JOURNAL OF AUTOMOTIVE AND MECHANICAL APPLIED TECHNOLOGY

Vol. 3 No 1, 2026, pp. 12-25 https://journal.uny.ac.id/publications/jamat https://doi.org/10.21831/jamat.v3i1.3018

ArUco Marker-Based Autonomous UAV Navigation for Reconnaissance Operations in Urban Terrain Environments

Cahya Dwi Pradika[1*], Imanuel Dindin[1], Erzi Agson Gani[1], Ardan Nagra Coutsar[1], Mochamad Riza Pratama[2]

[1]Department of Power Motion Technology, Faculty of Engineering & Technology, Universitas Pertahanan, Citereup, Bogor 16810, Indonesia
[2]Department of Electrical Automation Engineering, Faculty of Vocational, Institute Technology Sepuluh Nopember, Surabaya 60111, Indonesia

*Corresponding author: cahya.pradika@tp.idu.ac.id

ARTICLE INFO

Article history:
Received 28.01.2026
Revised 01.03.2026
Accepted 27.03.2026

Keywords:
ArUco-Marker navigation; Unmanned aerial vehicle; Reconnaissance operations; LiDAR sensor

ABSTRACT

This study demonstrated the feasibility of autonomous UAV navigation in GPS-denied indoor environments using ArUco marker-based visual localization integrated with a VL53L1X LiDAR sensor and PX4 Offboard control. The developed system successfully validated markers, performed real-time pose estimation, and navigated sequentially through waypoints without human intervention. The web-based monitoring interface and QGroundControl integration operated reliably throughout all trials, enabling effective dual-platform telemetry monitoring and manual setpoint adjustment from a safe standoff position. The ArUco-marker-based detection, implemented using the OpenCV DICT_5×5_250 dictionary, validated marker identities within a functional altitude range of 40 to 200 cm. Third, across 61 trials discrete movement samples spanning four path configurations—straight-axis, lateral-right, lateral-left, and compound multi-direction—the system achieved an overall navigation success rate of 70%. Navigation failures caused by synchronization lag between UAV translational velocity and the camera's image processing frame rate, which prevented timely marker validation during high-speed maneuvers. These results confirm that ArUco marker-guided UAV navigation is a viable, low-infrastructure solution for initial indoor reconnaissance in GPS-denied military environments, and establish a quantitative baseline for future enhancements, including precision landing algorithms and dynamic marker placement strategies.

1. Introduction

Contemporary armed conflict has shifted decisively toward urban and subterranean theatres. Military Operations in Urban Terrain (MOUT) routinely require forces to clear multi-story buildings, underground facilities, and hardened infrastructure, where reliable intelligence on room layout and threat disposition can determine mission success [1]. The conventional approach of deploying scouts or point personnel entails an immediate risk of casualties before any actionable information is obtained. Autonomous UAVs operating within C4ISR frameworks significantly enhance situational awareness for commanders, enabling data collection in urban areas without exposing personnel [2].

Unmanned Aerial Vehicles (UAVs) or drones can navigate vertical obstacles without dedicated infrastructure and hover to capture or interpret sensor data [3]. The primary challenge remains navigation, as outdoor deployments typically rely on GPS for continuous position feedback. However, GPS signals attenuate sharply inside reinforced concrete structures [4]. Without GPS, the flight controller cannot maintain a position loop, causing the UAV to drift uncontrollably unless an alternative localization source is provided [1].

Marker-based localization systems are recognized as a practical solution for indoor environments. ArUco markers are particularly effective, combining compact size, binary encoding, and efficient detection via the OpenCV library, enabling real-time six-degree-of-freedom pose estimation from a single camera frame, making them a computationally efficient

option for resource-constrained platforms such as companion computers onboard UAVs [5]. Each marker encodes a unique identifier within a grid of black-and-white cells. T OpenCV ArUco module can locate a marker's corners in a single camera frame, correct for perspective distortion, and output a six-degree-of-freedom pose estimate [5].

This article presents the design, implementation, and experimental evaluation of an autonomous UAV system that navigates exclusively by reading ArUco markers mounted in advance on the floor. The UAV hardware combines a Diatone Mamba F405 flight controller running PX4 firmware, a Raspberry Pi 3 B+ running ROS, a camera, and a VL53L1X LiDAR distance sensor [6]. Navigation setpoints are entered through a web-based. The drone then executes the corresponding trajectory in offboard mode without any manual intervention beyond an emergency radio-controller override. Four path configurations were tested, and the results provide a quantitative analysis of how the system performs in terms of operational reliability.

2. Methodology

This section describes the methodology used to design, implement, and validate the ArUco Marker-based autonomous UAV navigation system. The methodology is structured into four sub-sections: (2.1) Hardware Configuration, (2.2) Software Architecture & Navigation, (2.3) ArUco-Marker Navigation, and (2.4) Simulation for Reconnaissance in Urban Terrain Environments.

2.1 Hardware Configuration

The UAV 3D Design shown in **Fig. 1.** is built on a length and width of 450 × 450 mm, a height of 150 mm, with a carbon-fiber frame in a standard X configuration (Quadcopter), where four brushless DC motors sit at the frame corners and contribute jointly to roll, pitch, yaw, and thrust. This geometry maximizes the forward camera field of view. It keeps the motor downwash away from the nadir-pointing sensor payload [7].

