Technical Design Report for Charybdis

Tyler Gragg
Software
KSU AUV
Powder Springs, USA
robots@tylergragg.com

Jacob Larson
Mechanical
KSU AUV
Powder Springs, USA
jacob.m.larson@gmail.com

Charlie McDermitt
Software
KSU AUV
Marietta, USA
charlie@charliemcdermitt.com

Cody Meier
Electrical
KSU AUV
Austell, USA
codyrivers3321@gmail.com

Blair Sailor
Mechanical
KSU AUV
Woodstock, USA
rrhs.sailor.ryan@gmail.com

Logan Spencer
Mechanical
KSU AUV
Dunwoody, USA
lspencer9349@gmail.com

***Abstract*— The Kennesaw State University Autonomous Underwater Vehicle Team (KSU AUV) built and designed Charybdis with the intent to modify and enhance it over seasons to come. Initially developed over the course of one year, the AUV's motor setup and control systems run in parallel with common technology being used in aerial drones. This vehicle utilizes a PixHawk flight controller as both a motor controller and gyroscopic sensor. The communications between a dual camera system and aforementioned flight controller govern the movement of the AUV. Successfully creating an autonomous underwater vehicle requires interdisciplinary cooperation between team members.**

Keywords—autonomy, underwater, vision, design, navigation

I. Competition Strategy

At the 2018 RoboSub competition, the former AUV model, Via Maris, qualified for the semifinals by passing through the start gate [1]. Because of a series of setbacks in manufacturing, autonomy, and design oversight, this sub was unable to consistently complete the qualifying task and make use of its peripheral systems. In addition, its large size cost us points and made transportation difficult and expensive. With this experience in mind, this year's vehicle needed to be compact, significantly lighter, and consistently able to find and perform its tasks.

For this year, the team decided to focus on designing a maneuverable sub that would act as a foundation for future improvements. We decided to shift our design and manufacturing process to a two-year cycle that will be more appropriate for our budget and team size. This is the first year of that cycle; we plan to use an improved version of Charybdis next year as well. Charybdis is smaller and lighter than previous KSU AUVs, which leaves more physical space on the sub and more room under the competition weight limit. In the future, we will be able to add additional subsystems and improvements to the sub, including hydrophones, improved manipulators, and other sensors. We also made the sub light to leave open the possibility of constructing another smaller sub in the future that could communicate with Charybdis and complete simpler tasks while Charybdis itself completes the more complex ones.

II. Vehicle Design (Novel Aspects)

A. Mechanical Design

1) Outer Structure: The external frame of Charybdis is made primarily of 6061-T6 aluminum. The aluminum sheets were cut with a waterjet by team members, reducing manufacturing costs.

This frame design is a drastic change from the team's previous frames, which were modular and made from t-slotted aluminum extrusion. The weight saved by changing to aluminum sheets outweighed the benefits of using the modular extrusion. The previous sub weighed approximately 120 lbs. in air to operate at neutral buoyancy. The new sub, Charybdis, is 46% lighter at only 65 lbs. The reduced weight increases efficiency, maneuverability, and ease of transportation and should qualify Charybdis for a point bonus instead of the penalty we received last year.

The sub was also built small enough to easily fit through doors. As seen in the rendering of Charybdis in Fig. 1, four motors are mounted on aluminum extrusion on hinged mounts, which fold in toward the sub's body. When folded, the sub should easily fit through tight doorways from any angle, and with the motors folded or removed it fits in a small case for shipping.

