IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 4, December 2026, pp. 927~943
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i4.pp927-943

An improved anti-windup PID architecture with feed-forward compensation for onboard-computer trajectory and velocity control of hexarotor UAVs

Nguyen Thi Dieu Linh[1], Nguyen Van Bang[2]
[1]Department of Science and Technology, Hanoi University of Industry, Hanoi, Viet Nam
[2]Faculty of Missiles, Air Defence - Air Force Academy, Ha Noi, Viet Nam

Article Info

Article history:

Received Jul 24, 2026
Revised Sep 17, 2026
Accepted Nov 10, 2026

Keywords:

Actuator saturation
Anti-windup control
Companion computer
Hexarotor UAV
Trajectory tracking

ABSTRACT

Onboard trajectory tracking of hexarotor unmanned aerial vehicles (UAVs) degrades when reference-induced phase lag, actuator saturation, wind, and communication delay act together. This paper proposes a cascaded proportional-integral-derivative (PID) architecture for a PX4/ROS 2 companion-computer implementation, augmented by reference-acceleration feed-forward and a vector anti-windup law that maps the thrust-saturation residual into all three world-axis integral channels. A six-degree-of-freedom Newton-Euler model was simulated with a full-rank control-allocation matrix, first-order motor dynamics, a 250 Hz attitude loop, a 50 Hz outer loop, and a 20 ms delayed noisy state. Twenty common-random-number Monte Carlo runs were evaluated in three saturation regimes against a tuned super-twisting sliding-mode benchmark. Under a nominal thrust envelope, the augmentation reduced the mean three-dimensional root-mean-square error (RMSE) by 11.5% and the peak error by 15.9%; under a tighter envelope, it reduced RMSE by 26.1% and the maximum integral-state norm by 77.0%. Beyond the envelope, the unprotected baseline diverged in all twenty runs while the proposed controller completed every mission. The augmentation added 31 scalar operations per cycle and no additional state. Although the simulation embeds realistic delay, noise, and actuator limits, the results remain purely numerical, so hardware-in-the-loop and flight testing remain necessary before any operational claim.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Nguyen Thi Dieu Linh
Department of Science and Technology, Hanoi University of Industry
Hanoi, Viet Nam
Email: nguyen.linh@haui.edu.vn

1. INTRODUCTION

Multirotor unmanned aerial vehicles (UAVs) have become indispensable platforms for aerial surveillance, infrastructure inspection, precision agriculture and search-and-rescue, because they take off and land vertically, hover precisely and fly slowly in confined or global-positioning-system-degraded spaces [1], [2]. Among multirotor configurations the hexarotor is particularly attractive: its six-actuator layout provides higher payload capacity, a larger attainable-wrench set and an inherent control redundancy that improves robustness to single-rotor faults relative to the quadrotor [3], [4]. These advantages have motivated a steady stream of research into hexarotor modeling, actuation design and flight control [5], [6]. The architecture studied here is directly applicable to industrial inspection drones that must hold a camera on a structure in gusty air, and to automated warehouse inventory systems in which a hexarotor scans racking aisles under a strict positioning tolerance.

The classical way to fly a multirotor autonomously is to run every control loop on the embedded flight controller. Modern missions, however, increasingly require onboard perception, motion planning and learning that exceed the computational budget of a microcontroller-class autopilot. This has driven the adoption of a companion computer, for example an NVIDIA Jetson, an Intel NUC or a Raspberry Pi, that hosts high-level autonomy under Linux and issues offboard setpoints to the autopilot [7], [8]. In the PX4/ROS 2 ecosystem the coupling is realized through the micro extremely-resource-constrained-environment data-distribution-service (micro-XRCE-DDS) bridge, which exchanges autopilot telemetry and external position, velocity or attitude commands at configured update rates [9]. Executing the trajectory-level control law on the companion computer decouples autonomy from the ground station, which is decisive when the radio link is weak or intermittent, as in dense urban, forest or maritime scenarios [2], [7].

Within this architecture the trajectory-level law is usually implemented as a PID or cascaded PID regulator, because it is computationally inexpensive, transparent to tune and already familiar to flight-test engineers [10], [11]. Two practical effects nevertheless remain problematic. First, sustained saturation lets the integral state accumulate beyond the achievable control authority, which delays recovery once the command re-enters the feasible set and, in the worst case, destabilizes the loop entirely [4], [12]. Second, feedback alone cannot reproduce the acceleration of a moving reference without phase lag, and the transport delay of the companion-computer link increases the along-track error still further [9], [10].

A large body of work replaces the PID law outright. Sliding-mode designs achieve strong robustness: a fuzzy PID sliding surface improves quadcopter tracking [13], a higher-order sliding-mode law has been applied to fixed-wing vehicles [14], and an extended-state observer sharpens sliding-mode attitude control [15]. Observer-based adaptive neural control handles model uncertainty and external disturbance simultaneously [16], while combining model predictive control (MPC) with a super-twisting algorithm yields robust quadrotor tracking [17]. Modified reaching laws further reduce chattering [18]. Reinforcement learning has been used to predict controller gains [19], to stabilize a quadrotor under a Lipschitz-constrained policy [20] and to adapt continually in wind [21]. Fuzzy nonlinear controllers have been compared on a coaxial octorotor [22]. For over-actuated and fully actuated platforms, trajectory optimization exploits the enlarged wrench set [3], a tilt-rotor hexacopter prototype has been modeled and controlled for precision agriculture [6], and a hierarchical composite adaptive sliding-mode scheme has been proposed for hexarotor position and attitude [23]. Wind rejection has been addressed by antidisturbance sliding mode [24], by frequency-dependent H-infinity design [25] and by active disturbance rejection control [26].

These methods are powerful, but each buys robustness with an additional requirement. Sliding-mode laws need a disturbance bound and a boundary layer, and they leave their own integral or auxiliary states unprotected against saturation. Predictive schemes need an online optimizer and a reliable model [1], [12], [27]. Observer-based and learning-based schemes need estimator tuning or training data [16], [19]–[21]. A parallel line of work keeps the PID structure and tunes it automatically, using bio-inspired optimization [28], evolutionary learning [29] or genetic algorithms with system identification [30]; these improve the gains but not the behavior once the command leaves the feasible set. Two gaps remain. First, published comparisons frequently change the controller structure, the gains and the disturbance realization at the same time, so the measured benefit cannot be attributed to a single mechanism. Second, the anti-windup literature for multirotors is dominated by scalar clamping, whereas the saturation of a multirotor is a constraint on the magnitude of a three-dimensional force vector, which couples the three integral channels. A further consideration decides which of these routes is practical in the field. The autopilots that actually fly production multirotors, PX4 and ArduPilot foremost among them, implement their position and velocity loops as cascaded PID and expose those loops through a small, stable and well-documented set of gains. Replacing that structure means rewriting and re-qualifying flight-critical firmware, retraining the operators who tune it and discarding the airworthiness evidence already accumulated for the released control path, whereas augmenting it leaves the gain set, the tuning workflow and the certification basis intact. For a standard industrial autopilot, improving the PID is therefore the shortest path from a published result to a deployed aircraft, and it is the route taken here. Table 1 positions the present work against representative recent studies along these two axes.

a. Novelty

To the best of the authors' knowledge, this is the first work to combine gravity feed-forward, reference-acceleration feed-forward and a dual-mechanism vector anti-windup law within a PX4/ROS 2 offboard architecture for hexarotor control, and to validate the combination under a strictly matched comparison protocol. Three elements distinguish the study. The baseline and the proposed controller share the same pole-placed gains, the same gravity compensation, the same plant, the same delayed measurements and bit-identical wind and noise samples, so that only the two added mechanisms can explain any difference. The anti-windup residual is formed in controller-output coordinates and mapped into all three world-axis integral channels rather than clamped axis by axis. Finally, the evaluation does not stop at a nominal

operating point: the thrust envelope is tightened until the unprotected baseline fails, which locates the boundary at which the mechanism stops being a refinement and becomes a requirement.

Table 1. Comparison with representative recent UAV-control studies

Study	Platform and method	Constraints and disturbances	Evidence	Difference from this work
Xu *et al.* [4]	Fully actuated tilt-rotor hexarotor; anti-windup active disturbance rejection with allocation	Actuator amplitude and rate saturation	Simulation	Observer-based design; no matched-gain comparison and no companion-computer delay
Lopez-Sanchez and Moreno-Valenzuela [10]	Survey of quadrotor PID architectures	Broad review of PID limitations	Review	No isolated augmentation and no matched-disturbance experiment
Zhou *et al.* [12]	Hybrid MPC-PID trajectory control	Model uncertainty and disturbance	Numerical study	Online predictive optimization with far larger model and computation requirements
Han *et al.* [23]	Hierarchical composite adaptive sliding mode for a hexarotor	Parametric uncertainty and disturbance	Numerical study	Nonlinear adaptive law; saturation handling is not isolated
Wang *et al.* [24]	Antidisturbance sliding-mode trajectory control	Wind disturbance	Numerical study	Discontinuous nonlinear control rather than a drop-in augmentation
Present work	Underactuated hexarotor; reference feed-forward plus vector anti-windup added to a pole-placed PID	Wind, gusts, noise, 20 ms delay, motor lag and three thrust envelopes	20-seed Monte Carlo with paired t-tests, ablation and a tuned super-twisting benchmark	Identical gains and common random numbers isolate two computationally light additions; the failure boundary is located explicitly

b. Contributions and achieved results

The paper contributes: A six-degree-of-freedom hexarotor model with an explicit full-rank control-allocation matrix, per-rotor thrust limits and first-order motor dynamics, embedded in a 50 Hz companion-computer and 250 Hz flight-controller architecture with a documented 20 ms transport delay; a cascaded PID augmented by reference-acceleration feed-forward and a vector conditional-integration anti-windup law, with a documented gain-selection procedure derived from an explicit time-scale-separation constraint; a twenty-seed common-random-number evaluation in three saturation regimes, with paired significance tests, an ablation study, a tuned super-twisting sliding-mode benchmark and a measured computational cost. Relative to the matched baseline, the proposed controller reduced the mean three-dimensional root-mean-square error (RMSE) by 11.5% under the nominal envelope and by 26.1% under a tighter envelope, reduced the peak error by 15.9% and 21.0% respectively, and reduced the maximum integral-state norm by 77.0% in the tighter case. Beyond the envelope the baseline diverged in twenty runs out of twenty whereas the proposed controller completed every mission. The augmentation costs 31 additional scalar operations per 20 ms cycle and no additional persistent state. Section 2 develops the model and the controller, section 3 describes the simulation and evaluation protocol, section 4 reports and discusses the results, and section 5 concludes.

