SWARM DRONE

Tugas Akhir

Oleh:
Sarah Anggraini (4222111007)

Program Studi Teknologi Rekayasa Robotika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2026

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam

Oleh:
Sarah Anggraini (4222111007)

Dengan judul:
Implementation of Real-Time Swarm Drone Formation Using Firebase and MIT App Inventor with Interpolation-Based in Gazebo

Tanggal Sidang: 15 Januari 2026

Disetujui oleh :

Dosen Penguji I	**Dosen Pembimbing I**
1. Rifqi Amalya Fatekha, S.ST, M.Tr.T NIK: 199007172021211001	1. Ryan Satria Wijaya S.Tr.T., M.Tr.T. NIK: 199706112025061009

Dosen Penguji II

2. Naurah Nazhifah, S.Kom., M.C.S.(AI)
NIK: 199909132025062010

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Implementation of Real-Time Swarm Drone Formation Using Firebase and MIT App Inventor with Interpolation-Based Control in Gazebo" adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan**, dan **bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 26 Januari 2026

SEPULUH RIBU RUPIAH
10000
METERAI
TEMPEL
387B5ANX264276059

Sarah Anggraini
NIM: 4222111007

Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Implementation of Real-Time Swarm Drone Formation Using Firebase and MIT App Inventor with Interpolation-Based Control in Gazebo

Ryan Satria Wijaya[1]*, Hendawan Soebhakti[2]*, Rifqi Amalya Fatekha[3]*, Sarah Anggraini[4]*
* Teknik Robotika, Politeknik Negeri Batam
ryan@polibatam.ac.id [1], hendawan@polibatam.ac.id [2], rifqi@polibatam.ac.id [3], anggrainis1723@gmail.com[4]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

swarm drone, formation control, Firebase, MIT App Inventor, ROS, Gazebo, interpolation.

ABSTRACT

Swarm drone systems require reliable real-time control to ensure smooth coordination, stable formation transitions, and synchronized motion among multiple unmanned aerial vehicles. However, many existing swarm control studies focus on static formations or predefined trajectories, with limited attention to smooth formation transitions and real-time performance, particularly for low-cost and educational platforms. This study presents the implementation of a real-time swarm drone formation control system using Firebase Realtime Database and an Android-based control interface developed with MIT App Inventor, integrated with the Robot Operating System (ROS) and the Gazebo simulation environment. To improve motion smoothness during formation changes, an interpolation-based motion control approach is applied to generate intermediate waypoints between target positions. The proposed system is evaluated through simulation using five UAVs performing line, triangle, circle, star, and dynamic formation sequences. Simulation results indicate a real-time response delay of approximately 0.4–0.6 seconds and stable hover performance within ±0.05 m at an altitude of 3.5 meters during formation transitions. These results demonstrate that the interpolation-based approach effectively enhances formation smoothness and synchronization in a cloud-integrated swarm drone system, providing a practical and low-cost reference for educational and early-stage swarm drone research.

I. INTRODUCTION

The development of swarm drone systems has gained significant attention due to their potential applications in surveillance, environmental monitoring, and collaborative robotic research[1], [2], [3]. Swarm drones rely on coordinated motion among multiple agents, where each drone must be able to maintain formation, avoid collisions, and respond to control commands in real time[4], [5]. One of the main challenges in swarm control lies in achieving smooth formation transitions while maintaining system stability and synchronization among drones[6], [7], [8].

Several previous studies have explored swarm drone control using centralized or decentralized architectures, as well as mobile-based interfaces and cloud communication platforms[9], [10], [11], [12]. However, many existing works primarily focus on static formations or predefined paths, with limited attention given to smooth transitions between formations[6], [13], [14]. In addition, real-time performance issues such as response delay, synchronization accuracy, and hover stability are often not analyzed in detail[4], [15]. These limitations become more critical when swarm control is implemented using low-cost platforms intended for educational or experimental purposes.

Cloud-based control approaches offer practical advantages for swarm drone research, particularly in terms of accessibility, scalability, and cost efficiency[11], [16]. By utilizing cloud services such as Firebase Realtime Database, control commands can be transmitted with minimal infrastructure requirements while enabling real-time interaction between user interfaces and robotic systems[11], [12]. This approach is especially suitable for educational environments, where researchers and students require flexible and affordable solutions without relying on expensive ground control stations or proprietary systems.

In this research, Firebase Realtime Database is integrated with an Android-based control interface developed using MIT App Inventor. This combination allows users to send formation commands intuitively through a mobile application, which are then processed by a ROS-based control node. Although similar combinations of mobile interfaces, cloud communication, and ROS-based control have been explored in previous studies, most of these works primarily focus on basic command execution or static formation control. In contrast, this research emphasizes the implementation of interpolation-based motion control to explicitly address smooth formation transitions and motion stability during dynamic formation changes, which are often insufficiently discussed in existing studies.