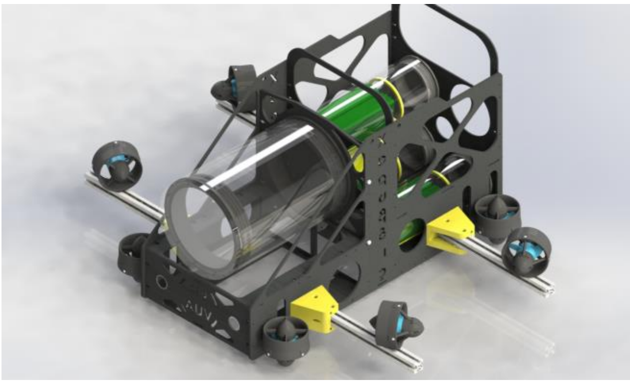

Figure 1: Charybdis Current Design

2) Housing: The acrylic housings, the most crucial pieces of structure on the sub, protect its onboard electronics from the water. Instead of using a single large housing like our previous designs, Charybdis has five different clear acrylic tubes manufactured by Blue Robotics that hold electronic components. This makes it easier to repair and test individual components and also offsets the modularity lost with the new frame because individual pods can be modified without affecting the components in others.

Four 4 in. tubes in the back of the sub hold batteries, and one 8 in. diameter tube in the front holds the rest of the electronics. Acrylic is an ideal material for this purpose because it is light and transparent enough to contain cameras. The four smaller tubes have about the same buoyancy as the large tube, giving a center of buoyancy near the center of gravity and the center of thrust.

3) Inner Structure: The inner structure shown in Fig. 2, which rests inside the large acrylic housing, was designed to allow easy and stable access to electronics, act as a heat sink, keep components from shifting with movement, and be modular enough to change with any future upgrades. Manufactured from 3D printed PLA plastic and waterjet-cut 6062-T6 aluminum sheets, it balances heat sink capability and durability with a slightly decreased weight.

This structure (and the attached electronics) stay statically mounted to the frame and are covered by the acrylic tube sliding over it, unlike last year's design in which the tube was static and the electronics were inserted. This design is more convenient for us to work on and keeps the electronics safer, since we have to move them less often. We used a shelving design similar to the previous year's, but modified to give us easier access to sub internals and steadier mounting points for them.

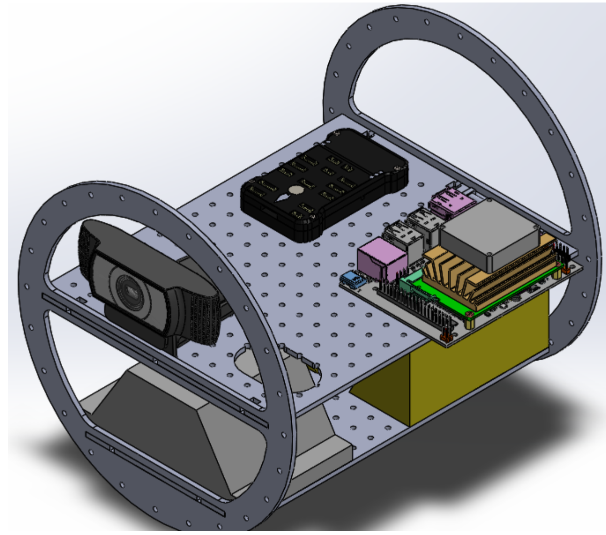

Figure 2: Charybdis Internals Mockup

4) Robotic Arm and Deployment Mechanism: This year's competition makes use of manipulable tokens in the form of garlic and crucifixes, so our manipulator must be designed to retrieve, store, and deposit these objects. The largest design challenge was increasing the margin of error for object retrieval to allow the sub's positioning to be less precise. The manipulator must effectively store and retrieve markers from an onboard storage system and deposit them accurately in the scoring areas. This was especially challenging because of the multiple odd shapes that must be retrieved and stored.

Due to the design's complexity, we determined that we would need two years to design, manufacture, construct, and implement a fully functional robotic arm in addition to our other systems. As a result, we began the design process this year and plan to complete it in time for RoboSub 2020.

5) Torpedo System: We chose unpowered torpedoes for this year's competition to conserve space. Unpowered torpedoes can be smaller, and compressed air can be stored separately from the torpedoes, letting us fit the system across smaller spaces. We chose to 3D print our torpedo designs so that we could design, build, test, and iterate on them rapidly while staying within our budget. The torpedoes shown in Fig. 3 are based on military torpedoes and stabilize by spinning. Like rifling in firearms, the spin keeps the torpedo from straying off-course, and the effect is magnified by water's higher density.

