

<https://doi.org/10.31891/csit-2026-3-6>

Mykola SUSLO

PhD Student, Department of Information and Computer Systems
Chernihiv Polytechnic National University,
Chernihiv, Ukraine

<https://orcid.org/0009-0006-6692-1234>

e-mail: nikolaysuslo0@gmail.com

Volodymyr KAZYMYR

Doctor of Technical Sciences, Professor,
Department of Information and Computer Systems

Chernihiv Polytechnic National University,
Chernihiv, Ukraine

<https://orcid.org/0000-0001-8163-1119>

e-mail: vkazymyr@gmail.com

Received: 22/07/2026

Accepted: 04/09/2026

Published: 30/09/2026

© Copyright
2026 by the author(s)



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

UDC 681.5:629.7.05

ADAPTIVE PID CONTROL OF UAV POSITION USING VISUAL DISPLACEMENT

Visual-displacement-based control of unmanned aerial vehicles (UAVs) requires the position controller to operate directly on image-space errors while adapting to changes in motion conditions and limited control authority. This study presents an adaptive PID gain-scheduling method for UAV route following using visual displacement estimated from a downward-facing camera. The controller operates in image pixel space and adjusts the PID gains according to the current motion condition, visual-positioning confidence, route progression, cross-track error, and control-signal saturation. Saturation is handled independently for the two image-plane axes, allowing the affected axis to use a saturation-limited gain schedule while the other axis retains the configuration corresponding to the underlying motion condition. The proposed method was implemented and evaluated in a Gazebo/ArduPilot Software-in-the-Loop environment and compared with a fixed-gain PID controller. Preliminary development tests also included triangular and zigzag route geometries, but these runs were excluded from the final statistical validation. The final validation dataset consisted of 100 experimental runs, with 50 runs performed for each controller mode under the same predefined square route and simulation configuration. Performance was evaluated using route-completion rate, trajectory-error measures, cross-track error, route-completion time, and control-signal saturation occupancy. Route-completion proportions were compared using Fisher's exact test, while continuous run-level metrics were analysed using Mann-Whitney U tests with Holm correction for multiple comparisons. The adaptive controller completed 46 of 50 validation runs, corresponding to a route-completion rate of 92%, whereas the fixed-gain controller completed 35 of 50 runs, corresponding to 70%. The difference in completion proportions was statistically significant. For successfully completed routes, no statistically significant differences were observed after Holm correction in the principal trajectory-error measures. Mean control-signal saturation occupancy decreased from 52.60% for the fixed-gain controller to 48.00% for the adaptive controller, and this difference remained statistically significant after correction for multiple comparisons. The results indicate that, under the investigated simulation conditions, the proposed adaptive PID gain-scheduling method was associated with a higher route-completion rate and reduced operation at the control limits without statistically significant deterioration in observed trajectory-following accuracy. The conclusions are limited to the investigated square-route validation and Software-in-the-Loop environment and require further evaluation using additional route geometries, disturbances, and hardware-based UAV experiments.

Keywords: adaptive PID control; unmanned aerial vehicle; visual displacement; visual navigation; Gazebo; ArduPilot

Introduction

Unmanned aerial vehicles (UAVs) are widely used in monitoring, inspection, mapping, object search, infrastructure observation, and route-following tasks in environments where direct human involvement is undesirable or impractical. In multi-agent UAV systems, an individual vehicle acts as an autonomous agent that must perform local motion control while remaining a component of a higher-level coordinated scenario. Recent reviews of UAV formation-control approaches indicate that the effectiveness of such systems depends not only on high-level coordination but also on the accuracy and robustness of the local control performed by each individual vehicle [1].

PID controllers remain widely used for UAV position control and stabilization. Their main advantages are structural simplicity, relatively low computational complexity, and suitability for real-time implementation. However, a controller with fixed gains does not necessarily provide the same control quality during different stages of motion. During active motion toward a target point, a rapid response to the control error is

required, whereas near the target, reducing overshoot, suppressing oscillations, and maintaining accurate stabilization become more important.

An additional challenge arises when position control relies on visual information. When a UAV is controlled using a downward-facing camera, the control error may be estimated in image pixel coordinates rather than directly in metric coordinates. In this case, control performance depends on the reliability of visual feature matching, surface texture, illumination conditions, flight altitude, and the stability of the displacement-estimation algorithm. Consequently, fixed PID gains may provide limited flexibility when the operating conditions or reliability of visual positioning vary during motion.

The aim of this study is to develop and experimentally evaluate an adaptive PID controller with phase-based gain scheduling for UAV position control using visual displacement and to compare it with a fixed-gain PID controller under identical simulation conditions.

Related Work

Research on UAV control covers several interconnected areas, including local stabilization, trajectory tracking, adaptive control, visual navigation, and simulation-based evaluation. In multi-agent systems, particular attention is also given to formation control, decentralized interaction, and coordination between individual vehicles [1]. However, the overall performance of such systems remains strongly dependent on the accuracy and robustness of the local control of each UAV.

PID control remains one of the fundamental approaches to autonomous quadrotor control because of its relatively simple structure and practical applicability [2]. Several studies have therefore investigated adaptive modification of PID parameters. Melo et al. [3] proposed a fuzzy gain-scheduling PID approach for UAV position and altitude control and demonstrated the potential benefits of adjusting controller gains instead of maintaining fixed parameters. A related approach was presented by Boubakir et al. [4], who considered a robust adaptive PID-like controller for quadrotor UAV systems. PID autotuning for multicopters has also been investigated under variations in vehicle dynamics [5].

These studies demonstrate the relevance of adaptive gain adjustment for UAV control. However, their primary focus is on vehicle dynamics, controller autotuning, or general trajectory control. The present study considers a more specific problem: UAV position control in image pixel space using displacement estimates obtained from a downward-facing camera.

Another relevant research direction is vision-based positioning and navigation. Conte and Doherty [6] investigated UAV navigation based on visual information in situations where GPS data are unavailable or insufficiently reliable. Visual odometry and simultaneous localization and mapping (SLAM) provide alternative approaches for estimating the motion of mobile robots from sequences of images [7]. Vision-based servoing has also been applied directly to UAV control tasks [8].

