Technical Design Report of Matsya 7, Autonomous Underwater Vehicle

Shubham T, Kaushik C, Sameer M, Archit S, Ayush P, Anshu A, Anirudh J, Urvi G, Sparsh B, Adwai K, Khushajh V, Jigar M, Shresth K, Aniket G, Shlok R, Manit J, Divyanshu G, Jash D, Akshika J, Khushi T, Divya M, Anuj P, Ayush K, Jinay V, Arush S, Avani G, Supriya M, Shivakshi G, Preetham P, Denver Z, Pratham C, Dhruv S, Kiara T, Nilkrishna, Tanvi G, Mridhula V, Saket V, Satyadev S, Shivansh G, Abbas H, Shaurya G, Aaditya P, Aayush S, Ankush, Kunal P, Rucha R
Faculty Advisors: Prof. Leena Vachhani and Prof. Shashi Ranjan Kumar

***Abstract*—At RoboSub 2025, AUV-IITB is unveiling Matsya 7, a vehicle built from the ground up with a focus on rapid testing and robust performance. Prioritising quick, informed iteration, a compact single-board electrical stack was developed, integrating sensors, status LEDs for real-time feedback, and onboard SD card logging. Wetlink penetrators and cable sleeves streamlined assembly, halving stack build time. The actuation system was redesigned from scratch, eliminating motor stalling and introducing a simplified reloading mechanism. To improve performance in vision-based tasks, the front camera was upgraded to the OAK-D, enabling stereovision-based depth estimation. Aesthetically and ergonomically refined, Matsya 7 features a jet-black frame and carbon fibre handles, marking a confident return to the monohull design philosophy.**

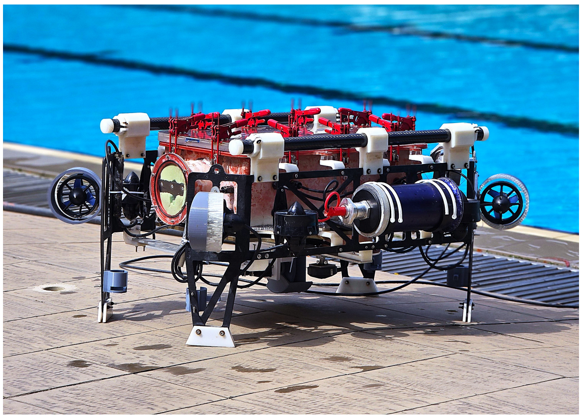


Fig. 1: Matsya 7

I. Competition Strategy

Matsya 7 is the team's first new vehicle post-COVID. The design philosophy behind this vehicle was to make simple, yet robust and functional systems in order for it to be competition ready i.e, capable of attempting all tasks, within just 6 months. We achieved this by revamping all our electrical and mechanical systems with a strong focus on reliability, incorporating our cumulative learnings from Matysa 6[1]-[4].

Driven by critical limitations in our previous monocular setup which relied on depth inferred from 2D images, we've shifted our front camera to the OAK-D this year. The previous approach introduced significant errors due to its dependence on object size priors and geometric approximations, resulting in unreliable 3D-mapping. In contrast, integrating the OAK-D's stereovision-based depth estimation[5] into our YOLOv8[6] pipeline has proven far more consistent (Fig. 2), producing more accurate and precise depth estimates. See Appendix B for further details.

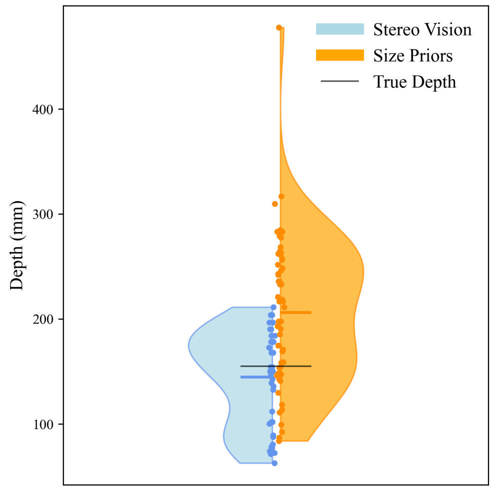


Fig. 2: Comparison of Depth Estimates

To address the complexity of Matsya 6's multi-hull system and asymmetric thruster layout, Matsya 7 was redesigned for practicality. The monohull has fewer potential points of failure and streamlined both manufacturing and assembly. Wetlink penetrators replaced our in-house epoxied ones, increasing configurability and making the waterproofing more reliable. The thrusters are now mounted in a 4-surge, 4-heave arrangement, enhancing control and tunability, at the marginal cost of a 5% reduction in top speed.

I-A. Course Strategy

- Heading Out (*Coin Flip*) & Collecting Data (*Gate*): We start the run off with a yaw scan to detect and align to the gate.
- Navigate the channel (*Slalom Poles*): By filtering on confidence and sorting by depth, we detect the first red pole using a yaw scan for alignment, navigate forward to clear the triplet and then repeat for the second and third poles. Depending on the offset direction (right or left) of the second pole, we've found that the estimated setpoints are more reliable if the vehicle also navigates the same side of the poles.
- Drop a BRUVS (*Bin*): We first detect the bin's white flange to reduce the chance of missing the task, then align to the images inside. A bottom-heavy design ensures the markers fall along a vertical path.
- Random Pinger: In the tests we have conducted, we can reliably detect pings about 65% of the time. Yaw and position estimation are unreliable as there is a small margin for error in ping detection. Hence, we use a right-or-left strategy. See Appendix C for further details.
- Tagging (*Torpedoes*): Our spring-loaded torpedo shooter has been improved, achieving an increased range of 2 feet in water. We'll attempt shooting our torpedoes from far away for extra points. Using the stereovision pipeline, we align to the torpedo flex, and size priors for finer alignment to the holes.
- Ocean cleanup (*Octagon*): To better our chances at successfully picking up the cup and the spoon, we designed a flat-fingered gripper. Monitoring the current drawn by the motor, the vehicle detects if the object was gripped or not, saving run time by skipping visual verification of task completion.
- Return home: To conclude the mission, we'll chart a path back home ensuring none of the tasks are hit on the return journey, and perform a barrel roll upon successful return. We are 100% confident that we will finish in style.

II. Design Strategy

II-A. Mechanical Subsystem

The mechanical subsystem ensures robustness through structural, hydrodynamic, and thermal analyses, iteratively prototyping each design until all requirements are met.

Vehicle Structure

- *Hulls:* Reverting to a monohull design greatly expedited Matsya 7's timeline while improving electrical safety and convenience. The modular, quick-access battery hull was retained. Unlike Matsya 6, which required unscrewing endcaps, Matsya 7's cameras are housed in cylindrical extrusions on the front and bottom, allowing easier access and maintenance.
- *Frame and Handles:* The frame supports the hulls and the actuators. Static structural simulations were performed on ANSYS (see Appendix G for further details) to reduce weight without compromising structural integrity. The handles were changed to a more ergonomic design using hollow carbon fibre tubes which offer a better strength to weight ratio and a comfortable grip along its entire length.

