

Article

Iterative Learning Sliding Mode Control for UAV Trajectory Tracking

Lanh Van Nguyen ^{1,*}, Manh Duong Phung ^{1,2} and Quang Phuc Ha ¹

¹ School of Electrical and Data Engineering, University of Technology Sydney (UTS), 15 Broadway, Ultimo, NSW 2007, Australia; manhduong.phung@uts.edu.au (M.D.P.); quang.ha@uts.edu.au (Q.P.H.)

² VNU University of Engineering and Technology (VNU-UET), Vietnam National University, Hanoi (VNU), 144 Xuan Thuy, Cau Giay, Hanoi 100000, Vietnam

* Correspondence: vanlanh.nguyen@uts.edu.au

Abstract: This paper presents a novel iterative learning sliding mode controller (ILSMC) that can be applied to the trajectory tracking of quadrotor unmanned aerial vehicles (UAVs) subject to model uncertainties and external disturbances. Here, the proposed ILSMC is integrated in the outer loop of a controlled system. The control development, conducted in the discrete-time domain, does not require a priori information of the disturbance bound as with conventional SMC techniques. It only involves an equivalent control term for the desired dynamics in the closed loop and an iterative learning term to drive the system state toward the sliding surface to maintain robust performance. By learning from previous iterations, the ILSMC can yield very accurate tracking performance when a sliding mode is induced without control chattering. The design is then applied to the attitude control of a 3DR Solo UAV with a built-in PID controller. The simulation results and experimental validation with real-time data demonstrate the advantages of the proposed control scheme over existing techniques.

Keywords: iterative learning; sliding mode control; unmanned arial vehicles; trajectory tracking



Citation: Nguyen, L.V.; Phung, M.D.; Ha, Q.P. Iterative Learning Sliding Mode Control for UAV Trajectory Tracking. *Electronics* **2021**, *10*, 2474. <https://doi.org/10.3390/electronics10202474>

Academic Editor: Umberto Papa

Received: 19 September 2021

Accepted: 5 October 2021

Published: 12 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the interest in developing and utilizing unmanned aerial vehicles (UAVs) has been growing, with numerous applications in practice, such as mapping [1,2], inspection, search and rescue [3,4], construction automation [5], and agricultural surveillance [6]. When a quadrotor drone performs a desired trajectory, accurate tracking is highly necessary. In trajectory tracking control, feedback linearization (FL) has been widely used [7,8]. This control method works well under the assumption of known system dynamics. When faced with large uncertainties and disturbances, FL-based approaches may lead to poor tracking performance, and other advanced control laws are required. An adaptive feedback linearization controller was applied in [9], allowing for adjustments of the control parameters to enhance control performance. Robust control methods were also developed to improve control performance [10]. A backstepping controller was introduced in [11] to improve the tracking accuracy and robustness of UAVs' attitude control, wherein the external disturbances are estimated using a nonlinear disturbance observer.

Sliding mode control (SMC), a well-known control method for improving system robustness, has been successfully applied to various control systems [12,13] in general and particularly to UAVs [14,15]. However, information on disturbance bounds is required in these techniques. Adaptive SMC was developed to overcome this requirement [16]. This approach still reveals the main disadvantage of SMC, i.e., control signals usually present a chattering behavior, especially when dealing with large uncertainties and disturbances, which often require excessively high control gains. Various techniques have been suggested as a remedy, mostly using an approximation of the sign function to avoid or reduce chattering with some trade-offs with system robustness for control signal smoothing. Deep-learning-based techniques with convolutional neural networks have recently

shown promise in learning systems that are subject to highly complex and time-varying uncertainties, for example, in vision-based applications such as in robust face tracking [17]. For control in robotic manipulators, a deep convolutional neural network was developed to integrate learning schemes into a fractional-order terminal sliding-mode controller for enhancing tracking accuracy [18]. Although the deep learning methods can achieve high-quality performance with continuous control signals, a major disadvantage is the high computational cost incurred for implementing a deep neural network.

Iterative learning control (ILC) is an effective technique for dealing with repetitive tasks. It allows for learning from system data to update the control input repeatedly to improve system performance [19]. Through trial-based learning, ILC is able to achieve highly accurate tracking performance even with large model uncertainties and disturbances. Unlike nonlearning control techniques, the system in ILC is reset to zero after the system reaches the final time, and then repeatedly follows the same reference again. Thereby, the control input can be adjusted through the repetitions to result in perfect tracking. Since ILC can learn from the system response to provide feedforward control in the iteration domain, it is more robust and can effectively compensate for model uncertainties and unknown disturbances, particularly iteration-invariant disturbances [20].

The application of iterative cybernetics, first proposed in [21], has emerged for iterative learning control in robotic systems [22], later developed for industrial control [23]. In the past decades, ILC has become an effective tool in various control systems, such as robot arm manipulators [24], chemical batch processes [25], wafer scanner systems [26], and video-rate atomic force microscopy [27]. Unlike other learning techniques such as artificial neural networks, which obtain the inverse dynamics from a training set [28] or adaptive controller, which tunes the control parameters [29,30], requiring a time-consuming process, ILC updates the control input from the information from previous executions and can hence converge quickly after a limited number of repetitions [19]. Moreover, as ILC does not require a system model, it is quite beneficial in practical applications that deal with unknown characteristics.

In this paper, integrating the learning capacity of ILC with the strong robustness of SMC, we propose a novel iterative learning sliding mode controller (ILSMC) to achieve high-accuracy trajectory tracking for UAVs while retaining strong robustness as well as alleviating control chattering. In terms of iterative control, several existing techniques have been introduced for UAVs. In [31], a plug-in controller was designed and implemented in aerial robots. Although the average tracking error is reduced for periodic reference trajectories, the technique does not concern the effect of disturbances. In [32], fuzzy PID-typed ILC was introduced; where fuzzy logic is used to tune the control parameters, high-accuracy tracking performance is hard to attain in comparison to other rigorous control strategies. In [33], optimization-based ILC was developed to improve the UAV trajectory tracking performance. In this approach, learning and filtering schemes are formulated into convex optimal problems. Although the two-step convex optimization problem can be solved using software, it involves high computational complexity.

The proposed ILSMC offers a simpler design, and thus is more robust and effective. The main contributions of this work include (i) the comprehensive development of an iterative learning term with fast convergence after several iterations to compensate for system uncertainties and unknown disturbances, and (ii) the integration of ILC and SMC schemes to a built-in PID controller in cascade to yield high performance for quadcopter trajectory tracking.

This paper is structured as follows: The control development of ILSMC is presented in Section 2. The convergence and stability analysis of the proposed learning algorithm is also provided in this section. Next, Section 3 presents system modeling, including kinematic and quadcopter dynamics. Then, the integration of ILSMC with PID control for the UAV is described in Section 4. Section 5 provides numerical simulation results, and Section 6 presents experimental validation with real-time data. Finally, a conclusion is drawn in Section 7.

2. Iterative Learning Sliding Mode Control

Iterative learning control (ILC) is a tracking control strategy for systems performing repetitive tasks, which are commonly required in industry. This technique aims to generate a feedforward control signal so that the system can learn from the previous responses to improve tracking performance and repeatedly eliminate disturbance after each iteration. The basic structure of an ILC is depicted in Figure 1 for an iteration number j . At this iteration, the input $u_{(j)}(k)$ and the deviation $e_{(j)}(k)$ between the reference $y_d(k)$ and the output $y_{(j)}(k)$ are stored to compute the control signal for the next iteration, with k starting from an initial time instant ($k = 0$). In this section, an iterative learning sliding mode control scheme is designed to deal with large uncertainties and disturbances.

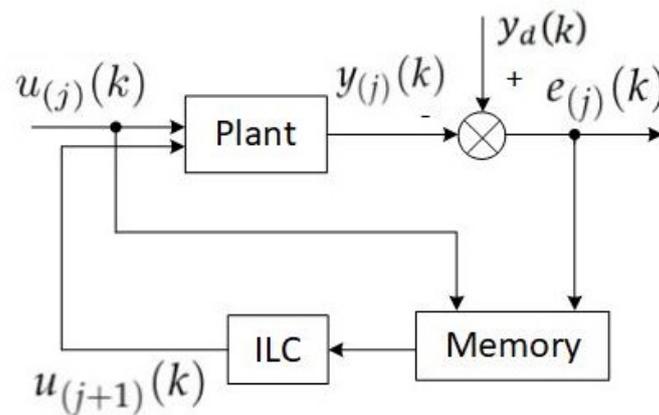


Figure 1. Basic structure of ILC.

2.1. ILSMC Design

Consider the following general discrete-time control system:

$$\begin{cases} x_{1(j)}(k+1) = x_{1(j)}(k) + \Delta_t x_{2(j)}(k), \\ x_{2(j)}(k+1) = f(x_{(j)}(k)) + \Delta_t B [u_{(j)}(k) + d_{(j)}(k)], \end{cases} \quad (1)$$

where k is the time instant and Δ_t is the sampling period. The subscript j denotes the iteration index, also called trial, run, cycle, or repetition in the ILC literature. The system state vector is $x_{(j)}(k) = [x_{1(j)}(k) \quad x_{2(j)}(k)]^T \in \mathbb{R}^{2m}$, where m is the dimension of state $x_{1(j)}(k)$, $x_{2(j)}(k)$ is its derivative, $u_{(j)}(k) \in \mathbb{R}^m$ is the control signal, $B \in \mathbb{R}^{m \times m}$ is a positive definite matrix, and $f(\cdot)$ is a vector function. The influence of parameter variations and loading conditions, model uncertainties and external disturbances can be lumped into a vector $d_{(j)}(k)$. In each iteration, the input and state variables comprise an N -sample sequence each, where N is a finite number of samples.

