



OPEN Multi-criteria decision model for multicircular flight control of unmanned aerial vehicles through a hybrid approach

Noorulden Basil¹✉, Hamzah M. Marhoon², Bayan Mahdi Sabbar³, Abdullah Fadhil Mohammed¹, Osamah Albahri^{4,5}, Ahmed Albahri^{6,7}, Abdullah Alamoodi^{8,9}, Iman Mohamad Sharaf¹⁰, Amare Merfo Amsal¹¹✉, Mahrous Ahmed¹², Enas Ali¹³ & Sherif S. M. Ghoneim¹²

This study presents a novel approach for optimizing UAV (unmanned aerial vehicle) Multicircular flight control by developing a fractional order proportional integral derivative (FOPID)-based hybrid Eagle strategy particle swarm optimization ant lion optimizer (HESPSOALO). The proposed algorithm combines the strengths of particle swarm optimization (PSO) and the ant lion optimizer (ALO), which are enhanced by the Eagle strategy to systematically fine-tune the FOPID controller parameters. This hybrid optimization method aims to improve system stability, responsiveness, and disturbance rejection in UAVs, particularly in challenging dynamic flight conditions. The proposed approach was validated against traditional control methods that utilize FOPID (Base), the Base HESPSOALO algorithm, the FOPID-based HPSOGWO (Hybrid Particle Swarm Optimization-Gray Wolf Optimizer), and the FOPID-based HGWOALO (Hybrid Gray Wolf Optimization-Ant Lion Optimizer) with a set of benchmark functions used in the analysis. The results demonstrate a minimization of position and angular errors, reduced oscillations, and overall improved control stability for the FOPID-based HESPSOALO compared with the other methods. Furthermore, a multicriteria decision-making (MCDM) framework is applied to evaluate the overall performance of alternative control strategies utilizing the CRiteria importance through intercriteria correlation (CRITIC) and technique of order preference by similarity to ideal solution (TOPSIS) techniques. The MCDM analysis demonstrates that among the evaluated criteria, K_p has the highest importance, with a weight of 0.244019, whereas K_d is deemed the least significant, with a weight of 0.161023. The ranking results reveal that the HESPSOALO algorithm (Base) is the best-performing controller method, with a ranking score of 0.571161, indicating its superior control performance across major metrics. In contrast, the FOPID + HPSOGWO controller method ranks the lowest, with a score of 0.282794. The findings have significant industrial implications, particularly in sectors where UAVs are critical for precision tasks, such as logistics, agriculture, surveillance, and environmental monitoring. By optimizing the FOPID controller parameters, the HESPSOALO algorithm enhances UAV stability, responsiveness, and reliability in dynamic environments, resulting in more precise control and robust performance under varying conditions. This improvement may reduce operational risks and maintenance costs while increasing efficiency, prolonging UAV service life, and achieving energy savings. This study provides a robust solution for UAV control based on the potential of hybrid optimization algorithms to improve UAV precision and reliability in autonomous flight.

Keywords UAV multicircular flight control, FOPID, Hybrid optimization, CRITIC, TOPSIS

¹Department of Electrical Engineering, College of Engineering, Mustansiriyah University, Baghdad, Iraq.

²Department of Automation Engineering and Artificial Intelligence, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq. ³College of Engineering and Engineering Techniques, Al-Mustaqbal University, Babylon, Iraq. ⁴Computer Techniques Engineering Department, Mazaya University College, Nasiriyah, Iraq.

⁵Victorian Institute of Technology, Melbourne, Australia. ⁶Technical Engineering College, Imam Ja'afar Al-Sadiq University (IJSU), Baghdad, Iraq. ⁷University of Information Technology and Communications (UOITC), Baghdad, Iraq. ⁸Applied Science Research Center, Applied Science Private University, Amman, Jordan. ⁹GUST Engineering

and Applied Innovation Research Center (GEAR), Gulf University for Science and Technology, Mishref, Kuwait. ¹⁰Department of Basic Sciences, Higher Technological Institute, 10th of Ramadan City, Egypt. ¹¹Department of Mechanical Engineering, Faculty of Technology, Debre Markos University, P. O. Box 269, Debre Markos, Ethiopia. ¹²Department of Electrical Engineering, College of Engineering, Taif University, 21944 Taif, Saudi Arabia. ¹³Centre for Research Impact and Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab 140401, India. ✉email: noorulden@uomustansiriyah.edu.iq; amare_merfo@dmu.edu.et

There has been total integration of unmanned aerial vehicles (UAVs), otherwise known as drones, in various aspects of our lives, such as defense, surveillance, agriculture, infrastructure monitoring and commercial delivery, depending on the field. Owing to their ability to either operate autonomously or semiautonomous, UAVs are well positioned to perform tasks in hazardous or impractical environments where human intervention would be impossible. UAV applications are growing, and their control systems need to evolve to provide precise navigation, stability and reliability under dynamic and complex environments. Despite these advances, however, UAV control remains a challenging problem, as their flight dynamics are highly dynamic and nonlinear. UAV control is complicated by external disturbances, including wind gusts, varying payloads, and obstacles, and is sensitive to perturbations in both positioning and orientation; robust control strategies are needed that can maintain stability and limit error during these disturbances. To control a UAV, effective UAV control systems need to ensure that the UAV remains on its intended trajectory with minimum errors in position, velocity, and angular displacement (yaw, pitch and roll). This level of control was achieved via advanced algorithms that can quickly and accurately respond to changing conditions^{1,2}.

As with other aspects of control theory, traditional proportional-integral-derivative (PID) controllers have been a mainstay in the UAV control community since they are simple and effective in linear systems. Unfortunately, PID controllers tend to fail to control the nonlinearities and time-varying behavior of UAVs in complex maneuvering. However, such limitations demand more complexity in control methods, as researchers resort to control techniques that involve the Fractional-Order Proportional-Integral-Derivative (FOPID) controller, a more sophisticated version of the classical PID controller that replaces the integral and derivative components with fractional order functions. Therefore, the extra flexibility in the FOPID results in better performance in terms of stability and better performance in terms of robustness since it is better able to handle the complicated dynamics of UAVs. Nevertheless, proper parameter tuning of the FOPID controller is a complex and time-consuming process. In dynamic environments, the parameters need to be tuned manually, and as a result, the performance is suboptimal. To address this problem, hybrid optimization algorithms are automated and improved tuning processes. Hybrid optimization methods use the strengths of various optimization techniques to search for an optimal set of control parameters that can be used only at the peak performance of the FOPID controller^{3,4}. To understand the importance of our topic, two important questions are introduced as follows:

The FOPID controller augments precision, flexibility and robustness. The control system for the circumnavigation of multiple targets or reference points via UAVs is a complex system that can be difficult to stabilize, especially in the dynamic environments where UAVs operate. It is well suited for these challenges, and it has multiple advantages over traditional PID controllers. Compared with the pure P controller, the FOPID controller in the DMCC is one of the major contributions from the perspective of the controller that serves to track the trajectory more accurately. In a DMCC situation, UAVs must travel around a series of target points on continuous circular paths. This level of precision must be maintained, particularly when many UAVs are involved in distributed control. Using the fractional order terms in the FOPID controller yields more refined control to track more accurately generated circular paths. Thus, the UAVs can stay on course in the presence of external disturbances or dynamic variations⁵.

This FOPID controller provides stability and robust improvement of the UAVs for these operations. The key to ensuring stability in the coordinated movements of UAVs as they circumnavigate is as UAVs collaborate. The major elements in the FOPID controller are introduced to the system, which can better handle the nonlinear dynamics of the UAV and increase the robustness of the control system to disturbances and stability risk. This added stability is critical in DMCCs with UAVs that must continuously realign their positions with respect to each other without inducing path deviations or collisions. A second major requirement of DMCCs is accurate coordination of various UAVs. In this area, the FOPID controller excels because each UAV can adjust its position and velocity with respect to the movement of other UAVs in the system. The FOPID controller fractional components guarantee smoother, more controlled responses to the positional and velocity errors, improving the overall tower coordination of the UAV fleet. As UAVs circumnavigate their targets, more seamless interactions between UAVs are needed, which is important for maintaining the integrity of the control system. In addition to increasing stability, coordination, and UAV adaptability to changing environments, the FOPID controller improves the tradeoff between the output dwell time and the gain used to specify it. This, however, can interfere with the UAV trajectory because of external interference from wind, obstacles, and communication delays in the DMCC. This fractional order allows the FOPID controller to be more flexible and make better adjustments to the control output; thus, it is suitable for a variety of disturbances, such as when UAVs respond effectively. This adaptability ensures that the system can keep operating as stable as possible, as it slows dynamic environments that might constrain traditional PID controllers. Furthermore, compared with the existing implementations, the FOPID controller presents more effective damping in the case of oscillations for the DMCC since, in that case, convergence of UAVs to their desired circular paths must be imposed without excessive overshoot. The FOPID controller provides better control of oscillatory behavior so that UAVs converge faster to their target. More refined control is achieved over the UAVs' response to deviations from their paths through the fractional order derivative and integral components, which help minimize oscillations and improve overall system convergence. Finally, the FOPID controller's parameters can be tuned in the DMCC context. Compared with traditional PID

controllers, the FOPID controller provides more control gain optimization with more degrees of freedom. In a multiagent system such as the DMCC, this is particularly useful, as UAVs may be required to do so with different circumnavigation ratios or speeds. The overall performance of the system can be improved by the ability to fine tune these parameters for each individual UAV, and each UAV can operate optimally under the distributed control framework⁶.

The key role of optimization algorithms in maintaining both PID and FOPID controllers in the distributed Multicircular circulation control (DMCC) framework for UAVs is demonstrated. The use of these algorithms to fine-tune the parameters of the controllers strongly affects the ability of the system to fly on stable, precise and coordinated flight paths. UAVs have complex and highly nonlinear dynamic problems, particularly if they are operating in distributed environments with multiple vehicles. When the controller parameters need to be adjusted for such scenarios, the result is often suboptimal performance. Optimization algorithms take place here to provide a more systematic and more efficient procedure for choosing the right control settings for both a PID controller and an FOPID controller⁷.

Optimization algorithms can overcome traditional PID controller limitations because of the inherent limitations of manual tuning. While simple and commonly used, PID controllers have difficulty handling the nonlinear dynamics and multiagent nature of DMCC systems. Inappropriately tuned, these controllers may lead to oscillatory, unstable, or ineffective disturbance responses. Using optimization algorithms, including Particle Swarm Optimization (PSO), Genetic Algorithms (GA), or hybrid algorithms, the parameters of a PID controller (proportional, integral, and derivative gains) can be modified so that better control can be achieved. This allows UAVs to travel along more stable circular paths around designated targets as perceived by the UAV, minimizing positional and velocity errors in the presence of external disturbances, e.g., wind or communication delays⁸.

Adding fractional terms to the FOPID controller makes optimization more essential when the fractional form is used. In contrast, FOPID controllers do not require all these parameters, nor do they require that the summation constant be exactly 1, as does a traditional PID controller; however, their flexibility comes at a cost since fractional calculus is used, meaning that more parameters are introduced, which must be tuned more finely. By including the fractional-order integral and derivative components, the control of the system's response is enhanced to make DMCC systems handle the dynamics of UAVs more efficiently. However, this flexibility involves the challenge of choosing appropriate values for these parameters, a task that cannot be achieved by manual tuning alone. This process is simplified by optimization algorithms that automate the search for the best fractional order parameters so that, under an FOPID controller, the UAVs' trajectories can be managed in the best way possible⁹.

The DMCC enables optimization algorithms to adapt PID and FOPID controllers to both dynamic and uncertain environments. Optimization algorithms are used to tune the control parameters for PID controllers such that the UAVs can be more responsive and adaptable to external disturbances in real time. The role of optimization increases even further for FOPID controllers, as the fractional order terms enable a more gradual, more precise control response. These terms are optimized so that the UAVs remain stable and coordinated under rapidly varying conditions and that the UAV system is as adaptable as possible. Additionally, the convergence and stability of the control system in the DMCC are improved significantly by optimization algorithms. Optimizing the PID and FOPID controllers for a UAV reduces oscillations and overshoot and results in better convergence for the UAV to its desired circular path. This stability is essential in multiagent systems such as DMCCs, where multiple UAVs must cooperate to achieve coordinated circumnavigation. Controllers are tuned via optimization algorithms to avoid collisions or deviations away from the desired flight path and prevent UAV interactions¹⁰.

Recently, a comprehensive literature review of recent research articles presenting new approaches for UAV control, autonomous system emulation, and energy-efficient data collection systems has been conducted. References are divided by the type of controller used, by whether a control or optimization method has been applied, and by tuning parameters (K_p , K_i , K_d , λ , μ) or flight dynamics (pitch, roll, yaw). Furthermore, the table shows the aims of each study, and the results obtained. The structured review provides a clear comparison of various control strategies, and the effectiveness in real-world applications where UAV control, voltage regulation, and vehicle maneuvering are used are shown in Table 1.

A combination of a fuzzy fractional-order PID (FOPID) controller and layered learning and proximal policy optimization (PPO) was proposed in¹¹ for simultaneous control and path planning of quadrotor drones. The use of this hybrid method provides improved path planning efficiency and overall control performance with better control and navigation of UAVs by adjusting the PID parameters (K_p , K_i , K_d) and fractional orders (λ , μ). This finding shows that the gain in system control caused by the enhanced control scheme in complex environments is sufficiently high to justify its use in real-time applications. The authors hybridize the Golden Jackal and Golden Sine Optimizers (GOJ-GSO) for the control parameter tuning (K_p , K_i , K_d) of the PID controller in¹². This method provides a robust tuning method, which significantly increases not only the accuracy but also the responsiveness of the system. The results demonstrated this hybrid optimization to achieve optimal control for several engineering applications. Furthermore, a fractional sliding mode control (FSMC) scheme that uses a potential field algorithm to minimize instability was introduced in¹³ for cooperative flight control. This is an approach for cooperative control of a fleet of quadrotors when pitch, roll and yaw motions are optimized in terms of coordinated flight. Multiple UAV operations in challenging environments are demonstrated in a study that successfully shows improvements in maneuverability and synchronization among multiple UAVs. The SMA motor control algorithm presented in¹⁴ was further explored in an SMA motor control algorithm for optical image stabilization via shape memory alloy (SMA) motors. The objective of this study is to establish a stable image stabilization method via precise robot control. This approach effectively shows the ability of SMA motors to maintain the system stability needed in applications where precise optical imaging is needed, such as aerial imaging.

Ref. & author name	Controller type	Method type	Criteria type	Aims	Achieved results
Shahbazi et al. ¹¹	Fuzzy fractional order PID (FOPID)	Layered Learning with proximal policy optimization (PPO)	$K_p, K_i, K_d, \lambda, \mu$, path planning, controlling	Simultaneous controlling and path planning in quadrotor drone	Improved control performance, efficient path planning
Mou et al. ¹²	PID	Hybrid golden jackal and golden sine optimizer (GOJ-GSO)	K_p, K_i, K_d	PID controller tuning for performance improvement	Enhanced tuning of PID controllers using hybrid optimization approach
Alabsari et al. ¹³	Fractional sliding mode control (FSMC)	Cooperative control, potential field algorithm	Pitch, roll, yaw	Cooperative flight control for a fleet of quadrotors	Coordinated quadrotor flight achieved through FSMC with potential fields
Li et al. ¹⁴	SMA motor control algorithm	Modeling and control algorithm for SMA motors	N/A	Optical image stabilization through SMA motor control	Stable control of SMA motors for image stabilization
Abdel-Basset et al. ¹⁵	Differential evolution-based system	Evolution-based Optimization for Energy-Efficient Data Collection	N/A	Energy-efficient data collection system for UAV-supported IoT	Optimized energy usage in UAV-IoT systems
Abdelaal et al. ¹⁶	Sliding mode control	Dynamic Voltage Restorer (DVR) control for voltage sag compensation	N/A	Voltage sag compensation using DVR with sliding mode control	Improved voltage regulation under sag conditions
Yang and Guan ¹⁷	PI (proportional-integral) controller	Improved Whale Optimization Algorithm (IWOA) for parameter tuning	K_p, K_i	PI parameters tuning for wireless power transfer system	Enhanced PI control for frequency tracking in wireless power systems
Shet et al. ¹⁸	Fractional-order sliding mode control	Intelligent control for autonomous vehicle maneuvering	N/A	Maneuvering of an autonomous vehicle through fractional sliding mode control	Improved vehicle maneuvering accuracy
Jie and Yuanqing ¹⁹	ADRC (active disturbance rejection control)	Nonlinear ADRC-based control system for inverted pendulum	N/A	Stability analysis and control of nonlinear systems	Demonstrated stability control in nonlinear ADRC-based systems

Table 1. Summary of recent literature on control methods and optimization techniques in UAVs and autonomous systems.

In¹⁵, researchers worked on energy-efficient data collection for UAV-supported Internet of Things (IoT) systems. To optimize the power consumption of UAVs during data collection tasks, this study employs a differential evolution-based optimization system. We present the method and show how it effectively reduces energy usage, prolongs the operational time of UAVs in IoT networks, and enhances large-scale data collection in real-world applications. Sliding mode control (SMC) was used in¹⁶ for voltage sag compensation by a dynamic voltage restorer (DVR). This method of voltage regulation allows the voltage instability to be corrected while allowing distributed power to ensure power delivery in critical systems. The results show that sliding mode control appears to improve the DVR's ability to sustain voltage levels, effectively providing a passive means to mitigate power sags. To tune the proportional integral (PI) controller parameters (K_p, K_i) in wireless power transfer systems, the IWOA was improved by the authors of¹⁷. This optimization improves the accuracy and stability of frequency tracking control in wireless systems. Enhanced PI control is imperative for power transfer efficiency under varying operating conditions. Fractional-order sliding mode control (FSMC) was proposed in¹⁸ as an intelligent control system for autonomous maneuvering of vehicles. The following study aims to improve the precision of vehicle control for autonomous systems, especially in complex environments that need dynamic adjustments. The analytical results show that the roll of the vehicle was significantly improved, which helped advance autonomous vehicle technologies. Finally, in¹⁹, a nonlinear active disturbance rejection control (ADRC) approach was taken for stability analysis of inverted pendulum systems. It is intended to show the stability control of nonlinear systems via ADRC, a robust control method that compensates for disturbances. We confirm the effectiveness of ADRC in maintaining system stability under such challenging dynamic conditions and, by extension, in systems sensitive to control accuracy.

