Comprehensive Multi-Level Optimization of Safe Swarm Motion

Leonid Hulianytskyi[1,†], Maksym Ogurtsov[1,†], Vyacheslav Korolyov[1,*,†] and Oleksandr Yarushevskyi[1,†]

[1] *V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Akademika Hlushkova Ave, 40, 03187, Kyiv, Ukraine*

Abstract

This paper presents a comprehensive approach to the safe motion planning of autonomous unmanned aerial vehicles (UAVs) operating in multi-agent swarm formations. The increasing deployment of UAV swarms in both civil and military domains necessitate robust, scalable, and adaptable control strategies that ensure reliable group behavior under dynamic and potentially adversarial conditions.

The proposed architecture integrates a multi-level optimization framework, which combines global trajectory planning with local collision avoidance. It is enhanced by reinforcement learning algorithms and safety-guaranteed maneuvering techniques. A hybrid control architecture is developed, supporting both decentralized and centralized coordination schemes, enabling agents to operate autonomously while maintaining real-time responsiveness to changes in the environment and swarm composition.

Inspired by biological systems and competitive behavioral patterns observed in nature, the architecture includes adaptive roles, leader–follower dynamics, and swarm clustering for obstacle avoidance and attack-defense positioning. A formal system model is defined, along with simulation algorithms for analyzing various swarm sizes and motion control strategies. The effectiveness of proportional-integral derivative parameter optimization for improving swarm dynamics is also investigated.

Experimental simulations demonstrate the model's capability to handle complex interaction scenarios, prevent inter-agent collisions, and maintain formation integrity. The proposed solution lays the groundwork for the development of AI-driven swarm systems that can execute coordinated operations in contested environments.

Keywords

UAV, Swarm, Combinatorial Optimization, Agents, Swarm Intelligence

1. Introduction

Countering a large number of unmanned aerial vehicles (UAVs) is a pressing issue in the fields of security and information technologies [1]. The advances in computer electronics and their miniaturization now enable the integration of onboard networked computational and communication systems with specialized artificial intelligence into groups of autonomous unmanned systems [2], facilitating the application of intelligent information technologies to address this problem.

The use of large UAV groups is currently at the stage of empirically accumulating successful field practices, with preliminary computer simulations of group control processes (agents) based on multi-agent system models in virtual environments [3–4], including group competition [4].

Information Technology and Implementation (IT&I-2025), November 20-21, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

leonhul.icyb@gmail.com (L. Hulianytskyi); maksymogurtsov@gmail.com (M. Ogurtsov); korolev.academ@gmail.com (V. Korolyov) olexand777@gmail.com (O. Yarushevskyi)

0000-0002-1379-4132 (L. Hulianytskyi); 0000-0002-6167-5111 (M. Ogurtsov); 0000-0003-1143-5846 (V. Korolyov); 0009-0004-9930-1643 (O. Yarushevskyi)

1.1. Patterns of Aggressive Competition in Large Groups Inspired by Nature

Competition for resources and territorial dominance is inherent to many living organ-isms on Earth [5]. Therefore, using conflict behavior patterns from groups of living creatures as simulation scenarios for AI-controlled multi-agent systems is a rational approach that can be further implemented and tested for managing autonomous UAV groups.

In the animal world, flock control is based on a strict hierarchical principle — there is a leader and subordinates who perform specific roles within the group. Control structures such as centralized leadership (in animals) and decentralized swarms are both used to model agent group control. Hybrid control models, including temporary leadership and swarm clustering for obstacle avoidance, are also applied [6]. In this paper, we define agents as virtual entities with position, velocity, and radius, following swarm movement algorithms.

1.2. Definitions

To avoid ambiguity, the following terms are defined:

Agent – an acting entity in any process or phenomenon, including simulations.

UAV (Unmanned Aerial Vehicle) – an aircraft without an onboard pilot, controlled remotely or programmed to fly autonomously.

Drone – a mobile unmanned device, such as UAVs, robotic systems, or ground/sea/aquatic unmanned vehicles.

Drone Group – a team or swarm of drones.

Flock – a group of animals, birds, fish, or other organisms that stay together.

Collective – a number of drones (more than one), operating permanently to per-form joint missions/tasks without direct interaction.

Team – several drones (more than one) that perform a mission/task without direct interaction.

Swarm – a group of drones equipped with swarm intelligence, capable of autonomous interaction, adaptation to changing conditions, and collective decision-making with minimal or no operator intervention.

1.3. Objective of the Study

To develop a hybrid control architecture for UAV swarms using reinforcement learning (RL), based on comprehensive multi-level optimization that integrates global trajectory planning and local collision avoidance with formal safety guarantees for emergencies. The architecture should ensure collision-free movement of agents in a dynamic 3D environment through multi-level optimization and cooperative maneuvering. Additionally, the study aims to investigate the optimization of consensus PID (Proportional-Integral-Derivative) controller parameters to improve the dynamic performance of UAV swarms and to develop corresponding algorithmic and software tools for simulating swarm movement of varying sizes under hybrid control systems.

1.4. Structure of the Paper

The remainder of the paper is structured as follows.

Section 2 reviews existing swarm control algorithms and analyzes their limitations. It also presents the general model of swarm agent motion and control, describes individual algorithms – Potential Field method (PFM), Vicsek, Particle Swarm Optimization (PSO), Reynolds, and evaluates their applicability to UAV swarm control.

Sections 3 introduces the proposed hybrid architecture with multi-level optimization, including PID parameter optimization.

Section 4 presents mathematical model of a multi-agent system using CBF-based safety mechanisms, RL for emergency cases and adaptive consensus PID controllers, its structural diagram and hybrid architecture.

