Autonomously Navigated Discovery Yacht (A.N.D.Y.)

K. Hattan, T. Giles, J. Sucharitakul, E. Harwood, D. Knapp
Oregon Institute of Technology (Oregon Tech), 3201 Campus Dr, Klamath Falls, OR 97601

I. Abstract

The goals of OTUS ANDY Team at Oregon Institute of Technology (Oregon Tech) are to obtain learning objectives and experience hands-on knowledge while strengthening theoretical aspects together at 2020 RoboSub online competition, even in restricted situations. The purpose of our project is to design and manufacture a functional autonomous robotic submarine capable of completing a couple of tasks underwater including passing through a mandatory gate. We've done it through 3D models using Blender like virtual reality this year as shown in Sec. 4. We have been fortunate to work at home remotely even in the spring term, July and also came to school labs in the early August so that we were able to build the vehicle through utilizing OIT machine shops and robotics lab while testing in a limited capacity, though, for essential propulsion test under autonomous mode (also see Section 4). We have designed four candidates for the 2020 RoboSub competition in the last fall 2019. With thorough design matrix and tough discussion, we selected the final version of hull and frame for the Competition that enables our strengths to focus on navigation to complete as many tasks as possible such as a torpedo to shoot either Bootlegger or G-man after opening via a robotic arm, which is designed for up/down motion to also lift barrel cover and to drop markers or drop bottle into 2 bins while using a new magnetic dropper. Thanks to the participation of computer/embedded/software teams, we were able to build a totally new Embed. computer & SW system with a package of a new Raspberry Pi 4 (SBC), Arduino, and Pixhawk autopilot device with Raspbian OS onto Linux, Pytorch library with Python and C++, compiling in the visual studio to incorporate image and vision system, state machine for mission control to develop competition strategies, depending on situation at the competition, and also IMU with pressure sensor systems to control attitude. Additionally, we have built a virtual simulation with 3D models to make scenarios and it is going to further develop all tasks for the next year. We look forward to competing our robot at RoboSub next year on site, TRANSDEC pool.

II. Competition Strategy

2.1 Software State Machine

The software side of the project had vast improvements due to the many contributors to it. Mission Control encompases any programs that need to be done on the onboard computer before the vehicle would descend into the water. Any instructions inputted there would then be translated into SQLite then exported to a text file. A more detailed explanation is in the next section. Once the sub is turned on underwater, the Mission Execution would start when the vehicle is turned on externally and take the text file from Mission Control. Based on the instructions from said file, the code would extract information from Tensorflow, which calculates any important variables from the camera via OpenCV like distance towards the object and the size of the object. Based on the results of the information, the code will tell the arduino to make the motors run in specified directions to reach the assigned objective. This process would be looped until all assigned tasks are complete.

2.2 Mission Control

The vast majority of Mission Control takes place in the graphical user interface (GUI), designed to be used for the club years on. The GUI has a select then drag and drop interface, simplifying the instructional process for the vehicle. Most of these drag and drop instructions are simple commands such as moving forward and rotating right, but more of the complex ones rely on camera and arduino during and after mission execution. These commands are:
'Log Start Point', 'Return to Start', 'Move To Target', 'Move Past Target'.
The first two rely on the coordinate position where the vehicle first starts out at, so after it finishes the tasks, it would automatically go towards the logged point using other instructions to do so. The last two commands rely on the vision processing info.

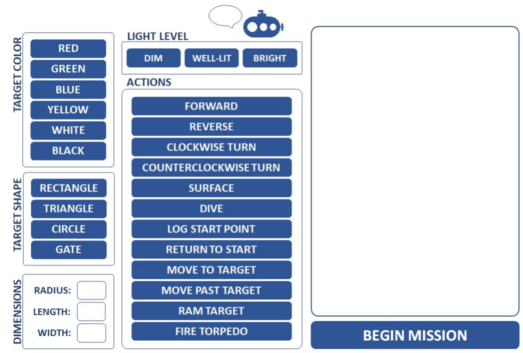
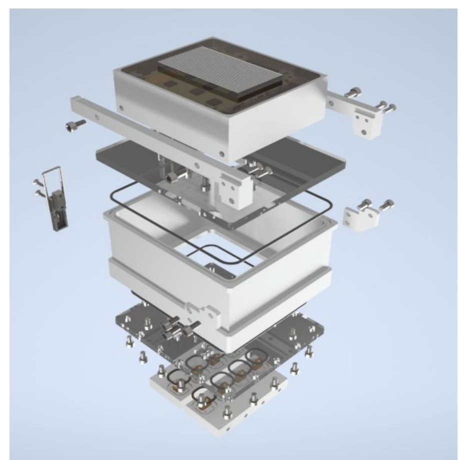


