Technical Design, Strategy, and Implementation of SEAHORSE

Aggie Marine Robotics

Written by Jacie Grow, Reed Thurber, Bryant Casutt, Konnor Andrews and Taylor Jones

Abstract - The Aggie Marine Robotics team developed a simple strategy for this year at the AUVSI Robosub competition, with the three main goals of increasing testing time, being capable of the qualification maneuver, and redefining the team's appearance and structure. We met these goals by developing a series of testing rigs, each one was designed to accomplish these three main goals and to score higher than any previous appearance made by USU at the Robosub competition. Our final rig SEAHORSE is the result of our efforts to design a simple adjustable vehicle with custom designed electronics and thruster controls.

I. Competition Strategy

Utah State University's Autonomous Underwater Vehicle (AUV) competition team has competed in the AUVSI Robosub competition for the past three years as USU Robosub Team. Each year our team has struggled to qualify at competition due to lack of in water testing and malfunctioning components. Following last year's competition, where most of our attendees were first time participants, we decided to restructure the way our team approached the competition. The first decision we made was to rename the team and design a new vehicle from the ground up, this allowed us to recruit many new team members and allowed us to abandon many of the previous practices we had identified in our design process.

Recognizing that our new team was young and inexperienced, we decided to approach the competition with a simple vehicle. We also hoped that by simplifying the design we would be able to limit the possibility of malfunctioning components and allow for easier repairs. Unlike previous years we wanted to focus our design on testing and make sure that the vehicle we took to competition was experienced and proven.

To accomplish these goals of simplicity and testing we decided to implement an iterative design involving a series of "testing rigs" or rather, multiple vehicles that would each be capable of performing the qualifying maneuver. By ensuring that even our most simple testing rig could qualify, we could then make sure that any improvements we made to the next vehicle would not jeopardize our ability to qualify and compete.

Once we had a vehicle capable of qualification, we performed an analysis of the tasks at the competition and determined which of these tasks would be attainable with our team's time, skill, and funding. The proposed competition strategy for Seahorse, our most recent vehicle, is to firstly go through the gate while attempting to perform a rolling movement. Seahorse will then swim toward the bouys in an effort to knock into them with our experimental vision analysis. After which Seahorse will attempt to surface to complete it's run. Additionally the weight and size of Seahorse will qualify our team for bonus points of 80 + (48.5-lb).

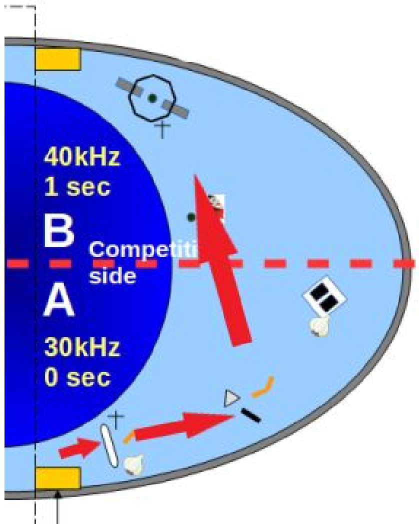

Fig. 1: The course plan for Seahorse (in red). Seahorse's main goals are passing through the gate, hit the buoys, and surface.

The strategy for attempting fewer tasks then we have planned for in the past, is based on our team's past experience at competition. Previously we have struggled to qualify at competition and experienced many problems which eventually rendered us unable to swim during semifinals. After reflection we identified the main issue was that we did not have a clear set of goals. We allocated the mechanical team's effort and funding to developing a torpedo launcher and dropper when our software team was still unable to perform simple movement underwater. Whereas this year we reduced the project's scope to only qualification to ensure that everyone was working towards the same goal. By setting specific attainable goals we are hoping to then use the same submarine at next year's competition and implement a more complicated strategy involving a dropper and torpedo launcher.

An additional strategy that we chose for the 2019 competition is to score the most points possible out of the water. In previous years elements such as the team uniforms and presentation have been afterthoughts to the vehicles design. However, this year with the new team name and improved organization, we have made an effort to create a name and image that will reflect positively on our standing as a university team and as competitors at the Robosub Competition.

II. Vehicle Design

This year our team chose a design approach that was unique to USU's history. We decided to split each portion of the vehicles design (mechanical, electrical, and software) into different stages that could be improved on independently without sacrificing function. We wanted to balance our ability to function reliably with the vehicles potential for innovative changes. We took an object oriented approach where every component we designed could be implemented on any one of our three testing rigs. By choosing simple and standard interfaces we could switch out any component that wasn't working without compromising the rest of the vehicle.

