

International Journal of Systems, Control and Communications

ISSN online: 1755-9359 - ISSN print: 1755-9340
<https://www.inderscience.com/ijsc>

Distributed collaborative control of unmanned aerial vehicle formation based on fuzzy autocoupling PID algorithm

Yijin Cheng

DOI: [10.1504/IJSCC.2026.10081653](https://doi.org/10.1504/IJSCC.2026.10081653)

Article History:

Received:	26 January 2026
Last revised:	15 May 2026
Accepted:	31 August 2026
Published online:	19 October 2026

Distributed collaborative control of unmanned aerial vehicle formation based on fuzzy autocoupling PID algorithm

Yijin Cheng

School of Aeronautical Engineering,
Henan Forestry Vocational College,
Luoyang 471000, Henan, China
Email: 18237956733@163.com

Abstract: In order to solve the problems of high total variation and high TV value in traditional control methods, a distributed collaborative control method of unmanned aerial vehicle formation based on fuzzy autocoupling PID algorithm is proposed. A nonlinear dynamic model of quadcopter unmanned aerial vehicles was established, and a virtual spring damping network was introduced to construct an elastic constrained unmanned aerial vehicle formation system model. This model can describe the compliant deformation and autonomous recovery capability of the formation. A multi parameter fuzzy auto coupled PID distributed controller is designed to address the complex control problem of coupling and superposition between internal dynamics and external formation in the model, thereby enhancing the adaptive and decoupling capabilities of the controller. Experimental results show that the proposed method can significantly reduce the total variation of control inputs, with the highest TV value being only 39.2, demonstrating better control smoothness and superiority.

Keywords: fuzzy autocoupling PID algorithm; unmanned aerial vehicle formation; distributed collaborative control; virtual spring damping network.

Reference to this paper should be made as follows: Cheng, Y. (2026) 'Distributed collaborative control of unmanned aerial vehicle formation based on fuzzy autocoupling PID algorithm', *Int. J. Systems, Control and Communications*, Vol. 17, No. 10, pp.1–18.

Biographical notes: Yijin Cheng received his Bachelor of Engineering in Industrial Engineering from the Tongji University, Shanghai, China, in 2009. He is currently a Lecturer at the Henan Forestry Vocational College, Luoyang, China. His academic work focuses on both teaching and research, with core courses covering mechanical design and electrical control. His primary research interests include drone (UAV) application and assembly technology and automotive manufacturing and testing technology, where he has conducted in-depth studies on field-oriented technical optimisation and practical engineering applications.

1 Introduction

Drones have rapidly penetrated into the civilian sector due to their significant advantages such as low cost, manoeuvrability, and zero personnel risk. However, a single drone has natural physical limitations in terms of mission execution capability, operational

coverage, and system robustness (Hasegawa et al., 2025). Inspired by high-order collaborative behaviours such as bird flocks and fish schools in nature, drone formation collaboration can effectively break through individual bottlenecks and achieve a ‘ $1 + 1 > 2$ ’ system emergence effect through organic cooperation among multiple aircraft. Specifically, the formation system is capable of parallel execution of tasks such as large-scale reconnaissance, multi angle stereo monitoring, and collaborative handling of complex objects, and significantly improves task success rate and system survivability through functional redundancy (Manaa et al., 2025). Among numerous formation control architectures, distributed collaborative control has become the mainstream research direction due to its high scalability, reliability, and flexibility (Li, 2025). Unlike centralised control that requires a central node, distributed control relies on limited local information exchange between machines, with each drone making autonomous decisions based solely on neighbour information. This architecture naturally fits the dynamic and self-organising application scenarios of drone clusters, avoiding the risk of single point failures (Alshdadi and Irshad, 2024), and making it easy to achieve dynamic reconstruction and scale expansion of the formation. Therefore, researching high-performance distributed collaborative control algorithms is a key core technology for achieving intelligent and practical unmanned aerial vehicle clusters. The core of distributed collaborative control for drone formations lies in how to design local control laws that enable all individuals to achieve and maintain the desired group configuration and motion state solely relying on neighbourhood interactions, in the absence of global centralised coordination.

Scholars at home and abroad have conducted in-depth research around this goal. For example, Yang et al. (2025) proposed a quadcopter unmanned aerial vehicle control method that combines multi-scale target tracking algorithms with second-class fuzzy control technology. Firstly, a multi-scale object detection technique based on kernel correlation filters was adopted, which can effectively respond to changes in target size and position through multi-scale analysis; Secondly, the second type of fuzzy control technology is used to handle the uncertainty in the control process, ensuring that the drone can accurately adjust its flight state when tracking the target. However, although type 2 fuzzy controllers can handle stronger uncertainties, their high dependence on expert experience leads to insufficient generalisation ability of the controller. Domoshnitsky et al. (2025) proposed a distributed feedback control method based on integral operators for achieving angular stability control of drone motion. Study the stability of integral differential equations in the case where the integral operator has infinite memory effect. However, the infinite memory of the integral operator means that all historical errors need to be considered, which is not achievable in digital systems and can only be approximated through a finite window. How to choose an effective memory length to achieve the best balance between stability, anti integral saturation ability, and correction effect on long-term deviations is a key challenge in practical applications. Xu et al. (2024) designed OmniDrones, an efficient and flexible reinforcement learning platform for drone control. The platform is built on Nvidia’s Omniverse Isaac Sim and adopts a bottom-up design philosophy. However, deep reinforcement learning typically requires massive amounts of interactive data, the training process is extremely time-consuming, and the computational cost is high. For complex drone swarm collaborative tasks, the state action space is huge; the exploration difficulty is high, which may lead to unstable training, slow convergence, and even difficulty in learning effective strategies. Mademlis et al. (2024) proposed a linear PID based motion control method for

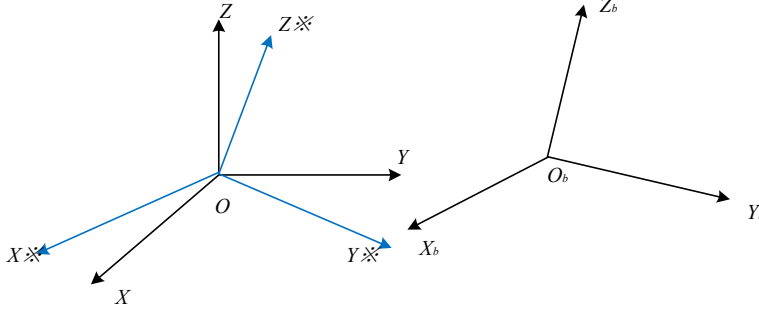
unmanned aerial vehicles, which proposes a new framework that can rephrase each required motion type as a set of interrelated requirement conditions, including 2D visual information, unmanned aerial vehicle motion trajectory, and camera orientation. By utilising these required conditions to generate appropriate error signals, visual information-based PID control algorithms can be implemented. However, this method is essentially a PID control framework based on visual error feedback. The ability of linear PID structures is limited in the face of strong nonlinearity, strong coupling, and parameter uncertainty in unmanned aerial vehicle mechanics. Although task decoupling has been achieved through ‘requirement conditions’, it has not fundamentally solved the problem of insufficient robustness of PID in dealing with complex dynamic models and external strong disturbances, and its adaptability is relatively weak.