Actuators

Matsya 7 has new, compact electromechanical actuators, designed using waterproof servo motors and 3D printed parts.

- *Marker-Dropper:* A servo-driven circular blocker moves across two barrels (Fig. 3), letting markers fall under gravity, one by one. This year, the design was made twice as compact by repositioning the servo and the blocker. A magnetic lid was also added to prevent accidental dropping of the markers during barrel rolls.

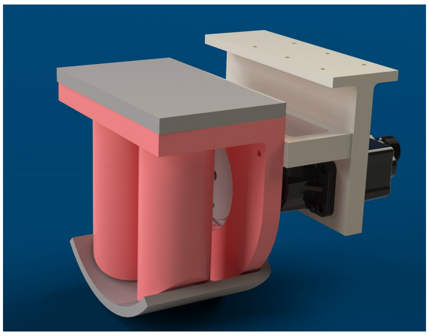


Fig. 3: Marker Dropper

- *Torpedo Shooter:* Two spring-loaded tubes with adjustable lengths (Fig. 4) allow for a tunable launch force, achieving a range of 2 feet in water.

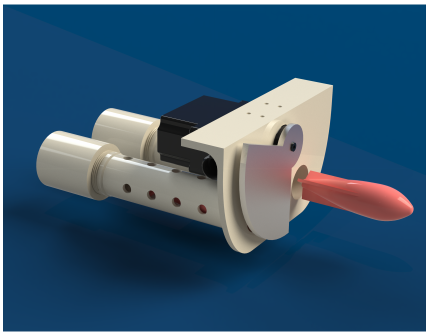


Fig. 4: Torpedo Shooter

- *Gripper:* Made of two 4-bar linkages coupled together with a gear (Fig. 5), the fingers follow an elliptic trajectory making the mechanism compact in the unactuated state. The 3D-printed fingers have rubber pads for better grip.

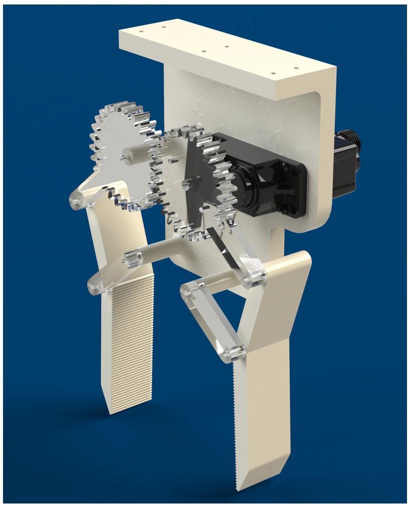


Fig. 5: Gripper

II-B. Electrical Subsystem

The electrical subsystem bridges the gap between the mechanical and software subsystems, designing boards in-house that interface peripherals, manage power distribution and handle efficient data packeting and transfer.

Improvements in the Electrical Stack

This year, a sleek single board employing the RPi Pico has replaced Matsya 6's modular four-board setup, leveraging maximum functionality using the USB's asynchronous duplex communication (Fig. 6).

An inline ESC layout and cable sleeves streamlined internal wiring, speeding up assembly. The ESCs are now mounted flush on a metal plate above the hull's baseplate, improving cooling while keeping them elevated and safe from potential leaks.

Communication

A custom driver was built on top of TinyUSB for bidirectional communication between the RPi Pico and the SBC, with a variable message length, and encoding optimisations for fast speeds of up to 1000 messages per second. A robust and modular three-way communication pipeline (Fig. 7) was setup to allow the Pico to stream data to the testing laptop through the SBC.

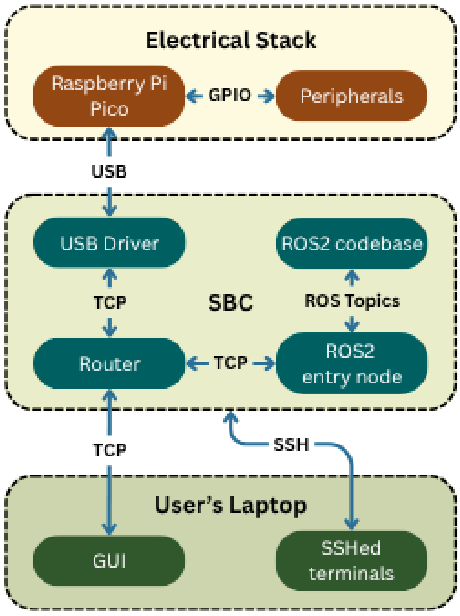


Fig. 7: Communication Pipeline

Reliability & Safety Features

The flow of information was central to Matsya 7's design. The RPi Pico lifted previous hardware limits on message headers, and new `VERBOSE` and `DEBUG` modes enabled rapid actuator tuning.

An SD card now logs all data routed through the stack with timestamps, helping us trace electrical failures and separate them from software issues.

Status LEDs linked to mission planner states provide real-time visual feedback for easier debugging (Fig. 8).

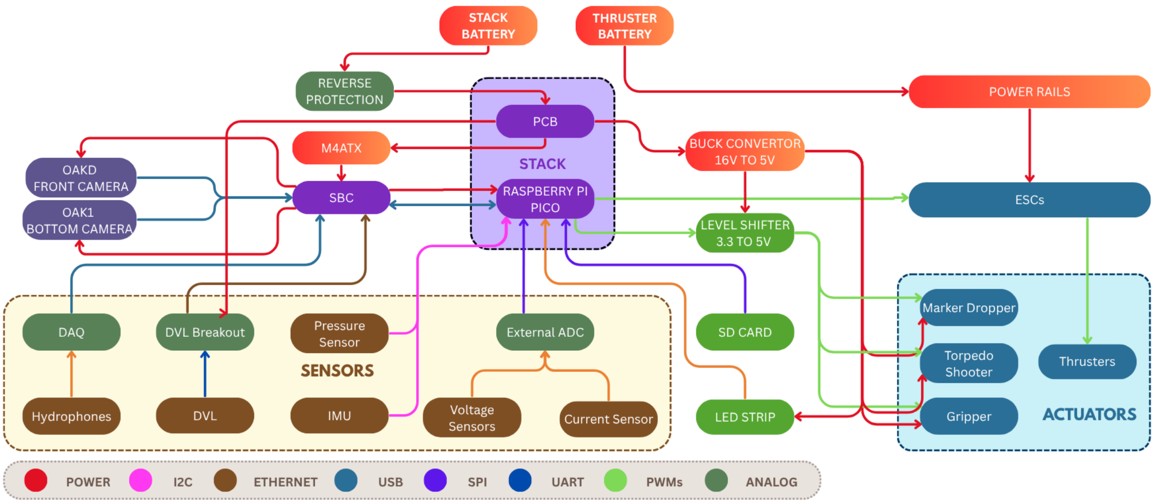


Fig. 6: Power & Data Flow of the Electrical Stack

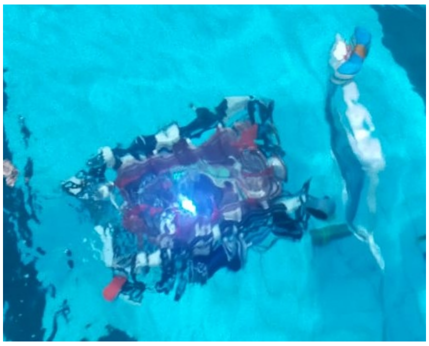


Fig. 8: Matsya 7 executing a scan (Blue) for Tagging

Current and voltage sensing was introduced on the power lines for early detection and warning of electrical failures. Fuses and MOSFET toggles on the actuators' power line act as fail safes against motor stalling, preventing them from burning up.

