Okanagan Marine Robotics: RoboSub 2025 Technical Design Report

Eryk Halicki, Jovan Sohi, Alexis Wong, Danielle Mackay
Okanagan Marine Robotics
University of British Columbia, Okanagan
Kelowna, Canada

***Abstract*—In 2025, Okanagan Marine Robotics (OKMR) presents Ogopogo, the team's second submission to RoboSub, replacing the Cascade AUV. Following a successful RoboSub debut, OKMR's second design cycle marks crucial improvements emphasizing reliability, modularity, and extended capabilities to meet the planned 2025 competition strategy. These include additional enclosures for Ogopogo's batteries and actuators, the standardization of all communication and interfacing with a modular system bus, and a significant rework of control and autonomy stacks.**

These upgrades enabled new hardware additions, including the 3-axis gripper arm, mechanically actuated torpedoes, and acoustic subsystems, allowing OKMR to attempt all tasks at RoboSub 2025. Facing numerous setbacks throughout the year, the team maintained momentum through continuous design, build, and test cycles, while remaining committed to system-wide improvements through weekly pool tests to fill gaps in Cascade's control systems. With this, OKMR is expecting to complete the Gate, Slalom, Dropper, and Return Home tasks, with attempts prepared for the Torpedo, Gripper, and Random Pinger tasks.

I. Introduction

Okanagan Marine Robotics is a student-led engineering design team at UBC Okanagan focused on the development of autonomous robotic systems. The team provides students with practical experience in electrical, mechanical, and software engineering as well as business administration through hands-on project development and fundraising efforts. By offering these opportunities, Okanagan Marine Robotics extends the academic experience at UBCO, contributing to both curricular and extracurricular learning in applied engineering fields.

In the team's second year of existence, OKMR has quadrupled its membership, which catalyzed the major overhauls to the previous AUV. This report details the work done to enable OKMR to be more competitive in upcoming RoboSub competitions, as well as pioneer robotics in the Okanagan region through professional development opportunities and outreach initiatives (Appendix D & E).

II. Competition Strategy

Following OKMR's inaugural year at RoboSub 2024, the team created year-long competition goals to guide development for the 2025 design cycle. Taking into account the team's significantly increased membership, funding availability, and timing, OKMR outlined a strategic overhaul to bring these goals to fruition. OKMR has split its ambitions into two categories: primary and secondary goals.

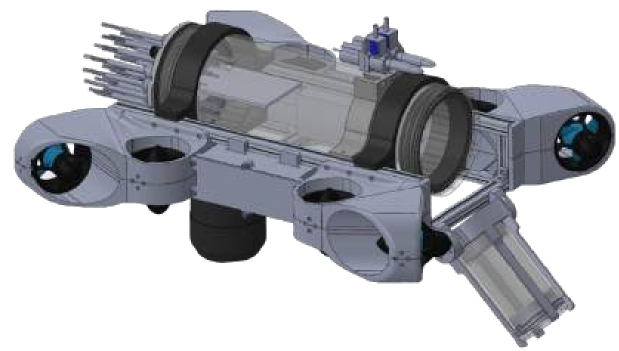

Fig. 1. SolidWorks Rendering of Ogopogo

Primary goals are OKMR's immediate tasks for the RoboSub 2025 competition that can be accomplished with a high degree of consistency. These tasks were selected due to their reliance on subsystems that were readily available for testing in late 2024 and early 2025.

- **Gate:** Attempting the Coin Flip, enabled by the new hierarchical state machine in the Automated Planner, and the new object detection subsystem. OKMR will also execute style points (2x Yaw + 2x Barrel Roll) using the new control stack.
- **Slalom:** Undertaking in the full slalom task, now feasible due to the new object detection system.
- **Dropper:** Pursuing the dropper task utilizing the new 3-axis arm with predetermined joint states. This de-risks the task by separating it from the complexities of the fully actuated gripper, allowing the team to secure extra points while manipulator development continues.
- **Return Home:** Aiming to complete the long-horizon task, made viable by the new localization system, which mitigates the accumulated drift that would have previously caused mission failure.

Secondary goals are tasks that the team has considered possible if development ends ahead of schedule. The subsystems required for the following tasks are currently being manufactured, with completion expected around early August. Due to this tight turnaround, OKMR will likely be performing initial tests during RoboSub 2025.

- **Torpedo + Surfacing / Octagon:** This attempt is contingent on the successful integration of the team's full acoustics pipeline and the proven reliability of the core autonomy stack.
- **Gripper:** The full Gripper task depends on the late-stage integration of the team's actuator hardware, control software, and the close-range Intel D405 camera.
- **Random Pinger:** Requires a fully functional acoustics system for localization and a mature automated planner capable of handling uncertain task order.

The focus of this year has been on reliable completion of the team's primary goals as well as the design and implementation of the subsystems required to complete OKMR's secondary goals. Following the successful completion of the primary goals, further testing and data collection will ensue to support attempts for the team's secondary goals at RoboSub 2025 or in preparation for RoboSub 2026. (See Appendix G: Competition Strategy Risk Mitigation Plan for more task selection rationale)

Fig. 2. Gripper Arm

III. Design Strategy

A. Mechanical Systems

*a) **Structures**:* The structure redesign improves on OKMR's prior AUV's suboptimal hydrodynamics, poor buoyancy, and overall size by adding an internal air pocket within a compact shell design (Fig. 1). This updated design also significantly improves durability, as the previous design would frequently snap at fatal stress points.

*b) **Electronics Tray**:* The electronics tray was completely redesigned with vertical mounts and custom snap-fit clips, exposing more modules per layer to simplify maintenance and upgrades. An aluminum tray runs through the hull and is connected to the end-cap to increase heat transfer outside of the control enclosure.

*c) **Torpedo**:* The torpedo design utilizes a spring-loaded launching system that operates by misaligning the torpedo fins with guide grooves and realigning them using a solenoid to fire. The torpedoes weight distribution is adjusted with variable infill in the head and fins, resulting in a maximum straight trajectory of 1 meter.

*d) **Gripper Arm**:* Ogopogo's gripper arm was originally a 3 axis design that utilized bio-inspired gecko fingers, however we chose to lock one of the servos for now to simplify testing and software development. In addition to the large range of motion, a pulley system was implemented in the hand of the gripper allowing the fingers to grip any unique shape effectively.

*e) **Battery Enclosure**:* To improve thermal management and serviceability of Ogopogo, the team relocated the batteries from the center of the control enclosure to an external, custom-machined aluminum enclosure. This results in simpler distribution of power to future enclosures, increased heat transfer and significantly more space within the control enclosure.

B. Electrical Systems

*a) **Overview**:* For OKMR's second RoboSub submission, the electrical team overhauled the communication and power systems, emphasizing modularity and expandability. This is to support the added hardware such as the hydrophones, gripper arm, torpedo's, and Zynq 7020 FPGA. The new layout and design improve on Cascade's lack of circuit protection and unified communication system for data collection and debugging.