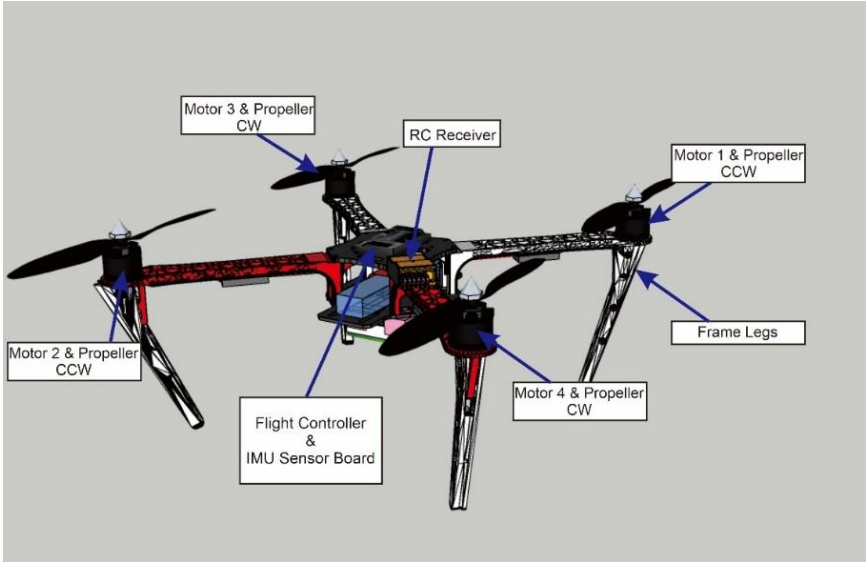

Figure 1. 3D Design UAV

Table 1. Hardware Components and Role in Reconnaissance System

Component	Specification	Role in system
Flight Controller Unit (FCU)	Diatone Mamba F405 Mk.II - STM32F405RGT 168 MHz 192-1024 KB PX4 v.1.13.0	Motor controller
Main Computer	Raspberry Pi 3 B+ 1.2 GHz 1 GB - Ubuntu 20.04 LTE	Vision processing, navigation logic, GCS communication
Camera	Sony IMX219 – 640×480 @ 90 fps CSI	ArUco marker-based detection
Motor (x4) + Propeller	Brushless DC DJI 2212 T, 2xCW, 2xCCW, 920kV + ESC	Propulsion in X-frame configuration
Altitude Sensor	VL53L1X LiDAR	Z-axis seeding prior to marker acquisition
Acceleration Sensor IMU	MPU6500 *Accelerometer (3 axes) Gyroscope (3 Axes)* SPI (*Serial Peripheral Interface*)	
Airframe	Carbon fiberglass X-frame, 450 × 450 mm F450	Structural platform

Table 1 shows the hardware components and their roles in the reconnaissance system. The Diatone Mamba F405 Mk.II hosts PX4 Autopilot firmware v1.13.0 and handles motor mixing, attitude estimation via the onboard IMU, and execution of position setpoints received over the MAVLink protocol. An STM32F103 microcontroller mediates signal routing between the RC safety transmitter and the Raspberry Pi GPIO outputs. The Raspberry Pi 3 B+ runs Ubuntu 20.04 LTS and ROS, it performs all image processing, navigation logic, and GCS communication. The MAVROS ROS package bridges ROS topics to the PX4 MAVLink interface serial connection, enabling the Raspberry Pi to command the flight controller in Offboard mode without proprietary middleware. The VL53L1X time-of-flight LiDAR is mounted approximately 19 cm above the landing gear's floor-contact surface. Its output therefore, requires a fixed offset correction to yield absolute altitude above ground. The sensor is initialized during the pre-flight check sequence. If its reading exceeds a configurable threshold (set to 5 cm in these trials), the PX4 altitude estimator accepts the LiDAR as the primary Z-axis source until the first Ar-Uco marker-based estimate becomes available.

2.2 Software Architecture & Navigation

The autonomous UAV navigation architecture for reconnaissance is shown in **Fig. 2.** The software architecture, shown in blue dashed lines, represents the primary focus of this research: the computer vision-based navigation system. The Pi Camera is connected to a Raspberry Pi 3 B+, which serves as the companion computer. Within this block, image processing is performed using the OpenCV library. The process begins with preprocessing, specifically image thresholding, to segment the background from potential marker candidates. The system then performs edge detection to locate the four corner points, allowing the image perspective to be transformed into a symmetrical square [8]. The validation process is conducted through binarization, matching the black-and-white matrix against the ArUco 5x5 dictionary. Once a marker is validated, the system estimates a 3D pose, which is subsequently transmitted to the Flight Controller Unit (FCU) [3].

Navigation setpoints are managed via a terminal on the web interface using Cartesian coordinates (X, Y, Z). These coordinate values align with the pre-designed ArUco marker floor plan. The coordinate data is sent to the Raspberry Pi for further processing. Upon initiation of flight instruction, the drone executes take-off procedures and navigates using the Autonomous Offboard Mode in the PX4 firmware, enabling full control from the companion computer via MAVROS.