In the field of distributed collaborative control of drone formations, traditional control methods mainly face the following key problems: firstly, the drone dynamics model has strong nonlinearity, strong coupling, and parameter uncertainty, and traditional linear controllers are difficult to maintain consistent dynamic response quality under all operating conditions; Secondly, there are complex aerodynamic coupling and communication constraints between aircraft in formation flight, and external disturbances and internal state coupling overlap, making it difficult for a single fixed parameter control strategy to balance the accuracy and flexibility of formation maintenance; Thirdly, existing methods often require precise system models or a large amount of offline training data when dealing with formation reconstruction or sudden disturbances, resulting in insufficient generalisation ability and real-time performance; Fourthly, many control strategies neglect the smoothness of control inputs while pursuing tracking accuracy, resulting in excessive overall control variation and exacerbating actuator wear and energy consumption. In response to the above issues, this article proposes a distributed collaborative control method based on fuzzy self coupling PID algorithm, which systematically solves the problem from three aspects: modelling, control structure, and parameter self-tuning. At the modelling level, a virtual spring damping network is introduced to construct a drone formation system model based on elastic constraints. This model does not require precise aerodynamic modelling between aircraft, but only generates elastic recovery force through the error between the expected distance and the actual distance, making the formation system naturally capable of flexible deformation and autonomous recovery, thereby reducing the dependence on accurate models and offline training. At the controller structure level, a multi parameter self coupling PID framework is designed to tightly couple the proportional, integral, and derivative stages through a velocity factor, forming a unified collaborative control force, avoiding the complexity of independent tuning of the three parameters, and enhancing the system’s ability to suppress uncertain disturbances. At the parameter adaptive level, fuzzy logic is introduced to adjust the speed factor online and dynamically change the controller gain based on real-time tracking error: when the error is large, the proportional effect is enhanced to respond quickly, and when the error is small, the gain is weakened to avoid overshoot and oscillation. This dual adaptive mechanism enables the controller to adapt to strong nonlinearity and strong coupling dynamics without the need for an accurate system model, while significantly reducing the total variation of control inputs through smooth control instructions, balancing the speed, stability, and economic efficiency of formation tracking and actuators.

2 Establishment of unmanned aerial vehicle dynamics model

Due to the study of the transformation from the drone body coordinate system to the ground coordinate system in this article (Faki, 2025), only the translation and rotation of the coordinate system need to be considered. The ground coordinate system $OXYZ$ and the body coordinate system $O_bX_bY_bZ_b$ are shown in Figure 1.

Figure 1 Schematic diagram of three-dimensional space translation (see online version for colours)



Converting the $O_bX_bY_bZ_b$ coordinate system to the $OXYZ$ coordinate system requires one translation and three rotations (Kang and Kim, 2025). The specific operation is as follows:

1 Translate

The translation operation is shown in Figure 1. After coordinate translation, the body coordinate system $O_bX_bY_bZ_b$ is transformed into a transitional coordinate system $O' = \{X', Y', Z'\}$. At this point, the origin of the body coordinate system coincides with the origin of the ground coordinate system (Meguro et al., 2025), and the transformation relationship can be expressed as:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X_0 + X_b \\ Y_0 + Y_b \\ Z_0 + Z_b \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} \quad (1)$$

Among them, $[X_0 Y_0 Z_0]^T$ represents the unit vector of the ground coordinate system.

2 Rotate along the X' axis at an angle of φ .

By rotating the roll angle φ from the transition coordinate system O' to the transition coordinates $O'' = \{X'', Y'', Z''\}$, we can obtain:

$$\begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} = \begin{bmatrix} X' \\ Y' \cos \varphi + Z' \sin \varphi \\ -Y' \sin \varphi + Z' \cos \varphi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (2)$$

3 Rotate the angle σ along the Y' axis.

The pitch angle σ from the transition coordinate system O'' to the transition coordinate system $O''' = \{X''', Y''', Z'''\}$ can be described as:

$$\begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} = \begin{bmatrix} X'' \cos \sigma - Z'' \sin \sigma \\ Y'' \\ X'' \sin \sigma + Z'' \cos \sigma \end{bmatrix} = \begin{bmatrix} \cos \sigma & 0 & -\sin \sigma \\ 0 & 1 & 0 \\ \sin \sigma & 0 & \cos \sigma \end{bmatrix} \begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} \quad (3)$$

4 Rotate the angle ϕ along the Z' axis.

The rotation of the yaw angle ϕ from the transition coordinate system O''' to the ground coordinate system $OXYZ$ can be expressed as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X''' \cos \phi - Z''' \sin \phi \\ -X''' \sin \phi + Z''' \cos \phi \\ Z''' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} \quad (4)$$

Simultaneous formulas (2) to (4), the complete rotation from the airframe coordinate system to the ground coordinate system can be obtained by sequentially left multiplying each rotation matrix to obtain $R_{b2e} = R_z(\phi)R_y(\sigma)R_x(\varphi)$, first calculating $R_{temp} = R_y(\sigma)R_x(\varphi)$, and then left multiplying $R_z(\phi)$ to obtain the final rotation matrix R_{b2e} , which

$$R_{b2e} = R_z(\phi)R_{temp} = \begin{bmatrix} \cos \sigma \cos \phi & \sin \phi \sin \sigma \cos \phi & \cos \phi \sin \sigma \cos \phi + \sin \phi \sin \phi \\ \cos \sigma \sin \phi & \sin \phi \sin \sigma \sin \phi & \cos \phi \sin \sigma \sin \phi - \sin \phi \cos \phi \\ -\sin \sigma & \sin \phi \cos \sigma & \cos \phi \cos \sigma \end{bmatrix} \quad (5)$$

Among them, R_{b2e} represents the transformation matrix from the body coordinate system to the ground coordinate system, which has orthogonality.