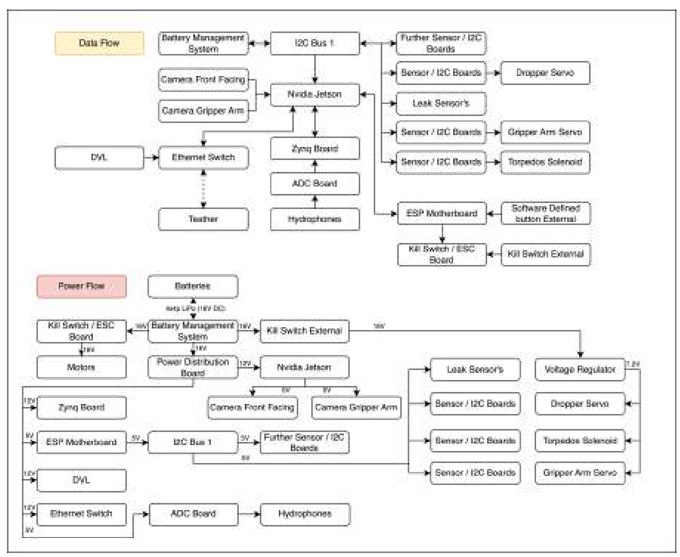

Fig. 3. Ogopogo Power and Communication Flowchart

*b) **I2C Sensor Bus**:* The I2C Sensor Boards are the backbone of the I2C bus unifying four out of five of Ogopogo's enclosures. This has been the most significant electrical upgrade to support the team's newly implemented systems.

These STM32-based PCBs provide current and environmental sensor data, simplifying system-wide troubleshooting. They feature PWM for actuator control, analog inputs for low-level interfacing, and I2C pass-through to connect to other I2C devices to the system bus.

This replaces a point-to-point wiring scheme that was difficult to debug and scale. The I2C bus architecture improves modularity and standardizes diagnostic data across enclosures, simplifying maintenance and the addition of new systems.

Fig. 4. I2C PCB.

*c) **Hydrophones**:* The Hydrophones began as a project when OKMR was formed and are being implemented into Ogopogo as the foundation for its onboard acoustics. The analog signal generated by the Aquarian S1 hydrophones is amplified and filtered through a custom-designed 6th-order Chebyshev filter and ADC board. The digital signal is then fed into the fabric of the Zynq 7020 FPGA, where the cross-correlation is calculated to determine the time delay of arrival (TDOA). The peak of the resulting output is fed through a custom driver and utilized by the team's Jetson to aid in navigation during competition.

*d) **Pingers**:* To aid in the system-wide testing of the hydrophones, a Digital to Analog Converter PCB was designed to power a transducer to mimic the ones used at RoboSub for a fraction of the cost. The AD5760 DAC is driven through a custom SPI driver compiled with MBed OS on an STM32. This delivers a 12 V 25-35 kHz sine wave through a bandpass filter and amplifier, enabling the transducer to reach 120dB SPL at 30 cm.

*e) **Killswitch**:* The Killswitch serves as a software and hardware interface capable of killing power to the propulsion system and beginning autonomous missions. The design utilizes MOSFETs in series with the motors on the low side, disconnecting them from the batteries on command. Flyback diodes were included to enable safe dissipation of any stored energy inside the thrusters.

*f) **System Improvements**:* The other electrical updates included circuit protection for the team's ESP32 in the form of the ESP32 Motherboard, which serves as the master for Ogopogo's sensor integration, system bus and actuator control. Other systems were implemented to improve on Cascade's safety and sensing capabilities. These include the Battery Management System (BMS), an I2C device on a custom PCB made for high power applications, and the leak sensor, a transistor-based circuit that acts as a switch if water is detected at determined failure points.

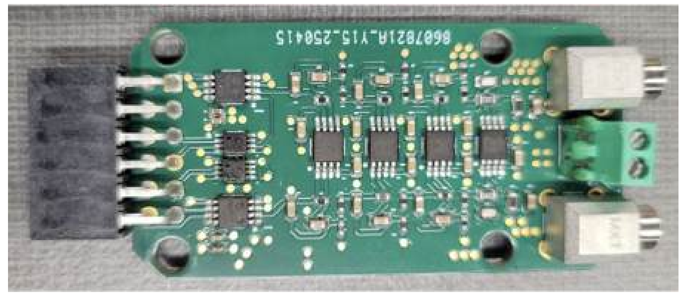

Fig. 5. Hydrofilter PCB.

C. Software System Design

*a) **Overview**:* OKMR's software stack has matured significantly to support more advanced competition objectives. Based on the control hierarchy model commonly used in subsea robotics [1], the system is organized into layered subsystems (L5–L1), with higher layers responsible for mission planning and lower layers focused on high-rate control.

Although the ROS2-based architecture and subsystem names remain the same, each module has been improved or redesigned to enable more sophisticated autonomous behaviour. See Fig 15 in Appendix A for a full system diagram.

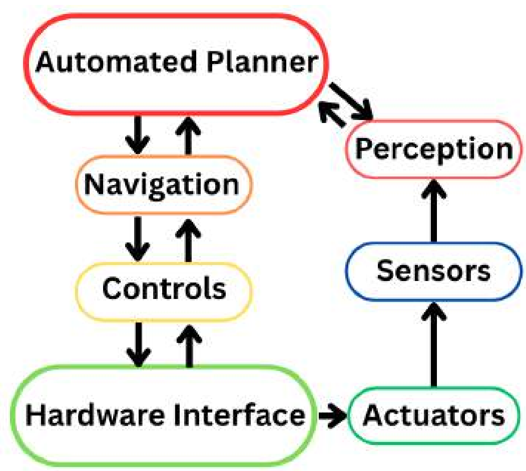

Fig. 6. Heavily simplified Software System Diagram, please see Appendix A for full diagram. This version shows the overall data flow from Automated Planner (L5) to Hardware Interface (L1)

*b) **Automated Planning**:* Ogopogo's automated planning system manages high-level decision-making and coordinates various subsystems. To handle more complex missions, the software team replaced the minimal 2024 implementation with a modular, hierarchical state machine framework built using an in-house extension of PyTransitions. This design eliminates hardcoded mission flows and integrates cleanly with ROS2. (See Appendix: F)

Although behaviour trees are an increasingly popular alternative [2], the team deliberately chose state machines for their superior traceability and simplicity. This design choice accelerates debugging during pool tests and simplifies the onboarding process for new team members, critical for OKMR's long-term sustainability.

*c) **Mapping**:* The mapping subsystem remains largely unchanged, continuing to utilize a modified Bonxai voxel grid for high-frequency semantic mapping. The system still supports 3-D object localization using the object detection system, allowing Ogopogo to understand both where objects are and what they are. A key enhancement for 2025 is the integration of hydrophone data, which enables the localization of pinger sources in real-time.

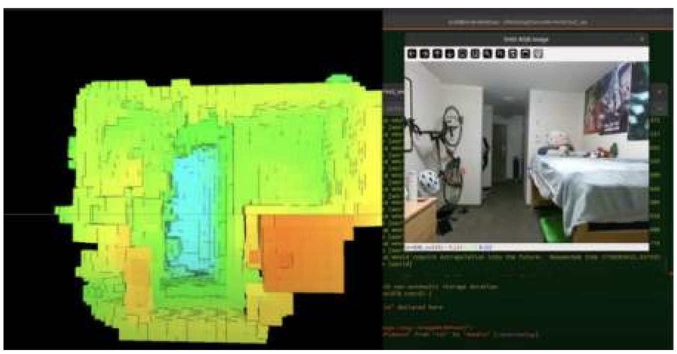

Fig. 7. Mapping System test using Intel D455 Camera.

*d) **Object Detection**:* The object detection pipeline underpins perception-based planning and mapping. The system has transitioned from a pre-trained You Only Look Once (YOLO) model to a custom PyTorch-based implementation using the ResNet backbone. While OKMR's previous YOLOv8 model offered rapid and simple implementation, it provided limited flexibility for integrating RGB-D data. Transitioning to a custom ResNet-based pipeline gives OKMR full control over the model architecture, allowing for future fusion of depth information to improve object localization accuracy and reduce false positives in cluttered environments.