Definition 1. At iteration j , an exogenous input $\delta_{(j)}(k)$ is called iteration-invariant if it occurs repeatedly over iterations, or persistent within the iteration domain, that is, $\delta_{(1)}(k) = \delta_{(2)}(k) = \dots = \delta_{(j)}(k)$ for all $k = \{0, 1, \dots, N-1\}$.

To proceed with the ILC methodology, the following assumption [20,27,34] is made:

Assumption 1. In this paper, the lumped disturbance $d_{(j)}(k)$ is assumed to be iteration-invariant.

In this paper, an ILSMC law is developed aiming to drive the tracking error asymptotically to zero from any initial condition and under external disturbances $d_{(j)}(k)$. The control algorithm consists of two steps. The first step is to induce a desired sliding surface to drive a learning sliding function to zero after some iterations regardless of external disturbance

and system uncertainties. In the second step, the tracking error of the system is driven to zero in the sliding mode associated with the control sliding function. Instead of using a discontinuous gain as in the conventional SMC methodology, here, an iteration learning process is involved; hence, a priori information of the disturbance bound is not required, and chattering is fully alleviated.

The control design is initiated by considering the tracking error in an iteration as

$$e_{(j)}(k) = \begin{bmatrix} e_{1(j)}(k) & e_{2(j)}(k) \end{bmatrix}^T = x_{(j)}(k) - x_d(k), \quad (2)$$

where $x_d(k) = [x_{1d}(k) \quad x_{2d}(k)]^T$ is the desired trajectory vector, which is also iteration-invariant during the execution of repetitive tasks.

Let us define the control sliding function as:

$$\sigma_{(j)}(k) = e_{2(j)}(k) + ce_{1(j)}(k), \quad (3)$$

where $c = \text{diag}(c_i) \in \mathbb{R}^{m \times m}$, $c_i > 0$.

By denoting

$$\Delta\sigma_{(j)}(k) = [\sigma_{(j)}(k+1) - \sigma_{(j)}(k)]\Delta_t^{-1}, \quad (4)$$

we have from (3):

$$\begin{aligned} \sigma_{(j)}(k+1) &= e_{2(j)}(k+1) + ce_{1(j)}(k+1) \\ &= x_{2(j)}(k+1) - x_{2d}(k+1) + ce_{1(j)}(k+1). \end{aligned} \quad (5)$$

Substituting (1), (3), and (5) into (4) yields:

$$\begin{aligned} \Delta\sigma_{(j)}(k) &= [f(x_{(j)}(k)) + \Delta_t B [u_{(j)}(k) + d_{(j)}(k)] - x_{2d}(k+1) \\ &\quad + c [e_{1(j)}(k+1) - e_{1(j)}(k)] - e_{2(j)}(k)]\Delta_t^{-1}. \end{aligned} \quad (6)$$

By using the forward Euler method for discretization, we also have

$$e_{1(j)}(k+1) = e_{1(j)}(k) + \Delta_t e_{2(j)}(k) \Rightarrow e_{1(j)}(k+1) - e_{1(j)}(k) = \Delta_t e_{2(j)}(k). \quad (7)$$

Applying (7) into (6) produces:

$$\begin{aligned} \Delta\sigma_{(j)}(k) &= [f(x_{(j)}(k)) + \Delta_t B (u_{(j)}(k) + d_{(j)}(k)) \\ &\quad - x_{2d}(k+1) + (c\Delta_t - 1)e_{2(j)}(k)]\Delta_t^{-1}. \end{aligned} \quad (8)$$

2.1.1. Equivalent Control

Now let us consider the following dynamics to be induced by the learning process:

$$S_{(j)}(k) = \Delta\sigma_{(j)}(k) + \mu\sigma_{(j)}(k) = 0, \quad (9)$$

where $\mu > 0$ is a control parameter. Substituting (8) into (9) yields:

$$\begin{aligned} [f(x_{(j)}(k)) + \Delta_t B (u_{(j)}(k) + d_{(j)}(k)) - x_{2d}(k+1) \\ + (c\Delta_t - 1)e_{2(j)}(k)]\Delta_t^{-1} + \mu\sigma_{(j)}(k) = 0. \end{aligned} \quad (10)$$

In nominal conditions of the system under no model error or disturbance, the equivalent control of the system is described by:

$$u_{eq(j)}(k) = (\Delta_t B)^{-1} \left[-f(x_{(j)}(k)) + x_{2d}(k+1) - (c\Delta_t - 1)e_{2(j)}(k) - \Delta_t \mu \sigma_{(j)}(k) \right]. \quad (11)$$

2.1.2. Learning Control

In the learning step, to drive the system trajectories toward the sliding surface (9) regardless of disturbances, an iterative learning scheme is introduced using the stored data from previous iterations as:

$$\begin{aligned} u_{ilc(j)}(k) &= (\Delta_t B)^{-1} \sum_{i=0}^{j-1} \lambda S_{(i)}(k) \\ &= u_{ilc(j-1)}(k) - (\Delta_t B)^{-1} \lambda S_{(j-1)}(k), \end{aligned} \quad (12)$$

where the initial iterations $u_{ilc(0)}(k) = 0$ and $\lambda > 0$ are a design parameter for the learning rate.

From (11) and (12), the ILSMC law is described by:

$$u_{(j)}(k) = u_{eq(j)}(k) + u_{ilc(j)}(k). \quad (13)$$

We summarize the ILSMC design in the following theorem:

Theorem 1. For a discrete-time system (1) with sampling period Δ_t subject to iteration-invariant disturbance $d_{(j)}(k)$, under the iterative learning sliding mode control (13) comprising the equivalent control (11) and learning control (12), if the control parameter μ and learning rate λ are respectively chosen such that $0 < \mu < 2/\Delta_t$, $0 < \lambda < 2$ and $\lambda \neq 1$, then the tracking error (2) is driven to zero at a sufficiently large number of iterations and the control system is asymptotically stable.

Proof. By substituting (8) and (11)–(13) into (9), we obtain:

$$S_{(j)}(k) = - \sum_{i=0}^{j-1} \lambda S_{(i)}(k) + \Delta_t B d_{(j)}(k). \quad (14)$$

Similarly,

$$S_{(j-1)}(k) = - \sum_{i=0}^{j-2} \lambda S_{(i)}(k) + \Delta_t B d_{(j-1)}(k). \quad (15)$$

According to the Assumption, as $d_{(j)}(k)$ is iteration-invariant, from (14) and (15), we have:

$$\begin{aligned} S_{(j)}(k) - S_{(j-1)}(k) &= -\lambda S_{(j-1)}(k) \\ \Leftrightarrow S_{(j)}(k) &= (1 - \lambda) S_{(j-1)}(k) = (1 - \lambda)^2 S_{(j-2)}(k) \\ &= \dots = (1 - \lambda)^j S_{(0)}(k). \end{aligned} \quad (16)$$

From (16), the iterative learning algorithm converges to 0 at large values of the iteration number under the condition $|1 - \lambda| < 1$. Therefore, if the learning rate λ is selected to satisfy $0 < \lambda < 2$ and $\lambda \neq 1$, we have

$$\lim_{j \rightarrow \infty} S_{(j)}(k) = \lim_{j \rightarrow \infty} (1 - \lambda)^j S_{(0)}(k) = 0. \quad (17)$$

By substituting $\Delta \sigma_{(j)}(k)$ into $S_{(j)}(k)$, Equation (9) can be rewritten as

$$S_{(j)}(k) = \Delta_t^{-1} \sigma_{(j)}(k + 1) - (1 - \mu \Delta_t) \Delta_t^{-1} \sigma_{(j)}(k), \quad (18)$$

whereby $S_{(j)}(k) \rightarrow 0$, with a proper selection of the learning rate λ and at an adequate number of iterations j , the sliding function (3) becomes:

$$\begin{aligned}\sigma_{(j)}(k) &\rightarrow (1 - \mu\Delta_t)\sigma_{(j)}(k-1) \\ &= (1 - \mu\Delta_t)^2\sigma_{(j)}(k-2) = \dots = (1 - \mu\Delta_t)^k\sigma_{(j)}(0),\end{aligned}\quad (19)$$

where $\sigma_{(j)}(0)$ is the initial value of $\sigma_{(j)}(k)$ at the j th iteration. Therefore, given a positive constant μ with $0 < \mu < 2/\Delta_t$, the sliding function $\sigma_{(j)}(k)$ in (19) approaches zero at a sufficiently large value of k . Thus, since the sliding function $\sigma_{(j)}(k)$ as defined in (3) is driven to zero after some iterations j , a sliding mode is induced from the selection of parameter $c > 0$. It follows that

$$\lim_{j,k \rightarrow \infty} e_{(j)}(k) = 0. \quad (20)$$

Notably, the asymptotic convergence of the tracking error $e_{(j)}(k)$ here does not come from the switching of the control signal with a high discontinuous gain as in conventional SMC, but is a result of the proposed learning process (12). Hence, the sliding mode (3) induced for the tracking error can retain system robustness in the face of uncertainties and disturbances while avoiding the high-frequency switching of the control signal. This is the reason why the proposed ILSMC can achieve highly accurate tracking without control chattering. The tracking performance then depends on the convergence of the learning process, governed by the learning rate λ .