The Gazebo simulator is used as the experimental environment in this study due to its ability to provide a safe, repeatable, and controlled testing platform for multi-drone systems[17], [18]. Simulation-based testing allows the evaluation of swarm behavior without the risks associated with real-world flight, such as hardware damage or safety concerns. Moreover, Gazebo enables detailed observation of formation transitions, response delays, and hover stability under consistent conditions, making it suitable for validating swarm control algorithms[19], [20]. In addition, the use of Gazebo allows controlled evaluation of formation transition behavior and system responsiveness under consistent conditions, making it suitable for early-stage validation of swarm control approaches.

This study evaluates five swarm formations, namely line, triangle, circle, star, and a dynamic formation sequence. The dynamic formation consists of a continuous transition through multiple sub-formations, designed to demonstrate the effectiveness of interpolation-based control in handling complex motion patterns. Interpolation is applied to generate intermediate waypoints between formations, ensuring gradual position changes and preventing abrupt movements that could lead to instability or collisions[6], [21], [22], [23].

Previous studies published in the Journal of Applied Informatics and Computing (JAIC) have demonstrated practical UAV control implementations, including gimbal stabilization systems and gesture-based interaction mechanisms for multi-UAV applications[24], [25].

Despite the growing use of cloud-based interfaces and mobile applications in swarm drone research, there remains a lack of studies that explicitly focus on smooth and stable formation transitions under real-time control, particularly in low-cost and educational platforms. Most existing works do not systematically evaluate interpolation-based motion control during dynamic formation changes, nor do they provide detailed analysis of response delay, hover stability, and transition consistency in multi-drone scenarios. This research addresses these gaps by focusing on interpolation-driven formation transitions within a cloud-integrated swarm control framework, validated through systematic simulation-based testing.

The main contribution of this research lies in the implementation and evaluation of a cloud-integrated swarm drone control framework combined with interpolation-based motion control in a simulation environment. The proposed system demonstrates smooth formation transitions, stable hover performance, and consistent response behavior during real-time operation. By focusing on motion smoothness, system simplicity, and real-time responsiveness, this work provides a practical and low-cost reference for educational and early-stage swarm drone research using open-source tools. Based on this motivation, the proposed system architecture, control logic, interpolation method, and evaluation scenarios are described in the following methodology section.

II. METHOD

This research employs a simulation-based methodology to design and evaluate a real-time swarm drone formation control system[17], [18]. The system integrates an Android-based interface developed using MIT App Inventor, Firebase Realtime Database as a cloud communication layer, and a Python-based ROS control node within the Gazebo simulation environment[19], [20], [26]. This approach is designed to support low-latency command transmission, smooth formation transitions, and stable multi-drone coordination during simulation-based evaluation.

A. *System Architecture*

The proposed system architecture is designed to enable real-time swarm drone formation control using a cloud-based communication framework. The overall system consists of three main components: an Android-based control interface developed using MIT App Inventor, Firebase Realtime Database as the cloud communication layer, and a ROS-based swarm control system running in the Gazebo simulation environment. The overall system architecture is illustrated in Figure 1.

The MIT App Inventor application serves as the user interface, allowing users to send formation commands intuitively through predefined buttons. Each command is transmitted to Firebase Realtime Database, which acts as a lightweight cloud mediator between the mobile application and the robotic system. Firebase is selected due to its real-time data synchronization capability, low configuration complexity, and suitability for educational and experimental applications.

On the robotic side, a ROS node continuously monitors changes in the Firebase database and translates incoming commands into swarm control instructions. These instructions are processed using interpolation-based motion control, where target positions for each drone are generated gradually to ensure smooth transitions between formations. This approach helps maintain synchronization among drones while preventing abrupt movements that could lead to instability or collisions.

On the robotic side, a ROS node continuously monitors changes in the Firebase database and translates incoming commands into swarm control instructions. These instructions are processed using interpolation-based motion control, where target positions for each drone are generated gradually to ensure smooth transitions between formations. This approach helps maintain synchronization among drones while preventing abrupt movements that could lead to instability or collisions.

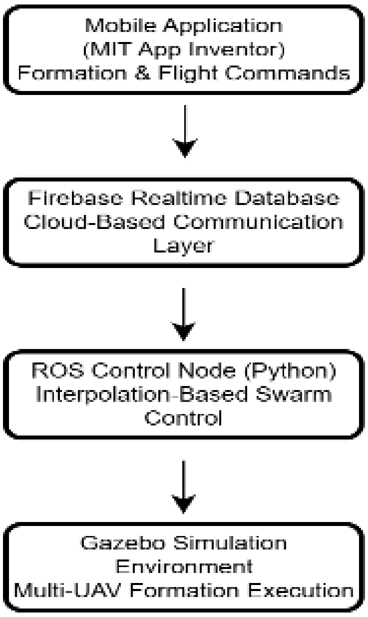


Figure 1. System architecture of the proposed real-time swarm drone formation control using MIT App Inventor, Firebase Realtime Database, ROS-based interpolation control, and Gazebo simulation environment.