*e) **Navigation**:* The navigation subsystem is responsible for inertial localization and interpreting motion requests. Several significant upgrades were implemented in 2025:

- Introduction of a ROS2 Action interface to facilitate integration with the automated planning system
- A full C++ reimplementation of the localization node to improve performance during long horizon tasks (ex. Return Home task)
- Incorporation of complementary-filtered accelerometer data to mitigate pitch and roll drift

*f) **Actuator Control**:* A dedicated actuator control subsystem has been introduced to interface with servos and

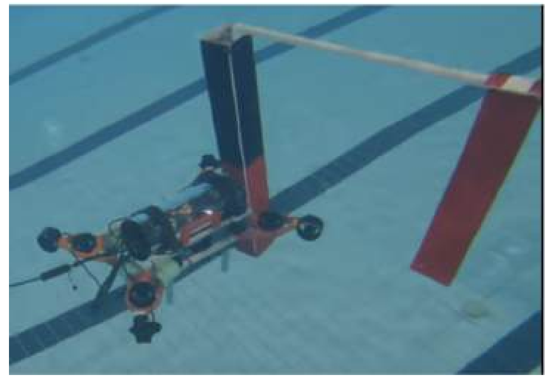

Fig. 8. Cascade crashing out due to object detection failure.

solenoids. This subsystem also hosts the inverse kinematics solver, which is used for gripper arm motion planning, allowing Ogopogo to turn desired Cartesian coordinates into actionable joint angles that Ogopogo can send to the hardware interface.

While additional testing is contingent on hardware availability throughout the summer, the architecture has been designed with modularity and future scalability in mind, enabling Ogopogo to use gripper arms with significantly more degrees of freedom in future years.

*g) **Motor Control**:* The motor control subsystem translates high-level motion commands into stabilized thrust outputs, taking into account environmental disturbances and model uncertainties. Previously implemented as a two-layer PID system, it has been restructured into a five-layer control stack incorporating PID control, feedforward components, thrust allocation, and throttle conversion. Although this increased complexity introduces greater tuning requirements, it yields significantly enhanced control, stability, and precision during complex operational scenarios like the gripper task.

This increased complexity is exemplified by the number of tunable parameters rising from approximately **45** in OKMR's 2024 system to over **150** in the current implementation. This necessitated the team's shift to a more formalized, data-driven tuning methodology.

By increasing the number of control layers, the team was able to implement more stable barrel rolls and yaw rotations, improving camera feed clarity during normal operation and causing less positional drift during style point task attempts. Furthermore, the new thrust allocation matrix control layer can be reused for any motor configuration, improving future scalability.

IV. Testing Strategy

A. *Mechanical Testing*

OKMR's Mechanical testing focused on rapid design and iteration, as real-world feedback would yield much better returns than relying solely on simulation and software. An emphasis was placed on usable design and rigidity, as the team's prior design, Cascade, faced many setbacks with clearances in transportation and fatal weak points.

a) Structures Testing: Utilizing SolidWorks, multiple FEA's were conducted as a prerequisite for 3-D printing a new thruster arm design. As shown in Fig. 9, weak points from Cascade's design, such as the thruster mounts, are no longer present,

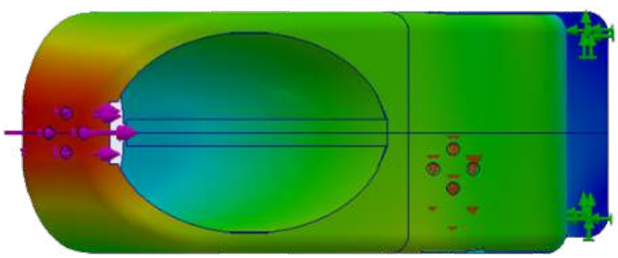

Fig. 9. Thruster Arm FEA Done in SolidWorks.

B. Electrical Testing

The electrical testing strategy aimed to verify the results received in simulations and confirm key calculations. With the scale of Ogopogo's electrical overhaul, isolated and system-wide tests were necessary to ensure functionality.

a) ***Hydrophones Testing****:* The digital filter implemented on the Zync FPGA has been rigorously tested to ensure a clean output even past the 6th-order analog filter. The visual shown in Fig. 10 is a simulation run in Vivado, which added the equivalent of 12mV of random noise. The results indicated that the digital filter implemented was able to accurately process the signal while also effectively removing all added noise.

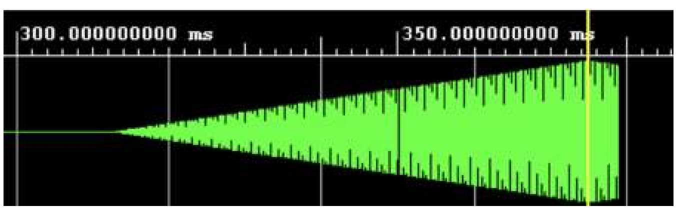

Fig. 10. Vivado Simulation Results.

b) ***Hardware Testing****:* With the timeline OKMR team faced, prototyping and iterating designs proved to be a challenge, requiring testing to be as rigorous and well thought out as the design. The team's hardware testing required simulating peak load conditions of the AUV's power and communication system.

c) ***I2C Addressing****:* To ensure timing and data integrity across the I2C system bus, the team utilized a digital oscilloscope during a typical sensor data transfer. This test read temperature, pressure, and humidity values from I2C address 0x42. (Fig. 11)

d) ***Chebyshev Filter Testing****:* The amplification and analog filtering of the Chebyshev filter is paramount to the success of the hydrophones' acoustic localization system. In Fig. 12, the success of the filter is shown with three different applied signals.

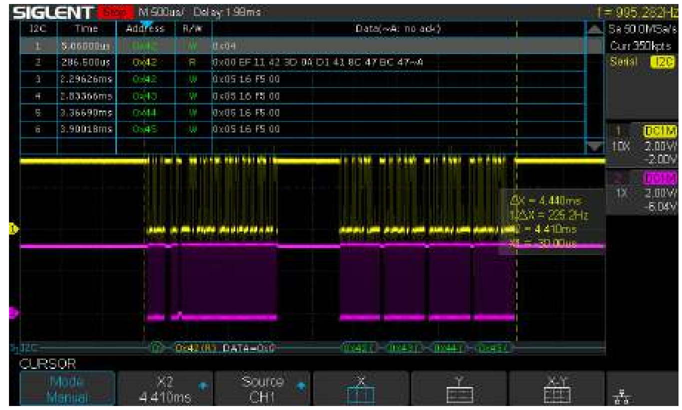

Fig. 11. 1.54 ms Transfer Time Including Request Time for Environmental Sensor Capture

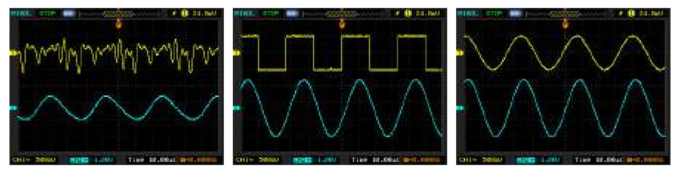

Fig. 12. Different Sampled Outputs of Chebyshev Filter vs. Inputs

e) ***Power System Testing****:* The testing of the Battery Management System and Killswitch involved validating control logic, data transfer, and power throughput. Control logic was required for the Killswitch, ensuring it is functional with Ogopogo's software Killswitch as well as the hardware Blue Robotics Killswitch. This test could be completed by probing with a multimeter at the motor output while applying a 5 V control signal.

