An Introduction to Nonlinear Robust Control for Unmanned Quadrotor Aircraft

How to Design Control Algorithms for Quadrotors Using Sliding Mode Control and Adaptive Control Techniques

ANDREA L'AFFLITTO, ROBERT B. ANDERSON, and KEYVAN MOHAMMADI

Quadrotor aircraft are drawing considerable attention for their high mobility and capacity to perform tasks with complete autonomy, while minimizing the costs and risks involved with the direct intervention of human operators [1]–[4]. Moreover, several limitations characterizing these rotary-wing unmanned aerial systems (UASs), such as their underactuation, make quadrotors ideal testbeds for innovative theoretical approaches to the problem of controlling mechanical systems.

Designing autopilots for autonomous quadrotors is a challenging task, which involves multiple interconnected aspects. Numerous researchers are currently addressing the problem of designing autonomous guidance systems [5]–[9], navigation systems [10]–[12], and control systems for quadrotors. The primary goal of this article is to present an analysis and synthesis of several nonlinear robust control systems for quadrotors, as discussed in "Summary." First, the article presents and analyzes the equations of motion of quadrotors under three sets of progressively restrictive modeling assumptions: 1) the vehicle's inertial properties (such as the mass and matrix of inertia) vary in time, 2) the quadrotor's main frame is a rigid body and the propellers are thin spinning discs, and 3) the pitch and roll angles are small. This analysis is instrumental to both explain the assumptions underlying the dynamical models used in quadrotors' control systems design and understand the architecture of autopilots for quadrotors. Next, we explain how sliding mode control, model reference adaptive control (MRAC), and adaptive sliding mode control design techniques can be applied to create autopilots for quadrotors. These techniques were chosen over others for their popularity and relative ease of implementation. Finally, we show a qualitative and quantitative approach to the problem of selecting a nonlinear robust control law for quadrotors. As part of this selection process, the performances of sliding mode control, MRAC, and adaptive sliding mode control laws are evaluated, compared, and contrasted through numerical simulations. The most promising of these control laws for precision, robustness, and computational time is successively implemented and tested on a quadrotor.

This article targets both control practitioners and engineering students who have been exposed to graduate-level courses in the analysis and control of nonlinear dynamical systems. Both sliding mode control and model reference adaptive control are illustrated by explicitly referencing graduate-level textbooks such as [13] and [14] so that the problem of designing autopilots for quadrotors results in a direct application of the theoretical notions usually presented in class. The adaptive sliding mode control technique is presented to illustrate how a significant limitation

Summary

In the near future, quadrotor vehicles will be involved in many complex tasks, such as transporting injured people from secluded locations or manipulating objects in industrial premises. These mission scenarios require exploiting the quadrotors' ability to operate in the proximity of people and obstacles, be robust to adverse weather and malfunctions of the propulsion system, and guarantee the safety of their payloads. These strict requirements can be met by employing nonlinear robust control algorithms to design autopilots for quadrotors. This article is aimed both at control practitioners and engineering students who have been exposed to graduate-level courses in the analysis and control of nonlinear dynamical systems. It presents, in a tutorial manner, how sliding mode control, model reference adaptive control, and adaptive sliding mode control can be applied to design autopilots for quadrotors that are robust to model uncertainties, external disturbances, sensor noises, and faults of the control system. Numerical examples and experimental results are used to illustrate both the applicability of the theoretical results and the performances of control algorithms presented. The numerical examples have been produced using the open-source virtual simulator Matlab/Simulink *A Simulator for Quadrotors*, created by the authors.

Digital Object Identifier 10.1109/MCS.2018.2810559
Date of publication: 18 May 2018

of sliding mode control, namely chattering [13, pp. 554–555], can be mitigated by merging two popular nonlinear control design techniques.

Several analytical examples are provided to better explain how the control design techniques presented here can be used to design control systems for quadrotors. Moreover, we cite recent publications that present how sliding mode control, MRAC, and adaptive sliding mode control have been utilized to synthesize autopilots for quadrotors.

The performance of each nonlinear control technique considered here is illustrated using numerical examples produced using *A Simulator for Quadrotors*, a freeware virtual simulator created by the authors [15]. Since this simulator is freely available and open source, readers are provided with working examples of how the theoretical notions presented in this article can be applied to create control systems for quadrotors. This simulator will also allow researchers to compare the performance of their original control techniques against those coded in the toolbox presented in [15]. The results produced by this simulator have been validated using jMAVSim [16]. To challenge the robustness of the sliding mode control, MRAC, and adaptive sliding mode control, the aircraft's inertial properties are considered as unknown in the numerical examples. Moreover, the noisy signal of sensors is mimicked by injecting Gaussian noise into the quadrotor's state vector that is fed back to the controller. Furthermore, the efficiency of one of the four propellers is reduced during the course of the mission. A quadrotor's motors may not be able to deliver the requested thrust forces because of issues such as wear and tear of the motors, low battery levels, or faulty inputs. Lastly, to simulate some fault in the quadrotor's gyroscopes, the measured roll angle is set to zero for a few seconds.

An analysis of the numerical simulations highlights several advantages and disadvantages of sliding mode control, MRAC, and adaptive sliding mode control. For instance, a sliding mode control law is easy to implement, but it is difficult to tune and affected by chattering, that is, high-frequency oscillations. An MRAC law guarantees good trajectory tracking but may be particularly aggressive and, hence, would strain the quadrotor's control system. Finally, an adaptive sliding mode control law provides good trajectory tracking performance and is not affected by chattering, but it is difficult to implement and tune. Based on these considerations and the quality of the numerical results presented, the adaptive sliding mode control has been selected, implemented on an actual quadrotor, and tested through indoor flight experiments.

To perform these flight experiments, the adaptive sliding mode control law was coded in the C++ programming language and executed on an Odroid X4 microcomputer installed on a quadrotor; a Pixhawk flight controller receives the control input from the microcomputer and coordinates the four motors. In the flight experiments, estimates on the UAS's position, velocity, and orientation are affected by noise and delays. The UAS's translational position and velocity are measured by a motion capture system and transmitted over a Wi-Fi signal to the vehicle, and the UAS's orientation and angular velocity are measured by the Pixhawk flight controller. To verify the robustness of the adaptive sliding mode control, the vehicle's mass and inertial properties are considered as unknown, and a payload (whose mass is 27% of the overall vehicle's mass) is deployed, while the quadrotor follows a given reference trajectory. Moreover, the efficiency of one of the motors is reduced to 50%.

The results in this article will allow the reader to appreciate additional control techniques for quadrotors, such as fault tolerant control, which merges the characteristics of fault detection and isolation theory, robust control, and reconfigurable control [17], [18].

NOTATION

Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}^n$ the set of $n\times 1$ real column vectors, and $\mathbb{R}^{n\times m}$ the set of $n\times m$ real matrices. We denote 1_n for the $n\times n$ identity matrix and $0_{n\times m}$ for the zero $n\times m$ matrix. Given $a,b\in\mathbb{R}^3$, with $a=[a_1\ \ a_2\ \ a_3]^{\mathrm{T}}$, the cross product of a and b is denoted by $a^{\times}b$, where

$$a^{\times}\triangleq\begin{bmatrix}0 & -a_3 & a_2\\ a_3 & 0 & -a_1\\ -a_2 & a_1 & 0\end{bmatrix}.$$

Let $\mathcal{I}=\{O;X,Y,Z\}$ denote an inertial, orthonormal reference frame centered at some point O, and let $\mathcal{J}=\{A;x(t),y(t),z(t)\}$, $t\geq t_0$, denote an orthonormal reference frame fixed with the quadrotor and centered at some point A, which is conveniently chosen. It is worthwhile to recall that the axes of orthonormal reference frames have unit length and are orthogonal to one another. The axes of the reference frames $\mathcal{I}$ and $\mathcal{J}$ form two orthonormal bases of $\mathbb{R}^3$, and, if a vector $a\in\mathbb{R}^3$ is expressed in $\mathcal{I}$, then this vector is denoted by $a^{\mathcal{I}}$; if $a\in\mathbb{R}^3$ is expressed in $\mathcal{J}$, then no superscript is used.

It is common practice in aeronautical engineering to set the Z-axis of the reference frame $\mathcal{I}$ so that the weight of an aircraft of mass m is given by $F_{\mathrm{g}}^{\mathcal{I}}=mgZ$, where g denotes the gravitational acceleration; see Figure 1. The matrix of inertia of the quadrotor (excluding its propellers) with respect to A is denoted by $I\in\mathbb{R}^{3\times 3}$, and the matrix of inertia of each propeller with respect to A is denoted by $I_{\mathrm{P}}\in\mathbb{R}^{3\times 3}$.

The position of the point A relative to the origin O of the inertial reference frame $\mathcal{I}$ is denoted by $r_A:[t_0,\infty)\to\mathbb{R}^3$, and the velocity of A with respect to $\mathcal{I}$ is denoted by $v_A:[t_0,\infty)\to\mathbb{R}^3$. The position of the quadrotor's center of mass C with respect to the reference point A is denoted by $r_C:[t_0,\infty)\to\mathbb{R}^3$. Numerous researchers choose the vehicle's center of mass as the reference point (that is, $A\equiv C$) and, hence, set $r_C(t)=0$, $t\geq t_0$. However, the center of mass of a mechanical system may not be easily identifiable. Furthermore, the addition of payloads (especially those connected

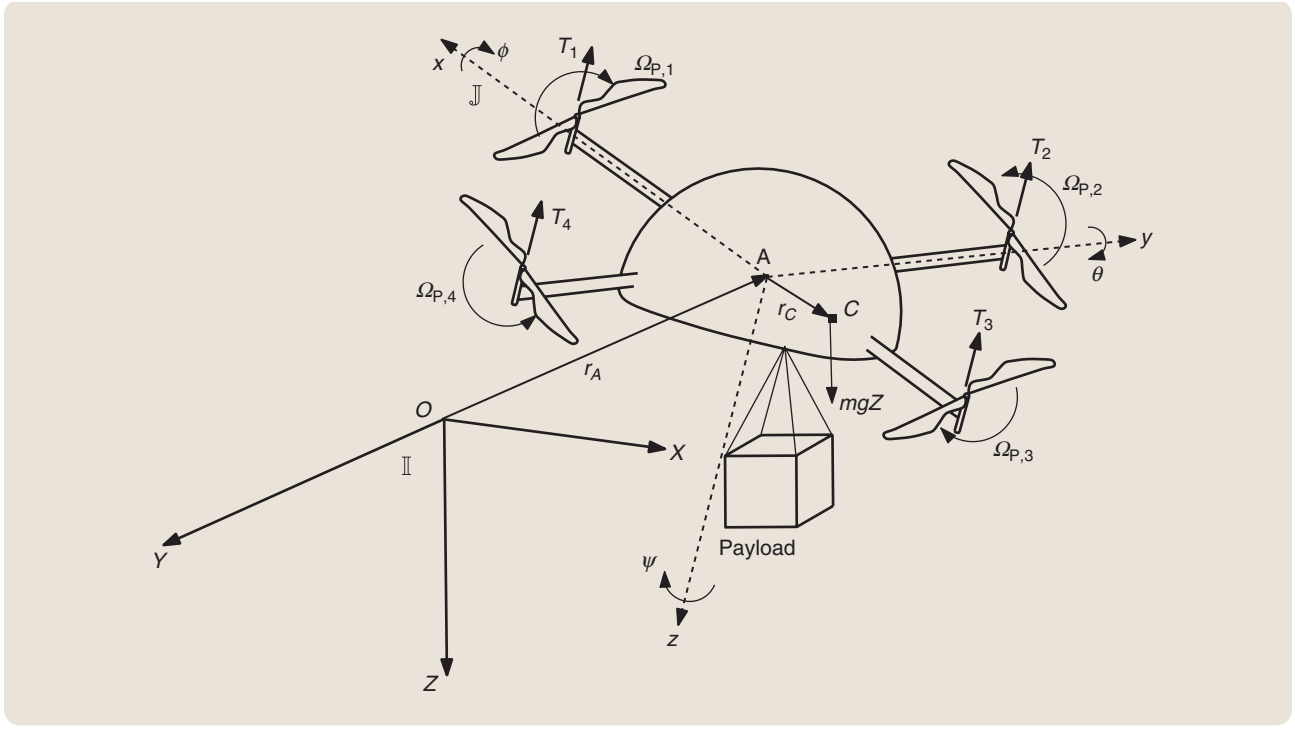

FIGURE 1 A schematic representation of a quadrotor aircraft. The reference point A does not necessarily coincide with the quadrotor's center of mass, whose location may be unknown or vary due to oscillations of the payload. According to the aeronautical convention, the Z-axis of the inertial reference frame $\mathbb{I}$ is aligned with the gravitational force.

to the quadrotor by flexible structures such as ropes [19]) may significantly alter the position of the vehicle's center of mass and introduce uncertainties in the vehicle's dynamical model over time.

The attitude of the reference frame $\mathbb{J}$ with respect to the reference frame $\mathbb{I}$ is captured by the roll, pitch, and yaw angles using a 3-2-1 rotation sequence [20, Ch. 1]. In particular, we denote by $\phi, \psi : [t_0, \infty) \to [0, 2\pi)$ the roll and yaw angles, respectively, and $\theta : [t_0, \infty) \to (-(\pi/2), (\pi/2))$ the pitch angle. It is worthwhile to recall alternative methods to capture the attitude of quadrotors that are based on Euler parameters [21], [22], classical and modified Rodrigues parameters [23], [24], and the SO(3) representation [25], [26]; for details on these attitude representation techniques, see [27]. The angular velocity of $\mathbb{J}$ with respect to $\mathbb{I}$ is denoted by $\omega : [t_0, \infty) \to \mathbb{R}^3$. Lastly, the spin rate of the ith propeller is denoted by $\Omega_{P,i} : [t_0, \infty) \to \mathbb{R}$, $i = 1, \ldots, 4$. Figure 1 illustrates some of the variables introduced to capture the motion of a quadrotor.

EQUATIONS OF MOTION OF A QUADROTOR

In this section, we present the equations of motion of a quadrotor and analyze some of their properties. It is common practice in the control literature to model a quadrotor's dynamics using simplified equations of motion. For instance, in several publications the inertial countertorque is neglected, the vehicle's inertial properties are considered as constant, and the roll and pitch angles are usually considered to be small; the validity of these assumptions is discussed in this section. Although these assumptions hold in most cases of practical interest, it is worthwhile to analyze a quadrotor's equations of motion in detail. Specifically, this analysis is instrumental to parameterize the vehicle's nonlinear dynamics and design more effective adaptive controllers, estimate the largest upper bound on the closed-loop system's uncertainty and design more effective sliding mode control laws [28], and address those less common cases, wherein the quadrotor's inertial properties vary with time [29]. Moreover, this analysis is fundamental to understand the architecture of autopilots for quadrotors.

The translational kinematic equation of a UAS is given by [20, Ex. 1.12]

$$\dot{r}_A^{\mathbb{I}}(t) = R(\phi(t), \theta(t), \psi(t)) v_A(t), \quad r_A^{\mathbb{I}}(t_0) = r_{A,0}^{\mathbb{I}}, \; t \geq t_0, \tag{1}$$

where

$$R(\phi, \theta, \psi) \triangleq \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix},$$
$$(\phi, \theta, \psi) \in [0, 2\pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \times [0, 2\pi), \tag{2}$$

and the rotational kinematic equation of a UAS is given by [20, Th. 1.7]

$$\begin{bmatrix}\dot\phi(t)\\ \dot\theta(t)\\ \dot\psi(t)\end{bmatrix}=\Gamma(\phi(t),\,\theta(t))\,\omega(t),\qquad \begin{bmatrix}\phi(t_0)\\ \theta(t_0)\\ \psi(t_0)\end{bmatrix}=\begin{bmatrix}\phi_0\\ \theta_0\\ \psi_0\end{bmatrix},\tag{3}$$

where

$$\Gamma(\phi,\theta)\triangleq\begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix},$$
$$(\phi,\theta)\in[0,2\pi)\times\left(-\frac{\pi}{2},\frac{\pi}{2}\right).$$

Recall that $\Gamma(\phi,\theta)$ is invertible, since $\theta\in(-\pi/2,\pi/2)$ [20, pp. 18–19]. For additional discussions on a quadrotor's rotational kinematics, see "Facts on Angular Position and Velocity."