Local visual features can be used to estimate displacement between consecutive or reference images. One computationally efficient approach is Oriented FAST and Rotated BRIEF (ORB), which combines FAST keypoint detection with a compact binary feature descriptor [9]. Such characteristics make ORB suitable for applications in which computational efficiency is important. In the present study, however, the detailed feature-detection and matching procedure is not treated as a separate contribution; the resulting visual displacement estimate is considered an input to the proposed control method.

Simulation-based evaluation constitutes a third relevant research direction. Gazebo provides an open-source environment for multi-robot simulation and enables repeatable experiments under controlled conditions [10]. For UAV research, physical simulation can also be combined with autopilot software. For example, FlyNetSim integrates ArduPilot with the ns-3 network simulator to support joint simulation of UAV operation and communication networks [11].

The present study therefore focuses on the intersection of these areas: adaptive PID position control in route-defined image pixel space using visual displacement from a downward-facing camera, evaluated quantitatively against a fixed-gain baseline.

A closely related earlier study by Suslo and Kazymyr [12] investigated visual PID-based quadcopter position stabilization under GPS-denied conditions using image-derived displacement. The present study uses the visual-displacement estimate as an established feedback signal but addresses a different research question: phase-based online adaptation of PID gains and its run-level statistical comparison with a fixed-gain baseline. Thus, the contribution evaluated here is the adaptive gain-scheduling mechanism rather than the underlying visual feature-detection and displacement-estimation procedure.

Proposed Adaptive PID Control Method

The control method considered in this study is intended for local UAV route following using position information derived from visual displacement in image pixel space. Images captured by a downward-facing camera are used to estimate the displacement of the UAV relative to a reference, and the resulting visual information is transformed into the position-error signal supplied to the controller. Both FIXED_PID and ADAPTIVE_PID operate on the same visual-displacement input and use the same position-error calculation.

In the FIXED_PID mode, the proportional, integral, and derivative gains remain constant throughout route execution. In the ADAPTIVE_PID mode, the controller gains are recalculated during each control cycle according to the current operating condition. The adaptive mechanism considers the magnitude and dynamics of the position error, the confidence of the visual-displacement estimate, proximity to the current target, and control-signal saturation. Based on these quantities, the controller identifies the current motion phase and determines separate PID gains for the two image-plane axes.

Five operating phases are considered: APPROACH, BRAKING, STABILIZATION, LOW_CONFIDENCE, and SATURATION_LIMITED. The phase is not treated as a persistent state with predefined transitions; instead, it is reevaluated at every control update according to a fixed priority rule. The selected phase defines the basic gain schedule, after which the gains are additionally scaled according to the relative error contribution of each axis, cross-track error, and route progression. Gain values are constrained to predefined ranges before being applied to the controller.

Control-signal saturation is handled independently for the two axes. When saturation occurs on one axis, the saturation-limited gain schedule is applied only to the affected axis, while the other axis can retain the gain schedule corresponding to the underlying motion condition. Integral action is suppressed on saturation-limited axes and under low visual confidence. This axis-specific treatment is intended to reduce operation at the control limits without unnecessarily restricting control authority in the unaffected direction.

The following subsections define the visual-displacement-based control error, the fixed-gain baseline, the motion-phase identification rules, the adaptive gain-scheduling law, confidence and gain constraints, and the complete adaptive control procedure.

Visual Displacement-Based Control Error

The position-control loop uses visual displacement estimated from images acquired by a downward-facing camera. At discrete control step (k), the estimated displacement relative to the reference image is represented as

$$d_k = \begin{bmatrix} d_{x,k} \\ d_{y,k} \end{bmatrix}$$

where d_k denotes the visual displacement vector at the k discrete control step, and $d_{x,k}$ and $d_{y,k}$ are its horizontal and vertical components expressed in image pixels, respectively.

For the current route target, the desired displacement is defined as

$$d_k^* = \begin{bmatrix} d_{x,k}^* \\ d_{y,k}^* \end{bmatrix}$$

Here, d_k^* denotes the desired visual displacement vector for the current route target, while $d_{x,k}^*$ and $d_{y,k}^*$ are its corresponding components along the image-plane axes.

The corresponding visual-displacement error is

$$e_k = d_k^* - d_k = \begin{bmatrix} e_{x,k} \\ e_{y,k} \end{bmatrix}$$

with

$$e_{x,k} = d_{x,k}^* - d_{x,k}, \quad e_{y,k} = d_{y,k}^* - d_{y,k}$$

Here, e_k is the visual-displacement error vector, and $e_{x,k}$ and $e_{y,k}$ denote its components along the x- and y-axes, respectively.

The scalar error magnitude used to characterize the current displacement from the target is calculated using the Euclidean norm,

$$e_k = \|e_k\|_2 = \sqrt{e_{x,k}^2 + e_{y,k}^2}$$

Because both the measured and desired displacements are represented in image pixel space, the resulting error quantities are also expressed in pixels. They therefore provide a consistent relative measure for controller operation and comparison between experimental runs but do not directly represent metric position errors without camera calibration and altitude-dependent geometric scaling.

The visual-displacement error is used directly as the position-control input. Its temporal evolution is additionally used by the adaptive controller to determine the current operating condition. For each image-plane axis $j \in \{x, y\}$, the discrete error derivative is evaluated as

$$\dot{e}_{j,k} = \frac{e_{j,k} - e_{j,k-1}}{t_k - t_{k-1}}$$

where t_k denotes the time of the current control update and $\dot{e}_{j,k}$ represents the rate of change of the controller error along axis j .

The visual-estimation subsystem also provides a normalized confidence measure $c_k \in [0,1]$ where larger values indicate greater confidence in the current visual-displacement estimate. This quantity does not alter the definition of the position error itself but is used by the adaptive mechanism when selecting the motion phase and determining the corresponding controller gains.

