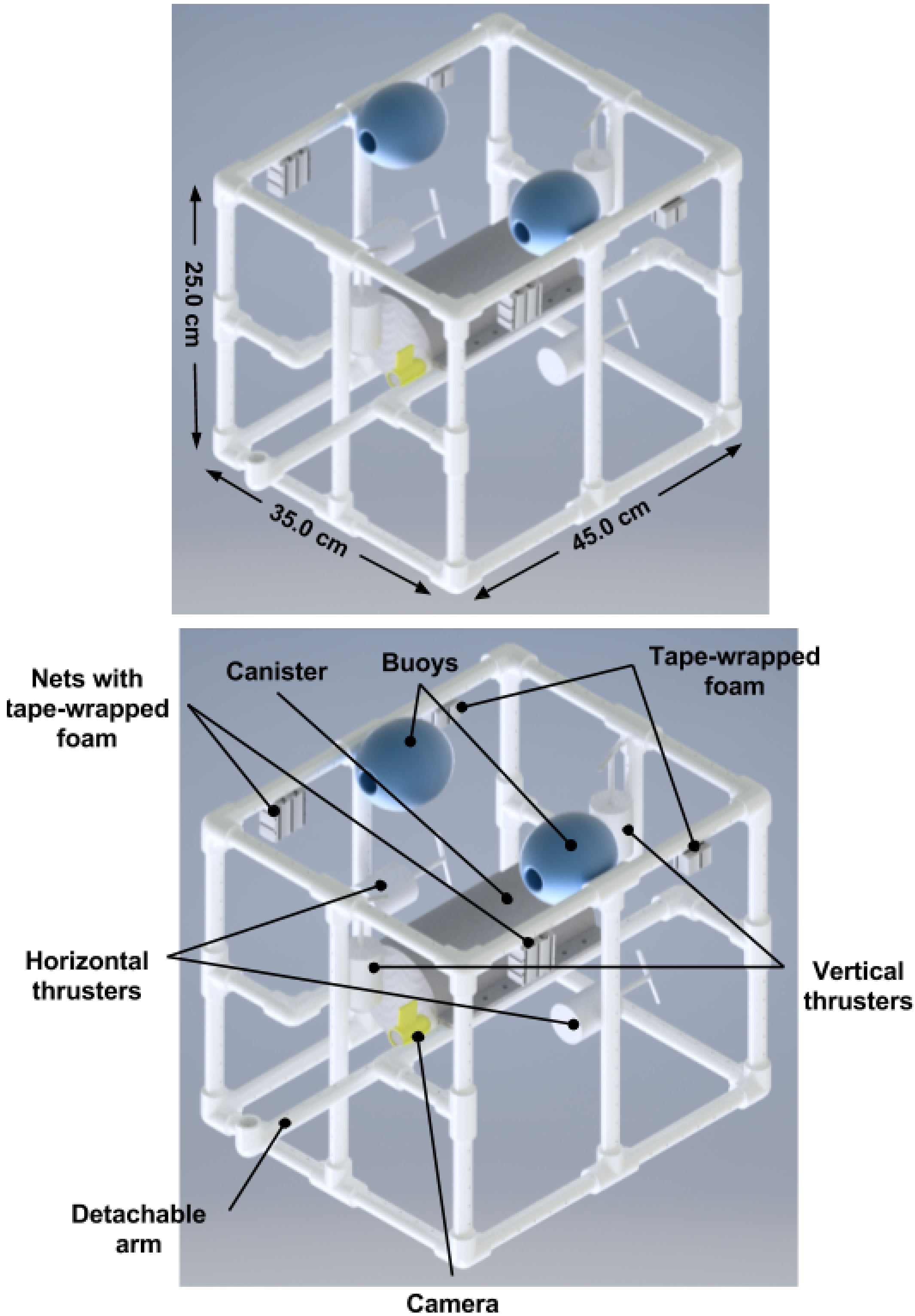


Figure 1. (top) CAD model of ROV design with dimensions. (bottom) CAD model of ROV design with labeled parts

Frame

The ROV prototype frame was built using PVC. The dimensions of the prototype are 35 x 45 x 25cm. The ROV frame was designed to incorporate adjustable thruster placement, allowing for fine tuning both during the competition, and particularly during uses in the Gulf of Mexico. Due to changes in tools and payloads, the centers of gravity and buoyancy will change, and the position of the thrusters will have to be adjusted to allow for proper propulsion. To achieve this, a vertical bar was placed on each side of the ROV to allow thrusters to be mounted at different heights along the bar. The front and back of the ROV contain mounts for the vertical thrusters, these were kept at a set height to maintain stability in heave and pitch. There are also two pieces of PVC that run down the length of the ROV to hold the payload in the center. The mechanical arm consists of one piece of PVC along with one elbow joint. The arm is attached to the camera and can be removed. All pieces were connected using PVC cement. An image of the assembled frame can be seen in Appendix A.

Thrusters

The thrusters were placed in the middle of the vertical and horizontal components of the ROV. The thrusters on the sides were mounted horizontally and are adjustable. As stated earlier in the frame section, there are three holes that allow for three different placements of the horizontal thrusters (on both sides), in order to accommodate for changes in the centers of buoyancy and gravity during varying ROV operations by BP. Currently, they are mounted in the center positions, as our team does not have extra tools and attachments that would raise or lower the center of gravity far from the center. The vertical thrusters were placed on the middle of the front and back sides of the ROV, and are used to control heave and pitch. They were mounted pointing downward, since the ROV was designed to be slightly positively buoyant. Thus, greater thrust is needed to submerge the ROV, constituting the orientation of our vertica thrusters.

Buoyancy

Pool buoys and foam wrapped in duct tape were used as buoyancy for the prototype. The buoys and two large pieces of foam are used for the essential, unchanging amount of buoyancy needed. Some foam, however, was cut into small pieces and held by mesh nets, as the adjustable buoyancy. The two buoys were put in the middle of the frame while the two large pieces of foam were put in the back of the frame, and the adjustable foam in the mesh was put in the front of the frame. This arrangement is shown in Figure 2.

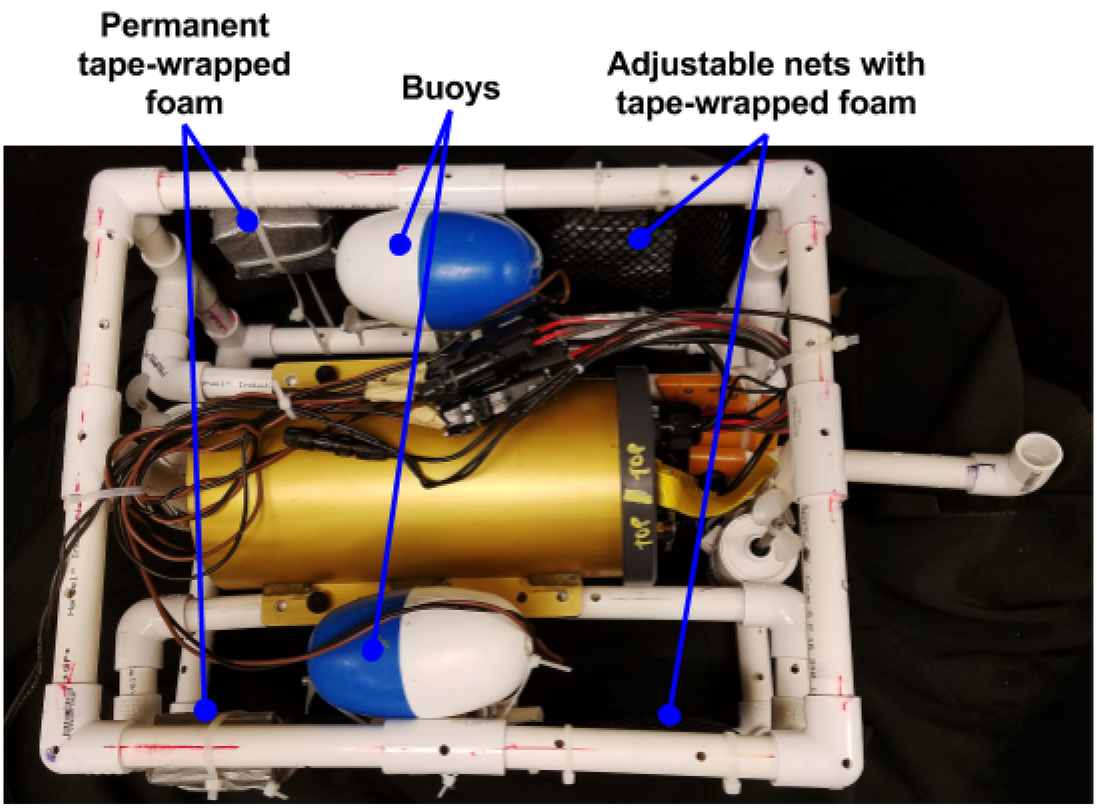


Figure 2. Top view of assembled ROV with buoyancy materials labeled.

