Article

Iterative Learning Sliding Mode Control for UAV Trajectory Tracking

Lanh V. Nguyen [1]*, M.D. Phung [1,2], and Q.P. Ha [1]

[1] School of Electrical and Data Engineering, University of Technology Sydney (UTS), 15 Broadway, Ultimo NSW 2007, Australia; {manhduong.phung; quang.ha}@uts.edu.au
[2] VNU University of Engineering and Technology (VNU-UET), Vietnam National University, Hanoi (VNU), 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
* Correspondence: vanlanh.nguyen@uts.edu.au

Abstract: This paper presents a novel iterative learning sliding mode controller (ILSMC) with application to 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 involves only 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 good 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. 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: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

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 required. 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. In the face of large uncertainties and disturbances, FL-based approaches may lead to poor tracking performance, and other advanced control laws are required. The adaptive feedback linearization controller is applied in [9], allowing for adjustments of the control parameters to enhance control performance. Robust control methods have also been developed to improve control performance [10]. A backstepping controller is 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 bound is required in these techniques. Adaptive SMC has been developed to overcome this requirement [16]. This approach, on the other hand, still reveals the main disadvantage of SMC, i.e., control signals usually present a chattering behavior, especially when dealing with large uncertainties and disturbances that 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 on system robustness for control signal smoothening. Deep learning-based SMC has recently been proposed to handle highly complex and time-varying uncertainties, such as deep convolutional neural network-based fractional-order terminal SMC [17] and integrated deep learning recurrent neural network with terminal SMC [18]. Although these methods achieve high 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 in 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 high tracking performance even in the face of large model uncertainties and disturbances. Indeed, unlike non-learning control techniques, the system in ILC is reset to zero after the system has reached 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 to iterative learning control in robotic systems [22] and later been developed for industrial control [23]. In the last 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 information from previous executions, and hence, can 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 of 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 has been 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 has been introduced where fuzzy logic is used to tune the control parameters, high tracking performance is hard to reached in comparison to other rigorous control strategies. In [33], optimization-based ILC is 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, thus more robust and effective. The main contributions of this work includes (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 the quadcopter trajectory tracking.

This paper is structured as follows. The control development for ILSMC is presented in Section 2. Convergence and stability analysis of the proposed learning algorithm is also provided in this section. Next, Section 3 presents system modelling, 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 feed-forward control signal so that the system can learn from the previous responses to improve tracking performance and eliminate disturbance repeatedly after each iteration. The basic structure of an ILC is depicted in the diagram of Fig. 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.

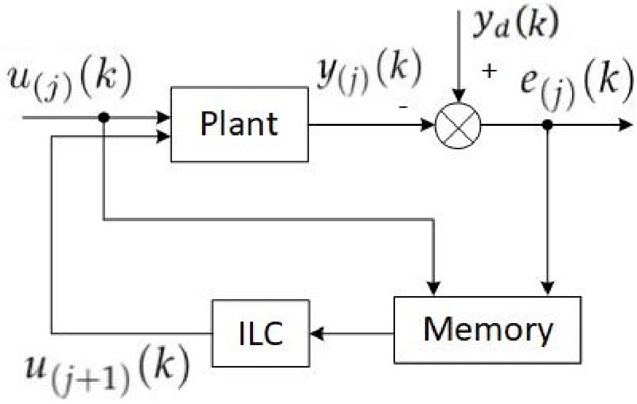


Figure 1. Basic structure of ILC

2.1. ILSMC design

Consider the following general discrete-time control system:

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

where k is the time instant, 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 $x_{(j)}(k) = \begin{bmatrix} x_{1_{(j)}}(k) & x_{2_{(j)}}(k) \end{bmatrix}^T \in \mathbb{R}^{2m}$, 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 $f(.)$ 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. *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, $\delta_{(1)}(k) = \delta_{(2)}(k) = ... = \delta_{(j)}(k)$ for all $k = \{0, 1, ..., N-1\}$.*

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

Assumption. *In this study, 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 will be involved, and hence *a priori* information of the disturbance bound is not required while chattering can be fully alleviated.

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

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

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

Let us define the control sliding function as below:

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

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

By denoting

$$\Delta_{\sigma_{(j)}}(k) = \left[\sigma_{(j)}(k+1) - \sigma_{(j)}(k)\right]\Delta_t^{-1}, \tag{4}$$

we have from (3) :

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

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

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

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

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

Applying (7) into (6) gives:

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

2.1.1. Equivalent Control

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

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

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

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

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

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

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 below:

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

where the initial iteration $u_{ilc_{(0)}}(k) = 0$ and $\lambda > 0$ is a design parameter for the learning rate.

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

$$u_{(j)}(k) = u_{eq_{(j)}}(k) + u_{ilc_{(j)}}(k). \quad (13)$$

We summarize the ILSMC design in the following theorem.

Theorem. *For the 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), (11), (12) and (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:

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

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

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

By substituting $\Delta_{\sigma_{(j)}}(k)$ into $S_{(j)}(k)$, Eq. (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, as $S_{(j)}(k) \to 0$ with a proper selection of the learning rate λ and at an adequate number of iterations j, the sliding function (3) becomes:

$$\begin{aligned}\sigma_{(j)}(k) &\to (1-\mu\Delta_t)\sigma_{(j)}(k-1) \\ &= (1-\mu\Delta_t)^2\sigma_{(j)}(k-2) = ... = (1-\mu\Delta_t)^k\sigma_{(j)}(0),\end{aligned} \tag{19}$$

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) will approach zero at a sufficiently large value of k. Thus, since the sliding funtion $\sigma_{(j)}(k)$ 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

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

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 face of uncertainties and disturbances while avoiding the high-frequency switching of the control signal. That 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 $[\sigma_{(j)}(k)]$:

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

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

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

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

We obtain, accordingly

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

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

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

From (22), we also have

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

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

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

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

Remark 1. *From (16), to fast 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 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 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 on tracking performance for a specific task of the system.*

3. System description and modelling

The UAV employed in this article is a quadcopter having a symmetric rigid structure and driven by four motors, as shown in Figure 2. For the quadcopter, the pitch angle, varying 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 controled 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 focus on the attitude tracking control, and thus, only the quadcopter orientation is concerned 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 as external disturbances.