Fixed-Gain PID Controller

The FIXED_PID mode was used as the baseline controller for the experimental comparison. It operates independently along the two image-plane axes using the visual-displacement error components $e_{x,k}$ and $e_{y,k}$.

For each control axis $a \in \{x, y\}$, the controller output is represented as

$$u_{a,k} = P_{a,k} + I_{a,k} + D_{a,k}$$

where $u_{a,k}$ is the position-controller output at control step k , and $P_{a,k}$, $I_{a,k}$, and $D_{a,k}$ are the proportional, integral, and derivative contributions, respectively.

The proportional contribution is calculated as

$$P_{a,k} = K_p e_{a,k}$$

where K_p is the proportional gain and $e_{a,k}$ is the controller error along axis a .

The integral contribution accumulates the controller error over time and is represented as

$$I_{a,k} = I_{a,k-1} + K_i e_{a,k} \Delta t_k$$

where $I_{a,k-1}$ denotes the integral contribution accumulated up to the previous control step, K_i is the integral gain, and Δt_k is the elapsed time between consecutive control updates, defined as

$$\Delta t_k = t_k - t_{k-1}$$

The derivative contribution is represented as

$$D_{a,k} = K_d \mathcal{D}_{a,k}$$

where K_d is the derivative gain and $\mathcal{D}_{a,k}$ denotes the discrete derivative estimate obtained from the temporal variation of the controller error along axis a .

The same fixed gains were applied to both image-plane axes throughout every FIXED_PID run:

$$K_p = 1.2, \quad K_i = 0.05, \quad K_d = 0.8$$

These gain values remained constant throughout each FIXED_PID run. No online gain adjustment was performed in response to error magnitude, error dynamics, visual-positioning confidence, motion phase, route progression, or control-signal saturation.

Motion Phase Identification

The motion phase ϕ_k is determined at each control step using a fixed-priority decision rule based on control-signal saturation, visual-confidence, target proximity, error magnitude, and error dynamics:

$$\phi_k = \begin{cases} \text{SATURATION_LIMITED}, & s_{x,k} \vee s_{y,k} \\ \text{LOW_CONFIDENCE}, & c_k < 0.55 \\ \text{STABILIZATION}, & n_k = 1 \vee e_k \leq 8 \\ \text{BRAKING}, & B_k = 1 \\ \text{APPROACH}, & \text{otherwise} \end{cases}$$

where $s_{x,k}$ and $s_{y,k}$ denote the saturation indicators for the x- and y-axis control channels, respectively; c_k is the normalized visual-confidence measure; n_k is a binary indicator that equals 1 when the UAV is identified as being near the current target; e_k is the current error magnitude; and B_k is the braking indicator. The low-confidence threshold was set to 0.55, while the stabilization error threshold was set to 8 px.

The conditions in the phase-selection rule above are evaluated in the order shown, so SATURATION_LIMITED has the highest priority, while APPROACH is selected only when none of the preceding conditions is satisfied.

If none of these conditions is satisfied, the braking condition is evaluated as

$$B_k = \begin{cases} 1, & |\dot{e}_{x,k}| \geq 40 \vee |\dot{e}_{y,k}| \geq 40 \vee \dot{e}_{x,k} \dot{e}_{y,k} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

where $\dot{e}_{x,k}$ and $\dot{e}_{y,k}$ are the rates of change of the visual-displacement error components. The braking threshold was set to 40.

Adaptive Gain-Scheduling Law

After the motion phase has been identified, ADAPTIVE_PID determines the controller gains separately for the two image-plane axes. The gain-scheduling procedure combines phase-dependent base gains with an axis-dependent scaling factor that accounts for the current route direction, route-step progression, the relative error along each axis, and the cross-track error.

The adaptive controller uses the base gains $K_{p,b} = 1.2, K_{i,b} = 0.02, K_{d,b} = 0.60$. The adaptive base gains define the reference values used by the gain scheduler and should not be interpreted as constant controller gains, because the effective gains are subsequently modified according to the selected motion phase and axis-specific scaling. For the APPROACH, BRAKING, and STABILIZATION phases, the resulting phase-dependent base gain vector is

$$(\bar{K}_{p,\phi}, \bar{K}_{i,\phi}, \bar{K}_{d,\phi}) = \begin{cases} (1.32, 0.005, 0.60), & \phi_k = \text{APPROACH} \\ (1.14, 0.007, 0.57), & \phi_k = \text{BRAKING} \\ (0.72, 0.011, 0.54), & \phi_k = \text{STABILIZATION} \end{cases}$$

where $\bar{K}_{p,\phi}, \bar{K}_{i,\phi}, \bar{K}_{d,\phi}$ denote the proportional, integral, and derivative base gains associated with the selected

motion phase ϕ_k . The LOW_CONFIDENCE and SATURATION_LIMITED phases apply additional confidence- and saturation-dependent modifications and are therefore treated separately from this nominal motion-phase schedule.

To account for route progression, a route-step factor is calculated as

$$\gamma_k = 1 + 0.02 \min(\max(j_k, 0), 4)$$

where j_k is the current route-step index. The factor increases by 0.02 per route step and is capped after four steps.

The route-step factor is then combined with the current completion-axis designation to obtain separate axis-scaling factors:

$$(A_{x,k}, A_{y,k}) = \begin{cases} (1.10\gamma_k, 1.20), & q_k = x, \\ (1.20, 1.10\gamma_k), & q_k = y, \\ (\gamma_k, \gamma_k), & q_k = xy. \end{cases}$$

where $q_k \in \{x, y, xy\}$ denotes the completion-axis designation for the current route step, while $A_{x,k}$ and $A_{y,k}$ are the corresponding scaling factors for the x- and y-axis controllers. The primary-axis gain factor was set to 1.10 and the cross-track-axis gain factor to 1.20.

For each axis $a \in \{x, y\}$, the relative contribution of the current axis error and cross-track error is normalized with respect to the dominant error quantity:

$$D_{a,k} = \max(|e_{a,k}|, |e_k|, |e_{ct,k}|, 1)$$

where $e_{ct,k}$ is the current cross-track error and $D_{a,k}$ is the normalization denominator.

