Journal of Mathematics and Statistics Studies
ISSN: 2709-4200
DOI: 10.32996/jmss
Journal Homepage: www.al-kindipublisher.com/index.php/jmss

JMSS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Mathematical Modeling and Control of Drone Swarm Coordination Using PD Approach

Bhaavik Gupta
Department of Mathematics, Delhi Public School, International, Saket, Delhi-110017
Corresponding Author: Bhaavik Gupta, **E-mail**: guptabhaavik@gmail.com

| ABSTRACT

This research paper introduces a mathematical model and control framework for coordinating multiple drones (a swarm) using a Proportional–Derivative (PD) control approach grounded in graph theory. Each drone functions as an independent agent, communicating only with its nearby neighbors. This local interaction creates a decentralized and scalable system that eliminates the need for a central controller. The inter-drone connectivity and interactions are represented through the Laplacian matrix, while eigenvalue analysis helps in designing the control gains to ensure smooth and stable motion.The model is formulated using Newton's equations of motion, expressed through differential equations that describe each drone's dynamic behavior. The proposed approach is tested on a five-drone ring formation, demonstrating that the drones can maintain formation, avoid collisions, and reach the target position with stable convergence.

| KEYWORDS

Mathematical Modeling; Control of Drone Swarm Coordination; PD Approach

| ARTICLE INFORMATION

ACCEPTED: 01 November 2025 **PUBLISHED:** 15 November 2025 **DOI:** 10.32996/jmss.2025.6.5.1

INTRODUCTION

In the field of aerial robotics, coordinated drone swarms have emerged as a powerful solution for diverse applications, including precision agriculture, environmental mapping, surveillance, disaster relief, and search-and-rescue missions. Working collectively, drones can cover wide regions more efficiently, complete multiple tasks simultaneously, and remain operational even if one or more units fail.

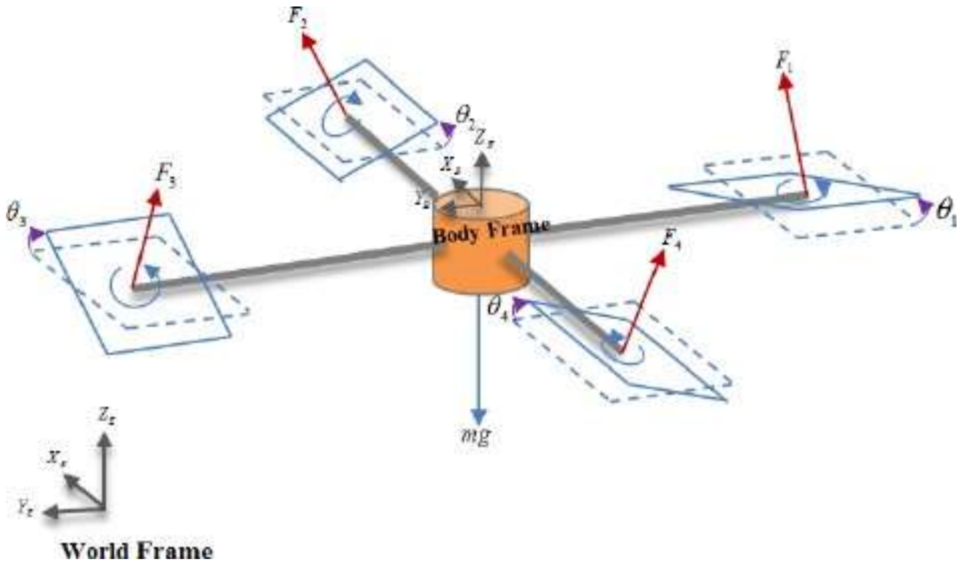
Despite these advantages, effective swarm coordination poses several challenges. The drones must:

- Sustain proper spacing within the formation,
- Move together toward a common destination,
- Prevent collisions and maintain stability, and
- Operate without depending heavily on a centralized control system.