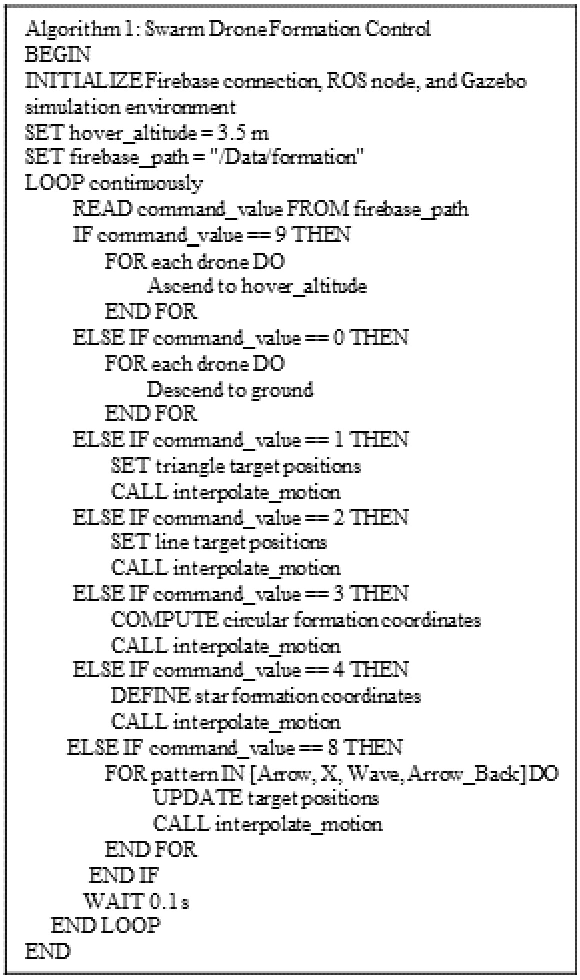
B. *System Control Logic (Pseudocode Representation)*

The system control logic is designed to manage real-time swarm drone behavior based on user commands issued through a mobile interface. Control commands are generated using an Android application developed with MIT App Inventor and transmitted to the system via Firebase Realtime Database. Firebase functions as a cloud-based communication layer that enables low-latency data synchronization between the user interface and the ROS-based control system.

Each command received from Firebase is continuously monitored by a ROS control node. The node interprets the command value and determines the corresponding swarm action, such as takeoff, landing, or formation change. This event-driven mechanism ensures that control actions are executed only when a command update is detected, thereby reducing unnecessary computation and maintaining system responsiveness.

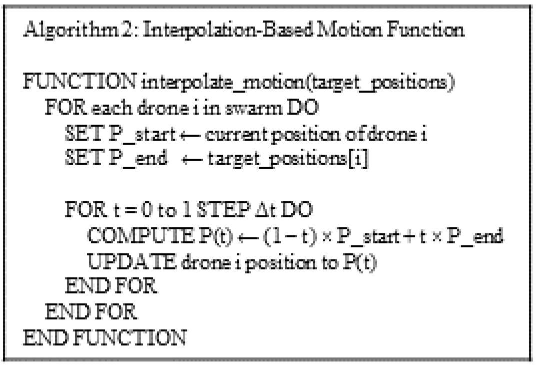
The control logic separates high-level decision-making from low-level motion execution. High-level commands define the desired swarm state, including the selected formation pattern, while the detailed movement of each individual drone is handled through interpolation-based position updates. This modular separation allows the system to maintain stability, synchronization, and smooth transitions during dynamic formation changes.

To ensure safe operation, the control logic enforces a fixed hover altitude of 3.5 meters during all formation maneuvers. Takeoff and landing commands are treated as special cases due to their significant vertical displacement, requiring gradual thrust adjustment and stabilization before and after horizontal movement. This design minimizes abrupt motion and reduces the risk of instability or inter-drone collision.



```
Algorithm 1: Swarm DroneFormation Control
BEGIN
INITIALIZE Firebase connection, ROS node, and Gazebo
simulation environment
SET hover_altitude = 3.5 m
SET firebase_path = "/Data/formation"
LOOP continuously
    READ command_value FROM firebase_path
    IF command_value == 9 THEN
        FOR each drone DO
            Ascend to hover_altitude
        END FOR
    ELSE IF command_value == 0 THEN
        FOR each drone DO
            Descend to ground
        END FOR
    ELSE IF command_value == 1 THEN
        SET triangle target positions
        CALL interpolate_motion
    ELSE IF command_value == 2 THEN
        SET line target positions
        CALL interpolate_motion
    ELSE IF command_value == 3 THEN
        COMPUTE circular formation coordinates
        CALL interpolate_motion
    ELSE IF command_value == 4 THEN
        DEFINE star formation coordinates
        CALL interpolate_motion
    ELSE IF command_value == 8 THEN
        FOR pattern IN [Arrow, X, Wave, Arrow_Back] DO
            UPDATE target positions
            CALL interpolate_motion
        END FOR
    END IF
    WAIT 0.1 s
END LOOP
END
```

Figure 2. Pseudocode of the swarm drone formation control algorithm with interpolation-based motion.



```
Algorithm 2: Interpolation-Based Motion Function

FUNCTION interpolate_motion(target_positions)
  FOR each drone i in swarm DO
    SET P_start ← current position of drone i
    SET P_end  ← target_positions[i]

    FOR t = 0 to 1 STEP Δt DO
      COMPUTE P(t) ← (1 − t) × P_start + t × P_end
      UPDATE drone i position to P(t)
    END FOR
  END FOR
END FUNCTION
```

Figure 3. Pseudocode of the interpolation-based motion function.