The motion parameters of a quadcopter drone include six, three of which are the positions (x, y, z) in the ground coordinate system, and the other three are the attitude of the quadcopter relative to the ground coordinate system (Chang et al., 2024), including roll angle φ , pitch angle σ , and yaw angle ϕ . Starting from the rotation of a single propeller, analyse and establish the motion equation of a quadcopter unmanned aerial vehicle (Abott, 2024). The relationship between the thrust provided by the four propellers and the lift F_{Z_b} on the Z_b axis in the body coordinate system can be expressed as:

$$F_{Z_b} = F_1 + F_2 + F_3 + F_4 \quad (6)$$

Combining the three-dimensional coordinate system transformation matrix R_{b2e} , the force form of the unmanned aerial vehicle converted from the body coordinate system to the ground coordinate system is:

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} = R_{b2e} \begin{bmatrix} F_{X_b} \\ F_{Y_b} \\ F_{Z_b} \end{bmatrix} = \begin{bmatrix} \cos \phi \sin \sigma \cos \phi + \sin \phi \sin \phi \\ \cos \phi \sin \sigma \sin \phi - \sin \phi \cos \phi \\ \cos \phi \cos \sigma \end{bmatrix} F_{Z_b} \quad (7)$$

According to Newton's second theorem, the following relationship can be obtained:

$$\begin{aligned} F_X &= m\ddot{x} = (\cos \varphi \sin \sigma \cos \phi + \sin \varphi \sin \phi) F_{Z_b} \\ F_Y &= m\ddot{y} = (\cos \varphi \sin \sigma \sin \phi - \sin \varphi \cos \phi) F_{Z_b} \\ F_Z &= m\ddot{z} = (\cos \varphi \cos \sigma) F_{Z_b} - mg \end{aligned} \quad (8)$$

Among them, m represents the mass of the drone; g represents the gravitational acceleration (Weon et al., 2024). The dynamic model of a quadcopter drone can be described as follows:

$$\left\{ \begin{aligned} \ddot{x} &= \frac{(\cos \varphi \sin \sigma \cos \phi + \sin \varphi \sin \phi) F_{Z_b}}{m} \\ \ddot{y} &= \frac{(\cos \varphi \sin \sigma \sin \phi - \sin \varphi \cos \phi) F_{Z_b}}{m} \\ \ddot{z} &= \frac{(\cos \varphi \cos \sigma) F_{Z_b}}{m} - g \\ \ddot{\phi} &= \frac{L_A (-F_1 + F_2 + F_3 - F_4)}{I_x} \\ \ddot{\sigma} &= \frac{L_A (F_1 + F_2 + F_3 - F_4)}{I_y} \\ \ddot{\varphi} &= \frac{(-M_1 + M_2 - M_3 + M_4)}{I_z} \end{aligned} \right. \quad (9)$$

Among them, L_A is the distance from the centre of gravity of the drone to the propeller, and I_x , I_y , and I_z respectively represent the moment of inertia along the lower X_b axis, Y_b axis, and Z_b axis. M_i ($i = 1, 2, 3, 4$) are the torques generated by the rotation of four propellers (Ili et al., 2024).

The quadcopter drone model is relatively complex. This article mainly studies the establishment of a drone formation system through a single drone and investigates its collaborative tracking control problem (Cetinsaya et al., 2024). The drones all use quadcopters, and their dynamic models can be expressed as:

$$\left\{ \begin{aligned} \ddot{x}_i &= (\cos \varphi_i \sin \sigma_i \cos \phi_i + \sin \varphi_i \sin \phi_i) \frac{u_z}{m} \\ \ddot{y}_i &= (\cos \varphi_i \sin \sigma_i \sin \phi_i - \sin \varphi_i \cos \phi_i) \frac{u_z}{m} \\ \ddot{z}_i &= \cos \varphi_i \cos \sigma_i \frac{u_z}{m} - g \end{aligned} \right. \quad (10)$$

Among them, x_i , y_i , and z_i respectively represent the position coordinates of the i drone in space, ϕ_i , σ_i , and φ_i respectively represent the yaw angle (Belfadel et al., 2024), pitch angle, and roll angle of the i drone, and u_z represents the total thrust F_{Z_b} on the Z axis.

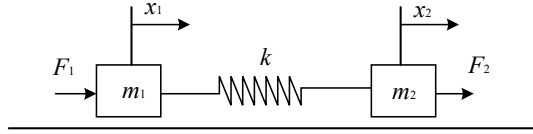
3 Modelling of unmanned aerial vehicle formation based on elastic constraints

The second section established a nonlinear dynamic model of a single unmanned aerial vehicle, providing a basic controlled object for the research of formation cooperative control. However, the core of drone formation lies in the collaborative interaction between multiple intelligent agents. To achieve distributed, adaptive, and compliant formation control, it is necessary to extend the individual model to the swarm system. On the basis of the single machine dynamics model described in formula (10) (Wang et al., 2024), this section introduces a virtual spring damping network to describe the elastic interaction between unmanned aerial vehicles, thereby establishing a model of unmanned aerial vehicle formation system based on elastic constraints. This model not only defines the expected equilibrium formation, but also characterises the compliant deformation and autonomous recovery ability of the formation under internal and external disturbances, providing an accurate mathematical model for subsequent collaborative controller design.

3.1 Elastic system model

Firstly, establish a system consisting of only two mass bodies and springs as shown in Figure 2, introduce the traditional elastic system model, and conduct force analysis on the system to establish a mathematical model of the system (Lai et al., 2025).

Figure 2 Spring system model



In this model, we use a system consisting of two objects with a mass of m_1 , m_2 and a spring with an elastic coefficient of k_f for analysis. Firstly, x_1 and x_2 represent the position of the object at this moment, and the relative distance between x_1 and x_2 is x_{12} , expressed as $x_{12} = x_1 - x_2$, with x_{12} and x_{21} having the same value. x_{12}^d is the expected relative distance between m_1 and m_2 , and $x_{12}^d = x_{21}^d$.