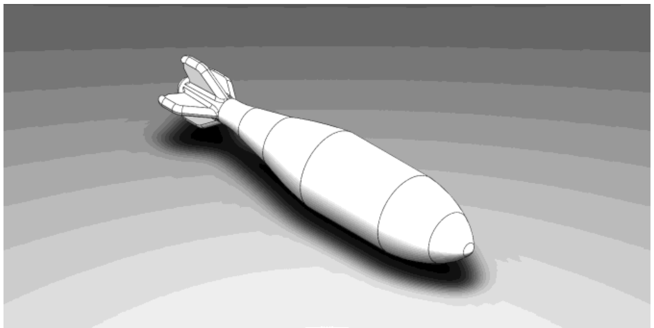

Figure 3: Current Torpedo Design

The first propulsion design used one 10 oz. paintball tank pressurized to 300 psi. Air moves through four tubes to flow restrictors, lowering the pressure to 120 psi, before electric solenoids join the flow restrictors to the torpedo tubes. The midsection of the torpedo forms a seal with the inner wall of the tube, allowing the compressed air to push the torpedo out. This means that the fins can be no larger than the torpedo's body, limiting their effectiveness.

B. Electrical Design

1) External Electronics: The connections between sub electronics are shown in Fig. 4. Charybdis utilizes eight BlueRobotics thrusters for maneuverability. Eight electronic speed controllers (ESC) control and regulate the speed of the thrusters. The ESCs receive instructions by pulse width modulation from the PixHawk and give us the ability to control the rotational speed and direction of the thrust.

2) Printed Circuit Board: For Charybdis, the electrical division designed a new circuit board for our killswitch. The killswitch board uses MOSFETs, with the gate connected to an optocoupler, which in turn is controlled by an Arduino-based potentiometer. The MOSFETs are set in parallel to minimize the heat output from each MOSFET and prevent burning them out.

3) Power Distribution: Five lithium polymer batteries power the sub. The sub power systems are divided into two primary categories: computer systems and propulsion. Separating these systems into two categories simplifies power distribution and reduces noise and crosstalk for electrical components. There are two primary voltage rails: 19.5 V rail that will power the motherboard and the 5 V rail which powers sensors and controllers.

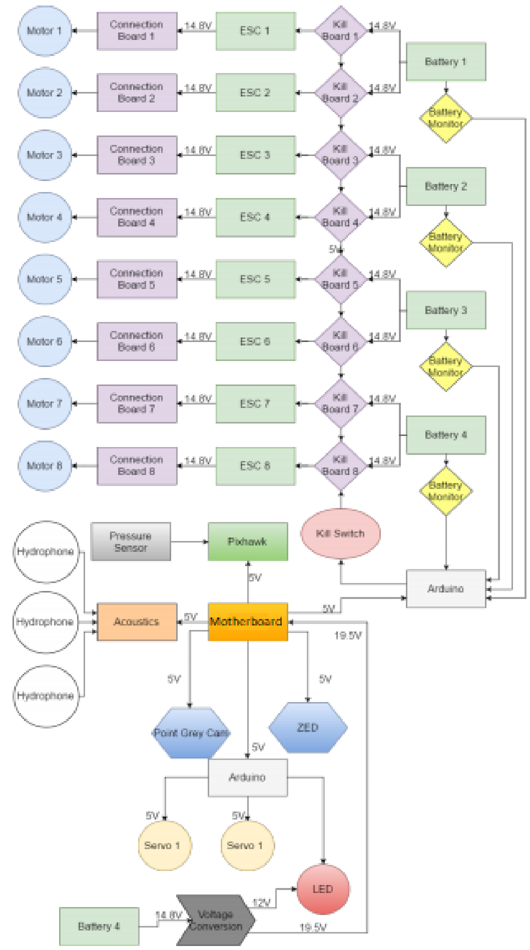

Figure 4. Electrical System Overview

4) Sensors: The sub has a full sensor suite corresponding to the events in which the team chose to compete. It has two cameras: one facing forward to locate and work through the challenges, and one facing the floor to track task objects and path markers on the pool floor. It also has a pressure sensor, which tracks the sub's depth and is used by the Pixhawk flight controller to hold the sub steady.

