Applied Ocean Research 166 (2026) 104903

Contents lists available at ScienceDirect

Applied Ocean Research

journal homepage: www.elsevier.com/locate/apor

Research paper

Experimental validation of a modular navigation architecture for marine autonomous surface vehicles with reactive collision avoidance

Raphael Zaccone *, Filippo Ponzini, Michele Martelli

Department of Naval Architecture, Electric, Electronic, and Telecommunication Engineering (DITEN), University of Genoa, Via Montallegro 1, Genoa, 16145, GE, Italy

ARTICLE INFO

Keywords:
Autonomous surface vehicle (ASV)
Modular architecture
Collision avoidance
Path planning
Marine robotics
LIDAR-based obstacle detection
Sensor fusion
Publish–subscribe middleware (MQTT)

ABSTRACT

This paper presents a modular software architecture for the autonomous navigation of surface vehicles, designed around a layered awareness, navigation, guidance, and control structure. The proposed framework separates global path management, reactive local planning for collision avoidance, and control, while situation awareness combines LiDAR perception with INS/GNSS localization to maintain an up-to-date, and realistic representation of the surrounding environment. The architecture is designed around the concepts of modularity and scalability, enabling distributed computation and the flexible integration of modules. The implementation employs a lightweight publish/subscribe protocol to enable efficient real-time communication among modules. The experimental validation of the proposed architecture in a collision avoidance test featuring a research ASV is reported and discussed. The vehicle successfully executed polygonal paths, adapting its trajectory to avoid multiple unexpected obstacles while still reaching its prescribed waypoints. These results demonstrated the reliability of the proposed framework in supporting path following and adaptive collision avoidance under realistic operating conditions.

1. Introduction

Autonomous Surface Vehicles (ASVs) are emerging as critical tools for a wide range of maritime applications, including environmental monitoring, surveying, and coastal operations. The dynamics of marine environments pose significant challenges to the autonomy of the vessel, which requires fast and reliable situation awareness, path planning, collision avoidance, and control. These functions must interact seamlessly under real-time constraints, often while processing data from heterogeneous sensors such as RADAR, LiDAR, and INS/GNSS. As the complexity of missions and sensor suites grows, the software architecture enabling autonomy becomes a key factor for system scalability, adaptability, and performance.

Research on autonomous marine vehicles (Zheng and Liu, 2025) has produced various platforms for various applications, including environmental monitoring, surveying, and surveillance, both on the surface and underwater. Several contributions focused on ecological measures such as water quality monitoring (Dunbabin et al., 2009; Nambiar et al., 2023; Qadir et al., 2024), while others presented designs for multipurpose modular platforms for generic research applications (Odetti et al., 2020; Ferretti et al., 2023; Stanghellini et al., 2020; Carlson et al., 2023). Recent work has emphasized advanced sensing, multisensor integration, and surveillance-oriented autonomy (Gogendeau et al., 2025; Ponzini et al., 2024), also extending to medium-sized boats of 5 to 7 m of length (Brekke et al., 2022; Zhang et al., 2023; Luo et al., 2024; Kalliovaara et al., 2024; Hinostroza et al., 2025; Eide et al., 2025), or model-scale cargo vessels (You et al., 2020).

Although most of the existing research focuses on the description of hardware systems, fewer works have emphasized the design of navigation software architectures that combine modularity, distributed computation, and lightweight communication while integrating multiple high-bandwidth sensors. The present work addresses this gap by introducing a layered and modular navigation framework that modularly organizes situation awareness, path planning, collision avoidance, and control, relying on lightweight communication to support flexible integration and resilience. The architecture includes modules dedicated to situation awareness, integrating INS-GNSS and LiDAR sensors, guidance and collision avoidance, and ASV control using algorithms known from the literature. WiFi data exchange among the modules relies on the MQTT (Message Queuing Telemetry Transport) protocol (Standard, 2019), widely used in the Internet of Things (IoT) to achieve pragmatic and lightweight communication, allowing distributed modules to exchange information efficiently and ensuring cohesive system-level operation. The architecture has been implemented to control a

* Corresponding author.
E-mail address: raphael.zaccone@unige.it (R. Zaccone).

https://doi.org/10.1016/j.apor.2025.104903
Received 8 September 2025; Received in revised form 10 December 2025; Accepted 24 December 2025

research ASV and experimentally validated in field trials, including scenarios with and without reactive collision avoidance, thus demonstrating both reliable path-following and adaptive behavior in dynamic environments.

The remainder of the paper is structured as follows: Section 2 reviews related work on ASV hardware and autonomy architectures; Section 3 presents an overview of the proposed navigation pipeline, emphasizing the information flow and the logical relationships among the different modules; Section 4.1 describes the architectural implementation using the Publish/Subscribe broker-based paradigm, highlighting modularity and sensor integration; Section 5 illustrates the materials and methods used to validate the proposed architecture, including the SWAMP ASV, software, communication frameworks, sensors, and hardware, and the location and details of the experiments; Section 6 reports the experiment results; Section 7 discusses the implications, limitations, and potential extensions of the approach; eventually, Section 8 draws the conclusions of the paper.

2. Related work

Autonomous surface vehicles (ASVs) have become increasingly important platforms for environmental monitoring, surveying, and maritime surveillance, with research highlighting specialized designs and general-purpose architectures.

Caccia (2006) presented one of the first reviews of ASVs and their applications, identifying the main aspects to be analyzed in the field of research. Since then, numerous prototypes have been presented in the scientific literature. One of the first demonstrations of the utility of ASV in environmental applications was presented by Dunbabin et al. (2009), who developed a solar-powered catamaran for real-time monitoring of water quality. Their platform integrated navigation, control, laser-based obstacle detection and avoidance, allowing the vessel to perform long missions efficiently. Building on this theme, Nambiar et al. (2023) also addressed water quality monitoring, focusing on the design of a compact and practical ASV capable of continuous data collection. The ASV featured an inexpensive and essential architecture without collision avoidance capabilities.

More recently, Gogendeau et al. (2025) introduced an ASV designed for acoustic tracking, bathymetric, and photogrammetric surveys, combining multi-sensor integration with autonomous navigation to deliver a versatile tool for oceanographic research. The vessel is built on a paddleboard equipped with two thrusters, focusing on affordability and ease of realization. It features an extensive suite of sensors, including cameras and echosounders, but without autonomous collision avoidance or decision-making capabilities. Similarly, Carlson et al. (2023) developed the NORDACC ASV, a compact and modular platform that supports climate research through autonomous and remote-controlled operations.

Odetti et al. (2020) presented SWAMP, an ASV expressly designed for extremely shallow waters, with a hull and propulsion system specifically designed and optimized for challenging shallow-water conditions. SWAMP is a multipurpose platform capable of carrying high-weight payloads on a relatively compact platform. Since SWAMP is the platform used to validate the proposed architecture, further details will be given in Section 5, describing the materials and methods of this study. Complementary to this, Stanghellini et al. (2020) proposed OpenSWAP, an open-architecture low-cost ASV class for geophysical surveys in shallow water environments.

Beyond environmental and survey missions, ASVs are also increasingly being explored for maritime security and surveillance. Ponzini et al. (2024) proposed a constrained stochastic coverage algorithm for real-time monitoring of critical marine infrastructure, demonstrating how stochastic but constrained path planning can provide a balance between adaptability and systematic coverage in dynamic contexts. Their emphasis on multi-sensor integration and real-time responsiveness resonates with the need for robust situational awareness in autonomous platforms. Ponzini et al. (2023) developed a multi-sensor indoor tracking system for model-scale ASVs, enabling controlled testing of navigation and control algorithms before field deployment. The testing framework relies on the MQTT messaging system to share data among the sensors. Costanzi et al. (2020) reviewed interoperability across unmanned maritime vehicles, highlighting the challenges of ensuring seamless communication and coordination in multi-vehicle systems above and below water.

More recently, Kalliovaara et al. (2024) presented the eM/S Salama USV, a boat-sized vessel capable of autonomous collision avoidance and dynamic positioning, featuring a rich sensor suite including LiDAR, RADAR, and panoramic cameras, as well as deep-learning-based obstacle detection and tracking. Remarkably, the complex software architecture controlling eM/S Salama relies on MQTT and HTTPS protocols for communication with the remote control station.

The Milliampere boat has been the object of several scientific papers (Brekke et al., 2022; Hinostroza et al., 2025; Eide et al., 2025). It is a small autonomous ferry over 8 m long that has the typical features of a high-TRL vessel that is almost production-ready, with a focus on reliability and safety. It is equipped with numerous redundant sensors, including RGB cameras, LiDAR, RADAR, IMU, GNSS, and anemometer. The communication system is based on 4G and 5G networks with a particular focus on reliability and cybersecurity.

You et al. (2020) proposed a 7-m prototypal model of the well-established KVLCC2. The vessel was equipped with GPS, LiDAR, and CCTV systems for perception and data acquisition. Communication capabilities included an Automatic Identification System (AIS) operating with a 160 MHz antenna, a Long Range (LoRa) module with a 433 MHz antenna, and a 4G router. A series of experiments was conducted and described, including autonomous docking and undocking maneuvers, path-following navigation, and autonomous turning operations.

Zhang et al. (2023) proposed the Maverick, a 6-m-long ASV equipped with numerous sensors integrated through NATS, a broker-based publish–subscribe messaging system. The original version of the ASV is capable of autonomously navigating predefined waypoint sequences, although it does not have autonomous collision avoidance capabilities. However, it is equipped with perception sensors and has a modular structure for research applications. Relying on the Maverick platform, Luo et al. (2024) presented an acquisition pipeline based on LiDAR and INS/GNSS for updating nautical charts for inland navigation.

Together, these contributions demonstrate the diversity of ASV research, spanning environmental monitoring, shallow-water operations, surveillance, modular designs, and testing frameworks. Many papers present platforms built with simple elements, while some authors focus on optimizing hull shapes and propulsion systems to achieve high performance in navigation. Onboard architectures are often very simple when ASVs do not have advanced navigation features such as collision avoidance. In contrast, for large applications rich in sensors and features, architectures become more complex, exploiting paradigms such as microservices and messaging brokers. In general, few papers address the development of a modular, message-driven architecture for distributed computation that can flexibly support navigation, guidance, path planning, and collision avoidance.

This paper presents an architectural framework for an ASV awareness, navigation, guidance, and control system, developed using a modular and scalable approach, with a focus on a simple description of its basic modules. The system includes obstacle detection and navigation capabilities provided by LiDAR and INS-GNSS sensors and a hierarchy of guidance, collision avoidance, and control modules, developed in a platform-agnostic and inherently modular manner. Furthermore, this paper presents an experimental validation of the proposed architecture to demonstrate its robustness and reliability, identify its limitations, and suggest strategies to overcome them.

