IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

Heterogeneous Multirobot Team: Maritime Inspection and Intervention in Global Navigation Satellite System-Denied Scenarios

Barbara Arbanas Ferreira[1,*], Antun Ivanovic[2], Marijana Peti[2], Luka Mandić[2], Matko Batoš[2], Jakob Domislović[2], Goran Vasiljević[2], Juraj Obradović[1], Fausto Ferreira[1,2], Frano Petric[2], Antonella Barišić-Kulaš[2], Marko Car[2], Jurica Goričanec[2], Natko Kraševac[2], Marko Križmančić[2], Ivan Lončar[2], Lovro Marković[2], Robert Milijaš[1], Dario Stuhne[2], Matko Orsag[2], Stjepan Bogdan[1,2], and Nikola Mišković[1,2]

***Abstract*—Autonomous robotic systems play an increasingly important role in overcoming complex maritime challenges where the environment requires high adaptability, robustness, and precision. Marsupial robotic systems, combining aerial systems and surface vessels, offer a powerful solution for coordinated missions in GNSS-denied scenarios, leveraging the unique capabilities of each robot to form a versatile team. This paper presents a system integrating advanced sensor fusion, localization, mission planning, and communication to enable seamless collaboration between Uncrewed Aerial Vehicles (UAVs) and Autonomous Surface Vehicles (ASVs). The focus is on optimizing their interaction to efficiently locate, retrieve, and transport objects from a target vessel in an adverse sea environment. This paper provides a complete system overview, including the hardware setup, algorithms used, implementation details, and field trial results. The results analysis demonstrates the advanced capabilities of the developed ASV-UAV system in complex, real-world maritime operations, underscoring its potential for coordinated multi-robot tasks in environments where traditional navigation systems are unavailable. Notably, this system was developed by the team that secured first place in the Mohamed Bin Zayed International Robotic Challenge (MBZIRC) competition, proving its excellence in challenging conditions.**

I. Introduction

Pressed by the increasing demand and rapid technological development, the uptake of autonomous technologies in the maritime sector has seen significant growth in recent years. Particularly interesting and advantageous are heterogeneous systems in which Autonomous Surface Vehicles (ASVs) work together with Unmanned Aerial Vehicles (UAVs), leveraging the strengths and advantages of each robot as a team to accomplish complex tasks [1]–[9]. ASVs have the advantage of a longer operational range supported by a large battery capacity. At the same time, UAVs offer additional mobility, agile operation, and the ability to interact with the environment.

Competitions, such as the Mohamed Bin Zayed International Robotic Challenge (MBZIRC) [10], are an important driver for developing such advanced robotics systems. As one of the most challenging and complex international robotics competitions, MBZIRC takes place every three years in Abu Dhabi, setting new tasks that demand cutting-edge robotics technologies. The MBZIRC 2023 Maritime Grand Challenge demonstrated the advanced use of ASV-UAV systems to search and identify vessels in a vast GNSS-denied maritime environment and the safe recovery of objects from a target vessel. This included a smaller object recovered by a UAV and a larger object retrieved collaboratively with UAVs assisting a robotic arm mounted on the ASV. This complex competition scenario led to advances in visual identification of mission targets, vehicle navigation systems without Global Navigation Satellite System (GNSS), aerial manipulation, and mobile manipulation of objects with a robotic arm mounted on a floating base (the ASV), among others. While this paper focuses on results from the MBZIRC competition, similar capabilities—such as autonomous navigation, docking, and object detection—are also demonstrated in other international challenges, including RoboNation's RobotX and RoboBoat competitions, though with different platforms, operational scenarios, and scoring frameworks.

In this paper, the implementation of the robotic system is described, which was used in the grand finale of the MBZIRC 2023 competition (shown in Figure 1), ultimately leading to our winning performance. The hardware configuration of the UAVs and the ASV is presented in detail, the strategies and advanced algorithms used, and the results obtained in a real scenario.

To provide a comprehensive overview, the paper is organized as follows: In the next section, related works in the collaborative use of sea-aerial systems are covered. Furthermore, the mechanical design of robot platforms used with their complete hardware structure is discussed in Section III. In Section IV, the methods used for each subproblem of the Grand Maritime Challenge are outlined. Section V presents the experimental results and evaluates the system's performance in the competition. Finally, the paper concludes with the key findings and lessons learned from this implementation scenario in Section VI.

[1]CoE MARBLE - Centre of Excellence in Maritime Robotics and Technologies for Sustainable Blue Economy, Unska 3, Zagreb, Croatia

[2]University of Zagreb Faculty of Electrical Engineering and Computing, Unska 3, Zagreb, Croatia;

*Corresponding author: barbara.arbanas@marble.eu

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

II. Related work

Collaborative use of sea-aerial systems began in 2008 with the pioneering work presented in [2], which demonstrated that unmanned surface vehicles (USVs) and micro aerial vehicles (MAVs) can be effectively used in disaster response to assess damage to coastal structures. The study highlighted their role in post-disaster recovery and cooperative autonomous operations near structures. Although both vehicles were piloted, the research laid the groundwork for future studies by demonstrating how UAV camera perspectives could assist the USV teams in navigation.

Multirotor drones have a limited battery life, which restricts the operational range of the robotic team. Marsupial configurations address this limitation, where an ASV carries and charges the UAVs. The ASV's long range and large battery capacity enable UAV transport and charging during missions. One such application is presented in [3], where the authors propose a marsupial system for environmental monitoring. In this system, the UAVs support the ASV by mapping the environment and aiding navigation by providing long-range navigation cost maps.

The Sea-RAI project [4] demonstrated the first portable USV that carries and deploys a UAV to inspect coastal areas for military, environmental, and disaster response applications. The system combines a twin-screw USV platform with a commercially available UAV, enhancing capabilities by utilizing the UAV as a communications relay and providing an aerial perspective. At the same time, the USV reliably navigates near-shore environments and observes both above and below the waterline.

In [5], a cooperative USV-UAV system is proposed to improve the visual perception of the USV. The performance of the system is validated by multiple object detection, semantic segmentation, and obstacle detection from both USV and UAV perspectives. The results show that integrating USV and UAV perspectives enables accurate object detection, classification, and discrimination of navigable and non-navigable regions while accurately measuring obstacle distances.

Besides enhancing the perception and navigation range [6]–[8], marsupial sea-air systems can be used for various manipulation tasks, such as object recovery. In [9], a cooperative USV-UAV system is designed to recover floating objects at sea. The mission consists of three phases: a coarse approach using GPS-based navigation, a fine approach where the UAV visually tracks the USV and provides relative distance data, and a tugging phase where the USV retrieves the object. However, this scenario was performed only in a simulation environment.

While marsupial ASV-UAV systems are advancing rapidly and several successful applications have been demonstrated, none have been conducted in GNSS-denied marine environments, which pose a major challenge for UAV deployments. In GNSS-denied environments, UAV localization has been addressed using optical, acoustic, radar, and radio-frequency (RF) methods [11]. Optical techniques, including those using cameras, LiDAR, or markers, are precise but limited to small areas and are impractical in featureless settings such as the sea; similar limitations apply to acoustic approaches. Radar offers longer range but is bulky, expensive, and prone to misidentification. RF-based systems — using ultra-wideband (UWB), Wi-Fi, Bluetooth, ZigBee, Radio Frequency Identification (RFID), LoRa, or cellular networks — can cover large areas at a lower cost but produce noisier, lower-rate measurements whose accuracy depends on the range. Consequently, state-of-the-art methods apply Kalman filtering and often fuse RF positions with inertial data [12], [13], sometimes extending coverage with fixed anchors equipped with GNSS or combining RF with vision for precision tasks such as takeoff and landing [14]. In our work [15], we presented and evaluated the complete implementation of our approach, demonstrating long-range RF–based localization combined

Fig. 1: ASV-UAV robotic team in an object pickup task during the MBZIRC competition. The ASV is docked to the target vessel, and a UAV is scanning the deck of the target vessel in search of target objects. The robotic arm onboard the ASV (yellow) tracks and localizes the UAV using the LiDAR attached to its end effector.

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

with vision-assisted navigation for UAVs operating in large, featureless, GNSS-denied maritime environments, which is briefly presented here in Section IV-A.

For ASVs, GNSS-denied navigation typically begins with an inertial navigation system (INS), which integrates accelerometer and gyroscope data but gradually accumulates error ("drift"). This drift can be reduced by complementary sensors or learned models to improve accuracy and robustness [16], [17]. Techniques adapted from UAV research, such as visual odometry and simultaneous localization and mapping (SLAM), can enhance localization when distinct features are present. In contrast, radar, LiDAR, and radio-based systems, such as UWB and LoRa, enable reliable or wide-area positioning where landmarks are available [18], [19].

Besides the novelty of the GNSS-denied marsupial ASV-UAV system, no studies in the literature address aerial manipulation in real maritime conditions, especially involving the interaction between two floating platforms. Equipping an ASV with a manipulator arm to recover objects from another vessel in cooperation with UAVs remains unexplored and goes far beyond the current state of the art.

III. Mechanical Design of the Robotic Platforms

This section describes the mechanical architecture of the robotic platforms in detail. The focus is on the integration of sensors, communication, and control systems, which have been developed for robust operation under GNSS-denied conditions.

A. Hardware setup of the UAVs

Two aerial platform types were used: one to perform search tasks and another, slightly smaller, to perform object pickup tasks. The frame of the search UAV (Figure 2) consists of four arms, a body, and two legs with skis, all made of carbon fiber. The vehicle is equipped with four Flame 60 A 12S ESCs that drive T-motor P60 KV170 12S motors with 22.4x8.0 carbon fiber propellers, giving it a maximum takeoff weight of 15 kg with a 7 kg payload capacity.

The onboard computer used in the UAV is an Intel NUC 13 with an Intel Core i5–1340P processor, integrated Intel Iris Xe graphics, and 16 GB DDR4 RAM. The NUC runs the flight stack software [20] and all high-level algorithms.

The UAV uses ArduPilot's Copter firmware[1] as a low-level controller, a robust open-source platform that supports applications ranging from FPV racing to fully autonomous missions. The entire package is designed to be safe, feature-rich, and open for custom applications. The copter firmware is loaded onto the Cube Orange autopilot[2], which contains a 32-bit ARM Cortex M7 core processor with a fail-safe 32-bit coprocessor, 1 MB RAM, and 2 MB flash memory. Integrated into the Cube is a triple-redundant inertial measurement unit (IMU) with an accelerometer, a gyroscope, a magnetometer, and two barometers, which are isolated and damped.