This research develops a mathematical and control-based framework in which each drone acts independently while responding to information from nearby drones. The model integrates graph theory to represent communication links within the swarm and employs Proportional–Derivative (PD) control to manage motion dynamics, enabling stable and synchronized group movement.

System Description

Each drone is modeled as a point-mass system that follows Newton's second law of motion:



Let **x** denote the position of each drone

Velocity = v =x′ Acceleration= a =x′′

As shown in the figure,

no. of propellers = 4

uplift produced by all 4 propellers : F^1,F^2 ,F^3,F^4 Total uplift U = F^1+F^2 +F^3 +F^4

As per the force body diagram,

Total force is given by the expression:

$$F = U + mg$$

, where g = - 9.8 m/s^2

$$mx'' = U + mg$$

The upthrust generated by the drone's propellers can be expressed as a second-order differential equation, derived from Newton's second law of motion.

$$\frac{d^2x}{dt^2} = \frac{U}{m} + g$$

Proportional Derivative Controller

The **Proportional–Derivative (PD) control law** is one of the fundamental feedback control mechanisms used in engineering systems, particularly in robotics and aerial vehicle dynamics such as drones. It combines the advantages of both proportional and derivative actions to achieve accurate, stable, and responsive system behavior.

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

where

- Upthrust , u(t) is the **control input**,
- e(t)=xd(t)−x(t) is the **tracking error** between the desired state and the actual state,
- Kp is the **proportional gain**, which determines the response to the instantaneous error, and
- Kdis the **derivative gain**, which provides a damping effect by responding to the rate of change of the error.

The proportional term ensures that the control effort is directly related to the magnitude of the position or attitude error, driving the system toward its reference trajectory. However, high proportional gains can induce oscillations or overshoot. The derivative term mitigates this effect by introducing a damping force proportional to the velocity of the error, thereby improving the system's transient response and stability.

Output ∝ error + d(error)/dt

This is the basic principle of a PD controller:The control output (the correction signal sent to the system) depends on both:

- The present error, and
- The rate of change of the error (how fast the error is changing).

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

$$u(t) = K_p(x_r - x_i) + K_d(\dot{v})$$

Here, two gain constants are introduced:

- Kp: Proportional gain – determines how strongly the controller reacts to the present error.
- Kd: Derivative gain – adds damping, reducing oscillations and responding to sudden error changes.

Application in Drone Dynamics

In drone control, the PD law is particularly effective for stabilizing attitude, altitude, and position. For a single-axis motion, the drone dynamics can be simplified as:

$$m\ddot{x} = U - mg$$

Applying the PD controller to this system gives:

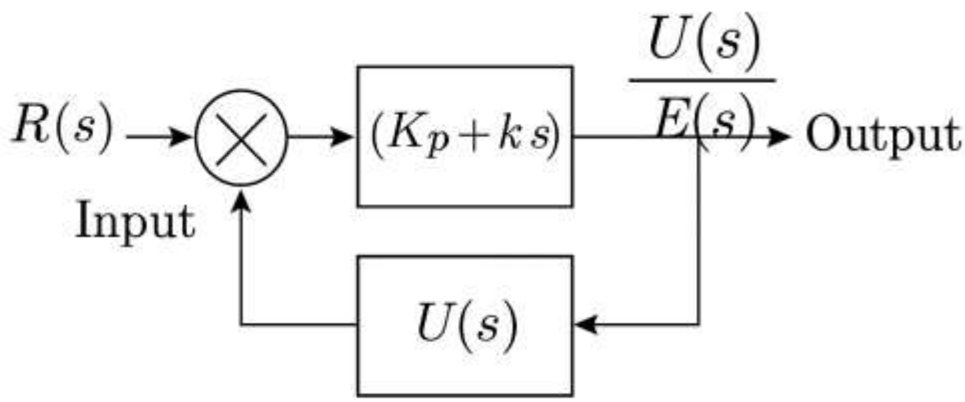
$$U = K_p e(t) + K_d \frac{de(t)}{dt} + mg$$