The overall control flow of the system is summarized in the pseudocode shown in Fig. 2, which illustrates how Firebase commands are processed in real time, how formation targets are selected, and how interpolated motion updates are applied to the swarm. Furthermore, Fig. 3 presents the interpolation-based motion function used to compute smooth position transitions for each drone during formation changes. Together, these pseudocode representations provide a clear and concise description of the event-driven control mechanism and the motion generation strategy implemented in this study.

C. Drone Formation and Interpolation Control

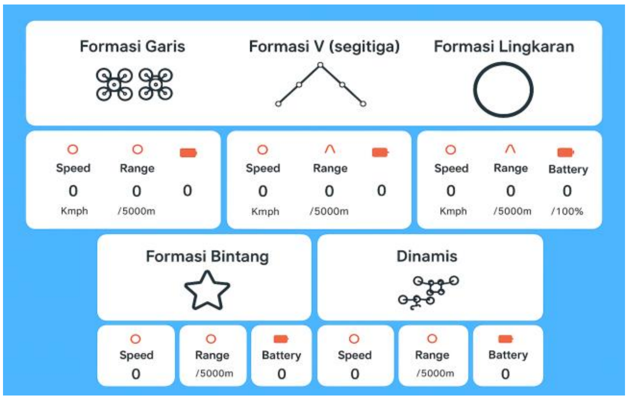


Figure 4. MIT App Inventor interface for swarm drone formation control.

The Android application developed using MIT App Inventor provides an intuitive interface for controlling swarm drone formations within the simulation environment. The application functions as a mobile-based remote control, where each button corresponds to a specific command transmitted to the Firebase Realtime Database. The command values stored in Firebase are continuously monitored by a ROS-based control node, which interprets the received data and executes the corresponding swarm behavior in the Gazebo simulation environment.

The mobile interface includes control buttons for takeoff, landing, and formation selection, as illustrated in Figure 4. Through this interface, users can initiate different swarm formation commands without direct interaction with the simulation software, enabling real-time control through a cloud-based communication mechanism.

Once a formation command is received, the selected formation is executed in the Gazebo simulation environment. The system supports several formation types, including line, triangle, circle, star, and a dynamic formation sequence. In the line formation, drones are positioned horizontally with equal spacing along the X-axis to form a straight alignment. The triangle formation arranges the drones into a V-shaped structure, while the circle formation distributes drones evenly along a circular trajectory generated using polar coordinates. The star formation creates a symmetric configuration combining horizontal and diagonal spacing.

The dynamic formation is executed as a sequence of multiple sub-formations, allowing the swarm to demonstrate continuous motion while remaining airborne. This scenario is designed to evaluate the system’s capability to handle formation changes in real time within a simulated environment.To ensure smooth and stable transitions between formations, an interpolation-based motion control method is applied. Instead of commanding drones to move directly to their target positions, intermediate positions are generated using linear interpolation, as expressed by:

$$P(t) = (1 - t) \cdot P_{start} + t \cdot P_{end} \qquad (1)$$

where P(t) represents the interpolated position at time t, P_start denotes the initial position, and P_end denotes the target position. This interpolation scheme enables gradual position updates, thereby reducing abrupt motion and supporting stable hover behavior during formation transitions.

III. RESULT AND DISCUSSIONS

This section presents the simulation results and analysis of the swarm drone formation control system implemented using Firebase, MIT App Inventor, and ROS in the Gazebo simulator. Several formation scenarios were tested to evaluate responsiveness, accuracy, and stability, with a specific focus on the performance of interpolation-based control.

The interpolation method introduced in section II-E was experimentally validated in the simulation phase. The evaluation focused on four key aspects: transitions smoothness, timing consistency, positional accuracy, and hover stability. These metrics directly measure the effectiveness of interpolation in ensuring smooth formation changes, preventing collisions, and maintaining synchronized movements in real-time.

