Southern Polytechnic State University Autonomous Underwater Vehicle Design Rationale for Sublight

Douglas Allyson, Albert Cheng, Alex Evans, Andrew Geiman
Cody Meier, Jake Massingill, Jason Wilkins, Matthew Strauss,
Joseph Burley, Marly Iheanacho

Abstract

The Southern Polytechnic State University Sub-light is a littoral-class autonomous underwater vehicle (AUV) built by undergraduate members of the SPSU AUV Team. The vehicle has been continuously modified and enhanced over the past sever years, however the current configuration is a complete re-design from past vehicles, the product of a ten-month development period. The vehicle was designed almost completely using three-dimensional CAD and simulation in Solidworks design software. Among the new design's features are redesigned new main housing, and riveted and screw 6061 aluminum exoskeleton to provide structural security while minimizing weight. The submarine is equipped with two cameras for challenge recognition and maneuvering computer vision tasks, a pressure sensor for active depth control, and an inertial measurement unit for orientation control.

Dry Weight	13.6 kg	Thrusters	6x SeaBotix Brushless BTD150
Dimensions	78.74 cm x 38.10 cm x 48.26 cm	Camera 1	Logitech Web Cam
Max Depth	18.29 m	Camera 2	Hewlett-Packard
Degrees of Freedom	Yaw, Surge, Heave, Sway	Motor Drivers	Hercules Dual Motor Controller

Software

Operating System and Languages

To avoid unnecessary complication, an external software stack was used to provide communication and interfacing with the sensors and cameras. This allowed for additional time to be spent addressing the challenges, rather than perfecting these utilities. These functions were provided by Robot Operating System (ROS). As explained on ROS's website, the Robot Operating System is a set of software libraries and tools that assist in building robot applications. ROS possesses open source drivers, state-of-the art algorithms, and other powerful developer tools. As ROS releases a new version every six months, throughout the development period versions of ROS had to be updated, from "Groovy" to "Hydro" to "Indigo". Fortunately, little code had to be rewritten between each update.

Using ROS narrowed down the choices of operating system for the on-sub computers. We ran Ubuntu and Debian on the onboard computers, mainly because they were easy to use ROS on. Code was generally not developed on the on-sub computer; instead, team members use personal laptops or lab computers, which have ROS installed, to develop any software for the sub. One reason for this was for the concern that doing many builds or compiles on the microSD cards of our onboard computers could quickly wear them out. It was decided to also use Ubuntu on the team's development machines, for ease of use and greatest support of ROS.

Because we are using ROS, our options for programming languages were also limited. The two options for languages that can use the ROS libraries are C++ and Python. The team decided to go with Python after comparing it with C++ the previous year, and use OpenCV for our computer vision library.

Architecture

We decided to use a closed or strictly layered architecture while designing our Robot Operating System package. This architecture was chosen because we believe it is easy to comprehend and teach to students who are new at computer programming. We have four layers in our architecture: Control, Decision, Calculating, and Device. The nodes in each layer generally only interact with nodes from the layers directly above or below, as well as nodes from its own layer.

Control Layer

The control layer's job is task scheduling. The control layer manages when nodes from the Decision layer have control of the sub. It runs only one Decision node at a time so that no two nodes have control of the sub at any one time.

Decision Layer

This layer makes all of the decisions the sub makes. Each task has its own node in this layer; which are run by the Control layer. We used this approach to simplify updating code and to streamline the review and merging process for our content management system.

Calculating Layer

This layer handles all of the heavy number crunching functions and making use of sensor data. Our communication nodes that communicate with our motor controllers are also in this layer.

Device Layer

This layer contains software that pulls the data from the hardware and converts it into a format we can use. This layer contains ROS drivers for cameras, motor controllers, IMU, and pressure sensor.

Source Control

This year, we tried to make pulling code from our git repository and then building it easier and simpler. Our ROS package can be cloned from GitHub into a ROS catkin build workspace and then built right away, provided that the package's dependencies are installed. We also keep images of both our onboard computer's microSD cards in case a new card is needed immediately.

Hardware Restructure

This past year we lost our embedded PCs (EPC) to power source malfunctions. However, due to the popularity of ROS and our choice of Python as a programming language, we were able to easily port most functionalities on to a Raspberry Pi. We had planned to use two Pies that split the work and use ROSs networking functionality to have the two communicate. However, we ended up with a Raspberry Pi and an Odroid U3 networked together as explained in the next section.

