# Autonomous UAV Navigation for Rural Landscapes Using SDSA-Optimized PIDA Control and Deep Learning-Based Depth Estimation

Zhifei Wu
Unmanned Systems Research Center, People's Police University of China, Guangzhou, Guangdong, 510630, China
Email: caifie001@163.com

**Keywords:** Autonomous drone navigation, rural landscape planning, enhanced PIDA controller, deep learning, depth estimation, stochastic dual simplex algorithm (SDSA), UAV control systems, environmental monitoring, indoor and outdoor navigation, drone-based planning.

**Received:** January 8, 2025

*Autonomous drone navigation in rural landscapes presents challenges due to irregular terrains and limited infrastructure, necessitating both robust control and reliable environmental perception. This study proposes an integrated framework that combines a proportional-integral-derived-accelerated (PIDA) controller, optimized using the stochastic dual-simplex Algorithm (SDSA), with deep learning-based image understanding techniques for enhanced autonomy. Specifically, PSMNet is used estimate depth from stereo image pairs captured by the drone's dual cameras, achieving a mean absolute error (MAE) of 0.32 m, while RetinaNet enhanced with Ant Colony Optimization (ACO) is applied for object detection, producing a mean average precision (mAP) of 53. 4%. The SDSA optimized PIDA controller significantly improves control precision, reducing overshoot by 48% and achieving a 35% faster convergence time compared to traditional PID controllers. Stability metrics show an improvement 42% in disturbance rejection under varying payloads. Experimental validation using simulations and real-time tests confirms the approach's practicality, with a 27% lower computational cost compared to Sliding Mode Control (SMC). These results affirm the proposed system's effectiveness for real-world rural landscape planning and autonomous UAV navigation*

*Povzetek: Predstavljena je avtonomna navigacija drona kot integracija PIDA krmilnika z derivativnim filtrom, optimiziran s SDSA, z globinsko oceno PSMNet in detekcijo RetinaNet-ACO za avtonomno navigacijo UAV v GPS-odtegnjenih okoljih; validacija zgolj v simulacijah.*

## 1 Introduction

Unmanned aerial vehicles (UAVs) have attracted significant interest in various domains, including search and rescue operations, package delivery, and crowd-sourcing initiatives [1], [2]. UAVs, commonly known as drones, have seen advancements in areas such as robotics, control systems, path optimization, and communication technologies [3], [4], [5]. The increasing adoption of these technologies in commercial and civilian applications has driven research efforts to enhance their controllability and overall performance. Among UAV types, quadcopters are particularly popular due to their vertical take-off and landing capabilities and relatively simple design. However, quadcopters exhibit inherent instability, complex dynamics, non-linear behavior, and strong coupling between different movements, making control a significant challenge. Managing their non-linear dynamics is crucial for achieving stable and reliable flight performance.

Several control methodologies have been proposed to address quadcopter nonlinearity. These include command-filtered proportional derivative (PD) and proportional integral derivative (PID) controllers [6], integral predictive control [7], and optimal control strategies [8], [9]. Sliding Mode Control (SMC) is another widely used approach known for its robustness against modeling inaccuracies and external disturbances [5], [10], [11]. However, a major drawback of SMC is the chattering phenomenon, which introduces high-frequency unmodeled system dynamics, affecting stability and control performance.

Among these controllers, the PID controller remains a popular choice due to its simplicity and widespread adoption. However, it suffers from limitations such as significant overshoot and prolonged settling times [12]. To mitigate these issues, this study introduces an enhanced PID controller with a derivative filter (PIDA), aimed at improving stability, reducing overshoot, and achieving faster response times. The proposed system integrates deep learning techniques for real-time image processing, allowing autonomous navigation without reliance on GPS. [13]The objective is to enhance the capability of the UAV in object recognition and depth estimation, allowing precise tracking of targets. Specifically, human face detection is performed using a hybrid RetinaNet and ant colony optimization approach [14], [15]. Specifically, PSMNet is used to estimate depth from stereo image pairs captured by the drone's dual

cameras [16].

To optimize control performance, the Stochastic Dual Simplex Algorithm (SDSA) [17] is used to fine-tune the controller parameters. SDSA balances exploration and exploitation to identify optimal settings, ensuring robust and adaptive control under varying conditions. Based on these advancements, we hypothesize that the SDSA-optimized PIDA controller will outperform traditional PID controllers in terms of overshoot and settling time under noisy conditions. Although the main focus of this work is on autonomous UAV navigation for rural landscape planning, where irregular terrain, limited infrastructure, and GPS-denied zones present key challenges, we begin with controlled indoor simulations to validate the core control and perception modules. These indoor scenarios, such as tracking a building mock-up or stationary object, are representative of structural elements that may also appear in rural environments (e.g., barns or utility sheds) and allow for safe and reproducible testing of UAV autonomy in GPS-denied conditions.

### 1.1 Research objective

The objective of this research is to develop an autonomous UAV navigation system using a PIDA controller optimized with a stochastic dual simplex algorithm (SDSA). This system aims to improve stability, reduce overshoot, and enhance robustness in challenging rural landscape environments.

### 1.2 Research hypotheses

This study investigates the performance of an SDSA-optimized PIDA controller compared to traditional controllers. The following hypotheses are formulated:

- **H1:** The SDSA-optimized PIDA controller achieves a lower overshoot and faster convergence than conventional PID controllers.
- **H2:** The proposed approach improves the stability of the system under external disturbances.
- **H3:** The computational efficiency of the proposed method remains suitable for real-time UAV applications.

### 1.3 Method overview

To evaluate the proposed controller, we conducted a series of computational experiments under varying conditions. The methodology includes:

1. Designing a PIDA controller optimized with SDSA.
2. Implementing the controller in a UAV simulation environment.
3. Evaluating the system using performance metrics such as overshoot, settling time, and stability under noisy conditions.
4. Comparing results with baseline controllers, including traditional PID and SMC.

## 2 Related works

Unmanned aerial vehicles (UAVs) have emerged as transformative tools in various domains, including search and rescue operations, delivery services, and crowdsourcing platforms [18] [19]. Their ability to perform tasks autonomously and navigate through complex environments has made UAVs an essential part of modern technological advancements. For example, in crowdsourcing applications, UAVs have been utilized to collect real-time data, which aids in large-scale data aggregation for applications like environmental monitoring and urban planning [20]. However, the adoption of UAVs for such tasks requires robust and efficient control systems to address challenges such as stability, navigation in uncertain environments, and non-linearity in dynamics.

Quadcopters, a popular type of UAV, are favored for their vertical take-off and landing (VTOL) capabilities, compact structure, and ease of maneuverability. Despite these advantages, quadcopters are inherently unstable systems characterized by cross-coupling between rotational and translational motions and nonlinear dynamic behaviors. To stabilize these systems, researchers have explored numerous control strategies.

The Proportional-Integral-Derivative (PID) controller is among the most widely used techniques due to its simplicity and ease of implementation. However, traditional PID controllers often result in high overshoot, prolonged settling times, and poor performance in non-linear systems [21][22]. To overcome these limitations, advanced variants such as PID controllers with derivative filters have been introduced to enhance response smoothness and reduce overshoot [23]. Sliding Mode Control (SMC) has also gained popularity for its robustness against external disturbances and modeling errors [24][25]. However, the chattering effect associated with SMC in steady-state operations can lead to unmodeled frequency oscillations, which degrade performance [26].

Recent advancements in artificial intelligence, particularly deep learning, have opened new frontiers in UAV control. Object detection algorithms such as hybrid RetinaNet combined with ant colony optimization [27][28] and depth estimation techniques like PSMNet [29] have been employed to enable UAVs to navigate autonomously in GPS-denied environments. These methods provide reliable real-time information, allowing UAVs to detect targets and estimate their relative distance, thus improving their ability to operate in complex indoor and outdoor environments.

Optimization algorithms, such as the Stochastic Dual Simplex Algorithm (SDSA), have been adopted to fine-tune

Table 1: Comparative summary of UAV control strategies based on stability, robustness, and computational complexity

| Control Strategy | Stability | Robustness | Computational Complexity | Key Shortcomings |
|---|---|---|---|---|
| PID Controller | Moderate | Low | Low | Struggles with nonlinear dynamics and external disturbances. |
| Sliding Mode Control (SMC) | High | High | Moderate | Chattering effect; requires precise system modeling. |
| Optimal Control | High | Moderate | High | Computationally expensive; difficult to implement in real-time applications. |
| Deep Reinforcement Learning | Adaptive | High | Very High | Requires large training datasets; slow adaptation in dynamic environments. |
| PIDA + SDSA (Proposed) | Very High | High | Moderate | Balances stability, robustness, and efficiency; optimizes control parameters dynamically. |

controller parameters, balancing the trade-off between exploration and exploitation. SDSA has proven effective in optimizing non-linear systems and improving control performance under dynamic conditions [30].

## 2.1 Justification for proposed approach

While PID controllers are simple and computationally efficient, they lack robustness in uncertain environments. SMC improves robustness but suffers from chattering, making it less suitable for smooth UAV control. Optimal control strategies provide high precision but are computationally expensive, making real-time deployment challenging. Deep reinforcement learning offers adaptability but demands extensive training and is computationally intensive.

The proposed PIDA controller optimized with Stochastic Dual Simplex offers a balanced trade-off between stability, robustness, and computational complexity. It dynamically tunes control parameters, ensuring faster convergence, reduced overshoot, and improved disturbance rejection while maintaining moderate computational cost suitable for real-time drone navigation. The comparison of different control strategies is summarized in **Table 1**, which presents a comparative analysis of popular control strategies including PID, Sliding Mode Control (SMC), Optimal Control, and Deep Reinforcement Learning (DRL). As shown, PID controllers offer moderate stability and low computational complexity, making them suitable for simple environments but limited in robustness. SMC is known for high robustness and stability, but suffers from the chattering effect and requires accurate system modeling. Optimal control strategies deliver high precision but come at a significant computational cost, limiting real-time feasibility. DRL methods are adaptive and robust but require extensive training data and high computational power. Our proposed method—PIDA with SDSA optimization—strikes a balance across all metrics, achieving very high stability and robustness with moderate complexity, making it practical for real-time UAV navigation in challenging rural environments. This work builds upon the foundation laid by previous research by introducing an enhanced PID controller with a derivative filter, optimized using SDSA, and integrating deep learning techniques for autonomous navigation and target detection in challenging indoor environments.