Fig. 1: GUI for ANDY

III. Vehicle Design

3.1 Electronics Housing



Found at the heart of ANDY is a central aluminum waterproof containment system designed to house and protect key electronic components. The body and lid are machined from aluminum stock, then supplemented with epoxy, polycarbonate, and rubber gaskets to create a functional, easily accessible housing. The waterproof housing consists of four major components:

Fig 2. electronics housing.

Main Body:

Now machined from billet aluminum, it serves as a mounting platform for electrical components such as relays, a flight controller, and a power distribution board. Previous designs required welding to construct the housing. Creating the housing from a single billet has allowed us to reduce weight while increasing pressure and depth capabilities. The Electronics housing consists of three main components. The lid, the potted ESC's, and the Polycarbonate sheets. The lid construction is dual purpose component machined from billet aluminum bolted to a polycarbonate base, potted Electronic Speed Controllers for thruster control, however, this year the ESC's are smaller, lighter weight, and using polycarbonate sheets allows for a visual inspection of the gasket surface.

Torpedos:

The torpedoes pack a lot of technology in a small form. Using a watertight resin 3d printed housing, the torpedoes incorporate on board water cooled motor controllers, counter rotating propellers powered by brushless motors, and light activation sensors. They are designed to be loaded into a tube and activated by a simple flashing light.

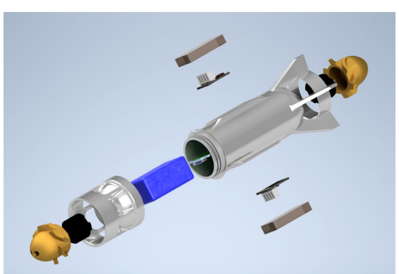


Fig 3: Exploded view of torpedo

Control system:

The control system remains largely unchanged from the last iteration, however there are a few important improvements. These include a fully potted computer that is passively water cooled.

Marker dropper:

We have elected to use two electro-magnets capable of holding 1 inch ferrous stainless-steel ball bearings as markers.

Arm & Gripper:

The gripper consists of four 3D printed claws to increase gripping area and reduce the need for accuracy when deploying it to pick up an object. The arm and jaws are manipulated using a hydraulic system powered by two stepper motors powering 3D printed gear pumps. The arm extends the length of the sub and provides a stable mount for the 4 gripping jaws. It is lowered and raised by means of a hydraulic cylinder attached to the frame rail of the sub.

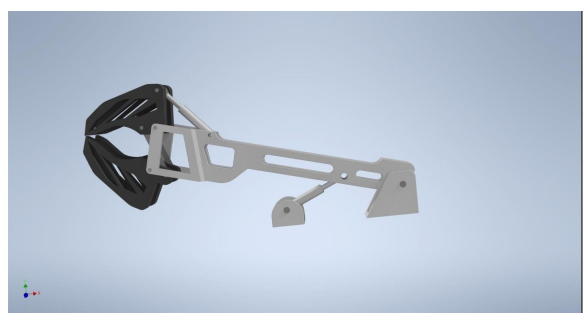


Fig 4: Gripper assembly.

3.2 Embedded System Design

The purpose of this system is to create both a navigation library for future students at the Oregon Institute of Technology and use this library to control an autonomous robotic submarine that will be submitted in a competition. For the autonomous submarine, it must be able to perform various tasks through communication between hardware and software and without human interference.

Hardware:

The RoboSub has 8 T200 BlueRobotics thrusters, A Raspberry Pi for main control and computation, a PixHawk 3.5 for independent, no-interference gyro positioning data, and an Arduino Mega for independent thruster control. Each peripheral could be disconnected and enable the sub to run as best it could without it in case we did not have the board at the time, or wanted to debug a certain part of the navigation system. This allowed for an application that can be added onto and modified to perfection later on, even by future team programmers who could use our Bitbucket repository and Wiki with ease.

