San Diego City College Autonomous Underwater Vehicle: Design and Implementation of the Zoidberg AUV

Giovanni Angel, Mathew Bailey, Nikita Bobrov, Gina Bochicchio, Emma Brand, DJ Brown, Linda Clark, Benny DeLaRosa, Rusty Dillard, Caroline Dillman, Calvin Dixon, Leila Firestone, Ivan Galindo, Andy Hernandez, Shane Jost, Timothy Mann, Luis Medina, Steven Nieder, Glenn Oberlander, Isabel Paredez, Jorge Perez, Jason Plojo, Christian Pratt, Bob Pruitt, Dominic Quijano, Ned Richards, Cindy Rios, Danny Rosales, Bumi Sanchez,Arnold Suarez, Naylynn Tañón, Parker Togut, Anthony Verduzcopaz

***Abstract* - Zoidberg is the autonomous underwater vehicle (AUV) built by San Diego City Robotics (SDCR) to compete in the 2018 Robosub competition. A team of approximately 30 undergraduate students, along with the support of advisors and sponsors, designed and built an all new, low-cost AUV capable of completing the complex tasks the Robosub competition provides.**

I. COMPETITIVE STRATEGY

San Diego City Robotics' primary objective is to develop our members' engineering, leadership, collaboration, and communication skills through the development of an autonomous underwater submarine. Our main focus was to develop a low-cost platform that allows new two-year students to understand and contribute to the project quickly. Our newly designed AUV, Zoidberg, was built to compete in the annual Robosub competition cosponsored by Unmanned Vehicle Systems International (AUVSI) and the Office of Naval Research. The competition provides a variety of obstacles mimicking real-world scenarios requiring navigation, visual and acoustic observation, mechanical manipulation, and complex decision making. Keeping our primary objective in mind, we chose to focus on a simplistic, cost-effective approach to tackling the basic navigation and visual aspects of the competition.

II. DESIGN CREATIVITY

Our design was based around our strategy of keeping the platform simple and cost effective. The mechanical design incorporated upcycled parts from previous team builds, 3D printed components, and low-cost components and materials, while the electrical design included hardware from the previous year's sub along with several vital components donated to us by sponsors. The software design focus was keeping the learning curve manageable for new team members with minimal experience. We accomplished this by minimizing the use of complex custom code when possible.

1. MECHANICAL

The mechanical design started with the choice between an all new custom fabricated chassis or modifying a previously used custom chassis. To keep costs to a minimum, we chose to modify the existing chassis we had. The high-density polyethylene (HDPE) side panels were

flipped and new holes were cut to allow optimal water flow for the thrusters. The thrusters were mounted with custom designed 3D printed mounts in a vectored layout *(fig. 1)* providing maximum maneuverability.

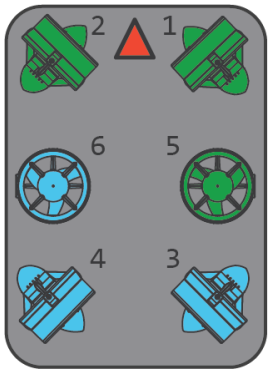

fig. 1

After experiencing difficulties with our previously used custom end caps, we chose to use a water tight enclosure from Bluerobotics. Their end caps give us the ability to easily access the components housed inside by removing just the faceplate leaving the main o-ring seals in place when poolside. We went through multiple design changes for mounting the enclosure that consisted of 3d printed straps, and several iterations of lower cradle mounts. After testing the different ideas, we found that a pair of stainless steel u-bolts wrapped in a closed-cell rubber and a set of custom printed cradle mounts gave us the best stability *(fig. 2)*.

Our biggest design challenge was integrating the new doppler velocity logger (DVL) from Rowe Technologies. Because the position of the DVL was crucial, we decided on a top mount design preventing potential interference with the signals. To support the weight of the sensor, we chose to fabricate a custom mount from HDPE which was attached to the chassis with anodized aluminum rails. This also provided the mounting location for the main enclosure.

For our camera housing, we chose to use a smaller version of the same enclosure as the main hull for similar reasons. The design is proven and provides quick, easy access to the camera. We repurposed a pair of aluminum light brackets to mount the enclosure and tied them into the chassis with custom designed 3d printed mounts.

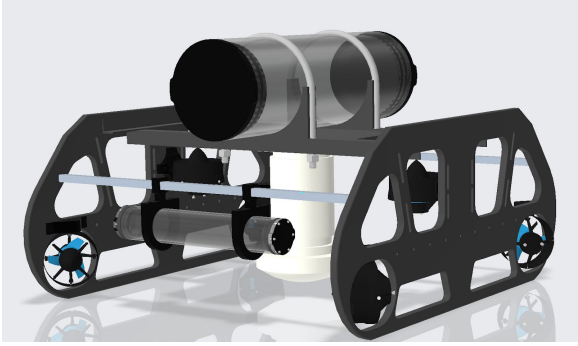

fig. 2

The internal mounts for our camera and electronic hardware were designed to be lightweight, low-cost, and effective. The main enclosure features a 3D printed electronics panel that slides into the enclosure and fixes to multiple ring guides. This gave us a modular, secure design and easy access to the internal components.

2. ELECTRICAL

Our biggest change this year in electrical design was the use of multiple off-the-shelf electrical modules instead of custom circuit boards. Most of the components are quadcopter based solutions retrofitted for a submarine, which saved the

team money and development time. The implementation of a power management system which monitored the power consumption to the motors helped out this year, in the prevention of damaging the Lithium Polymer batteries. The batteries used this year are of a larger power capacity and allow the submarine dive for longer periods. Our killswitch consist of a hall effect sensor on a custom circuit board connected directly to a 12 volt 100 amp relay.

