RoboSub 2025 Cabrillo Robotics Club Technical Design Report

Liam Gilligan, Stephanie L'Heureux, James Truong, Isaac Wax, Fox Williams

***Abstract*—This paper details the design of Cabrillo Robotics Club's (CRC) first Autonomous Underwater Vehicle (AUV), LazerShark. As a new competitor in the RoboSub competition, CRC focused on building an AUV that balances functionality and simplicity—capable of meeting competition goals, yet feasible to develop within a year by a five-member student team. Key developments include LazerShark's vehicle chassis, custom tooling for RoboSub 2025 tasks, electrical stackup, and advanced navigation, mapping, and control software. Each component was designed with competition objectives in mind and was tested rigorously using various strategies to ensure reliability.**

I. Introduction

Cabrillo Robotics Club (CRC) is a student-led organization from Cabrillo Community College in Aptos, California. Over the past three years, CRC has focused on developing Underwater Remotely Operated Vehicles (ROVs), completing three successful projects. Building on this foundation, the team chose to expand into autonomous systems by creating its first-ever Autonomous Underwater Vehicle (AUV)–LazerShark. CRC makes an effort to create custom, in-house solutions whenever possible. As a result, LazerShark features many custom mechanical, electrical, and software systems tailored specifically for LazerShark's debut at RoboSub 2025.

II. Competition Strategy

With a year to design, build, and test LazerShark, CRC initially focused on creating a minimum viable product (MVP). In the planning stage, CRC defined the MVP as an AUV which could estimate its pose, detect objects and their relative locations, and navigate to relative coordinates. The team set a single competition goal: successfully completing the one required task of navigating through the start gates. This functionality represented the minimum capability for basic autonomy. LazerShark was built around these MVP objectives, but always with a modular design in mind to support future expansion. Once the MVP was achieved, new goals were established by evaluating tradeoffs in time, resources, difficulty, and expected return.

While most team members had experience in underwater robotics through CRC's previous ROV projects, a completely autonomous vehicle with onboard batteries was a new frontier. The team predicted the autonomous element would introduce many new design, integration, and debugging challenges within all subsystems. Anticipating these hurdles, the team set realistic goals of an MVP and utilized prior ROV experience. Previously designed ROV systems were reimagined for autonomy whenever possible. Drawing upon CRC's history in ROVs allowed for incorporation of proven components and approaches to reduce new failure points and development time, leaving most of the efforts to focus on components unique to autonomy.

A. Competition Task Selection

CRC's MVP was a vehicle which could complete *Task 1 - Collecting Data (Gate)*. Completing the Gate task is not only essential to the Autonomy Challenge, but also represents basic autonomous functionality. Once LazerShark could reliably complete this task, course goals were expanded. Additional competition goals were added incrementally and selected based on ability to integrate required additional functionality with minimal changes to existing systems. Tasks which met these criteria were prioritized over tasks with potentially higher point values that would require major additions and changes. This strategy was adopted because increasing system complexity, especially rapidly, heightens risk of introducing failure points which can propagate to already stable systems. The only task which was an exception to this rule was *Task 6 - Return Home* which was given a higher priority due to the practical importance of vehicle retrieval.

Task 5 - Ocean Cleanup (Octagon) subtask of

#	Task	Additional Capabilities
1	Collecting Data (Gate)	MVP
2	Heading Out (Coin Flip)	Additions to mission planner
3	Navigate the Channel (Slalom)	Additions to mission planner
4	Return Home	SLAM, refined path planner (not point-to-point navigation)
5	Ocean Cleanup (Octagon) - Surface only	Additions to mission planner, upward camera
6	Drop a BRUVS (Bin)	Additions to mission planner, downward camera, dropper tool, droppers
7	Tagging (Torpedoes)	Additions to mission planner, torpedo launcher, torpedos

TABLE I: Task Prioritization

moving the samples to baskets was excluded from CRC's objectives. An autonomous claw was deemed not feasible within the constrained timeline.