A debugging interface (Fig. 9) was also integrated into the main GUI to streamline the testing of the electrical stack and its peripherals. This allows us to test commands and get comprehensive logs quicker than before.

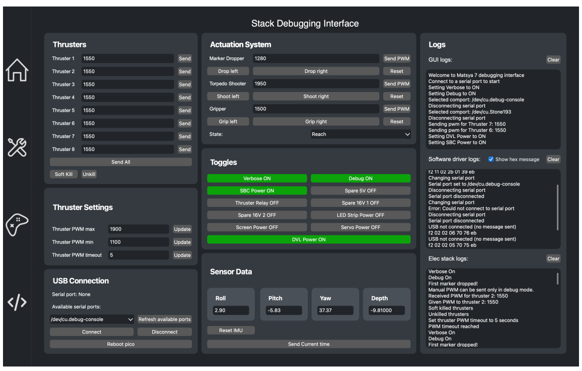


Fig. 9: Debugging Interface

II-C. Software Subsystem

The software subsystem encompasses a comprehensive stack designed to decode Matsya's environment as captured through its various visual, acoustics, and inertial sensors and process this information to facilitate finely tuned autonomous underwater operations.

Our source code is written primarily in Python and is organized into distinct core modules, including vision, acoustics, localization, navigator, controller, and mission control. We employ Robot Operating System (ROS) to enable effective communication between these modules (Fig. 10).

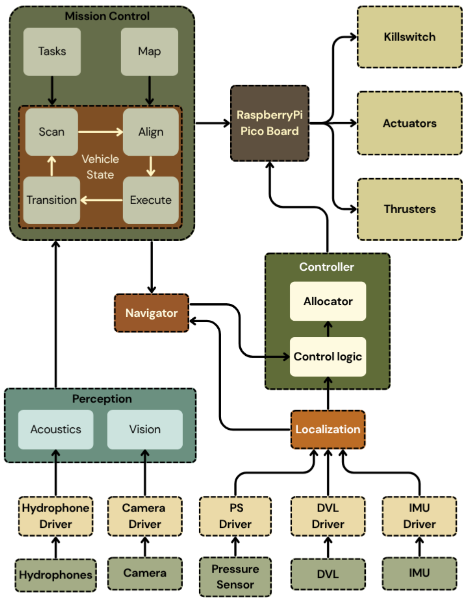


Fig. 10: Software Architecture of Matsya 7

Software Upgrades

As we upgraded our systems to Ubuntu 22.04 LTS, and with Ubuntu 20.04 LTS nearing end-of-life, we transitioned from ROS1 (Noetic) to ROS2 (Humble), as ROS1 lacks official support for this platform. As part of this migration, we moved away from the traditional service-client model and adopted an action-server architecture. This enabled more robust handling of long-running tasks by leveraging asynchronous execution, real-time feedback, and task cancellation, ensuring reliability and responsiveness in dynamic environments.

Improvements in Vision

Matsya 6's software stack estimated the distance of a task by calculating the ratio of the size of the detected bounding box, and the expected size of the object. This proved to be effective as the vehicle is always level when performing a task.

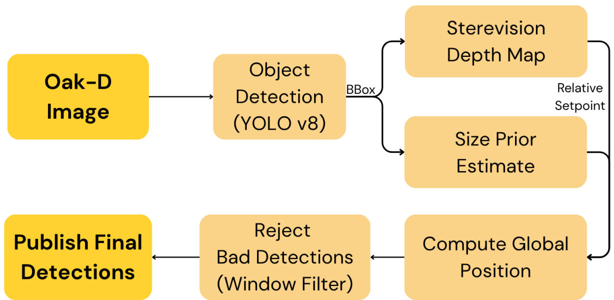
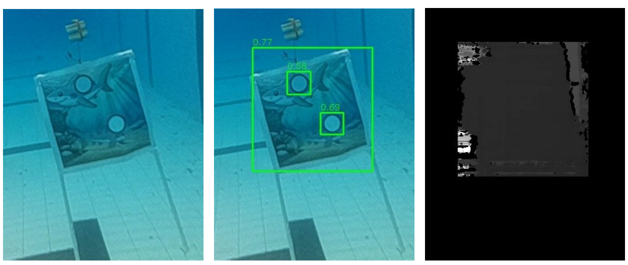


Fig. 11: Image Processing Pipeline

The image processing pipeline was revamped (Fig. 11) to utilise the OAK-D's stereovision. To get rid of distortion in the received image frames, we first perform a corrective transform (see Appendix B for more details), following which we estimate the depth of the object by averaging out the values in the depth frame in a buffered region surrounding YOLOv8's bounding box (Fig. 12).



(a) Raw Image (b) YOLOv8 BBox (c) Depth Estimate

Fig. 12: Image Processing Pipeline using Stereovision

However, we see that the stereovision depth frames are not able to detect small objects like the holes in the torpedo flex. Thus, for certain tasks, we revert to our original image processing algorithm using size priors.

Improvements in Acoustics

In the acoustic pipeline, we first apply a bandpass filter to isolate the ping signal, then narrow in on the peaked regions. The envelope is then found, and its derivative is used to find the point of sharpest rise.

We found that the envelope peak right after the rise best marked the ping's start, based on extensive analysis of gigabytes of noisy acoustic data.

Even after processing, time difference of arrival (TDoA)[7] was unreliable, so we adopted a simpler left-or-right method: whichever hydrophone detects the ping first guides the vehicle to the next task.

Testing Interface

A major development this year is our new Graphical User Interface (GUI), which aims to accelerate software testing. It features:

- Pose, setpoint, and common data logging, along with a course map view.
- Raw & processed camera feeds using JPEG compression, saving over 90% bandwidth versus X11 forwarding with ROS2's native rqt image view.
- An ROV mode, for data collection.