The normalized axis-error and cross-track-error components are then defined as

$$r_{a,k} = \min\left(\frac{|e_{a,k}|}{D_{a,k}}, 1\right), \quad r_{ct,k} = \min\left(\frac{|e_{ct,k}|}{D_{a,k}}, 1\right)$$

and combined into the axis-specific gain multiplier

$$M_{\{a,k\}} = A_{\{a,k\}}(0.8 + 0.2r_{\{a,k\}} + 0.1r_{\{\mathrm{ct},k\}})$$

here $r_{a,k}$ and $r_{ct,k}$ denote the normalized axis-error and cross-track-error components, respectively. In the final controller configuration, the base scaling factor was 0.8, the axis-error weight was 0.2, and the cross-track-error weight was 0.1.

For the nominal motion phases, the resulting scheduled gains before application of the final gain constraints are

$$\hat{K}_{p,a,k} = \bar{K}_{p,\phi} M_{a,k}, \quad \hat{K}_{i,a,k} = \bar{K}_{i,\phi} M_{a,k}, \quad \hat{K}_{d,a,k} = \bar{K}_{d,\phi}$$

The proportional and integral gains therefore vary independently between the two image-plane axes, whereas the derivative gain is determined by the selected motion phase. Although the implementation supports additional continuous derivative-gain scaling based on the error derivative, this mechanism was disabled in the final ADAPTIVE_PID configuration by setting its scaling coefficient to zero. Error dynamics therefore affect the controller through motion-phase identification but do not continuously modify the derivative gain within a phase.

Confidence Scaling and Gain Constraints

The nominal gain schedule is modified when the confidence of the visual displacement estimates decreases or when the available control authority becomes limited. These conditions affect the proportional, integral, and derivative gains differently and are handled independently for the two image-plane control axes.

When the visual-confidence measure falls below the threshold $c_k < 0.55$, the LOW_CONFIDENCE phase is selected. For each axis $a \in \{x, y\}$, the corresponding gains are

$$K_{p,a,k} = \text{clip}(1.2(0.55 + 0.25c_k)M_{a,k}, 0, 1.68), \quad K_{i,a,k} = 0, \\ K_{d,a,k} = \text{clip}(0.60[1.10 + 0.10(1 - c_k)], 0, 0.85)$$

where $M_{a,k}$ is the axis-specific gain multiplier. In this phase, the proportional gain decreases with decreasing visual confidence, while the integral gain is set to zero. The derivative gain increases moderately as confidence decreases. Thus, uncertain visual-displacement estimates do not contribute to integral accumulation.

The gains are constrained to predefined intervals:

$$0 \leq K_{p,a,k} \leq 1.68, \quad 0 \leq K_{i,a,k} \leq 0.05, \quad 0 \leq K_{d,a,k} \leq 0.85$$

where $\text{clip}(z, l, u)$ limits z to the interval $[l, u]$. The maximum proportional, integral, and derivative gains were set to 1.68, 0.05, and 0.85, respectively.

Control-signal saturation is detected independently for the two control channels. In the final configuration, a channel was considered saturated when the magnitude of its control adjustment reached 98% of the available maximum correction. Saturation hysteresis was disabled; therefore, the saturation state followed the current threshold condition directly.

For a saturation-limited axis, the resulting gains are

$$K_{p,a,k} = 0.95 \text{ clip}(0.90M_{a,k}, 0, 1.68), \quad K_{i,a,k} = 0, \quad K_{d,a,k} = 0.85$$

where the proportional gain is reduced after axis-specific error scaling, the integral gain is set to zero, and the derivative gain reaches its configured upper limit of 0.85. Integral accumulation is also clamped on the saturated

axis. Under LOW_CONFIDENCE, integral accumulation is suppressed on both axes because the integral gain is zero.

These constraints ensure that the adaptive scheduler remains within the predefined gain ranges while allowing confidence and saturation conditions to modify the controller response independently of the nominal APPROACH, BRAKING, and STABILIZATION schedules.

At each control update, the current motion phase and axis-specific scaling factors are recalculated from the visual-displacement error, error dynamics, visual confidence, route information, cross-track error, and saturation state. The corresponding gains are then determined independently for the two image-plane axes and constrained to the predefined ranges. If an axis is saturated, the SATURATION_LIMITED configuration is applied only to that axis, while the other axis retains the gain schedule associated with the underlying motion condition. Integral accumulation is suppressed under saturation and low visual confidence. The resulting axis-specific PID gains are supplied to the position controller for the current control cycle.

Evaluation Metrics

Controller performance was evaluated using run-level measures of route completion, trajectory-following error, cross-track deviation, route-completion time, and control-signal saturation. Each experimental run was treated as a single experimental unit, and samples recorded within the same run were not considered independent observations.

For experimental run r , let N_r denote the number of valid trajectory samples, where $k = 1, \dots, N_r$, and let $e_{r,k}$ denote the visual-displacement error magnitude at sample k . For each run, the mean error, root mean square error (RMSE), 95th-percentile error, and final error were calculated from the corresponding visual-displacement error samples. The RMSE was defined as

$$\text{RMSE}_r = \sqrt{\frac{1}{N_r} \sum_{k=1}^{N_r} e_{r,k}^2},$$

The mean error represents the arithmetic mean of $e_{r,k}$, the 95th-percentile error represents the value below which 95% of the recorded error samples fall, and the final error is the error magnitude at the last valid trajectory sample. All trajectory-error quantities are expressed in pixels.

Cross-track error characterizes deviation from the prescribed route rather than distance from the current target point. Let $d_{r,k}$ denote the minimum distance, in image pixels, between the current trajectory position and the corresponding segment of the prescribed route. The mean cross-track error for run r was calculated as

$$\bar{d}_r = \frac{1}{N_r} \sum_{k=1}^{N_r} d_{r,k}$$

Route completion was treated as a binary run-level outcome. The route-completion rate was calculated as the percentage of experimental runs in which all prescribed route steps were successfully completed. Route-completion time was measured in seconds for successfully completed runs and represents the elapsed time required to complete the prescribed route.