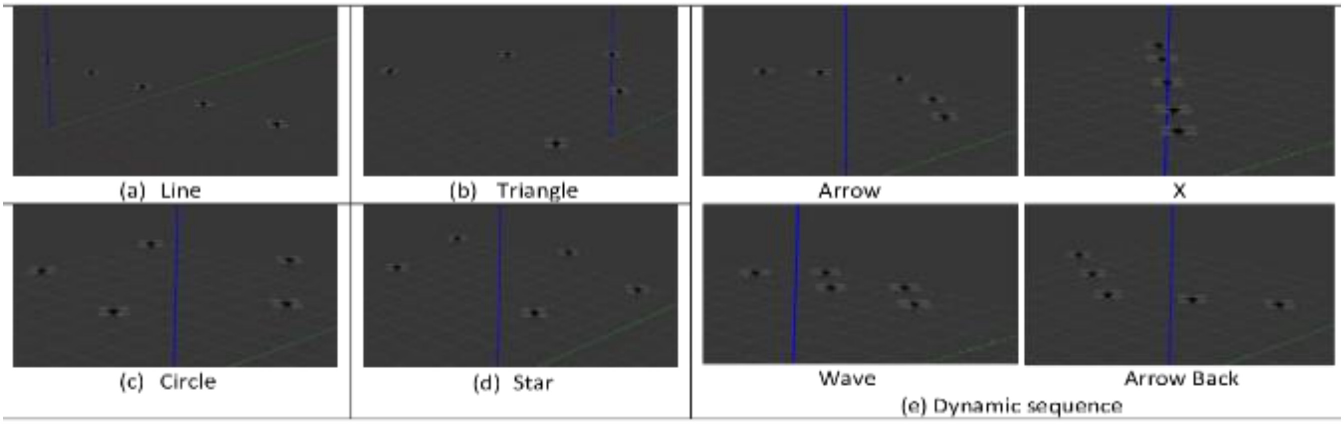


Figure. 5. Formation scenarios of swarm drones in Gazebo: (a) Line, (b) Triangle, (c) Circle, (d) Star, and (e) Dynamic sequence.

A. *Scenario Testing*

In this scenario testing, formation commands were sent through the MIT App Inventor interface, relayed to Firebase, and executed by the ROS node in real-time. Figure 4 illustrates all five formation patterns tested in the Gazebo simulator. In the Line formation (Fig. 5a), the five drones align horizontally at equal spacing along the X-axis at an altitude of 3.5 meters. The Triangle formation (Fig. 5b) arranges the drones in a V-shape, with three drones forming the base and two drones at the apex. For the Circle formation (Fig. 5c), the drones are positioned evenly along a circular path generated through polar coordinate calculation. In the Star formation (Fig. 5d), drones form a symmetric five-point pattern combining horizontal and diagonal alignment. Finally, the Dynamic formation (Fig. 5e) demonstrates a sequential transition through four sub-patterns—Arrow, X, Wave, and Inverted Arrow—executed continuously to showcase smooth and stable interpolation control.

The results from all formations confirm that interpolation-based motion control successfully ensures smooth movement, prevents abrupt position changes, and maintains a stable hover altitude throughout the simulation.

This scenario-based evaluation is conducted entirely within the Gazebo simulation environment to provide a controlled and repeatable setting for analyzing swarm formation behavior without hardware-related risks.

A. *Real-Time Response*

The delay between pressing a command on the app and drone movement in Gazebo averaged 0.4-0.6 seconds. This range is acceptable for real-time control, enabling smooth transitions. The delay from writing a command in Firebase to execution in ROS was consistently around 1 second, showing efficient integration. This response delay range is considered acceptable for real-time swarm control in simulation-based educational and experimental applications, where command-level responsiveness is prioritized over low-level flight dynamics.

TABLE I
AVERAGE RESPONSE DELAY: 0.4-0.6 S, SUFFICIENT FOR SMOOTH FORMATION CONTROL.

Command	Formation	Avg. Delay (s)
1	Triangle	0.45
2	Line	0.42
3	Circle	0.47
4	Star	0.43
8	Dynamic (4 steps)	0.58

B. *Transition Duration Analysis*

Most formation transitions—such as Line, Triangle, Circle, Star, and Dynamic—were completed within a range of 4.87 to 4.99 seconds. These relatively consistent durations indicate that the interpolation-based motion control ensures uniform timing across different patterns. The smooth transition between formations minimizes abrupt movements and maintains formation stability throughout the simulation.

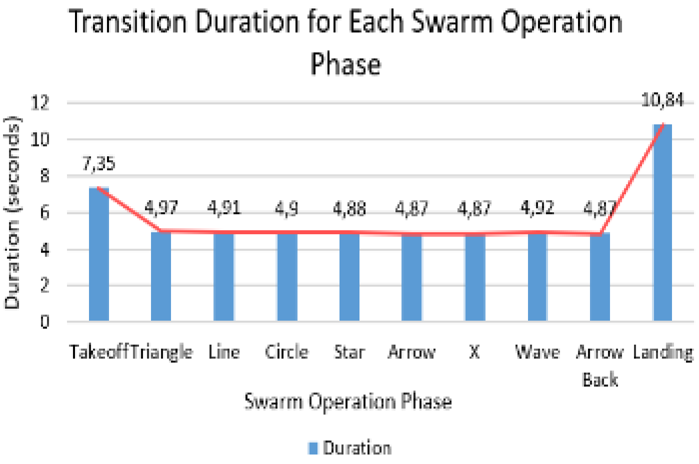


Figure 6. Duration of takeoff, landing, and formation transitions.

