IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 June 2025, accepted 21 June 2025, date of publication 24 June 2025, date of current version 1 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3582653

RESEARCH ARTICLE

Design of a ROS2-Based Hybrid Aerial-Ground Robot for Autonomous Inspection Applications

MUTAZ RYALAT, (Senior Member, IEEE), GHAITH AL-REFAI, (Member, IEEE), NATHEER ALMTIREEN, (Member, IEEE), AND HISHAM ELMOAQET, (Senior Member, IEEE)
Mechatronics Engineering Department, German Jordanian University, Amman 11180, Jordan
Corresponding author: Mutaz Ryalat (mutaz.ryalat@gju.edu.jo)

ABSTRACT This study presents the design, modelling, and experimental validation of a lightweight hybrid aerial-ground robot developed for extended-duration inspection and monitoring tasks. The proposed system uniquely integrates a differential drive ground base with a quadrotor aerial module, thereby enabling adaptive mobility that facilitates effective navigation across constrained, cluttered, and complex environments. Built upon the Robot Operating System (ROS 2) framework, the robot architecture supports real-time control, modular software integration, and distributed task execution. A distinctive feature of the presented design is its energy-efficient operational strategy, emphasizing ground-based autonomous navigation and mapping for extended endurance while selectively utilising aerial capabilities for vertical access, obstacle circumvention, and challenging terrain traversal. Comprehensive experimental evaluations have been conducted to assess the robot's performance, confirming its capability for smooth transitions between ground and aerial modes while reliably maintaining stability, precise localisation, and robust obstacle avoidance. These results illustrate the significant potential of the developed robotic platform for autonomous deployment in applications demanding extended operation times and versatile mobility, including infrastructure inspection, environmental monitoring, and search-and-rescue missions in complex scenarios.

INDEX TERMS Autonomous systems, differential drive, hybrid robotics, long-endurance, mechatronics system design, multimodal locomotion, ROS2, SLAM.

I. INTRODUCTION

Multi-domain inspection in constrained environments demands robotic systems capable of adapting to dynamic challenges such as uneven terrain, narrow passages, and vertical obstacles [1]. Ground mobile robots, also called terrestrial mobile robots, provide extended operational durations, enhance adaptability to complex environmental conditions, and increase payload capacity. Nonetheless, their mobility is often restricted by uneven or obstructed terrain, which limits maneuverability and impairs navigational agility [2]. Conversely, aerial robots offer superior mobility and access to hard-to-reach areas but are limited by shorter operational times and payload capacities [3]. Hybrid ground-aerial robots address these limitations by combining the strengths of both locomotion modes, thereby enabling seamless transitions between environments [4]. However, existing designs frequently lack modularity, rely on proprietary hardware, or fail to integrate robust autonomy frameworks.

Autonomous hybrid ground/aerial robots represent a cutting-edge fusion of autonomous aerial vehicles (AAV) and autonomous ground vehicles (AGVs), designed to leverage the strengths of both modalities. These robots can operate in diverse environments, from urban settings to remote and hostile terrains, making them invaluable for various applications. The following subsections explore the types, technological advancements, applications, and design challenges of these systems.

A. HYBRID AERIAL-GROUND ROBOTICS

Hybrid aerial-ground robots, also called bi-modal, dual-mode, and multimodal robots, have attracted considerable attention owing to their ability to navigate complex

The associate editor coordinating the review of this manuscript and approving it for publication was Hassen Ouakad.

environments by leveraging both aerial and terrestrial mobility. Numerous studies have explored the integration of flight and ground movement capabilities, typically by mounting a quadrotor platform onto a ground mobility mechanism. Depending on the type of terrestrial mechanism employed, hybrid robots are commonly classified into four categories: track-, legged-, wheel-, and cage-based systems.

1) TRACK-BASED HYBRID ROBOTS

Tank-like or track-based robots are ground-based robotic platforms equipped with continuous tracks or treads, similar to those used in military tanks. These systems are designed to traverse the complex, debris-laden terrains typically encountered in disaster zones. Owing to their robust traction and high stability, tracked robots rely on mechanical force and ground contact to overcome obstacles such as rubble, uneven surfaces, and inclines. While effective in terms of mobility, these platforms often have large physical footprints and may pose risks of secondary damage to unstable structures during rescue operations [3], [5], [6]. Table 1 shows hybrid land-air robots that utilise tank-like or tracked ground mobility systems for traversing rugged terrain.

2) LEGGED-BASED HYBRID ROBOTS

Hybrid legged-aerial robots are robotic systems that integrate legged locomotion, which mimics animal-like walking or climbing, with aerial mobility (e.g., flight via rotors or wings) to navigate complex, unstructured environments. These robots combine the terrain adaptability of legged platforms with the aerial maneuverability of drones, enabling multimodal operation in scenarios such as search-and-rescue missions, subterranean exploration, or disaster response [8]. They offer high versatility in unstructured environments but often involve a trade-off between mechanical complexity and functional adaptability. Table 2 presents some examples of legged hybrid robots proposed in the existing literature.

3) WHEEL-BASED HYBRID ROBOTS

Wheel-based hybrid robots utilise conventional wheeled locomotion for energy-efficient ground travel combined with aerial capabilities such as hovering or vertical take-off and landing (VTOL). These platforms are mechanically simpler than legged systems, allowing for smoother control and longer operational endurance in structured and semi-structured environments. However, they are typically limited to rough or highly irregular terrains. Table 3 presents representative examples of hybrid robots in the wheeled category.

Wheel-based robots are favoured for their speed and efficiency on flat terrains and are relatively easy to control.

4) CAGE-BASED HYBRID ROBOTS

Cage-based hybrid robots are mobile platforms that integrate protective rolling cage structure with aerial mobility. The cage functions both as a protective shell and as a mechanism for ground locomotion, enabling the robot to roll across surfaces and withstand impacts from obstacles. This design improves the robustness and facilitates navigation in cluttered or hazardous environments.

However, the inclusion of a cage can increase the overall size and weight of the robot, which may negatively affect the flight efficiency and maneuverability. To be effective, cage-based hybrid robots should demonstrate strong collision resilience and debris tolerance, retain agile maneuverability, and balance payload capacity against the limitations of the onboard battery life. These diverse approaches demonstrate significant innovation in the design of hybrid aerial-ground systems. Each category brings distinct benefits: tracks for stability, legs for adaptability, wheels for efficiency, and cages for protection, enabling hybrid robots to operate across a broad range of environments and tasks.

B. TECHNOLOGICAL ADVANCEMENTS

Technological advancements in robotics can be approached from multiple perspectives, including innovations in mobility, autonomy, material science, and energy efficiency. A particularly dynamic area of progress lies in the development of hybrid robotic platforms that leverage capabilities across both the terrestrial and aerial domains.

1) HYBRID MOBILITY AND ENERGY EFFICIENCY

Hybrid robots are emerging as versatile mobile systems capable of navigating diverse environments by integrating ground-based and aerial locomotion. Terrestrial mobility modes, such as walking, rolling, and hopping, are effectively combined with aerial techniques such as gliding, hovering, and powered flight [2]. This multimodal adaptability enables a wide array of field applications, including infrastructure inspection, precision agriculture, disaster response, environmental surveillance, topographic mapping, and cinematography.

Current research in this domain has increasingly drawn inspiration from biological locomotion [18], seeking to emulate the efficiency, adaptability, and terrain-conforming abilities observed in nature. These biologically inspired systems not only expand our understanding of cross-domain mobility but also contribute to the optimisation of energy consumption, agility, and control under complex environmental conditions.

One of the most notable advancements in this field is the seamless integration of ground and aerial modes, enabling robots to adapt their mobility strategies in real time. A prime example is the Rollocopter [17], which combines a rolling ground mechanism with rotor-based aerial propulsion. This hybrid approach allows for power-efficient navigation at low speeds. Similarly, the DoubleBee robot utilises a bicopter configuration for flight and a two-wheel self-balancing system for terrestrial movements [12]. Its design demonstrated high energy efficiency in both

TABLE 1. Examples of tracked hybrid land-air robots.

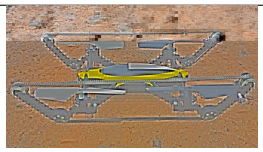
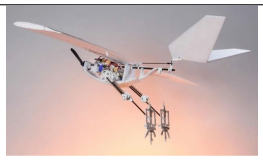
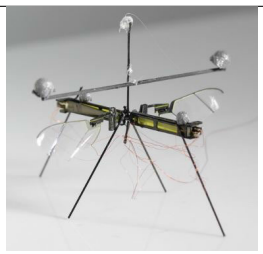
Name	Mobility	Design Features	Applications	Image
H.E.R.A.L.D.	Tracked ground + quadrotor flight	Tracked chassis, quadrotors for lift, compact design	Urban search and rescue, rubble traversal [5]	
B-Unstoppable	Tracked ground + aerial (VTOL)	Caterpillar tracks, quadrotor propulsion	Surveillance, rough terrain [3]	
HUUVER	Tracked ground + quadrotor flight	Compact tracked base with integrated quadrotors for aerial lift	designed for tight, cluttered environments Infrastructure inspection, confined area exploration [7]	

TABLE 2. Legged-based hybrid land-air robots.

Name	Mobility	Design Features	Applications	Image
LEONARDO	Bipedal (walking on a slackline, hopping, skateboarding) + Hovering	multi-joint legs and propeller-based thrusters + ducted fans for balance and lift	Terrain traversal, space exploration [9]	
RAVEN	Walking, hopping, jumping into flight	Avian-inspired legs with springs and motors; efficient ground-to-air transitions	Inspection across diverse environments [10]	
RoboFly	Flight + Ground + Water Surface	Flapping-wing robot, ultra-light, minimal design	Environmental monitoring in tight or aquatic areas [11]	

locomotion modes, providing an extended operational range and improved maneuverability. Innovations in propulsion and structural efficiency are also evident in platforms such as the ground-aerial dual actuator monocopter (G-ADAM) [19], which employs a single motor to drive both flight and ground movement. This unified actuation system not only reduces the overall mass but also simplifies the mechanical complexity and enhances the responsiveness during mode transitions.

Collectively, these advancements have significantly increased the endurance, adaptability, and mission versatility of hybrid robots, rendering them suitable for deployment in long-duration missions and extreme or inaccessible terrain [3].

2) ADVANCED CONTROL AND NAVIGATION SYSTEMS

Hybrid ground-aerial robots combine terrestrial and aerial locomotion, enabling versatile navigation across complex environments, such as urban disaster zones, confined infrastructure, collapsed buildings, subterranean tunnels, dense urban areas, and cluttered industrial sites. These systems require intelligent control architectures capable of managing transitions between vastly different mobility modes, while accounting for dynamic constraints, energy efficiency, and environmental uncertainty. Therefore, advanced control and navigation strategies are essential for optimising performance, ensuring safety, and achieving high levels of autonomy.

TABLE 3. Examples of wheel-based hybrid land-air robots.

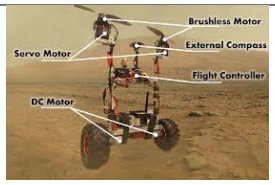
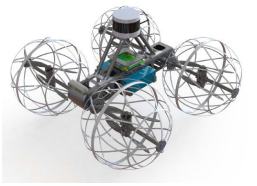
Robot Name	Mobility Modes	Design Features	Applications	Image
DoubleBee	Two-wheel ground + bicopter aerial mode	Self-balancing wheels and dual propellers with decoupled control scheme	Energy-efficient ground flight transition, Challenging indoor and outdoor missions [12]	
Rollocopter	Rolling on terrain + quadrotor flight	a quadrotor system to fly or roll along on two passive wheels	Subterranean navigation, caves [13]	
Skater	Passive wheels + longitudinal bicopter flight	Lightweight bi-copter with two passive side wheels; unified actuation for both modes	Adaptive locomotion on slippery and rough terrains; efficient trajectory tracking [14]	

TABLE 4. Cage-based hybrid land-air robots.