B. Competition Task Execution

1) *Heading Out (Coin Flip):* At the start of the run, CRC will request a coin flip. LazerShark will use its object detection system to determine if it is facing the gate. If not, the vehicle will reverse slightly to avoid hitting the side of the pool and spin clockwise until it visually identifies the gate.
2) *Collecting Data (Gate):* LazerShark will navigate to the gate and select the Reef Shark as its marine animal by passing below the Reef Shark image. CRC will obtain style points by rolling and spinning (roll).
3) *Navigate the Channel (Slalom):* After the vehicle identifies the channel, it will position between the left-most and center poles. It will determine the center of the next left and middle pole and move to that position using point-to-point navigation. The process repeats for the last set of poles. CV and stabilization will be used to ensure the vehicle remains within the pipe area, not above or below.
4) *Drop a BRUVS (Bin):* Once the bin is located, LazerShark will use its navigation to move to the task location. The downward camera will assist in positioning the vehicle directly above the Reef Shark and use the dropper tool to release two markers into the bin.
5) *Tagging (Torpedoes):* Using the front facing-camera, the LazerShark will place itself at specific offsets determined during testing to fire the torpedoes, aiming for maximum points.
6) *Ocean Cleanup (Octagon):* CRC only plans to complete the surface portion of the octagon task. This task is executed last because of the danger of surfacing outside of the designated area and preemptively ending the run.
7) *Return Home:* Return Home: Using the map of the competition area that is built, the LazerShark will use a modified path planning algorithm to find a path back to the starting area.

III. Design Strategy

A. Mechanical Subsystems

The vehicle measures about 920mm in length, 700mm in width, and 300mm in height. It has consists of two main components: the chassis and the electrical box. The placement of subassemblies was carefully chosen to optimize the center of mass (COM) and center of buoyancy (COB). These two points are aligned along the longitudinal and lateral axes but are slightly offset vertically, providing weak passive stability that is easily overcome during maneuvers such as flips. The mechanical design of LazerShark prioritizes modularity and expandability. These qualities, along with rapid development cycles, were favored over producing a polished final product due to the limited team size and time constraints.

1) Chassis

The electrical box is a major structural component of the LazerShark chassis and consists of an aluminum frame built from 20×20mm aluminum extrusions. The chassis is attached to the electrical box using four FDM-printed mounting brackets, which are bolted directly to the frame. This setup provides mounting points for the thrusters, tools, and a protected location for the Water Linked A50 Doppler Velocity Logger (DVL). The DVL is mounted so that the frame does not interfere with acoustic propagation, while still remaining as protected as possible. The frame is highly customizable and allows for easy adjustments to support future tooling. LazerShark's weight is balanced by the buoyancy of the internal volume of the electrical box. For fine buoyancy tuning, FDM-printed cartridges filled with lead shot can be added to or removed from unused interior spaces in the electrical box.

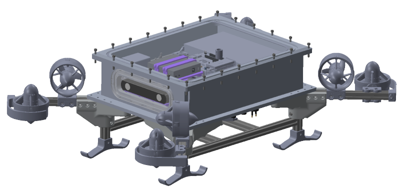

Fig. 1: Mechanical CAD

2) Electrical Box

The electrical box houses the complete LazerShark electrical stack. It is constructed from machined MIC 6 aluminum plates that are TIG welded together. Plates requiring O-ring seals are 0.5in thick, while the remaining structural panels are 0.25in thick. The design allows for compact and efficient stacking of electronics. The aluminum enclosure provides excellent thermal conductivity, helping to prevent overheating and pressure buildup. Batteries are mounted above the base plate, with a fan-assisted convection system used to dissipate heat. Switching power supplies are thermally sunk to the bottom of the box via conduction. The internal chassis was designed for easy access to all components. The enclosure includes two polycarbonate windows: a top window and a front window for the SterioLabs ZED 2i camera. These windows are secured with bolts and sealed with O-rings sized according to parameters from the Parker O-Ring Handbook [1]. The windows are cut from 10mm polycarbonate sheets, and their thickness was selected based on the maximum expected pressure on the top window. The aspect ratio of the box was chosen to minimize the cross-sectional area in the X-Z and Y-Z planes, reducing drag and, more importantly, the added mass during translation in the X-Y plane [2] [3].