If there is an error between the relative distance between two objects and the expected distance, it will result in an expected position error. Due to the existence of the error, the spring in the system will undergo elastic deformation. The existence of the elastic coefficient of the spring will generate a relative constraint force between the two substances, which can be used to adjust the two objects towards the expected distance through elastic constraint force while maintaining system stability after reaching the equilibrium point.

F_1 and F_2 are the external forces applied to objects respectively. When an external force is applied to one object in the system, in order to maintain the balance of the entire system and keep the actual distance between the two objects unchanged from the expected distance, the elastic force will cause the other object to move towards the desired position. After analysing the overall force of the system, a single object in the system is subjected to the combined effect of the inertial force generated by a motion and

the elastic force generated by a spring. The equation can be obtained as the system model:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} m_2 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + k_f \begin{bmatrix} x_{12}^d - (x_1 - x_2) \\ x_{21}^d - (x_2 - x_1) \end{bmatrix} \quad (11)$$

Assuming that the mass between two objects is the same and expressed as $m_1 = m_2 = m$, after system analysis, to establish the connection between the two systems,

$(\ddot{x}_2 - \ddot{x}_1) = \frac{F_2 - F_1}{m} + \frac{2k_f}{m}(x_2 - x_1)$ can be obtained using $F_2 - F_1$. Based on this, the state

space equation of the elastic system is obtained as follows:

$$\begin{cases} \dot{\eta} = A\eta + Bu \\ P_{ij} = C\eta \end{cases} \quad (12)$$

Among which, $u = \frac{F_2 - F_1}{m}$, $A = \begin{bmatrix} 0 & 1 \\ \frac{2k_f}{m} & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\eta = \begin{bmatrix} x_2 - x_1 \\ \dot{x}_2 - \dot{x}_1 \end{bmatrix}$.

After analysing formula (12) as the state space equation of the traditional elastic system model, the same analysis method is used to model the drone swarm formation and establish a drone swarm formation with elastic constraints.

3.2 Modelling of drone formation based on elastic system model

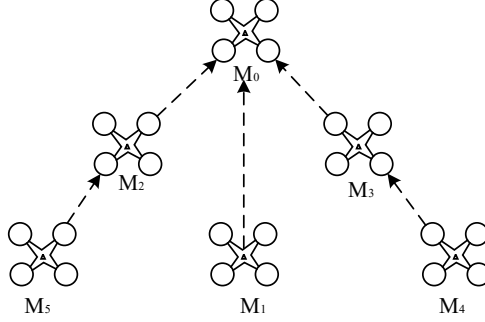
When the actual distance between two drones differs too much from the expected distance, the size of the formation elastic constraint strength K_f can adjust the speed of drone adjustment, but it also affects the stability of the drone when it reaches the equilibrium point. An excessively large K_f value will make the drone follow too quickly, and after reaching equilibrium, it is difficult to control due to the large control force. Therefore, adjusting an appropriate K_f value can better achieve control performance. By using formation constraints, each drone can follow the leader to achieve trajectory tracking, while also ensuring that the drone can move to the desired position according to the expected distance during formation reconstruction in the drone cluster, achieving the function of drone formation reconstruction.

Assuming there are n drones and drone S_1 is set as the virtual leader of the formation, responsible for providing the expected trajectory of the drone cluster. In order to reduce the communication pressure of the system, each drone only communicates data with another drone. The entire system is a tree structure as shown in Figure 3, with the parent node as the leader of the child node. At the same time, through a distributed control strategy, only the virtual leader is given the expected motion trajectory, and the virtual leader is given the expected distance and trajectory to the follower.

Afterwards, based on the traditional elastic system model described above, a separate force analysis is conducted on each drone in the system. Each drone S_i in the system will have a vector force that drives the drone S_i to move, represented as L_i . This force is calculated based on the drone dynamics model after giving the desired position to the drone, and is achieved by adjusting the motor speed. At the same time, each drone will be subject to the formation constraint force of all neighbouring drones in the formation

system, represented as $\sum L_i$. This force is the resultant force of the elastic force generated by the distance error between drone S_i and neighbouring drones, where i represents the sum of the constraint forces received by drone S_i . When the formation is in balance, the resultant force received by drone S_i should be kept at 0.

Figure 3 Structure of unmanned aerial vehicle cluster system



After placing a single drone into a system model based on elastic constraints, $P_i = (x_i, y_i, z_i)'$ are the current position information of the drone S_i . This article combines the elastic model into the drone cluster model to achieve a three-dimensional drone cluster formation control. In $P_i = (x_i, y_i, z_i)'$, x_i, y_i, z_i represent the coordinate points of drone S_i on the x axis, y axis, and z axis in the inertial coordinate system.

The actual distance between drone S_i and S_j is expressed as $P_{ij} = P_i - P_j$. And P_{ij}^d represents the expected distance between two drones. The distance error generated between the expected distance and the actual distance between two machines is expressed as $e_{ij} = P_{ij}^d - (P_i - P_j) = P_{ij}^d - P_{ij}$.

According to the elastic system modelling method, each drone node is set to have the same mass m . At this point, the formation force acting on drone node S_i only comes from its own output generated by approaching the desired position, as well as the formation constraint force formed by adjacent drone nodes on S_i under the elastic model. To maintain the stability of the formation, the drone S_i needs to move towards the desired position, relying on the inertial force generated by Newton's second law of motion, as well as the formation constraint of neighbouring drones. Therefore, based on the elastic system model combined with Newton's second law, a force analysis is conducted on each drone in the elastic system model. The communication topology of a drone formation is defined as an undirected connected graph $G = (V, E)$, where $V = 1, 2, \dots, n$ is the set of drone nodes, and $E \subseteq V \times V$ is the set of edges. If there is information exchange between drone S_i and drone S_j , then $(S_i, S_j) \in E$. Define the adjacency matrix $W = [w_{ij}] \in R^{n \times n}$, where $w_{ij} = \begin{cases} 1, & \text{if } (S_i, S_j) \in E, S_i \neq S_j \\ 0, & \text{otherwise} \end{cases}$. And it is agreed that $w_{ii} = 0$. This article adopts

undirected topology, so $w_{ij} = w_{ji}$. This weight matrix directly represents the existence of inter machine communication links without introducing weighting coefficients, in order to simplify the transmission model of elastic constraints, and a drone formation system model is established:

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} m & 0 & \cdots & 0 \\ 0 & m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m \end{bmatrix} \begin{bmatrix} \ddot{P}_1 \\ \ddot{P}_2 \\ \vdots \\ \ddot{P}_n \end{bmatrix} + k_f \begin{bmatrix} \sum_{j=1}^n w_{1j} e_{1j} \\ \sum_{j=1}^n w_{2j} e_{2j} \\ \vdots \\ \sum_{j=1}^n w_{ij} e_{ij} \end{bmatrix} \quad (13)$$

Among them, the sum $\sum_{j=1}^n w_{ij} e_{ij}$ represents the sum of the distance error vectors of all neighbouring drones received by drone S_i , and $e_{ij} = P_{ij}^d - (P_i - P_j)$ represents the relative position error.