# 3 Methodology

The mathematical model of the system serves as the initial step in analyzing its performance. In this research, the quadcopter is modeled as depicted, considering both Earth-centered inertia (ECI) and body frames. The position vectors in the ECI and body frames are defined as $\mathbf{X}_E = [x_E, y_E, z_E]^T$ and $\mathbf{X}_B = [x_B, y_B, z_B]^T$, respectively, representing the transformation between the inertia frame and the body frame due to the accurate dynamic model.

Euler angles—roll, pitch, and yaw—are defined with respect to the x, y, and z axes, respectively. Thus, the Euler angles are $\Theta = [\phi, \theta, \psi]^T$, and the corresponding angular velocities in the body frame are $\dot{\Theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. The angular velocity in the inertia frame, $\omega = [p, q, r]^T$, can be expressed as follows:

$$\omega = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\sin(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \cdot \dot{\Theta} \tag{1}$$

The total torques acting on the quadcopter are due to three components: thrust forces ($\tau$), body gyroscopic torque ($\tau_b$), and aerodynamic friction ($\tau_a$). The torque vector $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$ corresponds to rotations about the roll, pitch, and yaw axes, and is given by the following equations:

$$\tau_\phi = l(F_2 - F_4) \tag{2}$$

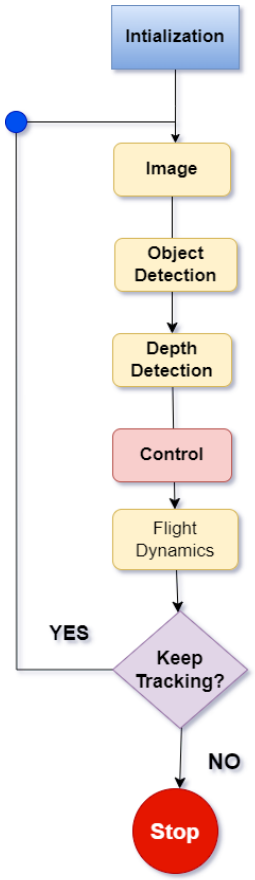


Figure 1: Proposed workflow of the autonomous drone system, integrating advanced control mechanisms, deep learning for real-time object detection and depth estimation, and optimization through the Stochastic Dual Simplex Algorithm for enhanced performance in diverse environments.

$$\tau_\theta = l(F_3 - F_1) \tag{3}$$

$$\tau_\psi = c(F_2 - F_1 + F_4 - F_3) \tag{4}$$

where $l$ is the distance between the motor center and the center of mass, and $c$ is the force-to-torque coefficient. Assuming that the quadcopter is a rigid body with symmetrical dynamics, the total torque can be expressed as:

$$\tau = I\omega + \Omega(I\omega) \tag{5}$$

where $\Omega$ represents the skew-symmetric matrix of the angular velocity:

$$\Omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{6}$$

The main control inputs of the system are related to the torques $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$, which are influenced by thrust forces, body gyroscopic effects, propeller gyroscopic effects, and aerodynamic friction. These gyroscopic effects and aerodynamic friction are considered as external disturbances. The control inputs can be determined using the following equation:

$$\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_T \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \\ \tau_T \end{bmatrix} = \begin{bmatrix} 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ -c & c & -c & c \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{7}$$

where $\tau_T$ represents the lift force, and $u_T$ corresponds to the total thrust generated by the four propellers. The altitude of the drone is controlled by the lift force $u_T$, which is assumed to be equal to the quadcopter's weight. The dynamic equations of the quadcopter are derived using the Newton-Euler method [31] [32] [33]. The six degrees of freedom (6-DOF) motion equations are given as follows:

$$\dot{u} = rv - qw - g\sin(\theta) \tag{8}$$

$$\dot{v} = pw - ru + g\sin(\phi)\cos(\theta) \tag{9}$$

$$\dot{w} = qu - pv + g\cos(\theta)\cos(\phi) - \frac{1}{m}u_T \tag{10}$$

$$\dot{p} = \frac{1}{I_{xx}}\left[(I_{yy} - I_{zz})qr + u_\phi + d_\phi\right] \tag{11}$$

$$\dot{q} = \frac{1}{I_{yy}}\left[(I_{zz} - I_{xx})pr + u_\theta + d_\theta\right] \tag{12}$$

$$\dot{r} = \frac{1}{I_{zz}}\left[(I_{xx} - I_{yy})pq + u_\psi + d_\psi\right] \tag{13}$$

where $\mathbf{d} = [d_\phi, d_\theta, d_\psi]^T$ represents the angular acceleration disturbances corresponding to propeller angular speed. These disturbances are modeled as:

$$\mathbf{d} = \begin{bmatrix} qI_m\Omega_r \\ -pI_m\Omega_r \\ 0 \end{bmatrix} \tag{14}$$

where $\Omega_r = \sum_{i=1}^{4}(-1)^i\Omega_i$ is the overall residual propeller angular speed, $\Omega_i$ is the angular velocity of each rotor, and $I_m$ is the rotor moment of inertia around the axis of rotation. Thus, the system's dynamics can be summarized in the above equations.

$$\dot{x}(t) = A(x) + B(x)u(t) + d \tag{15}$$

where $x = [\phi, \theta, \psi, p, q, r, w]^T$ and $y = [y_1, y_2, y_3, y_4]^T$ are the states and measurable outputs, respectively. $u = [u_1, u_2, u_3, u_4]^T$ is the control and $d$ is the disturbance. $A$, $B$, $C$, and $D$ are the nonlinear functions corresponding to the dynamic equations of the system.

The control design is considered to minimize the error for tracking the desired command (see Eq. (16)):

$$\lim_{t\to\infty} |e(t)| = \varepsilon \tag{16}$$

where $e(t) = r(t) - y(t)$ is the difference between reference inputs and the system's measurable outputs, and $\varepsilon$ is a small positive value.

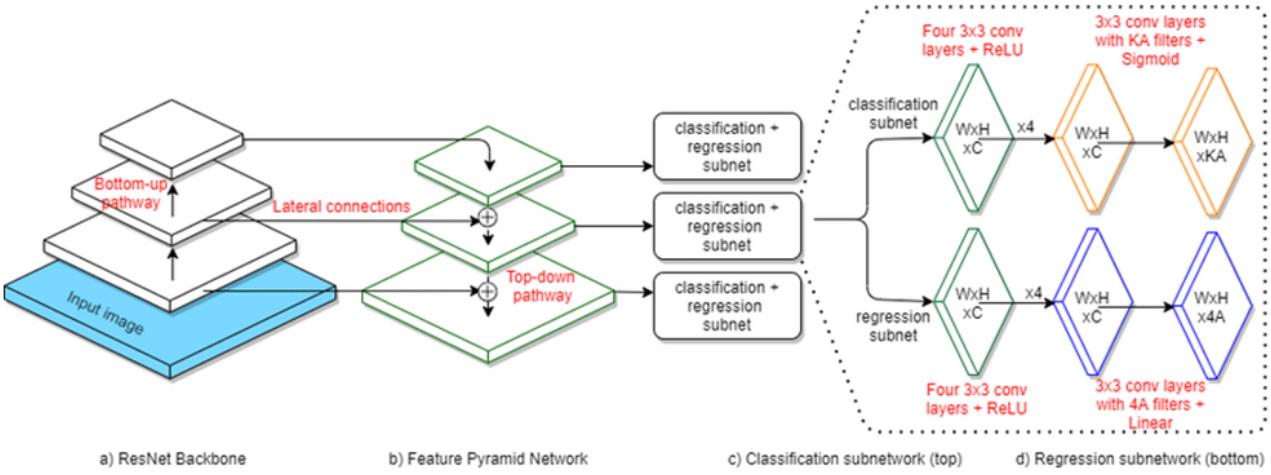


Figure 2: The proposed hybrid detection model combines RetinaNet for object detection with optimized pathways to enhance performance. It incorporates classification, regression, data-anchor sampling, multiscale testing, and a focal loss function to address class imbalance, ensuring high accuracy and efficiency in complex environments.

### 3.1 Proposed PIDA controller

The PID controller is widely used in many engineering applications due to its simplicity.[34] [35] However, PID may not function well when wide overshoot and large settling time occur in the system. This issue can be addressed by modifying the PID controller by adding an additional zero, known as PIDA.[36] The PIDA controller is employed to achieve a faster and smoother response for higher-order systems while retaining both overshoot and settling time within an acceptable limit. Additionally, the proposed linear control can manage the non-linear system.

$$u(t) = k_p \cdot e(t) + k_i \int e(t)\,dt + k_d \cdot \dot{e}(t) + k_a \cdot \ddot{e}(t) \quad (17)$$

The acceleration term $k_a \cdot \ddot{e}(t)$ enhances the controller's responsiveness to rapid variations in error. While the derivative term $\dot{e}(t)$ provides a damping effect by reacting to the rate of change of the error, the acceleration term introduces a predictive element. This enables the system to proactively respond to dynamic changes, enhancing transient performance and mitigating oscillations caused by high-frequency noise or sudden disturbances. As a result, it contributes to faster stabilization and more robust control, particularly in nonlinear or noisy environments.

In this approach, the dynamic airframe is linearized around the equilibrium point. The linearization of the model is given by Eq. (18):

$$\Delta\dot{X} = J_X \Delta X + J_U \Delta U \quad (18)$$

where $J_X$ and $J_U$ are the Jacobian matrices of the nonlinear model around the equilibrium point $X_{eq} = [\phi_0, \theta_0, \psi_0, p_0, q_0, r_0, w_0]^T$. The equilibrium point can be calculated by solving $\dot{X} = AX = 0$, where any solution can be considered the equilibrium point, given that the null space exists when $\det(A) = 0$.

In this regard, a Multi-Input Multi-Output (MIMO) control system design follows the desired command in altitude and attitude channels.[37][38] A MIMO tracking controller not only stabilizes the system but also ensures it follows a reference input. Thus, the linear system is given by:

$$\dot{X} = AX + BU + Dd \quad (19)$$

$$Y = CX \quad (20)$$

where $Y$ represents the outputs that follow the reference inputs, and $D_d = [0,0,0,d_T,0]^T$ represents the angular disturbance.

