UMBC Robosub - Technical Design Report

2019 International Robosub Competition

Connor Strang, Brandon Guerrero, Adam Grosse, Kevin Wegner,
Igor Savchenko, Logan Courtright, Daniel Lee

1. Abstract

UMBC Robosub is proud to be participating in the International Robosub Competition for the second time. Last year, we fielded a sub built over many semesters by Mechanical Engineering and Computer Engineering capstones. While we recognize and appreciate their contribution, the isolation between the various groups led to a disjointed result. This year, we've integrated requirements from the point of view of hardware, software, and electrical design to develop a set of design guidelines that we hope will drastically increase the stability and autonomous capability of this year's entry.

2. Competition Strategy

Compared to last year, UMBC supported more student organizations, and as a result, the funds granted to any single organization (including UMBC Robosub) decreased somewhat. We implemented a crowdfunding campaign this year and have received generous donations, however our budget still had to be reduced from last year's total.

Between our budget constraint, limited member count, and even more limited availability for the majority of members we decided to focus on building a cost-efficient frame and hull that can be improved or replaced with minimal waste. Similarly, our hardware and software designs emphasize completing the first few tasks well and providing a base for next year's team to work on developing solutions to the more complex tasks. We will also be creating manuals and justifications to help future members understand our choices of components and techniques in order to ease knowledge transfer.

The tasks we will confront this year include: Journey to the Undead Realm, Enter the Undead Realm, Pickup Garlic, Path, Slay Vampires, and Drop Garlic. These tasks mainly test the computer vision and motion systems, with some very simple active or even passive manipulators for picking up and dropping garlic. The other two main tasks are located merely by pingers and would be challenging to locate with a vision-only solution. In addition, they require more complex manipulators. Based on the requirements of the selected tasks and the manufacturing capabilities of the current team, we generated a list of guidelines for our new submarine:

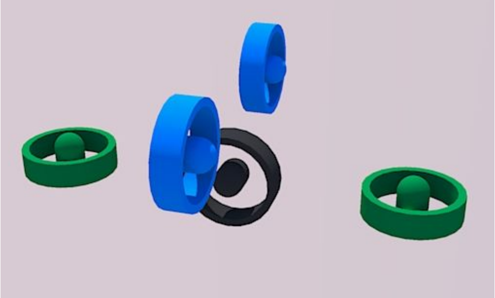
1. It needed to be lighter. The previous submersible was a bit over 50 lbs when fully ballasted. Due to low member count, especially on the hardware side, it is important that one member be able to carry the sub and tether.
2. It needed a *lot* more computational power than a Raspberry Pi. The complexity of the identifying images led us to the conclusion that simple, computationally-light algorithms like color detection would be insufficient.
3. We needed a frame that things could be mounted to. The solid wings and ballast plates of the previous sub made expansion challenging.
4. It needed to stay level while moving forward. This may seem obvious, except that the wing of the previous sub coupled with the large flat surface on the front amplified any pitch changes into an oscillation of pitching up and down during continuous forward motion.
5. It needed to be able to move sideways. Again obvious, but the previous sub had 4 vertical thrusters and 0 lateral thrusters. We realized that backing up, turning, and moving forward again is a horribly inefficient solution, and not all that compatible with a computer vision system which prefers to maintain sight of a target, from a consistent angle, at all times.

