Technical Design Report of Matsya 7B, Autonomous Underwater Vehicle

Sparsh B, Anirudh J, Khushajh V, Shubham T, Anuj B, Mayank B, Anurag D, Anshu A, Adwai K, Ayush P, Urvi G, Manit J, Divyanshu G, Aniket G, Jigar M, Anuj P, Divya M, Shresth K, Shlok R, Kiara T, Pratham C, Dhruv S, Denver Z, Aayush S, Ankush, Preetham P, Abbas H, Shivansh G, Shivakshi G, Tvisha M, Avadhoot K, Shur R, Vedant B, Arya M, Mantra S, Harith S, Aditi N, Tarun K, Sahil L, Soumya B, Jai S, Jainam R, Pratyush T
Faculty Advisors: Prof. Shashi Ranjan and Prof. Leena Vachhani

Abstract—**For RoboSub 2026, AUV-IITB presents two vehicles: Matsya 7B and Mini Matsya. The vehicles are assigned complementary tasks, Matsya 7B handles manipulation-heavy tasks (Bin, Torpedoes, Octagon) with redesigned fin-ray gripper; Mini Matsya targets motion-dominated tasks (Gate, Slalom) using DVL-free localisation. Mechanical upgrades to Matsya 7B include a toggling gripper mount extending reach by 20 cm, integrated camera cooling, and an IR-rejecting hull reflector. Electrical improvements add over/undervoltage protection, a single-source PWM demultiplexer for eight thruster channels, current telemetry, and an in-house OLED-menu debug board for rapid fault isolation. The perception stack offloads a custom four-channel RGB-D YOLOv11 detector to the OAK-D Myriad X VPU. A distance-weighted long-short range PID with a jerk-limited trajectory planner stabilises transit and close-in alignment under a single controller, and a history-aware spatio-temporal weighted estimation layer with task-tuned weights filters detection noise before navigation. Mini Matsya layers an APF reactive obstacle-avoidance module over PID and pursues an ESKF/GTSAM trade study for DVL-free localisation, while a Gazebo simulator carrying the full competition prop set enables off-pool mission rehearsal across all five tasks.**

I. Competition Strategy

A. Core Strategy

Our core objective for RoboSub remains 100% task completion accuracy. To achieve this, we are undertaking a comprehensive overhaul and optimization of all critical mechanical, electrical, and software systems of Matsya 7A to develop Matsya 7B, incorporating key learnings from the vehicle's first deployment at RoboSub 2025. The overhaul is focused on significantly improving the vehicle's overall reliability, robustness, and functionality.

In parallel, we are also developing a new mini-AUV platform. While this has resulted in slower short-term progress, particularly in hardware development and testing, it is a deliberate strategic decision aimed at maintaining long-term competitiveness and accelerating future development cycles.

B. Course Strategy

For RoboSub 2026, AUV-IITB will deploy two vehicles: Matsya 7B, and Mini Matsya, a newly developed compact 5-DOF platform. Task allocation between the two is determined by matching task requirements

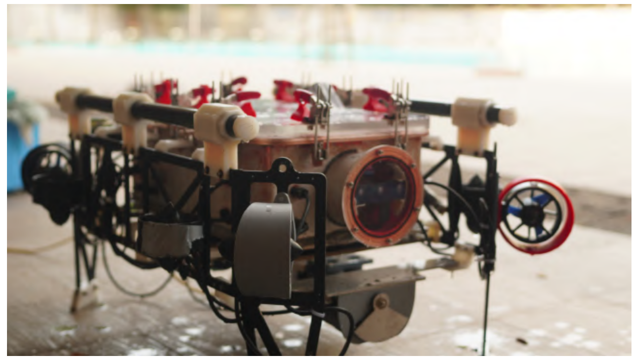


Figure 1: Matsya 7B

to each platform's actuation, sensing, and dynamic characteristics.

Matsya 7B is configured for tasks requiring precise stationkeeping and actuation. Its sensor suite includes a Doppler Velocity Log (DVL), and it carries the redesigned gripper. These attributes are suited to tasks that depend on alignment to small target windows namely *Bin*, *Torpedoes*, and *Octagon*. Mini Matsya is purpose-built for the motion-dominated tasks. Without a DVL, its localization is unaffected by aggressive roll manoeuvres, so it can claim full *Gate* style points via a barrel roll without carrying drift into subsequent tasks. Its smaller footprint and lighter frame also make it the better fit for the *Slalom* course, where clearance and agility matter more.

Running two vehicles in parallel offers two core benefits. A failure on one vehicle is isolated to its assigned tasks, so the rest of the run still scores. And by working in parallel, the effective run time available for scoring is greater than on a single-platform run. The assignment of tasks follows the actuation, maneuverability, vision, and acoustics requirements in Table I.

Task	Capability Requirement				Vehicle
	Act.	Manv.	Vis.	Acou.	
Gate	No	Yes	Yes	No	Mini Matsya
Slalom	No	Yes	Yes	No	Mini Matsya
Bin	Yes	No	Yes	No	Matsya 7B
Torpedoes	Yes	No	Yes	Yes	Matsya 7B
Octagon	Yes	No	Yes	Yes	Matsya 7B

Table I: Task-requirement-vehicle assignment.

- **Gate [Mini Matsya]:** We will opt into both the *Coin Flip* and *Random Role Assignment*; the vehicle then performs a yaw scan to align with the assigned side, passes through the gate, and executes a 720° barrel roll to maximize style points.

- **Slalom Poles [Mini Matsya]:** We will use SLAM to localize the red and white poles, then an Artificial Potential Field controller treats the poles as repulsive obstacles to weave through the poles on the side dictated by the assigned role.
- **Bin [Matsya 7B]:** This will be the first task after passing through the gate for the main vehicle. We will leverage the Map layer for robust bin localization and drop markers into both designated bins based on the assigned role.
- **Random Pinger [Matsya 7B]:** With acoustic detection reliable in 75% of test pings, the vehicle uses a left-or-right side-selection strategy instead of a continuous heading estimate, trading angular precision for decision robustness.
- **Torpedoes [Matsya 7B]:** The vehicle will fire torpedoes through both holes dictated by the assigned role from a standoff of about 50 cm.
- **Octagon [Matsya 7B]:** The vehicle will attempt to lift each object once and drop it into the designated bin; time-bound exit strategies guarantee that, at minimum, it surfaces with the right heading.

C. System Overhauls

Each subteam's overhaul targets a specific shortcoming observed during the 2025 run.

Mechanical Overhaul

The gripper has been redesigned for stronger grip and wider object compatibility, addressing inconsistent grasps during *Octagon* pick-and-place, and is mounted on a pin-based two-position platform that extends reach. Battery hulls have been removed and the batteries relocated into the main hull, yielding a more compact, lighter vehicle with one fewer seal failure mode. The hull's internal layout has been redesigned around thermal management with a dedicated camera-cooling fan and direct metal contact between heated components and the hull.

Electrical Overhaul

Communication reliability and long debugging cycles were the two main shortcomings from 2025. The communication architecture has been redesigned around a more robust UART-based framework, and dedicated in-house debug hardware now enables rapid validation and fault isolation. Power delivery and protection have also been strengthened.

Software Overhaul

The control loop now uses two gain regimes, with a trajectory planner generating jerk-bounded setpoints to eliminate startup transients. The perception stack offloads YOLO [13] inference to the OAK-D's VPU and uses a four-channel RGBD [15] detector for improved robustness. At the mission level, time-bound exit strategies and an adaptive map-correction layer keep runs alive through vision loss and localization drift, and a Gazebo simulator [12] replicating the competition environment enables mission rehearsal off-pool.

II. Design Strategy

Matsya 7B

1) Mechanical Updates

The Gripper Mechanism

Last year, the gripper was based on the four-bar chain mechanism with rigid PLA-based fingers. The gripper was unable to generate sufficient normal force and was suited only for certain kind of geometries; moreover, it had limited reach. To counteract this, a series of upgrades were made.

- *Gripper Mounting Platform:* To enhance reach, the frame mounting the gripper has been given the ability to toggle between two positions mechanically. This ensures that while on ground, the vehicle's gripper is at a safe height, but when in water, the height of the gripper is brought down by nearly 20 cm, ensuring sufficient reach.

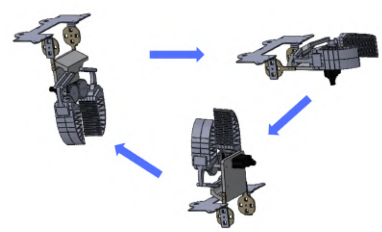


Figure 2: Gripper Height Toggle

- *Soft Robotics:* We used a fin-ray structure [1][2][3], a compliant mechanism where the structure deflects toward the contact point, naturally conforming to curved geometries and printed the fingers using Thermoplastic Polyurethane that helps in elastic recovery across repeated cycles. The final design balances the trade-off between flexibility to assume the shape of the geometry and rigidness to apply enough normal force on the body to hold it in place.
- *Prevention of object slipping:* The link lengths were calibrated to maximize the amount of servo torque being transmitted as normal force to the object, holding it in place. The internal surface of the fingers was also made more rough to increase friction.

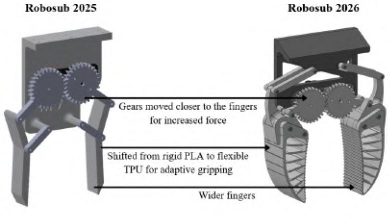


Figure 3: Gripper Finger Progress

Cooling Solutions

The hull was expected to have aggravated heating problems as the batteries have been brought inside the main hull. The following changes have been made to mitigate this problem this year:

- *Prevention of Hot Spots:* To prevent the accumulation of high temperatures, convective cooling is performed using fans. Eg: for the camera, the custom fan-mount [5] directs heat away, enabling a temperature drop from 65°C to 49°C (Appendix O).
- *Greenhouse Effect:* The hull experiences greenhouse effect since it has a clear acrylic endcap. This leads to high temperatures in the hull. To prevent this, we have introduced an aluminum-based reflector, rejecting nearly 98% of IR rays, reducing solar heat gains from 21.6W to less than 0.5W.

	Before	**After**
SBC Temperature	84–88 °C	60–63 °C
Camera Temperature	65 °C	49 °C
Solar Heat Gain (Greenhouse Effect)	21.6 W	<0.5 W

Table II: Thermal Performance Improvement

2) *Electrical Updates*

Under-Voltage, Over-Voltage and Reverse Polarity Protection

Dedicated input protection circuitry was integrated using the **LTC4365** [9] overvoltage/undervoltage protection controller with back-to-back MOSFETs configured as an ideal diode stage. The circuit continuously monitors the supply rail and autonomously isolates downstream electronics when the input exceeds programmable thresholds, while simultaneously blocking reverse polarity and reverse current conditions. Programmable resistor arrays in the threshold-setting network allow hardware-level reprogramming of UV/OV limits.

Dedicated Power Regulation for Pico

Initially, powering the Raspberry Pi Pico from the UART bridge 5V rail caused unstable startup behavior and unintended PWM outputs during battery connection. This was resolved by introducing a dedicated onboard buck converter to provide an isolated regulated 5V supply to the Pico.

Internal Telemetry Monitoring

To improve fault detection and system awareness, internal **pressure and temperature sensors** were integrated within the hull for continuous monitoring. Pressure measurements enable identification of abnormal conditions such as hull depressurization caused by seal failures or water ingress events, while temperature monitoring near the SBC provides visibility into thermal behavior during operation. These measurements also provide feedback to the software stack for real-time health monitoring and safety responses.

PWM Demultiplexing Architecture

The limited GPIO availability of the Raspberry Pi Pico restricted direct generation of independent PWM signals for all eight thruster channels. To overcome this, a **PWM demultiplexing architecture** was implemented, where a single PWM source is sequentially routed across eight outputs using digital select lines. Since standard ESC signals operate at **50 Hz** with pulse widths of approximately **1100–1900** μ**s**, the available **20 ms** frame provides sufficient timing margins to transmit valid pulses to all channels.

Actuator State Monitoring

To improve actuator reliability and enable early fault detection, **INA219**-based [10] current monitoring was integrated for continuous monitoring of servo channels. Real-time current measurements allow detection of abnormal conditions such as stalled servos, excessive loading, and unexpected current spikes during operation. The current signatures are also used to provide feedback to the software stack during manipulation tasks, where an increase in gripper motor current acts as an indicator of successful object grasping.

Function	**Component**	**Purpose**
Servo monitoring	INA219	Stall detection and grasp feedback
Input protection	LTC4365	UV/OV and reverse polarity protection
Power regulation	Buck converter	Dedicated regulated Pico supply
PWM distribution	Demultiplexer IC	Multi-channel thruster control with reduced GPIO usage

Table III: Electrical monitoring, protection, and signal distribution features integrated into Matsya 7B

3) *Software Updates*

Vision

YOLO inference has been offloaded from the host SBC onto the OAK-D's on-chip Myriad X VPU. The DepthAI [14] driver was migrated from v2 to v3 and collapsed from two parallel host-side processing nodes (one for RGB, one for depth) into a single unified node that emits a hardware-synchronized RGB-D frame (Appendix F.2). The base detector has been upgraded from YOLOv8 to YOLOv11, and a custom four-channel RGB-D variant of YOLOv11 (Appendix F.3) has been built in-house, feeding co-registered depth into the network as a fourth input channel alongside RGB so that depth cues contribute to detection from within the model itself. Mid-fusion emerged as the strongest configuration (Table IV): mAP@50-95 rises from 0.9388 to 0.9658 at roughly a 1.4ms increase in the inference latency.

Metric	YOLOv8n	YOLOv11n	RGB-D (mid)
mAP@50-95	0.886	0.9388	**0.9658**
mAP@50	0.992	0.9950	0.9950
Precision	0.995	0.998	1.000
Recall	0.996	0.999	1.000
Validation loss	1.58	1.43	1.27
Inference latency	1.9 ms	1.3 ms	2.7 ms

Table IV: Detector performance comparison

Mission Control

We added an exit strategy for every actuator task that, on a timeout, attempts the task on a best-effort basis using the last good navigation estimate and an emergency scan that recovers the target when vision drops out suddenly or the vehicle aligns incorrectly. Together, these additions make mission control resilient to partial task failures. (Appendix F.10).

Controller and Navigator

The controller operates in two discrete gain regimes. The long-range regime is tuned for stable, sustained surge motion over long distances, and the short-range regime for the precise short-distance manoeuvres required during actuation (Appendix F.1). A trajectory planner [26] sits between the mission planner and the PID loops and converts each waypoint into a jerk-limited setpoint stream. This addresses two issues: large steps produced abrupt thruster starts and erratic motion at the beginning of a transit, and a single far-away setpoint did not let error on the other degrees of freedom accumulate fast enough for the controller to act on it, so off-axis drift went uncorrected until it grew large. Streaming intermediate setpoints keeps every axis under active correction and removes the discontinuity at the regime switch (Appendix F.4).

Map & Localisation

We added a history-aware, spatial-filtering layer between the detection pipeline and the mission control wherein all inertial-frame detections corresponding to a task are logged into a comprehensive temporal buffer. Three parallel algorithms are then applied to the data: Spatio-Temporal Weighted Estimation [23], which locates the highest-density cluster within a set radius and then returns a representative point using a composite weight derived from spatial distance, temporal decay, and model confidence, Histogram-Based Filtering which places detections into a fixed 2D grid, designating the highest-frequency cell as the target to reject gross outliers, and DBSCAN Clustering [22] which isolates dense point clouds and drops sparse points as noise, to yield maximum spatial resolution.

This layer is designed to handle the issue of occasional false positives caused by refraction, backscatter, or dynamic lighting during underwater visual detections, which we faced during Semi-Final 1 of RoboSub 2025. The parallel algorithms resolve this by offering distinct precision-versus-accuracy profiles, while DBSCAN provides the spatial precision required for strict tasks like the torpedo launch, histogram filtering ensures robust general-vicinity localization for tasks like the bin drop and STWE balances both by optimizing for temporal relevance and detection confidence. Because the exact tolerances for every task are known a priori, the mission control fuses the outputs of all three methods using predetermined, task-specific weights. This leverages the strengths of each algorithm to deliver a single, optimally stabilized coordinate to the navigation filter.

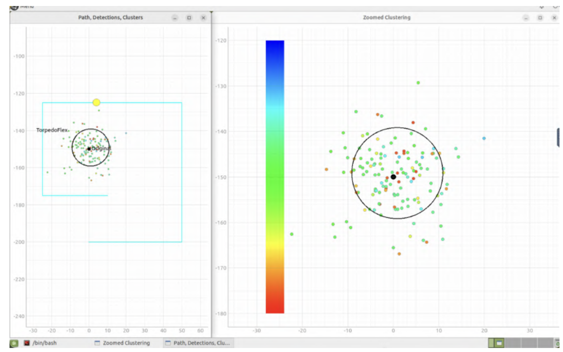


Figure 4: Map STWE clustering demonstration.

Mini Matsya

1) Mechanical Development

Mini Matsya's design philosophy prioritised cost and time efficiency in line with its intended role at RoboSub '26. This drove decisions that reduced manufacturing complexity, yielding a vehicle built in under 5 months at one-fifth the cost of Matsya 7. To further simplify the design, Mini Matsya operates with 5 degrees of freedom, requiring enhanced measures for mechanical pitch stability (see Appendix D).

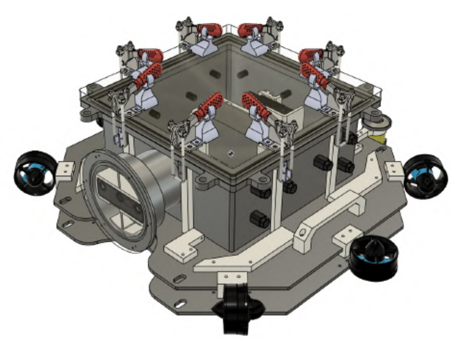
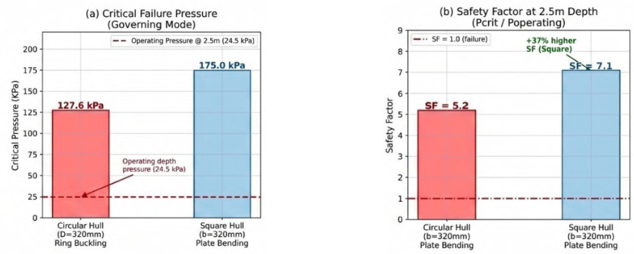


Figure 5: Mini Matsya

Designing and Manufacturing

- *Hull shape:* A cuboidal hull was chosen, due to cost-efficient and readily available raw material. We chose a cuboidal hull over a cylindrical hull for its superior performance in static structural simulations (safety factor and critical failure pressure).
- *Manufacturing:* Manufacturing was largely centred around fast processes like laser cutting, metal bending, and welding. Milling was intentionally avoided to save on cost and time (Appendix G).



(a) Critical Pressure (b) Safety Factor

Figure 6: Superiority of Cuboidal Hull

2) *Electrical Development*

The electrical architecture was designed with emphasis on modularity, reduced wiring complexity, and ease of debugging. Mini Matsya adopts a dedicated two-board architecture consisting of a Power Board and a Signal Board. Separating high-current power handling from communication and control circuitry reduces electrical interference and simplifies system integration.

Dedicated Power Distribution Board

A dedicated power board was developed to provide a compact and monitorable power architecture. The board integrates power distribution, protection circuitry, and telemetry onto a single PCB, reducing wiring complexity, connector count, and potential failure points. The board was designed to support sustained current operation up to 60A, with thermal performance treated as a primary design consideration. Wide high-current planes and 2 oz copper pours were employed across power paths to minimize resistive losses and localized heating. Integrated voltage and current telemetry additionally provide real-time visibility into power consumption, enabling rapid fault detection.

Signal Board

The Signal Board retains the communication and interfacing framework developed for Matsya 7B while being redesigned into a more compact form factor for Mini Matsya's reduced footprint. Reusing a validated signal framework reduced integration effort and enabled faster system bring-up.

Thruster Current

Reliable thruster current monitoring was implemented using the **INA3221** [8], providing stable multi-channel current sensing for detection of abnormal loading, stalled thrusters, and transient current spikes. The monitored current signatures additionally enabled rapid identification of unintended PWM outputs and controller-side anomalies within the software stack.

Power Delivery Architecture

Initial Mini Matsya prototypes powered camera modules directly through the **Jetson Orin Nano**, introducing additional loading and reducing power stability. The final architecture incorporated a **powered USB hub** and an **HW-773 USB PD trigger module** to offload peripheral power from the SBC and improve overall system reliability.

3) *Software Updates*

The Mini Matsya software stack is shaped by two core challenges: the vehicle has to localise without a DVL, and it has to navigate around obstacles in its path.

Localisation and Vision

Mini Matsya operates over a small course area and the camera sees a near-constant set of competition props through the run, giving the estimator a stable set of landmarks to anchor against. We are exploring two options. The first is a modified ESKF [17] where the vision detections act as the source of ground truth and are folded in as direct pose corrections on top of the IMU-driven nominal state. The second is a modified GTSAM factor-graph smoother [11] that fuses the same vision and IMU streams over a sliding window. Both are being characterized in pool tests. (Appendix F.8).