2. SYSTEM ARCHITECTURE AND CONTROLLER DESIGN

2.1. Onboard-computer offboard control architecture

The control system follows the two-rate cascaded architecture of Figure 1, which is representative of a PX4/ROS 2 deployment. A companion-computer node executes the trajectory-level law at 50 Hz and publishes offboard setpoints, while the flight controller executes the 250 Hz attitude and mixer loops and returns state estimates through the micro-XRCE-DDS bridge [9]. Mission logic is organized as a finite-state machine covering pre-flight checks, take-off, offboard trajectory tracking, body-frame velocity control and return-to-launch. The present study concentrates on the trajectory-tracking and velocity-control states, in which the quality of the outer law determines mission accuracy directly. Hardware-validated autopilot-tuning studies support retaining a computationally economical controller structure for practical onboard implementation [30].

The transport delay is modeled explicitly. The position and velocity returned to the outer loop are delayed by one outer-loop sample, that is by exactly 20 ms, using a first-in first-out buffer. The delay is fixed and deterministic, it is applied to the measured feedback only and never to the setpoints traveling in the opposite direction, and it is identical for the three controllers compared in this paper. Serialization jitter is not modeled. Packet loss is not present in the nominal configuration; it is introduced separately in section 4.6 as an independent Bernoulli process acting on the feedback samples, a lost sample causing the outer loop to hold its previous command until the next successful update. This is therefore a bounded-delay abstraction of the link rather than a packet-level network model.

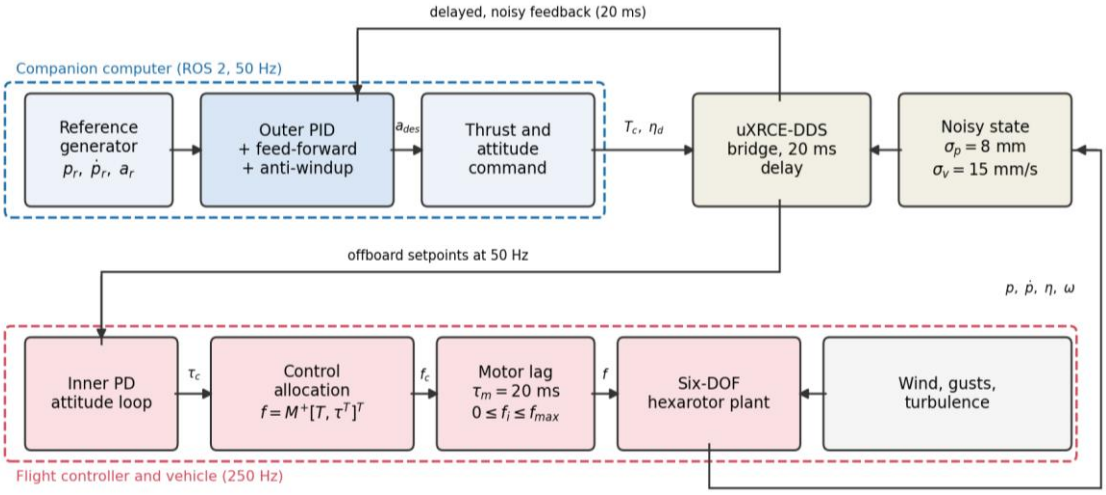


Figure 1. Complete control architecture: the upper dashed boundary encloses the 50 Hz companion-computer node and the lower dashed boundary encloses the 250 Hz flight controller and vehicle; the uXRCE-DDS bridge applies the fixed 20 ms feedback delay.

2.2. Six-degree-of-freedom hexarotor dynamic model

Let the world frame be denoted by the subscript w and the body frame by b. The hexarotor position is $p = [x, y, z]^{\mathrm{T}}$ and its attitude is described by the Euler angles denoted $\eta = [\varphi, \theta, \psi]^{\mathrm{T}}$ for roll, pitch and yaw. With m the mass, g the gravitational acceleration, R the body-to-world rotation matrix, $e_3 = [0, 0, 1]^{\mathrm{T}}$, T the collective thrust and F_d the aerodynamic and wind force, the translational Newton-Euler equation reads

$$m\ddot{p} = R(\eta)Te_3 - mge_3 - D\dot{p} + F_d \tag{1}$$

where the linear drag $D = \mathrm{diag}(k_d)$ captures translational aerodynamic damping. The rotational dynamics in the body frame are

$$J\dot{\omega} + \omega \times (J\omega) = \tau \tag{2}$$

with $J = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$ the inertia matrix, ω the body angular rate and τ the control torque. The state vector is formed by stacking the position, its time derivative, the Euler angles and the body angular rate, giving twelve states in total. This confirms the six-degree-of-freedom, underactuated nature of the platform: four independent wrench components steer six configuration coordinates [5], [14].

2.3. Control allocation, actuator limits and motor dynamics

The vehicle is a planar, non-tilted hexarotor in the PX4 hexa-X layout. Rotor i $l$ies at radius l from the center of mass at azimuth β_i measured from the body x-axis, with $\beta = [30, 90, 150, 210, 270, 330]$ degrees, and all six thrust axes are parallel to the body z-axis. The spin directions alternate, $\sigma = [+1, -1, +1, -1, +1, -1]$, where +1 denotes counter-clockwise rotation seen from above. Because the rotor position is $r_i = l[\cos\beta_i, \sin\beta_i, 0]^{\mathrm{T}}$ and the rotor thrust is $f_i e_3$, the moment contributed by rotor i is the cross product of r_i with $f_i e_3$, and the reaction torque about the body z-axis is $\sigma_i(k_m/k_t)f_i$. Collecting the six rotor thrusts in f gives the wrench relation

$$\begin{bmatrix} T \\ \tau \end{bmatrix} = Mf, \quad M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.140 & 0.280 & 0.140 & -0.140 & -0.280 & -0.140 \\ -0.242 & 0 & 0.242 & 0.242 & 0 & -0.242 \\ 0.017 & -0.017 & 0.017 & -0.017 & 0.017 & -0.017 \end{bmatrix} \tag{3}$$

The four rows of M correspond to the collective thrust and to the roll, pitch and yaw torques; the entries are given in meters for $l = 0.28$ m and $k_m/k_t = 0.0171$ m. The matrix has full row rank 4 and a condition number of 58.3, so the commanded wrench is always attainable in the unsaturated region and the two-dimensional null space of M represents the actuator redundancy of the hexarotor. Allocation uses the minimum-norm right inverse followed by a per-rotor clip,

$$f_c = \mathrm{sat}\left(M^{+}\begin{bmatrix}T_c\\ \tau_c\end{bmatrix}, 0, f_{max}\right), \quad f_{max} = 7.55\mathrm{N} \tag{4}$$

so that the six-rotor redundancy is resolved at the mixer without altering the outer control law [3], [4]. Recent hexarotor and fully actuated multirotor studies confirm that actuator redundancy, control allocation and disturbance-aware hierarchical control remain central design issues [6], [22], [23], [25].

The rotors are not treated as ideal thrust sources. Each rotor responds to its commanded thrust through a first-order lag,

$$\tau_m \dot{f}_i + f_i = f_{c,i}, \quad \tau_m = 20\mathrm{ms}, \quad i = 1, \ldots, 6 \tag{5}$$

which represents the combined electronic-speed-controller and rotor-inertia response of a 2 kg-class vehicle. The wrench actually applied to the airframe is therefore Mf and not Mf_c. Blade flapping, induced-flow transients and rotor-to-rotor aerodynamic interaction are not modeled; this limitation is restated in section 4.8.

2.4. Baseline cascaded PID controller

The outer loop generates a desired acceleration from the position error $e_p = p_r - p$ and the velocity error e_v, each defined as the reference value minus the measured value, through

$$a_{des} = K_p e_p + K_d e_v + K_i \int_0^t e_p d\sigma \tag{6}$$

The desired world force includes gravity compensation, $F = ma_{des} + mge_3$. Its magnitude provides the collective-thrust command after clipping to the software envelope, while its direction defines the desired roll and pitch angles for zero reference yaw. The inner loop uses the proportional-derivative torque law $\tau = J(K_{pa}e_\eta - K_{da}\omega)$ [10], [11]. In the baseline the translational integrator is unbounded and no reference-acceleration feed-forward is applied.