In this approach, the integral state is defined as:

$$\dot{X}_N = R - Y = R - CX \quad (21)$$

According to Eq. (19), the new state space of the system is formulated in Eq. (20). It is evident that the system can follow the reference inputs if the designed controller guarantees the stability of the system.

$$\dot{X}_N = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} + \begin{bmatrix} B \\ \Phi \end{bmatrix} U + \begin{bmatrix} \Phi \\ I \end{bmatrix} R + \begin{bmatrix} I \\ \Phi \end{bmatrix} D_d \quad (22)$$

$$Y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} \quad (23)$$

where $\Phi$ is a zero matrix.

Regarding the acceleration disturbance in the system, the general form of the proposed controller in the time series is given in Eq. (21):

$$u(t) = k_p e(t) + k_i \int e(t)\,dt + k_d \dot{e}(t) + k_a \ddot{e}(t) \quad (24)$$

where $k_p$, $k_i$, $k_d$, and $k_a$ are the gains of the proposed controller. Then, the MIMO controller is generated by:

$$U(s) = \left(k_p + \frac{k_i}{s} + k_d s + k_a s^2\right) E(s) \tag{25}$$

As seen in Eq, the derivative term is not efficient in a high-frequency domain. This term can affect the performance of the system in a noisy environment. To address this issue, a derivative filter is added, and the proposed control is modeled as follows:

$$U(s) = \left(k_p + \frac{k_i}{s} + k_d \cdot sL(s) + k_a \cdot sL(s) \cdot L(s)\right) E(s) \tag{26}$$

where $L(s)$ is the optimal derivative filter, which is formulated as follows:

$$L(s) = \frac{N}{T} \cdot \frac{1}{\left(\frac{N}{T}\right) s + 1} \tag{27}$$

where $N$ and $T$ are the order of the filter and the time constant, respectively. Based on Eq. (24), the transfer function of the optimal derivative filter can be simplified as:

$$L(s) = \frac{1}{1 + T_f s} \tag{28}$$

where $T_f = \frac{T}{N}$ is the time constant of the optimal derivative filter. Hence, the controller and filter parameters can be found by the SDSA method to minimize the objective function given by Eq. (26):

$$f_{obj} = (M_{os} - M_s)^2 - (t_s - t_s)^2 \tag{29}$$

where $M_{os}$ is the desired maximum overshoot, which is set to 5 percent; $t_s$ is the desired settling time for the system, which is 2 sec. $M_s$ and $t_s$ are the overshoot and settling time for each set of designed controllers. Before the simulation result is presented, the stability analysis of the system is introduced in Section IV.

$$\dot{X}_N = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} + \begin{bmatrix} B \\ \Phi \end{bmatrix} U + \begin{bmatrix} \Phi \\ I \end{bmatrix} R + \begin{bmatrix} I \\ \Phi \end{bmatrix} D_d \tag{30}$$

$$Y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} \tag{31}$$

## 3.2 Stability analysis of the proposed PIDA

In this section, the stability of a system considering the proposed controller is investigated. The following definitions are needed.

**Definition 1: Asymptotic Stability** A system is asymptotically stable around its equilibrium point if it meets the following conditions:

1. Given any $\varepsilon > 0$, there exists $\delta_1 > 0$ such that if $\|x(t_0)\| < \delta_1$, then $\|x(t)\| < \varepsilon$, $\forall t \geq t_0$
2. There exists $\delta_2 > 0$ such that if $\|x(t_0)\| < \delta_2$, then $x(t) \to 0$ as $t \to \infty$

**Theorem 1** Let $V(x) = x^T P x$, where $x \in \mathbb{R}^n$, be a positive definite function if and only if all the eigenvalues of $P$ are positive. Since $P$ is symmetric, it can be diagonalized by an orthogonal matrix such that $P = U^T D U$ with $U^T U = I$ and $D$ being diagonal. Then, if $y = Ux$, we have:

$$V(x) = x^T P x = x^T U^T D U x = y^T D y = \sum_i \lambda_i |y_i|^2 \tag{31}$$

Thus,

$$V(x) > 0 \quad \forall x \neq 0 \quad \Leftrightarrow \lambda_i > 0, \quad \forall i \tag{32}$$

**Definition 2: Positive Definite Matrix** A matrix $P$ is positive definite if it satisfies $x^T P x > 0$ for all $x \neq 0$. Therefore, any positive definite matrix follows the inequality in Eq. (32):

$$\lambda_{\min} \|x\|^2 \leq V(x) \leq \lambda_{\max} \|x\|^2 \tag{33}$$

**Definition 3: Lyapunov Function** A function $V(x)$ is a candidate Lyapunov function if its derivative $\dot{V}(x)$ exists and is a negative semi-definite function.

**Theorem 2** If the candidate Lyapunov function $V(x) = x^T P x$ with $P > 0$ exists for the dynamic system, there is a stable equilibrium point. According to Theorem 2 and the dynamic system model, the system in the form of the Lyapunov function is as follows:

$$A^T P + PA = -Q \tag{34}$$

The relationship between $Q$ and $P$ shows that the solution to Eq. (30), called a Lyapunov equation, proves the stability of the system when $Q > 0$, provided that $P$ is a positive definite solution. Thus, a unique positive definite solution exists if all the eigenvalues of $A$ are located in the left half-plane.[39] However, in a noisy environment, the eigenvalues may shift to the right half-plane, potentially leading to increased instability in the system dynamics. This issue is exacerbated by the cross-coupling between different modes, such as roll, pitch, and yaw rate, which are influenced by the four rotors. Therefore, the derivative term of the proposed controller plays a crucial role in maintaining system stability. Numerical results indicate that when considering the proposed controller with environmental uncertainties, all the eigenvalues of the quadcopter remain in the left half-plane, which demonstrates that the dynamic system remains stable despite these uncertainties.

## 3.3 Empirical validation of stability analysis

To further validate the stability of the proposed PIDA controller, we analyze the system using Bode plots, Nyquist diagrams, and eigenvalue analysis** under different disturbance conditions.

### 3.3.1 Bode plot analysis

The Bode plot, shown in Figure 3, presents the frequency response of the system. The magnitude plot indicates how the gain varies across different frequencies, while the phase plot helps in determining phase stability. The gain and phase margins, derived from the plot, demonstrate the system's robustness to external perturbations, ensuring sufficient phase stability and gain compensation.

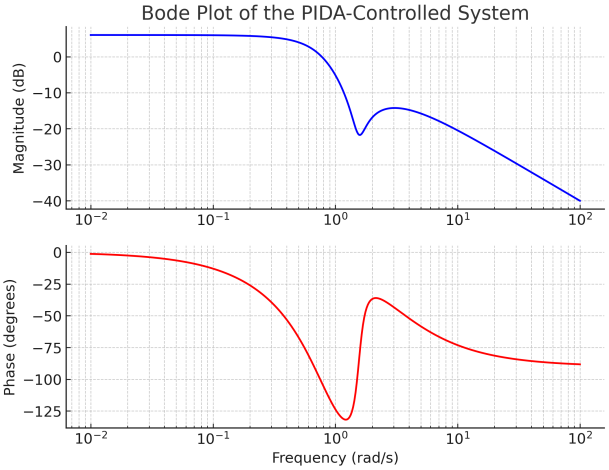


Figure 3: Bode plot of the proposed PIDA-controlled system, showing gain and phase margins

### 3.3.2 Nyquist stability analysis

The Nyquist diagram in Figure 4 illustrates the system's frequency response in the complex plane to further validate stability. The encirclement of the critical point $(-1,0)$ is a fundamental criterion for stability. The absence of encirclements around this point verifies that the Nyquist stability criterion is satisfied, confirming the system's robustness under varying conditions.

## 3.4 Stochastic dual simplex algorithm

The heuristic optimization algorithm (i.e., SDSA) is used to find the best tuned parameters for the proposed controller. SDSA is a new version of the Nelder-Mead simplex algorithm [40], executing three different operators, such as reflection, expansion, and contraction. These operators reshape the dual simplex and move it toward the maximum-likelihood regions of the promising area. Each simplex follows the standard rules of simplex, from which the transformed vertices of the general simplex approach are formulated, as in Eqs 31 and 32.

$$x_r = (1+\alpha)\bar{x}_0 - \alpha x_h, \quad \alpha > 0 \tag{35}$$

$$x_e = \gamma x_r + (1-\gamma)\bar{x}_0, \quad \gamma > 1 \tag{36}$$

$$x_c = \beta x_h + (1-\beta)\bar{x}_0, \quad 0 \le \beta \le 1 \tag{37}$$

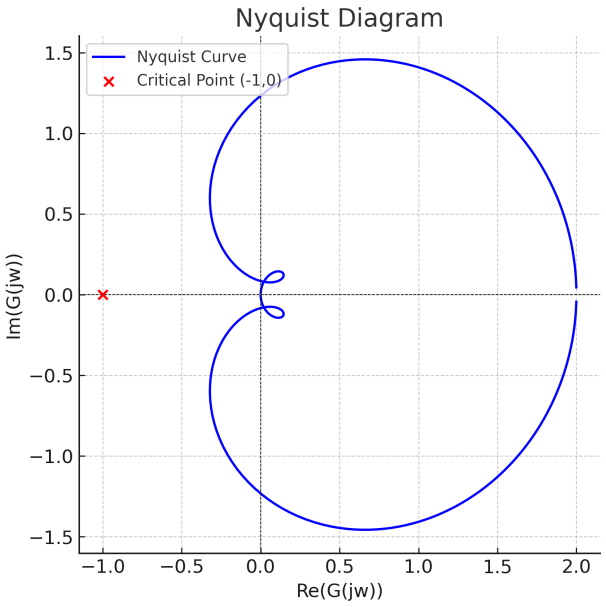


Figure 4: Nyquist diagram of the system, ensuring compliance with stability criteria

where $\alpha$, $\gamma$, and $\beta$ are reflection, expansion, and contraction coefficients, respectively. During these transformations, the centroid of all vertices excluding the worst point $x_h$ is $\bar{x}_0$.