The UAV is equipped with a downward-facing Intel RealSense D435[3] camera for takeoff and landing, using

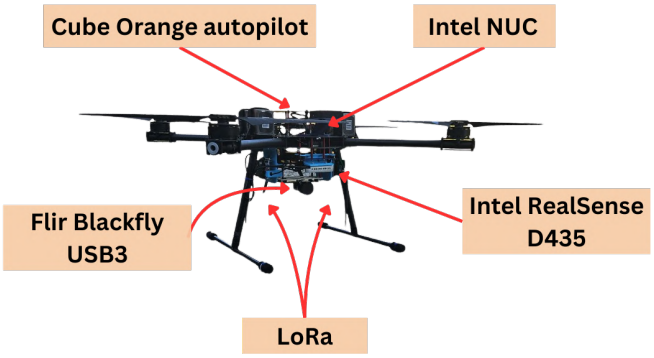


Fig. 2: Custom UAV hardware setup designed for search over large featureless areas without GNSS. The UAV is equipped with Cube Orange Autopilot and Intel NUC that handle control, state estimation and navigation, down facing RGB-D camera (Intel RealSense D435) for takeoff and landing using feedback from a custom marker, forward-facing camera (Flir Blackfly USB3) for target detection and localization, and two Semtech SX1280 LoRa modules for ranging and UAV localization.

TABLE I: MBZIRC UAV technical specification details.

Feature	Search UAV	Pickup UAV
Length	0.8 m	0.6 m
Width	0.8 m	0.6 m
Height	0.55 m	0.65 m
Weight	8 kg	4 kg
Maximal payload	7 kg	3 kg
Motor power	4×1.8 kW	4×0.9 kW
Battery system	12S LiPo 44.4 V	6S LiPo 22.2 V
Battery capacity	12,000 mAh	12,000 mAh
Max vertical speed	7 m/s	7 m/s
Max horizontal speed	20 m/s	12 m/s

feedback from a custom marker, a forward-facing FLIR Blackfly USB3[4] camera for target detection and localization, and two Semtech SX1280 LoRa[5] modules for ranging and UAV localization. The entire aerial platform is powered by two 12S 12,000mAh LiPo batteries, which enable a flight time of up to 30 minutes. Relatively long flight time makes this platform ideal for performing target vessel search missions.

The object pickup UAV is a scaled-down version of the search UAV, with specific hardware to enable pickup tasks, sharing a similar design. The Cube Orange autopilot and Intel NUC 13 are the same as described for the search UAV. Due to a smaller scale, this platform uses four T-motor MN4014 KV 400 6S motors with 17x4.0 carbon fiber propellers and T-motor F35A ESC, giving it a maximum takeoff weight of 7 kg with 3 kg payload capacity. The object pickup UAV is equipped with a downward-facing Intel RealSense D435 camera for object detection, mounted on a custom-built robotic gripper for object pickup. As discussed in Sections IV-E and IV-F, two different objects had to be picked up, requiring slightly different gripper designs. The small object gripper features two Dynamixel XM430-W350-R motors, while the

[1] https://ardupilot.org/copter/
[2] https://docs.cubepilot.org/user-guides/autopilot/the-cube
[3] https://realsenseai.com/products/stereo-depth-camera-d435/
[4] https://www.teledynevisionsolutions.com/en-150/products/blackfly-s-usb3
[5] https://www.semtech.com/products/wireless-rf/lora-connect/sx1280

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

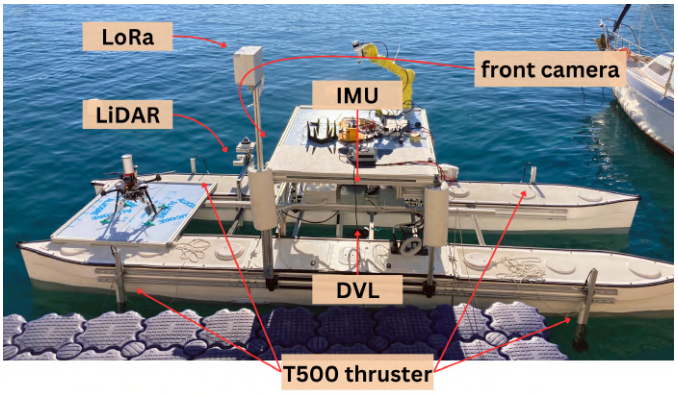


Fig. 3: Custom ASV hardware setup featuring additional sensors and actuators for enhanced navigation and perception capabilities. The IMU and DVL are used for long-range navigation over a GNSS-denied environment, while LoRa modules provide a fixed reference point for locational coordination with UAVs. LiDAR is used in the target vessel approach and docking, supported by additional T500 thrusters to enable fine maneuvering.

large object gripper features two Dynamixel MX-64 motors, which in both cases open and close the gripper when required. The object pickup UAV is powered by a single 6S 12,000mAh LiPo battery, achieving up to 10 minutes of flight time, making the platform suitable for object pickup. Detailed specifications for both UAVs are given in Table I.

B. Hardware setup of the ASV

All MBZIRC finalist teams received the same ASV without sensing or autonomy capabilities, which was designed as a symmetrical catamaran with modular dimensions, as shown in Figure 3. The ASV configuration was 2.5 to 3 meters wide, 7 meters long, and weighed approximately $500\,\mathrm{kg}$. Thanks to the modular design, additional equipment could be easily attached, allowing a maximum payload of $600\,\mathrm{kg}$. Each catamaran hull was originally equipped with a propulsion system with two $4\,\mathrm{kW}$ Torqueedo thrusters, which could deliver a thrust of up to $100\,\mathrm{kg}$. The thrusters enabled a maximum dynamic rotation velocity of $15\,^\circ/\mathrm{s}$, powered by a battery system operating at 44 to $58\,\mathrm{V}$. In the standard configuration, the vehicle could reach a cruising velocity of 4 knots ($\sim 2\,\mathrm{m/s}$) and a maximum velocity of up to 8 knots ($\sim 4\,\mathrm{m/s}$). The manual control system consisted of a remote control (RC), which was connected to a CAN master and enabled precise steering and control of the vehicle. All specifications are listed in Table II for convenience.

To achieve precise positioning, which is crucial for navigation and especially during the docking procedure, an additional set of four Blue Robotics T500 thrusters[6] was installed in an "X" configuration. The four additional T500 thrusters were integrated to complement the Torqeedo units, whose firmware introduces a latency of about $2\,\mathrm{s}$ into the control loop, which limits precise maneuvering at low speed.

[6]https://bluerobotics.com/new-product-the-t500-thruster/

TABLE II: MBZIRC catamaran ASV technical specification details.

Feature	Specification
Length	$7\,\mathrm{m}$
Beam	$3\,\mathrm{m}$
Weight	$500\,\mathrm{kg}$
Maximal payload	$600\,\mathrm{kg}$
Motor power	$2 \times 4\,\mathrm{kW}$
Thruster dynamic speed of rotation	$15\,^\circ/\mathrm{s}$
Battery system	44 to $58\,\mathrm{V}$
Cruise speed	4 knots ($\sim 2\,\mathrm{m/s}$)
Maximum speed	8 knots ($\sim 4\,\mathrm{m/s}$)

This setup, inspired by [21] with omnidirectional surface vessels, enables movement in the horizontal plane in any orientation and allows advanced control of the catamaran to easily maintain position during station keeping or docking, even in challenging weather conditions. These capabilities were essential to enable the recovery of objects with a robotic arm onboard the ASV.

The navigation system was developed using state-of-the-art technology [22] based on the fusion of DVL (Doppler Velocity Logger) and IMU (Inertial Measurement Unit) sensors to achieve highly accurate relative positioning from the catamaran's starting position, which is defined as the origin. Specifically, the system used a NavQuest 600 Micro DVL[7] and an Xsens MTi-630R IMU[8]. This combination of sensors ensured precise path following throughout the mission. The DVL determines the vessel's speed over the seabed using the Doppler effect, while the IMU captures accelerations and angular velocities for improved navigation accuracy.

The perception system of the ASV relied on a high-resolution OS1-128 Ouster LiDAR and the FLIR Blackfly USB3 color camera fitted with a RICOH VGA lens[9]. The lens, with a focal length of $50\,\mathrm{mm}$, provided an exceptional zoom with horizontal field of view of $14.4°$ and a vertical field of view of $11.0°$. These sensors enabled effective obstacle and object detection and collision avoidance. The camera using this lens provided high-resolution visual data over long distances, which was crucial for target detection and visual servoing. The LiDAR point cloud data proved invaluable for real-time obstacle detection and interaction with the target vessel during the missions [23].

The sensor data processing was distributed across two Intel NUC 13 computers, each running ROS2 Humble in Docker containers. One NUC was dedicated to navigation, while the other managed the perception sensors. This division ensured efficient data processing and optimized control of the vessel's subsystems, prioritizing autonomous navigation and obstacle avoidance at all times.

The custom-built hardware for docking to the target vessel consisted of two systems. Based on pneumatics, the first system comprised six suction cups distributed over two compartments. Each suction cup was connected to a vacuum chamber attached to the pump. Pressure sensors were placed

[7]https://www.link-quest.com/html/intro_nq.htm

[8]https://www.movella.com/products/sensor-modules/xsens-mti-630r-ahrs

[9]https://www.ricoh-iosd.eu/en/product/FL-BC5014A-VG/VGA-Lenses.html

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

directly in front of each suction cup, connected to the ROS2 and the vacuum pump via serial Arduino modules. The pressure sensors were critical indicators of contact between the ASV and the target vessel as the final approach to docking was performed in an open-loop procedure due to occlusions in LiDAR. The second system was based on a mechanical bridge with hooks that ensured a connection with the target vessel even if the pneumatic system lost contact. This way, a lost connection could be detected using the pressure sensors to trigger a fallback procedure for retrying the target vessel approach and re-establishing the pneumatic grip.

Another crucial component of the ASV system is the robotic arm, which was used to retrieve large objects from the target vessel. Given the physical and geometric constraints of the ASV and target vessel, as well as the estimated payload of the objects, we selected the Universal Robots UR10e arm. The arm was equipped with a custom parallel gripper powered by a linear actuator with claws, support legs, and compliance springs for stable grasping. A downward-facing Intel RealSense D435 RGB-D camera was integrated on the gripper for object detection and localization, while the arm's built-in force/torque sensor enabled compliant control during contact with the ship's deck. All components were powered and interfaced through the arm's tool supply, with control and perception running on the ASV's onboard computer. More details on the mechanical design of the gripper can be found in [24].

IV. Robot capabilities

In this section, the individual components of the system are described from the perspective of capabilities, and it is shown how each part contributes to the overall functionality. This includes a brief overview of the key algorithms, implementation strategies, and solutions developed to address the main challenges of the task.

A. GNSS-denied navigation using LoRa modules

A global localization system using Long-Range Radio (LoRa) devices has been integrated into the system to provide a common reference point for all robots. Particularly, Semtech SX1280 transceivers are used. This LoRa-based system allows the robots to determine their global positions in relation to the starting gate, where four fixed anchors with a known geometric layout are installed. Detailed information about the implemented localization system can be found in [15].