2.5. Proposed augmentation

The proposed controller uses exactly the same gains, the same gravity compensation and the same inner loop as the baseline. Any difference in the results is therefore attributable to the two additions below.

Reference-acceleration feed-forward. A weighted reference-acceleration term is added to reduce the phase lag along a time-varying trajectory,

$$a_{des} = K_p e_p + K_d e_v + K_i I + k_{ff} a_r, \quad 0 \le k_{ff} \le 1 \tag{7}$$

where I is the discrete integral state defined below. The weight k_{ff} is kept below unity so that feedback authority is preserved and so that the numerical differentiation noise of a planner-generated reference is not injected at full gain into the command; section 2.7 quantifies this choice.

Vector anti-windup. Let F be the unconstrained world-force command and F_{sat} its magnitude-limited counterpart. The saturation residual is formed in controller-output coordinates, that is after the gravity-compensation term has been removed,

$$\Delta a = \frac{F_{sat} - mge_3}{m} - a_{des} \tag{8}$$

This detail matters. If the residual is formed in force coordinates, it inherits a constant bias along the negative z-axis from the gravity-compensation term, and the anti-windup feedback then drives the vertical integral state instead of unwinding it. With Δa defined by (8), the residual vanishes whenever the command is feasible and otherwise points from the infeasible command back into the feasible set. The integral state is updated by

$$I(k+1) = \mathrm{clip}[I(k) + (e_{int}(k) + K_{aw}\Delta a(k))\Delta t, -I_{lim}, I_{lim}] \tag{9}$$

in which the conditional-integration rule selects, component by component,

$$e_{int,j} = \begin{cases} 0 & \text{if} K_{i,j} e_{p,j} \Delta a_j < 0 \\ e_{p,j} & \text{otherwise} \end{cases}, \quad j = x, y, z \tag{10}$$

Equation (10) is the mechanism that distinguishes this law from the usual scalar treatment. Because the multirotor constraint acts on the magnitude of a three-dimensional force vector, saturation on one axis changes the feasible set of the other two. The sign test in (10) compares each integral channel with the direction in which the command must move to become feasible, and suppresses only the components that would push the already infeasible command further outward. The remaining components continue to integrate, so steady disturbance rejection is not sacrificed. The clamp in (9) then guarantees that the integral state remains in the compact set defined by I_{lim}. The attitude loop is proportional-derivative and therefore contributes no attitude-integrator windup.

2.6. Stability and time-scale separation

This subsection states the stability properties that can be justified for the implemented model. The analysis is local to the small-angle, unsaturated operating region of the cascaded system. It establishes exponential stability of the nominal linearized error dynamics and input-to-state stability (ISS) with respect to bounded lumped disturbances. It is not a global proof for the full nonlinear vehicle under arbitrary saturation or delay.

a. Inner attitude loop

With the proportional-derivative torque law and the rotational dynamics (2), the small-angle attitude-error dynamics reduce to

$$\ddot{e}_\eta + K_d^a \dot{e}_\eta + K_p^a e_\eta = 0 \tag{11}$$

which is Hurwitz for positive gains. The chosen gains $K_{pa} = [225, 225, 100]\,\mathrm{s}^{-2}$ and $K_{da} = [24, 24, 16]\,\mathrm{s}^{-1}$ place the roll and pitch channels at a natural frequency of 15 rad/s with damping ratio 0.8, and the yaw channel at 10 rad/s with the same damping ratio.

b. Time-scale separation

The cascaded design is only valid if each outer channel is substantially slower than the inner element that actuates it. The correct comparison differs by axis. The horizontal channels are actuated by tilting the airframe and must therefore be slower than the attitude loop, whose dominant time constant is $1/(\zeta\omega_n) = 83$ ms. The vertical channel is actuated directly by collective thrust, so its fastest neighboring dynamics is the motor lag of 20 ms and not the yaw loop. With the outer gains of section 2.7 the slowest outer closed-loop pole gives a dominant time constant of 500 ms horizontally and 400 ms vertically, so the separation ratios are

$$\frac{T_{out}^{xy}}{T_{att}} = \frac{500}{83} = 6.0, \quad \frac{T_{out}^{z}}{\tau_m} = \frac{400}{20} = 20.0 \tag{12}$$

Both exceed the customary factor of five, and the attitude loop is itself 4.2 times slower than the motor lag, so the ordering from fastest to slowest, motor then attitude then translation, is respected throughout. The sampling-frequency ratio of 250 Hz to 50 Hz is not used as the justification, because a ratio of sampling rates is not a singular-perturbation parameter. The outer-loop analysis assumes only that the residual attitude-tracking error remains bounded and includes it in the lumped disturbance.

c. Outer translational loop

With gravity compensation applied to both controllers, the nominal slow-manifold acceleration equals the commanded acceleration a_{des} reduced by the linear drag term and displaced by the disturbance force divided by the mass. For each diagonal axis let e be the axis $e$rror and ζ its time integral. Feed-forward mismatch, drag, wind, bounded delay and noise, and bounded attitude-tracking residuals are grouped into the lumped disturbance δ. In the unsaturated region the error model is

$$\dot{z} = Az + B\delta, \quad z = [\zeta \quad e \quad \dot{e}]^T, \quad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -K_i & -K_p & -K_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{13}$$

The matrix A is in controllable canonical form with characteristic polynomial

$$\det(sI - A) = s^3 + K_d s^2 + K_p s + K_i \tag{14}$$

By the Routh-Hurwitz criterion A is Hurwitz if and only if $K_p, K_d, K_i > 0$ and $K_d K_p > K_i$. The selected gains satisfy this on every axis, with $K_d K_p = 72.0$ against $K_i = 8.0$ horizontally and $K_d K_p = 140.6$ against $K_i = 15.6$ vertically.

Since A is Hurwitz, for any positive-definite symmetric Q the Lyapunov equation $A^{\mathrm{T}}P + PA = -Q$ admits a unique positive-definite symmetric P. The quadratic $V(z) = z^{\mathrm{T}}Pz$ is positive definite and radially unbounded, and along disturbance-free trajectories

$$\dot{V} = -z^T Qz \leq -\lambda_{min}(Q)\|z\|^2 < 0 \tag{15}$$

which proves exponential stability of the origin for the nominal linear outer-loop error model. Under a bounded disturbance,

$$\dot{V} \leq -\lambda_{min}(Q)\|z\|^2 + 2\|PB\|\|z\|\bar{\delta} \tag{16}$$

so the linearized error state is ISS and uniformly ultimately bounded for bounded δ. The bound is conservative and holds only while the local model and the input-feasibility assumption hold. For a constant matched disturbance such as the steady wind bias the unique equilibrium of (13) is $z^* = [\delta/K_i, 0, 0]^{\mathrm{T}}$, so an ideal unsaturated integral loop rejects it with zero asymptotic position error. The finite-horizon post-mission errors reported in section 4 additionally reflect delay, noise, reference transitions and the command envelope; their reduction is empirical rather than a direct consequence of that equilibrium.

d. Anti-windup property

Equations (9) and (10) guarantee that the translational integral state remains in the compact set defined by I_{lim} and, more importantly, that the integral does not grow in the direction that aggravates saturation. Boundedness of the complete nonlinear vehicle state additionally requires a feasible reference, bounded disturbances and sufficient control authority. These conditions are enforced in the nominal and moderate scenarios of section 3; the severe scenario deliberately violates the third one in order to expose the failure boundary.

2.7. Gain-selection procedure

The gains were selected by the following four-step procedure, executed entirely on the tuning seeds {1, 2, 3}, which are disjoint from the twenty evaluation seeds. No evaluation seed influenced any gain.

Step 1, inner loop: The attitude gains were placed for a natural frequency of 15 rad/s in roll and pitch and 10 rad/s in yaw, both with damping ratio 0.8, giving the time constants used in (12).

Step 2, outer loop: The three outer poles of each axis were placed at a single real value ω, which for the canonical form (13) gives the closed-form gains

$$K_d = 3\omega, \quad K_p = 3\omega^2, \quad K_i = \omega^3 \tag{17}$$

The time-scale-separation constraint of (12) is then applied, and it binds differently on the two axes. Horizontally the relevant inner element is the attitude loop: a five-fold separation caps ω at 2.4 rad/s, and $\omega = 2.0\ \mathrm{rad/s}$ was retained so that a margin remains against the small-angle approximation, giving the ratio of 6.0 quoted in (12). Vertically the constraint is not binding, because the relevant inner element is the 20 ms motor lag, which would permit up to 10 rad/s. The vertical channel is limited instead by the thrust envelope: sweeping ω upward showed that as soon as it exceeds 2.5 rad/s the larger integral gain drives the unprotected baseline into windup divergence in the moderate scenario, the baseline RMSE jumping from 0.22 m at 2.5 rad/s to several hundred meters at 3.0 rad/s and above. Because a diverging baseline cannot be compared with anything, $\omega = 2.5\ \mathrm{rad/s}$ was retained as the largest value at which both controllers remain comparable. The resulting gains are $K_p = [12.0, 12.0, 18.75]\ \mathrm{s}^{-2}$, $K_d = [6.0, 6.0, 7.5]\ \mathrm{s}^{-1}$ and $K_i = [8.0, 8.0, 15.625]\ \mathrm{s}^{-3}$. This rule replaces manual tuning by a single interpretable design parameter per axis and guarantees the Routh-Hurwitz condition automatically, because $K_d K_p = 9\omega^3$ always exceeds $K_i = \omega^3$.