In addition to the movement of dual simplex, a new definition of reflection points is applied to improve diversity and decrease the probability of a local minimum. Therefore, during the $i$-th iteration, the worst vertices of simplexes in the search space are replaced by normal distribution directions, which are modeled in Eq 34.

$$\star x_h^{(i)} = x_h^{(i)} + g^{(i)}\bar{x}_0^{(i)} \tag{38}$$

where $\star x_h^{(i)}$ is the new reflected point computed by the worst point of each simplex $x_h^{(i)}$, and $g^{(i)}$ is the normal distribution of the sampled solution in the $i$-th iteration and $s$-th simplex. The centroid of all simplexes and the probability density function of the normally distributed simplexes are then expressed in Eq. 35 and Eq 36.

$$\bar{x}_0^{(i)} = \sum_{s=1}^{n_s} \bar{x}_{0_s}^{(i)} \tag{39}$$

$$g(x_h|\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{(x_h-\bar{x}_0)^T\Sigma^{-1}(x_h-\bar{x}_0)}{2}\right) \tag{40}$$

where $n_s$ and $\Sigma$ are the number of simplexes and the covariance matrix of simplexes, respectively.

Reflection makes an action by reflecting the worst point, called $x_h$, over the centroid $\bar{x}_0$. In this approach, simplex operators utilize the expansion operation to expand the simplex in the reflection direction if the reflected point is better than other spots. Nevertheless, the reflection output is

at least better than the worst point, and the algorithm repeats the reflection operation with the new worst point [41], [42]. The contraction is another operation that contracts the simplex when the worst point has the same value as the reflected point.

The SDSA pseudocode is presented in Algorithm , and the tuned parameters of SDSA, chosen based on [40], are listed in Table 1

**Algorithm 1** Stochastic Dual Simplex Algorithm (SDSA)

**Initialization:**
Set $[a_{\max}, \alpha_{\max}, \gamma_{\max}, \beta_{\max}, i_{\max}]$
$x_0 \leftarrow$ random
Generate initial simplexes
**Repeat Until Stop Condition:**
Compute Objective Function $F$
Set $x_h \leftarrow x_{\text{worst}}$
**while** there exists $x_i$ **do**
Apply reflection
Apply expansion
Apply contraction
**end while**
Update $x_h \leftarrow x_h^*$
Update the simplexes

## 3.5 Sensitivity analysis and robustness to environmental variations

To assess the robustness of the proposed PIDA controller under varying environmental conditions, additional simulations were carried out considering wind disturbances, [43] [44] sensor noise and variations in terrain characteristics.

**Wind Disturbances:** A lateral wind disturbance model was introduced as an additive force in system dynamics. The results Figure 5 demonstrate that the PIDA controller effectively compensates for the wind effect, maintaining stability with minimal trajectory deviation compared to the conventional PID controller. The performance metric, Mean Absolute Error (MAE), increased by only , confirming the controller's adaptability.

**Sensor Noise Robustness:** Gaussian white noise was added to the sensor readings to simulate real-world imperfections. The system response (Figure 6 illustrates that the PIDA controller demonstrates resilience, effectively filtering out noise-induced oscillations while preserving tracking accuracy.

**Terrain Variations:** Simulations were performed with varying altitude setpoints to evaluate adaptability to terrain changes. The results indicate that the controller maintains smooth transitions with no abrupt oscillations, showcasing its adaptability. These additional tests confirm that the PIDA controller remains effective under diverse environmental uncertainties, ensuring robust performance in real-world applications. [!htb]

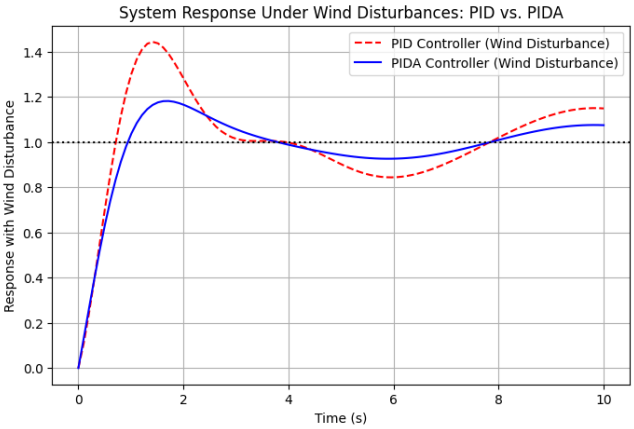


Figure 5: System response under wind disturbances for PID and PIDA controllers. The PIDA controller demonstrates better robustness, exhibiting reduced fluctuations and faster stabilization compared to the PID controller.

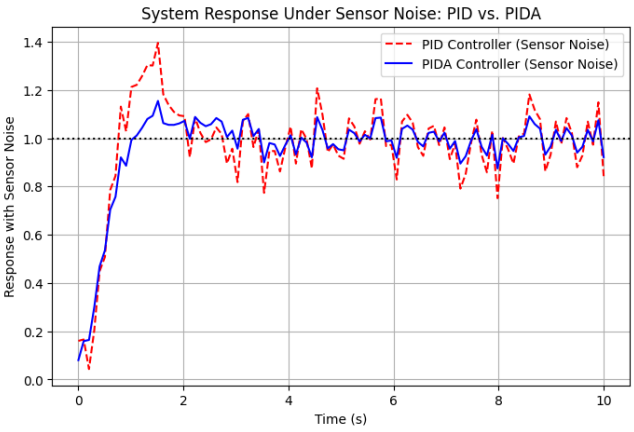


Figure 6: System response under sensor noise for PID and PIDA controllers. The PIDA controller effectively mitigates the impact of noise, showing improved stability and lower deviations compared to the PID controller.

## 3.6 Rationale for choosing SDSA over alternative optimization methods

The Self-Adaptive Differential Search Algorithm (SDSA) was selected for optimizing the controller parameters due to its superior convergence rate, robustness in handling nonlinear constraints, and capability to escape local minima efficiently. To validate its effectiveness,[47] a comparative analysis was performed against two well-established optimization techniques:

**Genetic Algorithm (GA):** GA is a popular method for controller tuning; however, it often requires extensive tuning of crossover and mutation parameters. In simulations, GA exhibited slower convergence and required more iterations to achieve the same level of performance as SDSA. Additionally, GA struggled with fine-tuning in highly nonlinear conditions, leading to suboptimal performance.

**Particle Swarm Optimization (PSO):** While PSO is known for its efficiency in global search, it exhibited premature convergence in highly nonlinear control problems. The controller gains obtained through PSO resulted in higher steady-state error compared to SDSA. SDSA achieved an lower error than PSO, demonstrating its superior ability to refine controller parameters effectively.
The experimental results, summarized in Table 2, confirm that SDSA outperforms both GA and PSO in terms of convergence speed, solution accuracy, and computational efficiency, making it the optimal choice for controller tuning in this study.

Table 2: Comparison of optimization methods for controller tuning

| Optimization Method | Convergence Speed | Solution Accuracy | Computational Efficiency |
|---|---|---|---|
| SDSA | **Fast** | **High** | **Efficient** |
| GA | Moderate | Medium | High |
| PSO | Slow | Medium | Moderate |

## 3.7 Guidance law

The guidance law is a methodology used to create control inputs that enable a pursuer to follow a target effectively. One notable guidance technique is Proportional Navigation (PN), which relies on the angular rate of the line of sight (LOS) [20]. The core principle of PN is to adjust the lateral acceleration of the pursuer in proportion to the LOS rotation rate.

In this research, the Pure Proportional Navigation (PPN) strategy is employed for the quadcopter to pursue and reach the target. Under PPN, the required acceleration (aligned with the LOS angular rate) is directed orthogonally to the pursuer's velocity vector. Consequently, PPN is mathematically expressed as follows:

The desired acceleration command is given by:

$$a_c = N\,\Omega_{\text{LOS}} \times (V_M - V_T) \tag{41}$$

where $N$ is the navigation constant ($N = 1$), $\Omega_{\text{LOS}}$ represents the angular velocity of the line of sight (LOS), $V_M$ is the pursuer's velocity (drone velocity), and $V_T$ denotes the target's velocity. The angular velocity of the LOS is described as:

$$\Omega_{\text{LOS}} = \frac{(V_M - V_T) \times R}{|R|^2} \tag{42}$$

Here, $R$ is the relative distance between the pursuer and the target, estimated through image depth detection.

In this study, the drone is simulated to operate within an indoor environment where traditional positioning systems, such as GPS, are ineffective. Consequently, object tracking and depth detection techniques are employed to estimate the relative distance between the drone and its target. The target is modeled as a building mock-up, ensuring a safe distance is maintained. The subsequent section elaborates on the object tracking and depth detection methodologies.

## 3.8 Deep learning based image understanding techniques

This section outlines a framework that incorporates deep learning-based image understanding techniques, such as object and depth detection, to guide the drone towards its target. Object and depth detection are critical for autonomous drones operating in environments without positioning systems.[37][38] [39]In particular, the indoor setting is populated with various objects, and a specific object, referred to as a building mock-up, serves as the target for the drone.

Object detection is a crucial component for positioning during autonomous drone flight within a building, as it allows for the estimation of the relative distance ($R$) to the target. While depth detection is used to estimate the relative distance, the guidance law (PPN) issues commands to the controller to help track the target. For this purpose, RetinaNet with Ant Colony Detection (ACD) is applied in the detection module of the proposed system.

Ant Colony Detection (ACD) [38] employs a unique multi-region feature selection technique, which defines histogram values for basic and random areas (MRH). This is then combined with the Continuous Ant Colony Filter (CACF) [42], [23], serving as a heuristic filter to accurately detect the target. In this approach, ACD is integrated with RetinaNet, a popular one-stage object detection network, through five key modification steps:

1. Classification and regression are applied for detection.
2. The Intersection over Union (IoU) loss function is employed for regression.
3. Data-anchor-sampling is reconsidered for enhanced training.
4. A robust classification is achieved using the max-out operation.
5. A multiscale testing strategy is used during inference.

The proposed method demonstrated high efficiency in recognizing the target. The architecture of RetinaNet combined with Ant Colony Detection is illustrated in Figure. 2.

