2020 Colorado RoboSub Overview of Physical and Virtual Systems for the *Leviathan* and *Papa* Autonomous Underwater Vehicles

Luke Barbier, Adam Chehadi, Daniel Fairbanks, Michael Marques, Ryan Oroke, Izzy Russo, Phillip Wu, William Wang, Zachary Faith, Soroush Khadem, Luke Morrissey, Chase Williams, Atharv Desai, and Luke Ethan Schwab

1. Abstract - ***At the 2019 RoboNation RoboSub competition Colorado RoboSub successfully completed numerous tasks beyond the start gate and therefore, for the 2020 competition season, our team focused on improving our ability to reliably complete both previously successful and new tasks in the TRANSDEC. For the purpose of completing these objectives our team has developed a more robust software stack capable of handling the increased complexity of our runs as well as introduced a second competition vehicle that will enable quick run times and increase our team's physical capabilities.***

2. Competition Strategy

Our team has closed gaps in our competition strategy and submarine design for several years. We have gone from ranking 35th, to 19th, 11th, and last year we ranked 7th place. In order to continue our momentum, and reach our goal of being a top three team this season, we have evaluated key improvements in our strategy and submarine design. Last season, our submarine had the ability to complete the start gate, path, buoys, droppers, and octagon tasks. In order to maximize our score, we have made design improvements to decrease task failure rate, and have designed a secondary submarine to achieve inter-sub communication and cooperation.

During the final competition run, our submarine timed out while attempting droppers. This happened because the dropper game piece exited our submarine's field of view for too long. In response to this shortcoming, we have vastly improved our perception pipeline to decrease object detection time. In addition to this, we have developed a new memory system that allows the submarine to maintain a record of each object bearing relative to the submarine's current time and position. This allows us to retrace portions of the competition if a task game piece exits the camera frame.

For the first time, our team has developed a secondary submarine to accompany our primary submarine during competition. Having a secondary submarine will increase our score by at least 1000 points, and we will earn additional points from time saved completing the competition course.

Although COVID-19 has hindered development of our two competition vehicles, we have continued progress towards completing our season goals. Software development was able to continue with simulator testing. The electrical and mechanical development was handicapped by the closing of our university and machine shops as March-August are generally our team's procurement and integration period, activities virtually impossible to complete while still following local disease prevention guidelines.

Fig. 1: A screengrab from our high-fidelity graphics simulator. In the simulation, Leviathan approaches the torpedo task.

3. Vehicle Design

A. Mechanical Sub-Team

This season the mechanical sub-team for Colorado RoboSub has been focused on expanding our underwater capabilities through the addition of new physical abilities and improving our existing sub-systems.

Despite efforts to continue development during pandemic closures, there were many plans in motion for the procurement, integration, and development of subsystems which had to be halted due to lack of resources and disease prevention guidelines.

The most significant development this season has been the creation of a brand new competition vehicle, Papa, which will allow us to complete more tasks in less time. Significantly different from our other submarine designs, Papa uses a PVC (Polyvinyl Chloride) to take advantage of plastic's anti-corrosive properties resulting in less maintenance and with the benefit of easier modification potential. Because Papa was in construction during the onset of the pandemic, physical construction and testing has not progressed much from the image shown below, although software continues to actively develop for Papa in our simulator.

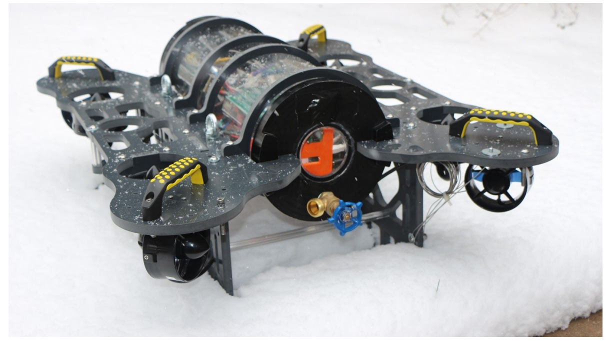

Fig 2: Our newest vehicle Papa rests in the snow shortly after final integration of the new frame.

One major issue encountered last year was Leviathan's weight, which has been addressed by offloading our pneumatics box and cylinder to Papa. This will better distribute the weight across our submarines, as well as increase the possibilities for future manipulator and hydrophone enclosures within Leviathan's frame perimeter.