Robot	Mobility Modes	Design Features	Applications	Image
HyTAQ	Rolling cage + quadrotor flight	Quadrotors integrated in a rolling cage structure	Urban and indoor navigation [15]	
Drivocopter	Spherical rolling + flight	a spherical cage that allows rolling in any direction; integrated with quadrotors for aerial mobility	Surveillance, tight space navigation [16]	
Rollocopter	Rolling cage + quadrotor flight	Multi-rotor propellers enclosed in a spherical shell to produce the necessary forces to roll on the ground and fly.	Subterranean and disaster response [17]	

Recent developments have introduced innovative solutions tailored to the unique challenges faced by hybrid platforms. One such system is HE-Nav, a high-performance navigation framework specifically designed for aerial-ground hybrid paths in cluttered spaces [20]. HE-Nav integrates terrain-aware planning with aerial dynamics to facilitate autonomous

mode switching based on environmental demands. This system exemplifies a modular design philosophy that emphasises adaptability to complex inspection and exploration tasks.

Recent developments in advanced control strategies, such as Model Predictive Control (MPC) and Sliding Mode Control (SMC), have demonstrated promising results in robust path tracking and constrained dynamic systems. For example, MPC has been effectively employed for green dynamic positioning under input saturation constraints in marine vessels [21], while SMC has shown strong resilience to disturbances in autonomous off-road vehicle tracking [22]. Although this work does not implement such advanced controllers directly, the modular ROS 2 architecture enables their future integration, especially for aerial motion planning and constrained navigation tasks in inspection scenarios.

To support obstacle avoidance and autonomous maneuverability, hybrid robots often employ dynamic modelling and integrated motion-control schemes. For example, Premachandra et al. [23] demonstrated a dual-mode control strategy that coordinates both ground and aerial movement through unified motion controllers, supported by LiDAR and vision-based sensing for real-time obstacle detection.

Another example is CapsuleBot, a bi-copter platform equipped with actuated wheel-rotors, that utilizes a hybrid cascaded control approach. This design achieves versatile locomotion with minimal actuators, reducing mechanical complexity while maintaining high functionality [24].

Data-driven navigation systems utilising AI and ML have emerged as powerful tools for robotic operations in feature-sparse and GPS-denied environments. Misra et al. [25] introduced a novel machine learning-enhanced navigation system for autonomous vehicles operating in tunnels and underground settings. The system constructs real-time maps using multisensor fusion and predictive analytics, enabling robust path planning and significantly improved Simultaneous Localisation and Mapping (SLAM) performance even in the absence of external positioning signals.

Overall, these advanced control frameworks commonly integrate multiple sensing modalities, including IMUs, LiDAR, and vision-based SLAM with intelligent algorithms to facilitate robust, adaptive, and fully autonomous operation in diverse environments.

3) COLLABORATIVE SYSTEMS AND MULTI-AGENT FRAMEWORKS

Recent advancements in hybrid robotics have emphasised the development of collaborative systems that integrate heterogeneous robotic agents, including aerial and ground vehicles, into cohesive operational units. These systems exploit the complementary capabilities of different platforms to enhance mission effectiveness in dynamic environments.

A representative example is the *Flourish* project, which demonstrates UAV-UGV collaboration in precision agriculture. In this system, UAVs conduct aerial surveys to generate high-resolution field maps, whereas UGVs perform targeted interventions such as selective spraying based on UAV derived data [26]. This collaborative workflow enables efficient monitoring and implementation in large agricultural fields.

The emergence of scalable multi-agent frameworks has further enhanced the capabilities of hybrid robotic teams. For instance, a robust multi-agent architecture proposed by Mondal et al. supports UAV-UGV collaboration in disaster management scenarios, utilising asynchronous decentralised planning to optimise path planning, resource allocation, and role assignment [27]. The framework ensures effective coordination even under communication constraints and varying team sizes, making it suitable for time-critical missions in hazardous environments.

These developments in collaborative robotics and multi-agent systems open new frontiers for hybrid robot deployment in agriculture, disaster response, and smart infrastructure inspection.

C. INTEGRATION OF ROS IN MULTIMODAL GROUND-AERIAL ROBOTICS

The Robot Operating System (ROS) serves as a critical enabler for software integration and control in hybrid ground-aerial robots that combine terrestrial locomotion (e.g., wheeled, tracked, or legged mobility) with aerial flight capabilities. These systems require robust middleware to synchronise heterogeneous subsystems, such as ground propulsion actuators, multirotor flight controllers, sensor fusion pipelines, and mission planners, while managing mode transitions, resource constraints, and environmental uncertainties [28]. The modular architecture of ROS simplifies the development of distributed control frameworks, enabling seamless communication between terrestrial and aerial modules through standardised topics, services, and actions [29].

The core strength of the ROS lies in its simulation-to-deployment ecosystem. Platforms such as *Gazebo* and *RViz* allow developers to model hybrid robots (e.g., drones with retractable wheels or tracked ground modules) and simulate multimodal navigation in complex environments (e.g., urban debris fields and uneven terrain) [30].

ROS 2 further advanced hybrid robotics by addressing real-time constraints and scalability. In addition to facilitating hardware abstraction and sensor integration, ROS also supports high-level capabilities such as SLAM, path planning, and multimodal control via open-source packages such as `navigation stack` [31]. These tools have significantly accelerated the development and testing of researchers in building hybrid systems. The use of ROS 2 is also gaining momentum in hybrid systems owing to its enhanced real-time communication, DDS-based architecture, and improved scalability, making it better suited for industrial and multi-robot applications [32].

The key challenges include computational resource allocation and interoperability. Lightweight ROS 2 variants

like *Micro-ROS* are increasingly deployed on embedded flight controllers to manage low-level motor commands, while ground modules use full ROS stacks for high-level perception and planning [33]. Open-source packages such as *ros_control* standardise hardware interfaces for hybrid platforms enable plug-and-play integration of diverse actuators (e.g., wheels and rotors). Emerging trends in hybrid robotics emphasise the integration of AI-enhanced ROS workflows. Chen et al. [34] introduced the *ROS-X-Habitat* (''ROS-Cross-Habitat''), which bridges the AI Habitat simulation platform with the ROS ecosystem, enabling realistic, photo-simulated training environments for embodied AI agents.

On the hardware deployment side, NVIDIA's Jetson AI Lab has contributed to this trend by supporting edge-AI integration with ROS 2. The *ros2_nanollm* package [35] provides ROS 2 nodes for running optimised large language models (LLMs) and vision-language models (VLMs) locally within containerised environments. Built on top of NanoLLM and ROS 2 Humble, this package enables the deployment of generative AI models directly onboard Jetson-powered robots, facilitating real-time on-device inference for advanced perception and interaction capabilities.

Overall, ROS plays a critical role in enabling robust, modular, and scalable software architectures for hybrid ground-aerial robots operating in dynamic and uncertain environments.

D. MOTIVATION AND CONTRIBUTIONS

Hybrid robots have shown promise in navigating constrained or uneven terrains. However, many existing platforms lack modular software design, open-source compatibility, and cost-effective manufacturing approaches.

This study addresses these gaps by deploying a practical, low-cost hybrid robot, leveraging ROS 2, Commercial Off-the-Shelf (COTS) hardware, and a custom 3D-printed chassis.

Key contributions

The main contributions of this paper are summarized as follows:

- **Modularity and ROS 2 Integration**: the proposed system leverages a ROS 2-centric architecture, facilitating rapid prototyping and interoperability with SLAM, planning, and navigation stacks, enabling real-time communication and distributed control—unlike prior systems built on custom firmware or legacy ROS 1.
- **Low-Cost and Open Architecture**: The robot is built from low-cost, off-the-shelf components and relies entirely on open-source tools (PX4, MAVROS, ROS 2), enabling reproducibility and future scalability (e.g., for VIO and full 3D SLAM).
- **Autonomous Mode Switching**: Unlike semi-automatic or mechanically triggered transitions in existing platforms, the proposed system employs LiDAR-based automatic ground-to-air mode switching, enhancing both autonomy and mission continuity.
- **Quantified Energy Profiling**: We provide a detailed power and energy analysis across operational modes and task phases (idle, mapping, navigation, hover), which is often overlooked or qualitatively described in related work. This power analysis is provided to support mission planning, showing the feasibility of extended hybrid operation using an onboard LiPo battery.
- **Validation**: The system is validated in indoor inspection scenarios, demonstrating SLAM-enabled navigation using LiDAR and odometry for robust terrestrial mobility, reliable mode transitions for inspection tasks.

Design challenges addressed:

- *Mode Transition:* Seamless aerial-ground switching requires careful weight distribution and actuator design.
- *Energy Efficiency:* Lightweight frames, efficient propulsion, and control algorithms help reduce the power overhead.
- *Navigation Robustness:* Operating in GPS-denied, cluttered environments requires advanced local planning and sensor fusion.

E. PAPER STRUCTURE

The remainder of this paper is structured as follows: Section II describes the mechanical and electronic integration in the hybrid platform under the title *Mechatronics System Design*. Section III presents the *Software Architecture and Control Framework*, detailing the ROS 2-based modular implementation. Section IV describes the *Modelling and Control* of both the ground and aerial subsystems. Section V outlines the *Hybrid Robot Programming* approach, including locomotion mode switching and mission logic. Section VI provides an *Experimental Evaluation and Results*, demonstrating the performance of the system in inspection tasks. Finally, Section VII concludes the study and suggests directions for future work.

II. MECHATRONICS SYSTEM DESIGN

This section presents the mechanical design, hardware selection, and system integration of the proposed hybrid land-air robot. The platform was engineered to support both differential drive terrestrial mobility and quadrotor-based aerial movement.

A. MECHANICAL DESIGN AND FRAME CONFIGURATION

The hybrid robot comprises a custom 3D-printed chassis that integrates ground and aerial propulsion units within a modular structure. The aerial section was built on the Holybro S500 V2 frame, chosen for its stability, lightweight carbon-reinforced plastic arms, and compatibility with 12-inch propellers and D3548 1100KV brushless DC motors.

The ground mobility system was based on a differential drive layout using two GJA25-370 130 RPM DC geared motors with encoders. To reduce aerodynamic drag and weight, oval-shaped acrylic plates were laser-cut to form the

body compartments of the robot, resulting in an estimated 20% weight reduction compared to rectangular frames.

The weight distribution was optimised using CAD software (SolidWorks), with the centre of mass aligned with the geometric centre. The heaviest component, the 4S 6000mAh LiPo battery, was mounted centrally to enhance the stability during flight and minimise wiring complexity. Figure 1 shows the proposed CAD model of the hybrid robot, and Figure 2 shows its exploded view, detailing its main components.

B. MATERIALS AND FABRICATION

The structural components are fabricated using two key materials:

- **3D Printed PCE Plastic:** Used for custom mounting brackets, sensor housings, and structural reinforcement.
- **Laser-Cut Acrylic:** Flat plates for compartmental divisions and support layers.

The modular approach enables rapid prototyping and reconfiguration, while also ensuring strength-to-weight efficiency, which is critical for flight stability.

C. ELECTRONIC COMPONENTS

The system integrates the following key electronics:

- **Microcontroller Units:** Raspberry Pi 4 (8GB) for high-level processing and ROS2 integration; ESP32 WROOM for motor control and real-time encoder feedback.
- **Flight Controller:** SpeedyBee F405, configured via Betaflight for manual aerial control. The ESCs (Skywalker 80A) were externally connected to high-current brushless motors.
- **Sensors:** RPLIDAR A1 for 360-degree 2D scanning; onboard IMU for aerial stabilisation; Raspberry Pi Camera Module for visual inspection.
- **Power Supply:** A 4S 6000mAh LiPo battery powered the entire robot. Voltage regulation is achieved via 3A and 12A buck converters.