To verify the system stability, let us consider the control sliding function $\sigma_{(j)}(k)$ at iteration j . According to [35], the discrete-time control system will be asymptotically stable if, for all its entries $[\sigma_{(j)}(k)]$:

$$\begin{cases} [\sigma_{(j)}(k+1) - \sigma_{(j)}(k)]\text{Sign}([\sigma_{(j)}(k)]) < 0, \\ [\sigma_{(j)}(k+1) + \sigma_{(j)}(k)]\text{Sign}([\sigma_{(j)}(k)]) \geq 0, \end{cases} \quad (21)$$

where $\text{Sign}(\cdot)$ is the signum function.

To verify the above conditions, from (18), we have,

$$\sigma_{(j)}(k+1) = (1 - \mu\Delta_t)\sigma_{(j)}(k). \quad (22)$$

We obtain, accordingly

$$\sigma_{(j)}(k+1) - \sigma_{(j)}(k) = -\mu\Delta_t\sigma_{(j)}(k). \quad (23)$$

Therefore, the first condition of (21) is satisfactory as

$$-\mu\Delta_t[\sigma_{(j)}(k)]\text{Sign}([\sigma_{(j)}(k)]) < 0. \quad (24)$$

From (22), we also have

$$\sigma_{(j)}(k+1) + \sigma_{(j)}(k) = (2 - \mu\Delta_t)\sigma_{(j)}(k), \quad (25)$$

and with the choice $0 < \mu < 2/\Delta_t$, the second condition of (21) is also satisfactory since

$$(2 - \mu\Delta_t)[\sigma_{(j)}(k)]\text{Sign}([\sigma_{(j)}(k)]) \geq 0. \quad (26)$$

Therefore, the control system is asymptotically stable. The proof is completed. \square

Remark 1. From (16), to quickly induce a sliding surface, a high rate of convergence is required, subject to the condition $|1 - \lambda| < 1$. This condition is similar to the ILC convergence condition presented in the frequency domain [36]. On one hand, the closer λ is to 1, the faster the convergence in the learning step. On the other hand, under the effect of noise and nonrepeating disturbances, a rapid learning rate could affect robustness. In practice, one can choose λ as close to 1 for fast convergence and gradually lower this value if required to reduce the system sensitivity.

Remark 2. The learning process can be terminated upon satisfaction of a required tracking performance index (TPI), e.g., when the integral time absolute error (ITAE) of the control error satisfies the requirement for tracking performance for a specific task of the system.

3. System Description and Modeling

The UAV employed in this study was a quadcopter with a symmetric rigid structure and driven by four motors, as shown in Figure 2. For the quadcopter, the pitch angle, varied in accordance with the quadcopter's longitudinal motion, is controlled by adjusting the front and rear propellers' velocities, which generate the force F_1 and F_3 . Meanwhile, its lateral displacement is governed by the roll angle, which is controlled through the right and left rotors' speeds, resulting in the forces F_2 and F_4 . Finally, the yaw angle, associated with the UAV yaw motion, is regulated by the difference between torques generated by these pairs of rotors. In this work, we focused on the attitude tracking control, and thus, only the quadcopter orientation is considered here. The torques acting on the quadcopter include the thrust forces τ , the gyroscopic torques caused by the rotation of the quadcopter's rigid body τ_b and of four propellers τ_p , as well as the torque due to aerodynamic friction τ_a . Here, the propellers' gyroscopic effects and the drag from air resistance are considered external disturbances.

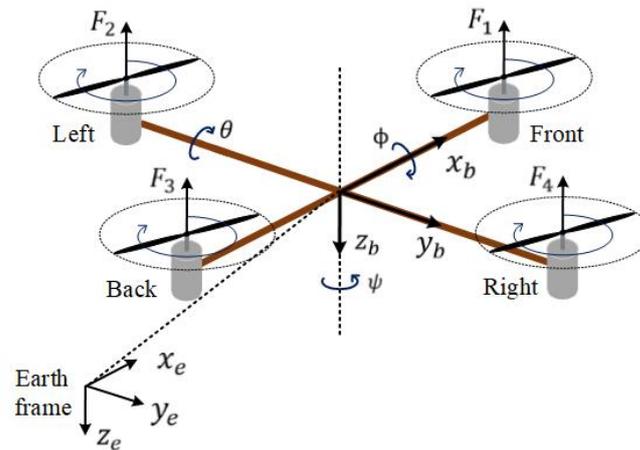


Figure 2. Configuration of a quadcopter.

3.1. Kinematics

As shown in the configuration in Figure 2, an earth frame, $\{x_e, y_e, z_e\}$, is fixed at the ground and a body frame, $\{x_b, y_b, z_b\}$, is attached to the CoG of the quadcopter, both with the z axis pointing downward. The position of the UAV's mass center in the earth frame is defined by a vector $P = (x, y, z)^T$. The UAV orientation is represented by angles $(\phi, \theta, \psi)^T$, corresponding to roll, pitch, and yaw motion, respectively. For attitude control, these angles are limited as $\phi \in [-\pi/2, \pi/2]$, $\theta \in [-\pi/2, \pi/2]$ and $\psi \in [-\pi, \pi]$. With respect to the earth frame, the orientation of the quadcopter is obtained a rotation transformation resulting from successively rotating around the x_b , y_b and z_b axes, and characterized by an orthonormal rotation matrix R [37]:

$$R = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\theta & c_\psi s_\theta c_\phi + s_\psi s_\theta \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\theta & s_\psi s_\theta c_\phi - c_\psi s_\theta \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \quad (27)$$

where s_x and c_x denote $\sin x$ and $\cos x$, respectively.

Denoting the angular velocity vector of the quadcopter in the body frame as $(\omega_\phi \ \omega_\theta \ \omega_\psi)^T$, the rotational kinematics can be obtained as follows [38]:

$$[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T = W^{-1}[\omega_\phi \ \omega_\theta \ \omega_\psi]^T, \quad (28)$$

where

$$W^{-1} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \theta \sec \theta \end{bmatrix}.$$

3.2. Quadcopter Dynamics

From the quadcopter description, the components of torque vector $\tau = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$, corresponding to rotation in the roll, pitch, and yaw directions, respectively, are calculated as

$$\tau_\phi = l(F_2 - F_4), \tag{29}$$

$$\tau_\theta = l(-F_1 + F_3), \tag{30}$$

$$\tau_\psi = l(-F_1 + F_3), \tag{31}$$

where l is the distance from each rotor to the CoG, and β is the apparent radius for converting the force into the yaw torque.

From (29)–(31), the control inputs are described as:

$$\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_z \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \\ F \end{bmatrix} = \begin{bmatrix} 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ -\beta & \beta & -\beta & \beta \\ l & l & l & l \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}, \tag{32}$$

where u_ϕ, u_θ , and u_ψ are the roll, pitch, and yaw torques, respectively; $F = \sum_{n=1}^4 F_n$ is the lift force, representing the total thrust utilizing from the four motors. As the only attitude of the quadcopter is controlled, u_z is assumed to balance with gravity.

The gyroscopic torque due to the rotation of the symmetric body of the quadcopter is described by [29]:

$$\tau_b = -SI[\omega_\phi \ \omega_\theta \ \omega_\psi]^T, \tag{33}$$

where

$$S = \begin{bmatrix} 0 & -\omega_\psi & \omega_\theta \\ \omega_\psi & 0 & -\omega_\phi \\ -\omega_\theta & \omega_\phi & 0 \end{bmatrix}$$

is a skew-symmetric matrix. As shown in the configuration in Figure 2 with the body frame assigned to the quadcopter, given a mass point m_i with its coordinates (x_i, y_i, z_i) in the body, the quadcopter’s inertia can be obtained as a diagonal matrix:

$$I = \begin{bmatrix} \sum_i (y_i^2 + z_i^2) m_i & 0 & 0 \\ 0 & \sum_i (x_i^2 + z_i^2) m_i & 0 \\ 0 & 0 & \sum_i (x_i^2 + y_i^2) m_i \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{34}$$

Accordingly, Equation (33) can be rewritten as

$$\tau_b = [(I_{yy} - I_{zz})\omega_\theta\omega_\psi \quad (I_{zz} - I_{xx})\omega_\phi\omega_\psi \quad (I_{xx} - I_{yy})\omega_\phi\omega_\theta]^T. \tag{35}$$

The gyroscopic torque due to the rotation of four propellers is determined as [29]:

$$\tau_p = [I_r\omega_r\omega_\theta \quad -I_r\omega_r\omega_\phi \quad 0]^T, \tag{36}$$

where I_r is the moment of inertia of the rotor of each motor, and $\omega_r = -\omega_{r1} + \omega_{r2} - \omega_{r3} + \omega_{r4}$ is the residual angular velocity, in which $\omega_{r1}, \dots, \omega_{r4}$ are the corresponding angular velocities of the propellers.

