Triton Robosub Technical Design Report 2019: *Ra*
University of California, San Diego

Ezequiel Ansaldi, Christopher Delgado, Donovan Drews, Steven Gov, Johnathan Huynh, Richard Lai, Patrick Paxson, Kevin Resler, Alex Rivera, Dan Truong, Phillip Vally, Peter Xu

Abstract - The Triton Robosub team, representing the University of California, San Diego, is competing in Robosub for the first time in 2019. In this paper, we will discuss our competition strategy, vehicle design, testing strategies and results, acknowledgements of our supporters, and references. This information will give an overview capabilities of our robotic submarine and how we overcame issues related to hardware and software aspects.

I. Competition Strategy

Our newly formed organization was keen to compete in its first tournament. Due primarily to a lack of experience in the field of underwater robotics, we chose to buy the BlueROV2 stock model rather than build a robot from scratch, and use it as a starting point for our autonomous control and custom design choices.

The competition provided many objectives that we would have to complete. To ensure that we obtain the maximum points possible with our time, hardware, and experience limitations, we narrowed down the objective and decided on focusing on passing through the gate and identifying images on the buoys. By setting these goals, we had a clear picture of the additional components and code that would be required. Since this was our first year, we did not want to attempt the other challenges because we wanted to focus on and guarantee success in the initial tasks.

Our main goal for this year was to establish our team and learn as much as we can, which we believe we succeeded at.

II. Vehicle Design

i. Hardware

The BlueROV2 acted as the base of our vehicle and was modified for the purposes of the competition. This stock model provided us with a frame, motors, waterproof enclosures, and computer/control electronics. We chose it as the basis for our vehicle this year rather than constructing one from scratch because this is our first year competing in Robosub. By modifying a robot that has already gone through the design challenges of buoyancy, freedom of movement, and waterproofing, we are able to get a jumpstart on tackling the competition challenges, and get comfortable with underwater robotics before proceeding with designing our own vehicle.

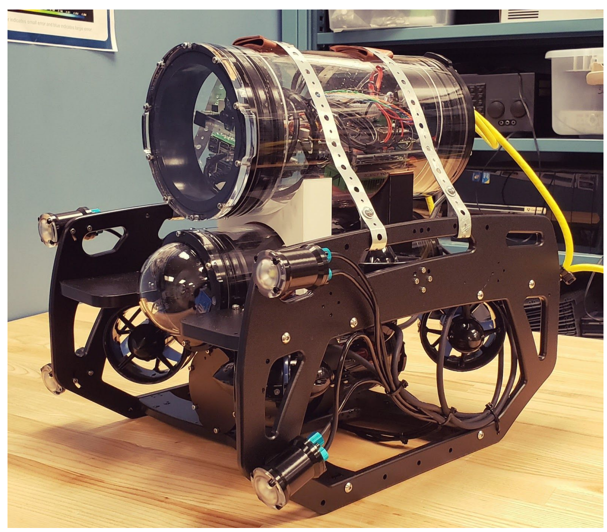

Figure 1: *Side view of the full robot. The Blue Robotics frame can be seen below the custom chamber.*

The frame is composed of seven pieces of black HDPE (high-density polyethylene) and two aluminum enclosure cradles secured together. This structure fits the enclosures, motors, fairings, and subsea buoyancy foam in fixed positions, while allowing flexibility with the mounting of 200g lead ballast weights and four lumen subsea lights. This flexibility increases adaptability, as the lights can be positioned to their optimal angles depending on use case and numerous weights can be added at various positions to stabilize the robot and reach neutral buoyancy in various water conditions.

There are six Blue Robotics T200 thrusters used to move the vehicle through the water. Four of them are positioned at 45 degree angles in each corner of the frame in order to get more precise movement in the xy-planes, and in order to move up and down in the water, the last two motors are positioned in the center of the robot facing up in the z-axis. One design challenge faced as a result of our custom third enclosure (discussed below) is the effectiveness of the motors. In order to combat the changed weight distribution introduced by the enclosure, we added lead weights and decreased the speed of the motors for autonomous navigation.

Two enclosures were provided in the stock model specifically for the battery and electronics. These are 3” and 4” acrylic tubes, respectively, enclosed with aluminum end caps and secured directly to the frame with aluminum enclosure clamps. These end caps are fitted with cable penetrators nuts and a vent plug. The former is fitted with O-rings and the cables are secured with epoxy to allow cables to be run into the enclosures while keeping the enclosure waterproof. The latter is similar, but functions to vent pressure within the tubes as well as to pressure test the enclosure. The electronics enclosure also includes a dome end cap to account for a forward facing camera.

