Roboboat 2025: Technical Design Report AGH Solar Boat Team

Anastasiia Vudvud, Łucja Wuls, Olivier Lichota, Filip Nawalaniec, Aleksander Młynarski, Przemysław Domagała, Michał Zawada, Dawid Steposz, Mateusz Kalandyk, Mateusz Jaźwiecki, Martyna Czerniak, Patryk Kozłowski, Kacper Kustra, Błażej Dyksiński
AGH University of Krakow, Krakow, Poland

***Abstract*—The autonomous boat Barka, developed by the AGH Solar Boat Team, returns to the RoboBoat competition with a strategic focus on localization reliability and deterministic performance. This year's mission readiness is driven by the integration of an RTK base station and a ZED2i camera to eliminate positioning errors, while the modular catamaran platform has been updated with a microphone for the new Harbor Alert task.**

The software architecture has transitioned to a more scalable Behavior Tree system, replacing traditional state machines to facilitate independent node testing. Real-time perception now combines YOLOv10 vision detection with LiDAR-based Euclidean clustering, providing a robust environmental model for SLAM-based mapping and A-Star path planning. Validated through MATLAB simulations and extensive trials at Bagry Lake, this integrated approach ensures Barka handles complex autonomous tasks with high precision and repeatability.

Competition Strategy

Building on the existing system, we integrated a ZED2i wide-angle camera and an RTK GNSS base station to enhance environmental awareness and eliminate localization drift. Our strategic priority this year is improving performance in Navigate the Marina, Emergency Response Sprint, and Harbor Alert. To manage increased system complexity, we transitioned to a modular Behavior Tree architecture, allowing for independent testing and deterministic mission execution

I. Evacuation Route & Return

To address past instabilities caused by GPS-only localization and buoy proximity issues, we now utilize high-precision RTK localization and sensor fusion. Camera reliably detect and track buoys, with spatial filtering preventing duplicate detections. The ASV selects the nearest red and green buoys and navigates through a waypoint at their midpoint. RTK corrections eliminate localization drift, ensuring consistent and accurate passage through the center of the gates.

II. Debris Clearance

In 2025, during the Follow the Path task, we encountered issues with obstacle counting and localization, with some yellow buoys being detected multiple times.

This year, our Debris Clearance approach is based on perception data from the processing layer, which provides tracked buoy positions and classifications to the control system. When gate buoys are visible, the ASV navigates through the channel by generating waypoints at the midpoint of each red–green buoy pair, while storing the followed path.

If the gates are no longer visible, the system switches to a debris scanning behavior, where the boat reacts to detected objects based on their type. After completing the debris scanning phase, the ASV returns through the channel by following the previously stored path.

III. Emergency Response Sprint

To overcome limited camera FOV and unreliable GPS localization from previous years, we implemented RTK corrections and a structured search pattern to detect the color indicator buoy using vision-based recognition. It then performs a structured search, gradually approaching the expected location of the yellow buoy until it is detected. Sensor fusion from the ZED2i camera, LiDAR, and IMU ensures accurate buoy localization during the approach. Once confirmed, the ASV executes a clockwise or counterclockwise circling maneuver, continuously refining its trajectory to maintain high speed, smooth motion, and safe clearance before exiting the course.

IV. SUPPLY DROP

At previous competition, we completed "Object delivery" on qualifications but did not attempt the task during semi-finals or finals, because of other challenges and problems.

This year the Supply Drop task employs a selection algorithm that identifies the nearest target boat, after which the ASV generates a two-waypoint approach sequence: an initial standoff point at a configurable distance (default 3.0m) and a closer approach point at half that distance, both oriented toward the target boat. The tracking system assigns unique IDs to detected boats and publishes their positions and types to coordinate delivery actions. Once the ASV reaches the target position, it executes the appropriate action sequence-either shooting or spilling-based on the detected boat classification.

V. NAVIGATE THE MARINA

Recognizing that placard detection alone was insufficient, we developed a geometry-driven strategy using LiDAR-based marina layout modeling for precise alignment. The ASV identifies marina structures as line primitives from LiDAR scans to model slip centers in the global frame. This model generates safe navigation waypoints, including specific entry and exit points.

