RoboSub 2025 - Technical Design Report
Stanford University

Mechanical: Lawton Skaling*, Kai Song, Félicie Hoffmannn, Chisa Ogaki, Julian Schultz, Luca Higgins
Electrical: Vassilis Alexopoulos*, Ryota Sato, Angelina Krinos
Software: Scott Hickmann*, Rohan Bhowmik
Research: Selena Sun*, Elsa McElhinney

*indicates Sub-Team Lead

June 30th, 2025

STANFORD
ROBOSUB

Abstract

We proudly introduce our 2nd-generation platform "Crush," developed with a focus on both the competition and research goals. Regarding the RoboSub competition, we have developed precise localization, custom hydrophones, and a spring-loaded torpedo system with the goal of completing all tasks. Our design choices continue the easy iterability and robustness found on our 1st-gen platform while improving on electrical organization, autonomy capabilities, all while drawing on our research experiences.

1 Mission Statement

Last year marked the inaugural season for the Stanford RoboSub team. Our V1 submarine, "Dory," built in just over five months, proved to be remarkably robust at competition! For the 2025 season, we built upon our past success through strategy and design considerations informed by lessons learned and inspirations from interactions at last year's competition. We introduce "Crush," our V2 platform, a manifestation of our renewed competition strategy and emerging research directions.

1.1 Overall Methodology

The design and development of our AUV this season are steered by the following guiding principles:

- **Reliability-First Design:** We prioritize cost- and space-efficiency when considering sensors, actuators, and software to increase flexibility, accelerate progress, and minimize catastrophic failures.
- **Modular Organization**: Perception, localization, mission planning, control, and hardware I/O reside in self-contained ROS 2 packages with clear interfaces, allowing sub-teams to develop, test, and swap software components independently. The same deliberate modularity informs the physical build: standardized mounts and quick Molex connectors let us replace or upgrade hardware without full tear-downs.
- **Data-Driven Iteration**: All sensor and localization streams are logged during bench and pool runs; these logs feed a robust simulation interface that mirrors real-world topics, enabling repeatable testing, parameter tuning, and informed iterations.
- **Research Ambition Beyond Competition**: Upgrades are chosen not only for RoboSub scoring but also for long-term autonomous-marine research value, ensuring the platform remains useful well after the event. Utilizing Crush in real-world scenarios enabled us to learn more about sensors, robustness, and practicality in robotics.

1.2 Competition Strategy

Last year, our submarine focused solely on localization. This year, with fully functional camera feed and vision, we're expanding our efforts to include vision-based tasks. We're also developing systems for auxiliary tasks such as torpedoes, a claw, and pinger detection.

Altogether, each design decision outlined in this report derives from the need for a robust sub capable of general autonomy or from specific tasks and requirements. Specifically, the fabrication of Crush's main body, electronics, and the base software stack enables physical durability and functional resilience under anticipated scenarios; meanwhile, auxiliary systems including the gripper, hydrophone system, and task-planning/perception code are designed to aid the completion of certain competition tasks. Each section will outline design requirements tailored toward general/specific competition tasks as well as tests and corresponding observations to validate their inclusion.

2 Design Overview

"To do all the dimensions, I had to learn trigonometry, which I never got in highschool!"
- Bob Gurr

2.1 Mechanical Design

Crush's Mechanical Design was guided by three main principles: robustness, iterability, and ease of manufacturing. The entire sub is modeled in Fusion 360 Computer Aided Design (CAD). Balance distribution and buoyancy were derived from the CAD model. Crush was designed with the tools available at Stanford in mind: laser cutters (Fiber and CO2) and Fused Deposition Modeling (FDM) 3D printers.

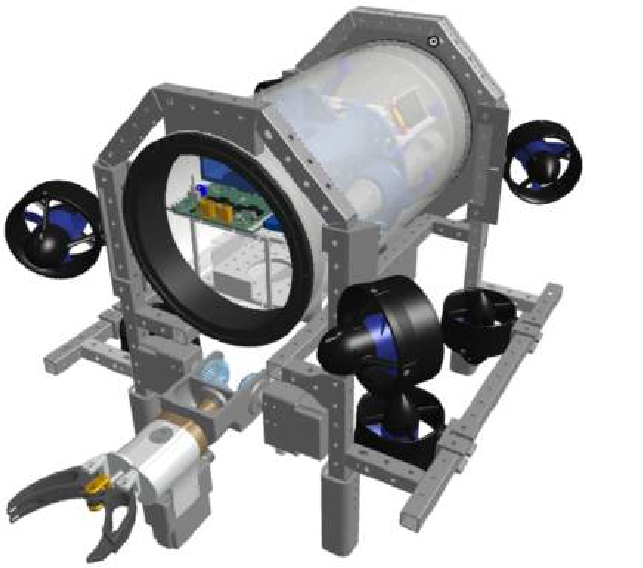


Figure 1: Crush's full CAD model. (Trimetric)

2.1.1 Frame

The frame optimizes protection and accessible mounting points for the sensors, thrusters, and pressure hull. Being stainless steel, it is resistant to salt water and durable. Our robot's research-driven focus necessitates the ability to operate in unknown environments. Lake Tahoe has strong tides and rock formations, creating a need for robust protection of important components: the pressure hull, camera tube, and Doppler Velocity Logger (DVL). Our continuous design improvements have yielded tremendous physical durability during testing and research missions.

2.1.2 Pressure Hull

Crush has one cast acrylic pressure hull to house all sensitive electronics. A clear front window allows for camera vision. The rear bulkhead, equipped with Blue Robotics Wet-Link penetrators for cables, is 9mm High-Density PolyEthylene (HDPE), which helps compensate for buoyancy balance. The pressure hull was tested to a depth of 10 meters at the Monteray Bay Aquatics Research Institute (MBARI).

2.1.3 Thrusters

Crush has eight T200 Blue Robotics thrusters. Four thrusters provide vertical height adjustment in water. Four other lateral thrusters are angled 45° to increase mobility in turning and forward movement, as T200s vary in thrust characteristics and power curves between forward and reverse states. The combination allows for complete and controlled translation and rotation within space.