According to equation (13), for any drone i and its leader, write their respective dynamic equations:

$$\begin{cases} F_i = m\ddot{P}_i + k_f \sum_{j=1}^n w_{ij} e_{ij} \\ F_1 = m\ddot{P}_1 + k_f \sum_{j=1}^n w_{1j} e_{1j} \end{cases} \quad (14)$$

Subtracting the two equations yields:

$$F_i - F_1 = m(\ddot{P}_i - \ddot{P}_1) + k_f \left(\sum_{j=1}^n w_{ij} e_{ij} - \sum_{j=1}^n w_{1j} e_{1j} \right) \quad (15)$$

Define the control quantity $u_i \triangleq \frac{F_i - F_1}{m}$ as the relative control force per unit mass, and the relative position vector is $P_{i1} \triangleq P_i - P_1$. Then $\ddot{P}_{i1} = \ddot{P}_i - \ddot{P}_1$. Assuming there is an elastic coupling term $E_{i1} \triangleq \sum_{j=1}^n w_{ij} e_{ij} - \sum_{j=1}^n w_{1j} e_{1j}$, these definitions are introduced into formula (15) and both sides are divided by m to obtain the dynamic model of the system

$$u_i = \ddot{P}_{i1} + \frac{k_f}{m} E_{i1} \quad (16)$$

The above equation is a dynamic model of a drone swarm formation with elastic constraints established by combining an elastic system model. In this model, the expected distance between the drone S_i and the virtual leader, as well as the formation constraint caused by elastic constraints between the drone S_i and its neighbouring drones in the elastic system model, are used to calculate the value of the control quantity that the drone needs to output in order to maintain the stability of the swarm formation. This ensures that the model can maintain fast formation transformation and reliable stability during formation reconstruction and even trajectory tracking.

4 Design of multi parameter fuzzy autocoupling controller

The elastic constraint formation model u_i established in Section 3 reveals that the dynamics of each drone are not only dominated by its own nonlinear dynamics, but also strongly coupled with the states of neighbouring nodes through the elastic term $\frac{k_f}{m}E_{i1}$.

The combination of internal dynamic coupling and external formation coupling makes it difficult for traditional PID control to ensure the stability and accuracy of the system under parameter uncertainty and external disturbances. To solve this complex control problem, this section designs a multi fuzzy self coupled PID distributed controller based on the aforementioned model.

If the relative position state of the i^{th} drone is defined as $\eta_{1i} = P_{1i}$, $\eta_{2i} = \dot{P}_{1i}$, then formula (16) can be rewritten as:

$$\begin{cases} \dot{\eta}_{1i} = \eta_{2i} \\ \dot{\eta}_{2i} = u_i + \frac{k_f}{m}E_{i1} \end{cases} \quad (17)$$

Among them, u_i is the distributed collaborative control law to be designed, and E_{i1} is the neighbourhood elastic coupling term. Define the global state vector $\tilde{\eta}_1 = [\eta_{11}^T, \eta_{12}^T, \dots, \eta_{1n}^T]^T$, $\tilde{\eta}_2 = [\eta_{21}^T, \eta_{22}^T, \dots, \eta_{2n}^T]^T$, then the state space model of the entire formation system is:

$$\begin{cases} \dot{\tilde{\eta}}_1 = \tilde{\eta}_2 \\ \dot{\tilde{\eta}}_2 = U + \frac{k_f}{m}E \end{cases} \quad (18)$$

Among them, U represents the global control input, and E represents the global elastic coupling vector.

The goal of formation control for the i drone is to generate the acceleration command u_i shown in formula (17) for maintaining an elastic formation. To achieve this goal, the position tracking error of the machine is defined as $e_{1,i} = P_{1i}^d - P_{1i}$, where P_{1i}^d is the expected relative position. Taking this error e_1 and its integral $e_{0,i}$, and derivative $e_{2,i}$ as inputs, and substituting them into the following formulas (19)~(22) of the self coupled PID control framework, the controller output u corresponds to the required u_i . Through this setting, the self coupling PID control method of a general second-order system is applied to solve the problem of elastic formation cooperative control. Consider a second-order nonlinear system as follows:

$$\begin{cases} \dot{c}_1 = c_2 \\ \dot{c}_2 = f(c_1, c_2) + b_0 u + d \\ h = c_1 \end{cases} \quad (19)$$

Among them, $c_1, c_2 \in R$ represent the state variables of the system; u, h represents system control input and output respectively; $f(c_1, c_2)$ represent nonlinear functions; $b_0 \neq 0$ represents the estimated value of the control channel gain; d represents an unknown bounded external disturbance, where $|d| \leq D$ and D is a constant. If the total perturbation

is $h_3 = f(c_1, c_2) + d$, and $h_3 < \infty$, then the second-order nonlinear system is mapped as $\dot{h}_1 = h_2, \dot{h}_2 = h_3 + b_0 u, h = h_1$. Among which, $b_0 \neq 0$. Obviously, the above system mapping is an uncertain system with linear properties; therefore, the system mapping is equivalent to the control of second-order nonlinear systems.