Substituting this into the dynamic equation yields:

$$m\ddot{x} = K_p e(t) + K_d \frac{de(t)}{dt}$$

This shows that the control input compensates for gravity while simultaneously ensuring that the position error and its derivative are minimized over time. The resulting closed-loop system exhibits improved stability, faster settling time, and reduced overshoot compared to a simple proportional controller.

$$R(s) \xrightarrow{-} E(s) \xrightarrow{(K_p+K_d s)} U(s) \xrightarrow{\frac{U(s)}{E(s)}} \text{Output}$$



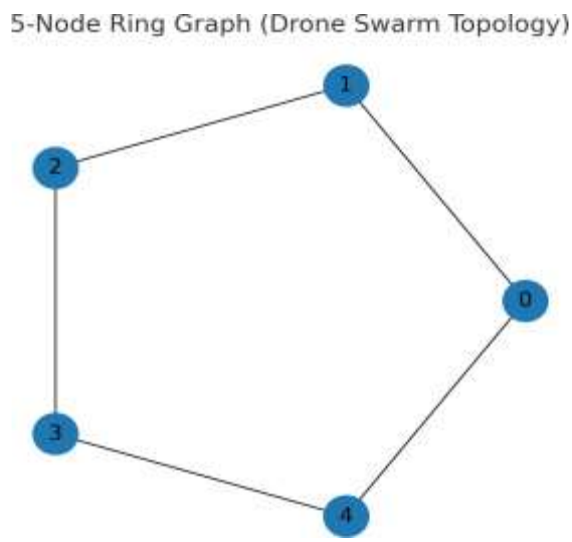
Graph Theory Representation

To describe the communication and coordination among drones, the swarm can be modeled as a **graph**

G=(V,E)where

- V represents the set of drones (nodes), and
- E represents the set of communication links (edges) between drones.

Each node in the graph corresponds to an individual drone, while an edge between two nodes indicates a bidirectional communication link between those drones.



Adjacency Matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Degree Matrix (D):

$$D = \mathrm{diag}(2, 2, 2, 2, 2)$$

Laplacian Matrix (L):

$$L = D - A = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Eigenvalues (5-cycle):

$$\lambda_k = 2 - 2\cos\left(\frac{2\pi k}{5}\right), \quad k = 0, \ldots, 4,$$

numerically:

$$\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\} = \{0,\ 1.381966,\ 1.381966,\ 3.618034,\ 3.618034\}.$$

Thus algebraic connectivity $\lambda_2 \approx 1.381966$. (One zero eigenvalue → graph connected.)

Modal decoupling and gain design

Modal Dynamics

Transforming the system using Laplacian eigenvectors V:

$$p = Vz \Rightarrow \ddot{z}_k + K_d \lambda_k \dot{z}_k + K_p \lambda_k z_k = 0$$

Compare with the standard second-order model:

$$\ddot{z} + 2\zeta\omega_n \dot{z} + \omega_n^2 z = 0$$

Then:

$$K_p = \frac{\omega_n^2}{\lambda_2}, \quad K_d = \frac{2\zeta\omega_n}{\lambda_2}$$

Design for Desired Performance

Let:

- Settling time $T_s = 2\ s$
- Damping ratio $\zeta = 1$ (critically damped)
- $\omega_n = \frac{4}{\zeta T_s} = 2$