With respect to multiple criterion optimizations of PID and FOPID parameters in distributed multicircular circumvention control (DMCC), we identify several critical gaps and challenges in recent studies on UAV control and optimization. Although significant advances in control theory and optimization algorithms have been made, many researchers have not fully addressed the inherent trade-offs when these controllers are tuned. This domain is especially challenging, as most approaches that have been proposed overrely on traditional optimization tools, which tend to ignore the multiobjective nature of UAV control systems. UAVs work in complicated settings where controllers must search for two or more struggle sketches with stability, exactness, flexible handling, and computational productivity. Most normal optimization techniques attempt to optimize a single cost function or collection of metrics but do not often explore the sensitive relationships among key performance indicators. The inherent complexity and multiple dimensions of this problem pose major challenges in UAV optimization, as normal methods are not well suited to this problem. Standard optimization algorithms such as Particle Swarm Optimization (PSO) Genetic Algorithms (GA) and others can provide acceptable solutions in few cases, but they tend to neglect the intricate trade-offs of performance metrics such as positional accuracy, response time, energy consumption and an aspect of stability while keeping in mind the fulfillment of security risks and business law. For example, if a tuned FOPID controller is better at stabilizing UVAs, it will not be as fast or precise. In such cases, there is no 'best' set of parameters since bettering one performance parameter may necessitate hurting the other. The result of this limited feasibility in these trade-offs is a problem where existing approaches fall short in fully assessing the interdependencies between these trade-offs to realize suboptimal UAV systems for real-world applications. Furthermore, most current research does not address how to assign proper weightings in

the performance criteria space in UAV control systems. Every criterion is important because, without carefully crafting the relative importance of each criterion, it is almost impossible to achieve balanced outcome-driven optimization. Usually, standard optimization techniques do not offer a suitable way to determine the relative significance of various objectives. This is a large gap, as the weights of different performance metrics, such as position error and velocity error, can dramatically change the optimization results. In practice, a UAV system rarely has true equal weighting or arbitrary choices, as is often a default in the literature. The further inability to critically analyze the importance of each criterion further complicates the selection of optimal parameters; thus, these solutions are not applicable in the DMCC, which is a complex multiagent environment. These challenges need to be overcome by researchers using more sophisticated multicriteria decision-making (MCDM) techniques. One way to achieve objective weights for different criteria is by CRiteria importance through intercriteria correlation (CRITIC), whereby we analyze the correlation between criteria to assign weights. Using CRITIC, we address the critical gap in traditional optimization by providing a systematic way to evaluate the relative importance of different performance indicators such that trade-offs are handled properly. After CRITIC has determined the weights, the values of possible solutions can be ranked via the technique of order preference by similarity to ideal solution (TOPSIS), which is based on the closeness of these solutions to an ideal solution. Six steps are given for implementing TOPSIS for evaluating each solution, as the best and worst possible outcomes for each criterion are considered, eliminating the shortcomings of standard approaches and providing a more holistic view of the optimization problem. Traditional optimization methods do not incorporate these multicriteria methods, and the gap in related research is significant. Discrete multicriteria combing (DMCC) tasks have high complexity in UAV systems, and the level of evaluation beyond mere determination of costs indicates inevitable coordination between executive and planning components. Common methods lack the ability to consider critical values and find trade-offs between different performance criteria, which often leads to suboptimal solutions that cannot be applied in real settings. Unaccounted issues can cause researchers to propose control systems that perform well in hypothetical situations but do not consider their limits during circumnavigation UAV tasks requiring dynamic, multifaceted demands.

However, recent studies must fully consider the complex trade-off that underpins UAV control optimization. Since their ability to address competing objectives, assign meaningful weights to performance criteria, and evaluate solutions in a multidimensional space are very limited, normal optimization methods cannot achieve the desired balance. This has resulted in a significant gap in the field, which requires researchers to explore traditional techniques and adopt more advanced techniques, such as CRITIC and TOPSIS. The application of these methods to UAV control systems should enable the system to be more accurately tuned to the complex and often conflicting demands of real-world applications so that UAVs can perform the necessary stability, precision and adaptability required for tasks of DMCC.

The main contributions of this study are two main areas: UAV control and optimization and the specific area of multicircular flight paths. The key contributions and novelty of this research are as follows:

1. This study develops the HESPSOALO algorithm, a hybrid of PSO and ALO with the Eagle strategy, to optimally tune FOPID control parameters for effective UAV control in dynamic settings.
2. The hybrid algorithm improves UAV multicircular flight control, optimizing parameters (K_p , K_i , K_d , λ , and μ) for better stability, accuracy, and disturbance rejection.
3. MCDM techniques (CRITIC and TOPSIS) are applied for an objective evaluation of control strategies, ensuring robust comparisons across stability, precision, and robustness metrics.
4. The FOPID-based HESPSOALO outperforms other methods (e.g., FOPID, HESPSOALO base, HPSOGWO, HGWOALO) in minimizing error and enhancing system response in demanding UAV conditions.
5. This study introduces fractional order dynamics (λ , μ) for UAV control, enhancing performance under complex conditions by optimizing disturbance and uncertainty handling.
6. This research provides practical advancements for industries reliant on UAV precision, such as logistics, agriculture, and infrastructure inspection. The optimized FOPID-based HESPSOALO control system enhances reliability, reduces maintenance needs, and extends UAV operational life, supporting broader UAV adoption in demanding environments through improved stability and energy efficiency.

There are five sections to this article. The multicircle circumnavigation problem with a moving target and its solutions are explained in the second section. The third section explains the proposed methodology, which introduces three hybrid optimization algorithms based on the FOPID controller. The results are presented and discussed in Section four. Section five concludes with a discussion of its limitations and future work.

UAV model and derivations

This work investigates a multicircle circumnavigation problem with a moving target, and a preset formation of N UAVs is considered in the cluster, as shown in Fig. 1.

Within the X–Y earth-fixed reference, the position of the i -th UAV is denoted by $p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$. Equations (1), (2), and (3) show how the kinematics of the i -th UAV are expressed^{20–23}:

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = v_i \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix} \quad (1)$$

$$\dot{\theta}_i = r_i \quad (2)$$

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = v_i \begin{bmatrix} \cos r_i \\ \sin r_i \end{bmatrix} \quad (3)$$

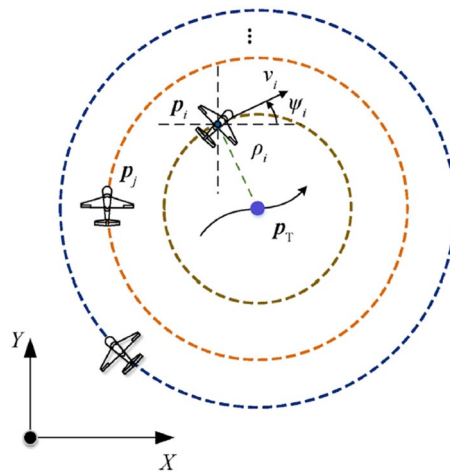


Fig. 1. A geometric illustration of cooperative circular enclosing.

where the angular rate and velocity are indicated appropriately, r_i denotes the linear component, and $v_i > 0$ indicates the heading angle. $0 < v_i \sim v_{max}$ and $r_{min} r_i \sim r_{max}$ are the UAV states in which v_{max} represents the maximum velocity and where r_{min} and r_{max} are the lower and upper bounds of the angular rate, respectively. For input, there is an angular rate r_i and a bounded and measurable velocity v_i ²⁴. Equation (4)^{25,26} expresses the kinematics of the moving target, $p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$.

$$p_T = v_T = \begin{bmatrix} v_{xT} \\ v_{yT} \end{bmatrix} \quad (4)$$

where $v_T = \begin{bmatrix} v_{xT} \\ v_{yT} \end{bmatrix}$, $\|v_T\| \leq z_T$ is a positive constant and v_T is the target velocity. It is assumed that v_T and its derivative are bounded and that there is an upper bound on v_T because of the maneuvering constraints.

Several UAVs are recommended. The ultimate control goal, as stated in Eq. (5), is to force all UAVs to evolve along a predetermined circular trajectory and maintain a desired neighboring angular spacing if target information is available to all of them.

$$T = T_{max} \cdot e^{-k \cdot d} \quad (5)$$

where d is the displacement from the equilibrium position, T is the tension in the circle, T_{max} is the maximum tension capacity, and k is a constant that represents the circular stiffness.

For Undirected Topology UAVs, this section provides a distributed multicircular circumnavigation control. suitable angular spacing and the corresponding control mechanism^{27,28}.

Newton's second law and the Euler equations for rigid body motion can be used to determine the translational and rotational movements of a multicircular UAV. The rotational Eq. (6) that follows:

$$\begin{cases} m\dot{v}_f = R_b \cdot F_T + F_f + mg + T \\ M_0 = I\dot{\omega}_b + \omega_b \times (I\omega_b) \end{cases} \quad (6)$$

where I is the inertia; mg is the weight of the UAV system; T is the tension from the multicircular and tensor of the multicircular UAV; $\omega_b = [\dot{\psi}, \dot{\theta}, \dot{\phi}]^T$ is the angular velocity of the multicircular UAV; the yaw pitch and roll angles are represented by ψ , θ , and ϕ , where m is the mass of the multicircular UAV; $v_f = [\dot{x}, \dot{y}, \dot{z}]$, x , y , and z are the positions of the UAV in three directions; R_b is the rotation matrix; and M_0 is the total moment on the multicircular UAV²⁹.

Multiple types of tension, aerodynamic drugs, and gravity are the primary complex forces at work. In this work, a "concentrated mass-lightweight rod" model is used to establish the tether model, considering the impact of the tether on the UAV's balance. Considering that multicircular UAVs usually operate in favorable weather conditions, we ignore resistance interference with multicircular UAVs³⁰.

Considering that the multicircular element experiences T_i and has a length of ds . Additionally, assuming that the tether is in contact with the earth tension indicated by T_0 and T_n , the motor is at point O and the UAV is at point On . The equation of balance for the I -th multicircular segment is displayed in Eq. (7)³¹:

$$\begin{cases} T_i \sin(\alpha_i) - T_{i-1} \sin(\alpha_{i-1}) = \epsilon ds \\ T_i \cos(\alpha_i) = T_{i-1} \cos(\alpha_{i-1}) \end{cases} \quad (7)$$

where α_i is the included angle between the i -th segments and the x -axis and where ϵ is the gravity per unit length of the multicircular, which is dependent on the material of the tether³². Figure 2 depicts the geometric correlation between a target and UAVs.

Owing to the symmetry of the force of the cable in the x and y directions, we convert three-dimensional tether modeling to two-dimensional modeling because of the symmetry of the force of the cable in the x and y directions. Two-dimensional space is viewed from two-dimensional space. The multicircular does not exhibit any vertical protrusions along the y -axis. According to dz , we can determine the values of x and z . These values are derived from Eq. (8)³³.

$$\begin{cases} x = \frac{T_n \cos(\alpha_n)}{\epsilon} \ln \left(\sqrt{1 + \left(\frac{\tan(\alpha_n) + S\epsilon}{T_n \cos(\alpha_n)} \right)^2} / (\sec(\alpha_n) + \tan(\alpha_n)) \right) \\ z = \frac{T_n \cos(\alpha_n)}{\epsilon} \sqrt{1 + \left(\frac{\tan(\alpha_n) + S\epsilon}{T_n \cos(\alpha_n)} \right)^2} - \frac{T_n}{\epsilon} \end{cases} \quad (8)$$

The lengths in the x - and z -axes correspond to the drone's position with respect to the motor, and the extended length S of the multicircular is considered. The formulas allow us to compute T_n and n , which in turn allows us to calculate the tension at the point where the multicircular and UAV connect. As a result, tension T has the following Eq. (9)³⁴:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = T_n \begin{bmatrix} \cos(\lambda_1) \cos(\lambda_2) \\ \cos(\lambda_1) \sin(\lambda_2) \\ \sin(\lambda_1) \end{bmatrix} \quad (9)$$

We now address coordinate transformation. The angle between the tether's projection in the XY plane and the plane of X is λ_2 . The kinematic relationships of multicircular UAVs are described via the Earth-fixed coordinate system. It is common practice to use the body coordinate systems.

$B = (Xb, Yb, Zb)$ and $E = (Xe, Ye, Ze)$. Equation (10) presents the rotation matrix from the Earth-fixed coordinate system to the body coordinate system³⁵:

$$TR_b = \begin{bmatrix} \cos(\theta) \cos(\psi) & \cos(\psi) \sin(\theta) \sin(\phi) - \sin(\psi) \cos(\phi) & \cos(\psi) \sin(\theta) \cos(\phi) + \sin(\psi) \sin(\phi) \\ \cos(\theta) \sin(\psi) & \sin(\psi) \sin(\theta) \sin(\phi) + \cos(\psi) \cos(\phi) & \sin(\psi) \sin(\theta) \cos(\phi) - \cos(\psi) \sin(\phi) \\ -\sin(\theta) & \cos(\theta) \sin(\phi) & \cos(\theta) \cos(\phi) \end{bmatrix} \quad (10)$$

where S and C stand for the trigonometric functions $\sin$ and $\cos$, respectively. The yaw, pitch, and roll angles of the multicircular UAV are denoted by ψ , θ , and ϕ , respectively³⁶. Figure 3 presents the transforming multicircular UAV position diagram:

Equation (11) uses the rotation matrix Reb to express the lift force (FT) produced by the rotating rotors.

$$F_T = R_b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \sum_{i=1}^4 C_i \omega_i^2 \quad (11)$$

Let C_i stand for the motor, Reb for the rotation matrix from the body coordinate system to the Earth-fixed coordinate system, and lift coefficients. The rotational speed of the four rotors is ω_i ($i = 1, 2, 3, 4$).

The system's geometry is altered by the swinging motion of the mechanical arm when the multicircular UAV grasps fruit. Moreover, the total mass of the multicircular UAV changes once it grasps the target object. As a

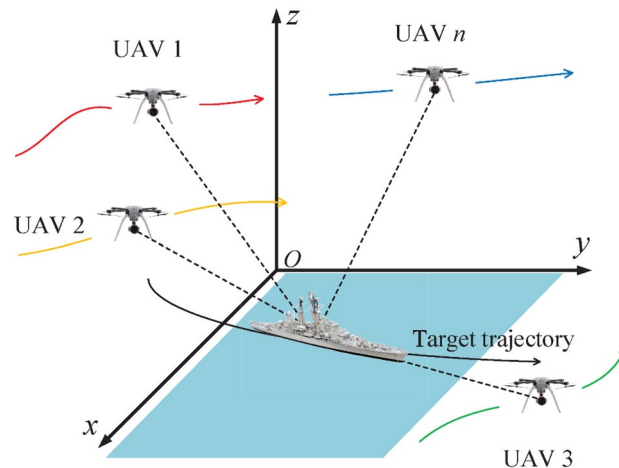


Fig. 2. The geometric relationship between a UAV and a target.

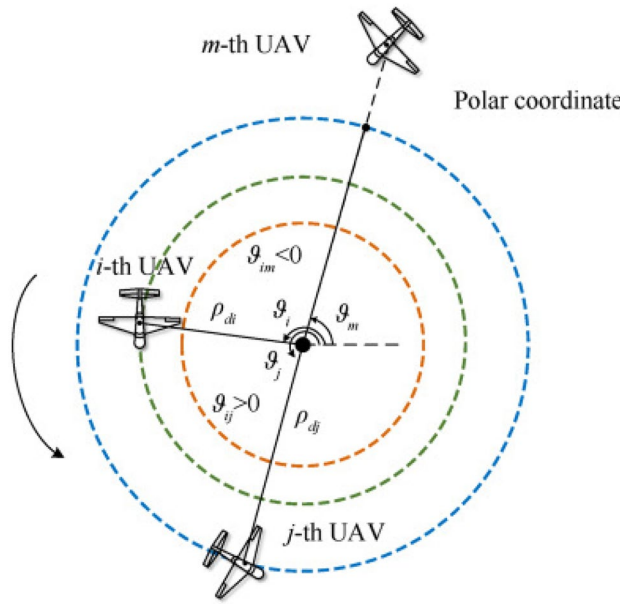


Fig. 3. Transforming multicircular UAV positions.

result, the inertia tensor matrix's apparent physical transformation is observed. Equation (12) illustrates how the grasping state causes a shift in the center of gravity, which changes the inertia matrix³⁷:

$$I = \begin{pmatrix} I_{x0} + \gamma_x & 0 & 0 \\ 0 & I_{y0} + \gamma_y & 0 \\ 0 & 0 & I_{z0} + \gamma_z \end{pmatrix} \quad (12)$$

Using the rotation matrix and the inertia tensor matrix as our guides, we can determine the equations $I_x = I_{x0} + \gamma_x$, $I_y = I_{y0} + \gamma_y$, and $I_z = I_{z0} + \gamma_z$. Which represents the correction made for the discrepancy between the inertia at the grasping operation and the initial inertia tensor.

