Team BengalBoat

RoboBoat 2026: Technical Design Report of BijoyTori 1.0, An Autonomous Surface Vehicle

An Nafew, Rubaiyat H Rahman, Hasin Israque Chowdhury Taha, Sudipto Mondal, Md Masrul Khan, Khondokar Radwanur Rahman, Sirajum Munir, Ariyan Siddique, Md. Rifat Hossain, Md Toslim, Md Tanvir Rahman, Abeed Mahmood, Nafia Bashar Suhani, Sasmit Banik, Mohammed Mashrur Arefin Bhuiyan, Zarif Bin Salek, Samnun Siam, Pranchoy Tarafdar, Shababa Islam Prionti and Rasel Ahmed

***Abstract* — This Technical Design Report presents BijoyTori 1.0, hereafter referred to as BTASV (Bijoy Tori Autonomous Surface Vehicle), developed by Team BengalBoat for RoboBoat 2026. BengalBoat is proud to be the first team from Bangladesh to participate in the RoboBoat competition, representing the organization Tech Autocrats, marking a significant step toward global inclusion in autonomous maritime robotics. BTASV is a fully autonomous surface vehicle designed to perform navigation, perception, and mission tasks in dynamic marine environments. The autonomy stack is built on ROS 2, with extensive simulation and validation conducted using Gazebo and visualization through RViz to ensure reliable system integration prior to on-water testing. Environmental perception is achieved through LiDAR-RGB sensor fusion, utilizing an RPLIDAR A2M12 for 360-degree range sensing and an Intel RealSense RGB camera for visual detection and classification tasks. The vehicle is propelled by Blue Robotics T200 thrusters, providing efficient and maneuverable thrust, and powered by a custom-designed 5S3P lithium battery pack to meet endurance and power requirements. Emphasis is placed on modular design, scalability, and system reliability. Beyond competition goals, the project promotes STEM education and engagement, offering hands-on experience in robotics, autonomy, and marine engineering while representing Bangladesh on an international platform.**

***Keywords* — Autonomous Surface Vehicle, RoboBoat 2026, BTASV, LiDAR-RGB Sensor Fusion, RPLIDAR A2M12, Intel RealSense, ROS 2, Gazebo Simulation, RViz Visualization, Autonomous Navigation, Marine Robotics.**

I. Competition Strategy

This season, our primary objective is to develop an autonomous surface vehicle [1] called BTASV capable of performing advanced mission tasks with a high degree of consistency and reliability. Rather than attempting to cover the full breadth of competition challenges at once, we adopted a focused and incremental development strategy, emphasizing a strong and dependable autonomy foundation. Guided by this approach, our efforts this year are centered on achieving consistent performance in Task 1 (Navigation Channel), Task 2 (Avoid the Crowds), and Task 3 (Snack Run). These tasks collectively stress precise navigation, perception, and control in dynamic environments, making them ideal benchmarks for validating and refining our software and hardware stack. By prioritizing these core and advanced challenges, we aim to establish a stable and repeatable system behavior before expanding to more mechanically and operationally complex tasks in future iterations.

A. Course Approach

The course focuses on end-to-end autonomous surface vehicle development through integrated sensing, perception, and localization. A RPLIDAR A2M12 [2] provides 360-degree range data for obstacle detection, while an Intel RealSense D455 [3] RGB-D camera enables visual perception and object detection [4]. The autonomy stack is built on ROS 2 with Python, Behavior Tree mission planning [5], supporting modular sensor integration and real-time communication. Visual processing combines OpenCV-based pipelines with YOLO8 [6] models for detection of buoys and obstacles. Vehicle state estimation and localization are achieved using an Extended Kalman Filter (EKF) [7], fusing onboard sensors with centimeter-level RTK-GPS data from a ZED-F9P dual-antenna system corrected by a stationary base station.

B. Navigation Channel and Avoid the Crowds

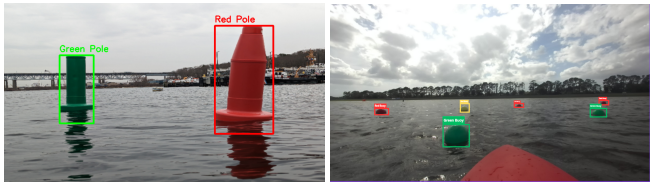


Fig. 1. Red, green, and yellow buoy detection using Yolov8.

This task is implemented using a tightly coupled vision-LiDAR perception pipeline. Buoys and gate markers are detected in RGB images from the Intel RealSense D455 using a YOLOv8 object detection model trained on competition-specific classes. The corresponding depth stream is used to estimate object range and filter false positives based on valid depth returns. Precise planar positions of buoys are computed using RPLIDAR A2M12 data. LiDAR point clouds [8] are clustered to extract buoy candidates, and red-green buoy pairs are geometrically associated to form gate hypotheses. Vision-based classifications are fused with LiDAR-derived positions to reliably identify gate elements and distinguish them from intermittent obstacle buoys [9]. The navigation module computes the gate midpoint in the vehicle frame and generates a collision-free trajectory that passes through the channel while avoiding crowd buoys. This sensor fusion approach ensures robust gate traversal and obstacle avoidance in dense and dynamic environments.

C. Find a Seat at the Show (Docking)

Docking is addressed using vision-based bay identification and precise short-range localization, although this task is assigned lower priority compared to core navigation missions [10]. The Intel RealSense D455 RGB stream is processed using a YOLOv8 model to detect docking bays based on the assigned shape and color. Depth measurements are used to estimate the distance to the dock face, while LiDAR data supports lateral alignment and

clearance estimation. Once the correct bay is identified, a target pose is defined at the bay centroid, and a low-speed closed-loop controller executes the docking maneuver. Given time and resource constraints, this task is considered secondary and may receive limited tuning.

D. Snack Run

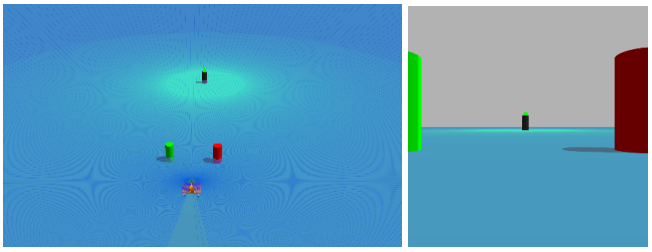


Fig. 2. Snack Run Mission Execution using Gazebo Simulation.

The Snack Run task is executed using a perception-driven, time-optimized navigation strategy. Entry gate buoys are detected and classified using a YOLOv8 model applied to the RealSense RGB stream, while LiDAR-based clustering provides accurate relative positions. After gate traversal, the mark buoy is continuously tracked, and a curvature-constrained trajectory is generated using a circular path-following controller to maintain a fixed standoff radius. Motion planning emphasizes minimum-time execution under dynamic constraints, with real-time obstacle checking and feedback control ensuring stable and repeatable maneuvering before exiting through the same gate [11].

E. Skeeball Game

The Skeeball task is considered a low-priority objective and is not a primary competition target; however, a partial implementation is planned if time permits. The shooting gate is detected using YOLOv8-based visual classification, with depth data used to estimate firing distance. Vehicle alignment is achieved through heading stabilization and lateral error minimization using LiDAR feedback. Ball delivery is performed via a mechanically actuated launcher system, with sequential firing controlled through a timed or state-based actuation sequence while maintaining station-keeping [12].

F. Water Blast

Similar to the Skeeball task, Water Blast is treated as a lower-priority mission and may be implemented opportunistically. The target face is detected using RGB-based object classification [13], and depth sensing is used to estimate target range and vertical offset. The BTASV performs station-keeping using feedback from LiDAR and state estimation while actuating an onboard water pump and nozzle system. Water discharge is sustained until the target indicator reaches the required threshold, subject to stability and alignment constraints.

G. Return to Home

The return-to-home task is executed using GPS-based waypoint navigation fused with onboard state estimation. The BTASV plans a global path back to the dock while continuously avoiding obstacles detected by LiDAR and vision. A final approach controller ensures the vehicle comes to a full stop within the required distance of the dock, completing the mission autonomously.

II. Design Strategy

The design strategy of BijoyTori 1.0 was developed to directly address the RoboBoat competition's emphasis on stable autonomous navigation, mechanical robustness, and reliable subsystem integration in an outdoor marine environment. From the earliest design stages, the mechanical architecture was treated as a mission-enabling system rather than a passive structure, ensuring that hull configuration, material selection, structural layout, and environmental isolation collectively support autonomous perception, propulsion, and control. This system-level approach allows the vessel to maintain consistent performance across all competition tasks while remaining resilient to environmental disturbances such as waves, wind, and vibration.

A. Hulls

The hull system of BTASV 1.0 forms the primary hydrodynamic interface and structural foundation of the Autonomous Surface Vehicle. A dual-hull catamaran configuration was selected to enhance transverse stability, reduce roll motion, and maintain consistent trim during autonomous navigation tasks. Each hull was initially developed using CAD-based modeling to define precise curvature profiles, internal volume distribution, and buoyancy zones, ensuring predictable hydrodynamic behavior under varying payload conditions. The physical construction process begins with the application of a resin sealing layer over the FDM-printed PLA structure to reduce surface porosity and improve inter-layer bonding, as shown in Fig. 3 (a). This is followed by the application of three successive layers of fiberglass reinforcement, which significantly enhance structural stiffness, impact resistance, and fatigue durability, as illustrated in Fig. 3 (b). A final plaster finishing layer is then applied to eliminate surface irregularities, achieve a uniform exterior finish, and reduce hydrodynamic drag, resulting in a mechanically robust hull, as shown in Fig. 3 (c).

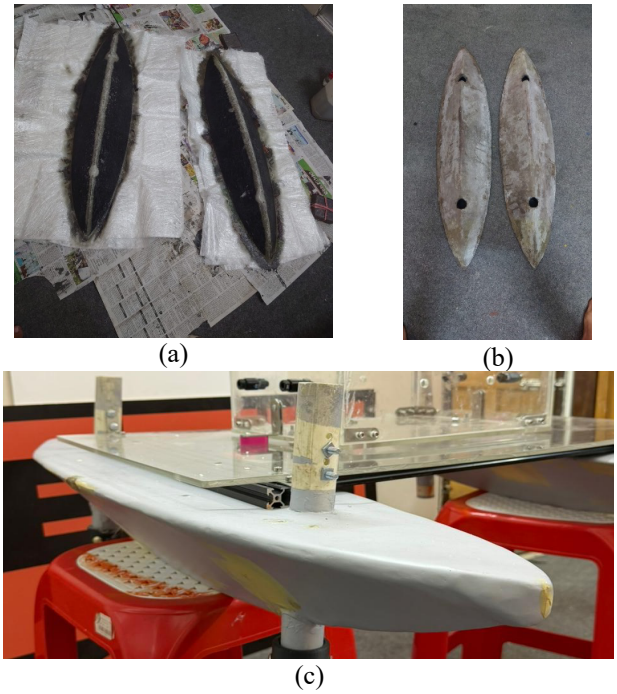


Fig. 3. Hull fabrication stages: (a) resin sealing over 3D-printed hull, (b) hull after application of three fiberglass-resin layers, and (c) final hull after plaster finishing.

B. Structural Frame and Deck Platform

The structural frame and deck platform provide the primary load-bearing and subsystem integration backbone of the BTASV. The dual hulls are mechanically interconnected using an aluminum v-slot extrusion frame to maintain rigid hull alignment while distributing mechanical loads generated by propulsion, payloads, and environmental forces. Aluminum was selected for its high stiffness-to-weight ratio, corrosion resistance, and dimensional stability in marine environments. The frame geometry was developed through CAD modeling to ensure uniform load transfer and minimal torsional deformation during dynamic maneuvers, as illustrated in Fig. 4. Mounted above the frame is a deck platform fabricated from 5-millimeter-thick acrylic sheet, which serves as the universal mounting interface for propulsion units, sensors, and onboard electronics. The acrylic deck provides sufficient structural rigidity while offering electrical insulation and visual accessibility for inspection and maintenance. The integrated frame and deck assembly, shown in Fig. 2, forms a modular and robust structural foundation that supports stable autonomous operation and facilitates efficient system integration and future upgrades.