2.1.4 Electronics Bay

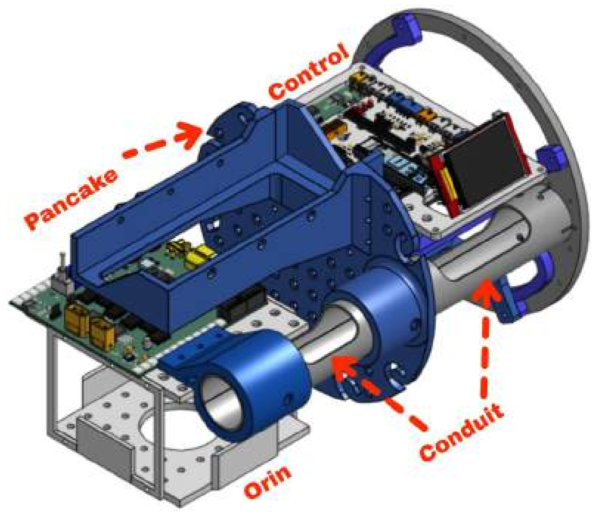


Figure 2: Electronics Bay. (Isometric)

The electronics bay is composed of three mounting modules: NVIDIA Orin, Pancake, and Control. A key feature of our electronics bay is the Conduit, a slotted sleeve that allows for neat wiring between modules while enabling Pancake rotation. These design considerations allow for compact fitting of on-board components while providing the electrical team easy access to the electronics bay, making iteration extremely easy.

All mounting modules are 3D printed on FDM printers using PolyLactic Acid (PLA) plastic, chosen for its print quality and cost-effectiveness.

All modules make use of a standard hole spacing sized for M3 heat-set inserts to incorporate threads into 3D prints. Heat-set inserts have been a big improvement for us over "Dory" in terms of iterability and ease of assembly, allowing for quick changes. Stainless steel heat-set inserts allow us to incorporate this feature into underwater mounts such as our DVL, preventing corrosion. All electronics were tested to a depth of 10 meters at MBARI, demonstrating a strong depth rating.

2.1.5 Camera Tube

A water-tight enclosure located below the main pressure hull holds the downward-facing camera and PCB for the arm and gripper. The gripper electronics are separate from the main electronics bay for heat-dissipation, space, and modularity considerations.

2.1.6 Arm and Gripper

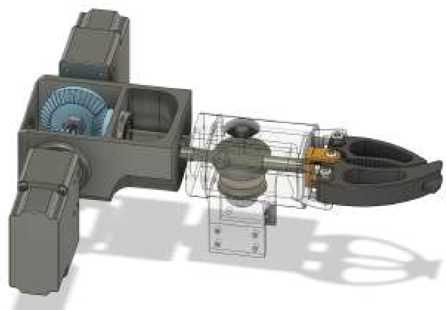


Figure 3: Gripper and Differential, Side

We designed our gripper to prioritize effectiveness, speed, cost-efficiency, and the appropriate number of degrees of freedom. It features two Blue Trail SER-20XX Underwater Servos arranged in a differential drive configuration, allowing roll and pitch. The connected claw mechanism uses another servo to convert rotational motion into linear movement, which in turn rotationaly moves a grabbing mechanism, modeled similarly to Blue Robotics' Newton Subsea Gripper. The system is encased in a custom-build with printed underwater bearings.

2.1.7 Torpedo Launcher

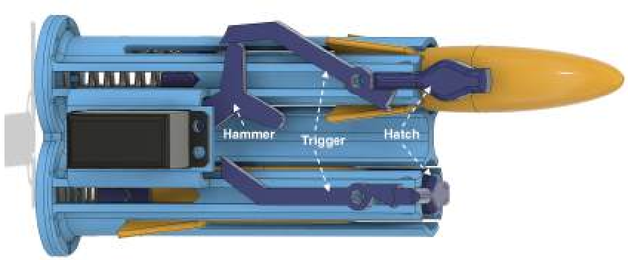


Figure 4: Torpedo Launcher, Top

The torpedo launcher balances power and simplicity. The launcher body is a single 3D printed PLA piece, for ease of assembly and manufacturing. The launcher holds two torpedoes, each spring loaded with a maximum force of 75N (17lbs). To properly contain and release this force using a single servo, there is a mechanical chain involving a hatch, trigger, and hammer, which grants mechanical advantage to the servo and prevents external radial or axial loading of the servo shaft. The torpedo is ellipsoidal with large fins to allow for stable motion across long paths.

2.2 Electrical System

Crush's electronics design for 2025 addressed last year's pain points, prioritizing mechanical integration and emphasizing the importance of future-proof engineering. We designed three custom PCBs to support the functions of Crush, all of which are publicly available on GitHub. Major challenges included handling high currents on custom hardware, implementing diagnostics for system failures, and fitting components within the space-constrained electronics bay.

2.2.1 Microcontroller Breakout PCB

The high density of signal and power lines between the microcontroller, peripheral devices, and other onboard hardware necessitated a custom four-layer PCB. This microcontroller breakout board is mounted at the rear of the sub on the Control module. Bulkhead peripherals interface with the PCB through dedicated Molex Micro-Fit connectors, providing robust electrical connections and clear pinout standardization.

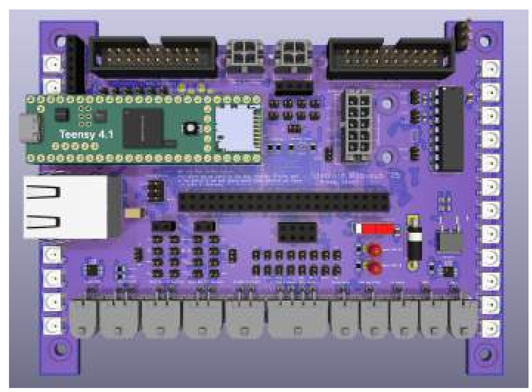


Figure 5: Microcontroller PCB, Top View

Several Molex ports are also routed through jumper headers and Dupont-style sockets. This architecture enables reassignment of I/O pins and integration of new bulkhead sensors without requiring PCB refabrication.

The Teensy is powered separately from the systems that power the Orin. This allows for full resets via an externally mounted switch, enabling recovery from firmware faults, even while Crush is submerged.

To monitor internal environmental conditions, the PCB integrates two MCP9808 temperature sensors and a BME680 environmental sensor. These are polled by the Teensy and logged to its onboard SD card for post-mission diagnostics and electronics bay monitoring. An Ethernet port allows for live monitoring of this data via a webpage that's accessible during tethered operation.

2.2.2 Power Distribution PCB

This board serves as a mounting point for the ESCs and a way to compactly distribute power to all eight thrusters. It is mounted directly under the Teensy PCB. Battery power is supplied via an XT60 connector. ESC inputs are locked in place via a set of terminal blocks. PWM connections for the ESCs are attached directly to the board and are routed to the Teensy through a ribbon cable, making the entire system tightly integrated.