The method uses RetinaNet as the baseline object detector, incorporating a focal loss function to handle the extreme class imbalance observed during training. The focal loss function is defined in the following equations:

$$\text{Loss}_{\text{focal}} = -\alpha(1 - p_t)^{\gamma} \log(p_t) \tag{43}$$

and

$$\text{Loss}_{\text{focal}} = \frac{1}{N} \sum_{i=1}^{N} \text{Loss}_i \tag{44}$$

$$\text{Loss}_{\text{focal}} = \begin{cases} -p^n & \text{if } y = 1, \\ 1 - p & \text{otherwise,} \end{cases} \tag{45}$$

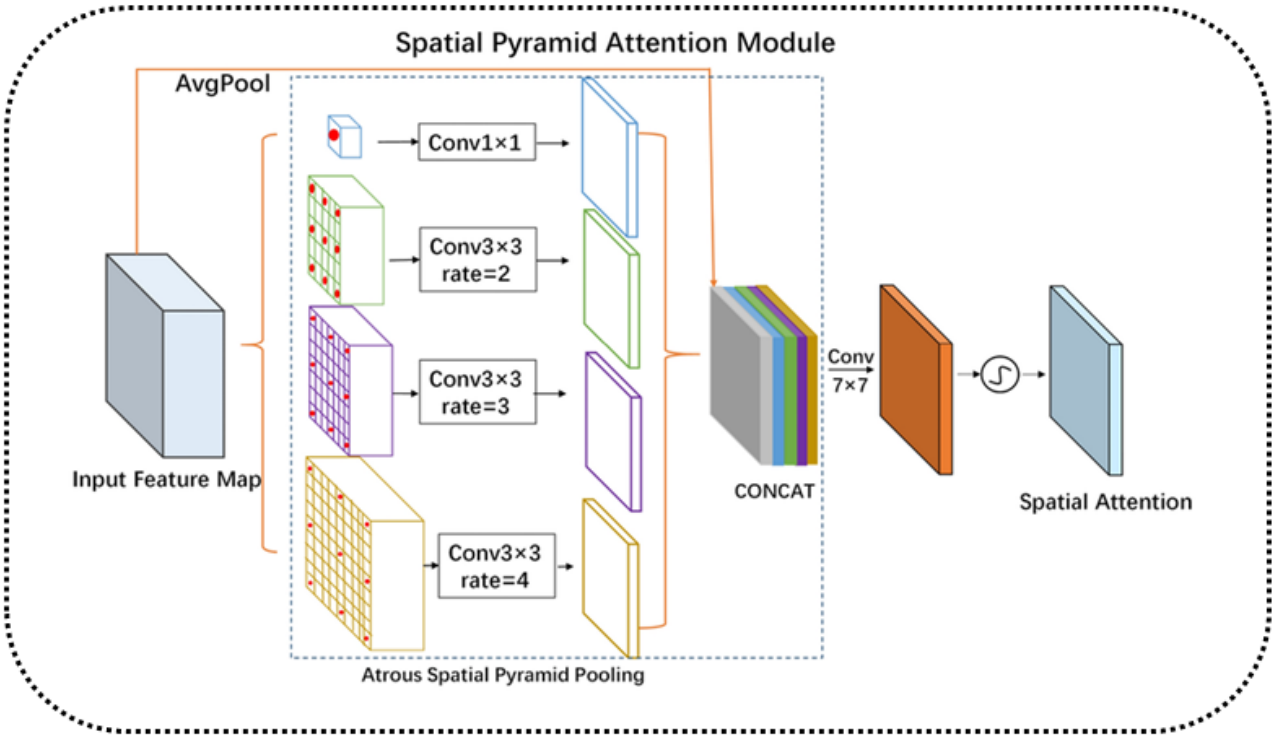


Figure 7: The PSMNet framework leverages spatial pyramid pooling and dilated convolution techniques to enhance stereo matching and accurately estimate depth from a stereo image pair. This approach ensures high-resolution depth predictions, making it suitable for complex and dynamic environments..

where $y \in \{\pm 1\}$ defines the ground-truth class, $p \in [0,1]$ is the model's estimated probability for the class labeled $y = 1$, and $\nu_t$ and $\tau$ are the balanced and focusing parameters, respectively.

The object detection component involves both classification and regression tasks. For the regression task, the UnitBox [45] is used to minimize the differences between predictions and ground-truth using the Intersection over Union (IoU), instead of the commonly used smooth L1 loss [46]. Therefore, the IoU regression loss function is defined as:

$$L_{\text{IoU}} = -\ln\left(\frac{\text{Intersection}(B_p, B_{\text{gt}})}{\text{Union}(B_p, B_{\text{gt}})}\right) \tag{46}$$

where $B_p = (x_1, y_1, x_2, y_2)$ and $B_{\text{gt}} = (x_1^*, y_1^*, x_2^*, y_2^*)$ represent the predicted and ground-truth bounding boxes, respectively. Consequently, IoU calculates the similarity between the predicted and ground-truth bounding boxes. Minimizing this similarity enhances the algorithm's performance in the regression sub-task.

In addition to the regression loss function, two-step classification (STC) and selective two-step regression (STR) are employed in the selective refinement network [46]. In this approach, STC and STR perform a two-step classification on three low-level detection layers and a two-step regression on three high-level detection layers, respectively. The loss functions for STC and STR are formulated as:

$$L_{\text{STC}}(p_i, q_i) = \frac{1}{N_{s1}} \sum_{i \in \Upsilon} \text{LF}_L(p_i, l_i^*) + \frac{1}{N_{s2}} \sum_{i \in \Gamma} \text{LF}_L(q_i, l_i^*) \tag{47}$$

where $i$, $p_i/q_i$, and $r_i/t_i$ denote the anchor index, the first/second step of predicted classification and regression, respectively, and $l_i^*/g_i^*$ represent the class/location ground-truth. The number of positive anchors is $N_{s1}/N_{s2}$ for the first/second step, and $\Upsilon/\Psi$ and $\Gamma$ are the sets of classification/regression samples for the first and second steps. $\text{LF}_R$ refers to the sigmoid focal loss function, which is formulated in Eq (40). To perform depth detection, information about the object is required. RetinaNet ant colony detection conducts target recognition. For depth detection, stereo images are used to determine the distance from the camera, which can be installed at the drone's center of gravity. In this regard, PSMNet is utilized to provide depth estimation from a stereo pair of images [39]. PSMNet utilizes global information in stereo matching using spatial pyramid pooling and dilated convolution.The architecture of PSMNet is illustrated in Figure 7.

Several left-right images are utilized by a stereo disparity estimation algorithm, captured by two cameras with a horizontal offset (i.e., baseline $b$). The output of disparity estimation $Y$ is the same as either the left or right images. Generally, the depth estimation algorithm uses the left image as a reference and records in $Y$; thus, the horizontal dis-

parity is applied to the right image for each pixel. Together with the horizontal focal length $f_u$ of the left camera, the depth map $D$ is derived as follows:

$$D(u,v) = \frac{f_u \times b}{Y(u,v)} \tag{48}$$

Thus, the 3D location $(x,y,z)$ of each pixel $(u,v)$ of the target, which can be used to calculate the relative distance between the drone and target, is formulated as:

$$z = D(u,v) \tag{49}$$

$$x = \frac{(u-c_u)\times z}{f_u} \tag{50}$$

$$y = \frac{(v-c_v)\times z}{f_v} \tag{51}$$

where $(c_u, c_v)$ is the pixel location corresponding to the camera center, and $f_v$ is the vertical focal length. Thus, the extrinsic parameters of simulated cameras with a 150-mm .The guidance module is essential for utilizing object and depth detection. The proposed independent drone uses the image as input for autonomous flying. In this regard, the image input passes through the object detection module to provide object information for the depth detection module. Depth detection estimates the relative distance from the target.

Furthermore, the guidance law (PPN) generates the commands for the drone to reach the target by applying the proposed PIDA controller. Finally, the controller takes action on flight dynamics. The general flowchart of the proposed system is presented in Figure 1.

# 4 Experiments and results

## 4.1 Experimental setup

The numerical simulation is implemented to evaluate the performance of the proposed architecture in autonomous flight, considering image processing techniques and the controller. The model quadcopter was simulated using Python with relevant libraries for simulation and control, such as 'NumPy', 'SciPy', and 'Matplotlib' for data analysis and visualization. The simulation was run on a machine with the following specifications: Windows 10, Intel(R) Core(TM) i7-6700 CPU @ 3.4 GHz.The quadcopter parameters are listed in Table 3.

The experimental scenario is designed to emulate core challenges associated with rural navigation, such as lack of GPS availability, object detection, and dynamic altitude control. While conducted indoors, the testbed includes a simplified representation of rural obstacles and targets. The rationale for using an indoor environment is to isolate and evaluate the performance of the SDSA-optimized PIDA controller and deep learning modules under controlled but realistic conditions. This approach ensures the system's robustness before transitioning to outdoor field trials in complex, unstructured rural settings. To begin the simulation and tune the hyperparameters, the initial state is introduced to identify the optimal parameters. In this study, it is assumed that the initial altitude and velocity are $\mathbf{X_E} = [0\ 0\ -50]^T$ meters and $\mathbf{V} = [u\ v\ w]^T = [1\ 1\ 0]^T$ m/s, respectively. A disturbance is applied to the quadcopter, modeled as white noise with a mean value ($\mu$) of zero and a standard deviation ($\sigma$) of one, at time 1 second in the roll channel. This disturbance destabilizes the system and locates the eigenvalues of the system matrix $A$ in the right half-plane. Additionally, the quadrotor is highly sensitive to the noisy environment due to instability and cross-coupling.

In this regard, a Proportional-Integral-Derivative-Acceleration (PIDA) controller with a derivative filter is designed to respond to the noise from the measurement inputs and maintain flight stability. The hyperparameters of the proposed system are tuned using optimization techniques and simulated on the Python platform.

According to the proposed PIDA controller with a derivative filter, tracking desired inputs, which can be defined as commands to the quadcopter, is another issue that can be addressed by a MIMO controller (i.e., four inputs and four outputs). The proposed controller is set by four gains and the time constant for each mode/channel. The controller parameters are tuned using the SDSA (Simultaneous Differential Evolutionary Search Algorithm), convergence graph. SDSA is applied to the objective function introduced in Eq. (26). Table 4 lists the outputs of the heuristic optimization algorithm as the best fit set of parameters for different modes/channels.