If the expected output and actual output of the controlled system are r and h , respectively, then the tracking error e_1 and its integral e_0 and derivative e_2 are:

$$e_1 = r - h, \quad e_0 = \int e_1 d\tau, \quad e_2 = \dot{e}_1 = \dot{r} - h_2 \quad (20)$$

Taking the differential of e_2 and combining it with the object e_1 , we have $\dot{e}_2 = \ddot{r} - \dot{h}_2 = \ddot{r} - h_3 - b_0 u$. The controlled error dynamic system can be obtained as follows:

$$\dot{e}_0 = e_1, \quad \dot{e}_1 = e_2, \quad \dot{e}_2 = \ddot{r} - h_3 - b_0 u \quad (21)$$

In order to enable the controlled error system (18) to asymptotically stabilise from any non-zero initial state to the origin $(0, 0)$, a reasonable and effective control law u needs to be designed. Based on the design concept of the self coupled PID controller, its collaborative controller can be constructed in the following form:

$$u = (q_c^3 e_0 + 3q_c^2 e_1 + 3q_c e_2) / b_0 \quad (22)$$

Substitute the controller (22) into the controlled error dynamic system (21) to obtain the closed-loop error system:

$$\begin{cases} \dot{e}_0 = e_1 \\ \dot{e}_1 = e_2 \\ \dot{e}_2 = \ddot{r} - h_3 - b_0 u = \ddot{r} - h_3 - (q_c^3 e_0 + 3q_c^2 e_1 + 3q_c e_2) \end{cases} \quad (23)$$

The total perturbation estimation error is defined as $\delta = \ddot{r} - h_3$, assuming $|\delta| \leq \Delta 2 = 3$ and $\Delta \cdot \Delta$ is a bounded constant. The closed-loop error system can be written as

$$\dot{e}_2 = -q_c^3 e_0 - 3q_c^2 e_1 - 3q_c e_2 + \delta \quad (24)$$

Selecting state vectors $E = [e_0, e_1, e_2]^T$, the closed-loop system can be represented as

$$\dot{E} = A_c E + B\delta. \text{ Among them, } A_c = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -q_c^3 & -3q_c^2 & -3q_c \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Theorem 1: When $q_c > 0$, matrix A_c is Hurwitz stable.

Proof: Calculate the characteristic polynomial:

$$\det(sI - A_c) = s^3 + 3q_c s^2 + 3q_c^2 s + q_c^3 = (s + q_c)^3 \quad (25)$$

The eigenvalues are $s_{1,2,3} = -q_c < 0$, all of which are negative real roots, so A_c is strictly Hurwitz.

Theorem 2: For a closed-loop error system, there exist normal numbers $\lambda, \gamma > 0$, such that:

$$\|E(t)\| \leq \|E(0)\| e^{-\lambda t} + \frac{\gamma \Delta}{q_c^3} \quad (26)$$

Proof: Selecting the Lyapunov function $V(E) = \frac{1}{2} E^T P E$, where $P = P^T > 0$ satisfies the Lyapunov equation $A_c^T P + P A_c = -I_c$. Derive along the system trajectory:

$$\dot{V} = -\frac{1}{2} \|E\|^2 + E^T P B \delta \quad (27)$$

According to Young's inequality:

$$E^T P B \delta \leq \frac{1}{4} \|E\|^2 + \|P B\|^2 \Delta^2 \quad (28)$$

By substituting formula (28) into formula (27), we can obtain

$$\dot{V} \leq -\frac{1}{4} \|E\|^2 + \|P B\|^2 \Delta^2 \quad (29)$$

From $\lambda_{\min}(P)\|E\|^2 \leq 2V \leq \lambda_{\max}(P)\|E\|^2$, we can obtain:

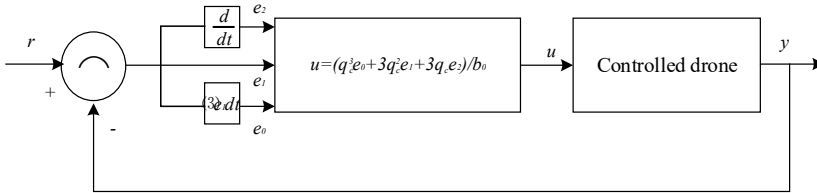
$$V(t) \leq V(0) e^{-\lambda t} + \frac{\|P B\|^2 \Delta^2}{\lambda} (1 - e^{-\lambda t}) \quad (30)$$

Among them, $\lambda = \frac{1}{2\lambda_{\max}(P)}$. The conclusion can be obtained by converting back to $\|E(t)\|$.

Inference: when the external disturbance d and model uncertainty satisfy $\Delta = 0$, the closed-loop error system is globally asymptotically stable, i.e., $\lim_{t \rightarrow \infty} \|E(t)\| = 0$.

Among them, $q_c > 0$ represents the velocity factor of the self coupled PID. The closed-loop control system composed of an auto coupled PID cooperative controller is shown in Figure 4.

Figure 4 Closed loop controller based on self coupling PID



Compared with the classical PID controller, the self coupling PID controller introduces a velocity factor q_c that is independent of the controlled object model, tightly coupling the physical links of proportional and differential properties together to form a collaborative control force with different functions and consistent objectives. The collaborative tuning rule for self coupling PID can be obtained as follows:

$$\begin{cases} k_p = 3q_c^2 \\ k_i = q_c^3 \\ k_d = 3q_c \end{cases} \quad (31)$$

According to the self coupling PID tuning rule, the speed factor q_c is linked to tune the three parameters of the PID, simplifying the debugging process. However, its initial value is fixed and cannot adapt to dynamic changes in the system online. To this end, this article introduces fuzzy control: at each sampling period, based on the current tracking error e_1 , the fuzzy rule table is queried to calculate the correction amount q_{c1} of q_c in real-time, thereby achieving adaptive online adjustment of q_c . This mechanism effectively enhances the dynamic adjustment capability of the controller, which helps to further suppress overshoot and oscillation, improve system stability and response quality. The initial set value is q_{c0} . The calculated setting value Δq_c is substituted into the following formula to obtain the corrected value q_{c1} :

$$q_{c1} = q_{c0} + \Delta q_c \quad (32)$$

The fuzzy optimised q_{c1} is substituted into formula (20) to improve PID.

5 Experimental analysis

5.1 Experimental environment

To verify the effectiveness of the distributed collaborative control algorithm for unmanned aerial vehicle formation based on fuzzy self coupling PID proposed in this paper, multiple sets of experimental tests are designed in this section. Using small and medium-sized quadcopter drones, as shown in Table 1. To simplify the simulation and focus on the performance verification of formation collaboration algorithms, it is assumed that the underlying attitude loops of each drone are stabilised by high-performance controllers, and there is no communication delay or packet loss.

