Technical Documentation

Xavier O'Keefe, Jake Tucker, Andrew Heck, Grant Riddle
University of Colorado, Boulder

***Abstract*—Over the past year, Colorado RoboSub has focused on preparing its new autonomous platform, Chimera, for competition. Chimera was deliberately designed as fully modular submarine, capable of operating both autonomously and via surface tether, to support a planned two-year development cycle. In the first year the team focused on building a robust and adaptable hardware platform. This past year, we shifted focus toward autonomy, integrating autonomy-critical components like batteries, an improved camera system, a modified frame, and an autonomy-ready software stack using image recognition and behavior trees. These innovations were guided by the demands of the RoboSub competition. Chimera now serves as a strong foundation from which we aim to expand capabilities to complete a greater number of tasks in 2026.**

I. Competition Strategy

Colorado RoboSub has a long history with the RoboSub competition, competing from 2014 to 2019 with our first vehicle, Leviathan. Leviathan was able to achieve moderate success, placing 7th in the 2019 competition. However, Colorado RoboSub did not compete in the RoboSub competition during the 2020 season due to the pandemic, and lost much of the talent and knowledge that had allowed us to be competitive in past competitions. The last 4 years have seen grow from a small team of underclassmen with minimal knowledge to a 20-person strong team with strong experience from building particpating in the MATE ROV competition.

Recognizing the challenge of returning to RoboSub after a multi-year hiatus, we adopted a deliberate two-year development plan. The first year focused on building Chimera, a modular submarine designed for reliability and future scalability. The second year, this current season, has been dedicated to enabling autonomy through battery integration, navigation enhancements, and an all-new autonomy stack.

Chimera is equipped with five cameras and a Doppler Velocity Logger (DVL), giving it the theoretical capability to complete most of the RoboSub tasks. However, our strategy prioritizes reliability of attempting every task. We chose to focus on a smaller number of tasks that align with our current experience and offer a good balance of point potential and complexity, as well as to maintain an effective engineering velocity by focusing our limited efforts on achievable goals.

Our competition plan begins with completing "Task 1 - Collecting Data", where we drive through the gate while remembering which sea creature we passed under. We anticipate that we will be able to earn style points in the way of a 360° yaw. If successful, we will then attempt "Task 3 - Drop a BRUVS", using the memory of our selected sea creature to determine where to drop our marker. This is considered a stretch goal for our team. We will designate a set amount of time for our AUV to attempt this task, before commanding it to complete "Task 6 - Return Home". Using the path generated by Chimera's DVL, our hope is that our robot will be able to complete the return home task to end its run.

By focusing on reliability and incremental success, we aim to ensure robust mission execution while laying the groundwork for more complex tasks next year.

II. Design Strategy

Colorado RoboSub had initially developed our current submarine platform, Chimera, during the 2023-2024 school year with the express goal of making it a hybrid submarine that could be tethered and controlled from the surface or run autonomously powered off of onboard batteries. This year, we shifted focus to implement everything needed for an autonomous sub, so the majority of this year's design projects focused on optimizing Chimera for autonomous control.

A. Electrical

Chimera's electrical system was designed with a philosophy of modularity; Each component should be responsible for only one task, and be minimally dependent on the specifics of each other component. This philosophy means that if one component breaks it can be easily replaced, or a component could be changed for a different one without impacting the rest of the submarine. Each component falls into one of three categories: power conversion and delivery, Control and locomotion, and sensing, with a main computer to connect each of the systems together.

1) Power Delivery: The power delivery system was designed to operate using a 48V tethered power supply or onboard batteries. To accommodate the two modes, we opted to use a 52V battery system. This allows us to use the same buck converter to step down tether power, or battery power to the voltages used by the systems on the sub. The decision to use a higher voltage from the battery also decreased the current required for power delivery, which allowed us to use a smaller gauge wire.

The power delivery system uses a custom DC-DC conversion board to convert the input power to a 12V line and a 5V line. The 12V line is used for driving the motors and DVL sensor and the 5V line is used for logic components, like the computer and PWM generator.