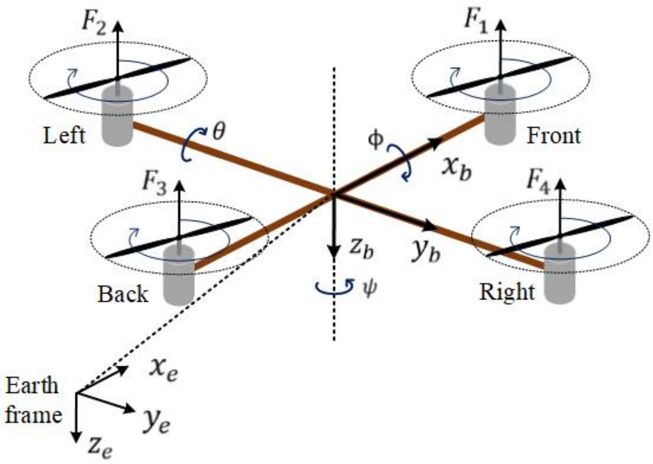


Figure 2. Configuration of a quadcopter

3.1. Kinematics

As shown in the configuration of Fig. 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 $(\phi, \theta, \psi)^T$, respectively corresponding to roll, pitch, and yaw motion. 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 x_b, y_b and z_b axes, and characterized by an orthonormal rotation matrix $\mathscr{R}$ [37]:

$$\mathscr{R} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\theta & c_\psi s_\theta c_\phi + s_\psi s_\theta \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\theta & s_\psi s_\theta c_\phi - c_\psi s_\theta \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{27}$$

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]:

$$\begin{bmatrix}\dot{\phi} & \dot{\theta} & \dot{\psi}\end{bmatrix}^T = W^{-1}\begin{bmatrix}\omega_\phi & \omega_\theta & \omega_\psi\end{bmatrix}^T, \tag{28}$$

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, are calculated as

$$\tau_\phi = l(F_2 - F_4), \tag{29a}$$
$$\tau_\theta = l(-F_1 + F_3), \tag{29b}$$
$$\tau_\psi = \beta(-F_1 + F_2 - F_3 + F_4), \tag{29c}$$

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), the control inputs are given as:

$$\begin{bmatrix}u_\phi \\ u_\theta \\ u_\psi \\ u_z\end{bmatrix} = \begin{bmatrix}\tau_\phi \\ \tau_\theta \\ \tau_\psi \\ F\end{bmatrix} = \begin{bmatrix}0 & l & 0 & -l \\ -l & 0 & l & 0 \\ -\beta & \beta & -\beta & \beta \\ l & l & l & l\end{bmatrix}\begin{bmatrix}F_1 \\ F_2 \\ F_3 \\ F_4\end{bmatrix}, \tag{30}$$

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

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

$$\tau_b = -\mathscr{S} I\begin{bmatrix}\omega_\phi & \omega_\theta & \omega_\psi\end{bmatrix}^T, \tag{31}$$

where

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

is a skew-symmetric matrix. As shown in the configuration of Fig. 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:

$$I = \begin{bmatrix}\sum_i(y_i^2 + z_i^2)m_i & 0 & 0 \\ 0 & \sum_i(x_i^2 + z_i^2)m_i & 0 \\ 0 & 0 & \sum_i(x_i^2 + y_i^2)m_i\end{bmatrix} = \begin{bmatrix}I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz}\end{bmatrix}. \tag{32}$$

Accordingly, Eq. (31) can be rewritten as

$$\tau_b = \begin{bmatrix}(I_{yy} - I_{zz})\omega_\theta\omega_\psi & (I_{zz} - I_{xx})\omega_\phi\omega_\psi & (I_{xx} - I_{yy})\omega_\phi\omega_\theta\end{bmatrix}^T. \tag{33}$$

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

$$\tau_p = \begin{bmatrix}I_r\omega_r\omega_\theta & -I_r\omega_r\omega_\phi & 0\end{bmatrix}^T, \tag{34}$$

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

The air drag torque is calculated as [29]:

$$\tau_a = \begin{bmatrix} k_{ax}\omega_\theta^2 & k_{ay}\omega_\phi^2 & k_{az}\omega_\psi^2 \end{bmatrix}^T, \tag{35}$$

199 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:

$$\begin{bmatrix} \dot{\omega}_\phi & \dot{\omega}_\theta & \dot{\omega}_\psi \end{bmatrix}^T = I^{-1}(\tau_b + \tau + \tau_p - \tau_a). \tag{36}$$

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

$$d = \begin{bmatrix} d_\phi & d_\theta & d_\psi \end{bmatrix}^T = \tau_p - \tau_a, \tag{37}$$

where d_ϕ, d_θ and d_ψ are disturbance components correspondingly, then substituting (30), (33), and (37) and into (36) yields:

$$\dot{\omega}_\phi = I_{xx}^{-1}\left[(I_{yy} - I_{zz})\omega_\theta\omega_\psi + u_\phi + d_\phi\right], \tag{38a}$$
$$\dot{\omega}_\theta = I_{yy}^{-1}\left[(I_{zz} - I_{xx})\omega_\phi\omega_\psi + u_\theta + d_\theta\right], \tag{38b}$$
$$\dot{\omega}_\psi = I_{zz}^{-1}\left[(I_{xx} - I_{yy})\omega_\phi\omega_\theta + u_\psi + d_\psi\right]. \tag{38c}$$

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:

$$\ddot{\phi} = I_{xx}^{-1}\left[(I_{yy} - I_{zz})\dot{\theta}\dot{\psi} + u_\phi + d_\phi\right], \tag{39a}$$
$$\ddot{\theta} = I_{yy}^{-1}[(I_{zz} - I_{xx})\dot{\phi}\dot{\psi} + u_\theta + d_\theta], \tag{39b}$$
$$\ddot{\psi} = I_{zz}^{-1}\left[(I_{xx} - I_{yy})\dot{\phi}\dot{\theta} + u_\psi + d_\psi\right]. \tag{39c}$$

200 *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:

$$\begin{aligned}\phi(k+2) = {} & 2\phi(k+1) - \phi(k) + I_{xx}^{-1}(I_{yy} - I_{zz})[\theta(k+1) - \theta(k)][\psi(k+1) - \psi(k)] \\ & + \Delta_t^2 I_{xx}^{-1}\left[u_\phi(k) + d_\phi(k)\right],\end{aligned} \tag{40a}$$

$$\begin{aligned}\theta(k+2) = {} & 2\theta(k+1) - \theta(k) + I_{yy}^{-1}(I_{zz} - I_{xx})[\phi(k+1) - \phi(k)][\psi(k+1) - \psi(k)] \\ & + \Delta_t^2 I_{yy}^{-1}[u_\theta(k) + d_\theta(k)],\end{aligned} \tag{40b}$$

$$\begin{aligned}\psi(k+2) = {} & 2\psi(k+1) - \psi(k) + I_{zz}^{-1}(I_{xx} - I_{yy})[\phi(k+1) - \phi(k)][\theta(k+1) - \theta(k)] \\ & + \Delta_t^2 I_{zz}^{-1}\left[u_\psi(k) + d_\psi(k)\right].\end{aligned} \tag{40c}$$

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

$$x_1(k) = \begin{bmatrix} \phi_1(k) & \theta_1(k) & \psi_1(k) \end{bmatrix}^T = \begin{bmatrix} \phi(k) & \theta(k) & \psi(k) \end{bmatrix}^T, \tag{41a}$$
$$x_2(k) = \begin{bmatrix} \phi_2(k) & \theta_2(k) & \psi_2(k) \end{bmatrix}^T = \begin{bmatrix} \frac{\phi(k+1)-\phi(k)}{\Delta_t} & \frac{\theta(k+1)-\theta(k)}{\Delta_t} & \frac{\psi(k+1)-\psi(k)}{\Delta_t} \end{bmatrix}^T. \tag{41b}$$

201 From (40) and (41), we obtain the UAV state equation in discrete-time of the form (1) as
202 below:

$$x_1(k+1) = x_1(k) + \Delta_t x_2(k), \tag{42a}$$
$$x_2(k+1) = f(x(k)) + \Delta_t B[u(k) + d(k)], \tag{42b}$$

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

4. Integrated ILSMC for UAV attitude control

The proposed ILSMC is now applied to the outer loop of a quadcopter with a built-in PID controller in the inner loop for flight control. Here, the ILSMC is integrated in cascade control to improve the performance of the UAV trajectory tracking in dealing with noise, non-repeating uncertainties and disturbances. Figure 3 shows the block diagram of the proposed controller wherein the reference signal of a feedback controller is generated by the ILSMC signal $\hat{u}_{(j)}(k)$ at a time instant k.