Section 5 presents proposed hybrid architecture and optimization strategy for UAV swarm control, including structural diagram of a swarm oriented multi-agent system, hybrid strategy and architecture for safe UAV swarm control.

Section 6 reports the results of numerical experiments and compares the proposed system with classical swarm control methods.

Section 7 concludes the paper and outlines future research directions, including the study of “swarm vs. swarm” scenarios.

1.5. Main Contributions of the Paper

This paper addresses the problem of safe autonomous UAV swarm control under conditions of aggressive interaction and complex dynamic environments. The key contributions are development of a general model of swarm agent motion based on multi-agent dynamics and inter-agent interaction functions; critical analysis of existing swarm control methods (PFM, Vicsek, PSO, Reynolds) and identification of their limitations with respect to safety and controllability; proposal of a hybrid architecture that integrates Model Predictive Control, Control Barrier Functions, adaptive consensus-based PID controllers, and Reinforcement Learning for emergency maneuvers; design of a multi-level optimization strategy that combines global trajectory planning, local collision avoidance, and cooperative maneuvering with formal safety guarantees and implementation of numerical experiments, confirming zero collisions in dense swarm formations and demonstrating the superiority of the proposed system in swarm vs. swarm" scenarios compared to classical algorithms.

2. Review of Existing Models and Their Limitations

In the field of AI ontology, collective decision-making algorithms are part of swarm intelligence [7]. Computer modeling of UAV collective behavior uses multi-agent technologies, along with Information and Communications Technology solutions, to enable distributed computing.

Competitive behavior of biological collectives in resource and territory conflicts includes: reconnaissance (including combat reconnaissance), deep raids into enemy territory, frontal attacks, flanking and encircling maneuvers, military deception (e.g., diversionary tactics, feigned retreat), enemy force dispersion (multi-directional attacks), wave attacks (rotational assaults), blocking enemy retreat (encirclement or siege) and tactical adaptation via route updates based on information exchange.

Multi-agent AI technologies allow modeling of autonomous agent collectives as self-organizing swarms or leader-controlled flocks using ad hoc radio networks. Agents collaboratively solve tasks such as directional movement, obstacle avoidance, and spatial positioning for attack or defense [8]. Therefore, control of agents in swarm/flock formations requires navigation algorithms that maintain acceptable positioning accuracy to prevent collisions and preserve group cohesion.

2.1. General Model of the Agent Swarm Control System

The basis of agent swarm behavior lies in the principles of cohesion, alignment, and separation (the Reynolds model) [9]. These are supplemented by collision avoidance strategies with static and dynamic obstacles and task-specific behavioral patterns (attack, defense, evasion, reconnaissance, deception, and others).

The swarm consists of N agents $A_1, A_2, \ldots, A_N$, each of which is characterized at any given moment by:

Position vector:

$$x_i(t) \in R^3 \tag{1}$$

Velocity vector:

$$v_i(t) \in R^3 \tag{2}$$

Radius (occupied space):

$$r_i \in R^+ \tag{3}$$

The movement of agents occurs in a limited three-dimensional space that may include static and dynamic obstacles. The motion of the agents is governed by the following differential equation:

$$\dot{x}_i = f(x_i, u_i), \tag{4}$$

where f is a function describing the agent's dynamics, and the vector u_i describes the influence of the swarm on the agent's movement and control of neighboring swarm agents. The interaction between swarm agents is introduced using the function g, which depends on their states:

$$u_i = g\left(x_i, \sum_{j \in N_i} h(x_j)\right), \tag{5}$$

where N_i is the set of neighbors of agent i, and is a function that determines the strength of influence on the movement of agent i of the neighboring agents in whose neighborhood it is located. The interaction function may include a description of the rules of movement based on a weighted sum of forces, for example, repulsion, alignment, attraction, and other methods of social interaction between N agents:

$$u_i = g(x_i, z_i) = \sum_{i \in N} \omega_i F_i(x_i, z_i) \tag{6}$$

where w_i are the weight coefficients, F_i, z_i are the forces acting on agent i:

$$z_i = \sum_{j \in N_i} h(x_j) \tag{7}$$

The collective dynamics of a system comprising N agents are described by the following system of differential equations:

$$\begin{cases} \dot{x}_1 = f(x_1, u_1) \\ \dot{x}_2 = f(x_2, u_2) \\ \quad \vdots \\ \dot{x}_N = f(x_N, u_N) \end{cases} \tag{8}$$

The proposed general model captures the motion of a multi-agent system and the control of a swarm of agents, accounting for both inter-agent interactions and individual dynamics. The functions f, g, and h are defined according to the specific characteristics of the system and the prescribed interaction rules. Previous studies by the authors [9] have demonstrated that existing algorithms and methods for modeling group agent motion exhibit several limitations, raising concerns regarding their suitability for controlling agent groups in counter-swarm operations. The developed general model of group motion and multi-agent system control is employed to analyze the limitations of existing models describing the motion of agent groups.

Let us examine the limitations of the PFM, the Vicsek algorithm, PSO algorithm, and the Reynolds algorithm in the context of their application to UAV swarm control under stringent collision-avoidance requirements.

2.2. Potential Field Methods

In PFM, each agent is influenced by artificial forces generated by virtual potential fields [10]. An attractive potential toward the target and repulsive potentials from obstacles and other agents are incorporated into the model. The resulting force determines the agent's motion. The control vector for an agent is defined as the gradient of the potential function:

$$u_i = F_{attr} + F_{rep} = -\nabla U(x_i) = -\nabla\left(U_{attr}(x_i) + U_{rep}(x_i)\right), \tag{9}$$

where $\boldsymbol{F}_{att}$ is the attractive force, and $\boldsymbol{F}_{rep}$ is the repulsive force, each defined as the gradient of the attractive potential of agent i toward the target and the repulsive potential from obstacles or other agents, respectively.