2) Control & locomotion: To control position of the submarine there are eight T200 thrusters which have their speed controlled using ESCs. For interaction there are 2 servo controlled effectors. The speed that each ESC dictates, as well as the position of the servos are set by a PWM signal output by the Pololu maestro PWM generator. The computer, a Jetson Orin Nano, controls the PWM generator using serial communication.

3) Sensing: The submarine is equipped with 4 cameras, a Doppler velocity logger (DVL), and inertial measurement unit (IMU). Each camera is connected to the computer using USB, while the DVL and IMU are connected by ethernet through an interface board.

B. Software

Colorado RoboSub's software was developed with a hybrid approach, allowing for the robot to operate in both an autonomous mode and through the use of a joystick. The platform was initially developed during the initial construction of the submarine, but had been expanded to use a behavior tree system for task construction in autonomous mode.Our entire software stack is constructed around ROS2 [1], a popular open-source framework that provides many important functionalities for robotics. We have found that the ROS2 middleware is incredibly convenient for connecting with hardware, connecting with simulators, and visualizing data.

1) Motor Drivers: Our system uses a Pololu Mini Maestro to interface with our motors. We use a provided Python library from Pololu to interface with our motors, and we created a custom ROS2 package to translate our desired velocity commands to values that work with our specific motor configuration. Our initial method of development focused on getting the motors to work with the controller in a natural way, and in order to do that we developed a package to take in joystick data and output a vector motor command to take the sub in the desired direction.

2) Camera Systems: We have an array of five cameras mounted around the sub. We stream all of the camera data in to our computer via several ROS nodes. We stitch together these four camera streams to create what is essentially a 360 degree field of view around the sub. Since we do not have a stereo vision system implemented on our submarine yet (a project for this upcoming year) we use an algorithm consisting of known object sizes, and OpenCV [2] image data to do a version of object localization and pose estimation.

3) Behavior Trees: At the highest level of our software stack, we control the submarine using a behavior tree. We find that the traditional state machine paradigm used in a lot of autonomous vehicles, while effective, can become cumbersome to manage, and thus lead to increased complexity. Behavior trees, contrarily, are much easier to scale, as one does not have to track complex and non-intuitive state transitions. This comes with the added benefit of increased understandability, a key component for debugging.

4) YOLO: We use the You Only Look Once (YOLO) [3] library as the base for our image recognition software. This easily trainable, blaz-

Fig. 1. Sample Behavior Tree

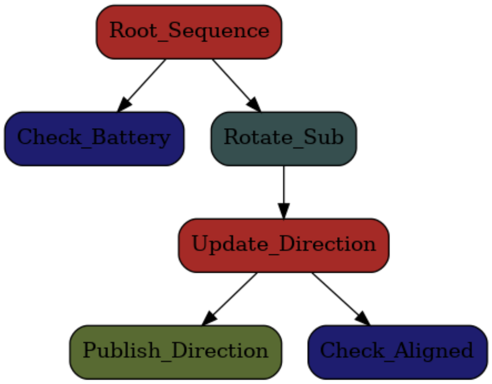
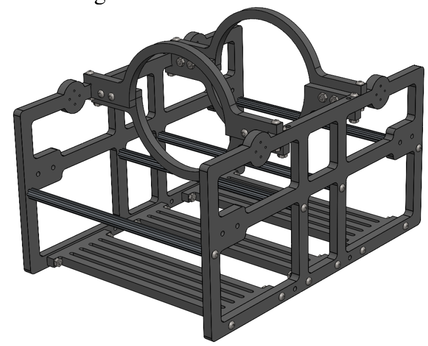


Fig. 2. Chimera's Frame



ingly fast, open source image classification algorithm allows our sub to make sense of its environment in the pool, and make informed decisions based on these observations.

C. *Mechanical*

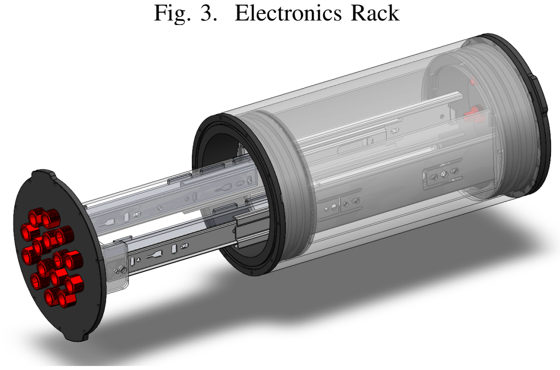
Chimera's mechanical and structural systems were designed to be modular with ease of manufacturing in mind. We choose rapid manufacturing methods like 3D printing and laser cutting when able to reduce down time and speed up the prototyping process. When traditional machining was required we used easily machinable materials like aluminum to speed up the process.