Step 3, feed-forward weight: With everything else frozen, k_{ff} was swept from 0 to 1. Table 2 gives the result twice: once with the exact analytic reference acceleration, and once with the reference acceleration corrupted by zero-mean Gaussian noise of standard deviation $0.6\ \mathrm{m/s^2}$, which emulates the numerical differentiation a real planner performs. With an exact reference the RMSE decreases monotonically and $k_{ff} = 1$ is optimal. With a differentiated reference the curve flattens: the improvement from 0.75 to 1.0 falls to 0.8%. The selection rule adopted was the smallest weight whose RMSE lies within 1% of the best value under the noisy reference, which gives $k_{ff} = 0.75$ horizontally and, after scaling by the ratio of vertical to horizontal reference acceleration in this mission, $k_{ff} = 0.34$ vertically.

Step 4, anti-windup gain. With k_{ff} frozen, the back-calculation gain K_{aw} was swept in the moderate saturation scenario. Table 2 shows a clear and somewhat unexpected outcome: once the vector conditional-

integration rule (10) is active, adding back-calculation is redundant. It increases the maximum integral-state norm from 0.49 to 1.77 m. s and degrades the RMSE from 0.166 m to 0.218 m as K_{aw} grows from 0 to 2.4 s^2, while buying only a 2.5 percentage-point reduction in the time spent on the thrust limit. The two mechanisms address the same phenomenon by different means and interfere when combined: conditional integration removes the offending error input, whereas back-calculation lets the integral converge to the value consistent with the saturated command. $K_{aw} = 0$ was therefore selected. The term is retained in (9) for generality and is examined again in the ablation study of section 4.4.

Table 2. Gain-selection sweeps on the tuning seeds. The feed-forward sweep uses the nominal scenario; the anti-windup sweep the moderate scenario. Selected values are shown in bold

k_{ff} (horizontal)	RMSE, exact reference (m)	RMSE, differentiated reference (m)	K_{aw} (s^2)	RMSE (m)	Max. integral norm (m.s)	Time on thrust limit (%)
0.00	0.0559	0.0559	**0.0**	**0.166**	**0.488**	9.27
0.25	0.0535	0.0538	0.6	0.174	1.515	7.77
0.50	0.0513	0.0521	1.2	0.188	1.656	7.47
0.75	**0.0494**	**0.0511**	2.4	0.218	1.765	6.75
1.00	0.0477	0.0507	-	-	-	-

3. SIMULATION SETUP AND EVALUATION PROTOCOL

3.1. Plant, disturbance and measurement models

The closed-loop model is implemented in Python and integrated with an explicit Euler scheme at 250 Hz, which also carries the attitude loop; the outer loop runs at 50 Hz. Table 3 lists every physical, actuator, disturbance and measurement parameter. The mission is a 60 s closed three-dimensional route comprising a climb, an X transfer, a combined Y transfer and climb, and a return to launch. Raised-cosine segments provide a continuously differentiable position reference and hence a bounded reference acceleration.

The measurement model is specified completely, because it conditions every reported number. The noise is additive, zero mean, Gaussian and white in time, and statistically independent between axes and between channels. It is applied only to the position and linear-velocity feedback that leaves the flight controller and enters the 50 Hz companion-computer loop, that is to the six states p_x, p_y, p_z, v_x, v_y, v_z. The standard deviations are 8 mm on position and 15 mm/s on velocity, values representative of a fused real-time-kinematic and inertial solution. Attitude and body-rate feedback used by the 250 Hz inner loop is treated as noise free, because it is produced by the onboard estimator at much higher bandwidth and its residual error is absorbed into the lumped disturbance of section 2.6. Noise enters after the delay buffer, so the outer loop sees a 20 ms old and noisy state.

Table 3. Vehicle, actuator, disturbance and measurement parameters

Parameter	Value	Parameter	Value
Mass m	2.10 kg	Physical thrust capability	45.3 N (2.20 mg)
Inertia diag(I_{xx}, I_{yy}, I_{zz})	diag(0.0411, 0.0411, 0.0745) kg.m^2	Thrust envelope, nominal	25.75 N (1.25 mg)
Arm $l$ength l	0.28 m	Thrust envelope, moderate	23.69 N (1.15 mg)
Linear drag coefficient k_d	0.12 N·s/m	Thrust envelope, severe	23.28 N (1.13 mg)
Rotor azimuths β	30, 90, 150, 210, 270, 330 deg	Body-torque limits	[2.2, 2.2, 1.0] N.m
Spin directions σ	+1, -1, +1, -1, +1, -1	Tilt command limit	0.60 rad
Yaw-torque ratio k_m/k_t	0.0171 m	Plant and attitude loop	250 Hz
Per-rotor thrust range	0 to 7.55 N	Outer trajectory loop	50 Hz
Motor time constant τ_m	20 ms	Feedback transport delay	20 ms (one outer sample)
Steady wind force	[2.8, -2.0, 0] N times wind scale	Position noise, per axis	zero-mean white Gaussian, 8 mm
Gust 1	4.0 N peak at 26 s, spread 5 s^2	Velocity noise, per axis	zero-mean white Gaussian, 15 mm/s
Gust 2	3.2 N peak at 40 s, spread 4 s^2	Attitude and rate noise	not modeled (section 4.8)
Vertical sinusoid	1.0 N amplitude at 0.7 rad/s	Turbulence, per axis	zero-mean Gaussian, 0.35 N

3.2. Operating scenarios

Three scenarios differing only in the software thrust envelope and the strength of the deterministic wind are used, because the value of an anti-windup mechanism cannot be judged at a single operating point. In the nominal scenario the envelope is 1.25 mg and the wind is at its nominal strength; the thrust command sits on its limit for about 1.3% of the samples. In the moderate scenario the envelope is tightened to 1.15 mg and the deterministic wind is increased by 50%, raising the time on the limit to about 9%; both controllers

still complete the mission, so they remain directly comparable. In the severe scenario the envelope is tightened one step further, to $1.13\,\mathrm{mg}$ with the same wind. This scenario lies just beyond the stability boundary of the unprotected baseline integrator and is used to locate that boundary rather than to compare tracking accuracy. In every scenario the envelope remains far inside the physical $2.20\,\mathrm{mg}$ capability, so saturation represents a conservative software limit and not motor failure.

3.3. Benchmark controller

To quantify the accuracy-versus-complexity trade-off rather than merely assert it, the two PID variants are compared against a state-of-the-art nonlinear controller. A super-twisting sliding-mode law was chosen in preference to a first-order sliding-mode law precisely because its integral term rejects the constant wind bias exactly, so the comparison is not handed an artificial advantage. With the sliding variable $s = e_v + \lambda e_p$ the law is

$$a_{des} = a_r + \lambda e_v + k_1\sqrt{|s|}\tanh\left(\frac{s}{\varepsilon}\right) + z, \quad \dot{z} = k_2\tanh\left(\frac{s}{\varepsilon}\right) \tag{18}$$

The hyperbolic tangent replaces the discontinuous sign function so that the law is implementable at 50 Hz without chattering, ε being the boundary-layer width. The benchmark was tuned by the same protocol as the PID family: a grid search over λ, k_1 and k_2 on the same tuning seeds, minimizing mean RMSE among the stable candidates, with the grid extended until the marginal improvement fell below 1%. The selected values are $\lambda = [2.8, 2.8, 3.92]\,\mathrm{s}^{-1}$, $k_1 = [4.0, 4.0, 5.2]\,\mathrm{m}^{0.5}/\mathrm{s}$ and $k_2 = [3.6, 3.6, 5.4]\,\mathrm{m/s}^3$, with $\varepsilon = [0.30, 0.30, 0.25]\,\mathrm{m/s}$. This benchmark is deliberately given every advantage the PID family is given, and one it is not: its gains were optimized directly for tracking error, whereas the PID gains were fixed by the pole-placement rule of (17).

3.4. Evaluation protocol

All results use common random numbers. Within one seed the baseline, the proposed controller and the benchmark receive bit-identical turbulence, measurement-noise and packet-loss sequences, so every reported difference is a paired difference and the seed-to-seed variability cancels. Twenty independent evaluation seeds are used, and results are reported as mean plus or minus one sample standard deviation. Paired two-sided Student t-tests on the twenty differences are used to test significance.

Ten metrics are recorded: the three-dimensional position RMSE and its three axis components; the peak error over the mission; the mean error over the final 3.5 s, denoted the post-mission error; the altitude overshoot during the first climb; the mean collective thrust; the maximum norm of the translational integral state; and the percentage of samples for which the thrust command is on a limit. A run is declared a mission failure when the peak position error exceeds 5 m, which for this mission means that trajectory tracking has been lost. Statistics for a scenario are computed over the runs that did not fail, and the failure count is reported separately, because averaging a divergent quantity is meaningless.