From the perspective of absolute safety and UAV swarm control, the PFM has inherent limitations. The presence of local minima in the force field can trap agents, preventing target acquisition and increasing the risk of collisions, which in swarm motion may cause delays, formation disruption, or partial immobilization.

2.3. Vicsek Model

The Vicsek model was developed to simulate the collective behavior of self-propelled particles in a simple way [11]. Each particle moves at a constant speed and, at each step, changes its direction by averaging the directions of its neighbors within a given radius, with the addition of random noise. The swarm influence vector is defined as the sum of forces acting on the agent, including alignment force and random noise:

$$u_i = F_{align,i} + \eta_i = k_{align} \sum_{j \in N_i} (v_j - v_i) + \eta_i, \tag{10}$$

where $\boldsymbol{F}_{align}$ is the alignment force, calculated as the sum of velocity differences between agent i and its neighbors j, multiplied by the alignment coefficient k_{align}. Vector η represents random noise, introduced to avoid the perfect synchronization of all agents' movement directions. Here, k_{align} determines the strength of alignment, v_j and v_i are the velocities of neighbors and the agent, and N_i is the set of neighbors of agent i.

The Vicsek model of swarm motion, when applied to UAV swarm control, has several inherent limitations. It is primarily oriented toward achieving global order through direction alignment and lacks explicit rules or mechanisms for preventing physical collisions between agents. As a result, collisions remain possible, particularly at high agent densities or in conditions with substantial noise.

2.4. PSO

PSO is a metaheuristic optimization algorithm inspired by the social behavior of birds in flocks or fish in schools. PSO algorithm has been applied to a wide range of optimization problems and can be adapted for swarm control applications [12]. Each agent in the swarm moves under the influence of a balance between its best-found position and the best position found by the entire swarm for that agent.

In PSO algorithm, the sum of the force vectors acting on each agent can be explicitly expressed. These forces include the inertia force, the attraction force toward the personal best position, and the attraction force toward the global best position.

For each agent i, the resulting control force $\boldsymbol{u}_i$ is the sum of three components:

$$u_i = F_{inertia,i} + F_{pbest,i} + F_{gbest,i} \tag{11}$$

where $\boldsymbol{F}_{inertia}$ is the inertia force determined by the agent's current velocity; $\boldsymbol{F}_{pbest}$ is the attraction force toward the agent's personal best position, determined by the difference between the personal best position and the current position; and $\boldsymbol{F}_{gbest}$ is the attraction force toward the global best position, determined by the difference between the global best position and the current position. The sum of these forces defines the agent's acceleration and motion within the search space.

2.5. Reynolds' Boids Algorithm

The Reynolds algorithm, also known as the Boids model, is one of the most well-known approaches to swarm control [13]. The motion of a swarm of agents under the Boids algorithm can be described in terms of a general multi-agent system model using the concept of forces acting on each agent. These forces define the interactions between agents and their behavior within the swarm.

In the Reynolds algorithm, each agent is influenced by three main forces:

Separation force $\boldsymbol{F}_{sep}$: determines the intensity of collision avoidance with neighbors.

Alignment force $\boldsymbol{F}_{align}$: aligns the agent's direction of motion with that of its neighbors.

Cohesion force $\boldsymbol{F}_{coh}$: attracts the agent toward the center of mass of its neighbors to maintain swarm cohesion.

The resulting force acting on an agent can thus be expressed as:

$$u_i = F_{rep,i} + F_{align,i} + F_{attr,i} \tag{12}$$

Let us consider the main disadvantages of the Boids algorithm from the perspective of UAV swarm control. The standard Boids model exhibits several limitations with respect to absolute safety in UAV swarm control. Its reactive nature means that agents respond only to the current positions, velocities, and proximity of neighbors, without predicting future motion trajectories, which can be insufficient to prevent collisions in dynamic environments with high speeds or complex maneuvers. Additionally, groups of agents governed solely by Boids rules may become trapped in locally stable configurations that carry latent collision risks. Conflicts between cohesion and alignment rules may further compromise safety, particularly if rule parameters are not optimally tuned. Finally, the standard Boids model lacks explicit mechanisms for predicting collisions and planning avoidance maneuvers.

In summary, while each of the considered algorithms offers advantages in specific contexts, their direct application to UAV swarm control, where zero-collision performance is critical, presents significant challenges. PFM may lead to agent trapping and does not guarantee safety. The Vicsek model focuses on global alignment, ignoring collisions. PSO is an optimization algorithm and requires specific extensions to enable motion control with real-time safety guarantees. Achieving zero collisions in a dynamic UAV swarm, while preserving controllability and task performance efficiency, is not feasible using algorithms based solely on simple reactive rules or general optimization approaches. This requires the development of hybrid approaches or specialized algorithms that combine proactive prediction, formal safety assurance methods, e.g., Control Barrier Functions (CBF), Reachability Analysis (RA) with optimization of UAV motor controller parameters, enabling the implementation of effective cooperative maneuvering strategies.

3. PID motor controller parameter optimization to improve the dynamic characteristics of a UAV swarm

A key challenge in integrating multiple real-world technical systems is the potential degradation of their original qualitative and quantitative performance metrics and characteristics. For UAV swarms, such subsystems include the flight controller and the PID controller [14], which directly controls the motors, as well as the autonomous swarm navigation program. Simulation of flight and optimization of such a combined system's parameters can be performed in UAV flight simulation software.