D. WIRING AND INTEGRATION STRATEGY

Figure 3 illustrates a schematic diagram of the main electronic components used in the system, while Table 5 provides a comprehensive list of all components, including their descriptions and reference numbers corresponding to the schematic.

A centralised power distribution board routes power from the battery to ESCs, buck converters, and microcontrollers. The Raspberry Pi communicates with the ESP32 over UART for encoder data exchange and motor commands. LiDAR data are processed directly on the Raspberry Pi via USB, whereas flight signals are relayed through the SpeedyBee controller and FlySky iBus receiver.

Signal and power lines were routed to reduce electromagnetic interference and maintain mechanical separation between the land and air systems. Care was taken to ensure the thermal dissipation and vibration isolation of the sensitive components.

TABLE 5. List of electronic components.

Ref. No.	Component Description
1	D3548 1100 KV Brushless DC Motor
2	Hobbywing Skywalker V2 80A ESC
3	SpeedyBee F405 Flight Controller
4	Multi-Axis Copter Power Battery ESC Connection Board
5	RPLiDAR A1 Sensor
6	LiPo 4S 6000mAh Battery
7	3A Buck Converter
8	12A Buck Converter
9	Raspberry Pi 4 Model B (8GB RAM)
10	PC (Host Computer)
11	ESP32 WROOM Microcontroller
12	Raspberry Pi Camera Module
13	L298N H-Bridge Motor Driver
14	GJA25-370 130 RPM Geared DC Motor with Encoder

E. CONTROL MODE CONFIGURATION

The robot operates in two switchable modes:

1) **Ground Mode:** Fully autonomous using ROS 2. SLAM, path planning, and obstacle avoidance were handled by Raspberry Pi with LiDAR and encoder fusion.
2) **Flight Mode:** 1- Manually controlled via an RC transmitter. 2- In this study, the Pixhawk 4 flight controller was selected to manage the aerial locomotion module owing to its high reliability, modularity, and widespread support in the open-source robotics community. The controller is compatible with both the PX4 and ArduPilot firmware, and it supports real-time sensor processing, attitude stabilisation, and autonomous mission execution, making it ideal for integration into hybrid platforms. The ROS 2–PX4 architecture enables deep coupling between the flight controller and onboard ROS 2 stack, allowing ROS 2 nodes to directly interface with PX4's internal *uORB* messaging system. Through this integration, ROS 2 publishers and subscribers can access flight state information and send control commands in real time, facilitating tight coordination between the ground and aerial modes [36].

Each mode has a distinct hardware control loop but shares core processing units and power systems. This dual-mode setup allows for extended ground operation with a quick deployment of aerial capabilities when required.

Figure 4 illustrates the hardware configuration of the proposed hybrid robot, which integrates both ground and aerial mobility components. The system combines a differential drive wheel base for terrestrial navigation with a quadrotor module for aerial deployment. Key hardware elements such as motor drivers, onboard processors, power distribution, and sensor placements are strategically arranged to ensure modularity, balance, and efficient mode switching. This configuration enables the robot to perform complex inspection tasks in cluttered and unstructured environments.

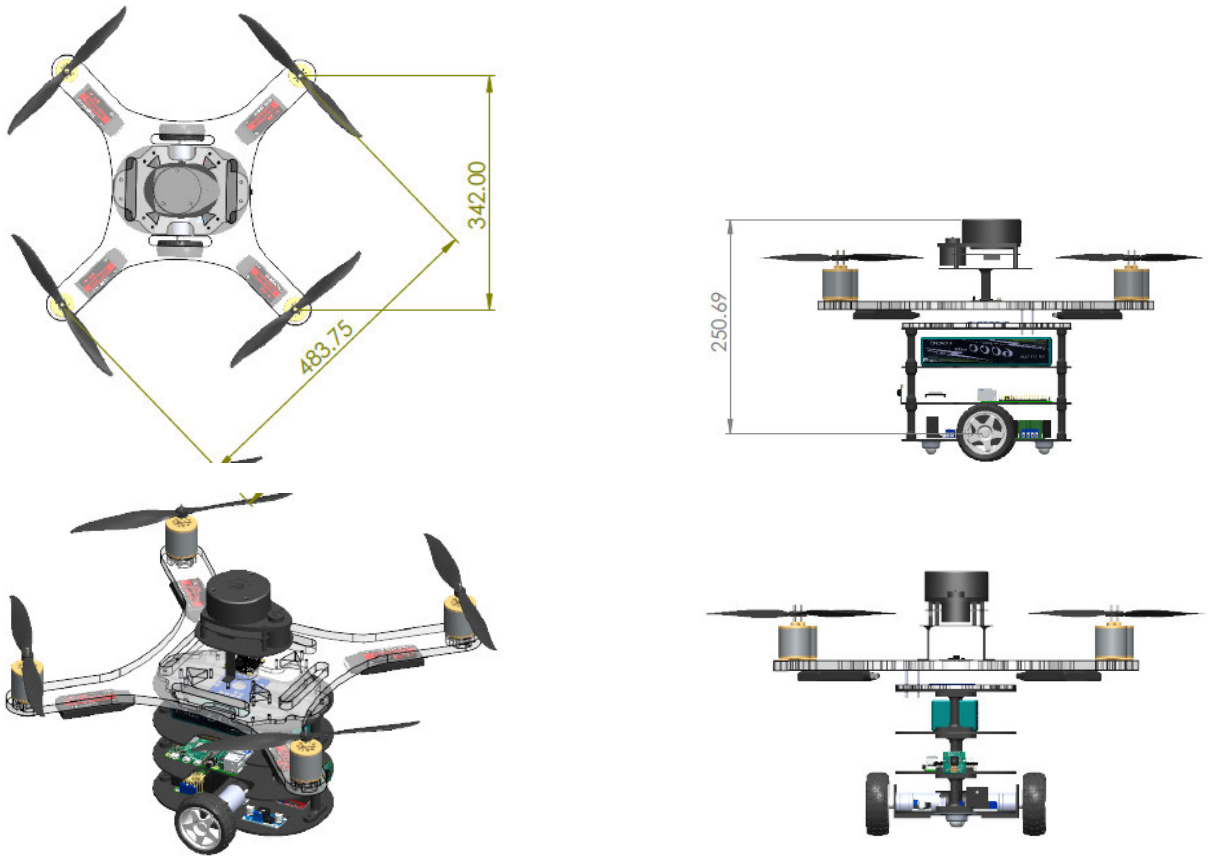


FIGURE 1. Cad Design with different views.

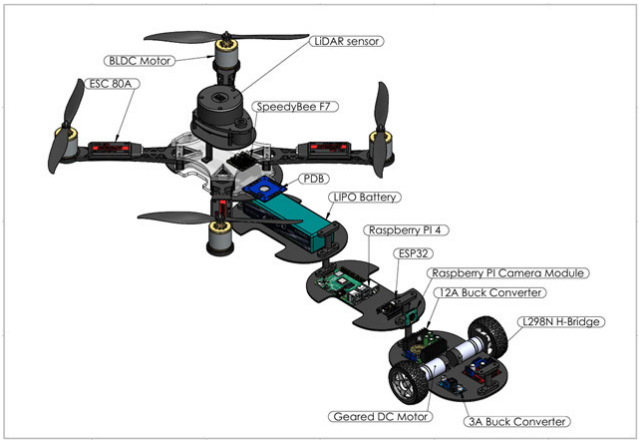


FIGURE 2. Exploded view of the robot.

III. SOFTWARE ARCHITECTURE AND CONTROL FRAMEWORK

The software architecture is designed around ROS 2 Humble distribution with Linux Ubuntu as the operating system (OS), selected for its real-time capabilities, modular communication model, and support for distributed robotics. This section outlines the core subsystems for ground and aerial operations, sensor integration, and navigation logic.

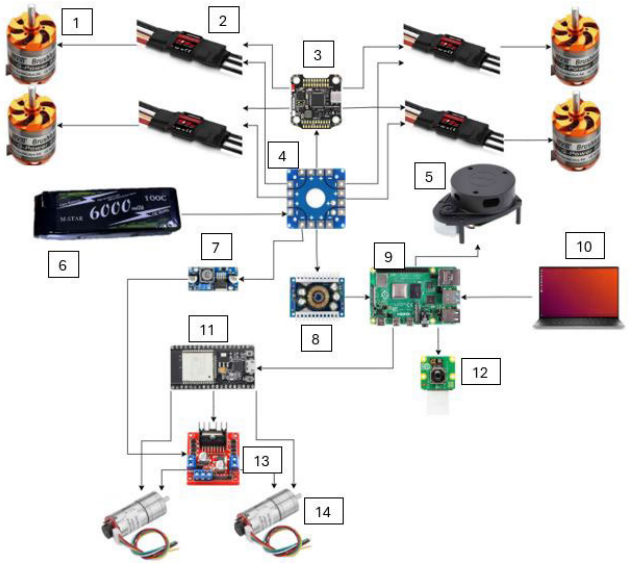


FIGURE 3. Electrical/electronic architecture of the robot.

A. ROS 2-BASED CONTROL PIPELINE

ROS is an open-source middleware framework that simplifies the development of robotic systems by providing a comprehensive suite of software libraries and tools that run

FIGURE 4. The Hybrid robot hardware configuration.

over conventional operating systems. Rather than acting as a full-fledged operating system, ROS serves as a flexible infrastructure for designing complex robotic behaviours. It enables developers to modularise robotic functionalities into distributed processes called nodes, each responsible for a specific task, such as sensor data acquisition, actuator control, or autonomous decision-making [37].

One of the key advancements introduced in ROS 2 is its native support for multimachine configurations, which enables the seamless distribution of nodes across multiple networked devices. This feature is particularly advantageous in complex robotic systems that require diverse computational resources or interface with various hardware platforms. By leveraging the Data Distribution Service (DDS) middleware, ROS 2 facilitates efficient and reliable communication between nodes regardless of their physical deployment. DDS ensures low-latency fault-tolerant messaging, making it ideal for distributed robotic architectures [38].

This capability is particularly relevant to our hybrid land-air robotic platform, where different subsystems, such as flight control, ground mobility, and sensor processing, are deployed on distinct machines. ROS 2 enables these heterogeneous components to operate in coordination while remaining loosely coupled, therby ensuring modularity and scalability.

We chose the ROS 2 Humble Hawksbill as our framework because of its substantial improvements over earlier versions. Notable enhancements include refined real-time performance, advanced security mechanisms, and optimised communication layers. These features are critical for managing the dynamic requirements of the robot, which must seamlessly orchestrate both terrestrial and aerial functionalities. Thus, the architecture of ROS 2 thus provides the robustness and flexibility necessary for integrating these subsystems into a cohesive structure.

The autonomous flight control system is built on the PX4-ROS 2 architecture, with computational tasks distributed between a Raspberry Pi 4 and an ESP32 microcontroller to optimise the performance across hybrid modes. ROS 2 nodes are deployed in a distributed configuration, leveraging the publisher-subscriber model for real-time communication among the subsystems:

- **High-Level Processing:** Raspberry Pi 4 executes high-level tasks such as LiDAR-based scan processing, SLAM, global path planning, and camera feed management. These operations are tightly integrated with the ROS 2 graph and interface with the flight controller via MAVROS or PX4-ROS 2 bridge.
- **Low-Level Control:** The ESP32 handles low-level actuator control and sensor feedback using *Micro-ROS*. It is responsible for motor command execution, encoder data acquisition, and communication with the main ROS 2 system over a serial transport layer.
- **Manual Aerial Control:** For manual overriding in aerial mode, a SpeedyBee F405 flight controller running Betaflight firmware was used. It operates independently of the ROS network and is controlled by an RC transmitter (FlySky) using the iBus protocol for direct input.

