Iterative Learning Control for Quadrotor Pose Tracking

Davi J. G. Sousa * **Débora N. P. Oliveira** * **Marcos R. A. Morais** ** **Antonio M. N. Lima** **

* *Electrical Engineering Graduate Program, Universidade Federal de Campina Grande (email: davi.j.g.sousa@ee.ufcg.edu.br, debora.oliveira@ee.ufcg.edu.br).*
** *Department of Electrical Engineering, Universidade Federal de Campina Grande (emails: morais@dee.ufcg.edu.br, amnlima@dee.ufcg.edu.br).*

Abstract: This work proposes a dual-layer control strategy for managing the pose of a quadrotor unmanned aerial vehicles in repetitive tasks. The first layer uses iterative learning control to reduce the error between the desired reference trajectory signal and the system output. This layer generates the desired flight trajectories and transmits them to the second control layer. The second control layer uses a dual-feedback proportional derivative strategy to achieve trajectory tracking accuracy. We conducted experiments using a lemniscate trajectory mission to verify the efficiency of the proposed method.

Keywords: Quadrotors; Trajectory Tracking; Iterative Learning Control; Unmanned aerial vehicle; double-layer control.

1. INTRODUCTION

The autonomy of Unmanned Aerial Vehicles (UAVs) makes them highly suitable for rescue missions or monitoring in remote areas. Due to their ability to operate without physical proximity to the controller, UAVs are commonly used for supervising power lines (Liu et al., 2023), crops (Amaral et al., 2020; Beniaich et al., 2022), or military assignments (Zhang and Chen, 2021). However, these vehicles are nonlinear and underactuated systems with significant dynamic coupling (Zhao et al., 2020). Consequently, UAVs are susceptible to aerodynamic disturbances.

Given this susceptibility to external interferences, it is crucial to have a robust controller that ensures accuracy in trajectory tracking. Therefore, the control strategy of a quadrotor is a fundamental aspect. Several approaches have been applied using classical feedback control techniques (Bouabdallah et al., 2004) and adaptive strategies (Cowling et al., 2006) (Herrera et al., 2015).

Typically, UAV control involves a dual-loop control, consisting of an inner loop related to attitude dynamics and an outer loop associated with altitude dynamics. These two feedback controllers are then connected in a cascaded scheme (Emran and Najjaran, 2018). However, this classical approach modifies rotor speeds using reference signals, resulting in a delay in the tracking response (Foudeh et al., 2020). Backstepping control or feedback linearization can be advantageous, but these methods require a model with disturbances.

The Iterative Learning Control (ILC) is a robust adaptive control strategy that employs the sliding mode approach to enhance disturbance rejection capabilities and reduce sensitivity to failures in quadrotor UAVs (Fu and He, 2021). When compared to control strategies like PID, LQR, Backstepping, FLC, SMC, MPC, Neural Network, H-infinity, Fuzzy Logic, and Adaptive Control, the ILC stands out for its advantages. It learns from errors across successive iterations, thereby improving tracking performance (Roy et al., 2021). The ILC design adopts a two-step optimization method that is practical and straightforward to implement (Adlakha and Zheng, 2021). This process maximizes learning efficacy and ensures system stability (Li et al., 2021). Furthermore, the ILC is versatile in handling systems with multiple inputs and outputs, taking into account references for future tasks, making it ideal for controlling quadrotor aircraft (Abitha and PK, 2020).

To achieve more accurate trajectory tracking, we can leverage learning methods, as mentioned by Pham et al. (2018) and Lee et al. (2021). One particular learning strategy that has been applied to quadrotors is Iterative Learning Control (ILC) (Zhaowei et al., 2015). This direct feedforward technique creates an intelligent system that memorizes previous iterations of the same trajectory (Adlakha, 2019). Although it requires repetitive disturbances and initial conditions, ILC does not require a model. Other efforts have been made to relax these prerequisites concerning the reference trajectory and experiment length (Chen et al., 2021).

In the case of a closed platform, such as commercial drones, with an inaccessible embedded controller, we may encounter unsatisfactory trajectory error for a specific mission. Faced with this problem, we can utilize ILC to minimize the tracking error of the trajectory (Pipatpaibul and Ouyang, 2013) (Adlakha and Zheng, 2020) (Foudeh et al., 2020).