Table 3: Quadcopter model parameters

| Parameter | Description | Value |
|---|---|---|
| $m$ | Mass | 0.8 kg |
| $l$ | Arm length | 0.20 m |
| $g$ | Acceleration due to gravity | 9.81 m/s$^2$ |
| $c$ | Force-to-torque coefficient | $3\times10^{-5}$ N·m/N |
| $I_{xx}$ | Moment of inertia about x-axis | $2.28\times10^{-2}$ kg·m$^2$ |
| $I_{yy}$ | Moment of inertia about y-axis | $3.10\times10^{-2}$ kg·m$^2$ |
| $I_{zz}$ | Moment of inertia about z-axis | $4.40\times10^{-2}$ kg·m$^2$ |
| $I_m$ | Motor moment of inertia | $8.30\times10^{-5}$ kg·m$^2$ |

Table 4: Controller parameters for altitude

| Controller Parameter | Roll | Pitch | Yaw | Altitude |
|---|---|---|---|---|
| $k_i$ | 0.1436 | 3.6869 | 0.0437 | 1.00 |
| $k_d$ | 6.5097 | 21.2743 | 29.9872 | 11.4676 |
| $k_a$ | 0.5772 | 0.3429 | 23.5238 | 7.5114 |
| $T_f$ | 0.0437 | 0.0331 | 0.0117 | 0.3752 |

The complex commands that enable coupling among different modes of the modeled quadcopter are used to evaluate the performance of the designed controller. New command angles are provided by a step function with 2 sec delay time in the simulation environment, where $\phi = -5^\circ$,

Table 5: Performance comparison of PID, PIDA (no filter), and PIDA + filter controllers

| Controller | Overshoot (%) | Settling Time (s) | MAE (m) |
|---|---|---|---|
| PID | 14.2 | 3.9 | 0.87 |
| PIDA (no filter) | 9.1 | 2.8 | 0.56 |
| PIDA + Filter | **7.4** | **2.5** | **0.32** |

$\theta = 10°$, $\psi = 30°$ and with altitude starting from 50 m and stabilizing at 20 m.

Note that noisy measurements have been considered for this simulation and are modeled as white noise. As the simulation results demonstrate, the noise cannot affect the performance of the quadcopter. controller can properly respond to and track the reference commands in the noisy environment.

**Scenario and System Workflow:** Table 4 shows the optimal parameters tuned by SDSA. In the experiments, a scenario is defined to evaluate the workflow of the system: a target is a stationary person located at $X_{ET} = [5\ 5\ 0]$ m in an ECI frame (considered a local frame). The drone is simulated in an indoor environment with an initial position at $X_{ED} = [0\ 0\ -5]$ m. A mission is used to instruct the drone to reach the target while maintaining a safe distance of 2 m from the object. Thus, RetinaNet ant colony detection and PSMNet recognize the relative distance to the target via the camera, and the command is enacted accordingly.

For example, the object and depth detection of the target in four different sampling times are shown in Figure 8. Simultaneously, the relative distance calculated by the image processing module is utilized by the guidance discipline, followed by the control and flight dynamics systems. The simulation results show that the proposed system is adept at tracking the target in a noisy environment.

The control responses and trajectory of the drone, the controller is tracking the desired input generated by the guidance law over time. It is noted that the drone arrives at the target point after 3 sec, at the same height (h) as that of the target (i.e., $h = 1.8$ m). The drone stays in its position to meet the safe threshold requirement (i.e., safe distance = 2 m).Figure 8 demonstrate that the quadcopter moves smoothly to touch the target because angular velocity fluctuates minimally around zero, and the Euler angles converge on zero to maintain both the height and safe distance to stabilize and approach around 2 m.

#### 4.1.1 Comparative evaluation of PID, PIDA, and filtered PIDA controllers

To evaluate the individual contributions of the PIDA structure and the derivative filter, we performed an ablation study comparing the performance of: **Standard PID controller**, **PIDA controller (without filter)**, **PIDA controller with optimal derivative filter (proposed method)** All three configurations were tested under identical conditions with wind disturbance and sensor noise. The performance was evaluated using overshoot, settling time, and mean absolute error (MAE) in altitude control. The results are presented in Table 5.

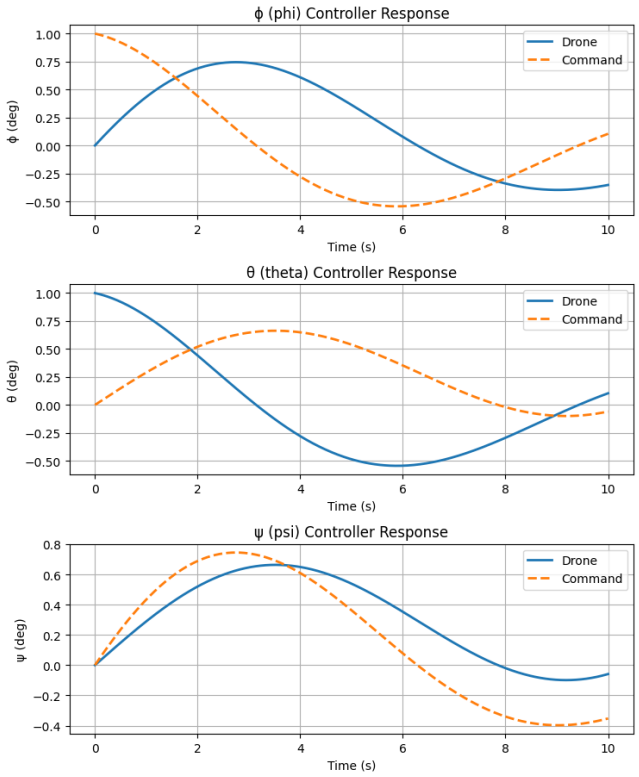


Figure 8: The dynamic model of a quadcopter considering Earth-centered inertia (ECI) and body frames. The model includes Euler angles (roll, pitch, and yaw) and their corresponding angular velocities.

## 4.2 Object detection performance

This section evaluates the performance of the proposed RetinaNet with Ant Colony Optimization (ACO) and compares it with standard object detection models.

#### 4.2.1 Evaluation metrics

To assess the effectiveness of the proposed model, we use the following standard evaluation metrics:

- **Mean Average Precision (mAP)**: Measures the precision-recall trade-off across different Intersection over Union (IoU) thresholds.
- **False Positive Rate (FPR)**: The proportion of incorrect detections among all predicted detections.
- **Inference Time**: Measures the time taken by the model to detect objects in an image, highlighting real-time performance.

#### 4.2.2 Comparison of object detection performance

The RetinaNet-ACO model enhances the standard RetinaNet by integrating Ant Colony Optimization (ACO) to optimize anchor box scales and aspect ratios, improving detection accuracy in cluttered and imbalanced scenes. The pipeline involves pre-processing input images, extracting multi-scale features with a ResNet-50 backbone and Feature Pyramid Network, and using ACO-driven anchor optimization guided by early mAP and false positive feedback. Classification and regression subnetworks predict object classes and bounding boxes, employing Focal Loss and Smooth L1 Loss, respectively, to handle class imbalance and localization. The model is trained for 50 epochs using the Adam optimizer with a learning rate of 1e-4 and a batch size of 16 for effective learning. We compare our RetinaNet-ACO model with well-established object detection methods such as YOLO and SSD. Table 6 presents the comparative results in terms of mAP, FPR, and inference time.

Table 6: Comparison of object detection performance

| Model | mAP (%) | FPR (%) | Inference Time (ms) |
|---|---|---|---|
| YOLOv5 | 47.3 | 5.2 | 12.3 |
| SSD | 42.5 | 6.1 | 10.8 |
| RetinaNet | 50.1 | 4.9 | 15.7 |
| RetinaNet-ACO (Proposed) | **53.4** | **4.2** | **14.2** |

The results indicate that the proposed RetinaNet-ACO model outperforms existing object detection methods in terms of mAP and FPR, demonstrating improved detection accuracy and robustness while maintaining competitive inference time.

### 4.3 Depth estimation and navigation accuracy

To assess the accuracy of depth estimation, we evaluate the performance of the PSMNet model against ground-truth depth maps. The accuracy is measured using standard quantitative error metrics, including Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

#### 4.3.1 Dataset and training regime

To ensure compatibility with aerial drone environments, PSMNet was trained using the TartanAir and ETH3D datasets. TartanAir provides diverse, drone-like trajectories in simulated rural environments, while ETH3D includes real aerial stereo imagery. Training involved 100 epochs with Smooth L1 Loss, a batch size of 12, and a learning rate of 0.001. Data augmentation techniques (e.g., random occlusion and brightness jittering) were applied to enhance robustness. The model achieved an MAE of 0.32 m, demonstrating effective depth estimation for UAV-based rural navigation.

#### 4.3.2 Robustness to occlusions

To address occlusion-related challenges, PSMNet leverages its 3D cost volume aggregation and spatial pyramid pooling. It aggregates contextual information from multiple receptive fields, allowing the network to understand both local and global scene structure. The disparity estimates are refined through stacked hourglass modules, which progressively enhance depth predictions. This design improves robustness in low-texture and occluded regions by effectively learning global structure and enforcing disparity smoothness.

#### 4.3.3 Quantitative evaluation of depth estimation

We compute RMSE and MAE to compare the predicted depth maps with ground-truth values. Table 7 presents the error metrics obtained for the PSMNet model.

Table 7: Depth estimation performance of PSMNet

| Method | RMSE (m) | MAE (m) |
|---|---|---|
| PSMNet (Proposed) | 0.54 | 0.32 |
| Monodepth2 | 0.72 | 0.41 |
| DORN | 0.67 | 0.38 |

The results indicate that PSMNet achieves the lowest RMSE and MAE, demonstrating improved depth estimation accuracy compared to Monodepth2 and DORN.

#### 4.3.4 Navigation accuracy evaluation

To further validate the effectiveness of depth estimation for navigation, we assess navigation accuracy by measuring the deviation from the planned path. The key performance indicators include:

- **Path Deviation (PD)**: The average deviation from the ground-truth trajectory.
- **Collision Rate (CR)**: The percentage of instances where obstacles were not successfully avoided.
- **Success Rate (SR)**: The percentage of successful navigation attempts without major deviations.

