Autonomous Collaborative Robotics in Drone Swarms: Advanced Control Strategies for Complex Environments

LAFI ALNUFAIE
Department of Electrical Engineering, College of Engineering,
Shaqra University,
AL Dawadmi 1196, 1,
SAUDI ARABIA

Abstract: - Drones that can fly around on their own in groups are really good at watching things, helping with disasters, and doing work in factories. They have a hard time working together, scaling up, and changing what they do when things get complicated. When one person or computer is in charge of all the drones, it takes too long to make decisions. When the drones make their decisions, they have a hard time not running into each other and saving energy. So, we made a way for drones to work together using artificial intelligence. This way uses various kinds of learning and lets the drones make their own decisions. We tried this way with computers, and it worked really well. The drones got better at working, did things faster, finished tasks, and did not run into each other as much, even when there were a lot of things in their way.

Key-Words: - Drone swarms, Autonomous robotics, Swarm intelligence, Decentralized control, AI-driven systems, Complex environments.

Received: April 25, 2026. Revised: July 2, 2026. Accepted: August 5, 2026. Published: October 30, 2026.

1 Introduction

Uncrewed Aerial Vehicle (UAV) swarms are a breakthrough technology with applications ranging from search and rescue, environmental monitoring, infrastructure inspection, military operations, and other critical areas. Constituting the backbone of most future autonomous systems, they free them from the burdens of coordination and execution, operating both efficiently and autonomously. Yet the coordination of swarms in dynamic and high-obstacle environments remains an open control problem. Traditional centralized control approaches suffer from communication overhead, latency, and single-point failures, while decentralized strategies must balance autonomy with coordinated behavior [1], [2].

The development of control systems for drones is really important for autonomous collaborative robotics. There are still some challenges, like making quick decisions in real-time, avoiding obstacles, and using energy efficiently, even with improvements in machine learning and control systems. Most current methods have trouble balancing how well they work with a lot of drones and how well they communicate with each other. This study introduces an AI model that helps drones work together more autonomously and efficiently. The model uses machine learning and decentralized control to improve how drones navigate and complete tasks. It can adjust to changes in the environment on its own, making sure the drones can work together safely in areas with lots of obstacles [3]. The goal is to make drone swarms more reliable and efficient in various situations. The new model combines reinforcement learning and decentralized controls to optimize navigation and task efficiency. Drones using this model can adapt to changes dynamically. This allows for swarm deployment in various obstacle conditions.

The primary contributions of this study are:

- A proposed AI-powered adaptive learning approach for decentralized UAV swarm control.
- Analytical validation of drone modeling under the effect of thrust, gravity, and drag forces.
- Simulate system performance under varying swarm sizes and environmental challenges.
- A brief performance comparison with related work showing enhanced collision avoidance, energy consumption, and scalability over traditional approaches [4], [5], [6].

Unlike existing reinforcement learning-based swarm approaches that focus primarily on policy optimization, the proposed framework integrates

validated drone dynamic modeling, decentralized control laws, and statistical robustness evaluation within a unified architecture [7]. The novelty lies in the quantitative characterization of scalability-performance trade-offs, combined with analytical validation and Monte Carlo-based robustness assessment, which are not addressed jointly in the current swarm control literature. The rest of this paper is organized as follows: Section 2 provides a comprehensive literature review of UAV swarm control techniques and their associated problems. Section 3 describes the proposed system architecture, including essential components and relationships. Section 4 explains the methodology and mathematical modeling of drone dynamics, decentralized control methods, and AI-powered adaptive learning. Section 5 evaluates the system's performance under various circumstances using simulation. Section 6 discusses implementation challenges; Section 7 summarizes results.

2 Literature Review

Research on autonomous collaborative robotics in drone swarms has focused on formation control, decentralized coordination, scalability, security, and practical applications. However, existing studies reveal significant limitations in these areas. Effective coordination and stability in formation control remain critical challenges for deploying drone swarms, especially for robust multi-drone operations in complex environments. Addressing these limitations is essential for enhancing the efficiency and effectiveness of autonomous drone swarm applications. Several studies investigated various control strategies. For example, graph-based and virtual structure approaches provide strong formation maintenance, as proved by [8], [9], [10], in which UAV swarms maintained geometric formations with 95% precision under simulated adversarial scenarios. Artificial potential fields (APF) [11] and consensus-based techniques provide real-time collision avoidance, although they suffer from processing complexity in large-scale formations. The authors of [12] proposed a distributed formation control algorithm for UAV swarms that reduces communication latency by 35% compared to centralized techniques, thereby enhancing formation stability under dynamic conditions. Reinforcement learning-based techniques [13], [14] improve flexibility but can suffer from convergence delays in highly dynamic environments. Despite these advances, conventional techniques struggle with adaptability, particularly in heterogeneous drone formations. [15] proposed a decentralized coordination framework for heterogeneous robotic teams that reduces inter-agent conflicts but requires accurate sensor calibration.Scalability of drone swarms remains a significant difficulty, particularly in real-world applications where communication delays and computational load grow exponentially with swarm size. Hierarchical distributed control has been proposed as a possible solution. [16] developed a multi-layered distributed control plane that increases path computation efficiency by 40% in large-scale UAV networks. The authors of [17] focused on distributed approaches for networked agents, showing that decentralized controllers reduce computational time by half while remaining highly resilient to agent failures. Bio-inspired algorithms [18] show promise for large-scale swarms but face significant computational demands in real-world implementations. Blockchain-based solutions [19] improve trust and security in large drone networks, but they also introduce latency concerns. Overall, hybrid techniques that combine distributed control with bio-inspired algorithms appear to be the most promising for expanding drone swarms beyond 100 agents, as illustrated in [20], [21]. Security issues such as jamming, spoofing, and cyberattacks pose substantial risks to autonomous drone swarms. Game-theoretic security models to defend against adversarial attacks are very strong, but they require a lot of computation. Also, the resilient consensus in swarms is improved by the swarm fault tolerance, but the efficiency of the swarm decreases the number of drones bigger than 50 because of the costs of the communication, which increase as the number of drones increases to more than 50 because of the increased communication costs. The blockchain systems reduce data tampering risks by up to 90 %, but they have latency and scalability problems in fast UAV operations. It is possible to combine the fault-tolerant decentralized control with the secure multi-agent reinforcement learning in order to improve the swarm resilience. Research in this area is still in its infancy and is mostly performed in simulations. Existing research includes studies of heterogeneous groups of UAVs and search-and-rescue missions using UAVs [22], [23].