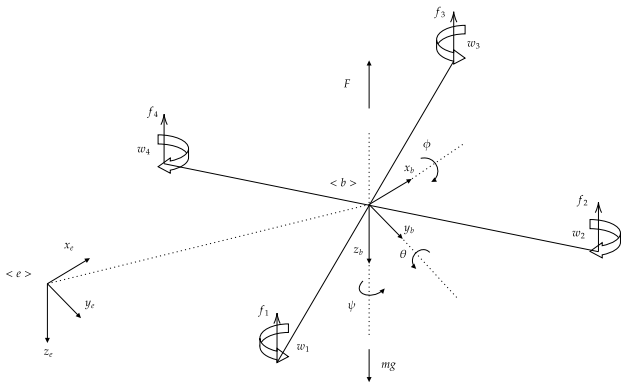

Figure 1. Coordinate system modelled with origin at the mass center of the drone.

Therefore, in this work, we consider a closed quadrotor platform where the performance of the commercial controller is not sufficiently accurate for our specific mission. Since this embedded controller cannot be modified, we implement improvements in the trajectory tracking loop through ILC, which takes the reference position and produces the error for the cascaded embedded controller.

2. QUADROTOR MODEL

The coordinate system used in this work is illustrated in Figure 1. The origin of the body coordinate system B is located at the center of gravity of the UAV. A quadrotor is a nonlinear system with multiple inputs and outputs, exhibiting strong state coupling. It is also underactuated, the number of actuators is smaller than the total degrees of freedom: there are four motors, three position states x, y, and z, and three angular states ϕ, θ, and ψ. The angles ϕ, θ, and ψ represent rotations around the x, y, and z axes, respectively, where $\phi \in (-\pi/2, \pi/2)$, $\theta \in (-\pi/2, \pi/2)$, and $\psi \in (-\pi, \pi)$. To simplify notation, we will use the following mathematical expressions: $c_{\text{angle}} = \cos(\text{angle})$, $s_{\text{angle}} = \sin(\text{angle})$, and $t_{\text{angle}} = \tan(\text{angle})$. Thus, the rotation matrices are defined as follows:

$$R_b^e(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \tag{1}$$

$$R_b^e(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \tag{2}$$

$$R_b^e(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Therefore, the rotation matrix from the body coordinate system B to the inertial coordinate system E is:

$$\begin{aligned} R_b^e &= R_b^e(\phi) R_b^e(\theta) R_b^e(\psi) \\ &= \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \end{aligned} \tag{4}$$

Next, the equation of motion with respect to E is:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R_b^e \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{5}$$

The mass of the drone is represented by m, and the lift force of each motor $i \in [1, 4]$, denoted as F_i, is defined as $F_i = \beta w_i^2$, where β is a positive lift constant and w is the angular velocity of each actuator. The relationship between the angular velocities $[p, q, r]^T$ in the body coordinate system B and $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ in the inertial coordinate system E is given by:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & -s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & \frac{s_\phi}{s_\theta} & \frac{c_\phi}{s_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{7}$$

Therefore, the translational model of the drone is:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} + g \begin{bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{bmatrix} + \begin{bmatrix} \dot{\psi} q - \dot{\theta} r \\ \dot{\phi} r - \dot{\psi} p \\ \dot{\theta} p - \dot{\phi} q \end{bmatrix} \tag{8}$$

The total drag force F is equal to the sum of the individual forces from each actuator F_i. On the other hand, the rotational dynamics model of the drone is as follows:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} l(F_4 - F_2)/I_x \\ l(F_3 - F_1)/I_y \\ -\alpha(w_1^2 - w_2^2 + w_3^2 - w_4^2)/I_z \end{bmatrix} + \begin{bmatrix} \frac{I_y - I_z}{I_x} \dot{\theta}\dot{\psi} \\ \frac{I_z - I_x}{I_y} \dot{\phi}\dot{\psi} \\ \frac{I_x - I_y}{I_z} \dot{\phi}\dot{\theta} \end{bmatrix} \tag{9}$$

Where l is the length of the arm, α is the drag constant, and $I = \text{diag}(I_x, I_y, I_z)$ is the inertia matrix. In summary, the equations can be solved in a state space representation as $\dot{X} = f(X) + g(U)$, where U is the input of force and torque and X is the state vector. When examining the equations 6 and 7, we can observe that the orientation of the drone $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ does not depend on the translational components $[p, q, r]^T$. However, the translational states assume that the rotational components are known.

3. CONTROL STRATEGY

In this work, we propose a control strategy grounded in a cascaded configuration of PD (proportional-derivative) controllers. Figure 2 shows the mechanism of double closed-loop PD control. The quadrotor UAV system is subdivided into a position subsystem and an attitude subsystem, which employs Euler angles. The outer control loop is designated for position control, while the inner control loop focuses on controlling the Euler angles. The aim is to follow the desired trajectory, denoted by $\Upsilon_d(t) = [x_d(t), y_d(t), z_d(t), \psi_d(t)]^T$, and the inner loop controller ensures that the Euler angles, θ and ϕ, converge to their desired angles θ_d and ϕ_d, which are calculated from the outer loop controller.