Networking

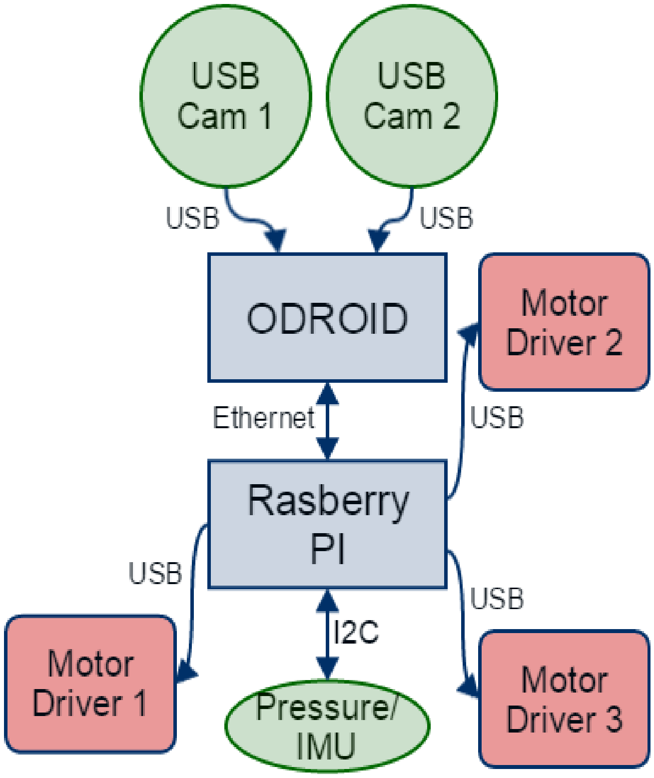
Because the sub is running ROS nodes on two separate computers (a Raspberry Pi and an Odroid U3), it was clear we needed a way for those computers to communicate. Furthermore, we needed an external tether to a laptop for maintenance and testing. The first solution was to simply run standard Ethernet cable between the three computers. The middle computer, the Raspberry Pi, was fitted with a USB ethernet dongle to allow it to accept both connections. While this allowed the Pi to communicate with the other two computers, the Odroid and the laptop could not ping one another. We then tried replacing one of the Ethernet cables with a crossover cable, but no matter how we configured the placement of this cable there was no change in our results.

Our third attempt was to use software called OpenFlow to turn the Raspberry Pi into a switch. This would allow us greater flexibility and expandability in the future if it worked. Unfortunately, the software we had access to encountered numerous errors. Because it was written in a language no one on the team knew (Erlang), we decided there must be a simpler approach that would achieve acceptable results. Finally, we bridged the connections between the Pi's eth0 and eth1 (USB to Ethernet) adapters. This allowed us to ping, SSH, and communicate through ROS with three computers in the network, the Odroid U3, the Raspberry Pi, and a testing computer connected by an ethernet tether. Though this solution doesn't provide as much adaptability as the switch, it does achieve the requirements set for it.

Electrical

Safety

The Kill Switch is a magnetic switch similar to a door sensor for some home security systems. When a magnet is brought close enough, the switch will change states. The switch is then hooked up to an power circuit and a 30 amp rated transistor. The transistor is connected between the thruster batteries and the motor drivers. When the switch is in the off position, no current can pass through the transistor. A magnetic switch was chosen in order to reduce the number of possible places the vehicle could leak. A magnet can be attached to a brightly colored handle so the safety diver can quickly de-energize the vehicle in the event of an emergency.



Sensor Suite

The number of sensors utilized was limited by the events the team chose to compete in; as such the submarine is currently set up in a “bare bones” configuration. It has two cameras: one facing forward to locate and work through the challenges, and one facing the floor to handle line following for the movement to each successive challenge. The decision was made to use “off the shelf” parts for the cameras, i.e. webcams, to minimize expenses and design complexity. One of the camera is Logitech brand, and the other being Hewlett-Packard. The primary sensors are a MS5803-14BA series pressure sensor, which acts as a depth gauge, and the IMU, used for underwater navigation. The pressure sensor is used to keep the submarine within the proper range of the tank floor, ensuring the sub does not breach the surface unexpectedly. The Inertial Measuring Unit (IMU) detects changes in the vehicle’s orientation in three major axes: pitch, roll, and yaw. The flow chart above shows how the electrical component are connected.