The rolling, pitching, and yawing moments brought about by the propellers' rotation are then added together to define the total lift, or U_1 . The expressions in Eq. (13) are as follows³⁸:

$$\begin{cases} U_1 = C_t (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ U_2 = \sqrt{2} C_t l (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ U_3 = \sqrt{2} C_t l (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ U_4 = C_m (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (13)$$

where ω_i ($i = 1, 2, 3, 4$) is the rotational speed of each of the four rotors and where l is the length between the center of the rotor and the drive center. For the multicircular UAV, the motor lift coefficient is C_t , and the inverse torque scaling coefficient is C_m . The nonlinear dynamics model of the operational multicircular UAV is shown below, combining the previously presented analysis with Eq. (14)³⁹:

$$\begin{cases} \ddot{x} = \frac{1}{m} [U_1 (-C_\psi C_\phi S_\theta + S_\psi S_\theta) - TC_{\lambda_1} C_{\lambda_2} - k_f \dot{x}] \\ \ddot{y} = \frac{1}{m} [U_1 (-S_\psi C_\phi S_\theta - C_\psi S_\theta) - TC_{\lambda_1} C_{\lambda_2} - k_f \dot{y}] \\ \ddot{z} = \frac{1}{m} [U_1 (-C_\theta C_\phi) - TS_{\lambda_1} - k_f \dot{z} - mg] \\ \ddot{\psi} = \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\phi} + \frac{I_x \dot{\theta} \Omega}{I_x} + \frac{l U_2}{I_x} \\ \ddot{\theta} = \frac{I_z - I_x}{I_y} \dot{\psi} \dot{\phi} + \frac{I_x \dot{\psi} \Omega}{I_y} + \frac{l U_3}{I_y} \\ \ddot{\phi} = \frac{I_x - I_y}{I_z} \dot{\psi} \dot{\theta} + \frac{U_4}{I_z} \end{cases} \quad (14)$$

The rotor's rotational inertia (I_r) is the coefficient of air resistance, the angles between the top end of the multicircular plane and the XOY plane and the X-axis plane are, respectively, and the UAV's rotational speed is equal to $\omega_1 + \omega_2 + \omega_3 + \omega_4$ ^{40–42}.

The arm's relatively high moment of inertia allows the changes to the body, rotor, and motor to be ignored. Thus, Eqs. (15) and (16) can be used to express the offset of the multicircular UAV's center of gravity position, denoted by rC ⁴³.

$$\begin{cases} \ddot{x} = \frac{1}{m} [U_1 (-C_\psi C_\phi S_\theta + S_\psi S_\theta) - TC_{\lambda_1} C_{\lambda_2} - k_f \dot{x}] \\ \ddot{y} = \frac{1}{m} [U_1 (-S_\psi C_\phi S_\theta - C_\psi S_\theta) - TC_{\lambda_1} C_{\lambda_2} - k_f \dot{y}] \\ \ddot{z} = \frac{1}{m} [U_1 (-C_\theta C_\phi) - TS_{\lambda_1} - k_f \dot{z} - mg] \\ \ddot{\psi} = \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\phi} + \frac{I_r \dot{\theta} \Omega}{I_x} + \frac{IU_2}{I_x} \\ \ddot{\theta} = \frac{I_z - I_x}{I_y} \dot{\psi} \dot{\phi} + \frac{I_r \dot{\psi} \Omega}{I_y} + \frac{IU_3}{I_y} \\ \ddot{\phi} = \frac{I_x - I_y}{I_z} \dot{\psi} \dot{\theta} + \frac{U_4}{I_z} \end{cases} \quad (15)$$

$$r_C = \frac{ma(l_1 - l_0) + (M - ma)l_1}{M} \quad (16)$$

Here, M is the total mass of the multicircular UAV, ma is the total mass of the four arms, and l_0 is the length of the arms before deformation. Let ΔI stand for the rotational inertia change, which is expressed as follows in Eq. (17):

$$\Delta I = ma(r - r_c)^2 \quad (17)$$

Equation (18) provides us with a new expression for the inertia matrix (I').

$$I' = \begin{bmatrix} I_x - \Delta I & 0 & 0 \\ 0 & I_y - \Delta I & 0 \\ 0 & 0 & I_z - \Delta I \end{bmatrix} \quad (18)$$

Equation (19) expresses how the quadcopter's lifting arms fold to form a "bowl"-shaped multicircular unmanned aerial vehicle system. The model is significantly impacted by the power produced by the spinning of the four rotors. Under these conditions, the thrust generated by the rotating propellers modifies the attitude and balance of the thrust and torque of the multicircular UAV. Changes in F_i result in variations in the total lift force U_1 , which is consistent with the objectivity principles of balance⁴⁴.

$$U_1 t = \sum_{i=1}^4 C_t \omega_i^2 \sin(\Delta \sigma) \quad (19)$$

With its arms bent at an angle to the horizontal and in a stable hovering position, the UAV folds its arms without affecting torque balance toward the ground. Thus, the folding deformation can be roughly interpreted as a telescopic change in arm length. The matrix of inertia can be calculated via the change in arm length, $l_2 = l_0 \sin(\Delta \delta)$. The derivation of motions can be expressed in Eqs. (20–45)¹.

$$\begin{bmatrix} \cos \theta_1 & \cos \theta_2 & \sin \theta_3 \\ -\sin \theta_1 & \cos \theta_2 & -\sin \theta_3 \\ 0 & 0 & \cos \theta_3 \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} \cos \theta_1 & \cos \theta_2 & \sin \theta_3 \\ -\sin \theta_1 & \cos \theta_2 & -\sin \theta_3 \\ 0 & 0 & \cos \theta_3 \end{bmatrix} \quad (21)$$

$$r_g + eRe_1 = r_g (\cos(\theta) \cos(\phi) e_1 + \sin(\theta) e_2 + \sin(\phi) \cos(\theta) e_3) + eRe_1 \quad (22)$$

$$a = \frac{g}{R} \left(\frac{d}{dt} (\cos(\theta) \cos(\phi)) - 2 \frac{d}{dt} (\sin(\theta)) - \frac{d}{dt} (\cos(\theta) \cos(\phi) + \sin(\theta) \sin(\phi)) \right) \quad (23)$$

$$a = \frac{g}{r} \left(\frac{d}{dt} \sin(\theta) + 2 \frac{d}{dt} (\cos(\phi) \cos(\theta)) \right) \quad (24)$$

$$ar = \frac{d^2}{dt^2} (\sin(\theta) \cos(\phi)) \quad (25)$$

$$\begin{bmatrix} \cos \theta_1 & \cos \theta_2 & \sin \theta_3 \\ -\sin \theta_1 & \cos \theta_2 & -\sin \theta_3 \\ 0 & 0 & \cos \theta_3 \end{bmatrix} \quad (26)$$

$$r_g + eRe_1 = r_g (\cos(\theta) \cos(\phi) e_1 + \sin(\theta) e_2 + \sin(\phi) \cos(\theta) e_3) + eRe_1 \quad (27)$$

$$a = \frac{g}{R} \left(\frac{d}{dt} (\cos(\theta) \cos(\phi)) - 2 \frac{d}{dt} (\sin(\theta)) - \frac{d}{dt} (\cos(\theta) \cos(\phi) + \sin(\theta) \sin(\phi)) \right) \quad (28)$$

$$a = \frac{g}{r} \left(\frac{d}{dt} \sin(\theta) + 2 \frac{d}{dt} (\cos(\phi) \cos(\theta)) \right) \quad (29)$$

$$ar = \frac{d^2}{dt^2} (\sin(\theta) \cos(\phi)) \quad (30)$$

$$\begin{bmatrix} \cos \theta_1 & \cos \theta_2 & \sin \theta_3 \\ -\sin \theta_1 & \cos \theta_2 & -\sin \theta_3 \\ 0 & 0 & \cos \theta_3 \end{bmatrix} \quad (31)$$

$$r_g + eRe_1 = r_g (\cos(\theta) \cos(\phi) e_1 + \sin(\theta) e_2 + \sin(\phi) \cos(\theta) e_3) + eRe_1 \quad (32)$$

$$a = \frac{g}{R} \left(\frac{d}{dt} (\cos(\theta) \cos(\phi)) - 2 \frac{d}{dt} (\sin(\theta)) - \frac{d}{dt} (\cos(\theta) \cos(\phi) + \sin(\theta) \sin(\phi)) \right) \quad (33)$$

$$a = \frac{g}{r} \left(\frac{d}{dt} \sin(\theta) + 2 \frac{d}{dt} (\cos(\phi) \cos(\theta)) \right) \quad (34)$$

$$ar = \frac{d^2}{dt^2} (\sin(\theta) \cos(\phi)) \quad (35)$$

$$X = R_1 \cos(\theta) - R_2 \sin(\theta) \quad (36)$$

$$Y = -R_1 \sin(\theta) - R_2 \cos(\theta) \quad (37)$$

$$w_i = \frac{(y_i - y_T)(x_i - x_T) - (x_i - x_T)(y_i - y_T)}{(x_i - x_T)^2 + (y_i - y_T)^2} \quad (38)$$

$$\frac{(v_T)^T (k(r_i - r_{di}) \phi_i + v_T r_{di} (u + ew_i) A \phi_i)^T}{A(p_T - p_i)} = \frac{ker_i \phi_i + r_{di} (6 + ew_i) (A \phi_i)^T}{r_i} \quad (39)$$

$$A \phi_i = \frac{r_{di} (6 + ew_i)}{r_i} = 6 + ew_i - \frac{er_i (6 + ew_i)}{r_i} \quad (40)$$

$$\dot{p}_{bTi} = bv_{Ti} - a_1 \sigma(b_i^0 (p_{bTi} - p_T) + \sum_{j=1}^N a_{ij} (p_{bTi} - p_{bTj}))^m - b_1 \sigma(b_i^0 (p_{bTi} - p_T) + \sum_{j=1}^N a_{ij} (p_{bTi} - p_{bTj}))^n \quad (41)$$

$$\dot{v}_{bTi} = -a_1 \sigma a_{ij} (v_{bTi} - v_{bTj}) - b_1 \sigma(b_i^0 (v_{bTi} - v_T) + \sum_{j=1}^N a_{ij} (v_{bTi} - v_{bTj}))^n - l_1 \text{sign}(b_i^0 (v_{bTi} - v_T) + \sum_{j=1}^N a_{ij} (v_{bTi} - v_{bTj})) \quad (42)$$

$$\dot{p}_{Ti} = v_{Ti} - a_1 \sigma a_{ij} (p_{Ti} - p_{Tj}) - b_1 \sigma(b_i^0 p_{Ti} + \sum_{j=1}^N a_{ij} (p_{Ti} - p_{Tj}))^n \quad (43)$$

$$\dot{v}_{Ti} = -a_1 \sigma(b_i^0 v_{Ti} + \sum_{j=1}^N a_{ij} (v_{Ti} - v_{Tj}))^m - b_1 \sigma(b_i^0 v_{Ti} + \sum_{j=1}^N a_{ij} (v_{Ti} - v_{Tj}))^n - l_1 \text{sign}(b_i^0 v_{Ti} + \sum_{j=1}^N a_{ij} (v_{Ti} - v_{Tj})) \quad (44)$$

$$\dot{e}_{ri} = \dot{r}_i = \frac{(p_T)^T - (p_i)^T}{(p_T - p_i) r_i} = \frac{(p_T)^T - u_{Ti} \phi_i}{(v_T)^T \left(k \left(\dot{r}_i - r_{di} \right) \dot{\phi}_i + \dot{v}_{Ti} r_{di} \left(6 + \dot{e} w_i \right) A \dot{\phi}_i \right)^T} \quad (45)$$

Methodology

Our methodology is divided into three phases, as shown in Fig. 4, which are centered around the UAV multicircular system design and FOPID controller in phase one; hybrid optimization for controller parameter tuning, such as FOPID-based HESPSOALO, FOPID-based HPSOGWO, and FOPID-based HGWOALO with the pseudocode of these proposed algorithms and their benchmark functions in phase two; and performance evaluation via multicriteria decision making under real-world conditions, such as the CRITIC method for weights and the TOPSIS method for ranks of UAV circumnavigation, which are all included in our detailed explanations of each step.

Phase one: UAV multicircular system

A 3D plane can be used to simulate the motion of the UAV, depending on the requirements of our simulation. The positions of the UAVs are represented by vectors in Cartesian coordinates. In a 2D model, Eq. (46) indicates the position of a UAV i at time (t) ⁴⁵:

$$p_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} \quad (46)$$

where $x_i(t)$ and $y_i(t)$ represent the positions of the UAV along the x- and y-axes, respectively, at time t . The velocity of each unmanned aerial vehicle is also expressed as a vector via Eq. (47):

$$v_i(t) = \begin{bmatrix} v_{x,i}(t) \\ v_{y,i}(t) \end{bmatrix} \quad (47)$$

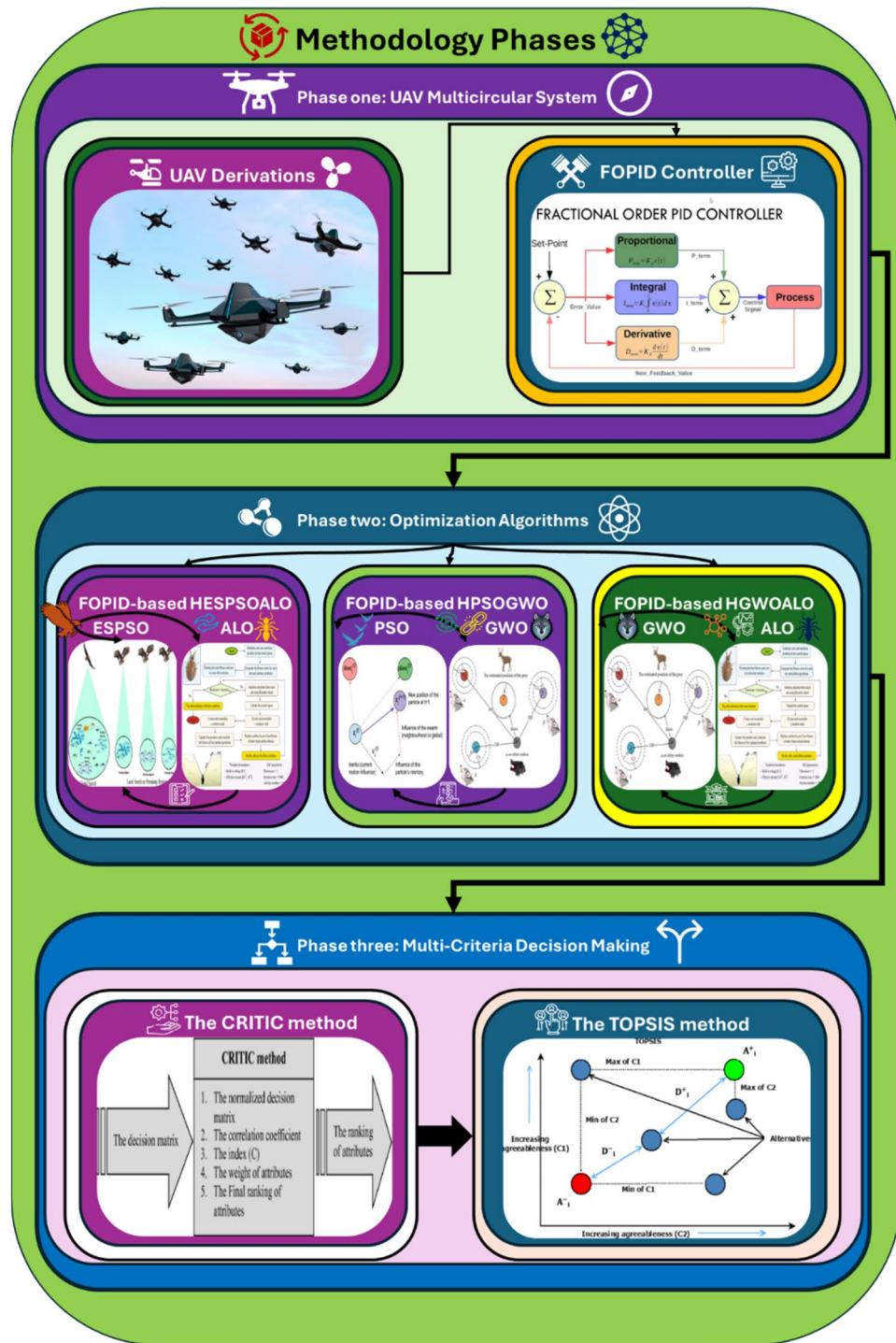


Fig. 4. The proposed methodology phases.

The angle formed by the velocity vector and the y-axis, known as the heading angle, or $\theta_i(t)$, determines the direction of movement of the UAV. Each UAV must circle a central target in a circular path to complete a circumnavigation. Equation (48) provides the desired position of UAV U on the circle at time t^{46} :

$$P_{i,\text{des}}(t) = P_c + r_i \begin{bmatrix} \cos(\omega_i t) \\ \sin(\omega_i t) \end{bmatrix} \quad (48)$$

The position of the central target, or the center of the circular path, is indicated by p_c^{47} . As stated in Tables 2, 3 and 4, the radius of UAV U 's circular trajectory is represented by r_i . The compensation equations in Eqs. (49), (50), (51), (52), and (53), respectively, are as follows:

Parameter name	Values
Initial positions of UAVs (m)	$p1(0) = [-50, -50] T$; $p2(0) = [50, -60] T$; $p3(0) = [-100, -100] T$; $p4(0) = [-200, 0] T$; $p5(0) = [-50, 150] T$; $p6(0) = [-100, 200] T$; $p7(0) = [-50, 300] T$; $p8(0) = [-100, -100] T$
Initial heading angles of UAVs (rad)	$\psi1(0) = -0.3957$; $\psi2(0) = \pi + 0.0977$; $\psi3(0) = 0.4259$; $\psi4(0) = 0.7771$; $\psi5(0) = \pi + 0.3685$; $\psi6(0) = -0.5923$; $\psi7(0) = -0.7151$; $\psi8(0) = 1.1425$

Table 2. Initial conditions of the UAVs.

Parameter name	Values
Initial positions of UAVs (m)	$p1(0) = [50; 50] T$; $p2(0) = [50; 60] T$; $p3(0) = [100; 100] T$; $p4(0) = [200; 0] T$; $p5(0) = [50; 150] T$; $p6(0) = [100; 200] T$; $p7(0) = [50; 300] T$; $p8(0) = [100; 100] T$
Initial heading angles of UAVs (rad)	$\theta1(0) = 0.3957$; $\theta2(0) = \pi + 0.0977$; $\theta3(0) = 0.4259$; $\theta4(0) = 0.7771$; $\theta5(0) = \pi + 0.3685$; $\theta6(0) = -0.5923$; $\theta7(0) = -0.7151$; $\theta8(0) = 1.1425$
Initial velocities of UAVs (m/s)	$v1(0) = [2; 2] T$; $v2(0) = [1; 2] T$; $v3(0) = [1; -1] T$; $v4(0) = [2; 1] T$; $v5(0) = [-2; -2] T$; $v6(0) = [-3; -2] T$; $v7(0) = [3; 3] T$; $v8(0) = [3; -2] T$

Table 3. Initial conditions of distributed observers with a fixed time.