Assuming that the quadrotor's mass is constant, its translational dynamic equation is given by [30]

$$\begin{aligned}F_{\mathrm{g}}(\phi(t),\theta(t))-F_{\mathrm{T}}(t)+F(t)&=m[\dot v_A(t)+\omega^{\times}(t)v_A(t)+\ddot r_C(t)\\&\quad+\dot\omega^{\times}(t)r_C(t)+2\omega^{\times}(t)\dot r_C(t)\\&\quad+\omega^{\times}(t)\omega^{\times}(t)r_C(t)],\\ v_A(t_0)&=v_{A,0},\qquad t\ge t_0,\end{aligned}\tag{4}$$

where $-F_{\mathrm{T}}(t)=[0,\,0,\,u_1(t)]^{\mathrm{T}}$ denotes the thrust force, that is, the force produced by the propellers that allows a quadrotor to hover,

$$F_{\mathrm{g}}(\phi,\theta)=mg[-\sin\theta,\,\cos\theta\sin\phi,\,\cos\theta\cos\phi]^{\mathrm{T}},$$
$$(\phi,\theta)\in[0,2\pi)\times\left(-\frac{\pi}{2},\frac{\pi}{2}\right),$$

denotes the quadrotor's weight, and $F:[t_0,\infty)\to\mathbb{R}^3$ denotes the aerodynamic forces acting on the quadrotor [28], [31]–[34]. The rotational dynamic equation of a quadrotor whose frame is modeled as a rigid body, payload is rigidly attached to the vehicle, and propellers are modeled as thin spinning discs is given by [30]

$$\begin{aligned}&M_{\mathrm{T}}(t)+M_{\mathrm{g}}(r_C(t),\phi(t),\theta(t))+M(t)=mr_C^{\times}(t)[\dot v_A(t)+\omega^{\times}(t)v_A(t)]\\&\quad+I\dot\omega(t)+\omega^{\times}(t)I\omega(t)+I_{\mathrm{P}}\sum_{i=1}^{4}\begin{bmatrix}0\\0\\ \dot\Omega_{\mathrm{P},i}(t)\end{bmatrix}+\omega^{\times}(t)I_{\mathrm{P}}\sum_{i=1}^{4}\begin{bmatrix}0\\0\\ \Omega_{\mathrm{P},i}(t)\end{bmatrix},\\&\omega(t_0)=\omega_0,\qquad t\ge t_0,\end{aligned}\tag{5}$$

where $M_{\mathrm{T}}(t)=[u_2(t),\,u_3(t),\,u_4(t)]^{\mathrm{T}}$ denotes the moment of the forces induced by the propellers, $M_{\mathrm{g}}(r_C,\phi,\theta)\triangleq r_C^{\times}F_{\mathrm{g}}(\phi,\theta),(r_C,\phi,\theta)\in\mathbb{R}^3\times[0,2\pi)\times(-(\pi/2),(\pi/2))$ denotes the moment of the quadrotor's weight with respect to A, and $M:[t_0,\infty)\to\mathbb{R}^3$ denotes the moment of the aerodynamic forces with respect to A. The terms $I_{\mathrm{P}}\sum_{i=1}^{4}[0,0,\dot\Omega_{P_i}(t)]^{\mathrm{T}}$, $t\ge t_0$ and $\omega^{\times}(t)I_{\mathrm{P}}\sum_{i=1}^{4}[0,0,\Omega_{P_i}(t)]^{\mathrm{T}}$ are the inertial counter-torque and the gyroscopic effect, respectively.

A quadrotor's control vector is given by $u(t)=[u_1(t),u_2(t),u_3(t),u_4(t)]^{\mathrm{T}}$, $t\ge t_0$, that is, the third component of the thrust force $F_{\mathrm{T}}(\cdot)$ and the moment of the forces induced by the propellers $M_{\mathrm{T}}(\cdot)$. It is possible to verify that [35, Ch. 2]

$$\begin{bmatrix}u_1(t)\\ u_2(t)\\ u_3(t)\\ u_4(t)\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -l & 0 & l\\ l & 0 & -l & 0\\ -c_{\mathrm{T}} & c_{\mathrm{T}} & -c_{\mathrm{T}} & c_{\mathrm{T}}\end{bmatrix}\begin{bmatrix}T_1(t)\\ T_2(t)\\ T_3(t)\\ T_4(t)\end{bmatrix},\qquad t\ge t_0,\tag{6}$$

where $T_i:[t_0,\infty)\to\mathbb{R}$, $i=1,\ldots,4$ denotes the component of the force produced by the ith propeller along the $z(\cdot)$-axis of the reference frame $\mathbb{J}$, $l>0$ denotes the distance of the propellers from the vehicle's barycenter, and $c_{\mathrm{T}}>0$ denotes the propellers' drag coefficient; the relation between controls and propellers' forces in (6) is bijective if and only if $lc_{\mathrm{T}}\neq 0$.

Since the thrust force generated by the ith propeller is related to the propeller's angular velocity by

$$T_i(t)=k\Omega_{\mathrm{P},i}^2(t),\qquad i=1,\ldots,4,\qquad t\ge t_0,\tag{7}$$

where $k>0$ [35], [36], some authors neglect the inertial countertorque and define $[\Omega_{\mathrm{P},1}^2(\cdot),\,\Omega_{\mathrm{P},2}^2(\cdot),\,\Omega_{\mathrm{P},3}^2(\cdot),\,\Omega_{\mathrm{P},4}^2(\cdot)]^{\mathrm{T}}$ as the quadrotor's control vector [37], [38]. For additional considerations on the modeling of quadrotors, see [32], "Inertial or Body Reference Frame?," "Special Cases," and the references therein.

CONTROL STRATEGIES FOR QUADROTORS

Before designing controls for any mechanical system, it is necessary to verify whether the control inputs can steer the system's configuration at will, that is, whether the mechanical system is fully actuated or underactuated [39, Def. 2.9]. This analysis is instrumental to design effective control

Facts on Angular Position and Velocity

The rotational kinematic equation (3) shows that the time derivative of the angular position is not equal to the angular velocity. Indeed, (3) can be considered as a nonholonomic constraint on the quadrotor's dynamics [40, pp. 140–147]. To design controllers for quadrotors, numerous researchers set $[\dot\phi(t),\dot\theta(t),\dot\psi(t)]^{\mathrm{T}}=\omega(t)$, $t\ge t_0$ [35]. In the first approximation, if it is known a priori that the Euler angles $\phi(\cdot)$ and $\theta(\cdot)$ are sufficiently small, then $\Gamma(\phi,\theta)\approx 1_3$, and, hence, it follows from (3) that this choice is justifiable; note that the yaw angle $\psi(\cdot)$ plays no role in the small angle approximation assumptions. The small angle approximation is quite realistic for numerous applications involving quadrotors.

Inertial or Body Reference Frame?

In aeronautical engineering, it is common practice to express the dynamic equations of an aircraft (4) and (5) in the body reference frame $\mathbb{J}$ [20, Ch. 2]. Several authors addressing the control problem for quadrotors capture the vehicle's translational dynamic equation in the inertial reference frame $\mathbb{I}$ [62], [S1]–[S4]. Specifically, (1) and (4) can be replaced by

$$m\ddot{r}_A^{\mathbb{I}}(t) = R(\phi(t),\theta(t),\psi(t))\begin{bmatrix}0\\0\\u_1(t)\end{bmatrix}+\begin{bmatrix}0\\0\\mg\end{bmatrix}+F^{\mathbb{I}}(t)-m\ddot{r}_C^{\mathbb{I}}(t),$$
$$r_A^{\mathbb{I}}(t_0)=r_{A,0}^{\mathbb{I}},\quad v_A^{\mathbb{I}}(t_0)=v_{A,0}^{\mathbb{I}},\quad t\geq t_0, \tag{S1}$$

where $F^{\mathbb{I}}(t)=R(\phi(t),\theta(t),\psi(t))F(t)$. This formulation is advantageous to directly control the quadrotor's position by regulating the thrust force and the pitch and roll angles. Moreover, (S1) is useful to design feedback controls, should the quadrotor's position in the reference frame $\mathbb{I}$ be measured using, for instance, a GPS receiver or a camera-based motion capture system. Since aerodynamic coefficients are usually measured in the body reference frame [S5, Ch. 5], a drawback of (S1) is that the aerodynamic forces acting on the vehicle must be computed in the body reference frame and then expressed in the inertial frame using the rotation matrix (2).

REFERENCES

[S1] T. Madani and A. Benallegue, "Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles," in *Proc. American Control Conf.*, 2007, pp. 5887–5892.

[S2] Z. Zuo and P. Ru, "Augmented $\mathcal{L}_1$ adaptive tracking control of quad-rotor unmanned aircrafts," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 3090–3101, 2014.

[S3] M. Abdolhosseini, Y. M. Zhang, and C. A. Rabbath, "An efficient model predictive control scheme for an unmanned quadrotor helicopter," *J. Intell. Robot. Syst.*, vol. 70, no. 1, pp. 27–38, 2013.

[S4] L.-C. Lai, C.-C. Yang, and C.-J. Wu, "Time-optimal control of a hovering quad-rotor helicopter," *J. Intell. Robot. Syst.*, vol. 45, no. 2, pp. 115–135, 2006.

[S5] B. Etkin, *Dynamics of Flight: Stability and Control*. New York: Wiley, 1982.

Special Cases

It is rather common to assume that the quadrotor's mass is constant over time; if the vehicle is battery operated and is not deploying any payload, then this assumption is realistic. However, there are applications, such as in precision agriculture, wherein the quadrotor's mission is to spread pesticides, herbicides, and fertilizer the crop [S6]. To account for the variations in the quadrotor's mass, the translational dynamic equation becomes

$$\begin{aligned}\hat{F}_{\mathrm{T}}(t)-F_{\mathrm{T}}(t)+F_{\mathrm{g}}(\phi(t),\theta(t))+F(t)&=m(t)[\dot{v}_A(t)+\omega^{\times}(t)v_A(t)+\ddot{r}_C(t)\\&\quad+\dot{\omega}^{\times}(t)r_C(t)+2\omega^{\times}(t)\dot{r}_C(t)+\omega^{\times}(t)\omega^{\times}(t)r_C(t)]\\v_A(t_0)&=v_{A,0},\quad t\geq t_0,\end{aligned} \tag{S2}$$

where $\hat{F}_{\mathrm{T}}(t)\triangleq\int_{\mathcal{V}}w(t)\delta\dot{m}(t)$ denotes the force generated by deploying some mass from the unmanned aerial system (UAS), $\mathcal{V}\subset\mathbb{R}^3$ denotes a volume containing the quadrotor and its payload but not the mass expelled from the UAS, $\delta m:[t_0,\infty)\to\mathbb{R}$ denotes the mass of an infinitesimally small subset of $\mathcal{V}$, whose position from A is denoted by $r_{mA}:[t_0,\infty)\to\mathcal{V}$, and $w:[t_0,\infty)\to\mathbb{R}^3$ denotes the exhaust velocity of each particle expelled from $\mathcal{V}$. Moreover, the rotational dynamic equation becomes

$$\begin{aligned}&M_{\mathrm{T}}(t)+M_{\mathrm{g}}(r_C(t),\phi(t),\theta(t))+\hat{M}_{\mathrm{T}}(t)+M(t)\\&\quad=m(t)r_C^{\times}(t)[\dot{v}_A(t)+\omega^{\times}(t)v_A(t)]+I(t)\dot{\omega}(t)+\omega^{\times}(t)I(t)\omega(t)\\&\qquad+\dot{I}(t)\omega(t)+\dot{h}(t)+\omega^{\times}(t)h(t)+J(t)\omega(t),\\&\qquad\omega(t_0)=\omega_0,\quad t\geq t_0,\end{aligned} \tag{S3}$$

where $\hat{M}_{\mathrm{T}}(t)\triangleq\int_{\mathcal{V}}r_{mA}^{\times}(t)w(t)\delta\dot{m}(t),t\geq t_0$ denotes the moment of the thrust force, $I(t)\triangleq-\int_{\mathcal{V}}r_{mA}^{\times}(t)r_{mA}^{\times}(t)\delta m(t)$ denotes the inertia matrix of the UAS with respect to the reference point A, $J(t)\triangleq\int_{\mathcal{V}}r_{mA}^{\times}(t)r_{mA}^{\times}(t)\delta\dot{m}(t)$, and $h(t)\triangleq\int_{\mathcal{V}}r_{mA}^{\times}(t)\dot{r}_{mA}(t)\delta m(t)$; $J(\cdot)$ captures the variation in the quadrotor's moment of inertia due to the variation of the vehicle's mass over time, and $h(\cdot)$ captures the contribution to the quadrotor's angular momentum because some components (such as the propellers) are moving. If the quadrotor's frame is modeled as a rigid body of constant mass, the payload is rigidly connected to the vehicle, and the propellers are modeled as thin spinning discs, then $\dot{h}(t),t\geq t_0$ specializes to the inertial counter torque, and $\omega^{\times}(t)h(t)$ specializes to the gyroscopic effect in (5). For details, see [30].

REFERENCE

[S6] L. P. Hafsal, "Precision agriculture with unmanned aerial vehicles for SMC estimations towards a more sustainable agriculture," Master's thesis, Hedmark Univ. Applied Science, Elverum, Norway, 2016.

strategies. To define the underactuated mechanical system, consider the second-order differential equation

$$\ddot{q}(t)=f(t,q(t),\dot{q}(t))+G(q(t))u(t),$$
$$q(t_0)=q_0,\ \dot{q}(t_0)=q_{\mathrm{d},0},\qquad t\geq t_0, \tag{8}$$

where $q(t)\in\mathcal{D}\subset\mathbb{R}^n$ denotes the vector of independent generalized coordinates [40, Ch. 2], $u(t)\in\mathbb{R}^m$, $f:[t_0,\infty)\times\mathcal{D}\times\mathbb{R}^n\to\mathbb{R}^n$, and $G:\mathbb{R}^n\to\mathbb{R}^{n\times m}$, and recall that all mechanical systems can be expressed in the same form as (8); for details, see [41].

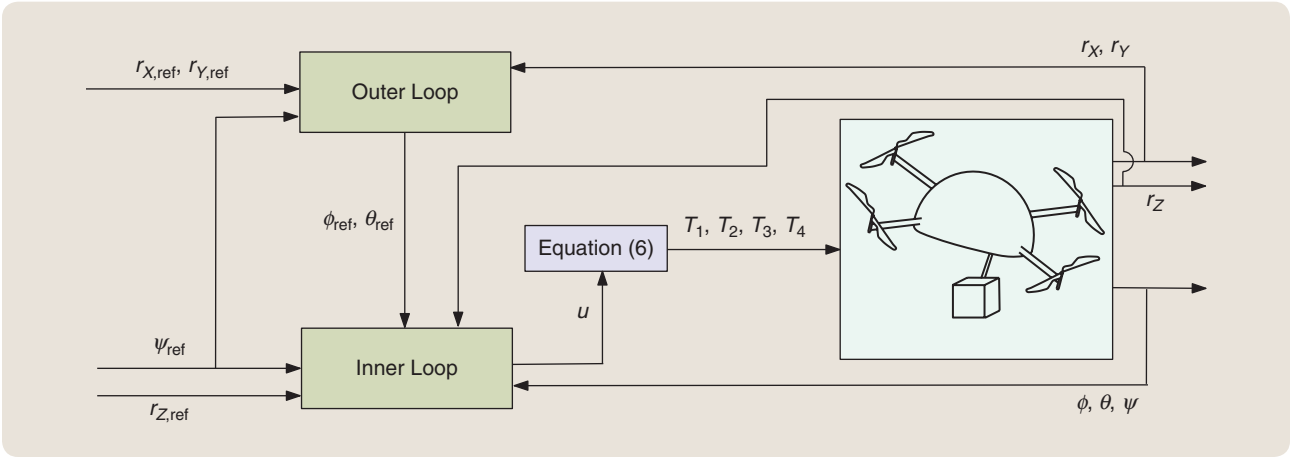

FIGURE 2 A control scheme for a quadrotor. To track both the reference position $[r_{X,\mathrm{ref}}, r_{Y,\mathrm{ref}}, r_{Z,\mathrm{ref}}]^{\mathrm{T}}$ and the reference yaw angle ψ_{ref}, conventional control schemes for quadrotors are divided into two parts, the outer loop and the inner loop. The outer loop determines both the roll angle ϕ_{ref} and the pitch angle θ_{ref} needed to steer the quadrotor's thrust force and track $[r_{X,\mathrm{ref}}, r_{Y,\mathrm{ref}}]^{\mathrm{T}}$. The inner loop determines the thrust forces $T_1, \ldots, T_4$ the propellers must deliver to track $[r_{Z,\mathrm{ref}}, \phi_{\mathrm{ref}}, \theta_{\mathrm{ref}}, \psi_{\mathrm{ref}}]^{\mathrm{T}}$.