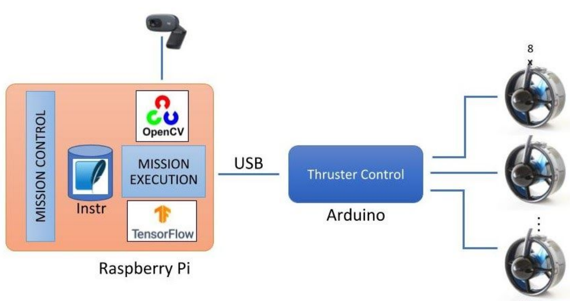


Figure 5: Control Tree

The system is not custom-integrated into one another, but the communication between the boards is extremely steady and fast, with no problems arising from the high-speed serial communication. The Raspberry Pi is able to efficiently decode the data sent out and in of itself, whether it be to the Arduino Mega, or from the Pixhawk.

Software:

The navigation system is built around allowing our RoboSub to travel to any orientation matrix or position. We do not use many external libraries, all the navigation code being our own, only some imported Python libraries for peripheral communication. The original code was written to incorporate ease-of-testing because of our limited time constraint of around three weeks to make a programmed system.

The RoboSub is completely autonomous within a state-like system, where it reads commands generated from GUI that enables any person, programmer or not, to generate commands for our RoboSub to parse, process, and navigate with. When the sub is run, either by loading a button listening program beforehand, or running a certain script in the console of the Raspberry Pi host computer, which runs the commands accompanied by supplementary matrix, target, or position data.

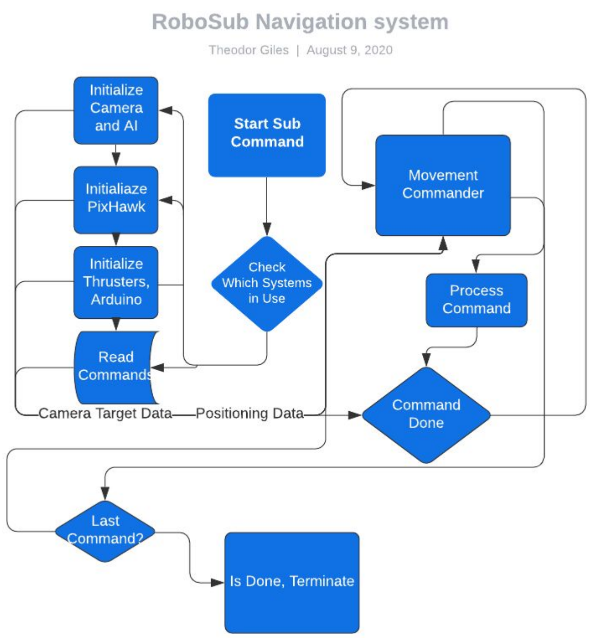
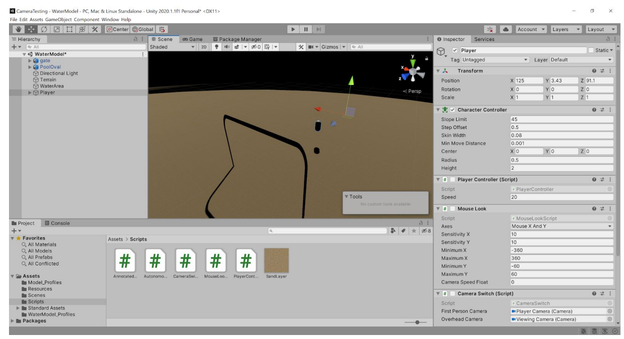


Figure 6: Serial Hookup between boards on the RoboSub

VI. Simulation/Experimental Results

With our ability to physically test propulsion during our short time working socially distanced, we developed a navigation system through data in water and out-of-water tests. We were able to develop a proportional auto-orienting system from this small amount of testing, which after a couple days of testing. Most of the code for the main navigational application was written in the weeks before having an actual RoboSub to work on, but we were able to recreate the electrical and board communication system in our own quarantine homes. As well as physical testing, we also developed a simulation for showing the entirety of what our vehicle could do.

To simulate the results of the procedure while the vehicle was under construction, we have designed a bareback simulation of the preliminary task of the Robosub Competition. This simulation was constructed in the Unity engine, with the script/code used based in the C# language, and attached towards the Player object in the engine. Alongside with manually controlling the object, there is a script meant to autonomously run through the sim by itself.



V. Acknowledgements

Thank NASA Oregon Consortium for sponsoring the project partially by NASA Oregon Space Grant (80NSSC20M0035). Special thanks to Oregon Tech MMET Sr. and CSET Jr. Project teams (teammates below) as well as AUVSI Club members, especially Kai, Theodor, and Jay who have dedicated themselves to the project.