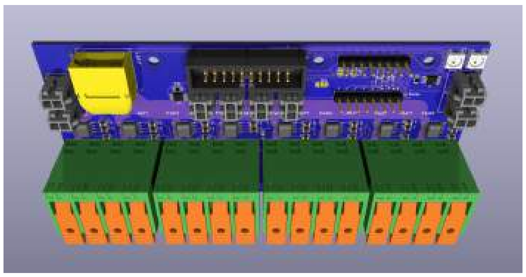


Figure 6: Power Distribution PCB, Bottom View

2.2.3 Power Management PCB

To support durational research missions exceeding one hour, we designed a dedicated power management PCB enabling expanded battery configurations, providing:

a. **Hot-swapping:** Seamless replacement of a depleted LiPo battery without power interruption or cross-charging risk.

b. **Dual-battery load negotiation:** Coordinated power draw from two batteries for extended runtime using an auxiliary enclosure.

The PCB also includes a hardware kill switch, per-battery voltage monitoring, and configurable under-voltage (UVP) and over-current (OCP) protection.

Each battery connects via a trio of low-R_{DS} N-channel Power MOSFETs, configured as high-side switches to fully isolate downstream circuitry when off. A gate driver boosts control voltage to 25V, ensuring reliable operation up to 16.8V of system voltage.

Voltage sensing is handled via resistive dividers; current is monitored in real time using a Hall-effect sensor for active OCP. All protection and switching logic is implemented in hardware (comparators, logic gates, trimmable references), ensuring fail-safe operation even if the microcontroller faults.

Arbitration logic uses the lower-voltage battery first; when it hits its UVP threshold, the system switches to the other battery. This serial (non-parallel) scheme avoids unsafe current imbalances. To protect downstream electronics, the board includes a TVS diode for spike suppression and a 20 mF capacitor bank to buffer brownouts.

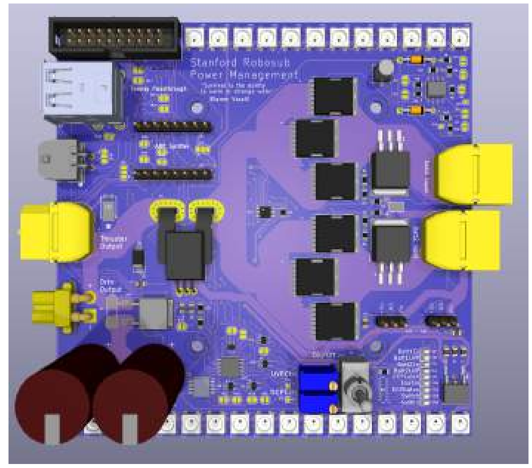


Figure 7: Power Management PCB, Top View

2.2.4 Ping Localization System

Another key endeavor this year was the development of a ping localization system. The system employs four custom-fabricated piezoelectric hydrophones, crafted with guidance from the Center for Computer Research in Music and Acoustics (CCRMA). These hydrophones are mounted at each corner of the AUV, maximizing their separation to enhance the accuracy of acoustic source localization.

The system's signal processing chain, as depicted in Figure 13, involves analog-to-digital (A/D) conversion, digital filtering, amplitude envelope detection, and internal logic, all managed by a Digital Signal Processor (DSP) chip. The system determines the precise arrival times of the acoustic ping at each hydrophone using peak amplitude tracking. These time differences are then utilized for geometric calculation of the ping source, based on Time Difference of Arrival (TDOA) principles. We have explored multiple DSP platforms, including the Analog Devices ADAU1701 (programmed with SigmaStudio), the Teensy Audio Library, and Bela chips. A significant challenge encountered across these iterations has been achieving the necessary high sampling rates (well above 90 kHz, ideally 100 kHz or more, given a 45 kHz signal) for accurate signal detection. Further research and development are ongoing to address these sampling rate requirements, potentially exploring external high-speed ADCs or more powerful dedicated DSPs.

2.3 Software Architecture

Last year, our focus was mainly on getting a submarine out in the water and being able to do basic control with some level of localization. Since then, we've learned a lot on what works well and less well and created a new software stack from these learning. This mainly boils down to better separation of concerns into separate modules, improvements to the control and localization modules, the addition of smooth path following capabilities, and the heavy use of the Gazebo simulator.

Figure 8 shows a simplified layout of our updated software stack, emphasizing its three cooporating layer.

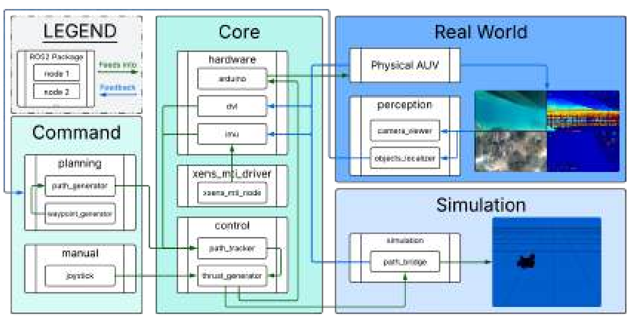


Figure 8: Simplified holistic software stack

The first layer **Command** hosts high-level decision making: a `planning` package converts mission goals into discrete waypoints then continuous paths, while a manual package exposes a joystick teleoperation fallback. These commands flow into the **Core** layer. Here, dedicated drivers interface with onboard devices in the `hardware` package (`arduino` for thrusters and sensors, `dvl` for velocity, and `imu` via the `xsens_mti_driver`). Ultimately, the `control` package (`path_tracker`, `thrust_generator`) synthesizes path following commands and DVL + IMU readings to generate thruster command for the Arduino. This interface can target either the **Real-World** vehicle or a **Simulation** environment: in hardware mode, perceptual nodes (`camera_viewer`, `objects_localizer`) powered by a fine-tuned YOLO11 model stream situational awareness back into Core, whereas in simulation a `path_bridge` mirrors those topics so the remainder of the system runs unchanged. This modular, feedback-rich design lets us switch seamlessly between joystick control, autonomous planning, real-world tests, and full-fidelity Gazebo trials without touching the control or perception code.

While the physical aspects of Crush were validated and stress-tested in MBARI, testing the software stack involves iterating isolated tests for each ROS2 package and node. Planning modules were tested by ensuring that generated paths passed through original wavepoints and remained C5 continuous in position and orientation over time (to minimize strain on thrusters), while the manual joystick was simply validated by succesfully driving the sub. Hardware components were verified by ensuring sensor readings matched expected trends (e.g., recording velocity/acceleration readings over predetermined path for IMU/DVL tests); the same was done for localization. The Gazebo simulation side was simply verified visually for consistency with expected sub motion. The simulation enables testing of the control package and PID tuning while further validating path generation visually. The validation and iteration for camera feed and perception. Validating each module through independent iterative testing informed by leveraging empirical feedback (real-world for hardware/input/perception/localization and simulation for control/planning) provided the basis for the team to incrementally build up a unified, cooperating software stack.