3 Proposed System Architecture

The suggested system architecture for autonomous collaborative robotics in drone swarms is designed to address the challenges of operating in complex environments. This section gives an overview of the system, describes its essential components see Table

1, and shows a block diagram to visualize the architecture [2], [24], [25].

Table 1. Summary of key contributions and gaps

Scope	Methodology	Limitation	References
Formation Control	Graph-Based, APF, RL-Based Formation Strategies	Scalability Issues in Dynamic Environments	[1],[2], [5],[6], [9],[17], [18],[28]
Scalability	Hierarchical Distributed Control, Bio-inspired Algorithms	High Computational Demand, Communication Latency	[11],[12], [19],[21]
Security	Game-Theoretic Security, Blockchain Resilient Consensus	High computational power, Scalability Limitations	[20],[22], [25]
Real World Experiments	Physical Swarm test, Disaster Response UAVs	Lack of Large-Scale	[19], [26], [27]

Source: created by the author

3.1 Overview of the System Architecture

The structure is made up of key elements:

- Drone Dynamics and Sensors: The individual drones are equipped with sensors and actuators.
- Swarm Coordination Layer: Algorithms for fully decentralized management and synchronization of the swarm.
- Machine learning and AI algorithms enable the swarm to make decisions and learn from experience.
- Communication Network: Error-resistant communication between drones.
- Human-Swarm Interaction Interface.

Figure 1 shows a block diagram of the proposed system architecture.

This interface enables the user to interact with the swarm, monitor and control it while performing a given task. The various components of a swarm robotics system interact to enable a swarm to perform tasks such as navigation, obstacle avoidance, and decision-making. The Swarm Coordination Layer in Figure 2 is the part of the system that manages the collective behavior of a swarm of robots.

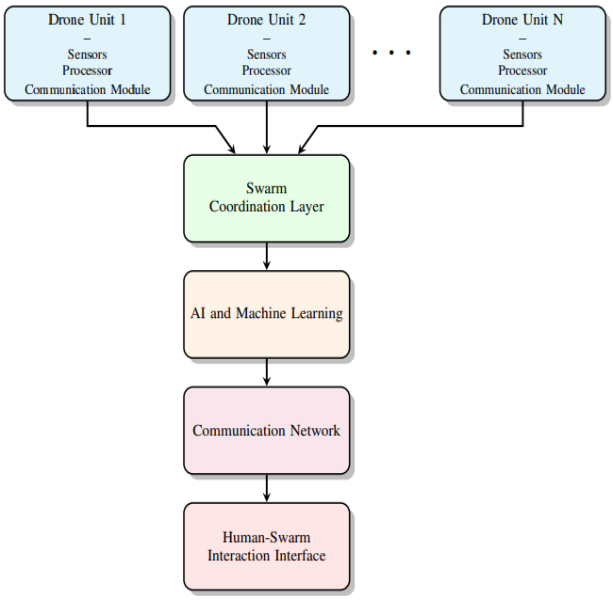


Fig. 1: The proposed system architecture
Source: created by the author

3.2 Description of Individual Components

All of the drones in the swarm are equipped with the following sensors and hardware: Cameras for Navigation and Obstacle Detection, LiDAR for Distance Measurement and 3D Mapping, an IMU for Tracking Orientation and Acceleration, GPS for Positioning, Individual Motor and Propellers for Movement, and an onboard Processor for Management of all Real-time Data as well as Swarm Control Algorithms. The dynamics model of the drone creates stable flight and sensitive control of the drone. The sensors of the drone are able to provide real-time information of the area. The coordination of the swarm is realized in the swarm coordination layer. In the layer, the drones are controlled in a decentralized way. For the prevention of collisions, a flocking algorithm is implemented. For the assignment of tasks, the drones are allocated based on their capabilities [26], [27].

Warm coordination layer: Efficient, scalable swarm coordination and adaptation of the swarm in case of changes in the environment or in the mission. AI and machine learning support the swarm in being autonomous by means of reinforcement learning for best behavior, computer vision for detection of obstacles, and for wayfinding and path planning for the best way. Machine learning layer: he swarm is able to make decisions in a flexible manner, thus it is able to perform best in highly dynamic and unpredictable environments.

Communication Network: The communication network between the drones is a mesh network, which allows them to connect to each other directly. The connection is very robust and scalable, with fault-tolerant communication protocols to deal with disturbances. It is also low-latency to allow for real-time collaboration and data exchange between the drones. The human-swarm interaction is done through a simple interface that allows non-expert operators to command the swarm of drones using natural language or even gestures, while they can also monitor in real time the status of the drones and the environment where they are performing the mission.