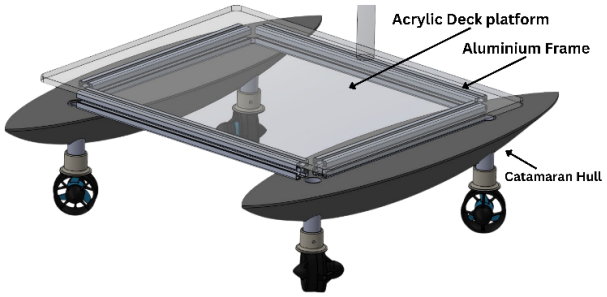


Fig. 4. Structural integration of the ASV showing CAD model of the aluminum frame with hull mounting interfaces and assembled acrylic deck platform mounted on the dual-hull structure.

C. Vision Setup Mount

The vision mount of BTASV 1.0 provides a stable, adjustable platform for onboard cameras, ensuring optimal alignment for autonomous navigation and target detection.

Fig. 5. Vision system mounting assembly showing CAD model of the adjustable 3D-printed and PVC-based camera mount, which has adjustable height and camera angle installed on the ASV.

It combines custom 3D-printed brackets with a lightweight PVC pipe frame, allowing quick fabrication and flexibility. The mount's height and angle can be adjusted to optimize the camera's field of view, reduce occlusion, and maintain consistent visual references. Once set, it offers sufficient rigidity to minimize vibration while remaining adaptable for future sensor upgrades. The CAD design and physical implementation are shown in Fig. 5.

D. Propulsion System

The propulsion of BTASV 1.0 is provided by four T200 thrusters, each mounted at a 45° angle relative to the hull, enabling precise maneuvering in both surge and sway directions. This angled configuration allows the ASV to generate combined thrust vectors for forward, lateral, and rotational motions, improving agility and control during navigation. The thrusters are secured to the hull using custom mounts that ensure stability and minimize vibration. The arrangement and orientation of the T200 thrusters are illustrate Fig. 6, highlighting their contribution to the vehicle's omnidirectional movement capabilities.

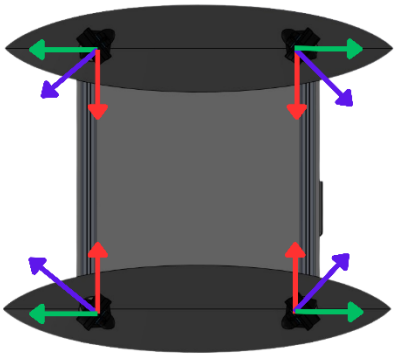


Fig. 6. Arrangement of Four T200 Thrusters Mounted at 45° Angles for Omnidirectional Propulsion.

E. Mountings and Assembly

The structural components, sensors, and propulsion units are securely mounted using a combination of custom brackets, 3D-printed fixtures, nut-bolts, and fasteners, ensuring stability and precise alignment across the platform. The modular assembly approach allows each subsystem, vision, propulsion, and electronics to be easily installed, adjusted, or upgraded without affecting the overall structural integrity.

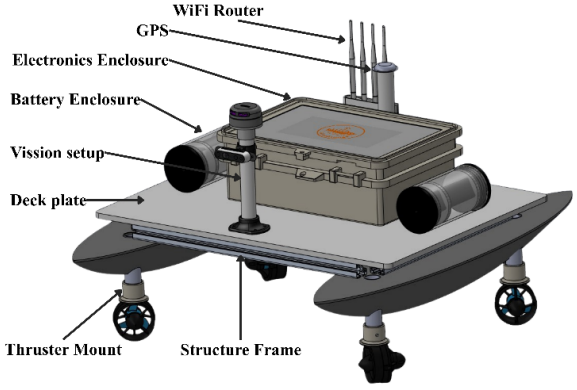


Fig. 7. Full CAD Model Showing Mountings and Assembly of BTASV.

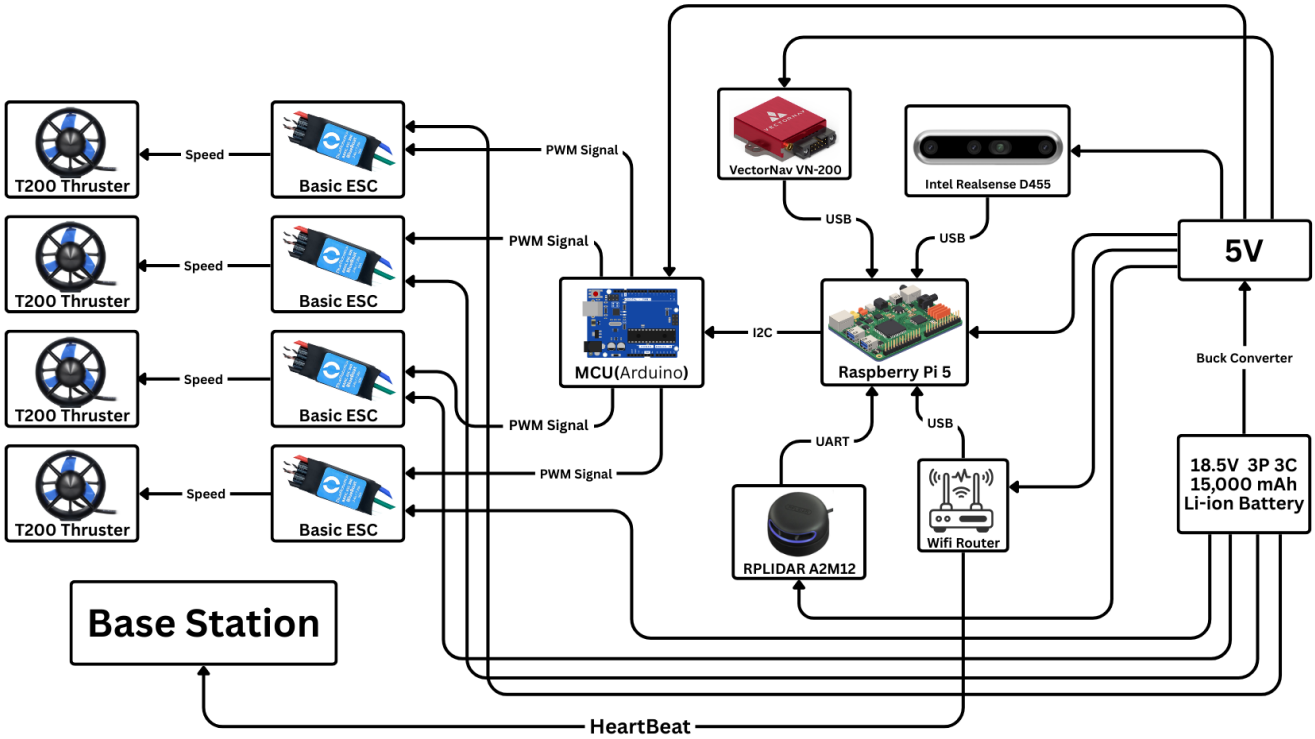


Fig. 8. Overall electrical system design architecture.

This design facilitates rapid prototyping and maintenance while maintaining a compact and balanced layout on the hull. The complete assembly, showing the integration of all subsystems, is illustrated in the full CAD model in Fig. 7.

F. Electrical System

The electronics subsystem of the BTASV manages power, sensing, and control. A custom made 5S3P battery powers the thrusters directly, along with computing and sensors module through regulated converters. The Raspberry Pi handles vision and mission planning, along with the communication. The arduino controls four T200 thrusters and actuators via PWM signals.

a) ***Overall Electrical System:*** The electrical architecture of BTASV follows a modular design that separates high-level autonomy, low-level actuation, sensing, and power management, as shown in Fig. 8. A Raspberry Pi 5 serves as the primary onboard computer running the ROS 2 autonomy stack and interfacing with all sensors. Perception is enabled through an RPLIDAR A2M12 (UART) for 360° ranging, an Intel RealSense D455 (USB) for RGB vision, and a VectorNav VN-200 IMU (USB) for state estimation. Low-level motor control is handled by an Arduino microcontroller, which receives motion commands via I²C and outputs PWM signals to four ESCs driving Blue Robotics T200 thrusters, ensuring reliable real-time propulsion control. Power is supplied by a custom 5S3P 18.5 V, 15,000 mAh lithium-ion battery pack. A buck converter provides a regulated 5 V rail for computing, sensors, and communications. Wireless connectivity and system monitoring are supported through an onboard Wi-Fi router, providing a robust and scalable electrical foundation for RoboBoat 2026.

b) ***Power Distribution System:*** BTASV is powered by a high-voltage 18.5 V lithium-ion battery pack (5S3P, 15,000 mAh), selected to provide sufficient energy capacity and peak current capability for sustained autonomous operation. The battery pack directly supplies the high-power propulsion bus, which feeds four ESCs that drive the four T200 thrusters, ensuring minimal voltage drop and efficient thrust delivery. To support onboard computing and sensors, the system includes a dedicated DC-DC buck converter that steps the 18.5 V battery voltage down to a regulated 5 V low-voltage rail. This 5 V rail powers the Raspberry Pi, MCU logic, sensors, and communication hardware, electrically separating sensitive electronics from propulsion transients and reducing noise-related instability. This distribution strategy provides a simple, robust power architecture with a clear separation between high-current motor loads and low-voltage control/perception subsystems.

1. ***High Voltage Power Calculation***: Each of the 4 T200 thrusters is driven by an ESC, operating at 18.5 V (5S) with a maximum current of 12 A under load.

$$I_{max} = 4 \times 12A = 48A$$

For continuous mission estimates (60% utilization):

$$I_{cont} = 0.6 \times 48A = 28.8A$$

Total power:

$$P_{thrusters} = V \times Icont = 18.5V \times 28.8A = 532.8W$$

2. ***Low Voltage Power Calculation***: The LVD supplies the Raspberry Pi, sensors, and actuators. Estimated consumption are depicted in Table 1:

Table I: Low voltage component power

Component	Power (W)
Raspberry Pi 4B	8
Arduino/MCU	2
VN200 IMU	1.5
Leak Sensor	0.5

Cameras	10
Lider	10
Total (approx.)	**32W**

c) ***Communication System:*** Fig. 9 illustrates the ESP32-based communication module PCB developed for BTASV. The left image shows the 3D rendered view of the assembled board, highlighting the compact layout, surface-mounted ESP32 module, and multiple JST connectors used for power, UART, GPIO, and peripheral interfaces. The right image presents the PCB layout view, detailing the component placement, signal routing, ground planes, and antenna keep-out zone. The design emphasizes reliable wireless communication, compactness, and ease of integration with the vehicle's electrical system. Proper separation between RF, logic, and power traces is maintained to reduce noise and improve signal integrity, while mounting holes at the corners enable secure installation within the electronics enclosure.

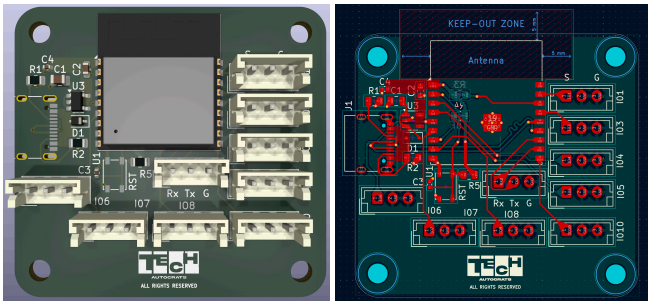


Fig. 9: ESP32 based communication module PCB.

d) ***Software System Architecture:*** The USV software stack is implemented as a modular, ROS 2 based layered architecture, designed for real-time operation, scalability, and safe autonomous control. The system follows a five-layer vertical pipeline, progressing from raw sensor acquisition to hardware actuation, with well-defined data interfaces between layers which is shown in Fig. 10.

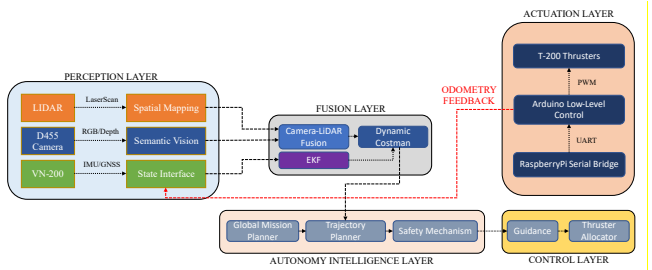


Fig. 10. High Level Overview of System Architecture.