For initial flight performance testing of UAVs, three test trajectories are used: a circle, a figure-eight (lemniscate), and a zigzag. Numerical experiments have shown that the relative deviation values for the lemniscate and circle are approximately the same. For the zigzag trajectory, its step and amplitude must be known precisely, as they are specific to the motion of certain UAV types and are useful for determining UAV inertial characteristics during sharp maneuvers — a task outside the scope of this work. Accordingly, in the subsequent discussion, PID controller parameter optimization will be tested for simulated agent motion along a circular path.

To demonstrate the effect of PID controller parameter optimization on deviations from the ideal trajectory, let us formulate the optimization problem for a drone's circular motion.

3.1. Drone motor model

The mathematical model of the drone motor controller is described by second-order differential equations for coordinates x and y. Let the drone move in a circle of radius r and angular velocity ω. Then, the equations for the ideal trajectory (circle) are as follows [14]:

$$x_c(t) = rcos(\omega t), y_c(t) = rsin(\omega t) \tag{23}$$

The optimization criterion consists of minimizing the RMSD from the ideal trajectory, described by the following loss function [15]:

$$J = \int_0^T \{[x(t) - x_c(t)]^2 + [y(t) - y_c(t)]^2\} dt \tag{34}$$

The PID controller's control signal for each direction u_x and u_y is defined as [14]:

$$u_x(t) = K_p e_x(t) + K_i \int_0^T e_x(\tau) d\tau + K_d \frac{de_x(t)}{dt} \quad (45)$$

$$u_y(t) = K_p e_y(t) + K_i \int_0^T e_y(\tau) d\tau + K_d \frac{de_y(t)}{dt} \quad (56)$$

where $e_x(t)$ and $e_y(t)$ are the position errors of the drone in space relative to the values obtained from the solution of the problem.

The solution to the optimization problem is a set of PID controller parameters (K_p, K_i, K_d) that minimize the loss function J, thereby achieving the smallest possible RMSD from the ideal trajectory. The algorithm for solving the PID parameter optimization problem was based on the gradient descent method. The results of the numerical experiments are summarized in Classical PID controller, circular motion, parameters, deviation.

Table 1

Classical PID controller, circular motion, parameters, deviation

Non-optimized PID parameters	Optimized PID parameters
K_p=2.0,K_i=0.1,K_d=0.5	K_p=10.0,K_i=0.0,K_d=5.0
Mean deviation: 0.4475	Mean deviation: 0.0754
Max deviation: 1.0101	Max deviation: 0.2088

Fig. 1 shows the simulation results of drone movement along a circle with a radius of 3 meters. The left graph illustrates the movement without optimization, while the right graph shows the movement with optimization according to the RMSD criterion. The optimization significantly reduces the deviation from the ideal trajectory.

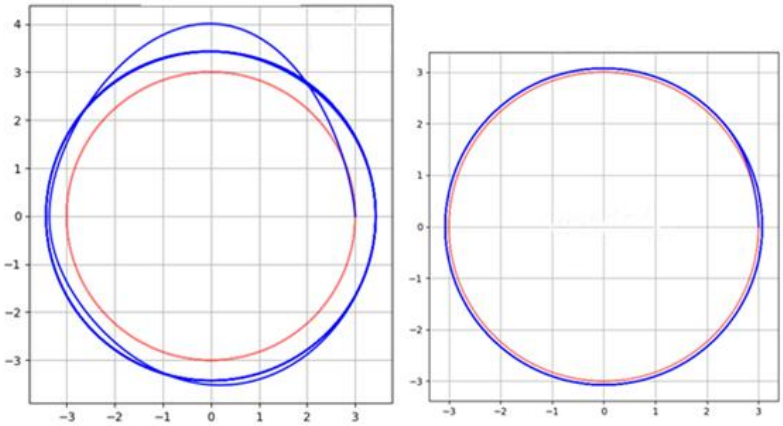

Figure 1: Classical PID controller, circular motion, parameters, deviation

Similar numerical experiments were conducted for fractional and robust controllers [16].

The average error before optimization was about 15%, and after optimization — 2.5%. Thus, PID parameter optimization reduced the trajectory deviation by approximately a factor of 6. This means that developing a multi-level algorithmic framework for group agent control without consensus-based PID parameter tuning can degrade the controllability and safety of flock movement or cooperative swarm maneuvering when implemented on real UAVs.

However, this may still be insufficient. To ensure collision avoidance within the swarm, additional safety mechanisms must be implemented, in particular, the CBF method.

4. Mathematical Model of a Swarm Oriented Multi-Agent System Using CBF and Adaptive Consensus PID Controllers

Performing rapid aerodynamic maneuvers in swarm movement scenarios — target pursuit or swarm-scale evasive action — requires a theoretically guaranteed ability to minimize collisions, i.e., ensuring that agents never leave the set of safe states.

4.1. CBF Method for Constrained System Optimization

Since the 2010s, Barrier Functions (BFs) have been used in robotics to mathematically formalize safety in terms of maintaining minimum distances between agents in constrained systems [17]. CBFs are an extension of these functions for controlled systems, where the motion controller actively maintains safety [18]. The concept of combining CBFs with Model Predictive Control (MPC) for robotics [19] gained traction in 2018, and CBFs became widely integrated into modern robotics between 2020–2023, with applications in autonomous vehicles, robotic manipulators, and swarm robotics.

Functionally similar to CBFs are Optimal Reciprocal Collision Avoidance (ORCA) algorithms, which provide weaker mathematical guarantees and are therefore considered as auxiliary solvers in cases where the CBF method cannot find a feasible solution.