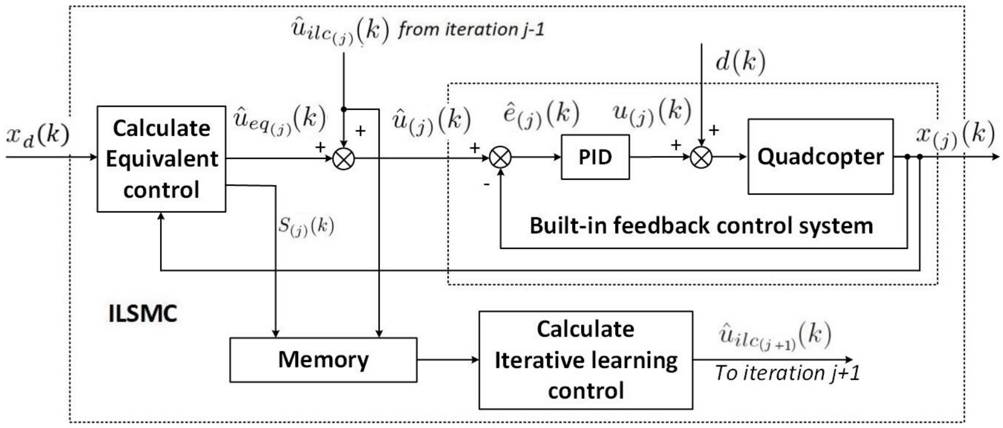


Figure 3. ILSMC in cascade PID-controlled quadcopter

4.1. Inner-loop PID controller

As shown in Fig. 3, the output of the quadcopter PID controller is computed as

$$u_{(j)}(k) = K_p \circ \hat{e}_{(j)}(k) + K_i \Delta_t \circ \sum_{\kappa=1}^{k} \hat{e}_{(j)}(\kappa) \quad + K_d \Delta_t^{-1} \circ [\hat{e}_{(j)}(k) - \hat{e}_{(j)}(k-1)], \tag{43}$$

where $\circ$ denotes the elementwise Hadamard product, $K_p = \left[K_{p_\phi}\ K_{p_\theta}\ K_{p_\psi}\right]^T$, $K_i = \left[K_{i_\phi}\ K_{i_\theta}\ K_{i_\psi}\right]^T$, and $K_d = \left[K_{d_\phi}\ K_{d_\theta}\ K_{d_\psi}\right]^T$ are PID control parameters. The error of the PID feedback loop $\hat{e}_{(j)}(k)$ is defined as

$$\hat{e}_{(j)}(k) = \hat{u}_{(j)}(k) - x_{(j)}(k), \tag{44}$$

where $\hat{u}_{(j)}(k)$ is the ILSMC control signal.

Substituting (44) into (43) yields:

$$u_{(j)}(k) = D \circ \hat{u}_{(j)}(k) + H(k), \tag{45}$$

where

$$D = K_p + K_i \Delta_t + K_d / \Delta_t, \tag{46}$$

$$H(k) = -D \circ x_{(j)}(k) + K_i \Delta_t \circ \sum_{\kappa=1}^{k-1} \hat{e}_{(j)}(\kappa) - K_d \circ \hat{e}_{(j)}(k-1)\Delta_t^{-1}. \tag{47}$$

4.2. Outer-loop ILSMC

As mentioned above, the proposed ILSMC is now added to the predesigned PID controller for improving the UAV tracking performance. Substituting (45) into (10) yields:

$$\Delta_t^{-1}[f(x_{(j)}(k)) + \Delta_t B(D \circ \hat{u}_{(j)}(k) + H(k) + d(k)) - x_{2d}(k+1) \\ + (c\Delta_t - 1)e_{2(j)}(k)] + \mu\sigma_{(j)}(k) = 0. \tag{48}$$

The equivalent control of the outer-loop is then given by:

$$\hat{u}_{eq_{(j)}}(k) = D^{-1} \circ (\Delta_t B)^{-1}\{-f(x_{(j)}(k)) - \Delta_t BH(k) + x_{2d}(k+1) \\ - (c\Delta_t - 1)e_{2(j)}(k) - \Delta_t\mu\sigma_{(j)}(k)\}. \tag{49}$$

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

$$\hat{u}_{ilc_{(j)}}(k) = \hat{u}_{ilc_{(j-1)}}(k) - D^{-1} \circ (\Delta_t B)^{-1}\lambda S_{(j-1)}(k), \tag{50}$$

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

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

$$\hat{u}_{(j)}(k) = \hat{u}_{eq_{(j)}}(k) + \hat{u}_{ilc_{(j)}}(k). \tag{51}$$

4.3. Implementation Procedure

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

- 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 (51) 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, go to Step 6.
- Step 6: Set $j = j + 1$, update $\hat{u}_{ilc_{(j)}}(k)$ from (50), then return to Step 3.

5. Simulation Results

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

5.1. Step response in nominal conditions

In this section, the performance of the proposed controller is evaluated via step responses in nominal conditions where disturbances and uncertainties are set to zeros. The desired reference attitude angles are set to $\phi_d = -20°$, $\theta_d = 20°$, and $\psi_d = 60°$ at 1s. Simulation results of step responses and control signals are shown in Fig. 4, in which the black step signal is the desired angle and responses of ATSMC, AFTC, PD,

Table 1: Parameters of the 3DR Solo drone

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

Table 2: Control paremeters

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

PD-ILC, and the proposed ILSMC controllers are depicted in cyan, green, magenta, blue, and red colors, respectively. It can be seen from Fig. 4 for all three orientation angles that while the ATSMC and AFTC provide some oscillations in the control, and the PD presents a slow response, both iterative learning-based techniques, the ILSMC and PD-ILC, exhibit fast responses with zero steady-state error. The 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 $c = \mathrm{diag}(c_\phi, c_\theta, c_\psi)$ and μ. A faster transient response, however, requires higher control efforts that may go beyond the physical limits imposed by the motors and power supply for the drone. Moreover, the proposed controller is chattering-free in the steady state.