Because there is more mass in the front of the ROV, the adjustable nets were placed there as well to balance it out. The reason for the adjustable buoyancy has to do with the variation of mass of the payload. The team was provided with a canister by BP, containing all of the electrical components and the battery; however, each canister was different in mass; therefore, the adjustable buoyancy accounted for the change in mass to keep the ROV positively buoyant. Also, the back, permanent foam pieces are adjustable in position. They can be moved backwards and forwards on the frame by sliding them, which also helps with correction in pitch, creating a more balanced ROV.

Camera and Detachable Arm

As specified by BP, the ROV must include a camera that allows users to see from the ROV underwater, as well as an arm that allows the ROV to interact with its surroundings. The camera was mounted to one side of the ROV using zip-ties. It was carefully placed such that it did not extend out at the end of the ROV, making sure that even if the ROV were to run into an obstacle or debris the camera would be kept safe and functional. Since the ROV is extremely close to the maximum dimensions placed by BP, a detachable arm was used so that the ROV will continue to fit in these specified dimensions. The detachable arm was made out of a ½” PVC rod along with a PVC elbow. It is attached directly underneath the camera with a bolt so that the arm can be seen through the camera. This is effective because the arm is always in sight of the camera, making it easy to control and see where the arm is with respect to its surroundings. An image of the camera and detachable arm system is shown below in Figure 3.

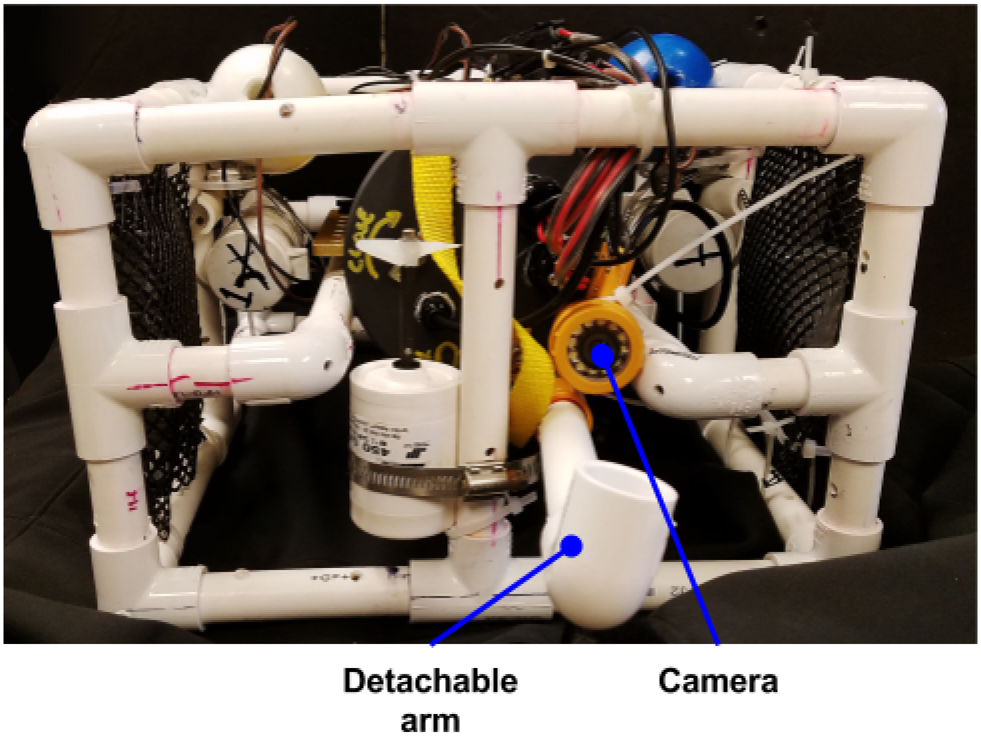


Figure 3. Front view of assembled ROV with detachable arm and camera attached.

Control Box

In order to allow for an intuitive, simple control method for the ROV, the team decided on a control box with 4 toggle switches that would be easy for anyone to operate. This configuration can be seen in Figure 4.

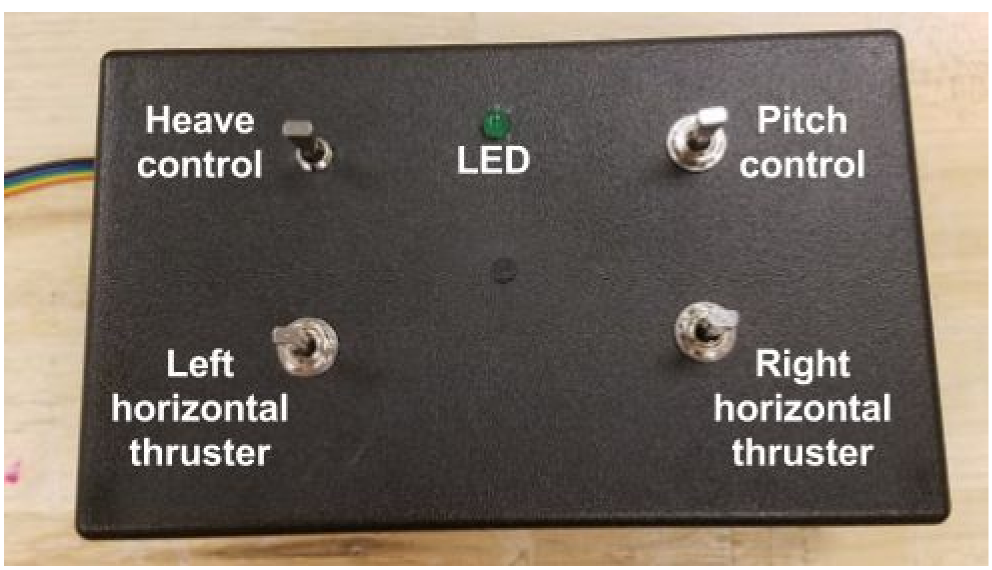


Figure 4. Custom constructed control box with labeled components

Four momentary switches are distributed across the front of the control box. The two bottom switches control the horizontal thruster in a tank drive fashion. Thus, the left toggle switch controls the left thruster going forwards and backwards, while the right toggle switch controls the right thruster going forwards and backwards. This allows for forward, backward, and rotational motion. The top two switches control the heave and the pitch of the ROV. Due to this specialized control, the team wrote a custom Arduino code, which can be referenced in Appendix B. Through this custom code, the top left switch can be flipped, and it will control the two vertical thrusters simultaneously. The thrusters will either both be on in the same direction, or they will both be turned off, creating a steady heave control for the ROV. The right toggle switch also turns on the two vertical thrusters simultaneously, but in opposite directions. As a result, the pitch of the ROV is controlled.

An Arduino board is nested inside the control box, and wired into a breadboard. The switches each have 3 connections soldered to them, in order to provide power and ground the two directions. The wiring of this control box can be seen in Figure 5.

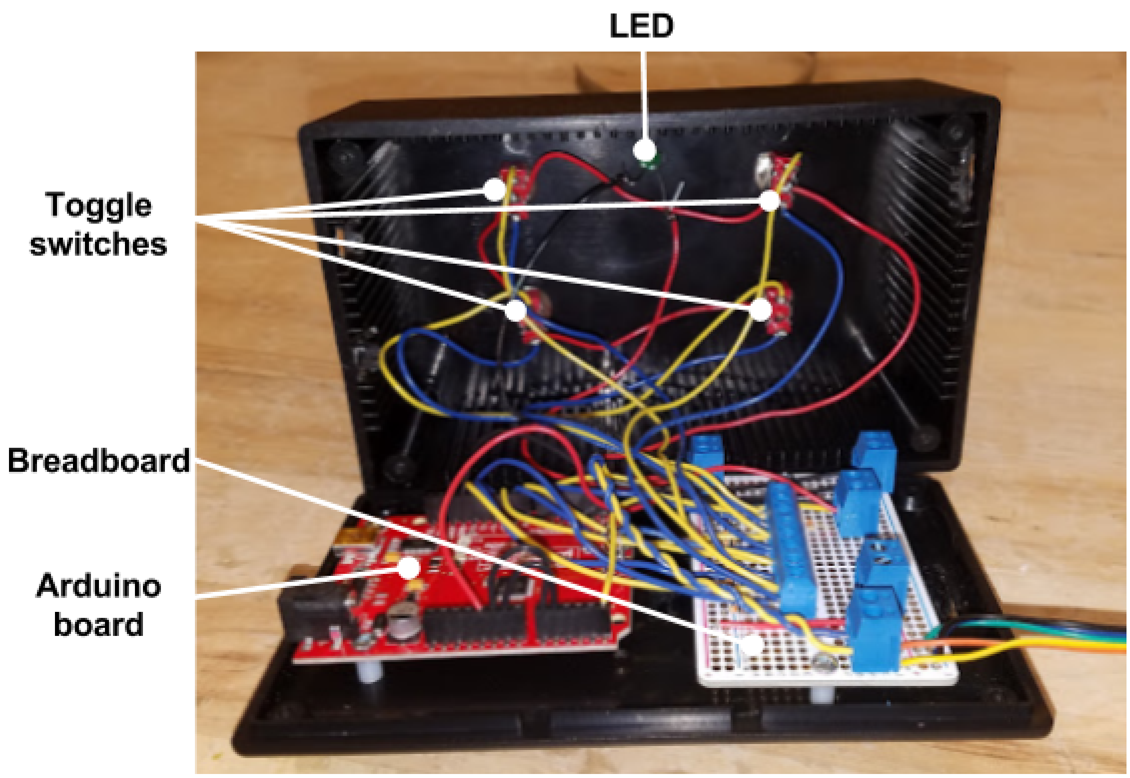


Figure 5. Wiring of the control box with labeled components.

