

TDP: DronKab (Mexico), Indoors Category

L. Carmona Vieyra^{1*}, A.E. Ibarra Paz¹, M. Torres Rivera¹, E. Bustamante Arroyo¹, M. Mendoza García¹,
M. Sanchez Herrera¹

¹Universidad Aeronáutica en Querétaro, Carretera Estatal 200 Querétaro – Tequisquiapan No. 22154
Col. Parque Aeroespacial de Querétaro. Colón, Querétaro. México

*lucycarmonavieyra@gmail.com

ABSTRACT

This Technical Description Paper presents the solutions proposed by the DronKab team for indoor missions in the 2025 International Micro Air Vehicle (IMAV) Competition. The competition focuses on autonomous drones for search and rescue in post-disaster environments, where unstable structures and limited human access pose challenge.

Key contributions include a Gazebo-based simulation environment, structural adaptations of commercial MAV frames for mission-specific tasks such as medical kit delivery and aerial whiteboard interaction, and the implementation of autonomous mapping and navigation algorithms with 2D LiDAR, neural networks, and sound detection via a microphone array. These capabilities enhance perception and control, enabling tunnel traversal, precision landing, and interactive panel manipulation.

Beyond addressing the competition's missions, the solutions presented here demonstrate a systematic approach to designing customized MAVs by integrating mechanical adaptations, electronic components and advanced algorithms. This integration lays the groundwork for the development of autonomous aerial systems with direct applicability to real-world search-and-rescue operations in post-disaster scenarios.

1 INTRODUCTION

The International Micro Air Vehicle (IMAV) Competition and Conference 2025 focus on advancing innovative autonomous drones for search-and-rescue operations in post-disaster environments. These missions are characterized by unstable structures, limited visibility, and restricted human access, which demand robust aerial systems capable of operating reliably under constrained conditions. Within this context, Micro Air Vehicles (MAVs)—small-scale aerial platforms designed to

perform fully autonomous tasks—are positioned as effective tools to address these challenges.

Building upon this framework, the DronKab team approaches the indoor missions of the competition through the development of customized MAVs. These platforms originate from commercial frames but are progressively adapted to mission needs by means of structural modifications, mechanical add-ons, and the integration of commercial electronic components. The process is complemented by the development and implementation of algorithms that enable autonomous mapping, navigation, and interaction, ensuring that each aerial system is tailored to the specific tasks defined by the competition.

The key contributions of this work include the creation of a Gazebo-based simulation environment for algorithm testing, the design of mechanical and structural adaptations for tasks such as medical kit delivery and aerial whiteboard interaction, and the implementation of perception and navigation modules based on 2D LiDAR, deep neural networks, and visual marker detection. These enhancements enable complex operations such as tunnel traversal, precision landing, and panel interaction.

Finally, this paper is organized as follows. Section II provides the background and theoretical foundations that guided the design decisions. Section III presents the methodology, describing the integration process from mechanical design, electronics, and sensor configuration to the development of software layers and algorithms. Section IV details the implementation and partial results, including simulations that provide preliminary evidence of the feasibility of the proposed approaches. Section V discusses conclusions and outlines future steps, followed by references in Section VI.

2 BACKGROUND

GPS-denied environments have become a significant challenge for autonomous drone navigation. In this context, odometry plays a crucial role in estimating the

position and orientation of MAVs. In [1], the authors propose RF2O, a method for estimating the planar odometry of mobile robots using a 2D LiDAR. The technique is based on the range flow equation, formulated as a dense optimization problem with robust estimators to handle dynamic and low-informative environments. Experimental results showed that RF2O is more accurate and up to twenty times faster than methods such as PL-ICP and PSM.

For successful autonomous navigation, it is also necessary to generate a map of the environment. Several tools have been developed to address this problem. In [2], the authors present a software package designed to tackle mapping and localization challenges in large and dynamic environments, such as warehouses, retail stores, or disaster zones, where maps must be frequently updated. This package provides multiple operating modes, including synchronous mapping, asynchronous mapping, and pure localization. Furthermore, [3] introduces a navigation system whose modular architecture is based on behavior trees, enabling route planning, motion control, and recovery strategies in a flexible and safe manner. This approach allows for reliable and secure autonomous navigation in complex and dynamic scenarios.

In [4] presents a navigation system that enables a quadcopter to operate autonomously in indoor environments without relying on GPS. Using the Gazebo simulator, the authors modeled the behavior of a quadcopter equipped with a LiDAR sensor. This setup was used to perform SLAM, after which the ROS 2 Nav2 package was employed to achieve autonomous navigation of the aerial vehicle. The study demonstrated the feasibility of applying Nav2, originally designed for ground robots to aerial platforms, with the limitation that navigation was constrained to the xy-plane.

With the development of LiDAR-based odometry SLAM, autonomous navigation of MAVs within a flight zone can be generally achieved. However, this approach is limited for operations requiring higher precision, such as controlled tunnel crossing, interaction with specific surfaces, acoustic source localization, and landing in confined areas. To overcome these limitations, it is necessary to integrate additional sensors, such as cameras and microphones, along with perception techniques based on deep neural networks, which enhance the robustness and versatility of navigation.

In tasks involving the identification of structural obstacles, such as the tunnels used in our MAV missions, convolutional neural network-based navigation systems

have been reported. In [5], for example, a YOLOv3-based model was implemented to recognize architectural elements in indoor environments, such as doors, windows, and intersections, using the monocular camera of a commercial quadcopter. This system was integrated into a modular ROS architecture, where detection is combined with autonomous control algorithms, enabling flight through narrow corridors and multiple intersections. The proposed methodology demonstrates that the detection of structural features can compensate for the absence of additional sensors and provide reliable navigation in GPS-denied environments. A complementary approach is presented in [6], which not only detects doors and windows but also classifies their state (open or closed).

Regarding target detection, such as MAV landing platforms, it is relevant to extend this visual identification logic to the estimation of safe contact zones. Several studies have employed neural networks specifically for landing site detection. For instance, in [7], a method based on image segmentation models was proposed to identify safe landing zones in indoor environments.

In addition to neural networks, other approaches for autonomous navigation using visual perception have been reported. In [8], an autonomous navigation system for MAVs in indoor environments is presented, relying on OpenCV ARUCO marker detection to estimate the MAV's position and orientation in space. This information is then used to implement control algorithms that adjust the MAV's trajectory accordingly.

In previous work, Grondin et al. [9, 10] introduced the ODAS framework, designed for robust sound source localization on low-cost embedded hardware. The system implements the GCC-PHAT algorithm and a modified 3D Kalman filter (M3K), optimized through hierarchical search strategies and hemisphere restriction models. These features enable Direction of Arrival (DOA) estimation on platforms such as the Raspberry Pi 4. Its selection as a reference is justified by its computational efficiency and feasibility for integration into aerial vehicles.

3 METHODOLOGY

The design methodology considered the adaptation of a commercial frame and the integration of a payload release system. An Mk4 7-inch frame was employed as the base platform due to its lightweight and robust construction. Structural modifications were performed by extending the plates to increase internal space and by incorporating a custom landing gear to enhance clearance and allow sensor installation. In addition, a servo-actuated release

mechanism was modeled in CAD and manufactured through 3D printing to enable controlled payload delivery during the indoor mission.

Hardware selection followed the guidelines provided by MicoAir Technology [11], which offer a validated integration framework. The MicoAir 743 Flight Controller, fully compatible with PX4, was employed as the central unit to ensure reliable flight control and peripheral compatibility. Manufacturer-recommended devices were integrated, including the Bluejay-48K 4-in-1 ESC for centralized motor management, and additional sensors to enhance the MAV's navigation and stability.

For propulsion, 9imod X2807 1500 KV motors were selected for their high torque and efficiency. Four Bluejay-48K ESCs were used, powered by a 3S LiPo battery (11.1 V, 5000 mAh), coupled with 5-inch tri-blade polycarbonate propellers, selected according to performance calculations. An OAK camera was also incorporated for vision-based tasks. System performance was preliminarily evaluated using the eCalc tool by inputting the MAV's configuration and dimensions.

Prior to developing the mission-specific algorithms, a simulation environment was implemented using Gazebo 7 (Gazebo Garden version). A world was defined through an SDF file, which described all the elements composing the simulated environment to reproduce the desired behavior. This included the definition of lighting points, the initial positioning of the UAV, the placement of objects such as window tunnels, platforms and whiteboard, as well as physical properties such as gravity, air density, and vehicle characteristics (e.g., inertia values). This virtual environment is intended to serve as a testbed for evaluating the functionality of the developed algorithms, thereby avoiding physical trials in cases where system safety cannot yet be guaranteed.

For mission execution, an autonomous navigation system was implemented, incorporating odometry and LiDAR data to generate a map necessary for trajectory planning. Additional autonomous navigation methods were also employed, including neural networks, ARUCO marker-based navigation, and acoustic source detection, all implemented using ROS2, PX4, and Gazebo.

For mapping, a 2D LiDAR sensor was integrated into the X500 model within Gazebo, a design provided by the PX4 Autopilot community. Simulation parameters, such as LiDAR detection range and coverage angle, were adjusted to better represent the MAV under development. To manage transformations between the MAV's components,

a URDF (Unified Robot Description Format) file was created. This file, together with the robot_state_publisher node, enables the construction of a TF2 tree that continuously publishes the position and orientation of each MAV component relative to the base, ensuring that sensors and navigation algorithms operate with consistent coordinates.

Odometry was implemented using the RF2O package, which subscribes to LiDAR data to estimate the MAV's position and orientation. Mapping relies on three components: LiDAR data, MAV odometry, and the transformation tree, generating an occupancy map. For NAV2-based navigation, the system requires the transformation tree, odometry, LiDAR data, and the previously generated SLAM map.

The simulation environment allowed observation of dynamic map generation and updates in RViz, including detected obstacles, walls, and other relevant objects. Autonomous navigation was visualized in RViz, displaying the planned path from the current position to the goal, along with obstacle avoidance and trajectory adjustments in response to environmental changes.

To perform tunnel-crossing missions, a neural network was trained to classify tunnels by size. A dataset of images captured from the X500 simulation model in Gazebo was used for training with YOLOv8. The trained model was exported to ONNX for ROS2 integration and implemented in a ROS2 node to execute inferences. The node calculates an error used by a separate control node to guide the MAV safely through tunnels.

Similarly, another neural network was trained to detect landing platforms. This model was integrated into ROS2, where the output error is used to execute a controller ensuring precise landing.

For ARUCO marker-based navigation, the ROS2 ARUCO_opencv package was employed to compute the MAV's position relative to visible markers. A custom node executes control commands to enable smooth MAV movements, such as drawing a line on the whiteboard in mission 2.

For acoustic source detection, the ReSpeaker USB Mic Array V3.1 was configured to capture four-channel audio. ODAS was employed for signal processing, with parameters tuned to enable source localization. Visualization was performed using the ODAS Web Studio GUI, displaying acoustic energy on a unit sphere. These experiments provide a foundation for future integration

with a Raspberry Pi 4, enabling the MAV to make navigation decisions based on real-time acoustic signals.

4 DEVELOPMENT AND IMPLEMENTATION OF AUTONOMOUS DRONES

4.1 Functional Description of the Drones

4.1.1 Mechanical Design and Structure

The Mk4 7-inch frame is selected as a lightweight and robust platform for long-range drone applications. The stock frame features a 295 mm wheelbase, carbon fiber construction with 5 mm arms and 2.5 mm top and bottom plates, and a weight of ~121 g. To adapt it to mission requirements, the frame underwent two structural modifications:

- Extended plates. The upper and lower carbon fiber plates were redesigned to reach a length of 210 mm, providing increased internal space for electronics, battery placement, and payload integration.
- Landing gear. A custom landing system was designed and manufactured, raising the clearance to 100 mm, which improves protection for onboard electronics and allows safe integration of cameras and sensors beneath the frame.



Figure 1: Modified frame

In addition to these modifications, a hook was developed for the controlled release of the load. The hook was designed to be opened upon actuation so that the rope that kept the payload in place could be let go. The rope contained a medical kit stored in a box of the size $20 \times 10 \times 5$ cm. The hook system was first developed in CAD using SolidWorks to achieve precise geometry and fit. Once the design was finalized, it was 3D printed to produce a working yet light component.

The actuation of the hook was achieved by a servo motor to facilitate precise and reliable control over the release process. The mechanism assembled underwent test runs to confirm that the hook securely held the rope during flight and released it on command. This setup ensured safe and precise delivery of the payload when conducting the indoor obstacle courses.

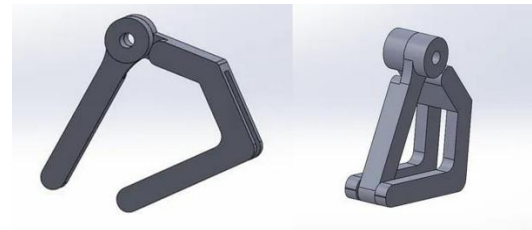


Figure 2: Hook system for med-kit delivery

4.1.2 Hardware Selection and Analysis

The MicoAir 743 Flight Controller, fully compatible with PX4, was employed as the core of the system. Following MicoAir Technology recommendations, the Bluejay-48K 4-in-1 ESC, the MTF-01 Optical & LiDAR Sensor, and the LR900-F telemetry module were integrated. For propulsion, 9imod X2807 1500 KV motors were selected due to their high torque and efficiency, powered by a 3S LiPo battery (11.1 V, 5000 mAh). The motors were coupled with 5-inch tri-blade polycarbonate propellers, chosen according to performance calculations. An OAK camera was also incorporated for vision-based tasks.

System performance was preliminarily evaluated using the eCalc tool, where the MAV configuration and dimensions were input. The results indicated that the maximum range occurs near 70 km/h, after which range decreases due to higher energy consumption. Under these conditions, the estimated flight time is approximately 14.2 minutes. These results are shown in Figure 1.



Figure 3: Range Estimation

4.1.3 Software Architecture

The system is composed of several subsystems classified according to their primary functions: SLAM for autonomous navigation, tunnel traversal, whiteboard

interaction, sound detection, and platform landing. The complete system is integrated through the implementation of ROS 2 nodes, which are described below.

For autonomous navigation, the architecture shown in Figure 4 provides a general overview of trajectory planning and current position estimation of the MAV, enabling the system to publish Twist messages to achieve autonomous flight.

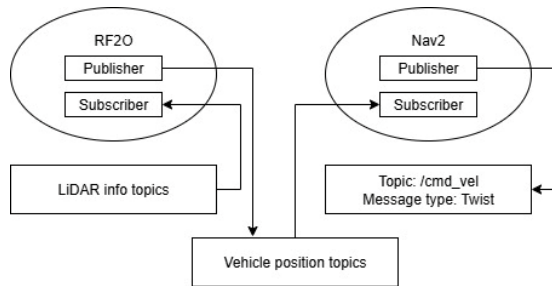


Figure 4: General ROS2 nodes and topics for autonomous MAV navigation

The diagrams presented in Figure 5 through Figure 8 illustrate the proposed architecture for the execution of the competition missions, including tunnel detections and traversal, platform landing, whiteboard interaction, and sound detection. These diagrams detail the nodes developed for each mission, as well as their publisher and subscribers, along with the corresponding message types.

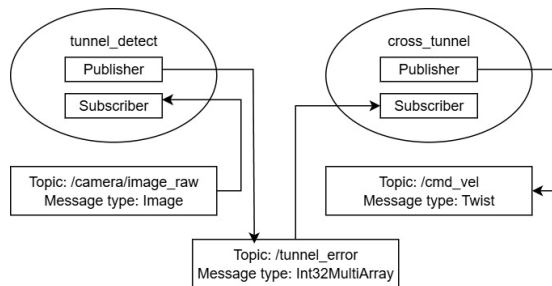


Figure 5: ROS2 nodes and topics to execute mission 1

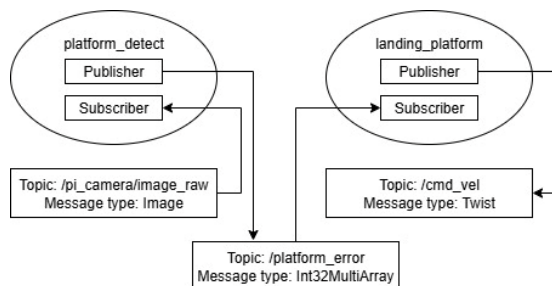


Figure 6: ROS2 nodes and topics for platform landing

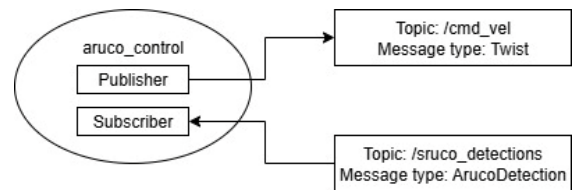


Figure 7: ROS2 nodes and topics to execute mission 2

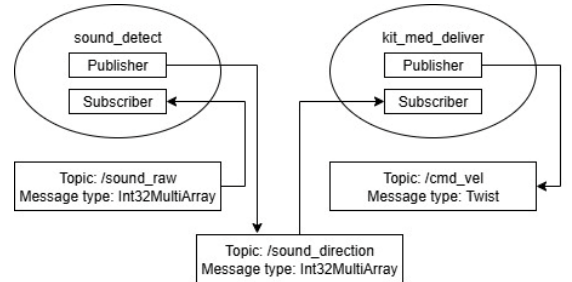


Figure 8: ROS2 nodes and topics to execute mission 3

4.2 Design Approach

The following subsections present the development of the solutions for each mission, outlining the specific requirements and the tools utilized for their execution. Additionally, the functioning of each MAV subsystem is described from a software perspective, emphasizing how these components collectively enable the successful completion of the missions.

4.2.1 Navigation and SLAM

Odometry plays a crucial role in the autonomous navigation of a MAV, as it determines its position and motion during displacement. In the case of ground robots, odometry can be calculated relatively easily by using the wheel radius, wheel separation, and revolutions per minute (RPM), which allows the estimation of the robot's position within its environment. However, this method of calculating odometry cannot be replicated in MAVs.

To address this problem, the RF2O [1] package was explored. Using a 2D LiDAR sensor and estimation algorithm, it can generate odometry in order to estimate the position and orientation of the MAV. At the same time, this package is characterized by its low computational cost, which makes it particularly suitable for the limitations of the onboard computer.

Odometry allows obtaining a global reference frame, which is essential for executing SLAM algorithms. This technique is used to map the area in which the robot will navigate. For this purpose, the SLAM Toolbox [2] package

is employed, which, based on data from the LiDAR sensor, can generate an occupancy map where obstacles are represented using a grayscale scale. The map is initially constructed through a manual traversal with the MAV and is then stored and used during the autonomous navigation phase.

To achieve MAV localization on the map, AMCL [12] (Adaptive Monte Carlo Localization) is used in conjunction with the Nav2 package, which combines the vehicle's odometry with LiDAR sensor readings on the map, thereby estimating its position and orientation within the environment. The Nav2 package provides a wide range of tools and algorithms for autonomous navigation, such as Path Planning, Obstacle Avoidance, Behavior Trees, etc. Using these functionalities, trajectories for the MAV are planned, and Twist-type messages are sent to execute movement. It is important to note that Nav2 is primarily designed for ground robots; therefore, navigation is currently limited to the XY plane, with no native support for movement along the Z axis

consists of 72 layers and was trained for 120 epochs, achieving accuracies of 99.3% for wide_tunnel, 98.5% for medium_tunnel, and 96.1% for small_tunnel.

For in-flight execution, two ROS 2 nodes were developed. The first one (tunnel detection node) receives images from the MAV's front-facing camera, runs the neural network model, and selects the most reliable detection corresponding to the target tunnel. Based on this detection, the node computes the error between the tunnel center and the image center along the horizontal and vertical axes, publishing these errors to the /tunnel_error topic.

The second node (control node) subscribes to the /tunnel_error topic and applies a proportional-derivative (PD) controller along the lateral and vertical axes to align the vehicle with the tunnel window. The controller runs every 20 ms and enforces maximum velocity limits to avoid instability. When the X and Y errors fall within a predefined threshold, the node allows forward motion along the longitudinal axis at constant velocity, ensuring that the MAV traverses the tunnel in a properly aligned and safe manner.

4.2.3 Blackboard Interaction

To simulate the vehicle's interaction with its environment, the MAV must draw a continuous line on a whiteboard. To achieve this, a visual marker located at the top-left corner of the whiteboard is used to estimate the distance between the MAV and the surface, as well as to position the MAV in front of the drawing area. Marker detection is performed using the ARUCO_opencv ROS 2 package, which provides the marker's position along the X, Y, and Z axes, as well as its orientation in quaternions.

A ROS 2 node was implemented to subscribe to the /ARUCO_detections topic to obtain the lateral, horizontal, and longitudinal distance between the MAV and the marker. Based on this information, a proportional-derivative (PD) controller aligns the MAV with the marker in the lateral and longitudinal axes and positions it at 30 cm from the whiteboard surface, while ensuring the MAV is oriented perpendicular to the whiteboard.

Once the MAV reaches the target position, it performs a lateral movement to the right to draw the line continuously across the whiteboard.

4.2.4 Acoustic Localization

The acoustic localization system is implemented using the optimized ODAS framework (Open Embedded Audition

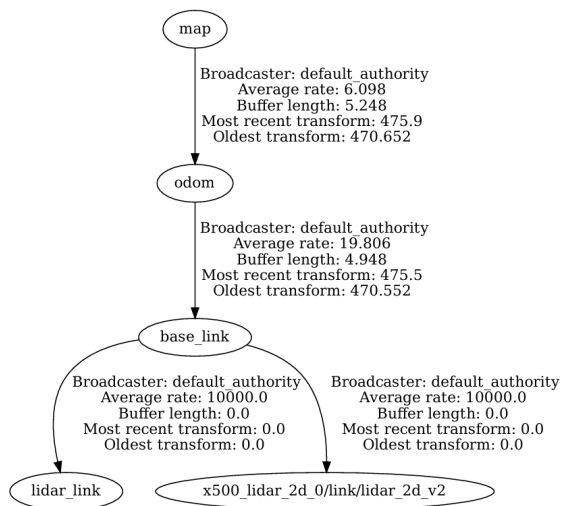


Figure 9: TF tree

4.2.2 Tunnel Crossing

At the beginning of mission execution, the MAV enters the flight zone by passing through a tunnel formed by windows. To identify the type of tunnel, a neural network was trained with three classes: "wide_tunnel", "medium_tunnel", and "small_tunnel". The network was trained using YOLOv8n, and the resulting model was exported to ONNX format for integration into ROS 2. The dataset used for training was built from 807 images of tunnels captured from different vehicle perspectives and was divided into three subsets: 70% for training, 20% for validation, and 10% for testing. The resulting network

System) [10]. Raw signals are captured from a ReSpeaker USB Mic Array V3.1, featuring four microphones in a circular configuration. This geometry is used for measuring the Time Difference of Arrival (TDOA) and phase shifts of sound waves across microphones, which is the fundamental principle for estimating the source direction.

Processing is centered on the SRP-PHAT algorithm (Steered Response Power with Phase Transform). For the 4-microphone array, the Generalized Cross-Correlation (GCC-PHAT) is computed for all 6 unique microphone pairs. The algorithm works by testing numerous candidate directions (azimuth and elevation). For each candidate, it calculates the theoretical TDOAs for every microphone pair, sums the corresponding GCC-PHAT correlation values at those precise delays, and selects the direction with the maximum summed power as the estimated Direction of Arrival (DOA).

To ensure performance on a Raspberry Pi 4, ODAS utilizes two critical optimizations [10] as detailed in Figure 10:

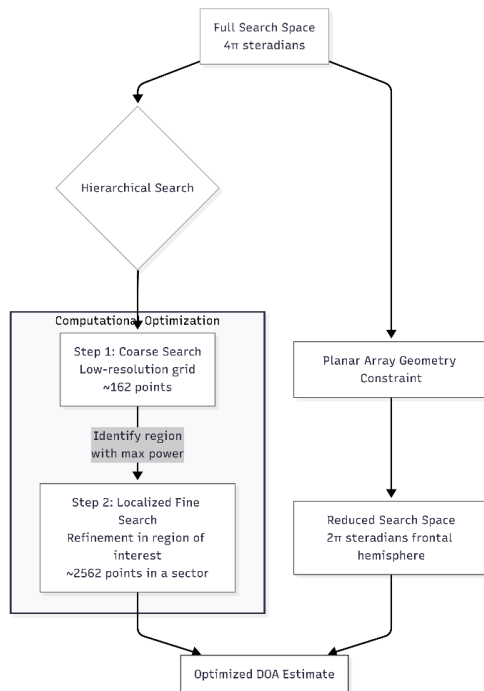


Figure 10: Optimizations for Efficient DOA Estimation in ODAS

- **Hierarchical Search:** The DOA estimation is split into two stages. First, a coarse search on a low-resolution grid (~162 points) identifies high-probability sectors. Second, a fine search refines the estimate within a confined sector of the high-resolution grid (~2562 points only in that sector), drastically reducing computational load.

- **Hemisphere Restriction:** The search space is constrained to the frontal hemisphere (2π steradians), leveraging the planar array geometry to discard acoustically unrealistic directions from below the array, instantly halving the number of points to evaluate.

This localization pipeline runs in a closed loop every 20 ms. The raw DOA estimates from each frame are smoothed and tracked over time using a modified 3D Kalman filter (M3K), which filters noise, manages source states, and maintains stable trajectories. This provides the MAV with continuous, updated directional feedback.

4.2.5 Precision Landing

Finally, the MAV lands on a platform identified through the execution of a neural network trained with YOLOv8n, whose model was exported to ONNX format for integration into ROS 2. The dataset used for training was built from 589 images of the platform captured from different angles by the onboard camera and was divided into three subsets: 70% for training, 20% for validation, and 10% for testing. The implemented network consisted of 72 layers and was trained for 120 epochs, achieving an accuracy of 98%.

For integration into the navigation system, a node of ROS 2 was developed to execute the detection model (platform detection node). This node receives images from the downward-facing camera, performs inference, and filters the detections. Once the platform is detected, the error between the center of the detection bounding box and the image center is computed. These errors along the X and Y axes are published to a topic named `/platform_error`.

Additionally, a second node was implemented to manage descent control and alignment (control node). This node subscribes to the `/platform_error` topic and applies a proportional-derivative (PD) controller along the horizontal and longitudinal axes to center the MAV over the platform. When the error falls within a predefined threshold, the controller generates a constant descending velocity to perform a safe landing.

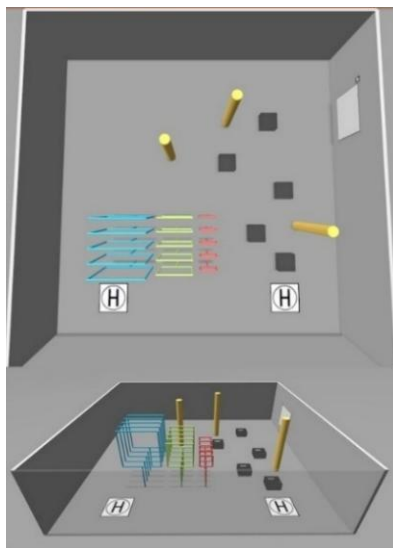


Figure 11: Gazebo world

5 RESULTS

For the initial experiments and the progress achieved thus far, a simulation environment was developed in Gazebo (Figure 11). This environment enabled the successful testing of mapping, trajectory generation, tunnel traversal, interaction with the whiteboard, and landing on the designated platform. The results of these experiments are presented below.

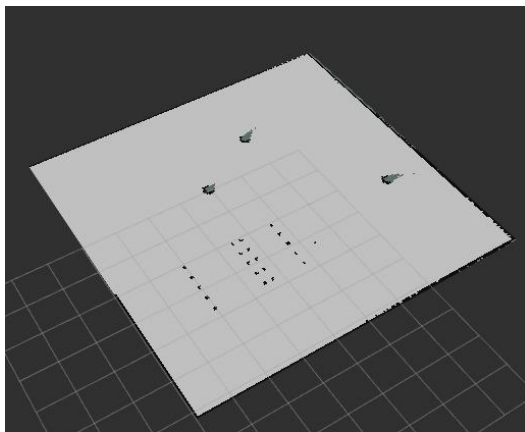


Figure 12: Map generated with slam toolbox

In Figure 12, the occupancy map generated by the MAV at a constant altitude can be observed, while it moves in the xy plane.

With the map previously generated using SLAM Toolbox, the cost map (Figure 10) is created, which is then used by the path planning to perform navigation.

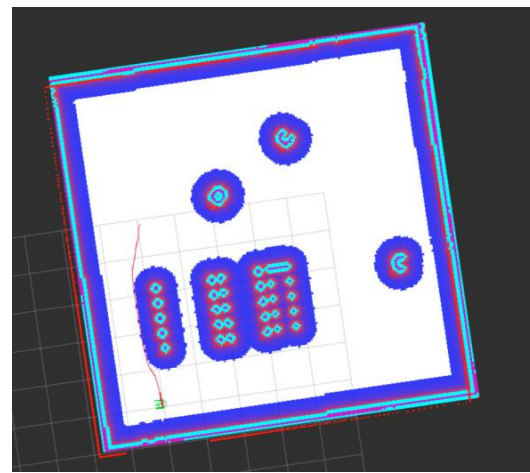


Figure 13: Path planning with NAV2

Regarding tunnel traversal, the system achieved consistent and successful detections (Figure 14) that enabled the MAV to pass through the selected tunnel without collisions or loss of control. Positive results were obtained when crossing both the wide tunnel and the medium tunnel; however, the MAV was unable to traverse the small tunnel due to its limited dimensions.

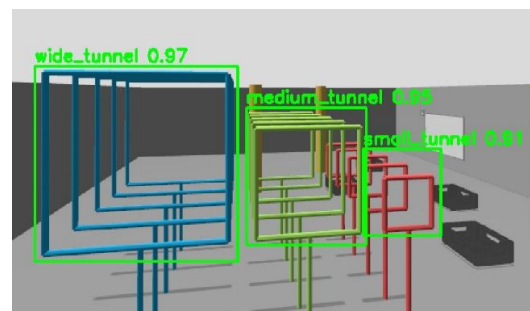


Figure 14: Tunnel detections

Continuing with the implementation of neural networks, accurate detections of the landing platform were obtained from different viewpoints of the MAV (Figure 15), even in cases where the platform was only partially visible. This capability proved essential for achieving precise landings on the designated platform.

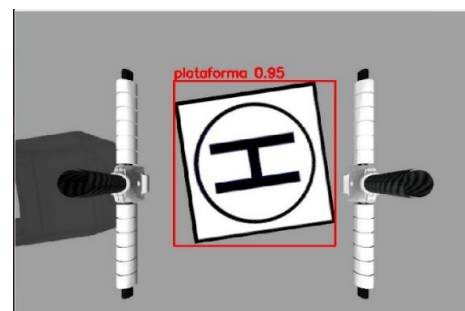


Figure 15: landing platform detections

Regarding the interaction with the whiteboard, drawing a continuous line in the simulation environment was not yet achieved. Nevertheless, reliable detections of the ARUCO marker were obtained, enabling smooth movements of the MAV through the implemented controller. These results (Figure 16) are expected to be valuable for the real execution of Mission 2.

For the acoustic localization component, the system validation was conducted in ROS2, where the functionality of the ReSpeaker USB Mic Array V3.1 was initially verified through multichannel audio recording and playback, ensuring correct signal acquisition and data quality. In parallel, the ODAS Web Studio interface was evaluated using a specific test code designed to generate a constant signal. In Figure 18, the results of the sound source visualization can be observed: the Source graph shows a fixed straight line at 0°, confirming the stable detection of a single acoustic source. Additionally, the Azimuth graph displays a slightly inclined straight line, demonstrating the consistency of the angular calculation without significant noise. The interface also exhibited a single color, corresponding to the detection of a single source, further corroborating its correct operation.

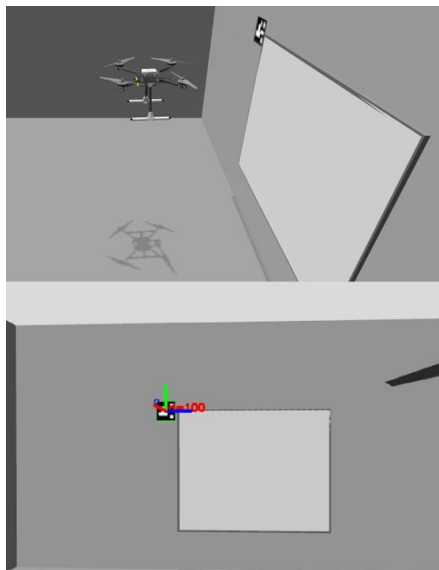


Figure 16: ARUCO detections

These findings validate both the audio acquisition by the ReSpeaker and the graphical representation. Finally, this stage lays the groundwork for integrating the six microphone pairs with the acoustic localization algorithm and the visual interface, aiming to identify multiple sound

sources and enable autonomous MAV navigation based on these signals.

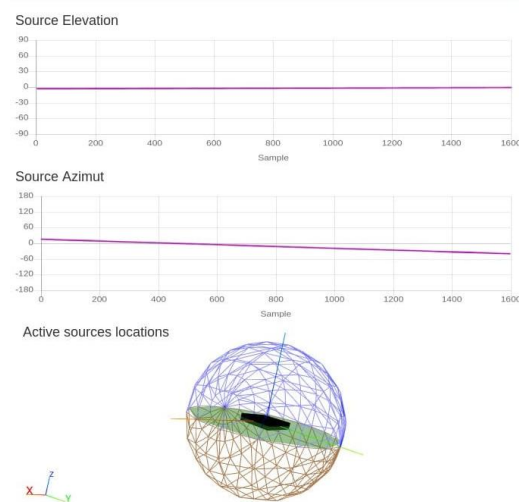


Figure 17: Graphical User Interface (GUI) for Data Visualization

6 CONCLUSIONS

This paper presents significant progress in the development of solutions for the indoor missions of the 2025 International Micro Air Vehicle (IMAV) Competition. The implementation of a Gazebo-based simulation environment allowed the validation of algorithms for mapping, trajectory planning, autonomous navigation, and visual marker detection, as well as interaction with mission-specific elements such as tunnels, landing platforms, and panels. To date, the results remain primarily in simulation, except for acoustic signal acquisition and processing using the microphone array; however, all developments were designed for subsequent implementation on the physical prototype.

Despite these advances, several areas still require refinement. Route planning needs further optimization to improve accuracy, and enhancements are necessary for landing on platforms under complex conditions, such as the presence of smoke or moving platforms. Additionally, MAV control for navigating toward sound sources is still under development, which will likely require the integration of filters to identify specific sounds and guide the MAV accurately for the delivery of medical kits.

These advances not only allow for the verification of the functionality of each MAV subsystem individually but also provide a great foundation for the complete integration of

all algorithms and systems to successfully accomplish all competition missions. Although some developments are still limited by ongoing hardware implementation and prototype design, the results achieved demonstrate that the proposed strategies constitute a robust framework upon which the MAV's full autonomous capabilities will be built and deployed during IMAV 2025.

7 REFERENCES

- [1] M. Jaimez, J. G. Monroy, and J. Gonzalez-Jimenez, "Planar Odometry from a Radial Laser Scanner. A Range Flow-based Approach," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2016.
- [2] S. Macenski and I. Jambrecic, "SLAM Toolbox: SLAM for the Dynamic World," J. Open Source Softw., vol. 6, no. 61, p. 2783, May 2021, doi: 10.21105/joss.02783.
- [3] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The Marathon 2: A Navigation System," arXiv preprint arXiv:2003.00368, submitted Mar. 1, 2020, revised Jul. 14, 2020, accepted for IROS 2020.
- [4] A. Iyer, S. Narayan, N. M., and M. K. Rajagopal, "Autonomous Systems: Indoor Drone Navigation," arXiv preprint arXiv:2304.08893, Apr. 2023. Accepted to EAMT 2023.
- [5] A. Garcia, S. S. Mittal, E. Kiewra, and K. Ghose, "A convolutional neural network feature detection approach to autonomous quadrotor indoor navigation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 2019, pp. 74-81, doi: 10.1109/IROS40897.2019.8968222.
- [6] O. Alparslan and O. Cetin, "Fast and effective identification of window and door openings for UAVs' indoor navigation," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, 2021, pp. 175-181, doi: 10.1109/ICUAS51884.2021.9476840.
- [7] A. R. Thomas, "Safe landing zone detection for indoor UAVs using image segmentation models," M.S. thesis, Univ. of Cincinnati, Cincinnati, OH, USA, 2024.
- [8] M. Irfan, S. Dalai, K. Kishore, S. Singh, and S. A. Akbar, "Vision-based guidance and navigation for autonomous MAV in indoor environment," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020, pp. 1-5, doi: 10.1109/ICCCNT49239.2020.9225398.
- [9] F. Grondin and F. Michaud, "Lightweight and optimized sound source localization and tracking methods for open and closed microphone array configurations," arXiv preprint arXiv:1812.00115, submitted Dec. 4, 2018. [Sec. 2, 3,4]
- [10] F. Grondin, D. Létourneau, C. Godin, J.-S. Lauzon, J. Vincent, S. Michaud, S. Faucher, and F. Michaud, "ODAS: Open embeddeD Audition System," *Frontiers in Robotics and AI*, vol. 9, Art. no. 854444, May 2022, doi: 10.3389/frobt.2022.854444. [Sec. 2.1, 2.2]
- [11] MicoAir Technology. (s. f.). MicoAir 743 Flight Controller User Manual. MicoAir. Recuperado el 29 de agosto de 2025, de <https://micoair.com/docs/micoair743-user-manual/>
- [12] W. Xiaoyu, L. Caihong, S. Li, Z. Ning y F. Hao, "On Adaptive Monte Carlo Localization Algorithm for the Mobile Robot Based on ROS," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 5207-5212, doi: 10.23919/ChiCC.2018.8482698.