During transit, the ASV fuses LiDAR and camera depth data to inspect each slip for occupancy, traffic light status, and placard numbers. After evaluating two marina sections, the system selects the optimal slip-one that is vacant, displays a green light, and has the lowest number. Finally, the ASV maneuvers to the slip center, exits, and holds at a designated safe position.

VI. HARBOR ALERT

The ASV continuously monitors the acoustic environment using an FFT-based audio pipeline. Audio frames are windowed, transformed to the frequency domain, and refined with parabolic interpolation for accurate tone detection.

Detected tones exceeding a magnitude threshold are grouped into blasts, classified as SHORT or LONG based on duration. Blast frequency and count are published to the decision module via ROS 2. When a predefined signal pattern is recognized, the ASV immediately interrupts its current task and navigates to the assigned target zone, maintaining collision avoidance throughout.

Design Strategy

Our strategy focuses on enhancing a proven modular catamaran platform through high-precision perception and a modular software architecture. To manage complexity and improve reliability, we transitioned from state-machine-based control to Behavior Trees, enabling scalable and reusable task execution. Key engineering decisions were driven by the need for better situational awareness and navigation precision. This includes upgrading to wide-angle stereo vision to eliminate blind spots and implementing RTK-GNSS for centimeter-level localization. By refurbishing reliable mechanical components and overhauling the autonomy stack, we achieved a robust system that balances physical stability with decision-making.

I. MECHANICAL AND HARDWARE MODIFICATIONS

A. ***Hull and Propulsion***

Fig. 1: Image of Barka

The ASV continues to use a modular catamaran hull that supports quick disassembly for transport and maintenance, including air travel. While the basic hull design remains unchanged, the float modules have been refurbished and refinished for improved durability. With a footprint of 1.3 m x 0.9 m, the ASV utilizes two BlueRobotics T200 thrusters located under the rear of each float for propulsion.

B. *Sensor Suite Integration*

To support the new autonomy software, the sensory system was physically expanded:

*1) **Visual**:* For tasks with multiple visual targets and dynamic maneuvering near gates and obstacles, we upgraded visual perception by replacing the standard Stereolabs ZED2i stereo camera with a wide-angle variant. This expanded the field of view from 72° to 120°, boosting situational awareness without mechanical complexity. The wider sector minimizes gates and obstacles leaving the frame during tight maneuvers, enabling stable detections and tracking. Consequently, planning and obstacle avoidance receive consistent perception streams, enhancing reliability and smoother trajectories with fewer corrections.

*2) **Audio**:* For the Harbor Alert task, we integrated the Xtrike Me XMC-03 condenser microphone. Its omnidirectional pattern ensures reliable blast detection regardless of vessel orientation. The USB interface avoids external amplification or ADCs, cutting EMI issues and simplifying drivers. Mounted at the front of the aluminum frame, away from thrusters, it enjoys a lower noise floor and better signal-to-noise ratio.

*3) **Precision Positioning**:* For reliable, repeatable navigation across the competition course, we added a high-precision system using RTK GNSS with INS support. It splits between a shore-based reference station and onboard unit, connected via long-range radio telemetry. RTK corrections boost accuracy and stability, improving waypoint tracking, map consistency, and maneuver repeatability. Broadcasting standard RTCM messages over a dedicated link ensures independence from cellular coverage, enhancing waterborne robustness. The shore station computes differential corrections from its fixed position, while the onboard GNSS/INS applies them to resolve carrier-phase ambiguities. This integrates seamlessly with onboard drivers and the navigation stack, delivering stable localization for autonomy modules.

II. Electrical and Communication Systems

A. *Power Management*

Electrical architecture remains mostly unchanged from the previous year, as the system has reached a state of reliability. The core of the system continues to rely on a custom power distribution PCB to ensure organized cable management and integrated fuse protection for all electronic components. The Battery Management System remains the primary safety layer, continuously monitoring battery voltage, current, and temperature. It features an automated safety cutoff for the thrusters in critical conditions, alongside an acoustic buzzer and a 360-degree LED Status Light Post to provide immediate visual feedback on the vessel's operational mode.