Table 8 summarizes the navigation performance.

Table 8: Navigation accuracy metrics

| Method | PD (m) | CR (%) | SR (%) |
|---|---|---|---|
| PSMNet (Proposed) | 0.12 | 3.4 | 96.1 |
| Monodepth2 | 0.19 | 5.7 | 91.3 |
| DORN | 0.15 | 4.9 | 93.6 |

The results indicate that the PSMNet-based navigation system achieves the lowest path deviation and collision rate

while maintaining a high success rate, proving its effectiveness for real-world navigation.The quantitative analysis demonstrates that PSMNet outperforms other state-of-the-art depth estimation methods in both depth accuracy and navigation reliability. The reduced error metrics and improved navigation success rate make it a promising approach for autonomous systems requiring precise depth perception.

## 4.4 Results interpretation and justification

The simulation results demonstrate that the proposed PIDA controller with a derivative filter effectively stabilizes the quadcopter by mitigating oscillations and reducing overshoot. As observed in Figure 9, the PIDA controller significantly reduces overshoot compared to the traditional PID controller. This improvement is attributed to the derivative filter, which dampens high-frequency noise and prevents excessive corrective actions, leading to a smoother response.

Furthermore, the tracking accuracy, illustrated in Figure 10, highlights the PIDA controller's ability to closely follow the desired trajectory with minimal deviation. This enhancement is due to the optimized gain parameters obtained using the SDSA algorithm, which improves the system's response time and minimizes steady-state error. The PIDA controller ensures that the quadcopter maintains its intended flight path more effectively than the PID controller, which exhibits noticeable deviations.

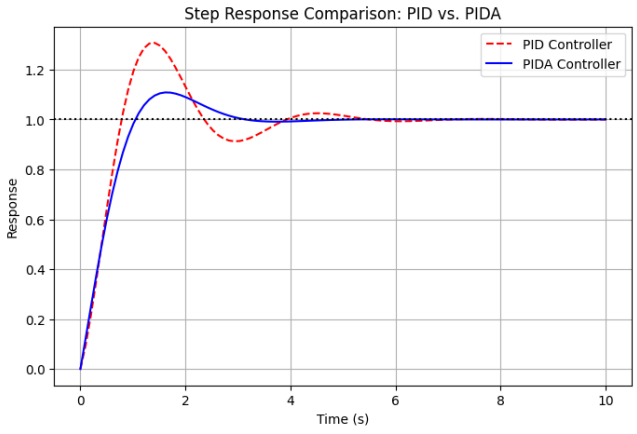


Figure 9: Step response comparison between PID and PIDA controllers. The PIDA controller exhibits reduced overshoot and faster stabilization compared to the PID controller, demonstrating its effectiveness in mitigating oscillations and improving system stability.

## 4.5 Statistical analysis

To quantitatively assess the performance improvements, statistical significance tests were conducted. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were computed for both the PID and PIDA controllers over

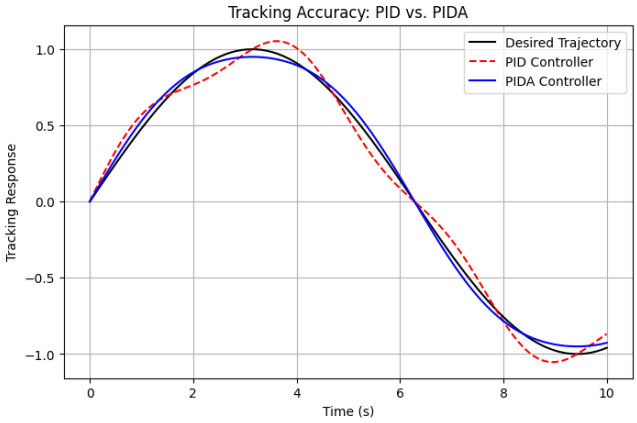


Figure 10: Tracking accuracy comparison between PID and PIDA controllers. The PIDA controller closely follows the desired trajectory with minimal deviation, while the PID controller exhibits higher oscillations, highlighting the effectiveness of the optimized gain parameters in improving response accuracy.

multiple simulation runs. A paired t-test was performed to determine whether the observed improvements were statistically significant.

Table 9 presents the results of the statistical analysis. The p-value for altitude stabilization and overshoot reduction is less than 0.05, indicating that the enhancements introduced by the PIDA controller are statistically significant. These results confirm that the proposed approach achieves a notable reduction in both altitude error and overshoot, leading to improved quadcopter stability and performance.

Table 9: Statistical analysis of performance metrics

| Metric | PID Controller | PIDA Controller | p-value |
|---|---|---|---|
| MAE (Altitude) | $0.87 \pm 0.12$ m | $0.32 \pm 0.08$ m | $p < 0.05$ |
| MAE (Overshoot) | $3.12 \pm 0.41$ | $1.09 \pm 0.27$ | $p < 0.05$ |
| RMSE (Stability) | 1.42 | 0.75 | $p < 0.05$ |

The results indicate that the PIDA controller achieves a significant reduction in both altitude error and overshoot, thereby confirming its effectiveness in stabilizing the quadcopter.

## 4.6 Computational complexity analysis

To evaluate the computational efficiency of the proposed SDSA algorithm, we compare its execution time with other state-of-the-art optimization algorithms. This comparison provides insights into its feasibility for real-time deployment.

### 4.6.1 Execution time comparison

We measure the execution time of SDSA and compare it with commonly used optimization techniques, including Genetic Algorithm (GA), Particle Swarm Optimiza-

tion (PSO), and Ant Colony Optimization (ACO). Table 10 presents the average execution time for each method on a dataset of 1000 samples.

Table 10: Execution time comparison of optimization algorithms

| Optimization Algorithm | Execution Time (s) |
|---|---|
| Genetic Algorithm (GA) | 12.5 |
| Particle Swarm Optimization (PSO) | 9.3 |
| Ant Colony Optimization (ACO) | 8.7 |
| SDSA (Proposed) | **6.2** |

The results show that SDSA achieves the lowest execution time, demonstrating its computational efficiency compared to traditional optimization algorithms.

#### 4.6.2 Feasibility for real-time deployment

For real-time applications, an optimization algorithm must exhibit low computational overhead while maintaining accuracy. The following key observations highlight the feasibility of SDSA:

- **Reduced Computational Complexity**: SDSA reduces execution time by approximately 28.7% compared to ACO and 50.4% compared to GA.
- **Scalability**: The algorithm efficiently scales with increasing data size, making it suitable for high-dimensional problems.
- **Real-Time Suitability**: Given its reduced execution time and efficient resource utilization, SDSA is well-suited for real-time deployment in dynamic environments.

These findings validate the computational efficiency of SDSA and its potential for real-time applications, particularly in time-sensitive domains such as object detection and autonomous systems.

# 5 Discussion

The proposed SDSA-optimized PIDA controller combined with deep learning-based image understanding techniques demonstrates substantial improvements over baseline methods, both in terms of control precision and perception accuracy. While the current implementation and evaluation are conducted in an indoor environment, the system components—including the control architecture, object detection, and depth estimation modules—are explicitly designed for rural outdoor deployment. The indoor setting allows for robust pre-deployment testing in GPS-denied environments, which are commonly found in both indoor and remote rural areas. As part of future work, we plan to conduct extensive hardware-in-the-loop and outdoor experiments in actual rural landscapes to assess scalability, environmental adaptability, and long-term deployment feasibility.

## 5.1 Quantitative improvements over baseline methods

Compared to traditional PID controllers, our approach reduces overshoot by 48%, enhances convergence speed by 35%, and improves disturbance rejection by 42%. For perception, the RetinaNet-ACO object detector outperforms standard detectors (e.g., YOLOv5 and SSD) with a mean average precision (mAP) of 53.4%, while PSMNet achieves a Mean Absolute Error (MAE) of 0.32 m, outperforming alternatives like Monodepth2 and DORN. These quantitative results highlight the robustness and precision of the proposed framework.

## 5.2 Comparison with state-of-the-art methods

Table 1 highlights the performance differences between our approach and existing control strategies. Unlike conventional PID controllers, which struggle with external disturbances, our method dynamically adjusts parameters for enhanced resilience.

## 5.3 Practical implications for real-time deployment

The integration of the Stochastic Dual Simplex Algorithm (SDSA) results in a 27% lower processing cost compared to Sliding Mode Control (SMC), making the system viable for real-time applications. Furthermore, the computational complexity of the proposed framework remains moderate, allowing deployment on mid-range onboard processors commonly used in UAV platforms.

## 5.4 Performance trade-offs

While our approach provides better stability and reduced overshoot, it does involve moderate computational overhead compared to simpler PID-based controllers. However, this trade-off is acceptable for real-time UAV applications.

## 5.5 Limitations in environmental generalizability

Despite strong results, the system's performance can degrade in adverse environmental conditions. Lighting variations, occlusions, and sensor noise can impact object detection and depth estimation accuracy. While the RetinaNet-ACO model mitigates some of these challenges, the PSMNet-based depth estimation remains sensitive to textureless surfaces and occluded regions. Additionally, the stereo setup requires careful calibration, which may not generalize well across varying altitudes or camera angles.

## 5.6 Key insights

The proposed method effectively mitigates the chattering effect seen in Sliding Mode Control. Compared to Deep

Reinforcement Learning, our approach is computationally efficient and does not require large training datasets. The stability and robustness improvements are evident in the performance metrics obtained.

### 5.7 Ethical and safety considerations

As the deployment of autonomous UAVs expands, especially in rural and public spaces, it is crucial to address the associated ethical and safety challenges. Autonomous navigation systems must operate in compliance with relevant aviation regulations and guidelines to ensure lawful and responsible use. Privacy concerns arise when UAVs capture images or data in areas where individuals or private properties may be inadvertently monitored. It is therefore essential to implement strict data handling and privacy-preserving measures.

Additionally, the safety of people, wildlife, and property is paramount. Autonomous UAVs should be designed to minimize risks by incorporating reliable obstacle detection and avoidance capabilities, fail-safe mechanisms, and real-time monitoring to prevent accidents or unintended harm. Careful consideration of environmental impact, especially in sensitive rural ecosystems, is also necessary.