PID Controller with APF

The controller is a standard PID loop with an Artificial Potential Field (APF) [25] module layered on top (Appendix F.9). The APF takes the tracked obstacles, generates repulsive forces from them, and adds these to the PID translational output before allocation. The APF only deflects the path when an obstacle enters the influence radius. The deflection is restricted to surge and sway; heave is held at a fixed depth and the rotational axes stay under pure PID.

Allocator

The allocator is responsible for translating the 5-DOF body force and torque vector commanded by the controller into individual forces for the six thrusters. This relationship is governed by a control effectiveness matrix, which is constructed column-wise using the force direction of each thruster and the moment arm of its line of action about the centre of mass. To ensure physical feasibility, the resulting thruster forces are saturation-capped to remain within operational limits and subsequently translated to PWM signals via an empirically tuned lookup table (Appendix F.7).

Let $\boldsymbol{\tau} \in \mathbb{R}^5$ be the commanded vector, $\mathbf{u} \in \mathbb{R}^6$ be the thruster force vector, and $A \in \mathbb{R}^{5\times 6}$ be the control effectiveness matrix, thus:

$$\boldsymbol{\tau} = A\,\mathbf{u} \quad \Longrightarrow \quad \mathbf{u} = A^{+}\,\boldsymbol{\tau} \tag{1}$$

where A^{+} is the Moore-Penrose pseudo-inverse, precomputed at startup to minimize runtime overhead.

III. Testing Strategy

Autonomous Test Runs and Validation

Software verification is organised in three layers, each targeting a different class of fault before it reaches the vehicle.

1) Simulator Testing

We validate new code paths in the Gazebo simulator (Appendix F.6) before deploying to the pool. The simulator mirrors the production sensor and control interfaces and carries the full competition prop set at field-representative positions. This allows resource-intensive mission rehearsals to be run virtually, ensuring that in-water time is reserved strictly for true hardware and environmental validation.

2) Pool Testing

Before the vehicle is submerged, a bench check on the electrical stack and a sensor-driver dry run are completed on shore. Once in the water, testing follows a strict, bottom-up progression: initial runs focus on mapping corresponding thrusters in the software and electrical stack, then tuning the force allocators for the thrusters, followed by baseline controller calibration. With vehicle stability confirmed, the vision pipeline is verified against the live underwater environment. Only after these subsystems are individually validated does the team proceed to full mission control runs (Appendix C). During all pool slots, on-board bags are recorded to capture sensor data and vehicle state for offline review and debugging.

Electrical Stack Verification

To reduce turnaround time and improve fault localization, a dedicated in-house verification board was developed with an OLED-based menu-driven UI, enabling rapid subsystem testing. The board integrates multiple diagnostic and validation capabilities into a single interface:

- *Thruster Validation & Signal Measurement:* Individual PWM signals can be dispatched to each of the eight thrusters to verify correct actuation, direction, and channel assignment prior to deployment.
- *Signal Path Verification and Communication Integrity:* End-to-end validation of the thruster command chain is performed by comparing PWM commands generated on the SBC side with the PWM signals measured at the thruster inputs. By verifying that the final received signals match the commanded values, communication failures, signal corruption, channel mismatches, and faults introduced at intermediary stages such as the microcontroller interface can be rapidly identified.
- *Connectivity Verification:* Due to the Ethernet spool length and repeated stress during operation, connector loosening and cable faults can occur. Thus, the board functions as an active continuity and short-circuit tester for rapid fault detection.

Figure 7: 3D model of the debugging board

Mechanical Verification

Before the electrical stack is assembled onto the vehicle, the mechanical subdivision ensures that the vehicle is completely leak-proof, through a series of methodical steps to capture failure points.

Table V: Waterproofing Test Methods

Method	Time	Target Failure Points	Identification Method
In Air Water proofing: The Hull is pressurised to 125 kPa, and then pressure is monitored for one hour	1 hr	Overall test (weld joints, o-ring, penetrators)	The internal hull pressure is monitored; in case of a dramatic pressure drop, a soap solution is applied to the hull to identify leaks. Over the one-hour period, the vehicle is considered leak-proof if the pressure drop is gradual and less than 0.5 kPa (corresponding to a leak rate of 0.01 cm^3/min). This method cannot detect very small leaks, so an in-water test is also performed.
Submerged waterproofing test: Vehicle is submerged in an 8 ft deep pool for a given time period.	15 min	O-rings (unpressed or damaged)	Cameras inside the hull are used to identify exact spots from where water is entering. When o-rings are not pressed, the ingress of water is relatively large, hence this failure can be identified quickly.
	1 hr	Faulty penetrator	Zip-lock bags are placed on the penetrator openings. In case a penetrator is faulty, water is collected in the corresponding bag. Inflow of water in such failures is relatively slower.
	3 hr	Weld failures or cracks in hull	Litmus paper is applied to weld seams and parts of the hull with suspected failures. The leakage is slowest in this case as the failures are minute, hence a long testing time is required. The failure is tracked by identifying the wet litmus papers.
	12 hr	Overall waterproofing	Only after the hull's interiors are absolutely dry, even after being submerged for 12 hours, is the vehicle approved to be used.

IV. Acknowledgements

The team extends heartfelt gratitude to Prof. Leena Vacchani and Prof. Shashi Ranjan Kumar for their mentorship and technical guidance. The team also thanks the Industrial Research and Consultancy Centre, the Swimming Pool Administration, and the Desai Sethi School of Entrepreneurship for providing monetary support, testing facilities, and workspace, respectively. We are extremely grateful to our sponsors for their support:

- Luxonis (Camera Sponsors)
- NCF Labs (Rapid Prototyping Partner)
- Solidworks
- Ansys

V. References

[1] M. S. Sofla, H. Golshanian, E. I. Sklar, and M. Calisti, "Modeling and modification of fin-ray effect grippers to improve their load capacity and grasp stability," *Sensors and Actuators A: Physical*, vol. 392, p. 116711, 2025, ISSN 0924-4247. doi: 10.1016/j.sna.2025.116711.

[2] G. L. Srinivas, A. Javed, and L. M. Faller, "Versatile 3D-printed fin-ray effect soft robotic fingers: lightweight optimization and performance analysis," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, p. 382, 2024. doi: 10.1007/s40430-024-04957-0.

[3] Y. Yang, K. Jin, H. Zhu, G. Song, H. Lu, and L. Kang, "A 3D-Printed Fin Ray Effect Inspired Soft Robotic Gripper with Force Feedback," *Micromachines*, vol. 12, no. 10, p. 1141, 2021. doi: 10.3390/mi12101141.

[4] J. Hernandez, M. S. H. Sunny, J. Sanjuan, I. Rulik, M. I. I. Zarif, S. I. Ahamed, H. U. Ahmed, and M. H. Rahman, "Current Designs of Robotic Arm Grippers: A Comprehensive Systematic Review," *Robotics*, vol. 12, no. 5, 2023. doi: 10.3390/robotics12010005.

[5] A. Jebelli, N. Lotfi, M. S. Zare, and M. C. E. Yagoub, "Advanced Thermal Management for High-Power ICs: Optimizing Heatsink and Airflow Design," *Applied Sciences*, vol. 14, no. 20, p. 9406, 2024. doi: 10.3390/app14209406.

[6] J. S. Barrowman, *The Practical Calculation of the Aerodynamic Characteristics of Slender Finned Vehicles*. Master's thesis, The Catholic University of America, Washington, D.C., USA, March 1967. NASA/TM-2001-209983.

[7] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK: John Wiley & Sons Ltd., 2011. doi: 10.1002/9781119994138.

[8] Texas Instruments, "INA3221 26V, Triple Channel, 13-Bit, I2C Output Current and Voltage Monitor With Alerts," Datasheet SBOS576C, Mar. 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina3221.pdf

[9] Analog Devices, Inc., "LTC4365: Overvoltage, Undervoltage and Reverse Supply Protection Controller," Datasheet, Rev. E, 2016. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ltc4365.pdf

[10] Texas Instruments, "INA219 Zerø-Drift, Bidirectional Current/Power Monitor With I2C Interface," Rev. G, Datasheet, Dec. 2015. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina219.pdf

[11] F. Dellaert and M. Kaess, *Factor Graphs for Robot Perception*, Foundations and Trends in Robotics, Vol. 6, 2017. [Online]. Available: http://www.cs.cmu.edu/~kaess/pub/Dellaert17fnt.pdf

[12] Open Robotics, "Gazebo Simulator," [Online]. Available: https://gazebosim.org/home

[13] Ultralytics, "YOLO by Ultralytics," GitHub repository. [Online]. Available: https://github.com/ultralytics/ultralytics

[14] Luxonis, "DepthAI Documentation (v3)," [Online]. Available: https://docs.luxonis.com/software-v3/depthai/

[15] D. Wan, R. Lu, Y. Fang, X. Lang, S. Shu, J. Chen, S. Shen, T. Xu, and Z. Ye, "YOLOv11-RGBT: Towards a Comprehensive Single-Stage Multispectral Object Detection Framework," *arXiv preprint arXiv:2506.14696*, 2025.

[16] D. Deng, "DBSCAN Clustering Algorithm Based on Density," in *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, 2020, pp. 949–953, doi: 10.1109/IFEEA 51475.2020.00199.

[17] T. I. Fossen, "Feedback error-state Kalman filter with time-delay compensation for hydroacoustic-aided inertial navigation of underwater vehicles," *Control Engineering Practice*, vol. 138, p. 105603, 2023, doi: 10.1016/j.conengprac.2023.105603.

[18] Open Robotics, "ROS 2 Humble Documentation," [Online]. Available: https://docs.ros.org/en/humble/index.html

[19] S. Macenski, A. Soragna, M. Carroll, and Z. Ge, "Impact of ROS 2 Node Composition in Robotic Systems," *IEEE Robotics and Automation Letters (RA-L)*, 2023, doi: 10.48550/arXiv.230 5.09933. [Online]. Available: https://arxiv.org/abs/2305.09933

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

[21] "Iterative Closest Point Algorithm," Stanford CS164 Course Notes. [Online]. Available: https://graphics.stanford.edu/courses/cs164-09-spring/Handouts/paper_icp.pdf

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. KDD*, 1996. [Online]. Availabl e: https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf

[23] I. Portugal, P. Alencar, and D. Cowan, "A framework for spatial-temporal cluster evolution representation and analysis based on graphs," *Scientific Reports*, vol. 14, no. 1, p. 2 2137, Sep. 2024, doi: 10.1038/s41598-024-72504-x.

[24] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017, doi: 10.1109/TRO.2016.2597321.

[25] Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In: Cox, I.J., Wilfong, G.T. (eds) Autonomous Robot Vehicles. Springer, New York, NY. https://doi.org/10.1007/978-1-4613-8997-2_29

[26] Han CJ, Song KR, Rim U-R. An asymmetric S-curve trajectory planning based on an improved jerk profile. Robotica. 2024;42(7):2184-2208. doi:10.1017/S0263574724000031

Appendix A: Component Specifications (Matsya 7B)

Component	Vendor	Model/Type	Specs	Cost (new)	Year of Purchase
Buoyancy Control	Designed In-House	Dead Weights & Foam	-	-	-
Frame	Designed In-House	Aluminium & Delrin	4 kg	200 USD	-
Waterproof Housing	Designed In-House	Aluminium Hulls w/ Acrylic End-cap	1 Hull weighing 20 kg Depth Rating : 70 ft	800 USD	-
Waterproof Connectors	Designed In-House	Aluminium	27 connectors weighing 1.7 kg in total	405 USD	-
Thruster	Blue Robotics	T200	5.25 and 4.1 kgf forward and backwards	1840 USD	2025
Motor Control	Blue Robotics	Basic ESC	30A PWM controlled brushless motor speed controller	200 USD	2025
High Level Control	Raspberry Pi	Pico	ARM Cortex M based dual core	5 USD	2025
Actuators	Blue Trail Engineering	SER-201X Economy	Stroke Length: 25mm	30 USD	2025
Battery	SkyCell	LiPo Battery	4 Cell and 16000mAh x 2	400 USD	2025
Converter	-	200W DC-DC Buck converter	Designed in-house DC-DC switching supply circuit which supports upto 20A.	50 USD	-
Regulator	Mini-Box	M4ATX	High efficiency 250W output, $< 1.25mA$ standby current	80 USD	2023
CPU	AMD	Ryzen 5 7600X	6 Cores (4.7GHz), 32GB RAM	180 USD	2026
Internal Comm Network	-	UART, USB	1 Mbps operation limit	5 USD	-
External Comm Interface	-	Ethernet	10-100 Mb/s	-	-
Inertial Measurement Unit (IMU)	Bosch	BNO085	Consists of accelerometer and 6-axis gyroscope	30USD	2025
Doppler Velocity Log (DVL)	Waterlinked	A50	Depth rating of 300m, +-1.01% accuracy	8710 USD	2022
Camera(s)	Luxonis	OAK-D S2, OAK-W	Stereo vision based cameras used for depth perception	330 USD	2025
Hydrophones	Teledyne	RESON Underwater TC 4013	High sensitivity microphones with a usable frequency range of 1Hz to 170kHz	1500 USD	2022
Algorithms: Vision	Ultralytics	YOLO v11	Real-time video processing up to 30+ fps	-	-
Algorithms: Acoustics	-	Time difference of arrival	Energy Based Arrival and Position Estimation	-	-

Algorithms: Localization and Mapping	-	Custom Mapping algorithms	Fusing Algorithms like DBSCAN, Time decay clustering etc.	-	-
Algorithms: Autonomy	-	State Machine & Mission Planner	state machine for mission planner, designed in-house	-	-
Open Source Software	OpenCV, Eigen, ROS, YOLO v11	-	-	-	-

APPENDIX B: COMPONENT SPECIFICATIONS (MINI MATSYA)

Component	**Vendor**	**Model/Type**	**Specs**	**Cost** (new)	**Year Of Purchase**
Buoyancy Control	Designed In-House	Dead Weights & Foam	-	-	2026
Waterproof Housing	Designed In-House	Aluminium Hulls w/ Acrylic End-cap	Hull weighing 22 kg Depth Rating : 70 ft	300 USD	2026
Waterproof Connectors	Designed In-House	Aluminium	12 connectors weighing 0.75 kg in total	180 USD	2026
Thruster	Apisqueen	U2	1.3 and 1.2 kgf forward and backwards	240 USD	2026
Motor Control	Apisqueen	ESC	35A PWM controlled brushless motor speed controller	100 USD	2026
High Level Control	Raspberry Pi	Pico	ARM Cortex M based dual core	5 USD	2026
Battery	SkyCell	LiPo Battery	4 Cell and 16000mAh x 2	400 USD	2025
Converter	-	200W DC-DC Buck converter	Designed in-house DC-DC switching supply circuit which supports upto 20A.	50 USD	2026
CPU	NVIDIA	Jetson Orin Nano	6-core ARM Cortex-A78AE cores with 1024 CUDA cores and 32 tensor cores	250 USD	2025
Internal Comm Network	-	UART, USB	1 Mbps operation limit	5 USD	2025
External Comm Interface	-	Ethernet	10-100 Mb/s	-	2026
Inertial Measurement Unit (IMU)	Bosch	BNO085	Consists of accelerometer and 6-axis gyroscope	30USD	2025
Camera(s)	Luxonis	OAK-D S2, OAK-W	Stereo vision based cameras used for depth perception	330 USD	2025
Algorithms: Vision	YOLO v11	-	Parallel and Sequential processing, lens formula	-	-
Algorithms: Localization and Mapping	GTSAM/-	Factor Graphs / Modified ESKF	SLAM	-	-
Algorithms: Autonomy	-	State Machine & Mission Planner	Finit state machine for mission planner, designed in-house	-	
Open Source Software	OpenCV, Eigen, ROS, YOLO v11	-	-	-	-