1) Frame and Thrusters: Chimera's frame, Fig. 2, is made from ½" thick marine-grade high density polyethylene (HDPE), ½" anodized aluminum churro-tubes, and held together by 1/4"-20 bolts. HDPE is relatively cheap, strong, and easy to machine, which allowed us to perform all manufacturing in house at CU's Integrated Teaching and Learning Laboratory's machine shop. The addition of the churro-tubes improved rigidity while also acting as a mounting surface for our cameras. When we had initially manufactured Chimera's frame in 2023 we left an 8" width below the electronics housing, but when implementing our batteries in 2024 we realized we needed more room so we re-manufactured the frame cross pieces to give us a width of 12". Due to the modular design of our sub, this change was easy to implement and did not effect many other components.

Chimera has four angularly vectored thrusters in the X-Y plane and four high mounted vertical thrusters for movement in the Z direction. Our previous ROV, Lazarus, only had two thrusters in the X-Y plane which limited our speed in the fore and aft directions. By having four angled motors we are able to now strafe in the port and starboard directions while also giving us more thrust when traveling in the fore or aft direction than if we had one set of two motors facing fore/aft and a second set facing port/starboard.

2) Batteries and Electronics Housing: The majority of our electronics are housed in a 6" acrylic tube with end flanges both from Blue Robotics. We had issues accessing and debugging electronics in our previous sub, so we made the development of an electronics' rack, shown in Fig. 3, a high priority. The shelving and connector pieces are made from 1/8" acrylic since it is safe to laser cut , non-conductive, and has a melting point high above the maximum operating temperatures of any of our electronics.

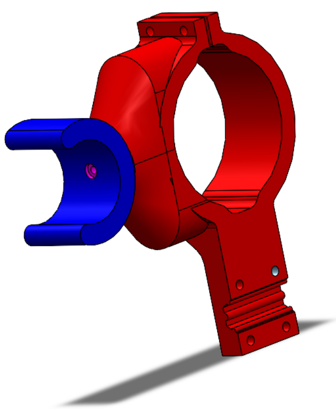
Fig. 3. Electronics Rack



Our batteries are housed in two separate 3" PVC housings, which allowed for some redundancy in case of leaks and better balancing of the sub. The battery housings are located inside of the sub's frame beneath the main electronics hull to protect them as they are our most critical component.

3) Mounts: To mount our angled thrusters, battery pods, and cameras we used mounts manufactured with 3D printed PLA. An example of one of these mounts is shown in Fig. 4. 3D printing allows for rapid production and prototyping of parts and allows for part geometries that would be infeasible from traditional machining. The only downside of using PLA is it is not waterproof and overtime the 3D printed layers will separate as it absorbs water. To account for this we sealed all of our 3D printed parts with two coatings of rubber cement.

Fig. 4. Example of a 3D Printed Camera and Light Mount



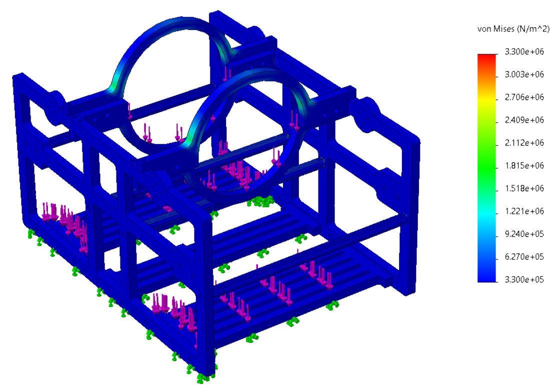
III. Testing Strategy

Colorado RoboSub's testing strategy is shaped by our limited availability of pool access, which we approach not as a constraint but as a core design challenge. As a result, we prioritize modularity and testability across all subsystems from the beginning of development. Both software and hardware components are tested individually in controlled environments before being integrated at the system scale.