Definition [39, Def. 2.9]

Consider the nonlinear dynamical system (8). If $\operatorname{rank}(G(q)) = n$ for all $q \in \mathcal{D}$, then (8) is fully actuated. Alternatively, if $\operatorname{rank}(G(q)) < n$, $q \in \mathcal{D}$, then (8) is underactuated.

A quadrotor's configuration is uniquely described by the vector of independent generalized coordinates

$$q(t) = \left[(r_A^{\mathbb{I}}(t))^{\mathrm{T}}, \phi(t), \theta(t), \psi(t)\right]^{\mathrm{T}}, \qquad t \geq t_0, \tag{9}$$

and, if $r_C(t) \equiv 0, t \geq t_0$, then the equations of motion of a quadrotor, that is, (S1), (3), and (5), can be expressed in the same form as (8) with $n = 6$, $m = 4$, and $\mathcal{D} = \mathbb{R}^3 \times [0, 2\pi) \times (-\pi/2, \pi/2) \times [0, 2\pi)$, and (10), found in the box at the bottom of the page, with $\omega = \Gamma^{-1}(\phi, \theta)[\dot{\phi}, \dot{\theta}, \dot{\psi}]^{\mathrm{T}}$,

$$G(q) = m^{-1} \begin{bmatrix} R(\phi,\theta,\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & 0_{3\times 3} \\ 0_{3\times 1} & m\Gamma(\phi,\theta) I^{-1} \end{bmatrix}, \tag{11}$$

$q_0 = \left[(r_{A,0}^{\mathbb{I}})^{\mathrm{T}}, \phi_0, \theta_0, \psi_0\right]^{\mathrm{T}}$, and $q_{d,0} = \left[(v_{A,0}^{\mathbb{I}})^{\mathrm{T}}, \omega_0^{\mathrm{T}} \Gamma^{\mathrm{T}}(\phi_0, \theta_0)\right]^{\mathrm{T}}$. If $r_C(t) \not\equiv 0, t \geq t_0$, then $G(\cdot)$ is the same as in (11). It follows from [20, Th. 1.5] that $\left\| R(\phi, \theta, \psi)[0, 0, 1]^{\mathrm{T}} \right\| = 1$, and, since $m\Gamma(\phi, \theta) I^{-1}$ is full rank, $(\phi, \theta, \psi) \in [0, 2\pi) \times (-(\pi/2), (\pi/2)) \times [0, 2\pi)$, it holds that $\operatorname{rank}(G(q)) = 4, q \in \mathcal{D}$. Thus, quadrotors are underactuated.

The fact that quadrotors are underactuated can be intuitively shown by noticing that quadrotors are characterized by six independent, generalized coordinates and four control inputs. Hexrotors are rotary-wing UASs similar to quadrotors, which have six parallel propellers. The reader is encouraged to prove whether these vehicles are fully actuated or not.

It follows from (S1) and (5) that the control input $u(\cdot)$ can be designed to regulate the altitude and orientation of a quadrotor. Indeed, it is possible to prove that the dynamical system that captures both the vertical motion and the orientation of a quadrotor is fully actuated. However, in practical applications, it is useful to regulate a quadrotor's position in the inertial space and its yaw angle, since these UASs are generally employed to transport detection devices (such as antennas or cameras) to a specific location and point these devices in a given direction. Thus, the following control strategy is generally used. Assume that the quadrotor's desired position $[r_{X,\mathrm{ref}}, r_{Y,\mathrm{ref}}, r_{Z,\mathrm{ref}}]^{\mathrm{T}} : [t_0, \infty) \to \mathbb{R}^3$ and its desired yaw angle $\psi_{\mathrm{ref}} : [t_0, \infty) \to [0, 2\pi)$ are given. Firstly, compute the roll angle $\phi_{\mathrm{ref}} : [t_0, \infty) \to [0, 2\pi)$ and the pitch angle $\theta_{\mathrm{ref}} : [t_0, \infty) \to (-(\pi/2), (\pi/2))$ needed to tilt the quadrotor and utilize the horizontal component of the thrust force to track $[r_{X,\mathrm{ref}}(t), r_{Y,\mathrm{ref}}(t)]^{\mathrm{T}}, t \geq t_0$; this step is known as outer-loop design. Successively, design a control law for $u(\cdot)$ to track the reference altitude $r_{Z,\mathrm{ref}}(t)$, $t \geq t_0$, and the reference orientation $[\phi_{\mathrm{ref}}(t), \theta_{\mathrm{ref}}(t), \psi_{\mathrm{ref}}(t)]^{\mathrm{T}}$; this step is known as inner-loop design. Figure 2 shows a schematic representation of how inner and outer loops contribute to computing the control inputs for a quadrotor. For examples

$$f(t, q, \dot{q}) = \begin{bmatrix} m^{-1} F^{\mathbb{I}}(t) + g[0, 0, 1]^{\mathrm{T}} \\ \dot{\Gamma}(\phi,\theta)\omega - \Gamma(\phi,\theta) I^{-1} \left(\omega^{\times} I \omega + I_{\mathrm{P}} \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \dot{\Omega}_{\mathrm{P},i}(t) \end{bmatrix} + \omega^{\times} I_{\mathrm{P}} \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \Omega_{\mathrm{P},i}(t) \end{bmatrix} - M(v_A, \omega) \right) \end{bmatrix} \tag{10}$$

Example 1: Outer-Loop Design

Let $r_A^{\mathcal{I}}(t) = [r_X(t), r_Y(t), r_Z(t)]^{\mathrm{T}}, t \geq t_0$. The authors in [S7] and [S8] assume a priori that $\phi(t), t \geq t_0$, and $\theta(t)$ are sufficiently small, set $r_C(t) = 0$, and neglect the aerodynamic forces. In this case, it follows from (S1) that

$$\begin{bmatrix} \ddot{r}_X(t) \\ \ddot{r}_Y(t) \end{bmatrix} = \frac{u_1(t)}{m} \begin{bmatrix} \sin\psi(t) & -\cos\psi(t) \\ \cos\psi(t) & \sin\psi(t) \end{bmatrix} \begin{bmatrix} \phi(t) \\ \theta(t) \end{bmatrix},$$
$$\begin{bmatrix} r_X(t_0) \\ r_Y(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} r_{A,0}^{\mathcal{I}},$$
$$\begin{bmatrix} \dot{r}_X(t_0) \\ \dot{r}_Y(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} v_{A,0}^{\mathcal{I}}, \quad t \geq t_0. \tag{S4}$$

Considering $\phi(\cdot)$ and $\theta(\cdot)$ as virtual control inputs of (S4), it is possible to set

$$\begin{bmatrix} \phi_{\mathrm{ref}}(t) \\ \theta_{\mathrm{ref}}(t) \end{bmatrix} = \frac{m}{u_1(t)} \begin{bmatrix} \sin\psi_{\mathrm{ref}}(t) & \cos\psi_{\mathrm{ref}}(t) \\ -\cos\psi_{\mathrm{ref}}(t) & \sin\psi_{\mathrm{ref}}(t) \end{bmatrix} \cdot \left(K_{\mathrm{p}} \begin{bmatrix} r_{X,\mathrm{ref}}(t) - r_X(t) \\ r_{Y,\mathrm{ref}}(t) - r_Y(t) \end{bmatrix} + K_{\mathrm{d}} \begin{bmatrix} \dot{r}_{X,\mathrm{ref}}(t) - \dot{r}_X(t) \\ \dot{r}_{Y,\mathrm{ref}}(t) - \dot{r}_Y(t) \end{bmatrix} + \begin{bmatrix} \ddot{r}_{X,\mathrm{ref}}(t) \\ \ddot{r}_{Y,\mathrm{ref}}(t) \end{bmatrix} \right), \quad t \geq t_0, \tag{S5}$$

where $K_{\mathrm{p}}, K_{\mathrm{d}} \in \mathbb{R}^{2\times 2}$ are symmetric, positive-definite. Note that (S5) is well defined since $u_1(t) \neq 0, t \geq t_0$ is a necessary condition for a quadrotor to fly.

REFERENCES

[S7] Z. Liu and K. Hedrick, "Dynamic surface control techniques applied to horizontal position control of a quadrotor," in *Proc. Int. Conf. System Theory, Control and Computing*, 2016, pp. 138–144.

[S8] D. Kotarski, Z. Benić, and M. Krznar, "Control design for unmanned aerial vehicles with four rotors," *Interdisciplinary Description Complex Syst.*, vol. 14, no. 2, pp. 236–245, 2016.

Example 2: Outer-Loop Design

The authors in [S9] use the following approach. It follows from (S1) that

$$\begin{bmatrix} \ddot{r}_X(t) \\ \ddot{r}_Y(t) \end{bmatrix} = \frac{u_1(t)}{m} \begin{bmatrix} u_X(t) \\ u_Y(t) \end{bmatrix},$$
$$\begin{bmatrix} r_X(t_0) \\ r_Y(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} r_{A,0}^{\mathcal{I}},$$
$$\begin{bmatrix} \dot{r}_X(t_0) \\ \dot{r}_Y(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} v_{A,0}^{\mathcal{I}}, \quad t \geq t_0, \tag{S6}$$

where $u_1(\cdot)$ is considered a time-varying parameter and

$$u_X(t) \stackrel{\Delta}{=} \cos\phi(t)\sin\theta(t)\cos\psi(t) + \sin\phi(t)\sin\psi(t), \tag{S7}$$
$$u_Y(t) \stackrel{\Delta}{=} \sin\phi(t)\sin\theta(t)\cos\psi(t) - \cos\phi(t)\sin\psi(t) \tag{S8}$$

are considered as virtual control inputs designed so that $|r_X(t) - r_{X,\mathrm{ref}}(t)| \to 0$ and $|r_Y(t) - r_{Y,\mathrm{ref}}(t)| \to 0$ as $t \to \infty$. In this case, $\phi_{\mathrm{ref}}(\cdot)$ and $\theta_{\mathrm{ref}}(\cdot)$ are such that

$$\sin\phi_{\mathrm{ref}}(t) = u_X(t)\sin\psi_{\mathrm{ref}}(t) - u_Y(t)\cos\psi_{\mathrm{ref}}(t), \quad t \geq t_0, \tag{S9}$$
$$\sin\theta_{\mathrm{ref}}(t)\cos\phi_{\mathrm{ref}}(t) = u_X(t)\cos\psi_{\mathrm{ref}}(t) + u_Y(t)\sin\psi_{\mathrm{ref}}(t). \tag{S10}$$

The virtual control inputs $u_X(\cdot)$ and $u_Y(\cdot)$ can be designed using proportional-integral-derivative control, sliding mode control, model reference adaptive control, the adaptive sliding mode control, or any other design technique. A similar approach is presented in [S10].

REFERENCES

[S9] S. Islam, X. P. Liu, and A. E. Saddik, "Adaptive sliding mode control of unmanned four rotor flying vehicle," *Int. J. Robot. Autom.*, vol. 30, no. 2, pp. 140–148, 2015.

[S10] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2891–2902, 2015.

of outer-loop design strategies, see "Example 1: Outer-Loop Design" and "Example 2: Outer-Loop Design."

Remark

If the aerodynamic forces and moments are negligible, $r_C(t) \equiv 0$, $t \geq t_0$, and

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix},$$

then it follows from (S1), (3), and (5) that the linearized equations of motion of a quadrotor are given by

$$\dot{x}(t) = Ax(t) + B\tilde{u}(t), \quad x(t_0) = 0, \quad t \geq t_0, \tag{12}$$

where $x = [q^{\mathrm{T}}, \dot{q}^{\mathrm{T}}]^{\mathrm{T}}$, q is given by (9), $\tilde{u} = [u_1 - mg, u_2, u_3, u_4]^{\mathrm{T}}$, $A \in \mathbb{R}^{12\times 12}$, $A_{1,7} = A_{2,8} = A_{3,9} = A_{4,10} = A_{5,11} = A_{6,12} = 1$, $A_{7,5} = -g$, $A_{8,4} = g$, $A_{i,j}$, $i, j = 1, \ldots, 12$ denotes the element on the ith row and jth column of A (every other element of A is equal to zero), $B \in \mathbb{R}^{12\times 4}$, $B_{9,1} = m^{-1}$, $B_{10,2} = lI_x^{-1}$, $B_{11,3} = lI_y^{-1}$, $B_{12,4} = I_z^{-1}$, and every other element of B is equal to zero. It follows from (12) that, in first approximation, the inner-loop dynamics (the dynamics of $z(\cdot), \phi(\cdot), \theta(\cdot)$, and $\psi(\cdot)$) is regulated by four decoupled double integrators. Moreover, the outer-loop dynamics (the dynamics of $x(\cdot)$ and $y(\cdot)$) are regulated by two decoupled quadruple integrators [42]. Thus, the inner-loop dynamics respond faster than the outer-loop dynamics [43, Ch. 5].

ROBUST NONLINEAR CONTROL TECHNIQUES FOR QUADROTORS

In this section, we explain how three popular nonlinear robust control techniques, namely sliding mode control, MRAC, and adaptive sliding mode control, can be applied to design autopilots for quadrotors. Moreover, we illustrate the features of these control techniques using numerical examples involving the quadrotor DJI F450. These numerical examples have been performed using the open-source, publicly available *A Simulator for Quadrotors*, which was created by the authors [15].

Sliding Mode Control

The sliding mode control architecture consists of steering in finite time the system's trajectory to a subspace of the state space (known as the sliding manifold), which is designed such that if the system's trajectory reaches this manifold, then the system's state asymptotically converges to zero [13, Ch. 14]. This approach is popular for its simplicity of implementation and robustness to uncertainties [44]. We discuss below how the sliding mode control can be employed to design an autopilot that steers a quadrotor to a given location with a given yaw angle.

Consider the nonlinear, time-varying plant [13, pp. 564]

$$\dot{x}_1(t) = f_1(t, x_1(t), x_2(t)), \quad x_1(t_0) = x_{10}, \quad t \geq t_0, \tag{13}$$

$$\dot{x}_2(t) = f_2(t, x_1(t), x_2(t)) + G(x_1(t), x_2(t))E(t, x_1(t), x_2(t))u(t) + \delta(t, x_1(t), x_2(t), u(t)), \quad x_2(t_0) = x_{20}. \tag{14}$$

where $x_1(t) \in \mathcal{D} \subseteq \mathbb{R}^{n-m}$, $t \geq t_0$, $0_{(n-m)\times 1} \in \mathcal{D}$, $x_2(t) \in \mathbb{R}^m$, $u(t) \in U \subseteq \mathbb{R}^m$, $f_1 : [t_0, \infty) \times \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}^{n-m}$, $f_2 : [t_0, \infty) \times \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}^m$, $G : \mathcal{D} \times \mathbb{R}^m \to R^{m\times m}$, and $E : [t_0, \infty) \times \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}^{m\times m}$ are continuous in their arguments, $E(t, x_1, x_2)$ is invertible for all $(t, x_1, x_2) \in [t_0, \infty) \times \mathcal{D} \times \mathbb{R}^m$, $G(\cdot, \cdot)$ is an unknown positive-definite diagonal matrix, that is, $G(x_1, x_2) \geq g_0 I_m$ for some $g_0 > 0$, and $\delta : [t_0, \infty) \times \mathcal{D} \times \mathbb{R}^m \times U \to \mathbb{R}^m$ is continuous and unknown. Both $\delta(\cdot, \cdot, \cdot, \cdot)$ and $G(\cdot, \cdot)$ account for model uncertainties and faults of the control system. Note that the equations of motion of many mechanical systems can be cast in the same form as (13) and (14). Specifically, (13) captures the kinematic equations, and (14) captures the dynamic equations. The kinematic equations of a mechanical system are usually considered to be perfectly known, whereas the dynamic equations are usually affected by modeling errors, uncertainties on the system's inertial properties, and disturbances.