- **Layer 1 (Perception Layer):** The perception layer acquires and preprocesses raw sensor data at 20-50 Hz. It includes: a Semantic Vision Node performing YOLOv8-based object detection (gates, buoys, obstacles), publishing Detection2DArray. A Depth Processing Node using the Intel RealSense D455, publishing filtered PointCloud2 data. A LiDAR Node (RPLIDAR A2M12) publishing filtered LaserScan messages. A VectorNav GNSS Node (VN-200) publishing IMU and NavSatFix data. This layer outputs raw, time-synchronized sensor observations.
- **Layer 2 (Fusion & State Estimation Layer):** Operating at 10-20 Hz, this layer fuses heterogeneous sensor data into a coherent world model:
  - This node fuses semantic detections from the vision system with spatial measurements from LiDAR to estimate 3D object poses, which are published as `PoseArray` messages.
  - An EKF fuses IMU, GNSS, and odometry data to estimate the vehicle's pose and velocity. The output is published as `nav_msgs/Odometry`, while maintaining a consistent TF tree (map → odom → base_link) for coordinate transformations.
  - Using fused object poses and state estimates, a multi-layer occupancy grid is generated and published as `nav_msgs/OccupancyGrid`, providing spatial awareness for navigation and collision avoidance.
- **Layer 3 (Planning & Decision Layer):** The planning layer converts state estimates into mission-level decisions at 5-10 Hz. A Global Planner manages waypoint-based goals, while a Local Planner (DWA/TEB) generates collision-free velocity commands. Mission sequencing and task transitions are handled by a Behavior Tree Executive, and a Safety Monitor continuously enforces geofencing and collision avoidance constraints. Outputs are published as `geometry_msgs/Twist` and PoseStamped.
- **Layer 4 (Motion Control Layer):** Operating at approximately 20 Hz, the motion control layer translates velocity commands into actuator-level thrust demands. A Guidance Controller (PID) computes control actions, a Thrust Allocator maps commands to port and starboard thrusters using differential drive kinematics, and a Disturbance Observer compensates for environmental disturbances. Control outputs are published as Float64MultiArray.
- **Layer 5 (Actuation Layer):** The actuation layer interfaces directly with propulsion hardware at 50 Hz. A Serial Bridge (Raspberry Pi) transmits motor commands over UART (115200 baud) to an Arduino controller, which generates PWM signals for BlueRobotics T200 thrusters in the 1000-2000 µs range

III. Testing Strategy

Before physical deployment, the system is validated in Gazebo, a high-fidelity 3D physics simulator, enabling a "Sim-to-Real" pipeline that reduces hardware risk and allows testing of edge-case scenarios which is shown in Fig. 11. The robot is modeled with accurate mass and friction coefficients to replicate real-world dynamics, and sensors are integrated via the Gazebo-ROS Bridge (gazebo_ros_pkgs), including a LiDAR plugin simulating the RPLidar A2M12 by publishing sensor_msgs/LaserScan at 10 Hz, and an RGB-D plugin simulating the Intel RealSense D455 to produce synchronized point clouds and aligned depth streams for the YOLO detection node. A custom Gazebo world incorporates static obstacles such as walls and furniture, as well as dynamic actors like moving pedestrians, to evaluate obstacle avoidance and object detection robustness. A multi-tier testing strategy ensures software reliability, combining unit tests for mathematical correctness such as depth-to-meter conversion, and integration tests that launch the full sensor suite while monitoring the /yolo/debug_image topic for accurate bounding box alignment. ROS 2 diagnostic tools including RQt Graph for node communication visualization, Rviz2 for

real-time 3D LiDAR and PointCloud2 visualization, and ros2 topic hz to verify stable high-bandwidth image streaming are employed for performance monitoring. The Sim-to-Real transition is seamless due to the modular architecture, where swapping the bijoy_tori_bringup launch parameters redirects subscribers from Gazebo plugins to physical drivers realsense2_camera and sllidar_ros2, ensuring that algorithms tested virtually execute identically on the physical Legion compute platform.

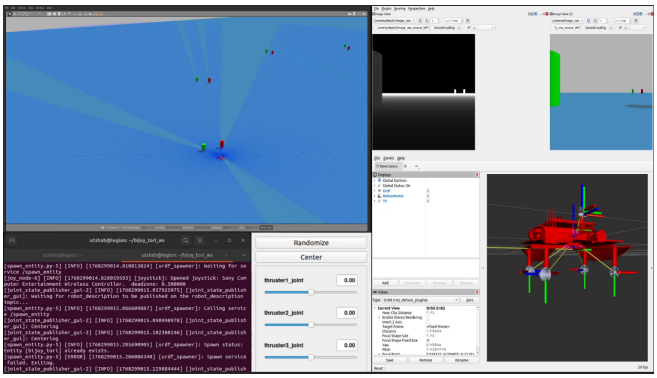


Fig. 11. ROS2 simulation in Gazeboo and Rviz.

IV. Acknowledgement

Team BengalBoat would like to express sincere gratitude to our sponsors for their invaluable support in making the development of BTASV and our participation in RoboBoat 2026 possible. We would like to especially acknowledge ICT Division, Government of Bangladesh for their significant support and encouragement toward innovation and student-led technological advancement. We also gratefully acknowledge Sony Smart Technology for their sponsorship and continued support. We extend our appreciation to our partners SolidWorks, Qatar Airways, Blue Robotics, and MATWORKS for providing essential tools, resources, and MATLAB licensing, which greatly contributed to our mechanical design, simulation, propulsion integration, and software development efforts. We would like to express our deepest gratitude to Md. Rasel Ahmed, who served as our Team Advisor, for his continuous guidance, technical insight, and leadership throughout the project. His support played a crucial role in shaping the team's direction and engineering decisions. Special thanks are also extended to our mentors Sudipto Mondal, Md Masrul Khan, Khondokar Radwanur Rahman, Md. Rifat Hossain and Md. Toslim, whose mentorship, design feedback, and hands-on guidance were instrumental in the growth and success of the team. Finally, we acknowledge the dedication and collaborative spirit of all BengalBoat members. As the first team from Bangladesh to compete in RoboBoat, we hope this project inspires future participation in STEM and autonomous maritime robotics.

References

[1] Z. Peng, D. Wang, Z. Chen, X. Hu and W. Lan, "Adaptive Dynamic Surface Control for Formations of Autonomous Surface Vehicles With Uncertain Dynamics," in IEEE Transactions on Control Systems Technology, vol. 21, no. 2, pp. 513-520, March 2013, doi: 10.1109/TCST.2011.2181513.

[2] https://static.generation-robots.com/media/pj2-ld310-slamtec-rplidar-datasheet-a2m12-v1-0-en-2874.pdf

[3] https://www.realsenseai.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf

[4] H. Liu et al., "Magneto-Inductive Magnetic Gradient Tensor System for Detection of Ferromagnetic Objects," in IEEE Magnetics Letters, vol. 11, pp. 1-5, 2020, Art no. 8101205, doi: 10.1109/LMAG.2020.2974178.

[5] Z. Wang, W. Gao, G. Li, Z. Wang and M. Gong, "Path Planning for Unmanned Aerial Vehicle via Off-Policy Reinforcement Learning With Enhanced Exploration," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 3, pp. 2625-2639, June 2024, doi: 10.1109/TETCI.2024.3369485.

[6] G. Jocher, A. Chaurasia, J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[7] S. Yamagishi and L. Jing, "The Extended Kalman Filter With Reduced Computation Time for Pedestrian Dead Reckoning," in IEEE Sensors Letters, vol. 7, no. 12, pp. 1-4, Dec. 2023, Art no. 6010404, doi: 10.1109/LSENS.2023.3331513.

[8] N. H. T. Nguyen, S. Perry, D. Bone, H. Le Thanh, M. Xu and T. T. Nguyen, "Combination of Images and Point Clouds in a Generative Adversarial Network for Upsampling Crack Point Clouds," in IEEE Access, vol. 10, pp. 67198-67209, 2022, doi: 10.1109/ACCESS.2022.3182697.

[9] C. Chen, J. -G. Lu, M. Wei, J. -C. Li, Q. -H. Zhang and H. -D. Wang, "Robust Real-Time Berthing Distance Estimation Based on Conical Field-of-View Search Using LiDAR-Based Dynamic Mapping," in IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-14, 2025, Art no. 8515414, doi: 10.1109/TIM.2025.3636659.

[10] C. Li, J. Sun, L. -W. Li, X. Wu and V. Palade, "An Effective Swarm Intelligence Optimization Algorithm for Flexible Ligand Docking," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 19, no. 5, pp. 2672-2684, 1 Sept.-Oct. 2022, doi: 10.1109/TCBB.2021.3103777.

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-Time Loop Closure in 2D LiDAR SLAM," Robotics and Automation (ICRA), 2016. GrabCad. Link: https://grabcad.com/library

[12] S. Song, T.-S. Kim, M. Sung and S.-C. Yu, "Attitude control of AUV using multiple buoyancy engines for exploration and water column profiling," in *Proc. IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, Porto, Portugal, 2018, pp. 1-5, doi: 10.1109/AUV.2018.8729827.

[13] Surface Water Dataset Mode. Link: https://universe.roboflow.com/roboboat-soe01/buoys-dkb8e

APPENDIX: A
COMPONENT SPECIFICATIONS

Component	Vendor	Model/ Type	Specifications	Custom/ Purchased	Year of Purchase	Cost
Buoyancy Control	Blue Robotics	Buoyancy Foam R- 3318	6 Pieces	Purchased	2025	$300
Frame	Local	Custom Aluminium with Acrylic Frame	Modular	Custom	2025	$270
Waterproof Housing	Local	Custom Acrylic Box	Modular	Custom	2025	$120
Hull	Local	Custom PLA print & Fibre-Resin	Dual Hull, able to float the robot in water	Custom	2025	$820
Waterproof Connectors	Local	Custom ABS	Ingress-Protected	Custom	2025	$20
Frame Anodizing	M/S Tanvir Enterprise	Gloss Black (39)	Reinforced, keeps the robot's all joint strong	Purchased	2025	$9
Thruster	Blue Robotics	T200 Thruster	Brushless,12-20 V, ~5 kgf thrust	Purchased	2025	$920
Motor Control	Blue Robotics	Basic ESC BR-100344	30 A bidirectional brushless speed controller	Purchased	2025	$160
LiDAR	Slamtech	A2M12	12-meter range, 10Hz rotational speed, & 16KHz sampling frequency	Purchased	2025	$350
Actuators	Robotics Bangladesh	IP54 DFRobot	Waterproof Metal Gear, 35kg, 360 degree	Purchased	2025	$32
Battery	EEE Tech		5S 3P 15,000 mAh	Custom	2025	$100
CPU	Raspberry PI Foundation	Raspberry PI 5 8gb	Quad CPU-core CPU, Enhanced PCIe GPU, multitasking applications	Purchased	2025	$120
Internal Comm Network	n/a	CAN, MAVLink Protocol	n/a	n/a	2024	n/a

External Comm Interface	Wi-Fi	TP-Link	802.11ac/Wi-Fi 5, 802.11ax/Wi-Fi 6,	Purchased	2025	$60
Programming Language 1	n/a	C++	n/a	n/a	2025	n/a
Programming Language 2	n/a	Python	n/a	n/a	2025	n/a
Compass and IMU	VectorNAV	VN200	GPS-aided IMU, 9 axis, high-precision	Purchased	2025	$3500
Camera 1	Luxonis	OAK-D-W	12MP RGB Camera Module with auto focus, Wide-FOV Stereo AI	Purchased	2024	$650
Camera 2	Intel	RealSense D455	Global -shutter stereo depth camera, ~6 m range	Purchased	2025	$625
Algorithms: Vision	n/a	YOLO V3	Real-time object Detection algorithm with high speed and good accuracy	n/a	2025	n/a
Algorithms: Acoustics	n/a	Time Difference of Arrival (TDoA)	n/a	n/a	2025	n/a
Algorithms: Localisation/ Mapping	n/a	ROS2	n/a	n/a	2025	n/a