The air drag torque is calculated as [29]:

$$\tau_a = [k_{ax}\omega_\theta^2 \quad k_{ay}\omega_\phi^2 \quad k_{az}\omega_\psi^2]^T, \tag{37}$$

where k_{ax} , k_{ay} , and k_{az} are aerodynamic friction factors.

The dynamics of the quadcopter in attitude control can then be represented as:

$$[\dot{\omega}_\phi \quad \dot{\omega}_\theta \quad \dot{\omega}_\psi]^T = I^{-1}(\tau_b + \tau + \tau_p - \tau_a). \quad (38)$$

Now, if the propeller gyroscopic and aerodynamic torques are considered as external disturbances, i.e.,

$$d = [d_\phi \quad d_\theta \quad d_\psi]^T = \tau_p - \tau_a, \quad (39)$$

where d_ϕ , d_θ , and d_ψ are disturbance components, then substituting (32), (35), and (39) into (38) yields:

$$\dot{\omega}_\phi = I_{xx}^{-1}[(I_{yy} - I_{zz})\omega_\theta\omega_\psi + u_\phi + d_\phi], \quad (40)$$

$$\dot{\omega}_\theta = I_{yy}^{-1}[(I_{zz} - I_{xx})\omega_\phi\omega_\psi + u_\theta + d_\theta] \quad (41)$$

$$\dot{\omega}_\psi = I_{yy}^{-1}[(I_{zz} - I_{xx})\omega_\phi\omega_\psi + u_\theta + d_\theta]. \quad (42)$$

To express the quadcopter dynamics via the orientation angles, the model can be simplified by considering $[\omega_\phi, \omega_\theta, \omega_\psi] \approx [\dot{\phi}, \dot{\theta}, \dot{\psi}]$. This approximation is acceptable since a minor model error can be adequately addressed by a good controller. Accordingly, the quadcopter model is obtained as:

$$\ddot{\phi} = I_{xx}^{-1}[(I_{yy} - I_{zz})\dot{\theta}\dot{\psi} + u_\phi + d_\phi], \quad (43)$$

$$\ddot{\theta} = I_{yy}^{-1}[(I_{zz} - I_{xx})\dot{\phi}\dot{\psi} + u_\theta + d_\theta], \quad (44)$$

$$\ddot{\psi} = I_{zz}^{-1}[(I_{xx} - I_{yy})\dot{\phi}\dot{\theta} + u_\psi + d_\psi]. \quad (45)$$

3.3. Discrete-Time Model

In the discrete-time domain, by considering the difference approximation for first and second derivatives using the forward Euler method, the transformed discrete-time model can be obtained as below:

$$\begin{aligned} \phi(k+2) &= 2\phi(k+1) - \phi(k) + I_{xx}^{-1}(I_{yy} - I_{zz})[\theta(k+1) - \theta(k)][\psi(k+1) - \psi(k)] \\ &\quad + \Delta_t^2 I_{xx}^{-1}[u_\phi(k) + d_\phi(k)], \end{aligned} \quad (46)$$

$$\begin{aligned} \theta(k+2) &= 2\theta(k+1) - \theta(k) + I_{yy}^{-1}(I_{zz} - I_{xx})[\phi(k+1) - \phi(k)][\psi(k+1) - \psi(k)] \\ &\quad + \Delta_t^2 I_{yy}^{-1}[u_\theta(k) + d_\theta(k)], \end{aligned} \quad (47)$$

$$\begin{aligned} \psi(k+2) &= 2\psi(k+1) - \psi(k) + I_{zz}^{-1}(I_{xx} - I_{yy})[\phi(k+1) - \phi(k)][\theta(k+1) - \theta(k)] \\ &\quad + \Delta_t^2 I_{zz}^{-1}[u_\psi(k) + d_\psi(k)]. \end{aligned} \quad (48)$$

Now, consider the system state vector $x(k) = [x_1(k) \quad x_2(k)]^T$ defined by:

$$x_1(k) = [\phi_1(k) \quad \theta_1(k) \quad \psi_1(k)]^T = [\phi(k) \quad \theta(k) \quad \psi(k)]^T, \quad (49)$$

$$x_2(k) = [\phi_2(k) \quad \theta_2(k) \quad \psi_2(k)]^T = \left[\frac{\phi(k+1) - \phi(k)}{\Delta_t} \quad \frac{\theta(k+1) - \theta(k)}{\Delta_t} \quad \frac{\psi(k+1) - \psi(k)}{\Delta_t} \right]^T. \quad (50)$$

From (46)–(50), we obtain the UAV state equation in discrete time of the form (1) as:

$$x_1(k+1) = x_1(k) + \Delta_t x_2(k), \quad (51)$$

$$x_2(k+1) = f(x(k)) + \Delta_t B[u(k) + d(k)], \quad (52)$$

where $f(x(k)) = x_2(k) + \Delta_t I^{-1}\tau_b(k)$ and $B = I^{-1}$.

$$\hat{u}_{eq(j)}(k) = D^{-1} \circ (\Delta_t B)^{-1} \{-f(x_{(j)}(k)) - \Delta_t B H(k) + x_{2d}(k+1) - (c\Delta_t - 1)e_{2(j)}(k) - \Delta_t \mu \sigma_{(j)}(k)\}. \tag{59}$$

In the learning step, the iterative learning term is computed as

$$\hat{u}_{ilc(j)}(k) = \hat{u}_{ilc(j-1)}(k) - D^{-1} \circ (\Delta_t B)^{-1} \lambda S_{(j-1)}(k), \tag{60}$$

where $S_{(j-1)}(k)$ is obtained from the learning process at a previous iteration ($j - 1$) as per (15).

This finally leads to the integrated iterative learning sliding mode control law (13) for the quadcopter:

$$\hat{u}_{(j)}(k) = \hat{u}_{eq(j)}(k) + \hat{u}_{ilc(j)}(k). \tag{61}$$

4.3. Implementation Procedure

In summary, a step-by-step procedure to implement the proposed control scheme is summarized as:

- Step 1: Declare $I_{xx}, I_{yy}, I_{zz}, K_p, K_i, K_d, c, \mu, \lambda$.
- Step 2: Set $x_d(k), j = 0$, and $\hat{u}_{ilc(j)}(k) = 0$.
- Step 3: Compute the ILSMC $\hat{u}_{(j)}(k)$ from (61) as a reference to the inner loop.
- Step 4: Compute, from the measured states $x_{(j)}(k), e_{(j)}(k), \sigma_{(j)}(k), S_{(j)}(k)$, and the selected TPI.
- Step 5: Check if the tracking performance requirement is met to terminate the learning process. Otherwise, proceed to Step 6.
- Step 6: Set $j = j + 1$, update $\hat{u}_{ilc(j)}(k)$ from (60), then return to Step 3.

5. Simulation Results

This section provides the simulation results of the proposed ILSMC design. The parameters used for simulation were obtained from the 3DR Solo drone [29], as listed in Table 1. The selected control parameters are described in Table 2. Here, in the learning process, a suitable value for λ was chosen to obtain a fast convergence rate, so $S_{(j)}(k)$ was driven to zero quickly. In the control phase, coefficients c_ϕ, c_θ, c_ψ and μ were chosen by the desired error dynamics described in (3). Initially, the iterative learning control signal was set to zero, $u_{ilc(0)}(k) = 0$, and then updated after each iteration. To evaluate performance of the proposed controller, we compared it with other available techniques including the PD feedback controller, PD-type ILC [19], adaptive twisting sliding mode controller (ATSMC) [29], and adaptive finite-time control scheme (AFTC) [30].

Table 1. Parameters of the 3DR Solo drone.

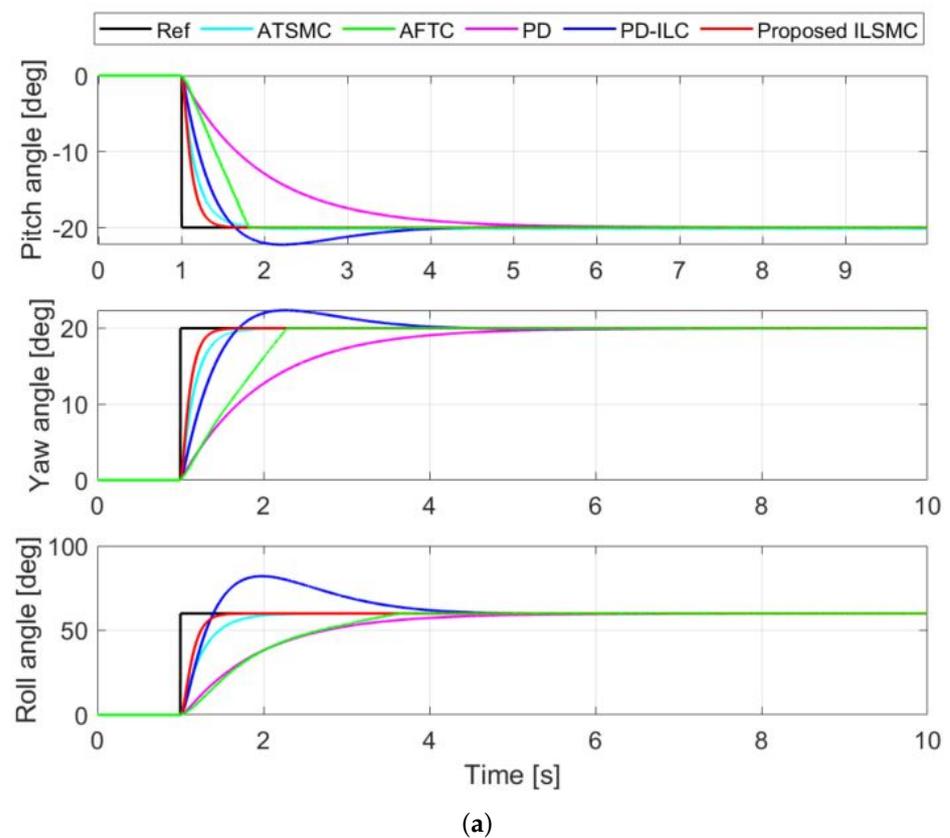
Parameter	Value	Unit
m	1.5	kg
l	0.205	m
g	9.81	m/s ²
I_{xx}	9.1×10^{-3}	kg m ²
I_{yy}	16.4×10^{-3}	kg m ²
I_{zz}	24.1×10^{-3}	kg m ²

Table 2. Control parameters.