The team considered adding hydrophones to the sub this year, but decided against it because they would have stretched our limited resources in terms of both time and money. We plan to include them for a future competition.

C. Software Design

1) Hardware: After using a full-size PC last year, we explored other options to reduce sub size and weight. Initially, we considered the same PC, a Raspberry Pi 3 B+, and an ODROID-XU4, but we chose the Nvidia Jetson Nano when it was released in March 2019. The Jetson Nano is small, cheap, and powerful for its size - preliminary tests indicate that the Jetson Nano with Tensorflow is able to process each frame almost twice as quickly as the same network running on the Intel Movidius Neural Compute Stick with Caffe, which we used last year [1]. It also has fewer hardware compatibility issues than the Movidius stick, which allowed us to use newer versions of some other software like Ubuntu and saved time that we can use on other projects.

2) Architecture: The software architecture of Charybdis is based on the Robot Operating System (ROS), which provides a message-passing system and networking capabilities, among other functions. [5]. The packages we created were designed to take advantage of ROS and the open-source libraries that use it, including SMACH, MavROS, and rosserial.

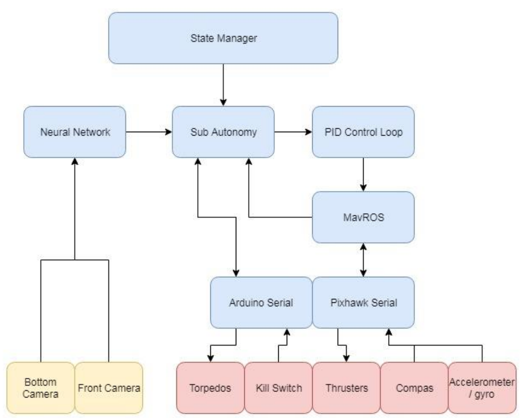

Figure 5: Software architecture

3) High-level Control: High-level decisions about what Charybdis should do are made in a state machine implemented in SMACH, a ROS package that defines a state machine structure. [6]. Each state performs a competition task or part of a task: for example, to get through the gate, Charybdis passes through up to seven different states. One is the start state, four (implemented as a smaller state machine) combine to form a search pattern, one tracks the gate once it has been detected, and one passes through the gate once Charybdis is close to it. The implementation of the other tasks is architecturally similar. SMACH allows us to create complex state machines

4) Vision: Video input is received from two USB cameras, one forward-facing and one downward-facing, and sent to the object detection algorithm via ROS. Due to the noise and variability of the competition environment, the light-scattering effects of water, and the time constraints at the competition, we deemed the use of machine-learning based object detection more effective than the creation of hand-crafted detection algorithms. We chose to use the SSD (Single Shot Detector) architecture with MobileNet, implemented in Tensorflow, because of its availability and performance - while less accurate than some other types of network like R-CNN, SSD is accurate enough for our needs while still performing well on our limited hardware.

Once the network visualizes the detections, we can perform movements based on that information. We take two points from the field of view: one provided by SSD and one provided by the center of the camera. The program calculates the error between the two points and processes the error through a PID control loop, then outputs an RC value published to MavROS.

5) MavROS: MavROS, a ROS wrapper for the Pixhawk's MavLink software, serves as an all-in-one package to control movement of the submarine by publishing virtual RC controller values to the Pixhawk flight controller. [7]. We used the Pixhawk because the open source community which developed Ardusub has created custom firmware for controlling AUVs that is easily wrapped with MavLink and MavROS for communication [8].

6) Arduino Auxiliary Control: Controlling external mechanisms on the sub requires an external interface, which we implemented through an Arduino over serial communication. The Arduino allows us to send commands to the torpedo launcher and manipulator and monitors the sub's killswitch to keep it aware of its current state.