VI. Appendix C: System Architectures & Testing Flows

A. System Architecture - Electrical

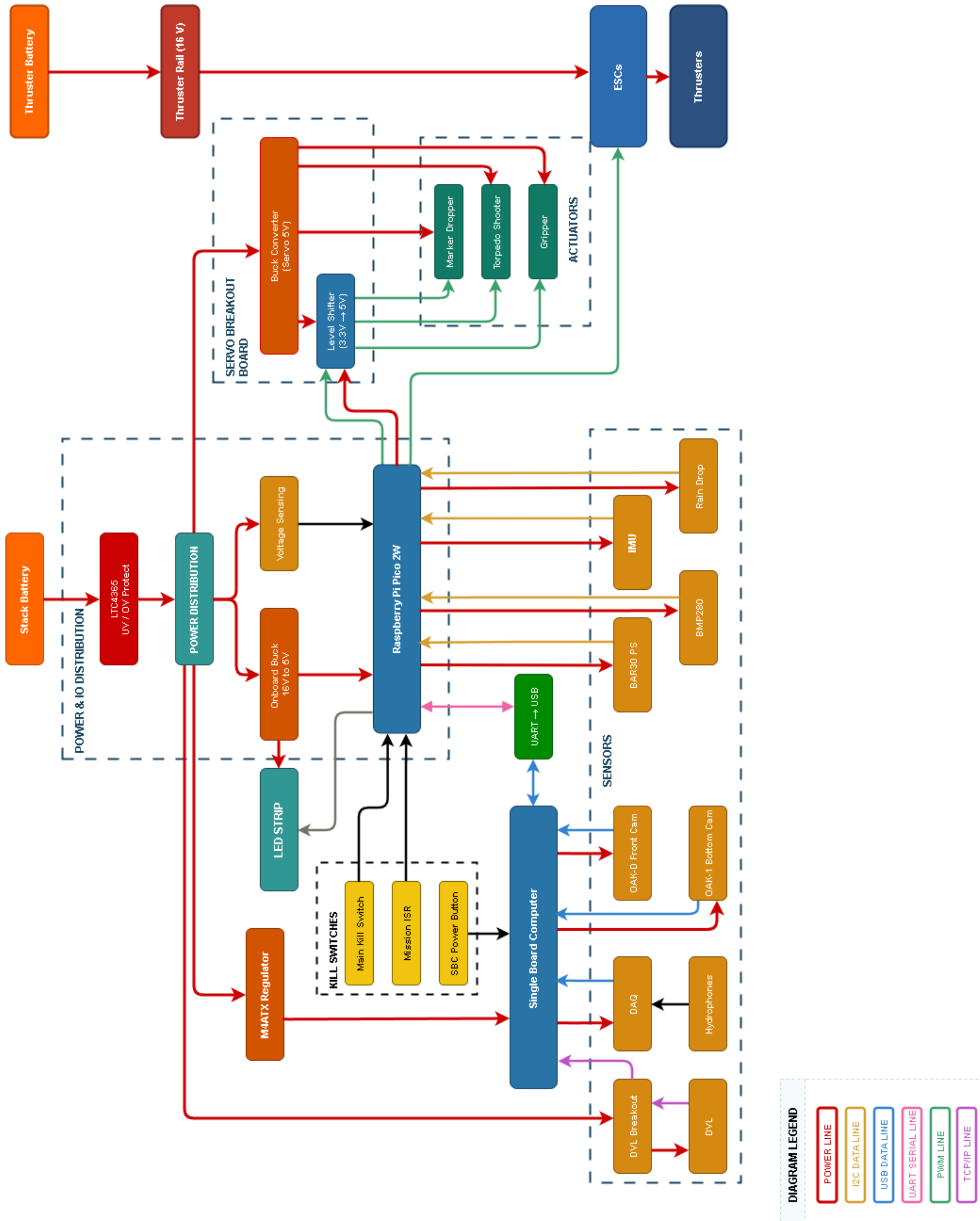


Figure 8: Matsya

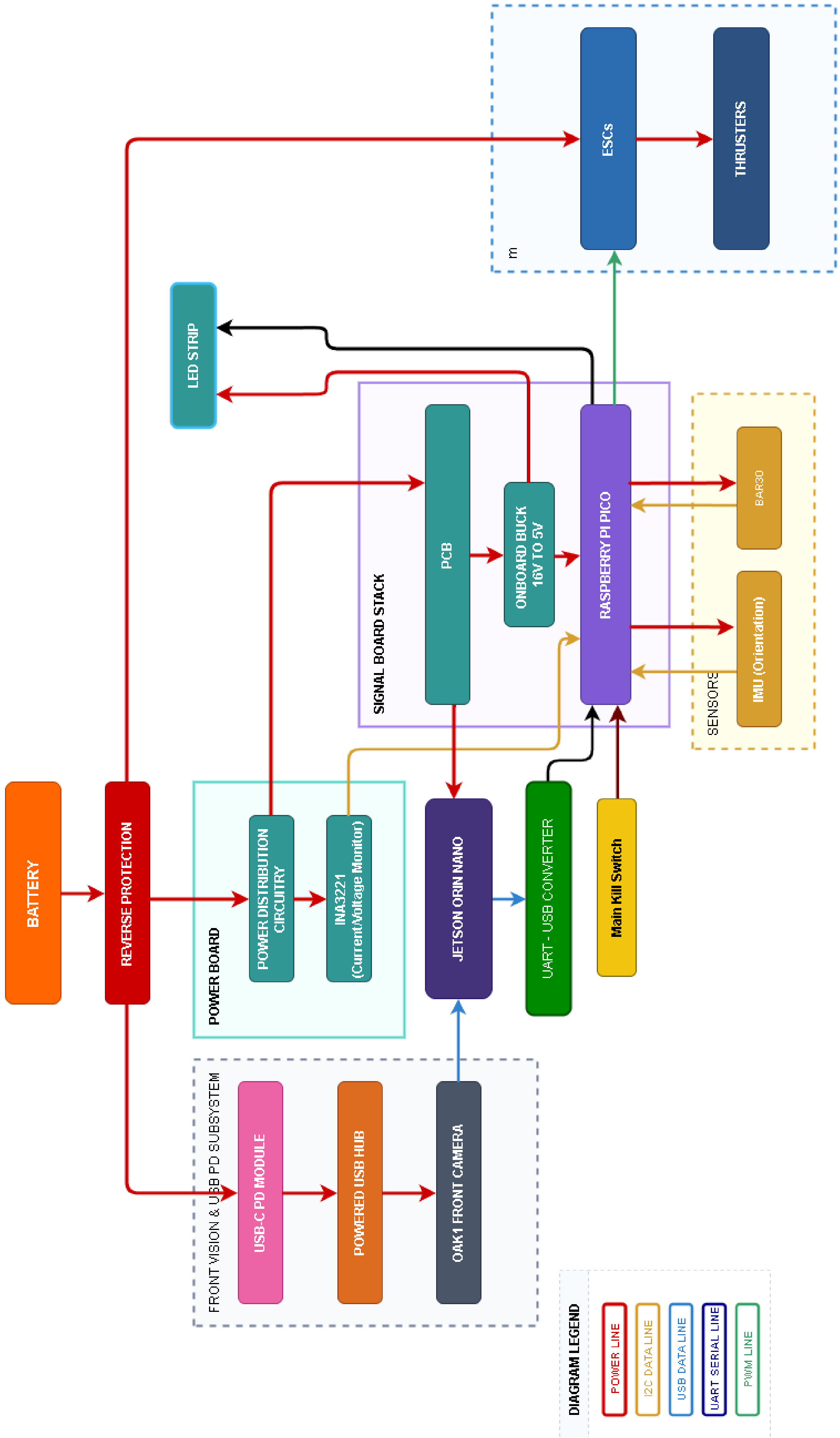


Figure 9: Mini Matsya

B. System Architecture - Software

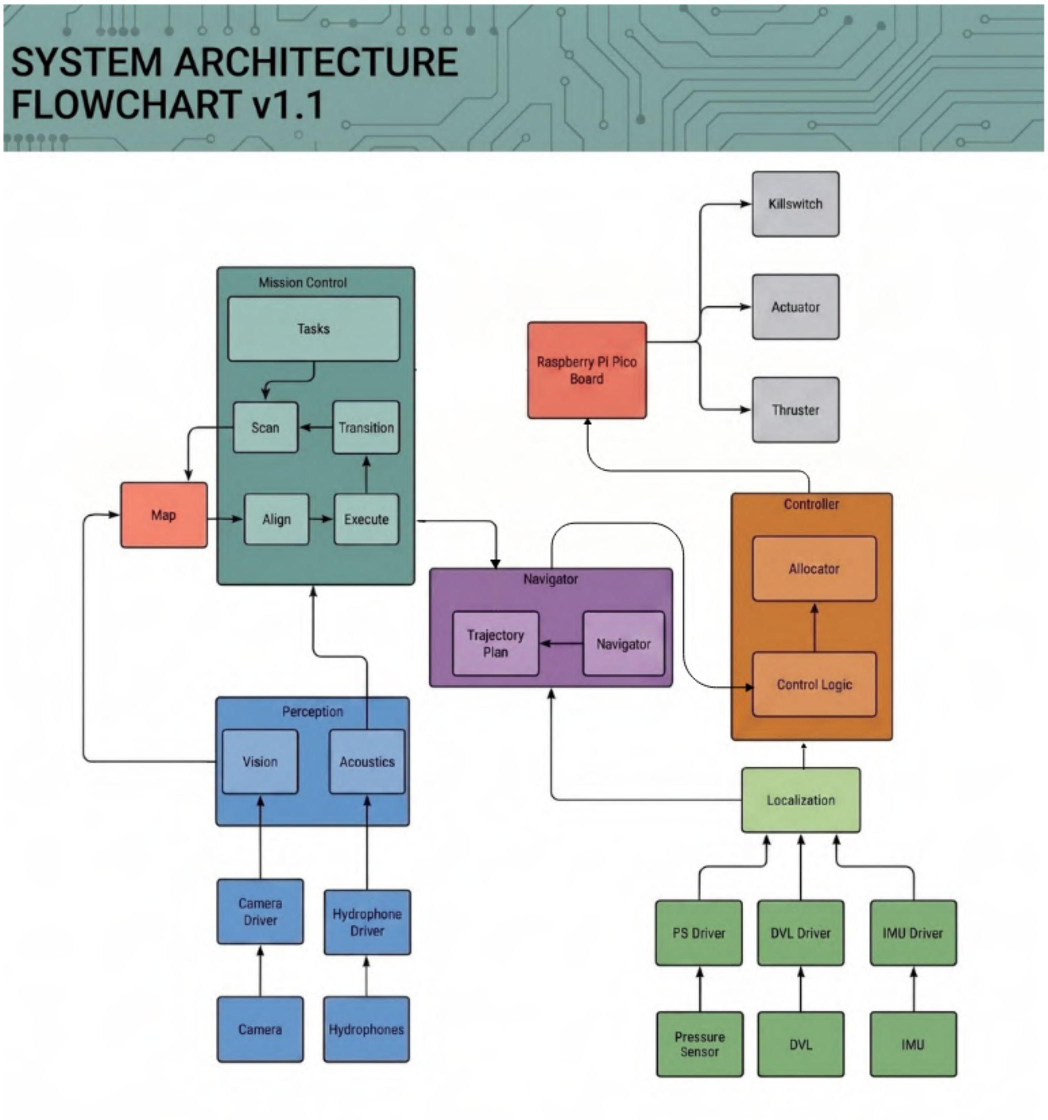


Figure 10: Matsya

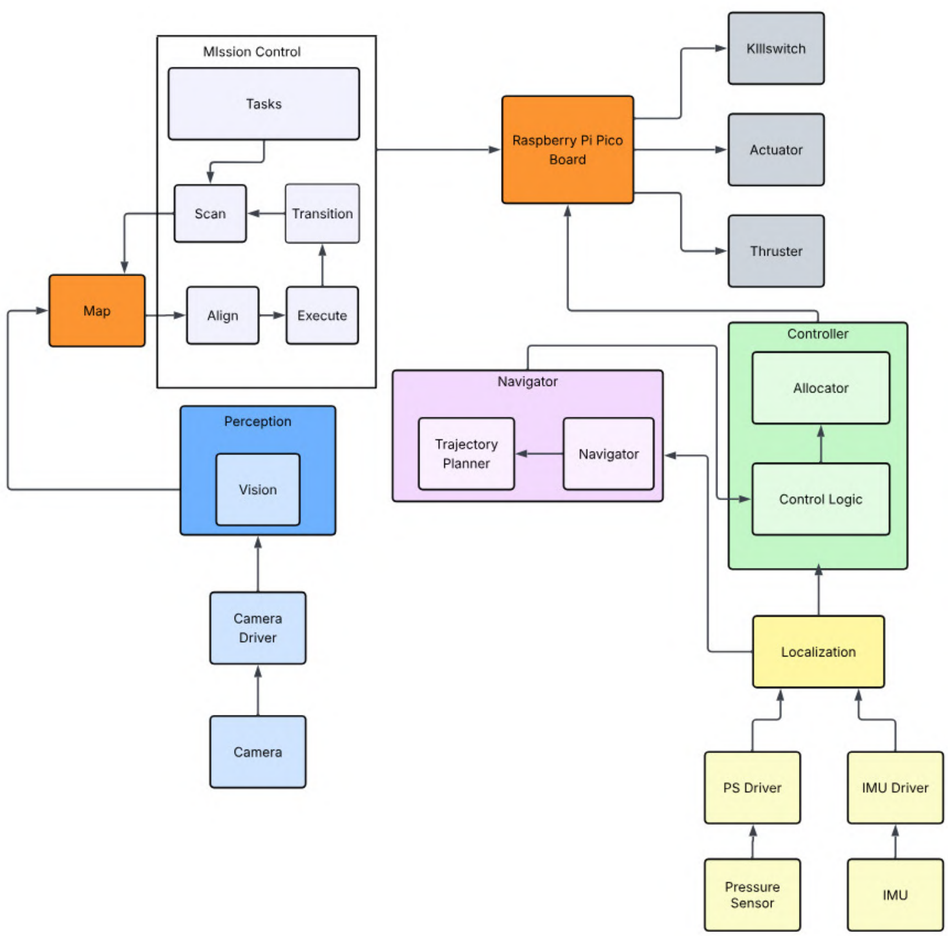


Figure 11: Mini Matsya

C. Testing Flow - Software

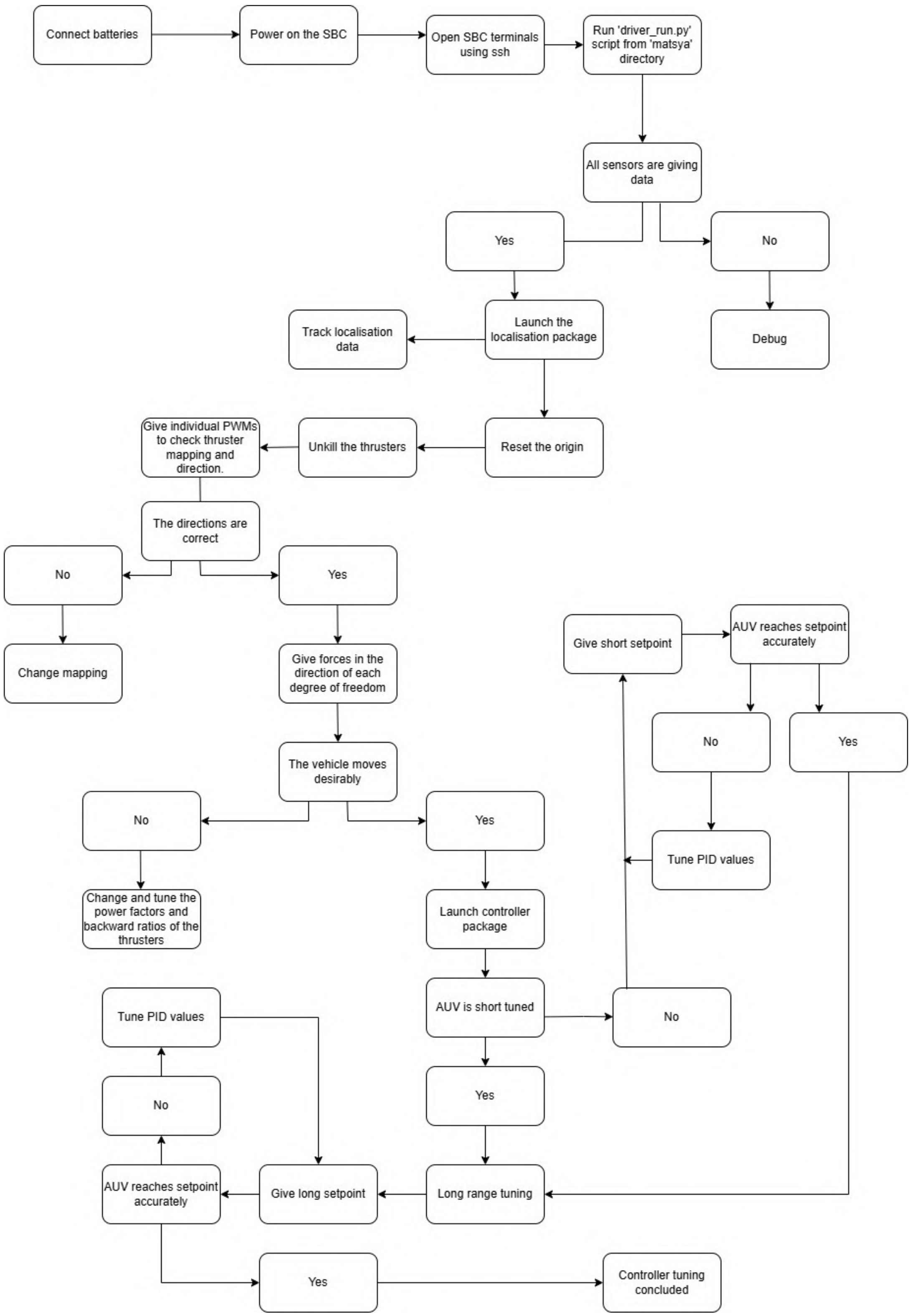


Figure 12: Controller & Allocator

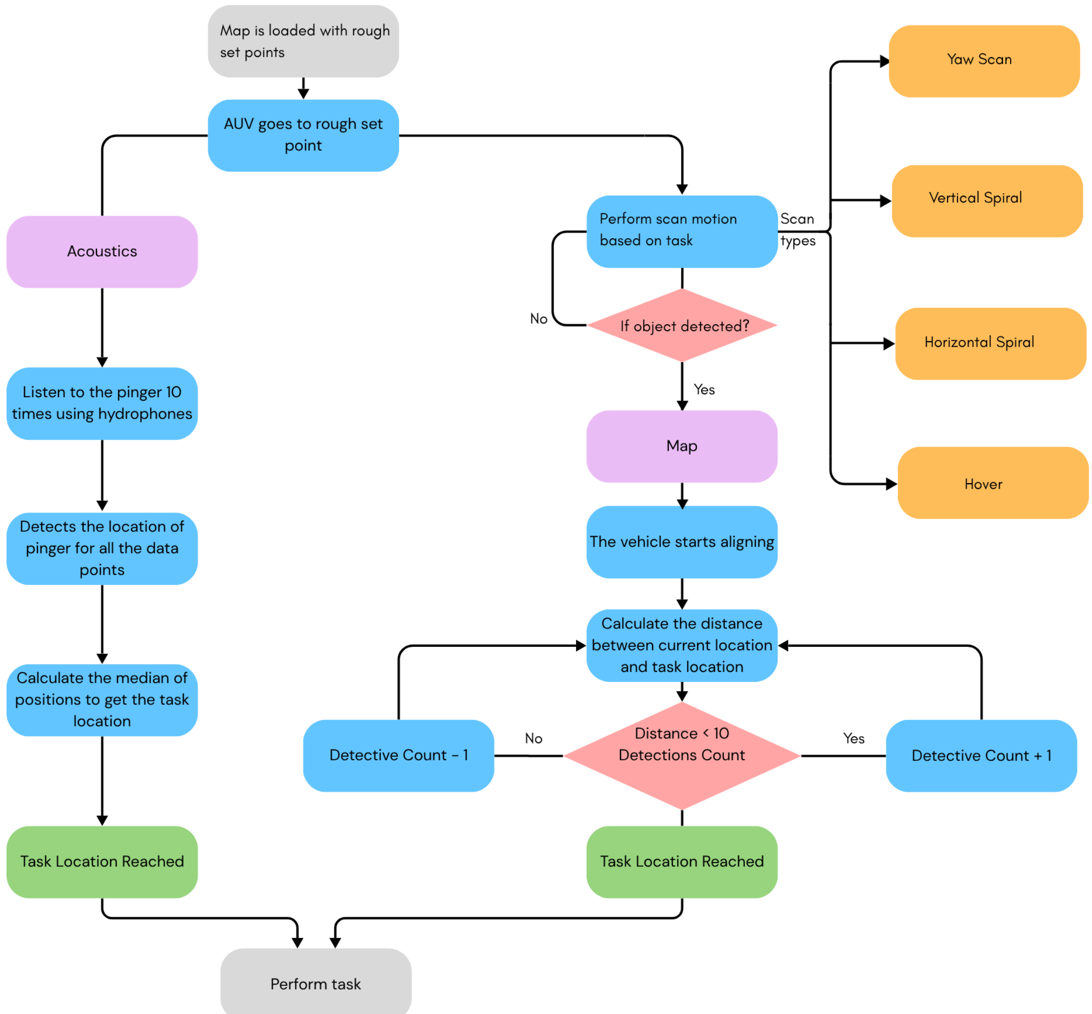


Figure 13: Mission Control

APPENDIX D: MATLAB MODEL TO DETERMINE WEIGHT DISTRIBUTION IN MINI MATSYA

Mini-Matsya has no active pitch correction — no dedicated pitch thrusters or dynamic ballast. Pitch stability in both upright and inverted orientations (required for the Barrel Roll) is instead achieved passively through the relative placement of the **Centre of Mass (CoM)**, **Centre of Buoyancy (CoB)**, and **Centre of Pressure (CoP)**. The key constraint is that the vertical separation Δz between COM and COB must be small and positive: too large a Δz produces a destabilising moment when inverted, as the COB drops below the COM. A MATLAB model was developed to sweep three structural variables — inter-plate gap, thruster arm length, and component placement — before hardware was finalised. The optimised configuration (50 mm inter-plate gap, battery shifted 100 mm rearward, thruster arms at 120 mm) reduced Δz from 51.2 mm to 35.4 mm and recovered a positive static margin.

Simulation Results

The preliminary layout failed two of the three passive stability criteria: Δz was too large (inverted pitch unrecoverable) and the static margin was negative (pitch-divergent under thrust). The optimised layout satisfies all three criteria simultaneously, as summarised in Table VIII.

Table VIII: Simulation Results: Previous vs. Optimised Layout

Configuration	**Δz (mm)**	**SM†**	**Peak Roll**	**Settling Time**
Previous layout	51.2	−0.048	11.2°	6.5 s
Optimised layout	35.4	+0.061	8.1°	4.2 s

†Static Margin (SM) is defined as the normalised distance between the Centre of Pressure and Centre of Mass along the vehicle's longitudinal axis. A positive SM indicates inherent pitch stability; a negative SM indicates pitch divergence.

Experimental Validation

The optimised layout was validated experimentally. The measured pitch angle under forward thrust (1.4° at 1.0 m/s) agrees closely with the model prediction and confirms a positive static margin. The vehicle demonstrated stable, self-aligning behaviour at cruise speed without active pitch compensation, as summarised in Table IX.

Table IX: Experimental Validation Against Model Predictions

Metric	**Model Prediction**	**Measured Result**
Static pitch trim error	1°	0.7°
Static roll bias	0.5°	0.3°
Roll settling time (10° step)	$\approx$ 4.2 s	4.5 s
Pitch under forward thrust @ 1.0 m/s	2°	1.4°
Pitch divergence rate when inverted	Bounded passively	No runaway divergence

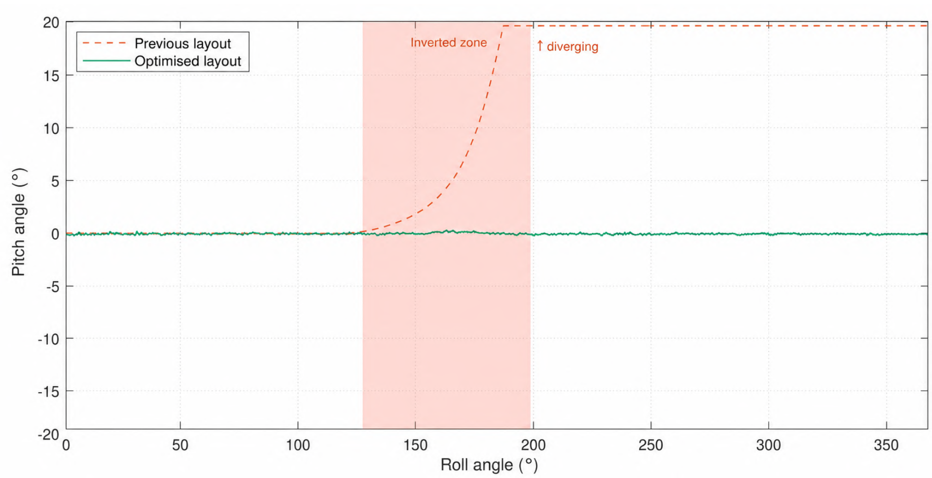


Figure 14: Inherent Pitch Correction after suggested improvements

Appendix E: Torpedo Development

The torpedo was designed for accurate launch over a 2 m underwater range. Rather than active guidance, a purely passive approach was adopted: stability is achieved entirely through geometry and mass distribution, with no electronics or control systems.

D. Hydrodynamic Design

The hull follows a **Myring profile**, minimising drag while maintaining smooth flow separation. Passive stability is enforced by two mechanisms: rear stabilising fins that generate restoring moments under pitch or yaw disturbances, and a **Centre of Pressure (CP) placed aft of the Centre of Gravity (CG)**, ensuring that any angular perturbation produces a corrective - not divergent - hydrodynamic moment.