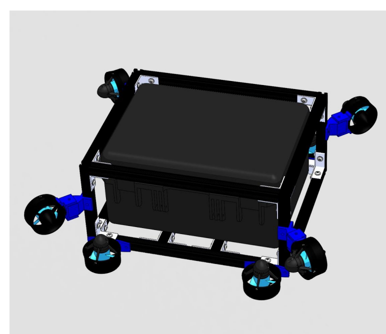

Figure 2: BLOCK I, the first testing rig for the 2018- 2019 year consisting of a large waterproof case with a frame of aluminum extrusions

A. Mechanical

The most unique decision the mechanical team made this year was using a series of testing rigs. Our team chose to split the phases of development into three separate vehicle bodies. Our first vehicle nicknamed T.I.J.A.R was designed to be a waterproof hull with only thrusters attached, this allowed the software and electrical teams to begin testing underwater before the mechanical team had finished fabricating the final vehicle body. We then installed sensors and other attachments as the software and electrical teams became ready to use and test different devices and functions. Our next rig named Seahorse could then be developed by the mechanical team without making the software and electrical teams stop testing and wait for fabrication to be complete.

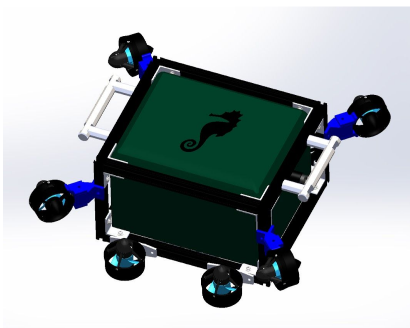

Figure 3: BLOCK III (Seahorse), the latest vehicle for 2019. The body consists of a small waterproof case, aluminum extrusion frame, 3d printed thruster mounts and steel handles.

We used this stage based development technique to also achieve our main two competition goals of ensuring everything we added to the next stage was well tested and did not compromise our ability to qualify. Before new components such as windows or handles were installed, we would first test them in separate waterproof enclosures to ensure that once we added them to our current rig, they wouldn't cause any problems that would impede our software and electrical testing time.

B. Electrical

Our electrical team followed a similar strategy to the mechanical team, with two separate stages of development. Our first stage was to modify our existing electronic system to supply the proper power and communication to thrusters to enable the software team to begin testing. The next stage involved designing custom PCBs for power distribution, motor ESCs, and communication with our sensors. Each of these boards would attach in an edge connector on a single bus to unify the electronics. In addition, each developed board consists of a processing chip that allows for easy integration into the network by using any of several serial communication between other boards. This custom PCB system allowed us to create a very compact overall assembly which is ideal for an AUV application where space is limited.

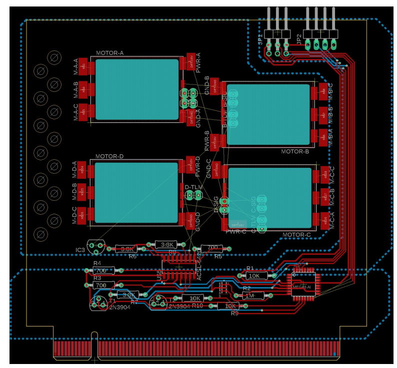

Figure 4: An example schematic of a custom PCB from the first stage of development, designed to be plugged into the edge connectors of the main bus.

This design also allowed us to modify any individual card without changing the entire system. When designing the bus we allowed for extra card slots to enable us to research and develop different boards that would be used on next year's design. Features on our next vehicle such as vision processing and manipulation of an arm or potentially launching torpedoes would all be located on these extra slots. In our final stage of design we sacrificed these research and development dedicated slots in order to make a smaller electronic block which was necessary to fit into the smaller vehicle bodies of the latest testing rigs and final submarine.

C. Software

This year our software team wanted to move towards using open source rather than rolling our system from scratch. We integrated the Robot Operating System (ROS) into our software architecture. ROS allows us to build modular components for different pieces of hardware and software. Then when we want to experiment with different parts of the software we can swap out the individual ROS packages. We architectured our software with multiple layers so we can manually control the system for testing, control semi-autonomously, or run fully autonomous scripts. Using ROS we can write C/C++ nodes on microcontrollers, C++ and python on the CPU, and JavaScript for autonomous scripting and testing web apps.

We built our system to be highly loggable for testing. With ROS we can selectively play back logs to simulate the real environment with different code in the lab. We can replace code after the test and replay the sensors to see how it responds. We built our system primarily on PID control using screw theory to help with path and control planning. Using eight motors we are able to have full control in every direction. Using ROS we can tweak the PID and kinematics control and retest the responses to sensor inputs without returning to the pool.