Figure 5 shows the complete operational flow of the hybrid ground-aerial robot, with a ROS 2-based architecture serving as the backbone for the control, perception, and decision-making processes. Upon system initialisation, ROS 2 nodes were launched to manage sensor data acquisition, localisation, and control, with LiDAR, IMU, and camera modules activated concurrently.

The middleware's distributed design allows independent ROS 2 nodes such as SLAM, navigation, and control to exchange data in real time through a publish-subscribe mechanism. After initialisation, the robot evaluates the mission requirements and environmental conditions to determine whether to operate in the ground or aerial mode. This decision is handled by a switching logic node that assesses terrain features, obstacle density, and target location. In the ground mode, the robot engages a differential drive controller using the SLAM Toolbox to process odometry and LiDAR inputs for mapping and navigation while executing dynamic obstacle avoidance. When ground-based traversal becomes infeasible, such as when encountering staircases, cluttered debris, or unreachable vertical inspection points, the system triggers a transition to the aerial mode. In the aerial mode, the robot supports both manual and autonomous flight; manual control allows for operator-guided inspection, whereas autonomous operation leverages onboard perception for obstacle detection and waypoint-based navigation. Both operational paths culminate in a return-to-ground sequence after which the mission is concluded. The modular and real-time architecture of ROS 2 enables seamless transitions between locomotion modes, thereby supporting hybrid mobility with minimal latency and robust task continuity. This hybrid architecture ensures both autonomous functionality and manual override capability, while maintaining modularity, fault isolation, and computational efficiency across resource-constrained onboard hardware.

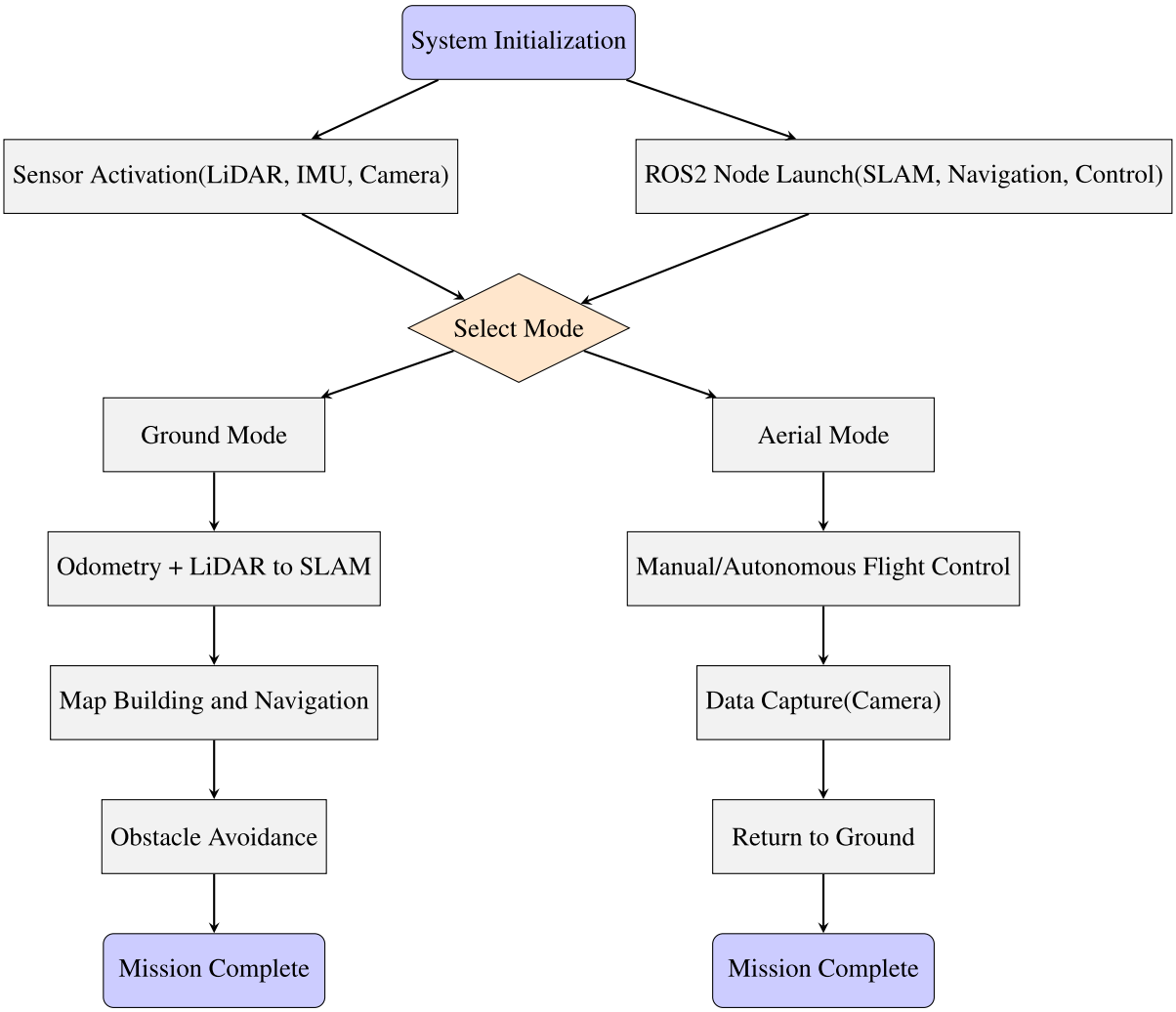


FIGURE 5. Operational flowchart of the hybrid ground-aerial robot using ROS2.

B. VISUALIZATION AND DEBUGGING WITH RVIZ

RViz (ROS Visualisation) is a core tool in the ROS 2 ecosystem and is designed to provide an intuitive graphical interface for visualising real-time data from a robotic system. It plays a vital role in the development, testing, and debugging of robotic applications by allowing users to observe how a robot perceives and interacts with its environment [39].

Here, RViz is utilised extensively to monitor sensor inputs, visualise robot state information, and validate navigation and perception modules during runtime. The tool supports a wide range of ROS message types and integrates seamlessly with standard topics, enabling the display of complex data, such as point clouds, occupancy grids, and transform frames. Key features of RViz include:

- **3D Visualization:** RViz provides a dynamic 3D environment to visualize sensor data such as LiDAR scans, odometry, and camera feeds. This aids in assessing the robot's spatial awareness and sensor accuracy in real time.
- **Interactive Controls:** The interface allows users to interact with the robot by setting navigation goals, manipulating frames, or sending commands directly through the GUI. This feature facilitates efficient testing of the planning and control algorithms.
- **Customizable Displays:** RViz offers a highly configurable layout, enabling developers to toggle various visualization elements on or off according to the task at hand. This ensures a focused and clutter-free view of the relevant data streams.

RViz has proven to be an indispensable tool in our development workflow, particularly for validating sensor fusion outputs, debugging navigation behaviour and visualizing multinode communication in a distributed system setup. Figure 6 shows the robot model spawned in RViz, with the transform (TF) frames of key components, such as the base link, wheels, and individual rotors, clearly visualized. Each frame is represented by coloured coordinate axes (red: x, green: y, blue: z), confirming the proper linkage within the

TABLE 6. List of parameters and abbreviations.

Parameter	Abbreviation	Unit
Total weight	W_{total}	N
Mass	M	g
Force	F	N
Torque	T	N·m
Total thrust	TT	g
Current encoder count	$C_{current}$	-
Last encoder count	C_{last}	-
Encoder counts per revolution	C_{rev}	-
Wheel radius	r	m
Wheel diameter	d	m
Wheel separation	l	m
Wheel displacement	D	m
Wheel velocity	ν_w	m/s
Linear velocity	ν	m/s
Angular velocity	ω	rad/s
New orientation angle	θ_{new}	deg
Old orientation angle	θ_{old}	deg
Orientation angle difference	$\Delta\theta$	deg
Time interval	Δt	ms
New X position	X_{new}	m
Old X position	X_{old}	m
New Y position	Y_{new}	m
Old Y position	Y_{old}	m
Voltage	V_o	V
Current	I	mA
Battery Capacity	BC	Wh
Total Power Consumption	TPC	W
Time	t	hr/min

robot's URDF and the integrity of the TF tree, which is essential for motion planning, state estimation, and sensor coordination in the ROS 2 environment.

IV. MODELLING AND CONTROL

The modelling and control of a hybrid land-air robotic system require careful consideration of both mechanical and electrical subsystems. This section presents the key parameters, mathematical models, and control calculations that underpin the robot's performance in both aerial and terrestrial modes.

To ensure accurate and efficient operation, several aspects were aanalysed, including the total weight distribution, thrust generation, torque output, power consumption, and kinematic equations for position tracking. The robot's components were selected not only for functionality but also with a strong emphasis on weight and energy efficiency. The control architecture is supported by sensor feedback and encoder data, LiDAR and cameras, which allow the system to estimate the position and orientation in real time.

Through a combination of physical modelling and empirical data, the robot's mechanical properties and control logic were optimised to deliver stable performance across multiple terrains and during aerial operation. The following subsections detail the critical calculations and parameter definitions that guided the development of the system.

Table 6 lists all parameters and their respective abbreviations used throughout this section.

A. WEIGHT AND WEIGHT DISTRIBUTION CALCULATIONS

During the design phase, minimising the robot's total weight was a critical consideration, particularly because of the additional constraints of the aerial functionality. After extensive iteration and component optimisation, the minimum total weight was determined to be **3370 g**, as shown in Table 7. Figure 7 shows the weight breakdown of the robot for the main subsystems.

To ensure both dynamic stability and energy efficiency, a battery was strategically placed at the centre of the robot mass (CoM). Using SolidWorks mass property analysis, the calculated CoM coordinates were (in mm):

$$X = 63.80, Y = 57.73, Z = 136.83$$

This central placement significantly improves the stability by minimising the moment of inertia, which is an especially important factor during flight. It also enhances traction and manoeuvrability during ground operation. The principal moments of inertia are as follows:

- $P_{xx} = 39{,}310{,}064.42$ g·mm^2
- $P_{yy} = 41{,}297{,}010.90$ g·mm^2
- $P_{zz} = 53{,}002{,}106.85$ g·mm^2

This optimised mass distribution reduces oscillations and control effort during both aerial and terrestrial modes, leading to a better dynamic response and power efficiency.

B. THRUST CALCULATIONS

One of the most critical design considerations for the aerial functionality of hybrid robots is the selection of motors capable of providing adequate thrust. To ensure a stable vertical takeoff and landing (VTOL), the total available thrust must exceed the gravitational force acting on the robot [40]. This condition can be expressed as

$$\frac{a}{g} > 1 \tag{1}$$

In other words, the *total thrust-to-weight ratio* must be greater than one to achieve upward acceleration. Overestimating this value may lead to unnecessary power consumption, increased battery weight, and the need for larger or heavier components, negatively impacting the overall design. Therefore, a balanced approach is adopted to ensure both safety and efficiency.

In this system, a safety margin was applied by targeting a total thrust value twice the total weight of the robot [40]

$$\begin{aligned} W_{total} &= 3370 \text{ g} \\ T_T &= 2 \times W_{total} = 2 \times 3370 = 6740 \text{ g} \end{aligned} \tag{2}$$

Since the quadcopter uses four motors, the required thrust per motor is

$$T_{motor} = \frac{T_T}{4} == 1.685 \text{ kg} \tag{3}$$

Thus, each selected motor must be capable of generating at least **1.685 kg** thrust. To accommodate minor variations

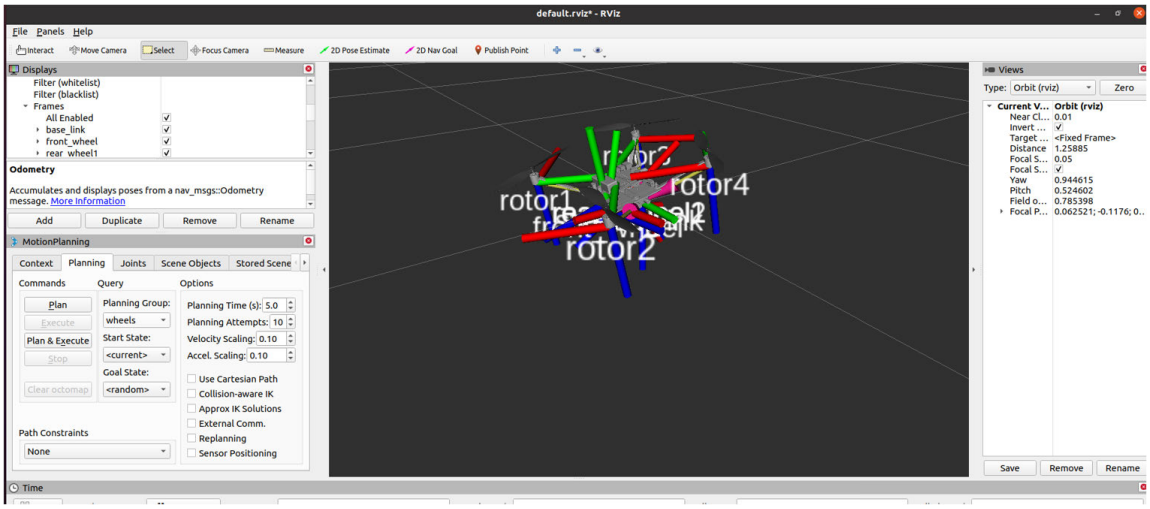


FIGURE 6. Drone transform tree visualisation using RVIZ.

TABLE 7. Power and weight specifications of system components.

Component	Qty	Total Mass (g)	Voltage (V)	Current (A)	Power (W)
Raspberry Pi 4 (8GB)	1	100	5	3	15
ESP32 WROOM	1	10	–	–	–
SpeedyBee F405	1	10.7	11.1–18.5	1	18.5
GJA25-370 130RPM DC Motor	2	186	12	0.48	5.76
D3548 Brushless DC Motor	4	624	11.1–18.5	246	3640
Skywalker 80A ESC	4	316	5	7	35
L298N H-Bridge	1	26	5	1.4	7
4S 6000mAh LiPo Battery	1	614	14.8*	6*	8880*
Power Distribution Board	1	8	–	–	–
RPLiDAR A1 Sensor	1	190	–	–	–
13A Buck Converter	1	70	–	–	–
3A Buck Converter	1	10	–	–	–
Raspberry Pi Camera Module	1	3	–	–	–
Propellers	4	40	–	–	–
Body and Other Components	1	1162.3	–	–	–
Sub-total	–	**3370**	–	**259.33**	**3723.51**

* Values based on battery specifications. The power consumption of the ESP32 and RPLiDAR is included in the drawing of the Raspberry Pi.

and ensure operational safety, motors with a thrust rating of $\geq$ **1.7 kg** were chosen. This calculation confirms that the motor thrust is sufficient to lift the robot during vertical flight while maintaining energy efficiency.

Selecting an appropriate propulsion system for aerial operation is critical in the design of a hybrid land-air robot. A key parameter in this process is the thrust required, which must exceed the total weight of the robot to enable a stable vertical lift [40]. It is essential to maintain a safe thrust-to-weight ratio while avoiding overestimation, which may lead to unnecessary power consumption and an increased component mass.

To ensure safe and efficient flight, a thrust margin of approximately 2:1 is typically adopted. Therefore, the total required thrust TT is calculated as follows:

$$TT = 2 \times W_{\text{total}} = 6740 \text{ g} \tag{4}$$

Assuming a quadcopter configuration with four brushless DC motors, the thrust required from each motor is

$$T_{\text{motor}} = \frac{TT}{4} = 1685 \text{ g} \tag{5}$$

Thus, each motor must be capable of delivering at least **1.685 kg** thrust. In practice, a minimum thrust rating of **1.7 kg**

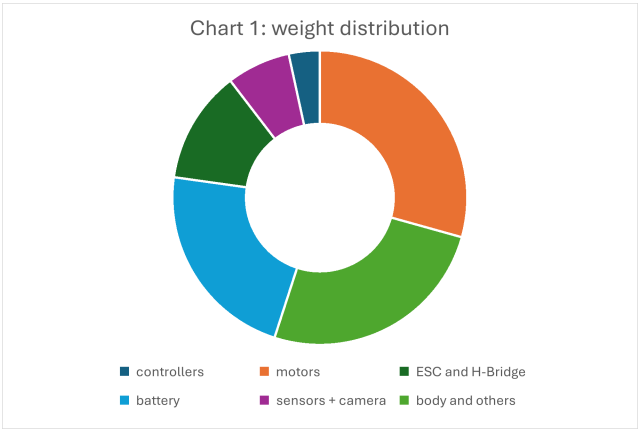


FIGURE 7. Weight distribution chart of the hybrid robot.

per motor was considered to ensure sufficient lift, even under minor variations in the component weight or environmental conditions. This value serves as a guiding specification for motor selection to achieve a stable and energy-efficient aerial performance.

C. UGV TORQUE AND LINEAR SPEED CALCULATIONS

To evaluate the robot's ground mobility, the torque and linear speed of the Autonomous Ground Vehicle (AGV) mode were calculated. These calculations consider the wheel geometry, motor RPM, mass of the robot, and rolling resistance due to friction.

1) ANGULAR VELOCITY

The angular velocity ω of the wheel is calculated from the rated rotational speed:

$$\omega = \frac{\text{RPM} \times 2\pi}{60} = \frac{130 \times 2\pi}{60} = 13.61 \text{ rad/s} \tag{6}$$

2) LINEAR VELOCITY

The linear velocity v of the robot is then obtained using the relation:

$$v = r \times \omega = 0.0325 \times 13.61 = 0.442 \text{ m/s} \tag{7}$$

3) WEIGHT AND FRICTION FORCE

The total weight W_{total} of the robot in Newtons is:

$$W_{total} = M \times g = 3.37 \text{ kg} \times 9.81 \text{ m/s}^2 = 33.05 \text{ N} \tag{8}$$

Assuming a coefficient of friction $\mu = 0.5$, the force F required to overcome the rolling resistance is

$$F = \mu \times W_{total} = 0.5 \times 33.05 = 16.525 \text{ N} \tag{9}$$

4) TORQUE CALCULATION

Finally, the required torque T at the wheel is:

$$T = F \times r = 16.525 \times 0.0325 = 0.537 \text{ Nm} \tag{10}$$

This torque value is used to select suitable geared motors that can provide an adequate driving force for ground movement without stalling under load.

D. DRIVE MODE KINEMATICS [41]

In land mode, the robot operates as a differential drive vehicle. To estimate its position, orientation, and trajectory on a 2D map, we applied forward kinematics based on the wheel encoder data. This involves computing individual wheel displacements and linear and angular velocities and using these to update the robot's pose over time.

To measure the distance travelled and orientation changes, encoder feedback from both the left and right wheels was used. The following equations govern the kinematic model:

1) WHEEL DISPLACEMENT

The displacement D of a wheel is calculated from the encoder count difference:

$$D = \left(\frac{C_{\text{current}} - C_{\text{last}}}{C_{\text{rev}}} \right) \times (2\pi r) \tag{11}$$

where:

- C_{current} is the current encoder count,
- C_{last} is the previous encoder count,
- C_{rev} is the encoder counts per full revolution,
- r is the wheel radius.

2) WHEEL VELOCITY

The instantaneous wheel velocity v_w is given by:

$$v_w = \frac{D}{\Delta t} \tag{12}$$

3) ROBOT LINEAR VELOCITY

The forward (linear) velocity v of the robot is the average of the left- and right-wheel velocities:

$$v = \frac{v_w^{\text{right}} + v_w^{\text{left}}}{2} \tag{13}$$

4) ANGULAR VELOCITY

The angular velocity ω is derived from the difference in wheel velocities, relative to wheelbase l as follows:

$$\omega = \frac{v_w^{\text{right}} - v_w^{\text{left}}}{l} \tag{14}$$

5) ORIENTATION UPDATE

The change in orientation angle $\Delta\theta$ and updated heading θ_{new} are calculated as

$$\Delta\theta = \omega \times \Delta t \tag{15}$$

$$\theta_{\text{new}} = \theta_{\text{old}} + \Delta\theta \tag{16}$$

6) POSITION UPDATE

Using the new orientation, the robot's position in Cartesian coordinates is updated using

$$x_{\text{new}} = x_{\text{old}} + v \times \Delta t \times \cos(\theta_{\text{new}}) \tag{17}$$

$$y_{\text{new}} = y_{\text{old}} + v \times \Delta t \times \sin(\theta_{\text{new}}) \tag{18}$$

7) EUCLIDEAN DISTANCE

To compute the straight-line distance from a reference point:

$$d = \sqrt{(x_{\text{new}} - x_{\text{ref}})^2 + (y_{\text{new}} - y_{\text{ref}})^2} \tag{19}$$

This forward kinematic model enables accurate tracking of the robot's pose and is essential for localisation, navigation, and mapping in land-mode operations.

E. BATTERY LIFETIME ESTIMATION

To evaluate the operational endurance of the robot in both flight and drive modes, the total power consumption and battery capacity were analysed. These values are derived from the component specifications listed in Table 7.

1) TOTAL POWER CONSUMPTION

The total power consumption (TPC) of the system is calculated as

$$TPC = V_o \times I = 3723.51 \text{ W} \tag{20}$$

This value represents the combined instantaneous power drawn by all components when operating at the maximum load.

2) BATTERY CAPACITY

The energy capacity of the LiPo battery is calculated from its rated voltage and charge:

$$BC = \frac{BC_{\text{mAh}} \times V_o}{1000} = 88.8 \text{ Wh} \tag{21}$$

3) FLIGHT MODE BATTERY LIFE

In the flight mode, where the primary power draw is dominated by the BLDC motors, the expected battery life is

$$t_{\text{flight}} = \frac{BC}{TPC} = \frac{88.8}{364} = 0.0244 \text{ hours} = 14.6 \text{ minutes} \tag{22}$$

Thus, the estimated maximum continuous flight time is approximately **14.46 minutes**.

4) DRIVE MODE BATTERY LIFE

In drive mode, the power consumption is significantly lower due to the use of geared DC motors. The estimated battery life is:

$$t_{\text{drive}} = \frac{BC}{TPC} = \frac{88.8}{5.76} = 15.41 \text{ hours} \tag{23}$$

Therefore, the robot is capable of operating on the ground for approximately **15.41 hours** under the maximum drive load conditions.

F. DETAILED BATTERY USAGE ANALYSIS

To enhance mission-level energy estimation and planning, a detailed power consumption analysis was conducted across the robot's main operational phases:

- **Idle Mode:** The system consumes approximately 6 W during idle phases on the ground, with energy drawn mainly by the onboard microcontroller, LiDAR, and communication modules.
- **SLAM Mapping:** When the robot is performing SLAM, CPU and sensor activity increase the average power consumption to 18 W. This reflects real-time processing and LiDAR scanning loads.
- **Navigation:** The highest energy consumption during ground operation occurs during navigation, reaching 21 W. This accounts for motor actuation and path planning computations.
- **Aerial Hover:** A significant power spike to 246 W is observed in aerial hover mode, driven by continuous operation of four brushless motors for lift. Although efficient for short-term traversal, this phase is energy-intensive and justifies the hybrid design.
- **Transition Phase:** The brief transition between modes consumes 12 W, attributed to the simultaneous shutdown of the ground system and preparation of the flight controller.

The total power consumption of the robotic platform can be estimated by summing the current draw of all onboard components, including sensors, actuators, microcontrollers, and other peripheral electronics [42].

Based on the current and voltage specifications listed in Table 7, Table 8 presents the average power consumption and task duration for each operational phase of the hybrid robot. Using these values, the total energy consumption is then computed in watt-hours (Wh) as follows:

$$E = \frac{P \times t}{3600}$$

Where P is the average power in watts, and t is the duration in seconds. The corresponding energy usage graph is shown in Figure 8.

TABLE 8. Energy consumption breakdown by task phase.

Phase	Power (W)	Time (s)	Energy (Wh)
Idle (Ground)	6	300	0.50
SLAM Mapping (Ground)	18	420	2.10
Navigation (Ground)	21	600	3.50
Aerial Hover	246	60	4.10
Ground-to-Air Transition	12	15	0.05
Total			**9.62**

The total energy consumed during the representative mission is approximately 9.62 Wh, which is significantly lower than the 88.8 Wh capacity of the 4S 6000 mAh LiPo battery. This confirms the system's energy efficiency and highlights its endurance potential for extended inspection tasks.

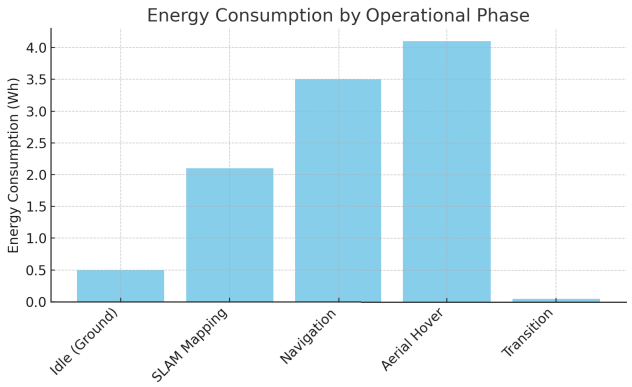


FIGURE 8. Energy consumption by operational phase.

V. HYBRID ROBOT PROGRAMMING

A. SLAM AND NAVIGATION STACK

The robot performed SLAM using the `slam_toolbox` package. This process involves fusing LiDAR data with encoder-based odometry to produce a map of the environment during exploration.

1) LiDAR scans are published to the topic `/scan`.
2) Odometry is derived from encoder pulses and integrated using Euler-based motion models.
3) The map is generated incrementally and stored in YAML/PGM formats for reuse.

The `Nav2` package is employed for localisation and autonomous path planning. The global planner uses Dijkstra's algorithm on a static costmap, whereas the local planner employs the Dynamic Window Approach (DWA) for real-time obstacle avoidance and smooth trajectory control.

B. ODOMETRY AND ENCODER INTEGRATION

The encoders attached to the rear wheels provide pulses that are converted into velocity estimates. ESP32 computes wheel displacements and publishes odometry via a Micro-ROS to the Raspberry Pi. The following kinematic equations govern the position update of the robot

$$v = \frac{v_r + v_l}{2}, \quad \omega = \frac{v_r - v_l}{L} \tag{24}$$

$$x_{t+1} = x_t + v \cdot \cos(\theta_t) \cdot \Delta t \tag{25}$$

$$y_{t+1} = y_t + v \cdot \sin(\theta_t) \cdot \Delta t \tag{26}$$

$$\theta_{t+1} = \theta_t + \omega \cdot \Delta t \tag{27}$$

where v_r, v_l are the right and left wheel velocities, respectively, L is the wheelbase, and θ is the heading of the robot.

C. CAMERA MODULE INTEGRATION

The Raspberry Pi Camera was integrated using the `Raspicam_node` in ROS 2. It streams video to the `/image_raw` topic, which is visualized in RViz or saved for subsequent inspection. This functionality supports remote monitoring and future extensions of computer vision-based inspections.

D. MANUAL FLIGHT MODE CONFIGURATION

A manual RC controller was used in flight mode. The motors were calibrated using PWM signals and ESC beeping sequences, followed by rotor direction adjustments via the BLHeli Configurator to align with the quad-X layout. The autonomous flight mode relies on the IMU within the flight controller to maintain orientation stability.

E. AUTONOMOUS MODE SWITCHING PROTOTYPE

The robot requires manual switching between ground and aerial modes, wherein the motors of one subsystem must be explicitly disabled before activating the other. A central power distribution board is used to deliver power from a shared 4S LiPo battery to both propulsion systems, with mode-specific voltage regulation handled via 3A and 12A buck converters.

To enhance autonomy and mission continuity, a preliminary prototype of automatic ground-to-air mode switching was developed and integrated with the existing ROS 2 framework. This prototype is based on real-time LiDAR scan analysis and local terrain assessment. A custom ROS 2 node continuously evaluates the minimum front-facing clearance and obstacle density using data from the SLAM and navigation stack. If the robot detects a non-traversable terrain (e.g., vertical obstruction, wall, or staircase), a transition flag is set to initiate an autonomous switch to aerial mode.

This flag initiates a controlled shutdown of the ground motors and activates the aerial flight controller. A minimum range threshold of 0.5 meters was used for obstacle detection, with transitions verified through simulation and confined-area testing. Figure 9 illustrates the decision logic for the proposed autonomous mode-switching process. The ROS 2 implementation for this switching logic is shown in Listing 1.

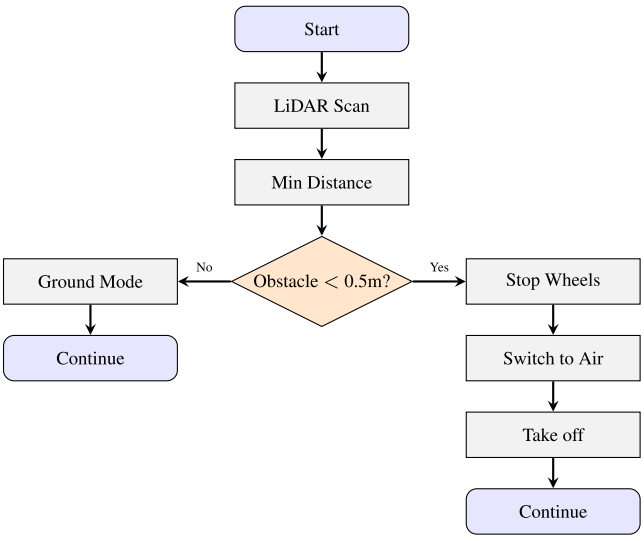


FIGURE 9. Mode-switching logic based on LiDAR obstacle distance.

While this approach is rule-based, future enhancements will integrate semantic terrain classification based on CNNs

```python
#!/usr/bin/env python3
import rclpy
from rclpy.node import Node
from sensor_msgs.msg import LaserScan
from std_msgs.msg import Bool

class ModeSwitcher(Node):
    def __init__(self):
        super().__init__('mode_switcher')
        self.threshold = 0.5  # meters
        self.subscription = self.
            create_subscription(
            LaserScan, '/scan', self.
                scan_callback, 10)
        self.publisher_ = self.
            create_publisher(Bool, '/
            trigger_aerial_mode', 10)
        self.get_logger().info("ModeSwitcher
            running...")

    def scan_callback(self, msg):
        valid_ranges = [r for r in msg.
            ranges if msg.range_min < r < msg
            .range_max]
        if not valid_ranges:
            return
        min_range = min(valid_ranges)
        if min_range < self.threshold:
            self.get_logger().warn("Obstacle
                detected! Switching to
                aerial mode.")
            self.publish_switch(True)

    def publish_switch(self, trigger):
        msg = Bool()
        msg.data = trigger
        self.publisher_.publish(msg)

def main(args=None):
    rclpy.init(args=args)
    node = ModeSwitcher()
    try:
        rclpy.spin(node)
    except KeyboardInterrupt:
        pass
    finally:
        node.destroy_node()
        rclpy.shutdown()

if __name__ == '__main__':
    main()
```

LISTING 1. ROS 2 Python node implementing autonomous mode switching based on LiDAR readings.

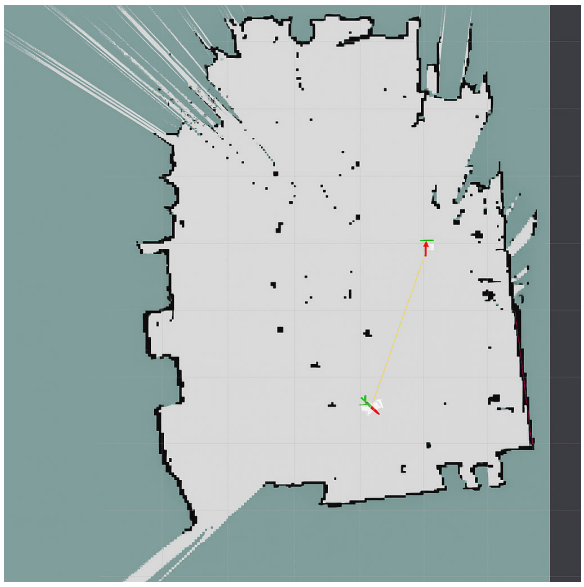


FIGURE 10. Robot maps obtained using SLAMalgorithm based on ROS.

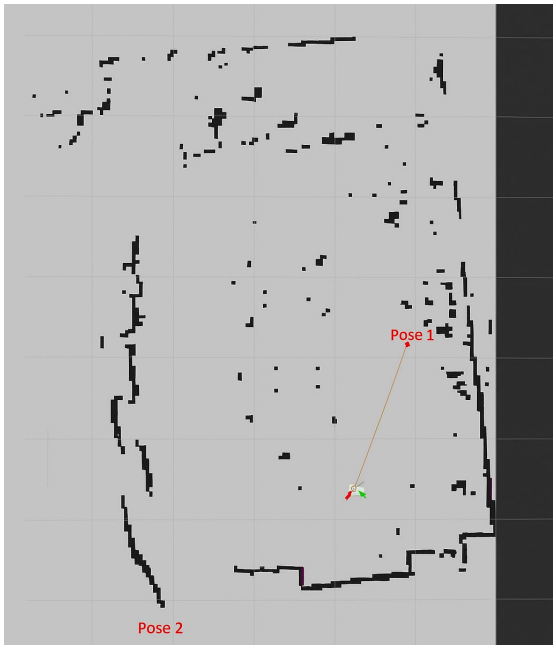


FIGURE 11. Final occupancy grid with Real-time SLAM.

trained on 2D LiDAR and RGB data to distinguish between climbable, passable, and fly-only regions. This will enable more context-aware decisions for transitioning between locomotion modes in unstructured or unknown environments.

VI. EXPERIMENTAL EVALUATION AND RESULTS

This section presents the experimental assessment of the hybrid land-air robot, including mapping accuracy, localisation performance, obstacle avoidance, flight control, and power consumption analysis. All experiments were conducted in an indoor laboratory environment simulating industrial inspection conditions.

A. DRIVE MODE EVALUATION

1) SLAM PERFORMANCE

Using the `slam_toolbox` and LiDAR integration, the robot successfully mapped a 12m × 8m area with clear obstacle boundaries. The mapping resolution was set to 0.05m per grid cell, and map drift was contained within 0.025m under normal operation. Figure 10 shows different maps for different scenarios using odometry data and LiDAR sensor scans to build the maps. Localisation was achieved using `Nav2` of the ROS 2 stack, where the robot's position remained within ±0.02m of its true coordinates. Figure 11 illustrates the final occupancy grid.

2) OBSTACLE AVOIDANCE AND NAVIGATION

The LiDAR sensor, combined with the DWA local planner, allowed real-time avoidance of static and dynamic obstacles. The robot was able to reroute with minimal oscillation and successfully reach nine out of ten randomly placed goals within a 30s window per trial. The minimum safe distance from obstacles was maintained at approximately 5cm, and the system responded to new obstacles within 0.2s. Figure 12 shows the hybrid robot navigating in an indoor environment.