Control-signal saturation was quantified independently for each run. Let N_r^c denote the number of valid position-control updates in run r , while $s_{x,r,k}$ and $s_{y,r,k}$ denote binary saturation indicators for the x- and y-axis control channels, respectively, at control update k . The saturation occupancy was calculated as

$$S_r = \frac{100}{N_r^c} \sum_{k=1}^{N_r^c} \mathbb{I}(s_{x,r,k} \vee s_{y,r,k})$$

where $\mathbb{I}(\cdot)$ is the indicator function. Thus, S_r represents the percentage of valid position-control updates during which at least one control channel operated at the configured saturation limit.

Implementation

The proposed control method was implemented in a Software-in-the-Loop (SIL) experimental environment integrating Gazebo and ArduPilot SIL. Gazebo simulated the UAV dynamics, operating environment, and downward-facing camera, while the external position-control software communicated with ArduPilot through the Micro Air Vehicle Link (MAVLink) protocol using the pymavlink library.

Images from the simulated downward-facing camera were processed to obtain the visual-displacement estimate used as the position-feedback signal. The same visual-displacement and position-error calculation pipeline was used for both controller modes, so the comparison between FIXED_PID and ADAPTIVE_PID differed only in the PID gain-selection mechanism. The resulting axis-specific PID outputs were converted into roll and pitch control commands and transmitted to ArduPilot, closing the control loop with the simulated UAV.

Experimental data were recorded for each run in comma-separated values (CSV) files. The logs contained trajectory, visual-estimation, controller, control-signal, saturation, and route-completion data required to calculate the run-level performance metrics and perform the statistical comparison between the two controller modes. The principal components and data flow of the experimental implementation are shown in Fig. 1.

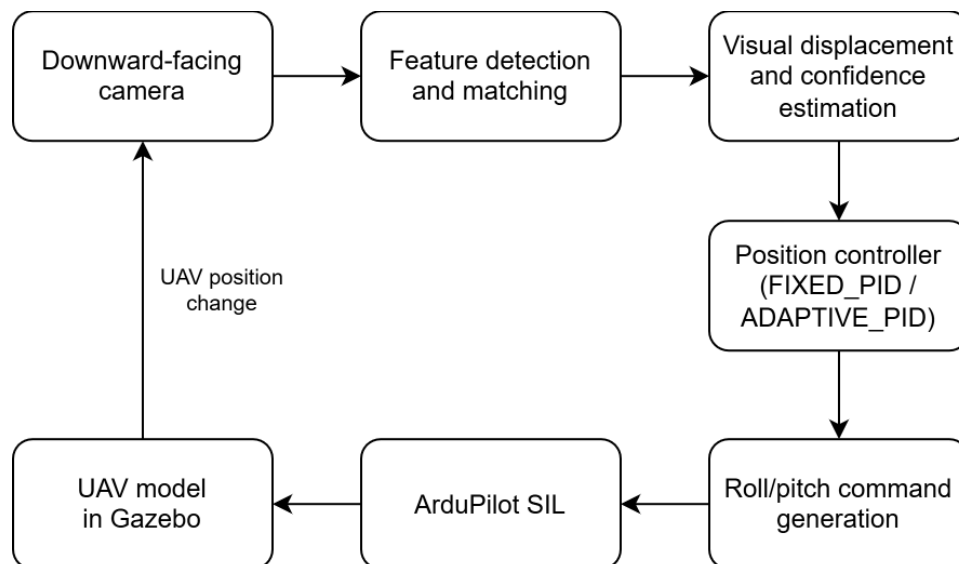


Fig. 1 Closed-loop structure of UAV position control based on visual displacement

Experimental Evaluation

The experimental evaluation was organized to separate controller configuration and tuning from the final validation experiments. The following sections describe the experimental design, statistical analysis, and results obtained from the final experimental dataset.

Experimental Design

The experimental study compared FIXED_PID and ADAPTIVE_PID under the same simulation conditions. The controller modes differed only in the PID gain-selection mechanism, while the route, simulation configuration, and data-acquisition procedure remained unchanged.

The final statistical validation was performed on a single predefined square route. Corresponding pseudorandom seed identifiers were used for the two controller modes to improve run-to-run comparability, although identical seed identifiers were not assumed to make Gazebo/ArduPilot SIL executions fully deterministic. Preliminary development tests also included triangular and zigzag route geometries; these runs were used only for controller development and were excluded from the final validation dataset.

Each experimental run was treated as a single experimental unit. Samples recorded within the same run formed one temporal trajectory and were not considered independent observations. Run-level metrics were calculated from these samples and used for the statistical comparison. The main parameters of the final experimental design are summarized in Table 1.

Table 1.

Experimental design

Parameter	Value
Simulation environment	Gazebo / ArduPilot SIL
Control modes	FIXED_PID, ADAPTIVE_PID
Runs per control mode	50
Total number of validation runs	100
Validation route geometry	Square
Route size	1000 × 1000 px
Route step	100 px
Number of route steps	40
Position-feedback source	Visual displacement from a downward-facing camera
Position-error units	Pixels
Seed sets	10001-10030; 40001-40020
Experimental unit	One experimental run

Controller parameters were selected during preliminary development runs and fixed before the final

validation series. These development runs were excluded from the statistical analysis. The adaptive configuration was refined using route completion, trajectory behaviour, control-signal saturation, and route-completion time as diagnostic criteria. The selected FIXED_PID and ADAPTIVE_PID configurations were then applied unchanged throughout all final validation runs.

Each validation run consisted of a complete route-following trial using either FIXED_PID or ADAPTIVE_PID. A run was classified as successfully completed when all prescribed route steps were executed and the route-completion event was recorded by the mission controller. Runs that did not complete the route within the maximum execution time of 480 s were classified as incomplete.