3) Tooling

All of LazerShark's actuators utilize a standard, FDM printed rack-and-pinion mechanism driven by continuous-rotation 20kg servos. These servos are internally waterproofed by displacing air with WD-40 and sealing the housing with epoxy. The marker dropper cartridge system is capable of holding up to three markers, each of which is deployed by linear actuation through a rack-and-pinion-driven plunger. Markers are constructed from FDM-printed shells ballasted with lead shot. The torpedo subsystem employs a passive elastic launch mechanism using pre-tensioned rubber bands, released via a servo-actuated trigger linkage. All mechanical components were designed for modularity, ease of maintenance, and robustness in an underwater environment.

B. Electrical Subsystems

The electrical systems of LazerShark are responsible for power delivery and distribution, system control, environmental monitoring, and embedded telemetry. These functions are supported by several custom circuit boards and carefully selected off-the-shelf components.

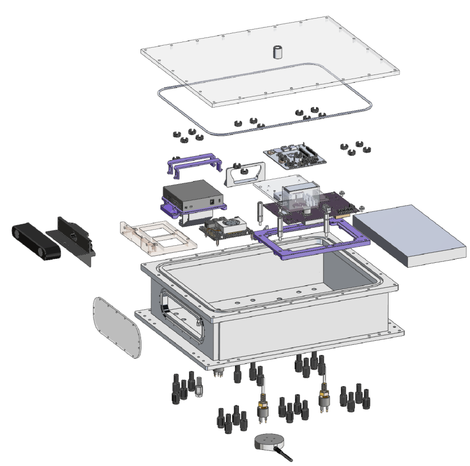

Fig. 2: Electronics box exploded view

1) Power Systems

Power is supplied by an off-the-shelf 340Wh, 43V lithium-ion battery pack. This pack connects to the Battery Management System (BMS) board, which handles all hardware-level control and protection. The BMS board includes reverse polarity protection using P-channel FETs and clamping diodes. It also features a dual power path system, allowing seamless switching between external bench power and battery power without needing to power cycle the system. After running through these, the 43V bus is passed through a current shunt IC, which provides measurements of energy consumption, power draw, current draw, and voltage level of the battery packs over an I2C interface [4]. To protect the batteries, LazerShark goes into a low power error state if the energy consumed from the battery is above 300Wh or the nominal voltage of the battery is under 40V. LazerShark has an absolute max power draw of 260W, meaning that in the most extreme circumstances, LazerShark could run safely for about an hour without the need to swap packs, but under normal operation, LazerShark can run on a single pack for about 2.5 hours. After passing through the protection and power path circuitry, the 43V input is stepped down via two 12V 200W buck converters. One converter supplies a clean 12V bus used for critical control electronics, while the other provides a dirty 12V bus to provide power to potentially dangerous systems such as thrusters and actuators. The clean 12V bus remains active during kill switch events, and the dirty 12V bus is fully disabled. A downstream 5V buck converter is also supplied from the clean 12V bus and provides both clean and dirty 5V outputs.