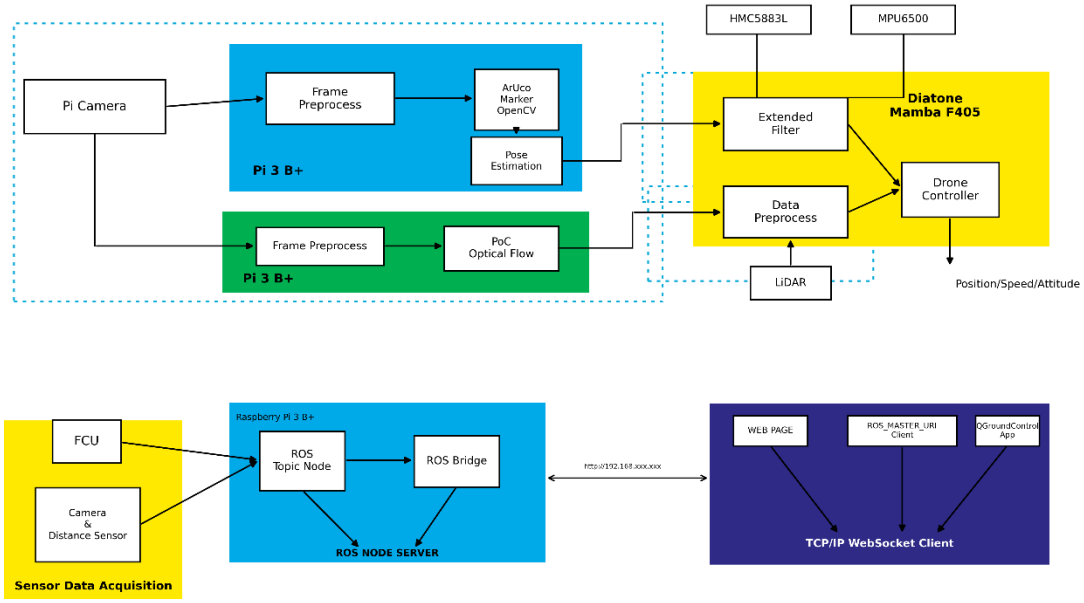

Figure 2. Software Architecture

The system integration concept utilizes the ROS framework as the primary middleware on the Raspberry Pi. The dark blue block illustrates the implementation of the ROS Node Server and ROS Bridge, which facilitate data communication [9]. The yellow block represents the data acquisition unit, encompassing the vision sensor, LiDAR, and inertial data (IMU) from the FCU (Mamba F405). Unidirectional arrows indicate one-way data transmission (e.g., raw sensor streams), while bidirectional arrows indicate bidirectional communication (e.g., control commands and status feedback). Communication between the drone system and external devices (UI) is conducted via HTTP/WebSocket protocols to the client-side (the rightmost dark blue block), which includes the web page and the Q-Ground Control application as the interface for remote monitoring and control [10].

2.3 ArUco-Marker Navigation

The ArUco markers used in this study are generated from the DICT_5X5_250 dictionary. Each marker consists of a wide black border enclosing a 5×5 binary matrix, where the encoded bit pattern uniquely identifies the marker ID. The detection pipeline, implemented using the OpenCV ArUco mule, performs image thresholding, edge detection to locate the four corner points, perspective transformation to normalize the marker to a square, and binarization against the dictionary to validate the marker identity [5]. The autonomous navigation in this study operates through PX4's Offboard control mode, in which position setpoints are transmitted to the flight controller via the MAVLink protocol [11].

2.4 Experimental Setup for Reconnaissance in Urban Terrain Environments

A validation environment was constructed to characterize the performance of the ArUco Marker-based navigation system under controlled conditions representative of Military Operations in Urban Terrain (MOUT). Four path configurations were defined and evaluated in the simulation environment, as shown in **Fig. 3**, each designed for a distinct maneuver type encountered in MOUT reconnaissance scenarios. The configurations range from a simple linear traversal to a compound multi-directional route representative of navigating through connected building corridors. Navigation setpoints were issued as coordinates (X, Y, Z) via the web-based terminal interface, with Z maintained at a constant 1m throughout all trials.