3. Vehicle Design

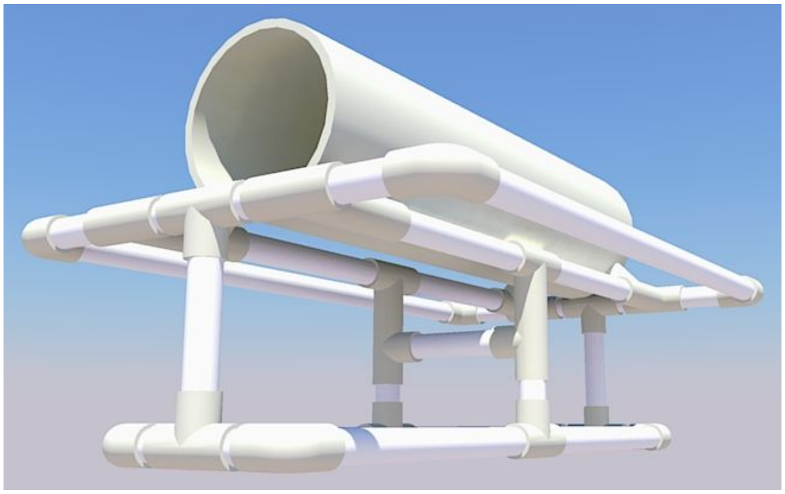
We discovered early on that the majority of our monetary investment would need to go towards a computer vision system. We considered the offer of the Jetson TX2 by the competition organizers, however we wanted a solution owned by the school that would persist even if lack of members meant the team couldn't compete for a year. In addition, we didn't want to potentially flood and destroy someone else's hardware.

We had two main software applications in mind. First, the ever present OpenCV for various image processing algorithms. Second, the YoloV3 neural network architecture. Unfortunately, the processing requirements for these two applications are nearly polar opposites. Most OpenCV algorithms run in a single thread which benefits from a fast single core speed and a complex instruction set. Neural network execution on the other hand prefers a GPU or specialized architecture with lots of cores. To balance these requirements, we selected an Intel NUC and low power desktop GPU. Having selected the largest components, we were able to identify a reasonable diameter for the hull.

We set a weight target of 30 lbs, then selected a length for the cylindrical hull that could reasonably fit the computer system and power components. Back-of-the-envelope math and experimentation indicated that we could more efficiently fill volume with off-the-shelf components by increasing the length of a narrow hull than by making a hull wider.

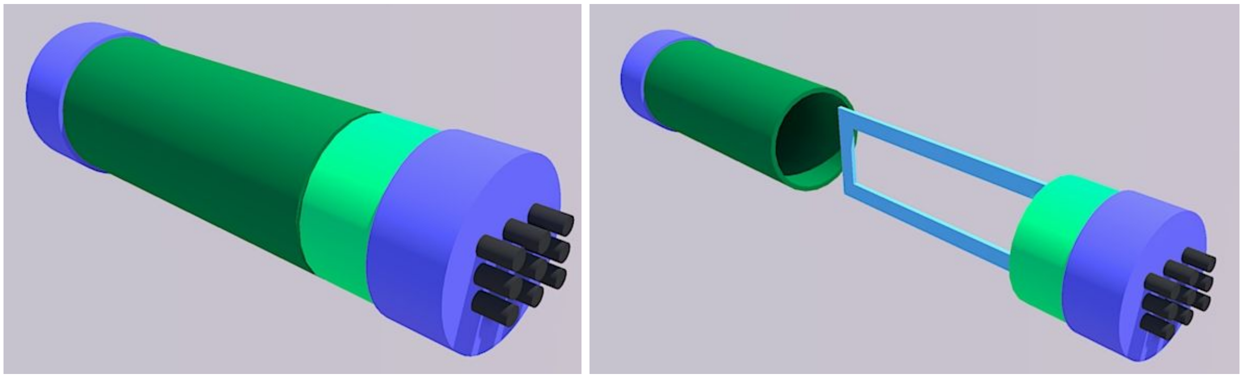


During return shipping from last year's competition, one of the team's 6 thrusters was cracked. A replacement thruster was not in the budget, so we designed a thruster layout for 5 thrusters keeping in mind the new requirement for lateral movement. Two thrusters are allocated for forward movement and turning (blue). Two thrusters are used to hold depth and for pitch control (green). A single thruster is placed in line with the center of mass for lateral movement (black). The frame will be designed to be stable on the roll axis, and any pushing will be done along the axis of the hull.