During localization, the robots send pings to the base station devices, which measure the range using Two-Way Time Travel (TWTT). The robot positions and the corresponding covariance measurement matrices are then obtained by performing trilateration with four anchors and solving the associated optimization problem. The optimization problem is limited to 2D and determines the robots' height (z component) using a barometer to make it tractable.

A partially unknown system horizontal location matrix is defined as:

$$\underset{(N_a+N_r)\times 2}{\mathbf{X}^{\text{sys}}_{\text{LoRa}}} = \begin{bmatrix}\mathbf{X}^{\text{a}}_{LoRa} & \mathbf{X}^{\text{r}}_{LoRa}\end{bmatrix}^T, \tag{1}$$

where $\mathbf{X}^{\text{a}}_{\text{LoRa}}$ is the 2D location matrix of N_a number of anchors with known locations, $\mathbf{X}^{\text{r}}_{\text{LoRa}}$ is the unknown 2D location matrix of N_r number of robots consisting of UAVs and ASV(s) with unknown locations.

Localization succeeds if each robot determines its horizontal position in a common reference frame. The common reference frame $\mathbf{L}_{\text{LoRa}}$ is an arbitrary East-North-Up (ENU) fixed frame in which known $\mathbf{X}^{\text{a}}_{\text{LoRa}}$ is represented.

The localization of the swarm of N_r robots relative to N_a localization anchors requires that each robot performs N_a range measurements, resulting in $N_r \cdot N_a$ ranging acquisitions. Robot-to-robot ranging enhances precision but requires $\frac{N_r(N_r-1)}{2}$ additional acquisitions. Therefore, it was not used during the competition but was implemented in the simulation phase of the competition [25].

Given the size of the competition area of approximately 2x2 km, consistent range measurements at a distance of $2\,\text{km}$ are desired. The $N_a = 4$ localization anchors are placed in the start gate area to increase the probability of successfully acquiring at least 3 ranges required for the 2D trilateration algorithm. The planned number of robots was in a range of $N_r = \{1, \ldots, 5\}$, consisting of $0-1$ ASV and $1-4$ search drones. The planned update rate of the localization system is $1\,\text{Hz}$.

1) Range Acquisition: A ranging procedure executes N exchanges to produce a single range measurement, where channel-hopping helps counteract environmental effects like multipath interference and frequency-dependent signal fading [26], [27]. It consists of a request-response phase ($2 \cdot T_{comms}$) and N exchanges ($N \cdot T_r$), where T_{comms} is the request-response duration and T_r is the duration of one exchange. A precaution is implemented to accommodate uncertainty in clock drift and packet drop, where each module waits for a fixed time T_{rw} between each range operation. Duration T_{comms} is commensurate to duration T_r so total time can be approximated by $\hat{T}_t = (N+2)T_{rw}$. Finally, the ranging cycle for N_a anchors and N_r robots is $T_l = \hat{T}_t N_a N_r$.

Simultaneous ranging is achieved by offsetting the start channel in the channel hopping sequence for the anchor LoRa modules. By synchronizing the channel hopping sequences between the modules, two pairs initiating the ranging procedure simultaneously on different channels can avoid interference. With M modules on each robot, M measurements can be collected in a single ranging procedure. The smaller value between $N_r \cdot M$ and N_a determines the maximum number of simultaneous ranging measurements. The simultaneous ranging procedure effectively shortens the distance measurement duration T_l by $min(N_r \cdot M, N_a)$ times.

2) Localization Procedure: After ranging, each robot knows a part of the noisy metric distance matrix $\mathbf{D}$, consisting of known inter-anchor, inter-robot and anchor robot distances. The localization problem is reduced to 2D by estimating the robot's height from barometer sensor measurements. The localization algorithm used in the simulation phase of the competition [25] is based on stress majorization, a method commonly used to solve metric multidimensional scaling problems (mMDS). In the final phase, the algorithm was adapted to a setup where the vehicles range only with the

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

base station, without measurements between the vehicles. The estimation of unknown $\mathbf{X}^{sys}$ from measured $\mathbf{D}$ and known $\mathbf{X}^{a}$ can be written in the following form:

$$J = \arg\min_{\mathbf{X}^{r}} \sum_{i,j} w_{ij}(d_{ij} - ||\mathbf{x}^{i} - \mathbf{x}^{j}||)^2, \mathbf{x}^{i,j} \in \mathbf{X}^{\mathrm{r}}, \quad (2)$$

where d_{ij} is taken from matrix $\mathbf{D}$ and w_{ij} represents the weight. This problem is commonly solved with the Scaling by MAjorizing a COmplicated Function (SMACOF) algorithm [28]. Since the reference frames are not preserved during optimization, the estimated anchor positions are aligned using a Procrustes transformation ($\mathbf{T}^{\mathrm{LoRa}}_{\mathrm{smacof}}$) that fits the estimated anchor positions to the known locations of the fixed anchors:

$$\mathbf{X}^{\mathrm{sys}}_{\mathrm{LoRa}} = \mathbf{T}^{\mathrm{LoRa}}_{\mathrm{smacof}} \mathbf{X}^{\mathrm{sys}}_{\mathrm{smacof}} \quad (3)$$

The extracted robot positions $\mathbf{X}^{\mathrm{r}}_{\mathrm{LoRa}}$ from $\mathbf{X}^{\mathrm{sys}}_{\mathrm{LoRa}}$ are provided to the robot state estimator (standard Kalman filter). All details on state estimation and control of the UAV under LoRa-based localization can be found in [15].

B. Collision-free GNSS-denied navigation of the ASV

In addition to the global LoRa-based positioning system, a relative localization system for the ASV was developed. It was designed to achieve precise and reliable positioning in relation to the starting point of the ASV. This system combined the data from the DVL and IMU to accurately track the movements of the ASV [22]. This combination of sensors ensured accurate path-following throughout the mission.

The implemented path-following method is based on the Regulated Pure Pursuit (RPP) [29], the most suitable NAV2 controller[10], as it enables smooth and precise waypoint following with additional collision detection. The global path is replanned every 2 seconds, using the cost map that consists of the latest updates. Initially, RPP transforms the global path into the reference frame of the ASV. Then, based on the current velocity of the ASV, a lookahead point ("carrot point") is calculated, and the appropriate velocities to control the ASV are determined. The lookahead point is calculated dynamically based on the current velocity of the ASV and the configured lookahead distance D_L. First, D_L is calculated based on the current velocity v of the ASV and the time parameter T_L:

$$D_L = v \cdot T_L. \quad (4)$$

Once the distance has been determined, the lookahead point $P_L(X_L, Y_L)$ can be calculated so that the distance from the current position $P_A(X_A, Y_A)$ of ASV to P_L is equal to D_L:

$$\sqrt{(X_A - X_L)^2 + (Y_A - Y_L)^2} = D_L. \quad (5)$$

The controller can interpolate between the waypoints to determine an accurate lookahead target for smoother and more precise path tracking. In addition, the controller calculates the curvature k of the path at the lookahead point P_L at a fixed distance to regulate the velocity of the ASV based on the path curvature. This is to avoid overly aggressive maneuvering or instability. To achieve a smooth and precise stop as the ASV approaches each waypoint, the controller adjusts its behavior by, for example, reducing velocity or adjusting the heading to match the target orientation. The curvature k from the current position P_A to the lookahead point P_L is calculated as

$$k = \frac{2 \cdot y_L}{L^2}, \quad (6)$$

assuming that the ASV is at the origin of this frame. Finally, the linear velocity v and the angular velocity ω of the ASV can be calculated as

$$v = V_{desired}, \quad (7)$$

$$\omega = k \cdot v. \quad (8)$$

$V_{desired}$ is dynamically regulated forward velocity of the ASV, and is adjusted based on several factors: the local path curvature (tighter turns result in lower velocity), the distance to the final waypoint (velocity decreases as the goal is approached), and the presence of nearby obstacles detected by the LiDAR costmap. This adaptive scaling ensures smooth trajectory tracking and prevents overshoot in sharp turns or unstable motion near the goal. Additionally, if the reference yaw deviates significantly from the actual yaw, the vehicle will halt until alignment is corrected.

Although the system is capable of lateral movement, the kinematics of the differential drive vessel is used for path following. The Torqueedo motors were used for the main drive and were not rotated during navigation, while the T500 thrusters were used for turning maneuvers, allowing efficient and controlled steering throughout the mission.

Collision avoidance: During the path-following procedure, checking and avoiding possible collisions at sea is essential. The main function of this method is to assess the collision potential based on the position of the ASV, its linear and angular velocities, and a lookahead distance [29]. Future positions are projected onto a cost map, which is updated with LiDAR data and checked for potential collisions. The pose and velocity of the ASV are used to calculate its future position step by step, taking into account both linear and angular velocities. This process continues until the projected path exceeds the lookahead distance or a collision is detected. If a collision is detected, the controller can adjust its course to avoid it or even stop to prevent the accident [30]. To address dynamic obstacles and vessels detected by LiDAR, the navigation cost map is periodically cleared every 2 seconds. The full navigation and collision avoidance algorithm is presented in [31].

C. Target vessel detection

During the survey of the designated area, the UAVs and the ASV focus on locating the target vessel, which is a critical element of the mission. A few hours before the mission start, a couple of images of the target were provided to facilitate its identification, as shown in Figure 4. The identification module consists of three different components:

[10]https://docs.nav2.org/configuration/packages/configuring-regulated-pp

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

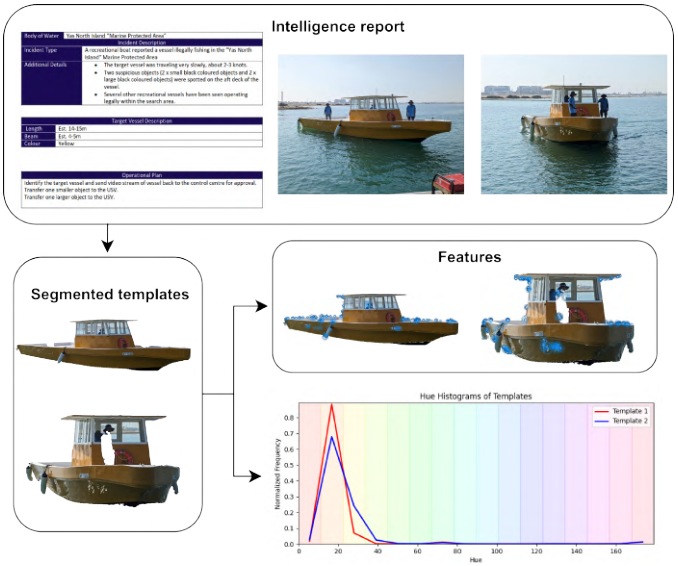


Fig. 4: The target identification module uses templates derived from the images in the intelligence report. It utilizes two metrics for matching: ORB feature-based matching and hue-histogram distance.

1) the **detection** of all maritime vessels within the competition arena, 2) the **identification** of the specific target vessel, and 3) the **localization** of the target vessel.