4. RESULTS AND DISCUSSION

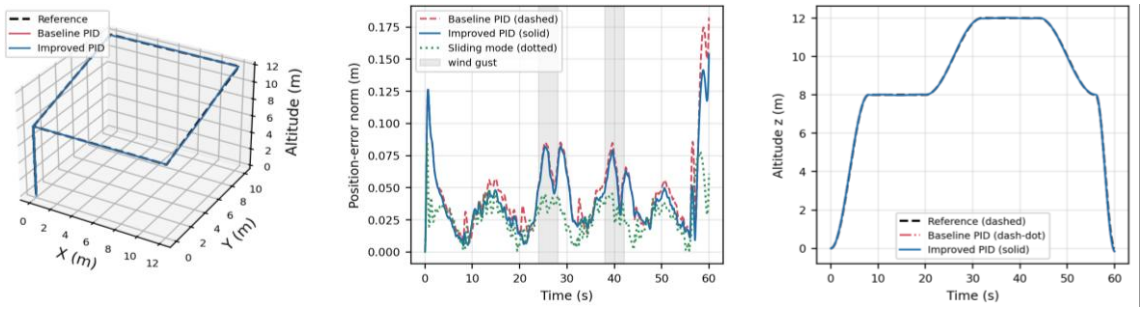
4.1. Nominal scenario

Table 4 reports the twenty-seed results for all three scenarios. Under the nominal envelope the augmentation reduces the mean three-dimensional RMSE from 0.0557 m to 0.0493 m, a reduction of 11.5% ($p < 10^{-30}$), the peak error by 15.9% and the post-mission error by 19.6%. The improvement is largest on the vertical axis, 14.9%, where the reference acceleration is largest during the climb segments. Altitude overshoot changes from 0.245% to 0.247%, a relative change of 0.7% that the paired test does not distinguish from zero ($p = 0.15$), and the mean collective thrust is unchanged to four significant figures, confirming that the improvement is not bought with additional control effort. The maximum integral-state norm is essentially identical for the two controllers, 0.4006 against 0.4000 $\mathrm{m \cdot s}$, which is exactly what should be expected: with the command on its limit for only 1.3% of the samples, there is almost no windup to prevent. Under nominal conditions the benefit therefore comes entirely from the feed-forward term.

Figure 2 shows the representative seed. Figure 2(a) confirms that both controllers follow the mission. Figure 2(b) shows that the separation between the controllers is largest during reference acceleration and during the two gusts, which is consistent with a feed-forward mechanism. Figure 2(c) shows that both controllers exhibit nearly identical altitude responses, consistent with the negligible change in altitude overshoot reported in Table 4. The integral-state history, not shown, is flat for both controllers, which again confirms that the anti-windup path is essentially inactive at this operating point.

Table 4. Twenty-seed Monte Carlo comparison in the three scenarios, reported as mean plus or minus one sample standard deviation. Percentages are the change of the improved controller relative to the matched baseline. In the severe scenario the baseline diverged in every run, so its statistics describe divergent trajectories and no percentage is quoted

Scenario	Metric	Baseline PID	Improved PID	Super-twisting SMC	Change
Nominal	3-D RMSE (m)	0.0557 ± 0.0003	0.0493 ± 0.0003	0.0313 ± 0.0002	-11.5%
	Peak error (m)	0.1898 ± 0.0076	0.1596 ± 0.0074	0.0870 ± 0.0030	-15.9%
	Post-mission error (m)	0.1202 ± 0.0012	0.0966 ± 0.0012	0.0603 ± 0.0008	-19.6%
	Altitude overshoot (%)	0.245 ± 0.020	0.247 ± 0.019	0.105 ± 0.018	+0.7% (n.s.)
	Mean thrust (N)	20.904 ± 0.003	20.903 ± 0.003	20.907 ± 0.003	0.0%
	Max. integral norm (m.s)	0.4006 ± 0.0017	0.4000 ± 0.0017	3.4964 ± 0.0334	-0.2%
	Time on thrust limit (%)	1.31 ± 0.05	1.23 ± 0.04	1.22 ± 0.06	-5.6%
	Mission failures	0 / 20	0 / 20	0 / 20	-
Moderate	3-D RMSE (m)	0.2266 ± 0.0107	0.1674 ± 0.0063	0.1370 ± 0.0073	-26.1%
	Peak error (m)	1.1770 ± 0.0659	0.9294 ± 0.0117	0.7993 ± 0.0156	-21.0%
	Post-mission error (m)	0.2145 ± 0.0029	0.1880 ± 0.0027	0.1409 ± 0.0031	-12.4%
	Max. integral norm (m.s)	2.1117 ± 0.1051	0.4850 ± 0.0035	9.2318 ± 0.0257	-77.0%
	Time on thrust limit (%)	10.50 ± 0.15	9.36 ± 0.12	10.04 ± 0.17	-10.9%
	Mission failures	0 / 20	0 / 20	0 / 20	-
Severe	3-D RMSE (m)	diverged	0.4128 ± 0.0107	0.3796 ± 0.0137	-
	Peak error (m)	diverged	2.1007 ± 0.0494	1.9807 ± 0.0622	-
	Max. integral norm (m·s)	$1.90 \times 10^4 \pm 0.47 \times 10^4$	0.5028 ± 0.0049	9.5674 ± 0.0434	-
	Time on thrust limit (%)	60.42 ± 0.14	13.09 ± 0.21	13.95 ± 0.27	-78.3%
	Mission failures	20 / 20	0 / 20	0 / 20	-



(a) (b) ©

Figure 2. Nominal scenario, representative seed: (a) Closed three-dimensional mission trajectory, (b) position-error norm versus time; the two gray bands mark the gust intervals 24.0 to 28.0 s and 38.0 to 42.0 s, and (c) altitude response. Line styles: reference black dashed, baseline PID red dashed in (b) and red dash-dot in (c), improved PID blue solid, super-twisting benchmark green dotted

4.2. Saturation-limited scenarios

The picture changes as soon as the actuator envelope tightens. In the moderate scenario the thrust command is on its limit for about 9% of the samples and the two mechanisms act together. The RMSE falls from 0.2266 m to 0.1674 m, a reduction of 26.1%, the peak error falls by 21.0%, and the maximum integral-state norm falls from 2.11 m · s to 0.49 m. s, a reduction of 77.0%. All of these are significant at $p < 10^{-12}$. The time spent on the thrust limit also falls by 10.9%, because a smaller integral state produces a smaller command. This is the operating region for which the mechanism was designed, and the size of the improvement scales with the severity of the constraint rather than being fixed.

In the severe scenario the difference is no longer quantitative. The baseline integrator diverges in twenty runs out of twenty: its mean maximum integral norm reaches 1.90×10^4 m. s and the vehicle leaves the mission volume entirely, with the thrust command pinned to its limit for 60% of the mission. The proposed controller completes all twenty runs with an RMSE of 0.4128 m and an integral norm of 0.50 m. s, and it spends only 13.1% of the mission on the limit. Figure 3 contrasts the two regimes on logarithmic axes: in the moderate case the baseline integral shows a transient peak during the first gust from which it recovers, whereas in the severe case the same peak initiates unbounded growth. The distance between these two envelopes is only 0.02 mg, roughly 0.4 N of thrust margin, which is a useful practical warning: the transition from a well-behaved to a divergent loop is abrupt, and a vehicle that is comfortable in one configuration may be one payload increment away from instability.

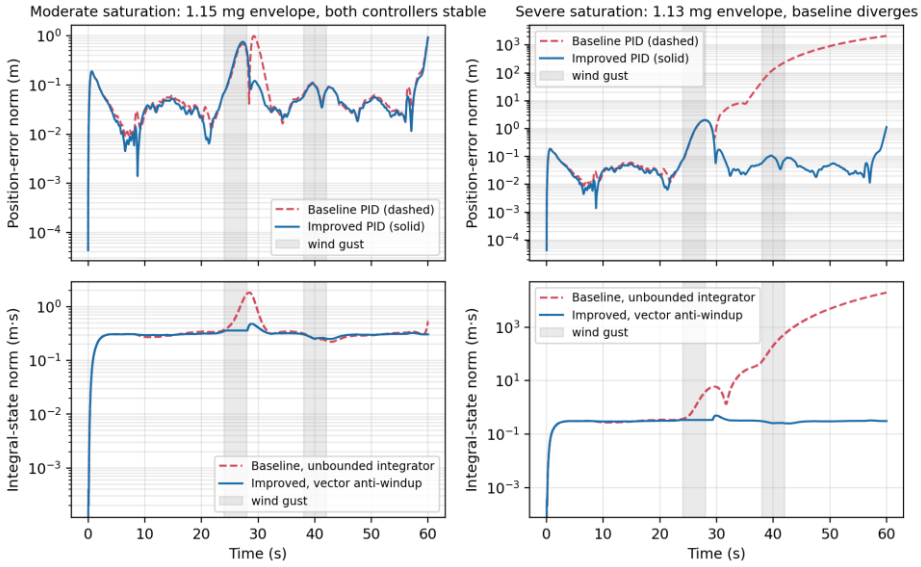


Figure 3. Behavior under actuator saturation on logarithmic axes, representative seed (red dashed line: baseline PID, blue solid line: improved PID, gray bands mark the gust intervals 24.0 to 28.0 s and 38.0 to 42.0 s)

4.3. Comparison with the sliding-mode benchmark

The tuned super-twisting controller is the most accurate of the three whenever it is stable. It reaches an RMSE of 0.0313 m in the nominal scenario, 36% below the proposed PID, and 0.1370 m in the moderate scenario, 18% below. This result is reported plainly rather than argued away: a well-tuned second-order sliding-mode law with an exact-rejection integral term is a stronger tracker than a pole-placed PID with a partial feed-forward, and the present work does not claim otherwise. What the benchmark does not provide is robustness to actuator saturation. Its super-twisting integrator is protected only by a magnitude clamp, so its integral norm reaches 9.2 m · s in the moderate scenario, nineteen times that of the proposed controller, and one run in ten is lost when the envelope is tightened to $1.12\,mg$. The benchmark also requires four tuned quantities per axis instead of one bandwidth, and its stability argument rests on a disturbance bound that must be known in advance.