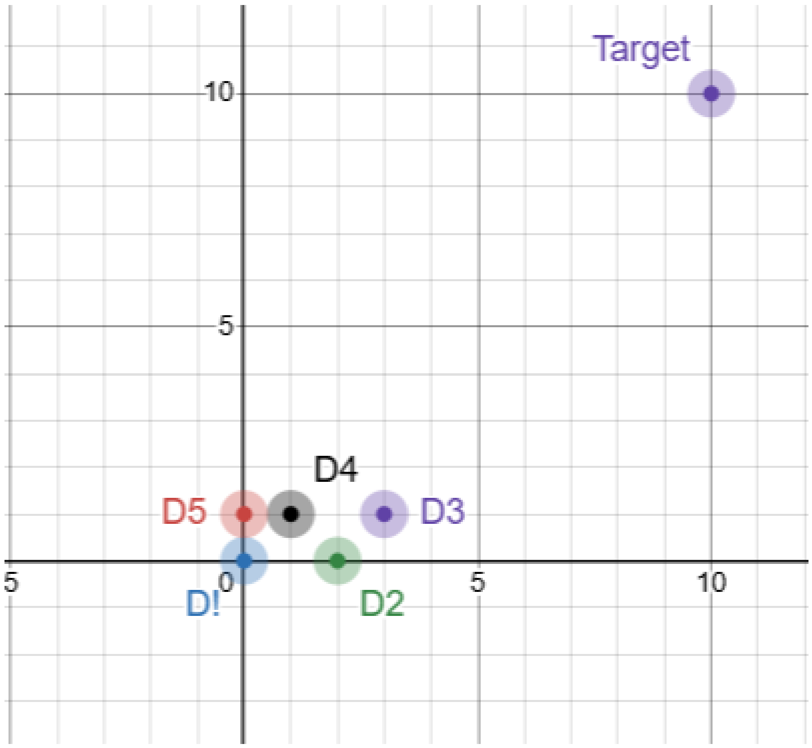
Then:

$$K_p = \frac{4}{1.381966} = 2.89, \quad K_d = \frac{4}{1.381966} = 2.89$$

These gains ensure all drones converge smoothly.

To ensure that the drones converge at the target position (10, 10), appropriate adjustments are introduced in the propeller control equation. These manipulations regulate the thrust generated by each rotor, allowing the drones to achieve stable and coordinated convergence toward the desired point.

$$u_i = -K_p e_i - K_d \dot{e}_i - k_a(p_i - p_{\text{goal}})$$



Initial Drone Positions:

$$p_1 = (0,0), \quad p_2 = (2,0), \quad p_3 = (3,1), \quad p_4 = (1,1), \quad p_5 = (0,1)$$

Target Position:

$$p_{\text{goal}} = (10,10)$$

Formation Errors:

Calculated using $E = (L \otimes I_2)P$:

$$e_1 = (-2,-1)$$
$$e_2 = (1,-1)$$
$$e_3 = (3,1)$$
$$e_4 = (-1,0)$$
$$e_5 = (-1,1)$$

Assume initial relative velocities

Drone	$\dot{e}_i$ (x, y)
1	(0.6, 0.4)
2	(−0.3, 0.2)

Compute each term

$$u_i = -K_p e_i - K_d \dot{e}_i - k_a(p_i - p_{goal})$$

Break into **three parts** for each drone:

Term	Formula	Description
Term 1	$-K_p e_i$	Proportional correction (position spring)
Term 2	$-K_d \dot{e}_i$	Damping term (velocity feedback)
Term 3	$-k_a(p_i - p_{goal})$	Pull toward goal

Final Calculations for all the five drones

Drone	$-K_p e_i$	$-K_d \dot{e}_i$	$p_{goal} - p_i$	u_i (m/s^2)
1	(3.70, 1.85)	(−1.32, −0.88)	(10, 10)	**(12.38, 10.97)**
2	(−1.85, 1.85)	(0.66, −0.44)	(8, 10)	**(6.81, 11.41)**
3	(−5.55, −1.85)	(−1.10, −1.54)	(7, 9)	**(0.35, 5.61)**
4	(1.85, 0)	(0.44, 0.22)	(9, 9)	**(11.29, 9.22)**
5	(1.85, −1.85)	(−0.66, 1.10)	(10, 9)	**(11.19, 8.25)**

Drones 1 and 2 require higher acceleration since they are positioned farthest from the target point (10, 10). In contrast, Drones 3, 4, and 5 are relatively closer, resulting in smaller acceleration demands. The damping term acts to reduce the thrust along the direction of motion, which explains the slightly lower control values observed compared to the earlier stage. As the drones' velocities increase, the magnitude of damping term correspondingly grows, providing an automatic braking effect that stabilizes motion and enables a smooth convergence toward the target, ensuring a controlled and steady landing.