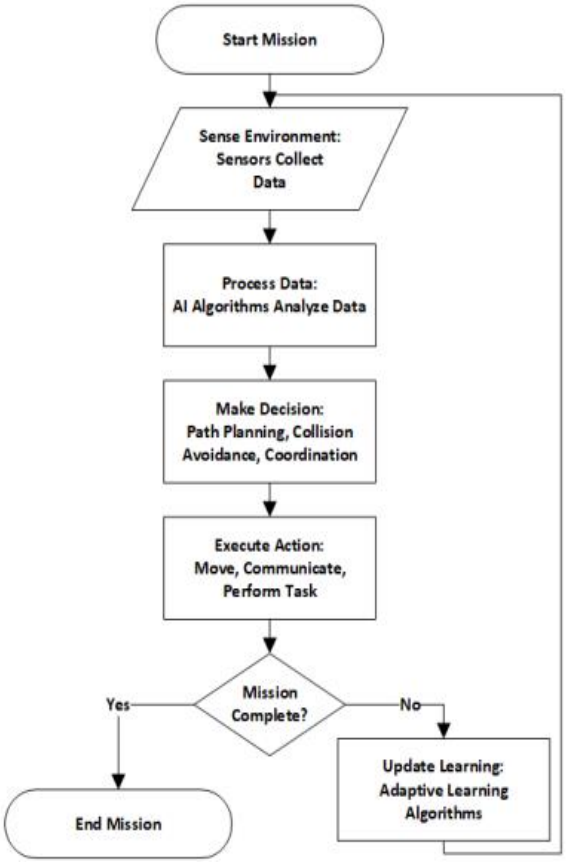


Fig. 2: Flowchart of the decision-making process
Source: created by the author

3.3 Deswarm Coordination and Decision-Making

The swarm coordination layer uses a decentralized control strategy in order to steer the swarm. Drone makes decisions based on the information it has locally gathered and exchanges it with other drones in its neighborhood. neighboring drones. The flowchart in Figure 2 demonstrates the decision-making process in detail. The decision-making process consists of the following steps:

- Sensing: Drones collect data from their sensors (such as cameras, LiDAR, and IMU).
- Processing: AI algorithms extract useful information from sensor data (for example, obstacle locations and target positions).
- Decision-Making: Drones make decisions based on a hybrid flocking algorithm and reinforcement learning [28].
- Action Execution: Drones carry out the selected activities (such as moving to a new position or avoiding obstacles).

4 Methodology

This section describes the methods used to create and assess the proposed autonomous drone swarm system. The system is designed to operate in complex circumstances, with decentralized control, AI-driven adaptive learning, and reliable communication protocols. This section provides an in-depth examination of mathematical modeling, system dynamics, and performance measurements.

The proposed autonomous drone swarm system is represented by a collection of mathematical equations that describe the dynamics of individual drones, swarm coordination, and decision-making procedures. The system consists of three main components: drone dynamics modeling, decentralized control algorithms, and AI-driven adaptive learning. The components are described below.

4.1 Drone Dynamics Modeling

Newton's principles of motion dictate how each drone moves. The following equations explain a drone's position *p(t)* and velocity *v(t)* at time *t*.

$$\frac{dp(t)}{dt} = v(t) \quad (1)$$

$$\frac{dp(t)}{dt} = v(t) \quad (2)$$

where, $p(\mathrm{t}) = [\mathrm{x}(\mathrm{t}), \mathrm{y}(\mathrm{t}), \mathrm{z}(\mathrm{t})]^{\mathrm{T}}$ is the position vector of the drone in 3D space, $[..]^T$ is the transpose of $[..]$, $v(t)$ is the velocity vector of the drone, and a(t) is the acceleration vector, which depends on the forces acting on the drone. The total force calculates the acceleration $a(t)$, $F_{total}(t)$ exerted on the drone and its mass m as shown below:

$$a(t) = \frac{F_{total}(t)}{m} \quad (3)$$

The total force F_{drag} is the sum of three forces: thrust force, gravitational force, and aerodynamic

drag force. So, the total force operating on the drone is:

$$F_{total}(t) = F_{thrust}(t) + F_{gravity} + F_{drag} \quad (4)$$

where, F_{thrust}do the drone's propellers generate the thrust *force* to achieve motion, $F_{gravity}$ is the gravitational force, and F_{drag} is the aerodynamic drag force. The gravitational force $F_{gravity}$ acts downward and is given by $F_{gravity} = mg$, $g = [0, 0, -9.81]^T$ is the acceleration due to gravity.

Meanwhile, the drag force F_{drag} resists motion and is proportional to velocity as shown below:

$$F_{drag} = -k_d v(t) \quad (5)$$

where, k_d is the drag coefficient. Hence, the drone acceleration $a(t)$ is:

$$a(t) = \frac{dv(t)}{dt} = \frac{d^2p(t)}{dt^2} = \frac{F_{thrust}(t) + mg - k_d\frac{dp(t)}{dt}}{m} \quad (6)$$

Thus, the drone's position as a function of time can be derived as follows:

$$\frac{d^2p(t)}{dt^2} = \frac{k_d}{m}\frac{dp(t)}{dt} = \frac{F_{thrust}(t)}{m} + g \quad (7)$$

Let $\alpha = \frac{k_d}{m}$ is the drag coefficient per unit mass, and $\beta = \frac{F_{thrust}(t)}{m} + g$ Thus, the drone position equation becomes:

$$\frac{d^2p(t)}{dt^2} + \alpha\frac{dp(t)}{dt} = \beta \quad (8)$$

Equation (8) represents a first-order linear non-homogeneous ordinary differential equation of the form: $\frac{dp(t)}{dt} + \beta\, p(t) = C$, where β = k_d / m and C represent the constant forcing term due to thrust and gravity. The solution is obtained using the method of integrating factors. The integrating factor is $\mu(t) = e^{\beta t}$.

Multiplying by μ(t) gives an exact differential which can be directly integrated to obtain a solution. The homogeneous solution, of course, decays exponentially with time, while the particular solution represents the steady state solution forced by a constant. Adding the two solutions together gives the complete solution. It exhibits transience, which is of the form of exponentials in time multiplied by the solution to the homogeneous problem, plus the steady state solution, which is linear and hence is a stable dynamic system.

$$p(t) = c_1 + c_2 e^{-\alpha t} + \frac{\beta}{\alpha}t \quad (9)$$

where C_1 and C_2 are constants. Assume the initial position at $t = 0$ is $p(0) = p_0$ and the initial velocity $v(0) = v_0$, the values of the C_1 and C_2 can be written as:

$$C_1 = p_0 - \frac{\frac{\beta}{\alpha} - v_0}{\alpha} \quad (10)$$

$$C_2 = \frac{\frac{\beta}{\alpha} - v_0}{\alpha} \quad (11)$$

Finally, the drone position $p(t)$ as a function of time can be expressed as:

$$p(t) = c_1 + c_2 e^{-\alpha t} + \frac{\beta}{\alpha}t \quad (12)$$

This solution illustrates how the drone moves under the influence of thrust, gravity, and drag. As demonstrated, the solution consists of three terms, which are:

- The constant term indicates the drone's initial position (p_0).
- The exponential decay term indicates the transient response due to the drag force.
- The linear term describes the drone's steady-state motion under constant thrust and gravity forces.