To design a sliding manifold, the following assumption is needed.

Assumption [13, p. 564]

Consider the nonlinear time-varying dynamical system (13). There exist $V : [t_0, \infty) \times \mathbb{R}^m \to \mathbb{R}$, which is continuously differentiable, and $W_1, W_2, W : \mathbb{R}^m \to \mathbb{R}$, which is continuous and positive-definite, and a control law $\alpha : \mathbb{R}^{n-m} \to \mathbb{R}^m$ so that

$$W_1(x_1) \leq V(t, x_1) \leq W_2(x_1), \quad (t, x_1) \in [t_0, \infty) \times \mathcal{D}, \tag{15}$$

$$\frac{\partial V(t, x_1)}{\partial t} + \frac{\partial V(t, x_1)}{\partial x_1} f_1(t, x_1, \alpha(x_1)) < -W(x_1). \tag{16}$$

Considering $x_2(\cdot)$ as an input for (13), it follows from [13, Th. 4.9] that this assumption postulates the existence of a control law $\alpha(\cdot)$, such that (13) with $x_2 = \alpha(x_1)$, $x_1 \in \mathcal{D}$ is uniformly asymptotically stable [13, Def. 4.4]. The sliding mode control framework steers $x_2(\cdot)$ toward $\alpha(x_1(\cdot))$ in finite time. The subspace $S_\alpha \triangleq \{(x_1, x_2) \in \mathcal{D} \times \mathbb{R}^m : \sigma(x_1, x_2) = 0\}$, where $\sigma(x_1, x_2) \triangleq x_2 - \alpha(x_1)$ is the sliding manifold. It is common practice to design linear sliding manifolds, that is, to set $\sigma(x_1, x_2) = x_2 + K_{x_1} x_1$, $(x_1, x_2) \in \mathcal{D} \times \mathbb{R}^m$, where $K_{x_1} \in \mathbb{R}^{m\times(n-m)}$.

To design a sliding mode control law for the nonlinear dynamical system given by (13) and (14), let the invertible matrix function $\hat{G} : \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}^{m\times m}$ denote an estimate of $G(\cdot, \cdot)$ and define

$$\psi(t, x_1, x_2, w) \triangleq u_{\text{eq}}(t, x_1, x_2) + E^{-1}(t, x_1, x_2)w, \quad (t, x_1, x_2, w) \in [t_0, \infty) \times \mathbb{R}^{n-m} \times \mathbb{R}^m \times \mathbb{R}^m, \tag{17}$$

$$u_{\text{eq}}(t, x_1, x_2) \triangleq -E^{-1}(t, x_1, x_2)\hat{G}^{-1}(x_1, x_2) \times \left[f_2(t, x_1, x_2) - \frac{\partial \alpha(x_1)}{\partial x_1} f_1(t, x_1, x_2) \right], \tag{18}$$

$$\Delta(t, x_1, x_2, w) \triangleq \delta(t, x_1, x_2, \psi(t, x_1, x_2, w)) + [1_m - G(x_1, x_2)\hat{G}^{-1}(x_1, x_2)] \times \left[f_2(t, x_1, x_2) - \frac{\partial \alpha(x_1)}{\partial x_1} f_1(t, x_1, x_2) \right]. \tag{19}$$

Furthermore, define

$$\gamma_i(x_1, x_2) \triangleq -\operatorname{sign}(\sigma_i(x_1, x_2))\beta_i(x_1, x_2), \quad (x_1, x_2) \in \mathcal{D} \times \mathbb{R}^m, \quad i = 1, \ldots, m, \tag{20}$$

where $\sigma_i(\cdot, \cdot)$ denotes the ith component of $\sigma(x_1, x_2)$, $i = 1, \ldots, m$, $k_i \in [0, 1)$, $\operatorname{sign}(\cdot)$ denotes the signum function [13, p. 19], $\beta_i : \mathcal{D} \times \mathbb{R}^m \to \overline{\mathbb{R}}_+$ is continuous and such that

$$\beta_i(x_1, x_2) \geq c + \frac{\rho_i(x_1, x_2)}{1 - k_i}, \tag{21}$$

$c > 0$, $\rho_i : \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}$ is continuous, and

$$\left| [G^{-1}(x_1, x_2)\Delta(t, x_1, x_2, \psi(t, x_1, x_2, w))]_i \right| \leq \rho_i(x_1, x_2) + k_i |w_i|, \quad (t, x_1, x_2, w) \in [t_0, \infty) \times \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^m. \tag{22}$$

Applying Lyapunov stability theory for nonlinear time-varying dynamical systems [13, pp. 564–566], it is possible to prove that if (15) and (16) are verified and (22) is satisfied, then the solutions $x_1(\cdot)$ and $x_2(\cdot)$ of (13) and (14) with

$$u(t, x_1, x_2) = u_{\text{eq}}(t, x_1, x_2) + E^{-1}(t, x_1, x_2)\gamma(x_1, x_2), \quad (x_1, x_2) \in \mathcal{D} \times \mathbb{R}^m \tag{23}$$

are such that both $x_1(t) \to 0$ and $x_2(t) \to 0$ as $t \to \infty$ uniformly in t_0.

It follows from (20)–(22) that, in the sliding mode architecture, larger uncertainties on the control imply stronger control actions. The control law (20) involves the signum function and, hence, is discontinuous. It is well known that solutions of ordinary differential equations with discontinuous right-hand sides may not exist or be unique [45, Ch. 2], [46]. In most cases of practical interest,

Example 3: Sliding Mode and Outer-Loop Design

Consider the problem of designing a quadrotor's outer loop so that the unmanned aerial system is steered to a given point in the horizontal plane. In this case, without loss of generality, it is possible to set $r_{X,\mathrm{ref}}(t) = r_{Y,\mathrm{ref}}(t) = 0, t \geq t_0$. If the quadrotor's mass $m > 0$ is unknown, then it follows from (S4) that the outer-loop dynamics can be cast in the same form as (13) and (14) with $x_1 = [r_X, r_Y]^{\mathrm{T}}$, $x_2 = [\dot{r}_X, \dot{r}_Y]^{\mathrm{T}}$, $u = [\phi_{\mathrm{ref}}, \theta_{\mathrm{ref}}]^{\mathrm{T}}$, $\mathcal{D} = \mathbb{R} \times \mathbb{R}$, $U = [0, 2\pi) \times (-(\pi/2), (\pi/2))$, $f_1(t, x_1, x_2) = [\dot{r}_X, \dot{r}_Y]^{\mathrm{T}}$, $f_2(t, x_1, x_2) = 0_{2\times 1}$, $B(x) = [0_{2\times 2}, 1_2]^{\mathrm{T}}$, $\delta(t, x_1, x_2, u) = 0_{2\times 1}$, $G(x_1, x_2) = m^{-1} 1_2$, and

$$E(t, x_1, x_2) = u_1(t) \begin{bmatrix} \sin\psi(t) & -\cos\psi(t) \\ \cos\psi(t) & \sin\psi(t) \end{bmatrix},$$

where both $u_1(\cdot)$ and $\psi(\cdot)$ are known time-varying parameters. The sliding manifold can be defined, for instance, by the proportional control law $\alpha(x_1) = -K_{x_1} x_1$, where $K_{x_1} \in \mathbb{R}^{2\times 2}$ is symmetric, positive-definite. In this case, it follows from (17)–(19) that

$$u_{\mathrm{eq}}(t, x_1, x_2) = -\hat{m}^{-1} u_1^{-1}(t) \begin{bmatrix} \sin\psi(t) & \cos\psi(t) \\ -\cos\psi(t) & \sin\psi(t) \end{bmatrix} K_{x_1} x_2,$$
$$\psi(t, x_1, x_2, w) = u_1^{-1}(t) \begin{bmatrix} \sin\psi(t) & \cos\psi(t) \\ -\cos\psi(t) & \sin\psi(t) \end{bmatrix} (w - \hat{m}^{-1} K_{x_1} x_2), \tag{S11}$$

where $\hat{m} > 0$ denotes an estimate of the quadrotor's mass and

$$\Delta(t, x_1, x_2, w) = \hat{m}^{-1}(\hat{m} - m) K_{x_1} x_2.$$

An upper bound on the uncertainties is given by (22) with

$$\rho_i(x_1, x_2) = \frac{\hat{m}}{4} \| K_{x_1} \| \| x_2 \|, \quad i = 1, 2$$

and $k_i = 0$, where $\| K_{x_1} \|$ denotes the Euclidean equi-induced norm of the matrix K_{x_1} [72, Def. 9.4.1].

discontinuous control inputs induce an undesired effect known as chattering, which consists of high-frequency oscillations of the system's state about the sliding manifold [13, Ch. 14], [47]. Chattering is one of the main drawbacks of classical sliding mode control, which can be mitigated by introducing a boundary layer, where the signum function is approximated by the saturation, the sigmoid, or the arctangent functions [13, pp. 556–558], [48], [49]. Alternatively, it is possible to resort to higher-order sliding mode control [47], [50], [51], whereby the sliding mode architecture is applied to higher derivatives of the sliding variable to contrast chattering, chattering attenuation conventional sliding mode control (whereby chattering is reduced by employing a time-varying sliding manifold [52]), or adaptive sliding mode control, discussed in the following. Terminal sliding mode is a variation of sliding mode control and guarantees finite-time convergence to the equilibrium point [53].

Remark

The function $\rho_i(\cdot, \cdot)$ provides an upper bound on the system's uncertainties and directly affects the amplitude of the sliding mode control law. Thus, accurate dynamical models are essential to estimate the system's uncertainties and minimize the control effort.

Sliding mode control and its numerous variations have been applied to design all or some of the four control inputs of a quadrotor. For instance, the classical sliding mode control architecture is applied in [54] to regulate a quadrotor's inner loop, [55] in combination with a backstepping control law that regulates the vehicle's outer loop, and [56] with an adaptive observer to estimate unknown parameters and hence dynamically estimate the upper bounds on the system's uncertainties. Terminal sliding mode is employed in [57] for trajectory tracking. Second-order sliding mode is applied in [58] to design the four control inputs of a quadrotor, [59] in conjunction with the block control technique [55], and [60] to regulate a quadrotor's orientation using nonlinear sliding surfaces. Lastly, higher-order sliding mode control has been applied to design outer loops [61] and estimators for quadrotors [62]. The sliding mode control technique for stochastic dynamical systems is discussed in [63]–[65]. For detailed examples on how to design the inner and outer loops of quadrotors using the sliding mode control architecture, see "Example 3: Sliding Mode and Outer-Loop Design" and "Example 4: Sliding Mode and Inner-Loop Design."

Figure 3 shows the trajectory of a quadrotor, which implements the sliding mode control law discussed in [54]. The aircraft takes off from the origin O of the inertial reference frame $\mathbb{I}$ and follows a cylindrical spiral reference trajectory with forward velocity of 1 m/s and vertical velocity of 0.15 m/s; this simulation has been performed using the toolbox presented in [15]. After 17 s from takeoff, the efficiency of one of the motors is reduced to 75%. Despite this efficiency loss, the quadrotor is still able to follow the reference trajectory. However, this control law fails for lower values of the motor's efficiency.

Every sensor is unavoidably affected by noise, and, to add realism to this numerical simulation, white Gaussian noise is introduced in the quadrotor's state vector being fed back to the inner and outer loops' control laws. The noise's variance is 0.188, which corresponds to an uncertainty of 2° in the Euler's angles capturing the vehicle's orientation. Moreover, the noise's frequency is 200 Hz, which is approximately the output frequency of sensors used for applications involving quadrotor helicopters, such as motion capture systems and gyroscopes. Figure 4 represents the measured roll angle, which is corrupted by noise. Although sliding mode control is not explicitly designed to be robust to stochastic disturbances, it appears from Figure 3 that

Example 4: Sliding Mode and Inner-Loop Design

Consider the problem of designing an autopilot for quadrotors so that the unmanned aerial system is steered to a given point in the reference frame $\mathbb{I}$ with a yaw angle. In this case, the autopilot's outer loop can be designed as in "Example 3: Sliding Mode and Outer-Loop Design," and the inner loop can be designed assuming without loss of generality that $r_{Z,\mathrm{ref}}(t) = 0, t \geq t_0$ and $\phi_{\mathrm{ref}}(t) = \theta_{\mathrm{ref}}(t) = \psi_{\mathrm{ref}}(t) = 0$. If the vehicle's mass m and matrix of inertia I are unknown, the location of the center of mass $r_C(\cdot)$ is constant and unknown, the aerodynamic force and moment of the force are unknown, and both the roll angle $\phi(t), t \geq t_0$ and the pitch angle $\theta(t)$ are small, then it follows from (S1), (3), and (5) that the inner-loop dynamics can be cast in the same form as (13) and (14) with $x_1 = [r_Z, \phi, \theta, \psi]^{\mathrm{T}}$, $x_2 = [\dot{r}_Z, \omega^{\mathrm{T}}]^{\mathrm{T}}$, $u = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$, $\mathcal{D} = \mathbb{R} \times (-\pi/2, \pi/2) \times (-\pi/2, \pi/2) \times [0, 2\pi)$, $U = \mathbb{R}^4$, $f_1(t, x_1, x_2) = [\dot{r}_Z, \omega^{\mathrm{T}}]^{\mathrm{T}}$,

$$f_2(t, x_1, x_2) = \begin{bmatrix} g \\ -I^{-1}\left(\omega^{\times} I\omega + I_{\mathrm{P}} \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \dot{\Omega}_{\mathrm{P},i}(t) \end{bmatrix} + \omega^{\times} I_{\mathrm{P}} \sum_{i=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \Omega_{\mathrm{P},i}(t) \end{bmatrix}\right) \end{bmatrix},$$

$$B(x_1, x_2) = \begin{bmatrix} 0_{4\times 4} \\ 1_4 \end{bmatrix}, \quad G(x_1, x_2) = \begin{bmatrix} m^{-1} & 0_{1\times 3} \\ 0_{3\times 1} & I^{-1} \end{bmatrix},$$

$$\delta(t, x_1, x_2, u) = \begin{bmatrix} [0, 0, m^{-1}] F^{\mathrm{I}}(t) \\ I^{-1}[M(v_{\mathrm{A}}(t), \omega) + r_{\mathrm{C}}^{\times} F_{\mathrm{g}}(\phi, \theta) - m r_{\mathrm{C}}^{\times}(\dot{v}_{\mathrm{A}}(t) + \omega^{\times} v_{\mathrm{A}}(t))] \end{bmatrix},$$

and $E(t, x_1, x_2) = 1_4$. The sliding manifold can be defined, for instance, by the proportional control law $\alpha(x_1) = -K_{x_1} x_1$, where $K_{x_1} \in \mathbb{R}^{4\times 4}$ is symmetric, positive-definite. In this case, it follows from (17)–(19) that

$$u_{\mathrm{eq}}(t, x_1, x_2) = -\begin{bmatrix} \hat{m} & 0_{1\times 3} \\ 0_{3\times 1} & \hat{I} \end{bmatrix} [f_2(t, x_1, x_2) + K_{x_1} x_2],$$

where $\hat{m}$ and $\hat{I}$ denote an estimates of the quadrotor's mass and matrix of inertia, respectively, $\psi(t, x_1, x_2, w) = u_{\mathrm{eq}}(t, x_1, x_2) + w$, and

$$\begin{aligned} \Delta(t, x_1, x_2, w) = {} & \delta(t, x_1, x_2, \psi(t, x_1, x_2, w)) \\ & + \left(1_4 - \begin{bmatrix} m^{-1} & 0_{1\times 3} \\ 0_{3\times 1} & I^{-1} \end{bmatrix} \begin{bmatrix} \hat{m} & 0_{1\times 3} \\ 0_{3\times 1} & \hat{I} \end{bmatrix}\right) \\ & \times [f_2(t, x_1, x_2) + K_{x_1} x_2]. \end{aligned} \tag{S12}$$

An upper bound on the uncertainties is given by (22) with $k_i = 0, i = 1, 2$; the expression for $\rho_i(x_1, x_2)$ is omitted for brevity.