A. Hardware

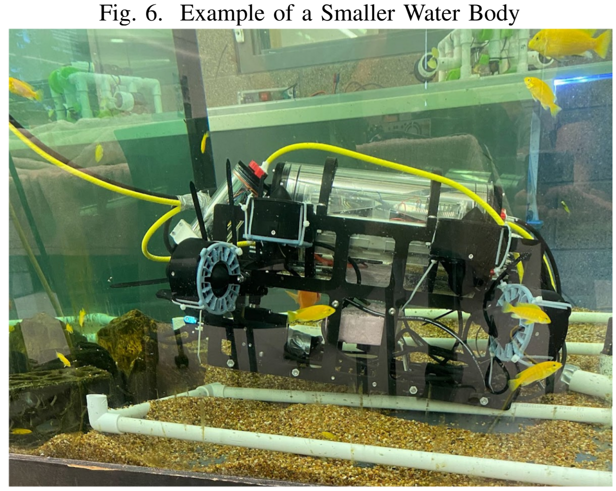
For hardware - both electrical and mechanical systems - we employ primarily land-based and simulated testing methods. Before manufacturing we use off the shelf simulation software like SolidWorks Simulation and AUTODESK Fusion to verify factors of safety and communication between electrical components. We receive the majority of our funding during the spring semester and this approach allows us to speed up the manufacturing process once materials arrive.

Fig. 5. Example of Structural Analysis of our Frame



Our engineers routinely design custom test setups alongside the hardware to ensure isolated functionality. When underwater functionality must be verified, these systems are submerged in smaller bodies of water - often in buckets or tubs, or sometimes fish tanks. After systems are verified to function in isolation they are integrated into the sub, and the resulting integration is confirmed to meet our requirements.

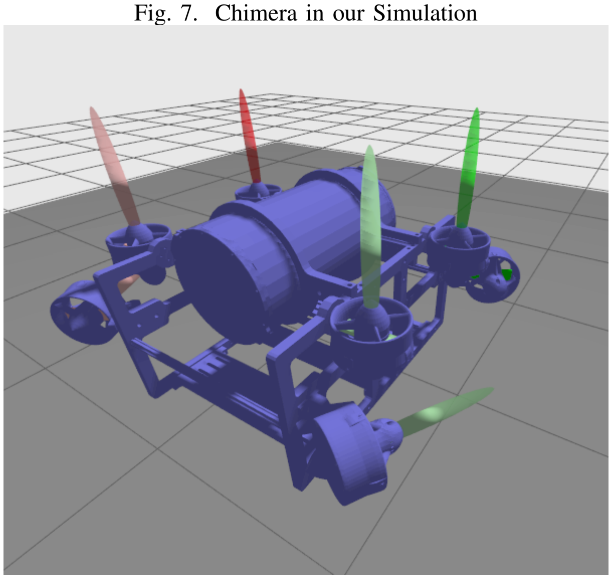
Fig. 6. Example of a Smaller Water Body



B. Software

Testing software at the system level requires a very high level of functionality, as well as access to a large body of water. These conditions are hard to come by, so much of our software testing takes place in a custom-built Gazebo simulation. This simulation is designed to mirror the real sub's physical characteristics and use the same code to produce the same movements. This allows us to perform rapid, repeatable testing of autonomy algorithms, mission planning, and vision pipelines.

Fig. 7. Chimera in our Simulation



Many common software development practices are also employed to ensure high code quality and functionality. We employ automated unit testing to ensure each piece of code is functioning as expected and that breaking changes are caught early. We also use version control combined with regression testing to ensure that when there is a breaking change, it is easy to revert to the last functional version of the repository.

Acknowledgment

Colorado RoboSub would like to thank the Engineering Excellence Fund for their generous funding, Cosmos Pizza for weekly team pizza supplies, and BlueTrail robotics for their hardware discounts and fees waived. We would also like to recognize our Faculty Advisor, Dr. Nikolaus Correll, for his thoughtful advice and support.

References

[1] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, pp. eabm6074, 2022. doi: 10.1126/scirobotics.abm6074

[2] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[3] G. Jocher and J. Qiu, "Ultralytics YOLO11," ver. 11.0.0, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics