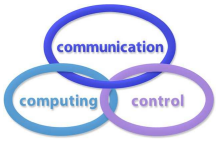
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 1, Month: February, Year: 2026
Article Number: 7197, https://doi.org/10.15837/ijccc.2026.1.7197



CCC Publications

Intelligent Autopilot Design for Fixed-Wing UAVs: Robust Stabilization and Trajectory Tracking Using Advanced Evolutionary and Swarm Optimization Algorithms

M. Fayti, M. Mjahed, H. Ayad, A. El Kari

Majid Fayti*
Electrical Systems, Energy Efficiency, and Telecommunications Laboratory
Department of Applied Physics
Faculty of Science and Technology, Marrakech, Morocco
*Corresponding author: majid.fayti@ced.uca.ma

Mostafa Mjahed
Maths and Systems Department,Royal School of Aeronautics, Marrakech, Morocco

Hassan Ayad
Electrical Systems, Energy Efficiency, and Telecommunications Laboratory
Department of Applied Physics
Faculty of Science and Technology, Marrakech, Morocco

Abdeljalil El Kari
Electrical Systems, Energy Efficiency, and Telecommunications laboratory
Department of Applied Physics
Faculty of Science and Technology, Marrakech, Morocco

Abstract

This paper presents the development and analysis of an intelligent autopilot system for a fixed-wing unmanned aerial vehicle (FWUAV), designed to stabilize and track critical flight parameters: airspeed, altitude, heading angle, and sideslip angle. The proposed system leverages a modular control architecture based on successive closed-loop control channels, enhanced by advanced optimization methods: Ant Lion Optimizer (ALO), Differential Evolution (DE), Bat Algorithm (BA), and Harmony Search (HS). Simulation results validate the robustness and efficiency of the autopilot system under various scenarios, including fixed set-point hovering and trajectory tracking for both straight-line and orbital paths, with and without wind perturbations. Comparative analyses with state-of-the-art methods from the literature demonstrate that the DE and ALO-based controllers consistently achieve superior performance in terms of precision and rapidity while maintaining adaptability to nonlinear dynamics and external disturbances. Furthermore, the findings highlight the potential of the proposed autopilot architecture and optimization methods in enhancing UAV control accuracy and robustness.

Keywords: Fixed-wing UAV, Intelligent Autopilot, Optimization, Differential Evolution, Ant Lion Optimizer, Bat Algorithm, Harmony Search, Trajectory Tracking.

1 Introduction

Since their development, control algorithms for stabilizing Fixed-Wing Unmanned Aerial Vehicles (FWUAVs) have gained significant attention within the scientific community. This interest stems from the inherent complexity of FWUAV dynamics, which pose a major challenge for researchers. Additionally, these vehicles are frequently deployed in critical missions such as wide-area surveillance, mapping, or search and rescue operations, requiring both precision and robustness. The diversity of these applications has driven the need for high-performance autopilot systems capable of maintaining accurate maneuverability under varying and often unpredictable conditions.
FWUAVs are particularly one of the most critical and complex systems due to the strong correlation among their main flight parameters. Furthermore, their classification as under-actuated systems, with fewer actuators than degrees of freedom, adds to the complexity of their control. Consequently, the development of autopilot architectures that ensure both stability and trajectory tracking accuracy remains a pressing research objective.

Over the past decades, a growing body of literature has explored various linear and nonlinear control strategies for FWUAVs. This exploration has led to the development of diverse linear and nonlinear methodological approaches. Classical linear methods such as Proportional-Integral-Derivative (PID) controllers, Linear Quadratic Regulators (LQR), and Root Locus (RL) techniques have laid a solid foundation due to their simplicity and effectiveness in well-modeled systems [1, 2, 3, 4]. These control methods largely rely on linearized models that can perform near the trimming conditions, where steady-state flight is assumed. For instance, the PID controller has been extensively utilized to stabilize FWUAVs, as demonstrated in [1], where cascade PID controllers were developed for decoupled lateral and longitudinal control, while the LQR technique had promising outcomes when it was applied to enhance the longitudinal stability in turbulent conditions [2, 3]. RL techniques have also been employed to design longitudinal control laws based on classical control theory [4]. Nevertheless, while traditional methods are effective in certain contexts, they exhibit limitations when managing high-performance maneuvers, particularly under the influence of non-linearities and coupling motion properties inherent in FWUAV dynamics. To address these limitations, nonlinear control methods have emerged as more robust alternatives. Significant advancements have been observed in techniques such as Integral Sliding Mode Control (ISMC), the backstepping method, and Nonlinear Model Predictive Control (NMPC). These approaches have demonstrated improved performance in handling complex flight scenarios. Studies like [5, 6] have shown that these methods can effectively reduce cross-track errors and manage large deviations in altitude and flight path angles by leveraging nonlinear sliding surfaces.

Despite these advancements, challenges remain, particularly in terms of computational complexity and sensitivity to model uncertainties. In response, the integration of Artificial Intelligence (AI) techniques has introduced new perspectives by combining the strengths of traditional methods with adaptive learning capabilities provided by AI technology. Within this field, Soft Computing (SC), Evolutionary Algorithms (EA), and Swarm Intelligence (SI) offer powerful tools for efficiently tuning autopilot systems across various flight modes using available input-output data [7].

Soft computing techniques, including Adaptive Neural Networks (ANN) and Fuzzy Logic (FL), are among the most well-known developments in this domain. These methods have proven their ability to manage complex non-linear dynamics, as in [8, 9]. Furthermore, they can handle those complex dynamics presented for FWUAVS. For example, in [10, 11], self-tuning fuzzy PID controllers have shown superiority in speed stabilization and error tracking when designing the autopilot systems for Aerosonde fixed-wing UAVs. On the other hand, the ANN method has been successful in dealing with complex behaviors and uncertainties, and it has been used effectively as a neural design for the autopilot of remotely controlled vehicles [12]. Furthermore, the Adaptive Neuro-Fuzzy Inference System (ANFIS) employs ANN and FL to dynamically enhance the UAV's feedback control performance. This method combines training a neural network with fuzzy logic reasoning, allowing for real-time adjustments to PID parameters [13, 14].
Otherwise, Evolutionary Algorithms (EA) and Swarm Intelligence (SI) methods offer a simpler and more effective alternative to optimize control settings in complicated, nonlinear systems. Unlike ANN

and FL, which require extensive parameter tuning, EA and SI methods automate the search for optimal solutions, reducing manual effort and improving adaptability, and enhancing their practicality for real-world applications [15, 16]. Techniques such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have demonstrated their effectiveness in optimizing control parameters and have been successfully applied to UAV control problems. In [17, 18], the authors have successfully applied the PSO and GA in improving the dynamics of quadcopters during trajectory tracking. These techniques are particularly suited for systems with unknown dynamics and external disturbances, as they naturally adapt to changing conditions without requiring precise modeling or extensive prior knowledge.

GA-based optimization has shown remarkable success in tuning PID controllers for FWUAVs, where both longitudinal and lateral dynamics can be effectively regulated, accounting for system nonlinearities and complexities. In [19], an intelligent flight control system was developed and optimized using GA for an Aerosonde fixed-wing UAV to address transient response issues. The GA-PID integration in the flight control system was successfully compared against classical PID tuning methodologies, demonstrating superior performance. Similarly, [20] presents a hybrid control architecture that leverages GA to optimize input weight feedback for a Robust Linear Quadratic Regulator (RLQR), enhancing aircraft performance. This comprehensive redesign of a UAV autopilot showcased improved response and robustness against uncertainties and nonlinearities compared to traditional PID controllers. Furthermore, GA has been compared with ANFIS and FL in intelligent autopilot system design for Aerosonde fixed-wing UAVs. The results highlighted GA's superior robustness and reliability with lower computational costs. Alternatively, other SI techniques, such as the Artificial Bee Colony (ABC) algorithm, have also shown promise. In [21], a cascade PID controller for UAV altitude control was designed and optimized using ABC. The Integral Absolute Error (IAE) was identified as the optimal fitness function, leading to significant performance improvements.

Research on FWUAV stabilization and trajectory tracking has highlighted several limitations. Linearizing dynamic models, while practical, often fails to capture the system's non-linear complexities, especially under disturbed conditions. Additionally, interactions between key parameters, such as airspeed, altitude, heading angle, and sideslip angle, are frequently underestimated, compromising stability. Conventional control methods like PID and LQR struggle to adapt to evolving dynamics and external disturbances, leading to suboptimal performance. In contrast, EA and SI methods offer robust solutions, addressing non-linearity and system complexity effectively. Unlike soft computing techniques, which rely on extensive parameter tuning, EA and SI adapt more naturally to dynamic environments. Their rapid advancement and innovative heuristics paved the way for more reliable and efficient UAV control systems, meeting the growing demands for stability and precision.

In this paper, we employ four modern evolutionary and swarm intelligence algorithms named Ant Lion Optimizer (ALO), Differential Evolution (DE), Bat Algorithm (BA), and Harmony Search (HS). They are carefully selected and used for the first time to deal with the control of FWUAVs because of their proven effectiveness in solving nonlinear control problems across various applications [22, 23]. These algorithms are not only utilized for the design of the control framework but also serve as a comparative and analysis platform to validate their robustness and ability to efficiently explore the solution space in complex optimization tasks [24, 25]. Our contribution introduces a novel control and stabilization framework that explicitly addresses parameter coupling, system nonlinearities, and environmental disturbances. The proposed methodology adopts a multi-objective optimization strategy to simultaneously regulate key flight parameters, including airspeed, altitude, heading angle, and sideslip angle. A reduced-parameter autopilot architecture, consisting of four successive closed-loop control channels, implements the control process. Each channel is dedicated to one flight variable and is optimized using ALO, DE, BA, and HS. Furthermore, a comparative performance evaluation is conducted to benchmark these algorithms against one another and against existing methods in the literature. The results confirm the robustness and efficiency of the proposed approach under diverse and challenging flight conditions. The rest of this paper is structured as follows: The FWUAV's dynamic description and mathematical modeling are shown in Section 2. An overview of the swarm intelligence and evolutionary algorithms used in this study is given in Section 3. The suggested autopilot system

architecture is described in detail in Section 4. The best way to tune the flight control system with the chosen algorithms is explained in Section 5. The simulation results and performance evaluation are covered in Section 6. And the conclusion part is in Section 7.

2 Fixed wing UAV dynamic of flight

The dynamic model of FWUAVs provided in Figure 1 is a particularly complex, non-linear, under-actuated system. It is based on strongly coupled differential equations between the various flight parameters. The system uses four main inputs to control its movements: three deflection control surfaces (Elevator δe, Ailerons δa, and Rudder δl) and one propulsion control (Throttle δt). Precise adjustment of these four inputs enables the performance of various high-performance flight maneuvers, such as hovering, climbing, landing, or coordinated turns. These maneuvers are influenced by the system's dynamic states, including position (p_n, p_e, p_d), attitude angles (roll angle ϕ, pitch angle θ, yaw angle r), angular velocity (roll rate p, pitch rate q, yaw rate r'), linear velocity components (u, v, w), and the stability of airframe components (airspeed V_a, angle of attack α, and sideslip angle β). As a result, the dynamics of the system are represented in the Equation 1.

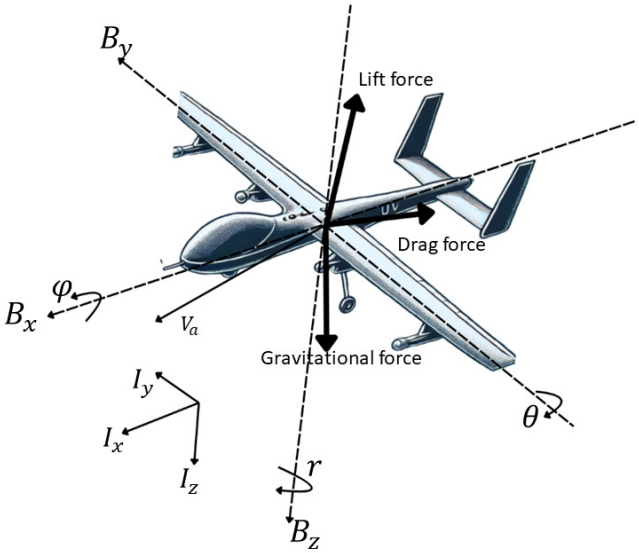


Figure 1: Fixed-wing UAV configuration

$$\begin{aligned} \dot{x} &= f(x,u) \\ x &= [p_n, p_e, p_n, u, v, w, \phi, \theta, r, p, q, r', V_a, \alpha, \beta] \\ u &= [\delta_e, \delta_a, \delta_l, \delta_t] \end{aligned} \tag{1}$$

To drive the FWUAV dynamical model, a set of assumptions is considered in this paper [26]:

- The UAV is considered a rigid body.
- The inertial mass of the UAV remains constant and is initially determined based on flight state values.
- The UAV is subjected to a uniform gravitational acceleration.
- The Earth is regarded as a fixed plane within inertial space.
- Changes in angular momentum due to rotating subsystems are omitted from consideration.

- Changes in angular momentum resulting from fuel movement and the motion of hinged components are also excluded.

The equations of motion of FWUAVs are established by taking into account two reference frames: a fixed inertial frame $I = (I_x, I_y, I_z)$ and a relative frame linked to the body of the system $B = (B_x, B_y, B_z)$. The translation and rotation dynamics, which describe all the dynamic phenomena acting on the UAV, are derived from the Newton-Euler formalization. This approach establishes a relationship between the linear and rotational accelerations and the forces and moments applied to the system. These fundamental equations, expressed as coupled relationships, are represented by Equations 2 and 3.

$$m \times \frac{dV}{dt}|_I = F_{aero} + F_{gravity} + F_{prop} \tag{2}$$

where $\frac{dV}{dt}$ is the linear velocity derivative vector expressed along the inertial reference frame I, while F_{aero} (lift and drag forces), $F_{gravity}$, and F_{prop} denote the external applied forces of the aerodynamic, gravitational, and propulsion forces, respectively.

$$J \times \frac{d\mathbf{\Omega}}{dt}\ |_B + \mathbf{\Omega} \times (J \times \mathbf{\Omega}) = \mathbf{T_{aero}} + \mathbf{T_{prop}} \tag{3}$$

Where $\frac{d\mathbf{\Omega}}{dt}$ represents the angular velocity derivative expressed in the body reference frame, $\mathbf{T_{aero}}$ and $\mathbf{T_{prop}}$ denote the moments acting on the UAV caused by the aerodynamic and propulsion effects following the three types of rotation (rolling, pitching, and yawing), and $\mathbf{J}$ is the moment of inertia matrix, which is derived under the assumption that the UAV's structure is symmetrically represented, leading to the following expression for $\mathbf{J}$:

$$\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & -J_{xz} \\ 0 & J_{yy} & 0 \\ -J_{zx} & 0 & J_{zz} \end{bmatrix} \tag{4}$$

Thus, the dynamic model of fixed wing in terms of rotation (ϕ, θ, r) rotation rates (p, q, r') and position (p_n, p_e, p_d) and velocity is given by the following equations:

$$\begin{aligned} \dot{p}_n &= c_\theta c_\psi u + (s_\phi s_\theta c_\psi - c_\phi s_\psi)v + (c_\phi s_\theta c_\psi + s_\phi s_\psi)w \\ \dot{p}_e &= c_\theta s_\psi u + (s_\phi s_\theta s_\psi + c_\phi c_\psi)v + (c_\phi s_\theta s_\psi - s_\phi c_\psi)w \\ \dot{p}_d &= -s_\theta u + s_\phi c_\theta v + c_\phi c_\theta w \end{aligned} \tag{5}$$

$$\begin{aligned} \dot{u} &= rv - qw - gs_\theta + m^{-1}\left(F_x(V_a, \alpha, q, \delta_e) + F_T(V_a, \delta_t)\right) \\ \dot{v} &= pw - ru + gc_\theta s_\phi + m^{-1}F_y(\beta, p, r', \delta_a, \delta_l) \\ \dot{w} &= qu - pv + m^{-1}F_z(V_a, \alpha, q, \delta_e) \end{aligned} \tag{6}$$

$$\begin{aligned} \dot{p} &= \Gamma^{-1}\left(\Gamma_1 rq - \Gamma_2 pr + T_l(\beta, p, r', \delta_a, \delta_l)\right) \\ \dot{q} &= J_y^{-1}\left((J_z - J_x)pr - J_{xz}(p^2 - r^2) + T_m(V_a, \alpha, q, \delta_e)\right) \\ \dot{r'} &= \Gamma^{-1}\left(\Gamma_3 pq - \Gamma_2 qr + T_n(\beta, p, r', \delta_a, \delta_l)\right) \end{aligned} \tag{7}$$

$$\begin{aligned} \dot{\phi} &= p + s_\phi \tan\theta\, q + c_\phi \tan\theta\, r' \\ \dot{\theta} &= c_\phi\, q - s_\phi\, r' \\ \dot{r} &= s_\phi \sec\theta\, q + c_\phi \sec\theta\, r' \end{aligned} \tag{8}$$

Where $c_i = \cos i, \quad s_i = \sin i \quad i = \phi, \theta, r$

$$\begin{aligned} \Gamma &= J_x J_z - J_{xz}^2 \quad \Gamma_1 = J_y J_z - J_z^2 J_{xz}^2 \\ \Gamma_2 &= (J_x - J_y + J_z)J_{xz} \quad \Gamma_3 = J_x^2 - J_y J_x - J_{xz}^2 \end{aligned} \tag{9}$$

$$\begin{aligned} V_a &= \sqrt{u^2 + v^2 + w^2} \\ \alpha &= \tan^{-1}\left(\frac{w}{u}\right) \\ \beta &= \sin^{-1}\left(\frac{v}{V_a}\right) \end{aligned} \tag{10}$$

In the equation above, m represents the UAV mass, and Γ, Γ_1, Γ_2, and Γ_3 are constants that are calculated by the moment of inertia matrix component J_x, J_y, J_z, J_{xz}.

F_x, F_y, F_z, and F_{prop} are the resulting aerodynamic, gravitational, and propulsion force components. T_l, T_m, and T_n denote the roll, pitch, and yaw resulting moment acting on the system, assuming that these parameters are non-linear functions depending on the aerodynamic coefficients and time-varying as the operating point of the vehicle, i.e., α, β, and V_a, changes that are updated using Equation 10. More details for the mentioned forces and moments, as well as all needed aerodynamic coefficients, are expressed in [26].

In a coordinated turn, an aircraft must adopt a specific roll angle to maintain a smooth turn without incurring lateral forces. During a turn, the lift generated by the wings is inclined, producing a vertical component to compensate for the force of gravity and a horizontal component to generate the centripetal force required to change heading. The roll angle ϕ is directly correlated to the rate of heading change $\dot{\psi}$. In the absence of wind and skid, the aircraft performs a smooth turn without lateral acceleration, ensuring a balance between gravitational and centrifugal forces. The relationship between the heading rate and the roll angle is kinematically expressed as follows:

$$\dot{\psi} = \frac{g}{V_a} \tan \phi \tag{11}$$

To ensure navigation at the desired speed and altitude, as well as the execution of coordinated turns while minimizing the disruptive effects of wind, it is essential to precisely control four fundamental parameters that directly influence the aircraft's longitudinal and lateral dynamics: airspeed V_a, altitude $z = -p_d$, heading angle ψ, and sideslip angle β. These parameters play a central role in the stability and control of the aircraft in flight. Therefore, an autopilot system must be designed with extreme rigor to regulate these parameters autonomously, thus guaranteeing stable, precise, and efficient flight performance, even under variable aerodynamic conditions.

3 Overview on the proposed methods

The Ant Lion Optimizer (ALO), Differential Evolution (DE), Bat Algorithm (BA), and Harmony Search (HS) are four chosen intelligent optimization methods used to tackle the complex control issues in FWUAV autopilot design. This section offers a comprehensive explanation of each method's application in this particular context.

3.1 Ant Lion Optimizer

Ant Lion Optimizer (ALO) is a collective intelligence-based optimization method developed by Sidali Mirjalili in 2015 [27]. The algorithm is inspired by the hunting behavior of antlions, which set traps for ants seeking food. In ALO, ants are used as search agents, while antlions guide them via a mathematically modeled random walk, as in Equation 12. Each ant adjusts its position according to the best solutions discovered in the search space. The algorithm is based on intelligent selection of the global optimum from a population of NP ants and antlions, initially distributed in a search space at D-dimension. Antlions influence the ants' random walk, increasing the probability of capturing them, and update their positions according to their fitness values to converge towards optimal solutions.

$$R(t) = [0, \text{cumsum}(2 \cdot r_1(t) - 1), \text{cumsum}(2 \cdot r_2(t) - 1), \ldots, \text{cumsum}(2 \cdot r_d(t) - 1)] \tag{12}$$

where $R(t)$ represents the position of the ant at time t, $r_j(t)$ (for $j = 1, 2, \ldots, D$) is a stochastic variable that represents a random binary value (either 0 or 1), and D is the dimension of the problem

space. This random walk is then normalized and confined within the boundaries of the search space by the antlion, ensuring that the solution search progresses effectively and balancing the exploration and exploitation processes in the algorithm.

3.2 Differential Evolution

Differential evolution, another strong metaheuristic algorithm used for complex optimization problems, was introduced by Strorn and Price in 1995 [28]. DE operates by evolving a population of candidate solutions over generations to be updated adequately by using three main operators of mutation, crossover, and selection all over to find the global optimal solution. It is an iterative process that tries to avoid the non-adoptive solution, starting with a randomly generated matrix of NP individuals within D-dimensional search space. Each individual is updated by operating the three main operators of the algorithm. The first one is the mutation that is created by using Equation 13 to promote a high level of search space exploration. While the crossover is used to gain the exploration of the obtained mutation element by combining each one to the current population individual, which is described by Equation 14. Finally, the selection process then tries to select the more adapted solution, retaining the better one for the fitness function for the next iteration. This process continues until a stop criterion, making DE effective for the multidimensional problem.

$$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}) \tag{13}$$

$$U_{i,j} = \begin{cases} V_{i,j} & \text{if } r_j \leq C_r \text{ or } j = j_{rand} \\ X_{i,j} & \text{otherwise} \end{cases} \tag{14}$$

where V_i is the mutant vector, F is the mutation coefficient, and X_{r1}, X_{r2}, X_{r3} are three randomly selected individuals from the current population. While U_{ij} denotes the trial vector element, C_r is the crossover probability.

3.3 Bat Algorithm

The BA algorithm is a nature-inspired intelligent optimization method created by Xin-She Yang in 2012 [29]. It is based on bats behavior in looking for food and prey, where the bats use the sound waves to navigate and locate their prey. It tries to emit pulses of sound and wait for the echoes to estimate the position-prey distance and the required velocity. This mechanism is modeled mathematically to create a process for getting the optimal solution and making the diverse search space exploitable and explorable. BA also requires a fitness function relevant to the search agent position, which is bats. In this case, it operates within a defined search space, ensuring diversity in the population by updating the bat position by its velocity and emission frequency, which also depend on the loudness and the pulse rate parameters. After a random initialized matrix of the NP bat population, the velocity of each bat in the current iteration is updated based on the current best position found so far, which is described by the following equations.

$$\begin{aligned} f_i &= f_{min} + (f_{max} - f_{min}) \cdot r_i \\ v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_*) \cdot f_i \\ x_i^t &= x_i^{t-1} + v_i^t \end{aligned} \tag{15}$$

where f_i, f_{min}, f_{max} are the current, minimum, and maximum frequencies, respectively; v_i denotes the velocity; x_i the current position; and x^* denotes the best position found in the current iteration.