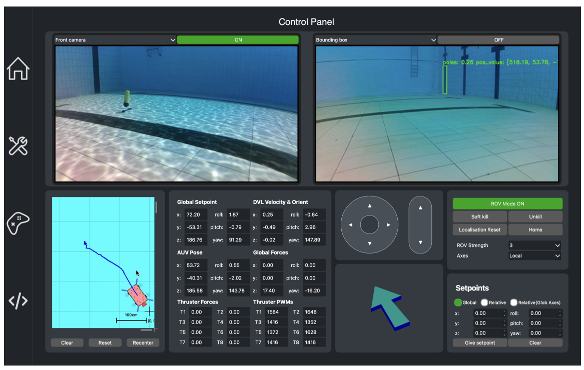


Fig. 13: Control GUI

III. Testing Strategy & Results

The mechanical and electrical subsystems work in parallel to get the vehicle up and running, following which the software subsystem tests algorithms and autonomy at the pool.

III-A. Mechanical Verification

To verify the waterproofing of the vehicle, it was left in the swimming pool at a depth of 2.5 meters overnight. However, waterproofing proved to be difficult due to multiple weld failures. To locate these leaks, we devised an efficient test in the lab by epoxying an air pump to a penetrator and filling the hull with water (Fig. 14). The pressure inside the hull was then increased, while being monitored by using an inverted pressure sensor, till until the water started to leak, isolating the source.

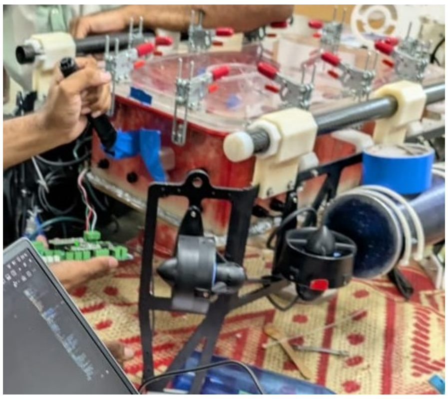


Fig. 14: In-Air Waterproofing

III-B. Hardware Reliability

In the week leading up to pool testing, the electrical stack and the actuators were systematically stress-tested thoroughly to ensure they do not burn up during operation. This was done by analyzing the toggle and PWM lines using an oscilloscope, calibrating the actuator positions to prevent motor stalls, and performing continuity checks on the electrical stack and the debug connector.

III-C. Autonomous Testing

With Matsya 7 waterproofed, and the electrical stack assembled on 23rd March, 2025, we had only three and a half months to test our software stack in the pool. Full fledged summer testing began at the end of April, wherein the vehicle underwent operational testing for upto eight hours a day, across two slots. Bug fixes, vehicle upgrades and in-air tests took place during nights.

This rigorous routine enabled us to complete pre-qualification as early as April 19, 2025, allowing us to promptly begin training for autonomous tasks.

1) *Software Calibration*

During testing, we found that relying solely on the DVL for position estimation resulted in a drift of around 15%. By combining the DVL's velocity data with orientation information from the IMU for coordinate transformation, we were able to reduce this drift significantly to between 0.5% and 2%, leading to much more reliable localization.

Additionally, our controller evaluations revealed that different navigation strategies worked best over different distances. Longer paths benefited from specific planning methods, while simultaneously controlling all degrees of freedom proved effective for fine, short-range movements. After tuning, the vehicle successfully reached target positions up to 20 meters away, enabling us to complete the pre-qualification run.

2) *Vision Detections*

Last year, our vision processing pipeline integrated OpenCV[8] and YOLOv8 as tasks were identified by simple colours. Since that is no longer the case, we're using YOLOv8 throughout this year, as it has always proved extremely reliable (Fig. 15).

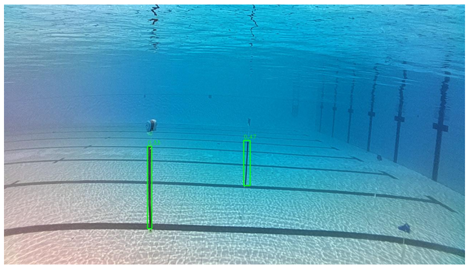


Fig. 15: Accurate Task Detection by Matsya 7

IV. Acknowledgements

We sincerely thank the IRCC of IIT Bombay for their vital administrative and financial support, which enabled our participation in RoboSub 2025. We also gratefully acknowledge the guidance of the Dean R&D's office in ensuring the success of our project.

We are deeply thankful for the generous support from our esteemed sponsors. Their contributions have been essential to the success of our project. We would like to offer special thanks to NCF Labs™ and Autodesk. Their support helped speed up prototyping the actuators, keeping us on track with our testing timeline.

References

[1] AUV-IITB. (2021). Technical Design Report of Matsya 6A.

[2] AUV-IITB. (2022). Technical Design Report of Matsya 6B.

[3] AUV-IITB. (2023). Technical Design Report of Matsya 6C.

[4] AUV-IITB. (2024). Technical Design Report of Matsya 6D.

[5] S. Gul. et al. (2021). "Advanced materials and sensors in well logging, drilling, and completion," in *Sustainable Materials for Oil and Gas Applications*, vol. 1, pp. 93–123.

[6] Jocher, G. et al. (2023). YOLOv8. GitHub repository. https://github.com/ultralytics/ultralytics

[7] Knapp, C. and Carter, G. (1976). The generalized correlation method for estimation of time delay. IEEE Transactions on Acoustics, Speech, and Signal Processing, 24(4), pp.320-327.

[8] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

Appendix A
Component Specifications

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost USD	Year of Purchase
Buoyancy Control	Designed In-House	Dead Weights & Foam	-	Custom	100	2025
Frame	Designed In-House	Aluminium	4 kg	Custom	200	2025
Waterproof Housing	Designed In-House	Aluminium Hulls w/ Acrylic Endcap	3 Hulls weighing 22 kg Depth Rating : 50 m	Custom	800	2025
Waterproof Connectors	Wetlink	Penetrators	24 penetrators of various wire diameters	Purchased	500	2025
Propulsion	Blue Robotics	T200	11 and 9.5 kgf forward and backwards	Purchased	1800	2024
Motor Control	Blue Robotics	Basic ESC	30A PWM controlled brushless motor speed controller	Purchased	320	2023
High Level Control	Designed In-house	GUI	Vehicle logs, sensor toggles and ROV control mode	Custom	-	2025
Current sensor	Allegro Microsystems	ACS712	5A current sensor	Purchased	3	2024
Status LED	Sparkfun	WS2812	8 neopixel circular LED strip	Purchased	1	2024
Pressure Sensor	Blue Robotics	Bar30	300m depth	Purchased	85	2024
Actuators	BlueTrail Engineering	SER-2000 casing	Tower Pro MG995 5V, 9.4kg/cm of torque, 140 degrees range	Purchased	830	2024
Battery	GenX	LiPo Battery	4 Cell and 16000mAh	Purchased	400	2025
Convertor	Mini-Box	M4ATX	High efficiency 250W output, $< 1.25mA$ standby current	Purchased	90	2024