The data is entered into the control box through the switches and is then transmitted over bluetooth to the ROV. A Sparkfun Bluetooth Mate is attached to the control box, and it sends the data to another Bluetooth Mate on the ROV. An LED on the front of the control box, as seen in

Figure 4, shows whether a connection has been established between the Bluetooth devices, indicating that the ROV can be operated.

Mass Budget

Each set of like components, such as the all the PVC T-Connectors, were massed together. The mass of each set was divided by the number of pieces in each set to find the average unit mass, as described in Equations 1 and 2:

$$\sum_{i=1}^{n} x_i = M_T \tag{1}$$

Where x_i is a component, n is the number of components, and M_T is the total mass for each group of components. The same notation is used in Equation 2, with M_U being the unit mass of each component in the group.

$$M_U = \frac{M_T}{n} \tag{2}$$

This method reduced the number of measurements necessary, increasing efficiency. Additionally, it can be assumed that all of these pieces were mass produced and would thus have very similar masses. In this sense, the mass of each component would also be very close to the average mass of all of the components of the same kind. Table 1 is the detailed mass budget for the ROV.

Table 1. Mass Budget Arranged by Descending Mass

Item	Count	Unit Mass (g)	Total Mass (g)
Canister	1	2432.0	2432.0
Thruster	4	238.9	955.7
6(⅞)” pipe	9	57.4	516.5
T-Connector	11	27.7	305.1
3-Way Elbow	8	29.1	232.5
5(¼)” Pipe	6	32.6	195.6
45 cm Pipe	2	93.1	186.3
Pool Buoy	2	47.4	94.8
Elbow Connector	4	21.8	87.1
3” Pipe	4	18.2	72.7
11(½)” Pipe	1	72.4	72.4
2(⅝)” Pipe	4	16.5	66.0
2(¼)” Pipe	4	13.9	55.5
Mesh Bag	2	21.9	43.9
7" pipe	1	43.8	43.8
Foam Piece Large	2	19.0	38.0
Foam Piece Small	8	4.0	32.1
4(⅞)” Pipe	1	18.5	18.5
Total Mass (g)			**5450.1**

As seen from Table 1, the majority of the design’s mass lies with the canister and PVC pipe; the buoyancy system had the lowest mass of any major system, as would be expected from effective buoyant structure design.

Center of Gravity

The center of gravity is the point where the weight of an object acts. For submerged vessels, such as an ROV, it is desirable to have a low center of gravity to maintain stability. The center of gravity was calculated using the formula below:

$$KG = \frac{\sum m * h}{\sum m} \tag{3}$$

Where KG is the height of the center of gravity, *m* is the mass of a component, and *h* is the height of the center of gravity of that component. Using this method, the center of gravity was determined to be located 11.9 cm vertically from the bottom of the ROV and 22.5 cm from the front of the ROV. This can be seen in Figure 6.

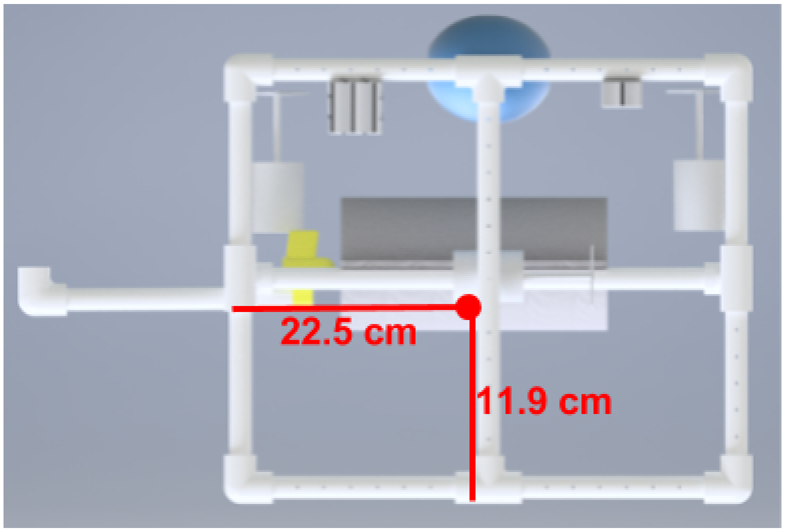


Figure 6. Center of gravity of the ROV, as seen from the side.

Due to the left and right side symmetry of the ROV, the center of gravity is located in the middle of the short side of the ROV. Also, since the center of gravity of each component was estimated, the calculated center of gravity is only an estimation. Tables showing all of the heights and masses used for these calculations in the height and depth directions are in Appendices C and D.

Center of Buoyancy

The center of buoyancy is the point where all of the buoyant forces of an object act. For submerged vessels such as an ROV, it is preferable to have a high center of buoyancy, particularly one that is above the center of gravity, in order to help maintain stability. The center of buoyancy was calculated using the formula below:

$$KB = \frac{\sum \nabla * h}{\sum \nabla} \tag{4}$$

Where KB is the center of buoyancy, ∇ is the volume of each component, and h is the height of the center of buoyancy of each component. The volume of all of the geometrically simple and symmetric pieces, such as the pieces of PVC pipe, were calculated by hand. The volumes of more complex pieces such as the mesh and the buoys were determined by filling a graduated cylinder with water, adding the piece to the water, and determining the difference in volume.

Using this method, the center of buoyancy was determined to be located 12.3 cm from the bottom of the ROV and 22.3 cm from the front of the ROV. This is shown in Figure 7.

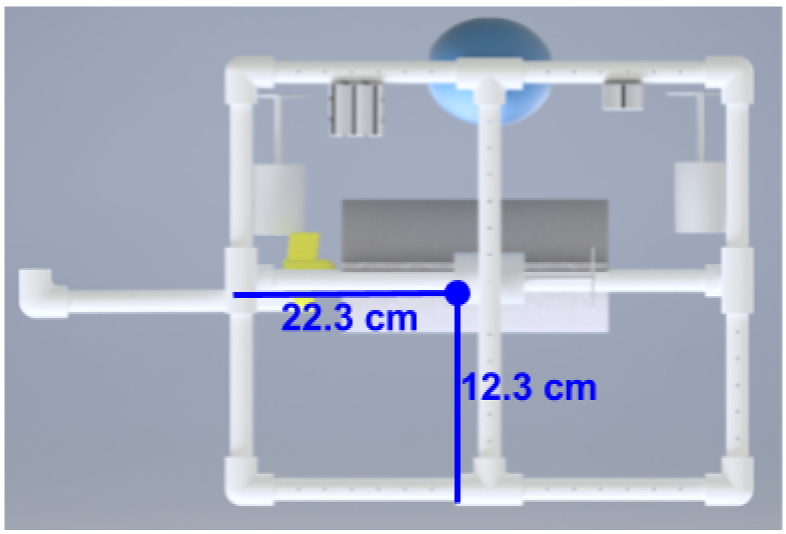


Figure 7. Center of buoyancy of the ROV, as seen from the side.

Due to the left and right side symmetry of the ROV, the center of buoyancy is located in the middle of the short side of the ROV. Also, since the center of buoyancy of each component was estimated, the calculated center of buoyancy of the ROV is only an estimation. A table showing all of the heights and volumes used for these calculation is in Appendices E and F.

These positions of the centers of buoyancy and gravity show the stability of the ROV. Since the center of buoyancy is above the center of gravity the ROV is self-righting, meaning that it will correct itself to remain upright in the water. Also, since these values are still rather close together, at only 0.4 cm apart in the z-direction, and only 0.2 cm apart in the x-direction, the ROV is more maneuverable.