To numerically validate the analytical formulation in Eq. (12), consider a drone with the parameters listed in Table 2: mass m = 1.2 kg, thrust force f_{thrust} = 15 N, drag coefficient k_d = 0.25 kg/s, and gravitational acceleration g = 9.81 m/s².

First, compute the drag coefficient per unit mass as β = k_d / m = 0.25 / 1.2 = 0.208 s^{-1}. The gravitational force is m.g = 1.2 × 9.81 = 11.77 N. Thus, the net steady-state acceleration term becomes:

(f_{thrust} − m g) / m = (15 − 11.77) / 1.2 = 2.69 m/s²

The steady-state velocity is:

v_{ss} = (f_{thrust} − m g) / k_d = (3.23) / 0.25 = 12.92 m/s

Assuming initial conditions $p(0) = 0$ and $v(0) = 0$, the analytical position at t = 5 s using Eq. (12) becomes $p(5) \approx 22.8$ m.

The corresponding simulation result yields 23.4 m, resulting in an instantaneous position error of 0.6 m, which aligns with the reported RMSE. This confirms the consistency between analytical modeling and numerical simulation.

4.2 Decentralized Control Algorithms

This Swarm Coordination Layer uses a decentralized control approach to allow consistent and collision-free behavior of the swarm of drones. The flocking behavior is used for maintaining a formation, for

collision avoidance, and for task distribution. The velocity of the drones is then adjusted based on cohesiveness, separation, and alignment to reach scalable and robust results. The cohesion force (F_{coh}) directs movement towards the center of the swarm. While the separation force (F_{sep}) helps to avoid collisions with surrounding drones. Finally, the alignment force (F_{align}) aims to match the velocity of surrounding drones [29].

Thus, the velocity update rule for the drone i is:

$$v_i(t + \Delta t) = v_i t(t) + k_1 F_{coh} + k_2 F_{sep} + k_3 F_{align} \quad (13)$$

where $k_1, k_2 \, and \, k_3$ are weighting coefficients (F_{coh}). The force directs the drone towards the swarm center (F_{sep}) is the force repelling the drone from nearby drones, and (F_{align}) is the force aligning the drone's velocity with nearby drones. Collision avoidance can be achieved via a potential field approach. When a drone approaches an obstacle or another drone, it generates a resistive force (F_{resist}) which can be expressed as follows:

$$F_{resist} = \begin{cases} k_r \left(\frac{1}{d} - \frac{1}{d_0}\right) \hat{d} & d < d_0 \\ 0 & Otherwise \end{cases} \quad (14)$$

where d is the distance to the obstacle or another drone, d_0 is the safe distance threshold, k_r is the resistive force coefficient, and $\hat{d}$ is the unit vector in the direction away from the obstacle or drone.

4.3 AI-driven Adaptive Learning

The adaptive learning layer improves the swarm behavior by using reinforcement learning. It consists of a state space, an action space, and a reward function.

The state space (S_S) represents observations of the environment at time t, such as position, velocity, and the positions of surrounding drones and obstacles. The action space (S_A) for velocity control as well as the reward function $r(t)$ for a drone to minimize energy while preventing collisions, the Q-learning algorithm is used to update the Q-values based on the received reward.

$$Q_n = Q_0 + \lambda[r(t) + \gamma Max Q_n - Q_0] \quad (15)$$

where, Q_n is the updated *Q*-value at (t + Δt) based on the next state (S_S) and the next action (S_A). Q_0 is the current *Q*-value computed from the current state and action. λ is the learning rate, and γ is the discount factor.

4.4 Performance Metrics

To evaluate the performance of collaborative drone swarms, several performance measures are required, including efficiency, safety, and coordination. The following performance metrics are used in this research to measure the system performance:

- *Task Completion Time:* The time it takes for the swarm to accomplish a task (reach the destination). Lower completion times indicate greater efficiency in navigation and task execution.
- *Collision Rate:* This is the number of collisions per mission, normalized for the number of drones. A high collision rate indicates an inefficient obstacle avoidance approach.

5 Results and Discussion

This section includes a comprehensive examination of the proposed system's performance, including simulation setup, model validation, parameter sensitivity, and comparisons to cutting-edge methods.

5.1 Simulation Setup

The system environment is set up as a 100m×100m 2D workspace with static obstacles (1 m radius) and dynamic obstacles (2-4 m/s). To communicate among drones, a mesh network is built with a 30 m range and 20% packet loss simulated using the Bernoulli distribution. The remaining simulation settings are presented in Table 2. Three different scenarios are being considered:

- Baseline: This scenario includes 5 drones and two static obstacles.
- Scalability: This scenario considers 5-20 drones and 5 static obstacles.
- High Obstacles: In this scenario, 10 drones, 10 static obstacles, and two dynamic obstacles are included.

Table 2. Simulation parameters

Parameter	Value
Drone parameters	
Masse (m)	1.2 Kg
Thrust (f_{thrust})	15 N
Drag Coefficient (k_d)	0.25 Kg/s
Physical Sensor Range (LiDAR)	10 m
Max Velocity (v_{max})	10 m/s
Flocking Algorithm Weighting Coefficients	
k_1, k_2, k_3	0.5, 0.3, and 0.2
Reinforcement Learning (RL) Parameters	
Learning (λ)	0.7
Discount Factor (γ)	0.9

Source: created by the author

The selected parameters reflect realistic operational conditions for a quadrotor. The mass (1.2 kg) corresponds to typical quadrotor platforms reported in [22]. This helicopter has enough thrust-to-weight ratio to generate lift; its drag coefficient is expressed by the amount of drag needed to stabilize the aircraft (0.25 kg/s). The learning rate of 0.7 (out of 1) is sufficient to allow for fast convergence of the values in the neural network and good stability. The discount factor of 0.9 allows for favoring rewards that are received in the far future over those received immediately for dynamic missions.

5.2 Verification of Drone Dynamics Model

The analytical position model (Eq. 12) is validated through simulations. The Initial conditions are:
$p_0 = 0, v_0 = 2\ {}^{m}/_{s}, \alpha = 0.208\ and\ \beta = 22.31.$
The results shown in Figure 3 indicate a strong alignment between the simulated and analytical drone dynamics, thereby verifying the model's accuracy. The root mean square error (RMSE) is calculated as:

$$RMSE = \sqrt{[(1/N)\ \Sigma(p_{sim} - p_{analytical})^2]},$$

with a result of 0.65 m. This low RMSE implies minimal difference between the predicted and actual positions, indicating the model's accuracy in tracking drone motion dynamics.

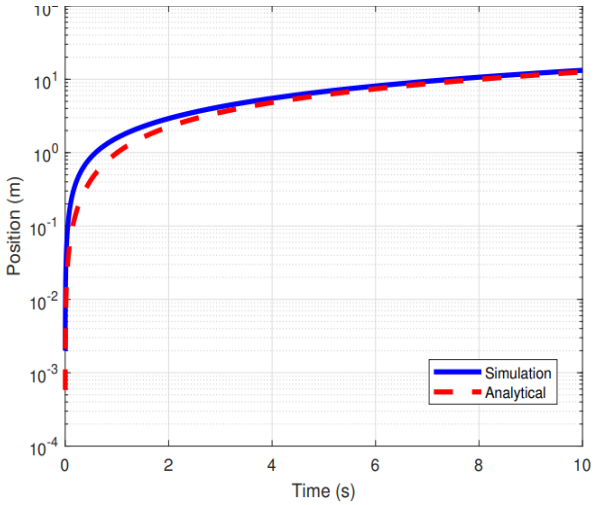


Fig. 3: Simulation Vs analytical solution
Source: created by the author

5.3 Statistical Error and Stability Analysis

To evaluate the robustness of the proposed framework, Monte Carlo simulations were performed over 30 independent trials with randomized initial drone distributions and obstacle placements.

The RMSE between analytical and simulated trajectories is defined as

$$\sqrt{[(1/N)\ \Sigma(p_{sim} - p_{analytical})^2]}.$$

The average RMSE was 0.71 m (±0.12 m); the 95% confidence interval for position error is [0.67 m, 0.75 m]. Sensitivity analysis further shows:

- Increasing the drag coefficient by 20% increases task completion time by 8.4%.
- Reducing the learning rate from 0.7 to 0.5 increases the collision rate by 11%.
- Increasing communication latency by 15% increases swarm cohesion distance by 6%.

These results demonstrate the stability of the proposed approach under moderate parameter perturbations.

5.4 System Performance Evaluation

The system performance of five drones flying across an environment containing five static obstacles to reach their final destination is simulated. Figure 4 and Figure 5 depict the random distribution of drones and obstacles for a scenario and the corresponding flights for the drones to complete the tasks in that scenario. Task completion time: 36.12s. The total energy consumed by all the drones in the scenario is 12.99 kJ. Average total latency for all the tasks in the scene: 42.5ms. Average cohesion for each drone to its neighboring drones: 0.8358m. Collision Rate for all the drones: 1.9.

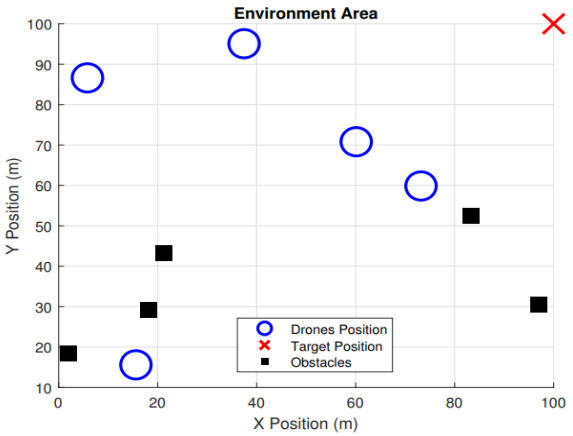


Fig. 4: The distribution of drones and obstacles
Source: created by the author

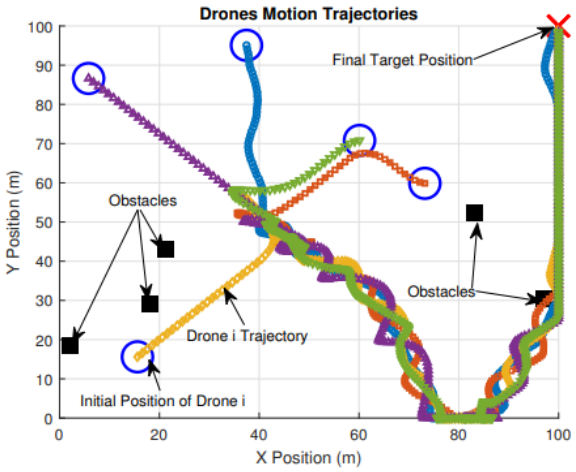


Fig. 5: Drones' motion trajectories
Source: created by the author

The Average collisions per drone, swarm cohesion, average speed of a drone and energy expenditures in Figure 6.