III. Experimental Results

A. Torpedo Experimentation

Torpedo design was one area in which we performed detailed experiments. The team simulated different combinations of pressure, drag and lift coefficients, leading us to base the torpedo design in Fig. 4 on military torpedoes which maximize lift. Since the torpedo needed to be positively buoyant, we thought that the fins needed to slightly push the torpedo downward to keep the straightest line of travel between the tube and the target.

The tests determined that since any rotation induced when the torpedo was launched could send it off course, shaping the fins to increase lift would likely do more harm than good. Instead, we decided to increase accuracy by spinning the torpedo instead.

With this in mind, the team performed a parametric "what-if" simulation in SolidWorks to find the dimensions that would give us the most torque and least drag. Using geometry from the initial design, the radii of the body, nose, tail, top of the fin, and bottom of the fin, plus the fin angle, were adjusted and run iteratively through SolidWorks Flow Simulation Add-on with the results shown in Fig. 6 and Fig. 7. This information was used to determine the most recent design.

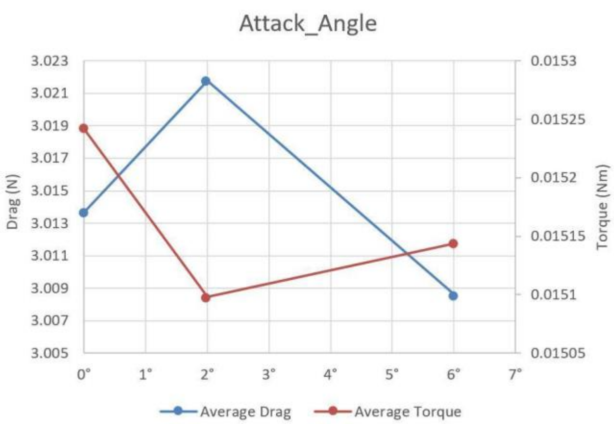

Figure 6: Flow Simulation Results for Attack Angle

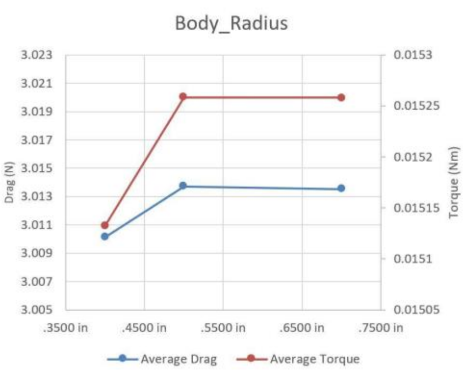

Figure 7: Flow Simulation Results for Body Radius

B. In-Pool Testing

Our pool testing focused on verifying that the new elements of the sub worked correctly, specifically the new frame and housing designs and the new neural network architecture. Our access to our university's pool was extremely limited, allowing us only to test our highest-priority items and preventing us from testing the neural network and competition tasks as carefully as we would have liked.

Mechanically, we initially discovered few issues. After an initial misstep with incorrect ESC settings, the sub appeared to work almost entirely as expected, allowing us to focus on software. Our software testing initially ran smooth, but not long into our pre-qualification preparation, issues with leaks were discovered. The large tube was pushed past the O-rings on its mount by a combination of movement and increased internal pressure from the sun's heat. Fortunately, damage was minimal and we were able to quickly manufacture a solution to stabilize and reinforce the housing to prevent future failures.