The proposed controller model is defined as:

$$\begin{bmatrix} \phi_d(t) \\ \theta_d(t) \end{bmatrix} = K_p \begin{bmatrix} -\sin(\psi) e_x + \cos(\psi) e_y \\ \cos(\psi) e_x + \sin(\psi) e_y \end{bmatrix} + K_d \begin{bmatrix} p(t) \\ q(t) \end{bmatrix} \tag{10}$$

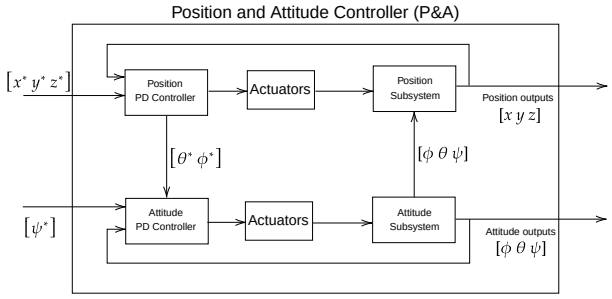

Figure 2. Position and Attitude Controller (P&A).

Where e_x and e_y represent the errors in x and y, respectively. The gains of the PD control law were determined using classical tuning techniques (Corke, 2017) and can be seen in Table 1.

Similarly, the inner control loop calculates the forces and torques necessary for the system to achieve the desired final pose ϕ, θ, and ψ. The torques $\tau_x = l(F_4 - F_2)$ and $\tau_y = l(F_3 - F_1)$ are determined by a PID control law as follows:

$$\begin{bmatrix}\tau_x(t)\\ \tau_y(t)\end{bmatrix} = K_p \begin{bmatrix}e_\phi\\ e_\theta\end{bmatrix} + K_i \begin{bmatrix}\int e_\phi dt\\ \int e_\theta dt\end{bmatrix} + K_d \begin{bmatrix}\dot{\phi}(t)\\ \dot{\theta}(t)\end{bmatrix} \quad (11)$$

Where e_ϕ and e_θ represent the errors in ϕ and θ, respectively. The torque $\tau_z = -\alpha(w_1^2 - w_2^2 + w_3^2 - w_4^2)$ is determined by a PD control law with ψ^* as the reference input. Therefore, we have:

$$\tau_z(t) = K_p e_\psi + K_d[\dot{\psi}(t)] \quad (12)$$

Where e_ψ represents the error in ψ. Finally, the vertical force F can be calculated by the altitude PID controller:

$$F(t) = K_p e_z + K_i \int e_z dt + K_d r(t) \quad (13)$$

Where e_z represents the error in z. To determine the values of τ_x, τ_y, τ_z, and F, it is necessary to know the relationship between the rotation velocity w and the generated moment. In this work, this relationship is described by a mixing matrix M as follows:

$$\begin{bmatrix}w_1^2\ w_2^2\ w_3^2\ w_4^2\end{bmatrix} = M^{-1}\begin{bmatrix}F\ \tau_x\ \tau_y\ \tau_z\end{bmatrix} \quad (14)$$

Where M is the mixing matrix defined as:

$$M = \begin{bmatrix} \beta & \beta & \beta & \beta \\ -\frac{\sqrt{2}}{2}l\beta & -\frac{\sqrt{2}}{2}l\beta & \frac{\sqrt{2}}{2}l\beta & \frac{\sqrt{2}}{2}l\beta \\ -\frac{\sqrt{2}}{2}l\beta & \frac{\sqrt{2}}{2}l\beta & \frac{\sqrt{2}}{2}l\beta & -\frac{\sqrt{2}}{2}l\beta \\ -\alpha & \alpha & -\alpha & \alpha \end{bmatrix} \quad (15)$$

This matrix describes the relationship between the forces and torques applied to the propellers and the squared angular velocities w_1^2, w_2^2, w_3^2, and w_4^2 of the motors.

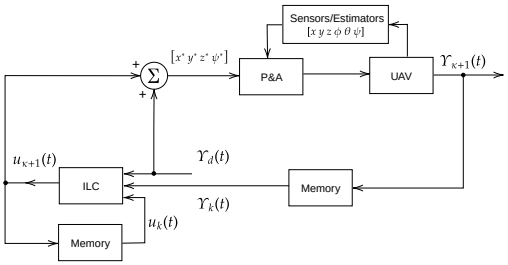

Figure 3. ILC configuration.