Maximum Velocity Estimation

BP required a maximum velocity estimation prior to testing. The estimation needed an approximate coefficient of drag; since much of the vehicle is built of PVC pipe, the drag coefficient was estimated with that of PVC. Research yielded a drag coefficient of 1.2 for a "hollow semi-cylinder opposite stream" (Drag Coefficient n.d.). The approximation also assumed drag to be equal to thrust at top speed.

$$T = D \tag{5}$$

From here, the equation for drag was set equal to thrust (Equation 6), and solved for velocity (Equation 7):

$$T = D = C_d(\tfrac{1}{2})\rho A_{ROV} V^{2} \tag{6}$$

$$V = \sqrt{\frac{T}{C_d(\frac{1}{2})\rho A_{ROV}}} \tag{7}$$

Where *V* is velocity, *T* is thrust, C_d is the coefficient of drag, *p* is water density, and A_{ROV} is the cross-sectional area of the ROV. To find thrust, a load cell was attached to a lever arm using copper wire. The lever arm pivoted about an axis above a tank of water. The thrusters were attached to the end of the lever arm, in the tank; the force of the torque on the load cell and the known distance of the thruster and load cell wires allowed us to calculate the thrust for each thruster in both the reverse and forward direction. These values are shown in Appendix G.

With these assumptions, the top speed of the ROV moving forward was estimated to be 0.4711 m/s, using the program in Appendix H. Likewise the maximum velocity estimates for vertical descents, vertical ascents, and backward motion are 0.2805 m/s, 0.4018 m/s, and 0.3000 m/s respectively.

Hydrodynamic Efficiency

Hydrodynamic efficiency is the percent of thrust that is actually used to accelerate the ROV through the water. The efficiency of the prototype was determined by first calculating the effective power of the ROV, using the equation:

$$P_E = TV \tag{8}$$

Where P_E is the effective power, *T* is the thrust, and *V* is the velocity of the model. The same thruster data that was collected in order to estimate the velocity was used, and the effective power was found to be 5.24992 Watts. Then, the brake power was determined by plotting the power output of different thrusters being on over time. This can be seen in Figure 8.

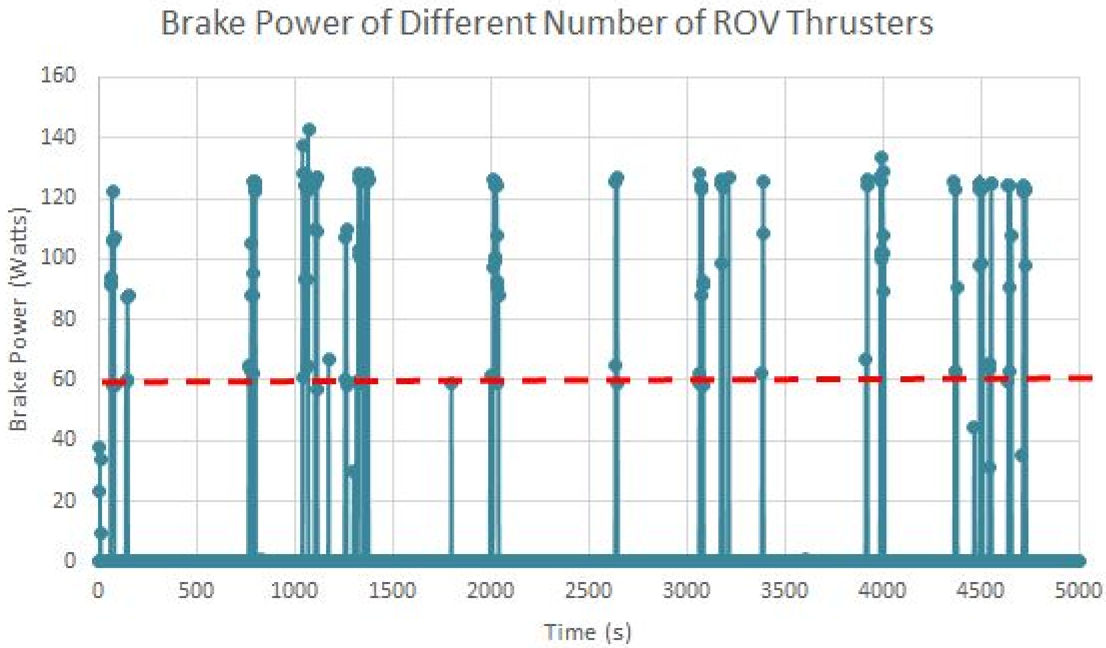


Figure 8. Brake power of thrusters, with average power for 2 thrusters labeled in red.

When the ROV moves forward, only two thrusters are on at the same time, and these points in time, the average brake power is 60.0 Watts. Then, using the following equation, the hydrodynamic efficiency of the model was determined:

$$\eta = \frac{P_E}{P_B} \tag{9}$$

Where η is the hydrodynamic efficiency, P_B is the brake power, and P_E is the effective power. The hydrodynamic efficiency was found to be 8.7%.

Sustainability and Cost

Cost and sustainability analyses were conducted on the ROV. The estimated cost of the model ROV is $166.54. Most of the cost of the model ROV comes from the ½ " PVC tubing needed for the construction of the ROV. Through the sustainability analysis, an overall sustainability score was calculated by adding up BP-assigned sustainability scores of individual components in the ROV. The lower the sustainability score of each component, the more environmentally friendly it is. Since the ROV frame is mainly built out of PVC, a form of plastic which might negatively impact its environment, most of the sustainability points come from the PVC frame. With this in mind, the team only used as much PVC as necessary in order to make a functional ROV. The team also decided to use pool buoys for the permanent buoyancy, since they are more sustainable than foam or duct tape. Overall, the team ended up with a total sustainability score of 65.183.

Performance

Throughout the construction of the ROV, multiple performance tests were conducted in order to aid in design changes, test controls, and test the final design. These performance tests were conducted at the University of Michigan GFL, the Marine Hydrodynamics Lab, and the final competition.

Testing at the GFL

During the testing at the Gorguze Family Laboratory (GFL), the ROV was placed into a large tank of water. Buoyancy adjustments were made throughout uses with different canisters, testing the effectiveness of the adjustable buoyancy. It proved to be sufficient to balance out the weight of the different canisters. Also, ROV steering was practiced for basic maneuvers such as traveling at a steady height, turning, and transporting rings. Figure 9 shows the ROV in the testing tank at the GFL.

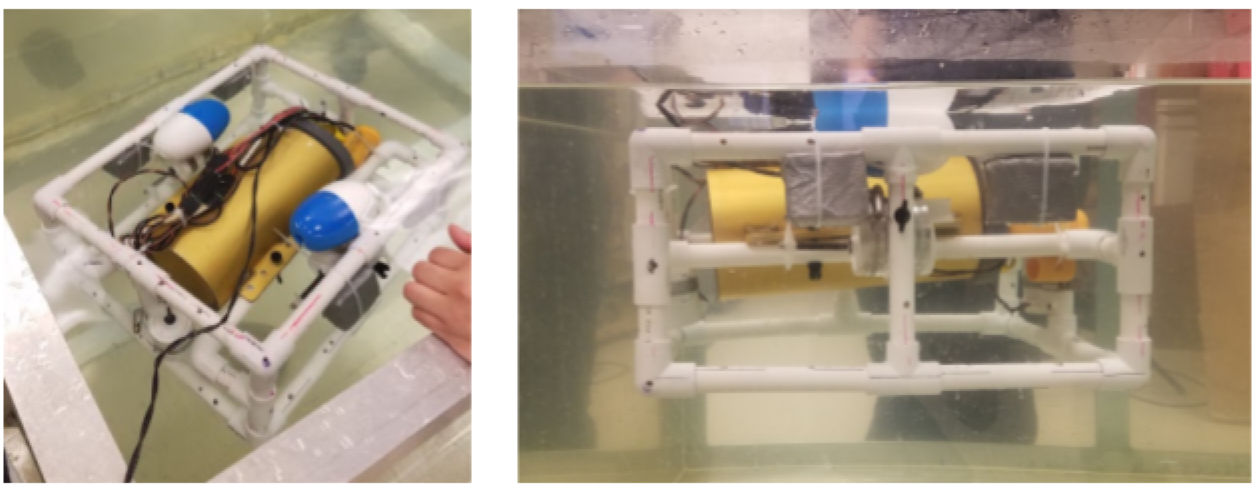


Figure 9. Testing of the ROV in the GFL testing tank

MHL Testing

When the team arrived at the Marine Hydrodynamics Laboratory (MHL), the ROV was fitted with a canister and placed in the water for testing.The biggest issue encountered was the lack of buoyancy, since the MHL water was much colder than that at the GFL, and the adjustable buoyancy had yet to be finalized. Thus, the ROV sank instead of being slightly positively buoyant. Still, due to the fact that no major adjustments could be made at the MHL, the team decided to still perform speed tests.