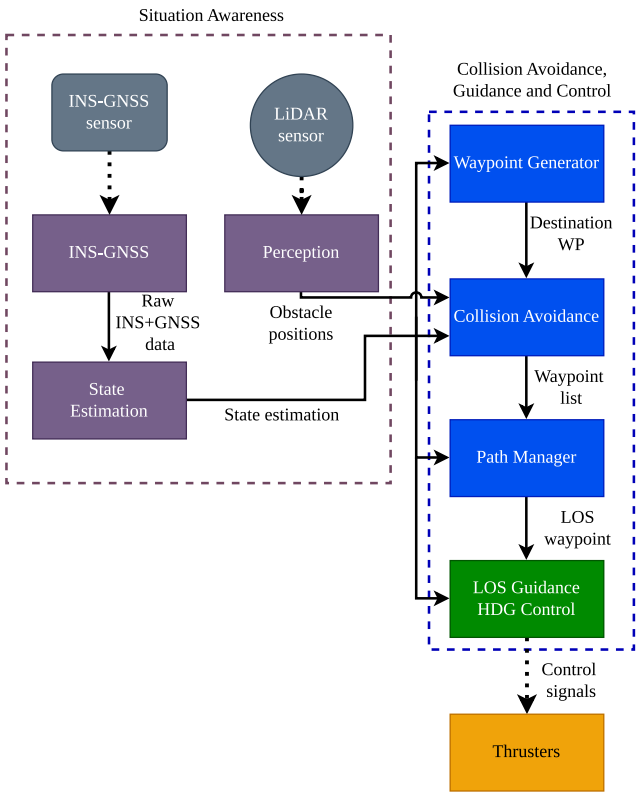


Fig. 1. A schematic representation of the proposed functional pipeline.

3. Autonomous navigation architecture

3.1. Overview

The architecture presented in this work envisions a vessel with autonomous abilities, capable of following a predefined sequence of waypoints for tasks such as experimental measurements or patrols, while detecting and dynamically avoiding unexpected obstacles encountered during navigation. The vessel needs to integrate proprioceptive and perceptual sensors that provide both self-localization relative to a fixed reference and real-time detection of surrounding obstacles. This information is processed by the Collision Avoidance (CAV) system, which determines optimal trajectories to safely reach each waypoint before proceeding to the next.

The overall logical pipeline is illustrated in Fig. 1. Within a situation awareness pipeline, an INS-GNSS system and a Kalman-filter-based state estimation continuously monitor the state of the vessel against the fixed reference. At the same time, a LiDAR-based perception algorithm provides information about nearby obstacles. The sensor data feeds the collision avoidance, guidance, and control pipeline, particularly the Collision Avoidance module, which handles reactive maneuvering to reach the destination point while avoiding collisions. Destination points are supplied by a Waypoint Generator, delivering the next waypoint each time the current target is reached. In this study, waypoints are assumed to be predefined, although they can also be generated dynamically using coverage algorithms to ensure comprehensive area exploration (Ponzini et al., 2024). Planned maneuvers are managed by a Path Manager, which communicates the current target waypoint to the Line-of-Sight (LOS) guidance and a heading control module, ultimately generating actuator commands for the thrusters.

The system architecture involves both serial and parallel asynchronous processing: some data produced by a module is used by multiple other modules, which must operate asynchronously and may be deployed on separate hardware units. These considerations motivate the design of the software architecture, described in detail in Section 4.1, and based on the publish–subscribe paradigm.

The remainder of this section presents some functional details of the main blocks of the pipeline shown in Fig. 1, in particular the Situation Awareness pipeline, including State Estimation and Perception modules, the Collision Avoidance, Guidance, and Control pipeline, with specific attention to the collision avoidance algorithm, Line-of-Sight guidance, and heading control.

3.2. State estimation

Accurate state estimation is a fundamental component of the navigation architecture of the ASV, as it provides the feedback required by other modules. The estimation process relies on measurements from the INS–GNSS system. The Latitude and Longitude from the GNSS system are converted into local coordinates through the UTM (Universal Transverse Mercator) coordinate transformation (Snyder, 1987). Considering the local system consistent with the NED (North-East-Down) standard and centered in $\boldsymbol{\Omega} = (lat_{\boldsymbol{\Omega}}, lon_{\boldsymbol{\Omega}})$, a generic position $\boldsymbol{P} = (lat_{\boldsymbol{P}}, lon_{\boldsymbol{P}})$ is converted into the local reference frame coordinates (x_{GNSS}, y_{GNSS}) according to:

$$(x_{GNSS}, y_{GNSS}) = \boldsymbol{\zeta}(lat_{\boldsymbol{P}}, lon_{\boldsymbol{P}}) - \boldsymbol{\zeta}(lat_{\boldsymbol{\Omega}}, lon_{\boldsymbol{\Omega}}) \tag{1}$$

where $\boldsymbol{\zeta}$ refers to the UTM conversion function. These measurements are fused through a linear Kalman filter (LKF) (Maybeck, 1990) implementing a purely kinematic constant velocity state transition model. The heading ψ_{mag} is derived from the magnetometer, and the angular velocity r_{gyro} is obtained from the gyroscope. A generic multi-sensor measure vector is represented as:

$$\mathbf{z} = \left(x_{GNSS}, y_{GNSS}, \psi_{mag}, r_{gyro}\right)^{\mathsf{T}} \tag{2}$$

Thus, the feedback vector is represented as:

$$\mathbf{x}_{ASV} = \left(x_0, y_0, \psi, v_x, v_y, r\right)^{\mathsf{T}} \tag{3}$$

3.3. Perception

LiDAR measurements are pre-processed to obtain a reliable obstacle representation for navigation. First, the raw point cloud is cropped to a suitable region of interest (ROI), accounting for the sensor placement and fixed structures such as the vessel hull. A noise reduction step follows, where spurious points are removed using a combined threshold on reflectivity ($\tau_I \in [0, 1]$) and vertical position, as water-surface effects (e.g., ripples, wakes, reflections) typically generate noisy returns near the waterline (indicated as $z_{\text{sea}} \in \mathbb{R}$ and derived from the mounting position of the sensor), as outlined by Faggioni et al. (2022). The filtering condition for a generic $\mathbf{p}_i = (x_i, y_i, z_i, I_i)$ ith LiDAR point, with a tolerance value $\delta_z > 0$, is defined as:

$$\mathbf{p}_i \text{ is discarded if } I_i < \tau_I I_{\max} \ \wedge \ |z_i - z_{\text{sea}}| < \delta_z. \tag{4}$$

The filtered points are then projected onto a bird-eye view (BEV) to simplify spatial reasoning. Finally, the point cloud is downsampled to an obstacle set $\hat{O} = \{\hat{\mathbf{o}}_i\}$ using a grid-based approach, yielding a reduced representation that preserves the relevant obstacle structures while providing a compact set of points suitable for real-time processing, according to Ponzini et al. (2024). The obstacle set $\hat{O}$ is expressed in the sensor's reference frame and contains only the coordinates on the plane, severely limiting the size of the cloud. Since the state estimation model is expressed in 3 DOF, the point cloud is not compensated for roll, pitch, and heave motions. This limitation does not affect the present case study due to the modest amplitude of the neglected dynamics; however, the framework already accounts for the possibility of applying such compensation in future trials in open-sea conditions, where these motions exhibit larger magnitudes.

3.4. Collision avoidance

The ASV follows a route defined by a sequence of waypoints. The Waypoint Generator module updates the destination waypoint once the ASV reaches the previous one within a specified acceptance radius.

Collision avoidance is handled by a dedicated module positioned upstream of the guidance and control system (Fig. 1), which dynamically computes a collision-free maneuver when obstacles are detected. Depending on the current position of the ASV and the location of the obstacles, the algorithm replans the trajectory towards the next waypoint, inserting intermediate points as needed to maintain a safe path. The waypoint updates provided to the LOS module are influenced both by the ASV's progress along the original path and by the replanning performed by the collision avoidance module.

The path planning algorithm at the core of the collision avoidance module is based on a Dynamic Programming (DP) algorithm proposed by Zaccone (2024), which computes the optimal sequence of waypoints on a discrete grid to reach the destination. This DP approach is particularly advantageous for real-time operation, as it balances computational efficiency with path optimality, enabling responsive and adaptive navigation in dynamic environments.

A generic maneuver R_N is represented as a sequence of N waypoints $\mathbf{x}_i = (x_i, y_i)^\mathsf{T}$, each expressed in the local reference system:

$$R_N = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N) \tag{5}$$

Where $\mathbf{x}_0 = (x_0, y_0)^\mathsf{T}$ represents the current position of the ASV available from ASV state estimation, and $\mathbf{x}_N$ is the destination waypoint generated by the waypoint generator. A few notations are now introduced:

- The length of the leg connecting waypoints $\mathbf{x}$ and $\mathbf{y}$ is denoted as $L(\mathbf{x}, \mathbf{y})$;
- The obstacle point cloud, i.e., the set of detected obstacles, is denoted as $O = \{\mathbf{o}_i\}$, where $\mathbf{o}_i$ are expressed in the local reference system;
- The minimum distance between the route leg $(\mathbf{x}, \mathbf{y})$ and the obstacle set O is written as $d_O(\mathbf{x}, \mathbf{y})$.

It is noteworthy that the collision avoidance algorithm formulation can handle both static and dynamic obstacles, i.e., those with zero and non-zero velocities, respectively. In the presented application, since the sensing system is not able to track obstacles, they are approximated as static. This assumption is considered acceptable in the context and purposes of this study and in the experimental scenario, which involves fixed or low-speed obstacles. Moreover, it is worth mentioning that the obstacle point cloud is usually expressed in the sensor reference system, that is, relative to the position and orientation of the sensor. If $\mathbf{x}_0$ and ψ are the currently estimated position and orientation of the ASV in the local coordinate system (as per Section 3.2), it is straightforward to apply a coordinate transformation:

$$\mathbf{o}_i = R_\psi \hat{\mathbf{o}}_i + \mathbf{x}_0 \tag{6}$$

Where R_ψ is the rotation matrix:

$$R_\psi = \begin{pmatrix} \cos(\psi) & -\sin\psi \\ \sin(\psi) & \cos\psi \end{pmatrix} \tag{7}$$

The algorithm used in this study aims to minimize the path length while keeping a safety distance from obstacles. Moreover, maintaining an additional safety distance is encouraged by a penalty term in the cost function. The optimization problem is formulated as follows:

$$\begin{cases} \underset{\mathbf{x}_1,\dots,\mathbf{x}_{N-1}}{\operatorname{argmin}} \sum_{i=1}^{N} \left[L(\mathbf{x}_{i-1}, \mathbf{x}_i) + \dfrac{\lambda}{d_O^2(\mathbf{x}_{i-1}, \mathbf{x}_i)} \right] \\ s.t.: \\ d_O(\mathbf{x}_{i-1}, \mathbf{x}_i) > d_s, \ i = 1, \dots, N \end{cases} \tag{8}$$

where the term $\frac{\lambda}{d_O^2(\mathbf{x}_i, \mathbf{x}_{i+1})}$ penalizes the proximity to the obstacles depending on the parameter λ, and d_s is the safety distance.

The cost function $C(R_N) = \sum_{i=1}^{N} \left[L(\mathbf{x}_{i-1}, \mathbf{x}_i) + \frac{\lambda}{d_O^2(\mathbf{x}_{i-1}, \mathbf{x}_i)} \right]$ can be additively separated as follows:

$$C(R_k) = C(R_{k-1}) + \left[L(\mathbf{x}_{k-1}, \mathbf{x}_k) + \frac{\lambda}{d_O^2(\mathbf{x}_{k-1}, \mathbf{x}_k)} \right] \tag{9}$$

Let $F(x_k) = \min_{\mathbf{x}_1,\dots,\mathbf{x}_{k-1}} C(R_k)$, i.e., the cost to reach the waypoint $\mathbf{x}_k$ following an optimal policy; the problem can be written recursively using the Bellman Equation (Bellman, 1954):

$$F(\mathbf{x}_k) = \min_{\substack{\mathbf{x}_{k-1}: \\ d_O(\mathbf{x}_{k-1}, \mathbf{x}_k) > d_s}} \left[F(\mathbf{x}_{k-1}) + L(\mathbf{x}_{k-1}, \mathbf{x}_k) + \frac{\lambda}{d_O^2(\mathbf{x}_{k-1}, \mathbf{x}_k)} \right] \tag{10}$$

The problem is solved using a tabulation-backtracking scheme (Zaccone, 2024). The outcome is an ordered list of waypoints: the first corresponds to the current position of the ASV, the last is the original destination, and the intermediate waypoints define a collision-free path that maintains a safe distance from detected obstacles. As the environment changes, the route is recomputed at fixed intervals, ensuring that the ASV continuously adapts its trajectory to navigate safely through dynamic conditions.