Power Distribution Network

Initially the power system was housed separately from all the electronics; however this was found to significantly increase the weight of the overall submarine, with little additional benefit. The current fully integrated setup is simple and organized, with the focus of keeping power system maintenance as easy as possible. Motor power is provided by two 14.8 volt lithium polymer batteries connected in parallel. The maximum voltage rating on the motors is 19 volts, so a parallel setup was used to stay under the limit while benefiting from increased mission time. The power from the batteries moves through a 30A transistor which functions as the “kill switch”, and then gets passed on to three motor drivers to individually power the motors. The main power for the computers is a lithium polymer battery. Originally all power was to be stepped down from the motor battery setup; however this would have required additional circuitry to the submarine and thus another point of possible failure. The 7.4 volt lithium polymer battery worked well with additional circuits, making powering the computer as simple as plugging it in. All other electronics aboard the submarine e.g. the IMU, Cameras, Etc. are powered through the computers.

Connectors and Ports

The vehicle has eight Teledyne Impulse connectors. Six of the Teledyne Impulse connectors are three pin connectors with an IP64 rating and are rated to 30 amps per pin. Finally, the last two Teledyne Impulse connectors is four pin connector of the same type and rating as the six three pin connectors.

Propulsion

The submarine utilizes six SeaBotix thrusters for maneuverability. They are brushed DC motors encased in a waterproof housing. They are able to produce a peak thrust of 2.9 kg force and are controlled by three Hercules motor controllers that communicate by USB to FTDI. These drivers give users the ability to control the rotational speed and direction of the thrusters.

Structure

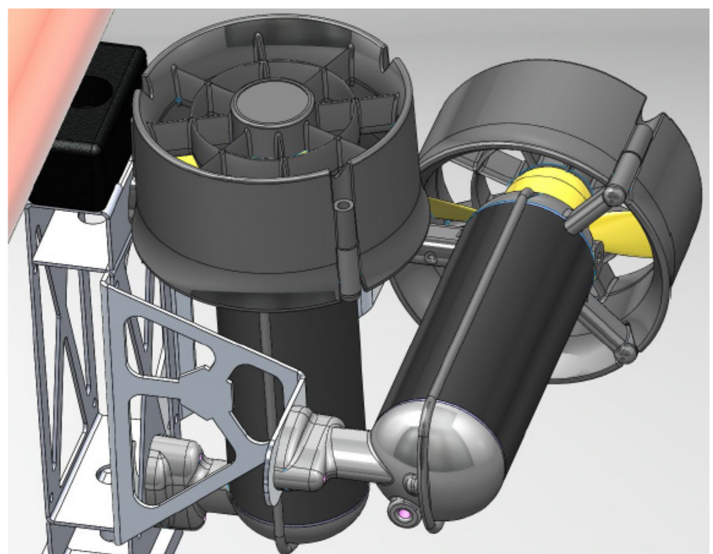
Main Housing

Our main housing is designed around an Ikelite marine videography case that was purchased towards the beginning of the season. In it, we house the on-board computers, motor controllers, cameras, and batteries. To enable us to connect to the motors outside of the housing a replacement end cap had to be made that included waterproof connectors for motors, an ethernet tether and a sensor cable. The end cap was designed in SolidWorks so that they could be made by CNC machine at a very high level of precision. The rationale for choosing aluminium for the end caps was that it has relatively low weight in comparison to the other metals, as well as a high rate of heat transfer. The end cap was designed to allow for multiple waterproof connectors to pass and connect to the electronics inside the tube. There were a total of 8 waterproof connectors passing through the housing. The two end caps utilize an O-ring channel so that the O-rings will be able to compress against the side wall of the tube, forming the watertight seal. The advantage of designing all the parts in SolidWorks was that it became possible to utilize the simulation and analysis suites available with SolidWorks. These tools enabled the Structures team to simulate the maximum pressure that the main housing could sustain before being destroyed.

Exoskeleton

The exoskeleton decision from previous years was to go small. In order to accomplish this, all motors needed to be kept close to the housing. To keep weight down, 6061-T6 aluminum was used. A few considerations aside from weight that we needed to take into account were directional flow/drag and packaging.