In order to determine the maximum forward and reverse speeds the ROV was timed while being driven along a 30 ft section of the pool. This was also repeated for the upward and downward directions, where the ROV had to descend to the bottom of the pool or ascend to the top, a distance of 9 ft. Three trials were completed in each direction in, order to account for variations in driving, and to achieve the best possible time. The following equation was used to determine the speed in each direction:

$$V = \frac{d}{t} \tag{10}$$

Where *V* is the velocity, *d* is the distance travelled, and *t* is the time it took to travel that distance. For each of the velocities in each direction, the average time and fastest time were determined and compared to the team's estimated top speed in each direction. This can be seen in Table 2.

Table 2. Actual and estimated speeds of the ROV

Trial	Forward Velocity (m/s)	Backward Velocity (m/s)	Upward Velocity (m/s)	Downward Velocity (m/s)
1	0.4160	0.1743	0.1289	0.3082
2	0.3703	0.1501	0.1567	0.3544
3	0.3790	0.1289	0.1385	0.3014
Average	0.3884	0.1511	0.1414	0.3222
Top Speed	0.4160	0.1743	0.1567	0.3544
Estimated	0.4711	0.3000	0.4018	0.2805

While the estimated velocities for the forward and downward directions were relatively close to the actual top speed, the backward and upward estimated velocities did not coincide with the determined maximum velocities. This is most likely due to the fact that there was not enough buoyancy at the time, and the canister started leaking due to a faulty seal on a wire, so the ROV was sinking. Together with the fact that the ROV was designed to be positively buoyant, which would require less thrust to get it to accelerate upward, and the fact that the thruster were mounted pointing downward due to this design, the upward estimated velocity did not match the actual upward velocity. For the backward velocity, one of the greatest sources of error was the estimated coefficient of drag used in the original estimation. The actual coefficient of drag was greater than the estimated, causing the decrease in maximum velocity.

Competition Results

The competition tasks provided by BP's original project proposal were changed on the day of the competition due to several complications with the camera systems among all ROVs at the competition. The new competition specifications were as follows: each team needed to load a ring onto their ROV's mechanical arm and then proceed to drive their ROV to the other end of the pool, where they would descend down to a "parts bin", and release the ring. Then, teams would send their ROV back to the starting position, and continue to send rings to the other side of the pool in the parts bin until 20 minutes had elapsed. The timer started as soon as the ROV was taken out of the container used to store it, meaning all set up time and initial testing were included as a part of those 20 minutes. A figure of the competition is shown below in Figure 10.

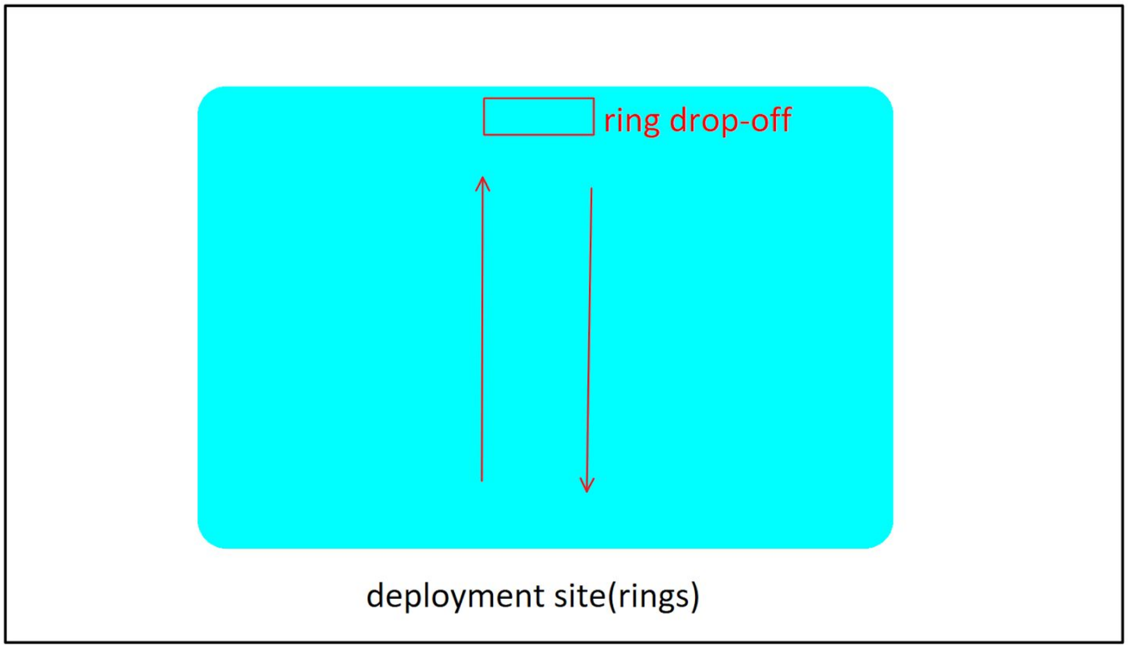


Figure 10. Diagram of Competition Pool.

Overall, the ROV managed to get two rings into the ring drop off site within the 20 minutes, and closely missed a third. This third ring missed its target due to the fact that it was difficult to maneuver the ROV when it came to dropping off the ring. The ring managed to stay on the ROV regardless of how much it pitched - at times even though the ROV was at a 90 degree angle with the pool floor, the ring still refused to fall off. Further iteration may need to be done in order to make the mechanical arm system more accurate and reliable.

Full-scale ROV

This section outlines the methods used to scale up the ROV prototype to full size. The challenges created by an ocean environment will be explained, and thus a proposal for the full-scale ROV materials is included to account for these changes in the work conditions.

Scaling to the Ocean Environment

The prototype was created to model a full-scale ROV. Thus, the dimensions and properties of the model had to be scaled up, per the required scaling factor of $\lambda = 2.5$. The scaling of principal particulars from the model to the full-size ROV is shown in Table 3

Table 3. Table of Principal Particulars, including model dimensions, full-size dimensions, and the scaling method.

Principal Particular	Prototype	Scaling Method	Full-Scale
Length	45.0 cm	$\lambda * L$	112.5 cm
Width	35.0 cm	$\lambda * L$	87.5 cm
Height	25.0 cm	$\lambda * L$	62.5 cm
KG_z	11.9 cm	$\lambda * L$	29.8 cm
KG_x	22.5 cm	$\lambda * L$	56.3 cm
KB_z	12.3 cm	$\lambda * L$	30.8 cm
KB_x	22.3 cm	$\lambda * L$	55.8 cm
C_D	1.2	$C_{D,p} = C_{D,v}$	1.2
$V_{top\ speed,\ forward}$	0.4160 m/s	$(V_m L_m v_s)/(v_m L_s)$	0.2244 m/s
$V_{top\ speed,\ backward}$	0.1501 m/s	$(V_m L_m v_s)/(v_m L_s)$	0.0810 m/s
$V_{top\ speed,\ up}$	0.1567 m/s	$(V_m L_m v_s)/(v_m L_s)$	0.0845 m/s
$V_{top\ speed,\ down}$	0.3554 m/s	$(V_m L_m v_s)/(v_m L_s)$	0.1422 m/s
η	0.087	$\eta_p = \eta_v$	0.087
P_B	60.0 Watts	$\eta = P_E/P_B$	76.5 Watts
Δ	6.9 L	$\Delta * \lambda^3$	107.8 L

As seen in Table 3, the linear dimensions were scaled linearly by multiplying the model length by λ. These include the height, length, width, center of gravity and center of buoyancy. It should be noted that the scaling of center of grabityis only a very rough estimate, since the full-scale materials have very different masses and densities that would alter the center of gravity.

The velocity was scaled using Reynold's scaling, since the viscous forces had a large effect on the ROV. The following equation was used:

$$V_v = \frac{V_p L_p v_v}{v_p L_v} \tag{11}$$

Where V_v is the velocity of the full-scale ROV, V_p is the velocity of the prototype, L_p is the length of the prototype, v_v is the viscosity of the fresh-water the prototype was in, taken to be

$1.004*10^{-6}$ m^2/s, v_p is the viscosity of salt-water which is $1.354*10^{-6}$ m^2/s, and L_v is the length of the full-scale ROV.

In order to scale up efficiency and brake power, it was assumed that the efficiency stays the same for both the prototype and the full-scale ROV. Then the thrust of a full-scale thruster was found to be 20N (*T100 Thruster*). This new thrust for each of the two thrusters was used, together with the scaled velocity, in order to determine the effective power P_E of the full-scale ROV, and this was then converted back into brake power with the known hydrodynamic efficiency.