APPENDIX: B
TOOLS AND EQUIPMENTS

1) Electrical Equipment

- Solder gun and stand
- Solder wire
- Wire strippers (standard and AWG precision)
- Ethernet cables and LAN spool
- HDMI cables
- Micro USB cables for screens
- Test screen
- Multimeter
- Spare, charged Li-Po batteries
- Battery chargers (Li-Po compatible)
- Extension cords and power strips
- Screwdrivers (various sizes)
- USB extenders and hubs
- Scissors
- USB ASP programmer
- Glue gun and glue sticks
- Arduino Uno boards with cables
- Spare ATmega and CAN ICs
- Duct tape and insulation tape
- Jumper wires
- Single-strand and AWG wires for harnessing
- Mouse and keyboard
- Heat shrink tubing and heat gun
- Crimping tools and connectors
- XT60, XT30 connectors
- Digital oscilloscope (for signal debugging)
- Bench power supply
- PDB test board
- USB-to-UART interfaces
- Hot air rework station (for PCB repairs)

2) Mechanical Equipment

- Allen key and spanner set
- Precision screwdrivers (PSD)
- Drill machine with assorted bits
- Epoxy resin, mixing cups, sticks
- Cardboard, tissue, chalk for layout work
- Duct tape, insulation tape
- Zip ties (various sizes)
- Rope for handling and lowering AUV
- Nuts, bolts, washers (stainless steel preferred)
- Buoyancy foam and floats
- Mini hacksaw and blade replacements
- Scissors and utility knife
- Calipers and measuring tape
- Sandpaper for edge finishing

- Threadlocker for critical fasteners
- Ballast weights and mounting straps

3) Testing and Integration Equipment

- Water tank or access to pool for testing
- Tether spool and slip ring if applicable
- Kill switch test harness
- Tether booster modules
- Laptop with ROS Noetic, MAVROS, and mission software
- Jetson Orin Nano and Raspberry Pi for integration testing
- Pixhawk Cube with accessories
- Hydrophone array with test pinger
- Cameras (OAK-D-W, low-light USB) for vision pipeline validation
- Spare thrusters and ESCs for quick swap during pool tests
- Digital scale for weight checks
- Leak detection sensors and vacuum pump for hull testing

APPENDIX: C
SYSTEM ARCHITECTURE, FLOWS AND DIAGRAMS

Electrical System Configuration

a) Propulsion and Actuation

The propulsion system of BTASV is designed to provide stable, efficient, and precise surface maneuvering during RoboBoat missions. The vehicle utilizes four Blue Robotics T200 thrusters, arranged to enable differential thrust control for forward motion, turning, and station keeping. Each thruster is driven by a dedicated electronic speed controller (ESC) and powered directly from the high-voltage propulsion bus at 18.5 V, supplied by the main battery pack. The ESCs receive PWM control signals from a low-level microcontroller, enabling smooth and responsive thrust modulation. Thruster placement is optimized to maintain balance around the vehicle's center of gravity, reducing yaw-induced torque and improving energy efficiency during navigation and docking tasks. The modular propulsion layout allows rapid replacement of individual thrusters or ESCs, minimizing downtime during testing and competition.

b) Microcontrollers and Computing Units

1. Low-Level Controller (MCU)

A dedicated microcontroller (Arduino) serves as the low-level control unit, responsible for deterministic real-time actuation. It receives motion commands from the high-level computer via I²C and converts them into precise PWM signals for the four ESCs. This separation ensures reliable propulsion control independent of high-level computation load.

2. High-Level Computer (Raspberry Pi 5)

The Raspberry Pi 5 functions as the primary onboard computer, running the ROS 2 autonomy stack. It handles mission logic, navigation, perception, and sensor fusion. High-level velocity and control commands are generated onboard and transmitted to the MCU for execution. The Raspberry Pi also manages system monitoring, logging, and communication with the base station.

c) Sensor and Camera Interfaces

BTASV employs a LiDAR-RGB perception system for navigation and obstacle avoidance. An RPLIDAR A2M12 connects via UART to provide 360° planar range data for mapping and collision avoidance. An Intel RealSense D455 RGB camera interfaces via USB and supports visual detection and color-based classification tasks. A VectorNav VN-200 IMU, connected via USB, provides inertial and orientation data for state estimation.

d) Communication Module

Wireless communication with the base station is achieved through an onboard Wi-Fi router, enabling real-time telemetry, debugging, and mission monitoring. A heartbeat-based link is implemented to support safety monitoring and system status feedback during operation.

e) Power and Sensor Interfaces

Power is supplied by a custom 5S3P lithium-ion battery pack rated at 18.5 V and 15,000 mAh. The high-voltage bus directly powers the ESCs and thrusters. A DC-DC buck converter steps down the battery

voltage to a regulated 5V rail, supplying the Raspberry Pi, MCU, sensors, and communication devices. This architecture ensures electrical isolation between high-current propulsion loads and sensitive low-voltage electronics, improving system reliability and safety.

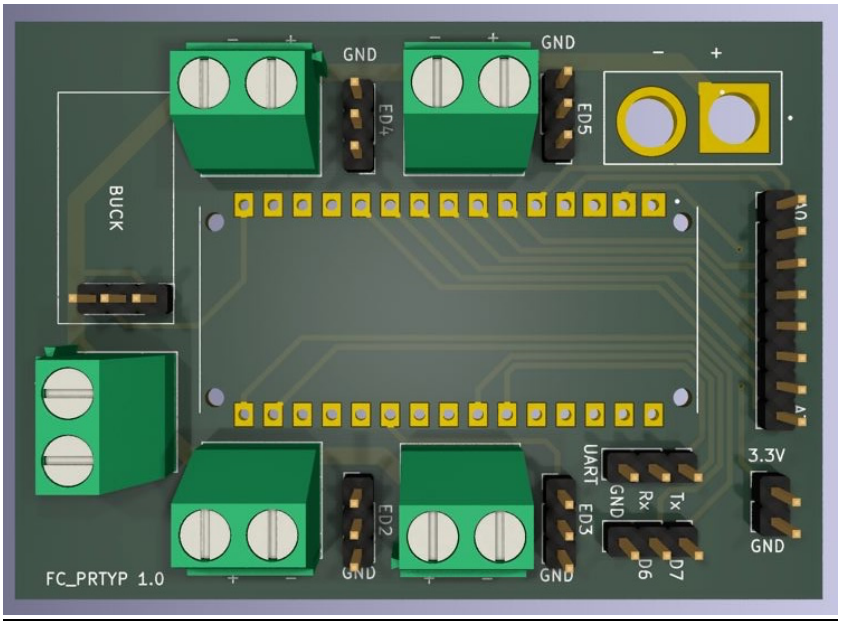


Fig. 12: Power Distribution Board (PDB) for BTASV showing wide copper pours and reinforced power planes on a 2 oz/ft²

The Power Distribution Board (PDB) of BTASV is designed to safely and efficiently distribute high-current power from the main battery pack to the propulsion and auxiliary subsystems which is shown in Fig. 12. The board is fabricated on a 1.6 mm FR-4 substrate with 2 oz/ft² copper, selected to support sustained high-current operation while maintaining mechanical rigidity and thermal stability in a marine environment. The PDB layout incorporates wide copper pours and reinforced power planes dedicated to the main positive supply and return paths, minimizing resistive losses and voltage drop during peak propulsion loads. Trace dimensions were designed following IPC-2152 current-carrying capacity guidelines, with conservative assumptions to ensure reliable operation under worst-case conditions. The design targets a maximum temperature rise below 20°C during continuous full-load operation, providing adequate thermal margin for extended missions. Each thruster power branch utilizes an effective copper width of approximately 12–15 mm, enabling a current-handling capability exceeding 10 A per channel. This exceeds the tested continuous current requirements of the Blue Robotics T200 thrusters, ensuring safe operation even during aggressive maneuvers or transient load spikes. Additional design considerations include symmetric current paths to reduce electromagnetic interference, short return loops to improve efficiency, and clearly labeled power outputs to simplify assembly and maintenance. This robust PDB design contributes to overall system reliability, scalability, and electrical safety during RoboBoat 2026 operations.

Fig. 13 illustrates the electrical power behavior of the vehicle under varying thrust conditions, showing battery voltage (blue), current draw (red), instantaneous power consumption (yellow), and average estimated power draw (green) over time. The supply voltage remains largely stable throughout the test, with only minor dips observed during periods of peak load, indicating adequate battery capacity and low internal resistance. Current and instantaneous power exhibit sharp spikes corresponding to rapid thrust changes, such as acceleration and aggressive maneuvering, while lower values are observed during steady-state operation. The average estimated power draw increases gradually over time, reflecting sustained propulsion demand during extended high-thrust operation. This analysis validates the robustness of the

power distribution system, supports battery sizing decisions, and demonstrates stable electrical performance under dynamic operating conditions relevant to autonomous mission profiles.

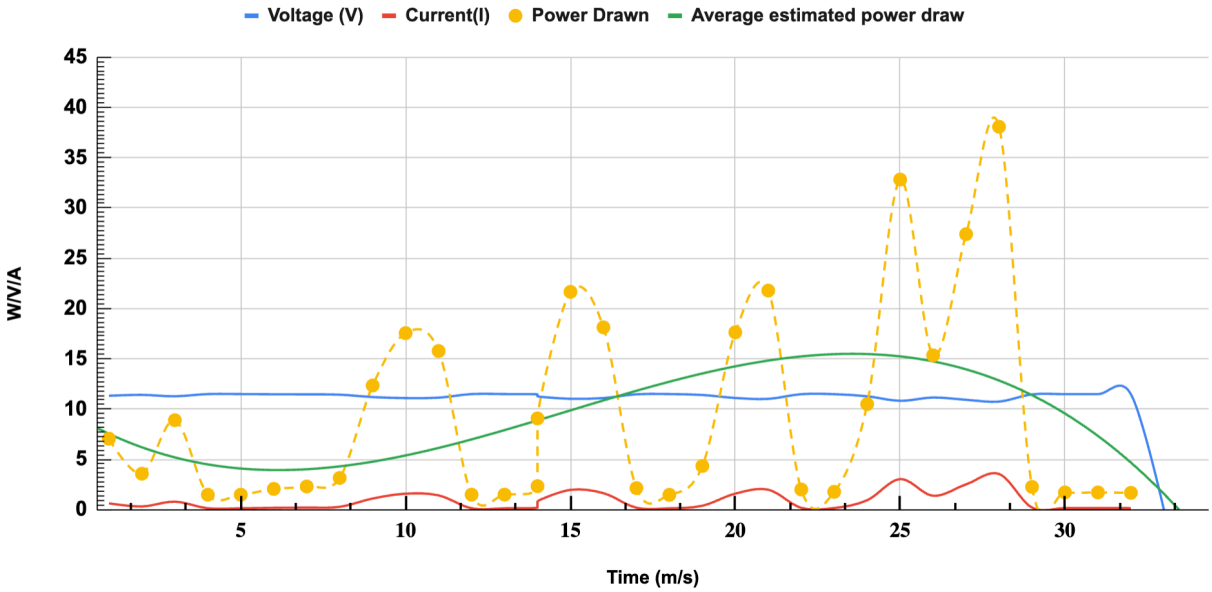


Fig. 13: Power draw curve under different thrust condition

Fig. 14 presents the results of repeated thrust testing conducted on the propulsion system to evaluate the performance and consistency of the thrusters. The horizontal axis represents successive test runs, while the vertical axis shows the measured thrust output in kilograms. The results indicate thrust values ranging approximately from 4 kg to 14 kg, reflecting variations due to changes in throttle input, test conditions, and transient load effects during operation.