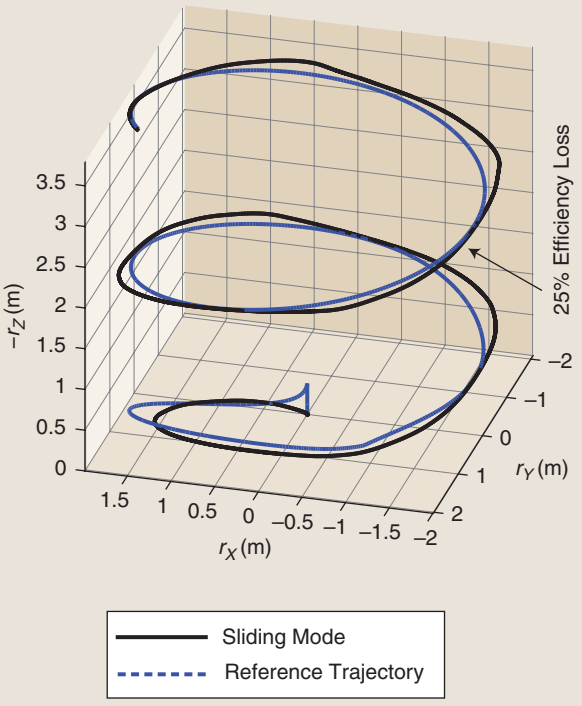

FIGURE 3 The trajectory of a quadrotor implementing a sliding mode control law. The aircraft takes off from the origin and follows a cylindrical spiral trajectory. After 20 s, the efficiency of one propeller is reduced to 75%. Despite this efficiency loss and uncertainties in the aircraft's inertial properties, the quadrotor still continues following the reference trajectory.

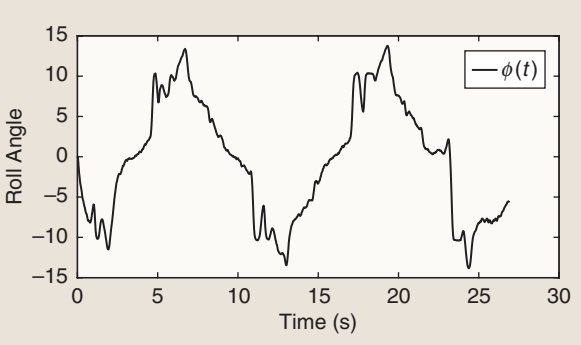

FIGURE 4 The measured roll angle for the quadrotor implementing the sliding mode controller. To increase the realism of this numerical simulation, white Gaussian noise is introduced into the integrated state vector to mimic the noisy behavior of sensors.

the quadrotor implementing the sliding mode control law is able to follow the given reference trajectory with acceptable tracking error, despite noise in the feedback signal and uncertainties on the vehicle's inertial properties.

Figure 5 represents the control inputs for the quadrotor implementing the sliding mode control law and clearly shows the effect of chattering. This discontinuous behavior of the control input would not allow the implementation of sliding mode control laws on actual quadrotors, unless a boundary layer is introduced.

MRAC with Integral Feedback Connections

MRAC is a design technique that allows regulating nonlinear dynamical systems whose dynamics are partly

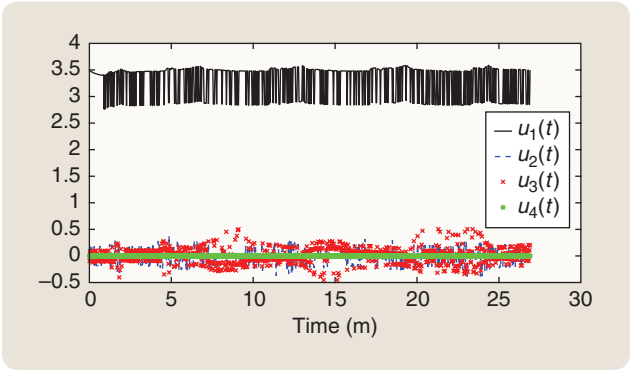

FIGURE 5 Sliding mode control inputs for a quadrotor following a cylindrical spiral trajectory. Clearly, these control signals are affected by high-frequency oscillations. This phenomenon is known as chattering.

known. Specifically, parameterizing the plant's nonlinear dynamics by means of a given regressor vector and a set of unknown coefficients, an MRAC law regulates the closed-loop system's response so that it asymptotically coincides with an ideal dynamical system's response [14]. In its original formulation, MRAC is robust both to parametric and matched uncertainties in the plant's model. Robustness to unmanteched uncertainties can be achieved by employing modified versions of this control technique [66], [67], [14, Ch. 10]. MRAC controls that are explicitly designed to guarantee robustness to random disturbances are discussed in [68]–[70].

Consider the nonlinear plant [14, pp. 293–294]

$$\begin{aligned} \dot{x}_{\mathrm{p}}(t) &= A_{\mathrm{p}}x_{\mathrm{p}}(t) + B_{\mathrm{p}}\Lambda[\tilde{u}(t) + \Theta^{\mathrm{T}}\Phi_{\mathrm{p}}(x_{\mathrm{p}}(t))], \\ x_{\mathrm{p}}(t_0) &= x_0, \quad t \geq t_0, \end{aligned} \tag{24}$$

$$y(t) = C_{\mathrm{p}}x_{\mathrm{p}}(t), \tag{25}$$

where $x_{\mathrm{p}}(t) \in \mathbb{R}^{n_{\mathrm{p}}}$, $t \geq t_0$, $\tilde{u}(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^m$, $A_{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{p}} \times n_{\mathrm{p}}}$ is unknown, $B_{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{p}} \times m}$, $C_{\mathrm{p}} \in \mathbb{R}^{m \times n_{\mathrm{p}}}$, $\Theta \in \mathbb{R}^{N \times m}$ is unknown, the regressor vector $\Phi(x) \in \mathbb{R}^N$, $x_{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{p}}}$, is Lipschitz continuous in x_{p}, and $\Lambda \in \mathbb{R}^{m \times m}$ is diagonal, positive-definite, unknown, and such that the pair $(A_{\mathrm{p}}, B_{\mathrm{p}}\Lambda)$ is controllable. Although the matrix A_{p} is unknown, its structure is usually known, and the controllability of the pair $(A_{\mathrm{p}}, B_{\mathrm{p}}\,\Lambda)$ can be generally verified. The matrix Λ captures malfunctions in the control system, such as the breakage of a propeller, and $\Theta^{\mathrm{T}}\Phi_{\mathrm{p}}(x_{\mathrm{p}})$, $x_{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{p}}}$, parameterizes the plant's nonlinearities.

Given the bounded, continuous reference signal $y_{\mathrm{cmd}} : [t_0, \infty) \to \mathbb{R}^m$, the goal of the MRAC architecture with integral feedback connections is to design the control input $u(\cdot)$ so that the measured output $y(\cdot)$ tracks $y_{\mathrm{cmd}}(\cdot)$ with uniformly ultimately bounded error, that is,

$$\|y(t) - y_{\mathrm{cmd}}(t)\| < b, \quad t \geq t_0 + T, \tag{26}$$

for some $b > 0$ and $T \geq 0$, both independent of t_0. To meet this goal, consider the augmented state vector

$$x(t) \triangleq \left[x_{\mathrm{p}}^{\mathrm{T}}(t), \int_{t_0}^{t} [y(\tau) - y_{\mathrm{cmd}}(\tau)]^{\mathrm{T}} \mathrm{d}\tau\right]^{\mathrm{T}} \in \mathbb{R}^n, \quad t \geq t_0,$$

where $n \triangleq n_{\mathrm{p}} + m$, and note that it follows from (24) and (25) that

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B\Lambda[\tilde{u}(t) + \Theta^{\mathrm{T}}\Phi(x(t))] + B_{\mathrm{ref}}y_{\mathrm{cmd}}(t), \\ x(t_0) &= \begin{bmatrix} x_0 \\ 0_{m\times 1} \end{bmatrix}, \quad t \geq t_0, \end{aligned} \tag{27}$$

$$y(t) = Cx(t), \tag{28}$$

where

$$A = \begin{bmatrix} A_{\mathrm{p}} & 0_{n_{\mathrm{p}}\times m} \\ C_{\mathrm{p}} & 0_{m\times m} \end{bmatrix}, B = \begin{bmatrix} B_{\mathrm{p}} \\ 0_{m\times m} \end{bmatrix}, B_{\mathrm{ref}} = \begin{bmatrix} 0_{n_{\mathrm{p}}\times m} \\ -1_m \end{bmatrix}, \Phi(x) = \Phi_{\mathrm{p}}(x_{\mathrm{p}})$$

and $C = [C_{\mathrm{p}}, 0_{m\times m}]$. Consider also the reference dynamical model

$$\dot{x}_{\mathrm{ref}}(t) = A_{\mathrm{ref}}x_{\mathrm{ref}}(t) + B_{\mathrm{ref}}y_{\mathrm{cmd}}(t), \quad x_{\mathrm{ref}}(t_0) = x(t_0), \quad t \geq t_0, \tag{29}$$

$$y_{\mathrm{ref}}(t) = Cx_{\mathrm{ref}}(t), \tag{30}$$

where $A_{\mathrm{ref}} \in \mathbb{R}^{n\times n}$ is Hurwitz, and define the tracking error $e(t) \triangleq x(t) - x_{\mathrm{ref}}(t)$, $t \geq t_0$.

Applying classical Lyapunov stability theory [14, Ch. 10] or, equivalently, partial-state stability theory [71], it is possible to prove that if there exists $K_x \in \mathbb{R}^{n\times m}$ such that

$$A_{\mathrm{ref}} = A + B\Lambda K_x^{\mathrm{T}}, \tag{31}$$

then the adaptive control law

$$\tilde{u}(x, \Phi(x), \hat{\Pi}) = \hat{\Pi}^{\mathrm{T}}\begin{bmatrix} x \\ \Phi(x) \end{bmatrix}, \quad (x, \hat{\Pi}) \in \mathbb{R}^n \times \mathbb{R}^{(n+N)\times m}, \tag{32}$$

steers the dynamical system (27) so that $\lim_{t\to\infty} e(t) = 0$ and (26) is verified, where

$$\begin{aligned} \dot{\hat{\Pi}}(t) &= \begin{bmatrix} -\Gamma_x & 0 \\ 0 & \Gamma_\Phi \end{bmatrix}\begin{bmatrix} x(t) \\ \Phi(x(t)) \end{bmatrix}e^{\mathrm{T}}(t)PB, \\ \hat{\Pi}(t_0) &= 0_{(n+N)\times m}, \quad t \geq t_0, \end{aligned} \tag{33}$$

$\Gamma_x \in \mathbb{R}^{n\times n}$ is symmetric positive-definite, $\Gamma_\Phi \in \mathbb{R}^{N\times N}$ is symmetric positive-definite, $P \in \mathbb{R}^{n\times n}$ is the symmetric positive-definite solution of the Lyapunov equation

$$0 = A_{\mathrm{ref}}^{\mathrm{T}}P + PA_{\mathrm{ref}} + Q, \tag{34}$$

and $Q \in \mathbb{R}^{n\times n}$ is symmetric positive-definite. In practice, the MRAC law (32) guarantees that the nonlinear plant (24) and (25) eventually mimics the behavior of the reference model (29) and (30) and the measured output eventually tracks the reference signal $y_{\mathrm{ref}}(\cdot)$. Since

$$\mathcal{L}[y_{\mathrm{ref}}(\cdot)] = C(s1_n - A_{\mathrm{ref}})^{-1}B_{\mathrm{ref}}\mathcal{L}[y_{\mathrm{cmd}}(\cdot)], \quad s \in \mathbb{C}\backslash\mathrm{spec}(A_{\mathrm{ref}}), \tag{35}$$

where $\mathcal{L}[\cdot]$ denotes the Laplace transform of its argument [43, Ch. 2] and spec $(\cdot)$ denotes the spectrum of its argument

Example 5: MRAC and Inner-Loop Design

Consider the problem of designing a quadrotor's inner loop, and assume that the vehicle's mass m, matrix of inertia I, and location of the center of mass $r_C(\cdot)$ are constant and unknown. In this case, it follows from (12) that the equations of motion of a quadrotor can be approximated by the nonlinear dynamical model (24) with $x_p = [q^T, \dot{q}^T]^T$, $q(\cdot)$ given by (9), $\tilde{u} \in \mathbb{R}^4$, $B_p = [0_{4\times 8}, 1_4]^T$, $\Lambda \in \mathbb{R}^{4\times 4}$ is a diagonal matrix such that $\Lambda_{1,1} = m^{-1}$, $\Lambda_{2,2} = I_x^{-1}$, $\Lambda_{3,3} = I_y^{-1}$, $\Lambda_{4,4} = I_z^{-1}$, and

$$\Theta^T \Phi_p(x_p) = \begin{bmatrix} 0_{1\times 3} \\ r_C^\times \end{bmatrix} [F_g(\phi,\theta) - m(\dot{v}_A + \omega^\times v_A)].$$

A typical output signal for inner-loop design problems is $y = [r_Z, \phi, \theta, \psi]^T$.

[72, Ch. 4], $A_{\text{ref}}, B_{\text{ref}}$, and C must be chosen so that $y_{\text{ref}}(\cdot)$ tracks $y_{\text{cmd}}(\cdot)$ with bounded error. This way, the MRAC law (32) guarantees that $y(\cdot)$ eventually tracks $y_{\text{cmd}}(\cdot)$ with bounded error [14, p. 297].

The authors in [42] modify the quadrotor's linearized equations of motion (12) to match the dynamical model (24) and (25) with $n_p = 12$, $m = 4, N = 17$, $x_p = [q^T, \dot{q}^T]^T$, q is given by (9), $\tilde{u} = [u_1 - mg, u_2, u_3, u_4]^T$, $\Phi_p(x_p) = [x^T, y_{\text{cmd}}^T, 1]^T$, $t \geq t_0$ and $y(t) = [(r_A^{\mathbb{I}}(t))^T, \psi(t)]^T$, and the MRAC is applied (32) to regulate the quadrotor's inner loop. MRAC has been also employed in [73] to design inner loops and regulate a quadrotor's pitch and roll dynamics, [37] to account for the misalignment in the propellers' axes, [74] by relaxing the matching condition (31), and [75] to counteract parametric and nonparametric uncertainties. For detailed examples on how to design the inner loops of quadrotors using the MRAC architecture, see "Example 5: MRAC and Inner-Loop Design" and "Example 6: MRAC and Inner-Loop Design."