We now present a mathematical model and a concise problem formulation for applying CBFs to ensure the safety of UAV motion. For each pair of neighboring UAVs, a barrier function is defined that becomes zero when the distance between them reaches a predefined safety threshold. The safety condition (imposed on the derivative of the barrier function) constrains the set of admissible control actions for each UAV, guaranteeing that the safety distance will never be violated.

Problem statement for optimizing the movement of a group of agents.

4.2. System description

We consider a multi-agent system with N agents, where each agent $i \in \{1,\dots,N\}$ has:

State $x_i(t)=[p_i(t),v_i(t)]^T$ (position and velocity of the agent at time t);

Local PID controller with parameters $\theta_i=[K_{P,i},K_{I,i},K_{D,i}]^T$;

Data exchange with neighboring agents through a network described by an undirected connectivity graph $G=(V,E)$, where V is the set of vertices and E is the set of edges, $V=\{1,\dots,N\}$.

4.3. The objective function

The objective function describes a joint minimization of the deviation of the current state of each agent from its goal; the control error in the agent's motion; the deviation of the current PID parameters from the consensus values, according to a quadratic criterion:

$$J = \int_0^T \left[\sum_{i=1}^N \left(\alpha_i \left\|x_i(t) - x_{i,goal}\right\|^2 + \beta_i \|u_i(t)\|^2 + \gamma_i \|\theta_i(t) - \bar{\theta}(t)\|^2\right)\right] dt \tag{17}$$

where $\alpha_i,\beta_i,\gamma_i$ are weighting coefficients;

$x_{i,\mathrm{goal}}$ is the target position of agent i;

$u_i(t)$ is the control input;

T is the planning horizon;

$\bar{\theta}(t)$ is the consensus PID parameter vector:

$$\bar{\theta}(t) = \frac{1}{N}\sum_{i=1}^N \theta_i(t) \tag{18}$$

4.4. Constraints

Constraints include agent motion dynamics, PID parameter adjustment, and safe motion enforcement using CBF.

The agent dynamics are described by:

$$\dot{x}_i = f(x_i, u_i), u_i = f_{PID}(e_i, \theta_i) \tag{19}$$

where e_i is the agent's position error relative to the computed trajectory.

The PID control law is:

$$u_i(t) = K_p e_i(t) + K_i \int_0^T e_i(\tau) d\tau + K_d \frac{de_i(t)}{dt} \tag{20}$$

where $e_i(t)$ is the control error.

4.5. The barrier function

The barrier function ensuring a minimum safe separation to avoid collisions:

$$h(x_i, x_j) \leq 0, \forall (i, j), i \neq j \tag{21}$$

where

$$h(x_i, x_j) = \|p_i - p_j\|^2 - (r_i + r_j + d_{safe})^2 \geq 0 \tag{22}$$

r_i, r_j are the safety radii of agents i and j, and d_{safe} is the minimum safe distance.

Consensus conditions for PID parameters have lower priority than safety constraints to ensure problem feasibility:

$$\dot{\theta}_i = -\sum_{j=1} a_{ij}(\theta_i - \theta_j) + \eta_i \nabla_{\theta_i} L_i(x_i, u_i, \theta_i) \tag{23}$$

where a_{ij} are adjacency matrix elements of G, $\eta_i > 0$ is the learning rate, and

$$L_i(x_i, u_i, \theta_i) = \|e_i\|^2 + \lambda_i \|u_i\|^2 + \mu_i \|\theta_i - \theta\|^2 \tag{24}$$

$\lambda_i, \mu_i > 0$ are regularization coefficients.

4.6. The safety conditions

The safety conditions are based on the second-order derivative of the CBF:

$$\ddot{h}(x_i, x_j) + K_d \dot{h}(x_i, x_j) + K_p h(x_i, x_j) \geq 0 \tag{25}$$

where

$$\ddot{h}(x_i, x_j) = 2\|v_i - v_j\|^2 + 2(p_i - p_j)^T (u_i - u_j) \tag{26}$$

Since the safety inequality is linear in u_i and u_j, it can be solved as a quadratic programming problem to find safe control inputs.

4.7. Summary

Developing hybrid and specialized algorithms that guarantee safe agent motion is a complex interdisciplinary task involving control theory, optimization theory, formal verification methods, and swarm dynamics. This approach is key to achieving the ambitious goal of guaranteeing zero collisions in autonomous UAV swarms.

5. Proposed Hybrid Architecture and Optimization Strategy for UAV Swarm Control

The basis of proposed hybrid architecture is the structural diagram of a swarm oriented multi-agent system for a multi-agent control.

5.1. Structural Diagram of a Swarm Oriented Multi-Agent System

Figure 1 illustrates a structural scheme for a multi-agent control and communication system using multi-level optimization [20], which includes an adaptive consensus algorithm for swarm agents' PID controllers and the data processing workflow.

Sensor data, after preprocessing and formatting, is distributed through the agent network via a data exchange algorithm. Based on the sensor data, the collision risk estimation algorithm performs a quick safety assessment.

Data received from other agents is used to check global trajectory compliance, build a short-term motion prediction model, and update the group PID consensus algorithm. With it:

High threat: the agent's PID controller performs emergency maneuvers to avoid collisions with other swarm agents. Low threat: PID parameters are synchronized with nearby agents using a swarm consensus algorithm. Medium threat: PID parameters are optimized using CBFs, local routing, trajectory prediction, and obstacle motion prediction algorithms.