Power throughput testing of each of the aforementioned boards involved utilizing a thermal camera to monitor for temperature spikes across high-power copper pours. Temperature spikes in these regions would indicate areas of high resistance caused by manufacturing defects or incorrect pour geometry.

C. Software Testing

a) ***Overview****:* With OKMR's first AUV build and pool test happening in the middle of RoboSub 2024, the software team was able to start real testing for the first time this design cycle. As such, the team had to adopt a more organized testing methodology to support hardware-in-the-loop testing. To accomplish this, OKMR introduced formalized testing plans and documentation.

This year's testing enabled the fine-tuning of Ogopogo's localization, navigation, and control systems, aiding in Ogopogo's reliability and support for the systems to be added.

b) ***Lake Testing****:* During summer 2024, OKMR was able to do system integration testing in the Okanagan Lake. These tests resulted in valuable insight into Ogopogo's control and localization systems and highlighted the need for a comprehensive data analysis tool. FoxGlove Studio was implemented to address this gap.

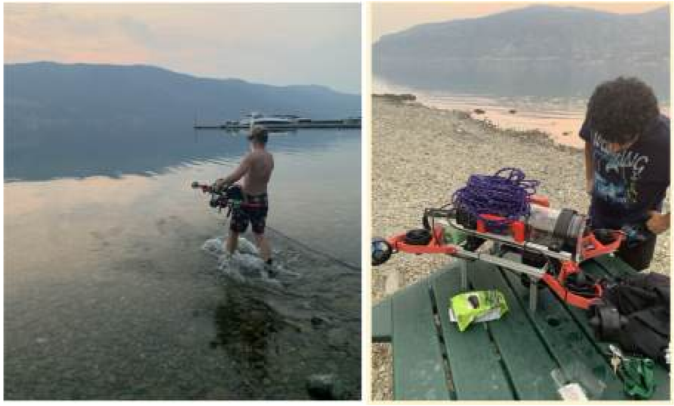

Fig. 13. Left: President David with Cascade at Okanagan Lake. Right: Electrical Lead Jovan inspecting Cascade.

*c) **Pool Testing**:* Starting in mid-November 2024, the Kelowna Family YMCA generously allocated weekly pool time to OKMR. To make the most of this opportunity, the software team began implementing formalized testing and result forms.

Key outcomes of these tests were enhancements to Ogopogo's PID and Dead Reckoning systems, as well as improvements to the team's debugging and data analysis workflows. Furthermore, pool testing allowed the team to collect critical perception data for training Ogopogo's object detection models, and initiated OKMR's usage of ROS2 bags for comprehensive data logging.

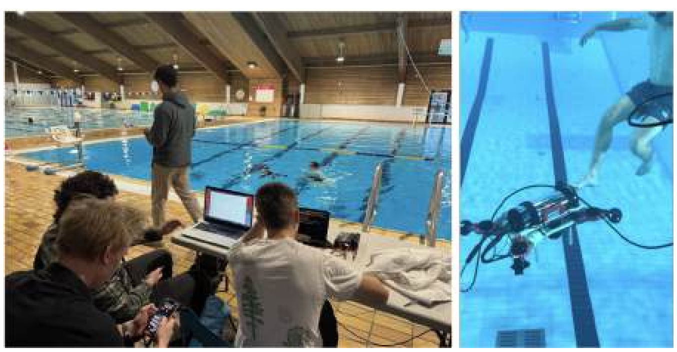

Fig. 14. Left: Software team at pool test. Right: Software lead Eryk testing localization and PID by kicking the AUV.

*d) **Automated Testing - CI/CD**:* With a growing codebase and software team, the need for automated unit and integration testing became clear this development cycle. By implementing a basic CI/CD pipeline using GitHub actions and the colon test tool built into ROS2, the software team was able to improve early recognition of bugs and continuously verify that new code did not cause any retroactive issues.

*e) **Simulation Testing**:* Simulation remains a cornerstone of OKMR's development workflow, allowing for testing without pool access. This year, the team migrated from Gazebo Sim to Stonefish Sim, a high-fidelity simulator designed specifically for marine robotics. This transition enabled significantly more scalable and realistic simulation of hydrodynamics, sensors, and thrusters, resulting in a more accurate simulator. This improved sim-to-real transfer and reduced tuning and debugging time during in-person pool tests, providing greater focus towards higher-level integration challenges.

V. Conclusion

The 2025 Okanagan Marine Robotics program demonstrates how a targeted system-level redesign can transform an AUV from a functional prototype to a competitive platform. The multi-enclosure architecture connected through an I2C system bus fosters sustainable growth, simpler troubleshooting, rapid battery changes, and drastically improved serviceability and thermal management. From September to April, weekly pool tests and high-fidelity simulations yielded great improvements to the perception, control, and automated planning software systems. This ensures the Gate, Slalom, Dropper, and Return Home tasks can be attempted with confidence. This builds the foundation where continued development will see the software implementation of the new Gripper Arm, Torpedo, and Acoustics System to maintain OKMR's continued growth at the RoboSub Competition.

VI. Acknowledgments

The accomplishments of Okanagan Marine Robotics this year were made possible through the generous support of individuals, organizations, and institutional partners. We sincerely thank our sponsors for their continued commitment:

UBC's Professional Activities Fund, Cellula Robotics, UBC Okanagan's Engineering Society, Blue Trail Engineering, Banksia Robotics, SKYTRAC Systems, Blue Robotics, Mearl's Machine Works, Mouser Electronics, Bloom, Rocky Mountain Motion Control, Altium

We would like to give special thanks to **Eric Jackson** of **Cellula Robotics** for his belief in our team. His support has gone far beyond sponsorship, providing us with valuable mentorship and enabling three of our members to pursue internships with his company.

We are also grateful to **Carley Hunt** and the staff at the **Kelowna Family YMCA** for providing access to pool facilities, which were essential for in-water testing.

Our appreciation extends to the **UBC Okanagan School of Engineering**, the **Students' Union of UBC Okanagan**, our faculty advisor, **Dr. Rudolf Seethaler**, and Student Professional Development Officer **Grant Topor** for their continued guidance, lab access, and technical support throughout the development process.

We especially thank **RoboNation** for its leadership of the RoboSub competition, which has played a key role in shaping our team and inspiring young engineers around the world.

Finally, we acknowledge the heart of our club: our team. This year presented both technical and personal challenges, but through resilience, collaboration, and a shared purpose, our members have consistently demonstrated exceptional dedication and professionalism. Their commitment and passion remain the driving force behind everything we accomplish.