Statistical Analysis

Route-completion proportions were compared using a two-sided Fisher's exact test. Continuous run-level metrics were compared between FIXED_PID and ADAPTIVE_PID using two-sided Mann–Whitney U tests to avoid reliance on normality assumptions.

Trajectory-error metrics, mean cross-track error, route-completion time, and saturation occupancy were analysed for successfully completed runs only. Saturation occupancy was treated separately from the trajectory-accuracy measures because it characterizes controller activity rather than trajectory-following error.

The significance level was set to $\alpha = 0.05$. Holm's correction was applied to the family of seven continuous-metric comparisons to control for multiple testing. Unadjusted and Holm-adjusted p -values were retained for reporting.

Experimental Results

The route-completion results are summarized in Table 2.

Table 2.

Route-completion results		
Outcome	FIXED_PID	ADAPTIVE_PID
Successfully completed	35	46
Incomplete	15	4
Completion rate, %	70	92

The route-completion proportion was 22 percentage points higher for ADAPTIVE_PID. A two-sided Fisher's exact test indicated a statistically significant difference between the two control modes $p = 0.009$. Representative individual completed trajectories for the two controller modes are shown in Fig. 2.

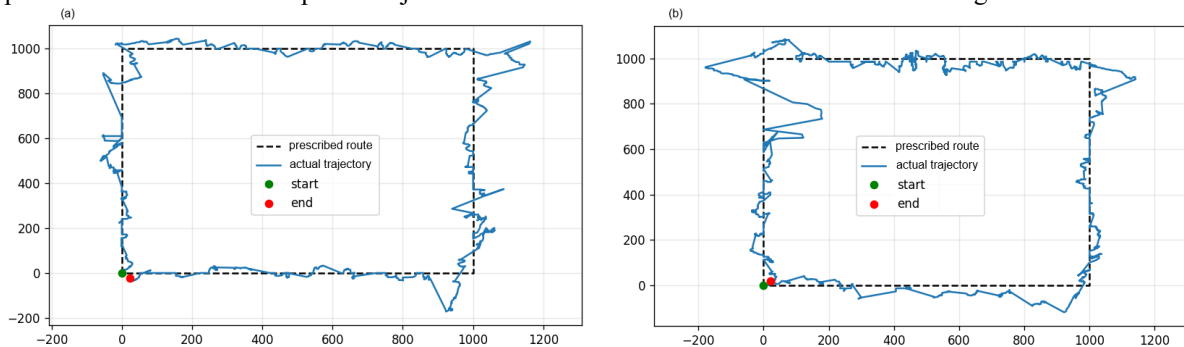


Fig. 2 Representative individual completed trajectories for (a) ADAPTIVE_PID and (b) FIXED_PID on the prescribed square route

Continuous performance metrics were evaluated using successfully completed runs only. Table 3 reports the descriptive values together with the Mann-Whitney U test results and Holm-adjusted p -values.

The trajectory-error metrics were numerically lower for ADAPTIVE_PID, with reductions of approximately 3-5% in mean error, RMSE, P95 error, and final error. Mean cross-track error was nearly identical between the two controller modes. None of these trajectory-accuracy measures reached statistical significance after Holm's correction.

The distributions of RMSE, route completion time, and control-signal saturation occupancy for successfully completed runs are shown in Fig. 3.

The distribution of route completion times differed significantly between the two controller modes after Holm's correction $p_{Holm} = 0.027$. The mean completion time was 144.63 s for FIXED_PID and 141.35 s for ADAPTIVE_PID, while ADAPTIVE_PID exhibited greater run-to-run variability.

The most pronounced difference among the continuous performance measures was observed for control-signal saturation. Mean saturation occupancy decreased from 52.60% for FIXED_PID to 48.00% for ADAPTIVE_PID, corresponding to a reduction of 4.60 percentage points. This difference remained statistically significant after correction for multiple comparisons $p_{Holm} < 0.001$.

Overall, ADAPTIVE_PID achieved a higher route-completion proportion and reduced control-signal saturation while maintaining trajectory-following accuracy comparable to FIXED_PID.

Table 3.

Performance metrics for successfully completed routes (FIXED_PID, n = 35; ADAPTIVE_PID, n = 46)

Metric	FIXED_PID	ADAPTIVE_PID	Difference	p-value	Holm-adjusted p-value
Mean error, px	150.11 ± 30.83	142.12 ± 23.03	−5.32%	0.255	1.000
RMSE, px	167.11 ± 32.39	158.99 ± 22.61	−4.86%	0.328	1.000
P95 error, px	278.57 ± 50.02	269.74 ± 38.93	−3.17%	0.545	1.000
Final error, px	134.87 ± 64.76	129.72 ± 73.68	−3.82%	0.584	1.000
Mean cross-track error, px	33.77 ± 8.40	33.85 ± 8.20	+0.24%	0.973	1.000
Route completion time, s	144.63 ± 15.26	141.35 ± 31.00	−2.27%	0.0046	0.027
Saturation occupancy, %	52.60 ± 4.66	48.00 ± 4.84	−4.60 pp	0.00013	<0.001

Values are reported as mean ± standard deviation

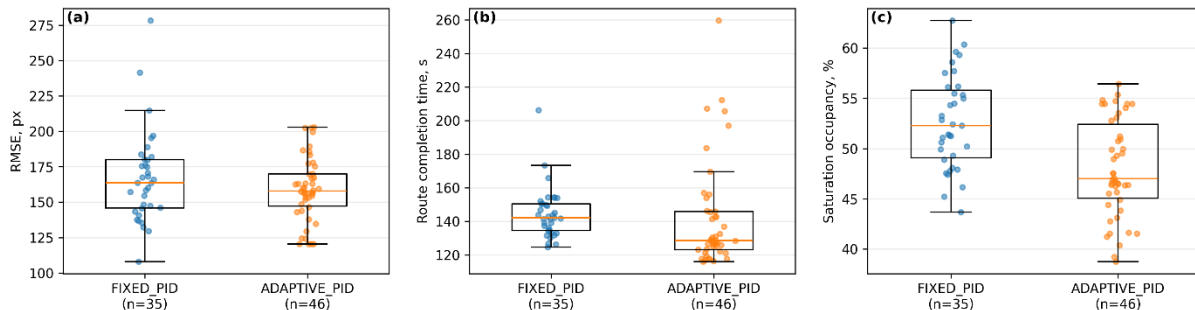


Fig. 3 Distributions of run-level performance metrics for successfully completed FIXED_PID and ADAPTIVE_PID runs: (a) RMSE, (b) route completion time, and (c) control-signal saturation occupancy