Table 1 Parameters of unmanned aerial vehicle model

<i>Parameter</i>	<i>Numerical value</i>	<i>Unit</i>
Quality	1.2	kg
Acceleration due to gravity	9.8	m/s ²
Moment of inertia (X-axis)	0.011	kg·m ²
Moment of inertia (Y-axis)	0.011	kg·m ²
Moment of inertia (Z-axis)	0.021	kg·m ²
Distance from centre of gravity to propeller	0.25	kg

5.2 Experimental indicators

The total variation of the control input measures the smoothness of the control variable. The smaller the value, the smoother the control action, which helps reduce actuator wear and energy consumption, and indirectly reflects system overshoot and oscillation. The expression is:

$$TV_{u,i} = \sum_{i=1}^n \int_0^T \left\| \frac{du_i(t)}{dt} \right\|_2 dt \quad (33)$$

Among them, T represents the total time; $\|\cdot\|_2$ represents the 2-norm of a vector; $\frac{du_i(t)}{dt}$ represents the derivative of the control vector (i.e., the rate of change of the control variable).

In actual simulation, control instructions are discretely output with a fixed sampling period T_s . If the control vector at the k^{th} sampling time is $u_i[k]$, then the discrete approximation of the continuous derivative adopts a first-order backward difference:

$$\frac{du_i(t)}{dt} \approx \frac{u_i[k] - u_i[k-1]}{T_s}, k = 1, 2, \dots, N \quad (34)$$

where $N = \lfloor T/T_s \rfloor$ is the total number of sampling steps. The total variation index of discretisation is:

$$TV_{u,i}^{(\text{disc})} = \sum_{k=1}^N \left\| \frac{u_i[k] - u_i[k-1]}{T_s} \right\|_2 T_s = \sum_{k=1}^N \|u_i[k] - u_i[k-1]\|_2 \quad (35)$$

The total variation is equivalent to the cumulative sum of Euclidean distances between adjacent control instructions, independent of the sampling period of T_s .

Noise suppression measures: directly differential amplifying high-frequency measurement noise on control commands (especially when there is noise in the sensor). To balance the accuracy of reference standards and noise robustness, the following processing strategies are adopted in this paper:

Apply a first-order low-pass filter to the original control instruction $u_i[k]$:

$$u_i^{\text{filtered}}[k] = \alpha u_i[k] + (1 - \alpha) u_i^{\text{filtered}}[k-1] \quad (36)$$

The formula for calculating the total variation after smoothing is as follows:

$$TV_{u,i}^S = \sum_{i=1}^n \int_0^T \|u_i^{\text{filtered}}[k] - u_i^{\text{filtered}}[k-1]\|_2 \quad (37)$$

5.3 Analysis of experimental results

In the initial stage, the positions of the three drones are scattered, with a distance of about 1,000 metres, and their heading is different. The system first completes the assembly and stable maintenance of loose formations; subsequently, the formation compression command will be executed from 200 s onwards, with the plan to achieve the target formation constraint at 230 s with a spacing of 4 metres. It is required that the distance between the planes and the vertical height be synchronously reduced at a rate of 2 metres per second. Figure 5 shows the attitude curve of the formation position.

From Figure 5(b), it can be seen that there was a significant change in pitch angle at 200 seconds, which was mainly adjusted to correct the altitude difference between the aircraft. The yaw angle in Figure 5(c) changes relatively little throughout the entire process, which is consistent with the actual flight control situation – under the condition

of compressing only 2 m of inter aircraft plane distance, there is usually no need to make significant adjustments to the yaw angle.

The total variation data of control inputs for the three algorithms at different formation spacing is shown in Figure 6.

Figure 5 Position and attitude curve of team, (a) speed response curve (b) pitch angle response curve (c) deviation angle response curve (d) plane position change curve

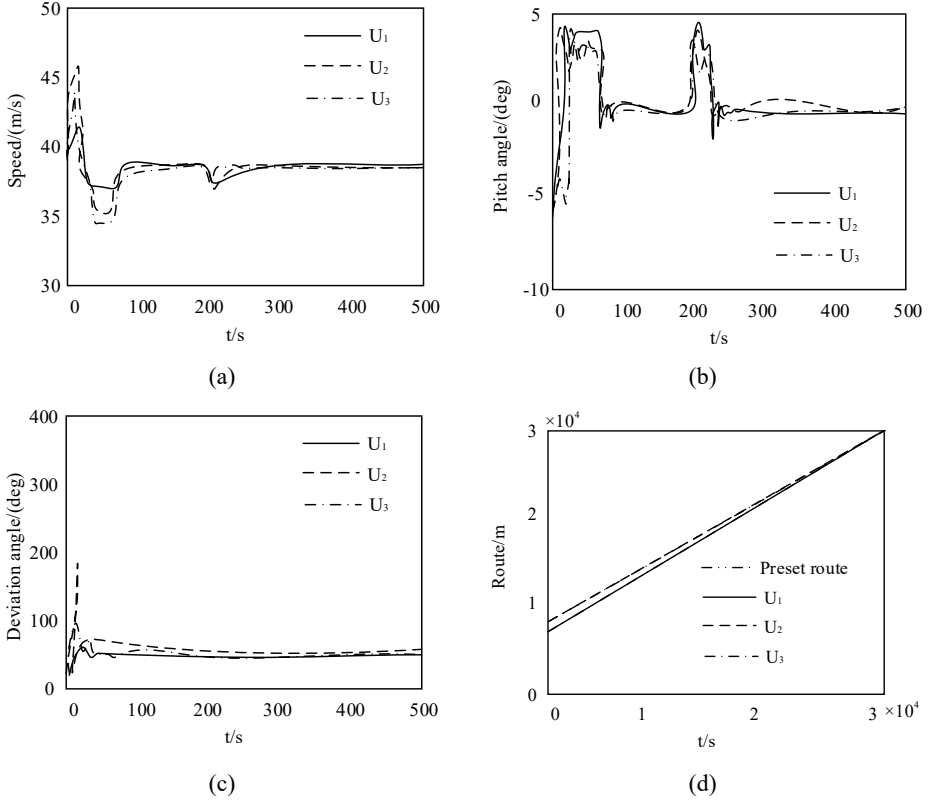
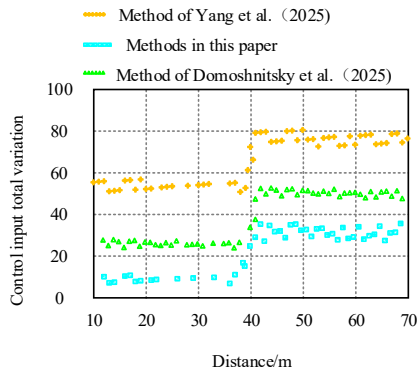


Figure 6 Control input total variation test results (see online version for colours)