B. *Internal Network*

To comply with competition communication rules, we implemented a local Wi-Fi network using two Ubiquiti Bullet M2 devices configured in bridge mode. The shore unit operates as an access point with a sector antenna, while the ASV unit connects in station mode using an omnidirectional antenna. Their high transmission power and dedicated antennas ensure stable long-range communication. Onboard, the Wi-Fi link is integrated with the ASV LAN connecting the Jetson computer and LiDAR via a 1 Gbps switch. To maintain proper bridge operation, only the two Bullet devices are included in the bridge network, while a separate shore access point is used for crew devices.

III. Software Architecture

A. *Layer 0: Hardware Drivers*

The L0 layer serves as the primary interface for all hardware communication, contains low-level messaging, CAN bus interactions, and motor control calculations. This layer utilizes the ZED wrapper, the Hesai ROS wrapper, and a custom-developed CAN driver to manage data from the hardware components. By utilizing GNSS data integrated with RTK corrections, the system achieves the high-precision positioning necessary for accurate navigation and mapping.

A central component of L0 is the custom CAN driver, which facilitates the essential communication bridge between the high-level software stack and the vehicle hardware. Beyond managing the thrusters, this layer contains the specific logic required to handle peripherals, formatting and sending the appropriate CAN messages to operate the water pump and the custom ball gun. This ensures that all sensors and actuators are synchronized with high-level control logic, providing a robust and reliable foundation for physical execution.

B. Layer 1: Object Detection

Following the recent architectural refinement, the system formerly known as the Data Processing Layer has been streamlined and renamed to Object Detection. The L1 layer now focuses exclusively on real-time perception, data fusion, and object tracking.

This layer provides the critical perception capabilities of the vessel, processing vision, and point cloud data to identify and track obstacles in real time. L1 serves as a dedicated detection engine that ensures high-fidelity environmental awareness.

*1) **Autolabeling & Augmentation**:* To streamline model training, we utilize a partial auto-labeling workflow. A subset of our dataset is manually labeled to train an initial model, which then labels the remaining raw data. This process reduces manual labor to a review-only phase, significantly accelerating model preparation. We also apply extensive data augmentation, simulating conditions like intense sunlight and glare, to ensure the model remains robust across varying maritime environments.

*2) **Optimized Neural Network Inference**:* For real-time vision, we use YOLOv10, which offers superior accuracy and speed compared to previous iterations. To achieve the low latency required for autonomous operations, we implement quantization, reducing weight precision from 32-bit to 16-bit with negligible accuracy loss. Using frameworks such as CUDA and TensorRT, we optimize parallel computing on the GPU. The workflow includes a 10-step warmup to initialize GPU resources, followed by image preprocessing (resizing and normalization), inference, and the conversion of outputs into ROS2-compatible formats.

*3) **Clusterization & LiDAR Processing**:* In parallel with the vision pipeline, we process LiDAR point clouds using the Euclidean Clustering algorithm via the Point Cloud Library (PCL). This method segments objects based on their spatial proximity, providing precise distance measurements. This year, we have enhanced this process by incorporating camera point cloud data from the ZED2i, which provides a denser data set that improves the initial estimation and reliability of the fusion algorithm.

*4) **Detection Fusion & Tracking**:* The core of L1 is the Detection Fusion system, which synchronizes data from both the vision and cluster-based detectors using precise timestamps. The system evaluates the angular positions of detections from both sources to determine if they represent the same physical object. Once fused, the system integrates LiDAR and camera data with the Xsens IMU/GNSS inputs to calculate global coordinates, relative to the vessel's starting position.

Each detection is processed through a tracking algorithm that compares new coordinates with the locations of previously identified objects. Based on a defined distance threshold, objects are either assigned to a new ID or matched to an existing one. This persistent tracking allows the system to distinguish between various buoys and obstacles, maintaining a stable environmental model for the higher-level SLAM and planning layers.

C. Layer 2: Data Processing and SLAM

*1) **Layer Components and Signal Processing**:* This layer provides the calculation necessary to navigate to a certain waypoint or detect complex structures from lidar such as a marina or dock. SLAM, nodes, mapper and planner are placed here. Apart from them, there are a few more components. The Odometry node is responsible for running a kalman filter to provide the most accurate position based on IMU accelerometers and GNSS with RTK corrections. Pointcloud_to_laserscan is responsible for degrading the 3D pointcloud from LiDAR to a 2D laserscan, which is easier to process for mapper and segment_scan. Segment_scan is used to aggregate points from laserscan into straight lines in order to detect docking slips or marina by dock or marina detectors. To precisely follow calculated path we use Controller with a physical model of our boat.