The primary electronics in this stock robot include: Pixhawk Autopilot, Raspberry Pi 3 Model B Computer, Fathom-X Tether Interface Board, Low-Light HD USB Camera, and Basic Electronic Speed Controllers. These electronics enable us to pilot the robot using Blue Robotics software called QGroundControl, a wired tether to another Fathom-X Tether Interface Board on the surface, and a game controller. This tether interface was very useful for driving the robot for testing, but since the competition requires the robot to be autonomous, it presented a challenge in software, discussed further in II.ii.

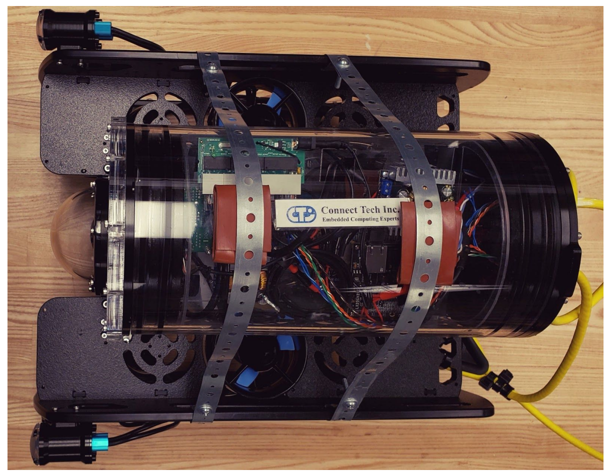

Figure 2: *Top view of the completed robot. The top of the robot is the custom chamber, containing our TX2, Ethernet switch, Raspberry Pis, and more.*

The creative aspect of our physical robot was the positioning of the largest container. The two options that were contemplated was the bottom and top of the BlueROV2. The bottom orientation would require that we build a frame to hold the enclosure so that it would be able to sit on a flat surface. Ultimately, we chose to place the enclosure on top of the sub. Using 3D-printed parts, the top enclosure rests on the middle one and is latched to the frame. Part of the motivation for choosing the top was because the large container will provide a large buoyant force upwards due to its large diameter. This would prevent the resulting buoyant force from flipping our sub upside down.

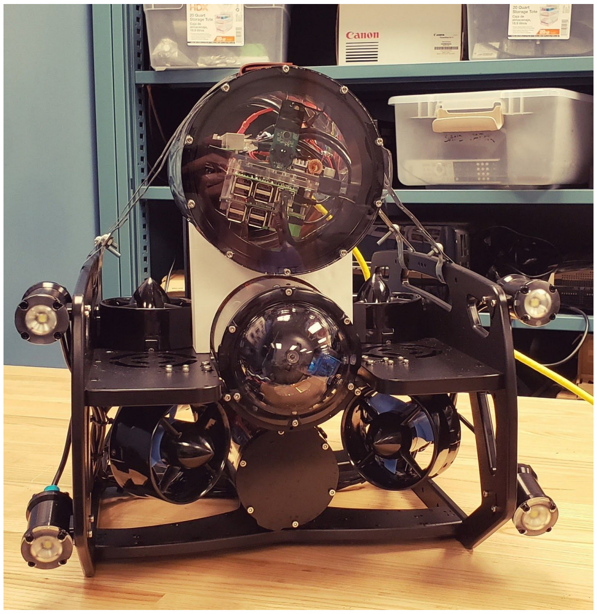

Figure 3: *Front view of the sub. The swivel camera in the middle enclosure and the Logitech C270 in the top enclosure can be seen here, which are used for computer vision.*

A 4S 14.8V lithium-ion battery is used to power our AUV. We chose the custom-made battery pack from BlueRobotics because it fit the specifications of the components in the BlueROV2 and it perfectly fit in the BlueROV2 3’’ battery enclosure. The top enclosure houses two Raspberry Pis, a Nvidia Jetson TX2, an Ethernet switch, a USB camera, buck converters, and a reed switch. To power these components, a power cable is connected to a terminal block in the middle enclosure and fed to the outside through a cable penetrator and into the third container through another cable penetrator. A 16-gauge wire used for the power cable to ensure that enough amps can be safely used. 14.8V is transferred through this power cable and then split to power each individual component in the top enclosure. To power each component, a voltage converter (Buck converter) is needed to convert the 14.8V to each component’s respective operating voltage.

A reed switch is used as the kill-switch. Using a Raspberry Pi, a reed switch is attached to the inside surface of the top enclosure [4]. A magnet is attached to the outside surface of the top enclosure to “arm” the kill-switch. Once the magnet is pulled, the reed switch will send a signal to turn off the thrusters.

ii. Software

Ra’s movement is derived from the ArduPilot project. The ArduSub autopilot software on the BlueROV2 provides robot mobility through controllers. In order to utilize this provided autopilot for vehicle autonomy, the MAVLink protocol is used to communicate between onboard components to operate the motors. ArduPilot software provided Ra with the ability to move in all axes, and necessary functions were accessed through the use of pymavlink, a Python implementation of the MAVLink protocol. Once the basic mobility functions were established for Ra, these needed to be incorporated with the state machine for Ra’s autonomy.