Regulator	Mini-Box	M4ATX	High efficiency 250W output, $< 1.25mA$ standby current	Purchased	90	2024
CPU	AMD	Ryzen 5 5500	6 Cores (4.2GHz), 8GB RAM	Purchased	400	2024
Internal Comm Network	Raspberry Pi	RPi Pico Microcontroller	-	Purchased	10	2025
External Comm Interface	Designed In-House	Ethernet	1 Gb/s	Custom	10	2025
Inertial Measurement Unit (IMU)	Adafruit	BNO055	3 axis accelerometer, gyroscope and magnetometer	Purchased	50	2025
Doppler Velocity Log (DVL)	Waterlinked	A50	1Mhz Doppler shift	Purchased	7900 (Sponsored)	2022
Front Camera	Luxonis	Oak D	Stereo camera with inbuilt AI inference	Purchased	400	2025
Bottom Camera	Luxonis	Oak 1	12MP AI camera	Purchased	200	2024
Hydrophones	Teledyne	RESON Underwater TC 4013	-	Purchased	13000	2019
DAQ	National Instruments	NI-9227	4 channel 500M samples per second	Purchased	500	2024
Algorithms: Vision	YOLO v8	-	Parallel and Sequential processing, Lens Formula	-	-	-
Algorithms: Acoustics	FFTW	Time Difference of Arrival	Filtering in Frequency & Time Domain	-	-	-
Algorithms: Localization and Mapping	Orocos BFL	Extended Kalman Filter	EKF applied on position found by integration of DVL velocity	-	-	-

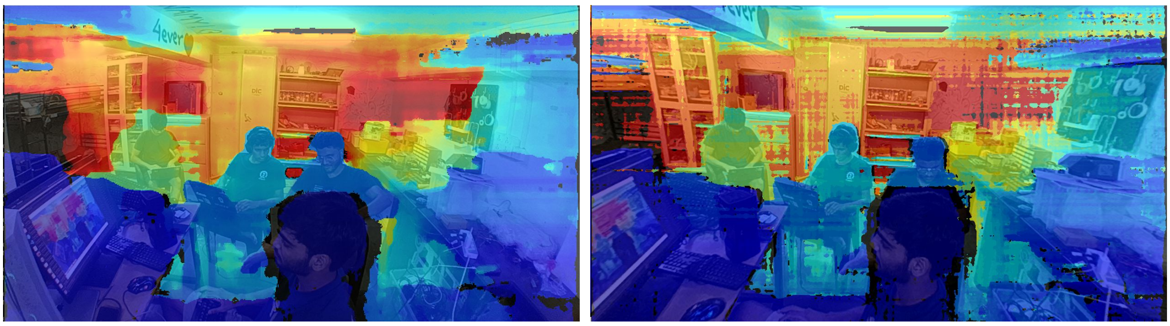
Algorithms: Autonomy	-	State Machine & Mission Planner	Probabilistic (or Finite) state machine for mission planner, designed in-house	-	-	-
Open Source Software	OpenCV, Eigen, ROS, YOLO v8	-	-	-	-	-

Appendix B
Stereovision

Depth was initially extracted using the median of all values within the bounding box, but this failed in cases like gates and holes where central pixels didn't belong to the object. Clustering techniques (e.g., DBSCAN) were tested but slowed the pipeline significantly. Using percentiles of the depth array (e.g., 20%) required object-specific tuning, which was unreliable with mixed-object detections. A method using size priors to constrain acceptable depth ranges was also tested but was dependent on the accuracy of the prior.

The current method applies a hue-based mask within the bounding box to isolate the object before computing depth. This improves reliability but still requires per-object color parameters.

A critical issue was the mismatch between depth maps and image frames. Depth distortion increased toward the image edges (Fig. 16a) due to differences in FOV between the stereo pair and the central RGB camera. This was corrected (Fig. 16b) using `WarpMeshSource` in the calibration node.



(a) Distorted depth map

(b) Corrected depth map

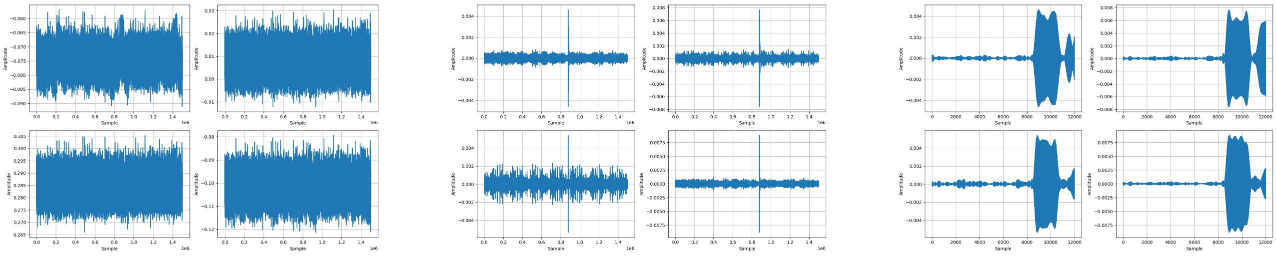
Fig. 16: Depth distortion before and after calibration with WarpMesh

Underwater, the OAK-D consistently underestimated depth. The cause was the change in effective focal length underwater, which affects depth computed from disparity. Scaling the depth by a factor of 1.33 (refractive index of water) corrected this error.

Appendix C
Acoustics

From the data we have gathered, we've found that the acoustic pings last for approximately 2-4ms. The goal of our acoustic pipeline is therefore to detect the initial rise of the ping across four hydrophones and compute the time difference of arrival (TDoA) between them. The rise occurs over only 200μs, while the rest of the ping is subject to multipath reflections, leading to unreliable phase information over longer distances. To mitigate this, all four hydrophones are mounted within a 6cm $\times$ 6cm square, ensuring that the maximum time difference between any two hydrophones is less than 60μs. This ensures minimal phase distortion within the useful part of the signal.

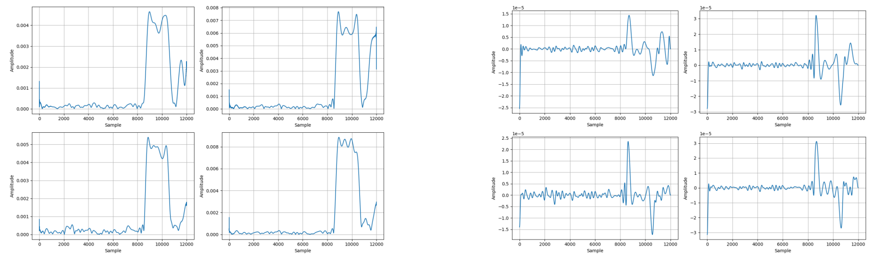
The raw signal is first passed through a bandpass filter to isolate the frequency range of interest. We then zoom into a small window of samples around the detected peak in one of the filtered signals to isolate the ping.



(a) Raw Data (b) Bandpassed (c) Narrowed

Fig. 17: Signal processing: Raw data to narrowed.