4.4. Ablation study

Table 5 isolates the contribution of each element. Three findings are worth stating explicitly. First, under nominal conditions the entire benefit comes from the feed-forward term: feed-forward alone reproduces the full improvement, and both anti-windup variants are indistinguishable from the baseline. Second, and conversely, under saturation the entire benefit comes from the vector conditional-integration rule, while feed-forward alone changes almost nothing. Third, and most importantly for practice, clamping the integral state is not sufficient. In the severe scenario the clamp-only variant holds the integral norm at exactly its bound of 5.196 m. s, yet still loses the mission in ten runs out of ten. Bounding the magnitude of the integral state does not prevent it from pointing in the wrong direction; only the directional test of (10) does. Since scalar clamping is the most widely deployed anti-windup measure in practical multirotor code, this is a concrete and actionable result. The final row confirms the choice of $K_{aw} = 0$: adding back-calculation on top of the conditional rule degrades the RMSE by 9% in the moderate scenario and multiplies the integral norm by 3.4.

Table 5. Ablation of the individual mechanisms, ten seeds per cell. FF: reference-acceleration feed-forward. CI: vector conditional integration. CL: element-wise clamp. BC: back-calculation

Variant	FF	CI	CL	BC	Nominal RMSE (m)	Moderate RMSE (m)	Moderate max. I (m.s)	Severe RMSE (m)	Severe failures
Baseline PID	-	-	-	-	0.0556	0.2222	2.068	diverged	10/10
Feed-forward only	yes	-	-	-	0.0492	0.2195	2.071	diverged	10/10
Clamp only	-	-	yes	-	0.0556	0.2222	2.068	diverged	10/10
Vector anti-windup only	-	yes	yes	-	0.0558	0.1689	0.487	0.4113	0/10
Proposed	yes	yes	yes	-	**0.0492**	**0.1653**	**0.485**	**0.4101**	**0/10**
Proposed with BC, K_{aw} = 1.2	yes	yes	yes	yes	0.0494	0.1807	1.666	0.4406	0/10

4.5. Is the mechanism specific to PID?

A reasonable objection to any anti-windup result is that the mechanism may be a patch for one particular controller. To test this, the conditional-integration rule (10) was applied unchanged to the super-twisting integrator of the benchmark, with no other modification and no retuning. Table 6 shows that the rule transfers. Its cost is negligible where saturation is absent, it reduces the benchmark integral norm from 9.2 to 3.5 ms in the moderate scenario, and at the $1.12\ mg$ envelope, where the unmodified benchmark loses one run in ten, the protected version completes all ten. The mechanism is therefore an architectural property of how a saturated vector command should be fed back to an integrator, not a PID-specific correction. Below about $1.10\ mg$ no controller holds the trajectory, because the thrust required by the mission then exceeds the envelope; that limit is physical rather than algorithmic.

Table 6. The proposed conditional-integration rule applied unchanged to the sliding-mode benchmark. Six seeds per cell, except the last column which uses ten seeds at a $1.12\ mg$ envelope

Controller	Nominal RMSE (m)	Moderate RMSE (m)	Moderate max. I (m.s)	Severe RMSE (m)	Severe failures	Failures at 1.12 *mg*
Baseline PID	0.0558	0.2268	2.112	diverged	6/6	-
Improved PID	0.0493	0.1681	0.485	0.4128	0/6	0/10
Super-twisting SMC	0.0314	0.1374	9.231	0.3781	0/6	1/10
SMC with the proposed rule	0.0314	0.1465	3.851	0.3955	0/6	0/10

4.6. Computational cost and sensitivity to the data link

The claim that the augmentation is computationally negligible is quantified in Table 7 rather than asserted. Two measures are given. The hardware-independent measure is the number of scalar operations executed per outer-loop cycle, obtained by counting the additions, multiplications, divisions, comparisons and transcendental calls of each law. The augmented controller needs 76 operations against 45 for the baseline, an increase of 31, and it stores exactly the same three floating-point numbers, that is 24 bytes of persistent state. The measured wall-clock figures were obtained for the interpreted NumPy reference implementation on an x86-64 machine under Python 3.10, averaged over 30 000 cycles after discarding the top 0.5% of samples. They are dominated by array allocation and interpreter overhead rather than by arithmetic, which is why the sliding-mode law, despite needing more arithmetic, measures slightly faster than the augmented PID: it makes fewer separate array calls. The meaningful quantity is therefore the incremental cost, 5.3 microseconds per 20 ms cycle, or 0.026% of the cycle budget, since the baseline and the augmented controller share the same array machinery. A compiled ROS 2 node would execute 31 additional scalar operations in well under one microsecond on any companion computer in current use.

This cost should be read against the published cost of the alternatives. Nonlinear sliding-mode and active-disturbance-rejection laws of the type in [13], [15], [18], [24], [26] add tens to a few hundred scalar operations and one to three auxiliary states per axis, and remain comfortably real time at 50 Hz. Predictive schemes such as [1], [12], [17], [27] solve a constrained optimization over a receding horizon at every cycle; their cost grows with horizon length and constraint count, and they typically require a dedicated solver and a memory footprint several orders of magnitude larger than a PID. Learning-based controllers [19]–[21] shift the burden offline but require a trained network in the loop, tens to hundreds of kilobytes of weights, and a validation argument for behavior outside the training distribution. Against that spectrum an augmentation that adds 31 operations and no state sits at the extreme low-cost end, which is precisely its intended contribution: it can be adopted without changing the software architecture or the hardware.

Table 7. Computational cost of one outer-loop cycle. Wall-clock values are for the interpreted NumPy reference implementation and are overhead dominated; the operation count is the hardware-independent measure

Controller	Scalar operations per cycle	Persistent state (bytes)	Mean time (microseconds)	95th percentile (microseconds)	Duty cycle of the 20 ms budget (%)
Baseline PID	45	24	67.6	162.8	0.338
Improved PID	76 (+31)	24	72.8 (+5.3)	168.0	0.364
Super-twisting SMC	105	24	68.1	160.1	0.341
SMC with the proposed rule	118	24	68.7	157.6	0.344

Table 8 examines how the result depends on the companion-computer link. The transport delay was increased from 20 ms to 40 and 60 ms, and an independent Bernoulli packet loss of 0 to 10% was applied to the feedback samples, a lost sample causing the outer loop to hold its previous command. Both controllers degrade

with delay, as expected, and the relative advantage of the augmentation narrows from 11.5% at 20 ms to 8.5% at 40 ms and 6.4% at 60 ms, because feed-forward cannot compensate a delay that also affects the reference-tracking loop. Packet loss up to 10% has a very small effect at every delay, changing the RMSE by about 1% and the relative advantage by less than 0.1 percentage points, because the loop runs at 50 Hz and a single dropped sample is corrected at the next update. The improvement is therefore robust across the range of link quality that a micro-XRCE-DDS bridge realistically delivers, and it degrades gracefully rather than abruptly.

Table 8. Sensitivity to transport delay and packet loss, nominal scenario, six seeds per cell

Transport delay (ms)	Packet loss (%)	Baseline RMSE (m)	Improved RMSE (m)	Reduction (%)
20	0	0.0558	0.0493	11.54
20	5	0.0564	0.0499	11.49
20	10	0.0570	0.0505	11.43
40	0	0.0737	0.0674	8.52
40	5	0.0742	0.0679	8.55
40	10	0.0748	0.0684	8.59
60	0	0.0943	0.0883	6.37
60	5	0.0948	0.0887	6.41
60	10	0.0952	0.0891	6.46

4.7. Discussion

Taken together, the results support a narrow but useful claim. The two augmentations are complementary and act in different regimes. Reference-acceleration feed-forward buys accuracy when the actuator has margin and the mission is dynamic; it is worth about 11% of RMSE here and nothing at all when the command is saturated. Vector conditional integration buys nothing when there is margin and becomes the difference between completing and losing the mission when there is not. A designer who implements only one of them will therefore see a benefit in only one of the two regimes, which may explain why the reported value of anti-windup in the multirotor literature varies so widely: it depends almost entirely on how hard the actuator is being pushed in the reported experiments.

The comparison with the super-twisting benchmark clarifies where a computationally light PID augmentation belongs. It does not compete with a well-tuned nonlinear controller on accuracy: the results show a clear 18% to 36% gap in favor of the benchmark whenever both are stable. What it offers instead is a mechanism that requires no disturbance bound, no observer, no online optimization and no additional state, that can be added to existing PID code in a few lines, and that protects the integrator of whatever controller it is attached to, including the benchmark itself. The value proposition is robustness per unit of engineering and computational cost, not peak accuracy.

The finding that a magnitude clamp is insufficient deserves emphasis because it contradicts common practice. Clamping is attractive precisely because it gives a hard guarantee on the integral state, and Table 5 confirms that the guarantee holds: the clamp-only variant never exceeds its bound. But a bounded integral state pointing consistently in the direction that aggravates saturation still delays recovery indefinitely, and in the severe scenario the clamp-only variant lost the mission in every run, exactly as the unprotected baseline did. The directionality of the integral state, not its magnitude, is what must be controlled, and that is a property of the vector nature of the constraint that a per-axis clamp cannot see.