The YOLOv8 model is used for detection, which was trained with a collection of open-source and custom-generated synthetic datasets. YOLOv8 was chosen for its state-of-the-art accuracy at the time, real-time inference speed, and lightweight design, making it ideal for resource-constrained platforms such as robots with limited payload capacity. The YOLOv8 detections serve as input to the identification module, which uses two methods: feature matching and color histogram distance. ORB (Oriented FAST and Rotated BRIEF) features [32] are used to match the YOLOv8 detection against known target images, as shown in Figure 4. Next, the distance between the color histograms of the templates and the detected object is calculated, with the comparison shown in the bottom graph of Figure 4. The detected vessel is classified as the target if the resulting matching score exceeds a predefined threshold.

The identification module has two additional tasks: 1) providing a yaw angle reference to keep the target in the camera's field of view while waiting for confirmation of valid identification, and 2) transmitting the target's position to all robots in the system. For both tasks, the direction vector from the camera to the target is needed, expressed in the LoRa reference frame $\mathbf{L}_{LoRa}$. First, the direction vector $\mathbf{v}_{cam}^{vessel} = [x_{bb}, y_{bb}, f]$ is calculated, pointing from the camera center to the detected object with the image coordinates (x_{bb}, y_{bb}), where f is the focal length of the camera calibrated beforehand. The normalized direction vector is transformed into the LoRa coordinate frame:

$$\mathbf{v}_{LoRa}^{vessel} = \mathbf{R}_{LoRa}^{rob}\mathbf{R}_{rob}^{cam}\mathbf{v}_{cam}^{vessel}, \tag{9}$$

where $\mathbf{R}_{LoRa}^{rob}$ is the rotation matrix from the robot (UAV or ASV) body frame to the global frame, and $\mathbf{R}_{rob}^{cam}$ is the rotation matrix from the camera frame to the robot body frame. From the direction vector, the required robot yaw angle is calculated relative to the LoRa frame:

$$\psi_t = \arctan 2(\mathbf{v}_{LoRa,y}^{vessel}, \mathbf{v}_{LoRa,x}^{vessel}). \tag{10}$$

Once the target has been confirmed externally by the organizers, the position of the target vessel can be determined by the intersection of the line $\mathbf{p}_{LoRa}^{cam} + t \cdot \mathbf{v}_{LoRa}^{vessel}$ from the camera position $\mathbf{p}_{LoRa}^{cam} = (cam_x, cam_y, cam_z)$, along the vector $\mathbf{v}_{LoRa}^{vessel}$ and the plane in which the center of the vessel lies $z = h$. A constant z component of the vessel's position is assumed, representing the height h above the sea surface, an assumption made based on the approximate known dimensions of the vessel. By assigning an empirical value to h, the x and y components of the position can be calculated using:

$$\begin{bmatrix} cam_x \\ cam_y \\ cam_z \end{bmatrix} + t \cdot \begin{bmatrix} \mathbf{v}_{LoRa,x}^{vessel} \\ \mathbf{v}_{LoRa,y}^{vessel} \\ \mathbf{v}_{LoRa,z}^{vessel} \end{bmatrix} = \begin{bmatrix} x_{LoRa}^{vessel} \\ y_{LoRa}^{vessel} \\ h \end{bmatrix} \rightarrow t = \frac{h - cam_z}{\mathbf{v}_{LoRa,z}^{vessel}}. \tag{11}$$

Inputting t calculated from the z component into the x and y parts of (11) yields the global position $x_{LoRa}^{vessel}, y_{LoRa}^{vessel}$ of the target in the LoRa frame.

D. ASV docking procedure

The proposed algorithm relies solely on LiDAR feedback for the approach phase to effectively handle challenging tasks such as docking to a target vessel without reliable localization data, such as GPS measurements. It uses point cloud data to determine a suitable docking position and target vessel heading. LiDAR provides precise relative positioning of the target vessel and identified key features required to determine the optimal docking point.

1) Point Cloud filtering: In the initial phase of the algorithm, basic filtering was required to reduce the number of points to maintain the computing power needed to complete the mission and minimize the possibility of false results due to irrelevant areas in the point cloud. To this end, a voxel filter with an empirically determined leaf size based on preliminary trials effectively reduces the number of points while preserving the essential details that define the boat as a single object. Using a too large leaf can result in sparse objects being split into multiple segments, which is undesirable. A filter mask is also created to remove the static ASV features from the LiDAR frame and hide irrelevant areas, such as very low and high regions, and the back of the ASV, where foam often forms. Including large foam segments in the point cloud could lead to clustered objects and affect the performance of the algorithm.

2) Clustering and tracking: Basic clustering methods in the LiDAR frame have proven adequate for detection algorithms in the open sea [33], primarily due to the invisibility of the sea surface in the LiDAR image. The clustering algorithm employed in this paper is Euclidean Cluster Extraction, which utilizes K-Dimensional (KD) trees to find the nearest neighbor [34]. This approach demonstrates robust performance and improved processing speed compared to other common methods (K-means, DBSCAN, and Super-voxel). It achieves

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

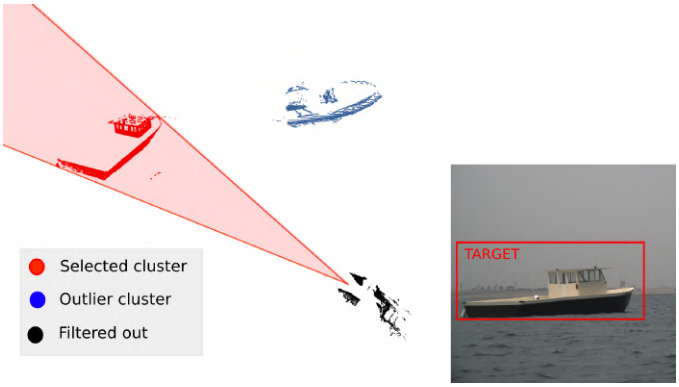


Fig. 5: Selection of the target object from the LiDAR frame involves filtering out undesired points represented by black dots in the LiDAR frame. Using the bounding box generated by the camera detection algorithm, the cluster representing the target vessel can be accurately identified (colored red in the image). It's important to note that the boat represented in blue is a decoy vessel and should not be selected as the target.

a processing time of $16.2\,\mathrm{ms}$ per frame, outperforming the next fastest method, DBSCAN, which required $17.8\,\mathrm{ms}$, as reported in [35]. This method correctly clusters the entire sample dataset without prior knowledge of the number of clusters, unlike the other methods. However, it is worth noting that it relies solely on the distance between points. Therefore, if parameters are not accurately set, a single vessel can be incorrectly clustered into two objects (e.g., the hull separated from the cabin), or two closely positioned objects can be incorrectly clustered together. Nevertheless, such cases are rare and more relevant in navigation scenarios involving heavy traffic in narrow marinas, which is not applicable in this context, where additional processing is unnecessary.

After segmenting the clusters, tracking was performed between frames to maintain the continuity of the objects. This procedure was applied to all clusters in the image and allowed the identification of classified objects, such as a target vessel, a decoy vessel, or any other object of interest. Tracking was based on Euclidean distance and the size and shape of each object. For each cluster, the centroid was calculated as a representative position. The centroid, shape, and size were then stored and compared across frames. The clusters were matched against predefined thresholds for distance, shape difference, and size variation to ensure accurate tracking across frames.

3) Target selection: The target is selected after all objects in the area have been tracked. If the target was present in the previous frame with sufficient confidence, matching the object in the new frame with the previous frame can reliably identify the target without additional target selection algorithms. However, if the target has been lost for some frames, e.g., due to occlusion or initial detection, a more advanced target selection approach is used.

First, an attempt is made to recognize the object in the camera image. Then, a frustum is created from the bounding box detection. The Intersection over Union (IoU) approach is used to select the cluster within the camera frustum, as proposed in the article [33]. This method ensures that the correct object is tracked in the LiDAR frame based on the information from the camera detection. In cases where the target boat has been lost but camera detection is not available, a larger frustum (e.g., 45 degrees width) is created based on the last known position of the target object. The same IoU criteria are applied to identify the object within this frustum. The masking, clustering, tracking, and target selection are visualized in Figure 5.

4) Orientation estimation: A principal component analysis (PCA) method is employed to estimate the heading of the target vessel from a 3D point cloud. Initially, the target boat cluster's centroid is determined, followed by computing the covariance matrix of the points within the cluster. Its eigenvalues and eigenvectors, often referred to as principal values, not only represent the boat's orientation but also serve as an indicator of any irregularities in the cluster shape that could potentially impact the accuracy of the orientation estimation.

5) Docking position estimation: Estimation of the docking position is one of the critical aspects of the mission, enabling the precise positioning of the ASV for mission continuation. This includes detecting the hull location to determine the more adequate docking side. Given the target cluster, the methodology involves dividing the point cloud into two distinct sections and computing the average point height for each segment. After identifying the side with the cabin, the docking point is adjusted by 25% of the total cluster length to the stern side as the objects to be picked up are on the deck behind the cabin. This offset from the center is calculated only once when ASV aligns with the target vessel's heading. To avoid approaching the vessel from the front or back, the ratio between principal values in the cluster must be larger than the fixed threshold value, which is deduced by the approximate ratio between the target length and width.

6) Control system and navigation: The docking phase begins with a $45°$ turn of the ASV on the port side to ensure better LiDAR coverage of the target vessel. At a distance of $50\,\mathrm{m}$ from the center of the target vessel, a circular path is planned, which the vehicle follows at a constant speed of $1\,\mathrm{m/s}$ to find an adequate relative position to the target vessel, from which the docking approach starts. Once a suitable position is reached, the ASV enters dynamic positioning mode and aligns the course to the target at the fixed position $10\,\mathrm{m}$ from the center of the target vessel cluster to identify the cabin and find the reference point for docking. During the approach to the target vessel, the ASV position controller is provided with a reference docking target point, which is calculated based on the shape of the target cluster.

Since the LiDAR feedback is strongly occluded when approaching the target vessel sideways, the consistency of the docking reference point calculation is compromised. In the field trials, the occlusion becomes unmanageable at a distance of $3\,\mathrm{m}$ from the center of the target vessel cluster. When the threshold value of $3\,\mathrm{m}$ is reached, the control system switches to an open-loop control and stops as soon as the pressure difference at the suction cups is measured. As this is an open-loop control, a timeout of 30 seconds is set from the start of

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

the control to complete the docking process. An unsuccessful docking attempt triggers a fallback procedure that returns the ASV to the $10\,\mathrm{m}$ distance from the center of the target vessel cluster for the next docking attempt. More details on the implemented docking algorithm are available in [36].

E. Small object pickup

The autonomous marsupial system shown in Figure 13 consists of an ASV, a UAV equipped with a robotic gripper, and a robotic manipulator (UR10e). The ASV serves as the mothership and the base reference coordinate system for the entire team for the subtasks of intervention/manipulation, with its center of mass set as the origin. This setup is intended to compensate for the displacement caused by wave-induced pitching and rolling of the vehicle.