The algorithms incorporate locale search x_{new} around the best-found position depending on the bat's pulse emission r_i^t and loudness A_i^t to intensify the exploration progress. The parameters of pulse rate and loudness are also dynamically adjusted as the algorithm progresses according to Equations 16, balancing exploration and exploitation to discover the global optimum.

$$\begin{aligned} A_i^{t+1} &= \delta A_i^t, \qquad 0 \prec \delta \prec 0 \\ r_i^{t+1} &= r_i^t(1 - \exp^{-\lambda t}) \qquad \lambda \succ 0 \end{aligned} \tag{16}$$

3.4 Harmony Search

The main idea behind the harmony search algorithm (HS) is that musical improvisation could be modeled as an optimization process, indeed, because trying to find a perfect harmony by improvising the notes of musical instruments and avoiding the false ones is similar to how the optimization works to find the best solution and avoid the worst ones. This methodology was developed by Greem in 2001 [30]. The algorithm process of HS uses to store the possible solution as a matrix of decision vectors, named by harmonic memory (HM). Harmony memory size (HMS) determines the size of this matrix and the number of candidates within HM. Similar to other metaheuristic algorithms, the HS also starts with the random initialization of HM within the possible range of the search space, and the updating solution x^i_{memory} is guided through memory consideration, pitch adjustment, and random consideration based on three parameters: harmony memory consideration rate $HMCR$, pitch adjustment PAR, and bandwidth bw. And the new solution, x^i_{new}, is computed using the following structure:

$$x^i_{\text{new}} = \begin{cases} x^i_{\text{memory}} & \text{avec une probabilité HMCR,} \\ x^i_{\text{memory}} \pm \text{rand}() \times bw & \text{avec une probabilité PAR,} \\ \text{rand}(x_{\min}, x_{\max}) & \text{avec une probabilité } 1 - \text{HMCR,} \end{cases} \tag{17}$$

where $rand()$ generates random numbers for broader exploration. HS iteratively updates the harmony memory by replacing the worst solution with a better one.

4 Autopilot System Design for Flight Control

The model equations described above exhibit strong non-linearity and coupling, making the system inherently complex and unstable while also making it particularly sensitive to environmental conditions and external disturbances. Consequently, controlling this type of behavior proves extremely difficult, especially when the control task must cover the entire system and be directly implemented while ensuring that the temporal response meets design requirements and delivers robust performance. To overcome these challenges, a successive loop closure approach is adopted for the autopilot system control tasks. This approach involves the use of several simple feedback loops chained around the open system, thus eliminating the need to design a single, complex control system. Instead, a system specifically designed to make rapid and precise control adjustments is preferred.

The autopilot system is designed to control a fixed-wing drone and provide precise navigation as required. It incorporates four blockchain controllers designed to respond to longitudinal and lateral movements with fully coupled system parameters. These chains are proposed for this application according to the control laws described by Equation 18, in order to maintain the desired airspeed, altitude, heading angle, and sideslip angle. The first controller is dedicated to maintaining the desired airspeed and is designed with a proportional gain of K_v and K_{dv}, due to the stable open-loop speed behavior observed. The second controller corresponds to the pitch chain with commanded altitude, taking into account current feedback of pitch rate q, pitch angle θ, and altitude z to ensure pitch and altitude stabilization by adjusting K_q, K_θ, K_z gains. The third chain concerns roll with commanded heading, developed in a similar way to the pitch chain, with Kp, K_ϕ, K_ψ gains to guarantee roll and heading angle stabilization. Finally, the fourth chain is dedicated to sideslip angle, using two gains, K_β and $K_{r'}$, for the same purpose of stabilization.

$$\begin{aligned} \delta t &= K_v(V_{ac} - V_a) + K_{dv}\dot{V}_a \\ \delta e &= K_\theta(K_z(z_c - z) - \theta) - K_q q \\ \delta a &= K_\phi(K_\psi(\psi_c - \psi) - \phi) - K_p p \\ \delta l &= K_\beta(\beta_c - \beta) - K_r r' \end{aligned} \tag{18}$$

where V_{ac}, z_c, ψ_c, and β_c denote the desired velocity, altitude, heading angle, and sideslip angle, respectively.

The overall stability of the proposed cascade control architecture for fixed-wing drones is addressed through Lyapunov stability analysis, a widely used method to rigorously demonstrate the stability of closed-loop systems. Indeed, the integration of cascade control architectures often poses hierarchical stability problems, requiring rigorous analysis based on the Lyapunov method to guarantee the system's stability and overall performance, as discussed in [31, 32]. Furthermore, several studies have shown that cascade structure controllers, combined with nonlinear control techniques, are analyzed using candidate Lyapunov functions to ensure the asymptotic stability and robustness of fixed-wing UAV systems [33, 34]. Although the proposed control is based on a cascade structure, its stability aligns with this well-established theoretical framework, as confirmed by our simulation results illustrating asymptotically stable trajectory tracking performance even in the presence of nonlinear dynamics and disturbances.

Figure 2 illustrates the comprehensive flight controller system for the fixed-wing UAV. The four blocs

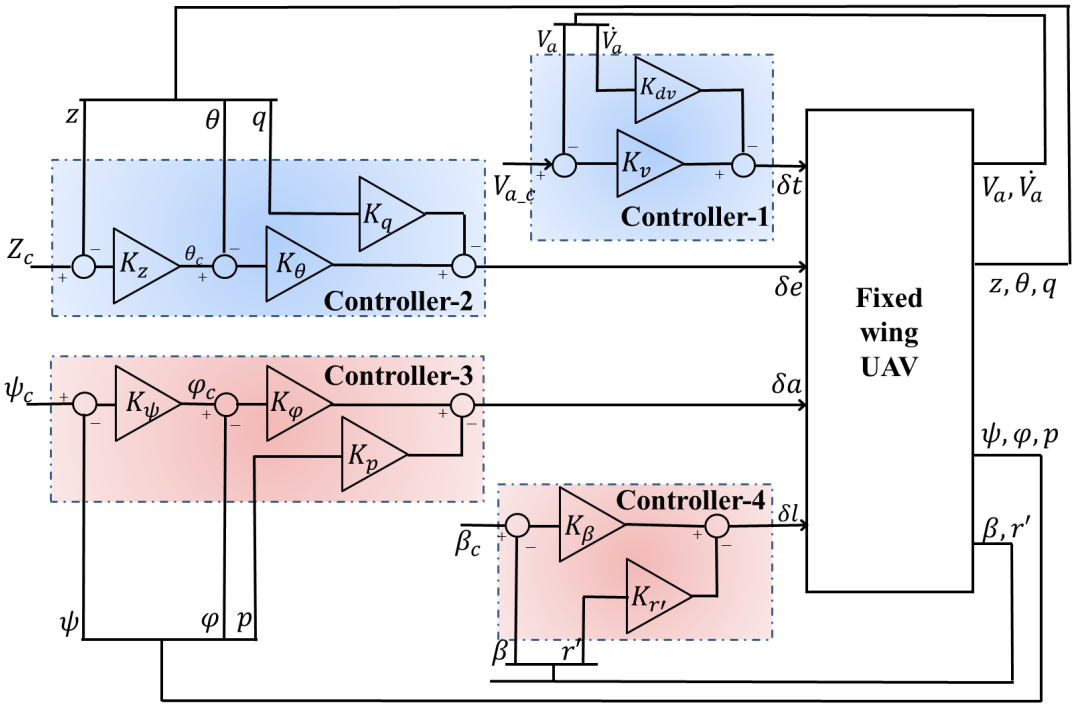


Figure 2: The autopilot system control loop

(controllers) operate as successive loop closure configurations, employing ten proportional actions to stabilize attitude, altitude, and velocity. The complete autopilot controller architecture receives as inputs the current variable state and the predefined desired signals for altitude, airspeed, heading, and sideslip angle $(V_{ac}, z_c, \psi_c, \beta_c)$ according to the autopilot system requirements. In contrast, the sensor signals for the desired pitch and roll angles (ϕ_c, θ_c) are derived from the commanded altitude and heading angles. All gains require appropriate optimal tuning inside a fully coupled system to ensure precise, rapid, stable, and resilient performance.

5 Optimal Flight Control System Tuning

The control system has been designed using Equations 18 is directly based on the architecture shown in Figure 2. Its main objective is to stabilize the system studied through its key flight parameters: airspeed, altitude, heading angle, and sideslip angle. To this end, 10 parameters corresponding to the proportional actions of the four controllers developed were identified as variables to be optimized for the desired performances. Consequently, the ALO, DE, BA, and HS algorithms were used to solve this optimization problem.

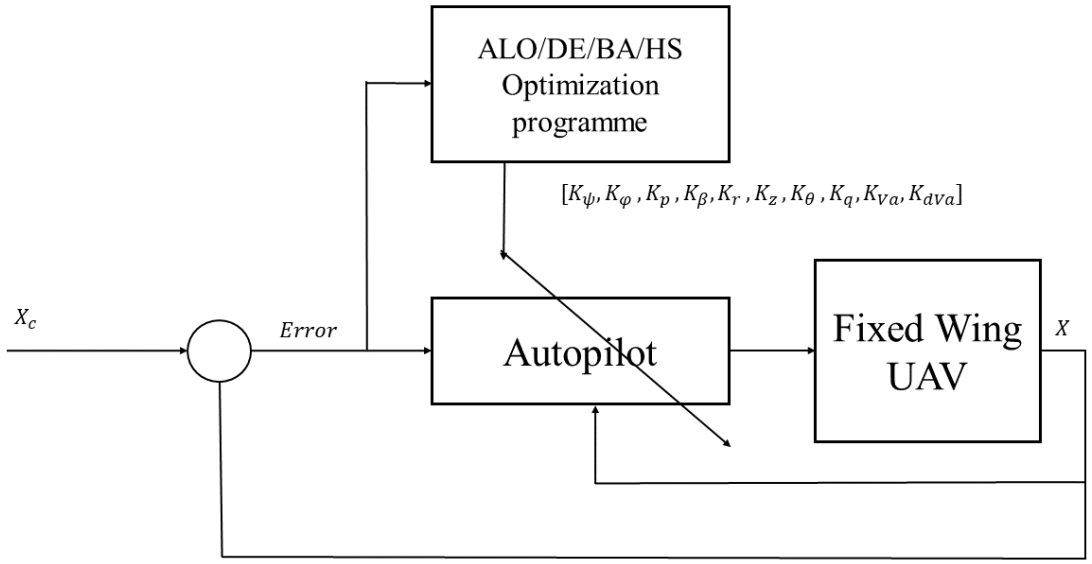


Figure 3: The block diagram of the optimal autopilot system using ALO, DE, BA, and HS.

A matrix of candidate solutions was randomly generated to initialize each parameter, thus exploiting the stochastic search dynamics specific to these algorithms. This matrix constitutes the search space where each element represents an agent or a candidate solution. Each candidate vector $P = [K_\psi, K_\phi, K_q, K_\beta, K_r, K_z, K_\theta, K_q, K_{V_a}, K_{dV_a}]$ contains all the controller gains and will be updated by the dynamics algorithms to ensure fast, accurate, and robust transient responses. An initial population of 100 agents was adopted in this process.

Minimizing the deviation between desired values and regulated responses was addressed by considering three performance measures: overshoot, settling time, and steady-state error. These criteria were combined to form a fitness function offering an overall assessment of the quality of candidate solutions and their ability to satisfy performance requirements. The optimization problem was divided into four elementary functions dedicated respectively to airspeed, altitude, heading, and sideslip, as shown in Equations 19 and 20.

$$f_i = aOv_i + bT_i + cE_i \qquad where \quad i = (V_a, \quad z, \quad \psi, \quad \beta) \tag{19}$$

$$F = \sum f_i \tag{20}$$

The composite fitness function F ensures a balanced evaluation by assigning different weights to the performance indexes where $a = 1, b = 0.75, and \quad c = 0.4$, and it was calculated using simulations on the block diagram shown in Figure 3. The tuning parameters specific to each algorithm, which influence their ability to explore solutions, were chosen by successive trials and are summarized in the following Table 1 below. An optimal set of parameters for all four controllers results in superior system performance and minimized performance indices. Table 2 compares the gains obtained using the ALO, DE, BA, and HS methods.

6 Simulations and Results

In this section, a series of simulations is conducted to evaluate the efficiency and robustness of the proposed autopilot architecture, and they are performed using the ALO, DE, HS, and BA algorithms applied to an FWUAV under various flight scenarios. These scenarios include stabilization at fixed reference points for airspeed, altitude, heading angle, and sideslip angle, as well as trajectory tracking in the presence of dynamic disturbances. The MATLAB/Simulink environment is used to implement

Table 1: Parameter tuning

Algorithm	Parameter	Signification	Value
ALO	NP	Population Size	100
	T	Number of iterations	40
DE	F	Mutant constant	0.6
	CR	Crossover	0.9
BA	$[f_{\min}, f_{\max}]$	Frequency range	[0.6, 0.9]
	r_i^0	Initial Pulse Rate	0.5
	A_i^0	Initial Loudness	0.5
HS	HMS	Harmony Memory Size	20
	N_Impro	Number of Improvisations	600
	HMCR	Harmony Memory Consideration Rate	0.9
	PAR	Pitch Adjustment Rate	0.5
	bw	Band Weight	0.7

Table 2: Optimal autopilot parameters obtained with Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS) for the fixed-wing UAV

		Intelligent Algorithms			
Controller	**parameter**	**DE**	**ALO**	**BA**	**HS**
Heading angle ψ	K_ψ	3.5620	6.4612	1.8249	3.8169
	K_ϕ	-2.4131	-3.3476	-2.1407	-3.6776
	K_p	-0.2118	-0.0010	-0.6998	-0.1836
sideslip anlge β	K_β	-0.2841	-0.0001	0	-0.0635
	$K_{r'}$	0	-0.6993	0	-0.0266
altitude z	K_z	-0.1518	-0.0523	-0.1860	-0.0440
	K_θ	-3.6923	-2.4745	-2.9002	-3.2881
	K_q	0	-0.0002	-0.0034	-0.3168
Airspeed V_a	K_{V_a}	29.0462	11.3555	29.8395	15.0553
	$K_{\dot{V}_a}$	12.5829	5.9856	0	16.2147

intelligent control strategies and determine the optimal autopilot parameters, aiming to enhance system performance and achieve the desired control objectives. Table 3 summarizes the geometric and technical specifications of the FWUAV model used in the simulations.

6.1 Desired Setpoint Holding Flight

The main objective of this scenario is to enable the controller to maintain a hover under standard conditions, around predefined fixed values. Initial flight conditions and setpoints for airspeed, altitude, heading angle, and sideslip angle are specified by Equations 21 to 22, guaranteeing a stable flight point to be reached and maintained.

$$[pn_0, pe_0, z_0, \phi_0, \theta_0, \psi_0, Va_0] = [0m, 0m, 100m, 0^\circ, 0^\circ, 0^\circ, 23m/s] \tag{21}$$

$$[V_{ac}, \quad z_c, \quad \psi_c, \quad \beta_c] = [25m/s, 110m, 30^\circ, 0^\circ] \tag{22}$$

The purpose of the sideslip control chain is to sustain the sideslip angle β at 0 to avert lateral drift of the drone during flight. The optimal controller parameters obtained via each optimization algorithm are summarized in Table 2, thereby enabling direct comparison and full comprehension of our tuning results. The time responses of the controlled flight parameters (airspeed, altitude, heading angle, and sideslip angle) are shown in Figures 4 to 7, while the corresponding control signals $(\delta e, \delta a, \delta l, \delta t)$ are illustrated in Figure 8. Additionally, the comprehensive analysis and comparison between the studies methodologies is offered by the metrics performance and include a statistical assessment shown in

Table 3: The parameters of the FWUAV

Parameter	signification	Value
J_x	The x-axis Moment of inertia	0.7795 $kg.m^2$
J_y	The y-axis Moment of inertia	$1.122kg.m^2$
J_z	The z-axis Moment of inertia	1.752 $kg.m^2$
J_{xz}	The xz-plane Moment of inertia	$0.1211kg.m^2$
J_{prop}	Propeller moment of inertia	0.002 $kg.m^2$
R_{prop}	Propeller radius	$0.254m$
c	Chord	$0.18m$
b	Wind span	$2.8956m$
S	Wing area	$0.55m^2$
g	Gravitational	$9.81m/s^2$
V_a	Airspeed bounds	$[15\ 50]m/s$
m	weight	13.5 kg

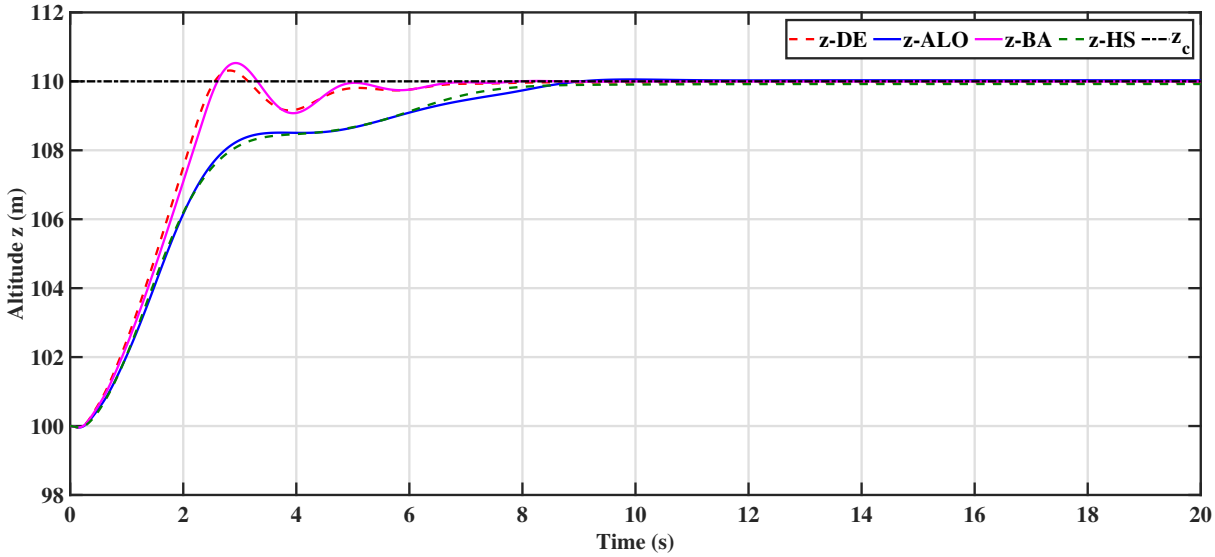


Figure 4: Fixed-wing UAV's altitude z using Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS).

Table 4 and Table 5, including the overshoot Ov, the settling time T_s and mean RMSE across 10 independant runs.

The simulation results demonstrate the efficacy of the suggested autopilot architecture in precisely controlling V_a, z, ψ, and β, despite the UAV system's coupled, non-linear dynamics. The curves in Figures 4 to 7 demonstrate that the chosen algorithms, ALO, DE, BA, and HS, proficiently utilize their exploration and exploitation dynamics to provide appropriate proportionate gains for each control loop. This feature ensures adherence to desired performance metrics across all dedicated flight parameters. Additionally, it can be clear that the autopilot's modular closed-loop architecture has successfully facilitated the management of interactions among flight dynamics while maintaining system stability. Whereas, the decrease in parameters enhances the optimization efficiency and allows the algorithms to calibrate the controllers with exceptional accuracy. These insights highlight the autopilot system's capacity to overcome the UAV control complexity while maintaining performance integrity.

Regarding the numerical metrics in Table 4, the performance among the optimization strategies utilized for the autopilot is highlighted. For the airspeed parameter, DE and ALO consistently outperformed the other algorithms, achieving the shortest settling times (3.1908 s), acceptable overshoot, and the lowest Integral of Squared Error (ISE), with values of 7.5618 and 8.8322, respectively. These

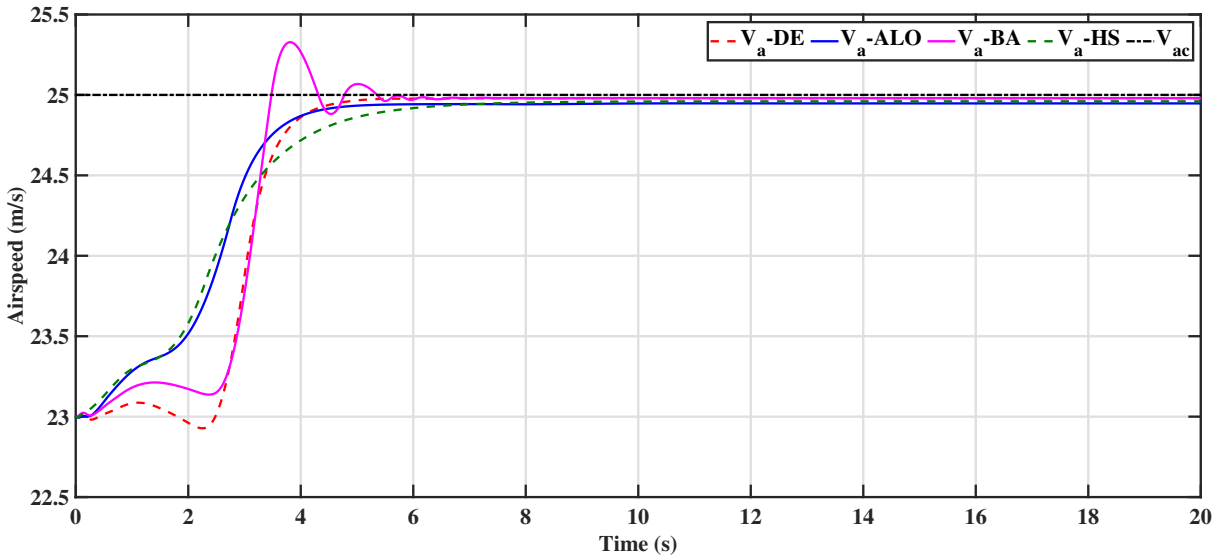


Figure 5: Fixed-wing UAV's airspeed V_a using Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS).

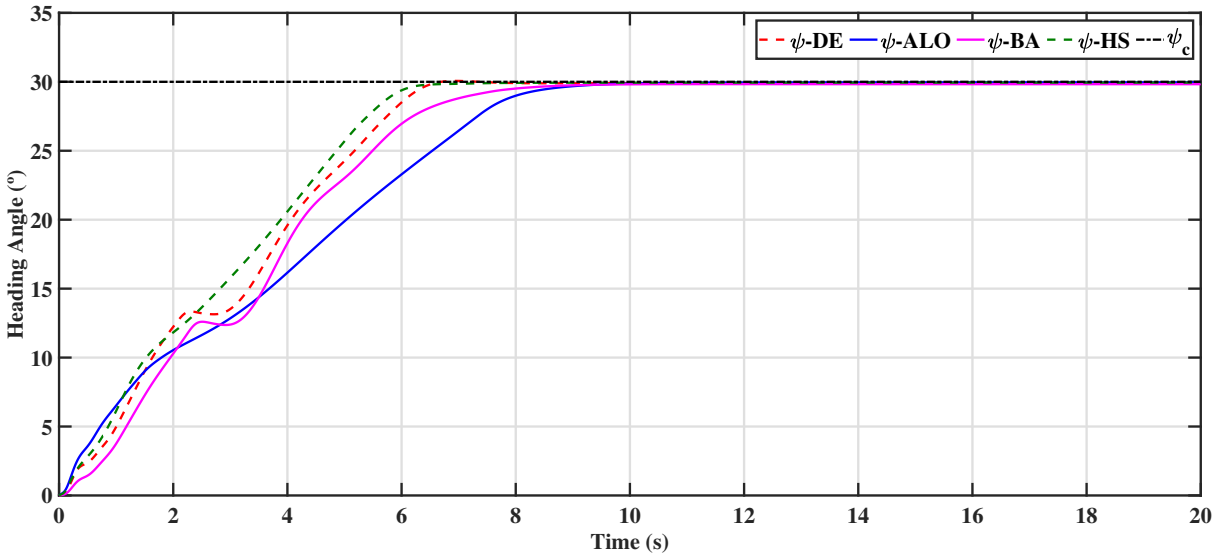


Figure 6: Fixed-wing UAV's heading angle ψ using Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS).