VII. References

References

[1] R. Skjetne and O. Egeland, ”Hardware-in-the-loop testing of marine control systems,” 2005. Available: https://www.researchgate.net/publication/330703311

[2] M. Iovino, E. Scukins, P. Albertelli, and D. Nardi, ”A Survey of Behavior Trees in Robotics and AI,” *arXiv preprint arXiv:2208.04211*, 2022. [Online]. Available: https://arxiv.org/abs/2208.04211

[3] Texas Instruments, ”BQ76952 Technical Reference Manual,” Revised May 2022. Available: https://www.ti.com/lit/ug/sluuby2b/sluuby2b.pdf

[4] Analog Devices, ”Ultra Stable, 16-Bit +/- 0.5 LSB INL, Voltage Output DAC,” Revised 2018. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ad5760.pdf

[5] H. J. Nussbaumer, The Fast Fourier Transform, in Fast Fourier Transform and Convolution Algorithms, vol. 2, Springer Series in Information Sciences. Berlin, Heidelberg: Springer, 1982, pp. 80–111. Available: https://doi.org/10.1007/978-3-642-81897-4$_4$

[6] Z. Peterson, “Getting started with high current PCB design,” Altium, November 2021 https://resources.altium.com/p/getting-started-high-current-circuit-board-design

VIII. Appendices

A. Appendix A: Software System Diagram

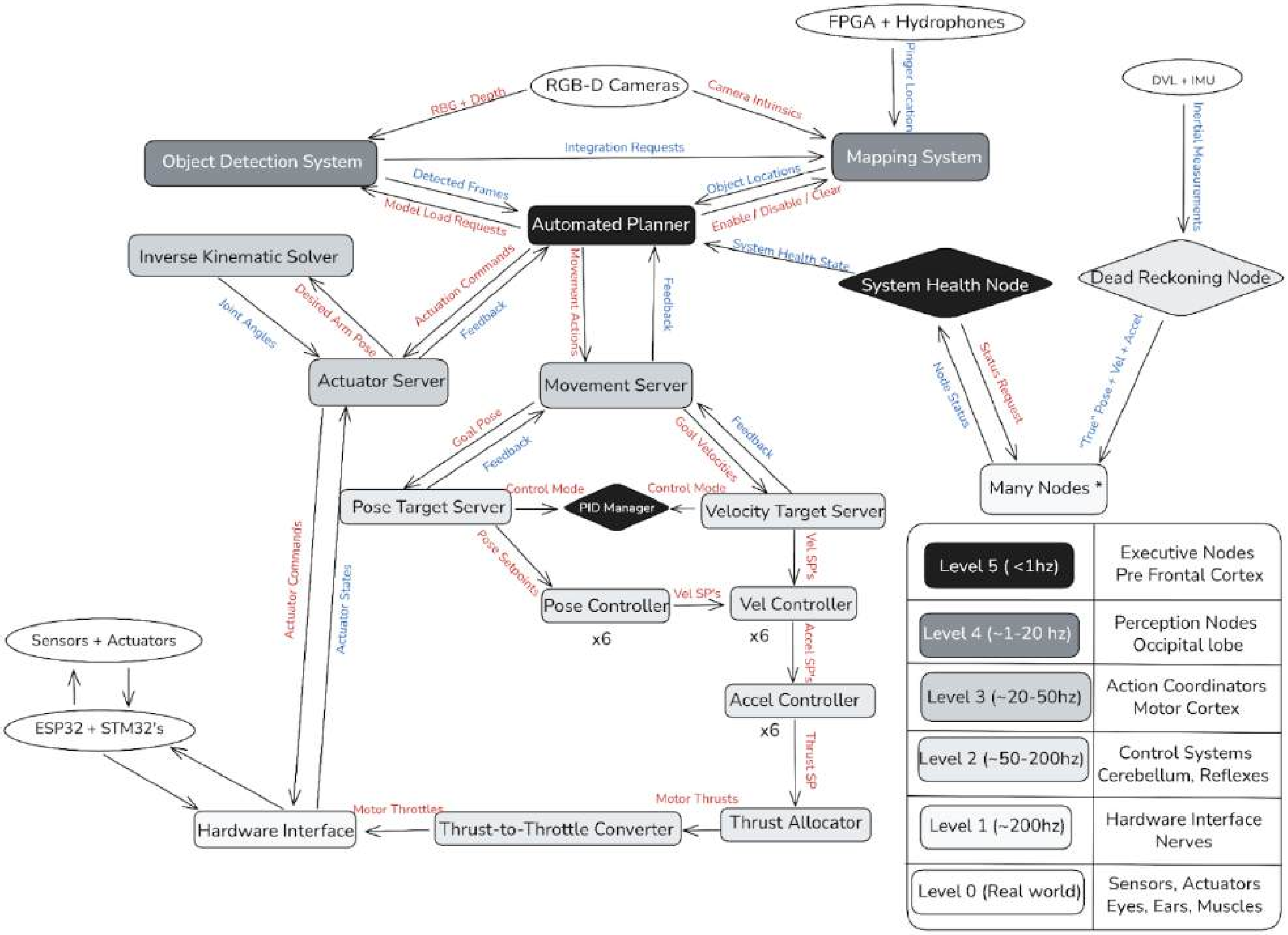

Fig. 15. Full Software System Diagram. Bottom Right Legend shows biological analogies for different parts of the system, as well as the colouring scheme for the control hierarchy (Black for L5 to Light Gray for L1). Note: the "Many Nodes" block represents most nodes in the system, and drawing connections between them all would result in a difficult-to-read diagram.

B. *Appendix B: Vehicle Information*

Component	**Vendor**	**Model/Type / Specs**	**Custom / Purchased**	**Cost**	**Year**
Buoyancy Control	N/A				
Frame	N/A		Custom		2024
Waterproof Housing					
Waterproof Connectors	Blue Robotics	Potted Cable Penetrator	Purchased		2024
Waterproof Connectors	Blue Trail	Various bulkheads and connectors	Purchased	3,703.94 CAD	2025
Thrusters	Blue Robotics	T200	Purchased		2024
Motor Control	Blue Robotics, JLCPCB	Basic ESC	Purchased		2024
High Level Control	N/A				
Actuators	Blue Trail	Underwater Servo SER-20XX	Purchased		2025
Propellers	N/A				
Battery	Amazon	HOOVO 2S LiPo Battery, 7.4V 8200mAh 120C			2024
Converter	JLCPCB	Power Distribution Board	Custom		2024
Regulator	JLCPCB	Power Distribution Board, Battery Management System: TI Bq76952, ACS770-150U	Custom		2025
CPU	Nvidia	Jetson Orin Nano: 1024 CUDA cores, 32 tensor cores, 8GB RAM, 25W, Jetpack 6.2	Purchased	650 CAD	2024
Internal Comm Network		Ethernet + I2C, STMG030K8T6-based PCB	Custom		
External Comm Interface		Ethernet	Purchased		
Compass	N/A				
IMU	Bosch	BMI055, inside Intel D455	Purchased	N/A	2024
DVL	Nortek	DVL 1000 Gen 3: 0.1–75m bottom track, 8Hz max	Sponsored / Donated	N/A	2024
Manipulator	Custom				2025
Algorithms	N/A	A* Pathfinding			
Vision	Intel	Realsense D455 and D405	Purchased	650 CAD (D455) + 450 CAD (D405)	2024/2025
Acoustics	Aquarian, Mouser, JLCPCB	S1 Hydrophones, Zynq 7020, filter and connector PCBs	Purchased / Custom	1598 USD	2025
Localization	N/A	**Localization:** Dead reckoning using DVL and IMU fusion **Mapping:** Point cloud insertion using Bonxai library **Rates:** 200Hz localization, 5Hz mapping	Custom	N/A	2024/2025
Autonomy	N/A	Nested state machines using PyTransitions	Custom	N/A	2025
Open Source Software	N/A	ROS2 Jazzy, IntelRealsenseSDK, PyTransitions, PyTorch, Bonxai	N/A	N/A	
Inter-Vehicle Communication	N/A				
Programming Languages	N/A	Python, C++	N/A		

TABLE I
VEHICLE INFORMATION SUMMARY FOR ROBOSUB 2025

C. Appendix C: Example Pool Test Plan and Results

Pool Test Plan Format

DATE, 2024 Pool Test

Pre-test

Inspections / tests:

1.