The takeoff and landing phases recorded the highest transition durations of 7.35 seconds and 10.84 seconds, respectively. This occurs because both phases involve significant vertical displacement and require stabilization before or after horizontal movements. During takeoff, each drone gradually increases thrust to reach the hover altitude of 3.5 meters while maintaining balance, which extends the transition time. Similarly, during landing, drones descend slowly to avoid overshoot and ensure synchronized touchdown, resulting in a longer completion time compared to horizontal formation transitions. The consistency of these transition durations across different formations indicates that the interpolation-based control successfully synchronizes multi-drone motion, independent of formation geometry.

These consistent transition times further confirm that the interpolation method effectively produces predictable and stable motion profiles for all tested formations.

C. Interpolation Method Performance

To evaluate the interpolation performance, several aspects were analyzed, including smoothness, consistency, accuracy, and stability. The results showed that all drones transitioned between formations without abrupt stops or jerks, confirming that the interpolation method effectively generated intermediate waypoints and ensured smooth movement. The transition durations were highly consistent, with variations of less than ±0.12 seconds, which indicates that the timing logic was stable throughout the tests. Furthermore, the drones successfully reached their final positions with minimal error of approximately ±0.01 m, even during the more complex dynamic sequence transitions. In terms of stability, the drones maintained a hover altitude of 3.5 meters with fluctuations of less than ±0.05 m, demonstrating reliable performance during both static and dynamic maneuvers.

This confirms that the interpolation method function effectively in all tested scenarios, ensuring reliable multi-drone motion control. Unlike direct position command approaches commonly reported in previous studies, the interpolation method applied in this work ensures gradual and predictable motion during both static and dynamic formation transitions

D. System Stability

During all formations and transitions, drones maintained a hover altitude of 3.5 meters with minimal oscillation. Interpolation minimized sudden movements and overshoot. Minor shaking occurred during rapid transitions, especially from Star to Dynamic, but was resolved by optimal spacing and speed adjustments.

The system successfully managed five drones simultaneously without observable collisions in the simulation environment or instability. The stable hovering performance and minimal oscillations indicate that the interpolation control approach is effective in maintaining coordinated multi-drone stability during rapid formation changes.

E. Interpolation Method Testing

TABLE II
INTERPOLATION METHOD TESTING RESULTS

Transition	**Planned Duration (s)**	**Actual Duration (s)**	**Position Error (m)**
Line → Triangle	4.90	4.88	±0.05
Triangle → Circle	4.90	4.87	±0.05
Circle → Star	4.90	4.89	±0.05
Star → Dynamic (Arrow)	4.90	4.91	±0.05
Dynamic (Arrow → X)	4.90	4.92	±0.05
Dynamic (X → Wave)	4.90	4.93	±0.05
Dynamic (Wave → Arrow Back)	4.90	4.94	±0.05

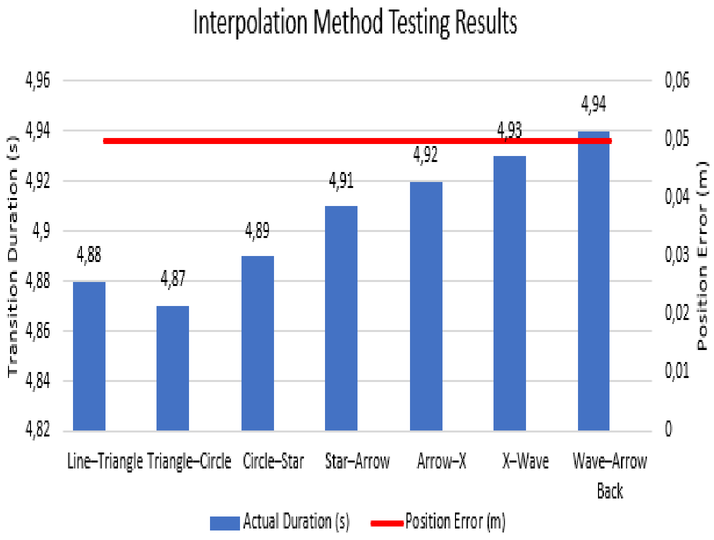


Figure 7. Comparison between planned and actual transition durations using interpolation-based control.

The interpolation method was evaluated by comparing the planned transition duration with the actual measured duration for each formation change, as presented in Table II and Fig. 7. Across all transitions, the planned duration was set to 4.90 seconds, while the actual execution times ranged from 4.87 to 4.94 seconds. The small deviation of less than ±0.03 seconds confirms that the interpolation-based control provides consistent and predictable timing behavior.

In addition, the position error remained within ±0.05 m throughout all transitions, indicating stable and accurate position tracking during motion execution. These results demonstrate that the interpolation logic effectively maintains synchronized swarm motion and stable trajectory execution, which is essential for smooth formation transitions in multi-drone systems.