E. Simulation and Validation

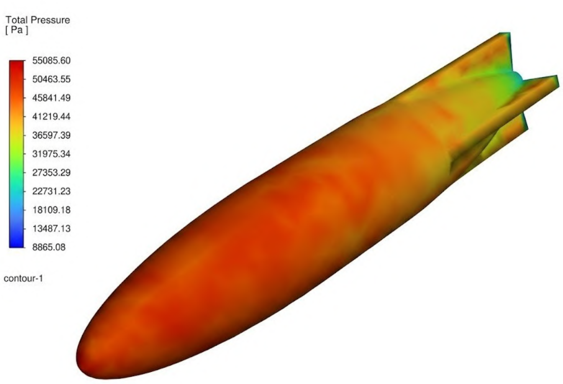
A **MATLAB** simulation based on the **Fossen equations of motion** was used to sweep key design parameters - nose shape index, tail semi-angle, fin span, and longitudinal CG position - and optimise for trajectory stability and damping. The finalised design showed smooth, predictable flight with minimal pitch deviation over the operating range (Figure 15a). Drag characteristics were subsequently confirmed via **CFD** in SimScale (OpenFOAM, k–ω SST).

F. Buoyancy and Manufacturing

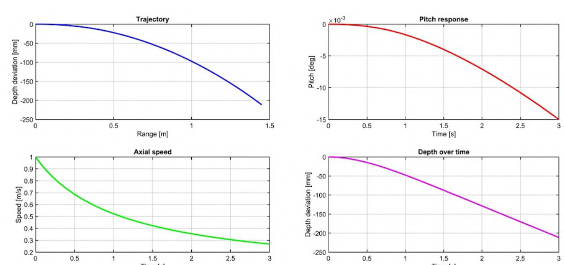
The torpedo was made **slightly negatively buoyant** for a controlled descent toward the target. Displaced volume was calculated analytically from the Myring geometry; two coats of **XTC-3D epoxy resin** were applied before ballast tuning so that the waterproofed mass was accounted for. The final design was manufactured entirely by **3D printing**.

G. Results

As shown in Figure 15b, the simulation confirms a smooth downward trajectory, well-damped pitch response with no oscillation, and predictable speed decay - all within acceptable bounds over the 2 m range. The torpedo achieved the required targeting accuracy using only passive hydrodynamic stabilisation.



(a) CFD contour plot of total pressure distribution over the torpedo surface.



(b) MATLAB plots of different trajectory parameters.

APPENDIX F: SOFTWARE SUBSYSTEM - TECHNICAL DETAILS

F.1 Long-Short Range Controller

The PID gain set required for stable, long-distance transit is incompatible with the gain set required for precise short-range manoeuvres during actuation. Far-field transit (>100 cm from target) requires aggressive proportional response and minimal integral action to prevent wind-up. Near-field approach (<100 cm) requires high derivative damping and tight integral limits for sub-decimetre alignment without limit-cycle behaviour. A distance-weighted blend of two canonical gain sets is therefore applied at every control cycle, eliminating the step discontinuity that hard switching at 100 cm produced in earlier iterations. Let d denote the instantaneous distance-to-target along the active translational axis, $K_{\text{short}}, K_{\text{long}} \in \mathbb{R}^3$ the canonical short- and long-range gain triplets (P, I, D), and $K(d)$ the active gain triplet emitted by the scheduler. The transition band $[d_{\text{lo}}, d_{\text{hi}}] = [85, 115]$ cm spans ± 15 cm around the nominal 100 cm boundary.

Distance-Weighted Gain Schedule

Three regimes are defined on d (Table X). Outside the transition band the canonical sets are used unchanged; inside, the gain on each channel is linearly interpolated.

Distance to target d	**Gain selection**
$d < 85$ cm	K_{short}
$85 \leq d \leq 115$ cm	linear blend $K(d)$
$d > 115$ cm	K_{long}

Table X: Distance regimes and corresponding gain selection.

Blending Law

Within the transition band the active gain is

$$K(d) \;=\; (1-\alpha)\, K_{\text{short}} \;+\; \alpha\, K_{\text{long}}, \qquad \alpha \;=\; \frac{d-85}{30}, \quad \alpha \in [0,1]. \tag{2}$$

The interpolation is applied element-wise to the P, I, and D channels on the surge, sway, and heave axes. The rotational channels (yaw, pitch, roll) retain a single fixed gain set across both regimes; this decoupling preserves attitude stability across phases, reduces the validation surface, and accelerates experimentation.

Settling time and first-overshoot peak were measured per translational axis against the prior fixed-gain controller (Table XI). Settling time falls by 51–56% across all three axes, and overshoot is reduced on every axis. Disturbance rejection improves automatically: any perturbation away from the target re-engages the aggressive long-range gains without operator intervention.

Axis	**Settling (s)**		**Overshoot (cm)**		Δt_s
	Static	Adaptive	Static	Adaptive	
Surge	28.5	13.6	44.7	19.2	−52%
Sway	30.4	13.4	40.5	33.7	−56%
Heave	32.4	15.8	36.7	19.1	−51%

Table XI: Per-axis step response: prior fixed-gain controller vs. distance-weighted adaptive controller.

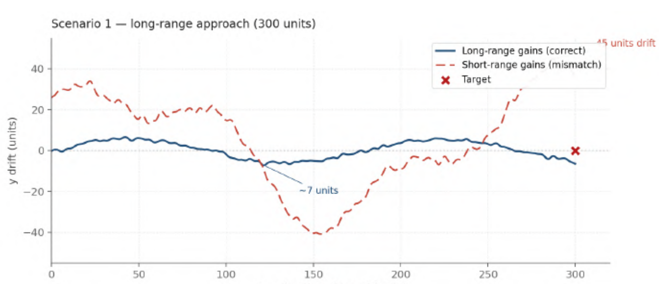


Figure 16: Long-range approach.

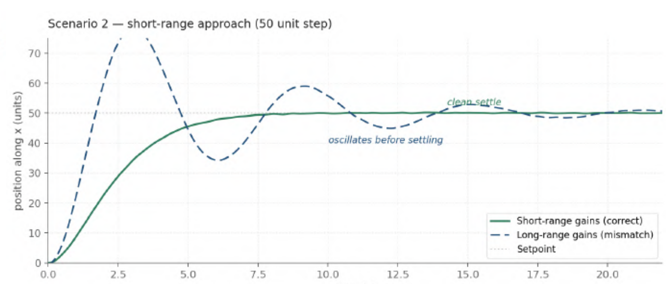


Figure 17: Short-range approach.

F.2 On-Camera Inference Pipeline (OAK-D Model Offloading)

A single unified camera node produces two synchronized 640×640 BGR taps from the primary camera via a single request output call. One tap feeds the publishing branch, paired with an image alignment-reprojected depth frame. The other tap feeds the spatial detection network, which runs YOLOv11n on the Myriad X VPU and emits detections with 3D coordinates in millimetres relative to the RGB optical centre. Both branches share the same stereo depth source. Pairing is performed on-device by hardware sequence number, bounding the residual RGB-to-depth offset to a deterministic ∼58 ms (the processing-latency difference between the RGB ISP path and the stereo-depth path), versus the unbounded and irregular offsets seen with the prior host-polled scheme.

Feature	DepthAI v2	DepthAI v3
Camera structure	Separate color and mono camera streams	Unified camera node with shared outputs
Depth alignment	Native stereo alignment only	Dedicated, reusable image projection
Frame pairing	Host-side queue polling	On-device hardware-level syncing
Model packaging	Standard blob format	Compressed architecture archives
IMU integration	Not supported in the core driver	Standardized IMU message delivery at 100 Hz

Table XII: Driver-level deltas, v2 baseline vs. current v3 implementation.

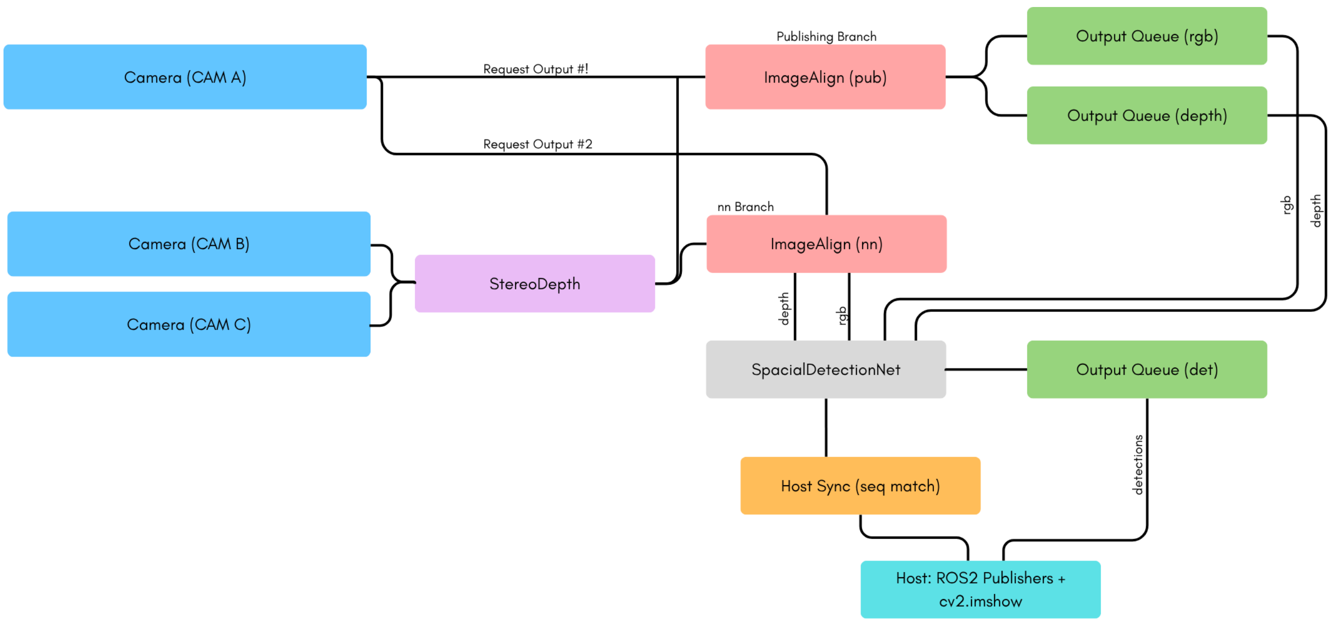


Figure 18: Current v3 on-camera inference pipeline.

Depth-to-RGB Alignment

The stereo pair and the RGB camera carry different positions, intrinsics, distortions, and fields of view, so raw depth referenced to the stereo baseline midpoint does not pixel-align with the RGB image. The standalone image alignment node, new in v3, performs per-pixel reprojection into the RGB frame using factory-stored calibration, producing a depth image in which every RGB pixel carries its corresponding depth in millimetres. The aligned depth is clipped to $[0, 32767]$ and cast to 16-bit integers, retaining millimetre precision out to ∼32.7 m. The same aligned depth feeds both the spatial detection network and the published depth topic.

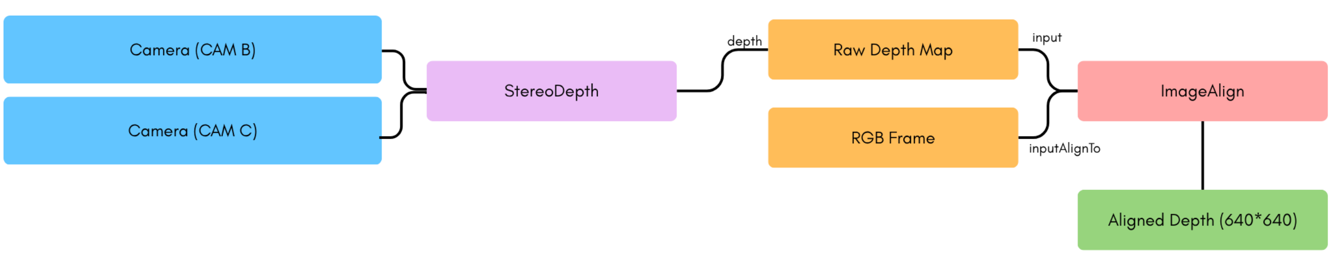


Figure 19: Depth-to-RGB alignment via the standalone image alignment block.

F.3 RGB-D Multi-Channel Detection (YOLO-RGBD)

YOLO-RGBD extends the standard three-channel YOLO detector with a fourth, co-registered depth channel so the network can exploit illumination-invariant geometric structure alongside RGB appearance. The fusion-strategy framework follows the multispectral methodology of YOLOv11-RGBT [15], adapted from RGB-Thermal to RGB-Depth.

Architecture

The input tensor is widened from three to four channels, with the fourth channel reserved for depth. The change propagates through the first convolutional layer, the data loader, the pretrained-weight initialiser, and the augmentation pipeline. A *channel-splitter* module is inserted at the entry point of every model: the loader emits a single four-channel tensor (preserving the native batching, augmentation, and resize routines), and the splitter separates RGB and depth immediately on entry into the backbone. Every fusion variant therefore shares the same data pipeline.

Fusion strategies

Let $I_{\text{RGB}} \in \mathbb{R}^{3\times H\times W}$ and $I_{\text{D}} \in \mathbb{R}^{1\times H\times W}$ denote the co-registered RGB and depth inputs, and $\phi_{\text{RGB}}, \phi_{\text{D}}$ denote per-modality feature extractors. The six implemented strategies, ordered by the network depth at which fusion occurs, are summarised in Table XIII. As representative cases, early and mid fusion can be written

$$F_{\text{early}} = \phi(\,[I_{\text{RGB}}\,;\,I_{\text{D}}]\,), \qquad F_{\text{mid}} = [\,\phi_{\text{RGB}}(I_{\text{RGB}})\,;\,\phi_{\text{D}}(I_{\text{D}})\,], \tag{3}$$

where $[\,\cdot\,;\,\cdot\,]$ denotes channel-wise concatenation.

Strategy	Fusion stage	Backbone cost
Early	Input concatenation, shared backbone	1×
Mid	Two parallel backbones, mid-stage concat	2×
Mid-P3	Depth injected at P3 of the feature pyramid	2×
Late	Separate streams to the detection head	2× full network
Score	Combination at the detection-score level	2× full network
Share	Intermediate cross-modal parameter sharing	∼1.5×

Table XIII: Implemented fusion strategies, ordered by the network depth at which RGB and depth are combined.

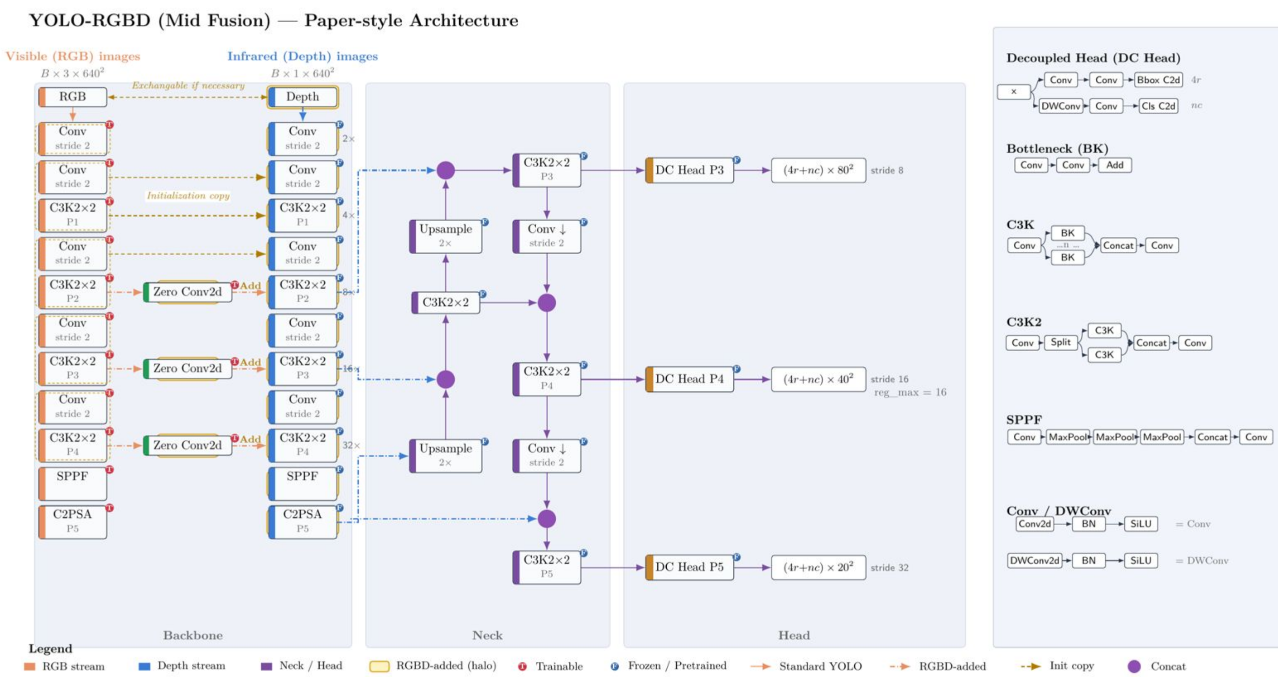


Figure 20: Mid-fusion architecture: two parallel backbones with channel-wise concatenation at an intermediate stage, followed by the shared neck and detection head.

Training pipeline

A three-stage transfer-learning pipeline ports COCO-pretrained weights onto the four-channel model. Each layer is enumerated; compatible parameters are copied directly, and the first convolution (the only structurally incompatible layer) is reshaped to accept the fourth input channel rather than initialised randomly. This preserves the appearance prior learned on COCO and permits effective training on modest RGB-D datasets. A dataset converter normalises external RGB-D and multispectral benchmarks, where depth is typically stored as 16-bit millimetre values, into the 8-bit grayscale format consumed by the loader, and derives bounding-box annotations from the accompanying segmentation masks.

Dataset collection

The training dataset was collected in-house using the Luxonis OAK-D mounted on Matsya 7, with the vehicle teleoperated through the pool to capture the Slalom prop. RGB and depth frames were recorded together through the synchronised vision pipeline. RGB frames were annotated for bounding boxes on the Roboflow platform; after annotation export, each annotated RGB image was re-paired with its corresponding depth frame and written out as a four-channel training tensor. This collection-then-overlay pipeline avoids the need to annotate depth maps separately, and guarantees that every (image, depth, label) triple in the training set carries the same temporal alignment that the deployed vision stack provides at inference time.

Evaluation protocol

Four configurations were trained for 25 epochs on the collected RGB-D set: RGB baseline, Early fusion, Mid fusion, and Mid-P3 fusion. Inputs are 640×640, batched at 64, with the standard YOLO augmentation set operating uniformly across both modalities through the unified loader. Both training and validation losses are tracked as the sum of the box, classification, and distribution-focal-loss (DFL) components. Ranking is reported on mAP@50–95; mAP@50 is reported for completeness.

Headline results

Mid fusion produces the strongest detector on the evaluation set: mAP@50–95 of **0.9658** against the RGB baseline's 0.9388, a +2.7-point absolute improvement, at an inference cost of 2.7 ms per image versus 1.3 ms for the baseline. mAP@50 saturates at 0.995 across all four configurations and is therefore uninformative for ranking; the discriminating metric is mAP@50–95 (Table XIV).

Metric	RGB	Early	Mid	Mid-P3
mAP@50	0.9950	0.9950	0.9950	0.9950
mAP@50–95	0.9388	0.9592	**0.9658**	0.9649
Precision	0.998	1.000	1.000	1.000
Recall	0.999	1.000	1.000	1.000
Val. loss	1.43	1.30	1.27	1.27
Latency (ms)	1.3	2.3	2.7	2.7

Table XIV: Headline metrics at epoch 25.

Loss decomposition

The validation-loss ordering (RGB 1.43, Early 1.30, Mid/Mid-P3 1.27) is preserved across each component-box, classification, and DFL-indicating that the depth modality contributes to localisation quality and not solely to classification. All three fusion variants overtake the RGB baseline by epoch 10 and continue to widen the gap through the remainder of training.

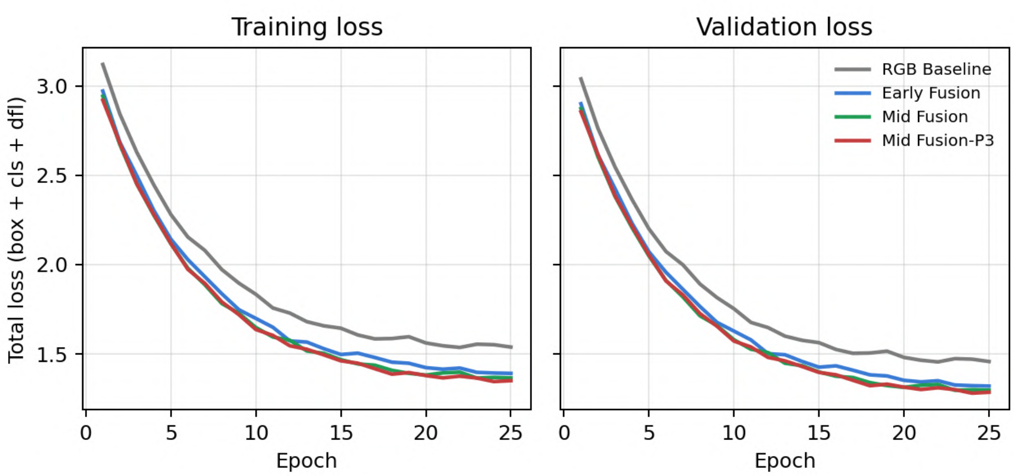


Figure 21: Training and validation loss curves: RGB baseline, Early, Mid, and Mid-P3 fusion.