In correcting our other issues from the previous year, the mechanical sub team also designed solenoid dropper replacements for our previously pneumatic droppers setup. This will eliminate any snagging, rupturing, or unplugging of pneumatics tubing, issues which plagued our droppers and torpedoes last year.

Another change for Leviathan is an upgraded stereo vision end cap, which will allow for the submarine to measure distances to the obstacles it sees. Unfortunately, we were preparing to order the machined parts for the new stereo vision end cap just as the coronavirus shut down our school's machine shop, and therein we will be procuring the parts for the sub-system as soon as the University of Colorado Boulder reopens its machine shops.

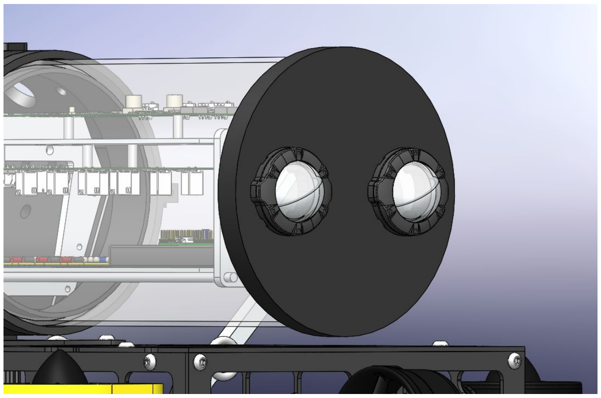

Fig 3: A rendering of Leviathan's stereo vision endcap upgrade.

Most significant for the competition divers, this year we designed new kill switches for our vehicles. By utilizing an intuitive "*green is on, red if off*" color coordination system, our students and divers will be able to more reliably, quickly, and safely disable our vehicle.

B. Electrical Sub-Team

In previous seasons, our MSP432 microcontroller offered more capabilities than we could make use of. Its hardware abstraction layer (HAL) was not as user friendly as we would have hoped for in some areas. As such we redesigned our embedded system to use an EFM32 Pearl Gecko. This allowed us to shrink our design by about 1.75 square inches without removing any of the functionality that we were utilizing. Its improved HAL also allowed us to more easily develop code for the embedded system. We also added this embedded design to our secondary submarine, implementing similar functionality.

We redesigned our 48V solution from the ground up. In previous years, we utilized two regulators to meet current demands; this year we designed around a single regulator that is able to meet all of the power requirements. Last year, we found our off the shelf 12V regulator to be unreliable. As such, we modified our 19V regulator design to run at 12V. We also replaced our 9V switching regulator to a much simpler and smaller load drop-out (LDO) regulator.

As we have added more components to the submarine, we have tried to see if we could run them on the unregulated battery voltage, and as such have added connectors straight to the battery voltage. Last year we attempted to use headers to connect our battery merge board and our power and embedded board. While this provided a secure connection, it made it difficult to separate the boards. A wired connection between the boards solved this issue.

In our previous design we found that running all of our MOSFETs with a single gate driver was damaging to the MOSFETs as the transition between states took too long. We have added a gate driver to every MOSFET to fix this.

C. Software Sub-Team

This season we have added Template Matching to our submarines, a new method of image classification / localization for the torpedo task. We use OpenCV to find keypoints in a reference image, and in a camera image. Those keypoints are matched into sets. Using the real world equivalent coordinates of the keypoints in the image we SolvePnP to get a pose estimate in the camera frame. That pose estimate is converted to the body frame and used to compute the ideal position of the submarine to take a torpedo shot. This is done by projecting a vector from the position of the torpedo tube along its axis to the plane formed by the pose solved target. The submarine then maneuvers to align itself with the plane at an appropriate distance. The system is also able to consider the relative offsets of the camera and the torpedo tube due to the known transform. This allows it to fire the torpedo without actually being able to see the hole in the target. Any part of the larger target that can be template matched against can be used.

This season we also developed a Z-Plane localizer which uses the fact that we know the depth of the target we are looking at. That plus our submarine's orientation and known depth allow us to project a vector to the point the object is detecting in the downward facing camera frame until that vector reaches the known world frame depth of the object. This gives us a 3D body frame position to use to maneuver the submarine. For octagon and droppers we can use this position to center and complete the task.