Student Players at Oregon Tech (OIT)

- Kai Hattan: OTUS ANDY team w/Mech. leader
- Jay Sucharitakul: Club president & Simulat. lead
- Theodor Giles: OTUS ANDY Embed & SW lead
- David Knapp: Sr. Project Mech Hub/Frame lead
- Eric Harwood: Robotic Arm/Web/Data lead
- Rowan Parker: CSET Jr. project teammate lead
- Dillon Wall: CSET Jr. project teammate
- Brett Sprague: CSET Jr. project teammate
- JJ In: an AUVSI Club member
- Nathan Fuller: MMET Sr. project teammate
- Dongbin "Don" Lee: faculty advisor and mentor

We cannot thank enough to Tom Keyser, Dean of College of Engineering and Erin Foley, VP of Student Affairs at OIT for partially sponsoring the project through RBC and their support and encouragement, and Brian Moravec, Associate Dean and acting MMET Dept chairperson for his huge support from lab availability to administrative guidance, etc. We especially thank Barbara Metcalf for her excellent administrative services and lab manager, Phil Dussel's amazing help for lab and facility utilization. The many staff members at Oregon Tech that deserve recognition for helping students work safely at recently remodeled labs these individuals include: Letticia, Sherry, Mary, Gaylyn, Edwards, Tabitha, Rachel, Cindy, Barb N., Jennifer, Sandi, etc.

IV. References

2020 RoboSub Task Ideas with Themes, AUVSI Foundation, Robonation, https://robonation.org/app/uploads/sites/4/2019/10/RoboSub-2020-Task-Ideas.pdf

Engineering Primer Document for RoboSub Competition, Maritime RobotX Challenge, AUVSI Foundation and ONR, 2006

Image Processing Primer Document for Autonomous Vehicle Competitions, AUVSI Foundation and ONR, 2014

Oregon Tech RoboSub Team, OTUS ANDY, https://sites.google.com/view/otusrobosub/home

Appendix A: component specifications

Component	Vendor	Model/Type	Specs	cost-new
Buoyancy Control	OIT	Motor Control	Programmed	
Frame	OIT	AL6160, ABS	Machine/3D print	
Waterproof Housing	O-ring, design	Polycarbonate tube, Resin, Wax	A bulk	
Waterproof Connectors	Blue Robotics/Amazon	XT30/60 (IP-68)	3-/9-pin connectors	
Thrusters	Blue Robotics	T200		
Motor Control	Speed Control	Basic ESC	30A	$27

High Level Control	Outer-loop PID control	Students' Programming		
Actuators	Hi-Tech	Servo/stepper brushless	Electric motors	
Motor Driver	H-bridge	Stepper motors		
Propellers	Blue Robotics	Fwd/Bwd set	Left/Right blades	
Battery	Panasonic Lipo	36EA(18,650mA)	2 * 18/stack	$15
Regulator	Built-in			
CPU	Rasberry Pi 4, 1.5GHz	ARM Cortex-A72	Q-core 64bit/4Gb	$56
Int comm network	Ethernet	Gigabit		
Ext comm interface	Wired for testing	RJ-45 wired	Fiber optic wires	
Program Lang. 1	Python/Pytorch lib			
Program Lang. 2	C++			
Compass	Pixhawk	magnetometer		
IMU	Pixhawk	Accel'/Gyros		
DVL	NA			

Camera	Logitech	C270		
Depth Sensor	Pressure			
Hydrophones	Teledyne Reson	T4019		
Pinger	JW Fisher	Mid-Freq range	20-50KHz/2k-3k feet	
Manipulators	OIT	Custom-made	Front & Bottom Foldable arms	
Algorithm: vision	VIGRA lib/OpenCV			
Acoustics	NA			
Local' & mapping	3D image building			
Autonomy	OIT			
Open Source SW	Python,C++,OpenCV,	Raspbian,Pytorch	Blender (3D)	
Team Size	Approx. 10			
HW/SW ratio	1:1 (5:5)			
simulation time	3 Months			
in-water testing	7 days			

Appendix B: outreach activities

Kai, David, Eric, and Nathan, Engineering Week Demo to OIT students and Visitors, 2/18/2020
Don Lee, Judge, VEX Robotics, Klamath Community College/Mt. Mazama HS, 2/01/2020
Kai Hattan et al&Don Lee, Service to Newcomers in Engineering, Fall Review, Oregon Tech, 10/19/2019