4. ILC CONTROL STRATEGY FOR UAVS

The primary objective of the proposed Iterative Learning Control (ILC) problem is to optimize and fine-tune the desired trajectory for the Unmanned Aerial Vehicle (UAV). For this purpose, the current states of the UAV are stored and used as a reference for future iterations. Specifically, the previously stored state is subtracted from the previously stored target to generate an adjustment value. This adjustment value is incorporated into the subsequent iteration as a correction factor, thus adjusting the current state of the UAV. The introduction of this adjustment value into subsequent iterations allows for incremental refinement of the UAV's trajectory, contributing to a quicker and more accurate convergence toward the desired state.

Figure 3 depicts the proposed complete control structure. An additional control layer has been designed based on the ILC strategy, aiming to tackle missions that are repeatedly executed. This ILC control layer generates the desired motion adjustment trajectories $u_k(t) = [x_{uk}(t), y_{uk}(t), z_{uk}(t), \psi_{uk}(t)]^T$ for the quadcopter, with this adjustment being refined with each mission repetition. As a result, the control system's performance progressively improves: the more the mission is repeated, the more refined the quadcopter's mission adjustment becomes.

The controller aims to minimize the error between the output $\Upsilon_k(t) = [x_k(t), y_k(t), z_k(t), \psi_k(t)]^T$ and the desired value $\Upsilon_d(t) = [x_d(t), y_d(t), z_d(t), \psi_d(t)]^T$. Therefore, the objective of formation control can be mathematically expressed as:

$$\lim_{k\to\infty} \Upsilon_k(t) = \Upsilon_d(t), \quad t \in [0, T] \quad (16)$$

5. ITERATIVE LEARNING

Firstly, we set a mission for the quadcopter. In this mission, it starts from an initial point, and upon completion, returns to the same location, similar to inspecting a specific area. In every mission, the quadcopter follows the same predetermined trajectory. To enable the quadcopter to learn and anticipate possible challenges that might arise during the missions, each stopping point along the trajectory is recorded. Based on this, the following assumptions are made:

- Use of on-board instrumentation,
- Each iteration ends within a fixed time interval,
- The initial state $\Upsilon_k(0)$ can be used in all iterations.
- The system is invariant.

- The output Υ_k is deterministic.
- The system dynamics are also deterministic.

Thus, the control law defined by iterative learning is:

$$u_{k+1}(t) = u_k(t) + \Gamma(\underbrace{\Upsilon_d(t) - \Upsilon_k(t)}_{e_k(t)}) \tag{17}$$

where $\Gamma \in R^{n_u}$ is a diagonal gain matrix for learning. Additionally, $u_0 \in R^{n_x}$ represents the initial state of u_k.

Given that each mission or iteration concludes within a predetermined time interval, a method has been proposed for storing data collected by onboard sensors on the UAV. This data is examined at different frequencies for more in-depth analysis. The storage occurs at specific moments along the trajectories where the drone receives control signals. This stored dataset serves as the foundation for adjusting control parameters in the subsequent iteration. This adjustment mechanism is cumulative; it aims to minimize errors caused by environmental disturbances, thereby improving trajectory accuracy over successive iterations.

6. RESULTS AND DISCUSSION

Table 1 presents the adjusted gains for the altitude and attitude controllers.

In this work, we used a Parrot Mambo Fly drone, which weighs 60 grams and has a width of 18 centimeters. The drone is equipped with a set of integrated sensors that provide information about altitude, velocity, and current position to the controller. To estimate altitude, a Kalman filter is used, combining data from a downward-facing ultrasonic altimeter and a compact barometer. Additionally, the drone has a built-in Inertial Measurement Unit (IMU) to measure velocity and acceleration. The optical flow from a stereo camera is also used to enhance the estimation of the Kalman filter.

Furthermore, we implemented a cascaded control approach that allows us to calculate the required forces and torques (Corke, 2017). To determine the control law, we used controllable reference values such as p^*, q^*, r^*, and ψ^*. Additionally, the estimated values $\hat{p}$, $\hat{q}$, $\hat{r}$, and $\hat{\psi}$ from the state space X are corrected using a Kalman filter.

The position and attitude control implemented on the Mambo drone are performed using the Simulink Support Package for Parrot, which allows for direct implementation of flight control algorithms on Parrot quadrotors. With the proposed controller, we conducted missions to investigate the quadrotor's performance in following a predefined geometric trajectory. We chose a lemniscate-shaped trajectory to simulate the worst maneuverability condition. Two experiments were conducted for the following trajectory:

Table 1. Tuned gains for the altitude and attitude controllers.