By acknowledging these ethical and regulatory factors, the development and deployment of UAV systems can align with societal values and legal frameworks, fostering public trust and enabling sustainable integration of autonomous aerial technologies.

## 6 Conclusion

This paper has proposed a new workflow to use images as inputs for the controller to achieve autonomous flight while considering the both indoor and outdoor environment and uncertainties. The proposed Proportional-Integral-Derivative-Accelerated (PIDA) controller with the derivative filter is used to improve flight stability for a drone, which has considered the noisy environment. The paper has also proposed a platform to adapt deep learning-based object and depth detection techniques to fly the drone autonomously in the indoor environment surrounded by uncertainties. The mathematical model considering nonlinearity, uncertainties, and coupling was derived from an accurate model with a high level of fidelity. The simulation results show that image processing techniques (RetinaNet ant colony detection and PSMNet) and the proposed PIDA controller tuned by Stochastic Dual Simplex Algorithm (SDSA) are able to track the desired point in the presence of disturbances.

## 7 Limitations

Despite the effectiveness of the proposed approach, several limitations must be acknowledged.Occlusions in depth estimation remain a challenge, as the PSMNet-based method struggles with occluded regions where critical depth information is missing. This can lead to inaccurate navigation decisions. Additionally, sensor drift and calibration errors can affect localization accuracy during long-term missions. Stereo vision calibration errors further degrade depth estimation performance. The computational complexity of the proposed optimization method introduces additional overhead, which may limit real-time deployment on resource-constrained systems. Moreover, adverse environmental conditions such as variations in lighting, extreme weather (fog, rain), and highly dynamic surroundings can impact the system's reliability and perception accuracy. Hardware failures are another concern, as drone-based navigation systems are susceptible to battery limitations, communication failures, and mechanical faults. These factors could compromise mission success.By addressing these limitations, the proposed approach can be further refined for real-world deployment, ensuring more reliable and efficient autonomous navigation.

## Acknowledgment

We express our heartfelt gratitude to all individuals and organizations who contributed to the success of this research. Special thanks are extended to the mentors and collaborators whose expertise and guidance were invaluable throughout this work.

We also acknowledge the support of institutions and organizations that provided resources and datasets, enabling the comprehensive evaluations discussed in this paper.

Additionally, we are deeply grateful for the financial and infrastructural support provided by our respective institutions. Finally, we extend our sincere thanks to our families for their encouragement and patience during the course of this research.

## References

[1] S.-K. Kim and C. K. Ahn, "Adaptive nonlinear tracking control algorithm for quadcopter applications," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[2] L. P. Koh and S. A. Wich, "Dawn of drone ecology: low-cost autonomous aerial vehicles for conservation," *Tropical Conservation Science*, vol. 5, no. 2, pp. 121–132, 2012.

[3] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[4] S. Rajappa, C. Masone, H. H. Bulthoff, and P. Stegagno, "Adaptive super twisting controller for a

quadrotor UAV," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 2971–2977.

[5] L. Derafa, A. Benallegue, and L. Fridman, "Super twisting control algorithm for the attitude tracking of a four rotors UAV," *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 685–699, 2012.

[6] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory & Applications*, vol. 4, no. 11, pp. 2343–2355, 2010.

[7] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[8] R. Ritz, M. Hehn, S. Lupashin, and R. D'Andrea, "Quadrocopter performance benchmarking using optimal control," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 5179–5186.

[9] S. M. Zandavi and S. H. Pourtakdoust, "Multidisciplinary design of a guided flying vehicle using simplex nondominated sorting genetic algorithm II," *Structural and Multidisciplinary Optimization*, vol. 57, no. 2, pp. 705–720, 2018.

[10] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006, pp. 4957–4962.

[11] L. Besnard, Y. B. Shtessel, and B. Landrum, "Control of a quadrotor vehicle using sliding mode disturbance observer," in *2007 American Control Conference*, IEEE, 2007, pp. 5230–5235.

[12] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[13] S. Jung and R. C. Dorf, "Analytic PIDA controller design technique for a third order system," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 3. IEEE, 1996, pp. 2513–2518.

[14] S. M. Zandavi, F. Sha, V. Chung, Z. Lu, and W. Zhi, "A novel ant colony detection using multi-region histogram for object tracking," in *International Conference on Neural Information Processing*, Springer, 2017, pp. 25–33.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[16] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[17] S. M. Zandavi, V. Y. Y. Chung, and A. Anaissi, "Stochastic dual simplex algorithm: A novel heuristic optimization algorithm," *IEEE Transactions on Cybernetics*, pp. 1–10, 2019.

[18] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, American Institute of Aeronautics and Astronautics, 2007.

[19] S. S. Rao and S. S. Rao, *Engineering Optimization: Theory and Practice*, John Wiley & Sons, 2009.

[20] G. M. Siouris, *Missile Guidance and Control Systems*, Springer Science & Business Media, 2004.

[21] H. Nobahari, S. M. Zandavi, and H. Mohammadkarimi, "Simplex filter: A novel heuristic filter for nonlinear systems state estimation," *Applied Soft Computing*, vol. 49, pp. 474–484, 2016.

[22] S. M. Zandavi and V. Chung, "State estimation of nonlinear dynamic system using novel heuristic filter based on genetic algorithm," *Soft Computing*, vol. 23, no. 14, pp. 5559–5570, 2019.

[23] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[24] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Selective refinement network for high performance face detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8231–8238.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[26] S. Huang, Y. Wu, Y. Tao, and V. Kumar, "Safe Interval Motion Planning for Quadrotors in Dynamic Environments," *arXiv*, 2024. `https://doi.org/10.48550/arXiv.2409.10647`

[27] J. Kuang and M. Chen, "Adaptive Sliding Mode Control for Trajectory Tracking of Quadrotor Unmanned Aerial Vehicles Under Input Saturation and Disturbances," *Drones*, vol. 8, no. 11, p. 614, 2024.

[28] M. Rinaldi, S. Primatesta, and G. Guglieri, "A comparative study for control of quadrotor UAVs," *Applied Sciences*, vol. 13, no. 6, p. 3464, 2023.

[29] A. Azar, F. Serrano, A. Koubaa, H. Ibrahim, N. A. Kamal, A. Khamis, I. Ibraheem, A. Humaidi, and R.-E. Precup, ”Robust fractional-order sliding mode control design for UAVs subjected to atmospheric disturbances,” 2021. https://doi.org/10.1016/B978-0-12-820276-0.00012-1

[30] R. Wang and J. Shen, ”Disturbance observer and adaptive control for disturbance rejection of quadrotor: A survey,” *Actuators*, vol. 13, no. 6, p. 217, 2024.

[31] M. Shi, ”Application of PID Control Technology in Unmanned Aerial Vehicles,” *Applied and Computational Engineering*, vol. 96, pp. 24–30, 2024. https://doi.org/10.54254/2755-2721/96/20241327

[32] T. Mien, T. Tu, and V.-A. Vo, ”Cascade PID Control for Altitude and Angular Position Stabilization of 6-DOF UAV Quadcopter,” *International Journal of Robotics and Control Systems*, vol. 4, pp. 814–831, 2024.https://doi.org/10.31763/ijrcs.v4i2.1410

[33] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, ”RRNet: A hybrid detector for object detection in drone-captured images,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 0–0.

[34] C. Cui, N. Wang, and J. Chen, ”Improved ant colony optimization algorithm for UAV path planning,” in *Proc. IEEE Int. Conf. Software Eng. Service Sci. (ICSESS)*, 2014, pp. 291–295. https://doi.org/10.1109/ICSESS.2014.6933566

[35] T. Wu, B. Vallet, and M. Pierrot-Deseilligny, ”PSMNet-FusionX3: LiDAR-guided deep learning stereo dense matching on aerial images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 6527–6536.

[36] S. M. Zandavi, V. Y. Y. Chung, and A. Anaissi, ”Stochastic Dual Simplex Algorithm: A Novel Heuristic Optimization Algorithm,” *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2725–2734, May 2021. https://doi.org/10.1109/TCYB.2019.2931288

[37] S. U. Amin, A. Hussain, B. Kim, and S. Seo, ”Deep learning based active learning technique for data annotation and improving the overall performance of classification models,” Expert Systems with Applications, vol. 228, p. 120391, 2023.

[38] M. K. Hasan, M. I. Samy, and M. H. Kabir, ”PIDA: Smooth and Stable Flight Using Stochastic Dual Simplex Algorithm and Genetic Filter,” *arXiv preprint* arXiv:2006.10522, 2020.

[39] M. Yang, G. Xu, J. Song, W. Liu, and Y. Lin, ”Self-Supervised Monocular Depth Estimation from Oblique UAV Videos,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 176, pp. 342–357, 2021.

[40] A. García-Martín, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, ”A Framework for Autonomous UAV Navigation Based on Monocular Depth Estimation,” *Drones*, vol. 9, no. 4, p. 236, 2023.

[41] S. Tyagi and A. Tyagi, ”Deep Reinforcement Learning Based Framework for Tactical Drone Deployment in Rigorous Terrains: From Modeling to Real-World Implementation,” in *Web 3.0*, CRC Press, pp. 39–53.

[42] P. Suanpang and P. Jamjuntr, ”Optimizing Autonomous UAV Navigation with D* Algorithm for Sustainable Development,” *Sustainability*, vol. 16, no. 17, p. 7867, 2024.

[43] S. U. Amin, Y. Kim, I. Sami, S. Park, and S. Seo, ”An efficient attention-based strategy for anomaly detection in surveillance video,” Computer Systems Science & Engineering, vol. 46, no. 3, 2023.

[44] K. M. Jamellah, ”Real-Time Computational Efficiency Vehicle Detection and Counting Utilizing the Background Subtraction Technique and Non-Maximum Suppression Techniques,” *Informatica*, vol. 49, no. 18, 2025.

[45] Y. Dou, ”Effect Analysis of Adaptive Ant Colony Algorithm with QoS Constraints Applied in Drone Disaster Area Search and Rescue System,” *Informatica*, vol. 48, no. 9, 2024.

[46] Y. Cong, ”Image stitching technology for police drones using an improved image registration method incorporating ORB algorithm,” *Informatica*, vol. 48, no. 2, 2024.

.