Discussion

The experimental results demonstrate a clear practical advantage of the proposed adaptive gain-scheduling method under the investigated simulation conditions. ADAPTIVE_PID increased the route-completion rate from 70% to 92%, corresponding to an improvement of 22 percentage points, and the difference between the controller modes was statistically significant. At the same time, no statistically significant differences were observed in the principal trajectory-error measures after correction for multiple comparisons. This combination is important because the higher route-completion rate was achieved without a statistically significant deterioration in the observed trajectory-following accuracy.

A second important result was the reduction in control-signal saturation occupancy from 52.60% for FIXED_PID to 48.00% for ADAPTIVE_PID. This difference remained statistically significant after Holm correction and is consistent with the intended behaviour of the adaptive controller. The gain-scheduling mechanism modifies the controller response according to the current motion condition and applies saturation-related gain adjustments independently to the affected image-plane axis. This allows the controller to reduce operation at the available command limits without unnecessarily restricting the unaffected control channel. The observed reduction in saturation therefore indicates that the adaptive mechanism reduced the proportion of control updates performed at the saturation limit while preserving similar trajectory-following accuracy.

The route-completion-time distributions also differed significantly between the controller modes. The mean completion time decreased from 144.63 s for FIXED_PID to 141.35 s for ADAPTIVE_PID, although the difference between the mean values was relatively small and the adaptive controller exhibited greater run-to-run variability. This result should therefore not be interpreted as evidence of a substantial improvement in route-execution speed. In addition, completion time was evaluated only for successfully completed runs, whereas the route-completion proportions differed between the controller modes. The completion-time result should consequently be interpreted as a characteristic of successful runs rather than as an unconditional comparison of execution speed.

Taken together, the results suggest that the main benefit of the proposed adaptive PID approach lies in improving the reliability of route completion and reducing operation at the control limits rather than in producing substantially lower trajectory-error values. This behaviour is consistent with the purpose of the proposed gain-scheduling strategy, which adapts the controller response to motion phase, visual confidence, route progression, cross-track error, and saturation state. However, the present comparison evaluates the complete adaptive controller against

the fixed-gain baseline and does not isolate the individual contribution of each adaptive mechanism. Therefore, the higher completion rate cannot be attributed exclusively to the reduction in saturation occupancy.

Several limitations constrain the generalization of the obtained results. The final quantitative validation was performed using a single predefined square route in image pixel space. Triangular and zigzag trajectories were used during preliminary controller development to verify operation under alternative motion patterns, but these runs were not included in the balanced final validation dataset and therefore do not provide statistical evidence of route-independent performance. The experiments were conducted in a Gazebo/ArduPilot Software-in-the-Loop environment, and corresponding runs with identical pseudorandom seed identifiers were not assumed to be fully deterministic.

The study also did not evaluate the effects of real-camera noise, illumination changes, motion blur, communication delays, wind disturbances, or discrepancies between simulated and physical UAV dynamics. Furthermore, the controller operates on visual displacement expressed in image pixels, so the reported trajectory-error values should not be interpreted directly as metric position errors without appropriate camera calibration and geometric scaling. No claim of formal closed-loop stability is made on the basis of the experimental observations. Further work should evaluate the proposed controller using multiple predefined route geometries, controlled visual and environmental disturbances, different operating conditions, and hardware-based UAV experiments to determine whether the observed improvements generalize beyond the investigated simulation scenario.

Conclusions

This study developed and experimentally evaluated an adaptive PID gain-scheduling method for UAV position control based on visual displacement in image pixel space. The proposed approach adapts the controller gains according to the current motion condition, visual-positioning confidence, route progression, cross-track error, and control-signal saturation, while handling saturation independently for the two image-plane axes.

The final validation consisted of 100 experimental runs under matched simulation conditions. ADAPTIVE_PID achieved a route-completion rate of 92%, compared with 70% for FIXED_PID, and the difference in completion proportions was statistically significant. For successfully completed routes, no statistically significant differences were observed in the principal trajectory-error measures after correction for multiple comparisons. At the same time, mean control-signal saturation occupancy decreased from 52.60% to 48.00% and remained statistically significant after Holm correction.

The obtained results indicate that, under the investigated simulation conditions, the proposed adaptive PID approach improved route-completion reliability and reduced operation at the control limits without statistically significant deterioration in trajectory-following accuracy. The conclusions are limited to the investigated square-route validation in the Gazebo/ArduPilot Software-in-the-Loop environment. Further work should extend the evaluation to multiple predefined route geometries, controlled visual and environmental disturbances, and hardware-based UAV experiments.

ADDITIONAL INFORMATION

AUTHOR CONTRIBUTIONS

Conceptualization, V.K.; methodology, M.S.; software, M.S.; investigation, M.S.; data curation, M.S.; formal analysis, M.S.; visualization, M.S.; writing - original draft preparation, M.S.; writing - review and editing, V.K.; supervision, V.K. All authors have read and agreed to the published version of the manuscript.

DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

In preparing this manuscript, the authors used ChatGPT (OpenAI) for language editing, translation, paraphrasing and rephrasing of individual sentences, consistency checking, and adaptation of the manuscript formatting to the journal requirements. All experimental data, controller implementation, statistical results, interpretations, and scientific conclusions were reviewed and verified by the authors. After using this tool, the authors reviewed and edited the resulting content and take full responsibility for the content of this publication.