To build up task-clearing capabilities for the competition, our main waypoint generator is built up as an ensemble of control flow statements informed by its real-world state, including pose, linear and angular velocities, acceleration, and perceived objects through camera view. Based on these inputs, the generator conditionally publishes mission-specific waypoints to accomplish tasks such as object approach, alignment, and traversal. For instance, when passing under a gate, the mission planner stores whether a sawfish or shark was detected, ensuring full detection by coming to a stop to run the YOLO pipeline. This control-flox structure allows for context-aware waypoint generation without hardcoding trajectories, enabling adaptive autonomy diverse competition scenarios.

2.4 Research Considerations

One of our major priorities this year was to optimize Crush's research applications. This initiative was driven by conversations with researchers at several leading marine and freshwater research institutes including MBARI, Hopkins Marine Station, and Tahoe Environmental Research Center. These discussions revealed a widespread need for more flexible, fine-scale tools that can support critical aquatic ecosystem research.

Across disciplines, researchers emphasized the importance of acquiring granular, spatially referenced environmental data (1). One major challenge researchers identified was monitoring the impacts of climate change, such as temperature stratification, oxygen depletion, eutrophication, and salinity shifts that alter aquatic habitat suitability. Another important task was detecting invasive species and other ecological stressors, demanding sensitive, repeatable, and spatially comprehensive biological sampling methods (2).

Crush is designed to directly address these needs by functioning as a mobile platform that combines advanced sensing, modular sampling, and precise localization. By facilitating repeatable, high-resolution data collection across both abiotic and biotic parameters, Crush aims to augment and extend traditional research methods, such as diver surveys, in-person sampling, and stationary monitoring stations.

- **3D Reconstruction**: To enable low-cost, frequent imaging of marine terrain, Crush uses a $300 OakD-S2 stereo vision camera to produce an accurate 3D map of the scene. We use NVIDIA's Instant-NGP software for map generation, which we run on the edge using our Nvidia Jetson Orin. This enables researchers to understand the accuracy and fidelity of the 3D map in real-time.

  Affordable, high-precision underwater terrain mapping fundamentally transforms how researchers monitor and understand dynamic ocean ecosystems. Democratizing environmental surveying enables scientists to conduct monitoring at the temporal resolution that marine conservation, evidence-based policy development, and cutting-edge research truly demand, rather than being limited by budgetary constraints in the thousands.

- **eDNA Collection for Invasive Species Identification**: A key addition in Crush's design is its cheap, modular environmental DNA (eDNA) collection (Figure 9). eDNA research works by collecting traces of DNA that all organisms naturally shed into their environment and capturing them in a membrane filter. DNA from these samples is then extracted and analyzed using quantitative PCR (qPCR), which amplifies target genetic markers to determine which species are present and in what abundance (3).

Figure 9: eDNA Collector

  Crush's design adapts passive eDNA samplers originally developed for use in trawling nets (4). These probes can be easily swapped depending on the deployment environment and are isolated to minimize cross-contamination between sites. Adding eDNA collection capacities allows us to test the efficacy of this new passive sampling method, expand its application in marine robotics, and generate species profiles from diverse aquatic ecosystems.

- **Environmental Sensing**: Crush is outfitted with temperature and salinity sensors that continuously record environmental conditions during operation. This data provides important context for researchers' experiments, and with our robot localization, this data can be easily displayed in a 3D visualization to track temperature and salinity gradients in bodies of water (Figure 15). This will help scientists answer a variety of research questions, such as: Why do phytoplankton cluster in specific areas? How are marine ecosystems responding to warming and changing precipitation patterns?

By integrating NeRF-based 3D mapping with environmental DNA collection and comprehensive sensing capabilities, Crush empowers aquatic researchers to conduct unprecedented high-resolution environmental assessments.

3 Acknowledgments

Stanford RoboSub's fast development and accomplishments would not have been attained without the supportive collaboration from our gracious community of supporters. Starting with our internal support mechanisms at Stanford, we wholeheartedly thank

- **Professor Oussama Khatib** — for his technical guidance, support in providing course units, and enthusiasm
- **Stanford Student Robotics** — who provide us a platform, leadership, funding, and wisdom to keep our team running.
- **Lab64** — thanks to Jeff Stribs and his CAs for providing a lab space and access to 3D printers, laser cutters, storage, and a machine shop.
- **AOERC Avery Recreational Pool** — for providing us with lane time for weekly testing.
- **Biome** — for providing lab space and resources for our research initiatives.
- **CCRMA** — for providing hydrophone resources and help in the ping localization system.

On the donor and external support side, we have been incredibly lucky to receive collaboration, test spaces, advice, and electronics from

- **Star Oddi** — for graciously discounted sensors for further water-column research.
- **Teledyne Marine** — for helping us acquire their Wayfinder Doppler Velocity Logger.
- **Blue Trail** — for providing help accessing their 3 quality Servos for our Gripper system.
- **Monterey Bay Aquatic Research Institute**(MBARI) - especially Jason Adelaars, for allowing us to use the facilities for subassembly and full assembly tests.
- **UC Davis** — especially Brandon Berry, Grace Rosburg-Francot, and the rest of the Tahoe Environmental Research Center at UCD for providing their time, expertise, and resources for future research endeavors.

References

[1] B. Berry, "Personal interview," 2025. Accessed: 2025-05-01.

[2] G. Rosburg-Francot, "Personal interview," 2025. Accessed: 2025-05-14.

[3] M. S. R. Krista M. Ruppert, Richard J. Kline, "Past, present, and future perspectives of environmental dna (edna) metabarcoding: A systematic review in methods, monitoring, and applications of global edna." https://www.sciencedirect.com/science/article/pii/S2351989418303500, 2019.

[4] G. Maiello, "Metaprobe-2.1 github." `https://github.com/GiuliaMaiello/Metaprobe-2.1`, 2024. Accessed: 2025-06-30.

4 Appendix

Six pages can contain only the tip of Crush's metaphorical iceberg. Hence, we have compiled any continuations, formulae, and wisdom in this appendix with the intention of explaining our design and methodology beyond the rubric to other curious teams.

A Mechanical

A lot of detail relating to design choices and manufacturing processes could not be covered in the main mechanical section. They are detailed below. This is meant to be both a guide for incoming team members, a resource for other teams, and a record of our learnings.