5.2. Trajectory tracking performance under disturbances and uncertainties

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

$$\begin{aligned} \phi_d(k) &= 20 - 20sin(2k), \\ \theta_d(k) &= 20 + 20sin(2k), \\ \psi_d(k) &= 20 + 60sin(2k). \end{aligned} \tag{52}$$

For the sake of performance evaluation, the system is injected with a disturbance at $t = 10s$ whose components are:

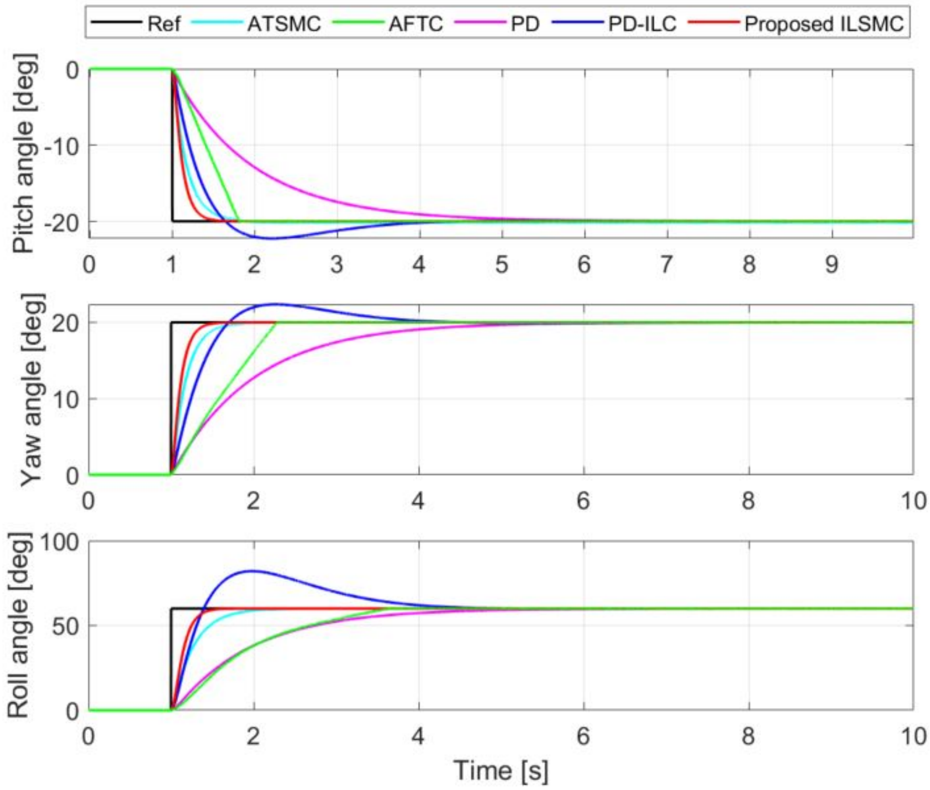
$$d_\phi = d_\theta = d_\psi = -0.2. \tag{53}$$

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

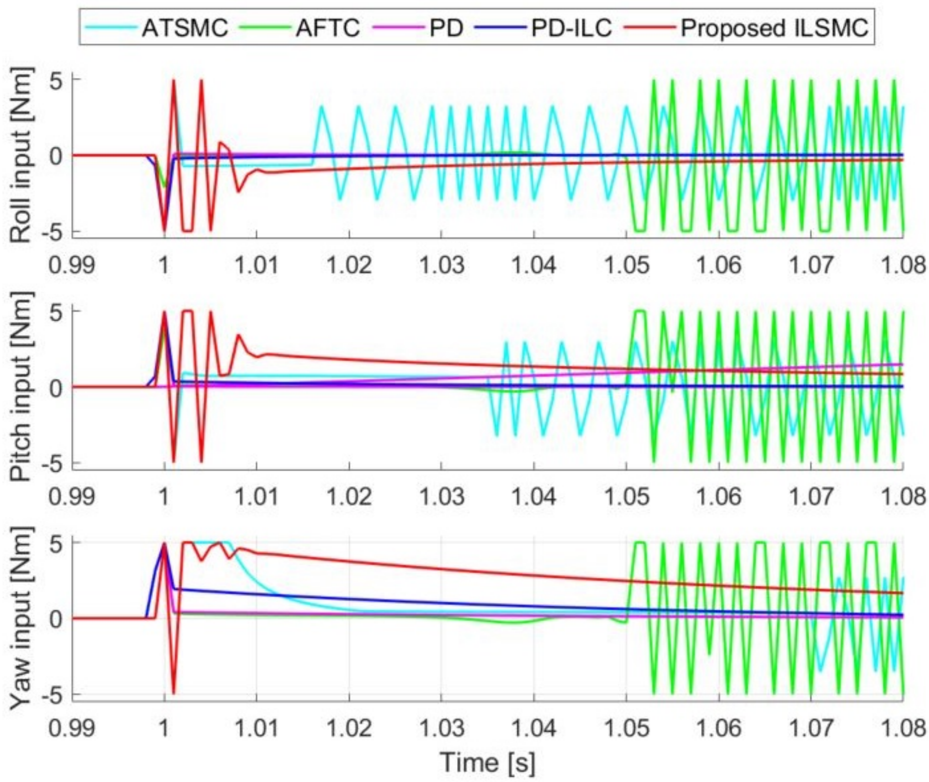
$$\hat{I}_{xx} = 1.2I_{xx}, \quad \hat{I}_{yy} = 1.2I_{yy}, \quad \hat{I}_{zz} = 1.2I_{zz}, \tag{54}$$