F. Discussion of Challenges

During the development of this system, several challenges were encountered. One of the issues was the delay in Firebase communication, where occasional delays of up to 0.6 seconds were observed during peak updates. Another challenge involved collision avoidance, as early testing sometimes caused overlaps in dynamic transitions, which were later resolved by adjusting the spacing between drones.

The MIT App Inventor interface also posed limitations, particularly because the dynamic button needed to cycle through multiple patterns using state logic in ROS, which added complexity to the control flow. Achieving smooth transitions was initially difficult as well, but the implementation of interpolation control significantly improved performance. Finally, the system currently lacks a feedback mechanism, meaning that telemetry data for real-time monitoring of drone states is not yet available.Despite these challenges, the system demonstrated stable, responsive formation control. These challenges highlight practical considerations for cloud-based swarm control systems and provide valuable insights for future refinement and real-world deployment.

G. System Evaluation

TABLE III
SYSTEM PERFORMANCE EVALUATION

Evaluation Aspect	**Criteria**	**Observation Result**	**Status**
Response Speed	< 2 s from Firebase to execution	±1 s from Firebase command to drone movement	Pass
Position Accuracy	Tolerance < 0.1 m	Final positions close to target	Pass
Hover Stability	Z fluctuation ±0.05 m	Stable hover at 3.49–3.50 m	Pass
Formation Success	No collisions	All formations executed without collision	Pass
Firebase–ROS Connection	Real-time connection	Data instantly received by ROS node	Pass

System performance was evaluated based on responsiveness, accuracy, stability, and formation success. The system demonstrated reliable behavior, with all formation maneuvers completed successfully without any collision. Hover altitude was maintained within ±0.05 m throughout the simulation, confirming stable vertical control. Overall, these results indicate that the implemented control framework provides high accuracy and real-time responsiveness, ensuring consistent performance during continuous formation transitions. Although this evaluation is limited to five drones in a simulated environment, the results demonstrate the feasibility and robustness of the proposed control framework as a foundation for future scalability studies.

It should be noted that all experiments in this study were conducted exclusively in a simulated environment using Gazebo. As a result, certain real-world factors such as sensor noise, wind disturbances, hardware latency, and network variability are not fully represented. These factors may introduce additional challenges in physical drone deployments. Nevertheless, the simulation-based evaluation provides a safe and controlled environment for validating the proposed control framework, formation behavior, and system responsiveness. The presented results serve as a foundational baseline for future extensions toward real-world implementation and hardware-based experimentation.

IV. CONCLUSION

The interpolation method proved to be the most effective approach for achieving smooth and stable formation transitions in the real-time swarm drone control system. By integrating Firebase as a cloud communication bridge, MIT App Inventor as the user interface, and ROS-Gazebo as the simulation platform, the system successfully demonstrated responsive and reliable formation changes. Five primary formation-Line, Triangle, Circle, Star, and Dynamic sequence-were executed smoothly with interpolation ensuring stable altitude, precise positioning, and collision-free operation even during rapid transitions.

The system testing demonstrated that the average response delay ranged between 0.4 and 0.6 seconds, which is considered acceptable for real-time simulation. The drones were able to maintain hover stability within ±0.05 meters at an altitude of 3.5 meters, ensuring consistent flight performance. Furthermore, all formation switching was executed reliably, with every transition completed smoothly and without any collisions.

The results confirm that interpolation-based motion control is an effective and scalable method for maintaining stability and accuracy in swarm drone simulation. Future work will focus on extending this approach to physical drones by integrating real-time telemetry feedback, implementing Firebase authentication for secure access, and applying PID tuning to enhance motion control precision.

A. Limitation and Future Work

This study has several limitations that should be acknowledged. First, the proposed swarm drone control system is evaluated solely in a Gazebo simulation environment and has not yet been implemented on physical UAV hardware. As a result, real-world factors such as network instability, wind disturbances, sensor noise, and

GPS inaccuracies are not fully captured in the current evaluation. Second, the number of UAVs used in the simulation is limited to five units, which may not fully represent scalability challenges that could arise in larger swarm deployments. Additionally, the cloud-based communication relies on Firebase Realtime Database, and its performance under higher communication loads and increased numbers of UAVs has not been extensively analyzed.

Future work will focus on extending the proposed system toward real-world implementation using physical UAV platforms to evaluate robustness under practical operating conditions. Further studies may also investigate the scalability of the system by increasing the number of UAVs and analyzing the impact on communication latency, synchronization accuracy, and computational load within the ROS framework. Moreover, alternative cloud or edge-based communication architectures could be explored to improve real-time performance and reliability for larger-scale swarm drone applications.

REFERENCES

[1] Q. Ouyang, Z. Wu, Y. Cong, and Z. Wang, “Formation control of unmanned aerial vehicle swarms: A comprehensive review,” *Asian J Control*, vol. 25, no. 6, pp. 2943–2965, 2023.