Parameter name	Values
Initial states of positions (m)	$p1(0) = [50; 50] T$; $p2(0) = [100; 200] T$; $p3(0) = [100; 100] T$; $p4(0) = [300; 250] T$; $p5(0) = [100; 100] T$; $p6(0) = [200; 100] T$; $p7(0) = [200; 200] T$; $p8(0) = [200; 100] T$
Initial states of velocities (m/s)	$v1(0) = [100; 100] T$; $v2(0) = [50; 100] T$; $v3(0) = [100; 100] T$; $v4(0) = [100; 50] T$; $v5(0) = [100; 100] T$; $v6(0) = [200; 100] T$; $v7(0) = [200; 200] T$; $v8(0) = [200; 100] T$

Table 4. Initial conditions of distributed fixed-time observers with elevated values.

$$H(e) = \begin{bmatrix} \beta_1 & -1 & 0 \\ \beta_2 \frac{m \text{fal}(e_1, A, M)}{e_1} & 0 & 1 \\ \beta_3 \frac{m \text{fal}(e_2, A, M)}{e_2} & 0 & 0 \end{bmatrix} \quad (49)$$

$$D = \begin{bmatrix} 1 & \frac{\beta_2 F_1}{B + \epsilon_1} - \eta \\ -\frac{\beta_2 F_1}{B - \epsilon_1} \eta & \frac{1}{B + \epsilon_2} \eta - \frac{1}{B - \epsilon_2} \eta \end{bmatrix} \quad (50)$$

$$V(t) = \int_0^t (DH(e)e, \dot{e}) dt + C_0 \quad (51)$$

$$V(t) = \int_0^t (DH(e)e, -H(e)e) dt + C_0 \quad (52)$$

$$V(t) = \int_0^t (H(e)e)' \cdot DH(e)e dt + C_0 \quad (53)$$

FOPID controller

The transfer function for the FOPID controller is given by Eq. (54):

$$C(s) = K_p + K_i s^{-\lambda} + K_d s^\mu \quad (54)$$

In this case, K_p represents the proportional gain, K_i represents the integral gain, K_d represents the first derivative gain, λ ($0 < \lambda < 1$) represents the integration order, and μ ($0 < \mu < 1$) represents the derivative order. The contributions of the fractional, integral, and proportional derivatives to the control of the UAV multicircular system are shown in this equation.

To noninteger (fractional) orders, fractional calculus extends conventional integer-order differentiation and integration. Equation (55) represents the general fractional derivative operator as follows⁴⁸:

$$\frac{d^\mu}{dt^\mu} f(t) \quad (55)$$

The fractional derivative operator is s^μ , and the fractional integral operator is $s^{-\lambda}$. Equation (56) expresses the overall control law in the time domain:

$$u(t) = K_p e(t) + K_i \frac{d^{-\lambda}}{dt^{-\lambda}} e(t) + K_d \frac{d^\mu}{dt^\mu} e(t) \quad (56)$$

where $K_p e(t)$ is the equivalent contribution, $K_i \int_0^t e(\tau) d\tau^\lambda$ uses a fractional order to integrate the error, and $K_d s^\mu e(t)$ distinguishes the mistake using the fractional order.

Phase two: optimization algorithms

In this phase, we apply advanced hybrid optimization algorithms to optimize the control parameters for UAVs operating within a distributed multicircular circulation-avigation control (DMCC) framework. The goal is to minimize errors in position, velocity, and orientation (yaw, pitch, roll) during UAV flight by tuning the control parameters for both PID and fractional-order PID (FOPID) controllers.

The optimization algorithms used in this phase are FOPID-based HESPSOALO, FOPID-based HPSOGWO, and FOPID-based HGWOALO. These algorithms further ensure that they are robust algorithms such that both exploration and exploitation techniques are utilized to search for optimal parameters. Each algorithm is discussed in detail with the corresponding equations⁴⁹:

The parameters are optimized for UAV control: the proportional gain (K_p), integral gain (K_i), and derivative gain (K_d) for the PID controllers; for the FOPIDs, there are additional fractional order components (λ and μ). These parameters are indispensable for controlling the UAVs precisely and stably to move along the desired circular paths. The objective function to be optimized seeks a low overall performance of the UAV on the basis of positional, velocity, and angular errors. The objective function J is given by Eq. (57):

$$J = w_p \sum_{i=1}^n (\text{PosError}(i))^2 + w_v \sum_{i=1}^n (\text{VelError}(i))^2 + w_a \sum_{i=1}^n (\text{AngError}(i))^2 \quad (57)$$

where w_p , w_v and w_a are the weights assigned to the position error, velocity error and yaw, pitch and roll, respectively. The errors are measured for each UAV i over sets of time intervals. The objective function we would like to minimize is related to the control parameters, as are the optimization algorithms.

FOPID-based HESPSOALO

The FOPID-based HESPSOALO algorithm is a hybrid approach that combines three key components: the global exploration solved via the Eagle strategy (ES) for the ES, particle swarm optimization (PSO) for population-based optimization and the ant lion optimizer (ALO) for the refinement of the solutions during exploitation. The hybrid method is designed to search for the solution space thoroughly and structure the results.

First, among all the strategies, there is global exploration that emulates the Eagle strategy (ES), which is applied as the first stage of HESPSOALO. Levy flights, a type of random walk where long jumps are characterized, are utilized in the Eagle strategy. With the addition of Levy flights, the algorithm's exploration capability is improved such that the particles (candidate solutions) jump to faraway regions of the search space. The position of a particle after a Levy flight is updated via Eq. (58):

$$X_{t+1} = X_t + \alpha \cdot |V_t|^{\frac{1}{\beta}} \sigma \quad (58)$$

where X_t is the current position, α is a scaling factor, σ is a normally distributed random variable, and β controls the jump distribution. Levy flight ensures that the particles explore large portions of the solution space to avoid becoming trapped in a local minimum.

Following the exploration phase, the algorithm transitions into the exploitation phase, which uses both particle swarm optimization (PSO) and the ant lion optimizer (ALO). In the PSO component, particles move through the search space by adjusting their positions and velocities on the basis of their personal best positions and the global best position within the swarm. The velocity of each particle is updated via Eq. (59)^{50,51}:

$$v_{i,j}(t+1) = w \cdot v_{i,j}(t) + c_1 \cdot r_1 \cdot (p_{best,j} - x_{i,j}(t)) + c_2 \cdot r_2 \cdot (g_{best,j} - x_{i,j}(t)) \quad (59)$$

where $v_{i,j}(t)$ is the velocity of particle i in dimension j at time step t , w is the inertia weight controlling the influence of the previous velocity, c_1 and c_2 are cognitive and social acceleration coefficients, r_1 and r_2 are random numbers between 0 and 1, and $p_{best,j}$ and $g_{best,j}$ represent the particle's personal best position and the global best position in dimension j , respectively. The position of the particle is then updated via Eq. (60):

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (60)$$

Simultaneously, the ant lion optimizer (ALO) is applied to refine the search process by simulating the hunting mechanism of ant lions. The ALO random walk model of prey movement within the search space is presented in Eq. (61):

$$X_{t+1} = X_t + (r_1 - r_2) \quad (61)$$

Here, r_1 and r_2 are random numbers representing the prey's movement, and X_t is the current position of the ant lion. The ability of ALO to trap and refine solutions ensures that the parameters are fine-tuned during the exploitation phase, providing a strong local search capability that complements the global exploration of PSO. The pseudocode for Algorithm 1 is as follows:

1. **Inputs:** Population size N , Maximum iterations ($MaxIter$), Set the parameter bounds for FOPID(Kp, Ki, Kd, λ, μ)
2. **Output:** Best solution X^*
3. Initialize the population of particles X_i ($i = 1, 2, \dots, N$)
4. Initialize velocity v_i for each particle
5. Assign personal best p_best and global best g_best
6. Initialize control parameters for Eagle Strategy, PSO, and ALO
7. **while** (stopping criteria not met) **do**
8. Evaluate the fitness of each particle using the FOPID controller performance
9. Apply Eagle Strategy for exploration:
10. **For** each particle i (**for** $i = 1, 2, \dots, N$)
11. Perform Levy flight-based exploration:
12. $X_i(t+1) = X_i(t) + Levy(\beta) \times step_size$
13. **else**
14. Apply PSO for exploitation:
15. Update position of the particle:
16. $X_i(t+1) = X_i(t) + v_i(t+1)$
17. Apply ALO for adaptive random walk
18. Update position using the ant lion adaptive random walk:
 $X_i(t+1) = 2X_{min} + X_{max} + (2X_{min} - X_{max}) \times R$
19. Bound the updated position within parameter limits
20. Evaluate the fitness of the new position $X_i(t+1)$
21. Update $p_best[i]$ and g_best if necessary
22. Update control parameters for PSO, ALO, and Eagle Strategy
23. **end while**
24. **Return** best solution X^*

Algorithm 1. Hybrid Eagle Strategy Particle Swarm Optimization Ant Lion Optimizer (HESPSOALO)

FOPID-based HPSOGWO

The gray wolf optimization (GWO) algorithm is integrated with particle swarm optimization (PSO) to generate the FOPID-based HPSOGWO algorithm. We design this hybrid approach to combine the fast convergence of PSO and efficient exploration in GWO.

The gray wolf optimization (GWO) algorithm is inspired by the social hierarchy structure and hunting behavior of gray wolves. In GWO, the population of wolves is divided into four groups: the alpha, beta, delta, and omega modules. The alpha wolves represent the best solution, followed by the beta and delta wolves, who represent almost equally bad solutions; finally, the rest of the population consists of the omega wolves. At each iteration, the alpha, beta and delta wolves work as guides to move toward the optimal solution. The position of a gray wolf is updated as in Eq. (62)⁵²:

$$X(t+1) = 3X^\alpha + X^\beta + X^\delta \quad (62)$$

The alpha, beta, and delta wolves are at positions X^α , X^β , and X^δ . The three best solutions among these wolves are at any iteration. The encircling behavior of the wolves is modeled via Eq. (63) and Eq. (64):

$$D = |C \cdot X_p(t) - X(t)| \quad (63)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (64)$$

These are equations (with subscript p on $X_p(t)$ for the position of the prey (optimal solution), coefficient vector A, coefficient vector C controlling the wolves' movements, and D, the distance between the area of the prey and the wolf). Particle swarm optimization (PSO) is used after GWO narrows it down to search. PSO adjusts the positions and velocities of the particles on the basis of their own experience (personal best) and the swarm's

experience (global best). This hybridization ensures that the algorithm balances exploration and exploitation, ultimately leading to more accurate tuning of the UAV control parameters⁵³. The pseudocode for Algorithm 2 is as follows:

-
1. **Inputs:** Population size N , Maximum iterations ($MaxIter$), Set the parameter bounds for FOPID(Kp, Ki, Kd, λ, μ)
 2. **Output:** Best solution X^*
 3. Initialize the population of particles X_i ($i = 1, 2, \dots, N$) within the bounds of FOPID parameters.
 4. Initialize velocity v_i for each particle i
 5. Set personal best positions $p_best[i]$ and initialize the global best position g_best .
 6. Initialize the control parameters for PSO and GWO components.
 7. Initialize the top three wolves (α, β, δ) for the Gray Wolf Optimizer (GWO).
 8. **While** (stopping criteria not met):
 9. Evaluate the fitness of each particle based on the performance of the FOPID controller (minimizing position and angular errors, improving stability, etc.).
 10. Apply PSO for position updates:
 11. **For** each particle i in the population $i = 1, 2, \dots, N$:
 12. Update the velocity of each particle:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_best[i] - X_i(t)) + c_2 \cdot r_2 \cdot (g_best - X_i(t))$$
 13. Update the position of each particle:

$$X_i(t+1) = X_i(t) + v_i(t+1)$$
 14. Apply GWO for further exploitation:
 15. **For** each particle i , update the position using the three best wolves:

$$D_\alpha = |C_1 \times \alpha - X_i(t)|$$

$$A_1 = 2a \times r_1 - a$$

$$X_1 = \alpha - A_1 \times D_\alpha$$

$$D_\beta = |C_2 \times \beta - X_i(t)|$$

$$A_2 = 2a \times r_2 - a$$

$$X_2 = \beta - A_2 \times D_\beta$$

$$D_\delta = |C_3 \times \delta - X_i(t)|$$

$$A_3 = 2a \times r_3 - a$$

$$X_3 = \delta - A_3 \times D_\delta$$
 16. Update the particle's position:

$$X_i(t+1) = 3X_1 + X_2 + X_3$$
 17. Bound the updated position within the limits of the FOPID parameters(Kp, Ki, Kd, λ, μ)
 18. Evaluate the fitness of the new position $X_i(t+1)$
 19. Update personal best $p_best[i]$ and global best g_best if the new fitness is better than previous values.
 20. Update control parameters for PSO (e.g., inertia weight w) and GWO (adjust a parameter for controlling exploration and exploitation).
 21. **End while**
 22. **Return** best solution X^* , representing the optimal FOPID parameters (Kp, Ki, Kd, λ, μ).
-

Algorithm 2. Hybrid Particle Swarm Optimization-Gray Wolf Optimizer (FOPID-based HPSOGWO)

FOPID-based HGWOALO

HGWOALO is a hybrid optimization algorithm that uses gray wolf optimization (GWO) to generate its population, which then undergoes the ant lion optimizer (ALO) to find its final population. The hybridization of this type is used to provide the optimization process with improved global exploration and local exploitation

abilities, making it possible to tune the control parameters for fractional-order PID (FOPID) controllers in an appropriate way. This optimization algorithm attempts to minimize positional, velocity and angular errors in the control of unmanned aerial vehicles (UAVs) via a distributed multicircular circulation control (DMCC) system.

In an FOPID controller, the control parameters to be optimized are K_p , K_i , and K_d , along with two additional fractional parameters, λ and μ , associated with the fractional orders of the integral and derivative components, respectively. The control law for an FOPID controller is expressed in Eq. (65):

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \quad (65)$$

Using the assumed structure, the proportional, integral, and derivative gains, K_p , K_i , and K_d , can be combined with $e(t)$ and $u(t)$ such that $u(t)$ is the control output associated with the error at time t . The fractional calculus operators for integration are $D_t = -\lambda$ and $D_t = \mu$. Having applied the HGWOALO optimization algorithm to minimize the errors in UAV control by varying these FOPID parameters, we hold an implementation available on GitHub. This algorithm operates in two main phases: for global exploration of the parameter space, an implementation of gray wolf optimization (GWO) is used, and for local exploitation of the parameter space, the ant lion optimizer (ALO) is used. This combined approach results in both a thorough search of the solution space and a fine-tuned parameter adjustment during exploitation. On the basis of the social hierarchy and hunting mechanism of gray wolves, gray wolf optimization (GWO) is introduced. In this algorithm, the population of wolves (candidate solutions) is divided into four types: gamma, delta, alpha, and omega wolves. The alpha, beta, and delta wolves represent the top three best solutions, whereas the remaining wolves (omega) explore the search space. The wolves surround and move toward the prey, which represents the optimal solution. The position of a gray wolf is updated on the basis of the positions of the alpha, beta, and delta wolves via Eq. (66):

$$X(t+1) = \frac{1}{3} * (X^{\alpha}(t) + X^{\beta}(t) + X^{\delta}(t)) \quad (66)$$

where $X(t+1)$ is the updated position of a gray wolf at iteration $t+1$ and where $X^{\alpha}(t)$, $X^{\beta}(t)$, and $X^{\delta}(t)$ are the positions of the alpha, beta, and delta wolves (the best, second-best, and third-best solutions, respectively) at iteration t . As the ant lion refines its trap, the search space around the prey is progressively reduced, forcing the prey toward the optimal solution. The shrinking of the search boundaries is mathematically modeled as in Eq. (67)^{54,55}:

$$I = \frac{\text{iter}_{\max}}{I_t}, f = \frac{\text{iter}_{\max}}{f_t} \quad (67)$$

I and f are the bounds present for the search space, and $\text{iter}_{\max}$ is the number of iterations. The search space enclosed decreases, and the prey is constrained from an optimal solution within decreasing boundaries.

As a hybrid method, FOPID-based HGWOALO is extremely powerful in that it combines the global searching capability of the gray wolf optimizer (GWO) and the local refinement feature of the ant lion optimizer (ALO). By combining this hybrid approach to explore the solution space with fine-tuning solutions toward convergence to optimal control parameters, efficiency is guaranteed, whereas efforts focused on solution fine-tuning converge toward optimal control parameters. HGWOALO achieves an effective optimization technique to control DMCC systems by minimizing the objective function that measures positional, velocity, and angular errors. The pseudocode for Algorithm 3 is as follows:

1. **Inputs:** Population size N , Maximum iterations ($MaxIter$), Set the parameter bounds for FOPID(Kp, Ki, Kd, λ, μ)
2. **Output:** Best solution X^*
3. Initialize the population of particles X_i ($i = 1, 2, \dots, N$) within the bounds of FOPID parameters.
4. Set initial positions for the top three wolves (α, β, δ) for the Gray Wolf hierarchy.
5. Initialize the control parameters for the Gray Wolf Optimizer (GWO) and Ant Lion Optimizer (ALO) components.
6. Assign personal best $p_best[i]$ for each wolf and initialize global best g_best .
7. **While** (stopping criteria not met):
8. Evaluate the fitness of each wolf using the FOPID controller performance in UAV Multicircular flight.
9. Apply GWO for exploration and exploitation:
10. **For** each wolf i :
11. Calculate the distance of each wolf from α, β , and δ :

$$D_\alpha = |C_1 \times \alpha - X_i(t)|$$

$$D_\beta = |C_2 \times \beta - X_i(t)|$$

$$D_\delta = |C_3 \times \delta - X_i(t)|$$
12. Update the position based on α, β , and δ :

$$X_1 = \alpha - A_1 \times D_\alpha$$

$$X_2 = \beta - A_2 \times D_\beta$$

$$X_3 = \delta - A_3 \times D_\delta$$
13. The new position for the wolf is calculated as:

$$X_i(t+1) = 3X_1 + X_2 + X_3$$
14. Apply ALO for additional exploration:
15. The wolves perform an adaptive random walk simulating ant lion behavior:

$$X_i(t+1) = 2X_{min} + X_{max} + (2X_{min} - X_{max}) \times R$$
16. Where X_{min} and X_{max} are the bounds of the FOPID parameters, and R is a random walk value generated by the ant lion's adaptive behavior.
17. Bound the updated position within the limits for FOPID parameters(Kp, Ki, Kd, λ, μ)
18. Evaluate the fitness of the new position $X_i(t+1)$ based on the FOPID controller performance (e.g., positional errors, angular stability).
19. Update personal best $p_best[i]$ and global best g_best if the fitness of $X_i(t+1)$ is better than previous values.
20. Update control parameters for GWO and ALO mechanisms. Adjust exploration and exploitation factors in GWO (a) and ALO's random walk adaptability.
21. **End while**
22. **Return** best solution X^* , representing the optimal FOPID parameters (Kp, Ki, Kd, λ, μ).