Figure 6 shows the trajectory of a quadrotor, which implements the MRAC law discussed in [37]. The aircraft takes off from the origin O of the inertial reference frame $\mathbb{I}$ and follows a cylindrical spiral reference trajectory with forward velocity of 1 m/s and vertical velocity of 0.15 m/s; this simulation has been performed using the toolbox presented in [15]. To simulate a fault in the quadrotor's gyroscope, the roll angle fed back to the autopilot's inner loop is set equal to zero in the time interval [5, 9.8] s. After 20 s from take off, the efficiency of one of the motors is reduced to 50%. Furthermore, the state vector fed back to the inner and outer loops is affected by the same white Gaussian noise as in the numerical example discussed in the previous subsection. The quadrotor is clearly able to

Example 6: MRAC and Inner-Loop Design

The inner-loop design strategy outlined in "Example 5: MRAC and Inner-Loop Design" underlies the assumption that the pitch and roll angles are small at all times. To circumvent this limiting assumption, one option is to feedback linearize the inner-loop dynamics and then apply the model reference adaptive control (MRAC) design framework as follows. If $r_C(t) = 0, t \geq t_0$, then it follows from (S1) that

$$\begin{aligned} m\ddot{r}_Z(t) &= \cos\theta(t)\cos\phi(t)u_1(t), \\ r_Z(t_0) &= [0,0,1]r_{A,0}^{\mathbb{I}}, \\ \dot{r}_Z(t_0) &= [0,0,1]v_{A,0}^{\mathbb{I}}, \quad t \geq t_0, \end{aligned}$$

and, by setting $u_1 = \eta_1(\phi,\theta,v_1)$, where

$$\begin{aligned} \eta_1(\phi,\theta,v_1) &= [\cos\phi\cos\phi]^{-1}v_1, \\ (\phi,\theta,v_1) &\in \left(-\frac{\pi}{2},\frac{\pi}{2}\right)\times\left(-\frac{\pi}{2},\frac{\pi}{2}\right)\times\mathbb{R}, \end{aligned}$$

the vertical dynamics is regulated by a double integrator; the virtual control input v_1 can be designed using any control law. Next, it follows from (3) and (5) that

$$[\dot{\phi}(t),\dot{\theta}(t),\dot{\psi}(t),\dot{\omega}(t)]^T = f(\phi(t),\theta(t),\psi(t),\omega(t)) + \begin{bmatrix} 0_{3\times 3} \\ I^{-1} \end{bmatrix} \begin{bmatrix} u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix},$$

$$[\phi(t_0),\theta(t_0),\psi(t_0),\omega^T(t_0)]^T = [\phi_0,\theta_0,\psi_0,\omega_0^T]^T, \quad t \geq t_0,$$

where $f(\phi,\theta,\psi,\omega) = [\omega^T\Gamma^T(\phi,\theta),(-I^{-1}\omega^\times I\omega)^T]^T$. By setting

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = I\Gamma^{-1}(\phi,\theta)\left(\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} - \begin{bmatrix} L_f^2\phi \\ L_f^2\theta \\ L_f^2\psi \end{bmatrix} + v\right), \quad (\phi,\theta,\psi,v) \in \mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}^3,$$

where $L_f^2(\cdot)$ denotes the second Lie derivative [13, p. 510] with respect to f of its argument, the closed-loop system is equivalent to

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p\Lambda[v(t) + \Theta^T\Phi_p(x_p(t))], \quad x_p(t_0) = x_0, \quad t \geq t_0, \\ y(t) &= C_p x_p(t), \end{aligned}$$

where $x_p = [\phi,\dot{\phi},\theta,\dot{\theta},\psi,\dot{\psi}]^T$, $[\dot{\phi}_0,\dot{\theta}_0,\dot{\psi}_0]^T = \Gamma(\phi_0,\theta_0)\omega_0$, $\Lambda \in \mathbb{R}^{3\times 3}$ is diagonal positive-definite, $\Phi(\cdot)$ conveniently chosen, and

$$A_p = \begin{bmatrix} 0&1&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&1 \end{bmatrix}, \quad B_p = \begin{bmatrix} 0&0&0 \\ 1&0&0 \\ 0&0&0 \\ 0&1&0 \\ 0&0&0 \\ 0&0&1 \end{bmatrix}, \quad C_p = \begin{bmatrix} 1&0&0 \\ 0&0&0 \\ 0&1&0 \\ 0&0&0 \\ 0&0&1 \\ 0&0&0 \end{bmatrix}^T;$$

explicit expressions for $L_f^2\phi$, $L_f^2\theta$, and $L_f^2\psi$ are omitted for brevity. In this case, the terms Λ and $\Theta^T\Phi(X_p(\cdot))$ have been introduced to account for model uncertainties and neglected nonlinear terms. The MRAC framework can be now applied to design a robust control law for v. For details, see [13].

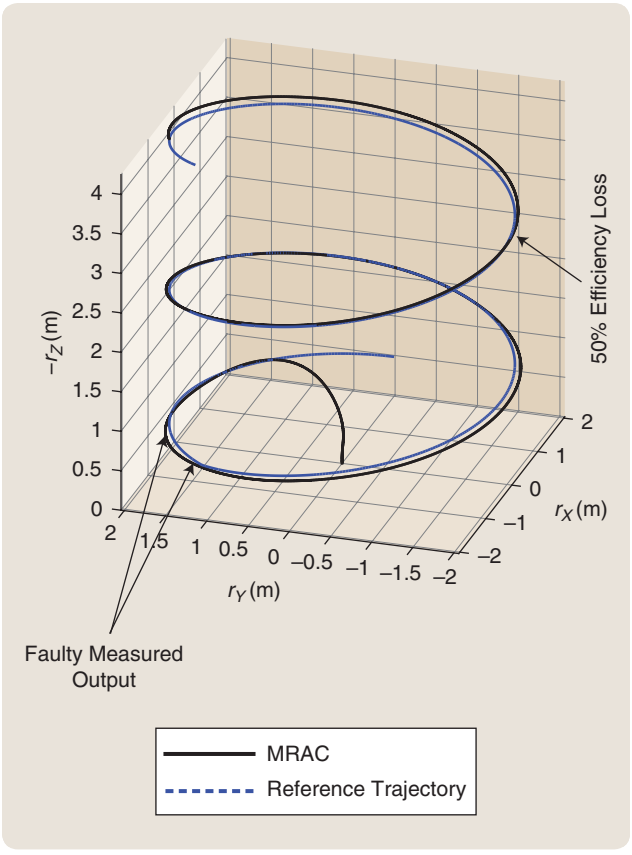

FIGURE 6 The trajectory of a quadrotor implementing a model reference adaptive control law (MRAC). The aircraft takes off from the origin and follows a cylindrical spiral trajectory. In the time interval [5, 9.8] s, the flight controller is unable to measure the roll angle $\phi(\cdot)$. After 20 s, the efficiency of one propeller is reduced to 50%. Despite these faults and uncertainties in the vehicle's inertial properties, the quadrotor still continues following the reference trajectory with good precision.

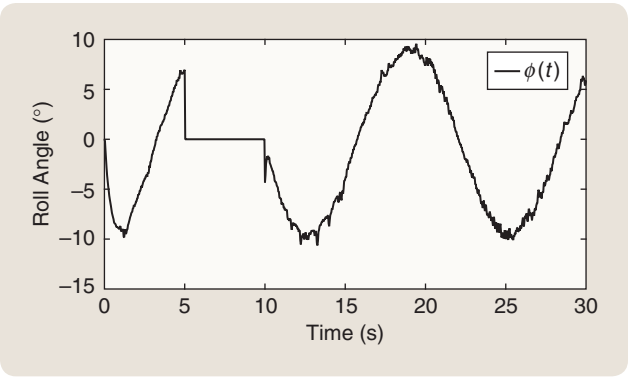

FIGURE 7 The measured roll angle for a quadrotor implementing a model reference adaptive control law. To simulate a fault in the flight controller, the roll angle fed back to the autopilot's inner loop is set equal to zero in the time interval [5, 9.8] s. Moreover, to mimic the noisy behavior of sensors, white Gaussian noise is introduced into the state vector fed back to the controller.

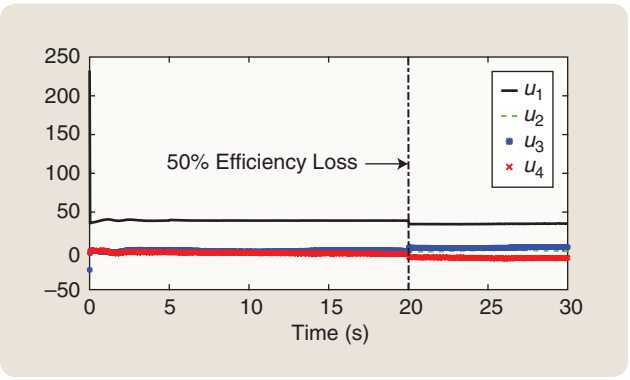

FIGURE 8 Model reference adaptive control inputs for a quadrotor following a cylindrical spiral trajectory. High adaptation gains guarantee satisfactory trajectory tracking but also significant variations in the control inputs.

continue following the reference trajectory despite these challenges. Figure 7 shows the quadrotor's roll angle fed back to the autopilot's inner loop, and Figure 8 shows the quadrotor's control inputs.

Adaptive Sliding Mode Control

As discussed earlier, chattering is a substantial problem in the implementation of autopilots for quadrotors based on sliding mode control. Adaptive sliding mode control provides an approach to overcome this difficulty by merging sliding mode control's robustness to uncertainties with adaptive control's ability to tune the controller's gains and eventually reduce the amplitude of the high-frequency oscillations to a minimal admissible level defined by the conditions for the sliding mode to exist [76]. To achieve this result, however, the closed-loop system's trajectory does not converge to the sliding manifold in the finite time.

The adaptive sliding mode control architecture can be applied to design a quadrotor's inner loop as follows [77]. Let (8) capture a quadrotor's equations of motion with the vector of independent generalized coordinates $q \in \mathbb{R}^6$ given by (9), the control input $u \in \mathbb{R}^4$, $f(t,q,\dot{q})$ given by (10), and $G(q)$ given by (11). Moreover, let $q_j, j = 1,\ldots,6$ denote the ith component of q, $q_{\text{ref},j}$ denote jth component of the vector $q_{\text{ref}} \in R^6$ of reference-independent, generalized coordinates, $f_j(\cdot,\cdot,\cdot)$ the jth component of $f(\cdot,\cdot,\cdot)$, and $G_{i+2,i}(\cdot)$, $i = 1,\ldots,4$ the element on the $(i+2)$th row and ith column of $G(\cdot)$; it is worthwhile to recall that $q_3 = r_Z$, $q_4 = \phi$, $q_5 = \theta$, and $q_6 = \psi$. Assume also that both the quadrotor's mass m and matrix of inertia I are unknown and both the pitch angle $\phi(t), t \geq t_0$, and the roll angle $\theta(t)$ are small; recall that in this case $[\dot{\phi}(t),\dot{\theta}(t),\dot{\psi}(t)]^{\mathrm{T}} = \omega(t)$, $\Gamma(\phi(t),\theta(t)) = \mathbf{1}_3$, and $\dot{\Gamma}(\phi(t),\theta(t)) = 0_{3\times 3}$ in (10) and (11). Finally, let the nonlinear terms $f_{i+2}(\cdot,\cdot,\cdot), i = 1,\ldots,4$ be parameterized so that

$$\begin{bmatrix} G_{3,1}^{-1}(q) f_3(t,q,\dot{q}) \\ G_{4,2}^{-1}(q) f_4(t,q,\dot{q}) \\ G_{5,3}^{-1}(q) f_5(t,q,\dot{q}) \\ G_{6,4}^{-1}(q) f_6(t,q,\dot{q}) \end{bmatrix} = \Theta^{\mathrm{T}} \Phi(t,q,\dot{q}), \quad (t,q,\dot{q}) \in [t_0,\infty) \times \mathcal{D} \times \mathbb{R}^6, \tag{36}$$

where $\mathcal{D} = \mathbb{R}^3 \times [0, 2\pi) \times (-(\pi/2), (\pi/2)) \times [0, 2\pi), \Theta \in \mathbb{R}^9$, captures of the vehicle's unknown inertial properties, the regressor vector is given by

$$\Phi(t, q, \dot{q}) = [\dot{z}, \bar{\Omega}_{\mathrm{P}}(t)\dot{\phi}, \bar{\Omega}_{\mathrm{P}}(t)\dot{\theta}, \dot{\theta}\dot{\psi}, \dot{\phi}\dot{\psi}, \dot{\phi}\dot{\theta}, \dot{\phi}^2, \dot{\theta}^2, \dot{\psi}^2]^{\mathrm{T}}, \quad (37)$$

and $\bar{\Omega}_{\mathrm{P}}(t) \triangleq \Omega_{\mathrm{P},1}(t) - \Omega_{\mathrm{P},2}(t) + \Omega_{\mathrm{P},3}(t) - \Omega_{\mathrm{P},4}(t)$.

The sliding surface is defined as

$$\sigma(q, q_{\mathrm{ref}}) \triangleq [\dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^{\mathrm{T}} - \eta(q, q_{\mathrm{ref}}), \quad (q, q_{\mathrm{ref}}) \in \mathbb{R}^6 \times \mathbb{R}^6, \quad (38)$$

where

$$\eta(q, q_{\mathrm{ref}}) \triangleq \begin{bmatrix} \dot{q}_{\mathrm{ref},3} \\ \dot{q}_{\mathrm{ref},4} \\ \dot{q}_{\mathrm{ref},5} \\ \dot{q}_{\mathrm{ref},6} \end{bmatrix} - K \left(\begin{bmatrix} q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} - \begin{bmatrix} q_{\mathrm{ref},3} \\ q_{\mathrm{ref},4} \\ q_{\mathrm{ref},5} \\ q_{\mathrm{ref},6} \end{bmatrix} \right) \quad (39)$$

and $K \in \mathbb{R}^{4\times 4}$ is a diagonal positive-definite matrix of control gains. Using Lyapunov stability theory, it is possible to prove that the control law

$$u(t, q, q_{\mathrm{ref}}, \hat{\Pi}_1, \hat{\Pi}_2) = \hat{\Pi}_1 [\dot{\eta}(q, q_{\mathrm{ref}}) - K\sigma(q, q_{\mathrm{ref}})] + \hat{\Pi}_2 \Phi(t, q, \dot{q}),$$
$$(t, q, q_{\mathrm{ref}}, \hat{\Pi}_1, \hat{\Pi}_2) \in [t_0, \infty) \times \mathbb{R}^6 \times \mathbb{R}^6 \times \mathbb{R}^{4\times 4} \times \mathbb{R}^{4\times 9} \quad (40)$$

guarantees that

$$\lim_{t\to\infty} |q_{i+2}(t) - q_{\mathrm{ref},i+2}(t)| = 0, \qquad i = 1, \ldots, 4,$$

where

$$\dot{\hat{\Pi}}_1(t) = -M(q(t))\Gamma_\eta \circ [\mathbf{e}_4 \otimes \dot{\eta}(q(t), q_{\mathrm{ref}}(t))],$$
$$\hat{\Pi}_1(t_0) = 0_{4\times 4}, \qquad t \geq t_0, \quad (41)$$
$$\dot{\hat{\Pi}}_2(t) = M(q(t))\Gamma_\Theta [1_9 \circ (\mathbf{e}_9 \otimes \Phi(t, q(t), \dot{q}(t)))],$$
$$\hat{\Pi}_2(t_0) = 0_{4\times 9}, \quad (42)$$

$M(q)$ is a diagonal matrix, $M_{1,1}(q) = 1$, $M_{2,2}(q) = \mathrm{sign}(G_{4,2}(q))$, $M_{3,3}(q) = \mathrm{sign}(G_{5,3}(q))$, $M_{4,4}(q) = \mathrm{sign}(G_{6,4}(q))$, $\Gamma_\eta \in \mathbb{R}^{4\times 4}$ is symmetric positive-definite,

$$\Gamma_\Theta = \begin{bmatrix} \gamma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_2 & \gamma_3 & 0 & 0 & \gamma_4 & 0 & 0 \\ 0 & \gamma_5 & 0 & 0 & \gamma_6 & 0 & 0 & \gamma_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma_8 & 0 & 0 & \gamma_9 \end{bmatrix}, \quad (43)$$

$\gamma_j > 0$, $j = 1, \ldots, 9$, $e_n = [1, \ldots, 1]^{\mathrm{T}} \in \mathbb{R}^n$, $\circ$ denotes the Schur product [72, p. 404], and $\otimes$ denotes the Kronecker product [72, Def. 7.1.2].

In practice, in the adaptive sliding mode control architecture, the closed-loop system's trajectory is steered to a sliding manifold and the gains of the control law are chosen using an adaptive approach so that the magnitude of the control vector (and, hence, the magnitude of chattering) decreases as the closed-loop system's trajectory approaches the sliding manifold. The adaptive sliding mode control has been employed in [78] and [79] to design autopilots for quadrotors that do not require prior knowledge of the uncertainties' upper bound, in [80] to counteract white Gaussian noise in the measured output, in [81] together with a backstepping control architecture that generates the reference pitch and roll angles needed to regulate the vehicle's position in the horizontal plane, and in [82] by employing the projection operator [14, p. 332], [83, Def. B.3] to verify constraints on the propellers' maximum thrust.

