Dynamics and Control Description for a Minimal 3D Swarm Simulation

1 Overview

This document provides a mathematical and physical explanation of the swarm simulation implemented in the function `demo_swarm_run`. The simulation models n unmanned aerial vehicles (UAVs), referred to as drones, moving in three-dimensional space to reach a specified target position while maintaining a coordinated formation. Each drone is modeled as a point mass with second-order dynamics:

$$\dot{\mathbf{p}}_i = \mathbf{v}_i, \qquad \dot{\mathbf{v}}_i = \mathbf{a}_i,$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the position, $\mathbf{v}_i \in \mathbb{R}^3$ is the velocity, and $\mathbf{a}_i \in \mathbb{R}^3$ is the commanded acceleration of drone i. The goal of the control law is to:

- Drive all drones to reach the desired goal point $\mathbf{g} \in \mathbb{R}^3$.
- Maintain a mean inter-drone spacing of L_s.
- Align drone headings so that their velocity vectors become similar.
- Maintain a common target altitude, typically the goal altitude.
- Avoid collisions with static obstacles in the environment.
- Ensure physical limits, including bounded acceleration and speed.

2 Drone Dynamics

For each drone i, the state evolves under a double-integrator dynamic model:

$$\dot{\mathbf{p}}_i = \mathbf{v}_i, \tag{1}$$

$$\dot{\mathbf{v}}_i = \mathbf{a}_i. \tag{2}$$

Numerical time integration is performed using a classical fourth-order Runge–Kutta (RK4) scheme with timestep Δt. At each step, four derivative evaluations are computed:

$$k_1^p = \mathbf{v}_i^n, \qquad k_1^v = \mathbf{a}_i(\mathbf{p}_i^n, \mathbf{v}_i^n), \tag{3}$$

$$k_2^p = \mathbf{v}_i^n + \tfrac{\Delta t}{2} k_1^v, \qquad k_2^v = \mathbf{a}_i(\mathbf{p}_i^n + \tfrac{\Delta t}{2} k_1^p, \mathbf{v}_i^n + \tfrac{\Delta t}{2} k_1^v), \tag{4}$$

$$k_3^p = \mathbf{v}_i^n + \tfrac{\Delta t}{2} k_2^v, \qquad k_3^v = \mathbf{a}_i(\mathbf{p}_i^n + \tfrac{\Delta t}{2} k_2^p, \mathbf{v}_i^n + \tfrac{\Delta t}{2} k_2^v), \tag{5}$$

$$k_4^p = \mathbf{v}_i^n + \Delta t\, k_3^v, \qquad k_4^v = \mathbf{a}_i(\mathbf{p}_i^n + \Delta t\, k_3^p, \mathbf{v}_i^n + \Delta t\, k_3^v), \tag{6}$$

and the state is updated via:

$$\mathbf{p}_i^{n+1} = \mathbf{p}_i^n + \frac{\Delta t}{6}(k_1^p + 2k_2^p + 2k_3^p + k_4^p), \tag{7}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \frac{\Delta t}{6}(k_1^v + 2k_2^v + 2k_3^v + k_4^v). \tag{8}$$

3 Control Law

The commanded acceleration is composed of several physically meaningful terms:

$$\mathbf{a}_i = \mathbf{a}_{\text{goal},i} + \mathbf{a}_{\text{align},i} + \mathbf{a}_{\text{sep},i} + \mathbf{a}_{\text{alt},i} + \mathbf{a}_{\text{obs},i} - c_d \mathbf{v}_i.$$

- $\mathbf{a}_{\text{goal},i}$ is an attraction term that pulls the drone toward the desired target point. It reduces the position error relative to the goal and provides the primary direction of travel.

- $\mathbf{a}_{\text{align},i}$ encourages the drone to match its velocity to the average velocity of the group. This promotes coordinated motion and reduces heading divergence among drones.

- $\mathbf{a}_{\text{sep},i}$ regulates spacing by pushing drones apart when they are too close and drawing them together when they are too far. The reference spacing is L_s, which prevents collisions and maintains loose formation.

- $\mathbf{a}_{\text{alt},i}$ adjusts vertical motion such that the drone maintains a desired altitude, typically chosen as the altitude component of the goal position.

- $\mathbf{a}_{\text{obs},i}$ generates repulsive forces to avoid collisions with static obstacles in the environment using short-range barrier functions.

- $-c_d \mathbf{v}_i$ is a damping term that opposes motion. It prevents overshoot, reduces oscillations, and models a simple aerodynamic-like drag that stabilizes the velocity.

3.1 Goal Tracking Term

Each drone is attracted toward the goal $\mathbf{g}$:

$$\mathbf{a}_{\text{goal},i} = k_g \left(\mathbf{g} - \mathbf{p}_i \right),$$

where k_g scales how strongly drones move toward the target. The terms $\mathbf{g} - \mathbf{p}_i$ represents the position error vector from drone i to the goal.

3.2 Velocity Alignment Term

To encourage coordinated motion, drones adjust their velocity to match the average swarm velocity:

$$\mathbf{v}_{\text{avg}} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{v}_j,$$

$$\mathbf{a}_{\text{align},i} = k_a \left(\mathbf{v}_{\text{avg}} - \mathbf{v}_i \right).$$

3.3 Separation Regulation Term

To maintain spacing, each drone exerts a spring-like force relative to every other drone:

$$\mathbf{a}_{\text{sep},i} = \frac{k_s}{n-1} \sum_{j \neq i} (d_{ij} - L_s)\,\hat{\mathbf{r}}_{ij},$$

where

$$\mathbf{r}_{ij} = \mathbf{p}_j - \mathbf{p}_i, \quad d_{ij} = \|\mathbf{r}_{ij}\|, \quad \hat{\mathbf{r}}_{ij} = \frac{\mathbf{r}_{ij}}{d_{ij}}.$$

If $d_{ij} > L_s$, the term encourages convergence inward; if $d_{ij} < L_s$, it pushes drones apart. The normalization factor $\frac{1}{n-1}$ ensures that the separation effect scales appropriately with swarm size, preventing the magnitude of this term from growing unbounded as more drones are added. This term ensures flock-like spatial cohesion.

3.4 Altitude Regulation Term

Drones are encouraged to maintain a target altitude h^* = goal's z coordinate:

$$\mathbf{a}_{\text{alt},i} = k_h(h^* - p_{i,z}) \cdot \hat{\mathbf{e}}_z,$$

where $\hat{\mathbf{e}}_z = (0, 0, 1)$.

3.5 Obstacle Avoidance Term

For obstacle avoidance, the simulation currently supports finite cylindrical obstacles. Each obstacle is defined by its axis location (c_x, c_y), radius R, and vertical extent $[z_{\min}, z_{\max}]$. For each drone i and obstacle k, the algorithm computes the nearest point on the cylinder surface and applies a repulsive acceleration if the drone is within the influence radius d_{safe}. The nearest point is determined by considering two cases:

- **Lateral surface**: The closest point on the cylindrical side wall, with z clamped to $[z_{\min}, z_{\max}]$.
- **Top/bottom caps**: The closest point on the circular disk at $z = z_{\min}$ or $z = z_{\max}$.

Let d be the distance from drone i to the nearest point on obstacle k, and let $\hat{\mathbf{n}}$ be the outward unit normal from that point. The repulsive force is:

$$f(d) = \begin{cases} \min\left\{k_o\left(\frac{1}{d} - \frac{1}{d_{\text{safe}}}\right), f_{\max}\right\} & \text{if } d < d_{\text{safe}}, \\ 0 & \text{otherwise,} \end{cases}$$

and the obstacle avoidance acceleration for drone i is:

$$\mathbf{a}_{\text{obs},i} = \sum_{k=1}^{N_{\text{obs}}} f(d_{ik})\,\hat{\mathbf{n}}_{ik},$$

where d_{ik} and $\hat{\mathbf{n}}_{ik}$ are the distance and normal direction for drone i relative to obstacle k. This barrier-like potential grows rapidly as the drone approaches the obstacle surface, ensuring avoidance while remaining zero outside the influence region.

3.6 Damping

A linear drag term prevents overshoot and excessive velocity:

$$-c_d \mathbf{v}_i.$$

4 Constraint Enforcement

To keep the system realistic:

- The norm of $\mathbf{a}_i$ is limited:
$$\|\mathbf{a}_i\| \leq a_{\max}.$$
- Drone speed is limited:
$$\|\mathbf{v}_i\| \leq v_{\max}.$$

Both are applied via scaling if necessary. If the computed acceleration or velocity exceeds the limit, it is rescaled to the maximum allowable magnitude while preserving direction.

5 Default Parameters

Table 1 lists the default parameter values used in the simulation.

Table 1: Default simulation and control parameters.

Parameter	Symbol	Default Value
Number of drones	n	5
Desired separation	L_s	5.0 m
Goal attraction gain	k_g	0.02 s^{-2}
Separation gain	k_s	0.8 s^{-2}
Alignment gain	k_a	0.6 s^{-2}
Altitude gain	k_h	0.5 s^{-2}
Damping coefficient	c_d	0.25 s^{-1}
Max acceleration	$a_{\max}$	6.0 $\mathrm{m/s}^2$
Max speed	$v_{\max}$	15.0 m/s
Obstacle repulsion gain	k_o	40.0 $\mathrm{m}^2/\mathrm{s}^2$
Obstacle influence radius	d_{safe}	12.0 m
Obstacle force cap	$f_{\max}$	10.0 $\mathrm{m/s}^2$
Time step	Δt	0.05 s
Goal tolerance	–	5.0 m
Settle steps required	–	20

6 Physical Interpretation and Balance of Objectives

The control law balances multiple competing objectives through the relative magnitudes of the gains. Notably, the separation gain $k_s = 0.8$ is significantly larger than the goal attraction gain $k_g = 0.02$. This hierarchy means that when the swarm is far from the goal, formation maintenance dominates over goal-seeking, resulting in a "cruise then converge" behavior: the swarm first establishes a cohesive formation, travels as a coordinated unit, and only tightens toward the goal as it approaches.

The desired separation L_s serves a dual purpose: it prevents inter-drone collisions (the control law pushes drones apart when $d_{ij} < L_s$) and maintains visual or communication range (it pulls them together when $d_{ij} > L_s$). The damping term c_d is essential for stability, as without it the second-order dynamics could lead to unbounded oscillations around equilibrium configurations.

The obstacle avoidance term acts as a local override: when a drone enters the influence region of an obstacle, the barrier function generates a strong repulsive acceleration that temporarily dominates the other terms, steering the drone away from collision. The cap $f_{\max}$ ensures that obstacle avoidance does not exceed actuator limits.

7 Initial Conditions

When initial positions P_0 are not provided, the simulation places drones on a regular grid in the xy-plane:

$$P_0 = [x_{\text{grid}},\, y_{\text{grid}},\, 20 \cdot \mathbf{1}_n]\,,$$

where $(x_{\text{grid}}, y_{\text{grid}})$ are arranged on a square lattice with 10 m spacing. Initial velocities default to $\mathbf{v}_i = [4, 0, 0]^\top$ m/s for all drones, giving the swarm a uniform forward motion.

8 Simulation Termination

The simulation may end early when all drones satisfy the convergence criterion:

$$\|\mathbf{p}_i - \mathbf{g}\| \leq 5.0 \text{ m} \quad \text{for all } i,$$

and this condition persists for 20 consecutive timesteps (equivalent to 1.0 s at $\Delta t = 0.05$ s). This ensures stable convergence rather than a transient flyby, confirming that the swarm has genuinely settled at the goal rather than momentarily passing through the tolerance sphere.

9 Output

The simulation exports:

- Per-drone time histories of position and velocity as individual CSV files.
- A combined master CSV file containing all drone trajectories.
- A 3D trajectory plot of the swarm with the goal location indicated and obstacles rendered as semi-transparent cylinders.

10 Modeling Assumptions and Limitations

The simulation makes several simplifying assumptions:

- **Point-mass model**: Drones have no orientation dynamics, rotational inertia, or actuator-level modeling. The commanded acceleration is assumed to be instantly achievable (subject to the saturation limits).
- **No inter-drone collision checking**: The separation regulation term is relied upon to prevent collisions. Explicit collision detection between drones is not implemented.
- **All-to-all communication**: The alignment and separation terms assume that each drone has access to the positions and velocities of all other drones. In practice, this would require communication or sensing over potentially large distances.
- **Static obstacles**: Obstacle positions are fixed. Moving or dynamically appearing obstacles are not supported.

Despite these simplifications, the model captures the essential collective behaviors of flocking, goal-seeking, and obstacle avoidance in a computationally efficient framework suitable for educational purposes and preliminary mission planning.

11 Summary

This swarm model couples basic Newtonian point-mass dynamics with biologically inspired flocking behavior, destination-seeking motion, and geometric obstacle avoidance. The interaction terms allow the swarm to move collectively, maintain formation spacing, avoid static hazards, and converge smoothly to the desired target in three-dimensional space. The relative tuning of control gains determines the emergent behavior, balancing cohesion, goal attraction, and safety in a principled manner.