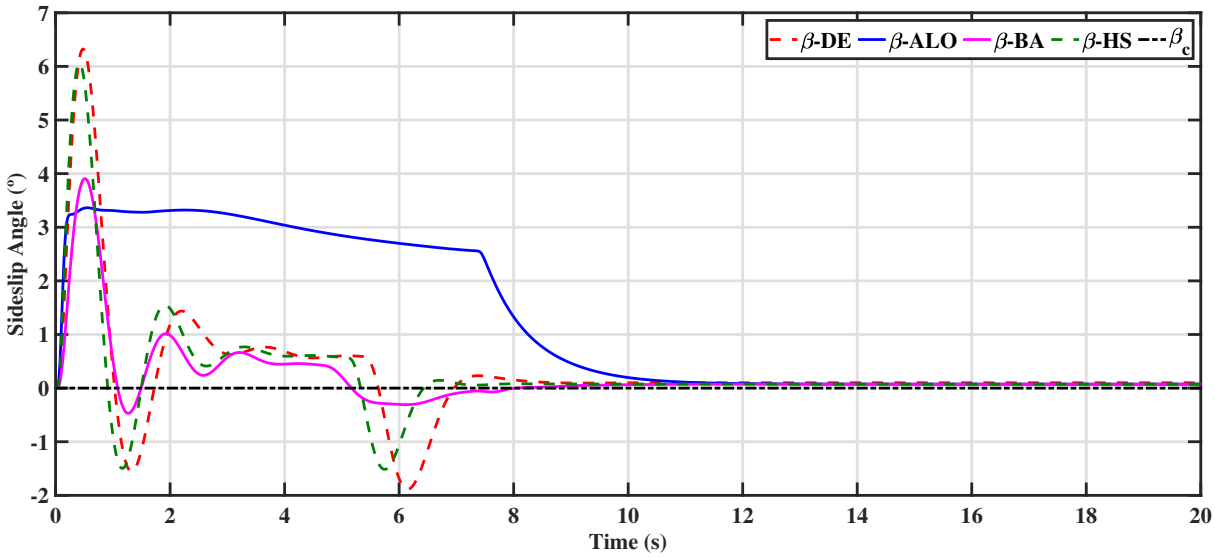


Figure 7: Fixed-wing UAV's sideslip angle β using Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS).

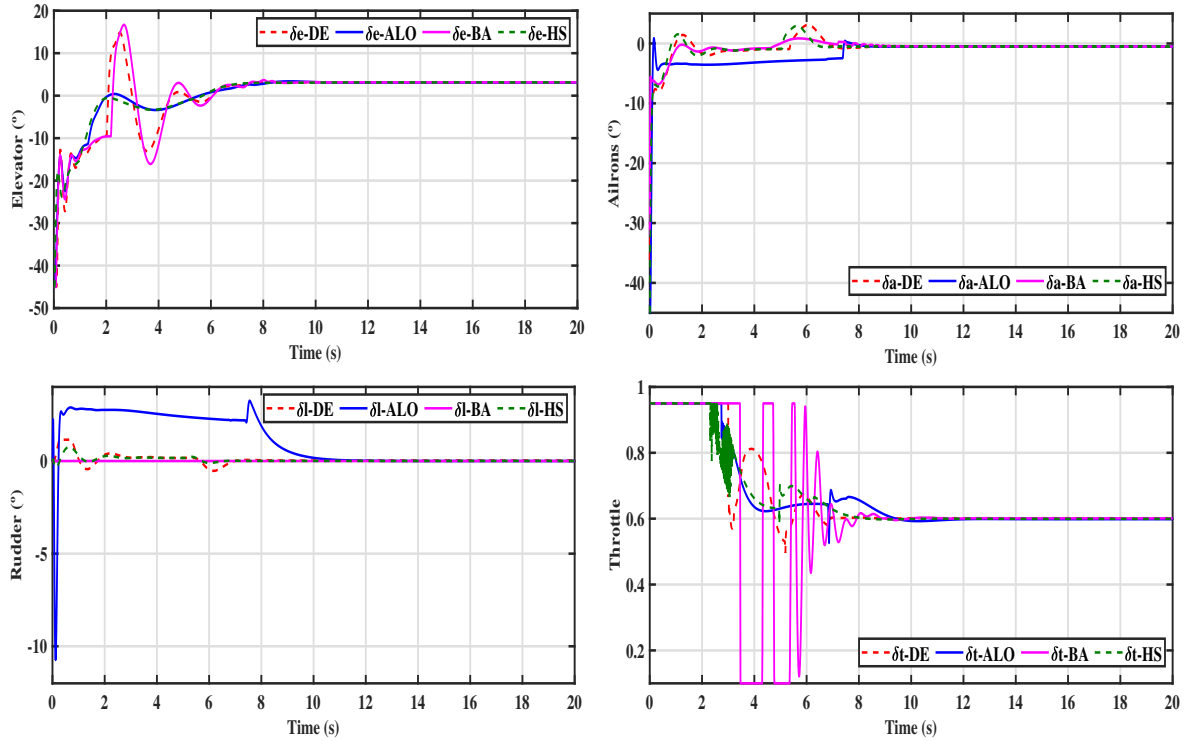


Figure 8: The obtained control laws for Elevator δe, Ailrons δa, Rudder δl, and Throttle (δt) using Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS).

Table 4: Performances obtained with Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS) for the fixed-wing UAV.

output parameter	**performance**	**Intelligent Algorithms**			
		DE	**ALO**	**BA**	**HS**
V_a	T_s	3.1908	3.2603	2.9498	3.3325
	Ov	0.0	0.0012	1.4019	0.0001
	ISE	7.5618	10.3176	7.8591	11.4846
	IAE	10.9121	8.8322	12.5040	9.2047
	$ITSE$	26.5700	19.5109	39.7467	21.6491
	$ITAE$	456.7204	236.4117	600.1468	243.2234
z	T_s	2.6087	2.1065	2.5892	2.0221
	Ov	0	0.4994	0.0181	0.3242
	ISE	133.8442	116.4806	134.6100	112.5975
	IAE	34.7789	17.8759	26.8220	21.2580
	$ITSE$	185.9892	86.9537	152.9012	90.9155
	$ITAE$	613.3255	31.4377	679.2017	303.8817
ψ	T_s	5.5139	6.5348	7.4238	5.7219
	Ov	0.3802	0.0402	0.0000	1.3935
	ISE	0.4965	0.6145	0.6078	0.5413
	IAE	1.6262	2.1886	1.9989	1.8249
	$ITSE$	0.7256	1.1012	1.1881	0.8618
	$ITAE$	14.1718	37.7356	12.1316	22.1279

Table 5: Mean RMSE and standard deviation over 10 runs for UAV control parameters

Algorithm	$RMSE_{V_a}$	$RMSE_z$	$RMSE_\psi$	$RMSE_\beta$
DE	1.19 $\pm$(0.04)	1.07$\pm$(0.07)	0.07$\pm$ (0.005)	0.007$\pm$(0.002)
ALO	1.59$\pm$(0.25)	1.07$\pm$(0.35)	0.14$\pm$(0.19)	0.011$\pm$(0.007)
BA	1.19$\pm$(0.06)	1.04$\pm$(0.07)	0.08$\pm$(0.01)	0.013$\pm$(0.008)
HS	1.20 $\pm$(0.06)	1.09$\pm$(0.15)	0.28$\pm$(0.53)	0.011$\pm$(0.007)

results indicate that DE and ALO, in particular, provide the most precise and stable control response. The Bat Algorithm (BA) exhibited moderate settling time and ISE values, suggesting acceptable but less optimal performance. In contrast, Harmony Search (HS) showed the minimum overshoot and a significantly longer settling time (3.3325 s), reflecting a slower response. Overall, DE emerges as the most effective algorithm for airspeed control within the proposed autopilot framework.

For the altitude, ALO and HS are performing slightly better at 2.1065 s and 2.0221 s, respectively. In terms of overshoot, DE and BA demonstrated near-zero values, indicating highly stable responses. However, DE exhibited a higher Integral of Squared Error (ISE) at 133.8442 compared to HS (112.5975) and ALO (116.4806), suggesting that while DE maintained stability, it was less accurate in minimizing cumulative error over time. Overall, HS achieved the best balance between rapid response and low error, while DE and ALO offered strong stability with slightly higher ISE values. These results highlight the trade-offs between precision and responsiveness in altitude control and demonstrate the value of selecting the appropriate algorithm based on mission requirements. For heading angle control, the simulation results indicate that DE and HS achieved the fastest settling times, at 5.5139 s and 5.7219 s, respectively. The BA demonstrated the best overshoot performance with a value of 0.0000, followed closely by ALO (0.0402). In terms of cumulative error, DE achieved the lowest Integral of Squared Error (ISE) at 0.4965, indicating a highly accurate response. Although ALO and BA had slightly higher ISE values (0.6145 and 0.6078), their performance remained competitive. HS, while fast in response, exhibited the highest overshoot (1.3935), which may affect stability in sensitive applications. Overall, DE again stands out for its balance of speed and accuracy, while BA offers excellent overshoot control with slightly slower convergence.

Regarding the results contained in Table 5, analysis of the RMSE results over 10 independent runs reveals that the DE and ALO algorithms offer significant accuracy and stability gains in the adjustment of parameters for controlling V_a, z, ψ, and β. Furthermore, BA and HS also offer competitive results, although their error level is slightly higher than that of DE, with moderate variability. These observations justify that the DE and ALO are preferred algorithms, due to its balance between accuracy and robustness in the context of automatic autopilot adjustment. The other algorithms show greater variability, particularly for certain parameters, highlighting the importance of robust statistical evaluation to ensure the reliability of optimizations.

Among the four algorithms evaluated, the DE and ALO algorithms excel in overall performance due to their capacity to integrate accuracy and stabilization speed, especially for crucial parameters like V_a and z. This feature emphasizes its strength and efficacy in managing UAV nonlinearities and dynamic couplings, making them a favored choice for applications necessitating accurate and rapid control.

In addition, it would be beneficial to mention that all optimization experiments were performed on a workstation equipped with an Intel i3 processor and 16 GB of RAM. Each algorithm was allowed a maximum number of predefined iterations/generations: 40 for the DE, ALO, and BA algorithms, and 600 improvisations for HS. Furthermore, the average convergence time was approximately 30.13 min for DE and 32.03 for ALO, while BA and HS required slightly less computation time, around 28.34 min. For all methods, convergence was defined either by reaching the maximum number of iterations and by minimizing the choosed objective function in Equation 20. In terms of computational costs, DE and ALO generally performed an accurate number of fitness function evaluations per run, due to their population-based nature. Hence, their robust convergence profiles resulted in better optimized parameters. All algorithms completed their calculations within practical limits for offline autopilot tuning. .

After examining the results, it is clear that the creation and improvement of our autopilot using the DE and ALO algorithms performs exceptionally well when compared with other evolutionary techniques. This underlines the value of the solutions obtained, making the studied method a particularly competitive option for comparisons with techniques drawn from the literature. As reported in [19], the Genetic Algorithm (GA) optimized Intelligent PID showed a clear improvement compared to the traditional PID, decreasing the stabilization time for altitude from 10.28 s to 2.73 s and for velocity from 11.39 s to 7.73 s. However, when comparing these results with those of our DE or ALO based approaches, it is clear that our configuration outperforms them, showing additional decreases in stabilization time of 22% and 58% respectively for altitude and airspeed, while maintaining similar simulation conditions.