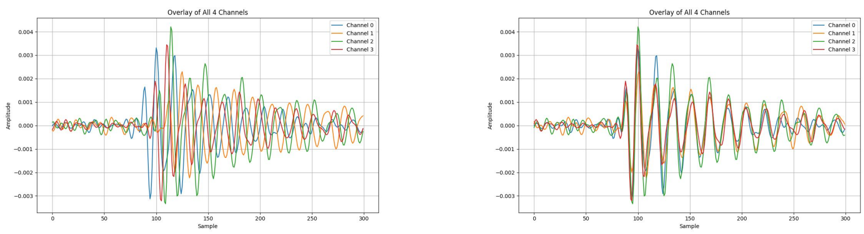
To precisely identify the start of the ping, the envelope of the narrowed signal is extracted. The derivative of this envelope often shows a sharp peak corresponding to the ping's rise.



(a) Envelope (b) Envelope derivative

Fig. 18: Envelope extraction and its derivative.

To improve alignment of the signals across hydrophones, we model each signal as a probability distribution by normalizing and shifting them. We then compute pairwise Kullback–Leibler (KL) divergences and apply temporal shifts to minimize total divergence. This method works reliably on tightly cropped signals but degrades when applied to larger time windows.



(a) Overlay of rise (b) Shifted overlay of rise

Fig. 19: Overlay of signals before and after alignment.

We explored multiple methods for detecting the rise of the ping:

1) The peak of the derivative of the envelope is theoretically aligned with the ping's onset. This method performs well on clean signals but is prone to false positives in noisy or distorted data.
2) An alternative is to detect prominent peaks in the derivatives of all four signals, and identify the set of peaks that are spatially closest. We then zoom into the window showing the best alignment.
3) A more robust approach involves element-wise multiplication of the signal derivatives. The resulting signal amplifies consistent rises across hydrophones, even when individual signals lack distinct peaks. This method shows promising results.

Appendix D
Testing Workflow

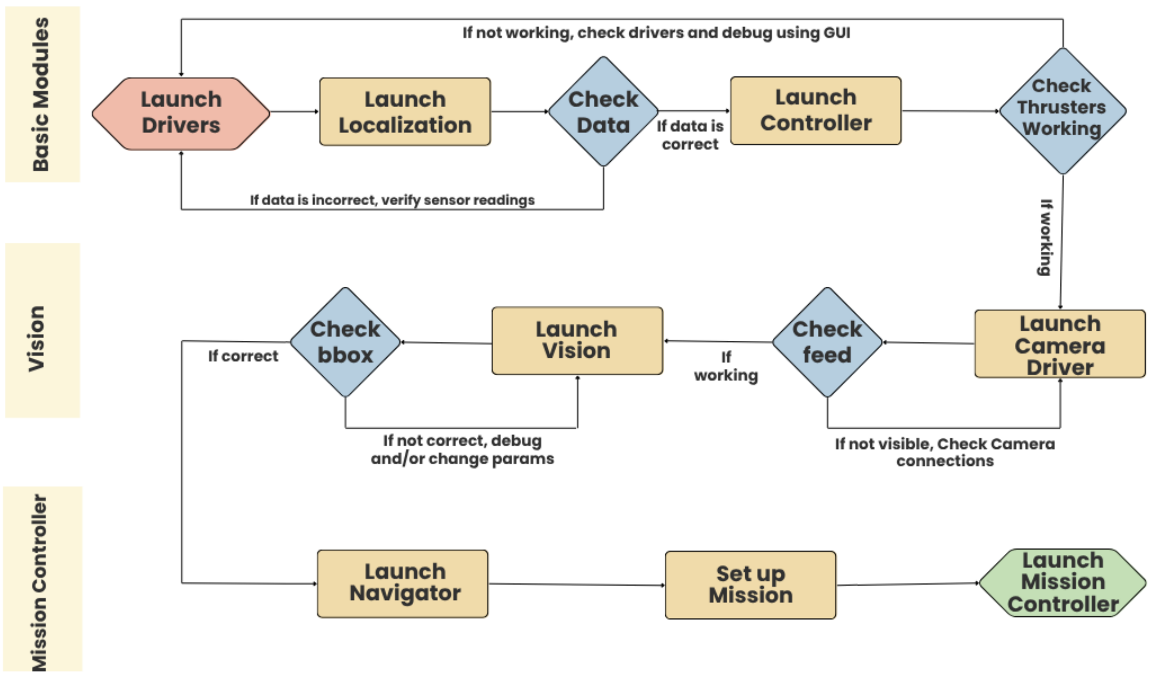


Fig. 20: Flow of Software Testing

The process begins by connecting Matsya's battery, powering on the onboard SBC, and establishing an SSH connection via the tethered laptop. Once connected, sensor drivers (excluding the camera) are launched, and ROS2 topics are checked to confirm the published data. If valid, the localization module is launched. The thrusters are then un-killed, and the controller is launched. If the AUV behaves erratically, thruster mappings, power factors, and PID constants are tuned accordingly.

For the vision stack, camera drivers are initialized and their feed is checked. If functional, the vision package is launched, and bounding boxes (bbox) are verified and tuned if needed. After launching the navigator,the mission is set up, and finally the mission controller is launched to start it. The entire flow ensures that all critical systems—drivers, localization, controller, navigator, vision, and mission controller—are validated and operational before full deployment.

The mission control has a modular design for clear logic flow. It comprises of three config files to modify behaviours independently - `mission.json`, `map.json` and `tasks.json`.

- The scripts have function calls in the following order: `main.py` → `perform_task.py` → `matsya.py`
- In `mission.json` the sequence of tasks is defined.

- The `map.json` stores absolute coordinates and orientations for each task location.
- Every task is described by a name, timeout period (seconds) and a list of navigator actions with its own set of subparameters in `tasks.json`.

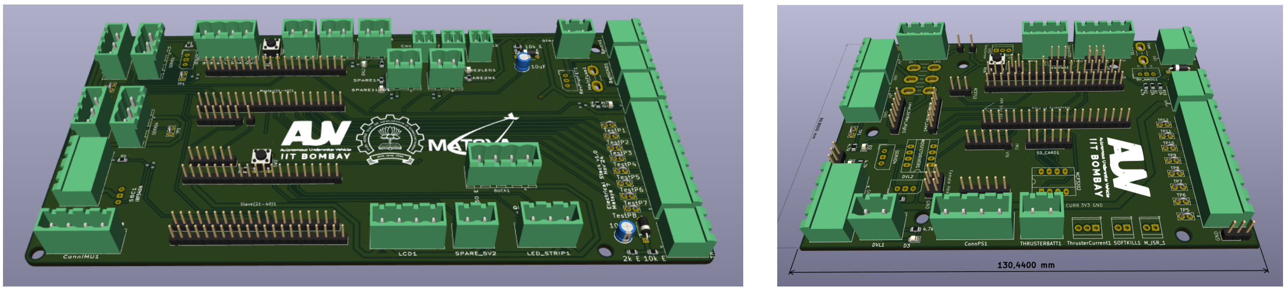
Sample for a task

`"Bin":`

- `"task_timeout": -1` → a task is aborted on exceeding the set time, mission control waits for all navigator actions to either exceed timeout or complete successfully.
- `"actions":`→ only actions described in navigator package are permissible.