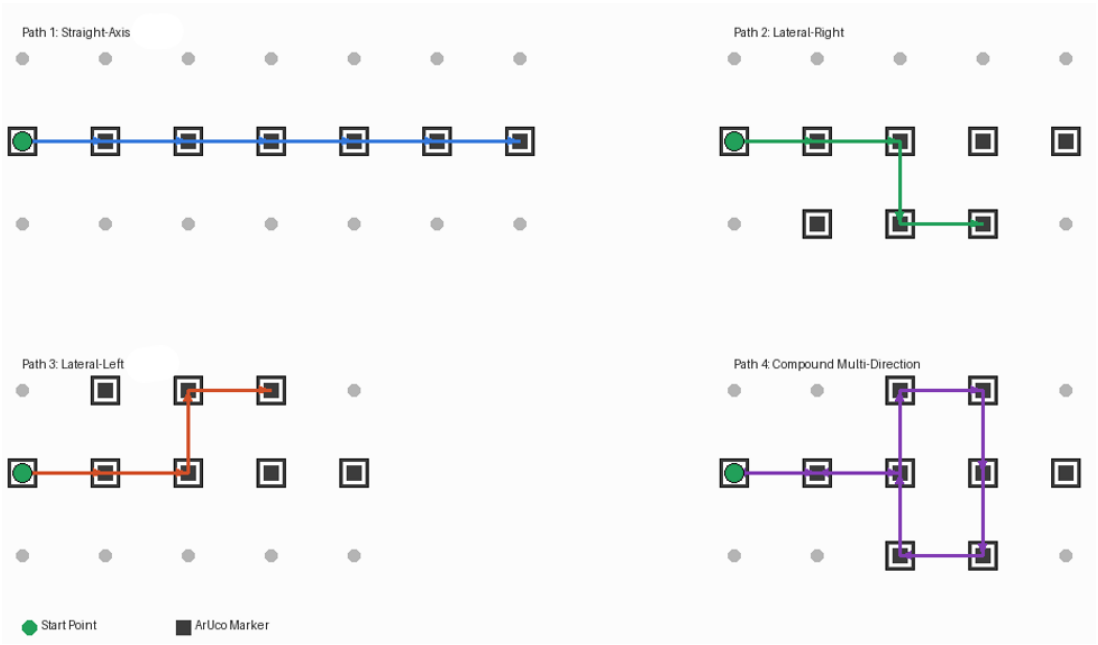

Figure 3. ArUco Markers-based Path Configuration

a. Path Configuration 1 – Straight-Axis Navigation: The UAV navigates along Segment A–B (IDs 0–8) in a single linear direction along the Y = 0 corridor axis. This configuration establishes the baseline navigation reliability for unidirectional travel. It is analogous to a UAV transiting a primary building corridor during initial entry reconnaissance.
b. Path Configuration 2 – Lateral-Right Transition: A compound path combining straight-axis travel along Segment A–B followed by a lateral right turn into Segment E–F (IDs 15–20), with a negative Y-axis setpoint. This configuration emulates the UAV transitioning from a primary corridor into a lateral room or side passage, representative of a reconnaissance sweep that requires coverage of multiple room entrances.
c. Path Configuration 3 – Lateral-Left Transition: Analogous to Configuration 2 but with a positive Y-axis lateral offset, directing the UAV into Segment C–D (IDs 9–14). This configuration emulates the UAV transitioning from a primary corridor into a lateral room or side passage, representative of a reconnaissance sweep that requires coverage of multiple room entrances.
d. Path Configuration 4 – Compound Multi-Direction Route: The most complex configuration, combining sequential transitions through all three segments (A–B → C–D → E–F → return to A), representing a full-building sweep trajectory. This configuration evaluated the system's endurance over the longest flight duration and the greatest number of marker-to-marker transitions.

3. Result and Discussion

3.1 Sensor Testing

The graph in **Fig. 4.** illustrates a strong linear correlation between the absolute height (ground truth) and the sensor readings, with the VL53L1X LiDAR tracking data points almost perfectly along the ideal reference line (y = x). The calculated coefficient of determination (R^2) is 0.9999, indicating that the sensor model accounts for nearly 100% of the observed variance and demonstrates exceptional precision across the tested range of 40 cm to 200 cm. The linear regression equation, y = 1.0001x - 0.95, further confirms the high degree of linearity,

indicating that the sensor's response remains proportional and consistent as altitude increases.

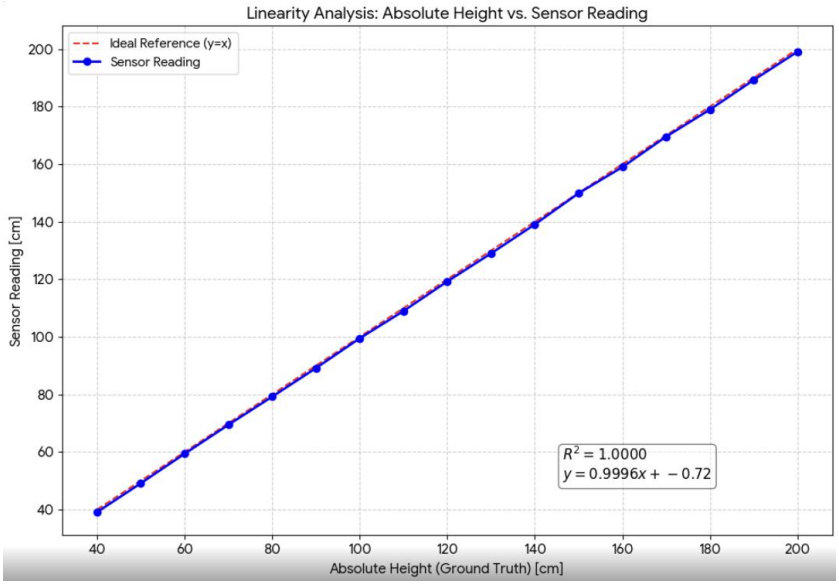

Figure 4. Linearity absolute height and sensor reading

Analytically, the results suggest that the LiDAR sensor is highly reliable for altitude estimation in GPS-denied environments. The slight negative intercept of approximately 0.95 cm represents a constant systematic offset, likely due to the sensor's physical mounting position on the drone frame or its internal zero-point calibration. However, the slope of 1.0001 indicates that the measurement gain is nearly with negligible with scaling errors. This level of accuracy is critical for maintaining a stable vertical position loop within the PX4 flight controller, as it ensures the autonomous system receives a truthful representation of its displacement from the floor with an average error of 0.77%.

3.2 System Integration Testing

System integration testing was conducted to validate data synchronization between the drone hardware, the custom web interface, and the QGroundControl (QGC) application, as shown in **Fig. 5**. The QGC architecture is used to monitor core parameters from the Flight Controller Unit (FCU) via the MAVLink protocol, including flight mode status (Offboard/Autonomous), essential telemetry, and battery levels [12]. Simultaneously, a dedicated web interface serves as the control hub for specialized features not natively supported by QGC in this configuration, including real-time ArUco marker detection and a terminal for manual navigation setpoint input. Experimental results indicate that pilot mode transitions via Remote Control are instantaneously synchronized across both platforms. This dual-monitoring ensures that high-bandwidth tasks, such as image processing on the Raspberry Pi, remain isolated from the FCU's critical stability loops while providing the operator with a comprehensive tactical overview through a unified web-based dashboard [13].