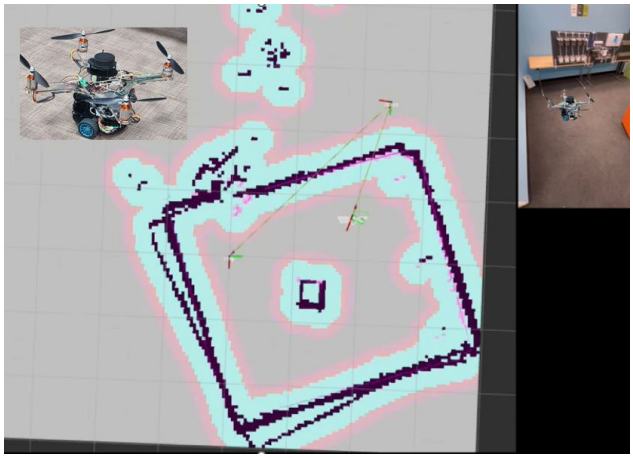


FIGURE 12. Hybrid robot during navigation operation.

3) TRAJECTORY TRACKING AND SMOOTHNESS

To assess trajectory adherence, we used the RViz to compare the planned and actual paths. The mean deviation was less than 0.015m for straight-line paths and less than 0.025m during curved motions. The wheel encoder data confirmed the accuracy of the turning radius and forward speed consistency. Global planning is responsible for computing an optimal path from the robot's starting position to its goal, while considering static obstacles. It relies on a ''global costmap'', which represents the environment based on a predefined map. In this study, we chose the ''Dijkstra'' algorithm, which is a breadth-first search algorithm that explores all possible paths and selects the one with the lowest accumulated cost. The ''global planner'' executes once when the robot receives a navigation goal, generating a path that avoids obstacles while adhering to the costmap constraints. Figure 13 shows the path plan of the costmap.

B. FLIGHT MODE EVALUATION

1) STABILITY AND HOVER TESTS

The robot was tested in manual flight mode under no-wind indoor conditions. This demonstrates a stable hover. The quad-X configuration ensures that the roll and yaw control remain within expected margins.

2) TAKEOFF AND LANDING PRECISION

Controlled takeoff and landing tests showed consistent vertical thrust delivery and no significant lateral drift. Central

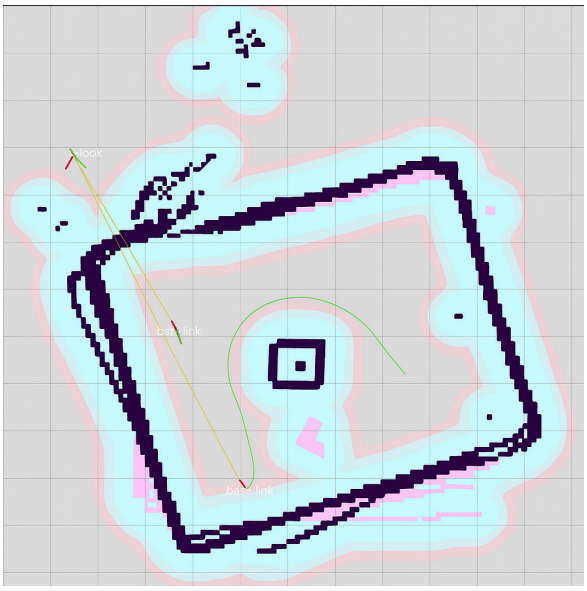


FIGURE 13. Occupancy grid path planning in ROS.

battery placement contributed to a low centre of gravity, preventing tip-over events during the descent.

3) MANEUVERABILITY AND ORIENTATION CONTROL

The built-in IMU of the flight controller enables responsive roll, pitch, and yaw control. However, due to aerodynamic asymmetries from the ground-wheel frame, the lateral flight exhibited slight drift during aggressive turns.

C. POWER CONSUMPTION AND BATTERY LIFE

The robot's power consumption was logged during both operation modes:

- **Ground Mode:** Operating at an average of 18 W with an estimated runtime of 15.4 hours.
- **Flight Mode:** Consumed approximately 246 W with a theoretical flight time of 1.5 minutes on a full 4S 6000mAh battery.

This highlights the efficiency of using the ground mode for the majority of inspection operations, reserving flight mode for brief, and situational deployment.

D. AUTONOMOUS AERIAL NAVIGATION

Although the current implementation supports only manual aerial control, we propose a structured development plan to enable full autonomy in aerial navigation. Given laboratory and hardware limitations, this work has not been implemented yet. However, the following roadmap outlines the future phases required to achieve this objective.

1) PHASE 1: ARCHITECTURE SETUP AND INTEGRATION

Objective: Enable ROS 2-based offboard control using PX4 for aerial navigation.

Tasks:

- Configure the PX4-ROS 2 bridge via `px4_ros_com` or MAVROS.

- Enable `OFFBOARD` mode control via ROS 2 publishers.
- Define the UAV pose/velocity reference interface for flight commands.

Software Stack: ROS 2 Humble, PX4 Autopilot (v1.13+), `px4_ros_com` or MAVROS, QGroundControl.

Hardware Requirements: Pixhawk 4 (existing), Raspberry Pi 4 (Ubuntu 22.04), RC failsafe system.

2) PHASE 2: WAYPOINT-BASED FLIGHT CONTROL

Objective: Create and execute waypoint missions autonomously in a predefined environment.

Tasks:

- Upload waypoints using MAVROS service `/mavros/mission/push`.
- Execute position commands via `/mavros/setpoint_position/local`.
- Test takeoff, position hold, waypoint tracking, and landing.

Software Stack: `mavros_msgs/WaypointPush`, PX4 in `OFFBOARD` mode, ROS 2 waypoint publisher node.

Hardware Requirements: Optical flow sensor (if GPS not used), onboard barometer and IMU (integrated with Pixhawk).

3) PHASE 3: VISUAL-INERTIAL ODOMETRY (VIO)

Objective: Enable onboard localisation in GPS-denied environments using depth + IMU fusion.

Tasks:

- Integrate depth camera (e.g., Intel RealSense D435i or ZED 2).
- Run VIO packages such as VINS-Fusion, ORB-SLAM3, or RTAB-Map on the companion computer.
- Fuse VIO output with IMU data via PX4 EKF2 or an external estimator.

Software Stack: `vins-fusion`, `rtabmap_ros`, `realsense2_camera` or ZED drivers, ROS 2-PX4 VIO bridge.

Hardware Requirements: Intel RealSense D435i or ZED 2 camera, lightweight camera mount.

4) PHASE 4: ONBOARD TRAJECTORY PLANNING

Objective: Generate and execute obstacle-aware 3D trajectories onboard the robot.

Tasks:

- Use ROS 2 planners (e.g., `moveit2`, `nav2`) to build 3D trajectories.
- Fuse environmental maps via OctoMap or point cloud-based costmaps.
- Implement obstacle-aware motion execution and recovery behaviors.

Software Stack: `nav2`, `octomap_server`, `voxblox`, `path_follower`, `mpc_local_planner`.

5) PHASE 5: MISSION INTEGRATION AND AUTONOMY

Objective: Achieve high-level autonomy including terrain-triggered aerial activation and landing.

Tasks:

- Monitor terrain via LiDAR, VIO, or map.
- Implement terrain-aware ROS 2 decision logic (e.g., height thresholds).
- Activate aerial mode based on mission triggers and execute return-to-ground behavior.

Software Stack: Custom ROS 2 `autonomy_manager` node, integrated with existing SLAM and mission planner.

Hardware Requirements: No additional hardware if LiDAR and VIO are already installed.

This plan will guide the staged implementation of autonomous aerial navigation in ROS 2. Each phase is modular and can be validated independently before full mission integration. Once realised, this autonomy framework will enable the hybrid robot to operate effectively in inspection scenarios with minimal user intervention.

E. VISUAL INSPECTION CAPABILITIES

The integrated Raspberry Pi camera successfully streamed real-time video to the ROS graph and visualised it using RViz. This allows the user to inspect assets remotely, capture frame snapshots, or extend vision-based anomaly detection.

VII. CONCLUSION AND FUTURE WORK

This article presents the design and validation of a lightweight, low-cost hybrid ground-aerial robot using a ROS2-based architecture. Addressing a gap in the literature, the proposed system integrates 3D-printed modular hardware and open-source software to enable seamless land-air operation within a unified control framework. The robot demonstrated autonomous SLAM-based navigation, real-time obstacle avoidance in the ground mode, and stable manual flight for rapid deployment in constrained environments. Experimental results confirm the effectiveness of the platform in mapping and inspection tasks, with energy-efficient performance favouring ground-based operation. These findings support the utility of hybrid mobile systems in complex indoor scenarios, particularly for inspection, surveillance, and exploratory missions that require adaptable mobility.

The proposed design distinguishes itself by combining a modular ROS 2-based architecture with seamless integration of LiDAR-based obstacle detection to initiate aerial mode, significantly improving switching autonomy over prior semi-automatic systems. Additionally, our quantified energy analysis across operational modes (Table 8) demonstrates superior efficiency (93% savings on ground vs. aerial), which is not systematically reported in other platforms. The planned trajectory planner and onboard VIO for GPS-denied flight expand the autonomy frontier, bridging practical deployment with research-focused innovation.

A. FUTURE WORK

Several directions are envisioned to enhance the autonomy, robustness, and field applicability of the proposed hybrid aerial-ground inspection robot.

1) AUTONOMOUS AERIAL NAVIGATION

As discussed in Section VI-D, future work will focus on implementing and validating fully autonomous aerial navigation using ROS 2-PX4 integration. This includes waypoint missions, onboard VIO for GPS-denied localization, and dynamic 3D path planning in cluttered environments.

2) OUTDOOR TESTING AND GPS INTEGRATION

While the current evaluation focused on indoor environments due to institutional restrictions on outdoor UAV operations, future work will extend testing to outdoor scenarios. The hybrid robot's controller and navigation stack are already compatible with GPS and RTK modules via the PX4-ROS 2 interface. Planned outdoor trials will integrate GPS-based global positioning fused with IMU and barometric data using the PX4 EKF2 estimator to enhance robustness under real-world conditions, including wind disturbances and multipath interference. Once regulatory approval is obtained, these experiments will validate full-system performance in open and semi-structured environments. Additionally, future phases will explore terrain-aware autonomy, aerial mapping, and long-range waypoint missions to support infrastructure inspection tasks at scale. The system will be tested in representative infrastructure inspection scenarios (e.g., power substations, warehouses, and bridges), including mixed indoor-outdoor operations. These scenarios will be used to evaluate real-time SLAM continuity, mode-switching robustness, and mission-level autonomy.

REFERENCES

[1] J. L. Chien, C. Leong, J. Liu, and S. Foong, "Design and control of an aerial-ground tethered tendon-driven continuum robot with hybrid routing," *Robot. Auto. Syst.*, vol. 161, Mar. 2023, Art. no. 104344.

[2] J. P. Ramirez and S. Hamaza, "Multimodal locomotion: Next generation aerial–terrestrial mobile robotics," *Adv. Intell. Syst.*, Dec. 2023, Art. no. 2300327, doi: 10.1002%2Faisy.202300327. [Online]. Available: https://advanced.onlinelibrary.wiley.com/action/showCitFormats

[3] M. Ryalat, N. A. Rawashdeh, N. Abu Alrub, and H. Elmoaqet, "Mechatronics design and control of a hybrid flying-ground robot for long-endurance mobility," in *Proc. 12th Int. Conf. Control, Mechatronics Autom. (ICCMA)*, Nov. 2024, pp. 336–342.

[4] A. Lopez-Lora, P. J. Sanchez-Cuevas, A. Suarez, A. Garofano-Soldado, A. Ollero, and G. Heredia, "MHYRO: Modular HYbrid RObot for contact inspection and maintenance in oil & gas plants," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 1268–1275.

[5] S. Latscha, M. Kofron, A. Stroffolino, L. Davis, G. Merritt, M. Piccoli, and M. Yim, "Design of a hybrid exploration robot for air and land deployment (H.E.R.A.L.D) for urban search and rescue applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1868–1873.