Appendix E
Electrical Stack: Iterations

The electronics stack saw a complete redesign this year, culminating in two versions: the feature-rich dual-Pico V1 and the compact single-Pico V2. Both boards integrate reverse polarity protection, ports for the pressure sensor and M4ATX (powering the SBC), and voltage monitoring for the stack and thruster batteries using voltage dividers. Test pins on PWM lines proved invaluable for signal debugging using a DSO.



(a) V1 (Dual-Pico) (b) V2 (Single-Pico)

Fig. 21: Iterations of the Electrical Stack

V1 used inter-Pico communication, removing the need for extra ICs and improving replaceability and resilience. It had a real-time display showing battery voltages and temperatures from sensors near heat-sensitive parts like ESCs and cameras. This setup helped with system diagnostics during development.

V2 focused more on reliability and core features. It added Hall-effect sensors (WCS1700 and ACS712) to measure current, sending the data to the Pico via an external ADC over SPI. This allowed real-time current monitoring of servo-based actuators, helping detect stalls and trigger protective actions. Based on this, we added fuses and MOSFET cutoffs to protect the DVL and servos. An onboard SD card logged telemetry during autonomous runs, making post-run debugging easier.

Appendix F
Actuators: Iterations

Solidworks was used to design various iterations of the torpedo. The main objective was to develop a neutrally buoyant torpedo, with the centre of gravity near the centre of the body. The torpedoes were 3D-printed with different infills, testing for the perfect buoyancy. Due to layering during 3D-printing, water seeping into the cavities was also considered while designing the torpedo.

To keep track of the torpedoes during a rapid prototyping phase, informative nomenclature was introduced, providing information about the fin shape, infill percentage, and the material used. For example, SF77A would refer to a torpedo with straight fins, 77 percent infill, made of Acrylonitrile Butadiene Styrene (ABS).

Fig. 22: Different Torpedo Iterations

After all these iterations, we arrived at a sufficiently neutrally buoyant torpedo that achieved a long range of more than 2 feet.

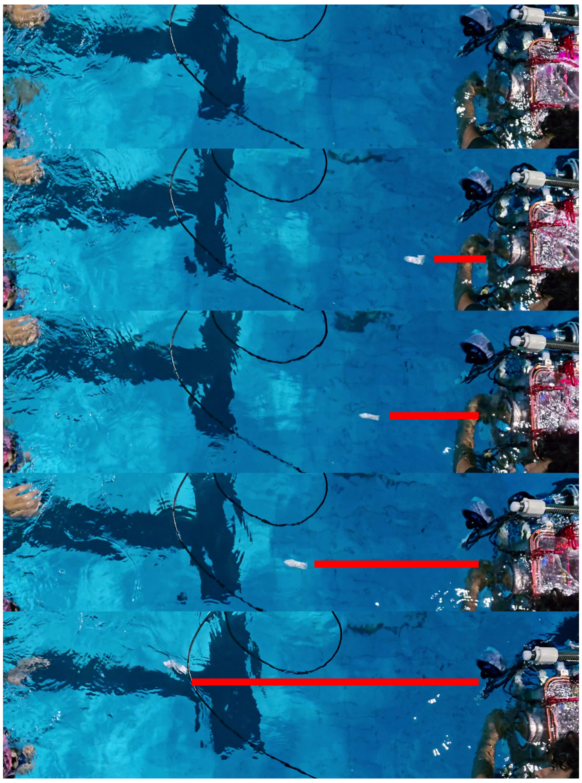


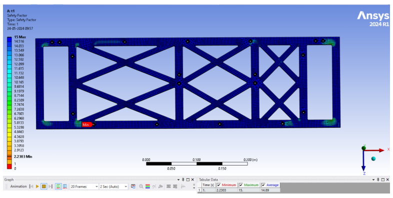
Fig. 23: Range of our Torpedo Shooter

Appendix G
ANSYS Structural Simulations

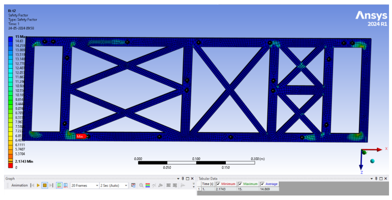
Matsya 7's frame, consisting of a mid frame and two leg frames inclined at 45°, was structurally analyzed and optimized using ANSYS Static Structural. The initial design was a rectangular plate with truss patterns, which was subjected to static loads from the hulls, actuators, and dynamic loads from the thrusters.

Boundary conditions were applied by fixing the sides of the frame, and a coarse mesh was initially used. The mesh was then refined using inflation layers with spheres of influence near holes and stress concentration zones. Finally, a mesh convergence study of five iterations was performed to ensure the accuracy of the solution.

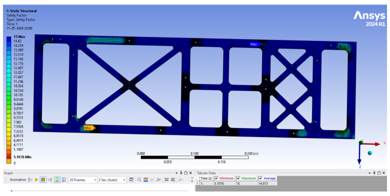
In each subsequent version of the frame, stress results were used to remove underloaded elements such as horizontal members. The truss angles were adjusted to improve the load paths, while maintaining a minimum safety factor of 2.5.