Saída	K_p	K_i	K_d
$[\phi^* \ \theta^*]^T$	$[-0,24\ 0,24]^T$	–	$[0,1\ -0,1]^T$
$[\tau_x \ \tau_y]^T$	$[0,013\ 0,01]^T$	$[0,01\ 0,01]^T$	$[-0,002\ -0,003]^T$
τ_z	$0,004$	–	$-0,0012$
F	$0,8$	$0,1$	$-0,5$

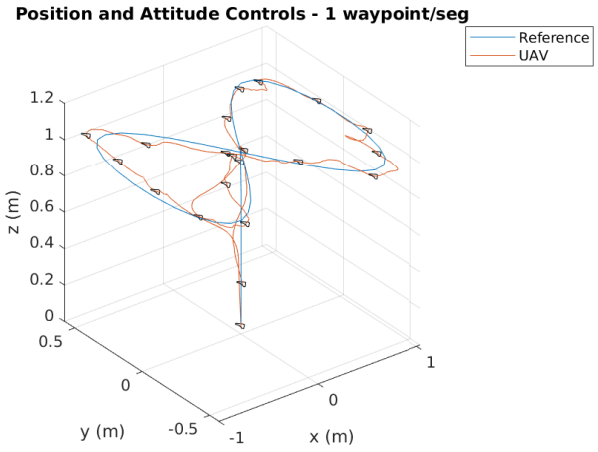

Figure 4. Experiment with position and attitude controller for 1 waypoint/seg.

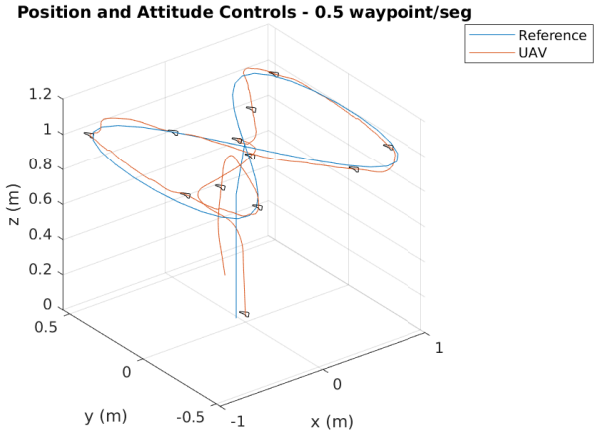

Figure 5. Experiment with position and attitude controller for 0.5 waypoint/seg.

$$\begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \\ \psi_d(t) \end{bmatrix} = \begin{bmatrix} \sin(0.3t_\sigma) \\ \sin(0.3t_\sigma)\cos(0.3t_\sigma) \\ 1 \\ 0 \end{bmatrix}, t_\sigma \in [0, 0.5, 1, 1.5, .., 21]$$

In the first experiment, the tracking speed was set at 1 reference point per second. The results of this experiment are visually represented in Figure 4. In the corresponding graph, found in Figure 9, the blue segment demonstrates the observed mean squared error. In the second experiment, the tracking speed was pre-defined at 0.5 reference points per second. The results for this case are illustrated in Figure 5. In this graph, the yellow segment represents the mean squared error associated with the second experiment. From these analyses, it becomes clear that the position and attitude controller, on its own, is not effective in minimizing the accumulated tracking errors from sensors caused by environmental disturbances.

To address the limitations of a stand-alone PD position and attitude controller in effectively mitigating accumulated sensor errors induced by environmental factors, this work introduces an innovative solution: a two-layered control system. In this dual-layer control architecture, the

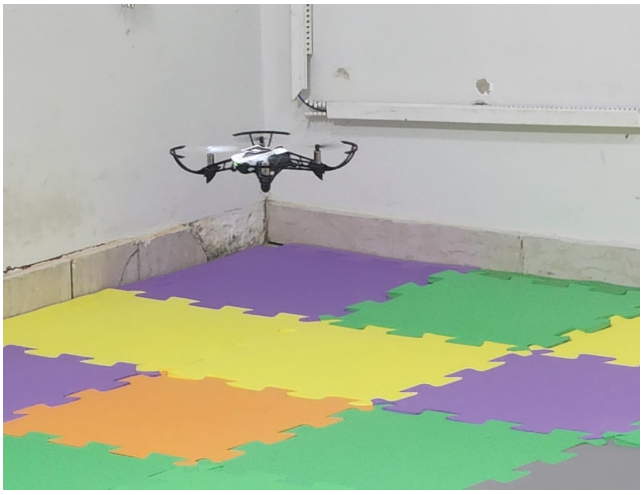

Figure 6. Laboratory experiment arrangement.

first layer employs Iterative Learning Control (ILC) to target and minimize the cumulative sensor error. The second layer integrates a PD position and attitude controller configured with a dual-supply mechanism.