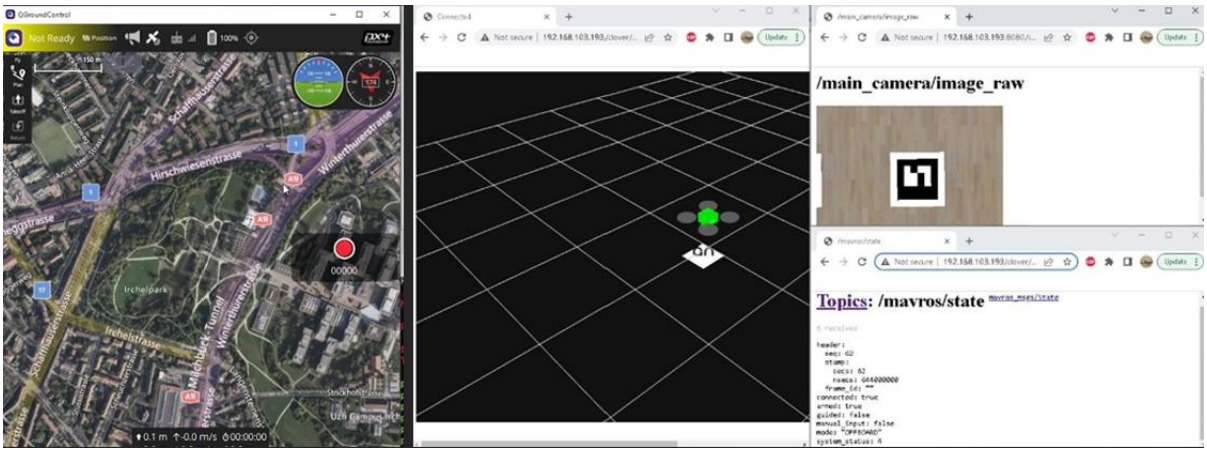

Figure 5. System Integration Between QGroundControl and Web-Based

3.3 Navigation System ArUco-Marker

This **Fig. 6.** Illustrates the ArUco marker-based navigation path configuration. A total of 21 markers (IDs 0–20) are systematically deployed and grouped into three primary trajectory segments: Segment A–B (IDs 0–8) serving as the straight navigation axis, Segment C–D (IDs 9–14) representing the left lateral path, and Segment E–F (IDs 15–20) representing the right lateral path. Each marker serves as a discrete spatial anchor mapped to Cartesian coordinates (X, Y), enabling the UAV to navigate autonomously in GPS-denied environments via real-time marker identification via computer vision. As illustrated in **Fig. 6**., categorized into three primary trajectory segments:

a. Segment A–B (Primary- Straight Axis): Consists of 9 markers (IDs 0–8). This segment serves as the central navigation spine, used for long-distance straight-axis trials and as the baseline for coordinate alignment.

b. Segment C–D (Upper Lateral Axis): Consists of 6 markers (IDs 9–14). This path is utilized to test lateral movement stability and the system's ability to maintain a heading while transitioning between marker sets.

c. Segment E–F (Lower Lateral Axis): Consists of 6 markers (IDs 15–20). Similar to segment C–D, this path provides additional data for lateral drift analysis and multi-directional waypoint sequencing.

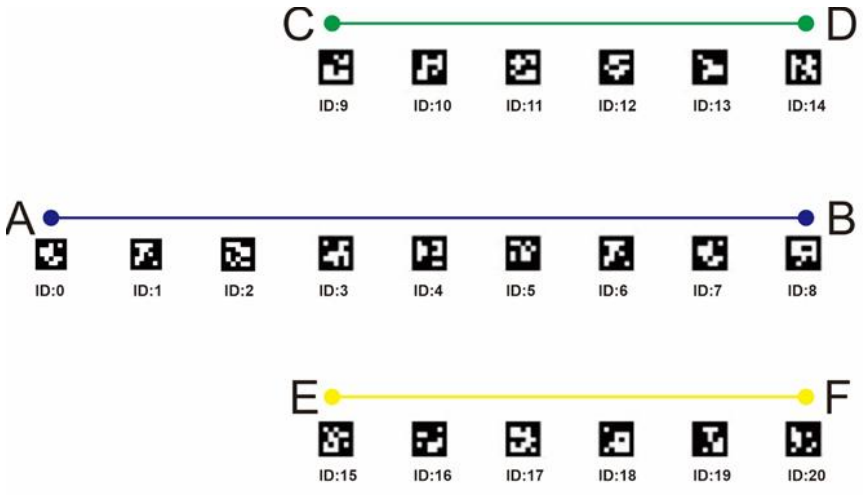

Figure 6. Testing ArUco Marker-Based

The UAV's companion computer (a Raspberry Pi 3 B+) decodes these IDs in real time to determine its current relative position. By mapping these identifiers to specific Cartesian coordinates (X, Y), the system can execute complex trajectories, such as a compound multi-directional route, by sequencing waypoints across these segments. This configuration enables quantitative evaluation of navigation reliability, particularly by identifying platform-specific issues such as the asymmetric drift observed during lateral-left maneuvers.

Figure 7. Camera Identification Trial Phase

Fig. 7 presents the results of the camera-based marker identification trial conducted as part of the ArUco marker–based UAV navigation system in an indoor environment. The onboard camera image shows real-time detection and identification of ArUco markers, with green bounding boxes, local coordinate axes, and corresponding marker ID labels highlighting validated markers. Marker detection and pose estimation are performed using the OpenCV ArUco module, which extracts the marker corner features and computes the relative pose with respect to the camera frame. This trial validates the robustness of the vision system under practical indoor lighting and background conditions. It confirms the reliability of marker identification as a localization input for autonomous UAV navigation in GPS-denied environments.