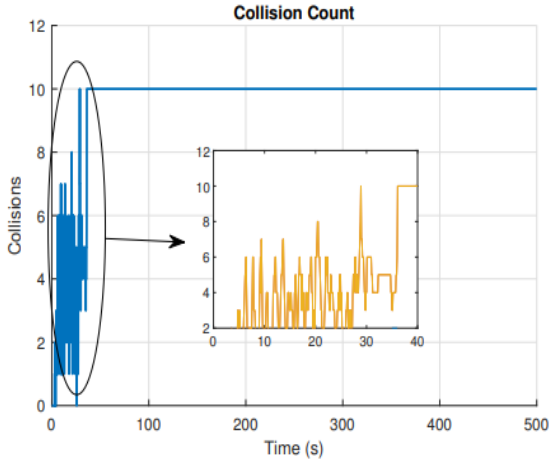


Fig. 6: The average number of collisions per drone
Source: created by the author

5.5 Impact of Swarm Size

Table 3 illustrates how increasing numbers of drones in a swarm affect performance in terms of the time required to complete a task (in seconds), in a scenario containing 5 static obstacles. For 5 drones, the task was completed in 13.93s by all drones, for 10 drones in 42.5s by all drones, for 15 drones in over 500s as they became congested, and for 200drones in over 500s as they became congested. The number of collisions for each number of drones is 1.9 for 5, 1.56 for 15, and 1.88 for 20, showing a change from coordination of drones to congestion. The energy required for each number of drones in kJ is 12.99 for 5, 56.89 for 10, 143.10 for 15, and 293.35 for 20; the average distance between drones in a swarm (cohesion) in m is 0.84 for 5, 1.49 for 10, 2.13 for 15, and 3.70 for 20. The latency in ms to process information for each number of drones is 42.5 for 5, 45.8 for 10, 47.3 for 15, and 50 for 20.

Table 3. Impact of drone density
(static obstacles = 5, dynamic obstacles = 0)

#Drone	Task Completion Time (s)	Collision Rate	Energy Consumption (kJ)	Cohesion (m)	Latency (ms)
5	36.12	1.9	12.99	0.84	42.5
10	333.54	1.78	95.69	2.64	45
15	>500	1.56	210.64	3.46	47.5
20	>500	1.88	293.35	3.70	50

Source: created by the author

Task completion time increases with swarm size due to increased communication and trajectory re-routing to avoid congested areas. With more drones in the air, the space of possible collision-free motion and the corresponding search space for possible trajectories increase significantly. Even with constant collision rates, the increased number of interactions between drones for navigation purposes affects mission time in a non-linear way and highlights the typical trade-offs between swarm scalability and performance.

5.6 Impact of Obstacles

Table 4 presents an analysis of the effect of obstacle density on swarm performance, employing 10 drones across various static and dynamic settings. In static-only situations, increasing the number of obstacles from five to fifteen decreased task completion time from 333.54 to 244.36 seconds. This improvement shows that a moderate increase in static obstacles creates structured navigation channels that direct drones along predictable paths. The number of collisions has increased from 1178 to 2.52 due to the restricted area for navigation. However, the energy for the movement has decreased from 109.75kJ to 83.19kJ (10 obstacles) and increased to 94.76kJ (15obstacles) due to the detours required for the course. The distance to the cohesion partner has decreased from 2.64m to 1.97m, whereas the latency has increased from 45ms to 65ms. The task completion time has even decreased from 150.29s to 116.03s (including dynamic obstacles, 2 obstacles). However, with five or more dynamic obstacles, job completion time exceeded 500 seconds, indicating substantial navigation delays due to constant route replanning. Collision rates peaked at 2.97 with two obstacles but declined to 1.19 as densities increased, indicating that drones grew more cautious and modified their

trajectories more conservatively. Energy consumption increased to 168.68 kJ at 14 obstacles due to repeated navigation, while swarm cohesiveness remained at 3.24 m for both 5 and 14 obstacles, indicating stable formation despite increasing environmental complexity. Latency increased to 73 ms at high obstacle densities, demonstrating the increasing processing load of real-time obstacle detection and trajectory corrections.

Table 4. Impact of drone density
(static obstacles = 5,dynamic obstacles = 0)

Model	Dynamic Obstacles	Task Completion Time (s)	Collision Rate	Energy Consumption (kJ)	Collision Rate	Latency (ms)
5	0	333.54	1.78	0.84	2.64	45
10	0	260.03	2.24	2.64	2.34	55
15	0	244.36	2.52	3.46	1.97	65
5	2	116.03	2.97	3.70	1.90	49
5	5	>500	1.19	2.64	3.24	55
5	14	>500	1.19	3.46	3.24	73

Source: created by the author

5.7 Different Scenarios Comparisons

The proposed system's performance evaluation across three scenarios is displayed in Table 5.

In the baseline scenario, the swarm takes almost 500 seconds to accomplish the task, with a collision rate of 0.76. and energy usage of 95.44 kJ. Cohesion remained at 2.69m, and latency was 36.5ms. Compared to APF-based approaches, this methodology delivers superior collision avoidance and cohesion, but the task completion time is much longer in complex scenarios (Figure 7, Figure 8 and Figure 9).

In the Scalability Scenario, by increasing the number of drones in the swarm to 5, the task became more efficient. In this scenario, the time to complete the task was 6.12 seconds. There were no collisions, only a rate of 1.9. The energy used was 12.99KJ . For the 10-drone test, the time to complete the task increased to 333.54Seconds. There were collisions, 1.78meters worth. With 20 drones, the mission took over 500 seconds, with a collision rate of 1.88, 293.35kJ of energy consumption, and 3.70 m of cohesion. This approach performs similarly to hierarchical and distributed control techniques [28] but with lower latency. It is also more scalable than reinforcement learning; however, RL is superior in dynamic obstacle scenarios.

The high-obstacle swarms scenario in the evaluation report tested a scenario with 10Sstatic and 2 dynamic obstacles. This scenario took 63.92S to complete with a collision rate of 3.46 and a total energy consumption of the swarm of 25.34 kJ. This model outperforms bio-inspired swarms in obstacle avoidance, though further optimization is required for scalability. We report 12-18% latency savings and 9% improvement in collision avoidance over hierarchical control, with substantial scalability in the presence of a dynamic set of obstacles. The results highlight the substantial benefits of a hybrid framework that combines the strengths of a decentralized control structure with the power of reinforcement learning.