Accuracy–latency Pareto

Per-stage latency decomposes into preprocess, inference, and postprocess. Inference dominates in all configurations, but postprocess is noticeably heavier for mid fusion because the dual-backbone features must be concatenated before non-maximum suppression. On the accuracy–latency plane, mid fusion sits on the Pareto frontier; Mid-P3 buys very little over mid fusion for its additional cost; the RGB baseline is fastest but materially less accurate.

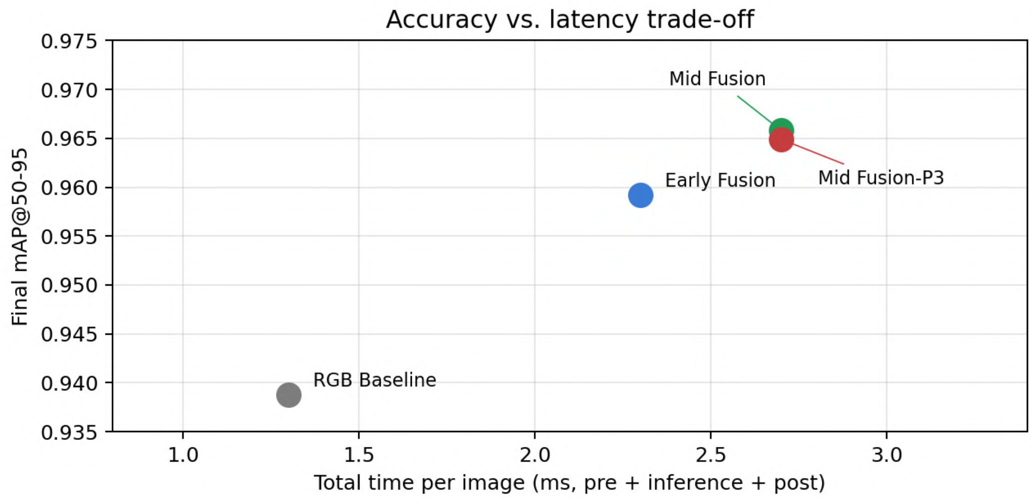


Figure 22: Accuracy–latency Pareto plot. Mid fusion dominates Mid-P3 and Early fusion on the accuracy-per-millisecond frontier.

F.4 Trajectory Planner: Jerk-Limited Setpoint Generation

The trajectory planner sits between the mission planner and the PID loops (Appendix F.1) and converts each commanded waypoint into a jerk-bounded stream of intermediate setpoints. Previously, applying step-change PWM commands caused asynchronous thruster spin-up, resulting in severe initial yaw instability before the vehicle could finally stabilise. The new trajectory planner eliminates these abrupt transients. By ensuring all thrusters spool up and wind down synchronously, even if minor yaw deviations still occur, they are now strictly limited and highly controlled.

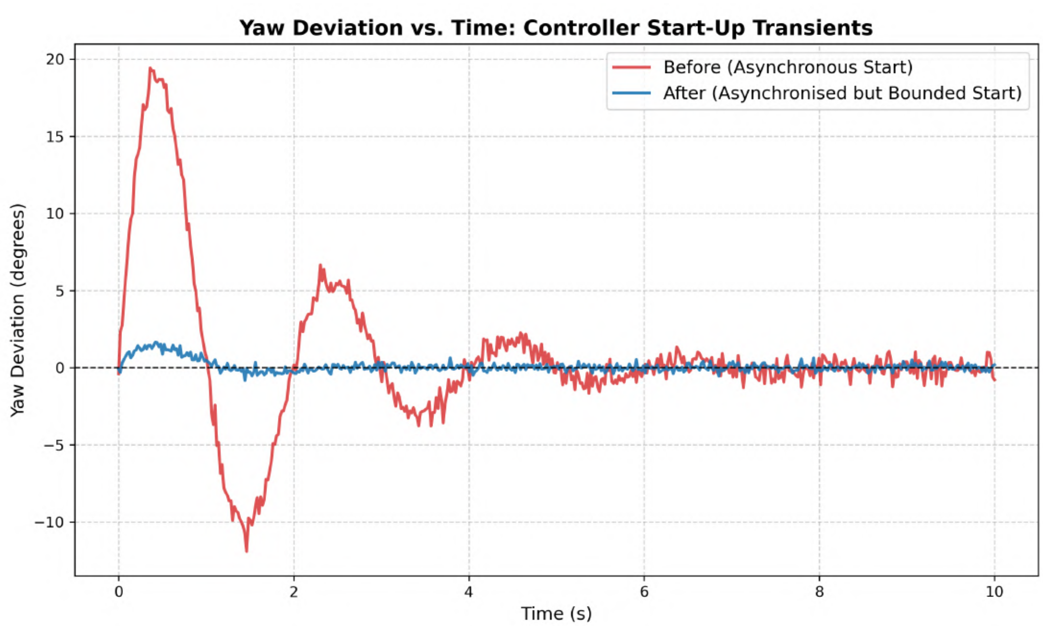


Figure 23: Yaw deviation vs. time during controller start-up.

F.4.1 Velocity profile

The continuous trajectory between two waypoints is parameterised by a normalised progress variable $x \in [0, 1]$ and a velocity profile

$$v(x) \propto \big(x(1-x)\big)^3. \tag{4}$$

This 6^{th}-order polynomial is theoretically constructed to ensure that velocity, acceleration, and jerk drop exactly to zero at both endpoints and all four quantities (S, V, A, J) are bounded and smooth across the full transit. The boundary conditions guarantee a perfectly smooth start and finish, eliminating abrupt thruster transients. Between the bounds, the velocity constantly increases to a single maximum at $x = 0.5$ before symmetrically decelerating. This profile ensures the vehicle reaches the final setpoint quickly by maintaining a high transit speed in the middle of the manoeuvre, rather than remaining excessively slow and sluggish throughout the entire path.

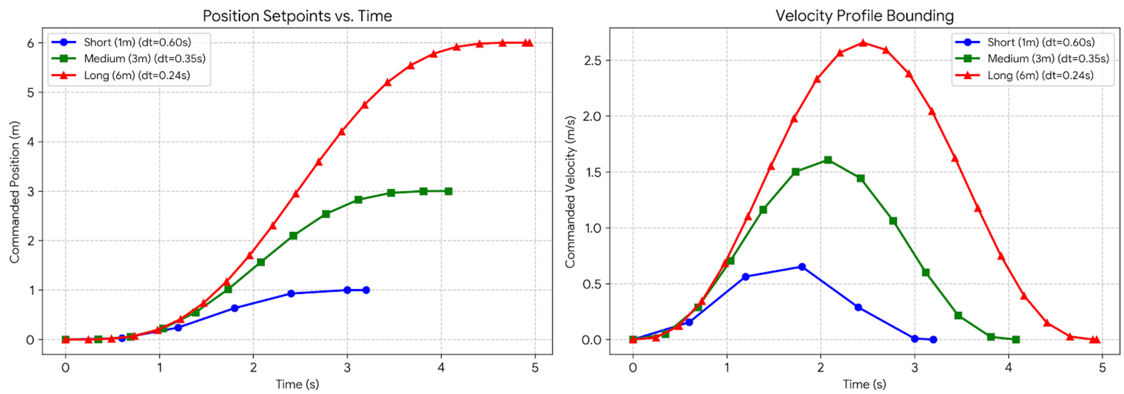


Figure 24: Normalised kinematic profile for Setpoints and Velocity derived from Eq. 4.

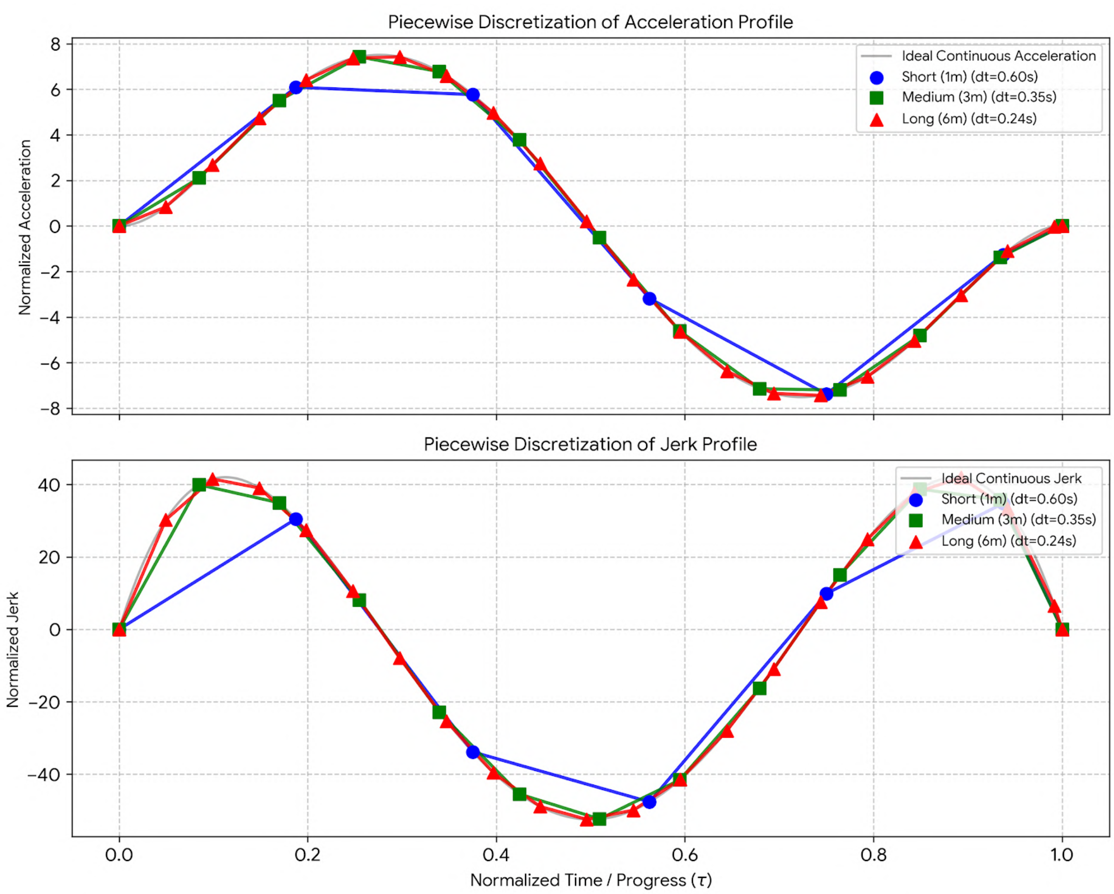


Figure 25: Normalised kinematic profile for Acceleration and Jerk derived from Eq. 4.

F.4.2 Implementation Strategy: Setpoints vs. Direct Forces

During development, two competing methodologies for applying the trajectory profile were evaluated:

1) **Direct Force Generation:** Differentiating the velocity profile to derive required accelerations, multiplying by the vehicle's mass matrix, and sending these global forces directly to the thruster allocator in an open-loop feedforward manner.
2) **Setpoint Tracking (Selected):** Integrating the velocity profile to derive positional setpoints (as detailed in

Appendix F.4.3) and feeding these into the closed-loop PID controller.

Empirical testing demonstrated that the Setpoint Tracking method is significantly superior in subsea environments. Direct Force Generation relies on highly accurate hydrodynamic models; however, unmodelled drag, added mass variations, and tether drag cause significant steady-state tracking errors and spatial drift. Furthermore, open-loop force control inherently suffers from state de-synchronisation and phase lag: because the pre-calculated force command at any time t assumes the vehicle has successfully reached the previous predicted state, any physical delay caused by inertia or drag means the applied forces execute out of sync with the vehicle's true position, rapidly exacerbating the error. By contrast, generating intermediate spatial setpoints leverages the existing closed-loop PID architecture to automatically compensate for these environmental disturbances, maintaining tight adherence to the ideal trajectory without lag.

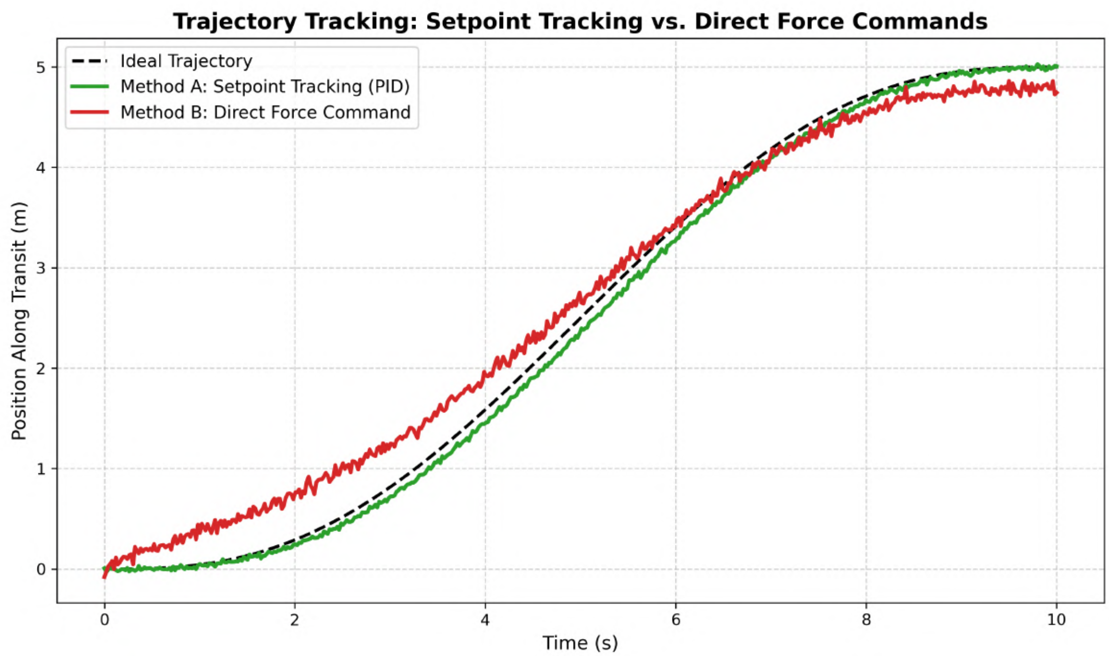


Figure 26: Trajectory tracking performance comparison.

F.4.3 *Dynamic interval scaling*

To generate the spatial commands, the continuous velocity profile is first integrated to produce a continuous setpoint (position) profile. We then use a dynamic profiling interval selection—dependent entirely on the total commanded distance $\|\Delta p\|$—to discretise this continuous profile into the final setpoints that are sent directly to the controller. The discrete sample interval dt is scaled dynamically depending on this total distance:

- **Short transits.** The interval dt remains large enough to keep the trajectory highly responsive while also keeping thrusters free of jerk.
- **Long transits.** The interval dt is dynamically adjusted to become short enough so that the peak commanded velocity $v_{\max}$ stays strictly below the thruster saturation limit.

This dynamic discretisation inherently bounds both peak commanded velocity and acceleration. Because $v_{\max}$ and $a_{\max}$ scale monotonically with $\|\Delta p\|$, the planner consistently produces hydrodynamically achievable trajectories ranging from sub-metre alignment corrections through to multi-metre transits.

F.4.4 *Integration with the controller*

At each control cycle, the planner publishes the current intermediate setpoint to the PID loop in place of the raw waypoint. Crucially, the controller calculates its error and selects its gain regime (Appendix F.1) based entirely on this intermediate setpoint, not the final waypoint. This is the fundamental purpose of the trajectory planner: by continuously presenting a slowly-advancing local target, the vehicle inherently operates within a stable tracking regime. This architecture entirely bypasses the discontinuous long-range to short-range gain boundary, seamlessly producing a smooth thruster command throughout the entirety of the manoeuvre.

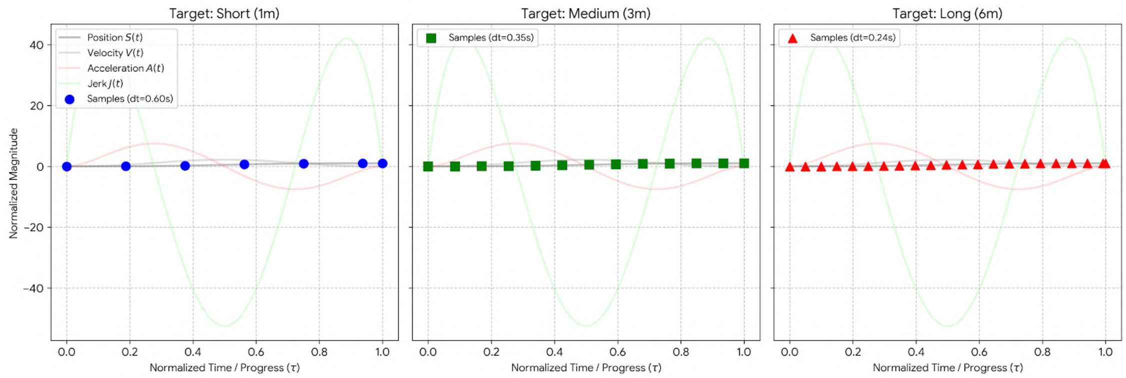


Figure 27: Dynamic setpoint generation across three transit distances demonstrating the discrete sample interval dt scaling with setpoint magnitude.

F.5 Adaptive Map Correction via Spatial Filter Fusion

Visual detection in scattered underwater environments produces false detections from refraction, particulate backscatter, and dynamic illumination. Feeding raw bounding-box detections directly into the state estimator introduces issues like the vehicle drifting to the wrong location and then going out of frame of the object. For instance, a raw, anomalous bounding-box detection could cause the AUV to deviate significantly off-course and fail its task. To resolve this, a history-aware spatial filter block is interposed between the detector and the estimator. This block stores all detections in a rolling buffer, which are then processed in parallel by three spatial algorithms. Their outputs are fused with task-dependent weights and passed to the navigation filter as a single high-fidelity observation.

Pipeline Architecture

The detector publishes 3D landmark positions in the inertial frame. These are accumulated into a temporal buffer and consumed in parallel by three estimators (STWE, Histogram, DBSCAN). A task-aware fuser combines the three candidate centres into a single observation handed off to the state estimator. Every fusion variant shares the same buffer and the same downstream interface; only the per-task weight vector changes.

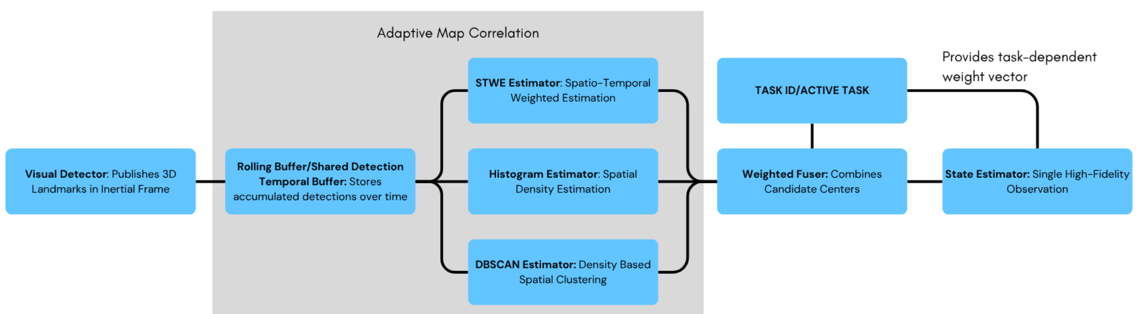


Figure 28: Adaptive map-correction block.

Notation

Let $\mathbf{p}_i \in \mathbb{R}^3$ be the inertial-frame detection, $c_i \in [0,1]$ be the detector confidence, and τ_i be the timestamp. Let $\hat{\mathbf{p}}_S, \hat{\mathbf{p}}_H, \hat{\mathbf{p}}_D$ denote the candidate centres returned by the STWE, Histogram, and DBSCAN algorithms respectively, and $\boldsymbol{\alpha} = (\alpha_S, \alpha_H, \alpha_D)$ with $\sum \alpha = 1$ the per-task fusion weights.

Algorithm 1 - Spatio-Temporal Weighted Estimation (STWE)

To identify the true target location, STWE evaluates the local density around every point in the buffer. For a given candidate centre $\mathbf{p}_i$, the weight of a neighbouring point $\mathbf{p}_j$ within a search radius R is calculated using temporal, confidence, and spatial components:

$$w_{i,j} = \underbrace{e^{-\lambda_t(t-\tau_j)}}_{w_{\text{temporal}}} \cdot \underbrace{c_j}_{w_{\text{confidence}}} \cdot \underbrace{e^{-\lambda_r\|\mathbf{p}_j-\mathbf{p}_i\|}}_{w_{\text{spatial}}} \tag{5}$$

The algorithm computes the total weight sum $W_i = \sum_{j:\|\mathbf{p}_j-\mathbf{p}_i\|\le R} w_{i,j}$ for every detection. The cluster with the maximum total sum is selected, designating its origin as the optimal centre $\mathbf{p}^\star$. The final estimator is the weighted mean of all points within this maximal cluster:

$$\hat{\mathbf{p}}_S = \frac{\sum_{j:\|\mathbf{p}_j-\mathbf{p}^\star\|\le R} w_{\star,j}\,\mathbf{p}_j}{\sum_{j:\|\mathbf{p}_j-\mathbf{p}^\star\|\le R} w_{\star,j}} \tag{6}$$

The thresholds $(R, \lambda_t, \lambda_r)$ are tunable, biasing the algorithm toward strict precision or broader accuracy as required by the operational context.