*2) **SLAM and Path Planning Strategy**:* In order to complete tasks and navigate precisely, we need a reliable tool to keep us oriented in the surroundings. The Simultaneous Localization and Mapping system turned out to fit perfectly in our architecture, specifically the Mapper and Planner functionality. Firstly, the data from LiDAR is processed to a local map and synchronized with position information from odometry in order to form and update a global map. Then the costmap is generated from the global map. Concurrently, the fused detections from Layer 1 are compared with those already seen and combined to keep track of objects on course. Finally, as soon as Layer 3 establishes the goal, the planning process takes place. We use the A-Star algorithm with a modified Euclidean heuristic run on the costmap to find a safe and efficient path to the goal. The plan is updated as the vessel makes its way to the waypoint to ensure no collisions with new objects.

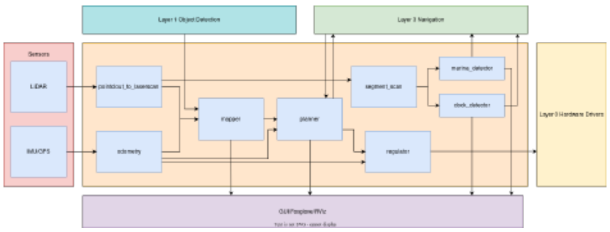


Fig. 2: Software architecture

3) ***Motion Control:*** The controller on the boat is responsible for processing signals from sensors such as position, speed, and heading, and comparing them with reference values derived from the planned trajectory. Based on the resulting error, it generates control commands for propulsion and steering systems to ensure stable motion and accurate path tracking. Its task is to minimize position and orientation errors while maintaining smooth and safe maneuvers. The controller must also respond to external disturbances, such as wind, waves, and varying load conditions, by compensating for their influence on the vessel's motion. As a result, the boat can autonomously perform navigation tasks, maintaining the desired course and speed even under changing environmental conditions.

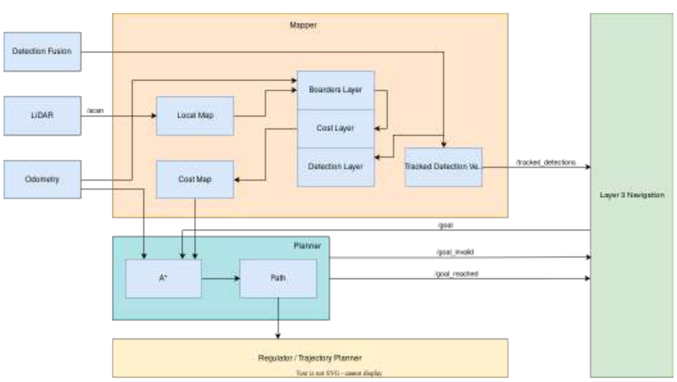


Fig. 3: High-level steering architecture

D. Layer 3: Navigation

The previous control architecture was based on a state machine that managed the execution flow of a control algorithm implemented for a specific task. This approach was effective during early development stages involving single, well-defined missions. However, as the system grew and the number and complexity of supported tasks increased, limitations of the state-machine-based approach became apparent. To improve scalability and maintainability, Behavior Trees were introduced. They are easier to extend and reorganize, allowing new behaviors to be added without redesigning the entire control structure. Instead of implementing full tasks, individual tree nodes are designed around smaller functionalities. This modular approach enables reuse of nodes across multiple tasks, reducing code duplication, and simplifies testing and debugging, as nodes can be developed and validated independently before integration into the central decision-making layer.