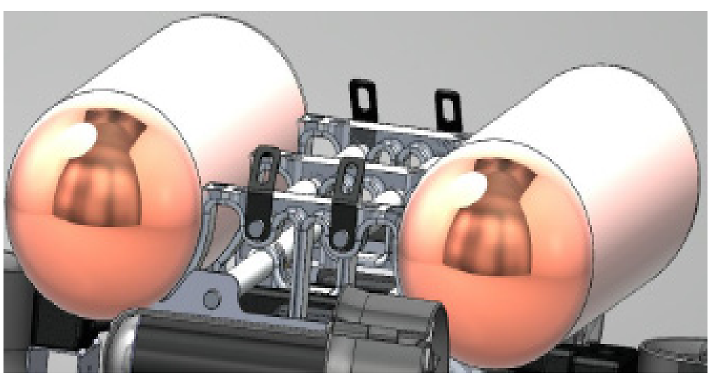
The forward and down thrusting mounts will be cut from 3.18mm 6061 aluminum using a Semyx waterjet and formed using a 175 ton Niagra Press Brake. Starting from our model, flat pattern drawings were converted to .dxf files for use by the waterjet. The ballast tank mounts will be milled out of 25.4x76.2mm Aluminum solid bar on a 4020 Fadal milling center.


mounting brackets for thruster

The camera housing and surrounding structure already had a nice fit around the center housing, the manner in which the camera housing mounts to the outer structure seemed to fit our needs so the rest of the exoskeleton had to follow suit.

Ballast

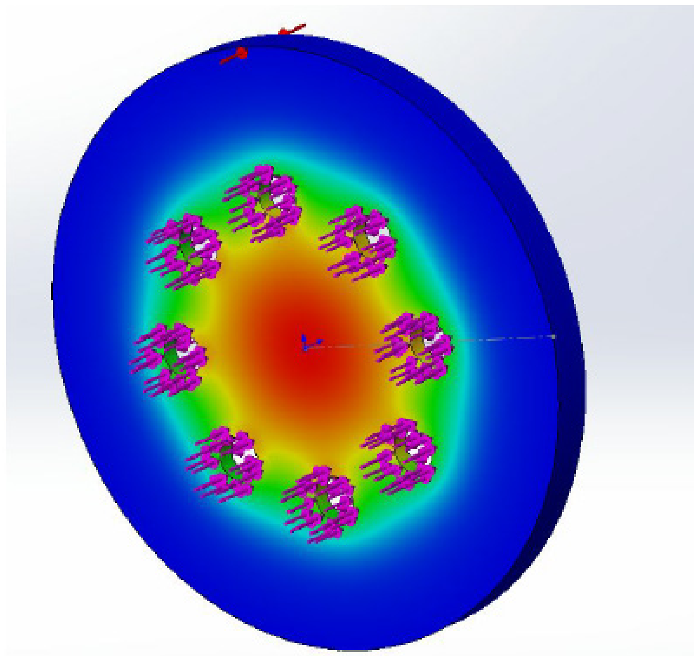
We chose to go with a cheap and easy design for the ballast tanks, leaving only the material choice and dimensions up for debate. We put two common plumbing caps on both ends of common piping; we just had to choose material and length. We determined in our first pool test with the sub that three PVC two inch diameter pipes almost works, this was done with old materials laying around the lab. We planned on using what we had first and make new ones with the data we received from the preliminary testing. When our test showed that the sub just barely sinks with the not streamlined and bulky old ballast, we then started the construction of the final tanks calculating in the proper size increase to make the submarine five percent buoyant. The new tanks we chosen to be made of two thin walled sections of copper tubing. We received a donation of this material and with the walls being able to support a pressure much higher than the sub will ever encounter, we did not have to worry about them failing. We also created a new PVC set that has smooth rounded ends instead of the bulky threaded end caps. This set was constructed in case the copper ones were not fabricated on time. In all cases, we used common clamps to hold everything in position, this allowed quick attachment and removal along with a weight that is negligible. These designs were later replaced with a series of lighter, but sufficiently buoyant bottles to reduce the overall weight of the sub.