[2] Y. Zhou, B. Rao, and W. Wang, “UAV swarm intelligence: Recent advances and future trends,” *IEEE Access*, vol. 8, no., pp. 183919–183932, 2020.

[3] M. et al. Abdelkader, “Aerial swarms: Recent applications and challenges,” *IEEE Robot Autom Mag*, vol. 28, no. 2, pp. 82–95, 2021.

[4] S. et al. Javed, “State-of-the-art and future research challenges in UAV swarms,” *IEEE Internet Things J*, vol. 11, no. 3, pp. 4120–4142, 2024.

[5] J. et al. Chen, “Review of unmanned aerial vehicle swarm communication architectures,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2699–2737, 2020.

[6] X. et al. Zheng, “Visually smooth multi-UAV formation transformation,” *Graph Models*, vol. 113, no., p. 101130, 2021.

[7] J. Zhang, J. Yan, and P. Zhang, “Multi-UAV formation control based on a novel back-stepping approach,” *IEEE Trans Veh Technol*, vol. 69, no. 8, pp. 8439–8451, 2020.

[8] J. et al. Guo, “Research on multi-UAV formation and semi-physical simulation with virtual structure,” *IEEE Access*, vol. 11, no., pp. 74291–74305, 2023.

[9] Y. et al. Wang, “Multi-agent bifurcation consensus-based multi-layer UAVs formation keeping control and its visual simulation,” *IEEE Trans Industr Inform*, vol. 19, no. 10, pp. 10145–10156, 2023.

[10] T. et al. Xu, “Distributed MPC for trajectory tracking and formation control of multi-UAVs with leader–follower structure,” *IEEE Access*, vol. 11, no., pp. 35641–35655, 2023.

[11] Various Authors, “CloudStation: A cloud-based ground control station for drones,” *Sensors*, vol. 21, no. 9, p. Article 3114, 2021.

[12] Various Authors, “A UAV-swarm control platform architecture based on cloud computing,” *Applied Sciences*, vol. 12, no. 14, p. Article 7123, 2022.

[13] J. et al. Wu, “Formation control algorithms for multi-UAV systems with unstable topologies and hybrid delays,” *IEEE Trans Netw Sci Eng*, vol. 11, no. 2, pp. 1385–1397, 2024.

[14] Y. et al. Liu, “Flexible multi-UAV formation control via integrating deep reinforcement learning and affine transformations,” *Aerosp Sci Technol*, vol. 147, no., p. 108195, 2025.

[15] Z. et al. Pan, “An improved artificial potential field method for path planning and formation control of multi-UAV systems,” *IEEE Trans Syst Man Cybern Syst*, vol. 51, no. 11, pp. 7098–7109, 2021.

[16] Various Authors, “Self-organizing UAV swarm simulation platform based on cloud computing,” *J Intell Robot Syst*, vol. 109, no. 3, p. Article 64, 2023.

[17] K. et al. Xiao, “XTDrone: A customizable multi-rotor UAVs simulation platform,” *IEEE Robot Autom Lett*, vol. 5, no. 2, pp. 2661–2668, 2020.

[18] Z. et al. Chen, “A survey on open-source simulation platforms for multi-copter UAV swarms,” *Robotics*, vol. 12, no. 3, p. Article 79, 2023.

[19] Y. et al. Li, “Design of UAV swarm simulation platform based on ROS1 and ROS2 hybrid architecture,” *IEEE Access*, vol. 11, no., pp. 91234–91248, 2023.

[20] S. et al. Khaliq, “Multi-platform hardware-in-the-loop simulation for decentralized swarm communication using ROS and Gazebo,” *IEEE Communications Magazine*, vol. 59, no. 8, pp. 70–76, 2021.

[21] B. Siciliano, L. Villani, and A. De Luca, “Trajectory Tracking for UAVs: An Interpolating Control Approach,” *Aerosp Sci Technol*, vol. 117, no., p. 106975, 2021.

[22] A. Alessandri, M. Baglietto, and G. Battistelli, “Interpolating control based trajectory tracking for UAVs,” *IEEE Transactions on Control Systems Technology*, vol. 30, no. 6, pp. 2601–2613, 2022.

[23] H. Zhang, Y. Wang, and L. Liu, “Cubic spline interpolation-based UAV path planning,” *J Intell Robot Syst*, vol. 103, no. 4, p. Article 87, 2021.

[24] JAIC Authors, “Design and Implementation of UAV Gimbal Stabilization System,” *Journal of Applied Informatics and Computing*, vol. 9, no. 1, pp. 45–52, 2025.

[25] JAIC Authors, “Hand Gesture-Based Control System for UAV Swarm Applications,” *Journal of Applied Informatics and Computing*, vol. 9, no. 1, pp. 53–61, 2025.

[26] M. Quigley *et al.*, “Robot Operating System (ROS): An Open-Source Robot Middleware,” *IEEE Robot Autom Mag*, vol. 16, no. 3, pp. 72–81, 2009.