The robotic manipulator is rigidly connected to the ASV and equipped with a LiDAR sensor, a camera, and a gripper as its end effector. The LiDAR tracks the UAVs, providing essential feedback that is backpropagated within the reference frame of the ASV. Continuous tracking is crucial to provide precise pose and velocity information to the aerial robots. To ensure the UAV remains within the LiDAR's line of sight, the manipulator adjusts its configuration and keeps it centered in the LiDAR's field of view.

The design of the aerial robots features a quadrotor configuration, with a gripper securely attached to the base plate. The grippers on both UAVs were developed based on concepts from [37] and are specifically tailored to the payload and dimensions of different objects.

An Intel RealSense D435 RGB-D camera, mounted at the gripper's central attachment point and pointing downwards, provides depth information to estimate the 3D object pose from 2D images. For the object pickup, the Position-Based Visual Servoing (PBVS) approach is implemented. For a successful pickup, the UAV must first determine the 4-DoF pose of an object (3D translation and yaw rotation) in the base frame, neglecting roll and pitch rotations as the object is assumed to lie on a flat surface. The object is segmented using the YOLOv8 neural network trained on a custom dataset of real and synthetic images. Details about the object pickup can be found in [38].

Precise relative localization of aerial robots: Accurate and robust localization is essential for UAV-based object pickup tasks. To address this requirement, a ρLiRLo (Reflectivity-based **Li**DAR **R**elative **L**ocalizati**o**n) is proposed, where ρ stands for reflectivity, a crucial property of retro-reflective markers that enhances point cloud filtering and clustering. In the proposed setup, a passive retro-reflective marker is mounted on top of a UAV, and it is tracked using a LiDAR sensor installed on a robotic arm aboard the ASV. The approach is outlined in the remainder of this section, with details provided in [39].

The ρLiRLo starts with acquiring the point cloud with intensity information. In the first step, *intensity filtering*, the LiDAR point cloud is pre-processed by retaining only points with intensities above a predefined threshold. *Clustering* step groups the remaining points based on their spatial distance. Since the number of UAVs is known in advance, the k-means clustering approach is used by setting k equal to the expected number of UAVs and defining the set of cluster centroids. Each point is assigned to the nearest cluster based on Euclidean distance. The k-means algorithm minimizes the within-cluster sum of squares (WCSS) to find an optimal centroid value for each cluster.

The centroid of the marker's cluster in the LiDAR coordinate frame represents the UAV's position. Placing a retro-reflective marker on the UAV isolates the marker while filtering out irrelevant points. This is particularly beneficial when the UAV operates close to other objects or surfaces, as it prevents the UAV and its surroundings from being falsely clustered together. As the cylindrical shape of the marker does not provide a way to track the orientation, the UAV's integrated IMU is used to estimate the heading.

The third step in the aerial robot's position-tracking system is to use a linear Kalman filter to *track the trajectory* of the detected centroid. The Kalman filter ensures robust tracking even if the UAV is not detected in every frame due to occlusions or temporary sensor limitations. The filter serves two important purposes: 1) smoothing noisy detection data and 2) predicting the UAV state when there is temporarily no detection data.

The final step in the localization process is to actively track the detected marker by adjusting the configuration of a UR10e robotic manipulator so that the object is centered within the LiDAR field of view. The manipulator mounted on the ASV increases the effective tracking volume and helps to prevent loss of tracking when the UAV is moving. Without this adjustment, the UAV's operational range would be limited to the LiDAR's fixed field of view, increasing the risk of tracking loss when approaching the edge of the scan area.

Applying the visual servoing approach [40] to the LiDAR tracking problem, the manipulator focuses on maintaining the marker's position in the center of the LiDAR field of view by adjusting its orientation while imposing minimal constraints on its position. This strategy improves both tracking reliability and operational flexibility. The Ouster OS0-128 LiDAR offers a 90° vertical field of view, while the UR10e provides six degrees of freedom, allowing precise and continuous view adjustment. Note that although an RGB-D camera is also mounted at the manipulator gripper center, it is not actively used to track the UAV; rather, it is used for large object pickup, discussed in later sections. In our experience, the camera-based detection did not work reliably, so it was not considered as feedback for the UAV.

The UAV control formalism in this paper follows a cascade PID structure, where the inner loop controls velocity and the outer loop controls position. In addition, the controller supports the use of feedforward signals for velocity and acceleration, which are utilized through the smooth reference trajectory generated by the MPC tracker. The main purpose of the localization results of ρLiRLo is the possibility to use this information as feedback for the control of the UAV. Specifically, the output of the Kalman filter is used as position and velocity feedback. The detailed controller and MPC tracker description can be found in our previous work [20].

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

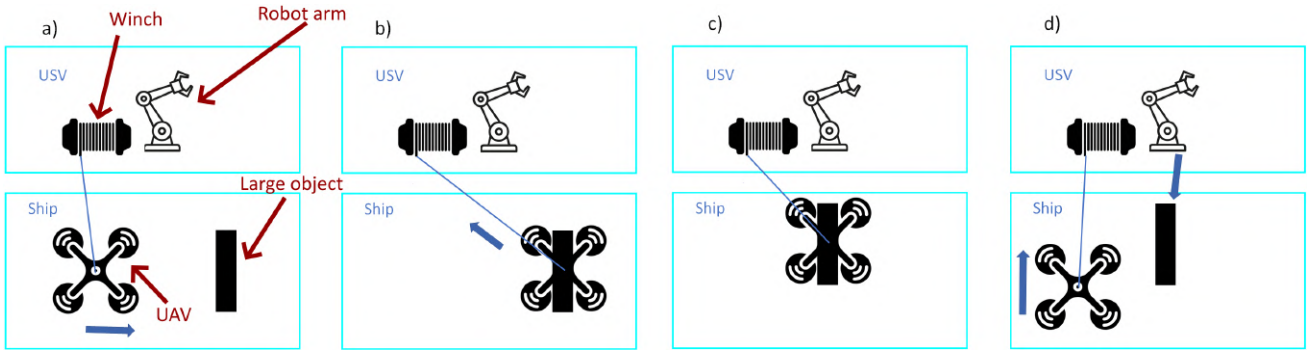


Fig. 6: Concept of the collaborative pick-up procedure: a) The large object pickup UAV searches for a large object above the ship deck; b) when the large object is found, the UAV lands on it and the winch of the ASV starts to pull the UAV together with the large object; c) when the UAV with the large object is close to the robotic manipulator, the winch stops and the UAV takes off; d) the UAV returns to the landing position and the robotic manipulator proceeds to pick up the large object.

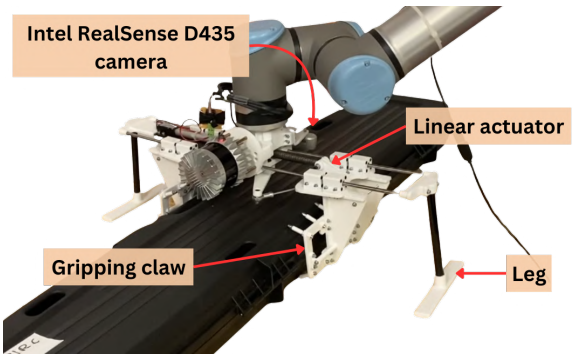


Fig. 7: The custom-built gripper on the robotic manipulator features stabilizing legs to maintain contact with the floating deck where the object is located, a gripping claw actuated by a linear actuator, and an eye-in-hand RealSense camera to guide the manipulator toward the object.

F. Collaborative large object pickup

Since the large object has dimensions of 132cm x 30.5cm x 9.5cm and a weight of over 3kg, it cannot be picked up by an individual UAV. For this purpose, a collaborative object pickup procedure is used. The procedure includes collaboration between a UAV, a robotic manipulator, and a winch mounted on the ASV, as illustrated in Figure 6.

The procedure starts similarly to the small object pickup described in Section IV-E, where the UAV takes off and searches for the object using its downward-facing camera. Once positioned above the object, the large object pickup UAV aligns its gripper to its orientation, lands, and closes the robotic gripper. Since the UAV cannot carry large objects independently, additional components had to be added to the system, namely a tether connecting the UAV to the winch located on the ASV next to the robotic manipulator.

While the UAV is searching for a large object in flight, the winch system is deactivated so that the UAV can extend the tether without restrictions. After the UAV lands on the large object and closes the gripper, the winch locks and starts to wind up the tether, pulling the UAV and the large object together towards the robotic manipulator. As the LiDAR on the robotic manipulator tracks the UAV, it can also determine the location of the large object simultaneously. When the large object comes within range of the robotic manipulator, the UAV takes off, and the robotic manipulator performs the pickup task, as described next. The location of the UAV before take-off is assumed to be the location of the large object.

To complete the retrieval task, the manipulator uses the RGB-D camera mounted on its gripper to scan the area around the large object's last known location, enabling precise object detection and localization. Once the object is located, the manipulator initiates the final approach and pickup operation.

Due to the influence of waves and wind, the relative pose of the object with respect to the robotic manipulator base experiences rolling and pitching motion, as well as vertical and, to a lesser extent, horizontal movements in both axes. Dealing with such a challenge requires a method for picking up objects that can perform well in such conditions. One such method (which is presented in detail in [24]) used in the proposed solution is based on the measurement of a force/torque sensor on the flange of the robotic manipulator.

The pickup procedure starts by aligning the custom-built gripper on the robotic manipulator directly above the large object. The gripper's design features two legs that must be in contact with the ship's deck, while the two arms of the gripper close around the object (see Figure 7). The robotic manipulator moves downwards until the gripper comes into contact with the ship's deck. From this moment on, the robotic manipulator controls force and torque so that the legs of the gripper maintain contact with the deck despite the influence of the waves, enabling the gripper to grasp the object correctly.

In this collaborative task, the distribution of control and perception responsibilities is clearly defined across the platforms. The ASV onboard computer is responsible for controlling the UR10e robotic arm, performing LiDAR-based object detection and tracking, operating the winch, processing RGB-D images for large object pickup, and executing the high-level state machine that coordinates the UAVs, winch, and manipulator. Each of the pickup UAVs runs its own onboard modules for control and navigation, gripper actuation for object pickup, RGB-D image processing for object detection,

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

and a dedicated state machine for the pickup task.

G. Communication overview

Both mission and system complexity required thorough consideration of the communication system. Figure 8 illustrates the main communication channels employed for the mission. The anchors at the base are used to provide localization for search UAVs and the ASV over the LoRa protocol at $2.4\,\mathrm{GHz}$. To avoid interfering with the localization, each UAV is equipped with a TBS Crossfire Nano RX at $868\,\mathrm{MHz}$ for manual control in the event of a pilot takeover. For the small object pickup, a local WiFi network supporting ROS communication is established between the ASV and the pickup UAVs. Again, to avoid interfering with the localization on the ASV, this network is set to operate on the $5.8\,\mathrm{GHz}$ WiFi band. Additionally, the ASV also housed two network switches. The Ethernet network established through these switches handled high-bandwidth data from two LiDARs, as well as communication with other sensors and the robotic manipulator.