This layered control strategy enables us to draw a comparative analysis between the newly proposed system and the original controller, thereby elucidating the benefits of incorporating an additional ILC layer.

Initially, the ILC layer was configured using the Simulink Support Package for Parrot for a tracking speed of 1 reference point per second. A predetermined learning stop criterion for the mean squared error (MSE) was set at 0.01, represented by the parameter Γ in Equation 18. After five iterative cycles, the system achieved a remarkable MSE of 0.0041. This optimized result was then implemented on the actual drone, culminating in a trajectory that is visually documented in Figure 6.

By adopting this two-layered approach, we aim to substantiate the efficacy of integrating an additional ILC control layer in improving the tracking accuracy, while potentially mitigating the influences of environmental perturbations on sensor data.

$$\Gamma = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

Figure 7 illustrates the first experiment using ILC with a desired tracking speed of 1 reference point per second. In the corresponding mean squared error (MSE) graph, designated as Figure 9, the red segment signifies the MSE achieved with this updated control architecture. Notably, the error demonstrated by the red segment shows a discernible reduction when compared to the blue segment and consistently stays below the 0.01 threshold, even under real-world experimental conditions. This compelling evidence underscores the efficacy of the ILC methodology in minimizing cumulative sensor errors and enhancing the overall performance of the control system.

As a subsequent step, we employed Iterative Learning Control (ILC) with a pre-defined tracking speed of 0.5

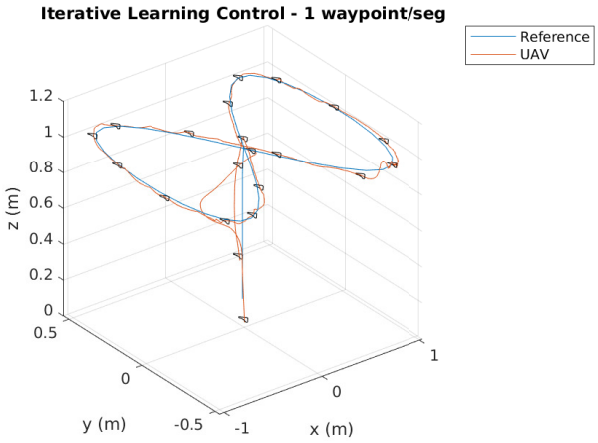

Figure 7. Experiment with ILC controller with k=5 for 1 waypoint/seg.

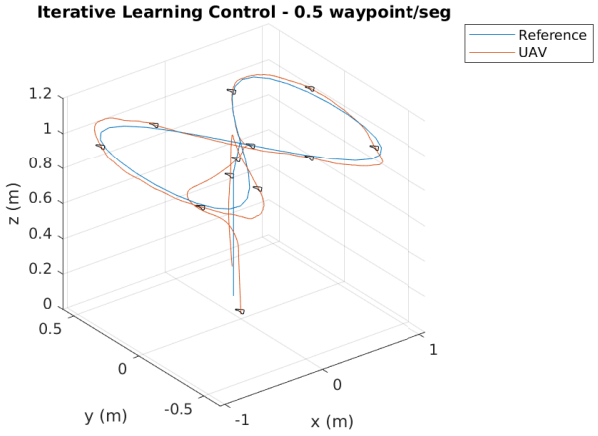

Figure 8. Experiment with ILC controller with k=5 for 0.5 waypoint/seg.

reference points per second and established a learning stop criterion for the mean squared error (MSE) at 0.02. Upon completing five iterative cycles, we successfully achieved an MSE value of 0.0183. When this optimized result was implemented on the actual drone, the resulting trajectory is visually captured in Figure 8. In the associated MSE graph, depicted in Figure 9, the purple segment symbolizes the MSE achieved with this new control configuration. It's noteworthy that although the error experiences a decline when compared to the yellow segment, it fails to remain below the 0.02 threshold during tests conducted under real-world experimental conditions. This behavior can be attributed to the high controller gains used in this particular setting.

Figures 10 and 11 provide a comprehensive comparative analysis of the quadcopter's trajectories under different tracking speeds, specifically 1 and 0.5 reference points per second. In these comparative graphs, the superior efficacy of the Iterative Learning Control (ILC) over the conventional position and attitude controller is unmistakably evident.