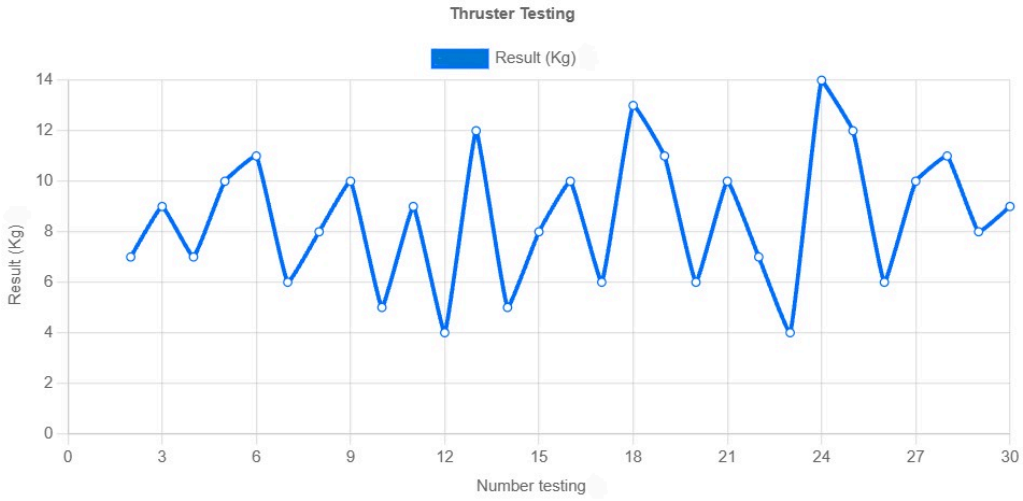


Fig. 14: Thruster test curve result

Higher thrust peaks correspond to maximum throttle commands, demonstrating the thrusters' ability to deliver sufficient force for aggressive maneuvers and rapid acceleration. Lower thrust values occur during reduced throttle or transitional states, highlighting the controllability and responsiveness of the propulsion system. Despite these variations, the overall trend shows repeatable performance across multiple test cycles, confirming reliable thrust generation and consistent ESC–motor response. This testing validates the suitability of the thrusters for autonomous navigation tasks such as station keeping, docking, and dynamic course traversal. The observed thrust range provides adequate margin for overcoming

environmental disturbances while maintaining stable and efficient operation during RoboBoat mission profiles.

Fig. 15 illustrates the battery capacity testing results under representative operating conditions, showing the battery voltage variation over time for different onboard loads. The black curve represents voltage drop primarily due to the T200 thrusters, the green curve corresponds to the Raspberry Pi, and the red curve represents other 5 V electronic components. The voltage decreases gradually from approximately 4.1 V to 3.7 V per cell over a 60-minute test period, indicating a controlled and predictable discharge profile.

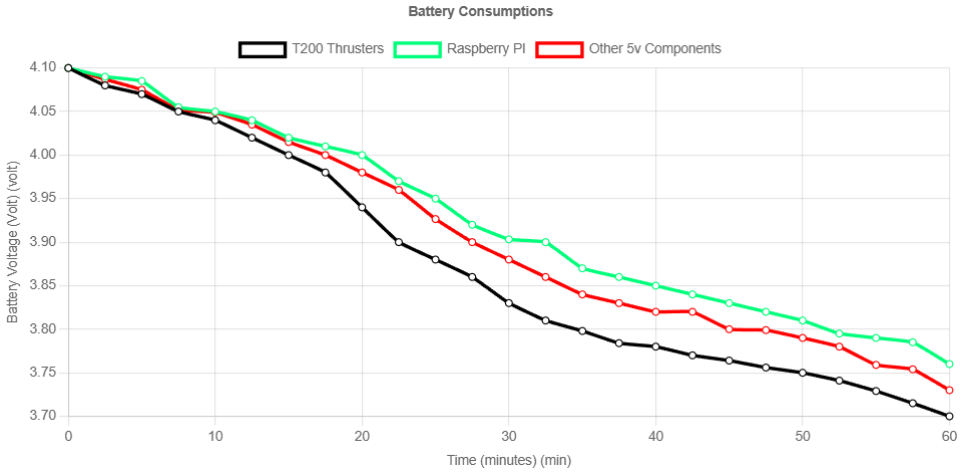


Fig. 15: Battery capacity testing data

The steeper voltage decline observed for the thruster load highlights the higher current demand of propulsion during operation, while the Raspberry Pi and other low-voltage electronics exhibit a more gradual voltage reduction, reflecting their comparatively lower and steadier power consumption. The consistent separation between the curves demonstrates effective power regulation and load distribution, with no sudden voltage collapses or instability observed during the test. This analysis confirms that the battery system provides sufficient capacity to support both propulsion and onboard electronics for extended mission durations. The results validate battery sizing, power management strategy, and overall energy efficiency, ensuring reliable autonomous operation throughout RoboBoat mission scenarios.

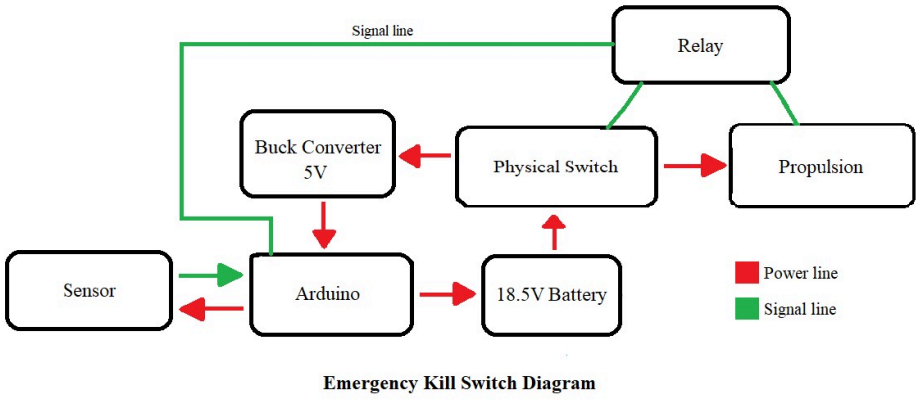


Fig. 16: Kill switch structure of the BTASV

Fig. 16 illustrates the emergency kill switch architecture implemented on BTASV, designed to ensure rapid and reliable shutdown of the propulsion system in the event of a fault or emergency. The diagram distinguishes between power lines (red) and signal lines (green), emphasizing the separation of high-current and low-voltage control paths for safety and robustness.

The 18.5 V main battery supplies power to both the propulsion system and the low-level control electronics. A physical kill switch is placed in series with the high-voltage power line, allowing immediate manual disconnection of propulsion power. When activated, the physical switch interrupts the battery supply and prevents further power delivery to the propulsion system. A relay, controlled via a low-voltage signal from the Arduino, provides an additional layer of electronic isolation. The Arduino receives regulated 5 V power from a buck converter and continuously monitors system state and sensor feedback. Under normal operation, the Arduino asserts the relay control signal, allowing propulsion power to pass. In the event of abnormal behavior, sensor fault detection, or loss of control signal, the Arduino deactivates the relay, cutting power to the propulsion system. This dual-layer kill switch design, combining a manual physical switch with a software-controlled relay, ensures compliance with competition safety requirements while providing both immediate human intervention and autonomous fault response, significantly enhancing operational safety during RoboBoat missions.

APPENDIX: D
DESIGN TO IMPLIMENTATION STRATEGY

Fig. 17 illustrates the structured workflow followed in the design and development of the mechanical subsystem of BTASV. The process begins with an analysis of handbooks and previous technical reports, allowing the team to leverage established design practices and lessons learned from earlier autonomous vehicle platforms. Based on this analysis, key design parameters such as structural constraints, weight limits, and functional requirements are selected, followed by the selection of suitable materials to ensure strength, durability, and compatibility with the marine environment.

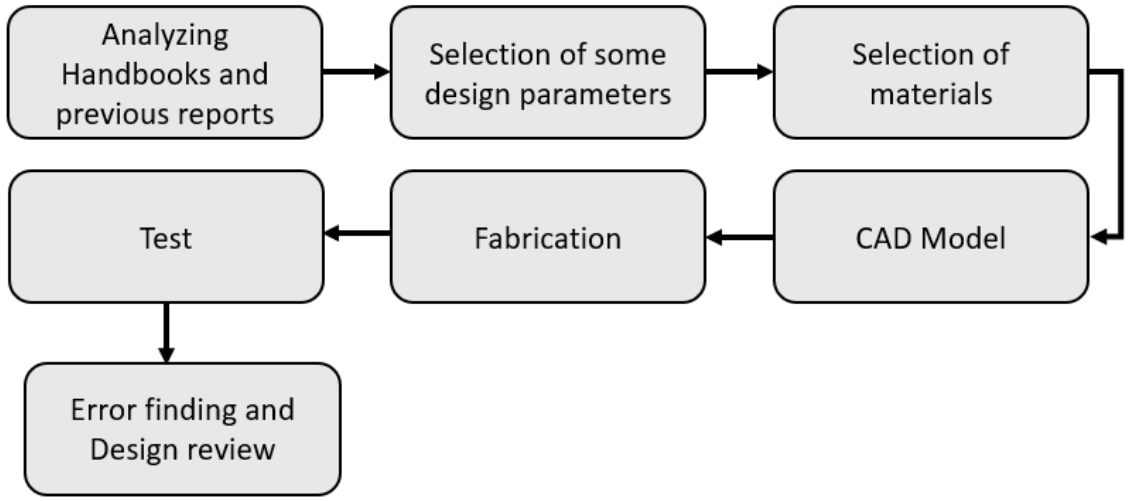


Fig. 17: Workflow diagram of the mechanical subsystem

These parameters are then translated into a detailed CAD model, which serves as the basis for mechanical integration and interference checking. Once the design is finalized, the components are fabricated using appropriate manufacturing techniques. The fabricated parts undergo testing to evaluate structural integrity, fit, and functional performance under operational conditions. Feedback from testing is used for error identification and design review, enabling iterative refinement of the mechanical design. This workflow ensures a systematic, repeatable, and optimization-driven approach, resulting in a robust and reliable mechanical subsystem suitable for RoboBoat mission requirements.