Another piece of useful information we had not been previously tracking was the bearing to detect obstacles. By storing the position of the submarine and vector to an object we can make better use of that information later. We also try to associate detections such that we can have multiple of a particular object class, such as startgate poles. We do this by seeing how close the previous bearings intersect with the new one to see if a new detection is a new obstacle or more information about an old one of the same class. Since we have a DVL we have very accurate knowledge of short term motion, which combines well with our 360° camera to get bearings to obstacles in every direction as we navigate the course. We can then intersect all those bearings to find a world space position to effectively map the object. We are making use of this for the start gate where pole detections may span across multiple camera frames. This allows us to consider all 3 poles when making decisions on how to maneuver the submarine.

An important addition to our state machine was the ability to retry a task. In a competition run where the clock is ticking it's good to be able to fail out quickly and move on if something goes wrong. We restructured our state machine to increase configurability of how long to spend on any part of a specific task as well as to return to that task in the state machine after skipping it and performing another task. This also includes more sophisticated searching. The stored bearing information helps us locate previously seen items involved in the new task so we can find it quickly. A variety of search patterns are also available if we cannot locate the objects where we expect them.

To accelerate our vision pipeline we nodeletted our CV algorithms in ROS. Previously we had used Python for ease of development but some of our algorithms could take seconds to process an entire frame, and sending image messages over TCP posed considerable overhead. By converting the code to C++ and Nodeliting so data could be passed directly in memory via pointers we significantly increased data throughput. This allows us to classify more frames of video and get more object detections to average and use as control feedback.

As previously mentioned, a second “mini” submarine is currently in development and a new DVL is cost prohibitive so a new accurate means of odometry is required. A considerably cheaper sensor is a camera, and in the TRANSDEC environment the bottom is textured with unique growths and visibility good enough to have accurate tracking. By using a stereo pair of cameras we can pull in an off the shelf Visual-Odometry algorithm into ROS to stand in for our DVL.

As an alternative to more stereo pairs as well as a fallback should we ever be put in a low visibility scenario we have added a BlueRobotics ping to our submarine. It has a 30 degree wide FOV and outputs the return of signal strength at different distances. We have simulated this with raytracing in Gazebo. We use a simulated gain pattern to weight the various rays in the beam to generate a return signal strength approximating what we would see from the pinger. We will use this data to help develop the algorithms in simulation that will be useful once we have the real hardware in the water.

We’ve started investigating using Unreal Engine alongside Gazebo for improved visuals. Unreal provides many benefits such as being able to directly import our SolidWorks models. The built in shaders and physics based lighting engine also provide excellent visuals. The ultimate goal is to be able to use it to train our YOLO neural network on simulated imagery before competition and have it perform well on real imagery.

4. Experimental Results

A significant portion of this year’s RoboSub season was completed virtually due to COVID-19. The University of Colorado at Boulder shut down campus, and access to our university shop space and rec center pool were fully restricted. In past seasons, our team completed real-world pool testing on a weekly basis during the semester and summer, but we were unable to continue our traditional, in-person testing routine this season.

Software testing was completed using both our Gazebo simulator and new Unreal Engine simulator. The additions of our template matching, Z-place localizer, and maintaining obstacle bearing yielded positive results over last year’s performance. Our submarines are completing tasks with greater accuracy. Obstacle task time-outs happen far less due to the submarine’s object bearing memory, and ability to return to a position where obstacles are visible. In addition to this, the improved ROS nodeletting, and code transition from Python to C++ made a noticeable difference in reducing object detection time.

One of the lessons our team learned during the testing phase is to have strong in-person and virtual testing options available. Whether the world is experiencing a pandemic or not, having multiple testing options provides greater flexibility and testing opportunities, which produces better submarines. Our team is eager to return to in-person testing, and we are additionally continuing development on our simulator.

5. Acknowledgements

Colorado RoboSub’s development would not be possible without the support from the University of Colorado at Boulder’s College of Engineering & Applied Science, and the Computer Science Department. Our team would like to acknowledge the support of the EEF (Engineering Excellence Fund) Committee, and the COHRINT lab for the funding they have provided. In addition to this, our team would like to thank the ITLL (Integrated Teaching & Learning Laboratory) for their machining work and collaborative workspaces.