A good resource to follow along with (and read it entirely!) is FTC's gm0. gm0 is a student written, student managed website for fundamental, yet in-depth, knowledge and skills in robotics. While it is oriented for FTC, it covers things that are so fundamental for all robotics.

I will begin with some philosophies and lessons learned:

1) Do it Right. The First Time.

> **"Just do it right...Every detail of everything you do, make sure you're doing it in the best possible way that you can. The choices a person makes, whether it's legal work, medical work, science, cartoons or Disneyland vehicles, do it your absolute best. That will make sure you'll be much more successful than if you just clock in and out to do the job and go home." - Bob Gurr**

Why is this important? Small mistakes compound. If you poorly tolerance something, the whole assembly will be even worsely toleranced. If you don't add cutouts for electrical wiring, you will have to drill/Dremel that cutout later. You're lazy and don't bother looking up the right drill-press speed for steel? You'll break the drill bit (or the drill-press!) When Boeing designed the 737 MAX, they didn't follow the industry standards of redundancy with regards to Angle-of-Attack (AOA) sensors. They took shortcuts to get their competitor to Airbus's A320NEO into production. Crash victims and families paid the price. Don't do shoddy work. Do it right. The First Time. By following this philosophy, the end result is better, you learn a tremendous amount, and you can take pride in your work.

2) **Curiosity & Experience:** As an engineer, being curious, especially in college, is essential. You should know what other sub-teams are working on to stay on track. Basic knowledge of most systems enhances your designs. You can't design electrical mounts if you don't know why an IMU moving is a problem or why electrical team needs standoffs for PCBs. You will never learn new manufacturing processes, better ways to CAD, or different components to make your life **easier**.

"There's simply no substitute for experience in terms of aviation safety" - Capt. Chesley Sullenberger

That quote applies to aviation safety, but also to engineering. You grow as an engineer by having experience. Doing over and over teaches you better ways to design. This goes hand-in-hand with curiosity: only through curiosity and passion are you pushed to do more and do better each time. Furthermore, being curious leads you to ask questions. Sometimes, the current design is not good, and you should question it. If you don't know why something is done the way it is, ask. If you need help or advice or feedback, ask. A good friend of mine once said: "God helps only those who help themselves." Finally, if you do something poorly, very likely someone else has to fix your mistake. And that is not fun.

3) **Blue Sky!** At the start of the brainstorming process should come Blue Sky. Blue Sky is central to Disney Imagineering's design process. Essentially, instead of going into brainstorming with one fixed mindset, have the whole team (in stages, from all subteams to more focused conversations) brainstorm with an open mind. Something may be good, but always ask: what if...? Conversations before actual CAD generates good ideas, many of which a single person alone would never have conceived. Once the team has a vision, it is so much easier to execute. As Woodie Flowers said: **"When you see your target, your aim is perfect."**

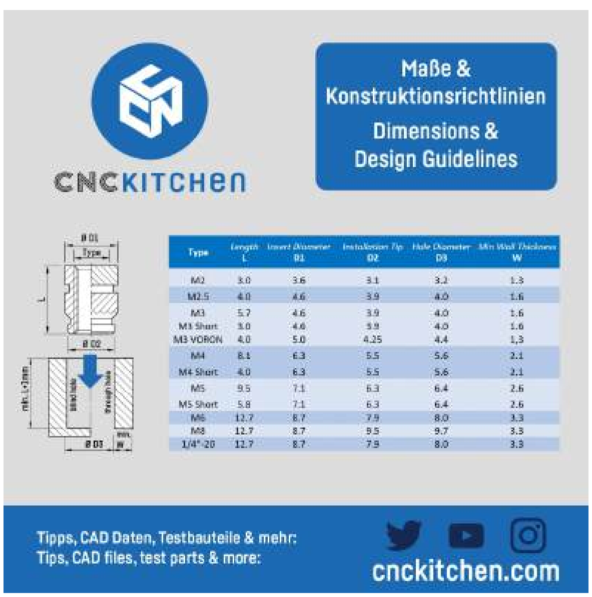
4) **Iterability:** One of our core design philosophies is iterability. Standardized mounting holes play a big role in that; just look at goBilda and the impact it has had on FIRST Tech Challenge (FTC) teams. The frame's mounting holes came in handy when the DVL had to be shifted to accommodate the camera tube and arm. It will also be useful next year when a new sub is designed, building on the same frame but perhaps reconfigured. The same principle applies for the electronics bay.

5) **Cross-Team and Smart-Design:** Cross-team communication is so key. A MECH (robot) is: **M**echanical **E**lectrical **C**omputational **H**appiness. Nothing in robotics is designed within just one subteam; everything has cross-team implications. Talk to and meet with other teams! It makes your life easier. Smart design is a part of this. For example, when designing electronics mounts, add holes or cutouts for wires to run. Simple things like this make the end result so much better.

6) **Good CAD:** This corresponds to gm0 section titled "Design Skills."
The importance of good CAD cannot be understated. Having up-to-date CAD for all members to reference is key; having buoyancy and weight balance calculated for you makes life easier. Good CAD (generally) involves: properly defining sketches, designing in millimeters (to make 3D printing & exporting easier and because metric is superior to imperial), and for the love of God dimensioning! You will thank yourself later when your CAD does **NOT** break after you change a dimension. Good CAD is an indication of good overall mechanical design. With a project like RoboSub, where there are so many parts, small errors in CAD will cause headaches in the assembly later on.

7) **Heat-Set Inserts:** This corresponds to gm0 section titled "Custom Manufacturing."
A lot of engineering (esp. in college and high-

school) is now centered around 3D printing. Heat-set inserts are a convenient, easy way to add threads to 3D printed parts. This is good for a simple reason: eliminate nuts. Nuts have their time and place but in most situations, you're better off without them. They require 2 hands to use: one to tighten the screw, one to hold the nut. It is almost always preferable to have tapped threads (for metal) or heat-set inserts (for plastic). Additionally, nuts used on soft plastics like PLA can easily be over-tightened and sink into the PLA. Then, the screw strips. That is not fun to get out (and usually the person putting it in is **not** the one taking it out. Heat-set inserts are quite literally life-changing. They making mounting electronics so much easier. 3D printed assemblies are easier to put together. You can use a standardized chart for heat-set insert hole sizing during CAD, shown in Figure 10. They're convenient, cheap, and everyone in this world should know about them. Just yesterday (as of writing) my friend struggled to figure out how to mount a 3D printed part to a 'shaker' for vibration testing; turns out, heat set inserts were the answer, and they were amazed!