The article [35] points out the versatility of soft computing methods, in particular through the use of hybrid systems combining fuzzy logic and neural networks (ANFIS). This technique has proved its competitiveness against PID-GA, with significant reductions in stabilization times. However, compared with ALO's performance, our method still manages to reduce this time by 16% for altitude and 59% for airspeed.

Our DE-based autopilot system also exceeds the performance of the fuzzy PID controller presented in reference [11], with regard to heading angle. It reduces the turn time from 9.41 to 5.5139 seconds, which represents an improvement of 41%. Overall, the particular modifications and innovations that the ALO and DE algorithms have introduced demonstrate superior competitiveness, outperforming not only traditional approaches but also soft computing techniques such as FL, ANFIS, GA, and NN.

6.2 Disturbances Rejection via Attitude Stabilization

The objective of this simulation is to assess the efficacy of our autonomous piloting system to track with variable reference, including external added disturbance. that is induced by a 2 m/s wind speed originating from a direction angled at 60° to the north and subsequently modeled. The disturbance was generated with advanced Simulink tools using the horizontal wind model, the wind shear model,

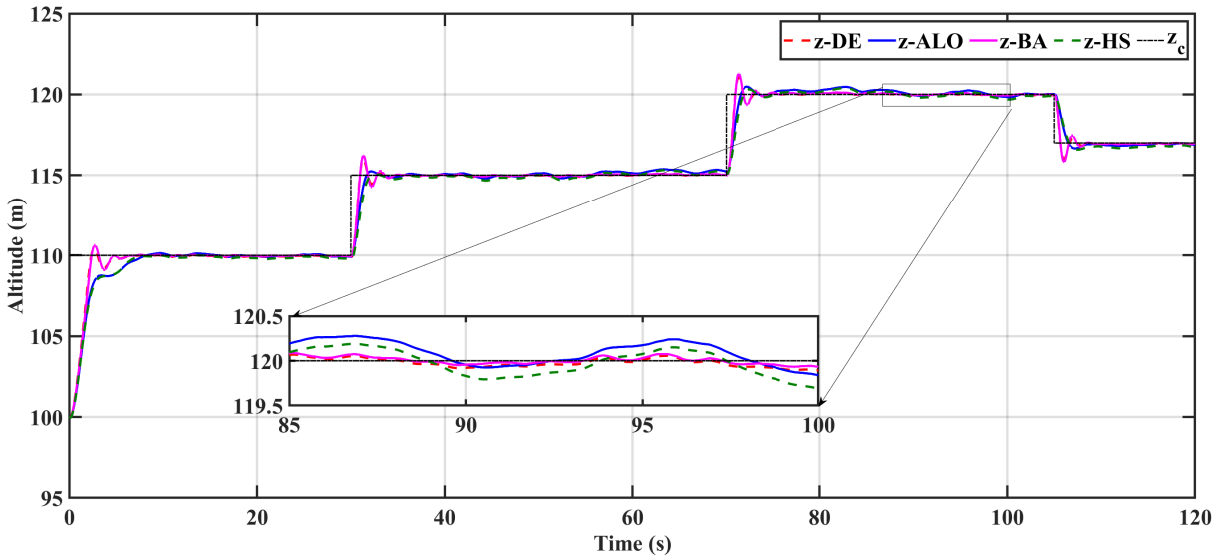


Figure 9: The fixed-wing UAV altitude z obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS). with changed reference and disturbance

and the Dryden wind turbulence model predefined in Simulink. These models are engineered to replicate real-world airflow properties precisely, considering turbulence events and unforeseen variations. Figures 9 to 12 illustrate the graphical progression of flight parameters (airspeed, altitude, heading, and sideslip) when the external disturbance of wind is added. Based on the observations made in the figures, it is evident that the intelligent algorithms effectively reject the influence of disturbances on the overall system dynamics. Target values are adhered to precisely, and wind disturbances are quickly mitigated, even with time differences in reference values. This result demonstrates the flexibility and effectiveness of the proposed controller. Furthermore, the study indicates that performance is significantly influenced by the autopilot's architecture, defined by its interconnected modules and loops. By isolating critical flight parameters, including airspeed, altitude, heading, and sideslip, and optimizing them dependently using ALO, DE, BA, and HS, the system ensures superior responsiveness and improved stability, even under dynamic turbulence.

When comparing algorithms, the DE and ALO algorithms excel in generating a high-quality response in complex scenarios, especially those involving wind-induced turbulence. Results show that in the most critical situations, the ALO algorithm facilitates rapid turbulence mitigation and enhances accuracy in maintaining the necessary parameters. This exceptional performance proves its suitability for applications requiring enhanced resilience in turbulent conditions.

6.3 Orbital and Straight line Trajectory Tracking

In order to evaluate the robustness and performance of the proposed controller under higher-level situations, its capacity to follow specified trajectories is evaluated with and without wind gusts. This evaluation verifies the system's precision and dependability when encountering difficult trajectories, regardless of external perturbations.

The two types of paths followed are a linear trajectory and an orbital trajectory, the obtained performance is depicted in Figures 13 and 15, respectively. These graphs illustrate the outcomes of intelligent controllers utilizing the DE, ALO, BA, and HS algorithms under optimal settings, devoid of disruptions. Conversely, Figures Figures 14 and 16 depict the impact of disturbances, simulated by winds (2 m/s, originating from a direction of 60° relative to north), on the persistence of these trajectories. Table 6 presents details for the performance index corresponding to each flight parameter, including speed, altitude, heading, and sideslip. These performance metrics, obtained by tracking errors, offer a quantitative evaluation of the efficacy of the different algorithms.

The simulations demonstrate that, despite the existing disturbance, all algorithms successfully

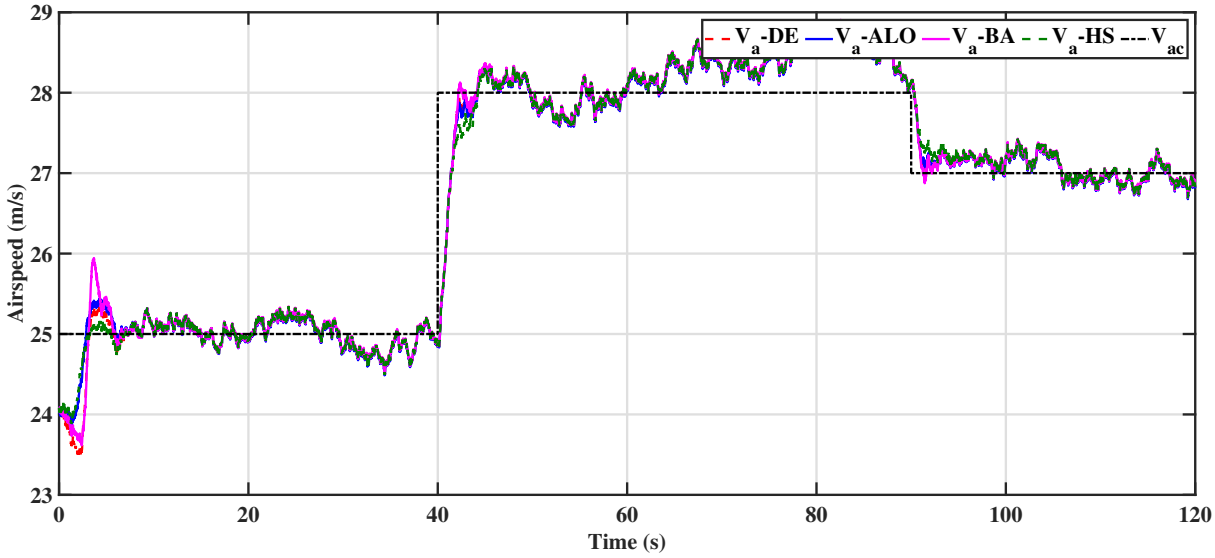


Figure 10: The fixed-wing UAV velocity V_a is obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS). with changed reference and disturbance

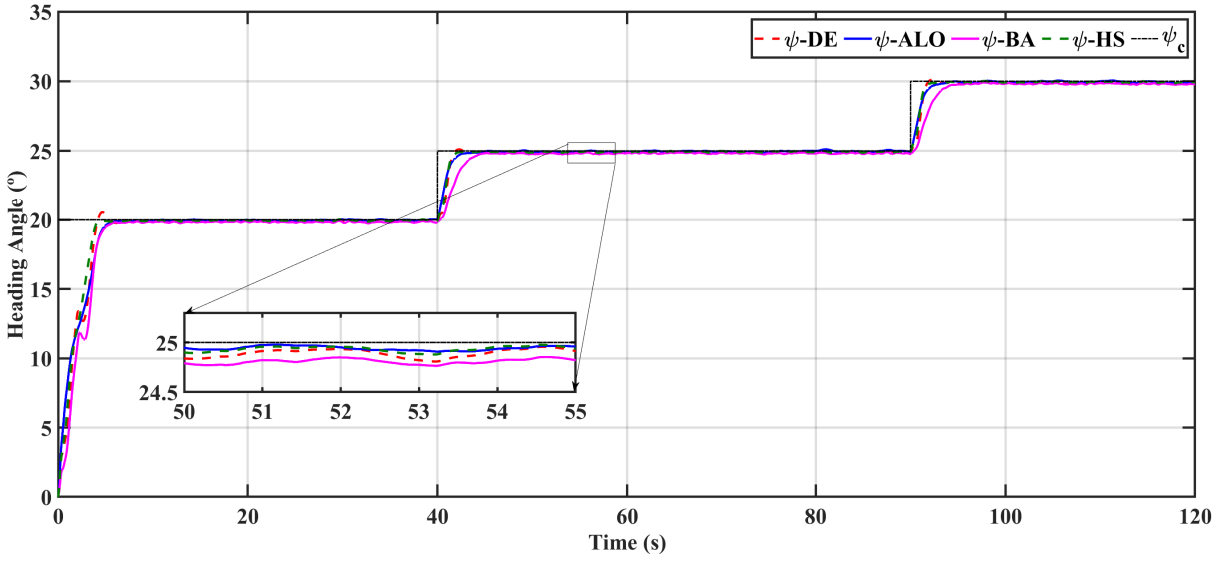


Figure 11: The fixed-wing UAV heading angle ψ is obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS). with changed reference and disturbance

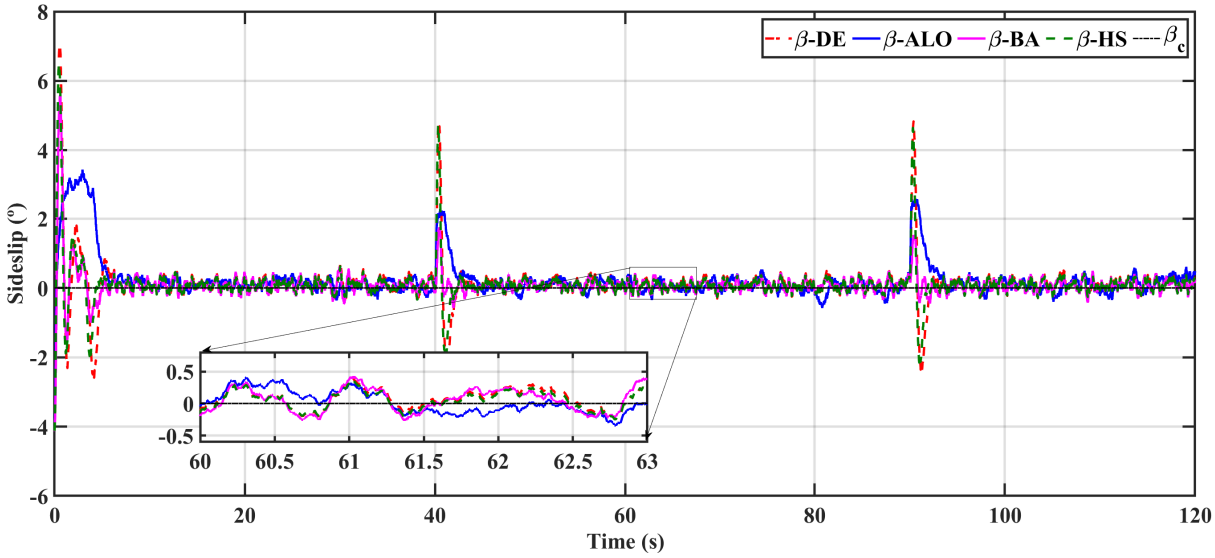


Figure 12: The fixed-wing UAV sideslip angle β obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS). with disturbance.

maintain appropriate trajectory tracking, hence affirming the robustness of the proposed system. However, performance inequalities remain: the Differential Evolution (DE) algorithm demonstrates greater accuracy in tracking linear trajectories, whereas the Artificial Lion Optimizer (ALO) shows enhanced ability in managing orbital trajectories, particularly under the presented disturbance. The results show the controller's ability to adjust to varying conditions while minimizing tracking errors.