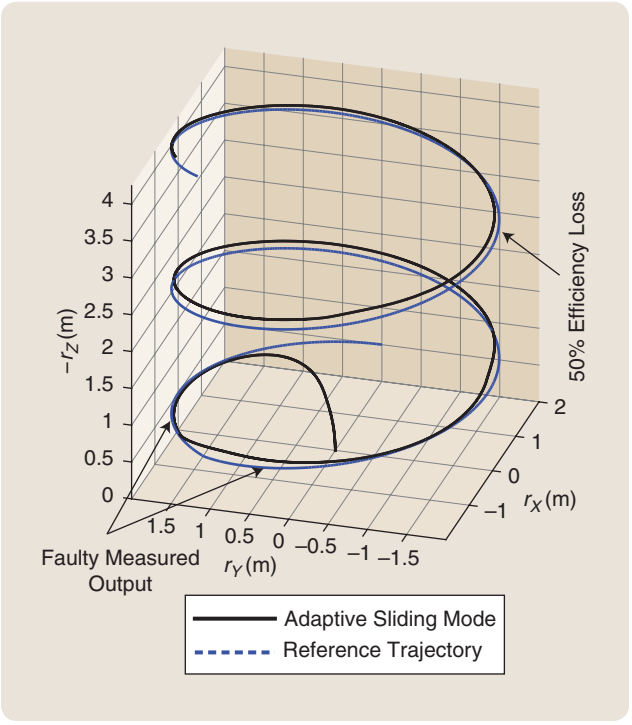

FIGURE 9 The trajectory of a quadrotor implementing an adaptive sliding control law. The aircraft takes off from the origin and follows a cylindrical spiral trajectory. In the time interval [5, 12] s, the flight controller is unable to measure the roll angle $\phi(\cdot)$. After 20 s, the efficiency of one propeller is reduced to 50%. Despite these faults, the quadrotor still continues following the reference trajectory with good precision.

Figure 9 shows the trajectory of a quadrotor, which implements the adaptive sliding mode control law presented in [84]. The aircraft takes off from the origin O of the inertial reference frame $\mathbb{I}$ and follows a cylindrical spiral reference trajectory with forward velocity of 1 m/s and vertical velocity of 0.15 m/s; this simulation has been performed using the toolbox presented in [15]. To simulate a fault in the aircraft's gyroscope, the roll angle fed back to the autopilot's inner loop is set equal to zero in the time interval [5, 9.8] s. Moreover, after 20 s from takeoff, the efficiency of one of the motors is reduced to 50%. The state vector fed back to the inner and outer loops is affected by the same white Gaussian noise as in the numerical examples discussed in the previous subsections. The quadrotor is clearly able to continue following the reference trajectory despite these challenges.

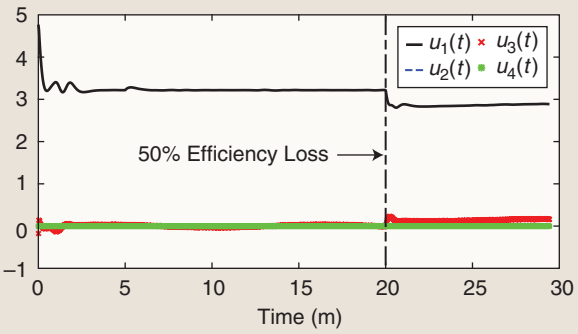

FIGURE 10 Adaptive sliding mode controls for a quadrotor following a cylindrical spiral trajectory. As soon as one of the motor's efficiency is reduced to 50%, the control inputs experience a sharp variation. Clearly, adaptive sliding mode control is not affected by chattering.

Are the Inner and Outer Loops Separated?

The inner and outer loops are usually designed separately. However, is it sufficient to juxtapose an outer loop and an inner loop to create an autopilot for quadrotors? Unfortunately, the answer to this question is negative: assume that the aerodynamic forces and moments are negligible and $r_C(t) = 0, t \geq t_0$. It follows from the equations of motion of a quadrotor (3), (5), and (S1) that the rotational dynamics affects the translational dynamics but not vice-versa. In this case, it follows from [S11, Prop. 4.1] that if both the inner and outer loops are asymptotically stable, then the interconnected system either is able to track a given reference trajectory or diverges. In practice, either the inner and outer loops work well together, or they do not.

Several approaches to solve this problem are available. For instance, one approach is to set the system's initial conditions sufficiently close to the reference trajectory; however, finding the maximum distance between the initial condition and the reference trajectory that guarantees asymptotic convergence of the tracking error is a nontrivial problem. Alternatively, one can assume that the roll and pitch angles are small at all times; in this case, the translational dynamics are decoupled from the rotational dynamics. Otherwise, the inner loop is designed using the backstepping control approach [13, pp. 589–603]; however, backstepping control is usually not robust to modeling errors. Lastly, one can design the control input so that the inner loop is asymptotically stable and the interconnected system given by the outer and inner loops does not diverge, as is discussed in [S12].

REFERENCES

[S11] R. Sepulchre, M. Janković, and P. V. Kokotović, *Constructive Nonlinear Control*. Berlin, Germany: Springer, 1997.

[S12] F. Kendoul, "Nonlinear hierarchical flight controller for unmanned rotorcraft: Design, stability, and experiments," *J. Guid. Control Dyn.*, vol. 32, no. 6, pp. 1954–1958, 2009.

Figure 10 shows the quadrotor's control inputs. This figure clearly shows that the control inputs $u_1(\cdot), \ldots, u_1(\cdot)$ for the adaptive sliding mode control are not affected by chattering. Moreover, comparing Figure 10 with Figure 8, it is apparent that adaptive sliding mode control is less aggressive than MRAC. In this example and the one presented in the previous section, we consider the same quadrotor following the same reference trajectory. Thus, the plot of the roll angle fed back to the inner loop of the quadrotor equipped with the adaptive sliding mode control law resembles the plot shown in Figure 7 and is omitted for brevity.

As previously discussed in this article, the autopilot design problem is usually divided into two parts (namely the outer- and inner-loop design problems), and we illustrated how to design the outer and inner loops separately. Before concluding this survey of nonlinear robust control techniques for quadrotors, the reader is referred to "Are the Inner and Outer Loops Separated?" for further details on the separation between inner and outer loops.

Pros and Cons—Which Control Law to Choose?

In the previous subsections, we presented numerical examples and separately analyzed the performances of autopilots based on sliding mode control, MRAC, and adaptive sliding mode control laws. In this subsection, we compare the results of these numerical examples and make several considerations, some of which are based on our experience in coding and implementing these control laws. This comparative analysis will eventually lead to the selection of a control law to implement on an actual quadrotor.

Although sliding mode control is easy to code and provides acceptable results, this technique is affected by chattering, which prevents its application on actual quadrotors unless a boundary layer is introduced. Moreover, it was difficult to identify the coefficients c and k_i, $i = 1, \ldots, 4$ in (21), that give satisfactory trajectory tracking performances; most satisfactory results have been achieved for smaller values of $c > 0$ and $k_i \in [0, 1)$, $i = 1, \ldots, 4$. For further considerations on the sliding mode control performance, see Table 1; for recommended values of c and k_i, $i = 1, \ldots, 4$, see [15]. Lastly, the quadrotor implementing the sliding mode control law crashed whenever the roll angle fed back to the inner loop was set to zero to mimic some faulty behavior of the control input.

MRAC is relatively easy to code, guarantees a low trajectory tracking error, and is the fastest of the nonlinear control techniques presented in this article. Moreover, the quadrotor implementing the MRAC law was able to follow the reference trajectory despite the fact that the roll angle fed back to the inner loop was set to zero for 4.8 s to mimic some faulty behavior of the aircraft's gyroscope. However, MRAC involves 24 coupled nonlinear differential equations, namely the adaptive laws (33), and eight coupled linear differential equation, namely the reference model (29). In addition, as shown by Figure 8, the control inputs may be very large and saturate the quadrotor's motors. In our experience,

finding Γ_x and Γ_Φ in (33) (which guarantee satisfactory trajectory tracking) may require some effort. If $\|\Gamma_x\| > 100$ and $\|\Gamma_\Phi\| > 10$, then the quadrotor is excessively responsive, and the time needed to integrate the control law (33) grows considerably; the operator $\|\cdot\|$ denotes the equi-induced Euclidean norm. Alternatively, if $\|\Gamma_x\| < 10$ and $\|\Gamma_\Phi\| < 0.5$, then the quadrotor is too slow to track the reference model (29) and (30). For further considerations on the MRAC performance, see Table 1; for recommended values of the adaptation gain matrices Γ_x and Γ_Φ, see [15].

The adaptive sliding mode control is not easy to code and involves 52 coupled nonlinear differential equations, namely, the adaptive laws (41) and (42). Finding Γ_η and Γ_Θ in (41) and (42) that guarantee good trajectory tracking performances was a difficult task. Also in this case, we verified that, if $\|\Gamma_\eta\| > 100$ and $\|\Gamma_\Theta\| > 10$, then the quadrotor is excessively responsive; alternatively, if $\|\Gamma_\eta\| < 10$ and $\|\Gamma_\Theta\| < 1$, then the quadrotor is too slow. For recommended values of Γ_η and Γ_Θ, see [15]. However, as shown by Table 1, the adaptive sliding mode control law is the second fastest among those presented in this article, and, as shown by Figure 10, it does not involve large variations in the control inputs. Lastly, the quadrotor implementing the adaptive sliding mode control law was able to follow the reference trajectory despite the fact that the roll angle fed back to the inner loop was set to zero for 4.8 s to mimic some faulty behavior of the control input.

Figure 11 shows the path-following error of three quadrotors following the same cylindrical spiral reference trajectory as in the examples shown in the previous subsections and implementing the sliding mode control law in [54], the MRAC law in [37], and the adaptive sliding mode control law in [84]. On each quadrotor, the efficiency of one of the propellers is reduced to 75% for all $t \geq 20$ s. From these plots, it appears that the quadrotor implementing the adaptive sliding mode control law suffers from the smallest path-following error. Indeed, the average path-following error is 18.15% for the sliding mode control law, 12.47% for the MRAC law, and 3.18% for the adaptive sliding mode control law. These simulations were completed in 62.72 s by the sliding mode control law, 13.11 s by the MRAC law, and 42.80 s by the adaptive sliding control law; for additional details, see Table 1.

TABLE 1 A comparison of sliding mode control (SMC), model reference adaptive control (MRAC), and adaptive sliding mode control (ASMC) for a quadrotor helicopter following a cylindrical spiral reference trajectory.

	SMC	MRAC	ASMC
Easy to code	Yes	Moderately	No
Quality of control input	Affected by chattering	Smooth, affected by large variations	Smooth and with small variations
Average path-following error with 25% efficiency loss in one propeller	18.15%	12.47%	3.18%
Simulation time	62.72 s	13.11 s	42.80 s

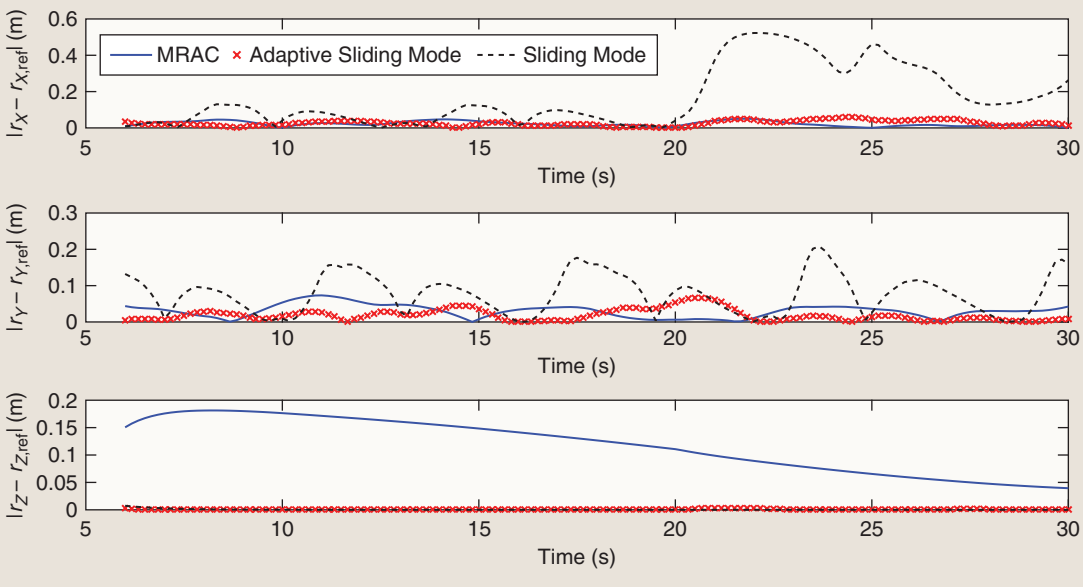

FIGURE 11 The path-following error of three quadrotors implementing a sliding mode control law, a model reference adaptive control (MRAC) law, and an adaptive sliding mode control law, while following a cylindrical spiral trajectory. After 20 s, the efficiency of one propeller is reduced to 75% on each quadrotor. The adaptive sliding mode control appears to suffer from the smallest path-following error.

These qualitative and quantitative considerations hint that the adaptive sliding mode control law may guarantee the most satisfactory results in actual flight experiments, despite its complexity. In the next section, we present the results of a flight test involving a quadrotor aircraft that implements an adaptive sliding mode control law.

FROM THEORY TO FLIGHT

In this section, we present the results of a flight test involving the quadrotor aircraft shown in Figure 12. This quadrotor, which was designed by the Army Research Lab and is used at the University of Oklahoma within the context of a collaborative agreement, employs a control system based on the adaptive sliding mode control framework presented in [84].

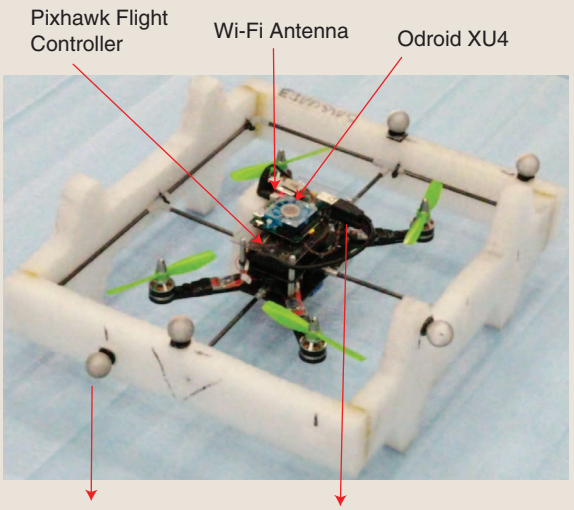

FIGURE 12 A quadrotor utilized for the flight experiments presented in this article. This quadrotor was designed by the Advanced Mobility and Manipulation Team of the Vehicles Technology Directorate at the Army Research Lab and is used at the University of Oklahoma as a testbed for nonlinear control algorithms.

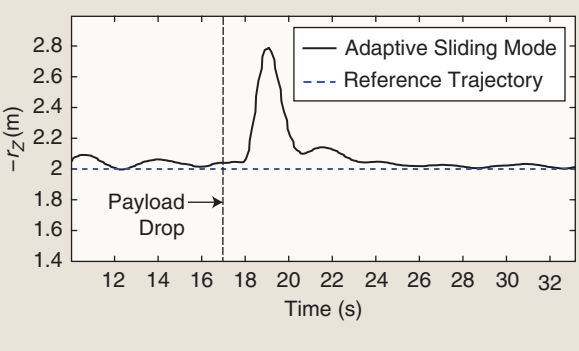

FIGURE 13 The altitude of the quadrotor in the first part of the experiment. The quadrotor's altitude increases rapidly after the payload is dropped. The adaptive sliding mode control law allows the aircraft to return to its reference altitude in approximately 8 s.

The quadrotor's control system comprises an Odroid XU4 microcomputer and a Pixhawk flight controller. The adaptive sliding mode control algorithm is coded in the C++ programming language and implemented on the Odroid XU4 microcomputer. Since the flight tests are performed indoors, the quadrotor's translational position and velocity are deduced by a Vicon motion capture system and transmitted over a Wi-Fi signal to the Odroid microcomputer; the motion capture system's sampling rate is 250 Hz. In outdoor experiments, the aircraft's position and velocity can be alternatively deduced using, for instance, a GPS system. The vehicle's orientation and angular velocity are estimated by the Pixhawk flight controller and transmitted to the microcomputer over a dedicated USB 2.0 cable; the flight controller's output rate is approximately 150 Hz. At each integration time step, the control input (40) is calculated by the microcomputer using the Runge–Kutta method to integrate the adaptation laws (41) and (42). The control input is then transmitted over a dedicated USB 2.0 cable to the flight controller, which employs (6) to determine the thrust force each motor must deliver.