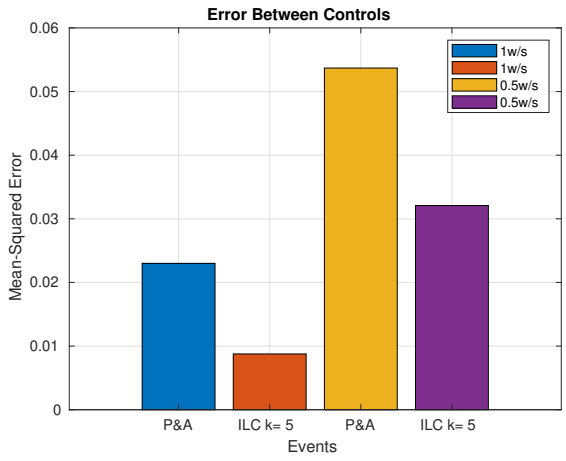

Figure 9. Mean-squared error between controls.

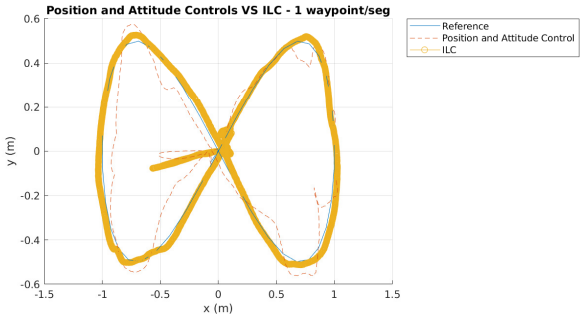

Figure 10. Comparison experiment between the controllers for 1 waypoint/second, considering the xyz axes as reference.

In Figure 10, it's apparent that the ILC controller not only produces a more accurate trajectory but also a smoother one, with a significant minimization of deviations from the established reference point. The quadcopter demonstrates remarkable capability in tracking the planned trajectory with both precision and stability.

Conversely, in Figure 11, even while operating at a lower tracking speed, the ILC controller maintains its superior performance, displaying reduced tracking error and less fluctuation relative to the reference point. This outcome suggests a robust adaptive efficiency of the ILC controller.

These results highlight the advantage of the ILC controller in terms of trajectory precision and tracking ability, showcasing its effectiveness compared to the traditionally used position and attitude controller. This approach offers greater reliability and improved performance of the quadrotor in tasks that require precise trajectory tracking.

In summary, the results and discussions of this work demonstrate that the initially implemented position and attitude controller cannot adequately minimize the cumulative tracking errors caused by environmental disturbances. However, by adding an additional layer of control with ILC, it is possible to significantly reduce these errors. ILC has shown an improvement in drone performance, resulting in a more accurate trajectory and reduced sensor tracking error. However, it was observed that the proper selection of learning parameters such as desired tracking

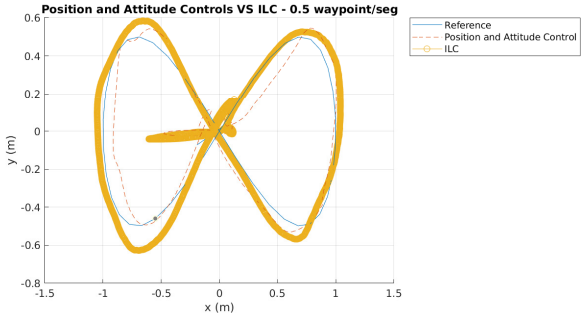

Figure 11. Comparison experiment between the controllers for 0.5 waypoints/second, considering the xyz axes as reference.

speed and stopping criterion is crucial to achieve satisfactory results.

Furthermore, it is important to note that the experiments were conducted in a laboratory environment, and further experiments in real-world conditions are required to validate the effectiveness of the proposed controller. This research contributes significantly to the field of drone control, offering an approach that improves trajectory tracking performance and reduces control errors caused by environmental disturbances. In future work, other control and optimization techniques can be explored to further enhance the drone's performance in different operational scenarios.

7. CONCLUSION

In this article, the implementation of iterative learning control for quadrotor position was proposed. The objective was to investigate whether the application of this iterative learning control on a Parrot Mambo drone would be able to reduce trajectory tracking error compared to the already implemented position and attitude controller. For this purpose, two experiments were conducted, varying the trajectory tracking speed. The obtained results demonstrated that the addition of iterative learning control minimized the trajectory tracking error.

As future work, it is suggested to propose an online trajectory training method and an evaluation approach for learning-based control algorithms, considering integral and derivative error. This would allow for a more comprehensive and enhanced analysis of controller performance in different scenarios.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support provided by PPgEE/UFCG, CAPES, and CNPq.

REFERENCES

Abitha, M. and PK, A.S. (2020). Comparative analysis of path control strategies for unmanned quadrotor aerial vehicle. In *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*, 1–6. IEEE.

Adlakha, R. (2019). *Design and Implementation of Iterative Learning Control for Quad-rotor UAV's Tracking*. Ph.D. thesis, State University of New York at Buffalo.