Table 6: The obtained performance ISE using DE, ALO, BA, and HS

output parameter	**performance**	**Performance Indexe ISE**			
		z	ψ	β	V_a
Straigth line	DE	0.7447	0.1518e-03	0.2188e-04	10.4787
	ALO	0.5024	0.2265e-03	0.1097e-04	10.8049
	BA	0.8238	0.1441e-03	0.3669e-04	10.5645
	HS	0.4831	0.1715e-03	0.3379e-04	10.8887
Orbital	DE	0.6110	0.0258	0.0340e-03	1.1914
	ALO	0.4865	0.0355	0.0147e-03	1.3849
	BA	0.6091	0.1211	0.4590e-03	1.2222
	HS	0.4719	0.0236	0.0511e-03	1.4102

The figures and Table 7 results demonstrate that the employed algorithms effectively follow the linear and the orbital trajectories, even in the presence of disturbances, showing a promising outcome. Significant variations are present in the performance metrics associated with flight parameters. The DE algorithm effectively regulates the altitude and the heading angle along a straight-line trajectory, achieving high performance metrics in terms of ISE. Such behavior confirms its effectiveness in maintaining this essential characteristic consistently. The HS algorithm demonstrates superior altitude performance for straight trajectories, closely followed by the ALO, highlighting their effectiveness for controlling altitude variations in the complexities of the system's non-linear dynamics.

The ALO stands out for its high precision in maintaining the sideslip angle, especially while following an orbital trajectory. This highlights its effectiveness in decreasing lateral disturbances caused by crosswinds. The DE and HS algorithms demonstrate superior performance in heading orientation, particularly in relation to orbital paths. Their ability to maintain high directional stability is crucial for the success of complex operations. The comparison of the two trajectory types illustrates the importance of enhanced disturbance management and synchronized dynamics in orbital trajectories, while ALO and DE are characterized by their greater robustness. The results highlight the autopi-

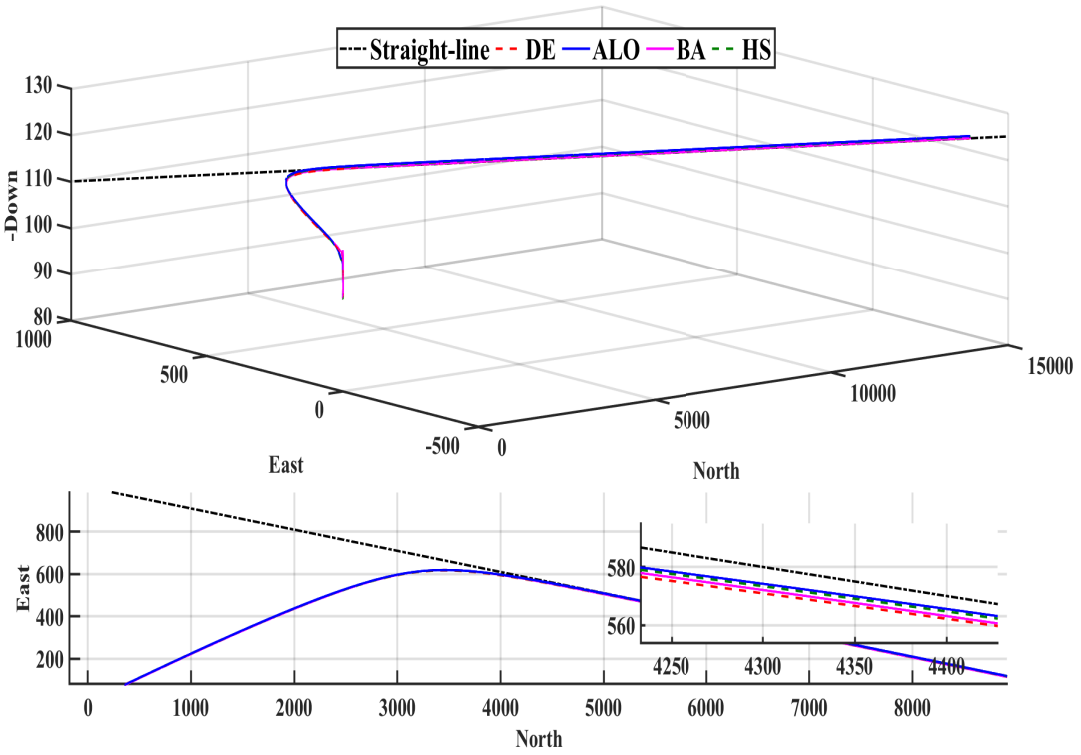


Figure 13: The fixed-wing UAV's position $(pn, pe, z = -pd)$ was obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS) (straight-line path following).

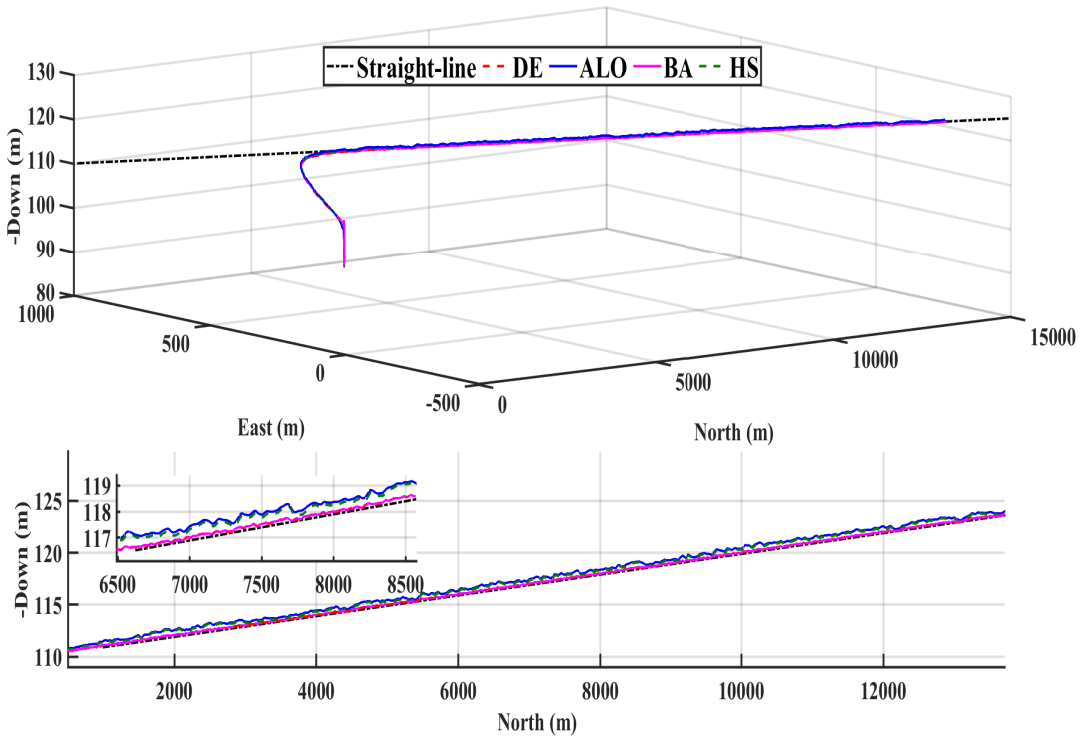


Figure 14: The fixed-wing UAV's position $(pn, pe, z = -pd)$ was obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS) with disturbance (straight line path follow).

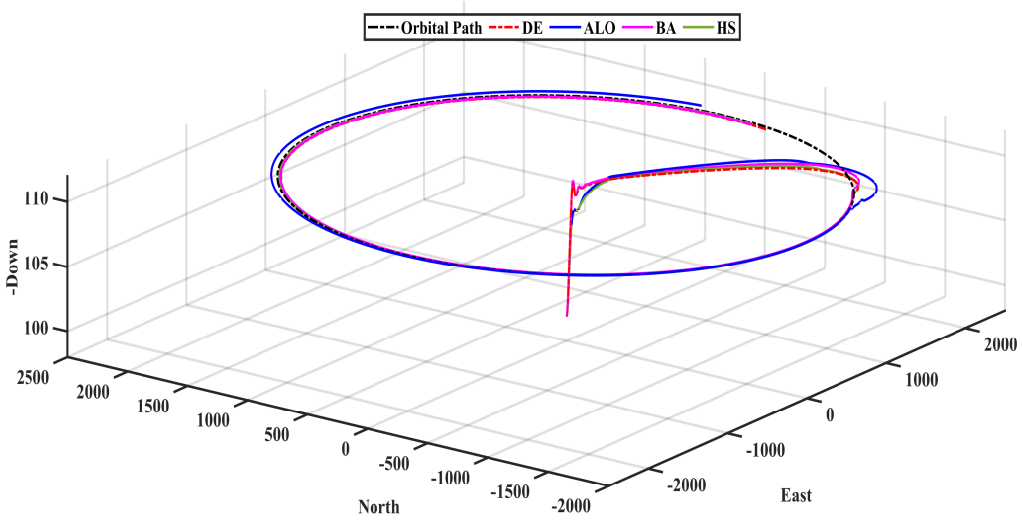


Figure 15: The fixed-wing UAV's position $(pn, pe, z = -pd)$ was obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS) (orbital path follow).

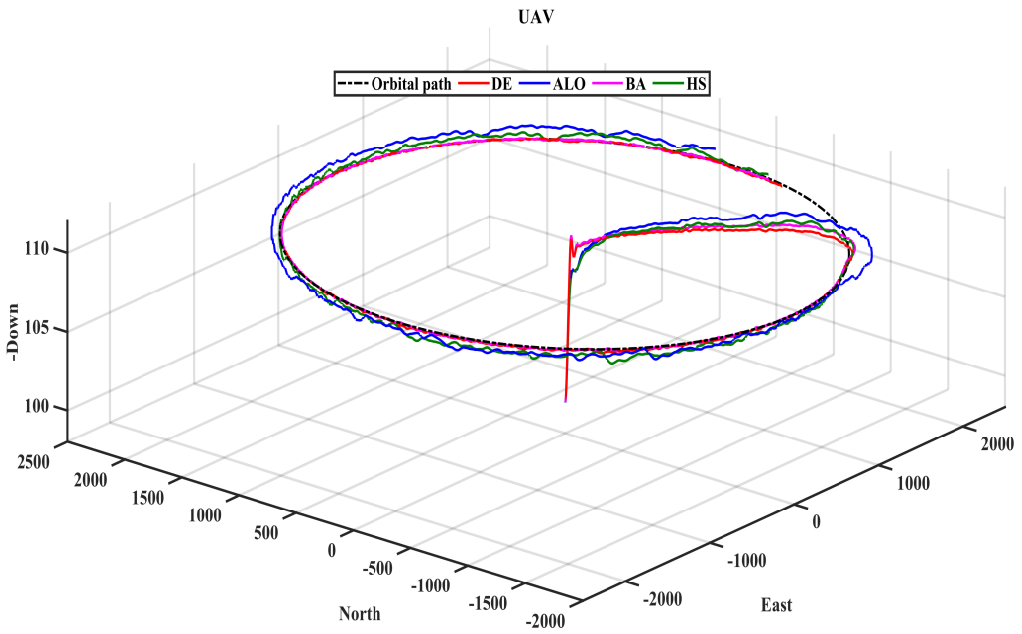


Figure 16: The fixed-wing UAV's position $(pn, pe, z = -pd)$ was obtained by Differential Evolution (DE), Antlion Optimizer (ALO), Bat Algorithm (BA), and Harmony Search (HS) with Disturbance (orbital path follow).