Finally, the volume of the prototype ROV was determined by adding up all of the volumes of the components. This was then scaled by λ^3, since this a three dimensional value and λ is a one dimensional value.

Influences of Ocean Environment

One of the issues that is created by an ocean environment is the effect of salinity on metals. Salt affects the oxidation of the metal, making it susceptible to rust and weakening the frame. Another issue is the lack of light at great ocean depths. An increasing lack of light could affect the user's control of the ROV, increasing the risk for accidents and mistakes.When creating this model, these potential influences were taken into consideration when designing a full scale ROV and choosing the materials.

Full-scale Materials

The material that would be most suitable for the full scale ROV would be aluminum metal. Aluminum is very durable and lightweight; this is most convenient for an underwater vehicle as it does not require bringing heavy machinery onto the water, making it easier to transport and deploy. Also, it is easily weldable, making it easy for a non-specialized crew to work with. Finally, in order to avoid corrosion over time, the metal will be sprayed with an anodizing paint. The anodizing paint is not harmful to the environment because the finish does break down due to environmental influences, and it does not produce any dangerous byproducts. This is suitable for ocean use because it will not harm the ecosystems and fish in the area.

For buoyancy, a subsea foam will be used. The lifting force is about 3.1lbs (*Subsea Foam*), which is suitable for the full scale ROV as it is not massive in size or mass. This buoyancy should allow the ROV to be positively buoyant in the case that the ROV is disconnected from its control box.

In order to see the ROV's path in the ocean, a camera is needed. The camera used will be a Raspberry Pi Camera Module. The advantages of this camera is that it will be able to rotate 110 degrees, allowing the operator to see around the ROV. This is unique because other cameras can only rotate 68 degrees. This camera will also allow for easy installation and use in the ROV. Since the light visibility decreases as depth increases, a Lumin Subsea Light will be used. The intensity of this light is greater than that of the headlights of a car, showing how it is able to light up ocean environments more easily. A light in the ocean is needed to guide the ROV through the water to avoid the danger of hurting the sea animals or crashing into any debris.

Moreover, a tether will be needed to descend into the ocean. Therefore, the tether that will be used is a 200 m Fathom ROV Tether. While the ROV tether is estimated at 200 m, the ROV can go from 30 to 3000 m, so the cost might change in accordance to the length of the ROV.

The cost of construction for the full scale ROV is $1791.73. The majority of the cost comes from the thrusters, the LED lighting, and the tether. The length of the tether might change, so the overall cost of the ROV might change but the given cost was calculated at 200 m. While the cost might be high, the materials have a major advantage in sustainability. When calculated, the sustainability score was 69.433; just as for the prototype, the lower the score, the more environmentally friendly the ROV will be. With higher quality materials, the ROV will be affected less by the environment; this will allow for more descents into the ocean without worry of pollution. Additionally, because the aluminum is anodized, it has a longer lifespan, allowing the ROV to be used for many years. This reduces the long term cost that is needed to operate the ROV because less money will be spent on ROV maintenance.

Conclusions and Recommendations

Agile^3 designed and constructed a readily deployable ROV. The modest prototype mass and dimensions of 25.0 x 35.0 x 45.0 cm meet BP’s stipulations, and strong securements give it the desired durability. The design also incorporates thruster adjustability, to allow for future changes in ROV applications. To ensure stability, buoyancy is attached to the top of the frame. The ROV was designed to be slightly positively buoyant to make it recoverable, should communications be severed. To cope with this, two thrusters are directed in the direction of the keel; this design provides sufficient downward thrust. The other two are directed forwards to increase maximum forward velocity. This increased maximum velocity will reduce time spent going from one work location to the next. Adjustable buoyancy accommodates for modified payloads. Additionally, the ROV has a mechanical that is attached with a bolt prior to deployment.

To make the controls intuitive, a custom code was written, which will make pitching, yawing, vertical, and horizontal motion linked to specific switches. This design will allow the driver to control multiple thrusters with four simple buttons, creating a more ergonomic design that can be easily handled.

Stability was achieved by designing the ROV such that the center of buoyancy was above and almost directly over the center of gravity. While self-righting, the ROV’s centers of buoyancy and gravity are close to each other to allow for effective pitching.

The full scale ROV will have an anodized aluminum frame, a maximum speed of 0.1884 m/s in the forward direction, and a per unit cost of $1791.73, excluding labor and facilities. Its final dimensions will be 112.5 x 87.5 x 62.5 cm, corresponding to length, width, and height respectively.

If any adjustments to the design are required, or if more information on any aspect of the design is desired, the team can be reached at agile3rov@umich.edu .

References

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. *Journal of Engineering Education, 94*(1), 103-120.

Drag Coefficient (n.d.). The Engineering ToolBox. Retrieved from http://www.engineeringtoolbox.com/drag-coefficient-d_627.html

Fathom ROV Tether (n.d.) BlueRobotics. Retrieved from http://www.bluerobotics.com/store/cables/fathom-tether-nb-4p-26awg-r1/

Lumin Subsea Light (n.d.) BlueRobotics. Retrieved from http://www.bluerobotics.com/store/electronics/lumen-light-r1/

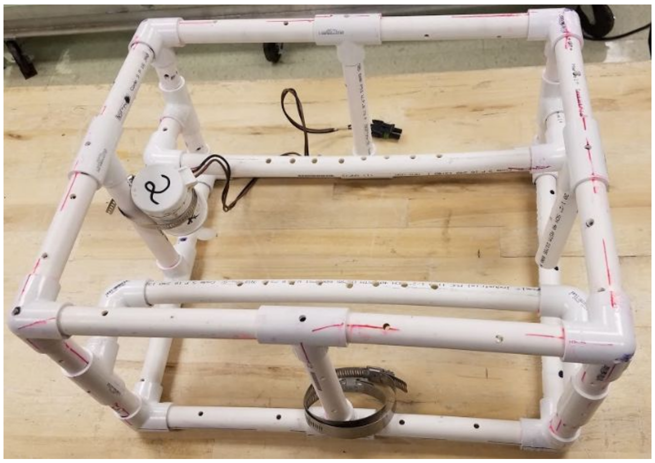
Raspberry Pi Camera Module (n.d.) BlueRobotics. Retrieved from http://www.bluerobotics.com/store/electronics/cam-rpi-wide-r1/

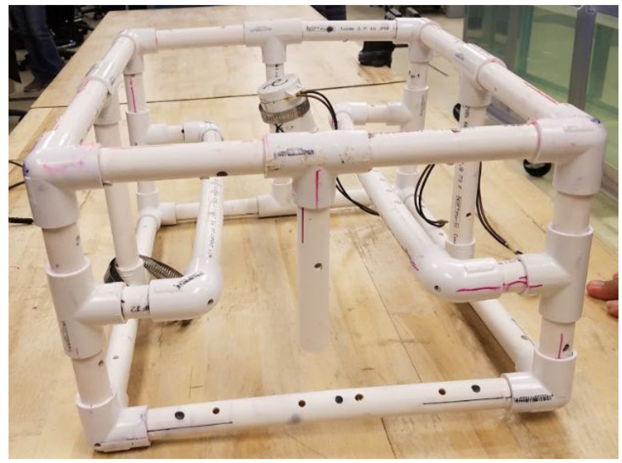
Subsea Foam (n.d.) BlueRobotics. Retrieved from http://www.bluerobotics.com/store/parts/float-r1/

T100 Thruster (n.d.) BlueRobotics. Retrieved from http://www.bluerobotics.com/store/thrusters/t100-thruster/

Appendices

Appendix A. Assembled ROV frame





Appendix B. Control Box Code.