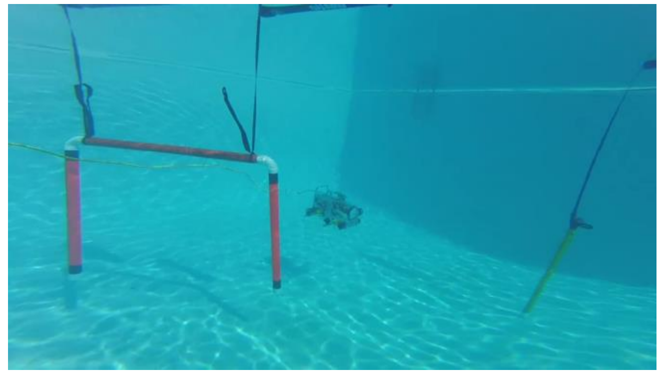

Figure 8: Charybdis with the prequalification gate and pole

Our software testing was largely centered on testing specific competition tasks. We created a model start gate and pole for prequalification, a path marker, buoys, and an octagon for the sub to interact with, as shown in Fig. 8. These objects are what we used to train the neural network and determine that our current hardware is faster than the previous year's. We also found that our first choice of neural network struggles to detect small objects, like the thin PVC pipe we used as a pole for prequalification, forcing us to explore other potential solutions. Due to the issues mentioned above, our prequalification efforts have not yet been successful, but we still hope to prequalify by the video submission deadline.

IV. Acknowledgments

The team would like to thank Dr. Kevin McFall for acting as our faculty advisor, all our engineering and technology professors for their instruction, the KSU Student Activities Board Advisory Committee and the KSU Alumni Association for funding this project, and the Smyrna Parks and Recreation Department for allowing us to use their aquatic facility to test Charybdis.

V. References

[1] A. Graves, A. Hagle, C. McDermitt, C. Meier, E. Neleski, B.Sailor "*Technical Design Report for Via Maris*." RoboNation. Accessed on July 1, 2019, https://robonation.org/sites/default/files/Kennesaw_TDR_RS18.pdf

[2] A. Cheng, H. Evans, J. Gragg, A. Graves, A. Hagle, C. Meier, J. Nguessan, V. Nguyen, B. Sailor. "Design rationale for Leviathan autonomous underwater vehicle." RoboNation. Accessed on July 1, 2018, http://robonation.org/sites/default/files/RS17_KennesawState_Paper.pdf

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C. Berg. "SSD: Single Shot MultiBox Detector." arXiv. Dec. 2016. Accessed on July 1, 2018, https://arxiv.org/pdf/1512.02325.pdf.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv. Apr. 2017. Accessed on July 1, 2018, https://arxiv.org/pdf/1704.04861.

[5] "ROS Melodic Morenia." wiki.ros.org, 2018. Accessed on 8 July, 2019, http://wiki.ros.org/melodic.

[6] “smach - ROS Wiki", wiki.ros.org, 2017. Accessed on 19 June, 2017, http://wiki.ros.org/smach.

[7] "mavros - ROS Wiki", Wiki.ros.org, 2017. Accessed on 19 June, 2017, http://wiki.ros.org/mavros.

[8] Overview ArduSub GitBook". BlueRobotics, 2017. Accessed on 19 June, 2017, https://www.ardusub.com/.

[9] R. Panez. “Simplified method for obtaining navigational information from hydrophone arrays”. University of Florida, 2004. Accessed on 19 June, 2017, https://www.mil.ufl.edu/publications/thes_diss/Rolando_Panez_thesis.pdf

[10] "rosserial - ROS Wiki", Wiki.ros.org, 2017. Accessed on 19 June, 2017, http://wiki.ros.org/rosserial_arduino

Appendix A: Expectations

Subjective Measures			
	Maximum Points	Expected Points	Points Scored
Utility of team website	50	50	

Technical Merit (from journal paper)	150	150	
Written Style (from journal paper)	50	50	
Capability for Autonomous Behavior (static judging)	100	100	
Creativity in System Design (static judging)	100	100	
Team Uniform (static judging)	10	10	
Team Video	50	50	
Pre-Qualifying Video	100	100	
Discretionary points (static judging)	40	40	
Total	650	650	
Performance Measures			
	Maximum Points		
Weight	See Table 1 / Vehicle	38	
Marker/Torpedo over weight or size by <10%	minus 500 / marker	0	
Gate: Pass through	100	100	
Gate: Maintain fixed heading	150	150	
Gate: Coin Flip	300	300	
Gate: Pass through 60% section	200	0	
Gate: Pass through 40% section	400	400	
Gate: Style	+100 (8x max)	100	
Collect Pickup: Crucifix, Garlic	400 / object	0	
Follow the “Path” (2 total)	100 / segment	200	
Slay Vampires: Any, Called	300, 600	600	
Drop Garlic: Open, Closed	700, 1000 / marker (2 + pickup)	0	
Drop Garlic: Move Arm	400	0	
Stake through Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200 / torpedo (max 2)	1600	
Stake through Heart: Move lever	400	0	
Stake through Heart: Bonus - Cover Oval, Sm Heart	500	0	
Expose to Sunlight: Surface in Area	1000	1000	
Expose to Sunlight: Surface with object	400 / object	0	
Expose to Sunlight: Open coffin	400	0	
Expose to Sunlight: Drop Pickup	200 / object (Crucifix only)	0	
Random Pinger first task	500	0	
Random Pinger second task	1500	0	
Inter-vehicle Communication	1000	0	
Finish the mission with T minutes (whole + fractional)	Tx100	800	