Hardware implications. Three consequences follow for an implementation on real hardware. The first is the computational budget. Table 7 puts the augmented outer loop at 76 scalar operations, 24 bytes of persistent state and 0.364% of the 20 ms cycle on an interpreted reference implementation; a compiled build on the ARM Cortex-A class processors carried by current companion computers would be cheaper again by at least an order of magnitude. The structure of the computation matters more than the figure itself: the law contains no matrix inversion, no iterative solver, no dynamic memory allocation and no data-dependent branching beyond the three sign tests of (10) and the six comparisons of the integral clamp, so its worst-case execution time is essentially its average and it can be placed in a hard real-time thread without a timing margin. The same property makes it implementable inside the flight-controller firmware, in the multicopter position-control module, when no companion computer is carried. The second concerns the origin of the saturation limit on a real vehicle. Here the thrust envelope was imposed as a fixed software parameter, whereas on hardware it is set by battery-voltage sag, by electronic-speed-controller current limiting and by rotor efficiency loss, all of which tighten the envelope progressively as a flight proceeds. A practical implementation should therefore schedule the envelope on the measured bus voltage and form the saturation residual from the value the mixer actually applied rather than from the value commanded; the two quantities required for this, the commanded and the applied collective thrust, are already published by standard autopilot firmware over the same uXRCE-DDS bridge [9], so the mechanism needs no additional sensing and

no change to the airframe. The third is that these unmodeled effects make the mechanism more relevant on hardware than in simulation, not less: blade flapping, induced-flow transients and voltage sag all erode the available thrust in exactly the sustained-saturation regime identified in section 4.2, which should move the failure boundary of the unprotected baseline to a looser nominal envelope than the $1.13\ mg$ found here. The delay and packet-loss sweep of Table 8 supports transferring the result to a real data link: over 20 to 60 ms of transport delay and up to 10% loss, representative of a Wi-Fi or serial-telemetry companion link, the reduction in RMSE degraded gracefully from 11.5% to 6.4% and never reversed sign.

4.8. Limitations

Simulation versus reality. All results are numerical. The model does not represent blade flapping, induced-flow transients, rotor-to-rotor aerodynamic interaction, ground and ceiling effects, battery-voltage sag under sustained high thrust, estimator drift, or the scheduling jitter of a non-real-time Linux companion computer. Each of these could alter the quantitative results, and the sustained-saturation scenarios are precisely where battery sag and thermal derating would matter most. The qualitative conclusion that an unprotected integrator diverges once the envelope is tight enough does not depend on those effects, but the numerical value of the failure boundary certainly does.

Model assumptions. The rotors are modeled as first-order lags with a single time constant and a static thrust map; the allocation matrix is assumed exactly known, so allocation error and rotor-efficiency mismatch are absent. Yaw is held at zero throughout the mission, so the coupling between yaw torque and collective thrust through the allocation matrix is never excited strongly. Measurement noise is white; a real global-navigation-satellite-system solution has colored error with multipath and outages. The transport delay is fixed rather than jittered, and packet loss is memoryless, whereas real wireless links lose packets in bursts.

Validity of the time-scale separation. The separation ratios of (12) are 6.0 horizontally and 20.0 vertically, which justify the cascaded design in the small-angle, unsaturated region. They do not hold during saturation, when the effective outer-loop gain collapses, nor at large tilt angles, where the small-angle approximation behind (11) fails. The stability results of section 2.6 are correspondingly local and say nothing about the severe scenario, whose outcome is established empirically over twenty seeds rather than proved. A global proof for the constrained nonlinear system, for example through a piecewise-quadratic Lyapunov function that accounts for the saturation region explicitly, remains open.

Scope of the comparison. One benchmark was implemented and tuned by the same protocol; a predictive or learning-based controller was not, so the cost discussion of section 4.6 is drawn from the literature rather than measured here. The single mission profile, vehicle mass and wind model also limit generality, and the reported percentages should be read as evidence for a mechanism rather than as a specification.

5. CONCLUSION

This paper presented a reproducible PID augmentation for companion-computer trajectory control of a hexarotor in a PX4/ROS 2-style two-rate architecture, and evaluated it under a strictly matched comparison protocol. The baseline and the proposed controller share pole-placed gains, gravity compensation, plant, delayed measurements and bit-identical disturbance realizations; only reference-acceleration feed-forward and a vector conditional-integration anti-windup law are added. Over twenty common-random-number Monte Carlo runs the augmentation reduced the mean three-dimensional RMSE by 11.5% and the peak error by 15.9% under a nominal thrust envelope, and by 26.1% and 21.0% respectively under a tighter envelope, where it also reduced the maximum integral-state norm by 77.0%. When the envelope was tightened by a further $0.02\ mg$ the unprotected baseline diverged in every run while the proposed controller completed every mission. The augmentation costs 31 additional scalar operations per 20 ms cycle and no additional persistent state.

Four design guidelines follow directly from the ablation and sweep data, and are the most transferable part of this work. First, size the outer loop from an explicit time-scale-separation budget rather than by trial and error: place the three outer poles at a single bandwidth ω using $K_d = 3\omega, K_p = 3\omega^2$ and $K_i = \omega^3$, and raise ω until the first binding constraint is met. Horizontally that constraint is the five-fold separation from the attitude loop; vertically, where the separation from the motor lag is never binding, it is the thrust envelope, because a large vertical integral gain will drive an unprotected integrator into divergence long before the separation limit is reached. This satisfies Routh-Hurwitz automatically and reduces tuning to one interpretable number per axis. Second, choose the feed-forward weight from a sweep performed with a numerically differentiated reference, not an analytic one; with an exact reference the sweep will always select unity and will overstate the achievable benefit. Third, do not rely on a magnitude clamp alone. The clamp-only variant respected its bound exactly and still lost every run in the severe scenario; it is

the directional test on the saturation residual, formed after gravity compensation has been removed, that prevents divergence. Fourth, do not stack back-calculation on top of vector conditional integration: the two mechanisms address the same phenomenon by different means, and combining them degraded accuracy by 9% and tripled the integral state in these experiments.

The method is most beneficial in a specific and identifiable regime: missions in which the collective-thrust command reaches its envelope for more than a few per cent of the time. In these experiments the transition lay between 1.3 per cent, where the anti-windup path is inactive and only the feed-forward term contributes, and 9 per cent, where the mechanism already carries the whole benefit; above that the benefit grows steeply and eventually becomes the difference between completing and losing the mission. Practically, this describes heavily loaded inspection platforms, flight in sustained wind, and any vehicle operating with a conservative software thrust limit for safety or certification reasons. The mechanism also proved not to be specific to PID: applied unchanged to the super-twisting benchmark it reduced that controller's integral state by a factor of 2.4 in the moderate scenario and 2.7 in the severe one, and removed its residual failure at the $1.12\ mg$ envelope.

The limitations restated in section 4.8 bound these claims. The evaluation is entirely simulation based; no hardware-in-the-loop or flight experiment was performed, and the model omits blade flapping, battery sag, estimator drift and scheduling jitter, all of which act most strongly in exactly the sustained-saturation regime where the mechanism matters most. The stability analysis is local and does not cover the saturated constrained system, whose behavior is established here empirically. The reported percentages come from one mission profile, one airframe and one wind model, and should be read as evidence for a mechanism rather than as a specification. Hardware validation is therefore required before any operational claim is made. Future work will pursue three directions: hardware-in-the-loop and flight tests on a 2 kg hexarotor with measured communication jitter and packet loss; extension of the vector conditional-integration rule from the collective-thrust magnitude constraint to the full attainable-wrench set of the allocation matrix, which would allow it to protect the attitude loop as well; and a piecewise-quadratic Lyapunov analysis that includes the saturation region explicitly, so that the failure boundary located empirically in this paper can be predicted rather than measured.

Measured against the standard technology readiness level (TRL) scale, the evidence presented here places the method at TRL 3. The critical function, integral protection under a vector magnitude constraint, has been demonstrated analytically and verified numerically in a closed-loop model that carries the elements known to govern the phenomenon: a full-rank allocation matrix, per-rotor thrust limits, first-order motor dynamics, a two-rate 50 Hz and 250 Hz architecture, a 20 ms transport delay, packet loss, wind and measurement noise, exercised over twenty paired Monte Carlo seeds and three saturation regimes with a tuned nonlinear benchmark for reference. What has not been done is what separates TRL 3 from the levels above it. No hardware element has been exercised, so the component validation in a laboratory environment that defines TRL 4 remains outstanding; hardware-in-the-loop tests with the flight-controller firmware and a real uXRCE-DDS link closing the loop would supply it. Validation in a relevant environment, TRL 5, would then require tethered and free flight of a 2 kg hexarotor under measured wind and a real battery discharge profile, which is where the unmodeled effects listed in section 4.8 would first be exposed. The two experimental steps named above map directly onto these two levels, and until at least the first of them is completed the numbers reported in this paper should be read as evidence for a mechanism rather than as a qualified performance specification.

FUNDING INFORMATION

The authors declare that no funds, grants or other support were received for this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nguyen Thi Dieu Linh		✓	✓	✓		✓	✓	✓	✓		✓		✓	
Nguyen Van Bang	✓	✓			✓					✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no competing financial or non-financial interests.

DATA AVAILABILITY

The complete simulation package that generated every number, table and figure in Section 4 accompanies this submission as a single supplementary archive, IJRA_Hexarotor_Supplementary.zip. It contains the simulation source (hexarotor_sim.py), the full parameter file (config.json), a README describing the environment, the exact commands and the seed policy, the three figure files at full resolution, and the machine-readable result files behind every table (model_report.json, tuning_sweep.json, metrics.json, monte_carlo_metrics.json, ablation.json, generalization.json, failure_boundary.json, compute_cost.json and sensitivity.json). Every stochastic quantity is drawn from a seeded generator, so re-running the package reproduces the reported values exactly. The archive will additionally be deposited in a public repository with a permanent DOI upon acceptance, and is available from the corresponding author on request in the meantime.