Algorithm 3. FOPID-Based Hybrid Gray Wolf Optimization Ant Lion Optimizer (HGWOALO)

Benchmark functions

A set of benchmark functions was employed to evaluate the performance of our proposed optimization algorithms: FOPID-based HESPSOALO (Hybrid Eagle Strategy Particle Swarm Optimization Ant Lion Optimizer), FOPID-based HPSOGWO (Hybrid Particle Swarm Optimization-Gray Wolf Optimization), and FOPID-based HGWOALO (Hybrid Gray Wolf Optimization-Ant Lion Optimizer). The benchmark functions selected for our tests cover a wide range of optimization problems, testing the algorithms' ability to find global minima while avoiding local optima, balancing exploration and exploitation, and handling unimodal and multimodal search landscapes. The algorithms were tested for their capacity to quickly converge to the global minimum of a smooth, convex landscape with the sphere function, which is a simple uniform function. The sphere function has no local minima, rendering it an ideal starting point to determine how well an algorithm will find its way to a minimum without the distraction of false minima. The Rosenbrock function is a harder function than the one given earlier because it includes a narrow, curved valley to the global minimum. Especially useful for testing how well your algorithms can optimize the complex landscape they travel into the global minimum, with that sensitivity and careful navigation around the valley. Its landscape is highly multimodal and is characterized by many local minima in the Ackley function. However, in the global minimum, we are in a deep and very narrow basin. This function measures how well the algorithms balance exploration and exploitation so that they are not trapped in local minima because they have no need to explore the whole global landscape.

In addition, a few tests, such as the Griewank function, Rastrigin function, Levi function and Booth function, were chosen to test the robustness and precision of the algorithms. These functions are known to be periodic, multimodal, and complex search spaces, which is key to testing the convergence of the algorithms to the global optimum as fast as possible. For our evaluations, we ran each algorithm through these benchmark functions to assess their performance on the basis of several key metrics: speed of convergence, precision in the search for the global minimum, and robustness in escaping from local minima⁵⁶. We list the detailed setup and specifications of each benchmark function along with its parameters in Table 5.

Phase three: multi-criteria decision making

FOPID-based HESPSOALO, FOPID-based HPSOGWO and FOPID-based HGWOALO are used to optimize the UAV control parameters. To select, evaluate and rank the alternatives, MCDM is applied. This phase considers the inherent trade-offs between multiple performance metrics and translates to the selection of the best control strategies for a more comprehensive evaluation. There are two stages of this applied model. Stage I shows the weights of the criteria based on CRITIC. Stage II prioritizes the alternatives via TOPSIS.

The CRITIC method

The CRITIC is a reliable objective weighting technique. It measures the criteria's weight according to the contrast intensity of a criterion and the degree of conflict between the criteria. The contrast intensity refers to the difference between the values of a criterion in each evaluation scheme. The conflict degree measures the similarity of information between different criteria. A criterion with a higher contrast intensity and higher degree of conflict should have a larger weight. The main advantage of CRITIC lies in its ability to detect the interdependencies among the criteria, enhancing the assessment of their relative significance⁵⁷.

Given the decision matrix “D” with n alternatives and m criteria, $r_{ij} \in R$ is the rating of the i^{th} alternative for the j^{th} criterion, where R is the set of real numbers as in Eq. (68).

$$\begin{array}{c|ccccc} \mathbf{D} & C_1 & C_2 & \cdots & C_m \\ \hline A_1 & r_{11} & r_{12} & \cdots & r_{1m} \\ A_2 & r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_n & r_{n1} & r_{n2} & \cdots & r_{nm} \end{array} \quad (68)$$

Step 1 Normalize the decision matrix

For every criterion, a membership function x_{ij} is defined that maps the value r_{ij} to the interval $[0, 1]$. This is accomplished via the following formulas: Eq. (69) and Eq. (70).

$$x_{ij} = \frac{r_{ij} - r_j^-}{r_j^+ - r_j^-}, \text{ for benefit criteria,} \quad (69)$$

$$x_{ij} = \frac{r_j^+ - r_{ij}}{r_j^+ - r_j^-}, \text{ for cost criteria,} \quad (70)$$

Equation (71) depicts $r_j^+ = \max_i r_{ij}$, and $r_j^- = \min_i r_{ij}$.

Benchmark function	Algorithm	Dimensionality	Search agents	Search range
$f_1(x, y) = x^2 + y^2$	FOPID-based HESPSOALO	2	20	$[-5, 5]$
$f_2(x, y) = 20 + (x^2 - 10 \cdot \cos(2\pi x)) + (y^2 - 10 \cdot \cos(2\pi y))$		2	20	$[-5, 5]$
$f_3(x, y) = -20 \exp\left(-0.2 \sqrt{0.5(x^2 + y^2)}\right) - \exp(0.5(\cos(2\pi x) + \cos(2\pi y))) + 20 + e$		2	20	$[-5, 5]$
$f_4(x, y) = 1 + \frac{x^2 + y^2}{4000} - \cos(x) \cdot \cos\left(\frac{y}{\sqrt{2}}\right) + 20 + e$	FOPID-based HPSOGWO	2	20	$[-5, 5]$
$f_5(x, y) = \sin^2(3\pi x) + (x - 1)^2 (1 + \sin^2(3\pi y)) + (y - 1)^2 (1 + \sin^2(2\pi y))$	FOPID-based HGWOALO	2	20	$[-10, 10]$
$f_6(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$		2	20	$[-5, 5]$
$f_7(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$		2	20	$[-5, 5]$
$f_8(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$	General (All algorithms)	2	20	$[-5, 5]$
$f_9(x, y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2$		2	20	$[-5, 5]$

Table 5. Benchmark functions for our proposed algorithms.

D_N	C_1	C_2	$\dots$	C_m
A_1	x_{11}	x_{12}	$\dots$	x_{1m}
A_2	x_{21}	x_{22}	$\dots$	x_{2m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_n	x_{n1}	x_{n2}	$\dots$	x_{nm}

(71)

Step 2 Computing the degree of conflict

Pearson's correlation coefficient between the j th criterion and the k th criterion is computed via Eq. (72).

$$PCOR(C_j, C_k) = \frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 \sum_{i=1}^n (x_{ik} - \bar{x}_k)^2}}, \quad (72)$$

where $\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$ is the mean of the alternatives' ratings for the j th criterion and where $\bar{x}_k = \frac{1}{n} \sum_{i=1}^n x_{ik}$ is the mean of the alternatives' ratings for the k th criterion.

As the correlation coefficient decreases, the degree of conflict increases. Hence, the degree of conflict for each criterion is calculated via Eq. (73).

$$Pdeg(C_j) = \sum_{k=1}^m (1 - PCOR(C_j, C_k)). \quad (73)$$

Step 3 Compute the contrast intensity

The standard deviation of the j th criterion is calculated via Eq. (74).

$$P\sigma_j = \sqrt{\frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}{(n-1)}}. \quad (74)$$

Step 4 Find the index of the criteria

The index of the j th criterion is found by multiplying (73) and (74), as expressed in Eq. (75).

$$PIn(C_j) = \sigma_j * deg(C_j). \quad (75)$$

Step 5 Determine the weights of the criteria

The weights of the criteria are obtained by normalizing the indices of the criteria depicted in Eq. (76).

$$W(C_j) = \frac{In(C_j)}{\sum_{j=1}^m In(C_j)}. \quad (76)$$

TOPSIS method

TOPSIS is one of the most widely applied classical MCDM methods^{58–69}. Like CRITIC, TOPSIS is initiated by the normalization procedure, followed by the formulation of the weighted decision matrix. After that, the best and worst solutions are determined. Then, the separation measures and the relative closeness coefficient are calculated for each alternative. Finally, the alternatives are ranked in decreasing order. The best alternative is the one with the largest relative closeness coefficient. The formulas that describe the procedure are defined as follows.

Step 6 Form the weighted decision matrix

The weighted decision matrix is formed by multiplying the normalized evaluations by the weights of the criteria via Eq. (77) and Eq. (78).

$$\omega_{ij} = w_j * x_{ij} \quad (77)$$

D_w	C_1	C_2	$\dots$	C_m
A_1	ω_{11}	ω_{12}	$\dots$	ω_{1m}
A_2	ω_{21}	ω_{22}	$\dots$	ω_{2m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_n	ω_{n1}	ω_{n2}	$\dots$	ω_{nm}

(78)

Step 7 Determine the extreme solutions

Since Eqs. (2) and (3) are used in normalization, the extreme solutions are given as follows:

$$\omega_j^+ = \max_i \omega_{ij}, \text{ and } \omega_j^- = \min_i \omega_{ij}, \text{ for } j = 1, \dots, m \text{ and } i = 1, \dots, n$$

Step 8 Calculate the separation measures

The measures of separation between alternatives and the extreme solutions are computed via Eq. (79) and Eq. (80):

$$S_i^+ = \sqrt{\sum_{j=1}^m (\omega_{ij} - \omega_j^+)^2}, i = 1, \dots, n \quad (79)$$

$$S_i^- = \sqrt{\sum_{j=1}^m (\omega_{ij} - \omega_j^-)^2}, i = 1, \dots, n. \quad (80)$$

Step 9 The relative closeness coefficient is calculated via Eq. (81):

$$R(A_i) = \frac{S_i^-}{S_i^+ + S_i^-} \quad (81)$$

Step 10 Prioritizing the alternatives

The alternatives can be prioritized according to the descending order of the relative closeness coefficient.

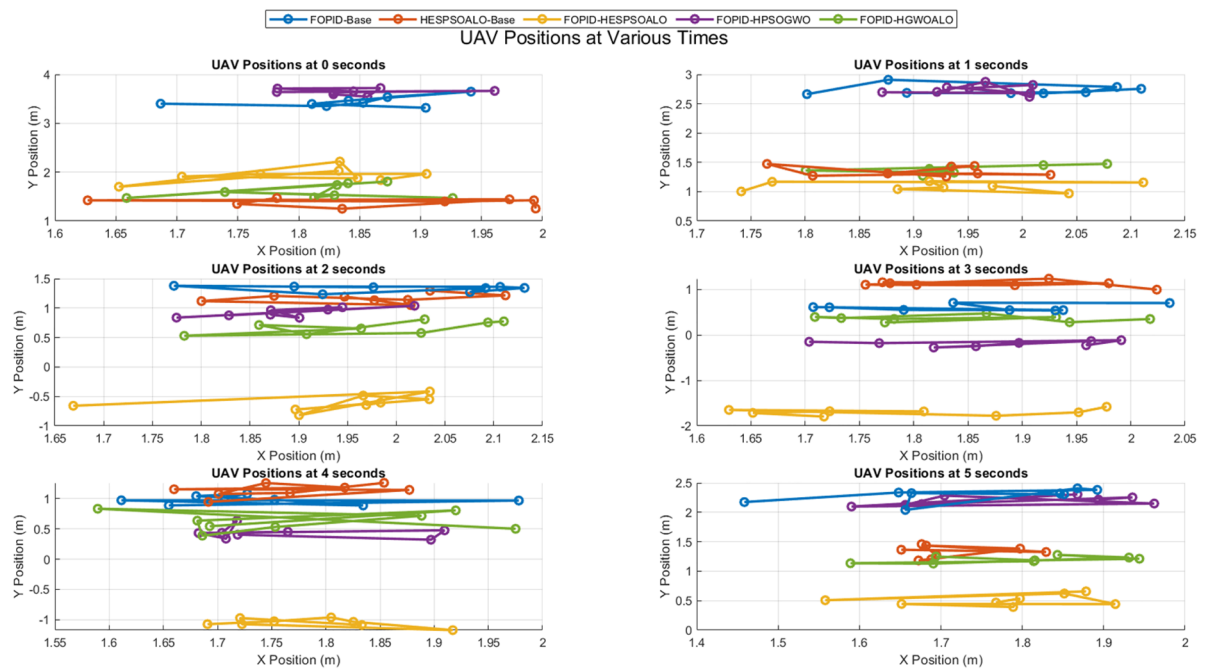
Results and discussion

This section includes a detailed analysis of the optimization algorithms for the UAV multicircular control system as well as the benchmark function testing results and the MCDM results when the best algorithm is selected. Each subsection reviews in detail the performance of the algorithms in managing the UAV control system on the basis of the capacity to minimize the number of errors and stabilize the system.

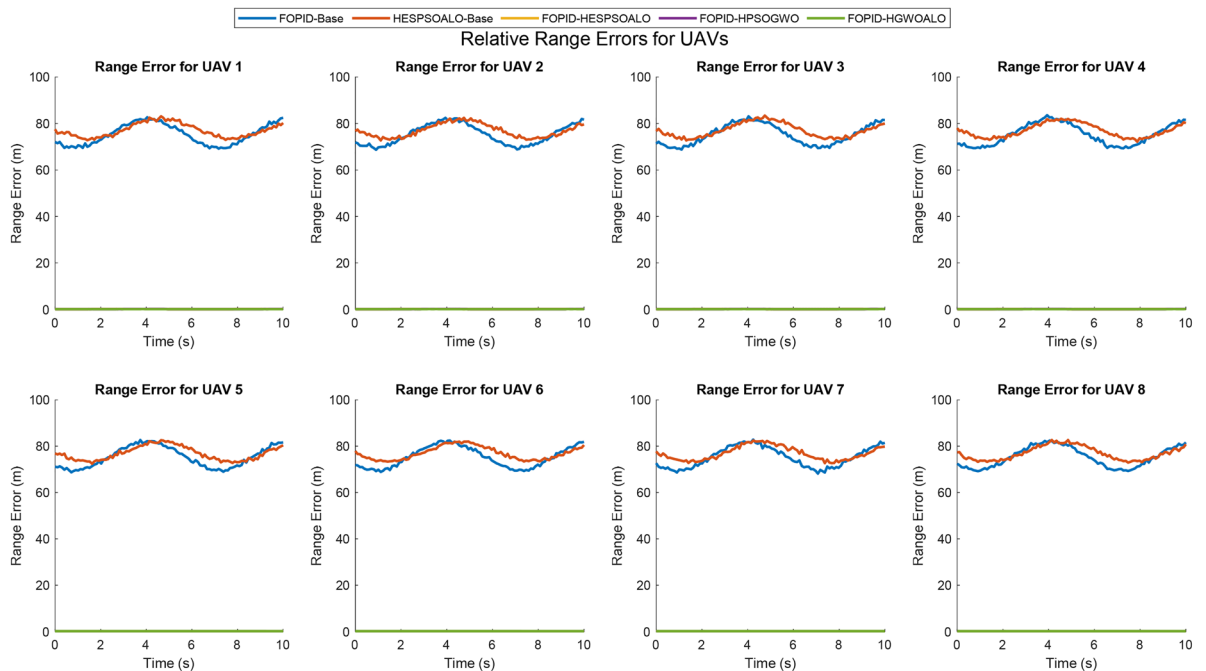
Results of the optimization algorithms for UAV multicircular

The performance of various control algorithms was analyzed with a UAV multicircular model for disturbance reduction in this study. FOPID and its hybrid forms, including FOPID-based HESPSOALO, and other hybrid control methods, such as HPSOGWO and HGWOALO, were compared in the second case. Throughout this work, the goal was to evaluate how these algorithms regulate the control of UAVs in multicircular flight scenarios, aiming specifically to minimize positional, angular and velocity errors. The results presented in Fig. 5 are divided into four main parts: Part (A): UAV positions over time; Part (B): relative range errors for UAVs; Part (C): angular errors (Yaw, Pitch, and Roll); and Part (D): overall MSE comparison for position, velocity, yaw, pitch, and roll. We evaluate the performance of this controller in terms of the position and velocity containment of the UAV by the UAVs as the proportional, integral and derivative gains are adjusted. The FOPID controller works with five dimensions (five controllers gain K_p , K_i , K_d , λ , and μ). While the performance of the FOPID controller is reasonable, compared with that of the hybrid optimization algorithms, it cannot produce comparable levels of positional and angular error minimization. However, we found that optimization was needed to improve controller performance because it could not reduce errors as effectively as the hybrid methods could. This study shows that HESPSOALO, which uses the FOPID as its optimizer, delivered the best results. The hybrid approach involving particle swarm optimization (PSO) and the ant lion optimizer (ALO) uses the Eagle strategy to optimize the balance between exploration and exploitation. A population size of 30 was chosen for the algorithm, which gives a heterogeneous set of candidate solutions, and the inertia weight was initially set to 0.7 and decreased to 0.4. In this configuration, before iterations diverge significantly from the point of good solution, the algorithm has more opportunity to expand its search space or tune the best solutions. The control performance of the HESPSOALO algorithm was also superior to that of the other algorithms, with both positional and angular errors improving for all time intervals. Over tiled matrices, its ability to avoid local optima and efficiently traverse the solution space makes it the most effective mechanism for optimizing the FOPID controller in this setup. Second, FOPID + HPSOGWO was proposed, which combines the gray wolf optimizer (GWO) with PSO to achieve UAV control. In this hybrid method, we adopted a population size of 25 search agents to achieve the reinforcement of both global exploration and local exploitation. Although the improvement over the base FOPID was somewhat greater than that over the base FOPID, it has not yet improved the HESPSOALO algorithm. In terms of error minimization, HPSOGWO achieves good performance but occurs in local optima traps to generate larger angular errors than HESPSOALO does. Similarly, the FOPID + HGWOALO algorithm performs FOPID controller tuning on the basis of the strengths of GWO and ALO. This algorithm's performance in controlling UAV flight for a population size of 20 and an inertia weight of 0.6 was moderate. However, these methods are not competitive at minimizing yaw and roll errors, falling behind HESPSOALO and HPSOGWO. The hybrid structure of HGWOALO permitted it to perform significantly better than the base FOPID controller, but it still faced some challenges in maintaining the level of precision of the top-performing algorithms. The HESPSOALO base algorithm was finally used without integration into the FOPID controller to serve as the baseline for comparison. On its own, the base HESPSOALO, with a population size of 30 and the same parameter configuration, revealed very strong results. However, further integration with the FOPID controller demonstrated the power of applying optimization algorithms for tuning FOPID controller parameters.