Adlakha, R. and Zheng, M. (2020). An optimization-based iterative learning control design method for uav's trajectory tracking. In *2020 American Control Conference (ACC)*, 1353–1359. doi:10.23919/ACC45564.2020.9147752.

Adlakha, R. and Zheng, M. (2021). A two-step optimization-based iterative learning control for quadrotor unmanned aerial vehicles. *Journal of Dynamic Systems, Measurement, and Control*, 143(7), 071006.

Amaral, L.R.d., Zerbato, C., Freitas, R.G.d., Barbosa Júnior, M.R., and Simões, I.O.P.d.S. (2020). Aplicações de uavs na agricultura 4.0. *Revista Ciência Agronômica*, 51(spe).

Beniaich, A., Silva, M.L., Guimarães, D.V., Avalos, F.A., Terra, F.S., Menezes, M.D., Avanzi, J.C., and Cândido, B.M. (2022). Uav-based vegetation monitoring for assessing the impact of soil loss in olive orchards in brazil. *Geoderma Regional*, 30, e00543.

Bouabdallah, S., Noth, A., and Siegwart, R. (2004). Pid vs lq control techniques applied to an indoor micro quadrotor. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2451–2456. IEEE.

Chen, Z., Liang, X., and Zheng, M. (2021). Iterative learning for heterogeneous systems. *IEEE/ASME Transactions on Mechatronics*, 1–1. doi:10.1109/TMECH.2021.3085211.

Corke, P. (2017). *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, volume 118. Springer.

Cowling, I.D., Whidborne, J.F., and Cooke, A.K. (2006). Optimal trajectory planning and lqr control for a quadrotor uav. In *International Conference on Control*.

Emran, B.J. and Najjaran, H. (2018). A review of quadrotor: An underactuated mechanical system. *Annual Reviews in Control*, 46, 165–180.

Foudeh, H.A., Luk, P., and Whidborne, J. (2020). Application of norm optimal iterative learning control to quadrotor unmanned aerial vehicle for monitoring overhead power system. *Energies*, 13(12), 3223.

Fu, X. and He, J. (2021). Robust adaptive sliding mode control based on iterative learning for quadrotor uav. *IETE Journal of Research*, 1–13.

Herrera, M., Chamorro, W., Gómez, A.P., and Camacho, O. (2015). Sliding mode control: An approach to control a quadrotor. In *2015 Asia-Pacific Conference on Computer Aided System Engineering*, 314–319. doi:10.1109/APCASE.2015.62.

Lee, W., Jeon, Y., Kim, T., and Kim, Y.I. (2021). Deep reinforcement learning for uav trajectory design considering mobile ground users. *Sensors*, 21(24), 8239.

Li, X., Xu, X., Shen, Z., and Sun, M. (2021). Iterative learning-based pid precision control for sports performance analysis. *Wireless Communications and Mobile Computing*, 2021, 1–13.

Liu, X., Miao, X., Jiang, H., Chen, J., Wu, M., and Chen, Z. (2023). Tower masking mim: A self-supervised pretraining method for power line inspection. *IEEE Transactions on Industrial Informatics*, 1–11. doi:10.1109/TII.2023.3268479.

Pham, H.X., La, H.M., Feil-Seifer, D., and Van Nguyen, L. (2018). Reinforcement learning for autonomous uav navigation using function approximation. In *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 1–6. doi:10.1109/SSRR.2018.8468611.

Pipatpaibul, P.i. and Ouyang, P. (2013). Application of online iterative learning tracking control for quadrotor uavs. *International Scholarly Research Notices*, 2013.

Roy, R., Islam, M., Sadman, N., Mahmud, M.P., Gupta, K.D., and Ahsan, M.M. (2021). A review on comparative remarks, performance evaluation and improvement strategies of quadrotor controllers. *Technologies*, 9(2), 37.

Zhang, J. and Chen, X. (2021). Multi-uav reconnaissance task assignment based on wide-area search scenario. In *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, 620–625. IEEE.

Zhao, Z., Wang, J., Chen, Y., and Ju, S. (2020). Iterative learning-based formation control for multiple quadrotor unmanned aerial vehicles. *International Journal of Advanced Robotic Systems*, 17(2), 1729881420911520.

Zhaowei, M., Tianjiang, H., Lincheng, S., Weiwei, K., Boxin, Z., and Kaidi, Y. (2015). An iterative learning controller for quadrotor uav path following at a constant altitude. In *2015 34th Chinese Control Conference (CCC)*, 4406–4411. doi:10.1109/ChiCC.2015.7260322.