[6] F. Michaud, D. Letourneau, M. Arsenault, Y. Bergeron, R. Cadrin, F. Gagnon, M.-A. Legault, M. Millette, J.-F. Pare, M.-C. Tremblay, P. Lepage, Y. Morin, and S. Caron, "AZIMUT: A multimodal locomotion robotic platform," *Proc. SPIE*, vol. 5083, pp. 101–112, Sep. 2003.

[7] B. Hu, Z. Dong, and L. Zhang, "Land-air amphibious robots: A survey," in *Proc. CAAI Int. Conf. Artif. Intell.*, L. Fang, J. Pei, G. Zhai, and R. Wang, Eds., Singapore: Springer, 2024, pp. 575–586.

[8] B. Lindqvist, S. Karlsson, A. Koval, I. Tevetzidis, J. Haluška, C. Kanellakis, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue," *Robot. Auto. Syst.*, vol. 154, Aug. 2022, Art. no. 104134.

[9] K. Kim, P. Spieler, E.-S. Lupu, A. Ramezani, and S.-J. Chung, "A bipedal walking robot that can fly, slackline, and skateboard," *Sci. Robot.*, vol. 6, no. 59, p. 8136, Oct. 2021.

[10] W. D. Shin, H.-V. Phan, M. A. Daley, A. J. Ijspeert, and D. Floreano, "Fast ground-to-air transition with avian-inspired multifunctional legs," *Nature*, vol. 636, no. 8041, pp. 86–91, Dec. 2024.

[11] Y. M. Chukewad, J. James, A. Singh, and S. Fuller, "RoboFly: An insect-sized robot with simplified fabrication that is capable of flight, ground, and water surface locomotion," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 2025–2040, Dec. 2021.

[12] M. Cao, X. Xu, S. Yuan, K. Cao, K. Liu, and L. Xie, "DoubleBee: A hybrid aerial-ground robot with two active wheels," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 6962–6969.

[13] NASA Jet Propuls. Lab. (2021). *Subterranean Rollocopter*. Accessed: Apr. 17, 2025. [Online]. Available: https://www.jpl.nasa.gov/robotics-at-jpl/subterranean-rollocopter/

[14] J. Lin, R. Zhang, N. Pan, C. Xu, and F. Gao, "Skater: A novel bi-modal bi-copter robot for adaptive locomotion in air and diverse terrain," *IEEE Robot. Autom. Lett.*, vol. 9, no. 7, pp. 6392–6399, Jul. 2024.

[15] A. Kalantari and M. Spenko, "Modeling and performance assessment of the HyTAQ, a hybrid terrestrial/aerial quadrotor," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1278–1285, Oct. 2014.

[16] A. Kalantari, T. Touma, L. Kim, R. Jitosho, K. Strickland, B. T. Lopez, and A.-A. Agha-Mohammadi, "Drivocopter: A concept hybrid aerial/ground vehicle for long-endurance mobility," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–10.

[17] S. Sabet, A.-A. Agha-Mohammadi, A. Tagliabue, D. S. Elliott, and P. E. Nikravesh, "Rollocopter: An energy-aware hybrid aerial-ground mobility for extreme Terrains," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–8.

[18] A. J. Ijspeert, "Biorobotics: Using robots to emulate and investigate agile locomotion," *Science*, vol. 346, no. 6206, pp. 196–203, Oct. 2014.

[19] B. L. Suhadi, T. W. Z. Heng, S. K. H. Win, L. S. T. Win, and S. Foong, "Design and control of a ground-aerial dual actuator monocopter (G-ADAM)," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jun. 2023, pp. 25–32.

[20] J. Wang, Z. Sun, X. Guan, T. Shen, D. Huang, Z. Zhang, T. Duan, F. Liu, and H. Cui, "HE-nav: A high-performance and efficient navigation system for aerial-ground robots in cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 11, pp. 10383–10390, Nov. 2024.

[21] Z. Yu, J. Du, J. Wang, and Z. Lin, "Prescribed-performance green dynamic positioning for fully actuated vessels under input magnitude and rate saturations," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 14940–14952, 2025.

[22] J. Matute, S. Diaz, and A. Karimoddini, "Sliding mode control for robust path tracking of automated vehicles in rural environments," *IEEE Open J. Veh. Technol.*, vol. 5, pp. 1314–1325, 2024.

[23] C. Premachandra, M. Otsuka, R. Gohara, T. Ninomiya, and K. Kato, "A study on development of a hybrid aerial / terrestrial robot system for avoiding ground obstacles by flight," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 327–336, Jan. 2019.

[24] Z. Zheng, Q. Cai, J. Wang, X. Xu, M. Cao, H. Yu, J. Li, J. Meng, and G. Lu, "CapsuleBot: A novel hybrid aerial-ground bi-copter robot with two actuated-wheel-rotors," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, pp. 120–127, Jan. 2025.

[25] S. Misra, K. Sundar, R. Sharma, S. Mukherjee, K. Gupta, S. Paul, S. Mondal, N. Kumar, and S. Singh, "Deployable, data-driven unmanned vehicle navigation system in gps-denied, feature-deficient environments," *J. Intell. Robotic Syst.*, vol. 105, no. 44, Jun. 2022, doi: 10.1007/s10846-022-01647-8.

[26] A. Pretto et al., "Building an aerial–ground robotics system for precision farming: An adaptable solution," *IEEE Robot. Autom. Mag.*, vol. 28, no. 3, pp. 29–49, Sep. 2021.

[27] M. S. Mondal, S. Ramasamy, J. D. Humann, J. M. Dotterweich, J.-P. F. Reddinger, M. A. Childers, and P. Bhounsule, "A robust UAV-UGV collaborative framework for persistent surveillance in disaster management applications," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2024, pp. 1239–1246.

[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009, p. 5.

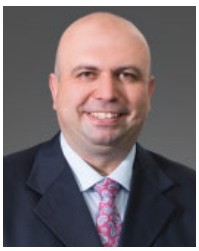
[29] Y. Maruyama, S. Kato, and T. Azumi, "ROS 2: Towards a real-time middleware for robotics," *J. Open Source Softw.*, vol. 5, no. 50, p. 2702, 2020.

[30] D. Chikurtev, "Mobile robot simulation and navigation in ROS and gazebo," in *Proc. Int. Conf. Automatics Informat. (ICAI)*, Oct. 2020, pp. 1–6.
[31] R. K. Megalingam, A. Rajendraprasad, and S. K. Manoharan, "Comparison of planned path and travelled path using ROS navigation stack," in *Proc. Int. Conf. Emerg. Technol. (INCET)*, Jun. 2020, pp. 1–6.
[32] A. S. Al-Batati, A. Koubaa, and M. Abdelkader, "ROS 2 key challenges and advances: A survey of ROS 2 research, libraries, and applications," *Preprint*, May 2024.
[33] A. Mamani-Saico and P. Raul Yanyachi, "Implementation and performance study of the micro-ROS/ROS2 framework to algorithm design for attitude determination and control system," *IEEE Access*, vol. 11, pp. 128451–128460, 2023.
[34] G. Chen, H. Yang, and I. M. Mitchell, "ROS-X-habitat: Bridging the ROS ecosystem with embodied AI," in *Proc. 19th Conf. Robots Vis. (CRV)*, May 2022, pp. 24–31.
[35] Jetson AI Lab. (2023). *ROS and Robotics*. Accessed: Apr. 17, 2025. [Online]. Available: https://www.jetson-ai-lab.com/ros.html
[36] PX4 Autopilot Develop. Team. (2024). *ROS 2 User Guide*. Accessed: Apr. 17, 2025. [Online]. Available: https://docs.px4.io/main/en/ros2/userguide
[37] S. Macenski, T. Moore, D. V. Lu, A. Merzlyakov, and M. Ferguson, "From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2," *Robot. Auto. Syst.*, vol. 168, Oct. 2023, Art. no. 104493.
[38] A. S. Al-Batati, A. Koubaa, and M. Abdelkader, "ROS 2 in a nutshell: A survey," *Preprints*, Oct. 2024.
[39] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "RViz: A toolkit for real domain data visualization," *Telecommun. Syst.*, vol. 60, no. 2, pp. 337–345, Oct. 2015.
[40] A. V. Javir, K. Pawar, S. Dhudum, N. Patale, and S. Patil, "Design, analysis and fabrication of quadcopter," *J. Adv. Res. Mech. Civil Eng.*, vol. 2, no. 3, pp. 16–27, Mar. 2015.
[41] G. Klanar, A. Zdear, S. Bla, and I. Krjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Oxford, U.K.: Butterworth, 2017.
[42] T. Abukhalil, H. Almahafzah, M. Alksasbeh, and B. A. Y. Alqaralleh, "Power optimization in mobile robots using a real-time heuristic," *J. Robot.*, vol. 2020, pp. 1–8, Feb. 2020.

MUTAZ RYALAT (Senior Member, IEEE) was born in Amman, Jordan. He received the B.Sc. degree in mechatronics engineering from the University of Jordan, Amman, in 2005, the M.Sc. degree in mechatronics engineering from Loughborough University, Loughborough, U.K., in 2007, and the Ph.D. degree in nonlinear and robust control from the University of Southampton, Southampton, U.K., in 2015. In Summer 2024, he was a Visiting Professor at the Hamlyn Centre for Robotic Surgery, Imperial College London, U.K. He is currently an Associate Professor with the Department of Mechatronics Engineering and the Director of the Robotics and Autonomous Systems Hub, German Jordanian University, Amman. His research interests include nonlinear and robust control of mechanical and electromechanical systems, Hamiltonian systems, mechatronics system design, robotic systems, Industry 4.0, and AI. He serves as the Chair for the IEEE Robotics and Automation Society (RAS) Jordan Section. He actively serves as a reviewer for several journals and international conferences in the field of mechatronics engineering.

GHAITH AL-REFAI (Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from Jordan University of Science and Technology, Irbid, Jordan, in 2010, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from Oakland University, MI, USA, in 2014 and 2018, respectively. He is currently an Assistant Professor with the Mechatronics Engineering Department, German Jordanian University, Amman, Jordan. His research interests include autonomous driving, computer vision, machine intelligence, AI, and robotics. He actively reviews various journals and conferences in the field of mechatronics engineering.

NATHEER ALMTIREEN (Member, IEEE) received the B.Sc. degree in mechatronics engineering from the University of Jordan, in 2010, the M.Sc. degree in advanced mechanical engineering from the University of Southampton, in 2012, and the Ph.D. degree in mechanical engineering from Karlsruhe Institute of Technology (KIT), in 2022. He is currently appointed as an Assistant Professor with the Mechatronics Engineering Department, German Jordanian University. His research topics mainly include control systems, sensors, automation, modeling and simulation of engineering systems, and engineering applications of artificial intelligence.



HISHAM ELMOAQET (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering from Jordan University of Science and Technology (JUST) in 2002 and 2006, respectively, with a focus on mechatronics, and the Ph.D. degree in mechanical engineering specializing in mechatronics from the University of Michigan, Ann Arbor, MI, USA, in 2015. He was a Research Fellow with the Department of Mechanical Engineering and Anesthesiology, University of Michigan, from 2015 to 2016. He joined the Mechatronics Engineering Department, German Jordanian University (GJU) as an Assistant Professor, in 2016. In Summer 2017, he was a Visiting Professor with the Institute of Technical Medicine, Furtwangen University of Applied Sciences, Villingen-Schwenninge, Germany. He is currently a Visiting Research Scientist at the Cardiovascular Research Group, Humboldt University, Berlin, Germany. His current research interests include dynamic systems, modeling, identification, and control, machine learning, physiological systems, and mechatronics systems design. He serves as the Chair for the IEEE Jordan Section Engineering in Medicine and Biology Society (EMBS) Chapter.