We selected PVC as a frame material because of its durability, rigidity, ease of modification, and extremely low price. In addition, it naturally presents a thin profile thereby avoiding any preference for a certain direction of travel. The frame should provide mount points for the hull, be structurally sound both in the water and on land, allow easy access to the hull, and provide mount points for the thrusters and other devices. PVC pipe and fittings are denser than water, allowing us to use a flooded frame as a start for our ballast and establish a center of buoyancy around the hull tube.

The frame has been optimized to reduce the number of fittings used while providing mount points for thrusters inside the profile of the frame. Although it decreases efficiency, mounting inside the profile reduces the chance that a diver grabs a thruster, or that a thruster becomes damaged during shipping. The hull is located by pins in angled tees, and fastened to the frame by straps.



Although not yet finalized, the hull will most likely be split into two parts. In the rear will be a short utility section containing the cable glands, structural mount points for the electronics sled, and potentially a water cooling loop. A single clamped gasket will be placed between this and the removable section of the hull which will be a similar diameter capped pipe. This method allows full access to the electronics sled with only one major joint to focus on waterproofing.

The software team is in the process of running experiments to determine whether a set of keypoints fed to a feature matcher is sufficient for locating task objects, or whether a neural network must be trained (and whether that is even feasible in the time remaining).

4. Experimental Results

Water testing has not been performed on the sub as the sub is still under construction. A team member has started acquiring video to test and train computer vision algorithms using an underwater camera at various resolutions and exposures.

The team has verified that a GPU can be used with an Intel NUC by adapting M.2 to PCIe and providing a stable 12V rail. The current adapter path is a bit convoluted: M.2 → PCIe 4x → PCIe 1x (bandwidth) → PCIe 16x (slot, not bandwidth) riser card. The limiting factor in this chain is currently the 1x riser which reduces available PCIe bandwidth to 1 GB/s. (Bowling, 2019) The Nvidia GT 1030 selected has 4 PCIe lanes enabled, meaning that if we can eliminate the 1x adapter from the chain, we can quadruple the bandwidth available to load network architecture into GPU memory. With about 1 Tflop and 2GB of VRAM, this GPU is at a disadvantage relative to the Jetson TX2. (En.wikipedia.org, 2019) Nonetheless it can load YoloV3 at up to 406x406 resolution and handle upwards of 5 frames per second when running through darknet. The main advantage of this method is that we can swap in an arbitrarily powerful GPU to the same riser card with a simple system reboot and train on the system while on-shore with the hull open.

It is important that the GPU be doing anything at all before the first CUDA call is introduced by darknet. For example, a forward pass of YoloV3 406x406 on an initially sleeping GPU is 400 ms, however that same network on the GPU when also running the display server produced a time of 170 ms. That yields a usable 5 fps when implemented for a video stream. Fortunately, even when booting headless, the Ubuntu operating system now selects the external GPU as the default GPU and starts a few processes on it that consume between 50-250 MB of memory. The 406x406 network consumes only 1.61 GB of VRAM, while the 620x620 network fails to load at all due to an *out of memory* error.

A significant amount of in-water testing is planned:

- 1 hr: Verify waterproofing of gasket and cable glands
- 2 hrs: Tune PID controllers
- 5 hrs: Acquire footage for training neural network and feature matchers
- 12 hrs: Test identification and motion algorithms on sub while submerged

5. Acknowledgements

We would like to personally thank each and everyone who has contributed to the Robosub project. Here is a list of outstanding people and organizations that have supported UMBC Robosub in 2019:

UMBC - University of Maryland, Baltimore County
Daniel Barvenik - Chief Systems Engineer | Strategic Systems Program, BAE Systems - Mentor
Jamie Gurganus - UMBC Mechanical Engineering Professor - UMBC Retriever Robotics/Robosub Faculty Advisor, Mentor

Keith Bowman - Dean, College of Engineering & Information Technology - Sponsor
UMBC Mechanical Engineering Department - Sponsor
IEEE Executive Board - Sponsor
BAE Systems - Sponsor

<u>Gritstarter Crowdfunding Campaign Donors</u>
Sarah Corley
Susan Bastress
Allen Bartlett
Ann Sieracki
Joshua Dishong
Dick and Anne Gillett
Anne Hairston-Strang
Nancy Selden
Catherine Bartlett
Melissa and Peter Hairston
Janet Vance

6. References

Bowling, M. (2019). *PCI Express Interface*. [online] Trentonsystems.com. Available at: https://www.trentonsystems.com/blog/pci-express-interface [Accessed 6 Jul. 2019].