This layer is responsible for taking decisions, giving waypoints to steer and accomplishing tasks. It contains only one node, basic_controller which handles behavior tree ticking. Not only it is made from control nodes but also custom subnodes covering every activity from spinning, breaking, and steering to wp to looking for buoys, boats and docks. This layer receives data about known objects from Layer 2 and calculates the waypoint to reach following the task rules. [**?**]

E. Human-Machine Interface (HMI)

To provide a clear view of the system's operational state, we use Foxglove GUI as a visualization and diagnostic tool integrated with our ROS 2 architecture. Foxglove serves as the main monitoring interface, allowing visualization of high-bandwidth ROS topic data transmitted from the vessel. Using the foxglove_bridge, a wireless link is established between the onboard computer and the operator station, enabling real-time visualization of camera feeds, point clouds, and maps generated by SLAM and planning modules. This level of feedback is essential during water trials, as it allows engineers to observe system behavior, diagnose issues in real time, and validate performance more efficiently and safely.

Testing Strategy

Tests are a crucial part of the development process and a key indicator of progress. System testing can be divided into four parts: individual component checks, simulation tests, dry tests, and on-water tests. This year we included MATLAB Simulation for tests of the boat controller.

I. Dry tests

To validate hardware robustness, we conduct dry tests in our workshop, including individual component checks to ensure that key elements (e.g., Jetson, ball shooter, or LiDAR) function correctly before full system activation. The nozzle is tested by connecting it to the boat and manually issuing commands to start and stop water flow, and the same procedure is applied to the ball shooter. The next stage involves activating the entire boat, beginning with verifying correct signal transmission from the radio to the thrusters. Then, the Jetson - serving as the main autonomous system component - is activated, and we confirm that all required information is properly received and transmitted. Finally, we test essential functionalities such as object detection and waypoint navigation and observe the resulting thruster outputs. These dry tests are performed near our workshop, providing a quick and convenient way to evaluate features that do not require water.

II. Simulation tests

The basis for software testing in simulation is VRX world in Gazebo simulation. Every new functionality undergoes thorough software testing in simulation before being implemented on the physical system. This allows us to maximize testing time on the water. Simulation allows us to refine the logic in algorithms and make sure the detection processes work well.

MATLAB simulation was created to conduct the validation of the control algorithm for the boat. It effectively minimises position and orientation errors against the planned trajectory. It allows us to test the system's ability to compensate for external disturbances, such as wind and waves. The simulation confirms the vessel's ability to autonomously maintain its desired course and speed under varying conditions without on-water testing.

III. Water tests

We conduct on-water tests on two lakes in Kraków: Bagry and Zakrzówek. Since last year, we have had buoys for Speed Challenge and Navigation Channel and placards for the Object Delivery task. This year, we built a dock from extruded polystyrene foam to enable testing of the Docking task. This setup allows us to test all possible on-water tasks. Additionally, we have a dataset that includes all types of objects, which we use to train object detection models. This approach ensures that our testing scenarios closely mirror the competition conditions, allowing us to refine and optimize Barka's performance in a controlled setting before deployment.

Acknowledgments

We extend our heartfelt gratitude to AGH University for their invaluable support and resources throughout the development of our project. We are especially thankful to our academic advisors, Krzysztof Sornek and Wojciech Sajdak, for their expertise, guidance, and dedication.

Our deepest thanks also go to TME, Xsens and ABB for their crucial support in design and funding. We thank Mine AGH for filming our promotional video and providing media materials. We also appreciate the HORN sailing club for enabling in-water boat tests and the VRX developers for their dynamic simulation environment.

A special gratitude goes to our team members Łucja Wuls, Michał Zawada, Ewa Nowicka and Przemysław Domagała for preparing the application for the Ministry of Science and Higher Education scholarship, without which participation in this competition would have been much more challenging.

Finally, we extend our deepest gratitude to every member of our team for their dedication, time, and passion that made this project a reality.

References

[1] Robonation Inc., 2026 Team Handbook, 2025, robonation.gitbook.io/roboboat-resources.

[2] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, ”Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.

[3] M. Colledanchise and P. Ögren, ”Behavior Trees in Robotics and AI: An Introduction,” CRC Press, 2018.

[4] A. Wang et al., ”YOLOv10: Real-Time End-to-End Object Detection,” arXiv preprint arXiv:2405.14458, 2024.

[5] Ultralytics, ”YOLOv10 - Ultralytics YOLO Docs,” [Online]. Available: https://docs.ultralytics.com/models/yolov10/.