Parameter	Value	Parameter	Value
c_ϕ	50	μ	10
c_θ	50	λ	0.9
c_ψ	20	-	-

5.1. Step Response in Nominal Conditions

In this section, the performance of the proposed controller is evaluated via the step responses in nominal conditions where disturbances and uncertainties were set to zero. The desired reference attitude angles were set to $\phi_d = -20^\circ$, $\theta_d = 20^\circ$, and $\psi_d = 60^\circ$ at 1 s. The simulation results of the step responses and control signals are shown in Figure 4, in which the black step signal is the desired angle, and responses of ATSMC, AFTC, PD, PD-ILC, and the proposed ILSMC controllers are depicted in cyan, green, magenta, blue, and red, respectively. It can be seen from Figure 4 for all three orientation angles that while ATSMC and AFTC provide some oscillations in the control, and PD presents a slow response, both iterative-learning-based techniques, ILSMC and PD-ILC, exhibit fast responses with zero steady-state error. PDILC, however, incurs a large overshoot, whereas ILSMC is able to maintain the desired dynamics without overshoot owing to the merits of sliding mode control. Notably, the fast response of ILSMC in comparison to ATSMC, AFTC, PD, ILC, and PD is attributed to the choice of $c = \text{diag}(c_\phi, c_\theta, c_\psi)$ and μ . A faster transient response, however, requires more control efforts that may exceed the physical limits imposed by the motors and power supply for the drone. Moreover, the proposed controller is chattering-free in the steady state.

**Figure 4.** Cont.

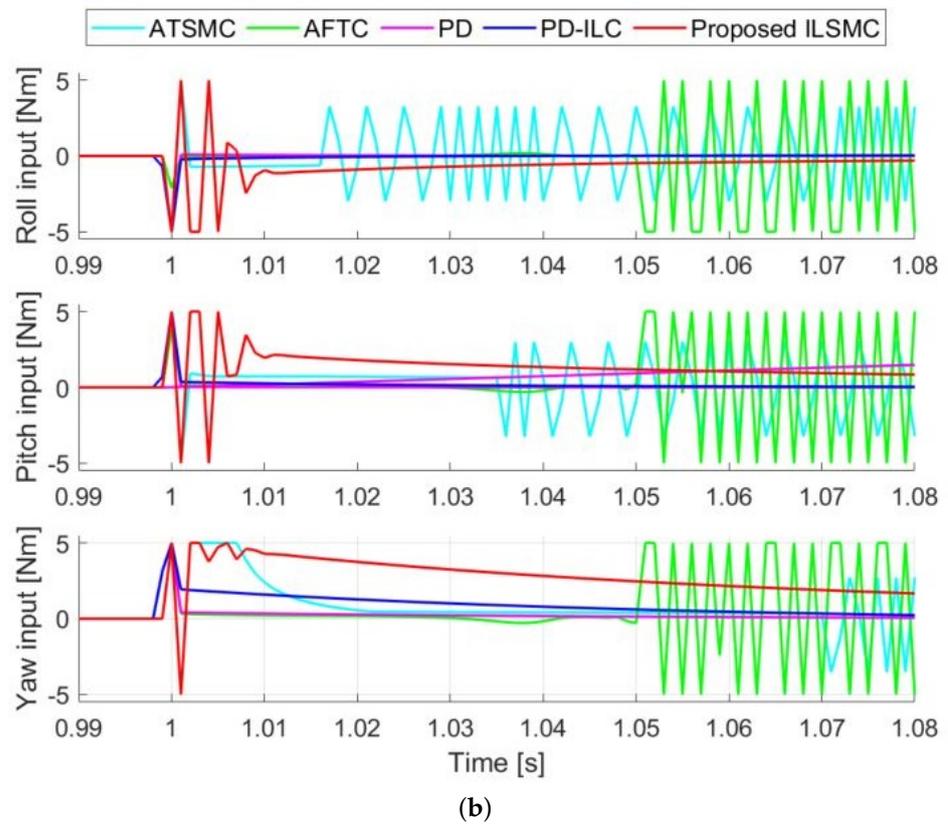


Figure 4. Step response in nominal conditions: (a) step response and (b) control effort.

5.2. Trajectory Tracking Performance under Disturbances and Uncertainties

To evaluate the tracking performance of ILSMC with the presence of uncertainties and disturbances caused by load variations, the reference attitude angles in this simulation were set, in degrees, as below:

$$\begin{aligned}
 \phi_d(k) &= 20 - 20\sin(2k), \\
 \theta_d(k) &= 20 + 20\sin(2k), \\
 \psi_d(k) &= 20 + 60\sin(2k).
 \end{aligned}
 \tag{62}$$

For performance evaluation, the system was injected with a disturbance at $t = 10$ s whose components were:

$$d_\phi = d_\theta = d_\psi = -0.2.
 \tag{63}$$

Considering 20% loading conditions, the model uncertainties were introduced by setting:

$$\hat{I}_{xx} = 1.2I_{xx}, \quad \hat{I}_{yy} = 1.2I_{yy}, \quad \hat{I}_{zz} = 1.2I_{zz},
 \tag{64}$$

where \hat{I}_{xx} , \hat{I}_{yy} , and \hat{I}_{zz} are the estimation of I_{xx} , I_{yy} , and I_{zz} , respectively.

Figure 5 shows the tracking performance of the attitude angles, while Table 3 presents the TPI, for which the integral time absolute error (ITAE) is adopted here for all angles. It can be seen that the PD controller cannot cope with disturbances with large tracking errors at $t = 10$ s and high values of ITAE. As with PDILC, it can suppress disturbances but suffers from control overshoot and a noticeable error. Both the ATSMC and AFTC techniques present relatively accurate tracking performance with a small ITAE, between 1.40 and 5.03. The proposed ILSMC presents a relatively large tracking error at the first iteration (since $u_{ilc(0)} = 0$), but owing to the learning ability, the tracking error decreases with increasing iterations by updating the iterative learning control term after each iteration. As the tracking performance is improved significantly, at the last iteration, ILSMC has the

smallest ITAE among the considered techniques. Additionally, the absolute error is also the smallest, almost zero in steady state, as shown in the magnified figure, demonstrating the advantage of the proposed ILSMC.