REFERENCES

[1] K. Panjavarnam, Z. H. Ismail, C. H. H. Tang, K. Sekiguchi, and G. G. Casas, "Model predictive control for autonomous UAV landings: A comprehensive review of strategies, applications and challenges," *The Journal of Engineering*, vol. 2025, no. 1, 2025, doi: 10.1049/tje2.70085.

[2] Z. Fang and A. V. Savkin, "Strategies for optimized UAV surveillance in various tasks and scenarios: A review," *Drones*, vol. 8, no. 5, p. 193, 2024, doi: 10.3390/drones8050193.

[3] M. Hachem, T. Miquel, M. Bronz, and C. Roos, "Trajectory optimization for fully actuated hexacopters: Enhancing maneuverability and applications," *International Journal of Micro Air Vehicles*, vol. 17, pp. 1–8, 2025, doi: 10.1177/17568293251332313.

[4] X. Xu, H. Fu, J. Zou, and T. Gui, "Anti-windup active disturbance rejection control of tiltable-rotors fully-actuated hexarotor," *International Journal of Advanced Robotic Systems*, vol. 22, no. 3, 2025, doi: 10.1177/17298806251331321.

[5] C. Rosales, C. M. Soria, and F. G. Rossomando, "Identification and adaptive PID control of a hexacopter UAV based on neural networks," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 1, pp. 74–91, 2019, doi: 10.1002/acs.2955.

[6] G. O. Pimentel, M. F. dos Santos, J. Lima, P. Mercorelli, and F. M. Fernandes, "An over-actuated hexacopter tilt-rotor UAV prototype for agriculture of precision: Modeling and control," *Sensors*, vol. 25, no. 2, p. 479, 2025, doi: 10.3390/s25020479.

[7] S. Bemposta Rosende, S. Ghisler, J. Fernández-Andrés, and J. Sánchez-Soriano, "Implementation of an edge-computing vision system on reduced-board computers embedded in UAVs for intelligent traffic management," *Drones*, vol. 7, no. 11, p. 682, 2023, doi: 10.3390/drones7110682.

[8] M. Aly *et al.*, "Real-flight-path tracking control design for quadrotor UAVs: A precision-guided approach," *Automation*, vol. 6, no. 4, p. 93, 2025, doi: 10.3390/automation6040093.

[9] PX4 Development Team, "uXRCE-DDS (PX4--ROS 2/DDS bridge)." 2026, Accessed: Aug. 07, 2026. [Online]. Available: https://docs.px4.io/main/en/middleware/uxrcedds.

[10] I. Lopez-Sanchez and J. Moreno-Valenzuela, "PID control of quadrotor UAVs: A survey," *Annual Reviews in Control*, vol. 56, p. 100900, 2023, doi: 10.1016/j.arcontrol.2023.100900.

[11] Z. Ahmed and X. Xiong, "Adaptive impedance control of multirotor UAV for accurate and robust path following," *Machines*, vol. 12, no. 12, p. 868, 2024, doi: 10.3390/machines12120868.

[12] W. Zhou, L. Zhou, T. Yuan, R. Chen, and D. Liu, "Robust performance optimization of UAV dynamic systems using MPC-PID hybrid control," *Scientific Reports*, vol. 16, no. 1, p. 2585, 2026, doi: 10.1038/s41598-025-32436-6.

[13] E. A. Gedefaw, C. M. Abdissa, and L. N. Lemma, "An improved trajectory tracking control of quadcopter using a novel sliding mode control with fuzzy PID surface," *PLoS ONE*, vol. 19, no. 11 November, p. e0308997, Nov. 2024, doi: 10.1371/journal.pone.0308997.

[14] Y. A. Admas, H. M. Mitiku, A. O. Salau, C. O. Omeje, and S. L. Braide, "Control of a fixed wing unmanned aerial vehicle using a higher-order sliding mode controller and non-linear PID controller," *Scientific Reports*, vol. 14, no. 1, p. 23139, 2024, doi: 10.1038/s41598-024-73901-y.

[15] X. Wu, Y. Pan, Q. Chen, N. Zheng, and Z. Chen, "Attitude control of a quadcopter UAV using sliding mode control with an improved extended state observer," *Electronics (Switzerland)*, vol. 14, no. 22, p. 4416, Nov. 2025, doi: 10.3390/electronics14224416.

[16] R. Mousavi *et al.*, "Observer-based adaptive neural control of quadrotor UAVs subject to model uncertainties and external disturbances," *Actuators*, vol. 13, no. 12, p. 529, 2024, doi: 10.3390/act13120529.

[17] S. Komiyama, K. Uchiyama, and K. Masuda, "Combined robust control for quadrotor UAV using model predictive control and super-twisting algorithm," *Drones*, vol. 9, no. 8, p. 576, 2025, doi: 10.3390/drones9080576.

[18] A. A. Mahmood, F. García, and A. Al-Kaff, "A novel design of a sliding mode controller based on modified ERL for enhanced quadcopter trajectory tracking," *Drones*, vol. 9, no. 11, p. 737, 2025, doi: 10.3390/drones9110737.

[19] S. Sönmez *et al.*, "Reinforcement learning-based PD controller gains prediction for quadrotor UAVs," *Drones*, vol. 9, no. 8, p. 581, 2025, doi: 10.3390/drones9080581.

[20] J. Quan, W. Hu, X. Ma, and G. Chen, "Reinforcement learning stabilization for quadrotor UAVs via Lipschitz-constrained policy regularization," *Drones*, vol. 9, no. 10, p. 675, 2025, doi: 10.3390/drones9100675.

[21] Y. Liu, L. Hao, S. Wang, and X. Wang, "Trajectory tracking controller for quadrotor by continual reinforcement learning in a wind-disturbed environment," *Sensors*, vol. 25, no. 16, p. 4895, 2025, doi: 10.3390/s25164895.

[22] K. Toudji, L. Cherroun, M. Nadour, V. Puig, and A. Hafaifa, "A comparative study between different fuzzy-based nonlinear controllers for UAV coaxial octorotor," *International Journal of Fuzzy Systems*, 2026, doi: 10.1007/s40815-025-02219-2.

[23] X. Han, H. Wang, N. Hui, and G. Yue, "Research on hierarchical composite adaptive sliding mode control for position and attitude of hexarotor UAVs," *Actuators*, vol. 14, no. 8, p. 401, 2025, doi: 10.3390/act14080401.
[24] Q. Wang, W. Wang, and S. Suzuki, "UAV trajectory tracking under wind disturbance based on novel antidisturbance sliding mode control," *Aerospace Science and Technology*, vol. 149, p. 109138, 2024, doi: 10.1016/j.ast.2024.109138.
[25] J. X. J. Bannwarth, S. Kazemi, and K. Stol, "Frequency-dependent control for wind disturbance rejection of a fully actuated UAV," *Robotica*, vol. 42, no. 6, pp. 1781–1795, 2024, doi: 10.1017/S0263574724000523.
[26] M. Ramírez-Neria, A. Luviano-Juárez, J. González-Sierra, R. Ramírez-Juárez, J. Aguerrebere, and E. G. Hernandez-Martinez, "Active disturbance rejection control for the trajectory tracking of a quadrotor," *Actuators*, vol. 13, no. 9, p. 340, 2024, doi: 10.3390/act13090340.
[27] M. A. Sadi, A. Jamali, and A. M. N. Abang Kamaruddin, "Optimizing UAV performance in turbulent environments using cascaded model predictive control algorithm and Pixhawk hardware," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 47, no. 8, p. 396, 2025, doi: 10.1007/s40430-025-05693-9.
[28] O. K. Soyinka, M. N. Ikpaya, and L. Luka, "Study on PID gain parameter optimization for a quadcopter under static wind turbulence using bio-inspired algorithms," *Discover Electronics*, vol. 2, p. 8, 2025, doi: 10.1007/s44291-025-00049-y.
[29] J. Yoon, M. Kim, J. Bang, S. Kim, and J. Doh, "Improvement of hovering stability for UAVs under crosswinds via evolutionary learning-based optimal PID control," *Journal of Mechanical Science and Technology*, vol. 39, no. 4, pp. 2151–2162, 2025, doi: 10.1007/s12206-025-0338-7.
[30] S. Bazzocchi, S. Warwick, and A. Suleman, "Automatic autopilot tuning framework using genetic algorithms and system identification," *Aerospace Science and Technology*, vol. 157, p. 109779, 2025, doi: 10.1016/j.ast.2024.109779.

BIOGRAPHIES OF AUTHORS



Nguyen Thi Dieu Linh received the Ph.D. degree from Harbin Institute of Technology, China, in 2013. She is Deputy Head of the Department of Science and Technology, Hanoi University of Industry, Vietnam. Her research interests include electronics, the Internet of Things, smart systems and telecommunications. She can be contacted at nguyen.linh@haui.edu.vn, via ResearchGate profile at https://www.researchgate.net/profile/Nguyen-Linh-16, and a personal website at https://sites.google.com/view/nguyenthidieulinh/home.



Nguyen Van Bang is a researcher and lecturer in the Department of Missile, Air Defence-Air Force Academy, Vietnam. He received the Ph.D. degree in Control Engineering and Automation in 2019. His research interests include guidance and control, unmanned aerial vehicles, nonlinear and robust control, fault-tolerant control and intelligent optimization. He can be contacted at banghvpkkq@gmail.com.