Type	Length L	Insert Diameter D1	Installation Tip D2	Hole Diameter D3	Min Wall Thickness W
M2	3.0	3.6	3.1	3.2	1.3
M2.5	4.0	4.6	3.9	4.0	1.6
M3	5.7	4.6	3.9	4.0	1.6
M3 Short	3.0	4.6	3.9	4.0	1.6
M3 VORON	4.0	5.0	4.25	4.4	1,3
M4	8.1	6.3	5.5	5.6	2.1
M4 Short	4.0	6.3	5.5	5.6	2.1
M5	9.5	7.1	6.3	6.4	2.6
M5 Short	5.8	7.1	6.3	6.4	2.6
M6	12.7	8.7	7.9	8.0	3.3
M8	12.7	8.7	9.5	9.7	3.3
1/4"-20	12.7	8.7	7.9	8.0	3.3

Figure 10: Heat-set Inserts Sizing Chart. Zoom in!

8) Bolts, Nuts, Washers, Through-Holes, Sizing, Assembly, Electric Tools and Character-Building, Bearings: This entire section is meant to be a basic review of mechanical engineering basics, aka how to put things together **properly**. This corresponds to gm0 section "Hardware Components."

Bolts are meant to be screwed into threads. There are special 'self-tapping' bolts for plastic, wood, and sheet metal; however, general bolts are not meant for this. Repeatedly tapping into plastic is not good; threads (heat-set inserts for plastic) should be used for this purpose. Please, always use metric bolts. Metric bolts are easier and you know exactly what the bolt diameter is. The number corresponds to diameter, in mm. ie M4 is a 4mm diameter bolt. M2.5 is a 2.5mm diameter bolt. Easy! Comparatively, I have no idea what the diameter of a #10 screw is. There are a couple basic types of bolts used commonly in robotics: socket-head, button-head, countersunk, shoulder, and machine-head. Socket-heads have a socket, with a cylindrical head. Button-heads have a half-size smaller socket than the same socket-head bolt. ie. for an M4 socket-head, the socket size is 3mm. For an M4 button-head, it is 2.5mm. Button-heads are used when more surface area is needed (ie. for screwing into plastic). Countersunk (or "flatheads") bolts have a head that is angled. They are meant to go into a hole that is countersunk (has the corresponding angled cutout.) Countersunk bolts sit flush, which saves space and looks beautiful. Shoulder bolts are screws which do not have threads along the entire length. Instead, a portion is a "shoulder" aka a smooth rod. The shoulder is the same diameter as the bolt. So for M5 the shoulder would be 5mm. Shoulder bolts should be used for applications where the shoulder will sit in a bearing, etc. Machine-heads are those with not sockets. You have to use a wrench or socket holder. Reference Figure 11.

Nuts are not preferred. There are some instances where they are unavoidable, but generally, tapping threads into metal or heat-set inserts is better. Nuts are more complex to assemble, they take up valuable space, and they do not hold as well as Loctite does. If nuts are needed, you almost always want Lock-Nuts. These have a small ring of nylon that prevents vibrations from loosening the nut.

Through holes are meant to prevent a very basic problem: screwing together two things that have threads. You should **never**, I repeat, **never** screw together two things with threads. It should always be one thread per bolt. Everything sandwiched in-between should be through holes. Otherwise, you will never get a tight connection no matter how hard you try. Same goes for screws that too long, which end up hitting something (usually t-slot rail). If the bolt is too long and hits something else, you can turn as much as you want, but it will never fully tighten.

On plastic parts, washers should always be used. This is especially true for 3D printed parts (PLA, PETG, TPU). Directly tightening a bolt (especially a socket-head) will cause it to sink into the plastic, making it overly-tight. This is incredibly annoying to remove when the time comes. If (you haven't read the nut section) there is a nut, that nut will sink into the plastic; you literally have to work so hard to get the screw out at that point. Please save your friends time and brain cells!

Metric sizing is easy! The number behind the "M" is the diameter, in millimeters, of the bolt diameter! Nice and simple. Do not use imperial; it makes no sense and no one knows what diameter a #8 screw is. No one wants to know either.

How should we properly assemble things? First, when bolting components together, always **1)** insert and partially tighten all screws; **2)** fully tighten all screws. We first partially tighten to ensure all screws are able to be inserted; if you fully fully tighten a single screw first, then you won't be able to get other screws in. When fully tightening, it is good to follow the standard tightening pattern, which ensures even load distribution. For robotics, bolts are generally found in groups of 4. Do opposing corners first, then the other opposing corners. **Do not go in a circle!** That does not distribute load nor align components correctly. Refer to Figure 12.

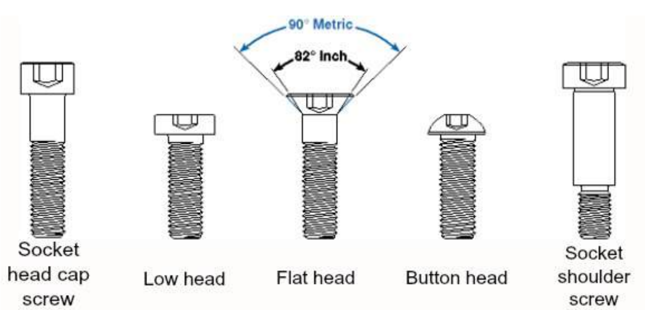


Figure 11: Standard types of bolts.

People generally don't have a good grasp of when to stop tightening. Tightening "as much as you can" is **not the way to do it.** You can easily over-tighten, especially into plastic. Hand-tightening using a screwdriver or hex-key is the best way to learn; for applications such as robotics, 3D printer assembly, etc hand-tightening should always be used. It is always sufficient, and there is minimal chance of over-tightening (unless you're going crazy). Screwing things in by hand, especially when learning about mechatronics, builds character; you should never consider it beneath you to be hand-screwing bolts in. The same goes for using hand-tools when needed over complex machines like lasers. The same also goes for considering it everyone's duty to clean up, organize, and sweep floors. An electric-screwdriver should **never** be used for small applications like robotics. You save no time and they over-tightens bolts (especially since many don't have torque limiting). Over-tightened screws often mean stripped screws. Some unfortunate soul then has to spend 15 minutes trying to remove that screw, but can't because it is stripped so badly it belongs in the trash. Also, please throw stripped screws away; they are a negative benefit.