As the expected formation spacing increases, the total variation of control inputs for all three algorithms shows an upward trend. This is because as the spacing increases, the drone needs to move a longer distance and a more complex path from the initial position to the desired position, resulting in a natural increase in the cumulative change in control instructions throughout the convergence process. Under all testing spacing conditions, the TV value of the fuzzy auto coupled PID (FA-PID) algorithm proposed in this paper is consistently the lowest, with a maximum value of 39.2, significantly lower than the 81.4 and 57.1 values of method of Yang et al. (2025) and method of Domoshnitsky et al. (2025). Its superiority stems from the dual adaptive mechanism of ‘fuzzy self coupling’. Fuzzy logic can adjust the speed factor in real-time based on the size of the error, thereby dynamically changing the gain combination of PID. When the error is large, the proportional effect is enhanced to respond quickly, and when the error is small, the gain is reduced to avoid overshoot, achieving online intelligent switching of control strategies. This enables FA-PID to generate control instructions more smoothly and accurately, completing formation tasks with minimal and necessary control actions, thus achieving the lowest TV value.

6 Conclusions

This article conducts systematic research on the key challenges in distributed collaborative control of drone formations. Established an elastic formation model with physical intuitiveness. Innovatively introducing a virtual spring damping network to describe inter machine interaction, a formation system model based on elastic constraints was constructed. The mathematical form of this model is clear, not only defining the expected formation, but also endowing the formation with the ability to flexibly deform and autonomously recover under disturbances, providing a more practical model foundation for the subsequent design of collaborative control algorithms. And proposed an intelligent PID controller that integrates fuzzy logic and self coupling mechanism. Specifically, a fuzzy self-coupled PID distributed controller is designed to address the strong internal and external couplings as well as parameter uncertainties inherent in the elastic formation model. This controller significantly improves the decoupling performance and robustness of the overall system.

Declarations

All authors declare that they have no conflicts of interest.

References

- Abott, R. (2024) ‘Navy works with general atomics and Lockheed Martin to demonstrate drone control station’, *Defense Daily*, Vol. 172, No. 8, pp.8–10.
- Alshdadi, A.A. and Irshad, A. (2024) ‘PDAC-SL: a PUF-enabled drone access control technique for smart logistics’, *Alexandria Engineering Journal*, Vol. 107, No. 8, pp.747–756.
- Belfadel, D., Cain, J., Haessig, D. and Chibane, C. (2024) ‘Optical flow and IMU fusion for drone horizontal velocity control’, *Journal of Advances in Information Fusion*, Vol. 19, No. 2, pp.72–74.

- Cetinsaya, B., Reinert, D. and Cruz-Neira, C. (2024) 'From PID to swarms: a decade of advancements in drone control and path planning – a systematic review (2013–2023)', *Swarm and Evolutionary Computation*, Vol. 89, No. 10, pp.1–24.
- Chang, Z., Zhang, H., Yang, Y., Jia, Y., Xu, S., Li, T. and Liu, Z. (2024) 'Fuzzing drone control system configurations based on quality-diversity enhanced genetic algorithm', *Lecture Notes in Computer Science*, Vol. 4, No. 2, pp.499–512.
- Domoshnitsky, A., Kupervasser, O. and Polonsky, A. (2025) 'Integro-differential equations in angular stabilization of drone's motion by distributed feedback control', *Mathematical Methods in the Applied Sciences*, Vol. 48, No. 17, pp.15671–15681.
- Faki, A. (2025) 'Gesture control drone: using gloves', *International Journal of Scientific Research in Engineering and Management*, Vol. 9, No. 2, pp.1–9.
- Hasegawa, D., Shindo, T., Hiraguri, T. and Itoh, N. (2025) 'Enhanced SHADE for drone control in a swarm-intelligent pollination system', *Journal of Signal Processing*, Vol. 29, No. 4, pp.123–127.
- Ili, M., Sunjevi, M., Skorin-Kapov, L. and Lorenzana, M.I. (2024) 'The impact of video encoding parameters on QoE of simulated FPV drone control', *Multimedia Tools and Applications*, Vol. 83, No. 28, pp.71575–71603.
- Kang, K. and Kim, W.J. (2025) 'A human-following drone providing gesture recognition to control IoT devices based on 3D body-landmark detection', *International Journal of Control, Automation and Systems*, Vol. 23, No. 9, pp.2718–2727.
- Lai, C.H., Lu, C.Y., Lin, C.L. and Chen, Y.Y. (2025) 'Enhancing drone swarm efficiency through a high-flexibility biomimetic formation algorithm', *Drone Systems and Applications*, Vol. 13, No. 9, pp.1–26.
- Li, H. (2025) 'Adaptive algorithms for drone flight control under communication constraints and information incompleteness', *The Aeronautical Journal*, Vol. 129, No. 2, pp.282–295.
- Mademlis, I., Symeonidis, C., Tefas, A. and Pitas, I. (2024) 'Vision-based drone control for autonomous UAV cinematography', *Multimedia Tools and Applications*, Vol. 83, No. 8, pp.25022–25083.
- Manaa, K.E., Laaidi, N., Ezzine, A. and Satori, H. (2025) 'Phoneme-based automatic speech recognition for real-time Arabic command drone control', *Iran Journal of Computer Science*, Vol. 8, No. 4, pp.1515–1531.
- Meguro, Y., Masuda, S. and Toyoda, M. (2025) 'Application and performance evaluation of backstepping control for drone 3D trajectory control', *Electrical Engineering in Japan*, Vol. 218, No. 1, pp.816–823.
- Wang, P., Hanif, A.S., Yu, S.H., Lee, C.G., Kang, Y.H., Lee, D.H. and Han, X. (2024) 'Development of an autonomous drone spraying control system based on the coefficient of variation of spray distribution', *Computers and Electronics in Agriculture*, Vol. 227, No. 1, pp.1–15.
- Weon, J., Yong, T. and Kim, J. (2024) 'Camera-based virtual drone control system using two-handed gestures', *IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*, Vol. 12, No. 6, pp.291–296.
- Xu, B., Gao, F. and Wang, W.Y. (2024) 'OmniDrones: an efficient and flexible platform for reinforcement learning in drone control', *IEEE Robotics and Automation Letters*, Vol. 9, No. 3, pp.2838–2844.
- Yang, Y. et al. (2025) 'Fuzzy control of multi-scale target tracking for quadrotor drones', *Drone Systems and Applications*, Vol. 13, No. 12, pp.1–11.