In Part (A), the UAV trajectory at time points ranging from 0 to 5 s is plotted to compare each control algorithm with its desired trajectory. The five algorithms that were compared are the following: the FOPID controller, HESPSOALO base, FOPID-based HESPSOALO, FOPID + HPSOGWO, and FOPID + HGWOALO.



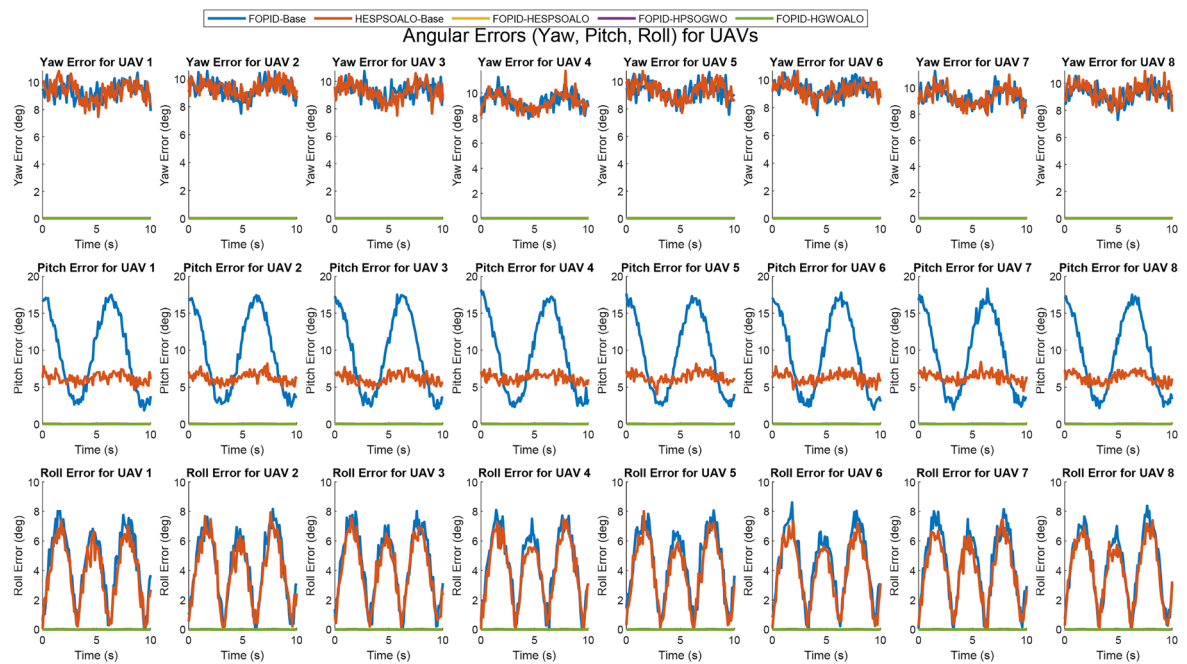
(A) UAV Positions at Various Times.



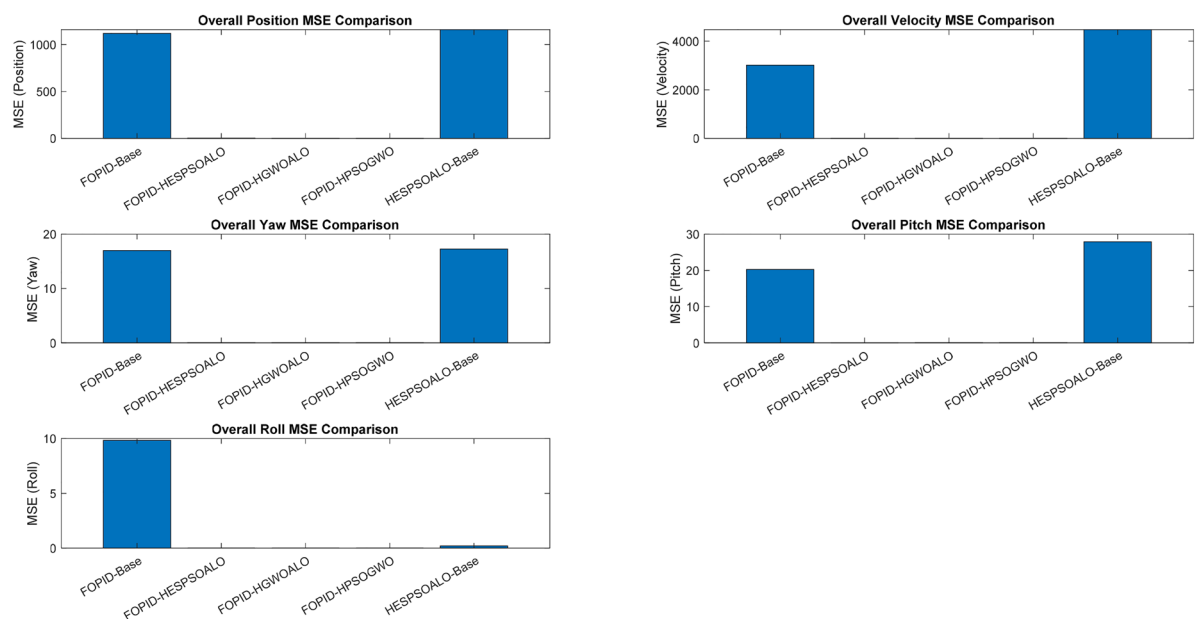
(B) Relative Range Errors for UAVs.

Fig. 5. Performance comparison of UAV control algorithms on the basis of position, range, angular error, and MSE.

Among various controllers, the FOPID-based HESPSOALO controller achieves the most stable performance, as UAVs maintain their positions more accurately for all intervals of time. However, the FOPID + HGWOALO and HPSOGWO algorithms have a larger positional deviation—particularly at later time steps where the UAVs start to deviate slightly from their intended path. The hybrid HESPSOALO algorithm greatly enhances the ability of the FOPID controller to hold the UAV position in dynamic environments. Its hybrid structure balances



(C) Angular Errors (Yaw, Pitch, Roll) for UAVs.



(D) Overall MSE Comparison for Position, Velocity, Yaw, Pitch, and Roll.

Fig. 5. (continued)

exploration and exploitation and improves control precision without allowing the UAVs to travel too far from their desired circular path.

In Part (B), the results of the relative range errors between the actual and desired UAV positions for eight UAVs are plotted. Range errors are measured through time, and the HESPSOALO method, which is based on the FOPID control method, is shown to generate the smallest range errors for these control methods. This algorithm has a strong ability to correct and minimize positional discrepancies during flight, but the average range error of this algorithm remains low throughout. However, the range errors of the base FOPID controller and HPSOGWO are greater than those throughout the transitions between time intervals. The superior performance of HESPSOALO is due to its hybrid nature. The method combines Particle Swarm Optimization

(PSO) and Ant Lion Optimizer (ALO) to avoid the local optimum problem, which is a common problem in single-strategy algorithms. Embed in the Eagle strategy into HESPSOALO is the ability to dynamically switch between global exploration and local exploitation to provide precise control over the UAV’s range dynamically during multicircular flight.

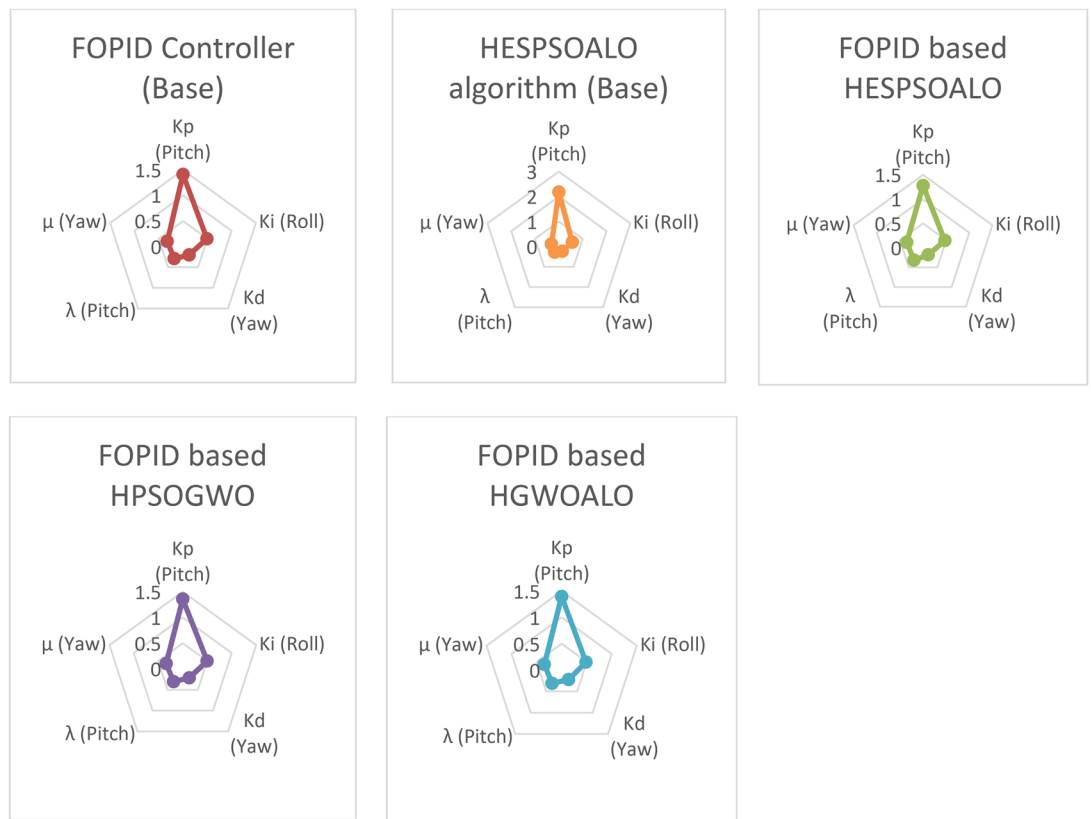
The yaw, pitch, and roll angular errors for the eight UAVs over time are presented in Part (C). The performance of reducing the maximum errors in these two criteria is consistently better for the FOPID-based HESPSOALO than for the other control strategies. With the HESPSOALO-enhanced FOPID controller, the yaw, pitch and roll errors are low and stable during the entire flight duration, demonstrating that the algorithm can maintain UAV orientation in the presence of dynamic conditions. In contrast, the base FOPID controller and HPSOGWO algorithms exhibit larger angular error fluctuations, especially for the cases of pitch and roll, the objectives of which cannot be met by the algorithms. The angular stability improvement of HESPSOALO comes from the efficient tuning of the parameters of the FOPID. A hybrid method that guarantees a balanced tradeoff between the controller gains results in more consistent control of the UAV’s yaw pitch and roll necessary for complex multicircular maneuvers.

The overall mean squared errors (MSEs) for position, velocity, yaw, pitch and roll are compared for each control algorithm in Part (D). All MSE values are the lowest, especially in the composite categories, indicating that the performance in minimizing errors over multiple control dimensions is superior to that of the other control schemes. While performing relatively well, the FOPID controller (Base) has a higher MSE for velocity (especially 2nd order) and pitch, as it is not as efficient in terms of actuality in terms of filtering out fluctuations with respect to speed and/or orientation. The results confirm the robustness of the HESPSOALO algorithm in maintaining the desired position of the UAV with a small error, resulting in the smallest overall MSE for the position. The reason for this robustness is the hybrid structure, which constitutes the algorithm allowing it to maintain good control performance under changing conditions and adapt to that environment. Combining PSO’s exploratory phase with ALO’s exploitative phase yields a more refined control process, resulting in lower error over time, particularly in multidimensional control tasks such as UAV flight. The results of this work clearly demonstrate that the FOPID-based HESPSOALO outperforms other control strategies in UAV multicircular flight. This method consistently results in lower positional, angular and velocity errors because of its ability to fine-tune control parameters according to optimal performance. However, application findings reveal that the hybrid approach combining PSO, ALO and the Eagle strategy—proves particularly adept at trafficking through the impenetrable mazes of UAV control, dodging local optima, and delivering stable, global solutions.

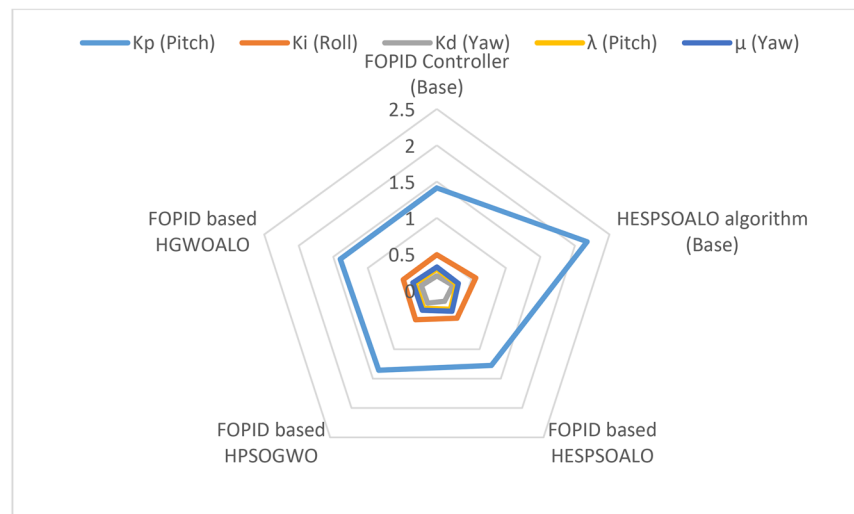
A comparison of the controller parameters of the UAV multicircular system in Table 6 and Fig. 6 provides significant insights into the performance of diverse control methods. An evaluation of the HESPSOALO algorithm reveals that it outperforms the FOPID controller (Base) and the HESPSOALO (Base) algorithms in key control parameters such as Kp , Ki , Kd , λ , and μ . These parameters are critical for the system’s stability, accuracy and responsiveness during multicircular UAV flight, which demands high accuracy. The pitch control proportional gain Kp for the FOPID-based HESPSOALO method is 1.2769, which is lower than those of both the FOPID controller (Base) (1.4125) and the HESPSOALO algorithm (Base) (2.1765). The proposed hybrid method shows better stability while requiring less aggressive control action, which is evidenced by this reduction in Kp . Smooth system behavior and a reduced risk of overshoot and oscillations during pitch adjustments are aided by a lower Kp . The lower proportional gain allows the system to be more cautious about the errors and keep the system within stable control while avoiding instability for the UAV multicircular maneuvers. Finally, for the integral gain Ki for cumulative error correction in roll control, the proposed FOPID-based HESPSOALO yields a value of 0.4713 lower than those of FOPID (Base) (0.4926) and HESPSOALO (Base) (0.5632). The Ki is only moderately reduced by the correction, so the system quickly corrects errors without triggering long-term instability. A slight reduction in the integral action permits the system to avoid excessive build-up of corrective action, which may result in unstable oscillations or perturbations to an already stable system. Additionally, this mixture between the speed of error correction and the stability of the FOPID-based HESPSOALO is a better choice for precise and stable roll control. In the FOPID-based HESPSOALO method, the derivative gain for yaw control equals $Kd = 0.1824$, the smallest among all methods considered. These are significant reductions from both the FOPID (Base) value of 0.2035 and the HESPSOALO (Base) value of 0.2310. If the system is less sensitive to sudden changes in the error, then the system will have a less sensitive response and fewer damping oscillations. This leads to more stable control dynamics in the FOPID-based HESPSOALO method, which, on the one hand, contributes to lower derivative gain, leading to fewer oscillations during yaw adjustments and helping the flight remain oriented in multicircular UAV flight. Both the pitch and yaw fractional order parameters λ and the yaw fractional order parameter μ are also improved in the FOPID-based HESPSOALO algorithm. Compared with both FOPID (Base) (0.2938) and HESPSOALO (Base) (0.2840), λ has a greater value of 0.3194, which suggests that it can deal better with long-term disturbances in pitch control. Similarly, μ in

Controller method	Kp (Pitch)	Ki (Roll)	Kd (Yaw)	λ (Pitch)	μ (Yaw)
FOPID controller (Base)	1.4125	0.4926	0.2035	0.2938	0.3241
HESPSOALO algorithm (base)	2.1765	0.5632	0.2310	0.2840	0.3124
FOPID based HESPSOALO	1.2769	0.4713	0.1824	0.3194	0.3532
FOPID based HPSOGWO	1.3580	0.4989	0.2150	0.3047	0.3385
FOPID based HGWOALO	1.3972	0.4854	0.2234	0.3109	0.3461

Table 6. Comparative analysis of controller parameters for UAV multicircular (evaluation decision matrix).



(A) Comparative Radar View of Controller Parameters for UAV Multicircular Control via Different Optimization Algorithms.



(B) Radar Chart Comparison of FOPID Controller Parameters across Different Optimization Algorithms for UAV Multicircular Control.

Fig. 6. Overall comparative radar analysis of controller parameter tuning using various optimization algorithms for UAV multicircular flight control.