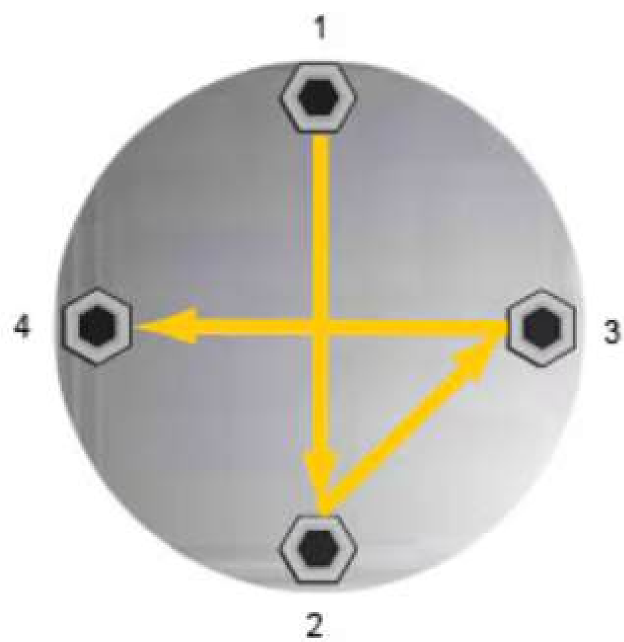


Figure 12: Standard 4 bolts tightening pattern.

Now follows specific technical details about our sub.

Frame and Conduit: The frame was laser-cut on a Fablight 4500 fiber laser out of 1.5mm stainless steel square tubing and Tungsten Inert Gas (TIG) welded together. A jig was constructed out of steel, with pins that fit into the screw holes on the frame tubing, to prevent the frame from warping during welding. A similar jig was constructed for the conduit, which was also laser cut and welded. There was slight warping noted in the frame, but not significant amounts; the jig seems absolutely essential to that.

Electronics Bay: The conduit actually acts as a structural element for the Pancake and Orin modules. It is mounted directly to the pressure hull flange. We use special shouldered set-screws and cutouts in the conduit to put mechanical limits for rotation. We want all modules to remain level during normal operation, and the set-screws allow for this.

Custom PCBs really help with space saving in the electronics bay. Even having smaller breakout boards wired via traces really saves space and reduces clutter. Electronics mounts are almost exclusively 3D printed on our team. This provides one key advantage: quick, cheap iterability. From CAD, we can quickly reprint a mount if we design a better one or find that an older version doesn't fit. That has already happened multiple times this year.

Next year, high-power and low-power PCBs should be located separately. This means: battery voltage and high-current components should be near each other to prevent running thick wires across the electronics bay. This year, this was not the case. Better mounting solutions for the ESCs should also be developed. Currently, they are not securely attached to anything and have no proper strain relief. Additionally, having a giant pancake in the center of the electronics bay has been a big impedance to wire-management. The key issue with this is that the conduit does not have enough space for all wires.

Camera Tube: The tube is secured onto the two center metal frames using PLA clamp rings with stainless steel heat-set inserts.

B Electrical

B.1 Electrical Component Selection

Battery: Crush runs on a Lumenier 13000mAh 4S 20C Lithium Polymer battery. The four-cell (4S) configuration provides a voltage range of 12.8V to 16.8V, which is well within the acceptable limits for both our thrusters and compute stack, eliminating the need for high-current buck or boost converters. The 20C discharge rating ensures sufficient burst current to power all eight thrusters simultaneously during complex maneuvers without throttling. Meanwhile, the 13,000mAh capacity provides approximately 40 minutes of runtime under a continuous 20A load during typical low-intensity maneuvers.

Thrusters and ESCs: Eight T200 Blue Robotics thrusters power Crush. Each thruster is paired with its own Blue Robotics electronic speed controller (ESC), which converts DC power from the battery into three-phase AC to drive the motors. Since the ESCs draw power directly from the LiPo battery—and thrust output from the T200 decays with input voltage—we compensate each PWM control signal based on the real-time battery voltage to ensure consistent thrust.

Low-Level Control: A Teensy 4.1 microcontroller handles time-critical and hardware-facing operations on Crush. This includes generating PWM signals for all ESCs, polling and processing data from non-critical sensors, interfacing with bulkhead components, and managing the internal electronics bay display. These tasks require deterministic timing and low-latency response, making the Teensy an ideal low-level controller. Communication with the high-level system is handled over a high-speed serial link, enabling clean separation between real-time control and mission logic.

High-Level Control: A Jetson Orin AGX serves as the central processor for autonomy and mission-level decision-making. It executes perception, navigation, and planning algorithms—leveraging onboard GPU acceleration for real-time computer vision and sensor fusion. By offloading hardware control to the Teensy, the Orin can focus exclusively on high-level tasks without being burdened by low-level constraints.

DVL: Teledyne's compact Wayfinder Doppler Velocity Log (DVL) met our constraints on weight, cost, and performance. Its 60-meter bottom-tracking range is more than sufficient for the depth profiles anticipated in Crush's research missions.

IMU: Movella's high-precision Xsens MTi-610 Inertial Measurement Unit (IMU) provides accurate real-time estimates of Crush's position and orientation in 3D space. To reduce drift over time, IMU outputs are fused with velocity data from the DVL and depth readings from a bulkhead-mounted BlueRobotics Bar02 depth sensor.

Vision: Two Luxonis Oak-D S2 cameras provide stereo and depth sensing, enabling high-resolution 3D reconstructions of underwater environments for mapping and navigation.

Connectors: We've learned the hard way that bad connectors can wreak havoc on the entire system. Robust harness engineering is key to Crush's reliability. We use XT60 and XT30 connectors for power distribution, Molex Micro-Fit connectors for sensor interfaces (based on recommendations from the folks at Saildrone), and ribbon cables for board-to-board interconnects.

B.2 Ping Localization System Diagram

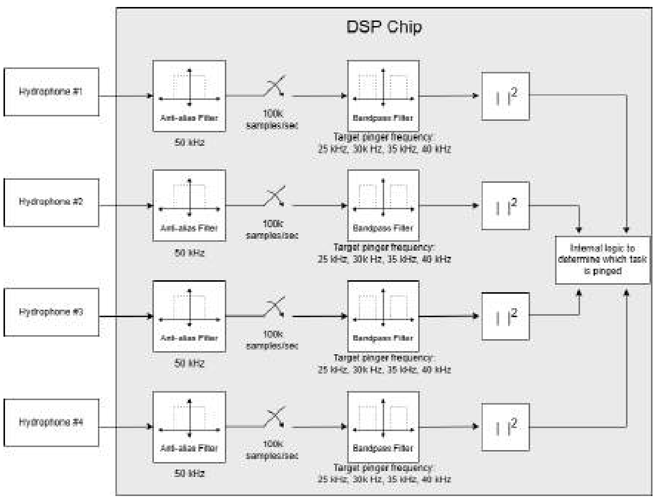


Figure 13: Ping localization system block diagram

C Software

Our entire software stack is open-source, available for reference here. The repository is a living breathing creature, meaning that code always changes and morphs; this report details the snapshots of progress up till the time of writing.