3.5. Guidance and control

Several approaches to ship guidance and control have been proposed in the literature (Donnarumma et al., 2016; Fruzzetti et al., 2024). In this study, the guidance system employs a pure Line-of-Sight (LOS) approach (Fossen, 2011) to guide the ASV towards the next waypoint. This simple method was chosen for its ease of implementation, requiring only a single waypoint to be stored in memory rather than two, as would be necessary for drift compensation. Drift compensation is considered unnecessary for the present application, where precise track following is not critical and the main task required to the ASV is to reach the assigned waypoints. Nonetheless, the proposed architecture is scalable and can accommodate more advanced guidance and control algorithms if needed. The LOS heading ψ_{los} to reach the desired waypoint (x_{wp}, y_{wp}) is expressed as:

$$\psi_{los} = \arctan\left(\frac{y_{wp} - y_{ASV}}{x_{wp} - x_{ASV}} \right) \tag{11}$$

The yaw moment required N_{req} is therefore determined with a PD controller, proportionally to the heading error against ψ_{los} and the measured rotation speed r:

$$N_{req} = K_p (\psi_{los} - \psi) - K_d r \tag{12}$$

The forward thrust required X_{req} is instead set in open loop. Therefore, the required force vector $\mathbf{X}$ is given by:

$$\mathbf{X}_{req} = (X_{set}, 0, N_{req})^\mathsf{T} \tag{13}$$

The architecture presented in this paper has been implemented and tested to control a vehicle called SWAMP (Shallow Water Autonomous Multipurpose Platform), a 1.2-m-long, fully electric, modular Autonomous Surface Vehicle (ASV) equipped with four pump-jet azimuthal thrusters, developed by Odetti et al. (2020) and used in this work as a testing platform. Additional details about the platform and the experiments will be provided in Section 5.

In the configuration used in this study, the thrust allocation is fixed at predetermined angles. The four pump-jet thrusters of the SWAMP vehicle are oriented as shown in Fig. 2, each positioned at 45° relative to the ahead direction. Although this configuration is not optimized for energy efficiency or maximum speed, it provides high responsiveness and simplifies control by reducing the degrees of freedom of the system. Each thruster is controlled by a normalized thrust signal $\tau_i \in [0, 1]$

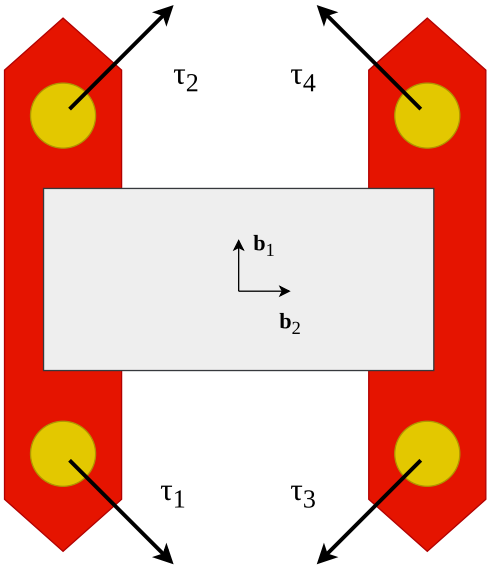


Fig. 2. The fixed thruster angle configuration used in the experiment.

with $i = 1, \ldots, 4$. The thrust signals are determined as a function of the required force vector $\mathbf{X} = \{X, Y, N\}$ as follows:

$$\begin{cases} \tau_1 = \dfrac{f^+(X) + f^+(Y) + \frac{1}{2} f^+(N)}{\max_i \tau_i} \\ \tau_2 = \dfrac{-f^+(X) + f^+(Y) - \frac{1}{2} f^+(N)}{\max_i \tau_i} \\ \tau_3 = \dfrac{-f^+(X) - f^+(Y) + \frac{1}{2} f^+(N)}{\max_i \tau_i} \\ \tau_4 = \dfrac{f^+(X) - f^+(Y) - \frac{1}{2} f^+(N)}{\max_i \tau_i} \end{cases} \tag{14}$$

where $y = f^+(x)$ returns the positive part of x.

4. Software architecture

4.1. Overview

The implemented architecture is built around the MQTT (Message Queuing Telemetry Transport) messaging system, a lightweight and widely adopted protocol in IoT applications (Standard, 2019; Soni and Makwana, 2017), following the publish–subscribe paradigm. The main features of MQTT are high efficiency and support for low-power devices in IoT applications, such as the testing setup described in this paper; moreover, it is a consolidated and well-established technology. MQTT operates on a broker-based, publish–subscribe model, where nodes can produce (publish) or consume (subscribe) data organized by topics. Publishers send updates to the broker, which immediately distributes them to all subscribing nodes. This decoupling of data sources and consumers allows the system to efficiently integrate multiple sensing, guidance, and control modules in real time.

Fig. 3 provides a schematic overview of the system, showing the various nodes connected to the central broker.

Each block represents an individual node that exchanges the data required for the ASV mission. At the software level, each node is an independent process that can be physically executed on a networked machine and exchanges information with other nodes using the publish/subscribe paradigm. The remainder of this section abstractly illustrates the interface of each node, i.e., the published or subscribed data. Moreover, Section 5 provides hardware data, some implementation details (e.g., languages, environments, and libraries), and information on how the processes were distributed across the physical machine setup used during the experiments.

Table 1
Waypoint generator node.

WPG	
Publishes:	Destination waypoint
Subscribes:	State estimation
Other Tx/Rx:	–

Table 2
INS-GNSS node.

INS-GNSS	
Publishes:	INS-GNSS raw data
Subscribes:	–
Other Tx/Rx:	Rx INS-GNSS raw data (USB serial)

4.2. Waypoint generator

The Waypoint Generator (WPG) module manages the sequence of destination points for the ASV. Table 1 shows the publish/subscribe interface of the module, which continuously monitors the position of the vessel by subscribing to the proper topic, and, when the ASV reaches a waypoint within a defined acceptance radius, it publishes the next waypoint on another dedicated topic. This allows other modules, such as the collision avoidance and guidance systems, to access the most up-to-date target location in real time. The module supports pre-defined waypoint sequences, but could be extended to manage dynamically updated paths generated and published by higher-level planning algorithms, by scaling the publish/subscribe relationship between the CAV and PM modules described later in this section.

4.3. INS-GNSS

The INS-GNSS module acquires raw navigational data from the onboard inertial measurement unit (IMU) and magnetometer, as well as position and velocity information from the GNSS system via serial USB communication. The data is acquired at 100 Hz and downsampled to 10 Hz to balance temporal resolution with computational end communication performance. Operating as a pure publisher (Table 2), the INS-GNSS module continuously publishes the acquired data on a dedicated MQTT topic, providing other modules with real-time information on the position, heading, and velocity of the ASV. By decoupling data acquisition from downstream processing, the module supports modularity: it allows multiple nodes, such as guidance or collision avoidance, to subscribe to the same data stream without interfering with the measurement process.

4.4. State estimator

The State Estimator (SE) module subscribes to raw data from the INS-GNSS module and fuses the information to produce a refined estimate of the ASV's position, velocity, and orientation. By filtering out sensor noise and compensating for measurement errors, the module generates a more accurate and reliable state estimate, which is published on a dedicated MQTT topic (Table 3). This enhanced state information can then be used by downstream modules, such as the collision avoidance or path-following controllers, ensuring robust navigation even in the presence of sensor uncertainty. Exploiting the modularity of the architecture, the SE module could be extended to receive data from multiple sensors, allowing for a more robust and redundant state estimation in case of sensor failure.

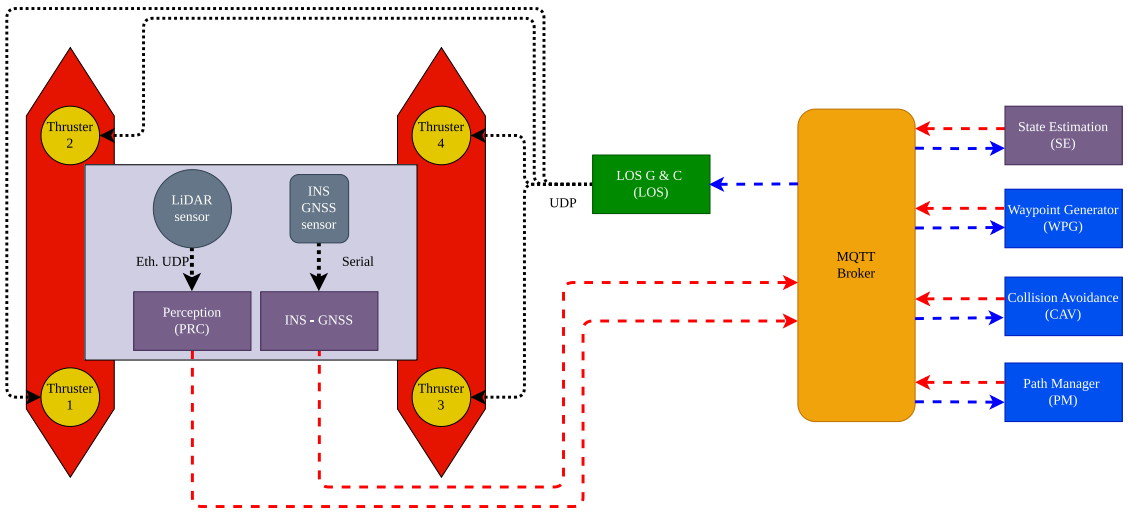


Fig. 3. An overview of the broker-based publish–subscribe software architecture.

Table 3
State estimation node.

LKF	
Publishes:	State estimation
Subscribes:	INS-GNSS raw data
Other Tx/Rx:	–

Table 4
Perception node.

PRC	
Publishes:	LiDAR subsampled point cloud
Subscribes:	–
Other Tx/Rx:	Rx LiDAR raw pointcloud (ethernet UDP)

Table 5
Collision avoidance node.

CAV	
Publishes:	CAV waypoint list
Subscribes:	State estimation LiDAR subsampled point cloud Destination waypoint
Other Tx/Rx:	–

Table 6
Path manager node.

PM	
Publishes:	LOS waypoint
Subscribes:	CAV waypoint list State Estimation
Other Tx/Rx:	–

4.5. Perception

The Perception (PRC) module acquires high-resolution point cloud data from the onboard LiDAR sensor via an Ethernet cable and processes it to generate a downsampled bird-eye-view representation of obstacles projected onto the water surface plane. This transformation reduces computational complexity and data traffic while preserving the essential spatial information needed for collision avoidance. The processed obstacle map is then published on a dedicated MQTT topic, as Table 4 shows, for use by the CAV module. By providing an efficient and timely representation of the surrounding environment, this module enables the ASV to react to obstacles in real time.

4.6. Collision avoidance

As Table 5 shows, the Collision Avoidance (CAV) module subscribes to multiple MQTT topics to gather real-time information from other system nodes, including the current state of the vehicle from the SE module, the next destination waypoint from the WPG, and obstacle data from the PRC. Using this information, the module computes a safe and feasible trajectory to the target waypoint, dynamically adjusting for any detected obstacles. The planned maneuver is published as a sequence of intermediate waypoints, starting from the current position of the ASV and ending at the destination, enabling the guidance and control system to avoid collisions while maintaining progress along the assigned path. This continuous replanning ensures that the ASV can adapt to changes in the environment in real time.

4.7. Path manager

The PM module acts as an intermediary between the CAV and the guidance and control module. It buffers the most recent sequence of waypoints published by the CAV and determines the next immediate waypoint to be reached by the ASV, taking into account its current position. Table 6 shows the publish/subscribe interface of the module. Waypoints are considered achieved once the ASV enters a predefined acceptance radius. By continuously updating and publishing the next target waypoint, the PM ensures smooth progression along the planned path while allowing the guidance and control system to focus on low-level trajectory tracking and actuation.

4.8. LOS guidance and heading control

The LOS (Line of Sight) guidance and heading control module adjusts the heading of the ASV to guide it towards the next waypoint provided by the Path Manager. Using the LOS logic, it computes the desired course correction based on the vehicle's current position relative to the target waypoint. This module operates solely as a subscriber within the MQTT architecture, receiving waypoint updates but not publishing data, as the resulting thrust commands are sent directly to the four thrusters via a WebSocket interface, as described in Table 7.

(a) (b)

Fig. 4. The SWAMP ASV in a render 4(a), and equipped for an autonomous mission 4(b). Notice the Pandar XT32 LiDAR and the Xsens MTi-G-710 INS-GNSS module at the bow.

Table 7
LOS guidance and heading control node.

LOS	
Publishes:	–
Subscribes:	LOS waypoint State Estimation
Other Tx/Rx:	Tx Thrust commands (UDP)

5. Materials and methods

5.1. The SWAMP platform

The testing was conducted using SWAMP (Shallow Water Autonomous Multipurpose Platform), a 1.2-m-long, fully electric, modular Autonomous Surface Vehicle (ASV) developed by Odetti et al. (2020). Designed for operation in extremely shallow and constrained environments, SWAMP features a catamaran hull with four azimuthal pump-jet thrusters that are flush with the hull, providing precise maneuverability and minimizing the risk of damage from obstacles or grounding. The hull is constructed of soft foam material, which ensures unsinkability, durability, and robustness against potential collisions and groundings in shallow water. SWAMP's compactness and light weight were specific design requirements to facilitate easy transport and deployment, making it ideal for use in remote or difficult-to-access areas. The motivation for its design stems from the practical need to operate in rivers, lagoons, estuaries, and wetlands, where water depths may be as low as a few tens of centimeters, to perform a wide range of missions such as environmental monitoring, water sampling, and geomorphological analysis (Karaki et al., 2022; Ferretti et al., 2023).

The platform features an open hardware and software architecture based on a local WiFi network, allowing for flexible payload integration and aiming towards modularity. In particular, the WiFi network enables communication between the various propulsion, control, and processing modules. The thrusters are controlled by appropriate commands sent via UDP websockets, while the network can be used for communication between additional nodes using any more abstract communication paradigm. A platform configured in this way is ideally suited to the architecture proposed in this paper, which, as will be clarified later, will be based on a network of nodes capable of communicating using a publish/subscribe paradigm through a broker.

Fig. 4(a) shows the ASV in a render, while Fig. 4(b) shows it fully equipped for a mission during the test campaign described later in this paper. The LiDAR and INS-GNSS sensors described in Section 5 can be seen arranged at the bow to ensure an adequate field of view for the former, while other sensors not relevant to this study are present on the deck.

5.2. Software

The autonomous navigation architecture was implemented using the MQTT protocol, a lightweight publish/subscribe messaging system widely adopted in IoT applications (Standard, 2019). Communication between nodes was ensured by the popular Mosquitto MQTT broker implemented by Eclipse (Light, 2017), ensuring asynchronous and reliable data exchange. Client functionality was provided by the Paho library also by Eclipse. Both implementations are Free and Open Source Software (FOSS) licensed under the EPL/EDL license.

The modular architecture ensured by the MQTT middleware allowed different computational nodes to be implemented in various languages, including Python/NumPy (Van Rossum, 1995; Harris et al., 2020) for rapid prototyping, relying on Rust (Klabnik and Nichols, 2023) for custom implementation of computationally intensive modules, such as the collision avoidance algorithm or the LiDAR drivers. FilterPy (Labbe, 2018) was used to implement the State Estimation module.

5.3. Sensors and hardware

The perception and state estimation were handled by a combination of onboard sensors and external computation. The ASV is equipped with a HESAI Pandar XT-32 LiDAR (Fig. 6), which provides 32 infrared laser beams with a vertical field of view of 32° at 1° resolution and a full 360° horizontal coverage. The acquisition frequency can be set in the 5–20 Hz range; for the case study, the LiDAR was operated at 5 Hz, ensuring a 0.09° horizontal angular resolution. Additional technical specifications are provided in Table 9. Inertial and positioning data are measured by an Xsens MTi-G-710 GNSS/INS system (Fig. 5), whose technical specifications are provided in Table 8. Raw sensor measurements were acquired onboard and published on dedicated topics, allowing a ground-based system for processing.

The SE module ran on a ground laptop, fusing INS and GNSS data to produce an accurate state estimate of the vehicle. The Waypoint Generator, Collision Avoidance (CAV) module, Waypoint Manager, and LOS module, forming the autonomous decision-making pipeline, were executed on a second ground-based laptop, together with the MQTT broker. More specifically:

- Onboard laptop for sensor acquisition: Acer Enduro, Intel Core i3-10110U, 2.10 GHz 8 GB RAM, Fedora 42 OS. Collected the raw INS/GNSS and LiDAR data via USB and ethernet cables, and published them to the MQTT broker.
- Ground laptop for state estimation (SE module): Asus ROG G14, AMD Ryzen 9-6900HS, 4.9 GHz, 16 GB RAM, Fedora 42 OS.

Fig. 5. A reference image of the Xsens MTi-G-710.

Table 8
Xsens MTi-G-710 INS-GNSS technical specifications.

Sensor	Spec.	Value
Gyroscope	Standard full range	450 deg/s
	In-run bias stability	10 deg/h
	Noise density	$0.01°/s/\sqrt{Hz}$
Accelerometer	Standard full range	20 g
	In-run bias stability	15 μg
	Noise density	$0.01°/s/\sqrt{Hz}$
Magnetometer	onboard	Yes
Barometer	onboard	Yes
GNSS receiver	Brand	u-blox
	Model	MAX-M8

Table 9
Pandar XT32 LiDAR main specifications.

Spec.	Value
Vertical FOV	31°
Vertical resolution	1°
Horizontal FOV	360°
Horizontal resolution	0.09° to 0.36°
Operating Frequency	5 Hz to 20 Hz
Range	120 m
Accuracy	0.01 m

Table 10
The buoys positioned in the experimental testbed.

	Coordinates [m]
BUOY 1:	(−30.0, 15.0)
BUOY 2:	(−35, 47)
BUOY 3:	(−27, 76)

- Ground laptop for collision avoidance, control, and MQTT broker: Lenovo ThinkPad, Intel Core i9-12900H, 5.10 GHz, 64 GB RAM, Fedora 42 OS. Hosted the Mosquitto broker, Waypoint Generator, CAV, Waypoint Manager, and LOS modules. Connected via an external ground-based WiFi antenna to the ASV to send the commands to the thrusters.

This configuration ensured that computationally intensive modules could run onshore, while the onboard system handled real-time sensor acquisition and publishing. The distributed setup was implemented as described above for practical reasons, determined by the availability of workstations in an outdoor experimental context, and trying to keep the most computationally demanding modules on high-performance machines. The proposed architecture, taking advantage of the flexibility offered by communication middleware, is inherently scalable and modular, and can be remodeled as needed while maintaining high performance and low latency.

5.4. Location of the experiments

To validate the proposed architecture, a series of experimental tests was carried out at the facilities of the Italian Research Council (CNR), located in Nemi, Rome, Italy. The station is located on the shores of the volcanic Lake Nemi, a lake area with almost no human activity, which is used for experimental studies concerning the maneuverability and controllability of self-propelled models. The lake surface is about 1.67 km^2 and the main dimensions span up to 0.71 nm along the E-W and up to 0.97 nm in the S-N direction, while the maximum depth is 26.5 m. Given its particular topology and minimal environmental noise in both air and water, the basin also allows high-quality measurements of radiated noise in the far field. In recent years, a new line of research has been established concerning Maritime Autonomous Systems (MASS). In the experimental basin, there are no restrictions on operative speed, and weather conditions are continuously monitored throughout the year and predicted during testing days, allowing the control and repeatability of experiments.

Table 11
Coordinates of the planned path for the LOS experiment.

Leg	Coordinates [m]	Dist.
WP 1-2:	(−68.0, 59.0) → (−18.0, 57.0)	50.0 m
WP 2-3:	(−18.0, 57.0) → (−31.0, 94.0)	39.2 m
WP 3-4:	(−31.0, 94.0) → (−56.0, 93.0)	25.0 m
WP 4-5:	(−56.0, 93.0) → (−50.0, −3.0)	96.2 m
WP 5-6:	(−50.0, −3.0) → (−15.0, 11.0)	37.7 m
WP 6-1:	(−15.0, 11.0) → (−68.0, 59.0)	71.5 m
	Total dist.:	319.7 m

The experiments were conducted and managed from a ground station placed on a semi-floating dock, relative to which the $\boldsymbol{\Omega}$ system of coordinates was expressed, centered in $41°43'11.4\varepsilon\ N$, $12°42'01.4\varepsilon\ E$. Fig. 7 shows a satellite view of the lake and a zoomed view of the semi-floating dock.

5.5. Experiment description

5.5.1. LOS experiment

Two experiments were conducted to evaluate the autonomous operational capabilities of the proposed architecture. The first experiment consisted of navigating a predefined obstacle-free polygonal route without the collision avoidance feature, thus feeding the waypoints from the WPG to the LOS module. The primary aim was to verify that the ASV could follow an assigned route autonomously under ideal conditions, while also assessing the robustness and stability of the foundations of the software architecture.

The assigned polygon consisted of six waypoints, as illustrated in Fig. 8, with a total route length of approximately 320 m. The coordinates of the waypoints in the $\boldsymbol{\Omega}$ system of coordinates, as well as the lengths of the individual legs, are reported in Table 11. It is worth noting that several buoys were positioned within the test field, with their locations detailed in Table 10. However, in this experiment, the route was deliberately planned to avoid the buoys, ensuring that the ASV could sail through the path without needing any collision avoidance maneuvers, since the CAV module was not operating.

This setup allowed for an initial validation of waypoint tracking performance and provided a baseline reference for subsequent experiments involving reactive collision avoidance.

5.5.2. CAV experiment

The second experiment represents a complete autonomous navigation scenario, requiring all the modules to operate, and was specifically designed to put the CAV module to the test. The ASV was tasked with following a polygonal route, as shown in Fig. 9, with waypoints listed in Table 12. The total path length is approximately 206 m.

During navigation, the ASV encounters buoys obstructing the planned path along several legs of the polygon. In other words, the polygon cannot be followed in the same manner as the first experiment, as this would result in multiple collisions. In contrast, once the perception system detects the obstacles, the CAV module is responsible for computing and updating alternative collision-free trajectories whenever

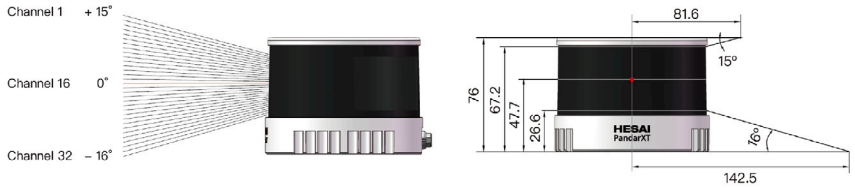


Fig. 6. A technical representation of Pandar XT32 LiDAR.

Fig. 7. A Google Earth view of Lake Nemi, the location where the experimental campaign took place.

Table 12
Coordinates of the planned path for the CAV experiment.

Leg	Coordinates [m]	Dist.
WP 1-2:	(−42.0, 63.0) → (−15.0, 88.0)	36.8 m
WP 2-3:	(−15.0, 88.0) → (−40.0, 70.0)	30.8 m
WP 3-4:	(−40.0, 70.0) → (−26.0, 4.0)	67.5 m
WP 4-5:	(−26.0, 4.0) → (−14.0, 17.0)	17.7 m
WP 5-1:	(−14.0, 17.0) → (−42.0, 63.0)	53.9 m
	Total dist.:	206.6 m

Table 13
Main features of the LOS experiment.

Distance	668.5	m
Total time	884.6	s
Avg. Speed	0.76	m/s

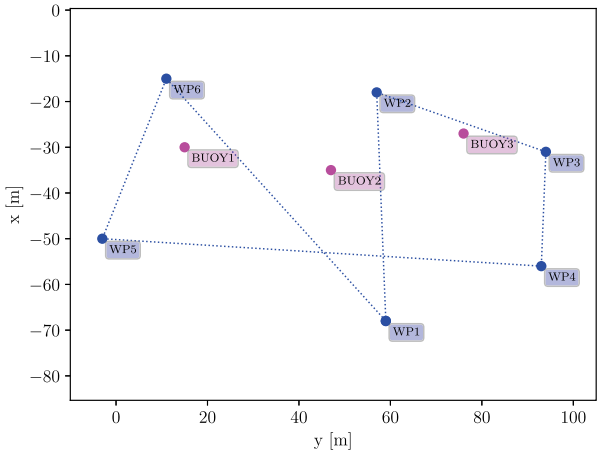


Fig. 8. An overview of the path for the LOS experiment.

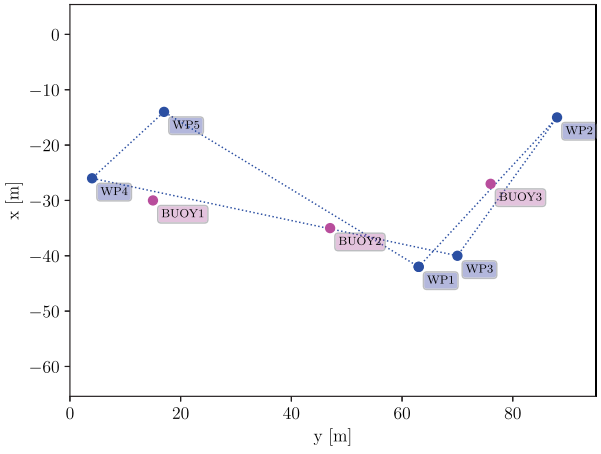


Fig. 9. Coordinates of the planned path for the CAV experiment.

it is needed, keeping a safety distance of 5 m from the obstacles while maintaining progress towards the assigned sequence of waypoints through the polygon.

Finally, it is worth noting that during the second experiment, other experimental activities were happening outside the authors' control, including other boats, ASVs, or swimmers potentially crossing the area and adding uncertainty and challenge to the obstacle detection and avoidance system. This second experiment aims to challenge the proposed architecture in real dynamic conditions, validating the ability of the system to adaptively plan maneuvers and maintain reliable navigation under real-world constraints.

5.6. Weather conditions

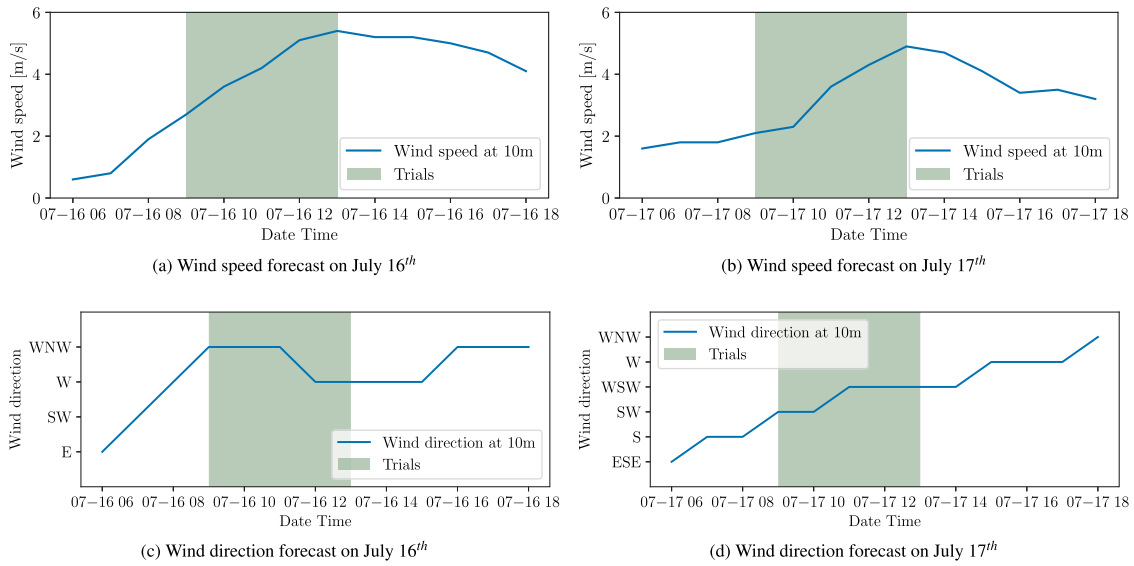
The experiments were carried out on 2025-07-16 and 2025-07-17, under good meteorological conditions. In particular, the LOS experiment took place on July 16th between 9:00 and 13:00, while the CAV experiment was carried out the following day in the same time window. Figure Fig. 10 shows the weather forecast for the test days, indicating winds of up to approximately 5 m/s on both days.

6. Results

6.1. LOS experiment

Fig. 11 shows the trajectory of the ASV during the LOS experiment. In the figure, the planned trajectory is represented by the blue dotted line connecting the waypoints in ascending order, while the solid curve represents the trajectory actually followed by the ASV in the experiment. Moreover, Fig. 12 presents the data measured during the experiment, including the time histories of the heading (Fig. 12(a)), the speed V, the velocity components in the body frame u and v (Fig. 12(b)), and the distance from the waypoints and from the current LOS waypoint (Fig. 12(c)).

The proportional heading control has an expected static error and some minor oscillations that do not affect the ASV's navigation. There is a slight delay in taking over the next waypoint due to the overall latency of the system and the discrete timing behavior of some network nodes. The implementation of the state estimation module generated



(a) Wind speed forecast on July 16th

(b) Wind speed forecast on July 17th

(c) Wind direction forecast on July 16th

(d) Wind direction forecast on July 17th

Fig. 10. Wind forecast data on the days the trials took place.

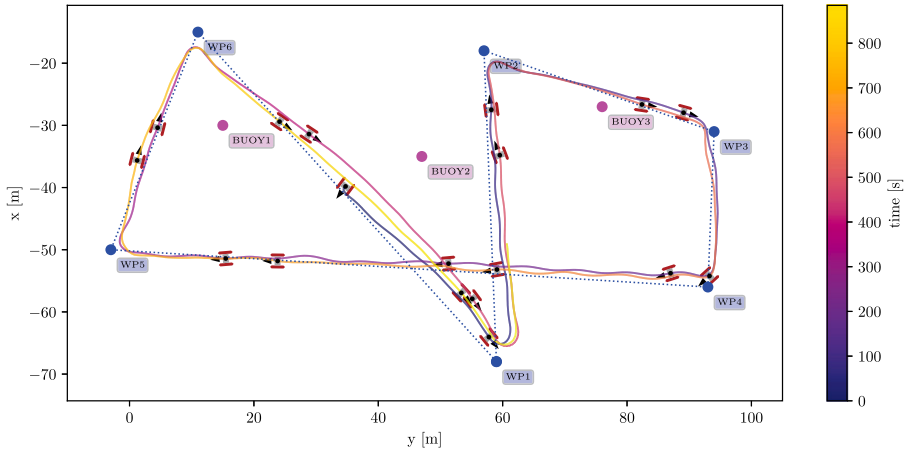


Fig. 11. The ASV's trajectory (solid line) and planned path (blue dashed line) during the LOS experiment.

Table 14
Main features of the CAV experiment.

Distance	260.1	m
Total time	600.0	s
Avg. Speed	0.43	m/s
CPA	6.4	m

an angle between −180 and 180 degrees. Therefore, the graph shows a physiological and sudden oscillation when the heading approaches the discontinuity value. It is worth noting that the Kalman filter effectively smoothed the oscillation, containing it and converging to the correct value in a few time steps.

Other average experimental metrics are summarized in Table 13. The trial lasted approximately 15 min, during which the ASV completed over two laps, covering a total distance exceeding 650 m at an average speed of 0.76 m/s.

The vehicle successfully followed the assigned polygonal path, smoothly reaching each waypoint within the acceptance threshold, demonstrating behavior consistent with the implemented guidance and control algorithms. Overall, the architecture was considered reliable and robust enough to proceed to the second, more challenging experiment.

6.2. CAV experiment

Fig. 13 shows the trajectory of the ASV during the collision avoidance experiment: unlike the LOS experiment, the trajectory does not strictly follow the preset polygonal legs represented by the dashed blue line, as several buoys obstruct the direct path. The ASV follows the trajectory variations suggested by the CAV module, which are updated every 3 s and are not shown in the figure for readability. Fig. 14 presents the data measured during the experiment: it is worth noting that the ASV's heading, instead of following the LOS angle to each destination waypoint, is guided by the CAV module (Fig. 14(a)), while still reaching all the destination waypoints (Fig. 14(c)), maintaining a safe distance of over 5 m (Fig. 14(d)). The experiment, whose average data are summarized in Table 14, lasted approximately 10 min, during which the ASV completed over one lap of the polygon, covering 260 m at an average speed of 0.43 m/s.