The decision-making aspect of the robot utilized a state machine, written in Python, that took inputs from the other systems in the robot like computer vision and the IMUs in the PixHawk, and outputted the instructions to motor control as well as the next state. This model allowed us to have a breadth of complex decision-making in relatively simple terms, since each state only needed to check one or two things. It was also quite easy to integrate using the network interfacing tool ZMQ, which allowed us to send and receive signals to and from other boards.

Computer vision was the most complicated algorithm in our software hierarchy. We initially attempted to use a static color detection algorithm in order to find the orange markers on the gate and comparison with bottom angle markers in the pool, but after experimentation and examining competition footage from last year, we determined that it would not be robust enough to successfully detect orange in a pool as murky and dark as the TRANSDEC pool. So, we turned to machine learning and Tensorflow [1] in order to detect objects underwater like the gate. We used the Object Detection API [2] as the basis of our training, and since the Nvidia TX2 was a GPU that allowed us to use CUDA, we optimized the neural network generated by our training with TensorRT [3]. Essentially, this allowed us to run the program on the TX2 across GPU cores instead of CPU cores, vastly increasing our speed, which was crucial for real-time processing. We trained it on a vast set of data gathered from our team members swimming through our prop gate with a GoPro, hand-labeling the gate in thousands of images. This grueling task yielded a very large training set that we used to train our network.

Overall, these individual components worked together to turn our originally manually-driven stock BlueROV2 into a fully autonomous sub capable of underwater object detection and autonomous movement. Given this foundation, we are excited to apply our newfound knowledge to next year's competition and make our software even better.

III. Experimental Results

After construction of the robot was completed, we moved on to pressure tests and water tests. The initial pressure tests we performed resulted in failure as there were cracks in the container that would have led water into the electronics container. Our second set of pressure tests also resulted in failure but we were unable to identify the source of the leak on land. We resolved this issue by doing a water test, because we suspected that our vacuum pump may have been the source. From this we found that water was leaking from a defective motor wire. After replacing the motor, our robot was waterproof and we moved to autonomous control tests.

The first autonomous control tests yielded some disastrous results because some of the motors needed to be electronically reversed for the robot to move normally. After reversing the motors, we were able to control the robot's movement in all planes, but faced an issue with our third chamber affecting our movement and buoyancy. We made several adjustments to the chamber in order to properly balance it, and attached lead weights to combat the buoyancy issue. We also lowered the throttle of our autonomous movements so that the extra frontal surface area of the chamber did not significantly affect our performance.

To test the computer vision we set up a test course in a pool to simulate the competition setting. We put the robot into the pool, and essentially simulated the competition, gathering some information from debugging tools we built in as well as the unused swivel camera that came with the BlueROV2.

We are excited to make even more improvements through experimentation in the coming weeks, and look forward to measuring our performance at the competition!

IV. Acknowledgements

Triton Robosub greatly appreciated the support of our faculty advisors: Ryan Kastner in the CSE department and Curt Schurgers in the ECE department. Their mentorship aided our decision making and kept our focus on getting results. We would also like to thank Eric Lo and Nathan for their help and advice throughout the year. We would not have a robot without them. A special thanks to the Kastner Research Group for providing the base BlueROV2 and a cutting edge laboratory space where we could build and test Ra.

We would also like to thank the Scripps Institute of Oceanography and the Canyonview Athletic Center for allowing us to perform tests in various pools on campus. The data gathered from these tests have proved invaluable to us.

Financial support in our first year was tremendous. Thanks to the generous support of UCSD benefactors like the Triton Research & Experiential Learning Scholars organizers, the Jacobs School of Engineering IDEA Center, as well as the support of local companies like Northrop Grumman, ConnectTech Inc, Qualcomm, and BrainCorp Inc, we were able to fund our robot's construction, as well as various competition fees. We look forward to using more of our donations and sponsorships in the future for even better innovative design.

V. References

[1] Tensorflow, "tensorflow/models," *GitHub*. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection. [Accessed: 09-Jul-2019].

[2] "Introduction and Use - Tensorflow Object Detection API Tutorial," *Python Programming Tutorials*. [Online]. Available: https://pythonprogramming.net/introduction-use-tensorflow-object-detection-api-tutorial/. [Accessed: 02-Jul-2019].

[3] "Deep Learning SDK Documentation," *NVIDIA Developer Documentation*. [Online]. Available: https://docs.nvidia.com/deeplearning/sdk/tensorrt-developer-guide/index.html. [Accessed: 08-Jul-2019].