En.wikipedia.org. (2019). List of Nvidia graphics processing units. [online] Available at: https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units#GeForce_10_series [Accessed 6 Jul. 2019].

Appendix A: Expectations

Subjective Measures			
	Maximum Points	**Expected Points**	**Points Scored**
Utility of team website	50	20	
Technical Merit (from journal paper)	150	100	
Written Style (from journal paper)	50	30	
Capability for Autonomous Behavior (static judging)	100	50	
Creativity in System Design (static judging)	100	50	
Team Uniform (static judging)	10	0	
Team Video	50	30	
Pre-Qualifying Video	100	0	
Discretionary points (static judging)	40	10	
Total	650	**290**	
Performance Measures			
Weight	See table 1 / Vehicle	100	
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	0	
Gate: Pass through 60% section	200	0	
Gate: Pass through 40% section	400	400	
Gate: Style	+100 (8x max)	200	
Collect Pickup: Crucifix, Garlic	400 / object	400	
Follow the "Path" (2 total)	100 / segment	200	
Slay Vampires: Any, Called	300, 600	600	

Drop Garlic: Open, Closed	700, 1000 / marker (2 + pickup)	700	
Drop Garlic: Move Arm	400	0	
Stake through Heart: Open Oval, Cover Oval, Sm Heart	800, 1000, 1200 / torpedo (max 2)	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 / 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	500	
Total		**3350**	

Appendix B: Component Specifications

Component	Vendor	Model / Type	Specs	Cost (USD)
Buoyancy Control	---			
Frame		¾” Sch 40 PVC	pipey	$20
Waterproof Housing		6” SDR-35	Teal (supposedly green)	$30
Waterproof Connectors		Cable Gland, Various	Hopefully waterproof : D	
Thrusters	Blue Robotics	T100	5.2 lbf, 135W	
Motor Control	Blue Robotics	Basic ESC R2	25A	
High Level Control	PJRC	Teensy 3.5		
Actuators	---			
Propellers	See *Thrusters*			
Battery	ZIPPY	2x Flightmax LiFePo4 Pack	2x 4200 mAh, 30C, 4S1P = 14.4V	
Converter	---			
Regulator	DROK	Buck/Boost	80W	$20
CPU	Intel, Nvidia	NUC 8i3BEH, GT1030	Intel 8109U, 256GB SSD, 8GB RAM, ~1 TFLOP (GPU), 2GB VRAM	~$500
Internal Comm Network		USB, I2C		
External Comm Interface		WiFi AP	2.4 GHz	$40
Programming Language 1		Python		
Programming Language 2		C++		
Compass	---			
Inertial Measurement Unit (IMU)	Pesky Products	Ultimate Sensor Fusion Solution, MPU9250 variant	EM7180 Sensor Hub coupled with MPU9250 IMU	$40

Doppler Velocity Log (DVL)	---			1 kidney and the first born child
Camera(s)	Logitech, Blue Robotics	C270, Low Light Analog Camera		
Hydrophones	---			
Manipulator	---			
Algorithms: vision		YoloV3, AKAZE, Stereo		
Algorithms: acoustics	---			
Algorithms : localization and mapping	---			
Algorithms: autonomy		If-then-else behavior selection		
Open Source software		OpenCV, USFS library, Arduino		
Team size (number of ~~people~~ ducks)	7			
HW/SW expertise ratio	3/4			
Testing time: simulation	---			
Testing time: in-water	1 hour recording training images	At least 20 hours planned		