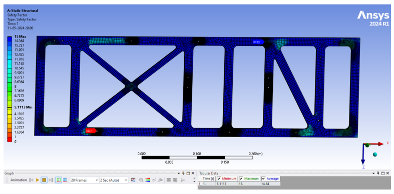
Iteration 1



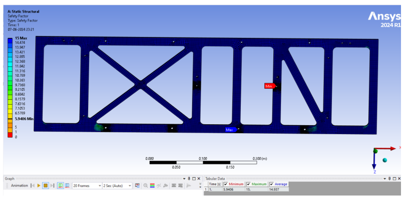
Iteration 2



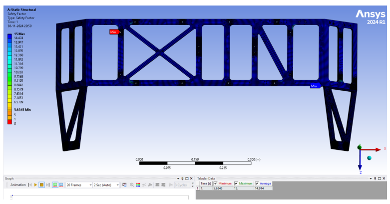
Iteration 3



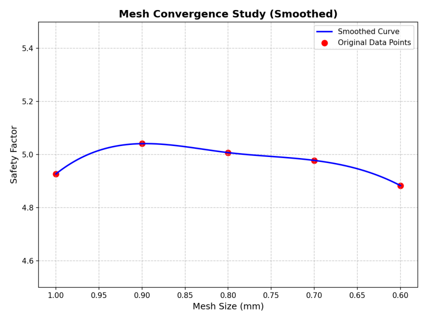
Iteration 4



Iteration 5



Final Optimized Frame

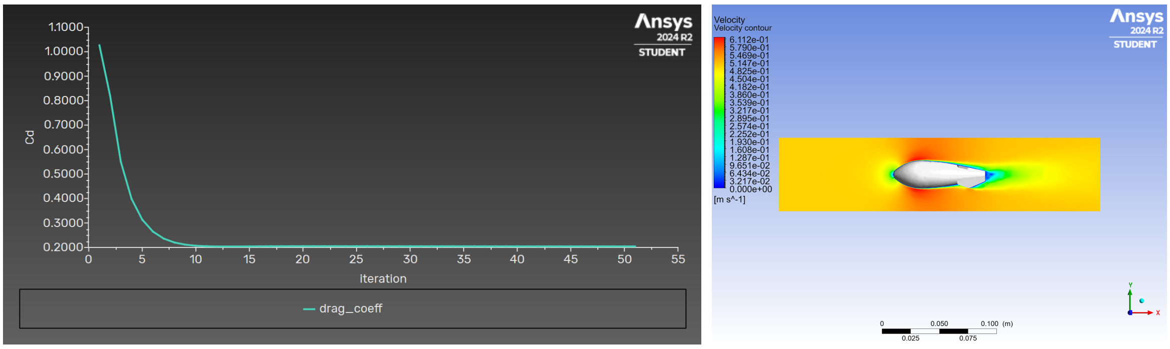


Mesh Convergence Study

The final design achieved a load capacity of 140kg with a component weight of only 2kg. This frame supports Matsya 7 (38kg) effectively, using two such structures on either side, ensuring a sturdy balance with the least amount of material.

Appendix H
CFD Simulations

ANSYS Fluent was used to simulate water flow around torpedo designs, allowing us to visualize drag sources and optimize shapes virtually. This enabled rapid testing and precise drag measurement. The new design achieves about 10% lower drag than its predecessor.



(a) c_d Convergence (b) Torpedo Velocity Contour

Fig. 24: ANSYS Simulation Results

Appendix I
Outreach Activities

The AUV-IITB Team actively engages with the community through various workshops and exhibitions each year. These events serve as platforms to inspire young students from schools and high schools to explore the field of robotics. During these exhibits, the team showcases the functionality of their AUV, followed by a detailed seminar and a Q&A session to enhance students' understanding and interest in AUVs and robotics.

Fig. 25: Displaying the vehicle to students at an exhibition

Technology Awareness Session: Hiware Bazar Rural Immersion Camp

As part of our commitment to outreach and knowledge dissemination, members of Team AUV, IIT Bombay participated in a Rural Immersion Camp. The camp was held in Hiware Bazar, Ahmednagar, Maharashtra. During the two-day camp, we conducted a Technology Awareness Session for primary school students in the village. This interactive session aimed to introduce young learners to the field of underwater robotics and the work undertaken. Through engaging demonstrations and simplified explanations, we showcased how autonomous underwater vehicles operate, the challenges they address, and the importance of robotics in marine research and exploration. The event served as an inspiring platform to spark curiosity about science, engineering, and technology among children in rural India.

Fig. 26: Technology outreach by Team AUV at Hiware Bazar

WISE: Empowering Future Women in Science and Engineering

The Women in Science and Engineering (WISE) program invites around 200 high school girls to IIT Bombay's campus every year for a few days of immersive exposure to cutting-edge technology and scientific exploration. As part of this initiative, AUV-IITB had the pleasure of presenting our autonomous underwater vehicle (AUV) to these aspiring young scientists, wherein we offered these bright young minds a glimpse into the exciting world of underwater robotics, encouraging them to see themselves as future leaders in science and engineering. Through WISE, we're proud to play a role in inspiring and empowering the next generation of female innovators.

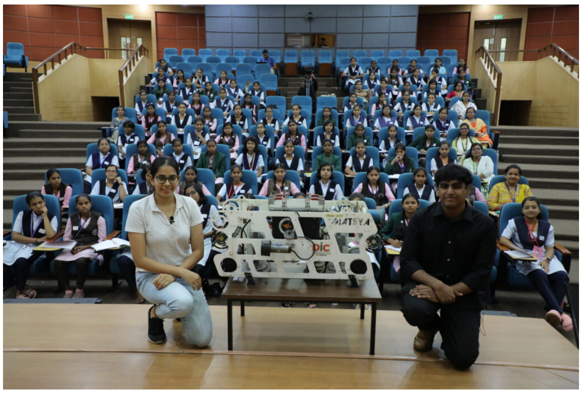


Fig. 27: An interactive session with the WISE community

Autodesk

Before decommissioning Matsya 6, we had the privilege of presenting the vehicle to the CEO of Autodesk. As part of this engagement, we conducted a study on handle design using Autodesk's Generative Design tool, successfully optimizing for both weight and ergonomics to improve handling. This collaboration exemplified the practical application of cutting-edge design tools in real-world engineering.

Fig. 28: Display at Autodesk

Ansys

During a visit by ANSYS executives to IIT Bombay, we demonstrated Matsya 7 performing a fully autonomous task in the pool. The session highlighted how advanced simulation tools are applied even at the student engineering level. It's a great pleasure to us that student-led engineering efforts like ours often serve as touchpoints between academic research and industry practice, where ideas flow in both directions and practical problem-solving meets cutting-edge technology.

Fig. 29: Presentation to ANSYS executives

TechConnect 2024

TechConnect is an annual technical exposition at IIT Bombay, held during the winter, where research labs and technical teams of IIT Bombay present their work to the general public, from professors to school children. Matsya 6, having just returned from RoboSub 2024, we had the pleasure of presenting it to Dr. Samir V. Kamat, Chairperson of the Defense Research and Development Organisation (DRDO).

Fig. 30: AUV-IITB Info Booth

The Matsya101 Instagram Series

We launched the Matsya101 Instagram series as a creative outreach initiative aimed at making our work more accessible and engaging. This series was designed to break down complex concepts related to our AUV, Matsya, into simple, easy-to-understand explanations. We used fun visuals and approachable language, and we were able to communicate the functionality and significance of components in an educational yet entertaining manner. Matsya101 highlighted the technical aspects of our AUV and served as a valuable tool to spark curiosity and interest in underwater robotics among students and tech enthusiasts alike.

Market Buzz - A Hull Cleaning Case Challenge

The Business Subdivision of Team AUV-IITB participated in 'Market Buzz' – A Hull Cleaning Case Challenge, organised during Techfest 2024 in collaboration with GAC India. The competition was focused on developing sustainable underwater hull cleaning solutions using ROVs at major Indian ports, which aligned with India's vision for greener maritime operations. Our team carried out a thorough feasibility study of using ROVs to clean the hulls in Indian port environments, taking into account important factors like economic viability, port safety procedures, and environmental regulations. We put forth a scalable business and deployment model with a focus on operational effectiveness, cost-effectiveness, and sustainability. The team received a significant cash prize and second place among national finalists for our strategic solution. This accomplishment demonstrates our dedication to using technical expertise to solve practical problems.