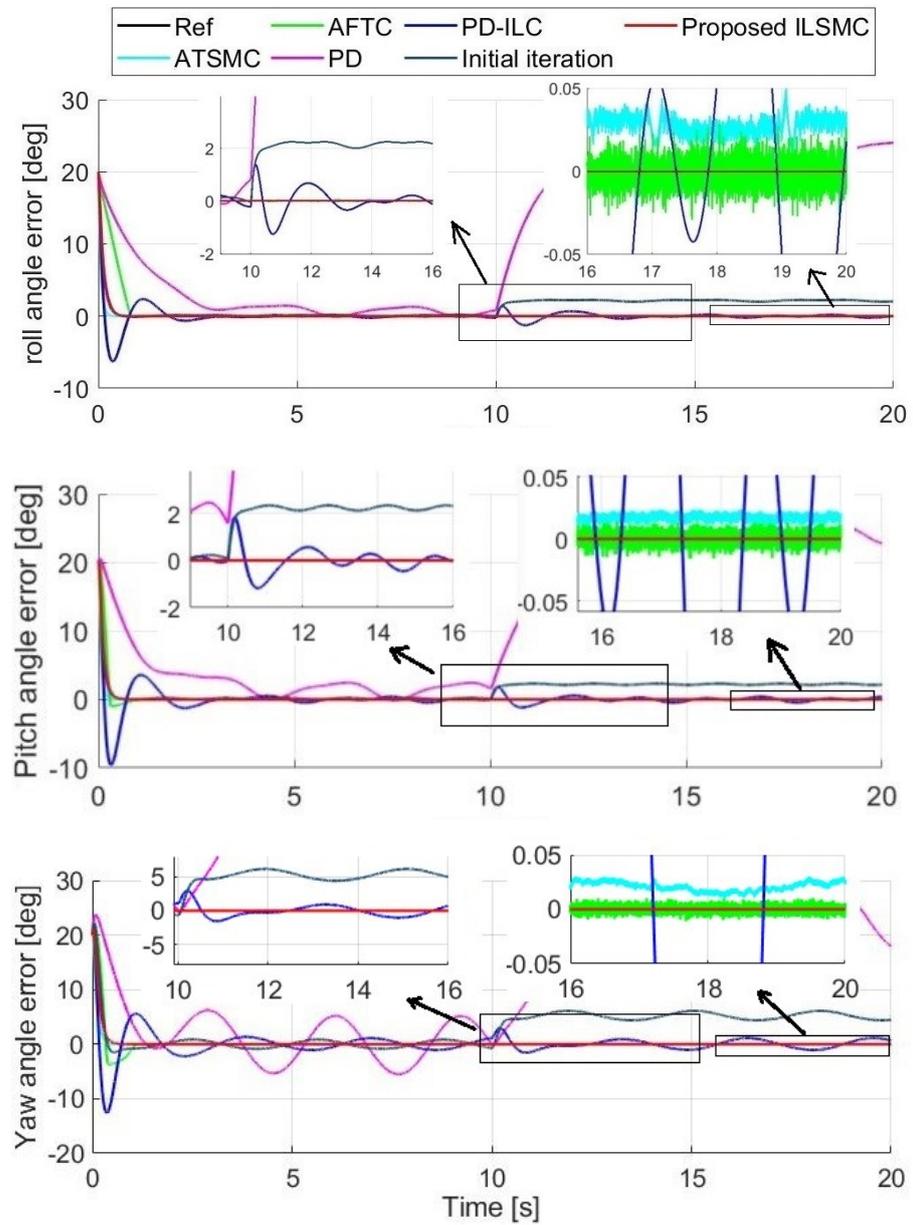


Figure 5. Performance in the presence of disturbances and uncertainties.

Table 3. ITAE of UAV attitude control angles.

UAV Angle (degrees)	ATSMC	AFTC	PD	PD-ILC	ILSMC
Roll	5.03	2.03	3344.2	39.91	0.255
Pitch	3.87	1.40	3468.4	55.60	0.261
Yaw	3.47	2.42	3323.0	149.89	0.444

Figure 6 shows the control efforts in the presence of disturbances and uncertainties. It can be seen that its magnitude increases after 10 s, which implies that more energy is required to compensate for the external disturbances. More importantly, the control efforts

of ILSMC display oscillation only in the transient state, but no chattering in the steady state, which is beneficial for practical implementation.

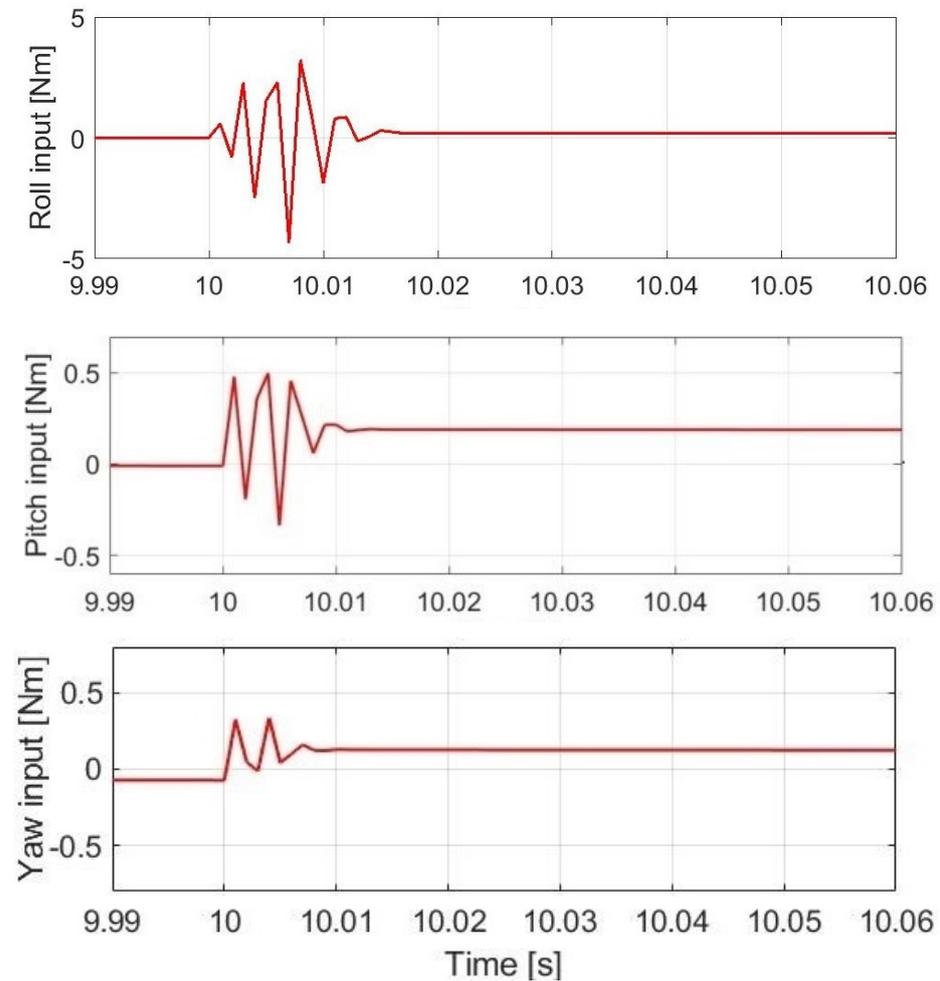


Figure 6. Control efforts in the presence of disturbances and uncertainties.

To evaluate the effect of the proposed learning mechanism, the ITAE values were computed for ILSMC after each iteration with different values of λ . The results up to 15 iterations are presented in Figure 7. They indicate that the ITAE of the all three attitude angle errors quickly decreases and converges to zero after several iterations. To induce a fast system sliding mode, a higher rate of convergence must be selected. It can be seen in Figure 7 that this can be obtained when λ is close to one. In this work, $\lambda = 0.9$ was chosen to achieve the desired control performance.

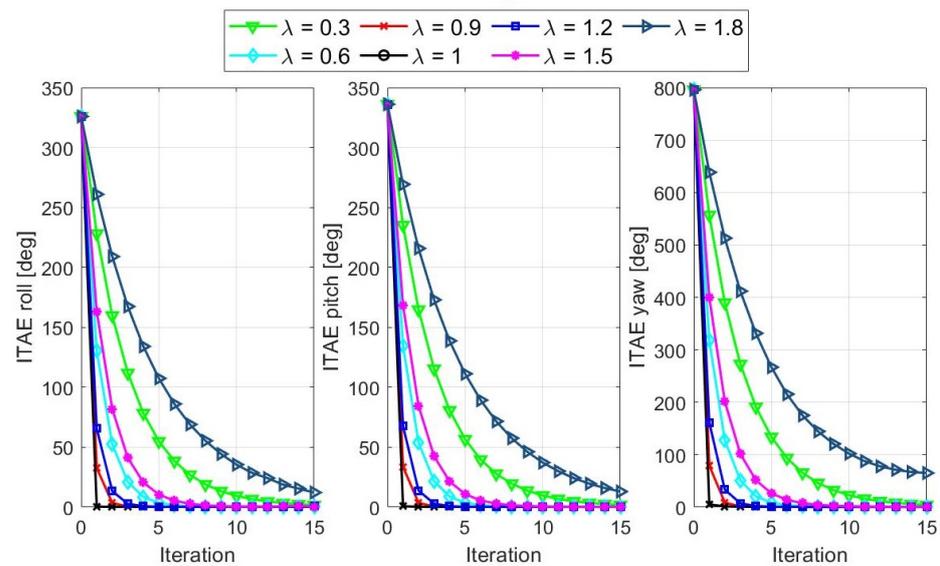


Figure 7. ITAEs of the tracking errors after each iteration.

6. Experimental Validation

In this section, we evaluate the performance of the combined ILSMC and PID control algorithm in the trajectory tracking problem for our UAV testbed, in which a built-in PID was already employed.

6.1. Experimental Setup

The setup for the experiments is shown in Figure 8, using a 3DR Solo drone with its parameters described in Table 1 [39]. It consists of two Cortex M4 168 MHz processors used for low-level control and one ARM Cortex A9 processor used for running the Arducopter flight operating system. The drone is equipped with a camera, a laser scanner, and environmental sensors for data acquisition. During the experiments, communication data, including control reference signals and drone sensor outputs, were transmitted to the ground control station via the local network established by the drone system. Mission Planner software was connected to the network to upload the flight plan to the drone and log flight data for analysis. In the experiments, the PID gains were set to their default values implemented in the 3DR Solo. From the desired and actual roll, pitch, and yaw angles, the tracking error was computed.



Figure 8. System architecture.

6.2. Real-Time Data Validation Results

The steps for conducting the experiments to validate the trajectory tracking performance of the proposed ILSMC were as follows: First, a trajectory was predefined with a starting point being set at the home position of the drone in an absolute frame of reference,

as depicted in Figure 9. After that, the longitude, latitude, and altitude of the waypoints forming the trajectory were imported into Mission Planner, as depicted in Figure 10. Next, those waypoints were uploaded to the 3DR Solo to fly automatically, as shown in Figure 11. Then, the reference and actual attitude angles were logged by Mission Planner, as shown in Figure 12, for comparison. The errors between those angles were used to update the iterative learning term. Finally, the trajectories of the 3DR Solo drone obtained by using the built-in PID controller were compared with the results obtained using ATSMC, AFTC, PD-ILC, and the proposed ILSMC.