The Aruco-Marker detection test only tested the web-integrated camera's readings at specific heights to determine the maximum computer vision capability for processing image data captured by the camera. Based on the test results, the system was already very difficult to validate at a height of 2.1 meters because the markers detected were too small, rendering the computer vision algorithm unable to run. The test result was lower than other research, which detected a 40x40 cm marker from a height of 5 - 14 m [15].

3.4 Autonomous Navigation System Test

Evaluating the experimental outcomes, it is critical to clarify the framework and criteria used to assess system performance. In this study, a 'trial' represents a single complete flight path execution. At the same time 'samples' refer to the documented discrete movements within those paths. To maintain scientific consistency, a strict binary classification was applied. A trial

is marked as a Success only if the UAV achieves full autonomous take-off, seamless node-to-node ArUco-marker validation without intervention, and a safe landing.

Table 2 presents the comprehensive results of the first trial phase for the autonomous quadcopter navigation system using ArUco markers as spatial anchors. The trial focused exclusively on straight-axis navigation along the Y=0 coordinate to evaluate the system's baseline effectiveness. A constant altitude of 1 meter was maintained throughout all samples to ensure sensor consistency. Most samples successfully navigated and landed at the designated ArUco marker IDs. Navigation attempts (at IDs 8) were classified as "Failed" because the UAV reached the setpoint but failed to stop or maintain controlled hovering. These failures were attributed to a synchronization lag, where the drone's velocity exceeded the camera's processing frame rate, preventing the system from identifying the target marker in time to execute the braking command. The following analysis details the performance and constraints observed during this phase:

Table 2. First Trial Phase – Straight Axis Navigation

Trials	Node Point / ID (Samples)	Navigation Result	Remarks
1	A – B (0 – 1)	Success	UAV reached setpoint and landed successfully at the target ID.
2	A – B (0 – 2)		
3	A – B (0 – 3)		
4	A – B (0 – 4)		
5	A – B (0 – 5)		
6	A – B (0 – 6)		
7	A – B (0 – 7)		
8	A – B (0 – 8)	Failed	Reached setpoint but became uncontrollable.

Based on the test results, the system's performance was quantified using the predefined binary classification framework, referred to as **Eq. 1**. Therefore, the SR value from the first trial phase was calculated as 87.5%.

$$\text{Success Rate (SR)} = \frac{Success\ Trials}{Total\ Trials}\ x\ 100\% \quad (1)$$

Comprehensive system testing at **Table 3** showed that the trajectory used a straight-line path combined with a rightward lateral shift. This configuration was designed to evaluate forward, backward, and lateral movements. The Z-coordinate setpoint was maintained at a constant 1 meter. To facilitate navigation to the right, a negative value (-) was assigned to the Y-axis setpoint.

Table 3. Second Trial Phase – Straight – Lateral Right Axis Navigation

Trials	Node Point / ID (Samples)	Navigation Result	Remarks
1	A – B – E – F (0 – 3 -15)	Success	UAV reached setpoint ID 3 and 15, and successfully landed at ID 15.
2		Success	
3		Success	
4	A – B – E – F (0 – 3 – 15 – 16)	Failed	Drones are losing control while capturing marker ID 15.

Trials	Node Point / ID (Samples)	Navigation Result	Remarks
5		Success	UAV reached setpoint IDs 3, 15, and 16 and successfully landed at ID 16.
6		Success	
7	A – B – E - F (0 - 3 – 15 – 17)	Failed	The UAV landed before reaching the designated ID 17 position.
8		Success	UAV reached setpoint IDs 3, 15, and 17 and successfully landed at ID 17.
9		Success	
10	A – B – E – F (0 – 8 – 20)	Failed	The system failed to validate ID 8, causing the drone to lose control but successfully land at ID 20.
11		Success	UAV reached setpoint ID 8 and 20, and successfully landed at ID 20.
12		Success	
13	A – B – E – F (0 – 8 – 20 – 15)	Failed	A system failure occurred during the initial take-off phase.
14		Success	UAV reached setpoint ID 8, 20, 15, and successfully landed at ID 15.
15		Success	
16	A – B – E – F (0 – 8 – 20 – 16)	Failed	A system failure occurred during the initial take-off phase.
17		Success	UAV reached setpoint ID 8, 20, 16, and successfully landed at ID 16.
18		Success	
19	A – B – E – F (0 – 8 – 20 – 17)	Failed	The initial take-off position was shifted too far to the right, causing the camera to miss the ID 0 marker and resulting in a navigation failure.
20		Failed	The initial take-off was uncontrollable.
21		Success	UAV reached setpoint ID 8, 20, 17, and successfully landed at ID 17.
22	A – B – E – F (0 – 8 – 20 – 18)	Failed	The initial take-off was uncontrollable.
23		Success	UAV reached setpoint ID 8, 20, 18, and successfully landed at ID 18.
24		Success	
25	A – B – E – F (0 – 8 – 20 – 19)	Failed	The initial take-off was uncontrollable.
26		Failed	Landing not on ID 19.
27		Failed	UAV can't reach setpoint ID 20.

Based on the test results, the system's performance was quantified using the predefined binary classification framework, referred to as **Eq. 1**. Therefore, the SR value from the second trial phase, with 27 trials, was 63%.