Table 7: The obtained performance ISE using DE, ALO, BA, and HS in the presence of disturbance

output parameter	**performance**	**Performance Indexe ISE**			
		z	ψ	β	V_a
Straigth line	DE	0.6746	0.1321e-03	0.2992e-04	16.5385
	ALO	0.4569	0.2078e-03	0.2612e-04	16.8461
	BA	0.7387	0.1054e-03	0.2670e-04	16.6652
	HS	0.4407	0.1484e-03	0.4429e-04	16.9158
Orbital	DE	0.7085	0.0222	0.0400e-03	1.6171
	ALO	0.4561	0.0312	0.0219e-03	1.8080
	BA	0.6626	0.1234	0.4731e-03	1.6234
	HS	0.4498	0.0199	0.0524e-03	1.8566

lot's ability to effectively adapt to critical situations due to its modular design and enhancement via the intelligent methods. These algorithms enhance system stability and accuracy, ensuring optimal performance across various conditions. In conclusion, the algorithms DE and ALO exhibit suitability for applications requiring high reliability, with each method offering unique advantages depending on the flight parameters considered.

7 Conclusion

This study proposed a novel autopilot architecture for fixed-wing UAVs and optimized it through the application of advanced swarm intelligence and evolutionary algorithms, including Ant Lion Optimizer (ALO), Differential Evolution (DE), Bat Algorithm (BA), and Harmony Search (HS). The primary objective was to achieve efficient stabilization and accurate trajectory tracking under nonlinear constraints and external disturbances while simplifying the control problem through a modular architecture based on successive control loops.

The simulation results prove the remarkable performance of the proposed system. The autopilot successfully controlled critical flight parameters such as airspeed, altitude, heading angle, and sideslip angle, demonstrating optimal performance and reduced stabilization time. The DE and ALO algorithms outperformed the other techniques, achieving significant improvements in key performance metrics across both static and dynamic scenarios. Moreover, simulations conducted under windy conditions demonstrated that the autopilot exhibited robustness and precision, with exceptional disturbance rejection and excellent tracking of both linear and orbital trajectories.

Compared to existing approaches, such as PID-GA controllers and ANFIS-based systems, the proposed autopilot exhibited superior performance, with notable reductions in tracking errors and stabilization times. These findings validate the proposed architecture and highlight the advantages of reduced-parameter control and the significant potential of swarm and developing intelligence methodologies.

Despite these promising results, certain limitations remain. The current evaluation is limited to simulated environments, and further testing in real-world conditions is necessary. Additionally, more complex scenarios such as long-duration missions under extreme climatic conditions should be explored. Integrating deep learning techniques with ALO and DE for real-time adaptation could further enhance the system's capabilities. Testing the architecture in more critical and realistic scenarios would also help strengthen its practical relevance in uncertain and dynamic environments.

References

[1] Abdelmoneim, M., & Osman, A. (2019). Lateral and Longitudinal Controllers Design for a Fixed Wing UAV, *International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 1-6, 2019.

[2] Li, Y., Chen, C., & Chen, W. (2013). Research on Longitudinal Control Algorithm for Flying

Wing UAV Based on LQR Technology, *International Journal on Smart Sensing and Intelligent Systems*, 6(5), 2155–2181, 2013.

[3] Putro, I. E., & Duhri, R. A. (2020). Longitudinal Stability Augmentation Control for Turbo-jet UAV Based on Linear Quadratic Regulator (LQR) Approach, *AIP Conference Proceedings*, 2226(April 2020).

[4] Duan, Z. (2013). Longitudinal Control Law Design and Simulation of Fixed-Wing UAV, *Applied Mechanics and Materials*, 241–244, 1128–1132, 2013.

[5] Chandar, E. A. S. (2022). A Review on Longitudinal Control Law Design for a Small Fixed-Wing UAV, *International Research Journal of Engineering and Technology*, 9(2), 197–202, 2022.

[6] Haridas, V., & Vivek, A. (2016). Longitudinal Guidance of Unmanned Aerial Vehicle Using Integral Sliding Mode Control, *Procedia Technology*, 25(Raerest), 36–43, 2016.

[7] El Gmili, N., El Hamidi, K., Mjahed, M., El Kari, A., & Ayad, H. (2024). Intelligent Sliding Mode Control for Quadrotor Trajectory Tracking Under External Disturbances, *Electrica*, 24(2), 304–317, 2024.

[8] Dahmad, H. A., Ayad, H., Cerezo, A. G., and Mousannif, H. (2024). IT-2 Fuzzy Control and Behavioral Approach Navigation System for Holonomic 4WD/4WS Agricultural Robot, *Int. J. Comput. Commun. Control*, 19(3), 2024.

[9] El Hamidi, K., Mjahed, M., El Kari, A., and Ayad, H. (2018). Neural and Fuzzy-Based Nonlinear Flight Control for an Unmanned Quadcopter, *Int. Rev. Autom. Control*, 11(3), 98–106, 2018.

[10] Korany, M., Youssef, A., and El-Sheikh, G. (2014). Comparative Synthesis of UAV Flight Control in Presence of Actuators and Sensors Dynamics, *Int. Conf. Electr. Eng.*, 9(9), 1–8, 2014.

[11] Sarhan, A., and Ashry, M. (2013). Self-Tuned PID Controller for the Aerosonde UAV Autopilot, *Int. J. Eng. Res. Technol.*, 2(12), 181–2278, 2013.

[12] Mjahed, M. (2019). Flight Control System Design Using Neural Networks, *Int. Robot. Autom. J.*, 5(3), 96–99, 2019.

[13] Mobarez, E. N., Sarhan, A., and Ashry, M. M. (2019). Classical and Intelligent Multivariable Controllers for Aerosonde UAV, *Proc. - 2019 IEEE 9th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2019*, 350–355, 2019.

[14] Mobarez, E. N., Sarhan, A., and Ashry, M. M. (2020). Multi-Variable Controllers for Cooperative Flight of Multi-Fixed Wing UAVs, *2020 12th Int. Conf. Electr. Eng. ICEENG 2020*, pp. 402–409, 2020.

[15] Azegmout, M.; Mjahed, M.; El Kari, A.; Ayad, H. (2023). New Meta-heuristic-Based Approach for Identification and Control of Stable and Unstable Systems, *International Journal of Computers Communications and Control*, 18(4), 5294, 2023.

[16] Oultiligh, A., Ayad, H., El Kari, A., Mjahed, M., and El Gmili, N. (2024). A Novel Improved Elephant Herding Optimization for Path Planning of a Mobile Robot, *International Journal of Electrical and Computer Engineering (IJECE)*, 14(1), 206–217, 2024.

[17] El Gmili, N., Mjahed, M., El Kari, A., and Ayad, H. (2019). Particle Swarm Optimization and Cuckoo Search-Based Approaches for Quadrotor Control and Trajectory Tracking, *Applied Sciences*, 9(8), 2019.

[18] Siti, I., Mjahed, M., Ayad, H., and El Kari, A. (2019). New Trajectory Tracking Approach for a Quadcopter Using Genetic Algorithm and Reference Model Methods, *Appl. Sci.*, 9(9), 2019.

[19] Elbatal, A., Youssef, A. M., and Elkhatib, M. M. (2021). Smart Aerosonde UAV Longitudinal Flight Control System Based on Genetic Algorithm, *Bull. Electr. Eng. Informatics*, 10(5), 2433–2441, 2021.

[20] D'Antuono, V., De Matteis, G., Trotta, D., and Zavoli, A. (2023). Optimization of UAV Robust Control Using Genetic Algorithm, *IEEE Access*, 11(10), 122252–122272, 2023.

[21] Durmaz, M., Cící, K., Sarikaya, M., Bílící, M., and Bílgíç H. H. (2023). Metaheuristic Algorithm-Based Cascade PID Controller Design for Fixed-Wing Unmanned Aerial Vehicle, *Eur. Mech. Sci.*, 7(4), 230–237, 2023.

[22] Lastomo,D., Setiadi,H., and Djalal,M. R.(2017). Optimization pitch angle controller of rocket system using improved differential evolution algorithm, *International Journal of Advances in Intelligent Informatics*,3(1), 27–34, 2017.

[23] Pradhan, R., MajhiS. K., Pradhan, J. K. and Pati, B. B.(2020). Optimal fractional order PID controller design using Ant Lion Optimizer,*Ain Shams Engineering Journal*,11(2), 281–291, 2020

[24] Fayti, M., Mjahed, M., Ayad, H., and El Kari, A. (2022). Differential Evolution Approach for Identification and Control of Stable and Unstable Systems, *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, Istanbul, Turkey, 2022, pp. 218-223.

[25] Fayti, M., Mjahed, M., Ayad, H., and El Kari, A. (2023). Recent Metaheuristic-Based Optimization for System Modeling and PID Controllers Tuning, *Studies in Informatics and Control*, 32(1), 57–67, 2023.

[26] Beard, R. W., and McLain, T. W. (2012). Small Unmanned Aircraft: Theory and Practice, *Princeton University Press*, Princeton.

[27] Mirjalili, S. (2015). The Ant Lion Optimizer. Advances in *Engineering Software*, 83, 80-98.

[28] Storn, R. and Price, K.(1997). Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Kluwer Academic Publishers*, 1997.

[29] Yang, X. and Hossein Gandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization, *Engineering Computations*, 29(5),464-483,2012

[30] Yi, J., Lu, C., and Li, G. (2019). A Literature Review on Latest Developments of Harmony Search and Its Applications to Intelligent Manufacturing, *Mathematical Biosciences and Engineering*, vol. 16(4), 2086–2117, 2019.

[31] Invernizzi, D. , Lovera, M. and Zaccarian, L. (2020). Integral ISS-Based Cascade Stabilization for Vectored-Thrust,*IEEE Control Systems Letters*, vol. 4(1), 43-48,2020

[32] Hosseini, S. M. , Bematol, I. , Ghazi, G. and Botez, R. M. (2024). Enhanced fuzzy-based super-twisting sliding-mode control system for the Cessna Citation X Lateral Motion, *Aerospace*,11(7), 2024.

[33] Qian, L. and Liu, H. H. T. (2022). Robust Control Study for Tethered Payload Transportation Using Multiple Quadrotors, *Journal of Guidance, Control, and Dynamics*, 45(3), 434–452, 2022.

[34] Guo, Y. , Luo, L. and Bao, C. (2022). Design of a Fixed-Wing UAV Controller Combined Fuzzy Adaptive Method and Sliding Mode Control, *Mathematical Problems in Engineering*, Article ID 2812671, 22 pages, 2022.

[35] Elbatal,A. , Elkhatib, M. M. and Youssef, A. M.(2020). Intelligent Autopilot Design Based on Adaptive Neuro -Fuzzy Technique and Genetic Algorithm, *2020 12th International Conference on Electrical Engineering (ICEENG)*,377-382, 2020.

Copyright ©2026 by the authors. Licensee Agora University, Oradea, Romania.
This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.
Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Fayti, M.; Mjahed, M.; Ayad, H.; Elkari, A. (2026). Intelligent Autopilot Design for Fixed-Wing UAVs: Robust Stabilization and Trajectory Tracking Using Advanced Evolutionary and Swarm Optimization Algorithms, *International Journal of Computers Communications & Control*, 21(1), 7197, 2026.

https://doi.org/10.15837/ijccc.2026.1.7197