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



(a) Step response



(b) Control effort

Figure 4. Step response in nominal condition

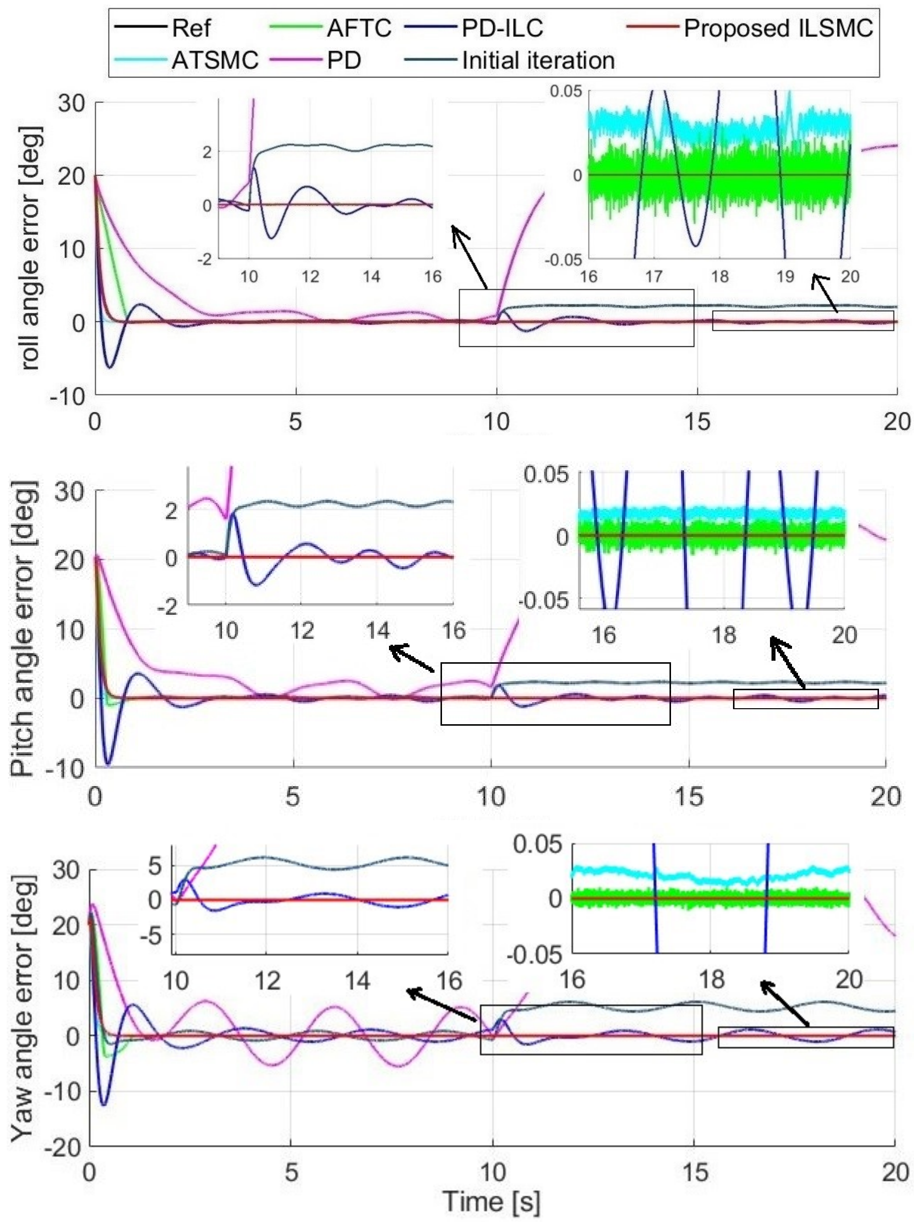


Figure 5. Performance in the presence of disturbances and uncertainties

Table 3: ITAE of UAV attitude control angles

UAV angle (deg)	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

with large tracking errors at $t = 10s$ 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 good tracking performance with 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 over iterations by updating the iterative learning control term after each iteration. As the tracking performance is improved significantly, at the last iteration, ILSMC results in a smallest ITAE among the considered techniques. Besides, the absolute error is also smallest, almost zero in steady state, as shown in the zoom-in figure, demonstrating the advantage of the proposed ILSMC.

Figure 6 shows the control efforts in the presence of disturbances and uncertainties. It can be seen that its magnitudes increase after 10s, 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.

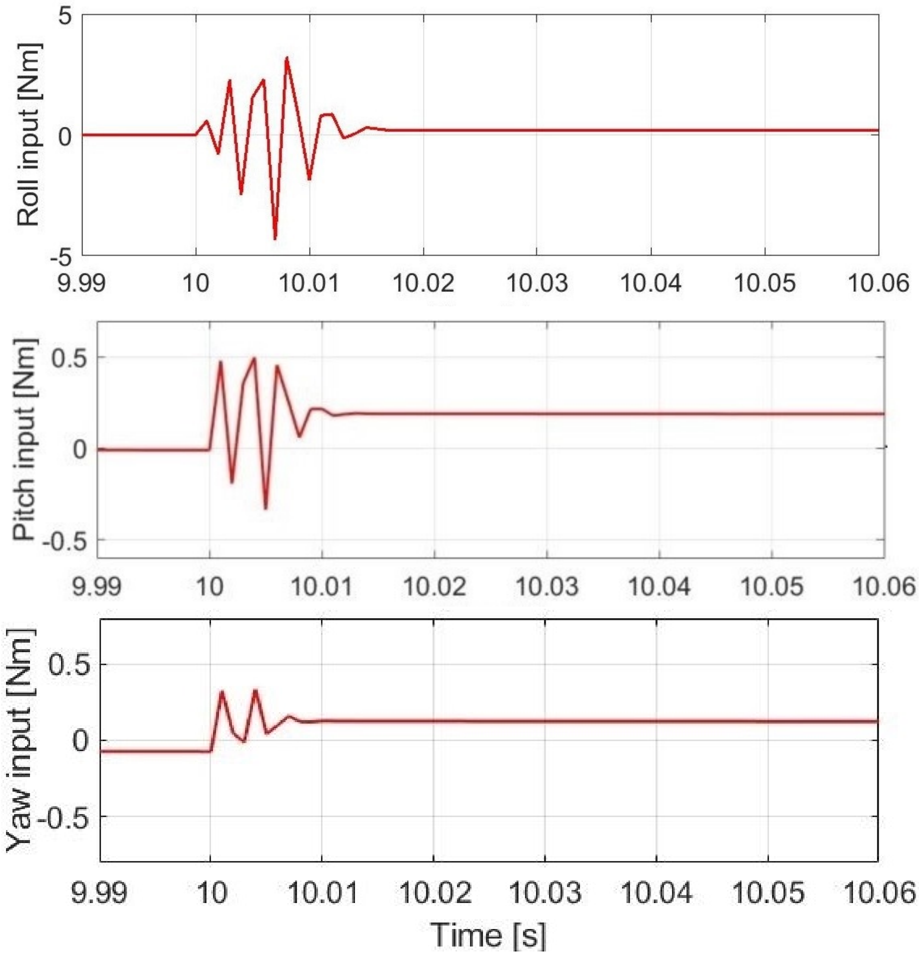


Figure 6. Control efforts in the presence of disturbances and uncertainties

To evaluate the effect of the proposed learning mechanism, the ITAE values are computed for ILSMC after each iteration with different values of λ. The results up to 15 iterations are presented in Fig. 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 is expected to select. It can be seen in Fig. 7 that this can be obtained when λ is close to 1. In this work, $\lambda = 0.9$ is chosen to achieve the desired control performance.

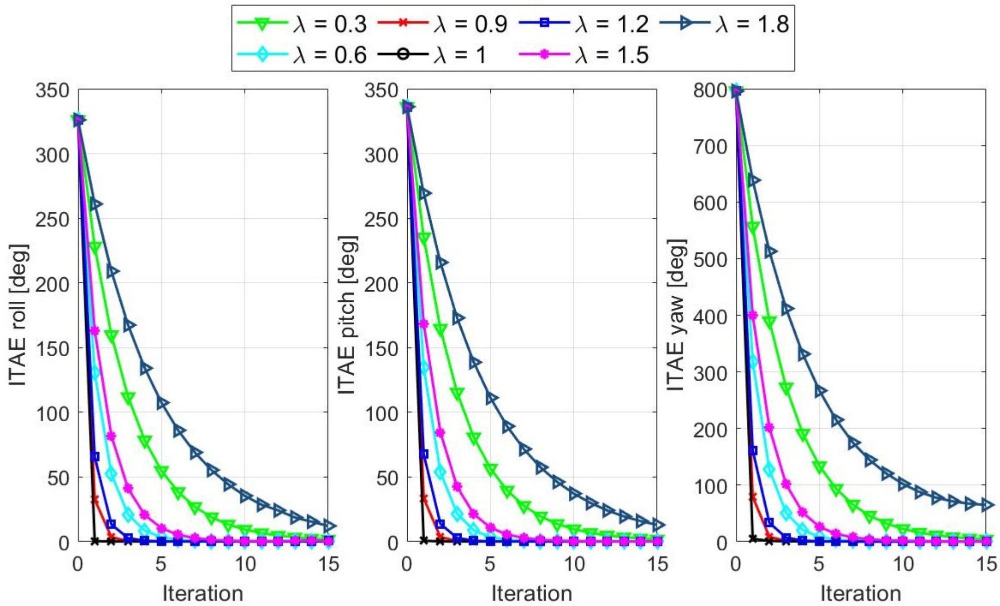


Figure 7. ITAEs of the tracking errors after each

6. Experimental Validation

This section evaluates 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 is already employed.