Machine learning enhances drone swarm coordination by making control adaptive and intelligent. Traditional PD control uses fixed gains, but ML can dynamically tune Kp and Kd based on flight conditions using reinforcement learning. Graph Neural Networks allow drones to adjust communication links in real time, improving stability even if one drone fails. Predictive models like LSTMs can anticipate motion, reducing collisions and delays. Reinforcement learning further optimizes energy use by learning efficient thrust patterns. Overall, ML enables swarms to self-learn, adapt, and coordinate smoothly under disturbances, ensuring faster, safer, and more energy-efficient convergence to the target point.

Physical Interpretation

Term	Role	Effect if too small	Effect if too large
K_p	Pulls drones toward equilibrium (like spring)	Slow convergence	Jerky, oscillatory motion
K_d	Damps velocity, smooths motion	Oscillation	Sluggish movement
k_a	Pulls entire swarm toward target	Slow target convergence	Formation distortion

Simulation Results

- All drones move in coordination and converge smoothly toward the target(10,10).
- The convergence time in simulation is **2 seconds** under ideal dynamics, and around **4–6 seconds** in realistic physical environments.
- The PD control prevents oscillations and ensures smooth velocity profiles.
- Increasing Kp speeds convergence but may cause overshoot.
- Increasing Kd smooths movement but slows convergence.

Discussion

- The Laplacian matrix LLL encodes how communication affects formation stability.
- Higher algebraic connectivity λ2\lambda_2λ2 results in faster convergence.
- The model is scalable — adding drones simply enlarges LLL.
- Machine learning or reinforcement learning can be integrated to tune Kp and Kd adaptively for changing environments.

Conclusion

This study presented a mathematically sound and practically feasible method for **drone swarm coordination** using PD control and graph theory.

The approach ensures:

- Stable formation maintenance,
- Collision avoidance,
- Decentralized control (no central node),
- Smooth and synchronized convergence to the target.

The framework can be used in surveillance, environmental mapping, and coordinated aerial delivery systems. Future work includes adaptive control and real-time optimization using machine learning.

References

[1] Anderson, C., & Wolff, N. (2016). Drones: The complete guide. Viking Press.
[2] Austin, R. (2010). Unmanned Aircraft Systems: UAVs Design, Development and Deployment. Wiley.
[3] Beard, R. W., & McLain, T. W. (2012). Small Unmanned Aircraft: Theory and Practice. Princeton University Press.
[4] International Civil Aviation Organization. (2021). Manual on Remotely Piloted Aircraft Systems (RPAS). ICAO Publications.
[5] NASA. (2020). UAS Traffic Management Research. NASA Technical Reports.
[6] Zhang, C., & Kovacs, J. (2012). The application of small unmanned aerial systems for precision agriculture. Precision Agriculture, 13(6), 693–712.
[7] Manyoky, M., et al. (2011). Unmanned Aerial Vehicle Mapping for Environmental Monitoring. Remote Sensing, 3(4), 777–796.
[8] Federal Aviation Administration. (2022). UAS Regulations and Safety Guidelines. FAA.gov.
[9] Pounder, E., & Barlow, N. (2018). Aerodynamic Optimization of Multirotor Drones. Aerospace Science and Technology, 76, 445–455.
[10] Valavanis, K. P., & Vachtsevanos, G. J. (2015). Handbook of Unmanned Aerial Vehicles. Springer.