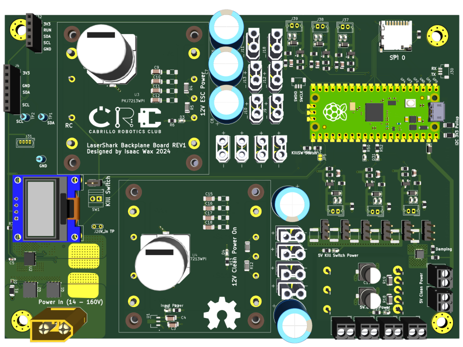

Fig. 3: BMS board

2) System Control and Interfaces

An RP2040 microcontroller on the BMS board drives universal 12V and 5V switching circuits and PWM generation for servos. It communicates with a second RP2040 on the Pi HAT using a custom UART protocol, sending environmental statuses such as temperature, humidity, current draw, voltage level, energy consumed since last battery swap, and kill switch status. The BMS board includes a BME280 temperature and humidity sensor to monitor the internal environment of the electrical box. An OLED display provides live sensor readouts during debugging. All telemetry data is logged to an onboard SD card over SPI to support post-fault analysis and hardware debugging.

3) Kill Switch

LazerShark's kill switch is implemented entirely in hardware for maximum reliability. A magnetic reed switch controls a logic circuit that disables all dirty power buses by turning off P-channel FETs when the magnetic read switch is in an open state. This ensures that power is fully cut from potentially dangerous systems such as thrusters and actuators while keeping power on to IMUs, DVL, Jetson Orin Nano, Raspberry Pi 4, and Ethernet Switch.

4) Pi Hat PCB

The Pi Hat is a custom legacy design developed last year for the MATE ROV. It interfaces between the RP2040 and the Raspberry Pi 4 and acts as the bridge between hardware-level electronics and the ROS network. It connects the Raspberry Pi 4 to an RP2040 over a UART protocol. The RP2040 runs a micro-ROS bridge, allowing it to control the Blue Robotis Basic ESCs with PWM and interface with the BMS board, providing direct access to the Pi Hat RP2040 from the ROS network. The Pi Hat also has several status LEDs for debugging purposes and includes switching circuits for controlling the cooling fans. The Pi Hat interfaces with the ROS network via connection to the Ethernet switch.

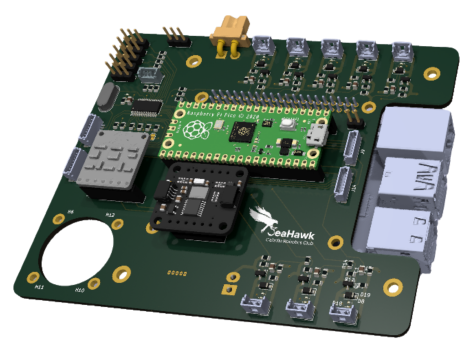

Fig. 4: Pi Hat PCB

C. Software Subsystems

1) Control System

The kinematics of the LazerShark were modeled by Thrust Allocation Matrix, which converts a twist vector into a vector of individual thruster forces. In order to compute the twist vectors needed to stabilize the AUV at a specific point, a Linear Quadratic Regulator was used. An LQR controller takes into account the dynamics of the AUV and is tuned by selecting a preference between pose error and input magnitude. The LQR was chosen over a PID controller both for performance [5] and as a means to reduce necessary tuning. An Extended Kalman Filter fuses together odometry from a Waterlinked A50 DVL, a PNI Naviguider IMU, a PNI TargetPoint-TCM IMU, and the ZED 2i camera in order to provide an accurate pose estimate.

2) Navigation

To complete *Task 6 - Return Home* and effectively navigate between tasks, LazerShark required a path planning system. Robotic path planning is fundamentally a graph theory problem of finding the shortest path, assuming an environment can be represented by a graph. Numerous algorithms could be applied to path planning and the software team did extensive research to identify the best approach. Since LazerShark will not have a predefined map of its environment, it must build the graph dynamically in real time. Therefore the selected algorithm must support incremental replanning. This constraint eliminated several common graph algorithms such as BFS, DFS, Dijkstra's and A*. Incremental graph algorithms compared include RRT, RRT*[6], D*[7], LPA*[8], and D* Lite[9]. These algorithms can be divided into Rapidly-Exploring Random Tree (RRT) algorithms and A* variants. RRT algorithms perform well in high dimensional complex environments, but produce non-reproducible, often inefficient paths which require smoothing. Paths produced by A* were found to be over 50 percent shorter than those by RRT [10]. While the AUV will need to move in 3 dimensions, from surveying the 2024 RoboSub competition area, it was determined the space was not highly complex. D* Lite was chosen over D* as it less complicated and proven to be at least as efficient [9]. Furthermore, D* Lite was also preferred over LPA* because it has more historical uses in robotics.