6.1. Experimental setup

The setup for experiments is shown in Fig. 8, using a 3DR Solo drone with its parameters given 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 experiments, communication data, including control reference signals and drone sensor outputs, are transmitted to the ground control station via the local network established by the drone system. The Mission Planner software is connected to the network to upload the flight plan to the drone and log flight data for analysis. In experiments, the PID gains are set to their default values implemented in the 3DR Solo. From the desired and actual roll, pitch, and yaw angles, the tracking error is computed.

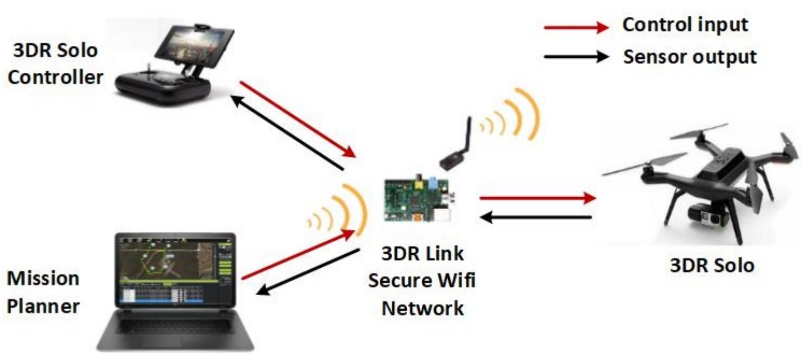


Figure 8. System architecture

6.2. Real-time data validation results

The steps for conducting experiments to validate the trajectory tracking performance of the proposed ILSMC are as follows. First, a trajectory is predefined with a starting point being set at the home position of the drone in an absolute frame of reference, as depicted in Fig. 9. After that, the longitude, latitude, and altitude of the waypoints forming the trajectory are imported into Mission Planner as depicted in Fig. 10. Next, those waypoints are uploaded to the 3DR Solo to fly automatically as shown in Fig. 11. Then, the reference and actual attitude angles are logged by Mission Planner as shown in Fig. 12 for comparison. The errors between those angles are used to update the iterative learning term. Finally, the trajectories of the 3DR Solo drone obtained by using the built-in PID controller are compared with the results obtained by using ATSMC, AFTC, PD-ILC, and the proposed ILSMC.

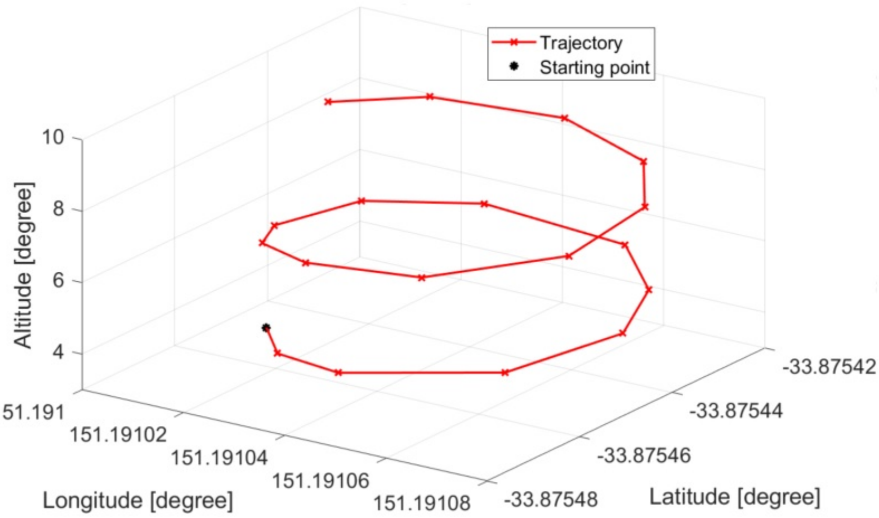


Figure 9. Predefined trajectory

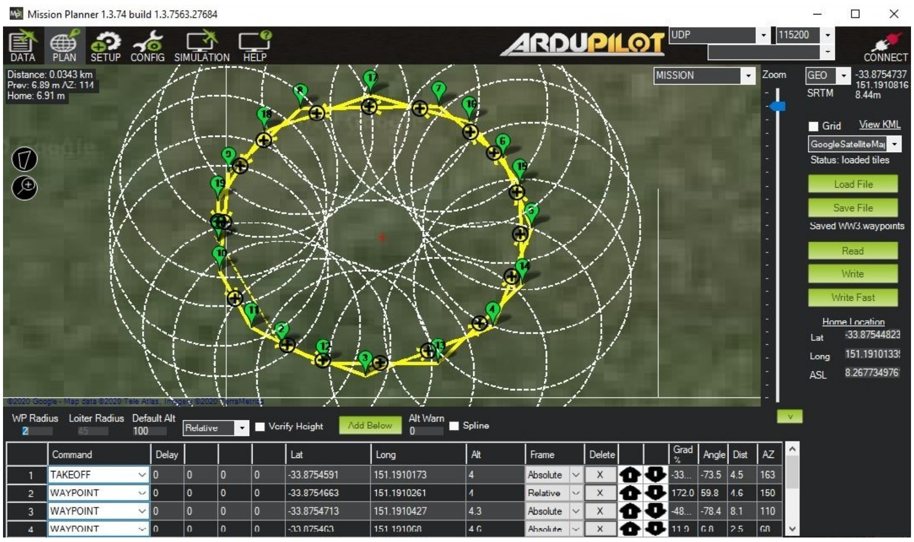


Figure 10. Imported trajectory

Figure 11. Flying 3DR Solo drone

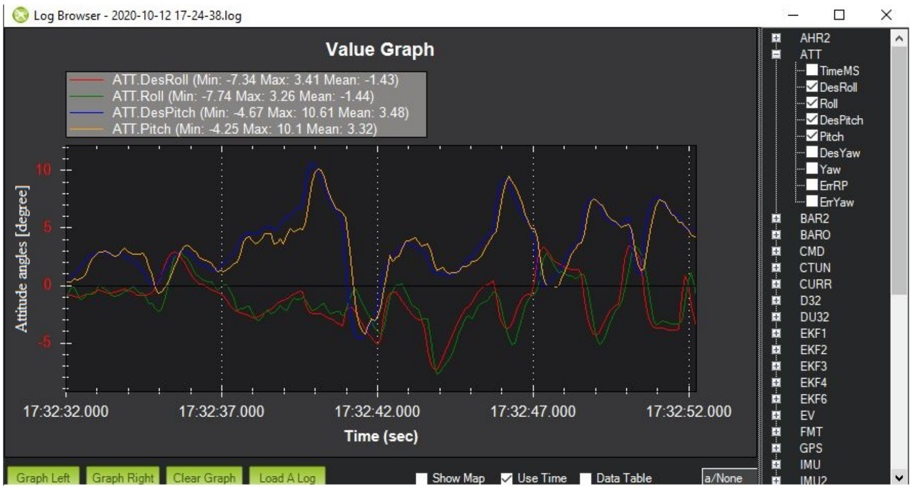


Figure 12. Logged flight data

The comparison is performed by setting the references obtained from the 3DR Solo drone under similar control settings as in the simulation. Figure 13 shows the comparison results typically 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. The advanced techniques can improve the tracking performance in which ILSMC remains the best by referring to its smallest tracking error, as can be seen clearly in the zoom-in insets. The results obtained confirm the validity and efficiency of the proposed approach.