[6] P. J. G. Teunissen and O. Montenbruck (Eds.), ”Springer Handbook of Global Navigation Satellite Systems,” Springer International Publishing, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-42928-1

[7] B. Bingham et al., ”Toward Maritime Robotic Simulation in Gazebo,” OCEANS 2019 MTS/IEEE SEATTLE, Seattle, WA, USA, 2019, pp. 1-10, doi: 10.23919/OCEANS40490.2019.8962724.

[8] R. B. Rusu and S. Cousins, ”3D is here: Point Cloud Library (PCL),” 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 1-4, doi: 10.1109/ICRA.2011.5980567.

[9] D. Bruijnen, J. van Helvoort and R. van de Molengraft, ”Realtime motion path generation using subtargets in a changing environment,” 2006 American Control Conference, Minneapolis, MN, USA, 2006, pp. 6 pp.-, doi: 10.1109/ACC.2006.1657385.

[10] N. Koenig and A. Howard, ”Design and use paradigms for Gazebo, an open-source multi-robot simulator,” 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 2004, pp. 2149-2154 vol.3.

[11] K. Miadlicki, M. Pajor and M. Saków, ”Ground plane estimation from sparse LiDAR data for loader crane sensor fusion system,” 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 2017, pp. 717-722.

[12] H. Durrant-Whyte and T. Bailey, ”Simultaneous localization and mapping: part I,” IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006.

[13] Foxglove Technologies Inc., ”Foxglove Documentation,” 2024. [Online]. Available: https://docs.foxglove.dev/docs

APPENDIX A
TEST PLAN & RESULTS

A. Scope

We conduct monthly tests of our boat, starting with basic functionalities and gradually progressing to more advanced ones. Basic tests include verifying that the electrical system, detection fusion, and waypoint navigation function as expected. As competition approaches, we focus on testing more advanced functionalities.

B. Schedule

The tests are planned in a way, so that we can test basic functionalities first, such as swimming by waypoints, detection fusion or receiving RTK corrections. Almost everything from that list was tested on Njord Challenge 2025, so now we are testing new features.

- **November:** Using depth from ZED 2i camera, SLAM mapper and planner
- **December:** Boat contorller, behavior trees for basic funtionalities (buoy gates)
- **January:** RTK base station, behaviour trees for Navigate the Marina, Emergency Responce Sprint, Evacuation, Debris Clearance
- **February:** Hardware & behaviour tree for Harbor Alert, ball launcher

C. Environment

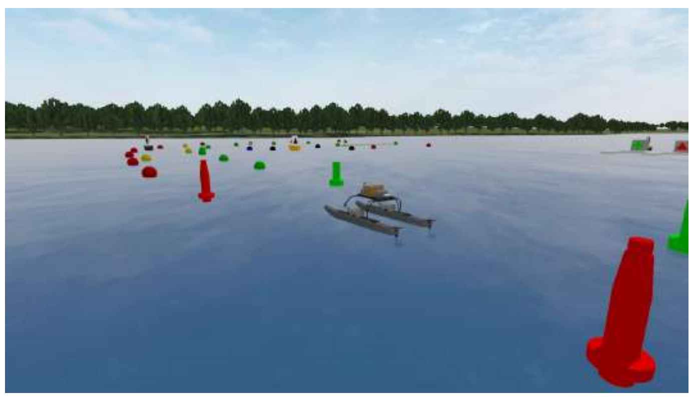


Fig. 4: VRX simulation environment for the whole course

Our simulation is built on a world originally created for the VRX competition. Using models from VRX or custom-made ones, we recreated each task and the entire RoboBoat 2026 course, except for the Harbour Alert task. While this is an idealized environment, it is perfect for developing algorithms that require continuous testing.

In MATLAB we are able to see how the steering algorithm controls the boat. This allows us to perfect the control parameters to ensure the vessel minimizes trajectory errors and executes smooth maneuvers. On Figure 5 steering vs. actual speed on both axes can be seen.

For in-water testing, we use Bagry Lake in Kraków, specifically the harbour of the HORN Sailing Club. It provides a safe environment, often used by beginner sailors, and allows us to position buoys and other visual marks as needed. However, in the winter it can freeze, so we move to Zakrzówek Lake.