As an example, Fig. 15 illustrates a sequence of events demonstrating the collision avoidance system in action. At 256 s into the test, the ASV has just reached WP 3 and is sailing towards WP 4. The detection system identifies buoy number 2 and plans a maneuver to avoid it (Fig. 15(a)). Shortly afterward, an unexpected obstacle enters the field of view of the LIDAR (Fig. 15(b)), a swimmer nearby. The CAV module immediately replans the trajectory to maintain a safe distance from the swimmer (Fig. 15(c)). As the swimmer moves away and disappears

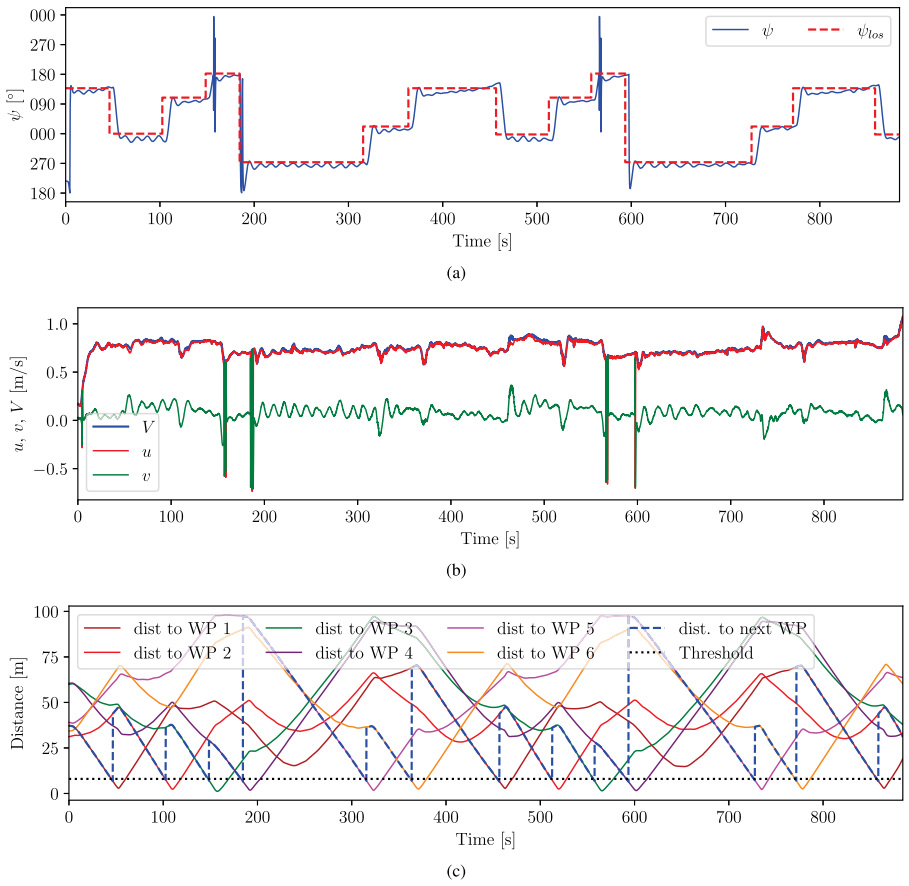


Fig. 12. Relevant data measured during the LOS experiment: ASV's heading 12(a), speed and velocity components 12(b), distances from the waypoints 12(c).

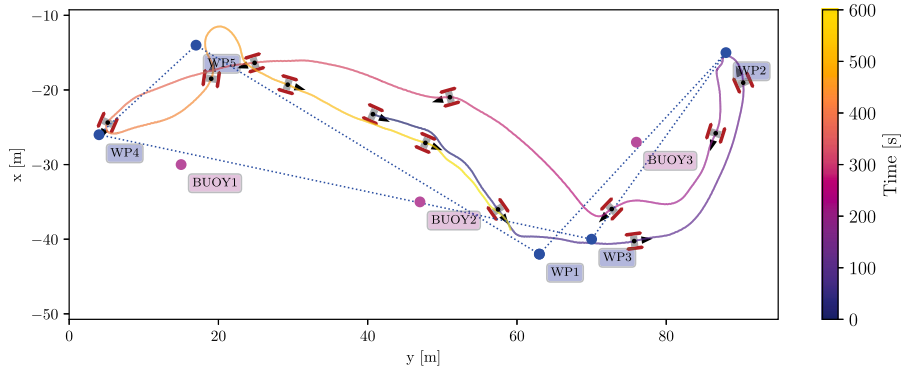


Fig. 13. ASV's trajectory (solid line) and planned path (blue dashed line) during the collision avoidance experiment.

from the FoV of the LiDAR (Fig. 15(d)), the ASV continues along its path until buoy 1 arrives in view (Fig. 15(e)), prompting further trajectory adjustment to maintain safe clearance (Fig. 15(f)). This sequence highlights the ability of the system to dynamically replan in real time, ensuring safe navigation around both static and unexpected obstacles.

In the sequence, it is also worth noting how the obstacle detection system identifies the position quite accurately. However, it is essential to highlight that the position of the buoys is only known approximately, as they move continuously due to disturbances such as wind. It is therefore impossible, in the current configuration, to separate the error in identifying the position of the buoys caused by the perception/proprioception pipeline from the actual movement of the buoys due to disturbances.

Finally, Fig. 16 shows some examples of obstacles acquired by the onboard sensors. In particular, Figs. 16(a), 16(c) and 16(e) show RGB images of two different buoys and the swimmer cited above. The images were acquired using a camera positioned onboard the ASV during the experiment. Figures Figs. 16(b), 16(d) and 16(f) show the corresponding LiDAR point clouds. The swimmer, barely visible in the RGB images, is effectively acquired by the LiDAR despite being mostly below the surface level, although with lower point cloud density compared to the buoys.

7. Discussion

The experimental results demonstrate the effectiveness and robustness of the proposed ASV architecture for both waypoint navigation and reactive collision avoidance. In the LOS experiment, the ASV successfully followed the predefined polygon at an average speed of 0.76 m/s, completing over two laps without human intervention. This

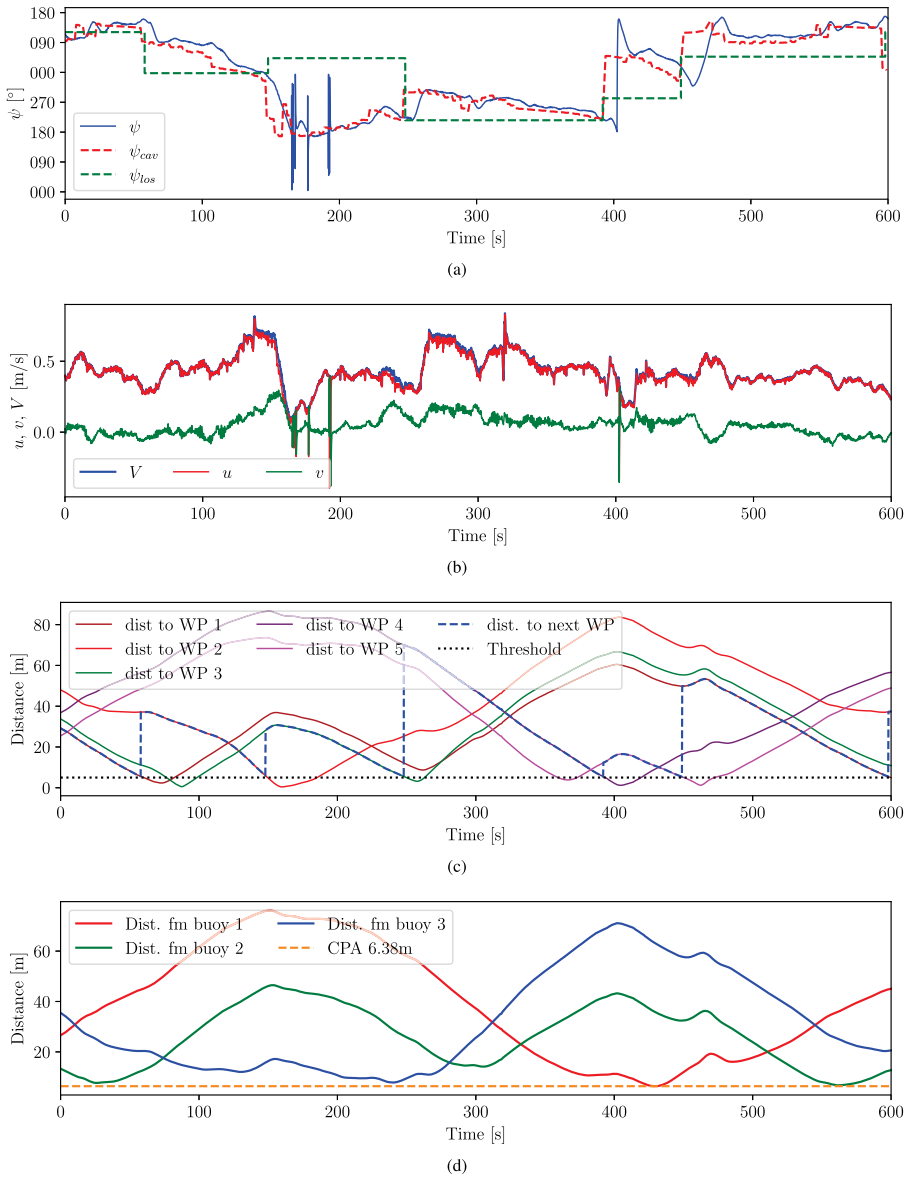


Fig. 14. Relevant data measured during the collision avoidance experiment: ASV's heading 14(a), speed and velocity components 14(b), distances from the waypoints 14(c), distances from the obstacles 14(d).

confirms that the implemented LOS guidance and control algorithm, combined with the waypoint generator and path manager, provides reliable navigation under ideal conditions, validating the core software architecture and message-passing framework.

The collision avoidance (CAV) experiment further highlights the adaptive capabilities of the system. When obstacles, including buoys and an unexpected swimmer, were detected, the ASV replanned its trajectory in real time, maintaining a safe distance of over 5 m while still reaching all designated waypoints. This demonstrates that the integration of perception (PRC), state estimation (SE), and collision avoidance (CAV) modules effectively supports autonomous decision-making in dynamic environments. The observed reduction in speed to 0.43 m/s compared to the LOS test reflects the additional maneuvers required to maintain safety in the presence of obstacles.

A key strength of the proposed system is its modular and open architecture. The MQTT-based messaging framework allows asynchronous operation of computationally intensive modules on separate hardware nodes while maintaining coordinated control. This modularity makes it straightforward to integrate additional sensors, implement more advanced control algorithms (Caharija et al., 2016; Martinovic et al., 2024), or expand the system for other autonomous tasks by extending the node network without requiring fundamental redesigns.

Some limitations of the presented work need to be highlighted. The current LOS-based guidance algorithm is simple, limiting the ASV's ability to precisely follow complex trajectories or compensate for disturbances. Additionally, while MQTT provides flexible communication between modules, it could possibly pave the way to exploit potential cybersecurity vulnerabilities (Longo et al., 2024). This issue must be addressed in any real-world critical application. Nevertheless, a broker-based architecture always presents a single point of failure, which is the broker itself. This problem could be addressed by ensuring broker redundancy, or by leveraging broker-less publish–subscribe messaging protocols such as DDS (Pardo-Castellote, 2004; Strelkovskaya and Zolotukhin, 2021), or ZeroMQ (Hintjens, 2013; Kang et al., 2020), which would ensure the same abstract structure, yet significantly increase the reliability.