CAD of Copper Ballast

Modified Rear Bulkhead

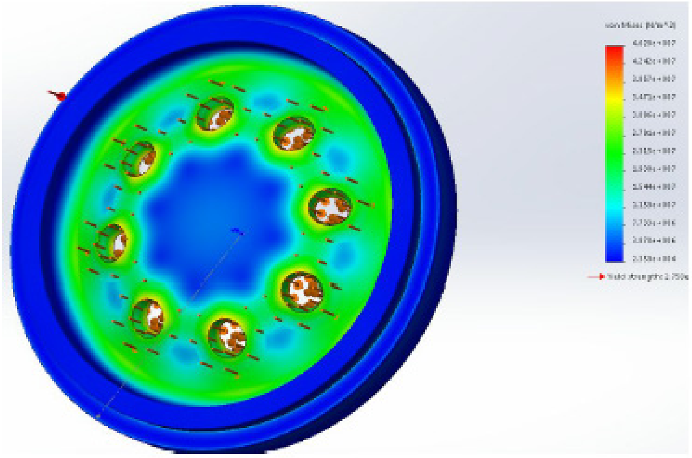
The modification of the Ikelite case's rear bulkhead was essential to the team's desired direction and success. It also proved that the machining schedule and processing could produce structure parts that met FEA expectations. This was very important for the team from an education perspective.The success of this component proved that the mechanical team could design, analyze, and manufacture aluminum 6061-T6 and not see any variation between the CAD model and the physical part.



Deformation fringe plot of the aluminum 6061-T6 Rear Bulkhead, water side.

In the past year, the mechanical team was weak in this area due to inexperience. Most of the members were only Juniors. This year, the team consisted of two mechanical engineering Seniors that had taken most of their high level analysis courses. The driving factors for the rear bulkhead are that it matched the original cap's outside dimensions, could hold six connectors, hold back all of the pressure without failing or fatiguing, and that the deformation was little enough to account for any possible o-ring separation.

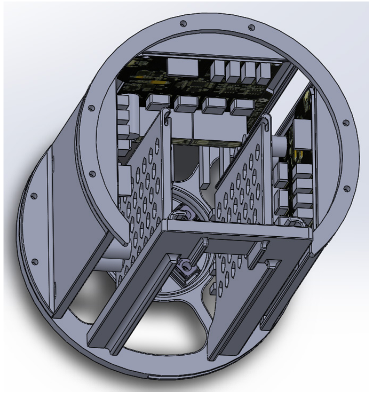
For the rear bulkhead to match the original cap's dimensions, digital calipers were used with multiple samples taken over the outside surface. This ensured that the tolerances between the outer and inner walls were very little and that the bulkhead's seal would seat properly. Holding six connectors of the electrical team's specification was a must. If the cap could not hold this amount of connectors safely, then the whole design was a bust. To ensure that this was possible, both the electrical lead and mechanical leads had equal say in the choice in regards to the connector diameter and length. The bulkhead's ability to keep the water at bay was an obvious driving condition; if the cap let any water in, the design was a failure. To meet and exceed this requirement, a proper radius was used in between the vertical and horizontal seating surfaces. This information was used from the Machinist's Handbook. The final driving factor was that the cap would deform so little that the connectors could maintain a properly sealed seat. This meant that the local deformation could be no greater that a quarter of the o-ring's cross sectional diameter. The connectors also needed to be tightened to the exact torque to ensure a proper bolt preload. Optimization of the rear bulkhead was also done to allow for the maximum housing space along with a million cycle fatigue life. The fatigue life of the part far exceeds the million cycle requirement and ensures that the team will not have to replace the bulkhead from a mechanical failure the entire year. The below fringe plots are from the SolidWorks simulations package and they include all of the forces acting on the cap, including the bolt preload force and the pressure at depth. The below fringe plots are from the SolidWorks simulations package and they include all of the forces acting on the cap, including the bolt preload force and the pressure at depth.



Stress fringe plot of the aluminum 6061-T6 Rear Bulkhead, Housing side

Inner Structure

Due to the size constraints of our main housing we had to get really creative for the inner structure. So that all of the electrical components and lithium polymer batteries will fit in the main husing . We first had to find largest component of all the onther electical parts that we have. By do these steps we saw that the main inner structure had to be very custom. The two options that made the most sense was CNC or 3d printing. The team compare the cost between CNC or 3d printing. The conclusion was 3d printing was a better and cheaper option for us. Do to the nature of 3d printing you are able to get very precise and complicated parts from the 3d printer. We were also to make many revisions at a very fast pace. We were also able to combine some of the electronic components into the 3d printed parts.



Acknowledgments

The team would like to thank our academic sponsors, Kevin McFall for putting up with us throughout the year. It has been a rather chaotic year for the team, but he has helped us through it and made sure we have made it as far as we have. We would also like to thank all of our sponsors Dassault Systemes, Portwell, Diver's Supply, Podio, Hobby Town USA, GitHub, MathWorks, Smyrna Recycling Center, and Maxair (see: http://spsu-auv-team.github.io/sponsors/) for providing us with equipment, time, and space in order for us to construct our vehicle.