For long-range, inter-robot communication, we use the Digi XBee XR 868[11] family of RF (radio frequency) modules. The RF modules operate in the 863-870 MHz range but implement LBT+AFA (Listen Before Talk and Adaptive Frequency Agility) to avoid interference with manual remote controllers and among themselves. Despite their small size and only 13 dBm ($20\,\mathrm{mW}$) of transmit power, thanks to smart proprietary modulation, they can achieve a line-of-sight range of up to $14\,\mathrm{km}$. Additionally, they support DigiMesh® network protocol that enables out-of-the-box multi-hop routing, ad-hoc network discovery and creation, self-healing, and reliable message delivery. With this setup, each robot in the system acts as a relay node, further strengthening and increasing the communication range beyond line-of-sight specifications.

To simplify development and usage, we developed a bi-directional ROS-XBee bridge. The software[12] connects to the XBee module using a serial USB connection and encodes/decodes messages coming from/to other ROS nodes running on the robot (e.g., mission state, vessel detection). The software supports both unicast and broadcast communication, ROS and ROS 2 middleware, and boasts automatic topic discovery and message queues to prevent loss of important messages. One additional XBee module is connected to the base station laptop to transmit the ”start mission” command and monitor mission execution.

As a consequence of the long range and low power, the maximum data rate is limited to 80 kbps, which is not enough for transmitting video. Instead, we opted for an analog video streaming common in the FPV (first-person-view) drone community. We use a digital-to-analog video converter to process video from the Intel NUC and transmit it via TBS Unify Pro32 HV VTX[13] using a circularly polarized omnidirectional antenna. This video transmitter operates in the $5.8\,\mathrm{GHz}$ band with 1000 mW output power. The receiver was provided by the organizers, and we equipped it with a pair of omnidirectional and 8 dBi patch antennas. Since only one video stream was allowed at a time, the robots negotiated which transmitter to turn on over the XBee inter-robot communication.

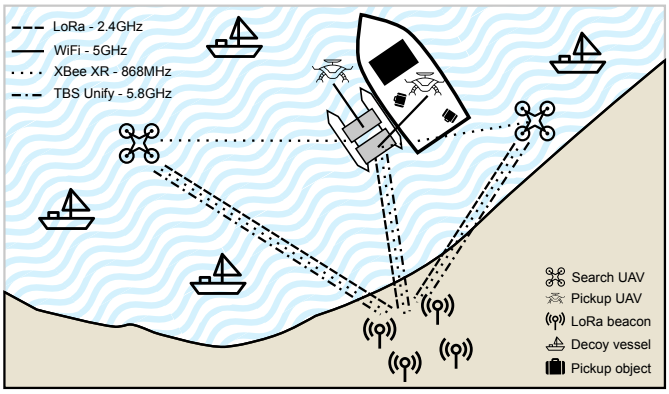


Fig. 8: The overall system main communication overview. Four beacons at the base provide localization over the LoRa protocol for the search UAVs and the ASV. Additionally, XBee mesh communication enables long-range mission-critical communication at $868\,\mathrm{MHz}$, while the video is streamed over TBS Unify at $5.8\,\mathrm{GHz}$. Locally, the pickup UAVs are connected to the ASV's WiFi network at $5.8\,\mathrm{GHz}$.

Apart from all the above-described communication interfaces, the organizer supplied a black box with a separate communication channel required for all robots. The main task of this box was to shut down all motors on UAVs and thrusters on the ASV in the event of a severe emergency. The specifics of this system were not disclosed to contestants.

V. Experimental Results and Evaluation

In the MBZIRC competition, we have chosen a pragmatic and robust approach in which the ASV is responsible for searching and identifying the target vessel. Given the logistical constraints and the risk of only being able to perform three full, uninterrupted mission attempts, the search UAVs remained on standby for possible deployment near the end of the time slots in case the ASV struggles to identify the target vessel. This paper focuses on the results obtained during the competition, while the outcomes of the UAV search mission are presented in [15].

Scores were awarded based on task completion and time efficiency. Points were given for successfully identifying the target, capturing small objects, and completing the entire mission (picking up a large object) – in that order. The entire mission had to be done fully autonomously, without manual interventions or re-runs in case some components fail. Our proposed strategy prioritized safely reaching the object pickup stage, accepting a potential compromise in search speed by not deploying UAVs at the start of the mission.

[11] https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/sub-1-ghz-rf-modules/digi-xbee-xr-868

[12] https://github.com/mkrizmancic/ros2_xbee_bridge

[13] https://www.team-blacksheep.com/products/prod:unifypro32_hv

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

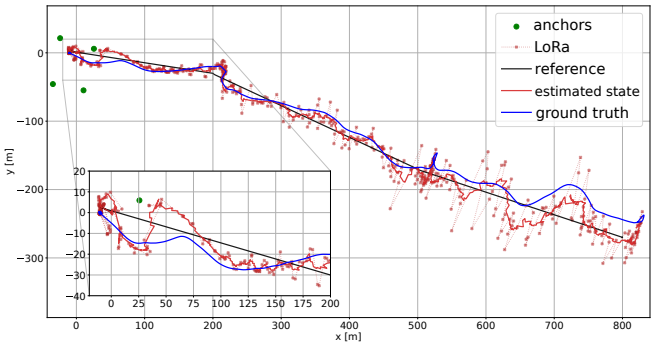


Fig. 9: The path of the UAV in the long-range experiment, comparing the localization with the ground truth. The localization error and variance increase as the UAV moves further from the anchors. Ground truth data from GNSS measurements is shown in blue. Raw locations based on LoRa measurements are depicted as a red x-dotted line, the estimated state is in a red full line, and the reference is shown as a black line.

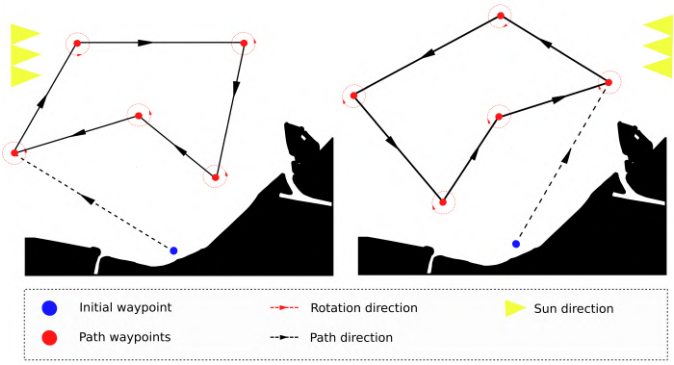


Fig. 10: The predefined path consisted of six waypoints: an initial waypoint to start the mission and the remaining waypoints forming a loop. The spinning procedure is initiated at each waypoint with a detection algorithm running in the background. Two path plans were created to ensure the ASV does not steer directly into the sun. If the target is not detected at the last waypoint, the ASV returns to the first waypoint and repeats the loop until the target is successfully detected.

A. GNSS-denied navigation using LoRa modules

To demonstrate the UAV localization with LoRa modules, an experiment is performed where the UAV is traversing approximately $800\,\mathrm{m}$ from the starting point, located in the area bounded by the fixed anchors. The experiment is shown in the Figure 9. The UAV is flying at an altitude of 50 m with a velocity of $2\,\mathrm{m/s}$. The path includes several predefined waypoints where the UAV hovers to survey the area. After reaching the furthest point, the UAV successfully navigates back to its home position. This experiment was done during the testing phase, when GNSS measurements were still allowed to obtain ground truth data. As shown in Figure 9, the standard Kalman filter used for state estimation reduces the noise significantly. Throughout the flight, the estimated position remained mostly within $30\,\mathrm{m}$ of the ground truth, demonstrating good localization and estimation performance. Larger errors were observed, particularly when the UAV moved further away from the anchors and the convex area formed by the anchors. This experiment showcased the reliability of the localization system. Despite the increases in localization error, the system consistently maintained a stable flight and returned to home position. Full results of the system's performance can be found in [15].

B. Target vessel search

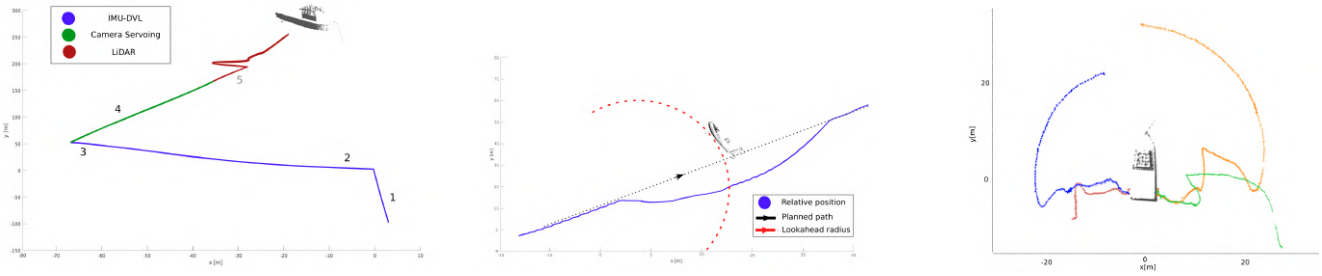
The ASV uses a predefined waypoint plan with six waypoints, as shown in Figure 10. This number of waypoints covers the entire search area as the integrated lens provides a high fixed zoom, whose target identification algorithm proved reliable at distances up to $\sim 900\,\mathrm{m}$. There were two possible plans for each mission, differing only in the direction of the path. This adjustment accounted for the sun's direction, as improper planning could result in the sunlight shining directly into the camera, making it much more difficult to detect the target vessel.

Figure 11a illustrates the catamaran's relative position throughout the mission, divided into distinct phases corresponding to key mission segments and their associated localization modes. Initially, navigation relies on DVL–IMU relative localization. In phase 1, the ASV is initially in station-keeping mode and waits for the start command for the mission. Once the mission begins, the ASV moves forward to the first waypoint. This initial movement ensures that the ASV steers clear of the starting gate behind it. In phase 2, Regulated Pure Pursuit is used to track the path to the second waypoint. Since the mission path is predefined, this method has proven to provide a highly accurate path following with safe and stable movement.

To ensure the safety of the catamaran as it moves between waypoints in a dynamic environment with various decoy vessels, a collision avoidance algorithm using LiDAR runs in the background. The collision avoidance results are analyzed at the end of this section and shown in Figure 11b.

Phase 3 is initiated when each of the predefined waypoints is reached. Upon arrival, the ASV performs a $360°$ in place spin, with the detection algorithm running in the background. In this test, the target vessel in Figure 11a was successfully detected from the first waypoint of the search trajectory. After detection, the ASV remains stationary, facing the target and streaming a video feed to the command center while waiting for confirmation.