Note: The code was written, and executed using an Arduino Uno and the Arduino IDE.

```
/* ROV remote controller arduino program    for eng100-600  by JAG, 10/2016
   the data structure is 4 bytes plus a 5th check byte which
   is the XOR product of the 4 data bytes.
   in the first byte, bits 3,2,1,0 are motor aFor(front),aRec(front),bFor(back),bRev(back)
   in the second byte, bits 3,2,1,0 are motor cFor(left), cRev(left), dFor(right), dRev(right)
   the third byte is the 1st servo position in degrees as the byte value
   the fourth byte is the 2nd servo position..

   to load the program, disconnect the BT xmitter to enable the USB connection to the programmer

*/

//a is up/down
//b is tilt
//c is left
//d is right

#define aFor 13                               //8 pins for 8 motor control functions
#define aRev 12                               // comment out the constant replacement
#define bFor 11                               // for the motor pins if programming
#define bRev 10                               // a custom control
#define cFor  9
#define cRev  8
#define dFor  7
#define dRev  6

#define servOnePin A0
#define servTwoPin A1
#define ledPin A5


const int setsPerSec = 7;                    //how fast to send data sets
const byte biteF = 0xFF;                     //this is the flag(start) byte
```

```
const byte PASS = 0x55;                       //confirmation byte
const byte FAIL = 0x0F;                       // Data check fail byte

byte chkbyte;
byte rdgs[5] = {0, 0, 0, 0, 0};                //the data bytes read from the bluetooth
int pinMap[8] = {aFor, aRev, bFor, bRev, cFor, cRev, dFor, dRev};
int arrayOfst;                                 //+1 -1 holder for motor setting loop
int delayTime;


void setup()   {

  Serial.begin(115200);
  delay(1000);

  for ( int i = 0 ; i < 8 ; ++i )   {                //initialize dig pins to read inputs
    pinMode( pinMap[i], INPUT );
  }

  pinMode(servOnePin, INPUT);
  pinMode(servTwoPin, INPUT);
  pinMode( ledPin, OUTPUT);
  delayTime = 1000 / setsPerSec;                       //calculate loop delay variable
}

void loop()  {

  if ( !Serial.available() ) {                       //if no return handshake turn off indicator
    digitalWrite ( ledPin, LOW );
  }
  else {
    for ( int j = Serial.available() ; j >  0 ; --j ) {
      chkbyte = Serial.read();                       //read a byte. Is it the confirmation value?
    }
    if ( chkbyte == PASS ) {                         //if payload is getting data, turn on the
indicator
      digitalWrite(ledPin, HIGH);
    }
```

```
    else  digitalWrite( ledPin, LOW);
  }



  // write data to the xmit array
  // the servo positions are the 3rd and 4th byte

  //  rdgs[2] = byte(constrain(map(analogRead(servOnePin),0,1023,0,180),1, 179));
  //  rdgs[3] = byte(constrain(map(analogRead(servTwoPin),0,1023,0,180),1, 179));
  rdgs[2] = rdgs[3] = 0;                  //no servos installed. With this line, A0 and A1 do not
need grounding.



  /*****************DO NOT ALTER ABOVE THIS LINE !************************************
     The block between the comment lines is where the motor control logic can be changed
     for custom control, comment it out and replace it with something else.
     Your logic shoud require that if both the forward and reverse buttons are
     pressed at the same time, the controller sends "off" for both
     to prevent any possibility of generating a shorted condition in the payload

     The loop below that sets the output bits is an example of how you can
     write very dense code in 'C'. This one loop is relatively arcane
     to figure out but is short and easy to debug once you do. In the lines
     further below is an example of the same operations done in single commands
     to set each bit where things are more explicit.
  */



  /*arrayOfst = 1;                                    // 1 or -1 picks the paired Foward or
Reverse
    for ( int i = 0 ; i < 8 ; ++i )   {
    int k = i / 4;                               //this is either 0 or 1 to pick the byte
    int j = 3 - (i % 4);                         //this counts from 3 to 0 to pick the bit

    if ( digitalRead( pinMap[i] ) && !digitalRead(pinMap[i + arrayOfst] ) )  {
      bitWrite ( rdgs[k], j, 1 );               //only turn on if both bottons NOT pushed
    }
    else bitWrite( rdgs[k], j, 0 ) ;
```

```
    arrayOfst = arrayOfst * (-1);                 //flip the counter. For a "forward", the
corresponding
    //'Reverse' to check is hte next value counting up. For a 'Reverse'
    // it is the next value counting down.
    }*/

  /**********************************************************************************/

  /* in this block it's easier to see how to set each bit
     based on some combination of switches. This can be used
     as a starting point for designing a custom motor
     switch control scheme.
  */


  if (digitalRead(aFor) && !digitalRead(aRev)) {
    //turn off everything
    bitWrite(rdgs[0], 0, 0);
    bitWrite(rdgs[0], 1, 0);
    bitWrite(rdgs[0], 2, 0);
    bitWrite(rdgs[0], 3, 0);


    bitWrite( rdgs[0], 2, 1);
    bitWrite(rdgs[0], 0, 1);
  } else if (digitalRead(aRev) && !digitalRead(aFor)) {
    //turn off everything
    bitWrite(rdgs[0], 0, 0);
    bitWrite(rdgs[0], 1, 0);
    bitWrite(rdgs[0], 2, 0);
    bitWrite(rdgs[0], 3, 0);


    bitWrite( rdgs[0], 3, 1);
    bitWrite(rdgs[0], 1, 1);
  } else if (!digitalRead(aFor) && !digitalRead(aRev) && digitalRead(bFor) && !digitalRead(bRev))
{
```

```
    //turn off everything
    bitWrite(rdgs[0], 0, 0);
    bitWrite(rdgs[0], 1, 0);
    bitWrite(rdgs[0], 2, 0);
    bitWrite(rdgs[0], 3, 0);



    //tilt forward
    bitWrite( rdgs[0], 3, 1);
    bitWrite(rdgs[0], 0, 1);
  } else if (!digitalRead(aFor) && !digitalRead(aRev) && !digitalRead(bFor) && digitalRead(bRev))
{
    //turn off everything
    bitWrite(rdgs[0], 0, 0);
    bitWrite(rdgs[0], 1, 0);
    bitWrite(rdgs[0], 2, 0);
    bitWrite(rdgs[0], 3, 0);



    //tilt backwards
    bitWrite( rdgs[0], 1, 1);
    bitWrite(rdgs[0], 2, 1);
  } else {
    //if none of the cases are met, then bitwrite 0 to all thrusters.
    bitWrite(rdgs[0], 0, 0);
    bitWrite(rdgs[0], 1, 0);
    bitWrite(rdgs[0], 2, 0);
    bitWrite(rdgs[0], 3, 0);
  }

  /*
    if ( digitalRead(aFor) && (!digitalRead(aRev)))  {                  //up
      bitWrite( rdgs[0], 3, 1);
      bitWrite(rdgs[0], 1, 1);

    }
    else  {
      bitWrite( rdgs[0], 3, 0);
```

```
    bitWrite(rdgs[0], 1, 0);
  }

  if ( digitalRead(bFor) && (!digitalRead(bRev)))  {                    //tilt_forward
    bitWrite( rdgs[0], 3, 1);
    bitWrite(rdgs[0], 0, 1);
  }
  else {
    bitWrite( rdgs[0], 3, 0);
    bitWrite(rdgs[0], 0, 0);
  }
*/
if ( digitalRead(cFor) && (!digitalRead(cRev)))  {                    //left
  bitWrite( rdgs[1], 3, 1);
}
else  bitWrite( rdgs[1], 3, 0);

if ( digitalRead(dFor) && (!digitalRead(dRev)))  {                    //right
  bitWrite( rdgs[1], 1, 1);
}
else  bitWrite( rdgs[1], 1, 0);
/*
  if ( digitalRead(aRev) && (!digitalRead(aFor)))  {                    //down
    bitWrite( rdgs[0], 2, 1);
    bitWrite(rdgs[0], 0, 1);
  }
  else {
    bitWrite( rdgs[0], 2, 0);
    bitWrite(rdgs[0], 0, 0);
  }

  if ( digitalRead(bRev) && (!digitalRead(bFor)))  {                    //tilt_reverse
    bitWrite( rdgs[0], 1, 1);
    bitWrite(rdgs[0], 2, 1);
  }
  else {
    bitWrite( rdgs[0], 1, 0);
    bitWrite(rdgs[0], 2, 0);
```

```
    }
  */
  if ( digitalRead(cRev) && (!digitalRead(cFor)))  {                  //left_reverse
    bitWrite( rdgs[1], 2, 1);
  }
  else  bitWrite( rdgs[1], 2, 0);

  if ( digitalRead(dRev) && (!digitalRead(dFor)))  {                  //right_reverse
    bitWrite( rdgs[1], 0, 1);
  }
  else  bitWrite( rdgs[1], 0, 0);



  /************************DO NOT ALTER BELOW THIS LINE**************/

  rdgs[4] = rdgs[0] ^ rdgs[1] ^ rdgs[2] ^ rdgs[3];
  //create check byte and send the start byte
  Serial.write(biteF);                                                 //and then the data array
to the payload
  for (int i = 0 ; i < 5 ; ++i ) {
    Serial.write(rdgs[i]);
  }
  delay(delayTime);                                                    //dont go faster than the
payload can process data
}
```