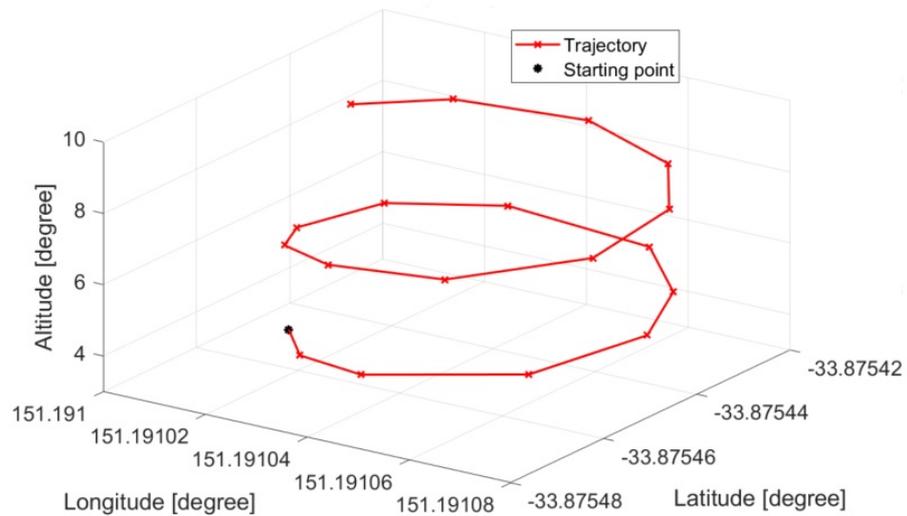


Figure 9. Predefined trajectory.



Figure 10. Imported trajectory.

The comparison was performed by setting the references obtained from the 3DR Solo drone under similar control settings as in the simulation. Figure 13 shows typical comparison results for the UAV roll, pitch, and yaw responses. It can be seen that the deviation between the reference and the actual roll angle controlled by the built-in PID is relatively high due to disturbances. Advanced techniques can improve the tracking performance in which ILSMC remains the best, as indicated by its smallest tracking error, as can be seen clearly in the insets in the figure. The results obtained confirm the validity and efficiency of the proposed approach.



Figure 11. Flying 3DR Solo drone.



Figure 12. Logged flight data.

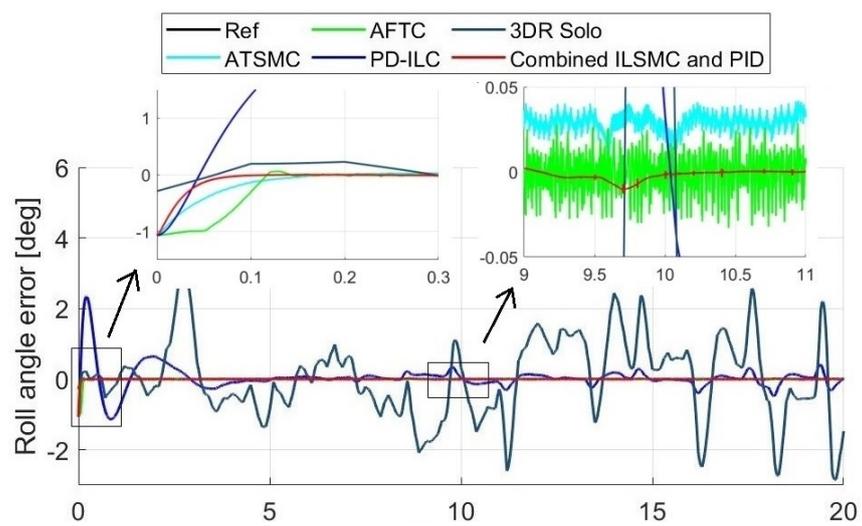


Figure 13. Cont.

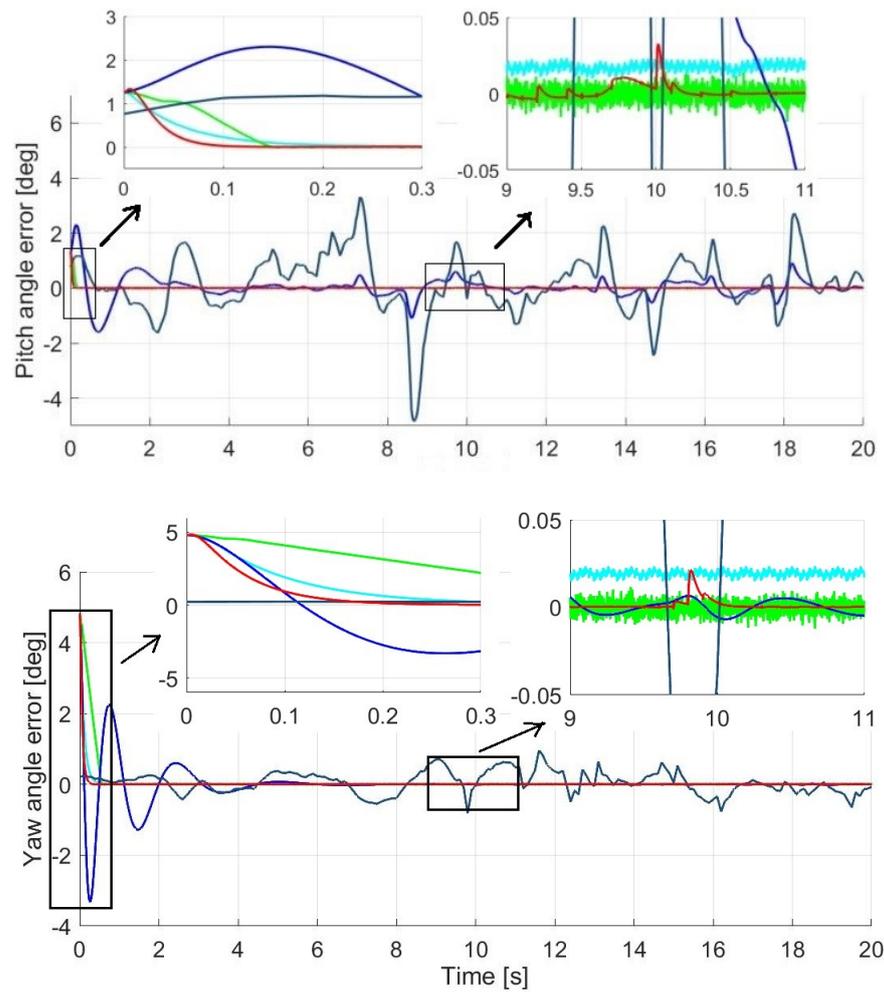


Figure 13. Tracking performance with real-time data.

From the real-time experiments, the recorded control efforts are shown in Figure 14, where the steady-state yaw torque is a constant as the quadcopter was controlled to lift up with a linearly increasing height while performing a circular trajectory during the test.

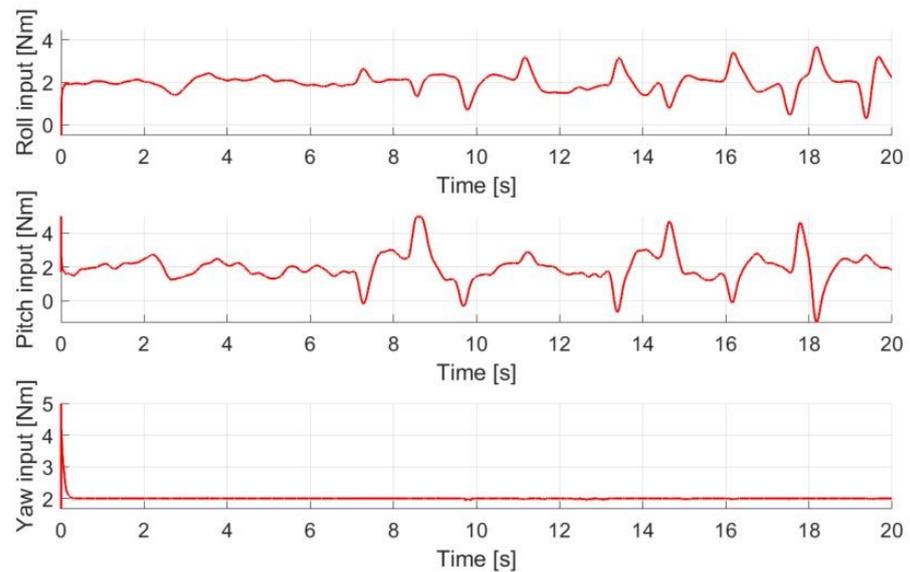


Figure 14. Control efforts with real-time data.

7. Conclusions

We proposed an effective control technique called ILSMC to address the tracking control problem experienced by quadcopters when subject to disturbances and uncertainties. The control signal consists of an equivalent term to control the system states within the desired sliding surface, and an iterative learning term to drive the system states toward the sliding surface and then remain in the sliding surface despite the presence of uncertainties and disturbances. The iterative learning signal is updated following some iterations to improve the tracking performance by using the data acquired from previous iterations. The simulation results showed, in the case of disturbances and uncertainties, that the iterative learning sliding mode controller presented the smallest tracking errors compared to some other existing control techniques used for quadcopter control. For UAVs with built-in PID controllers, the proposed control scheme can be integrated in a cascade structure to improve the trajectory tracking accuracy and robustness. Field tests were performed, and validation with real-time experimental data was conducted to confirm the advantages of the proposed approach. Our future work will focus on extending the learning mechanism to enable the control of multiple UAVs for real-time formation.