Notes / Things to watch out for:

1.

Pre-test Outcome:

Success

TEST NAME

Description / Goals:

1.

Steps / Outline:

1.

Results:

1.

Next Steps / Further Testing:

1.

Test Outcome:

Mixed

Overall details

Notes & comments:

1.

Next Steps / Improvements:

1.

Overall Test Outcome:

Incomplete

Fig. 16. Sample Empty Test Plan

Navigation test - Nov 22, 2024

Description / Goals:

1. Test the full navigation stack one layer at a time, starting with motor_cortex, then motion planner, then navigator
2. Be able to move based on position commands, ex. Move to (1,0,0)

Steps / Outline:

1. Send commands to each navigation layer, one at a time, and observe translation topics as well as movement characteristics visually in real life

Results:

1. Motor cortex was able to correctly calculate the desired xy_translation, however, it never stopped moving because of the localization system failure, i.e, it set a goal but never realized it reached it because it thought it wasn't moving

Next Steps / Further Testing:

1. Retry the test once the localization system is tested and fully functional

Test Outcome:

Failure

Data collection - November 22, 2024

Description / Goals:

1. Collect data for machine learning model training
2. Collect data for debugging inspection

Steps / Outline:

1. Run the ROS2 bag command and collect all topic data intermittently throughout the test

Results:

1. Forgot to collect data for most of the test
2. Collected 30 30-second clip of localization system failure in the last minute of testing, allowing for debugging of the dead reckoning node
3. We were not able to collect any meaningful data for the object detection team

Next Steps / Further Testing:

1. Implement standardized ROS2 bag collection procedures for each test

Fig. 17. November 2024 Test Plan Example

November 29, 2024 Pool Test Results

Pre-test

Inspections / tests:

1. Test dead reckoning node with artificial data from dvl_dummy_driver node and real data from camera
    a. Visualize with rviz2 and ensure all directions make sense
2. Ensure all topics are being published to and all synced subscriptions are working correctly
    a. Start from motor throttle and work backwards if not working
3. Create a document with the axis / motor / pid mapping
    a. Is clockwise yaw positive or negative?
    b. What is clockwise / positive / negative pitch and roll
    c. Motor mapping diagram
    d. https://www.andre-gaschler.com/rotationconverter/
4. Test tether and new switch
5. Verify status clients and servers are working correctly in simulation
6. Test if automated planner goes through commands correctly
    a. Inspect motor_throttle output at each stage
    b. Send ros2 topic pub /movement_command cascade_msgs/MovementCommand "{command: 1}" --once to artificially get "success" response from navigator and move through behaviour tree

Notes / Things to watch out for:

1. Carefully inspect all ros2 topic data for irregularities before finishing a test
2. Can we get a table?

Pre-test Outcome:

Success

Fig. 18. Second November 2024 Test Plan Example

Localization System Test

Description / Goals:

1. Test dead reckoning node with new changes
2. Ensure localization system is outputting reasonable pose estimates prior to navigation test

Steps / Outline:

1. Launch localization specific ros2 launch file
2. Inspect DVL outputs on /PID/axis/actual topics
3. Inspect pose message data, making sure it roughly lines up to real world translations
    a. Use foxglove and **rviz2**
    b. Test linear translation
        i. 1 meter forward / backward
        ii. 1 meter right / left
        iii. 1 meter up / down
    c. Test rotations
        i. 90/180/270/360+ degrees yaw,
            1. Check angle wrap around behaviour
        ii. +- 45 degrees pitch and roll
    d. Test continuous movements
        i. Circular arc
4. Launch navigation specific ros2 launch file
5. Inspect pose and xyz_translation data, making sure they roughly line up to real world translations
    a. Test same movements as before

Results:

1. Successful localization, no weird drifting, accurate readings

Next Steps / Further Testing:

1. Improve pitch and roll tracking using kalman filter

Test Outcome:

Success

Fig. 19. Localization Test Plan Example

Navigation System Test

Description / Goals:

1. Test the full navigation stack one layer at a time, starting with motor_cortex, then motion planner, then navigator
2. Be able to move based on position commands, ex. Move to (1,0,0), turn +90 degrees in yaw axis

Steps / Outline:

1. Send commands to each navigation layer, one at a time, and observe translation topics as well as movement characteristics visually in real life
    a. Navigator
        i. Commands are quite different from previous 2 layers
        ii. ros2 topic pub /movement_command cascade_msgs/MovementCommand "{command: 1, data0: {x: 1}}" --once
            1. 1 meter forward
        iii. ros2 topic pub /movement_command cascade_msgs/MovementCommand "{command: 1, data1: {z: 90}}" --once
            1. 90 degrees yaw

Results:

1. Successfully navigated rotational and translational movements
2. Accurate holding
3. One inner motor not turning on, stopped us from testing heave stabilization

Next Steps / Further Testing:

1. Fix all motors
2. Test heave
3. Reduce heave decay
4. Increase integration and proportional term on heave axis

Test Outcome:

Success

Fig. 20. Navigation Test Plan Example

Web interface test - Nov 22, 2024

Description / Goals:
Test the foxglove interface through the tether,

Steps / Outline:

1. Launch Foxglove Bridge
2. Connect through a laptop

Results:

1. Worked well on macOS and Windows, but very buggy on our Ubuntu laptop
2. Real-time camera data streaming and ROS2 topic data inspection worked perfectly
3. Was able to record and playback data afterwards

Next Steps / Further Testing:

1. Try to fix Ubuntu version
2. Overall success

Test Outcome:
Success

Fig. 21. Web Interface (Foxglove Bridge) Test Plan Example

D. Appendix D: Club Structure

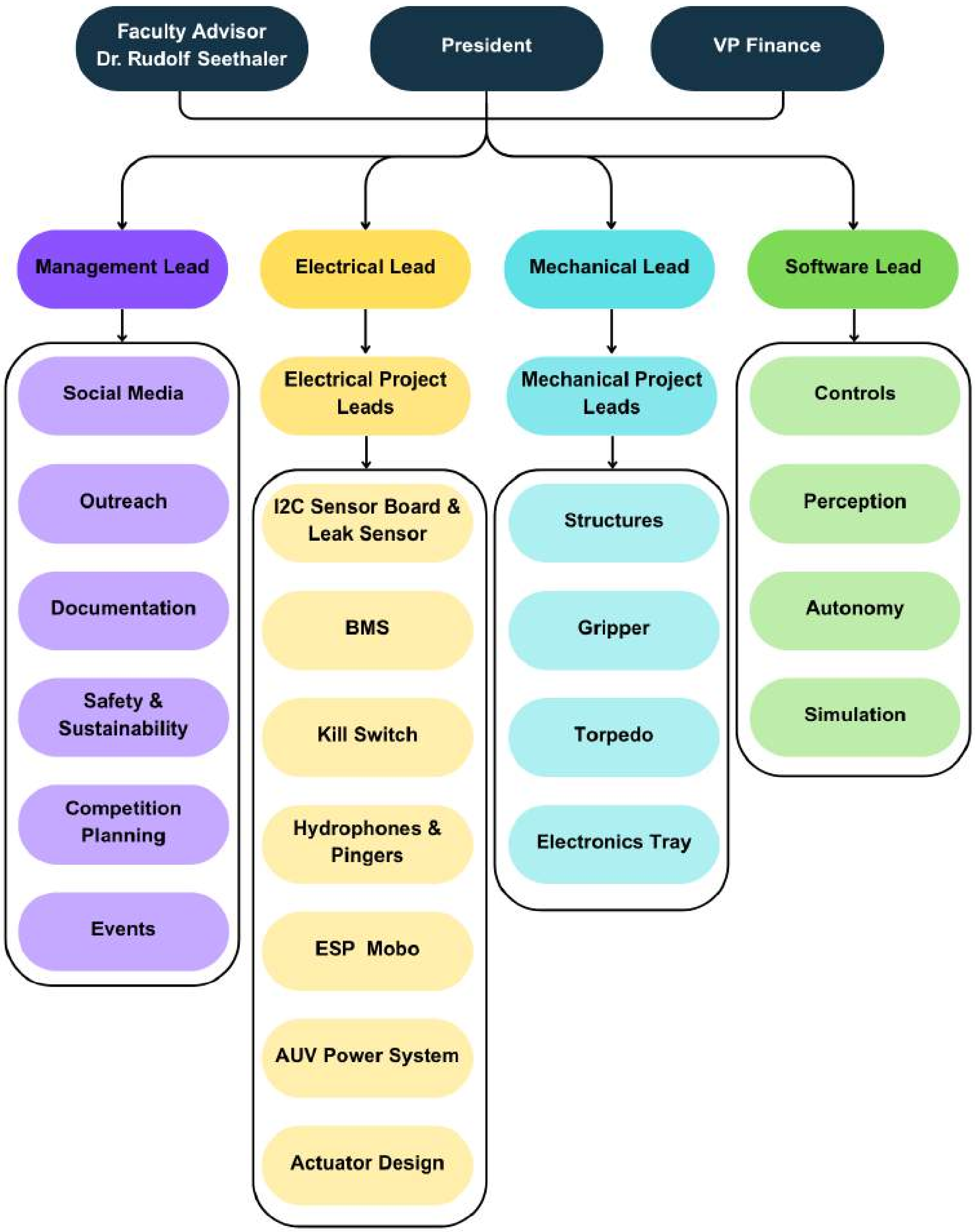

Fig. 22. Okanagan Marine Robotics club structure and projects.

E. Appendix E: Community Outreach

1) Kelowna Secondary School: Okanagan Marine Robotics led several outreach initiatives at Kelowna Secondary School (KSS), connecting with students in Grades 10 through 12. On our first visit, we met with KSS' robotics team, we introduced our team, explained how we operate as a university design group, and shared insights into how we manage sponsorships, funding, media, and competition logistics. These aspects helped show the real-world skills required to run an engineering project, beyond just the technical work.

We brought along a physical model of our Autonomous Underwater Vehicle (AUV), giving students a close-up look at the design's complexity. We showcased several functional systems still in progress, including the torpedo, grabber arm, grabber hand, leak sensors, and hydrophones. Students also had the opportunity to see early-stage builds and unfinished prototypes.

Fig. 23. Club members explaining the gripper arm to KSS robotics team.

The visit also covered our software systems. We walked through how the AUV collects data using sensors like the Doppler Velocity Logger (DVL), which tracks motion relative to the seabed, and the Inertial Measurement Unit (IMU), which includes a gyroscope and accelerometer. We explained how RGBD cameras, combined with object detection algorithms like YOLO and color filtering, are used to map the environment and identify targets. These inputs feed into our autonomy framework, enabling the AUV to make decisions, plan paths, and execute tasks, such as grasping an object, firing a torpedo, or navigating an obstacle course.

Beyond the demo, we worked directly with KSS students on their robot. Our team provided troubleshooting help, shared programming strategies, and supported them through design challenges. One of our members served as a dedicated liaison, keeping regular contact and connecting them with more specialized advice from other team members. This mentorship continued through their competition preparation, providing them with consistent support and feedback.

We also led a university application and engineering career workshop, focused on helping students understand the path to post-secondary engineering programs. We discussed how joining design teams like ours builds skills and fosters connections, and how early involvement in robotics can enhance applications and future opportunities.

Later in the year, we returned to present to the Engineering 11 class alongside other UBCO design teams, OK Motorsports and Aerial Robotics and Rocketry. Each team gave a brief presentation about their work, the tools and software they use, and the types of problems they solve. We emphasized that the balance between technical expertise and creativity was essential to our design process.

Fig. 24. Club members with our submarine at Kelowna Secondary School.

We also gave a more technical talk to the AP Physics 11 class. There, we focused on demonstrating how the physics concepts they were studying, such as motion, force, and energy, directly apply to real-world robotics design. It was a great chance to connect theory to practice and answer more detailed questions about how our systems work.

2) UBCO Prospective Student Tour: We took part in a UBCO-hosted tour for incoming and prospective engineering students. During this session, we introduced visitors to our AUV and explained how our interdisciplinary team, spanning mechanical, electrical, and software roles, comes together to build a fully autonomous system. We broke down how each subsystem fits into the bigger picture and what each subteam focuses on. The tour received a high level of interest, with many attendees expressing enthusiasm about joining or supporting the team.

3) École Élémentaire Glenmore Elementary School: We also visited a Grade 1 class as part of our elementary outreach efforts. Our goal was to introduce young students to basic STEM ideas in a way that was fun and easy to understand. We began

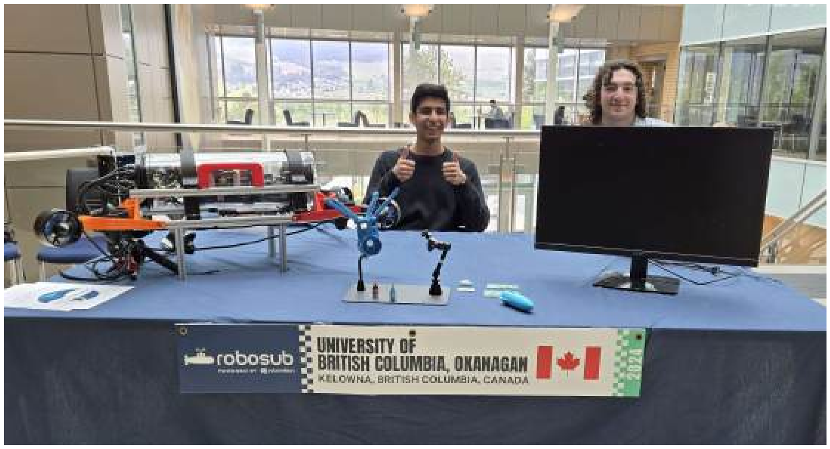

Fig. 25. Club members at our booth for the student tour.

by discussing what robots are and what marine robots can do, such as exploring underwater, taking photos of sea creatures, or cleaning up pollution.

We showed pictures of our robot and described its parts using simple language. We explained how thrusters make it move, how computers give it instructions, and how circuit boards power everything. To make things interactive, we led a "robot simulation" game where students acted like robots, following simple commands like "swim forward" or "turn right."

After that, we ran a hands-on activity where students built their pool noodle boats. They personalized their sails, decorated their boats, and then tested how many pennies their designs could hold before sinking. It was a fun way to introduce basic engineering principles, such as buoyancy and design iteration, and it gave students a chance to think creatively while learning through hands-on experience.

Fig. 26. First graders testing their pool noodle boats.