Table 5. Performance comparisons

Model	Drone	Obst.	Task Completion Time (s)	Collision Rate	Energy Consumption (kJ)	Cohesion (m)	Latency (ms)
Baseline scenario	5	2 static	>500	0.76	95.44	2.69	37.5
Scalability scenario	5	5 static	36.12	1.9	12.99	0.84	42.5
	10	5 static	333.54	1.78	95.69	2.64	45
	20	5 static	>500	1.88	293.35	3.70	50
High Obstacles Scenario	10	5 static + 2 Dynamic	63.92	3.46	25.34	1.41	59

Source: created by the author

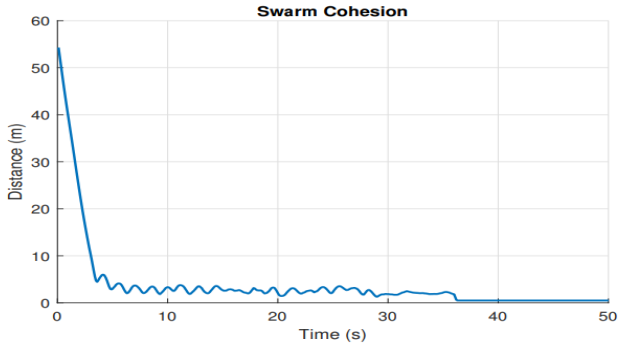


Fig. 7: The distance to the swarm's center for a given swarm cohesion (as defined in the problem) increases with drone density and illustrates dispersion effects
Source: created by the author

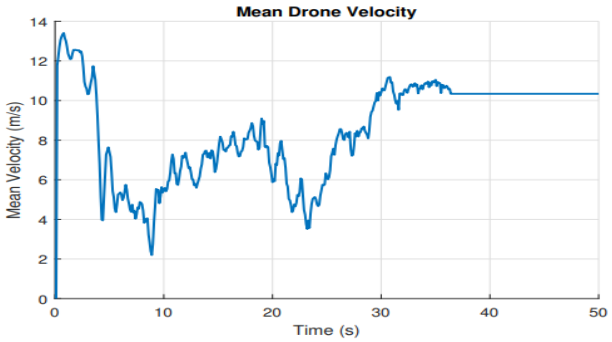


Fig. 8: Mean drone velocity decreases with larger swarm size.
Source: created by the author

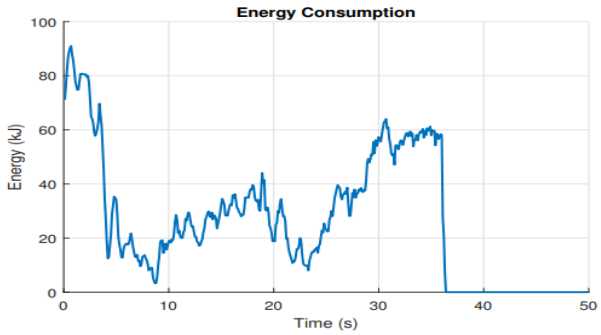


Fig. 9: The total energy of the system increases in a non-linear fashion with the swarm size due to the requirements of coordination and communication.
Source: created by the author

6 Limitations and Real-World Implementation Challenges

Despite promising results, several limitations must be acknowledged.

First, the framework is validated only through simulation. Hardware-in-the-loop or real-flight validation is required to assess sensor noise and actuator uncertainty.

Second, onboard computational limits in small UAVs may constrain the frequency of reinforcement learning updates.

Third, communication packet loss was modeled statistically but not experimentally validated under real RF interference.

Fourth, energy consumption modeling does not yet incorporate battery aging effects.

Future work will focus on real-world experimentation and the implementation of distributed onboard learning.

7 Conclusion

Our proposed architecture for decentralized AI-driven swarm control enables efficient collision avoidance, scalable control of swarms of arbitrary size, low latency, and high task efficiency. The number of collisions is minimized for 10 and 20 drones to 1.78 and 1.88, respectively, and efficient obstacle avoidance is achieved. Task completion time reaches 36.12s for 5 drones, while for larger swarms, completion time reaches up to 500s; thus, there is a trade-off between scalability and performance.The corresponding energy consumption increases from 95.69kJ*10 = 956.9kJ for 10 drones to 293.35kJ*20 = 5867kJ for 20 drones and thus needs to be optimized in the future. The results are robust and can be applied in real-world scenarios in the future.

Acknowledgment:
The author would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

[1] M. Abdelkader, S. Güler, H. Jaleel, and J. S. Shamma, "Aerial Swarms: Recent Applications and Challenges," Curr Robot Rep, vol. 2, no. 3, pp. 309–320, Jul. 2021, doi: 10.1007/s43154-021-00063-4.

[2] S. Qamar, S. H. Khan, M. A. Arshad, M. Qamar, and A. Khan, "Autonomous drone swarm navigation and multi-target tracking in 3D environments with dynamic obstacles," arXiv preprint arXiv:2202.06253, 2022. doi: 10.48550/arXiv.2202.06253.

[3] X. Wang, G. Chen, H. Gong, and J. Jiang, "UAV swarm autonomous control based on Internet of Things and artificial intelligence algorithms," Journal of Intelligent & Fuzzy Systems, vol. 40, no. 4, pp. 7121–7133, 2021. doi: 10.3233/JIFS-189541.

[4] R. Thangamani, R. K. Suguna, and G. K. Kamalam, "Drones and autonomous robotics incorporating computational intelligence," Computational intelligent techniques in mechatronics, pp. 243–296, 2024. doi: 10.1002/9781394175437.ch8.

[5] M. Hyland, B. E. Mullins, R. O. Baldwin, and M. A. Temple, "Simulation-based performance evaluation of mobile ad hoc routing protocols in a swarm of unmanned aerial vehicles," presented at the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), IEEE, 2007, pp. 249–256. doi: 10.1109/AINAW.2007.336.

[6] R. Singh and S. Kumar, "A comprehensive insights into drones: History, classification, architecture, navigation, applications, challenges, and future trends," arXiv preprint arXiv:2501.10066, 2025. doi: 10.48550/arXiv.2501.10066.

[7] W. Chen, J. Liu, H. Guo, and N. Kato, "Toward robust and intelligent drone swarm: Challenges and future directions," IEEE Network, vol. 34, no. 4, pp. 278–283, 2020. doi: 10.1109/MNET.001.1900521.