Author Contributions: L.V.N. and Q.P.H. proposed the main idea and the methodology of the research; L.V.N. performed the simulation, conducted the experiments, and wrote the original manuscript; M.D.P. and Q.P.H. reviewed the manuscript; the manuscript was directed and edited by Q.P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the Vingroup Science and Technology Scholarship (VSTS) Program for Overseas Study for Master's and Doctoral Degrees. The VSTS Program is managed by VinUniversity and sponsored by Vingroup.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SMC	Sliding mode control
ILC	Iterative learning control
ILSMC	Iterative learning sliding mode control
UAV	Unmanned aerial vehicle
PID	Proportional–integral–derivative
FL	Feedback linearization
CoG	Center of gravity
TPI	Tracking performance index
ITAE	Integral time absolute error
ATSMC	Adaptive twisting sliding mode control
AFTC	Finite-time control scheme.

References

1. Grzonka, S.; Grisetti, G.; Burgard, W. A fully autonomous indoor quadrotor. *IEEE Trans. Robot.* **2012**, *28*, 90–100. [[CrossRef](#)]
2. Grisetti, G.; Stachniss, C.; Burgard, W. Non-linear constraint network optimization for efficient map learning. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 428–439. [[CrossRef](#)]
3. Flint, M.; Polycarpou, M.; Fernandez-Gaucherand, E. Cooperative control for multiple autonomous UAV's searching for targets. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002.
4. Brown, A.; Anderson, D. Trajectory optimization for high-altitude long endurance UAV maritime radar surveillance. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2406–2421. [[CrossRef](#)]
5. Metni, N.; Hamel, T. A UAV for bridge inspection: Visual servoing control law with orientation limits. *Autom. Constr.* **2007**, *17*, 3–10. [[CrossRef](#)]
6. Herwitz, S.R.; Johnson, L.F.; Dunagan, S.E.; Higgins, R.G.; Sullivan, D.V.; Zheng, J.; Lobitz, B.M.; Leung, J.G.; Gallmeyer, B.A.; Aoyagi, M.; et al. Imaging from an unmanned aerial vehicle: Agricultural surveillance and decision support. *Comput. Electron. Agric.* **2004**, *44*, 49–61. [[CrossRef](#)]

7. Choi, Y.C.; Ahn, H.S.; Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 1179–1192. [[CrossRef](#)]
8. Park, J.; Kim, Y.; Kim, S. Landing site searching and selection algorithm development using vision system and its application to quadrotor. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 488–503. [[CrossRef](#)]
9. Prasenjit, M.; Steven, W. Direct adaptive feedback linearization for quadrotor control. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, USA, 13–16 August 2012; American Institute of Aeronautics and Astronautics: Minneapolis, MN, USA, 2012.
10. L'afflito, A.; Anderson, R.B.; Mohammadi, K. An introduction to nonlinear robust control for unmanned quadrotor aircraft. *IEEE Control Syst. Mag.* **2018**, *38*, 102–121. [[CrossRef](#)]
11. Qingtong, W.; Honglin, W.; Qingxian, W.; Mou, C. Backstepping-based attitude control for a quadrotor UAV using nonlinear disturbance observer. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015.
12. Chincholkar, S.H.; Jiang, W.; Chan, C.-Y. Continuous nonsingular terminal sliding mode control of DC–DC boost converters subject to time-varying disturbances. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2020**, *67*, 92–96. [[CrossRef](#)]
13. Ma, H.; Xiong, Z. Sliding mode control for uncertain discrete-time systems using an adaptive reaching law. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2021**, *68*, 722–726. [[CrossRef](#)]
14. Ali, S.U.; Samar, R.; Shah, M.Z.; Bhatti, A.I.; Munawar, K.; Al-Sggaf, U.M. Lateral guidance and control of UAVs using second-order sliding modes. *Aerosp. Sci. Technol.* **2016**, *49*, 88–100. [[CrossRef](#)]
15. Sankaranarayanan, V.; Mahindrakar, A.D. Control of a class of underactuated mechanical systems using sliding modes. *IEEE Trans. Robot.* **2009**, *25*, 459–467. [[CrossRef](#)]
16. Tian, B.; Yin, L.; Wang, H. Finite-time reentry attitude control based on adaptive multivariable disturbance compensation. *IEEE Trans. Ind. Electron.* **2015**, *62*, 5889–5898. [[CrossRef](#)]
17. Qi, Y.; Zhang, S.; Jiang, F.; Zhou, H.; Tao, D.; Li, X. Siamese local and global networks for robust face tracking. *IEEE Trans. Image Process.* **2020**, *29*, 9152–9164. [[CrossRef](#)] [[PubMed](#)]
18. Zhou, M.; Feng, Y.; Xue, C.; Han, F. Deep convolutional neural network based fractional-order terminal sliding-mode control for robotic manipulators. *Neurocomputing* **2020**, *416*, 143–151. [[CrossRef](#)]
19. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A survey of iterative learning control. *IEEE Control Syst. Mag.* **2006**, *26*, 96–114.
20. Yu, M.; Li, C. Robust Adaptive Iterative Learning Control for Discrete-Time Nonlinear Systems with Time-Iteration-Varying Parameters. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1737–1745. [[CrossRef](#)]
21. Uchiyama, M. Formation of high speed motion pattern of mechanical arm by trial. *Trans. Soc. Instrum. Control. Eng.* **1978**, *14*, 706–712. [[CrossRef](#)]
22. Arimoto, S.; Kawamura, S.; Miyazaki, F. Iterative learning control for robot systems. In Proceedings of the IECON, Tokyo, Japan, 22–26 October 1984; pp. 393–398.
23. Moore, K.L. Iterative learning control for deterministic systems. In *Advances in Industrial Control*; Springer: London, UK, 1993.
24. Tayebi, A.; Abdul, S.; Zaremba, M.B.; Ye, Y. Robust iterative learning control design: Application to a robot manipulator. *IEEE/ASME Trans. Mechatron.* **2008**, *13*, 608–613. [[CrossRef](#)]
25. Mezghani, M.; Roux, G.; Cabassud, M.; Lann, M.V.L.; Dahhou, B.; Casamatta, G. Application of iterative learning control to an exothermic semibatch chemical reactor. *IEEE Trans. Control Syst. Technol.* **2002**, *10*, 822–834. [[CrossRef](#)]
26. Zhu, Q.; Song, F.; Xu, J.; Liu, Y. An internal model based iterative learning control for wafer scanner systems. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2073–2084. [[CrossRef](#)]
27. Nikooienejad, N.; Maroufi, M.; Moheimani, R. Iterative learning control for video-rate atomic force microscopy. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 2127–2138. [[CrossRef](#)]
28. Madani, T.; Benallegue, A. Adaptive control via backstepping technique and neural networks of a quadrotor helicopter. *IFAC Proc. Vol.* **2008**, *41*, 6513–6518. [[CrossRef](#)]
29. Hoang, V.T.; Phung, M.D.; Ha, Q.P. Adaptive twisting sliding mode control for quadrotor unmanned aerial vehicles. In Proceedings of the 2017 Asian Control Conference (ASCC 2017), Gold Coast, Australia, 17–20 December 2017; pp. 671–676.
30. Tian, B.; Lu, H.; Zuo, Z.; Zong, Q. Adaptive finite-time attitude tracking of quadrotors with experiments and comparisons. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9428–9438. [[CrossRef](#)]
31. He, X.; Guo, D.; Leang, K.K. Repetitive control design and implementation for periodic motion tracking in aerial robots. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5101–5108.
32. Dong, J.; He, B. Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors* **2019**, *19*, 24. [[CrossRef](#)] [[PubMed](#)]
33. Adlakha, R.; Zheng, M. An Optimization-Based Iterative Learning Control Design Method for UAV's Trajectory Tracking. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1353–1359.
34. Chen, Y.; Moore, K.L. Harnessing the nonrepetitiveness in iterative learning control. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 3, pp. 3350–3355.
35. Sarpturk, S.Z.; Istefanopulos, Y.; Kaynak, O. On the stability of discrete-time sliding mode control systems. *IEEE Trans. Autom. Control* **1987**, *32*, 930–932. [[CrossRef](#)]
36. Norrlöf, M.; Gunnarsson, S. Time and frequency domain convergence properties in iterative learning control. *Int. J. Control* **2002**, *75*, 1114–1126. [[CrossRef](#)]

-
37. Guilherme, V.R.; Manuel, G.O.; Francisco, R.R. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. *Automatica* **2010**, *46*, 29–39.
 38. Craig, J.J. *Introduction to Robotics—Mechanics and Control*, 2nd ed.; Addison-Wesley Publishing Company, Inc.: Reading, MA, USA, 1989.
 39. Hoang, V.T.; Phung, M.D.; Dinh, T.H.; Ha, Q.P. System architecture for real-time surface inspection using multiple UAVs. *IEEE Sens. J.* **2020**, *40*, 4430–4441. [[CrossRef](#)]