F. Appendix F: Automated Planner Diagram Examples

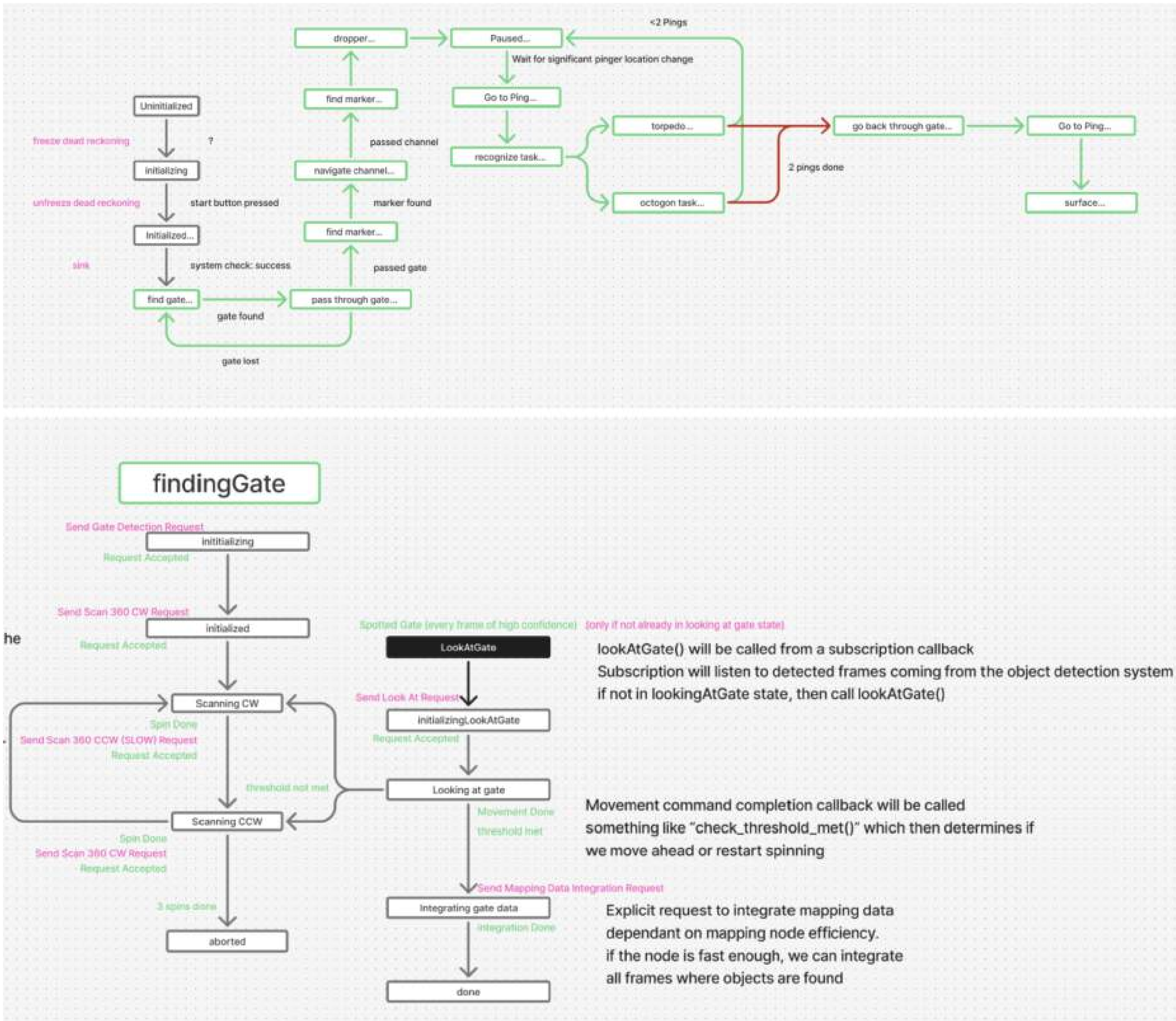

Fig. 27. Automated Planner Figma Diagrams

G. Appendix G: Competition Strategy Risk Mitigation Plan

To ensure the successful execution of our ambitious 2025 competition strategy, we have identified key risks tied to our primary and secondary goals. The following table outlines these risks and our corresponding mitigation strategies.

Risk Area	Impact on Competition Strategy	Mitigation Plan
Actuator & Acoustics Integration Delay	Prevents attempts at secondary goals (Torpedo, Gripper, Random Pinger tasks). Does not impact primary task performance.	• **Decoupled Development:** Software for these subsystems is being developed and tested in the Stonefish simulator, independent of hardware availability. • **Primary Task Priority:** Team resources are prioritized for completing and validating all primary tasks before dedicating full integration time to secondary goals. • **Tiered Approach:** If full integration is delayed, we will attempt simplified versions to test partial capability.
Object Detection Underperformance	Severely compromises all vision-based primary goals (Gate, Slalom, Dropper) by failing to reliably identify task elements.	• **Stable Fallback:** The previous, functional YOLO-based model is maintained as a stable branch for immediate deployment if the new ResNet model fails validation. • **Aggressive Data Collection:** Every pool test is used to capture ROS2 bag data, continuously expanding our training dataset to improve model accuracy.
Navigation System Instability	Degrades precision required for all tasks and could cause failure in long-horizon missions like the Return Home task.	• **Targeted Reimplementation:** The localization node has been fully reimplemented in C++ to enhance performance and reliability, addressing known 2024 bottlenecks. • **Formalized Tuning:** We use documented test plans (Appendix C) and data logging to systematically tune PID and Dead Reckoning systems during weekly pool sessions.

H. Appendix H: Electrical Test Plans

Described below is the simple procedures followed by our electrical team to effectively test the I2C Sensor Boards and the Hydrophone Filter Board.

I2C Communication Test

Description / Goals:

1. "Measure the time required for a complete data communication cycle over the I2C bus, with increasing numbers of sensor nodes."

Steps / Outline:

1. Connect one I2C Sensor Board to the ESP-Mobo board using the I2C protocol
2. Connect the I2C chain to channel 1 of an oscilloscope
3. Adjust the vertical scale and horizontal scales to fit the entire signal on the screen
4. Adjust the oscilloscope cursors to the beginning and end of the complete I2C transaction
5. Record the time from the beginning to the end of the I2C signal
6. Repeat steps 3-5 five times, adding an I2C sensor board to the communication bus each time

Results:

1. Observed that each additional sensor added approximately 0.540ms to the communication time

Next Steps / Further Testing:

- Determine the effects of peripherals, such as leak sensors, attached to each sensor board, on the total cycle time
-

Testing Outcome:

Success

Fig. 28. I2C Bus Test Plan

Hydrofilter Functionality Test

Description / Goals:

1. Evaluate the ability of the hydrofilter board to convert various input waveforms into clean sinusoidal outputs, and assess overall signal fidelity

Steps / Outline:

1. Generate a 0.3 MHz waveform with a 1.0 V peak-to-peak amplitude using a signal generator.
2. Connect the signal generator output to channel 1 of an oscilloscope and the input of the hydrofilter board
3. Connect the output pin of the hydrofilter board to channel 2 of the oscilloscope
4. Adjust channel 1 to 500mV/division, and channel 2 to 1V/division to account for amplitude change
5. Cycle through sine wave, triangular wave, square wave, sawtooth wave, and modulated wave, recording the output of each

Results:

1. The hydrofilter board consistently produced sinusoidal outputs across all input waveform types, preserving the original frequency with minimal noise

Next Steps / Further Testing:

- Test the hydrofilter board with noisy functions
- Test the hydrofilter board signals generated by the hydrophones

Testing Outcome:

Success

Fig. 29. Hydrophones Filter Test Plan