The target confirmation initiates phase 4, triggering a switch from DVL-IMU localization frame to visual servoing procedure. In the ASV trajectory in Figure 11a, the approach to the target vessel is displayed from $\sim 210\,\mathrm{m}$ distance. Based on the visual detection, references are passed to the heading control to maintain the correct yaw angle toward the target. When the ASV approaches within $\sim 50\,\mathrm{m}$ of the target, the cluster of the target vessel is detected in the LiDAR, and visual servo control is terminated. In phase 5, the docking process is



(a) The mission path of the ASV is shown with numbered phases: Phase 1 is takeoff; Phase 2 is path-following with collision avoidance; Phase 3 is stopping and target detection; Phase 4 is visual servoing; Phase 5 is docking.

(b) The relative position of the ASV is shown with the blue line, while the planned path is shown with the black dotted line. It can be seen that the ASV deviates from the planned path to avoid a collision with the obstacle ship at the predefined lookahead distance.

(c) Four recorded relative docking paths of the ASV to the target vessel, performed during the final preparation at the competition site.

Fig. 11: Analysis of the ASV performance during the entire mission. The figure shows (a) the relative position of the ASV during the entire mission, (b) an instance of collision avoidance protocol, and (c) four recorded docking paths of the ASV, all performed during the field trials at sea.

continued with the help of the LiDAR feedback.

Before the competition, the proposed localization system was evaluated by comparing it with the GNSS ground truth on two paths. On a circular trajectory, the ASV returned to its starting point, showing that the drift decreases as it revisits previous areas. In contrast, a longer linear trajectory showed an increasing drift, resulting in a larger deviation from the GNSS path. The mean RMSE was $14.78\,\mathrm{m}$ for the circular path and $53.23\,\mathrm{m}$ for the linear path, both covering almost $700\,\mathrm{m}$.

Figure 11b shows that the ASV deviated from its planned path to avoid an obstacle ship, using point cloud data from the integrated LiDAR for safe navigation. The RPP algorithm enables path following and flexible trajectory adjustment based on sensor inputs. With a look-ahead radius of $25\,\mathrm{m}$ at $2\,\mathrm{m/s}$ (3.88 knots), the ASV updates its costmap in real time based on LiDAR data and the initial occupancy grid. This dynamic approach allows the catamaran to adapt quickly, minimizing the risk of collision while maintaining control, a crucial step toward autonomous navigation in unpredictable marine environments.

C. Docking

In total, 11 out of 12 docking maneuver tests were successful during the competition, each successful on the first try (without triggering the fallback procedures) and lasting about 2-3 minutes. The target vessel size and orientation estimate effectively determined the orientation of the docking maneuver, although minor adjustments were required near the vessel due to lower LiDAR accuracy. Since the intelligence report only approximated the vessel size (e.g., a yellow boat 14–$15\,\mathrm{m}$ in length), an analytical evaluation of the vessel length estimate was not possible due to the lack of ground truth data. Nevertheless, the length estimation was sufficiently accurate to ensure correct docking to the target vessel, so additional tasks, such as manipulating objects with a robotic arm, could be performed.

During the competition, all teams were allowed to use some of the target vessels for tasks and final preparations for the competition. As the practice time was limited, the teams had to use it efficiently to familiarize themselves with the target vessel and the docking procedure. Four docking attempts were recorded during this time, as shown in Figure 11c. The image shows the relative position of the ASV to the target vessel for each docking attempt. Two docking attempts were performed from the left and right sides to cover multiple launch positions. The video of the full search and docking mission of the ASV is available on YouTube at https://youtu.be/VrwuCkpoud4.

D. Target vessel identification

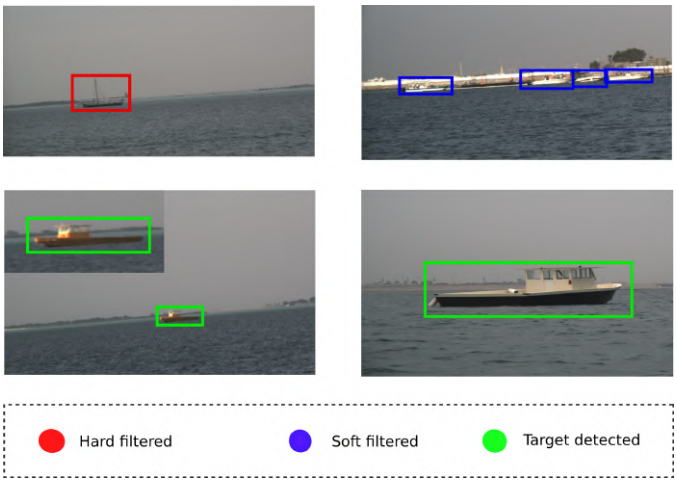


Fig. 12: In the upper left corner, the sailing ship is filtered out with a hard filter. In the upper right corner, a couple of decoy ships that have been filtered out using the soft filter method, as they may match the detection and hard filter from a distance, but do not belong in the HSV color ranges of the target ships shown in the lower left and right corners.

Figure 12 illustrates the detection results, where different vessels are filtered using hard and soft filtering methods. The hard filter eliminates vessels whose dimensions deviate from the required specifications. The soft filter further refines the

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

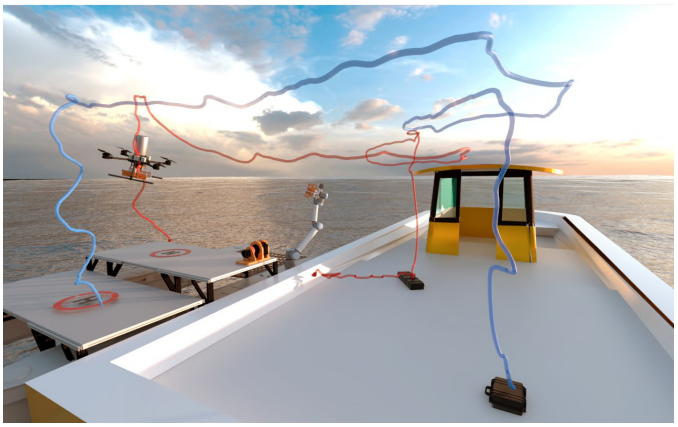


Fig. 13: The trajectories from the pickup task experiments visualized in the 3D environment, which was also used to train the segmentation modules by augmenting the dataset with realistic synthetic images. The blue line represents the trajectory of the first UAV, mapping objects and picking up a small object. The red line shows the trajectory of the second UAV, which is attached to the ASV winch via a thin line, and retrieves the large object by pulling it across the deck.

selection by removing false detections based on additional features such as color. This step is particularly effective in eliminating decoy vessels that may pass the hard filter but do not match the characteristics of the template image, as described in Section IV-C.

E. Small object pickup

Figure 13 shows the results of the pickup mission. As soon as the ASV has docked, the first UAV takes off and searches for objects of interest on the target vessel (blue trajectory in the image). It follows a predefined trajectory and maps the positions of the objects in the ASV base frame.

Once the search is complete, the UAV hovers over a smaller object and begins its approach using the visual servoing algorithm. It descends toward the object's centroid and begins landing when it is outside the camera's field of view. External disturbances, such as wind or prop wash, can lead to failed pickup attempts, causing the UAV to execute another attempt. After a successful pickup, the UAV takes off and transports the small object to the ASV. An analysis of the performance of the LiDAR-based relative localization method using a retro-reflective marker of the UAV can be found in [39]. The full video of the small object pickup is available on YouTube at https://youtu.be/l_ez7CVjDk4.

F. Collaborative large object pickup

The second pickup UAV, tethered to a winch, begins its mission as soon as the first UAV has landed on the ASV. It takes off and approaches the larger object using the object location map created by the first UAV (red trajectory in Figure 13). If the large object is not in the field of view, a local spiral search is executed until the object is detected. Following the same procedure for approach, landing, and grasping, it secures the object but remains tethered while the winch pulls it within the range of the manipulator arm. Once the UAV is close to the manipulator, it takes off and lands back on the ASV.

The robotic arm retrieval task begins by executing a predefined search trajectory, scanning the area with the RGB-D camera mounted at the center of the gripper. This step is essential, as the large object's final pose may vary depending on the dynamics of the winch pull.

After the large object is successfully detected in the camera image, the manipulator positions the gripper above it and decreases the height until the vessel deck is reached. As described earlier, force/torque sensing handles the relative movement between two vessels while the gripper closes and secures the object. Upon successful grasp, the object is transported and dropped in a net mounted on the ASV, successfully finishing the mission. The video of the full large object pickup mission is available on YouTube at https://youtu.be/efpaFqy4Hn8.

The proposed approach to the ship-to-ship large object pickup system is validated through controlled laboratory and real-world maritime experiments before the competition, as described in [24]. The gripper was tested for robustness against position and orientation errors. It could successfully grasp the object even with an offset of up to $\pm 20\,\mathrm{cm}$ in the long axis, $\pm 4\,\mathrm{cm}$ in the short axis, and yaw errors of up to $15°$. The force/torque-based servoing method effectively adapts to deck movements caused by waves, including roll and pitch compensation. Tests conducted on an autonomous catamaran at sea in varying light conditions and mild wave conditions (sea state 0–1) confirmed the reliability and robustness of the system, which repeatedly successfully picked up and transferred objects across different object placements. The video of the experiments is presented on YouTube at https://youtu.be/XpgYZ2Aideg.

G. Competition experience

The competition lasted four consecutive weeks. The first three weeks were dedicated to the final preparations and fine-tuning of the system, with limited daily testing time. In the last week, the five finalist teams carried out the official competition. Each team had three attempts, with no repeats allowed. The team set up and prepared their equipment for each attempt at the starting gate. There was then a 15-minute "hands-off" period during which no adjustments were allowed. During this time, the team waited to start the mission. One command was issued to launch the full, uninterrupted mission at the designated time. One of the biggest challenges for all teams was properly deploying the mission in its entirety under these very strict conditions, which are uncommon in these harsh experimental settings.

The mission was successfully launched in the *first run*, and the ASV began its search for the target vessel. The ASV managed to locate and identify the target. However, due to a setup error during mission preparation, the video stream of the detected target ship did not work, so no points were awarded.

The last two runs were conducted on the same day, as a bad weather forecast affected the schedule for the third run. These

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

consecutive runs only allowed a very short preparation time of just over half an hour to get ready for the final attempt. The *second attempt* resulted in a collision with the target vessel because the collision avoidance module had been inadvertently switched off during one of the tests. This resulted in one of the UAVs falling off the helipad, requiring a quick assessment and repair before the final attempt. The third and *final run* was our best achieving result: the ASV successfully identified the target in **7 minutes and 40 seconds**, taking nearly half the time of the second-placed team. The ASV also managed to dock to the target vessel autonomously. However, due to the earlier crash and limited setup time, the drone on board the ASV had a hardware malfunction and could not launch on its mission. It is important to note that the team was not allowed to debug or restart the UAV at this point, which ultimately prevented the completion of the pickup tasks. Additionally, no other team was able to complete the pickup task during the official runs or autonomously dock with the target vessel.