APPENDIX: E
SOFTWARE INTEGRATION

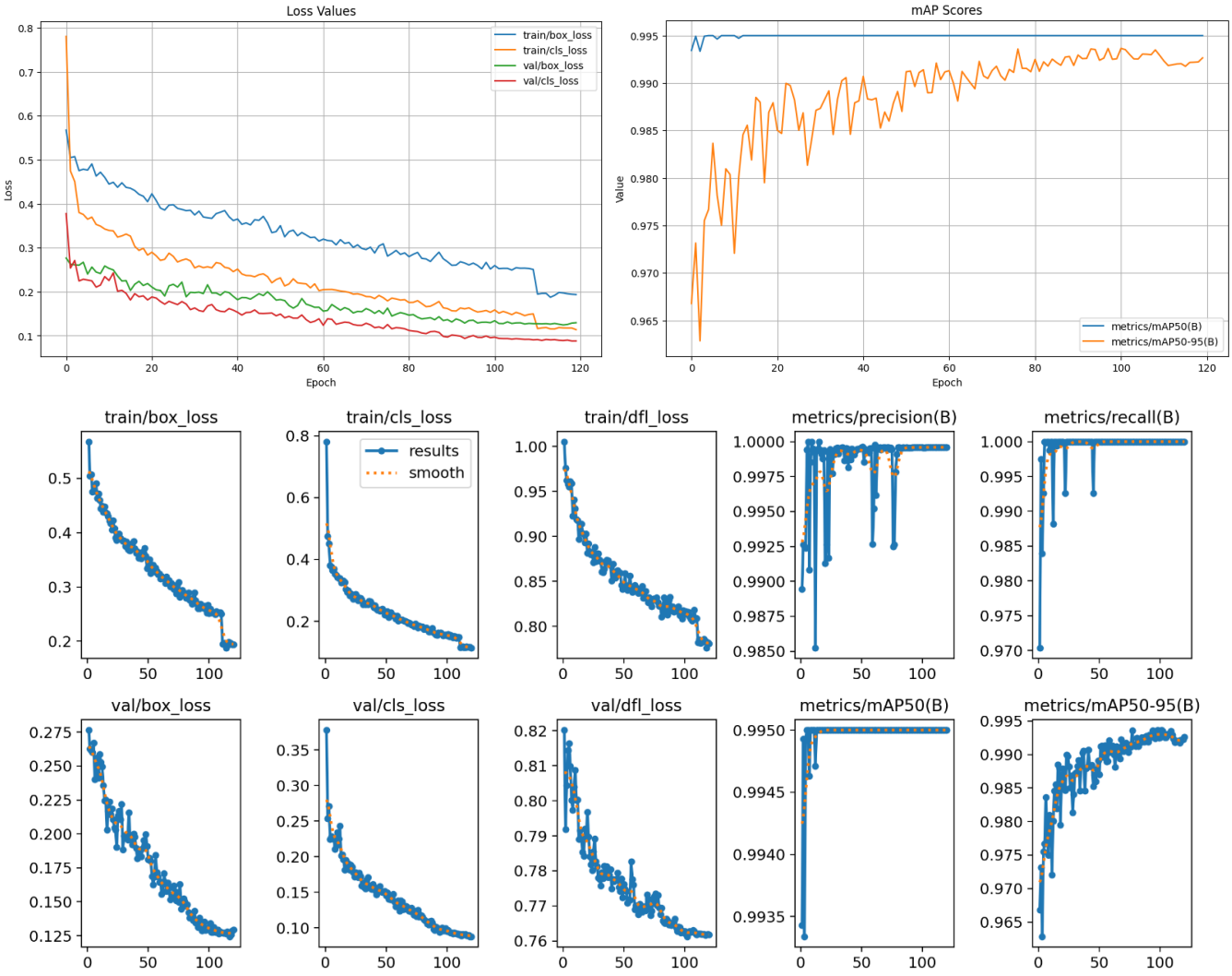


Fig. 18: YOLOv8x Training Metrics for BTASV

Our object detection pipeline for RoboBoat 2026 leverages a hybrid dataset comprising both publicly available RoboFlow maritime datasets and proprietary field-collected imagery captured under diverse environmental conditions specific to our competition venue. The annotation process was conducted using RoboFlow's cloud-based labeling platform, where we manually drew bounding boxes around three primary object classes: navigational gates (typically consisting of vertical poles or markers defining course waypoints), buoys (floating markers used for path planning and obstacle avoidance), and docking bay components (including dock structures, target zones, and mooring indicators). We selected YOLOv8x (extra-large variant) as our detection backbone due to its superior feature extraction capabilities through its enhanced C2f modules and SPPF spatial pyramid pooling layers, which provide robust multi-scale feature representation crucial for detecting objects at varying distances and under challenging maritime lighting conditions. The training regimen employed anchor-free detection with decoupled classification and regression heads, utilizing mosaic and mixup data augmentation strategies to improve model generalization across different water surface reflectivity patterns, weather conditions, and sun glare scenarios are illustrated in Fig. 18. Based on the training curves shown, our model achieved convergent loss values across bounding box regression (box_loss), classification (cls_loss), and distribution focal loss (dfl_loss) components, with validation metrics plateauing at approximately 0.995 mAP@50 and 0.993 mAP@50-95, indicating excellent localization precision and classification accuracy across IoU thresholds critical performance indicators for real-time autonomous navigation decision-making in the competition's GPS-denied obstacle course environment.

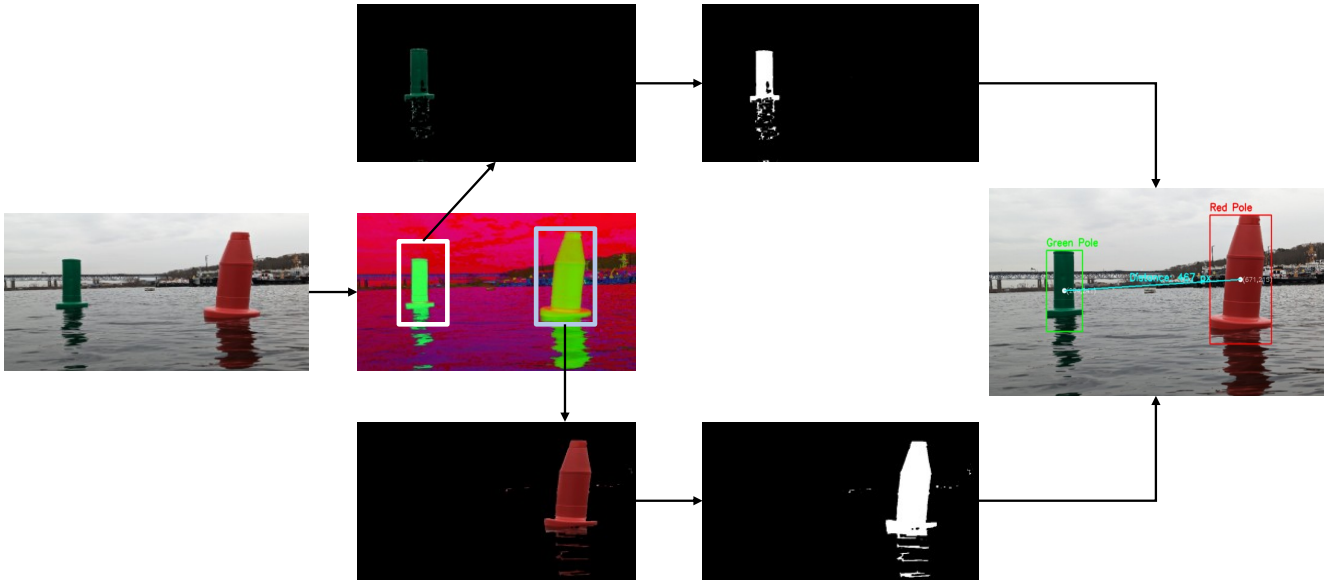


Fig. 19: Color-Based Buoy Detection Pipeline for Navigation Channel Task

Our Navigation Channel task implementation employs a color-based segmentation pipeline to identify and localize port (red) and starboard (green) channel markers for autonomous waypoint navigation are shown in Fig. 19. The processing workflow begins with the acquisition of RGB imagery from the onboard camera system, followed by HSV colorspace transformation to facilitate robust color thresholding under varying illumination conditions. Individual color masks are generated for green and red buoys using empirically determined hue, saturation, and value ranges that account for water surface reflections and ambient lighting variations. The segmented binary masks undergo morphological operations (erosion and dilation) to eliminate noise and consolidate fragmented regions, after which contour detection algorithms identify candidate buoy regions. For each detected marker, we compute the centroid coordinates in the image plane and apply Euclidean distance calculation between the green and red buoy centroids to estimate the channel width and relative positioning. This inter-buoy distance metric, combined with individual buoy positions relative to the image center, enables real-time heading corrections to maintain the vessel's trajectory along the channel centerline, ensuring safe passage between the lateral markers as required by RoboBoat competition rules.

Gate Heading Alignment Algorithm

Algorithm 1 Gate Heading Error Computation

$FrameCenterX \leftarrow \frac{ImageWidth}{2}$
$GateCenterX \leftarrow \frac{x_g + x_r}{2}$
$HeadingError \leftarrow GateCenterX - FrameCenterX$
if $HeadingError > 0$ **then**
    Turn RIGHT
else
    Turn LEFT
end if

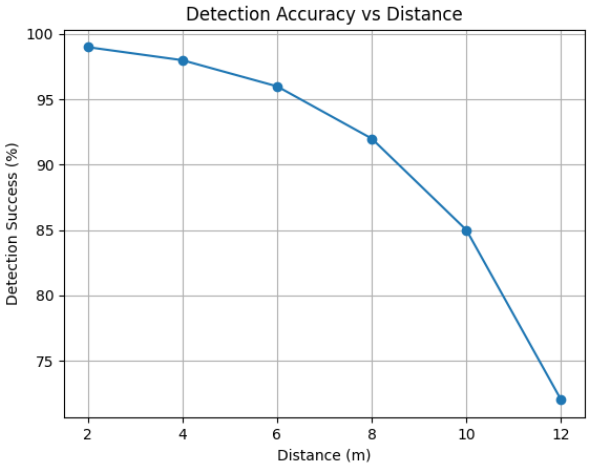


Fig. 20: Detection Accuracy vs Distance

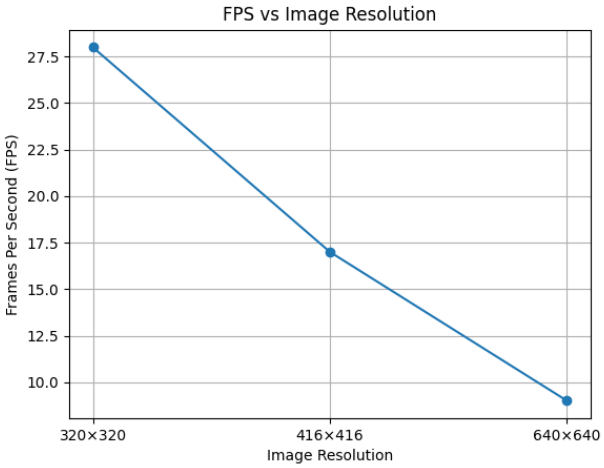
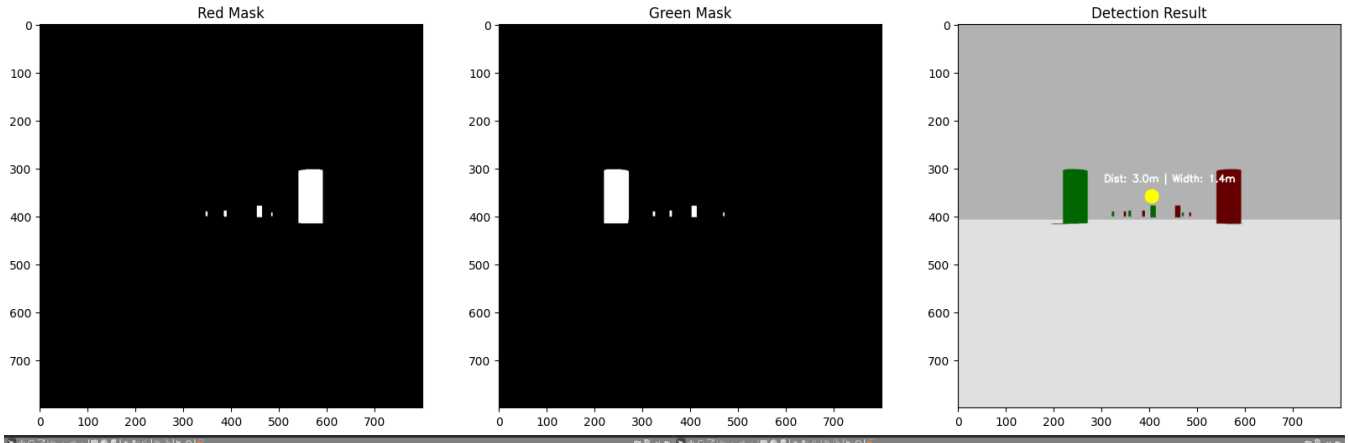


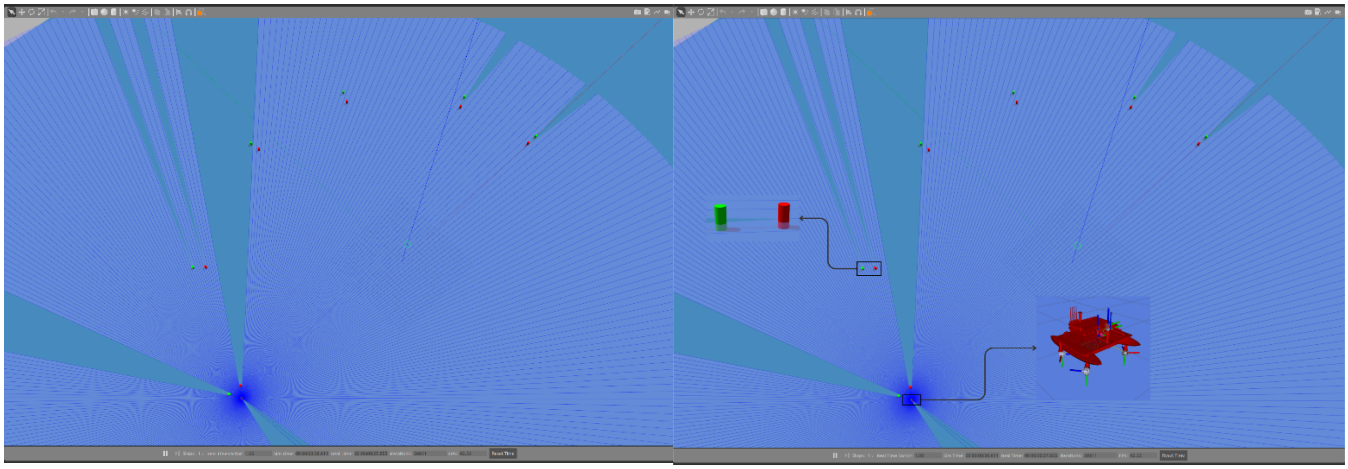
Fig. 21: FPS vs Image Resolution

Fig. 20 illustrates the variation of object detection success rate with respect to the distance between the camera and the target. The results show near-perfect detection performance at close and mid-range distances, with a gradual decline as distance increases due to reduced object scale and image resolution. Despite this degradation, the system maintains reliable detection within the operational range required for autonomous mission execution, demonstrating robustness under real-world conditions.