REFERENCES

- | | |
|--|--|
| Bu Y., Yan Y., Yang Y. Advancement Challenges in UAV Swarm Formation Control: A Comprehensive Review. 2020. Vol. 43, No. 1. P. 1-15. | N
o
.
3. Melo A. G., Andrade F. A. A., Guedes I. P., Carvalho G. F., Zachy A. R. L., Pinto M. F. Fuzzy Gain-Scheduling PID for UAV Position and Altitude Controllers. Sensors. 2022. Vol. 22, No. 6. Art. 2173. U
R
4. Boubakir A., Souanef T., Labiod S., Whidborne J. F. A Robust Adaptive PID-like Controller for UAV Positioning. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2021. Vol. 51, No. 1. P. 1-15. |
|--|--|

Controller for Quadrotor Unmanned Aerial Vehicle Systems. Aerospace. 2024. Vol. 11, No. 12. Art. 980.

U

R

5. Spencer J., Lee J., Paredes J. A., Goel A., Bernstein D. S. An Adaptive PID Autotuner for Multicopters with Experimental Results. 2022 IEEE Conference on Robotics and Automation

(

h

6. Conte G., Doherty P. Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. EURASIP Journal on Advances in Signal

P

7. Yousif K., Bab-Hadiashar A., Hoseinnezhad R. An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. Intelligent Industrial Systems. 2015. Vol. 1, No. 4. P. 289–311. URL: <https://doi.org/10.1007/s40903-015-0150-0>

0

8. Ma J., Chen P., Xiong X., Zhang L., Yu S., Zhang D. Research on Vision-Based Servoing and Trajectory Prediction Strategy for Capturing Illegal

D

7

9. Rublee E., Rabaud V., Konolige K., Bradski G. ORB: An Efficient Alternative to SIFT or SURF. 2011

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

IEEE International Conference on Computer Vision (ICCV). 2011. P. 2564–2571. URL:

10. Koenig N., Howard A. Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2004. Vol. 3. P. 2149–2154. URL:

<https://doi.org/10.1109/IROS.2004.1389727>

11. Baidya S., Shaikh Z., Levorato M. FlyNetSim: An Open Source Synchronized UAV Network Simulator Based on ns-3 and ArduPilot. Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of W

i

12. Suslo M., Kazymyr V. Method of Stabilizing a Quadcopter in a Preferred Position with Spatial PID-Adjustment Based on Video Data without GPS. Technical Sciences and Technologies. 2026. No. 3(44). P. 306–316. URL:

[https://doi.org/10.25140/2411-5363-2026-2\(44\)-306-316](https://doi.org/10.25140/2411-5363-2026-2(44)-306-316)

4

3

M

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

АДАПТИВНЕ PID-РЕГУЛЮВАННЯ ПОЛОЖЕННЯ БПЛА ЗА ВІЗУАЛЬНИМ ЗМІЩЕННЯМ

Керування безпілотними літальними апаратами (БПЛА) на основі візуального зміщення потребує роботи регулятора положення безпосередньо з похибками в координатах зображення з одночасною адаптацією до змін умов руху та обмеженого діапазону керуючих впливів. У дослідженні запропоновано метод адаптивного налаштування коефіцієнтів PID-регулятора для проходження БПЛА маршруту з використанням візуального зміщення, оціненого за даними камери, спрямованої вниз. Регулятор працює в піксельному просторі зображення та змінює коефіцієнти PID залежно від поточного режиму руху, достовірності візуального позиціонування, прогресу проходження маршруту, поперечної похибки та насичення керуючого сигналу. Насичення опрацьовується окремо для двох осей площини зображення, що дає змогу для осі, на якій виникло насичення, застосовувати обмежену конфігурацію коефіцієнтів, тоді як інша вісь зберігає конфігурацію, що відповідає поточному режиму руху. Запропонований метод було реалізовано та оцінено в середовищі Gazebo/ArduPilot Software-in-the-Loop і порівняно з PID-регулятором із фіксованими коефіцієнтами. Попередні тести під час розроблення також включали трикутні та зигзагоподібні маршрути, однак ці запуски не входили до фінальної статистичної валідації. Фінальний набір валідаційних даних складався зі 100 експериментальних запусків, по 50 запусків для кожного режиму керування, виконаних на одному наперед заданому квадратному маршруті за однакової конфігурації моделювання. Ефективність оцінювали за часткою успішного завершення маршруту, показниками траєкторної похибки, поперечною похибкою, часом завершення маршруту та часткою роботи в режимі насичення керуючого сигналу. Частки завершення маршруту порівнювали за допомогою критерію Фішера, а неперервні показники на рівні окремих запусків аналізували за допомогою критерію Манна-Вітні поправкою Холма на множинні порівняння. Адаптивний регулятор успішно завершив 46 із 50 валідаційних запусків, що відповідає частці завершення маршруту 92%, тоді як регулятор із фіксованими коефіцієнтами завершив 35 із 50 запусків, що відповідає 70%. Різниця часток завершення була статистично значущою. Для успішно завершених маршрутів після застосування поправки Холма статистично значущих відмінностей в основних показниках траєкторної похибки не виявлено. Середня частка роботи в режимі насичення керуючого сигналу зменшилася з 52,60% для регулятора з фіксованими коефіцієнтами до 48,00% для адаптивного регулятора, і ця різниця залишалася статистично значущою після поправки на множинні порівняння. Отримані результати свідчать, що за досліджених умов моделювання запропонований метод адаптивного налаштування коефіцієнтів PID-регулятора був пов'язаний із вищою часткою успішного завершення маршруту та меншою роботою на межах керуючих впливів без статистично значущого погіршення спостережуваної точності проходження траєкторії. Висновки обмежені дослідженою валідацією на квадратному маршруті та середовищем Software-in-the-Loop і потребують подальшої перевірки на додаткових геометріях маршрутів, за наявності збурень і в натурних експериментах з БПЛА.

Ключові слова: адаптивне PID-регулювання; безпілотний літальний апарат; візуальне зміщення; візуальна навігація; Gazebo; ArduPilot

U

R

L