3) Mapping

In order to do more advanced navigation, as well as to enable localization within the pool, CRC opted to use a VSLAM system. It was decided to use NVIDIA's NvBlox system, partially due to its support with the Zed 2i camera and ROS 2. NvBlox provides a Voxel Grid representation of the observed area, which is a data structure which can be easily traversed by the D* Lite. Ultimately, one of the key factors was the extensiveness of NvBlox documentation. Many alternatives that met CRC's criteria were poorly documented and offered no functional advantages. Aside from mapping, NvBlox also allows for localization in the map frame. This bounds any drift the EKF may be producing, easing difficulties in navigation. This localization also is what enables the ability to navigate to the starting area for *Task 6 - Return Home* without having to explicitly search for it.

4) Computer Vision System

The team is planning an integration of the Zed camera's object detection in conjunction with a custom trained YOLO model. The YOLO model will provide object classifications, which will be fused with depth and pose data that will return a 3D position of the detected target object. This perception is designed to support downstream tasks such as goal recognition, obstacle avoidance, and overall autonomy. Deployment will be supported by the onboard processing of the NVIDIA Jetson platform.

5) Mission Planner API

The Control and Computer Vision subsystems were designed to easily interface with a Mission Planner. This allows developers to easily create and modify logic to complete tasks. Not only does this futureproof the LazerShark for future competitions, it significantly lowers the work it takes to debug behavior during pool testing by abstracting away stabilization and vision from the mission logic.

IV. Testing Strategy

A. *Gazebo Simulation*

As a completely new vehicle, LazerShark's hardware and software were developed in parallel. To enable software testing without a physical vehicle, CRC created a custom simulation environment using Gazebo's simulation platform. Gazebo was chosen due to its built-in features for robotics simulation and integration with ROS. The simulator models the vehicle's characteristics including weight, buoyancy, inertias, and hydrodynamics. It accepts thruster commands as input and produces outputs of simulated sensors, enabling seamless integration into the software stack. The environment allowed for continuous development and rapid testing without a physical vehicle or pool access. It proved critical for testing control systems, path planning, and high level mission planning.

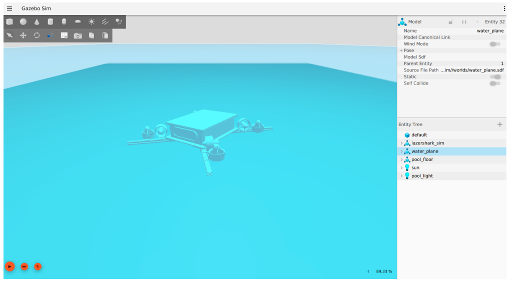

Fig. 5: Gazebo simulation

B. Fluid Dynamics Simulations

We used SolidWorks CFD to estimate first-order translational drag coefficients and approximate the added mass terms of the system. To validate these values, we conducted a pool experiment in which LazerShark was pulled across the bottom of a swimming pool at known speeds while force was measured using a load cell. By analyzing the load cell data, we were able to confirm that the predicted drag and added mass values were accurate enough to model the LazerShark's dynamics below approximately 0.7m/s when translating in X, 0.5 m/s when translating in Y and 0.2m/s when translating in Z. Above this threshold, modeling the system as linear is no longer a usable approximation. To stay within the bounds of this model, we intentionally operate the vehicle at low speeds.

C. Pool Testing

By accurately recreating the competition props, CRC can use the LazerShark's cameras to collect custom training data of the props in an Olympic-size swimming pool in order to best emulate the competition environment. CRC can then use the props to run mock runs of the competition in order to test and debug the various software systems.