3. SOFTWARE

With our main strategy revolving around simplicity, we chose to use the Robot Operating System (ROS), in conjunction with ArduSub, OpenCV, and custom software. All this software runs on the Jetson TX1 development board. Because most of our team members are two-year students, we wanted to use and create software that was easy to learn and was well documented. For this reason, everything was programmed in Python. It was compatible with OpenCV and allowed our team to explore topics of machine learning. Using ROS, OpenCV, and ArduSub gives our team a solid foundation to teach new students the basic functionality. This was designed to break the cycle of starting from scratch every year on designing a control system and basic navigation when core team members transfer.

We chose to use ROS because it gives us a versatile and module operating system that is adaptable to our application. It establishes communication between all of our components: Pixhawk, Jetson TXI, DVL, power module, GPS, and Zed camera. Implementing ROS gives us the capability to send all sensor information to a central location where it is published and ready for use when needed [1].

In previous team builds, the control system consisted of custom code written by core team members in C++ which hindered new members' understanding of its operation. Our changes reflect a system that is easy to learn and allows them to focus on individual tasks and more complex problems such as object recognition and color detection.

For our visual detection, we are running OpenCV. It provides us a common infrastructure for computer vision application and accelerates the use of the AUV's perception[2]. Using OpenCV we can detect objects positions relative to the AUV that we can use to set new headings and motor commands.

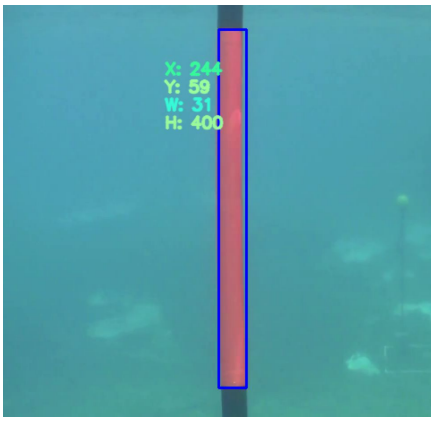

fig.3

In addition to OpenCV, we can navigate with the DVL. Our DVL is initiated via python code that can produce the serial commands normally sent by the pre-supplied control software, Pulse. Once

the DVL is engaged, the data received is sent over serial lines and published to a ROS node.

The mission control system is custom software created specifically for navigation and individual tasks. We chose to use our own design to tailor our system to accomplish the complex tasks required in the competition. Because this software was built from the ground up, our team devoted a considerable amount of time in developing it to be reliable, efficient, and well documented. As with all the subsystems, the mission control system was designed with our main goal of simplicity in mind.

III. EXPERIMENTAL RESULTS

Although our experimental testing was limited, we were able to successfully test the various systems of the AUV and make adjustments based off the data collected. We scheduled as much time possible to test in the pool and were able to spend over 35 hours of time in the water and even more time in simulation.

Due to time constraints, we planned individual water tests for the various systems onboard allowing the subteams to test independently. All of our tests were successful with very limited issues. We scheduled test time for each system after the engineering and design phases were completed. After the successful tests, the subsystems were integrated into the AUV and tested again. We spent the large majority of our time on the design and engineer phase which allowed our limited in time in the water to be as efficient as possible.

IV. ACKNOWLEDGEMENTS

SDCR would like to extend a special thanks to our faculty advisor Prof. Bob Pruitt who's supported our efforts since the teams inception. We'd also like to thank team advisors Jason Plojo (former team leader) and Patricia Shanahan (software advisor/donor). We'd also like to thank the incredible support we've received from our sponsors: Rowe Technologies (DVL), Robo3d (3D Printer), TE Connectivity (connectors), MRobotics (PixHawk/hardware), San Diego Maker's Guild (donor), Nvidia (TX1), Solidworks, Mathworks, and San Diego City College Engineering Department.

V. REFERENCES

[1] Wasowski, Andrzej, "Robot Operating System" Open Source Robotics Foundation. wiki.ros.org, June 9, 2018.

[2] OpenCV team, "Atomic Bomb" OpenCV. https://github.com/opencv/opencv/wiki, May 11, 2010.

Appendix A:

Component	Vendor	Model /Type	Specs	Cost (if new)
Buoyancy Control				
Frame	Used	Custom Fab	HDPE	
Waterproof Housing	BlueRobotics	6" Water Tight	w/ End Caps	$212.80
Waterproof Connectors	TE Connectivity	Wet-Conn	Mini	Donation
Thrusters	BlueRobotics	T200		$169.00 ea
Motor Control	MRobotics	Rev2		Donation
High Level Control				
Actuators				
Propellers				
Battery	Zande Elec.	YSD-12680	12V6800mA	$39.95 ea
Converter				
Regulator				
CPU	NVidia	Jetson TX1		Donation
Internal Comm Network	ROS			
External Comm Innterface	Serial			
Programming Language 1	C++			
Programming Language 2	Python			
Compass	Rowe Tech	SeaProfiler	600 khz	Donation
Inertial Measurment Unit (IMU)	MRobotics	Rev2		Donation
Doppler Velocity Log (DVL)	Rowe Tech	SeaProfiler	600 khz	Donation
Camera(s)	StereoLabs	ZED	Stereo Cam	$449.00
Hydrophones	Used	Custom		
Manipulator	N/A			
Algorithms: vision	Open CV			
Algorithms: acoustics	Custom			
Algorithms: localization and mapping	Custom			
Algorithms: autonomy	Custom			
Open source software	ROS	Open CV	ArduSub	
Team size (number of people)	35			
HW/SW expertise ratio	4:1			
Testing time: simulation	~40hrs			
Testing time: in-water	~35hrs			

Appendix B: Outreach

SDCR participated in:

- San Diego Maker's Fair
- Barnes & Noble Maker's Fair