Appendix B: Component Specifications

Component	Vendor	Model/Type	Specs	Cost (if new)
Buoyancy Control	N/A			
Frame	KSU AUV	Custom	28in x 36(motors out)in x 18in	$880
Waterproof Housing	Blue Robotics		1 in8 and 4 4in Enclosures	$972

Waterproof Connectors	Blue Robotics	red/black penetrators	N/A	$96
Thrusters	Blue Robotics	T200		N/A
Motor Control		EMAX Dshot 1200	45A ESC	$140
High Level Control	Amazon	Pixhawk 3		N/A
Actuators	N/A			
Propellers	Blue Robotics		N/A	N/A
Battery	HobbyKing	Multistar	10000mAh, 4s	N/A
Converter	N/A			
Regulator	Amazon	KNACRO	AC/DC to DC 20W Converter	$11.20
Embedded System	Nvidia	Jetson Nano	Quad-core ARM Cortex-A57 MPCore processor (1.43 GHz)	$99
Internal Comm Network	N/A	N/A	USB cables	N/A
External Comm Interface	Blue Robotics		Ethernet tether cable	N/A
Programming Language 1	Python			
Programming Language 2	C++			
Compass	Amazon	Pixhawk 3		N/A
Inertial Measurement Unit (IMU)	Amazon	Pixhawk 3		N/A
Doppler Velocity Log (DVL)	N/A			
Camera(s)	Logitech	C930E and C270	C930E: 1080p/30 FPS, 90° FOV C720: 720p /30 FPS, 60° FOV	N/A
Hydrophones	N/A			
Manipulator	N/A			
Algorithms: vision	Tensorflow	SSD MobileNet v2	N/A	$0
Algorithms: acoustics	N/A			
Algorithms: localization and mapping	N/A			
Algorithms: autonomy	KSU AUV	Custom	N/A	$0
Open source software	ArduSub, Ubuntu, ROS, MavROS, OpenCV, SMACH, Tensorflow,			
Team size (number of people)	20			
HW/SW expertise ratio	2 HW : 1 SW			
Testing time: simulation	2 hrs.			
Testing time: in-water	15 hrs.			

Appendix C. Community Outreach

KSU AUV participates in a number of community outreach efforts, both within KSU and in the surrounding area.

Our major outreach event each year is participating in Maker Faire in Atlanta, GA, which is held every October. The team sets up a booth at this event and uses the opportunity to show the community our work, including displaying our submarines and bringing an interactive display for attendees to experience, as shown in Figure 9. In the past we have shown a static motor testing rig, but this year’s project was a miniature remote-controlled submarine built using the same technology as Via Maris and Charybdis. The minisub is an especially good demonstration of our technology because it lets us show off multiple areas of expertise. We demonstrate the mechanical and electrical design needed to build a submarine as well as the software used to control it and process feedback.

We also participate in events at and hosted by KSU, including freshman orientations, events open to all university clubs, and events specific to KSU competition teams. The university hosts an annual competition team showcase, which we attend with the same display as at Maker Faire. We also participate in other events with the competition teams separate from those hosted by the university administration.

Figure 9: The team at Maker Faire Atlanta in October 2018