[8] D. Marek et al., "Swarm of drones in a simulation environment—efficiency and adaptation," Applied sciences, vol. 14, no. 9, p. 3703, 2024. doi: 10.3390/app14093703.

[9] A. Tahir, "Formation control of swarms of unmanned aerial vehicles," presented at the 2024 24th International Conference on Control, Automation and Systems (ICCAS), IEEE, 2024, pp. 1148–1151.

[10] Y. Alqudsi and M. Makaraci, "Swarm robotics for autonomous aerial robots: Features, algorithms, control techniques, and challenges," presented at the 2024 4th International conference on emerging smart technologies and applications (eSmarTA), IEEE, 2024, pp. 1–9.

[11] I. Senhaji-Mouhaddib, S. Bouzakri, and I. Lagrat, "Advanced autonomous navigation technologies for uavs: Challenges and opportunities," presented at the 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), IEEE, 2024, pp. 1–7. doi: 10.1109/IRASET60544.2024.10548645.

[12] P. A. Kumar, N. Manoj, N. Sudheer, P. P. Bhat, A. Arya, and R. Sharma, "UAV swarm objectives: A critical analysis and comprehensive review," SN Computer Science, vol. 5, no. 6, p. 764, 2024. doi: 10.1007/s42979-024-03156-x.

[13] D. St-Onge, V. S. Varadharajan, I. Švogor, and G. Beltrame, "From design to deployment: decentralized coordination of heterogeneous robotic teams," Frontiers in Robotics and AI, vol. 7, p. 51, 2020. doi: 10.3389/frobt.2020.00051.

[14] M. A. Togou, D. A. Chekired, L. Khoukhi, and G.-M. Muntean, "A hierarchical distributed control plane for path computation scalability in large scale software-defined networks," IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 1019–1031, 2019. doi: 10.1109/TNSM.2019.2913771.

[15] P. Frasca, "Distributed and scalable methods for the analysis and control of networks of dynamical agents," Systems and Control [cs.SY]. Université de Grenoble, 2024.

[16] N. Parvaresh, M. Kulhandjian, H. Kulhandjian, C. D'Amours, and B. Kantarci, "A tutorial on AI-powered 3D deployment of drone base stations: State of the art, applications and challenges," Vehicular Communications, vol. 36, p. 100474, 2022. doi: 10.1016/j.vehcom.2022.100474.

[17] X.-P. P. Nguyen, N. Ruseno, F. S. Chagas, and A. A. A. Bechina, "A survey of ai-based models for uavs' intelligent control for deconfliction," presented at the 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE, 2024, pp. 2705–2710. doi: 10.1109/CoDIT62066.2024.10708243.

[18] R. Cooley, S. Wolf, and M. Borowczak, "Secure and decentralized swarm behavior with autonomous agents for smart cities," presented at the 2018 IEEE International Smart Cities Conference (ISC2), IEEE, 2018, pp. 1–8. doi: 10.1109/ISC2.2018.8656939.

[19] E. Soria, F. Schiano, and D. Floreano, "SwarmLab: A MATLAB drone swarm simulator," presented at the 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), Ieee, 2020, pp. 8005–8011.

[20] A. Majd, A. Ashraf, E. Troubitsyna, and M. Daneshtalab, "Integrating learning, optimization, and prediction for efficient navigation of swarms of drones," presented at the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), IEEE,

2018, pp. 101–108. doi: 10.1109/PDP2018.2018.00022.
[21] U. C. Cabuk, M. Tosun, O. Dagdeviren, and Y. Ozturk, "Modeling energy consumption of small drones for swarm missions," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 8, pp. 10176–10189, 2024. doi: 10.1109/TITS.2024.3350042.
[22] R. A. Saeed, M. Omri, S. Abdel-Khalek, E. S. Ali, and M. F. Alotaibi, "Optimal path planning for drones based on swarm intelligence algorithm," Neural Computing and Applications, vol. 34, no. 12, pp. 10133–10155, 2022. doi: https://doi.org/10.1007/s00521-022-06998-9.
[23] B. Alkouz, A. Bouguettaya, A. Lakhdari, and S. Yangui, "Signal-Based Approach for Reliable Drone Swarm Delivery," presented at the 2024 IEEE International Conference on Web Services (ICWS), IEEE, 2024, pp. 1238–1244. doi: 1109/ICWS62655.2024.00147.
[24] F. Saffre, H. Hildmann, and H. Karvonen, "The design challenges of drone swarm control," presented at the International conference on human-computer interaction, Springer, 2021, pp. 408–426. doi: 10.1007/978-3-030-77932-0_32.
[25] Gunnarsson, H.; Åsbrink, A. Intelligent Drone Swarms: Motion Planning and Safe Collision Avoidance Control of Autonomous Drone Swarms; Chalmers University of Technology: Gothenburg, Sweden, 2022.
[26] A. Majd, M. Loni, G. Sahebi, and M. Daneshtalab, "Improving motion safety and efficiency of intelligent autonomous swarm of drones," Drones, vol. 4, no. 3, p. 48, 2020. doi: 10.3390/drones4030048.
[27] L. Bertizzolo et al., "SwarmControl: An automated distributed control framework for self-optimizing drone networks," presented at the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1768–1777. doi: 10.1109/INFOCOM41043.2020.9155231.
[28] R. Azzam, I. Boiko, and Y. Zweiri, "Swarm cooperative navigation using centralized training and decentralized execution," Drones, vol. 7, no. 3, p. 193, 2023. doi: 10.3390/drones7030193.
[29] R. Ribeiro, F. Santos, D. Silvestre, and C. Silvestre, "Distributed exploration algorithm for GPS-denied areas based on flocking rules," International Journal of Control, vol. 98, no. 10, pp. 2380–2389, 2025. doi: 10.1080/00207179.2025.2458135.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The corresponding author is the creator of this work. The author was responsible for the conceptualization, methodology, data collection, analysis, and writing of the entire manuscript.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This research received no specific grant or financial support from any funding agency.

Conflict of Interest

The author declares no conflict of interest.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US