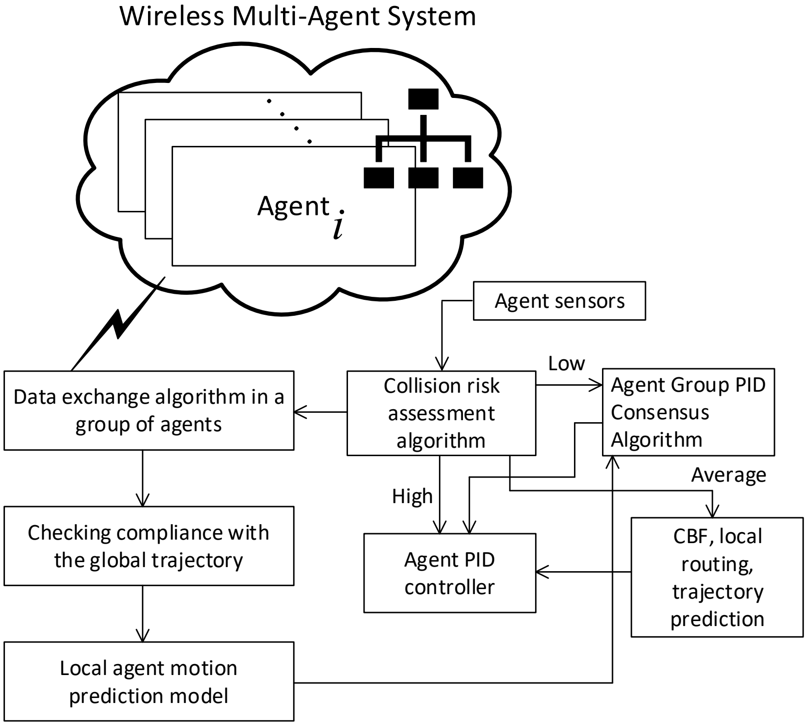

Figure 2: A structural scheme for a multi-agent control and communication system using multi-level optimization

5.2. Hybrid Architecture for Safe UAV Swarm Control

The proposed control system operates on three levels:

Global level – trajectory planning using MPC and Rapidly-exploring Random Tree Star (RRT*).

Local level – collision avoidance based on CBFs and adaptive PID controllers.

Emergency level – RL, specifically Q-learning, for critical maneuvers.

The safety system classifies situations into four risk levels: Normal, Warning, Threat, and Critical.

Main modules and algorithms:

1. Trajectory prediction: Estimates future UAV positions over a 5-step horizon considering control actions and neighbor responses.

Multi-level safety logic: Risk levels determined based on predicted minimum distances, ensuring early responses without false alarms.

Adaptive PID tuning: Kp, Ki, Kd dynamically change according to risk and past maneuver efficiency, preventing conflicts between trajectory tracking and collision avoidance.

Cooperative maneuvering: Coordinated changes in altitude or direction based on priority (agent ID, threat detection time, etc.).

Communication loss handling: In case of signal loss, the agent continues with avoidance maneuvers based on the last known neighbor data, preserving formation after reconnection.

Boundary reaction: Instead of abrupt stopping, UAVs smoothly change direction while staying inside operational space, coordinating maneuvers with other agents.

The proposed hybrid architecture enables a comprehensive multi-level optimization strategy for the safe movement of agent groups.

5.3. Hybrid Strategy for Comprehensive Multi-Level Optimization of Safe Group Motion

The strategy combines various optimization levels:

Global planning – MPC and RRT* build trajectories considering speed constraints and obstacles.

Local optimization – CBF ensures safe distances, while PID provides stabilization and trajectory tracking.

Emergency maneuvers – RL Q-table controls discrete actions in critical situations.

Cooperative coordination – Control parameters are collectively tuned via consensus mechanisms to improve robustness and energy efficiency.

The implemented technologies are summarized in Table 3 for a better understanding of the levels of comprehensive optimization for group safety.

It is also necessary to determine which trajectory computation approach should be used. Both centralized and decentralized computation can be applied, so both are considered:

Centralized – A single controller calculates all trajectories (using MAPF: Multi-Agent Path Finding) to coordinate movement. This is suitable for small swarms but lacks scalability.

Decentralized – Each agent plans its motion using local data and partial information from neighbors. This approach provides scalability and fault tolerance but makes it harder to guarantee global route optimality.

A compromise between centralized and decentralized approaches can be achieved through hierarchical coordination:

Global planning sets goals and routes.

Local agents adapt movement in real time according to threats.

Subgroups of agents coordinate actions via elected leaders.

Table 2

Levels of comprehensive safety optimization for agent motion

Component	Method / Algorithm	Purpose
Barrier Functions (CBF)	Quadratic Programming	Formal collision avoidance
Planning (MPC + RRT*)	Linear model with horizon constraints	Trajectory generation
Physical Model	Second-order equations (position + velocity)	Realistic dynamics
Safety	Risk levels + RL	Early response
Optimization	Gradient descent + swarm algorithm	Parametric tuning

The proposed hybrid architecture for safe UAV swarm control, combining global efficiency with local safety, is a promising direction for the development of secure swarms. Formal methods such as CBF will provide mathematical safety guarantees, while decentralization will ensure scalability and flexibility.

Next steps involve empirical validation, extension to multi-scenario missions, and further reduction of energy consumption. The feasibility and rationality of the proposed approaches can be confirmed through swarm motion simulation and relevant numerical experiments.

6. Numerical Experiments of the Multi-Level Agent Motion Control System and Comparison with Known Group Control Methods

We now present numerical experiments for the multi-level agent motion control system and compare its performance with established algorithms and methods for group control. The study focuses on ensuring the safe motion of agent groups, particularly UAVs, in a “swarm versus swarm” scenario. To evaluate the developed hybrid architecture for UAV swarm safety control, multi-level optimization of safe UAV motion was performed in a "swarm vs swarm" scenario.

6.1. Simulation objective

Assess the efficiency of the combined multi-level optimization strategy for safe and coordinated drone movement in an environment with an opposing swarm.