Fig. 21 presents the relationship between input image resolution and inference speed measured in frames per second (FPS). As resolution increases, FPS decreases due to higher computational load. The selected operating resolution provides a balance between detection accuracy and real-time performance, ensuring stable perception while meeting timing constraints of the autonomous control loop.

Simulation Results





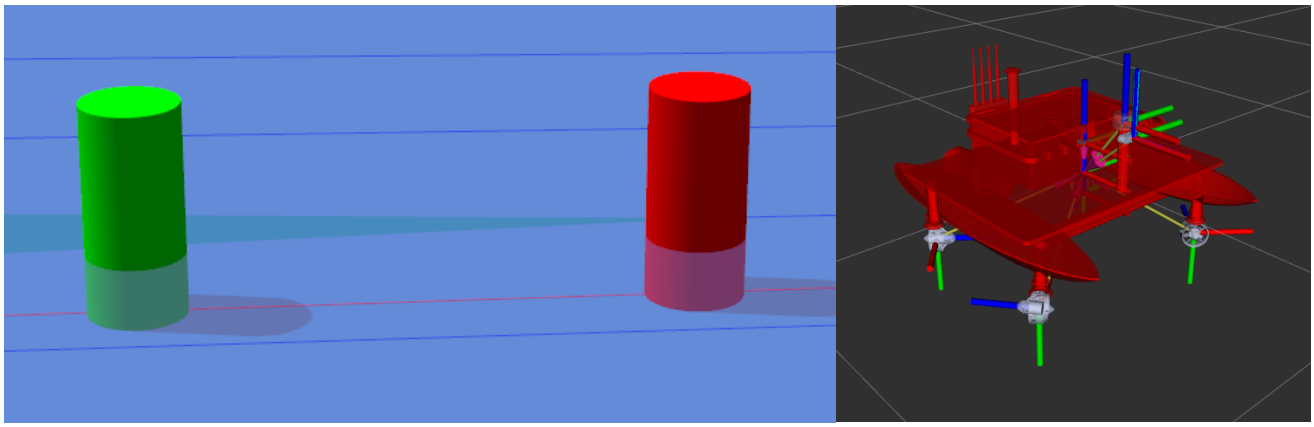


Fig. 22: Integrated ROS2-Gazebo-RViz Simulation Pipeline for A Path Planning and Navigation Testing

Our simulation environment integrates ROS2 with Gazebo physics engine and RViz visualization tools to validate navigation algorithms and control strategies in a virtual maritime setting before on-water deployment which is shown in Fig. 22. The top panels demonstrate our color segmentation pipeline, where HSV-based thresholding generates binary masks isolating red (left) and green (center) channel markers from the simulated camera feed, followed by contour detection and bounding box regression producing the detection result (right) with confidence scores and class labels overlaid on identified buoys. The middle panel displays the RViz 2D navigation interface showing the simulated competition course with occupancy grid mapping, where the vessel's planned trajectory (blue line) is computed using the A* (A-star) pathfinding algorithm. A* employs a heuristic-based graph search that evaluates nodes using the cost function $f(n) = g(n) + h(n)$, where $g(n)$ represents the actual cost from start to node n and $h(n)$ is the heuristic estimate to the goal, typically Euclidean or Manhattan distance in maritime applications. The algorithm efficiently explores the configuration space by maintaining an open list priority queue ordered by f-score, expanding the most promising nodes first while guaranteeing optimal path discovery when using an admissible heuristic. Our implementation discretizes the water surface into a grid-based representation with obstacle inflation around detected buoys and static hazards, enabling A* to generate collision-free trajectories that navigate between the green and red markers while maintaining safe clearance from obstacles. The blue star icon represents the vessel's current pose with orientation, while the red cluster indicates the goal region or docking target. The bottom-left panel shows the Gazebo 3D simulation environment rendering photorealistic cylindrical buoy models (green starboard and red port markers) with accurate hydrodynamic properties, water surface shaders with wave dynamics, and physically-based lighting to replicate real-world visual conditions. The bottom-right panel presents the vessel's URDF (Unified Robot Description Format) model visualized in RViz, displaying the robot's kinematic structure with coordinate frame axes (RGB representing XYZ), thruster mounting positions (green propellers), sensor placements, and collision geometry. This comprehensive simulation framework enables Hardware-in-the-Loop (HIL) testing of perception modules, iterative tuning of PID control gains for station-keeping and waypoint navigation, Monte Carlo testing of A*-generated paths under varied environmental conditions, and validation of sensor fusion techniques combining simulated IMU, GPS, and camera data streams before physical system integration.

ROS2 Node Structure

Manual Testing using JoyStick:

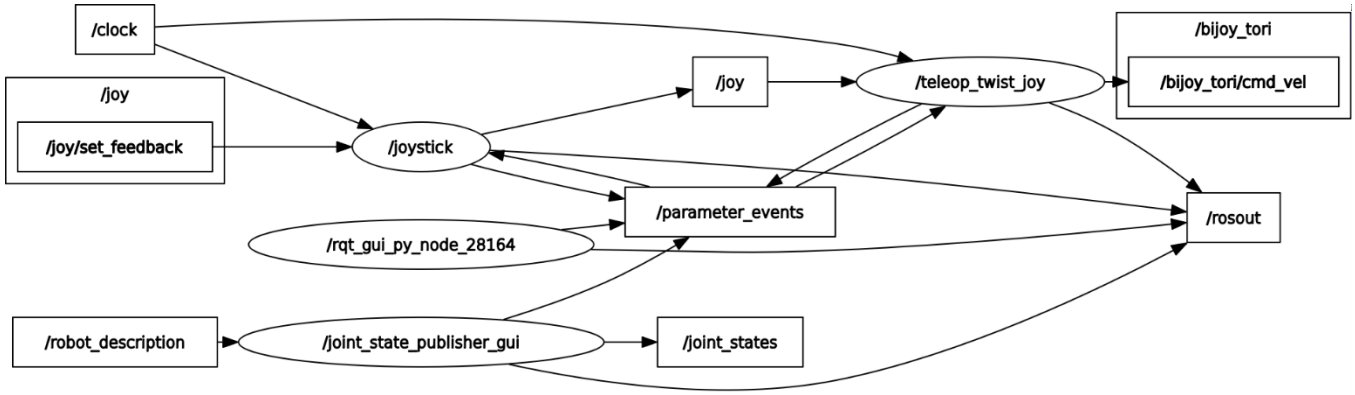


Fig. 23: ROS Node Graph for Manual Joystick Teleoperation System

Our manual teleoperation testing framework utilizes a joystick-based control interface integrated within the ROS (Robot Operating System) ecosystem to validate vessel dynamics and control responsiveness prior to autonomous operation deployment is shown in Fig. 23. The ROS computation graph illustrates the message-passing architecture where joystick inputs are captured via the `/joy` topic publishing sensor_msgs/Joy messages at a configurable frequency determined by the `/clock` node. The `/joystick` node serves as the primary input handler, subscribing to raw joystick data and `/joy/set_feedback` for haptic feedback control, while publishing processed commands to downstream navigation nodes. The `/teleop_twist_joy` node performs critical velocity mapping, translating joystick axis deflections and button states into geometry_msgs/Twist messages containing linear and angular velocity commands published on `/bijoy_tori/cmd_vel` for thruster actuation. This teleoperation pipeline interfaces with the `/rqt_gui_py_node_28164` for real-time parameter tuning and the `/parameter_events` topic for dynamic reconfiguration of control gains, deadzone thresholds, and velocity scaling factors. Additionally, the `/joint_state_publisher_gui` node provides visualization of the robot's kinematic state by publishing to `/joint_states` and subscribing to `/robot_description`, enabling operators to monitor actuator positions and vessel orientation during manual navigation trials. The `/rosout` logging node aggregates diagnostic messages from all active nodes, facilitating debugging and performance analysis. This comprehensive manual control architecture enables operators to assess thruster response characteristics, validate sensor integration, test collision avoidance behaviors, and establish baseline performance metrics under human supervision before transitioning to fully autonomous mission execution for RoboBoat competition tasks.

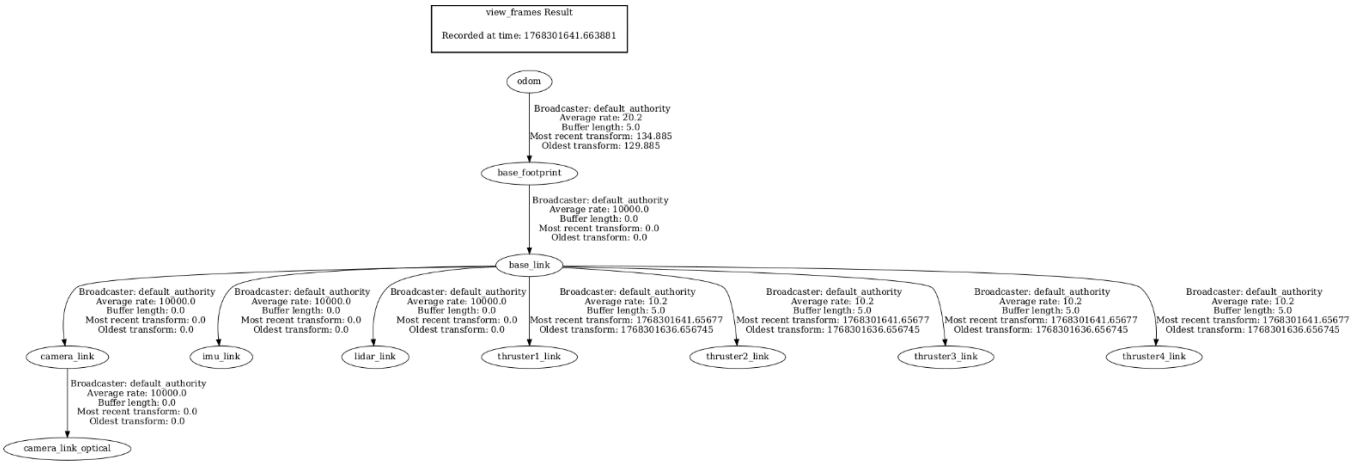


Fig. 24: ROS2 Transform Tree (tf2) for Vessel Coordinate Frame Hierarchy

Our RoboBoat vessel architecture is implemented using ROS2 (Robot Operating System 2), leveraging its improved real-time performance, Quality of Service (QoS) policies, and DDS (Data Distribution Service) middleware for reliable inter-process communication in maritime environments which is depicted in Fig. 24. The system employs a hierarchical node structure rooted at the `/odom` node, which serves as the central odometry publisher broadcasting the vessel's pose (position and orientation) and velocity estimates at 20.2 Hz with a 5.0-second buffer depth, accumulating 34,885 most recent transforms and 4,883 oldest retained transforms for temporal consistency. The odometry data propagates through transform tree branches including `/base_footprint` (vessel's ground projection frame) and `/base_link` (vessel's center of mass reference frame), which coordinate with sensor-specific frames: `/camera_link` for vision system mounting, `/imu_link` for inertial measurement unit data fusion, `/lidar_link` for 3D environmental mapping, and multiple `/thrusterN_link` nodes (where N = 1, 2, 3, 4) representing individual thruster actuation frames for differential thrust vectoring. Each transform link maintains DDS default authority with configurable buffer lengths and publishing rates, ensuring synchronized coordinate transformations across all sensor modalities. The `/camera_link_optical` frame represents the optical center transformation adhering to camera calibration parameters for accurate 3D point projection from image coordinates. This tf2 (transform) tree enables seamless coordinate frame conversions essential for sensor fusion algorithms, path planning in the /odom reference frame, and accurate control command generation in thruster-local frames. The timestamp-synchronized transform broadcasts (recorded at 1763016441.663881 UNIX time) ensure temporal alignment between asynchronous sensor streams, GPS/IMU fusion for dead-reckoning, and vision-based object detections, providing the foundational spatial reasoning infrastructure for autonomous navigation in the competition's GPS-challenged aquatic environment.