VI. Conclusion and Lessons Learned

This paper presents a flexible robotic system that operates in a maritime environment without GNSS to efficiently locate and identify a target vessel, retrieve, and transport objects from that vessel in an adverse sea environment. The system comprises an ASV and multiple UAVs to perform complex inspection and intervention tasks autonomously. The marsupial system configuration enables UAV deployment and retrieval from the ASV to combine the endurance and range of the ASV with the precision and maneuverability of the UAVs. We have demonstrated the system's capabilities in field trials involving vessel detection, precise docking, and object manipulation in challenging sea conditions. The system also demonstrated superior performance in the competition arena, securing a first-place finish among the finalist teams by completing the task in nearly half the time of the second-placed team.

The research presented in this paper lays the foundations for future maritime robotics developments that require autonomous operation without GNSS signals. The results show how combining multiple sensing technologies with advanced control and a reliable communication network enables state-of-the-art autonomous maritime operations. Possible future developments should focus on improving performance under severe conditions and optimizing the coordination of aerial and surface vehicles in joint complex missions.

Our competition experience shows how a competitive environment can provide important opportunities to showcase robotic innovations, but also how strict competition rules can prevent good performance. The limited testing time, one-chance missions, and long waiting times between mission setup and launch amplify the effects of small oversights that sometimes prevent teams from realizing the full potential of their systems. Many restrictions force teams to focus on dependable solutions and prepare conservatively rather than trying out innovative capabilities. Although stricter rules force a strong operational integration of the system, experience shows that progress in field robotics in harsh conditions also depends on learning from mistakes and adapting on the spot, which is not always possible given the rigid requirements. Nevertheless, we achieved all competition objectives during preparation, including autonomous target detection, docking, and coordinated ASV-UAV object recovery. In doing so, we have proven that heterogeneous robotic teams have the potential to realize complex maritime operations.

Acknowledgments

The authors would like to thank all members of LARICS and LABUST for their valuable discussions and contributions. We also wish to express our gratitude to the University of Zagreb Faculty of Agriculture and the University of Rijeka Faculty of Maritime Studies, Center for Marine Technologies, for providing access to their facilities during the competition preparations. Lastly, we extend our thanks to our sponsors and supporters.

References

[1] F. Marques, A. Lourenço, R. Mendonça, E. Pinto, P. Rodrigues, P. Santana, and J. Barata, "A critical survey on marsupial robotic teams for environmental monitoring of water bodies," in *OCEANS 2015 - Genova*, pp. 1–6, 2015.

[2] R. R. Murphy, E. Steimle, C. Griffin, C. Cullins, M. Hall, and K. Pratt, "Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane wilma," *Journal of Field Robotics*, vol. 25, no. 3, pp. 164–180, 2008.

[3] E. Pinto, F. Marques, R. Mendonça, A. Lourenço, P. Santana, and J. Barata, "An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 443–450, IEEE, 2014.

[4] M. Lindemuth, R. Murphy, E. Steimle, W. Armitage, K. Dreger, T. Elliot, M. Hall, D. Kalyadin, J. Kramer, M. Palankar, *et al.*, "Sea robot-assisted inspection," *IEEE robotics & automation magazine*, vol. 18, no. 2, pp. 96–107, 2011.

[5] C. Cheng, D. Liu, J.-H. Du, and Y.-Z. Li, "Research on visual perception for coordinated air–sea through a cooperative USV-UAV system," *Journal of Marine Science and Engineering*, vol. 11, no. 10, p. 1978, 2023.

[6] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6105–6110, IEEE, 2017.

[7] R. Mendonça, M. M. Marques, F. Marques, A. Lourenco, E. Pinto, P. Santana, F. Coito, V. Lobo, and J. Barata, "A cooperative multi-robot team for the surveillance of shipwreck survivors at sea," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–6, IEEE, 2016.

[8] G. Shao, Y. Ma, R. Malekian, X. Yan, and Z. Li, "A novel cooperative platform design for coupled USV–UAV systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4913–4922, 2019.

[9] N. Mišković, S. Bogdan, D. Nađ, F. Mandić, M. Orsag, and T. Haus, "Unmanned marsupial sea-air system for object recovery," in *22nd Mediterranean Conference on Control and Automation*, pp. 740–745, 2014.

[10] "Competition Spurs Robotics and AI Innovations for Maritime Challenges." https://spectrum.ieee.org/mbzirc-maritime-grand-challenge/about-mbzirc. Accessed: 2025-10-03.

[11] J. Yousaf, H. Zia, M. Alhalabi, M. Yaghi, T. Basmaji, E. A. Shehhi, A. Gad, M. Alkhedher, and M. Ghazal, "Drone and controller detection and localization: Trends and challenges," *Applied Sciences*, vol. 12, no. 24, 2022.

[12] A. Masiero, F. Fissore, A. Guarnieri, F. Pirotti, and A. Vettore, "UAV positioning and collision avoidance based on RSS measurements," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-1/W4, pp. 219–225, 2015.

[13] J.-L. Rullán-Lara, S. Salazar, and R. Lozano, "Real-time localization of an UAV using Kalman filter and a wireless sensor network," *Journal of Intelligent & Robotic Systems*, vol. 65, pp. 283–293, Jan 2012.

[14] T.-M. Nguyen, T. H. Nguyen, M. Cao, Z. Qiu, and L. Xie, "Integrated UWB-vision approach for autonomous docking of UAVs in GPS-denied environments," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9603–9609, 2019.

[15] M. Peti, L. Marković, I. Lončar, A. Barišić Kulaš, F. Petric, A. Milas, J. Goričanec, M. Car, M. Orsag, B. A. Ferreira, and S. Bogdan, "Aerial localization and navigation for surveillance of large, featureless, GNSS-denied maritime environments," *Journal of Field Robotics*, June 2025.

[16] R. Panish and M. Taylor, "Achieving high navigation accuracy using inertial navigation systems in autonomous underwater vehicles," in *OCEANS 2011 IEEE-Spain*, pp. 1–7, IEEE, 2011.

[17] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020.

[18] J. Han, Y. Cho, and J. Kim, "Coastal SLAM with marine radar for USV operation in GPS-restricted situations," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 300–309, 2019.

[19] L. E. Marquez and M. Calle, "Understanding LoRa-based localization: Foundations and challenges," *IEEE Internet of Things Journal*, vol. 10, no. 13, pp. 11185–11198, 2023.

[20] L. Markovic, F. Petric, A. Ivanovic, J. Goricanec, M. Car, M. Orsag, and S. Bogdan, "Towards a standardized aerial platform: ICUAS'22 firefighting competition," *Journal of intelligent & robotic systems*, vol. 108, no. 3, p. 52, 2023.

[21] Đ. Nađ, N. Mišković, and F. Mandić, "Navigation, guidance and control of an overactuated marine surface vehicle," *Annual Reviews in Control*, vol. 40, pp. 172–181, 2015.

[22] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[23] W. Zhang, F. Jiang, C.-F. Yang, Z.-P. Wang, and T.-J. Zhao, "Research on unmanned surface vehicles environment perception based on the fusion of vision and lidar," *IEEE Access*, vol. 9, pp. 63107–63121, 2021.

[24] G. Vasiljevic and D. Stuhne, "Robust ship-to-ship object pick-up with a 6-DoF robotic arm based on force/torque measurement and gripper design." In review for publication in Journal of Field Robotics, available online: 10.22541/au.172830862.29644392/v1, 2025.

[25] M. Peti, A. Milas, N. Kraševac, M. Križmančić, I. Lončar, N. Mišković, and S. Bogdan, "A search strategy and vessel detection in maritime environment using fixed-wing UAVs," in *2023 IEEE Underwater Technology (UT)*, pp. 1–8, 2023.

[26] SEMTECH, *Design of the SX1280 Ranging Protocol and Result Processing*, rev 1.1 ed., 2022. Available at https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000UypY/5mprGH6TIzeLnfosUgj1xK5ftoqDpoCnRk_dzY_jAx4.

[27] SEMTECH, *Application Note: An introduction to Ranging with the SX1280 transceiver*, rev 1.0 ed., 2017. Available at https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/44000000MDiH/OF02Lve2RzM6pUw9gNgSJXbDNaQJ_NtQ555rLzY3UvY.

[28] P. J. F. Groenen and M. van de Velden, "Multidimensional scaling by majorization: A review," *Journal of Statistical Software*, vol. 73, no. 8, p. 1–26, 2016.

[29] S. Macenski, S. Singh, F. Martín, and J. Ginés, "Regulated pure pursuit for robot path tracking," *Autonomous Robots*, vol. 47, pp. 685–694, Aug. 2023.

[30] F. Sobers, S. Anavatti, M. Garratt, and H. Abbass, *Real-Time Multi-obstacle Detection and Tracking Using a Vision Sensor for Autonomous Vehicle*, pp. 873–883. 06 2021.

[31] M. Batoš, L. Mandić, J. Obradović, N. Kraševac, B. A. Ferreira, and N. Mišković, "GNSS-denied search and object identification for autonomous surface vehicles using lidar and monocular camera," *Submitted for publication*, 2025.

[32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, pp. 2564–2571, 2011.

[33] J. Obradović, M. Fabijanić, M. Batoš, and N. Mišković, "Analysis of LiDAR-camera fusion for marine situational awareness with emphasis on cluster selection in camera frustum," *IFAC-PapersOnLine*, vol. 58, no. 20, pp. 434–439, 2024.

[34] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.

[35] L. Qi, L. Huang, Y. Zhang, Y. Chen, J. Wang, and X. Zhang, "A real-time vessel detection and tracking system based on lidar," *Sensors*, vol. 23, no. 22, p. 9027, 2023.

[36] J. Obradovic, L. Mandic, M. Batos, and F. Ferreira, "LiDAR-based docking at sea in GNSS-denied environments." Submitted for publication, 2025.

[37] B. Arbanas, A. Ivanovic, M. Car, M. Orsag, T. Petrovic, and S. Bogdan, "Decentralized planning and control for UAV–UGV cooperative teams," *Autonomous Robots*, vol. 42, no. 8, pp. 1601–1618, 2018.

[38] J. Domislovic, A. Ivanovic, F. Petric, and M. Orsag, "Remote grasping with a tethered unmanned aerial vehicle," in *2025 22nd International Conference on Ubiquitous Robots (UR)*, pp. 458–463, 2025.

[39] J. Domislovic, R. Milijas, A. Ivanovic, M. Car, G. Vasiljevic, B. A. Ferreira, F. Petric, M. Orsag, and S. Bogdan, "ρLiRLo: LiDAR-based relative localization with retro-reflective marker," in *2025 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 228–235, 2025.

[40] P. I. Corke, "Visual control of robot manipulators–a review," *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, pp. 1–31, 1993.