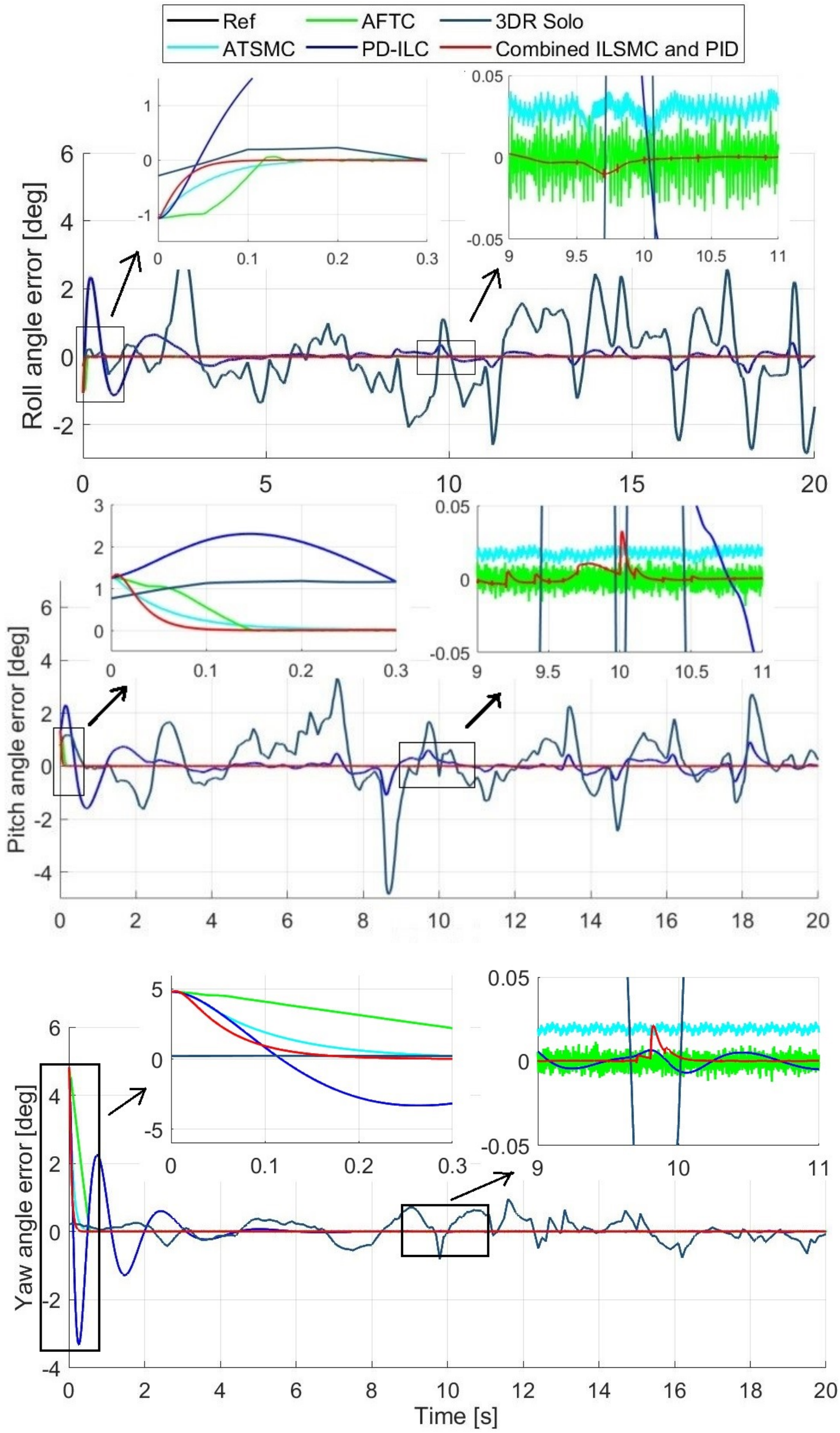


Figure 13. Tracking performance with real-time data

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

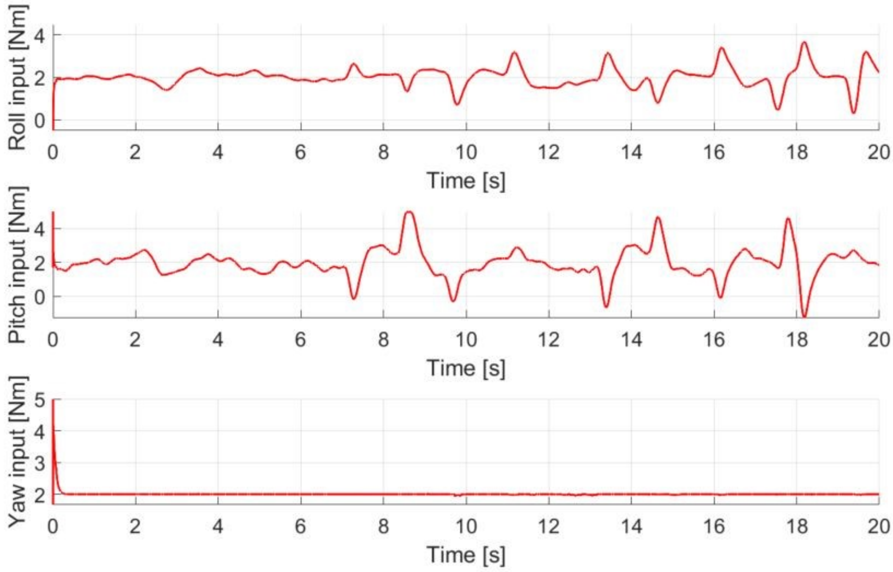


Figure 14. Control efforts with real time data

7. Conclusion

We have proposed an effective control technique called the ILSMC for the tracking control problem of quadcopters 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. Simulation results show in the case of disturbances and uncertainties that, the iterative learning sliding mode controller presents 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 have been performed and validation with real-time experimental data has been 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. Ha. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This work is 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.

References

1. S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, 2012.
2. G. Grisetti, C. Stachniss, and W. Burgard. "Non-linear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, 10:428–439, ISSN: 1524-9050, 2009.
3. M. Flint, M. Polycarpou and E. Fernandez-Gaucherand, "Cooperative control for multiple autonomous UAV's searching for targets," *Proceedings of the* 41st *IEEE Conference on Decision and Control*, Nevada USA, December 2002.
4. A. Brown and D. Anderson, "Trajectory optimization for high-altitude long endurance UAV maritime radar surveillance," *IEEE Transactions on Aerospace and Electronic Systems*, Volume 56, Issue 3, pp. 2406 - 2421, June 2020.