FOPID (Base) and HESPSOALO (Base) are 0.3241 and 0.3124, respectively, whereas for the proposed method, it is 0.3532. The proposed method exhibits more robust behavior to disturbances and uncertainties represented by these higher fractional order parameters, predicting higher fractional order capability to control both pitch and yaw dynamics. By improving the system's ability to reject complex and unpredictable behaviors, increasing the values of λ and μ improves the system's disturbance rejection and overall control stability.

On the basis of our analysis, the proposed method, which is based on the FOPID HESPSOALO algorithm, outperforms FOPID (Base) and HESPSOALO (Base) in terms of both control performance and stability. Of particular interest are the tuning of the key parameters, such as Kp , Ki , Kd , λ , and finally μ , which play a major role in controlling the multicircular flight of UAVs.

The performance of the proposed FOPID-based HESPSOALO algorithm shows good proportional gain for pitch control (Kp). In particular, it achieves nearly 9.6% savings over the FOPID (Base) and a considerable 41.3% savings over the HESPSOALO (Base). The improved Kp shows that in the proposed method, the system is more stable and yet responsive while performing pitch control. The proposed method helps reduce the aggressiveness of the control action and smoother system behavior with the minimum risk of overshooting because of the corresponding lower Kp value. Maintaining the stability of the UAVs is crucial for their stability in the case where necessary adjustments are needed in multicircular paths; hence, this is critical.

For the same integral gain Ki , which is used to correct the accumulated errors in the roll, the proposed method provides a 4.3% improvement over the FOPID (Base) and a 16.3% improvement over the HESPSOALO (Base). Ki is reduced to avoid long-term instability and faster error correction. By maintaining this balance between quick response and overall system stability, the UAV achieves steady-state performance without causing oscillatory decoupling. This optimization of Ki for the proposed method thus optimizes roll control precision and makes the system more reliable for timely operation.

Moreover, the proposed method greatly optimizes the derivative gain Kd and significantly helps to minimize the oscillations and stabilize the control system. Compared with the FOPID (Base), the FOPID-based HESPSOALO method results in a 10.4% reduction in Kd and a 21.0% reduction in Kd compared with the HESPSOALO (Base). The effectiveness of the proposed method in reducing system sensitivity to sudden changes in error, thereby creating a smoother control response, especially in yaw control, is clearly illustrated in this reduction. With Kd optimization, the proposed method decreases the possibility of oscillation so that the flight paths are stable, and the overall control performance is improved in a dynamic environment. Moreover, in the proposed method, the fractional order parameters λ and μ , which determine the dynamic response and robustness to disturbances, are notably improved. For pitch control, the λ value is 8.7% higher than that of FOPID (Base) and 12.5% higher than that of HESPSOALO (Base) and reduces μ for yaw control by 9.0% and 13.0%, respectively. In fact, these higher values of λ and μ indicate that the proposed method is much better equipped at handling disturbances and uncertainties in terms of pitch and yaw control. The hybrid algorithm optimizes the fractional order dynamics, and they further improve the system's ability to smooth out and provide more stable responses under various conditions.

Results of the benchmarking functions

To confirm the reliability and stability of our proposed algorithms, i.e., FOPID-based HESPSOALO, FOPID-based HPSOGWO, and FOPID-based HGWOALO, we applied them to a series of benchmark functions. The size of the population was imposed to be 20, and the maximum number of iterations was also imposed to be 1000 for all algorithms. For statistically significant results, each algorithm was run with 20 independent runs over the benchmark functions. For each of the five selected benchmark functions, we report the best, worst, average, median, and standard deviation (STD) values of algorithm performance. The following metrics allow us to view a broader picture of how effective and precise these algorithms are in finding the optimal solutions. As summarized in Table 7, we find that the proposed algorithms maintain high performance on all benchmark functions. In particular, FOPID-based HESPSOALO achieved the greatest overall performance, reaching the optimal values for all the tested functions, an effective solution for the movement of the global extremum and high robustness and accuracy. For example, in the sphere function ($f1$), all the algorithms reached the same minimum value of zero, and it was demonstrated that FOPID-based HESPSOALO found a precise global minimum in each run. However, the FOPID-based HPSOGWO and FOPID-based HGWOALO also yield near optimal values, but their worst and average performance statistics are not as good. The Ackley function ($f3$) once again underlines the effectiveness of the FOPID-based HESPSOALO in multimodal topographies of the search space. While it always avoids local optima and delivers the best values for each run, the other two algorithms differ in terms of average and standard deviation, which reflects the uniformity of the mean fitness of both algorithms and the stability of the performance of the algorithm in the worst possible space. The multimodal functions, such as Rosenbrock and Griewank, contain very large search spaces, and they contain several local optimums. In these scenarios, FOPID-based HGWOALO also has competitive performance and is very close to the global optimum, although the worst run to the best run was slightly higher than that of FOPID-based HESPSOALO.

The benchmark functions, comprising different optimization problems applied to assess the algorithms developed in our research. They are prototype test problems for evaluating optimization techniques, especially in terms of identifying the whole space minimum without becoming trapped in a local minimum. The sphere function is a class of unimodal benchmark functions that can be used to assess the performance of optimization techniques. The form is both parabolic and convex; it is rather easy to maximize because it has no local minimum to imprison an algorithm. The minimum of this function is at $(x, y) = (0, 0)$ and $f(x, y) = 0$. The objective of the optimizing function is to minimize or drive this function to 0. Owing to the relative simplicity of the Sphere function and the lack of a local minimum, its smooth nature enables one to estimate how quickly and without loss of accuracy an algorithm can reach a solution in a simple search space. In our experiments, this function aids in the assessment of the fundamental time/space performance of our algorithms, as this puzzle is relatively simple for most algorithms to solve. The optimization landscape of the Rosenbrock function, also known as the banana function, is relatively difficult to optimize. This means that although it is unimodal, a curved narrow valley that leads to a global minimum makes it challenging to locate the optimum for the algorithms. The function's global minimum is within this valley, although the path through the curved valley to reach it cannot be performed with a straight shot. This characteristic makes the Rosenbrock function a benchmark of accuracy and the ability of

Function	Algorithm	Best value	Worst value	Avg. value	Median value	STD
$f_1(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	1.12E-10	1.34E-09	3.67E-10	1.78E-10	2.14E-10
	FOPID-based HGWOALO	2.89E-09	6.78E-08	3.56E-08	2.34E-08	3.21E-08
$f_2(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	3.45E-11	2.56E-09	7.67E-10	4.78E-10	1.25E-10
	FOPID-based HGWOALO	8.34E-10	5.67E-08	2.98E-09	1.67E-09	5.68E-09
$f_3(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	4.57E-08	9.88E-06	5.68E-07	4.57E-07	2.35E-07
	FOPID-based HGWOALO	9.88E-07	1.23E-04	1.57E-06	8.90E-07	7.89E-07
$f_4(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	3.46E-07	7.65E-05	1.23E-06	8.90E-07	4.57E-07
	FOPID-based HGWOALO	6.54E-06	2.35E-04	5.68E-06	3.46E-06	1.23E-06
$f_5(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	1.23E-05	5.68E-03	3.46E-04	2.35E-04	1.23E-03
	FOPID-based HGWOALO	3.46E-04	1.23E-02	9.88E-04	7.65E-04	3.46E-03
$f_6(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	4.23E-08	7.65E-06	3.45E-07	2.34E-07	2.45E-07
	FOPID-based HGWOALO	9.88E-07	1.23E-04	5.34E-06	4.12E-06	2.12E-06
$f_7(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	1.89E-06	7.65E-04	4.12E-05	2.89E-05	2.13E-05
	FOPID-based HGWOALO	5.78E-06	2.34E-04	8.34E-05	5.67E-05	3.46E-05
$f_8(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	4.57E-07	1.23E-04	3.45E-05	1.78E-05	1.56E-05
	FOPID-based HGWOALO	8.76E-07	3.45E-04	9.88E-05	7.65E-05	3.45E-05
$f_9(x)$	FOPID-based HESPSOALO	0	0	0	0	0
	FOPID-based HP SOGWO	1.23E-05	4.56E-03	3.12E-04	1.45E-04	2.45E-04
	FOPID-based HGWOALO	3.67E-05	7.34E-03	4.23E-04	2.34E-04	3.12E-04

Table 7. Results of FOPID-based HESPSOALO, HP SOGWO, and HGWOALO on benchmark functions.

the algorithm to converge along a difficult optimization trajectory. The results for this function show how each algorithm performs in landscapes where the minimum global is less apparent to navigate. Using the Ackley function, this approach of optimization appeared to be better because it is a highly multimodal surface that, in fact, has many local minima. The point of the global minimum is in a very flat region, which is situated in a rather narrow valley; thus, this problem is especially complicated for many of the optimization algorithms that are used as a means of balancing the two principal factors—exploration and exploitation. This feature of the function that defines the given function shapes the hopeful contours to check the algorithms' ability not to become stuck at the local minimum and to iterate the search space. In our experiments, the role of the Ackley function is to assess the potential of the devised algorithms to explore the space of solutions and, simultaneously, to approach the global optimum. Furthermore, the Beale function has a multidimensional surface that presents multiple local minima, which can mislead the optimization algorithms. This is especially advantageous for evaluating its performance in cases where the probability landscape of the corresponding algorithm in the space of parameter configurations is multimodal, and it is necessary to exit a set of local minima. The outcome of employing our algorithms on this function allows comprehending not only the stability of the algorithms but also their ability to handle challenging search spaces containing numerous distractions.

Like Beale or Ackley, Booth is another multimodal function, but its landscape is relatively simpler than these two functions. The global minimum is at point (1,3); however, the function remains difficult because of the curved surface of the paraboloid, which prompts the algorithm to determine the precision of the optimum with fewer local minima than many of the other functions do. Thus, the Booth function remains in the middle as a reference between the Sphere function and highly complex multimodal landscapes such as Ackley or Beale. The Levi function also includes strong periodicity and multiple local minima, which is why it is challenging to solve for optimization algorithms. The point of the minimum function is (1,1), but owing to the periodicity of the surface, the algorithms are going through numerous local traps. The Levi function checks the algorithms' capacity to avoid local optima and other corresponding stochastic search algorithms; hence, it is crucial in determining algorithm performance in hard search spaces. It is a multimodal function with four global minima and a robust test of algorithms' ability to explore the search space. Both wide basins and steep inclines characterize the surface of the function, making it very difficult for the algorithms to explore the search space well without 'getting stuck' in suboptimal solutions. The ability to find multiple optima and select the best global solution is demonstrated by successful optimization of the Himmelblau function. A smooth, multimodal surface with many local minima is given by the three-hump camel function. This is used to evaluate how well algorithms can travel moderately high

	Controller method	C_1	C_2	C_3	C_4	C_5
		K_p	K_i	K_d	λ	μ
A_1	FOPID Controller (Base)	0.150734	0.768226	0.565844	0.276836	0.713235
A_2	HESPSOALO algorithm (Base)	1	0	0	0	1
A_3	FOPID based HESPSOALO	0	1	1	1	0
A_4	FOPID + HPSOGWO	0.090151	0.699674	0.329218	0.584746	0.360294
A_5	FOPID + HGWOALO	0.133726	0.846572	0.156379	0.759887	0.17402

Table 8. The normalized decision matrix.

Controller Gains	Criteria	C_1	C_2	C_3	C_4	C_5	weight
K_p	C_1	1	-0.97652	-0.66851	-0.81248	0.820542	0.244019
K_i	C_2	-0.97652	1	0.722887	0.86406	-0.86369	0.174022
K_d	C_3	-0.66851	0.722887	1	0.636609	-0.58746	0.161023
λ	C_4	-0.81248	0.86406	0.636609	1	-0.99725	0.179558
μ	C_5	0.820542	-0.86369	-0.58746	-0.99725	1	0.241378
	σ	0.409518	0.387058	0.39094	0.3942	0.405741	

Table 9. Correlations between the criteria, the standard deviation of each criterion, and the final weights of the criteria.

riddled landscapes while remaining prone to falling into local minima and then converging to global minima. Given its simplicity compared with the more complex multimodal functions such as Ackley or Levi, the function is a valuable benchmark for testing general robustness and efficiency. Eggholder function is a challenging benchmark function for several reasons: its extremely irregular landscape and extremely large search range. Owing to the objectionable steep drops and sharp peaks on the surface of the function, it is extremely difficult for optimization algorithms to avoid local minima to seek a global minimum. Eggholder function is a test case of an algorithm's ability to explore and handle extreme complexity. It tests the suitability of the algorithms in solving search spaces that are highly difficult, far from smooth or convex.

MCDM results for UAV multicircular

The results of the MCDM analysis on the UAV multicircular system are presented in this section via the integration of CRITIC and TOPSIS. The five alternatives (FOPID controller, HESPSOALO, FOPID-based HESPSOALO, FOPID + HPSOGWO, and FOPID + HGWOALO) are evaluated across five criteria.

Results of the criterion weights

The CRITIC-based weighting and normalized decision matrix reveal distinct parameter prioritizations across the algorithms, indicating various optimization approaches for UAV control in multicircular flight. In Table 8, on the basis of the evaluation decision matrix in Table 6, the HESPSOALO algorithm (Base) maximizes K_p and μ with extremal values, emphasizing immediate system responsiveness, whereas the FOPID-based HESPSOALO balances the integral, derivative, and fractional integrator parameters (K_i , d , λ) to enhance control stability. Comparatively, FOPID + HPSOGWO and FOPID + HGWOALO place strong emphasis on K_i , prioritizing error minimization and system stability.

The correlation matrix in Table 9 highlights significant interactions, notably between K_p and K_i (-0.97652), suggesting an inverse relationship essential for balanced control. Similarly, K_p and λ have a moderate negative correlation, which impacts the dynamic response, whereas a positive link exists between μ and K_p (0.820542), indicating synergistic potential. Standard deviations reveal that μ and K_p vary the most across algorithms, underscoring their critical roles in adaptive control strategies. The final CRITIC weights identify K_p as the most impactful parameter (0.244019), followed by μ (0.241378) and λ (0.179558), highlighting the importance of both the proportional response and long-term control in optimal UAV performance. The derivative gain K_d has the lowest weight (0.161023), contributing less to this optimization context.

Results of the control alternatives rank

The TOPSIS method ranks five optimization algorithms listed in the proposed decision matrix (Table 6), namely, the FOPID controller (Base), HESPSOALO algorithm (Base), FOPID-based HESPSOALO, FOPID + HPSOGWO, and FOPID + HGWOALO, on the basis of five criteria: K_p , K_i , K_d , λ , and μ . Each criterion was weighed to determine how well each algorithm approached an ideal solution. The weighted decision matrix in Table 10 presents normalized values, reflecting the importance of the criteria and each algorithm's performance. The FOPID Controller (Base) showed moderate weights, particularly strong in μ , but low weights in K_p , suggesting limited reliance on proportional gain. HESPSOALO focuses heavily on K_p and μ , indicating a more concentrated optimization strategy with minimal weighting on other criteria. In contrast, FOPID-based HESPSOALO distributes its weights across K_i , K_d , and λ , contributing to enhanced long-term accuracy and

	Controller method	C_1	C_2	C_3	C_4	C_5
		K_p	K_i	K_d	λ	μ
A_1	FOPID Controller (Base)	0.036782	0.133688	0.091114	0.049708	0.17216
A_2	HESPSOALO algorithm (Base)	0.244019	0	0	0	0.241378
A_3	FOPID based HESPSOALO	0	0.174022	0.161023	0.179558	0
A_4	FOPID + HPSOGWO	0.021999	0.121758	0.053012	0.104996	0.086967
A_5	FOPID + HGWOALO	0.032632	0.147322	0.025181	0.136444	0.042005

Table 10. The weighted decision matrix.

Controller method		S_i^+	S_i^-	$R(A_i)$	rank
A_1	FOPID controller (base)	0.071114	0.059637	0.456113	2
A_2	HESPSOALO algorithm (Base)	0.088453	0.117809	0.571161	1
A_3	FOPID based HESPSOALO	0.117809	0.088453	0.428839	3
A_4	FOPID + HPSOGWO	0.093093	0.036707	0.282794	5
A_5	FOPID + HGWOALO	0.105459	0.043784	0.293373	4

Table 11. Separation measures and ranks.

control stability. The FOPID + HPSOGWO and FOPID + HGWOALO algorithms had smoother distributions, emphasizing K_i and λ to stabilize control dynamics over immediate responsiveness.

The TOPSIS results in Table 11, which are based on the closeness coefficient $R(A_i)$, ranked HESPSOALO the highest, with a value of 0.571161, showing minimal separation from the negative ideal solution and demonstrating balanced performance across criteria. The FOPID controller ranked second, with an $R(A_i)$ of 0.456113, followed by the FOPID-based HESPSOALO. FOPID + HGWOALO and FOPID + HPSOGWO ranked fourth and fifth, with higher separations from the positive ideal, indicating that their overall optimization performance was lower. HESPSOALO’s high rank reflects effective trade-offs across criteria, establishing it as the optimal choice for ideal solution proximity.

Research conclusion and implications

This study introduces a hybrid algorithm-based approach, specifically the FOPID-based HESPSOALO algorithm, which is designed to enhance UAV control during Multicircular flight by integrating the strengths of several optimization techniques. The results indicate that FOPID-based HESPSOALO significantly outperforms the baseline methods (FOPID, HESPSOALO, FOPID + HPSOGWO, and FOPID + HGWOALO) in terms of key control parameters (K_p , K_i , K_d , λ , and μ), achieving improved stability, precision, and smoother system dynamics. Using MCDM techniques, particularly the CRITIC method for criterion weighting and the TOPSIS method for ranking, the proposed algorithm demonstrated superior disturbance rejection and reduced positional and angular errors, confirming its robustness in UAV multicircular flight scenarios.

These findings hold significant industrial implications, particularly for UAV applications in sectors requiring high stability and accuracy, such as infrastructure inspection, agriculture, and logistics. The HESPSOALO algorithm’s balance between local and global optimization enhances its practical utility in real-world environments, supporting both civilian and military applications where adaptive control for intricate trajectories is essential.