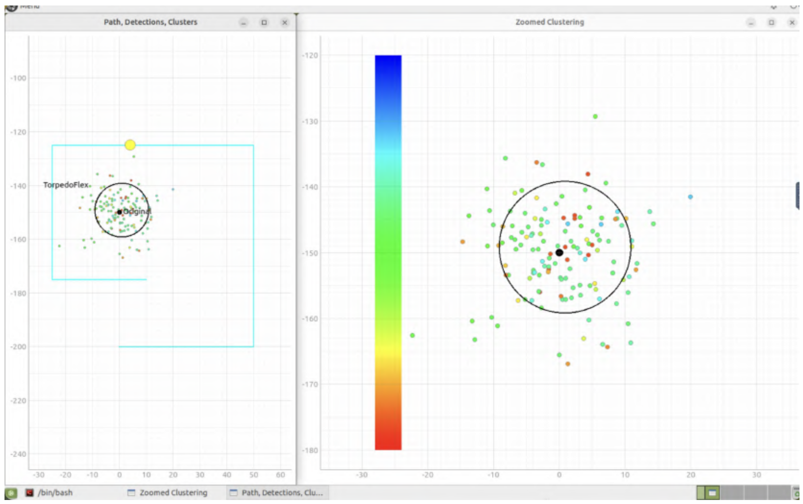


Figure 29: STWE diagnostic. Vehicle trajectory depicted by the line and raw visual scatter detections detected by the points.

Algorithm 2 - Histogram Filtering

The local coordinate space is partitioned into a fixed 2D grid of cell size Δ. Each cell c accumulates a vote $h(c) = \sum_{i \in c} c_i$; the estimator is the centroid of the maximum-vote cell:

$$\hat{\mathbf{p}}_H = \text{centroid}\Big(\arg\max_c h(c)\Big) \tag{7}$$

This grid-based approach natively filters gross outliers and ensures the vehicle is reliably guided to the correct macro-region, prioritising general-vicinity stability over pinpoint precision.

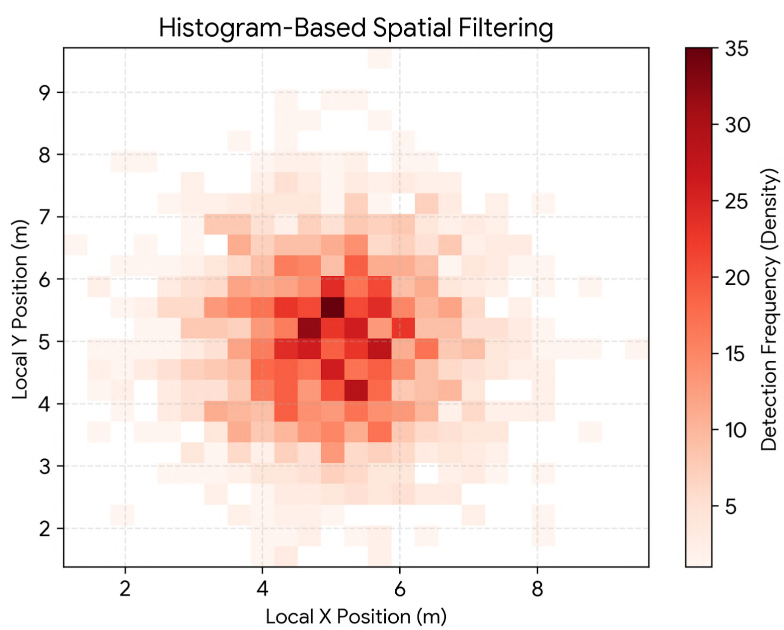


Figure 30: Histogram-based filtering.

Algorithm 3 - DBSCAN Clustering

Density-Based Spatial Clustering is run with parameters $(\epsilon, \text{minPts})$. The ϵ parameter defines the maximum spatial search radius around a detection, while minPts defines the minimum number of consistent detections required within that radius to form a dense region. By applying these thresholds, sparse and transient false-positive detections are explicitly labelled as noise and discarded. The estimator is the mean of the largest core cluster:

$$\hat{\mathbf{p}}_D = \mathrm{mean}(\mathcal{C}_{\max}), \qquad \mathcal{C}_{\max} = \arg \max_{\mathcal{C} \in \mathrm{DBSCAN}(\mathcal{B})} |\mathcal{C}| \tag{8}$$

DBSCAN yields the highest spatial resolution of the three filters, providing the exacting precision required for strict tasks, but requires a dense history of accurate detections to form a valid cluster.

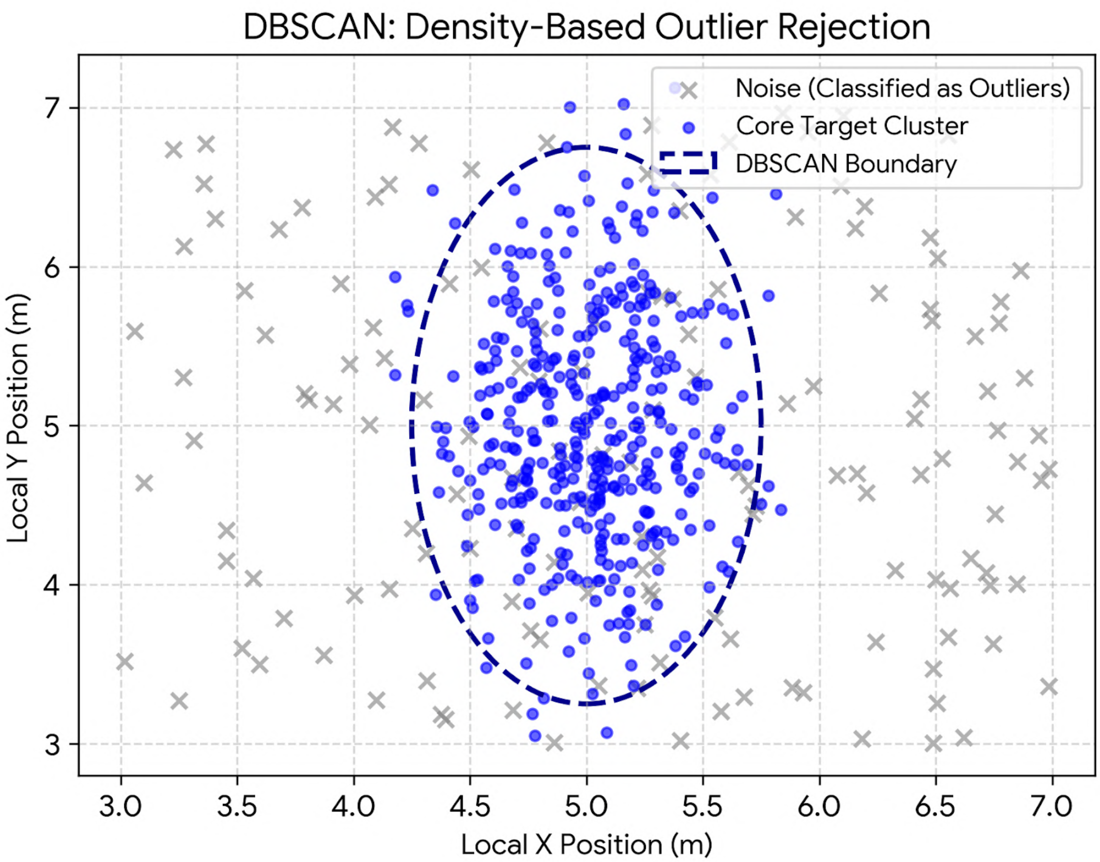


Figure 31: DBSCAN density-based outlier rejection.

Algorithm Trade-offs

The three algorithms have distinct precision-versus-accuracy and data-requirement profiles, summarised in Table XV.

Algorithm	Strength	Key parameters	Failure mode
STWE	Tunable precision/accuracy balance	R, λ_t, λ_r	Tuning-sensitive
Histogram	Robust gross outlier rejection	cell size Δ	Sub-cell quantisation error
DBSCAN	Sub-cm precision on dense clusters	ϵ, minPts	Degrades on sparse history

Table XV: Spatial filter trade-offs, key parameters, and dominant failure modes.

Task-Adaptive Weighted Fusion

Rather than hard-coding a single algorithm per task, mission control implements a convex combination of the three estimators based on *a priori* tolerances:

$$\hat{\mathbf{p}} = \alpha_S \hat{\mathbf{p}}_S + \alpha_H \hat{\mathbf{p}}_H + \alpha_D \hat{\mathbf{p}}_D, \qquad \alpha_S + \alpha_H + \alpha_D = 1 \tag{9}$$

Weights are switched based on the active task, biased toward the algorithm whose precision/accuracy profile matches the manoeuvre (Table XVI). Torpedo-launch alignment strictly biases toward DBSCAN while bin-drop vicinity acquisition biases toward histogram filtering and gate traversal uses a balanced default.

Task	α_S	α_H	α_D	Rationale
Bin	0.20	0.70	0.10	Macro-region acquisition
Torpedo	0.10	0.10	0.80	Sub-cm alignment required

Table XVI: Per-task fusion weights. Values reflect the precision/accuracy demand of each manoeuvre.

F.6 Gazebo-Based Simulator Development

Field testing is expensive in time and logistics, and reserving pool time for incremental algorithmic development is impractical. A high-fidelity simulator was developed to provide a repeatable environment for algorithm development, controller tuning, and mission validation, allowing pool time to be reserved for final-stage validation.

The simulator is built on three principles: (i) *Hardware-equivalent interfaces* - the simulator exposes the same sensor outputs and accepts the same control inputs as the real vehicle, so the autonomy stack runs unchanged across simulation and hardware; (ii) *modular assets and configuration* - world geometry, vehicle geometry, and runtime configuration are separated, with a versioned mapping file routing signals so topic, sensor, and actuator changes require no code modifications; (iii) *measured fidelity* - physical modeling (hydrodynamics, actuator allocation, sensor noise) is tuned to the vehicle's measured behavior across its operating envelope.

Hydrodynamic model:
Added-mass and damping coefficients (linear and quadratic) were derived from CFD sweeps over the vehicle's operating envelope and validated experimentally.

Actuation model:
The production configuration uses eight individual thruster plugins; force is applied per thruster, reflecting the physical layout and per-actuator limits. The body-level wrench abstraction used in earlier iterations was abandoned because it could not reflect actual propulsion layout or per-actuator authority limits. This configuration was validated through actuator-saturation and allocation tests and through trajectory-following experiments with injected disturbances.

Sensor models:
The simulator exposes a front-facing camera with matched field of view and resolution, an IMU with configurable noise parameters, and a depth measurement derived from rendered fluid pressure. Sensor topics use the same ROS 2 message types and frequencies as the hardware vehicle.

The simulator was developed over approximately five months (December 2025 – April 2026) in four phases, summarized in Table XVII.

Phase	Window	Focus
1	Dec'25 – Jan'26	Foundation + stability
2	Feb – Mar'26	Physics refinement
3	Apr'26	Integration + missions
4	Apr – May'26	Competition world + perception

Table XVII: Simulator development phases.

Phase 1 established a minimal pool scene with a placeholder vehicle to verify end-to-end signal paths. Phase 2 settled the eight-thruster plugin configuration and validated thrust allocation and disturbance rejection. CFD coefficients were imported and roll/pitch instabilities iteratively removed. Phase 3 added the service interfaces required for mission-level integration and ran multi-waypoint navigation experiments measuring timing, latency, and tracking error. Phase 4 populated the world with the full competition prop set at field-representative positions and the camera interface was also integrated for perception validation.

The simulator supports stable 6-DOF hover and attitude hold under nominal disturbances, repeatable waypoint navigation with bounded arrival tolerances, full task rehearsals for the gate, slalom, bin, torpedo, and octagon tasks with detection-triggered mission transitions, and vision-pipeline validation on simulated frames (bounding-box alignment, confidence metrics). A manual control mode is available for scene setup and debugging.

The simulator is the primary platform for PID tuning, multi-waypoint navigation, and perception validation prior to pool testing. External-interface differences between simulation and hardware are treated as defects, ensuring that the simulator remains a valid testbed and that hardware-integration surprises are minimised.

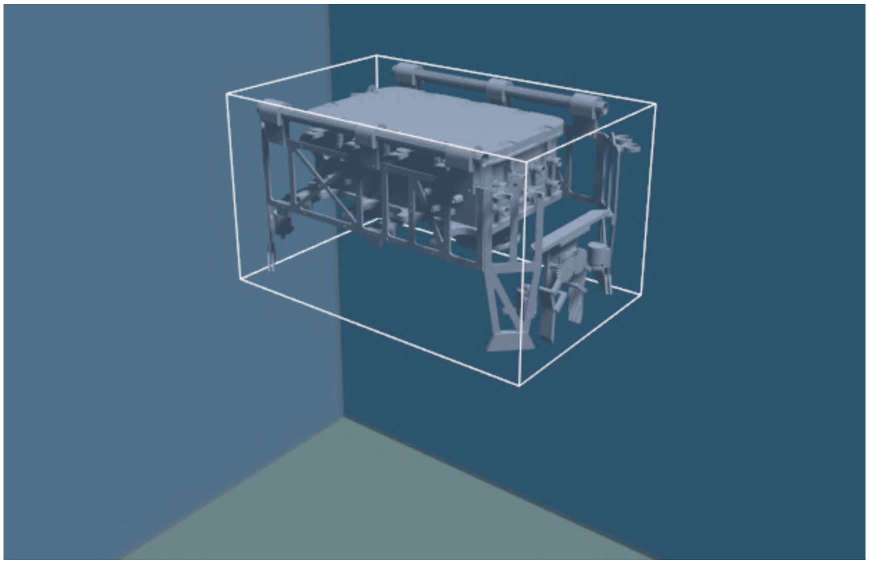


Figure 32: Simulator environment during the physics-model refinement phase.

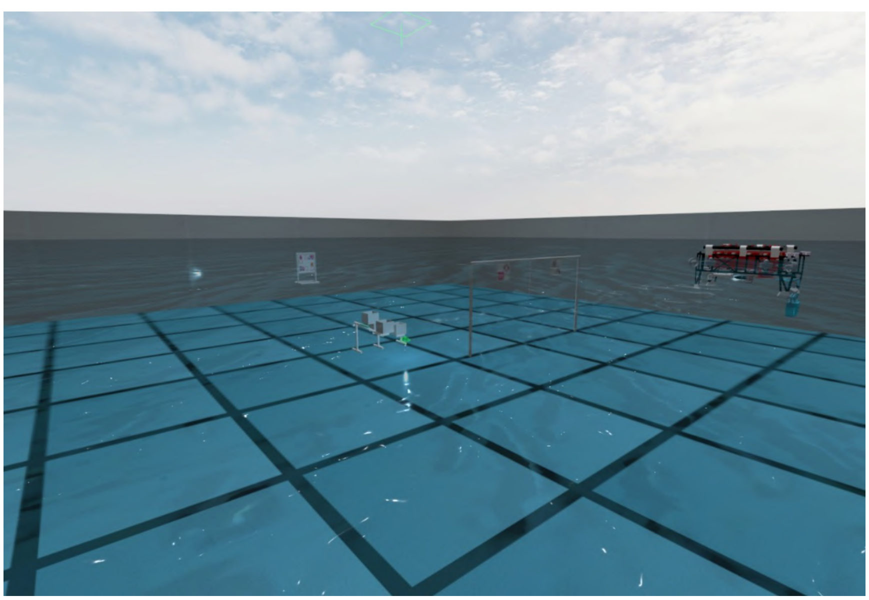


Figure 33: Competition-world environment populated with task objects (left) and a mission dry-run navigation sequence (right).

F.7 Mini Matsya Thrust Allocation

Mini Matsya operates with five controlled degrees of freedom (pitch is uncontrolled by design and passively stabilized through component placement) and carries six thrusters. The Matsya 7B allocator - a 6×8 matrix mapping a 6-DOF wrench command to eight thrusters - is therefore inapplicable and must be re-derived.

Allocator dimensions.
The wrench-to-thrust mapping is reduced from 6×8 to 5×6. The pitch row is removed entirely; the remaining rows correspond to surge X, sway Y, heave Z, roll K, and yaw N. Each of the six columns corresponds to one thruster's contribution to those five DOFs.

Matrix structure.
For a thruster i producing thrust T_i along its unit direction $\hat{\mathbf{d}}_i$ with moment arm $\mathbf{r}_i$ relative to the centre of gravity, the contribution to the wrench is

$$\begin{bmatrix} F_i \\ \boldsymbol{\tau}_i \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{d}}_i \\ \mathbf{r}_i \times \hat{\mathbf{d}}_i \end{bmatrix} T_i. \tag{10}$$

Stacking the six column vectors and dropping the pitch row yields the 5×6 allocator $\mathbf{B}$. The thrust command is recovered from the desired wrench $\boldsymbol{\tau}_d$ via the right-Moore–Penrose pseudo-inverse

$$\mathbf{T} = \mathbf{B}^{\dagger}\boldsymbol{\tau}_d = \mathbf{B}^{\top}\left(\mathbf{B}\mathbf{B}^{\top}\right)^{-1}\boldsymbol{\tau}_d. \tag{11}$$

The saturation logic from Matsya 7B is reused unchanged: per-thruster commands exceeding the individual thruster's operating envelope are clipped, and the entire thrust vector is rescaled to preserve the commanded wrench direction when collective saturation occurs.

The allocator is implemented as a configuration file load, with the matrix populated from the vehicle's thruster-layout JSON.

F.8 DVL-Free State Estimation: ESKF and GTSAM Trade Study

Mini Matsya does not carry a Doppler Velocity Log. Velocity-aided dead reckoning, the workhorse of Matsya 7B's localization, is therefore unavailable. Two properties of Mini Matsya's operating envelope justify a vision-inertial replacement: the effective course area for its assigned tasks (gate, slalom) is small compared to Matsya 7B's, based on the RoboSub'25 course layout, and the vision feed presents a near-constant set of competition props throughout the run. The combination of bounded operating volume and persistent visual landmarks makes visual-inertial fusion a viable primary localization source.

Two fusion variants are under evaluation. Both consume the OAK-D detection stream (3D landmark positions in the camera frame, refer Appendix F.2) and the BNO085 IMU stream at 100 Hz , and both produce a 6-DOF pose estimate at the same output rate as the existing Matsya 7B state estimator. The estimator slot in the pipeline is unchanged; only the internal implementation differs.

Variant 1: Error-State Kalman Filter (ESKF).
The ESKF maintains a nominal state $\mathbf{x}$ (position, orientation as a quaternion, linear and angular velocity, accelerometer and gyroscope biases) and an error state $\delta\mathbf{x}$. IMU measurements drive the propagation step; visual landmark observations from the OAK-D drive the update step. The error state is reset and injected into the nominal state after each update. The ESKF carries a fixed-size covariance matrix and runs in bounded time per cycle, making it the lower-latency option of the two.

Variant 2: GTSAM Factor Graph.
The GTSAM variant constructs a factor graph over a sliding window of recent poses. IMU pre-integration produces a single factor connecting consecutive pose nodes, and each visual detection produces a landmark factor anchoring the corresponding pose to the inertial landmark position. The graph is solved by incremental smoothing (iSAM2), yielding a smoothed trajectory over the window rather than a single filtered estimate. This produces lower drift over the window length at the cost of higher per-cycle compute.

Trade study (under evaluation).

Quantitative numbers from controlled experiments are pending. The qualitative trade-off is summarized in Table XVIII; final selection will be made on the basis of pool-test data once both variants are integrated against the simulator (Appendix F.6) and the production sensor harness.

Property	ESKF	GTSAM
Latency	Lower	Higher
Drift	Higher	Lower over window
Compute	Bounded	Window-dependent
Complexity	Lower	Higher

Table XVIII: Qualitative trade-off between ESKF and GTSAM visual-inertial fusion variants. Quantitative comparison is under evaluation.

The visual-inertial direction explored on Mini Matsya is not isolated. The long-term goal for the main Matsya line is a sensor-light, data-driven state estimator that reduces dependence on the DVL. This is a deliberate response to the budgetary constraints associated with sustaining a DVL-grade hardware suite year over year. Mini Matsya is the validation platform for that direction, and the ESKF/GTSAM trade study feeds directly into the Matsya 8 design decisions.

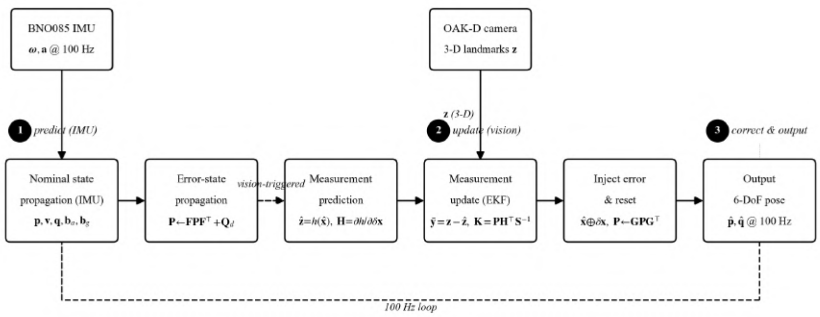


Figure 34: ESKF data flow: IMU drives propagation, OAK-D detections drive update, error state is reset into the nominal state each cycle.