III. Experimental Results

Our team focused on performing as many in water tests as possible before competition. The goal of every in water test time was to plot and traverse a path through our test gate. Each pool test involved a series of steps to ensure that we could replicate our tests. First, the mechanical team would perform a water test on the vehicle by removing all of the electronics and submerging the vehicle for a predetermined amount of time to ensure there were no leaks. Each dry test was then followed by a remapping of the thrusters ensuring the functionality of each individual motor. Following these tests we would then submerge the vehicle and analyze the readout from our sensors by performing basic translational movement while attempting to stay level, and eventually diving to constant depth. After these preliminary tests concluded we would then attempt to target and swim through the gate taking note of depth and translational errors. Replicating this pattern at every test helped us isolate any issues we had with our system.

For vehicle navigation this year our main problem was determining how to allow an arbitrary thruster layout to be controlled with a traditional control system. The following is our team's solution to this problem.

The resultant force and moment created by all the thruster generated forces F_i $i = 1, ..., n$, acting on the submarine's body is simply the sum of the individual wrenches W_i, that is

$$R = \sum_{i=1}^{n} W_i = \sum_{i=1}^{n} (F_i, P_i \times F_i).$$

Each thruster can be made to have a linear control over its force output. This results in each thruster's acting force to be

$$F_i = \hat{F}_i \cdot k_i,$$

where k_i is the magnitude of the thruster's force. The above equation for R can be rewritten in matrix form to be

$$R = \begin{bmatrix} \hat{F}_1 & \dots & \hat{F}_n \\ P_1 \times \hat{F}_1 & \dots & P_n \times \hat{F}_n \end{bmatrix} \begin{bmatrix} k_1 \\ \vdots \\ k_n \end{bmatrix}$$

The matrix form gives a solvable thruster output state for any given force and moment state. Applying the Moore–Penrose inverse to the above equation gives an almost optimal solution for fully defined thruster layouts. As a result, using this equation allows the rest of our control system to work as if it can generate forces and moments at the center of gravity of the vehicle.

Appendix A: Expectations

Subjective Measures			
	maximum points	Expected points	Points Scored
Utility of team website	50	27	
Technical Merit (from jourrnal paperr)	150	40	
Written Style (from journal paper	50	45	
Capability for Autonomous Behavior (static judging)	100	55	
Creativity in System Design (static judging)	100	50	
Team Uniform (static judging)	10	9	
Team Video	50	48	
Pre-Qualifying video	100	70	
Discretionary points (static judging)	40	20	
Total	650	364	
Performance Measures			
Weight	See Table 1 / Vehicle	91	
Marker/Torpedo over weight or size by <10%	minus 500 / marker	0	
Gate: Pass through	100	100	
Gate: Maintain fixed heading	150	140	
Gate: Coin Flip	300	0	
Gate: Pass through 60% section	200	200	
Gate: Pass through 40% section	400	0	
Gate: Style	100 (8x max)	200	
Collect Picup: Crucifix, Garlic	400 / object	0	
Follow the "Path" (2 total)	100 / segment	0	
Slay Vampires: Any, Called	300, 600	0	
Drop Garlic: Open, Closed	700, 1000 / marker	0	
Drop GarlicL Move Arm	400	0	
Stake through the Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200 / torpedo	0	
Stake through the Heart: Move lever	400	0	
Stake through the 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 + factional)	Tx100	0	
Total		1731	

Appendix B: Component Specifications

Component	Vendor	Model/Type	Specs	Cost(if new)
Bouyancy control	N/A			
Frame	80/20	Aluminum T-Slotted Profile	1" x 1"	$100
Water Proof Housing	Seahorse	SE-540 Waterproof Protective Hard Case Without Foam	Exterior Dimensions: 15.27" x 12.13" x 9.58" - Interior Dimensions: 13.50" x 9.80" x 8.37". Case weight: 6.47 lbs.	$35.99
Water proof connectors	Hilitchi	Nylon Plastic Waterproof Adjustable Cable Glands Joints	9.6 ounces, PG-9	$10.39
Thrusters	BlueRobotics	T200	7.8 lb f (forward), 6.6 lb f (backwards), 6-20 volts	
Motor Control	BlueRobotics	ESC Revision 1		
High Level Control	Jetson	TX2	NVIDIA Pascal™ Architecture GPU. 2 Denver 64-bit CPUs + Quad-Core A57 Complex.	
Battery				
Converter	N/A			
Regulator	N/A			
CPU	Jetson	TX2		
Internal Comm Network	ROS			
External Comm Interface	UART			
Programming Language 1	Python			
Programming Language 2	Javascript			
Programming Language 3	C++			
Compass	N/A			
IMU	Sparkfun 9DoF Razor			
DVL	N/A			
Cameras	N/A			
Hydrophones	N/A			
Manipulator	N/A			
Algorithms: vision	N/A			
Algorithms: acoustics	N/A			
Agorithms: localization and mapping	N/A			
Algorithms: autonomy	N/A			
open source software	ROS, Python, C++, Javascript,			
Team size	21			
HW/SW expertise ratio	18:3			
Testing time: simulation	N/A			
Testing time: in-water	20			