D. Risk Management

Potential risks on the water include someone slipping on a slippery entrance to the water. To place buoys, we use paddle board or kayak and there is a risk that someone can fall into the water. We use this lake regularly for the tests of both our boats – autonomous boat and solar powered racing boat, so members of our team have been through the safety training. We also make sure that the people on the kayak or paddle board wear lifejackets.

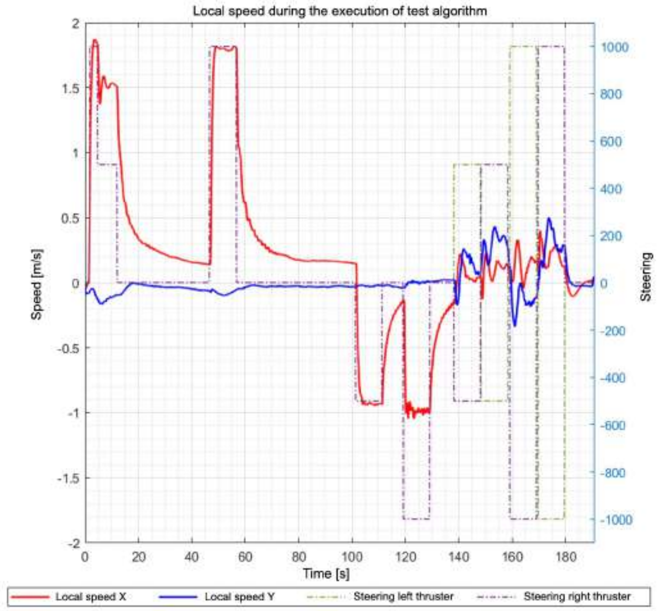
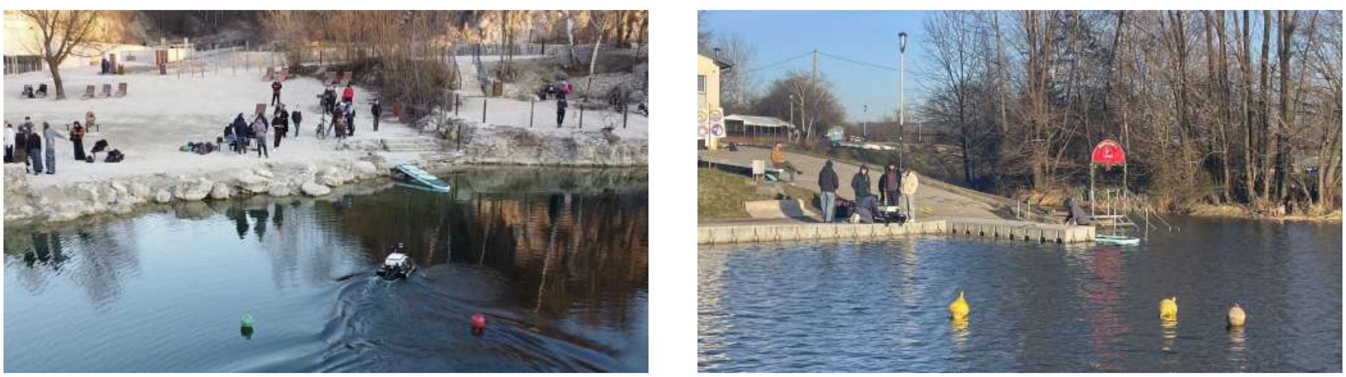


Fig. 5: Maximal displacement in previous load case in ball laucher



(a) Zakrzówek Lake

(b) Sailing club HORN in Krakow

Fig. 6: Water testing environments

E. Results

During the tests conducted in November, we confirmed that the hardware functioned as expected, with no issues detected. We also verified that the fusion of detections was working correctly and recorded some data. Global map is built correctly and costmap is generated inside SLAM. Tests in December showed that boat controller correctly computes the boat's path between the given points, which allows for high-precision navigation in the tasks. We have also started testing for tasks Evacuation Route and Debris Clearance.

In the next two months, we plan to test the remaining functionality, which is mentioned above in the plan. As always, we also plan to test in Sarasota before the competition.