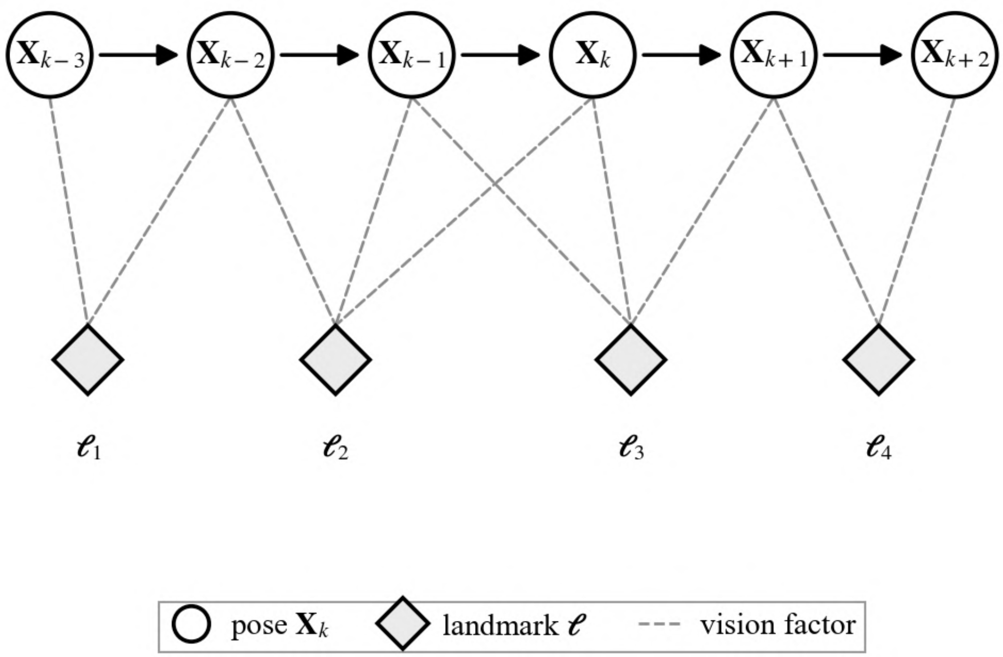


Figure 35: GTSAM factor graph over a sliding pose window.

F.9 APF-Based Reactive Obstacle Avoidance (Mini Matsya)

The APF module is deployed on Mini Matsya only. Mini Matsya's primary task is the slalom, which is the principal motivator for reactive avoidance; Matsya 7B does not carry the module.

Without reactive avoidance, the controller drives a straight-line path from the current position to the computed setpoint. Depending on the approach angle and any accumulated position error, this path can pass dangerously close to a slalom pole rather than cleanly through the corridor between poles. APF adds a tunable repulsive force around each obstacle so that the path is deflected sideways before contact, while the underlying PID controller continues to drive toward the setpoint.

Repulsive potential and force. The implementation uses the classical Khatib repulsive potential

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2} k_{\text{rep}} \left(\frac{1}{d(q)} - \frac{1}{r_{\text{rep}}} \right)^2, & d(q) < r_{\text{rep}} \\ 0, & d(q) \geq r_{\text{rep}} \end{cases} \tag{12}$$

and its negative gradient as the repulsive force:

$$\mathbf{F}_{\text{rep}}(q) = k_{\text{rep}} \left(\frac{1}{d} - \frac{1}{r_{\text{rep}}} \right) \frac{1}{d^3} (q - q_{\text{obs}}) \tag{13}$$

inside the influence radius, and zero outside. The magnitude grows sharply as the vehicle closes in: zero at the influence boundary, increasing nonlinearly toward infinity at zero distance.

The total repulsive force is the vector sum of per-obstacle forces. The computation is linear in obstacle count and produces a smooth field across most configurations.

A standard radial repulsion fails when an obstacle lies directly between the vehicle and the goal: the force opposes the PID's attractive drive and the vehicle stalls or oscillates. To address this, the repulsive force is decomposed based on its dot product with the goal direction. When the dot product is negative (the obstacle is in the forward path), a tangential component is added that steers the vehicle laterally around the obstacle:

$$\mathbf{F}_{\text{rep}} \mathrel{+}= k_{\text{tang}} \cdot \text{mag}(\mathbf{F}_{\text{rep}}) \cdot \hat{\mathbf{t}} \tag{14}$$

where the tangent direction $\hat{\mathbf{t}}$ is derived from the cross product of the world up-vector and the repulsion direction, oriented toward the goal side of the obstacle. The tunable gain k_{tang} controls steering aggressiveness; too low and the vehicle still stalls, too high and it overshoots the corridor.

Default parameters used during integration are listed in Table XIX. The influence radius r_{rep} trades detection envelope against intermediate-distance force strength: larger values trigger reaction earlier but shrink the $(1/d - 1/r_{\text{rep}})$ term across intermediate distances. The 0.69 m value reflects the sensor-detection range and the vehicle's stopping distance at operating speed.

Parameter	Symbol	Value
Repulsive gain	k_{rep}	5.0 (tuned per task)
Influence radius	r_{rep}	0.69 m per pole
Tangent gain	k_{tang}	tuned per task
Detection match	-	0.5 m
Control rate	-	10 Hz

Table XIX: Default APF parameters for the slalom task.

Without deduplication, the same physical pole observed across 20 frames produces 20 superimposed repulsive vectors, inflating the lateral force by an order of magnitude and causing violent deflection. A proximity-threshold-based deduplication merges incoming detections with the existing obstacle list: a new detection within 0.5 m of an existing obstacle updates the existing entry in place rather than appending. Existing obstacles with no matching detection are retained, providing robustness to single-frame occlusion.

The APF module is implemented as a standalone Python class with no `rclpy` dependency, allowing unit testing with pure NumPy. The controller node instantiates it during initialization with parameters from configuration. On each 10 Hz cycle, the PID forces are computed first; the APF repulsive force is then computed using the current AUV position and added *only* to the surge and sway components of the PID output. Heave, roll, pitch, and yaw are left unmodified - depth is held at a constant slalom level and the orientation channels remain under PID for heading hold. The combined force vector is then passed through the thrust allocator to produce PWM commands.

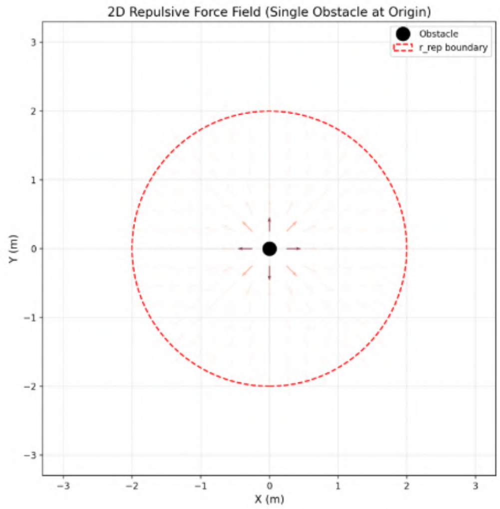


Figure 36: 2D repulsive force field around a single obstacle where intensity indicates magnitude. The dashed circle marks the r_{rep} influence boundary.

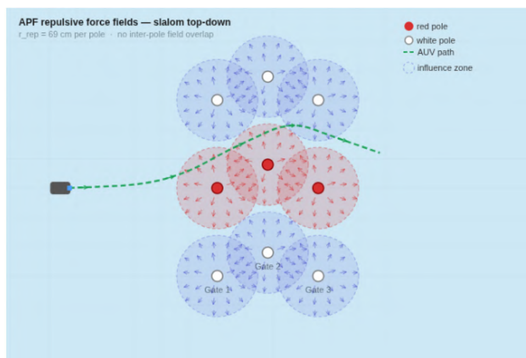


Figure 37: Slalom top-down view with per-pole influence zones (radius 0.69 m). The dashed trajectory shows the vehicle trajectory.

F.10 Mission Control & Localization

To ensure continuous operation during partial task failures, Mission Control incorporates a robust fault-tolerance framework. Every task is wrapped in a procedural exit strategy designed to prevent system stalls. Also, in the event of a sudden vision dropout or vehicle misalignment, this strategy dictates a best-effort task continuation relying on the last known best navigation estimate, paired with an automated emergency scan to reacquire the target. This guarantees that isolated sensor dropouts or mechanical delays do not cascade into complete mission failure.

Emergency Scan and Vision Recovery Protocol

Extended intervals without visual detections indicate the target has left the camera's field of view. Rather than executing a generic sweep, the system mitigates sudden vision dropouts by triggering a task-specific Emergency Scan tailored to the most probable failure mode. For example, during the Torpedo sequence, vision loss frequently occurs due to extreme proximity to the target plane, causing the bounding box to clip the camera's field of view. This triggers a vision-based fallback where the vehicle executes a horizontal reverse surge maneuver to regain optimal standoff distance, followed by a localized zig-zag search pattern sweeping laterally and vertically while avoiding collision. Conversely, for downward-facing tasks like the Bin Marker Drop, vision loss typically results from horizontal drift pushing the target outside the camera's footprint. If the elapsed time without a detection exceeds the threshold, the protocol commands a bounded vertical ascent to instantly widen the downward field of view, immediately followed by an expanding rectangular spiral scan in the horizontal plane to sweep the immediate seabed. Once the target is reacquired in the visual frame, the emergency scan terminates immediately, and the vehicle resumes closed-loop tracking.

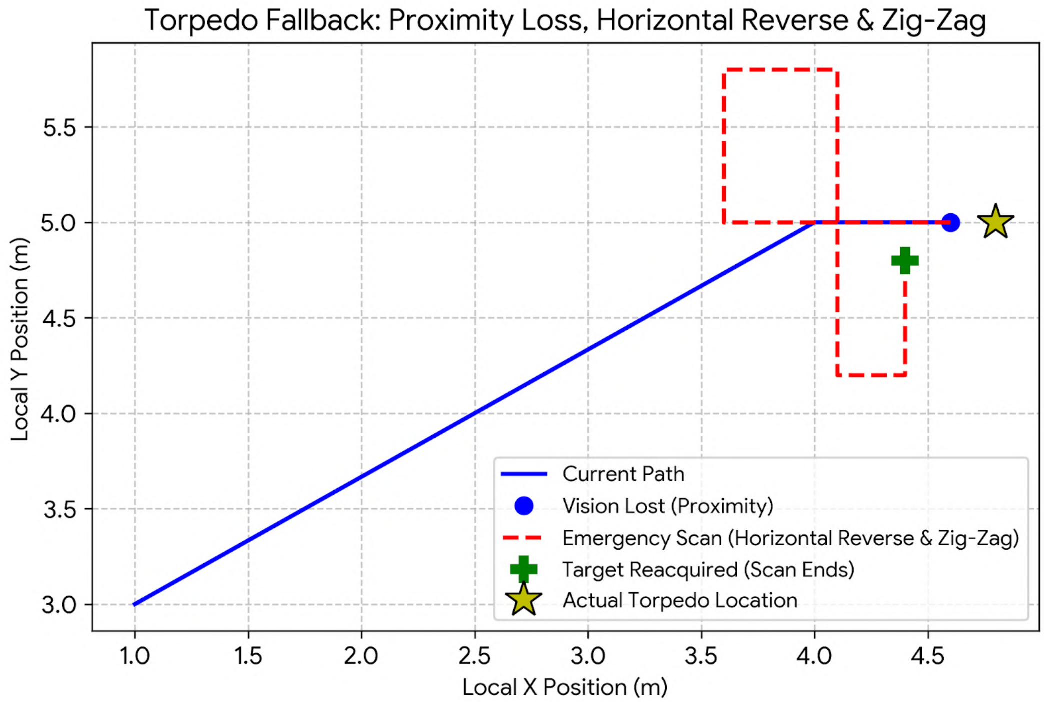


Figure 38: Torpedo Alignment Fallback.

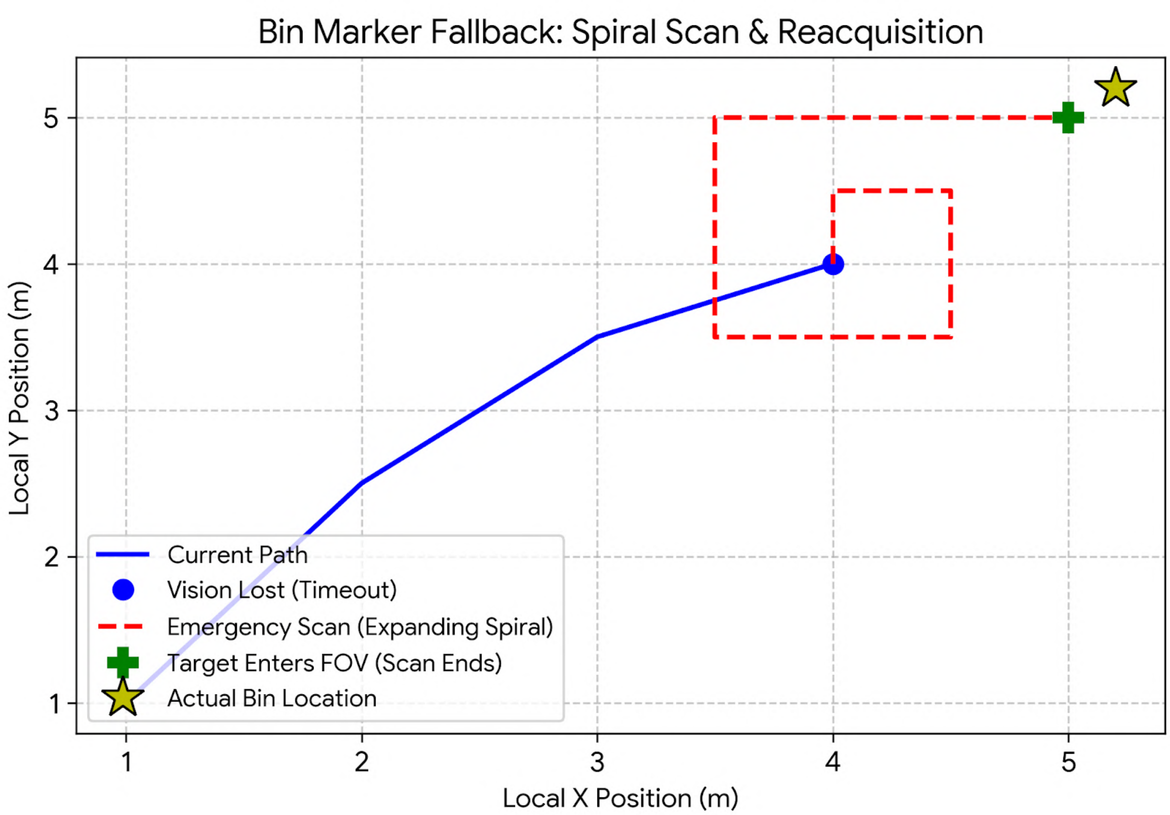


Figure 39: Bin Marker Fallback.

Timeout Thresholds and Trigger Conditions

Each task is assigned a strict execution window governed by predefined timeout thresholds monitored by the vehicle's mission planner. If a task fails to reach its precise closed-loop completion criteria within the allocated timeframe, the mission control triggers an open-loop exit strategy based on the current task. For instance, during the Bin Marker Drop, if the alignment timeout is reached before optimal visual centering is achieved, the vehicle abandons the active search phase, automatically navigates to a predefined target depth, and releases the marker. Similarly, during the Torpedo Launch sequence, which demands high-precision angular alignment, reaching the timeout causes the vehicle to cease fine-tuning its yaw and pitch. It instead maneuvers to the best-known spatial coordinates of the target and executes the launch to ensure partial point acquisition. This transition to a designated fallback protocol ensures the vehicle abandons the stalled phase and utilizes its last estimated navigation state, prioritizing an execution attempt over perfect accuracy. Ultimately, this bounds the maximum time spent in a degraded sensory state to maintain forward momentum and prevent mission deadlocks.

APPENDIX G: LTSPICE VALIDATION OF LTC4365 PROTECTION CIRCUIT

An LTSpice simulation model of the **LTC4365** protection circuit was developed prior to PCB fabrication to validate the resistor calculations used for setting the protection thresholds. The simulation was performed to verify cutoff voltages and validate reverse polarity protection before hardware implementation.

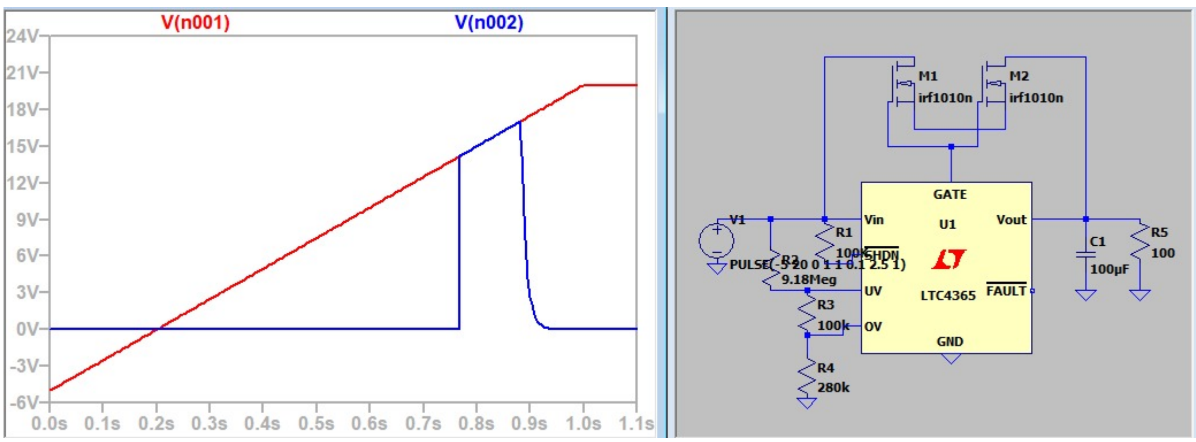


Figure 40: LTSpice simulation model of the LTC4365 protection circuit

Appendix H: PWM Demultiplexing Validation

A dedicated validation setup was developed to verify the PWM demultiplexing architecture used for multi-channel thruster control. Testing was performed to ensure that the demultiplexer IC correctly routed the PWM input signal across multiple output channels while preserving pulse width and timing characteristics required by the ESCs. The oscilloscope measurements confirmed successful sequential distribution of PWM signals without distortion.

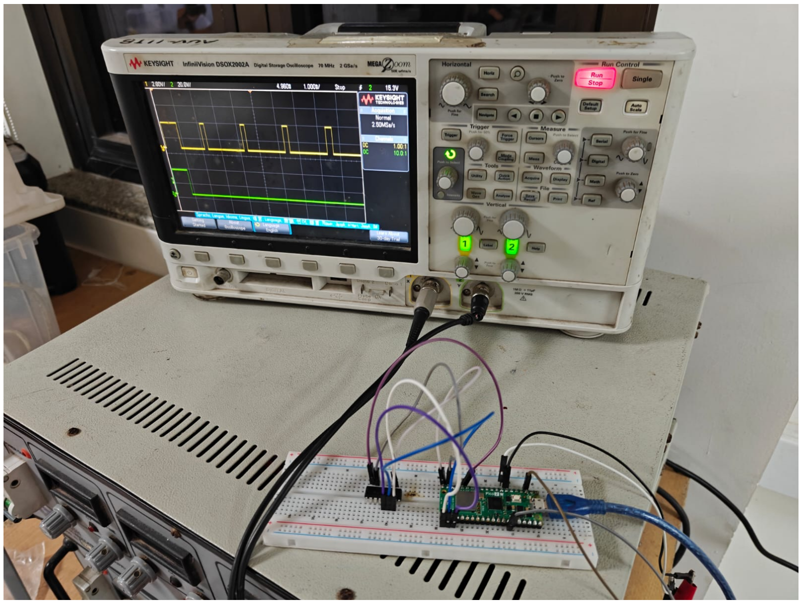


Figure 41: Experimental validation of the PWM demultiplexing architecture showing the demultiplexer setup and oscilloscope measurements

APPENDIX I: MINI MATSYA ELECTRICAL PCBS

The electrical architecture of Mini Matsya is implemented using a dedicated two-board design consisting of a Power Board and a Signal Board. Separating high-current power-handling and signal-interfacing circuitry reduces electrical interference while improving modularity and maintainability.

Figure 42: Mini Matsya electrical PCBs showing the Power Board and Signal Board

APPENDIX J: MATSYA 7B ELECTRICAL PCB

The Matsya 7B electrical stack integrates power regulation, monitoring, communication, and signal interfacing functionalities into a unified architecture designed for improved reliability and simplified debugging.