[4] "GPIO," *GPIO - Raspberry Pi Documentation*. [Online]. Available: https://www.raspberrypi.org/documentation/usage/gpio/. [Accessed: 09-Jul-2019].

[5] "BlueROV2 Assembly," *Blue Robotics.* [Online]. Available: https://bluerobotics.com/learn/bluerov2-assembly/. [Accessed: 09-Jul-2019].

Appendix A: Expectations

Subjective Measures			
Task	**Maximum Points**	**Expected Points**	**Points Scored**
Utility of Team Website	50	40	
Technical Merit (from Technical Design Report)	150	150	
Written Style (from Technical Design Report)	50	50	
Capability for Autonomous Behavior (Static Judging)	100	50	
Creativity in System Design (Static Judging)	100	70	
Team Uniform (Static Judging)	10	10	
Team Video	50	50	
Pre-Qualifying Video	100	0	
Discretionary Points (Static Judging)	40	40	
Total	650	460	
Performance Measures			
Task	**Maximum Points**	**Expected Points**	**Points Scored**
Weight	See Table 1	90	
Marker/Torpedo over weight or size by <10%	Minus 500 per 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 per object	0	
Follow the "Path" (2 total)	100 per segment	0	
Slay Vampires: Any, Called	300, 600	0	
Drop Garlic: Open, Closed	700, 1000 per marker (2+pickup)	0	
Drop Garlic: Move Arm	400	0	
Stake Through Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200 per torpedo (2 max)	0	
Stake Through Heart: Move Lever	400	0	
Stake Through Heart Bonus: Cover Oval, Sm Heart	500	0	
Expose to Sunlight: Surface in Area	1000	0	
Expose to Sunlight: Surface with Object	400 per object	0	
Expose to Sunlight: Open Coffin	400	0	
Expose to Sunlight: Drop Pickup	200 per 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)	$T \times 1000$	0	
Total		1140	

Appendix B: Component Specifications

I. Hardware

Component	Vendor	Model/Type	Specs	Cost
BlueROV2	Blue Robotics	Stock	- 457 x 338 x 254 mm - 10-11 kg	$3,663.00
Enclosure with Endcaps	Blue Robotics	Watertight Enclosure	- 6 in. diameter - 1 Acrylic Endcap - 1 Aluminum Endcap w/ 5 holes	$494.19
Thrusters	Blue Robotics	T200	- 3800 rev/min max	Included in BlueROV2, $169.00 ea
Motor Control	Blue Robotics	Basic ESC	- 30 A Max, 7-26 V - 400 Hz	Included in BlueROV2, $25.00 ea
Camera	Logitech	C270		$21.40
Camera	Blue Robotics	Low-Light HD	-Installed on a swivel	Included in BlueROV2, $118.00
Battery (x2)	Blue Robotics	Lithium Ion	- 14.8V, 18Ah	$501.61 ($250 ea)
Buck Converter	Amazon		- 8A 5-40V to 1.2-36V - 5A 4-38V to 1.25-36V	$29.13
Reed Switch	Amazon		- Glass Length:14mm, - Glass Diameter:2mm - Total Length: 45mm	$5.58

Penetrator	Blue Robotics	M10	-4-5mm cable width -8mm cable width -Blanks -Vent plug	$4.00 ea
Tether Cable	Blue Robotics	Fathom ROV tether	-100m length	Included in BlueROV2, $500.00
Vision Board	NVIDIA	Jetson TX2	- 87 x 50 mm - 8 GB 128-bit LPDDR4 - NVIDIA Pascal™ architecture with 256 NVIDIA CUDA cores - Dual-core Denver 2 64-bit CPU and quad-core ARM A57 complex	$479.00
Vision Daughter Board	ConnectTech	Orbitty	- 87x50mm - 1x GbE, USB 3.0, USB 2.0, 1x HDMI, 1x MicroSD, 2x 3.3V UART, I2C, 4x GPIO - +9V to +14V DC Nominal (+19V Peak)	$174.00
Motor Control Board	Raspberry Pi (through Blue Robotics)	Model 3B	- 5V/2.5A DC power input - 1GB LPDDR2 SDRAM - Broadcom	Included in BlueROV2, $35.00
State Machine Board	Raspberry Pi	Model 3B		$35.00

			BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz	

II. Software

Control Unit	**Language(s)**	**Tools/Libraries**
Computer Vision	Python	LabelImg Tensorflow Object Detection API
State Machine	Python	ZMQ
Motor Control	Python	ArduPilot

III. Team Information

Team Size	12 people
HW/SW expertise ratio	4:7 (4 hardware, 7 software, 1 social media manager)
Testing time: simulation	80 hours
Testing time: in-water	40 hours