6.2. Simulation conditions

Two swarms: Blue (friendly) and Red (hostile).

Number of drones: 10 in each swarm.

Simulation field: 100×100 units.

Blue swarm objective: reach a gathering point in sector B, avoiding collisions, threats, and detection.

Red swarm objective: patrol key areas and respond to Blue's approach.

Constraints: jamming zones, no-fly areas, dynamic obstacles.

The structure of multilevel optimization is presented in Table 3.

Table 3
Optimization structure in the simulation

Level	Algorithm	Task
1. Strategic	Genetic Algorithm	Selection of overall swarm trajectory (route segments, target priorities)
2. Tactical	PSO	Distribution of subgroups and individual drone paths within the chosen route
3. Local	A* + Avoidance Rules	Real-time obstacle avoidance based on sensor data

6.3. Evaluation metrics

We used the following metrix: Average distance between drones (swarm cohesion); Number of collisions or entries into risk zones; Time to reach the target; Percentage of detected/intercepted drones; Energy consumption per agent.

The multilevel optimization model, which integrates global strategic planning with local threat avoidance, demonstrates high efficiency in swarm-versus-swarm scenarios. This approach enables drones to achieve mission objectives with a higher probability while simultaneously reducing incidents and resource consumption. Thus, numerical experiments confirmed that the multi-level system achieves zero collisions in dense formations — a key indicator of its effectiveness. This is achieved through the integration of MPC, CBF, adaptive PID, and RL, which provide proactive prediction and formal safety guarantees.

6.4. Comparison with known algorithms

The analysis identified the limitations of existing group control algorithms (Table 4):

PFM – prone to local minima, oscillations near obstacles, and no formal collision avoidance guarantees. Vicsek model – lacks explicit collision avoidance, unpredictable behavior due to noise. PSO – not a direct control algorithm, lacks collision avoidance mechanisms. Boids model – reactive, susceptible to local optima, lacks explicit collision prediction. Unlike these, the proposed system provides formal safety guarantees through CBF and proactive collision avoidance, making it a superior choice for tasks such as "swarm vs swarm". For the simulations conducted in the Google Colab environment, the following algorithms were used: Hybrid CBF-PID algorithm (Control Barrier Function + PID) – a collision-avoidance algorithm that uses optimization constraints to guarantee the maintenance of a safe distance, achieved by combining an adaptive PID controller with barrier functions. Classical collision-avoidance algorithms: Boids, Vicsek, PFM, PSO; Algorithms without collision avoidance: Leader-Follower – one agent (leader) moves toward the target, while the others (followers) attempt to follow; the algorithm lacks explicit collision-avoidance mechanisms. Random-Walk – the simplest algorithm, in which agents move in random directions with speed adjusted to a fixed value. In the simulation, 1,000 agents with movement constraints were used, and 200 steps of the algorithmic suite were executed (Optimization structure in the simulation).

Table 4
Comparison of swarm control algorithms (1,000 agents, 200 steps)

Algorithm	Collisions	Avg. Speed (m/s)	Computation time per agent movement parameter (μs)
CBF-PID	0	3.00	6900.95
Boids	0	-	1.71
Vicsek	0	-	1.67
Potential Field	2605	-	10.68
PSO	85	-	2261.90
Leader-Follower	2203	-	1.64
Random-Walk	1790	-	13.97

The agents followed the following trajectory models: CBF-PID – swarm agents move along smooth parallel lines with guaranteed spacing. Boids – produces "cohesive" movement with random deviations but may create clusters. Vicsek – all agents move in the same average direction. Potential Field – generates curvilinear paths for agents. Leader-Follower – straight-line motion of the leader, with followers trailing behind. Simulation results show that the classical Boids and Vicsek algorithms with collision-control mechanisms can produce high-quality results using minimal computational power in the absence of mobile obstacles. However, they do not provide safeguards against agent collisions. CBF-PID requires significantly higher computational resources but guarantees zero collisions.

7. Conclusions

The implementation of safe, real-time autonomous big scale UAV swarm control is a critical challenge for practical deployment in complex environments. Abandoning standard Boids rules — or significantly modifying them — and developing custom swarm control rules is a justified and potentially highly effective approach for achieving zero collisions while maintaining coordinated swarm motion. One scientific problem requiring resolution is the multi-agent modeling of autonomous local navigation for groups, flocks, and large swarms to address the "swarm vs swarm" scenario.

This work addressed the problem of decentralized collective collision avoidance for a group of agents using predictive planning, adaptive control, and optimization methods with safety guarantees. For the first time, a hybrid control architecture for UAV swarms was proposed, based on comprehensive multi-level optimization that combines global trajectory planning and local collision avoidance with formal safety guarantees. The developed system integrates Model Predictive Control, Control Barrier Functions, adaptive consensus-based PID controllers, and Reinforcement Learning for emergency situations. The proposed approach ensures zero collisions for agent motion in a 3D dynamic environment through multi-level optimization and cooperative maneuvering. We propose a new collision avoidance strategy for swarms of agents that leverages control barrier functions in conjunction with a local adaptive consensus scheme for the dynamic adjustment of PID controller parameters.

In addition, algorithmic and software tools were developed to simulate UAV swarm motion of varying sizes with hybrid control systems. Computational experiments demonstrated zero collisions for tightly packed formations. The result is a comprehensive multi-level optimization system for safe group motion of agents, aimed at solving the collision problem during aggressive maneuvers in contested environments and during obstacle avoidance.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to grammar and spelling check.