5. N. Metni and T. Hamel, "A UAV for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, Volume 17, Issue 1, Pages 3-10, November 2007.
6. S. R. Herwitz, L. F. Johnson, S. E. Dunagan, R. G. Higgins, D. V. Sullivan, J. Zheng, B. M. Lobitz, J. G. Leung, B. A. Gallmeyer, M. Aoyagi, R. E. Slye, and J. A. Brass, "Imaging from an unmanned aerial vehicle: Agricultural surveillance and decision support," *Computers and Electronics in Agriculture*, vol. 44, no. 1, pp. 49-61, Jul. 2004.
7. Y. C. Choi and H. S. Ahn, "Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests," *IEEE/ASME Trans. Mechatron.*, Vol. 20, pp. 1179–1192, 2015.
8. J. Park, Y. Kim and S. Kim, "Landing site searching and selection algorithm development using vision system and its application to quadrotor," *IEEE Trans. Control Syst. Technol.*, Vol. 23, pp. 488–503, 2015.
9. M. Prasenjit and W. Steven, "Direct adaptive feedback linearization for quadrotor control," *AIAA Guid. Navig. Control Conf.*, American Institute of Aeronautics and Astronautics, Minneapolis, Minnesota, 2012.
10. A. L'afflitto, R.B. Anderson, and K. Mohammadi, "An introduction to nonlinear robust control for unmanned quadrotor aircraft," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 102-21, June. 2018.
11. W. Qingtong, W. Honglin, W. Qingxian, and C. Mou, "Backstepping-based attitude control for a quadrotor UAV using nonlinear disturbance observer," in *Proc. 34th Chin. Control Conf.*, pp. 771–776, 2015.
12. S. H. Chincholkar, W. Jiang, and C.-Y. Chan, "Continuous nonsingular terminal sliding mode control of DC–DC boost converters subject to time-varying disturbances", *IEEE Trans. Circuits Syst. II, Exp. Briefs,* vol. 67, no. 1, pp. 92–96, Jan. 2020.
13. H. Ma , Z. Xiong, "Sliding mode control for uncertain discrete-time systems using an adaptive reaching law", *IEEE Trans. Circuits Syst. II, Exp. Briefs,* VOL. 68, No. 2, Feb. 2021.
14. S. U. Ali, R. Samar, M. Z. Shah, A. I. Bhatti, K. Munawar, U. M. Al-Sggaf, "Lateral guidance and control of UAVs using second-order sliding modes," *Aerospace Science and Technology*, 49, 88-100, 2016.
15. V. Sankaranarayanan and A. D. Mahindrakar, "Control of a class of underactuated mechanical systems using sliding modes," *IEEE Transactions on Robotics*, Vol. 25, No. 2, pp. 459-467, 2009.
16. B. Tian, L. Yin, and H. Wang, "Finite-time reentry attitude control based on adaptive multivariable disturbance compensation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5889–5898, Sep. 2015.
17. M. Zhou, Y. Feng, C. Xue and F. Han, "Deep convolutional neural network based fractional-order terminal sliding-mode control for robotic manipulators," *Neurocomputing*, 416, 143-151, 2020.
18. Y. L. Wang, H. Jahanshahi, S. Bekiros, F. Bezzina, Y. M. Chu, and A. A. Aly, "Deep recurrent neural networks with finite-time terminal sliding mode control for a chaotic fractional-order financial system with market confidence," *Chaos, Solitons & Fractals*, 146, 110881, 2021.
19. D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, pp. 96-114, Jun. 2006.
20. M. Yu and C. Li, "Robust Adaptive Iterative Learning Control for Discrete-Time Nonlinear Systems With Time-Iteration-Varying Parameters," *IEEE Trans. Systems, Man, Cybernetics: Systems*, Vol. 47, No. 7, pp. 1737-1745, 2017.
21. M. Uchiyama, "Formation of high speed motion pattern of mechanical arm by trial", *Transactions of Society of Instrumentation and Control engineer*, 14(6), pp.706-712, 1978.
22. S. Arimoto, S. Kawamura, and F.Miyazaki, "Iterative learning control for robot systems", *Proc. IECON, Tokyo 1984*, pp.393-398, 1984.
23. K.L. Moore, "Iterative learning control for deterministic systems", *Advances in Industial Control* , London, Springer Verlag, 1993.
24. A. Tayebi, S. Abdul, M. B. Zaremba and Y. Ye, "Robust iterative learning control design: Application to a robot manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 5, pp. 608-613, Oct. 2008.
25. M. Mezghani, G. Roux, M. Cabassud, M. V. Le Lann, B. Dahhou and G. Casamatta, "Application of iterative learning control to an exothermic semibatch chemical reactor," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 822-834, Nov. 2002.
26. Q. Zhu, F. Song, J. Xu and Y. Liu, "An internal model based iterative learning control for wafer scanner systems," in *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2073-2084, Oct. 2019.
27. N. Nikooienejad, M. Maroufi, and R. Moheimani, "Iterative learning control for video-rate atomic force microscopy," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2127 - 2138, Aug. 2021.
28. T. Madani, and A. Benallegue, "Adaptive control via backstepping technique and neural networks of a quadrotor helicopter," *IFAC Proc. Volumes*, Vol. 41, pp. 6513–6518, 2008.
29. V. T. Hoang, M. D. Phung and Q. P. Ha, "Adaptive twisting sliding mode control for quadrotor unmanned aerial vehicles," *Proc. 2017 Asian Control Conference*, ASCC 2017, Gold Coast, Australia, December 17-20, pp. 671-676, 2017.
30. B. Tian, H. Lu, Z. Zuo, and Q. Zong, "Adaptive finite-time attitude tracking of quadrotors with experiments and comparisons," *IEEE Transactions on Industrial Electronics* Vol. 66, No. 12, pp. 9428-9438, 2019.
31. X. He, D. Guo and K. K. Leang, "Repetitive control design and implementation for periodic motion tracking in aerial robots," *2017 American Control Conference (ACC)*, pp. 5101-5108, 2017.
32. J. Dong and B. He, "Novel fuzzy PID-type iterative learning control for quadrotor UAV", *Sensors*, 19(1), 24, 2019.
33. R. Adlakha and M. Zheng, "An Optimization-Based Iterative Learning Control Design Method for UAV's Trajectory Tracking", *2020 American Control Conference (ACC)* (pp. 1353-1359). IEEE, July 2020.

34. Y. Chen and K. L. Moore, "Harnessing the nonrepetitiveness in iterative learning control," *Proceedings of the 41st IEEE Conference on Decision and Control,* Vol. 3, pp. 3350-3355. IEEE, Dec 2002.
35. S. Z. Sarpturk, Y. Istefanopulos, and O. Kaynak, "On the stability of discrete-time sliding mode control systems", *IEEE Trans. Autom. Control* No.32, pp. 930–932, 1987.
36. M. Norrlöf and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control", *International Journal of Control*, No. 75. No. 14, pp. 1114-1126, 2002.
37. V. R. Guilherme, G. O. Manuel, and R. R. Francisco, "An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, pp. 29-39, 2010.
38. J. J. Craig, "Introduction to robotics - Mechanics and control (2nd edn)," *USA: Addison-Wesley Publishing Company, Inc*, 1989.
39. V.T. Hoang, M.D. Phung, T.H. Dinh and Q.P. Ha, "System architecture for real-time surface inspection using multiple UAVs," *IEEE Sensors Journal,* Vol. 40, No. 8, pp. 4430 - 4441, 2020.