Appendix C. Center of Gravity Calculation Values for Z-Direction

Item	Count	Unit Mass (g)	Total Mass (g)	Height (cm)
Canister	1	2432.0	2432.0	11.6
Thruster	2	238.9	477.8	14.0
Thruster	2	238.9	477.8	9.0
6(⅞)” pipe	4	57.4	229.6	22.0
6(⅞)” pipe	4	57.4	229.6	2.0
6(⅞)” pipe	1	57.4	516.5	11.6
T-Connector	4	27.7	110.8	20.0
T-Connector	4	27.7	110.8	11.6
T-Connector	3	27.7	83.1	2.5
3-Way Elbow	4	29.1	116.4	20.0
3-Way Elbow	4	29.1	116.4	2.5
5(¼)” Pipe	4	32.6	130.4	22.0
5(¼)” Pipe	2	32.6	65.2	2.0
45 cm Pipe	2	93.1	186.3	11.6
Pool Buoy	2	47.4	94.8	22.0
Elbow Connector	4	21.8	87.1	11.6
3” Pipe	4	18.2	72.7	16.5
11(½)” Pipe	1	72.4	72.4	2.0
2(⅝)” Pipe	4	16.5	66.0	6.5
2(¼)” Pipe	4	13.9	55.5	11.6
Mesh Bag	2	21.9	43.9	13.5
7" pipe	1	43.8	43.8	11.6
Foam Piece Large	2	19.0	38.0	18.0
Foam Piece Small	8	4.0	32.1	15.0
4(⅞)” Pipe	1	18.5	18.5	13.0

Appendix D. Center of Gravity Calculation Values for X-Direction

Item	Count	Unit Mass (g)	Total Mass (g)	Depth (cm)
Canister	1	2432.0	2432.0	22.5
Thruster	2	238.9	477.8	22.5
Thruster	1	238.9	238.9	4.5
Thruster	1	238.9	238.9	40.5
6(⅞)” pipe	4	57.4	229.6	11.3
6(⅞)” pipe	4	57.4	229.6	33.8
6(⅞)” pipe	1	57.4	57.4	43.0
T-Connector	4	27.7	110.8	2.0
T-Connector	4	27.7	110.8	22.5
T-Connector	3	27.7	83.1	43.0
3-Way Elbow	4	29.1	116.4	2.0
3-Way Elbow	4	29.1	116.4	43.0
5(¼)” Pipe	4	32.6	130.4	2.0
45 cm Pipe	2	93.1	186.3	22.5
Pool Buoy	2	47.4	94.8	22.5
Elbow Connector	2	21.8	43.6	2.0
Elbow Connector	2	21.8	43.6	43.0
3” Pipe	2	18.2	36.4	2.0
3” Pipe	2	18.2	36.4	43.0
11(½)” Pipe	1	72.4	72.4	43.0
2(⅝)” Pipe	2	16.5	33.0	2.0
2(⅝)” Pipe	2	16.5	33.0	43.0
2(¼)” Pipe	2	13.9	27.8	43.0
2(¼)” Pipe	2	13.9	27.8	2.0
Mesh Bag	2	21.9	43.9	11.3

7" pipe	1	43.8	43.8	2.0
Foam Piece Large	2	19.0	38.0	33.8
Foam Piece Small	8	4.0	32.1	11.3
4(⅞)” Pipe	1	18.5	18.5	43.0

Appendix E. Center of Buoyancy Calculation Values for Z-Direction

Item	Count	Unit Volume (mL)	Total Volume (mL)	Height (cm)
Canister	1	3180.0	3180.0	11.6
Thruster	2	150.0	300.0	14.0
Thruster	2	150.0	300.0	9.0
6(⅞)” pipe	4	30.8	123.2	22.0
6(⅞)” pipe	4	30.8	123.2	2.0
6(⅞)” pipe	1	30.8	30.8	11.6
T-Connector	4	20.0	80.0	20.0
T-Connector	4	20.0	80.0	11.6
T-Connector	3	20.0	60.0	2.5
3-Way Elbow	4	20.0	80.0	20.0
3-Way Elbow	4	20.0	80.0	2.5
5(¼)” Pipe	4	23.4	93.6	22.0
5(¼)” Pipe	2	23.4	46.8	2.0
45 cm Pipe	2	79.3	158.6	11.6
Pool Buoy	2	350.0	700.0	22.0
Elbow Connector	4	15.0	60.0	11.6
3” Pipe	4	13.4	53.6	16.5
11(½)” Pipe	1	51.5	51.5	2.0
2(⅝)” Pipe	4	11.8	47.2	6.5
2(¼)” Pipe	4	10.0	40.0	11.6
Mesh Bag	2	25.0	50.0	13.5
7" pipe	1	31.4	31.4	11.6
Foam Piece Large	1	129.0	158.0	18.0
Foam Piece Small	8	38.1	305.0	15.0
4(⅞)” Pipe	1	21.9	21.9	13.0

Appendix F. Center of Buoyancy Calculation Values for X-Direction

Item	Count	Unit Volume (mL)	Total Volume (mL)	Depth (cm)
Canister	1	3180.0	3180.0	22.5
Thruster	2	150.0	300.0	22.5
Thruster	1	150.0	150.0	4.5
Thruster	1	150.0	150.0	40.5
6(⅞)” pipe	4	30.8	123.2	11.3
6(⅞)” pipe	4	30.8	123.2	33.8
6(⅞)” pipe	1	30.8	30.8	43.0
T-Connector	4	20.0	80.0	2.0
T-Connector	4	20.0	80.0	22.5
T-Connector	3	20.0	60.0	43.0
3-Way Elbow	4	20.0	80.0	2.0
3-Way Elbow	4	20.0	80.0	43.0
5(¼)” Pipe	4	23.4	93.6	2.0
45 cm Pipe	2	79.3	158.6	22.5
Pool Buoy	2	350.0	700.0	22.5
Elbow Connector	2	15.0	30.0	2.0
Elbow Connector	2	15.0	30.0	43.0
3” Pipe	2	13.4	26.8	2.0
3” Pipe	2	13.4	26.8	43.0
11(½)” Pipe	1	51.5	51.5	43.0
2(⅝)” Pipe	2	11.8	23.6	2.0
2(⅝)” Pipe	2	11.8	23.6	43.0
2(¼)” Pipe	2	10.0	20.0	43.0
2(¼)” Pipe	2	10.0	20.0	2.0
Mesh Bag	2	25.0	50.0	11.3

7" pipe	1	31.4	31.4	2.0
Foam Piece Large	2	129.0	258.0	33.8
Foam Piece Small	8	38.1	304.8	11.3
4(⅞)” Pipe	1	21.9	21.9	43.0

Appendix G. Thruster Data

Thruster	Direction	Load Cell Voltage	Battery Voltage	Battery Current	Load Cell Force (N)	Thrust (N)
1	forward	2.114	12.07	2.47	18.73	6.291
1	backward	1.934	-12.1	-2.224	7.205	2.421
2	forward	2.032	12.15	1.964	13.48	4.528
2	backward	1.899	-12.16	-1.747	4.965	1.668
3	forward	2.038	12.03	2.83	13.86	4.657
3	backward	1.952	-12.04	-2.728	8.358	2.808
4	forward	2.116	12.03	2.682	18.85	6.334
4	backward	1.947	-12.05	-2.395	8.037	2.700

Appendix H. Maximum Velocity Estimation Code

Note: The code was written, and executed in Matlab R2016b.

```
function [maxVel] = velMax(T)
%Water density estimated from data at
%https://www.ncsu.edu/chemistry/resource/
%H2Odensity_vp.html .
rho = 999.1026; %kg/m^3
A_ROV = 26 .* 36.5; %594; %Rough approx. assuming all piping 1" diameter, a 5" diameter
             %cannister, and 3" diameter buoyancy around piping. Thruster
             %drag was ignored. Given in cm^2

C_d = 1.2; %"Hollow semi-cylinder opposite stream" from
           %http://www.engineeringtoolbox.com/drag-coefficient-d_627.html
           %for approximation of C_d.
maxVel = sqrt(T ./ (C_d .* .5 .* rho .* (.0001 .* A_ROV)));
end

%Units(N,kg/m^3,m^2)
%Read in data...
[thrusterData,directions] = xlsread('Thruster Data.xlsx');
directions = directions(2:9,2);directions;
%Calculate total thrust (We assume Thrust = Drag at
%max velocity.)...
    %...forward...
T_f = thrusterData(1,7) + thrusterData(7,7);
maxVel_f = velMax(T_f);
display(maxVel_f), disp('m/s');
    %...backward...
T_b = thrusterData(2,7) + thrusterData(8,7);
maxVel_b = velMax(T_b);
display(maxVel_b), disp('m/s');
    %...up...
T_u = thrusterData(3,7) + thrusterData(5,7);
maxVel_u = velMax(T_u);
display(maxVel_u), disp('m/s');
    %...forward...
T_d = thrusterData(4,7) + thrusterData(6,7);
maxVel_d = velMax(T_d);
display(maxVel_d), disp('m/s');
```