C.1 PID Tuning and Control Logic

Our control module follows standard PID: a proportional–integral–derivative controller that continuously corrects the vehicle's state to reduce the difference between the desired and actual position and orientation. The general form of the PID control law is:

$$\boldsymbol{u}(t) = K_P \boldsymbol{e}(t) + K_I \int_0^t \boldsymbol{e}(\tau)\, d\tau + K_D \frac{d}{dt} \boldsymbol{e}(t)$$

where $\boldsymbol{u}(t)$ is the control wrench (force/torque), and $\boldsymbol{e}(t)$ is the full 6-DOF state error consisting of position and orientation.

To compute orientation error, we represent rotation as a unit quaternion and define error as the axis-angle residual between the current orientation $\mathbf{q}_{\text{cur}}$ and desired orientation $\mathbf{q}_{\text{ref}}$:

$$\boldsymbol{e}_\theta = 2 \cdot \text{vec}\left(\mathbf{q}_{\text{ref}} \cdot \mathbf{q}_{\text{cur}}^{-1}\right)$$

Here, $\text{vec}(\cdot)$ extracts the vector (imaginary) part of the quaternion representing the shortest rotation from current to desired attitude. This is then fed into the PID loop alongside positional error.

C.2 Path Generation from Waypoints

To produce feasible and dynamically consistent trajectories from sparse waypoints, we implement a trajectory generator that fits a **C^5-continuous spline** through both position and orientation. This generator allows downstream control policies to compute smooth velocity and acceleration references, avoiding abrupt changes that would strain the vehicle's actuators or violate underwater stability constraints.

Given a discrete set of N waypoints $\{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^N$ with associated orientation quaternions $\{\mathbf{q}_i \in \text{SO(3)}\}$, we fit a quintic B-spline $s(t)$ for position:

$$\boldsymbol{p}(t) = \text{B-Spline}_5\left(t; \{\boldsymbol{p}_i\}, \text{C}^5 \text{ boundary conditions}\right)$$

We enforce that first and second derivatives are zero: $\boldsymbol{p}'(0) = \boldsymbol{p}''(0) = \boldsymbol{p}'(T) = \boldsymbol{p}''(T) = \mathbf{0}$, ensuring continuity up to jerk for underwater control

To interpolate orientation over SO(3), we convert each quaternion into a rotation vector (logarithmic map):

$$\boldsymbol{\theta}_i = \log(\mathbf{q}_i) \in \mathbb{R}^3$$

Then we fit independent quintic splines for each component $\theta_x(t), \theta_y(t), \theta_z(t)$ to produce a smooth time-varying rotation vector $\boldsymbol{\theta}(t)$:

$$\boldsymbol{\theta}(t) = \begin{bmatrix} \theta_x(t) \\ \theta_y(t) \\ \theta_z(t) \end{bmatrix}, \quad \mathbf{q}(t) = \exp\left(\boldsymbol{\theta}(t)\right)$$

This results in a smooth rotation over time with consistent angular velocity and acceleration:

$$\boldsymbol{\omega}(t) = \frac{d}{dt}\boldsymbol{\theta}(t), \quad \boldsymbol{\alpha}(t) = \frac{d^2}{dt^2}\boldsymbol{\theta}(t)$$

The C^5 continuity of both position and orientation guarantees that the trajectory remains differentiable up to acceleration, which is essential for generating wrench commands using PID or model-based controllers.

This method also supports waypoint closure (looping back to the first point) and allows visual confirmation of generated trajectories before deployment via animated overlays, improving confidence during both simulation testing and real-world execution.

D Research

D.1 Instant-NGP and NeRFs

Instant-NGP is built on top of the Neural Radiance Field (NeRF) algorithm, which fuses 2D images and camera poses to produce a 3D continuous map, where each point contains a color and density value. A NeRF represents a scene as a continuous 5D function:

$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma) \tag{1}$$

where:

- F_Θ represents the function, with neural network parameters Θ
- $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ is a 3D spatial location
- $\mathbf{d} = (\theta, \phi)$ is a 2D viewing direction (typically parameterized as a unit vector)
- $\mathbf{c} = (r, g, b) \in [0, 1]^3$ is the emitted color
- $\sigma \in \mathbb{R}^+$ is the volume density

D.2 eDNA Collection for Invasive Species Identification

Metaprobes are small (4 inch) hollow balls created to house rolls of medical gauze which acts as a low-cost membrane filters to capture eDNA in aquatic environments. The Metaprobe design was printed from CAD released in the open source project created by Maiello et al. (4).

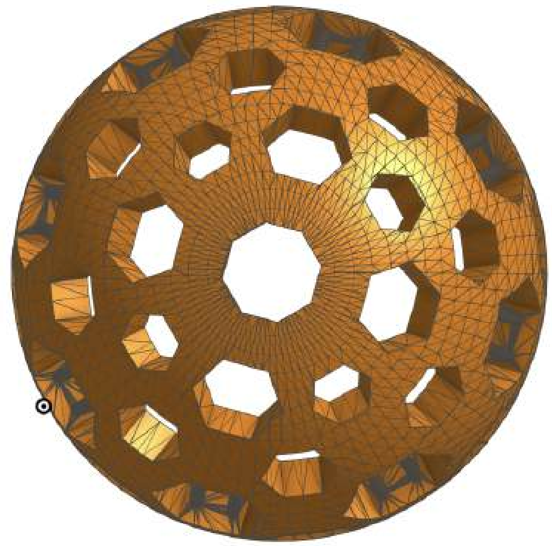
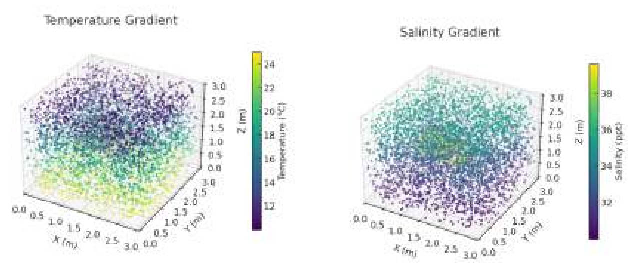


Figure 14: Metaprobe CAD.

D.3 Environmental Sensing

Keeping a figure here from above reference:



(a) Temperature gradient (b) Salinity gradient

Figure 15: Sample Gradients