References

[1] Artiushyn, L., Kononov, O., Nevzhliadenko, Y.: Analysis of prospects for the implementation of group use of military unmanned aerial vehicles. Collection of Scientific Papers of the State Research Institute of Aviation **19**(26), 42–48 (2023) https://doi.org/10.54858/dndia.2023-19-5

[2] Cheng, N., Wu, S., Wang, X., Yin, Z., Li, C., Chen, W., Chen, F.: AI for UAV-assisted IoT applications: A comprehensive review. IEEE Internet of Things Journal **10**(16), 14438–14461 (2023) https://doi.org/10.1109/JIOT.2023.3268316

[3] Phadke, A., Medrano, F.A., Sekharan, C.N., Chu, T.: Designing UAV swarm experiments: A simulator selection and experiment design process. Sensors **23**(17), 7359 (2023) https://doi.org/10.3390/s23177359

[4] Duan, H., Huo, M., Fan, Y.: From animal collective behaviors to swarm robotic cooperation. National Science Review **10**(5), nwad040 (2023) https://doi.org/10.1093/nsr/nwad040

[5] Scharre, P.: How swarming will change warfare. Bulletin of the Atomic Scientists **74**(6), 385–389 (2018) https://doi.org/10.1080/00963402.2018.1533209

[6] Bourjade, M., Thierry, B., Hausberger, M., Petit, O.: Is leadership a reliable concept in animals? An empirical study in the horse. PLoS One **10**(5), e0126344 (2015) https://doi.org/10.1371/journal.pone.0126344

[7] Priyadarshi, R., Kumar, R.R.: Evolution of swarm intelligence: A systematic review of particle swarm and ant colony optimization approaches in modern research. Archives of Computational Methods in Engineering, 1–42 (2025) https://doi.org/10.1007/s11831-025-10247-2

[8] Poghosyan, S., Poghosyan, V., Abrahamyan, S., Lazyan, A., Astsatryan, H., Alaverdyan, Y., Eguiazarian, K.: Cloud-based mathematical models for self-organizing swarms of UAVs: design and analysis. Drone Systems and Applications 12, 1–12 (2024) https://doi.org/10.1139/dsa-2023-0039

[9] Hulianytskyi, L., Ogurtsov, M., Korolyov, V., Rybalchenko, O., Yarushevskyi, O.: The group movement optimization of autonomous agents in a locally-centric navigation model. In: IV International Scientific Symposium "Intelligent Solutions" (IntSol-2025), 34-56, May 1–5, 2025 https://ceur-ws.org/Vol-4035/Paper4.pdf

[10] Liu, J., Yan, Y., Yang, Y., Li, J.: An improved artificial potential field UAV path planning algorithm guided by RRT under environment-aware modeling: Theory and simulation. IEEE Access **12**, 12080–12097 (2024) https://doi.org/10.1109/ACCESS.2024.3355275

[11] Zhang, L., Zhang, M., Feng, L.: A swarm intelligent model for AVs based on Vicsek and OV model. In: 2024 4th International Conference on Control Theory and Applications (ICoCTA), 101–105. IEEE, October 2024 https://doi.org/10.1109/ICoCTA64736.2024.00029

[12] Liu, Y., Zhang, H., Zheng, H., Li, Q., Tian, Q.: A spherical vector-based adaptive evolutionary particle swarm optimization for UAV path planning under threat conditions. Scientific Reports **15**(1), 2116 (2025) https://doi.org/10.1038/s41598-025-85912-4

[13] Jin, W., Tian, X., Shi, B., Zhao, B., Duan, H., Wu, H.: Enhanced UAV pursuit-evasion using Boids modelling: A synergistic integration of bird swarm intelligence and DRL. Computers, Materials & Continua **80**(3) (2024) https://doi.org/10.32604/cmc.2024.055125

[14] Moali, O., Mezghani, D., Mami, A., Oussar, A., Nemra, A.: UAV trajectory tracking using proportional-integral-derivative-type-2 fuzzy logic controller with genetic algorithm parameter tuning. Sensors **24**(20), 6678 (2024) https://doi.org/10.3390/s24206678

[15] Aliane, N.: A survey of open-source UAV autopilots. Electronics **13**(23), 4785 (2024) https://doi.org/10.3390/electronics13234785

[16] Zhang, QH., Lu, JG. & Zhu, Z. A review on robust control of continuous time fractional-order systems. Nonlinear Dyn (2025). https://doi.org/10.1007/s11071-025-11497-2

[17] Shi, K., Chang, J., Feng, S., Fan, Y., Wei, Z., Hu, G.: Safe human dual-robot interaction based on control barrier functions and cooperation functions. IEEE Robotics and Automation Letters (2024) https://doi.org/10.1109/LRA.2024.3458597

[18] Lee, J., Kim, J., Ames, A.D.: A data-driven method for safety-critical control: Designing control barrier functions from state constraints. In: 2024 American Control Conference (ACC), 394–401. IEEE, July 2024 https://doi.org/10.23919/ACC60939.2024.10644811

[19] Huang, J., Wang, H., Margellos, K., Goulart, P.: Predictive control barrier functions: Bridging model predictive control and control barrier functions. arXiv preprint arXiv:2502.08400 (2025)

[20] Korolyov, V., Ogurtsov, M., Khodzinskyi, O.: Research on algorithms for local robot positioning using methods of discrete optimization. Cybernetics and Computer Technologies **2**, 5–16 (2025) https://doi.org/10.34229/2707-451X.25.2.1

[21] Barhoumi, O., Zaki, M.H., Tahar, S.: A formal approach to road safety assessment using traffic conflict techniques. IEEE Open Journal of Vehicular Technology 5, 606–619 (2024) https://doi.org/10.1109/OJVT.2024.3387414

[22] Software used in an article for multi-agent system simulation. https://www.github.com/novice108/ITT2025