Comprehensive system testing at **Table 4** showed that the trajectory followed a straight-axis path, combined with a leftward lateral shift. This configuration was specifically designed to evaluate the UAV's forward, backward, and lateral movement capabilities. The altitude setpoint (Z-coordinate) was maintained at a constant 1 meter. To facilitate navigation to the left, a positif (+) value was assigned to the Y-axis setpoint.

Table 4. Third Trial Phase – Straight – Lateral Left Axis Navigation

Trials	Node Point / ID	Navigation Result	Remarks
1	A – B – C – D (0 – 3 – 9)	Success	UAV reached setpoint ID 3 and 9, and successfully landed at ID 9.
2			
3			
4	A – B – C – D (0 – 3 – 9 – 10)	Success	UAV reached setpoint IDs 3, 9, and 10 and successfully landed at ID 10.
5			
6			

Trials	Node Point / ID	Navigation Result	Remarks
7	A – B – C – D (0 – 3 – 9 – 11)	Failed	A system failure occurred, and the UAV landed at ID 9 instead of the designated target.
8		Success	UAV reached setpoint IDs 3, 9, and 11 and successfully landed at ID 11.
9			
10	A – B – C – D (0 – 3 – 9 – 12)	Success	UAV reached setpoint IDs 3, 9, and 12 and successfully landed at ID 12.
11			
12			
13	A – B – C – D (0 – 3 – 9 – 13)	Failed	A system failure occurred, and the UAV landed at ID 9 instead of the designated target.
14		Success	UAV reached setpoint IDs 3, 9, and 13 and successfully landed at ID 13.
15			
16	A – B – C – D 0 – 3 – 9 – 14	Failed	The UAV landed prematurely before reaching the designated ID 14 position.
17		Success	UAV reached setpoint IDs 3, 9, and 14 and successfully landed at ID 14.
18			

Based on the test results, the system's performance was quantified using the predefined binary classification framework, referred to as Eq. 1. Therefore, the SR value from the third trial phase, with 15 successes out of 18 trials, was 83%.

The final trial, the fourth phase of testing, involved complex navigation sequences that combined elements from the first three trial stages. This phase represented the longest flight trajectories conducted to evaluate the drone's endurance and performance during extended autonomous missions. The experimental results in **Table 5** confirm that the integrated system can operate effectively under these demanding conditions. In the majority of trials, the UAV did not land precisely on the center of the designated ArUco marker IDs. This outcome is considered acceptable within the scope of this research, as the current system architecture does not include a specific precision-landing algorithm. Certain "Failed" attempts were characterized by unvalidated Marker IDs, which occurred when the drone's horizontal velocity exceeded the camera's image processing frame rate. This lag in visual processing prevented the system from identifying the target markers in time to execute navigation commands. To mitigate these validation errors, system parameters were tuned using the QGroundControl application. Optimization was achieved by reducing the MPC_POS_XY parameter to 0.20 and adjusting MPC_VEL_VERTICAL to 4.60. These adjustments successfully constrained the drone's speed, enabling more reliable real-time marker detection and stabilization.

Table 5. Fourth Trial Phase – Compound Multi-Direction

Trials	Node Point / ID	Result	Remarks
1	A-B-D-C-D-B-F-E-F-B-A	Failed	Vertical ascent of 0.5 meters while navigating to Point B and loss of control during the transition to Point C. The UAV failed to land at the designated back-to-ID-A point.
2		Success	UAV reached all setpoints during multi-directional simulation recognition in an urban terrain environment.
3	A-B-F-E-F-B-A	Failed	During the transition toward Point F, the UAV ascended by approximately 0.2 meters, and the landing position did not coincide with Marker ID 0. Marker ID 8 could not be detected or validated by the vision system.
4		Success	UAV reached all setpoints during multi-directional simulation recognition in an urban terrain environment.

Trials	Node Point / ID	Result	Remarks
5	A-B-D-C-D-B-A	Failed	The marker ID was not successfully validated due to excessive UAV motion speed. The UAV did not land on Marker ID 0.
6		Success	The UAV reached all setpoints during multi-directional simulation recognition in an urban terrain environment.
7	A-B-D-B-F-B-A	Failed	During the transition from Point B to Point F, the UAV ascended by approximately 0.2 m, and the landing position did not coincide with Marker ID 0.
8		Success	The UAV reached all setpoints during multi-directional simulation recognition in an urban terrain environment.

Based on the test results, the system's performance was quantified using the predefined binary classification framework, referred to as Eq. 1. Therefore, the SR value from the fourth trial phase, with 8 trials, was 50%.

Table 6 summarizes the overall performance evaluation of the ArUco marker–based autonomous UAV navigation system across multiple predefined path configurations. The table is structured to compare different maneuvering scenarios, including straight-lane, lateral movements, and compound multi-directional routes, in order to assess system reliability under varying navigation complexities. Each configuration is evaluated based on the number of test or trials and the corresponding navigation success rate, providing a standardized framework for performance comparison. Based on the test results, the system's performance was quantified using the predefined binary classification framework, referred to as Eq. 1. The OSR value from the fourth trial phase, with 61 trials, was 70% successful.

Table 6. Comparison of Overall Performance and

Path Configuration	Number of Trials	Number of Successful Trials	Success Rate (%)
Straight-Axis	8	7	87.5
Lateral-Right	27	17	63
Lateral-Left	18	15	83
Compound multi-direction	8	4	50
Overall Success Rate			70.9

The results of the overall performance evaluation of the ArUco marker-based autonomous UAV navigation system achieved an overall success rate of 70%. This result was lower than that of studies using an object detection algorithm, which achieved a 90% success rate in tracking the yellow sphere [14].