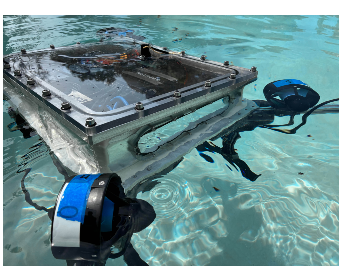

Fig. 6: Pool test

V. Conclusion

As a community college team, Cabrillo Robotics Club experiences the unique challenge of high member turnover. With members joining and graduating, long-term iteration on a single vehicle is difficult. This year marks the end for CRC's underwater vehicle team's current leadership. While the future of the underwater robotics program is uncertain, the accomplishments and experiences from past projects stand as a lasting achievement. Participating in RoboSub and building LazerShark has been the most challenging venture yet. Through this experience, members gained a deepened understanding of robotics and autonomous systems. With such a small team and limited time, CRC is proud of what has been accomplished. It is hoped that LazerShark and the underwater robotics team's legacy will be carried forward by the next era of Cabrillo Robotics Club.

VI. Acknowledgments

Cabrillo Robotics Club extends its sincerest gratitude to everyone who has supported the development of LazerShark. CRC's accomplishments would not have been possible without the generous donations by corporate sponsors, individual donors, and supporters. Thank you to PNI Sensor for providing us with their NaviGuider and TargetPoint-TCM IMUs, and to Water Linked for giving us a generous discount on their A50 DVL. Additionally, thank you to CITO Medical generously providing access to their facilities to machine the electrical housing, and SolidWorks for providing licences. The team is grateful to AMD, Intel, Cabrillo Engineering Department alumnus Mark Cowell, and many others for monetary donations. CRC recognizes the support of advisors Mike Matera and Andrew Thach. Finally, a big thank you to Marlene Coury for providing pool time for testing. Your generosity is greatly appreciated.

VII. References

[1] O-Ring Division. *Parker O-Ring Handbook*. Parker Hannifin, Parker Seal Group, 2007.

[2] Christopher E. Brennen. *Added Mass*. 2005. URL: http://brennen.caltech.edu/fluidbook/basicfluiddynamics/unsteadyflows/addedmass/introduction.pdf.

[3] C. E. Brennen. *A Review of Added Mass and Fluid Internal Force*. Naval Civil Engineering Laboratory, Jan. 1082.

[4] Texas Instruments. *INA740x 85 V, 16-Bit, Precision I2C Output Digital Power Monitor with EZShunt Technology (Datasheet)*. July 2023. URL: https://www.ti.com/lit/ds/symlink/ina740b.pdf.

[5] Ignacio Carlucho et al. "Comparison of a PID controller versus a LQG controller for an autonomous underwater vehicle". In: *2016 3rd IEEE/OES South American International Symposium on Oceanic Engineering (SAISOE)*. 2016, pp. 1–6. DOI: 10.1109/SAISOE.2016.7922475.

[6] Sertac Karaman and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning". In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894. DOI: 10.1177/0278364911406761.

[7] Anthony Stentz. "Optimal and Efficient Path Planning for Partially-Known Environments". In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. 1994, pp. 3310–3317. DOI: 10.1109/ROBOT.1994.351061.

[8] Sven Koenig, Maxim Likhachev, and David Furcy. "Lifelong Planning A*". In: *Artificial Intelligence* 155.1 (2004), pp. 93–146. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2003.12.001.

[9] Sven Koenig and Maxim Likhachev. "D* Lite". In: *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*. Edmonton, AB, Canada: AAAI Press / The MIT Press, 2002, pp. 476–483. DOI: 10.5555/645528.657484.

[10] Christian Zammit and Erik-Jan Van Kampen. "Comparison between A* and RRT Algorithms for 3D UAV Path Planning". In: *Unmanned Systems* 10 (Sept. 2021). DOI: 10.1142/S2301385022500078.