Finally, some limitations of the SE module were observed at points of discontinuity in the heading representation. To address this issue,

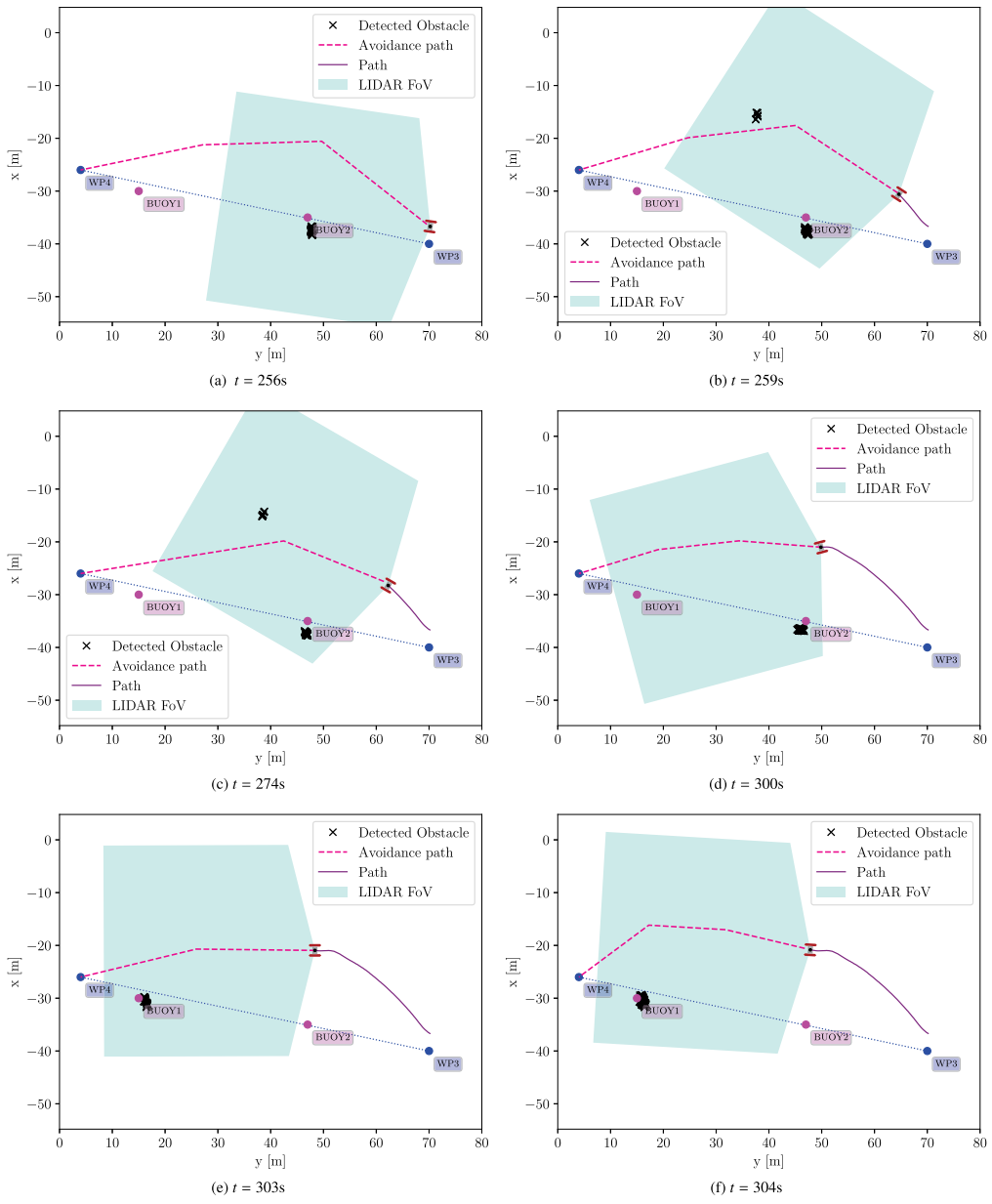


Fig. 15. An example of the collision avoidance system in action: 15(a) planned maneuver after detecting the first buoy; 15(b) an unexpected swimmer appears; 15(c) trajectory adjusted to avoid the swimmer; 15(d) the second buoy is not yet detected by the LiDAR; 15(e) the LiDAR detects the second buoy; 15(f) trajectory adjusted to safely avoid the second buoy.

the authors plan to adopt a component-based representation in future work, with the associated nonlinearity handled through an Unscented Kalman Filter.

The lake environment is different from the open sea but still valid and representative for a model unit approximately one meter long, for which the environmental conditions of the open sea would be at the limit of the platform's capabilities.

This paper did not address the influence of vessel motions on the performance of the onboard LiDAR system. The experimental context presented in this paper led to small-amplitude motions that did not require compensation. Nevertheless, to address scenarios with larger vessel motions, two extensions are under development. The first involves incorporating vertical motions into the filtering framework. The second is to enhance the state estimation system with a complementary filter specifically designed to provide accurate estimates vertical motions.

The collision avoidance approach used in this paper treated dynamic obstacles as fixed at the positions where the perceptual sensors detected them. Then it exploited continuous trajectory updates to avoid them even when they were moving. This approach, despite being approximate, proved effective with slow obstacles. However, obstacle tracking by the sensing system and modeling their motion within the path-planning algorithm would improve the system's performance.

The framework presented will be extended in the future to handle dynamic obstacles. The detection system can be extended to track dynamic obstacles by estimating their velocity and course angle using both single-sensor (Ponzini et al., 2025) and multi-modal (Ponzini and Martelli, 2025) approaches. The path planning algorithm used can handle dynamic obstacles if their velocity vectors are known (Zaccone, 2021, 2024; Zaccone and Martelli, 2023), planning evasive maneuvers based on predictions of the obstacles' motion. The individual modules have been demonstrated as standalone units in simulated and

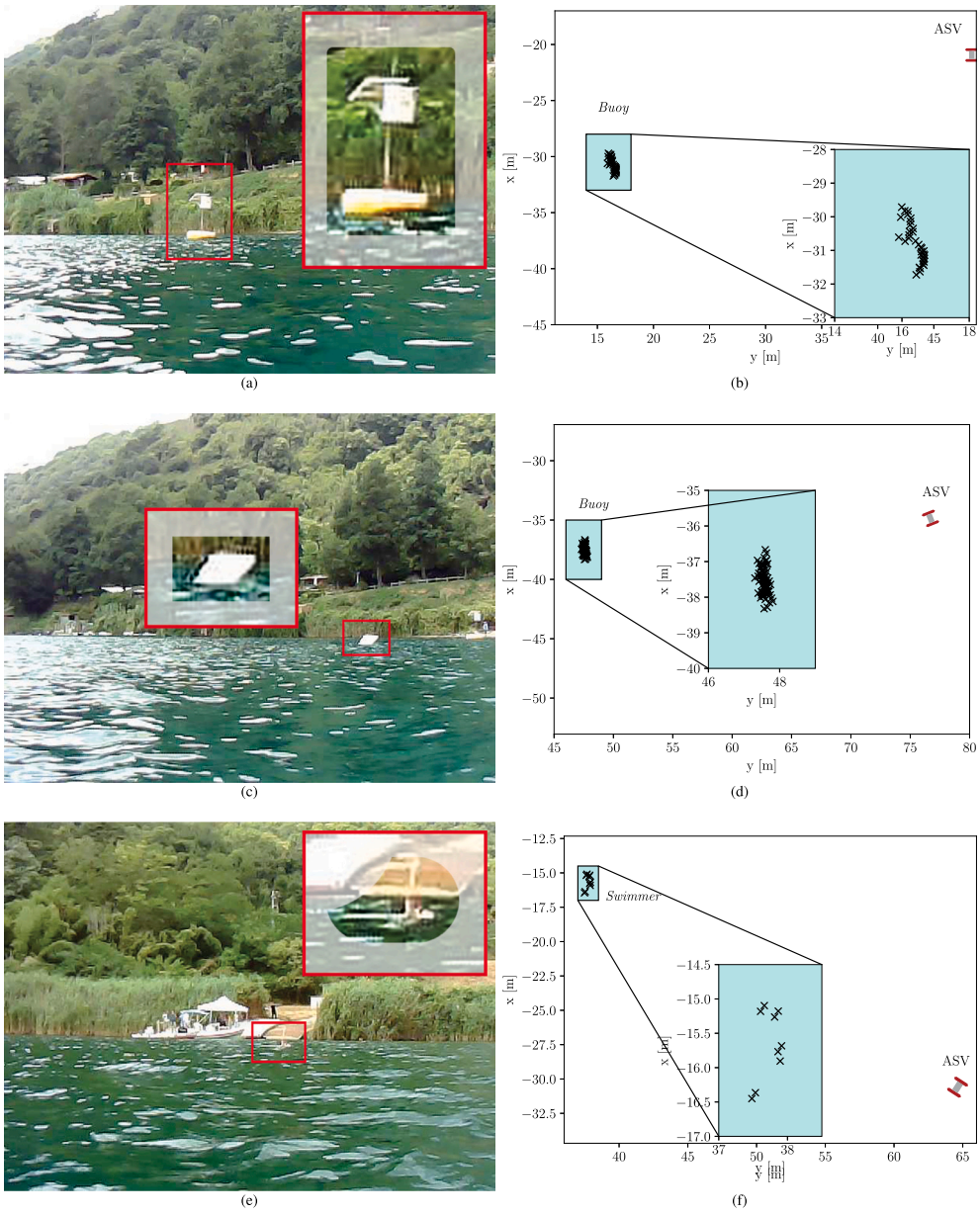


Fig. 16. Examples of obstacles: RGB views from a camera onboard the ASV and the corresponding LiDAR subsampled point clouds. A large buoy (16(a), 16(b)), a smaller buoy (16(c), 16(d)), and the swimmer cutting across the ASV's bow (16(e), 16(f)). Notice the different point cloud density for obstacles which are mostly above and below the water surface.

experimental scenarios; nevertheless, the integration of these upgraded modules will be the subject of future development and experimentation.

Overall, the experiments validate the architecture as a practical and adaptable solution for autonomous navigation and reactive collision avoidance. The modularity and open design of the system provide a solid foundation for future development, allowing enhancements in perception, planning, or control algorithms while maintaining robust operation. In the sight of extending the architecture to larger boats or ships, the integration of more sophisticated perception algorithms capable of detecting, classifying, and tracking obstacles (Ponzini et al., 2025; Ponzini and Martelli, 2025) could enable the implementation of COLREG-compliant collision avoidance (Zaccone and Martelli, 2023; Zaccone, 2024). With a view to higher TRL applications, the use of 4G/5G technology will be evaluated to increase the system's communication range, but with the primary objective of bringing as much of the computing effort on board as possible, to reduce latency and make the ASV as autonomous as possible.

8. Conclusions

This work presented a modular and open architecture for autonomous surface vessels, capable of waypoint navigation and real-time collision avoidance. The proposed system integrates perception, state estimation, collision avoidance, guidance, and control modules using an MQTT-based messaging framework, allowing asynchronous operation across multiple hardware nodes.

Experimental results were conducted in Lake Nemi using the SWAMP ASV as a research platform, and demonstrated that the architecture is effective and robust. In the LOS experiment, the ASV successfully followed a predefined polygon with minimal human intervention, validating the baseline navigation pipeline. In the collision avoidance experiment, the ASV detected and avoided unexpected obstacles, maintaining safe distances while reaching all assigned waypoints. These results highlight the reactive capabilities of the system and confirm the feasibility of real-time autonomous decision-making in dynamic environments.

The strength of the proposed approach lies in its modularity and openness, enabling straightforward integration of additional sensors, advanced control algorithms, and additional autonomous functionalities. Limitations include the simplicity of the current guidance and control algorithm and potential vulnerabilities associated with the MQTT communication platform. Despite these constraints, the architecture provides a flexible foundation framework for future development and experimental testing of autonomous maritime systems.

Future work will focus on improving the control strategies, integrating more sophisticated perception algorithms, and enhancing the communication framework to increase range and performance.

CRediT authorship contribution statement

Raphael Zaccone: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Filippo Ponzini:** Writing – review & editing, Software, Methodology, Investigation, Data curation, Conceptualization. **Michele Martelli:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was partially funded by the Compass Lab of Genoa University and by the European Union – NextGenerationEU. *Piano Nazionale di Ripresa e Resilienza, Missione 4 Componente 2 Investimento 1.4 "Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S su alcune Key Enabling Technologies"*. Code CN00000023 – Title: "Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)".

However, views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible.

The authors also express their gratitude to the CNR Institute of Marine Engineering for making their experimental basin available and for their kind support.

References

Bellman, R., 1954. The theory of dynamic programming. Bull. Am. Math. Soc. 60, 503–515.