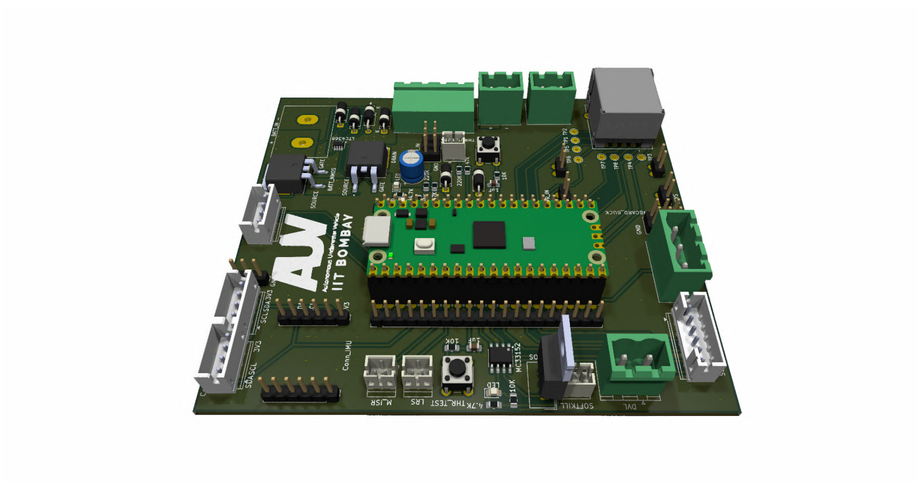


Figure 43: Matsya 7B electrical PCB

Appendix K: Hall Effect Sensor Characterization for Kill Switch Validation

A characterization test was performed on the Hall effect sensor used in the magnetic kill switch system to determine the output voltage levels corresponding to different magnetic field conditions. Oscilloscope measurements were used to map the sensor response and identify the operating voltage thresholds at the output pins.

The characterization was performed to establish reliable trigger thresholds and prevent unintended activation of the kill switch due to noise, minor magnetic field variations, or environmental disturbances during vehicle operation.

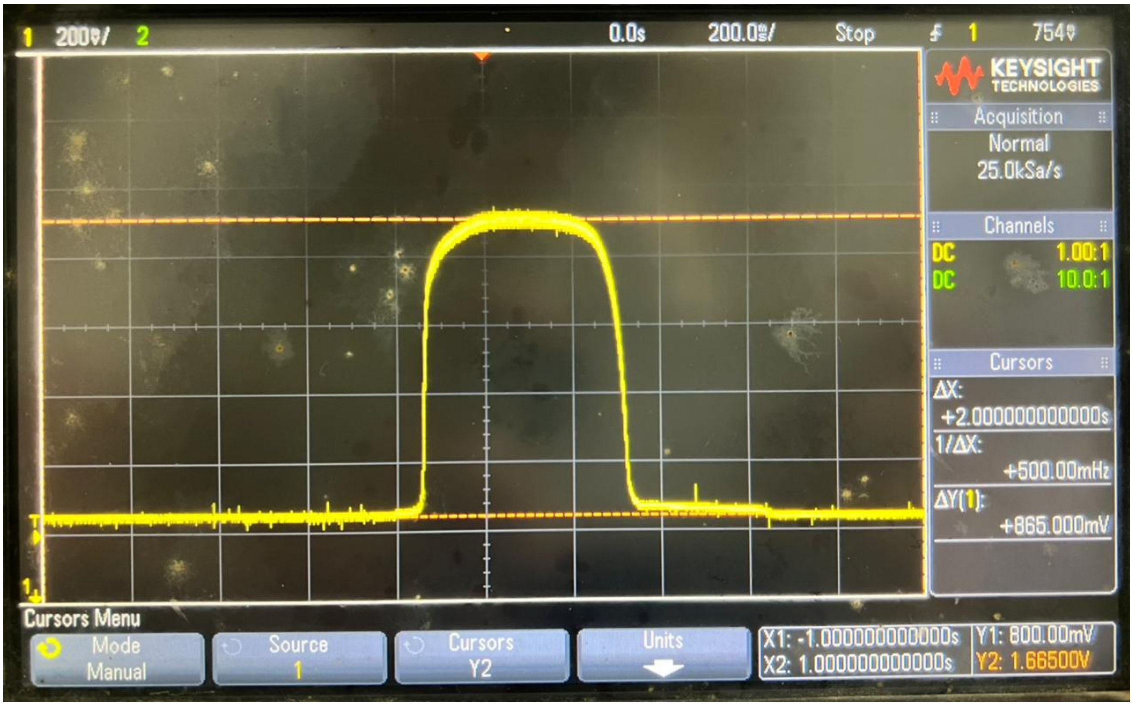


Figure 44: Oscilloscope capture used for Hall effect sensor characterization and trigger threshold verification

Appendix L: INA3221 Current Sensor Validation

The **INA3221** current sensor was validated by comparing its measurements against reference values obtained using a Digital Multimeter (DMM). Current measurements for a single thruster were recorded across different PWM values to verify the sensor response and measurement accuracy.

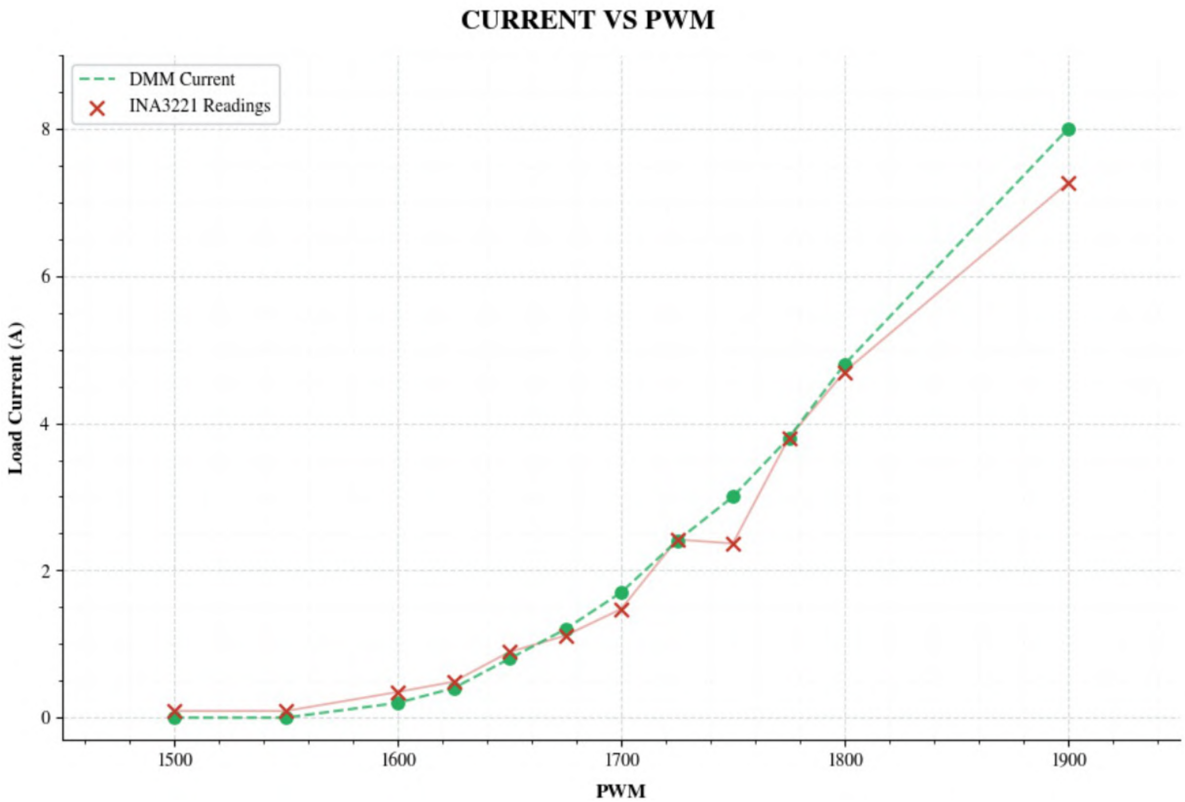


Figure 45: Comparison of DMM and INA3221 current measurements across varying PWM values

Appendix M: Mechanical Design and Manufacturing Process

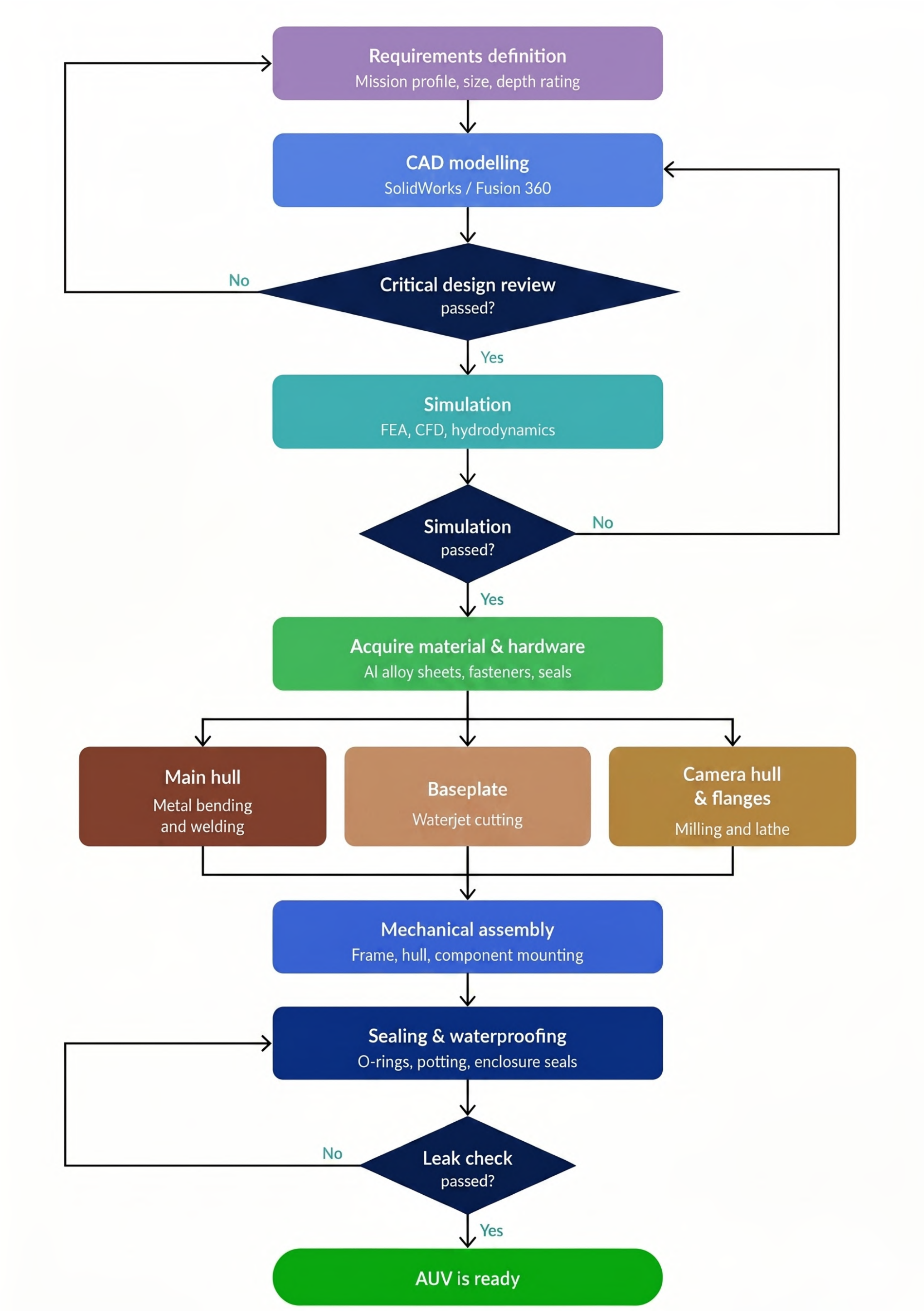


Figure 46: Gripper Finger Progress

Appendix N: Computation of Force to PWM mapping for Mini Matsya's new Thrusters

There were no published PWM-to-thrust curves or force ratings available for the Apisqueen U2 Thrusters fitted to Mini Matsya (decision driven to save costs). Since both the thrust allocator and PID controller depend on an accurate force-to-PWM mapping, a bench characterisation was conducted before deployment.

Experimental Setup

The thruster was mounted via a custom 3D-printed bracket onto a frictionless linear rail, constraining motion to a single axis aligned with the thrust direction. A spring balance was attached in-line to resist and measure the produced force. The thruster was swept across its full PWM range (1100 µs to 1900 µs in 25 µs increments), with spring balance readings recorded at each setpoint once steady state was reached. The following precautions were taken to minimise systematic error:

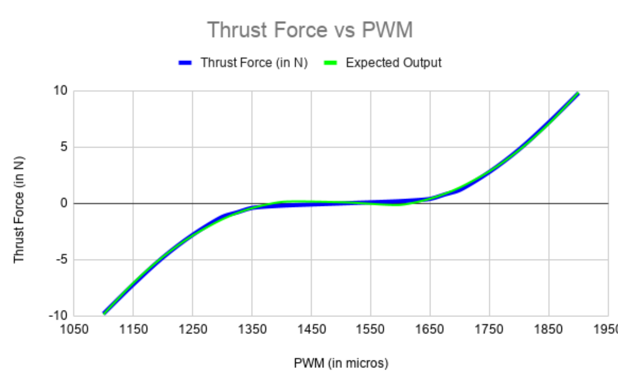
- **Frictionless rail:** Eliminated parasitic drag that would otherwise underread thrust at low PWM commands.
- **Steady-state reads only:** Readings were recorded only after motor RPM stabilised, avoiding transient overshoot.
- **Repeated runs:** Each setpoint was repeated three times; the mean value was taken to reduce noise.
- **Submerged operation:** The thruster was operated underwater to replicate actual operating conditions.
- **ESC warm-up:** The ESC was brought to operating temperature before measurements were recorded.
- **Spring balance zeroing:** The balance was re-zeroed before each run to eliminate drift from prior loading.
- **ESC deadband:** The ESC deadband region around neutral (1500 µs) was identified and excluded from curve fitting.

Results and Validation

The resulting thrust-force curve is shown in Fig. 50. Validation was performed on two levels. Analytically, thruster thrust is expected to follow a quadratic relationship with PWM: motor RPM scales approximately linearly with PWM offset from neutral, and thrust scales with RPM squared, yielding an inherently quadratic composite response. Fitting a quadratic curve to the measured data yielded $R^2 = 0.987$, confirming the model accounts for over 98% of the variance with no systematic deviation across the full PWM range.

Empirically, three independent runs at each setpoint produced a repeatability of ± 0.01 N, confirming the stability of the spring balance setup and the effectiveness of the frictionless rail. The ESC deadband was centred at 1500 µs with a width of approximately ± 30 µs, confirming accurate ESC calibration. The characterisation map is considered reliable for use in thrust allocation and PID gain tuning.

(a) Experimental Setup

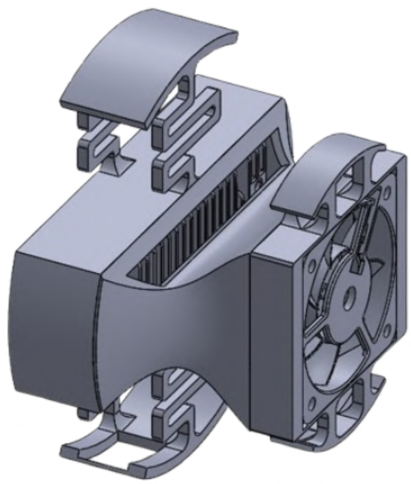


(b) Force to PWM mapping

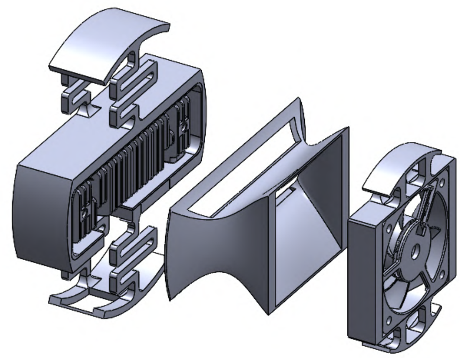
Figure 47: Empirical Estimation of Thrust Force to PWM mapping

APPENDIX O: CAMERA COOLING SYSTEM

The onboard camera is housed within a narrow cylindrical enclosure, which confines heat to a small volume elevating local temperatures, degrading performance, and raising the risk of long-term damage. During continuous YOLO inference, the OAK-D camera dissipates approximately 6 W of heat. To manage this, a 40 mm axial fan is tightly coupled to an aluminum heatsink, driving airflow through the fin channels and drawing heat away from the camera. This thermal management solution has been validated through both simulation and physical testing, reducing the camera's steady-state temperature from 65 °C to 49 °C, well within the 70 °C operating limit.

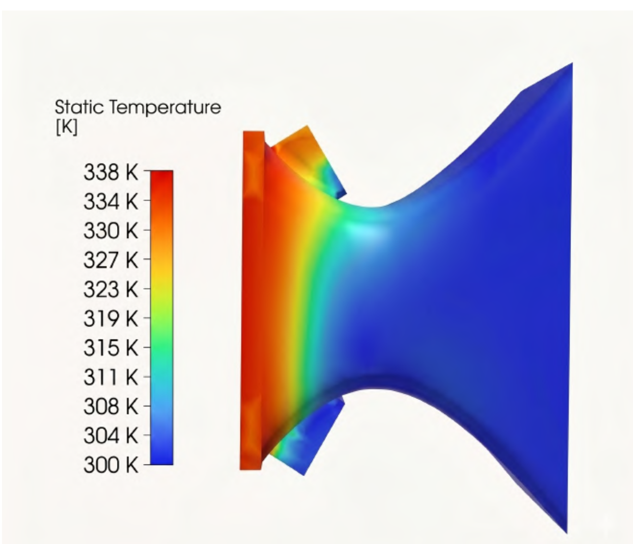


(a) Fan Mount

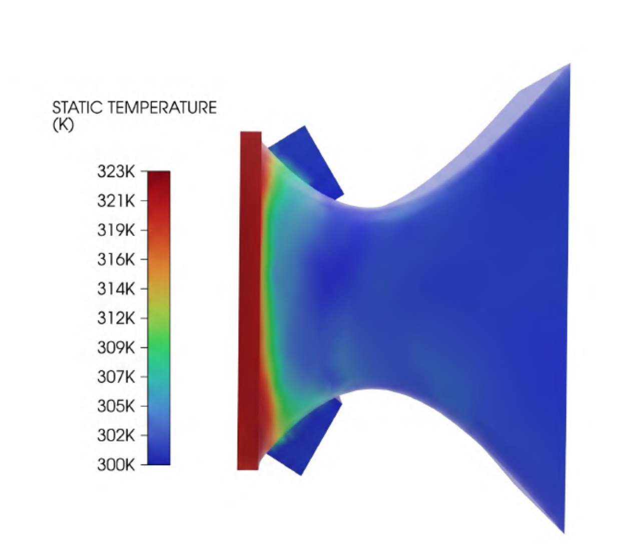


(b) Fan Mount (Exploded)

Figure 48: Camera Cooling System



(a) Temperature Profile at t=0



(b) Temperature Profile at t=20s

Figure 49: Temperature Profiles across 20 seconds

Appendix P: Outreach Activities

Community Development

In recent years, we have observed a massive spike in interest towards underwater robotics shown by university students and industries across India. We were getting calls from students of **NIT Rourkela**, **IIT Kanpur**, and industries like the **Defence Research and Development Organisation**, **Larsen & Toubro**, and **Tata Power** to discuss the nuances and applications of building underwater vehicles. Acknowledging this, we have focused this year on laying the foundation for building an underwater robotics community in India.
The first step towards this came with creating a **Discord server** and onboarding students from universities like IIT Kanpur and ISM Dhanbad as our early partners. We focus on putting regular updates regarding our technical updates and facilitating knowledge exchange. Our future aim is to leverage this network to further promote the development of student teams in underwater robotics.

Technical Exhibitions

We presented at the IIT Bombay's flagship technical exhibition, **TechConnect**, in December 2025. This exhibition was graced by the presence of top industrialists like **Mr Narayan Murthy (Founder of Infosys)**, top officials from the Indian defence, senior individuals in academia and students across India. The footfall of this exhibition was upwards of **150,000**.

(a) Presenting to Mr. Narayan Murthy (Founder of Infosys)

(b) Presenting to former Chief of Naval Staff, R. Hari Kumar

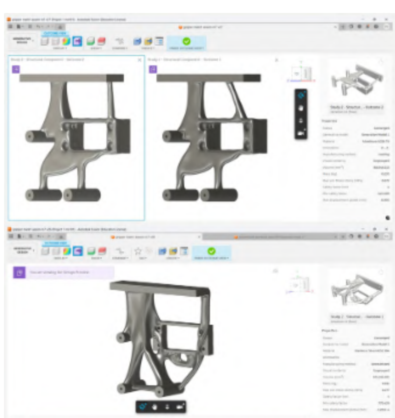
Figure 50: Presenting at TechConnect, 2025

Lab Open Week

In October 2025, we organised a lab open week, allowing campus residents to visit our lab, **increasing awareness** of Underwater Robotics

Industrial Partnerships

We have partnered with **Autodesk** on several project-based studies to evaluate their latest **Generative Design** feature. This year, we successfully concluded two long-term industrial projects with **Larsen & Toubro** (building an ROV to inspect underwater pipelines) and a **Towfish** project (a submersible being towed by a boat for the Centre of Excellence in Oil, Gas & Energy).



(a) Generative Designs for Gripper Frame

(b) Towfish Project

Figure 51: Industrial Partnerships