4. Conclusion

This study demonstrated the feasibility of autonomous UAV navigation in GPS-denied indoor environments using ArUco marker-based visual localization integrated with a VL53L1X LiDAR sensor and PX4 Offboard control. The developed system successfully validated markers, performed real-time pose estimation, and navigated sequentially through waypoints without human intervention. The web-based monitoring interface and QGroundControl integration operated reliably throughout all trials, enabling effective dual-platform telemetry monitoring and manual setpoint adjustment from a safe standoff position. The ArUco-marker-based detection, implemented using the OpenCV DICT_5×5_250 dictionary, validated marker identities within a functional altitude range of 40 to 200 cm. Third, across 61 trials, discrete movement samples spanning four path configurations—straight-axis, lateral-right, lateral-left,

and compound multi-direction—the system achieved an overall navigation success rate of 70%. Navigation failures caused by synchronization lag between UAV translational velocity and the camera's image processing frame rate, which prevented timely marker validation during high-speed maneuvers. Tuning the MPC_POS_XY and MPC_VEL_VERTICAL parameters via QGroundControl mitigated this issue in subsequent trials. These results establish ArUco marker-guided navigation as a practical, low-infrastructure baseline for initial indoor reconnaissance in military urban terrain operations. Future work should focus on integrating a dedicated precision landing algorithm, exploring adaptive marker sizing to extend detection range, and evaluating system robustness under varying indoor lighting conditions and dynamic obstacle environments.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Y. Pengyu, H. Yan, Z. Christopher, C. Wei, and S. Mao. (2025). "UAV autonomous navigation system based on air–ground collaboration in GPS-denied environments," Drones, vol. 9, no. 7, Art. no. 442, pp. 1–20. https://doi.org/10.3390/drones9060442.

[2] S. Petr, D. Jan, S. Karel, H. Jan, and P. Dalibor. (2020). "Collective perception using UAVs: Autonomous aerial reconnaissance in a complex urban environment," Sensors, vol. 20, no. 10, Art. no. 2926. https://doi.org/10.3390/s20102926

[3] W. P. H. B. Arie, M. Ronny, and N. I. Astria. (2024). "Drone position tracking system based on object detection and ArUco marker for autonomous navigation applications," in Proc. Int. Seminar on Intelligent Technology and Its Applications (ISITIA). https://doi.org/10.1109/ISITIA63062.2024.10668046

[4] S. Petr, D. Jan, M. Jan, H. Jan, and P. Dalibor. (2019). "Cooperative unmanned aerial system reconnaissance in a complex urban environment and uneven terrain," Sensors, vol. 19, no. 17, Art. no. 3754. https://doi.org/10.3390/s19173754

[5] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. (2014). "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 47, no. 6, pp. 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005

[6] B. Xu, Q. Zhang, F. Pan, and X. Feng. (2018). "Marker-based multi-sensor fusion indoor localization system for micro air vehicles," Sensors, vol. 18, no. 6, Art. no. 1706. https://doi.org/10.3390/s18061706

[7] Q. Quan. (2017). Introduction to Multicopter Design and Control. Singapore: Springer. https://doi.org/10.1007/978-981-10-3382-7

[8] A.-Z. Mohammed, M. A. Sabah, D. Ahmed, and R. R. Gharieb. (2014). "Edge detection with a preprocessing approach," Journal of Signal and Information Processing, vol. 5, no. 4, pp. 123–134. http://dx.doi.org/10.4236/jsip.2014.54015

[9] H. Lee, J. Yoo, M.-S. Jung, and K.-J. Park. (2021). "A robot operating system framework for secure UAV communications," Sensors, vol. 21, no. 4, Art. no. 1369. https://doi.org/10.3390/s21041369

[10] F. Kocer, M. Hossain, and W. Hardt. (2024). "Prototype for multi-UAV monitoring–control system using WebRTC," Drones, vol. 8, no. 10, Art. no. 551. https://doi.org/10.3390/drones8100551

[11] N. Bianchi, H. Garcia, J. Iglesias, and J. Ruiz. (2023). "Precision landing of a quadcopter drone by smartphone video guidance sensor in a GPS-denied environment," Sensors, vol. 23, no. 4, Art. no. 1934. https://doi.org/10.3390/s23041934

[12] T. Hakim, A. S. Priambodo. (2025). "Navigasi quadcopter berbasis ArUco marker dengan OpenCV," Jurnal Listrik Instrumentasi dan Elektronika Terapan, vol. 6, no. 1, pp. 28–39. https://doi.org/10.22146/juliet.v6i1.97135

[13] C. Ramirez-Atencia and D. Camacho. (2018). Extending QGroundControl for automated mission planning of UAVs," Sensors, vol. 18, no. 7, Art. no. 2339. https://doi.org/10.3390/s18072339

[14] O. Gharsa, Touba, M. M., Boumehraz, M., & Agram, N. (2025). Autonomous Vision-Based Object Detection and Tracking System for Quadrotor Unmanned Aerial Vehicles. *Sensors*, *25*(20), 6403. https://doi.org/10.3390/s25206403.

[15] A. S. Setyawan, R. Hendy, B. Ahsan. (2025). Implementation of Waypoint Navigation and Computer Vision for Monitoring Markers on a Quadcopter Based on ROS,". *International Journal of Marine Engineering Innovation and Research.* 10(1) pp. 40-46. http://dx.doi.org/10.12962%2Fj25481479.v10i1.22098