APPENDIX: F
ASV TESTING, COMMUNITY OUTREACH AND STEM ENGAGEMENT

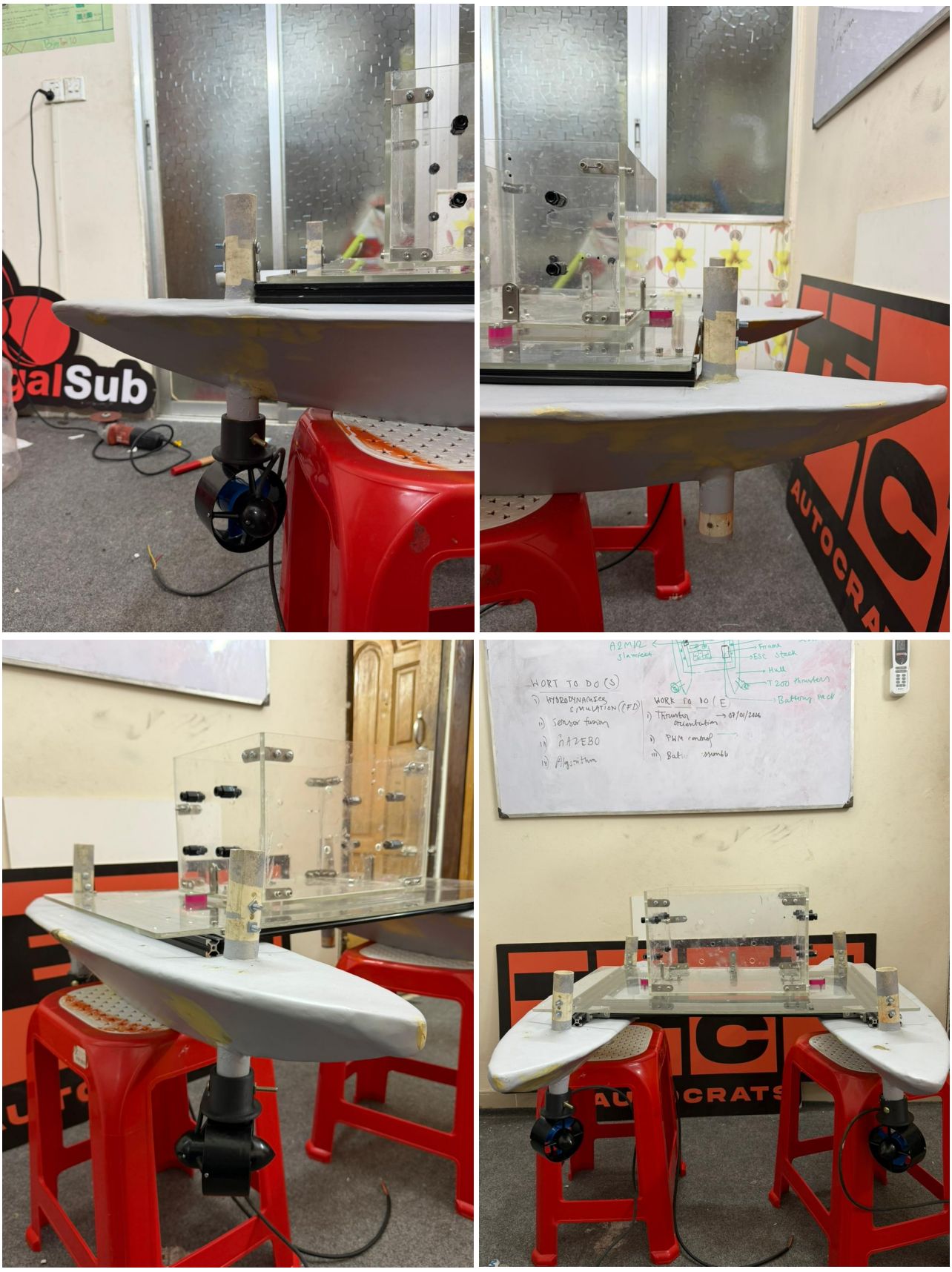


Fig. 25: Initial prototype of the BTASV 1.0

র শেষাংশ

নওরোজ ৭

যুক্তরাষ্ট্রে রোবোবোট প্রতিযোগিতায় বাংলাদেশি 'টিম বেঙ্গলবোট'

Dhaka Thursday, January 15, 2026

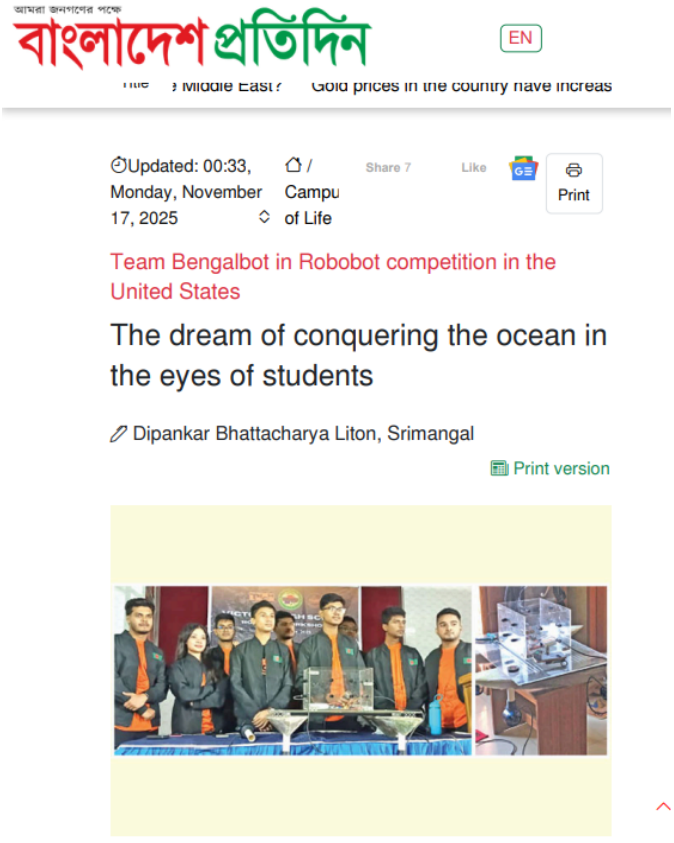
সমকাল

E-Paper English

Technology

Bangladeshi student team 'Team Bengalbot' is participating in robot competition in the United States

Members of Team Bengalboat



আমরা জনগণের পক্ষে

বাংলাদেশ প্রতিদিন

EN

Updated: 00:33, Monday, November 17, 2025

/ Campu of Life

Share 7 Like Print

Team Bengalbot in Robobot competition in the United States

The dream of conquering the ocean in the eyes of students

Dipankar Bhattacharya Liton, Srimangal

Print version

Fig. 26. News Media Coverage

Team BengalBoat has strengthened its community outreach through digital media, print journalism, and national television to share our mission and inspire young innovators across Bangladesh. Our road to RoboBoat 2026 has been featured on RTV and Boishakhi TV, highlighting our work on autonomous surface systems and the impact of hands-on engineering. In print and online news, outlets including Samakal, Daily Nawroj, Patakuri, and Holy Sylhet have covered our progress, reflecting the creativity and discipline behind our build. These platforms have expanded our national visibility, connected us with supporters, and encouraged students to see robotics as something they can learn, build, and lead.

Fig. 27: Community outreach program

Team BengalBoat is deeply committed to inspiring the next generation of innovators through active STEM outreach. Our team members have conducted interactive sessions at local schools, introducing students to robotics, autonomous surface systems, and real-world engineering. By showcasing key parts of our boat and explaining how control and navigation work, we helped students connect classroom concepts to practical applications. These sessions were designed to be engaging and inclusive, giving students space to ask questions, handle components, and learn through demonstration. Our goal is to make STEM more accessible and exciting, especially for students in underserved communities, and encourage them to dream big in technology and innovation.

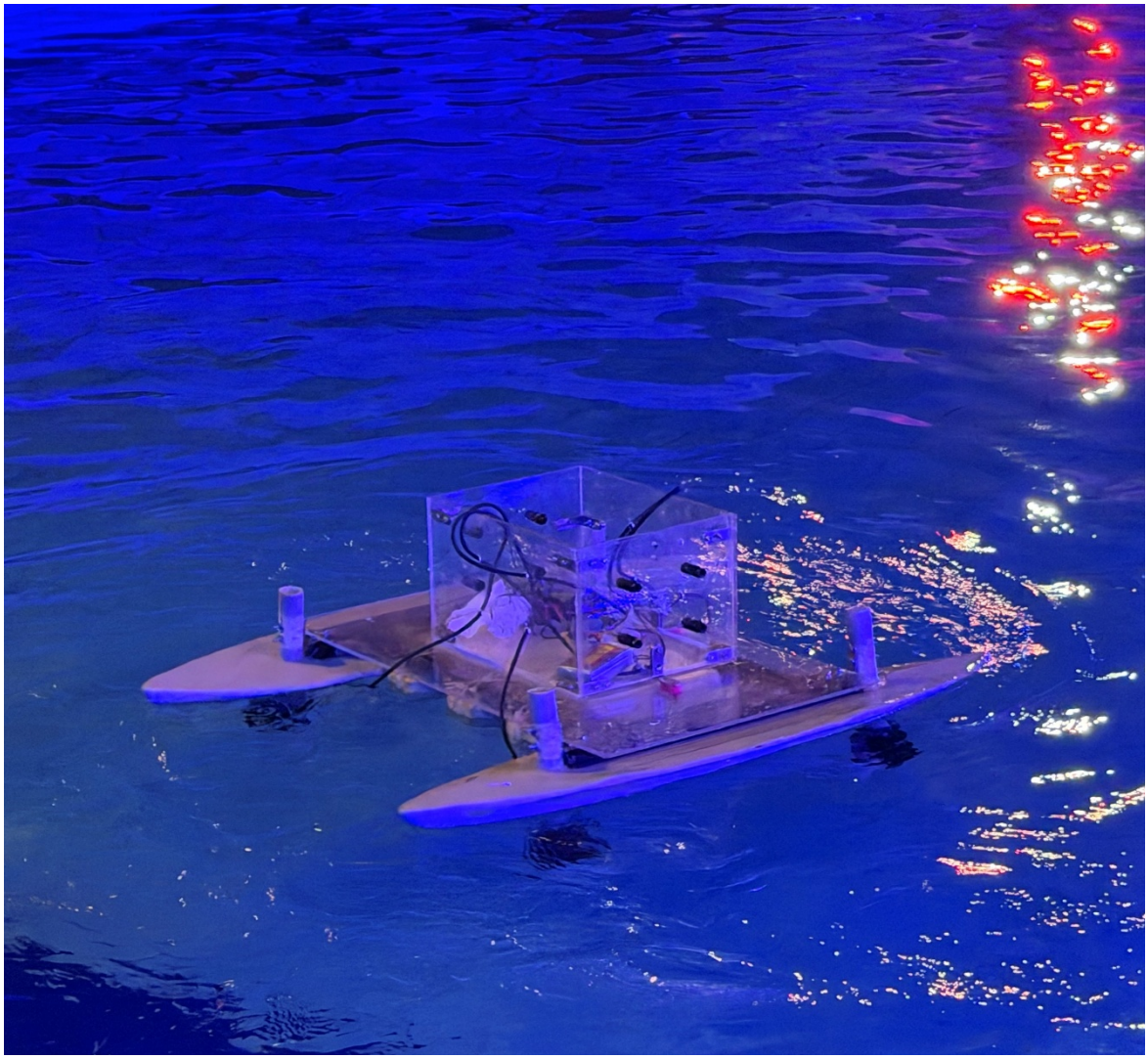


Fig. 28: An Autonomous Surface Vehicle, BijoyTori-1.0