Brekke, E.F., Eide, E., Eriksen, B.O.H., Wilthil, E.F., Breivik, M., Skjellaug, E., Helgesen, A.M., Martinsen, A.B., Thyri, E.H., et al., 2022. Milliampere: An autonomous ferry prototype. In: Journal of Physics: Conference Series. IOP Publishing, 012029. http://dx.doi.org/10.1088/1742-6596/2311/1/012029.

Caccia, M., 2006. Autonomous surface craft: prototypes and basic research issues. In: 2006 14th Mediterranean Conference on Control and Automation. IEEE, pp. 1–6. http://dx.doi.org/10.1109/MED.2006.328786.

Caharija, W., Pettersen, K.Y., Bibuli, M., Calado, P., Zereik, E., Braga, J., Gravdahl, J.T., Sørensen, A.J., Milovanović, M., Bruzzone, G., 2016. Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments. IEEE Trans. Control Syst. Technol. 24, 1623–1642. http://dx.doi.org/10.1109/TCST.2015.2504838.

Carlson, D.F., Akbulut, S., Rasmussen, J.F., Hestbech, C.S., Andersen, M.H., Melvad, C., 2023. Compact and modular autonomous surface vehicle for water research: The naval operating research drone assessing climate change (nordacc). HardwareX 15, e00453. http://dx.doi.org/10.1016/j.ohx.2023.e00453.

Costanzi, R., Fenucci, D., Manzari, V., Micheli, M., Morlando, L., Terracciano, D., Caiti, A., Stifani, M., Tesei, A., 2020. Interoperability among unmanned maritime vehicles: review and first in-field experimentation. Front. Robot. AI 7 (91), http://dx.doi.org/10.3389/frobt.2020.00091.

Donnarumma, S., Zaccarian, L., Alessandri, A., Vignolo, S., 2016. Anti-windup synthesis of heading and speed regulators for ship control with actuator saturation. pp. 1284–1290. http://dx.doi.org/10.1109/ECC.2016.7810466.

Dunbabin, M., Grinham, A., Udy, J., 2009. An autonomous surface vehicle for water quality monitoring. In: Australasian Conference on Robotics and Automation. ACRA, Sydney, Australia, pp. 2–4.

Eide, E., Breivik, M., Brekke, E.F., Eriksen, B.O.H., Wilthil, E., Helgesen, E.H., Veitch, E., Alsos, O.A., Johansen, T.A., 2025. The autonomous urban passenger ferry milliampere2: Design and testing. J. Offshore Mech. Arct. Eng. 147, 031409. http://dx.doi.org/10.1115/1.4067370.

Faggioni, N., Ponzini, F., Martelli, M., 2022. Multi-obstacle detection and tracking algorithms for the marine environment based on unsupervised learning. Ocean Eng. 266, 113034. http://dx.doi.org/10.1016/j.oceaneng.2022.113034.

Ferretti, R., Bibuli, M., Bruzzone, G., Odetti, A., Aracri, S., Motta, C., Caccia, M., Rovere, M., Mercorella, A., Madricardo, F., et al., 2023. Acoustic seafloor mapping using non-standard asv: Technical challenges and innovative solutions. In: OCEANS 2023-Limerick. IEEE, pp. 1–6. http://dx.doi.org/10.1109/OCEANSLimerick52467.2023.10244670.

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons.

Fruzzetti, C., Buzzurro, C., Donnarumma, S., Martelli, M., 2024. Time domain design of a marine target tracking system accounting for environmental disturbances. J. Mar. Sci. Eng. 12, http://dx.doi.org/10.3390/jmse12112058.

Gogendeau, P., Bonhommeau, S., Fourati, H., Julien, M., Contini, M., Chevrier, T., Nieblas, A.E., Bernard, S., 2025. An autonomous surface vehicle for acoustic tracking, bathymetric and photogrammetric surveys. Ocean Eng. 331, 121201. http://dx.doi.org/10.1016/j.oceaneng.2025.121201.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, M., Peterson, P., Gérard-Marchant, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. http://dx.doi.org/10.1038/s41586-020-2649-2.

Hinostroza, M.A., Eide, E., Brekke, E.F., Breivik, M., Lekkas, A.M., Skjetne, R., Håland Bryne, T., Gusev, A., Gudahl Tufte, A., 2025. Milliampere1 autonomous ferry prototype: Hardware and software. In: International Conference on Offshore Mechanics and Arctic Engineering. American Society of Mechanical Engineers, http://dx.doi.org/10.1115/OMAE2025-155401, V003T06A051.

Hintjens, P., 2013. ZeroMQ: Messaging for Many Applications. O'Reilly Media, Inc..

Kalliovaara, J., Jokela, T., Asadi, M., Majd, A., Hallio, J., Auranen, J., Seppänen, M., Putkonen, A., Koskinen, J., Tuomola, T., et al., 2024. Deep learning test platform for maritime applications: Development of the em/s salama unmanned surface vessel and its remote operations center for sensor data collection and algorithm development. Remote. Sens. 16, 1545. http://dx.doi.org/10.3390/rs16091545.

Kang, Z., Canady, R., Dubey, A., Gokhale, A., Shekhar, S., Sedlacek, M., 2020. A study of publish/subscribe middleware under different iot traffic conditions. In: Proceedings of the International Workshop on Middleware and Applications for the Internet of Things. pp. 7–12.

Karaki, A.A., Bibuli, M., Caccia, M., Ferrando, I., Gagliolo, S., Odetti, A., Sguerso, D. and, 2022. Multi-platforms and multi-sensors integrated survey for the submerged and emerged areas. J. Mar. Sci. Eng. 10 (753), http://dx.doi.org/10.3390/jmse10060753.

Klabnik, S., Nichols, C., 2023. The Rust Programming Language. No Starch Press.

Labbe, R.R., 2018. Filterpy documentation.

Light, R.A., 2017. Mosquitto: server and client implementation of the mqtt protocol. J. Open Source Softw. 2 (265).

Longo, G., Martelli, M., Russo, E., Merlo, A., Zaccone, R., 2024. Adversarial waypoint injection attacks on maritime autonomous surface ships (mass) collision avoidance systems. J. Mar. Eng. Technol. 23, 184–195. http://dx.doi.org/10.1080/20464177.2023.2298521.

Luo, Z., van Baelen, S., Slaets, P., Bruyninckx, H., 2024. Enhancing inland waterway transport through lidar-based updates to electronic navigational charts. In: OCEANS 2024-Halifax. IEEE, pp. 01–08. http://dx.doi.org/10.1109/OCEANS55160.2024.10754408.

Martinovic, L., Zecevic, Z., Bibuli, M., Caccia, M., Nad, D., 2024. Adaptive observer-based line-of-sight guidance law for path following of underactuated unmanned surface vehicle with sideslip compensation. IEEE Access 12, 144992–145002. http://dx.doi.org/10.1109/ACCESS.2024.3471790.

Maybeck, P.S., 1990. The Kalman Filter: An Introduction To Concepts. Springer New York, New York, NY, pp. 194–204. http://dx.doi.org/10.1007/978-1-4613-8997-2_15.

Nambiar, M.S., Ramakrishnan, A., Lakshman, A., PV, A., et al., 2023. Autonomous water quality monitoring surface vehicle. In: 2023 IEEE International Conference on Recent Advances in Systems Science and Engineering. RASSE, IEEE, pp. 1–5. http://dx.doi.org/10.1109/RASSE60029.2023.10363576.

Odetti, A., Bruzzone, G., Altosole, M., Viviani, M., Caccia, M., 2020. Swamp, an autonomous surface vehicle expressly designed for extremely shallow waters. Ocean Eng. 216, 108205. http://dx.doi.org/10.1016/j.oceaneng.2020.108205.

Pardo-Castellote, G., 2004. OMG Data-Distribution Service (DDS): Architectural Overview. Technical Report, Real-time innovations Inc Sunnyvale CA..

Ponzini, F., Fruzzetti, C., Sabatino, N., 2024. Real-time critical marine infrastructure multi-sensor surveillance via a constrained stochastic coverage algorithm. In: International Ship Control Systems Symposium. ISCSS 2024, p. 62.

Ponzini, F., Martelli, M., 2025. Marine obstacles multi-modal detection, classification and tracking via camera-lidar late fusion. In: 2025 IEEE International Workshop on Metrology for the Sea; Learning To Measure Sea Health Parameters. MetroSea, pp. 235–240. http://dx.doi.org/10.1109/MetroSea66681.2025.11245686.

Ponzini, F., Zaccone, R., Martelli, M., 2023. A multi-sensor indoor tracking system for autonomous marine model-scale vehicles. In: Journal of Physics: Conference Series. IOP Publishing, 012008.

Ponzini, F., Zaccone, R., Martelli, M., 2025. Lidar target detection and classification for ship situational awareness: A hybrid learning approach. Appl. Ocean Res. 158, 104552.

Qadir, M.I., Mumtaz, R., Manzoor, M., Saleem, M., Khan, M.A., Charlesworth, S., 2024. Unmanned surface vehicle for intelligent water quality assessment to promote sustainable human health. Water Supply 24, 2259–2270. http://dx.doi.org/10.2166/ws.2024.141.

Snyder, J.P., 1987. Map projections: A working manual. In: Professional Paper. vol. 1395, U.S. Government Printing Office, http://dx.doi.org/10.3133/pp1395.

Soni, D., Makwana, A., 2017. A survey on mqtt: a protocol of internet of things (iot). In: International Conference on Telecommunication, Power Analysis and Computing Techniques. ICTPACT-2017, pp. 173–177.

Standard, O., 2019. Mqtt version 5.0. URL https://docs.oasis-open.org/mqtt/mqtt/v5.0 1, 29.

Stanghellini, G., Bianco, F.Del., Gasperini, L., 2020. Openswap, an open architecture, low cost class of autonomous surface vehicles for geophysical surveys in the shallow water environment. Remote. Sens. 12, 2575. http://dx.doi.org/10.3390/rs12162575.

Strelkovskaya, I., Zolotukhin, R., 2021. Research the traffic characteristics of mqtt and dds protocols in low-bandwidth radio networks. In: 2021 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics. UkrMiCo, IEEE, pp. 137–141.

Van Rossum, G., 1995. Python reference manual.

You, X., Ma, F., Lu, S., Liu, J., Yan, X., 2020. An integrated platform for the development of autonomous and remote-control ships. In: Porceedings of the 19th Conference on Computer and IT Applications in the Maritime Industries. COMPIT 2020, pp. 316–327.

Zaccone, R., 2021. Colreg-compliant optimal path planning for real-time guidance and control of autonomous ships. J. Mar. Sci. Eng. 9 (405), http://dx.doi.org/10.3390/jmse9040405.

Zaccone, R., 2024. A dynamic programming approach to the collision avoidance of autonomous ships. Mathematics 12 (1546), http://dx.doi.org/10.3390/math12101546.

Zaccone, R., Martelli, M., 2023. Interaction between colreg-compliant collision avoidance systems in a multiple mass scenario. In: Journal of Physics: Conference Series. IOP Publishing, 012006. http://dx.doi.org/10.1088/1742-6596/2618/1/012006.

Zhang, Y.Y., Shuai, J., Billet, J., Slaets, P., 2023. Design and build of an autonomous catamaran urban cargo vessel. In: Journal of Physics: Conference Series. IOP Publishing, 012002. http://dx.doi.org/10.1088/1742-6596/2618/1/012002.

Zheng, H., Liu, C., 2025. An overview of unmanned surface vehicles: Methods, practices, and applications. Control Eng. Pract. 164, 106479. http://dx.doi.org/10.1016/j.conengprac.2025.106479.