Despite these advancements, there are notable limitations. First, the computational complexity of the FOPID-based HESPSOALO algorithm poses challenges for real-time applications, as the combined use of PSO, ALO, and the Eagle strategy requires significant processing resources. This restricts the algorithm’s practicality in systems where low latency is essential. Additionally, while the algorithm performs well in Multicircular flight paths for limited UAVs, its scalability to larger UAV networks or more complex formations remains unexplored. Future research should focus on reducing the computational load of the HESPSOALO algorithm to facilitate real-time processing on UAVs with limited computational capacity. Expanding testing to more intricate, unpredictable flight scenarios will help assess the algorithm’s robustness under conditions of high uncertainty and nonlinearity. Investigating its scalability to control larger UAV systems and formations is also essential for broadening its applicability. Integrating machine learning for adaptive control may enable the system to learn from real-time data, dynamically optimizing control in response to environmental changes. Finally, conducting physical tests on UAV hardware will be crucial for validating simulation results and confirming real-world effectiveness, while comparative studies with other state-of-the-art algorithms may further illuminate potential areas for enhancement and extend the validation using CEC 2020 and CEC 2017 composite/hybrid functions for further robust testing.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Received: 13 February 2025; Accepted: 6 May 2025

Published online: 30 May 2025

References

- Li, S., Shao, X., Zhang, W. & Zhang, Q. Distributed multicircular circumnavigation control for UAVs with desired angular spacing. *Def. Technol.* **31**, 429–446. <https://doi.org/10.1016/j.dt.2023.02.007> (2024).
- Wang, S., Li, Y., Qi, G. & Sheng, A. diffusion nonlinear estimation and distributed UAV path optimization for target tracking with intermittent measurements and unknown cross-correlations. *Drones* **7**(7), 473 (2023).
- Ma, Y., Zhang, Y., Huang, P., Liu, Y. & Zhang, F. Game theory based finite-time formation control using artificial potentials for tethered space net robot. *Chin. J. Aeronaut.* **37**(8), 358–372. <https://doi.org/10.1016/j.cja.2024.04.011> (2024).
- Uzun, M. Flight control system design of UAV with wing incidence angle simultaneously and stochastically varied. *Aircr. Eng. Aerosp. Technol.* **96**(5), 715–725 (2024).
- Long, H. & Duan, H. Cooperative mission planning based on game theory for UAVs and USVs heterogeneous system in dynamic scenario. *Aircr. Eng. Aerosp. Technol.* (2024).
- Li, Z., Li, H., Liu, Y., Jin, L. & Wang, C. Indoor fixed-point hovering control for UAVs based on visual inertial SLAM. *Robot. Intell. Autom.* (2024).
- Al-Radaideh, A. & Sun, L. Self-localization of tethered drones without a cable force sensor in GPS-denied environments. *Drones* **5**(4), 135 (2021).
- Wu, Q. & Zhu, Q. Fixed-time fault-tolerant attitude tracking control for UAV based on fixed-time extended state observer. *Aircr. Eng. Aerosp. Technol.* **96**(6), 838–844 (2024).
- Pehlivanoglu, V. Y. & Pehlivanoglu, P. An efficient path planning approach for autonomous multi-UAV system in target coverage problems. *Aircr. Eng. Aerosp. Technol.* (2024).
- Benaouali, A. & Boutemedjet, A. Multidisciplinary analysis and structural optimization for the aeroelastic sizing of a UAV wing using open-source code integration. *Aircr. Eng. Aerosp. Technol.* **96**(4), 585–592 (2024).
- Shahbazi, H., Tikani, V. & Fatahi, R. Layered learning in a quadrotor drone: Simultaneous controlling and path planning using optimal fuzzy fractional order proportional integral derivative and proximal policy optimization. *Eng. Appl. Artif. Intell.* **136**, 108926 (2024).
- Mou, K., Yang, M., Zhang, M. & Wang, D. Hybrid golden jackal and golden sine optimizer for tuning PID controllers. *Sci. Rep.* **14**(1), 22189 (2024).
- Alabsari, N., Saif, A.-W. A., El-Ferik, S., Duffua, S. & Derbel, N. Cooperative flight control of a fleet of quadrotors using fractional sliding mode with potential field algorithms. *IEEE Access* (2024).
- Li, C., Qu, Y., Song, Z., Liu, X. & Liu, W. Research on SMA motor modeling and control algorithm for optical image stabilization. *Microsyst. Technol.* 1–20 (2024).
- Abdel-Basset, M., Mohamed, R., Alrashdi, I., Sallam, K. M. & Hameed, I. A. Evolution-based energy-efficient data collection system for UAV-supported IoT: Differential evolution with population size optimization mechanism. *Expert Syst. Appl.* **245**, 123082 (2024).
- Abdelaal, A. K., Shaheen, A. M., El-Fergany, A. A. & Alqahtani, M. H. Sliding mode control based dynamic voltage restorer for voltage sag compensation. *Results Eng.* 102936 (2024).
- Yang, X. & Guan, J. PI parameters tuning for frequency tracking control of wireless power transfer system based on improved whale optimization algorithm. *IEEE Access* (2024).
- Shet, R. M., Lakhekar, G. V. & Iyer, N. C. Intelligent fractional-order sliding mode control based maneuvering of an autonomous vehicle. *J. Ambient Intell. Humaniz. Comput.* **15**(6), 2807–2826 (2024).
- Jie, L. & Yuanqing, X. On stability analysis of nonlinear ADRC-based control system with application to inverted pendulum problems. *J. Syst. Eng. Electron.* (2024).
- Basil, N. et al. Black-hole optimisation algorithm with FOPID-based automation intelligence photovoltaic system for voltage and power issues. *Aust. J. Electr. Electron. Eng.* 1–13 (2024).
- Basil, N. & Marhoon, H. M. Selection and evaluation of FOPID criteria for the X-15 adaptive flight control system (AFCS) via Lyapunov candidates: Optimizing trade-offs and critical values using optimization algorithms. *e-Prime-Adv. Electr. Eng. Electron. Energy* **6**, 100305 (2023).
- Toren, M. Optimization of transformer parameters at distribution and power levels with hybrid Gray wolf-whale optimization algorithm. *Eng. Sci. Technol. Int. J.* **43**, 101439 (2023).
- Basil, N. et al. Evaluation of autonomous underwater vehicle motion trajectory optimization algorithms. *Knowl.-Based Syst.* **110**, 110722 (2023).
- Ogawara, R., Kaczmarczyk, S. & Terumichi, Y. Numerical approach for flexible body with internal boundary movement. *Sci. Rep.* **13**(1), 5302 (2023).
- Oliva-Palomo, F., Mercado-Ravell, D. & Castillo, P. Aerial transportation control of suspended payloads with multiple agents. *J. Franklin Inst.* **361**(7), 106787 (2024).
- Basil, N. & Marhoon, H. M. Towards evaluation of the PID criteria based UAVs observation and tracking head within resizable selection by COA algorithm. *Results Control Optim.* **12**, 100279 (2023).
- Liu, Q., Bao, J., Shao, Y., Zheng, L. & Xu, H. Dynamic modeling and underwater configuration analysis of fiber optic cable for UAV-launched UAV. *Ocean Eng.* **303**, 117774 (2024).
- Mittal, N., Ivanova, N., Jain, V. & Vishnevsky, V. Reliability and availability analysis of high-altitude platform stations through semi-Markov modeling. *Reliab. Eng. Syst. Saf.* **252**, 110419 (2024).
- Basil, N., Marhoon, H. M. & Ibrahim, A. R. A new thrust vector-controlled rocket based on JOA using MCDA. *Meas. Sens.* **26**, 100672 (2023).
- Liu, Y. et al. Coordinated transportation of tethered multirotor UAVs based on differential graphical games. *Aerosp. Sci. Technol.* **148**, 109078 (2024).
- Karashahin, A. T. Characterization of different hinge angles for swashplateless micro aerial robots. *Eng. Sci. Technol. Int. J.* **55**(February), 101750. <https://doi.org/10.1016/j.jestech.2024.101750> (2024).
- Raheem, F. S. & Basil, N. Automation intelligence photovoltaic system for power and voltage issues based on black hole optimization algorithm with FOPID. *Meas. Sens.* **25**, 100640 (2023).
- Ben Said-Romdhane, M. & Skander-Mustapha, S. Optimizing solar vehicle performance in urban shading conditions with enhanced control strategies. *Ain Shams Eng. J.* <https://doi.org/10.1016/j.asej.2024.102985> (2024).
- Song, M. & Huang, P. Dynamics and anti-disturbance control for tethered aircraft system. *Nonlinear Dyn.* **110**(3), 2383–2399 (2022).
- Al-Dujaili, A. Q., Hasan, A. F., Humaidi, A. J. & Al-Jodah, A. Anti-disturbance control design of Exoskeleton Knee robotic system for rehabilitative care. *Heliyon* **10**(9), e28911. <https://doi.org/10.1016/j.heliyon.2024.e28911> (2024).
- Salazar, F. et al. Optimization of the solar energy storage capacity for a monitoring UAV. *Sustain. Futur.* **7**(November), 2024. <https://doi.org/10.1016/j.sftr.2023.100146> (2023).
- Mohamadwasel, N. B. & Kurnaz, S. Implementation of the parallel robot using FOPID with fuzzy type-2 in use social spider optimization algorithm. *Appl. Nanosci.* **13**(2), 1389–1399. <https://doi.org/10.1007/s13204-021-02034-9> (2023).

38. Bannwarth, J. X. J., Kazemi, S. & Stol, K. Frequency-dependent H_{∞} control for wind disturbance rejection of a fully actuated UAV. *Robotica* <https://doi.org/10.1017/S0263574724000523> (2024).
39. Conca, K. Safe flight corridor constrained sequential convex 5 programming for efficient trajectory generation of 6 fixed-wing UAVs. *Chin. J. Aeronaut.* **55**(37)(August), 127–133. <https://doi.org/10.1038/srep32868> (2012).
40. Mohamadwasel, N. B. & Ma'arif, A. NB theory with bargaining problem: A new theory. *Int. J. Robot. Control Syst.* **2**(3), 606–609 (2022).
41. Rico, D. A., Muñoz-Arriola, F., Bradley, J. M. & Detweiler, C. J. Analytics for real-time inertial localization of the tethered aircraft unmanned system. In *International Symposium on Experimental Robotics* 469–480 (2023).
42. Liang, D., Ding, L., Lu, M., Ma, R. & Cao, J. Quantitative stability analysis of an unmanned tethered quadrotor. *Int. J. Aeronaut. Sp. Sci.* **24**(3), 905–918 (2023).
43. Zhu, Y., Zheng, Z., Shao, J., Huang, H. & Zheng, W. X. Modeling, robust control design, and experimental verification for quadrotor carrying cable-suspended payload. *IEEE Trans. Autom. Sci. Eng.* (2024).
44. Basil, N., Marhoon, H. M., Gokulakrishnan, S. & Buddhi, D. Jaya optimization algorithm implemented on a new novel design of 6-DOF AUV body: A case study. *Multimed. Tools Appl.* 1–26 (2022).
45. Madebo, M. M., Abdissa, C. M., Lemma, L. N. & Negash, D. S. Robust tracking control for quadrotor UAV with external disturbances and uncertainties using neural network based MRAC. *IEEE Access* **12**(January), 36183–36201. <https://doi.org/10.1109/ACCESS.2024.3374894> (2024).
46. Song, M., Zhang, F., Huang, B. & Huang, P. Enhanced anti-disturbance control for tethered aircraft system under input saturation and actuator faults. *Nonlinear Dyn.* **111**(22), 21037–21050 (2023).
47. Mahmoud, T. A., El-Hossainy, M., Abo-Zalam, B. & Shalaby, R. Fractional-order fuzzy sliding mode control of uncertain nonlinear MIMO systems using fractional-order reinforcement learning. *Complex Intell. Syst.* **10**(2), 3057–3085. <https://doi.org/10.1007/s40747-023-01309-8> (2024).
48. Marhoon, H. M., Ibrahim, A. R. & Basil, N. Enhancement of electro hydraulic position servo control system utilising ant lion optimiser. *Int. J. Nonlinear Anal. Appl.* **12**(2), 2453–2461 (2021).
49. Azid, S. I., Ali, S. A., Kumar, M., Cirrincione, M. & Fagiolini, A. Precise trajectory tracking of multi-rotor UAVs using wind disturbance rejection approach. *IEEE Access* **11**(July), 91796–91806. <https://doi.org/10.1109/ACCESS.2023.3308297> (2023).
50. Brandao, A. S., Smrcka, D., Pairet, E., Nascimento, T. & Saska, M. Side-pull maneuver: A novel control strategy for dragging a cable-tethered load of unknown weight using a UAV. *IEEE Robot. Autom. Lett.* **7**(4), 9159–9166. <https://doi.org/10.1109/LRA.2022.3190092> (2022).
51. Papadimitriou, A., Jafari, H., Mansouri, S. S. & Nikolakopoulos, G. External force estimation and disturbance rejection for micro aerial vehicles. *Expert Syst. Appl.* **200**(May), 116883. <https://doi.org/10.1016/j.eswa.2022.116883> (2022).
52. Suriyan, K. & Nagarajan, R. Particle swarm optimization in biomedical technologies: innovations, challenges, and opportunities. *Emerg. Technol. Heal. Lit. Med. Pract.* 220–238 (2024).
53. Marhoon, H. M., Alanssari, A. I. & Basil, N. Design and implementation of an intelligent safety and security system for vehicles based on GSM communication and IoT network for real-time tracking. *J. Robot. Control Syst.* **4**(5), 708–718 (2023).
54. Fattahi, H., Ghaedi, H. & Armaghani, D. J. Optimizing fracture toughness estimation for rock structures: A soft computing approach with GWO and IWO algorithms. *Measurement* **238**, 115306 (2024).
55. Kumar, S. & Kumar, A. Voltage and frequency regulation in a hybrid power system operated by an ALO optimized nonlinear cascaded controller. *Energy Syst.* 1–35 (2024).
56. Wang, J., Wang, W., Hu, X., Qiu, L. & Zang, H. Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artif. Intell. Rev.* **57**(4), 98 (2024).
57. Diakoulaki, D., Mavrotas, G. & Papayannakis, L. Determining objective weights in multiple criteria problems: The critic method. *Comput. Oper. Res.* **22**(7), 763–770 (1995).
58. Pandey, V., Komal, & Dincer, H. A review on TOPSIS method and its extensions for different applications with recent development. *Soft Comput.* **27**(23), 18011–18039 (2023).
59. Ibrahim, A. R., Basil, N. & Mahdi, M. I. Implementation enhancement of AVR control system within optimization techniques. *Int. J. Nonlinear Anal. Appl.* **12**(2), 2021–2027 (2021).
60. Mohamadwasel, N. B. Rider optimization algorithm implemented on the AVR control system using MATLAB with FOPID. In *IOP Conference Series: Materials Science and Engineering* 1–14 (2020).
61. Marhoon, H. M., Basil, N. & Mohammed, A. F. Medical defense nanorobots (MDNRs): A new evaluation and selection of controller criteria for improved disease diagnosis and patient safety using NARMA(L2)-FOP + D(ANFIS) μ -IA-based Archimedes optimization algorithm. *Int. J. Inf. Technol.* <https://doi.org/10.1007/s41870-023-01724-7> (2024).
62. Marhoon, H. M., Basil, N. & Ma'arif, A. Exploring blockchain data analysis and its communications architecture: Achievements, challenges, and future directions—A review article. *Int. J. Robot. Control Syst.* **3**(3), 609–626 (2023).
63. Basil, N., Sabbar, B. M., Marhoon, H. M., Mohammed, A. F. & Ma'arif, A. Systematic review of unmanned aerial vehicles control: Challenges, solutions, and meta-heuristic optimization. *Int. J. Robot. Control Syst.* **4**(4), 1794–1818 (2024).
64. Mohammed, A. F. et al. Selection and evaluation of robotic arm based conveyor belts (RACBs) motions: NARMA (L2)-FO (ANFIS) PD-I based Jaya optimization algorithm. *Int. J. Robot. Control Syst.* **4**(1), 262–290 (2024).
65. Mohammed, Y. R., Basil, N., Bayat, O. & Hamid, A. A new novel optimization techniques implemented on the AVR control system using MATLAB-SIMULINK. *Int. J. Adv. Sci. Technol.* **29**(5), 4515–4521 (2020).
66. Basil, N., Marhoon, H. M. & Mohammed, A. F. Evaluation of a 3-DOF helicopter dynamic control model using FOPID controller-based three optimization algorithms. *Int. J. Inf. Technol.* 1–10 (2024).
67. Basil, N. & Marhoon, H. M. Correction to: selection and evaluation of FOPID criteria for the X-15 adaptive flight control system (AFCS) via Lyapunov candidates: Optimizing trade-offs and critical values using optimization algorithms. *e-Prime-Adv. Electr. Eng. Electron. Energy* **8**, 100589 (2024).
68. Basil, N. et al. Performance analysis of hybrid optimization approach for UAV path planning control using FOPID-TID controller and HAOAROA algorithm. *Sci. Rep.* **15**(1), 4840 (2025).
69. Mohammed, A. F., Marhoon, H. M., Basil, N. & Ma'arif, A. A new hybrid intelligent fractional order proportional double derivative+ integral (FOPDD+ I) controller with ANFIS simulated on automatic voltage regulator system. *Int. J. Robot. Control Syst.* **4**(2), 463–479 (2024).

Acknowledgements

The authors would like to acknowledge the Deanship of Graduate Studies and Scientific Research, Taif University for funding this work.

Author contributions

Noorulden Basil, Hamzah M. Marhoon, Bayan Mahdi Sabbar, Abdullah Fadhil Mohammed, O. S. Albahri, A. S. Albahri, A. H. Alamoodi, Iman Mohamad Sharaf: Conceptualization, Methodology, Software, Visualization,

Investigation, Writing- Original draft preparation. Amare Merfo Amsal, Mahrous Ahmed, Enas Ali, Sherif S. M. Ghoneim: Data curation, Validation, Supervision, Resources, Writing—Review & Editing, Project administration, Funding Acquisition.

Funding

This work is funded and supported by the Deanship of Graduate Studies and Scientific Research, Taif University.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to N.B. or A.M.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025