Appendix A: Component Specifications

Leviathan				
Component	**Vendor**	**Model/Type**	**Specs**	**Cost (If New)**
Buoyancy Control	N/A			$0.00
Frame	McMaster Carr, Front Range Cutting	Clear Anodized Aluminum	Aluminum Water Jet Sheet, Aluminum Machined 1/2" Square Rod	$1,000.00
Waterproof Housing	Custom	Custom Hull	Clear Anodized Aluminum Midcap, 2X Clear Anodized Aluminum Endcaos with 1/8" Acrylic Tubing	
Waterproof Connectors	Blue Robotics	Penetrators		
Thrusters	Blue Robotics	T200	5 kg f	$1,432.00
Motor Control	Pololu	Mini Maestro	12 Channel USB PWM Controller	$30.00
High level Control				
Actuators	Blue Trail	SER-110X	Waterproof Servo	$380.00
Propellors	Blue Robotics	T200 Propeller	3 Blade	$36.00
Battery	Turnigy	Multistar 4S 10.0AHr	4S 10AHr	$62.00
Converter	Custom			$399.19
Regulator	Custom		3.3/5/9/12/48V	$418.58
CPU	Intel	I7 7567	3.5GHz	$456.00
Internal Comm Network	NA	Ethernet / UART / I2C		
External Comm Network	Bluerobotics / NA / NA	Fathom / UART / Ethernet		
Programming Language 1		Python		
Programming Language 2		C++		
Compass				
Inertial Measurement Unit (IMU)	Sparton	AHRS-8		$1,350.00
Doppler Velocity Log (DVL)	Nortek	DVL 1MHz		$20,000.00
Camera(s)		Occam / Blackfly S	360 Degree	$1,500.00
Hydrophones	Custom			
Manipulator	Blue Robotics	Newton Subsea Griper	28lbf grip force	$329.00
Algorithms: Vision		YOLO / TemplateMatching		Free
Algorithms Acoustics				
Algorithms: Localization and Mapping				
Algorithms: Autonomy		SMACH		Free
Open Source Software		ROS		Free
Team Size (# of People)	15			
HW/SW Expertise Ratio	1			
Testing Time: Simulation	350			
Testing Time: In-Water	40			

Papa				
Component	Vendor	Model/Type	Specs	Cost (If New)
Buoyancy Control	N/A			$0.00
Frame	McMaster Carr, NoCo Waterjet	PVC Sheet	Waterjet 1/2" PVC Sheet	$700.00
Waterproof Housing	Custom	Custom Hull	Machined Alumin Endcaps, 1/8" Clear Acrylic Tubing	$1,000.00
Waterproof Connectors	Blue Robotics	Penetrators		
Thrusters	Blue Robotics	T200	5 kg f	$1,432.00
Motor Control	Pololu	Mini Maestro	12 Channel USB PWM Contr	$30.00
High level Control				
Actuators	Blue Trail	SER-110X	Waterproof Servo	$380.00
Propellors	Blue Robotics	T200 Propeller	3 Blade	$36.00
Battery	Turnigy	Multistar 4S 10.0AHr	4S 10AHr	$62.00
Converter	Custom			$399.19
Regulator	Custom		3.3/5/9/12/48V	$418.58
CPU	Jetson	Nano	Developer Kit	$100.00
Internal Comm Network	NA	Ethernet / UART / I2C		
External Comm Network	Bluerobotics / NA / NA	Fathom / UART / Ethernet		
Programming Language 1		Python		
Programming Language 2		C++		
Compass				
Inertial Measurement Unit (IMU)	Sparton	AHRS-8		$1,350.00
Doppler Velocity Log (DVL)	N/A			
Camera(s)	Amazon	Pi Camera		$75.00
Hydrophones	N/A			
Manipulator	N/A			
Algorithms: Vision		YOLO / TemplateMatching		Free
Algorithms Acoustics				
Algorithms: Localization and Mapping				
Algorithms: Autonomy		SMACH		Free
Open Source Software		ROS		Free
Team Size (# of People)	15			
HW/SW Expertise Ratio	1			
Testing Time: Simulation	350			
Testing Time: In-Water	40			

NB: Due to the COVID-19 Pandemic, development of Papa was significantly hindered as Leviathan recieved the most focus since the vehicle was already fully constructed. Some of Papa's fields have been left blank since a decision on whether or not to replace previous hardware has yet to be made.