The aircraft carries a payload of 0.364 kg and the quadrotor's mass, including its payload, is 1.380 kg. The adaptive sliding mode controller is designed assuming that the overall vehicle's estimated mass is 0.91 kg. The vehicle's principal moments of inertia are unknown, and the control law is designed assuming that the estimated principal moments of inertia are $\hat{I}_x = 0.0075$ kg m^2, $\hat{I}_y = 0.0075$ kg m^2, and $\hat{I}_z = 0.013$ kg m^2. In this flight test, the quadrotor's tasks are to reach an altitude of 2 m, drop its payload, and then follow a circular trajectory of 1.5-m radius at 1 m from the ground. The autopilot is not informed of when the payload is dropped and is expected to maintain the aircraft at a constant altitude, despite the sudden variation in the quadrotor's mass.

Figure 13 shows the quadrotor's altitude in the first part of the experiment. At 17 s, the payload is dropped and hence, at $t = 17$ s, the vehicle experiences an impulsive force captured by the term $\hat{F}(t) = -m_p g Z$ on the left-hand side of (S2), where $m_p = 0.364$ kg denotes the payload's mass. Approximately 8 s after the payload was dropped, the quadrotor returned to its reference altitude, despite a sudden decrease in the vehicle's mass of 27%. Figure 14 shows the quadrotor's trajectory during the second part of the experiment, that is, while descending to 1-ßm altitude and following a circular trajectory with forward reference velocity of 1 m/s. This is the same forward reference velocity as in the numerical examples presented in the previous section. After 58 s, the control input is modified so that the efficiency of one motor is reduced to 50%. Results achieved in this flight test are also satisfactory, despite large uncertainties in the vehicle's inertial properties, the efficiency loss in one motor, and the noise and delays in the sensors' output, the Wi-Fi connection that transmits the position

In this article, we presented, in a tutorial manner, an analysis and synthesis of nonlinear robust control techniques to design autopilots for quadrotor aircraft.

data to the quadrotor, and the two USB 2.0 cables connecting the flight controller to the microcomputer.

The Odroid XU4 microcomputer comprises a 2-GHz Samsung Exynos5422 Cortex A-15 processor with eight cores and a random access memory (RAM) of 2 GB. The C++ code used to manage the connections between the Odroid XU4 and the Pixhawk flight controller and integrate the adaptive control laws (41) and (42) occupy 10% of the microcomputer's RAM, which is approximately 200 MB. Moreover, these codes require 100% of the computational power of only one of the processor's cores, and it is estimated that each integration step of the Runge–Kutta method used to integrate both (41) and (42) requires only 33 μs to be completed.

CONCLUSION

In this article, we presented, in a tutorial manner, an analysis and synthesis of nonlinear robust control techniques to design autopilots for quadrotor aircraft. First, we presented and analyzed the equations of motion of quadrotors under three sets of progressively restrictive modeling assumptions. Second, we explained how sliding mode control, MRAC, and adaptive sliding mode control design techniques can be applied to create autopilots for quadrotors. Finally, we performed several numerical simulations to analyze the performance of these nonlinear control techniques and selected an adaptive sliding mode control law as the most promising one for precision, robustness, and computational time. The performance of an autopilot based on this adaptive sliding mode control law was shown through indoor flight tests. Future work directions involve a comparative analysis of linear and nonlinear control techniques for quadrotors. Moreover, flight tests in realistic outdoor settings will be performed using commercial off-the-shelf aircraft.

ACKNOWLEDGMENTS

This work was supported in part by the NOAA/Office of Oceanic and Atmospheric Research under NOAA-University of Oklahoma Cooperative Agreement #NA16OAR4320115,

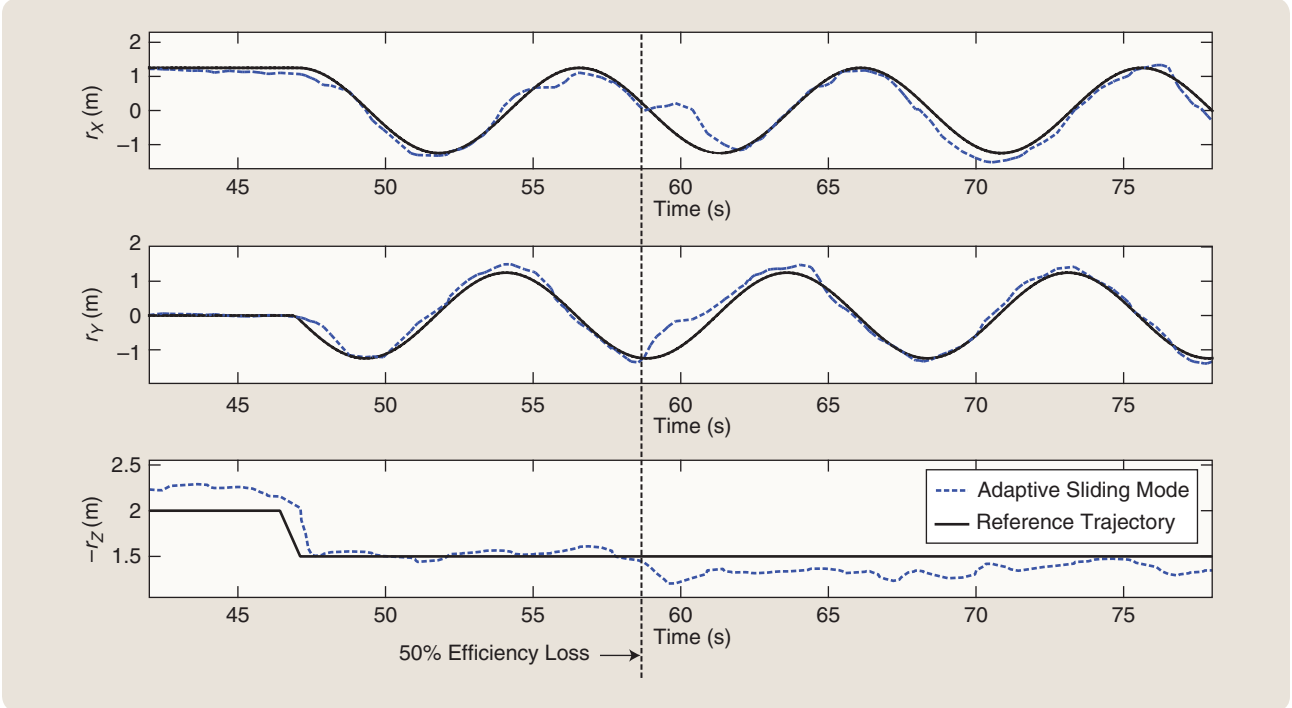

FIGURE 14 The position of the quadrotor in the second part of the experiment. The quadrotor must descend to 1-m altitude and follow a circular path of 1.5-m radius with forward reference velocity of 1 m/s. After 58 s, the efficiency of one propeller is reduced to 50%. Despite this efficiency loss, the uncertainties in the aircraft's inertial properties, and the noise in the signals fed back to the controller, the quadrotor still continues following the reference trajectory.

U.S. Department of Commerce, and the National Science Foundation under Grant 1700640.

The authors wish to thank Harris L. Edge at the Vehicles Technology Directorate of the Army Research Lab (ARL) for his research group's continuous collaboration enabled by the University of Oklahoma–ARL CRADA 17-011-01. The authors also thank James M. Dotterweich at ARL for having successfully integrated the Pixhawk flight controller with the Odroid XU4 microcomputer. Lastly, special credit must be given to the second author for having implemented the adaptive sliding mode controller on the Odroid XU4 microcomputer and performed the flight tests.

AUTHOR INFORMATION

Andrea L'Afflitto (a.lafflitto@ou.edu) received the Ph.D. degree in aerospace engineering from Georgia Tech, the M.S. degree in mathematics from Virginia Tech, and the M.S. and B.S. degrees in aerospace engineering and astronautics from the University of Napoli "Federico II." He is an assistant professor with the School of Aerospace and Mechanical Engineering, University of Oklahoma. His research interests include nonlinear robust control, differential games, and optimal control theory with applications to unmanned aerial systems, such as quadrotors. He is an associate editor for *Control Engineering Practice*, the author of a monograph on flight controls, and the author/coauthor of two book chapters and more than 30 articles in international journals and the proceedings of peer-reviewed conferences. He can be contacted at The University of Oklahoma, School of Aerospace and Mechanical Engineering, Norman, OK 73019 USA.

Robert B. Anderson received the B.S. degree in engineering physics from the University of Oklahoma, where he is currently a graduate student in aerospace engineering. Since the summer of 2016, he has been performing undergraduate research under Dr. L'Afflitto's direct supervision. He was recognized as one of the top eight undergraduate students among the interns at the Army Research Lab in the summer of 2017.

Keyvan Mohammadi received the M.S. degree in electrical engineering from the Amir Kabir University in Iran and the B.S. degree in electrical engineering from K.N. Toosi University in Iran. From August 2016 to August 2017, he was a graduate student at the University of Oklahoma.

REFERENCES

[1] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 90–100, 2012.

[2] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: Methodology and experiments," *Int. J. Robot. Res.*, vol. 27, no. 5, pp. 549–574, 2008.

[3] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *Int. J. Micro Air Vehicles*, vol. 1, no. 4, pp. 1–8, 2009.

[4] Y.-M. Ahn, D. J. Block, and R. S. Sreenivas, "Autonomous navigation and localization of a quadrotor in an indoor environment," *J. Aerosp. Inform. Syst.*, vol. 12, no. 12, pp. 699–709, 2015.

[5] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," in *Proc. 18th Int. Federation Automatic Control World Congress*, 2011, vol. 44, pp. 1485–1491.

[6] C. Richter, A. Bry, and N. Roy, *Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments*. Cham, China: Springer International, 2016, pp. 649–666.

[7] A. Chamseddine, Y. Zhang, C. A. Rabbath, and D. Theilliol, "Trajectory planning and replanning strategies applied to a quadrotor unmanned aerial vehicle," *AIAA J. Guid. Control Dyn.*, vol. 35, no. 15, pp. 1667–1671, 2016.

[8] R. C. Leishman, J. C. Macdonald, R. W. Beard, and T. W. McLain, "Quadrotors and accelerometers: State estimation with an improved dynamic model," *IEEE Control Syst.*, vol. 34, no. 1, pp. 28–41, 2014.

[9] Y. Chen, J. Yu, Y. Mei, S. Zhang, X. Ai, and Z. Jia, "Trajectory optimization of multiple quad-rotor UAVs in collaborative assembling task," *Chin. J. Aeronaut.*, vol. 29, no. 1, pp. 184–201, 2016.

[10] E. Páll, L. Tamás, and L. Buşoniu, *Vision-Based Quadcopter Navigation in Structured Environments*. Cham, China: Springer International, 2015, pp. 265–290.

[11] L. R. García Carrillo, A. E. Dzul López, R. Lozano, and C. Pégard, *Vision-Based Control of a Quad-Rotor UAV*. London: Springer, 2013, pp. 103–137.

[12] A. R. Vetrella, A. Savvaris, G. Fasano, and D. Accardo, "RGB-D camera-based quadrotor navigation in GPS-denied and low light environments using known 3D markers," in *Proc. Int. Conf. Unmanned Aircraft Systems*, 2015, pp. 185–192.

[13] H. K. Khalil, *Nonlinear Systems*. Princeton, NJ: Prentice Hall, 2002.

[14] E. Lavretsky and K. Wise, *Robust and Adaptive Control: With Aerospace Applications*. London, U.K.: Springer, 2012.

[15] R. B. Anderson and A. L'Afflitto. (2018, Jan.). A simulator for quadrotors. [Online]. Available: https://drive.google.com/file/d/1wuLwwEFN74lFqgWYojCiqAQVA7ayRZt/view?usp=sharing

[16] A. Babushkin. (2018, Jan.). jMAVSim. [Online]. Available: https://pixhawk.org/dev/hil/jmavsim

[17] H. Alwi, C. Edwards, and C. P. Tan, *Fault Tolerant Control and Fault Detection and Isolation*. London: Springer, 2011, pp. 7–27.

[18] Y. Zhang, A. Chamseddine, C. Rabbath, B. Gordon, C.-Y. Su, S. Rakheja, C. Fulford, J. Apkarian, and P. Gosselin, "Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed," *J. Franklin Inst.*, vol. 350, no. 9, pp. 2396–2422, 2013.

[19] P. Foehn, D. Falanga, N. Kuppuswamy, R. Tedrake, and D. Scaramuzza, "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *Proc. Robotics Science and Systems*, 2017, pp. 1–10.

[20] A. L'Afflitto, *A Mathematical Perspective on Flight Dynamics and Control*. London: Springer, 2017.

[21] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *Proc. European Control Conf.*, 2013, pp. 3864–3869.

[22] J. C. No, H. Abaunza, and P. Castillo, "Quadrotor quaternion control," in *Proc. Int. Conf. Unmanned Aircraft Systems*, 2015, pp. 825–831.

[23] Z. Zuo, "Global trajectory tracking control of quadrotors with input constraint," in *Proc. Chinese Control Conf.*, 2013, pp. 4426–4431.

[24] L. Wang and H. Jia, "The trajectory tracking problem of quadrotor UAV: Global stability analysis and control design based on the cascade theory," *Asian J. Control*, vol. 16, no. 2, pp. 574–588, 2014.

[25] T. Lee, "Robust adaptive attitude tracking on SO(3) with an application to a quadrotor UAV," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 5, pp. 1924–1930, 2013.

[26] A. Moutinho, M. Figueirôa, and J. R. Azinheira, "Attitude estimation in SO(3): A comparative UAV case study," *J. Intell. Robot. Syst.*, vol. 80, no. 3, pp. 375–384, 2015.

[27] M. D. Shuster, "Survey of attitude representations," *J. Astronaut. Sci.*, vol. 41, no. 4, pp. 439–517, 1993.

[28] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," *Abstr. Appl. Anal.*, vol. 2014, pp. 1–16, Oct. 2014.

[29] C. M. Korpela, T. W. Danko, and P. Y. Oh, "MM-UAV: Mobile manipulating unmanned aerial vehicle," *J. Intell. Robot. Syst.*, vol. 65, no. 1, pp. 93–101, 2012.

[30] K. Mohammadi and A. L'Afflitto, "Robust adaptive output tracking for quadrotor helicopters," in *Adaptive Robust Control and Its Applications*, L. A. Tuan, Ed. Croatia: InTech, 2017, pp. 77–100.

[31] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. AIAA Guidance Navigation and Control Conf.*, 2007, pp. 1–20.

[32] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Automat. Mag.*, vol. 19, no. 3, pp. 20–32, 2012.
[33] W. Z. Stepniewski, *Rotary-Wing Aerodynamics*. New York: Dover, 1984.
[34] M. Bangura and R. Mahony, "Thrust control for multirotor aerial vehicles," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 390–405, 2017.
[35] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard, *Quad Rotorcraft Control: Vision-Based Hovering and Navigation*. London, U.K.: Springer, 2012.
[36] S. Bouabdallah, "Design and control of quadrotors with applications to autonomous flying," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2007.
[37] B. T. Whitehead and S. R. Bieniawski, "Model reference adaptive control of a quadrotor UAV," in *Proc. AIAA Guidance Navigation and Control Conf.*, 2010, pp. 1–13.
[38] D. Xu, J. F. Whidborne, and A. Cooke, "Fault tolerant control of a quadrotor using $\mathcal{L}_1$ adaptive control," *Int. J. Intell. Unmanned Syst.*, vol. 4, no. 1, pp. 43–66, 2016.
[39] I. Fantoni and R. Lozano, *Non-Linear Control for Underactuated Mechanical Systems*. Berlin, Germany: Springer, 2002.
[40] D. Greenwood, *Advanced Dynamics*. New York: Cambridge Univ. Press, 2006.
[41] F. E. Udwadia and R. E. Kalaba, "What is the general form of the explicit equations of motion for constrained mechanical systems?" *Trans. ASME J. Appl. Mech.*, vol. 69, no. 3, pp. 335–339, 2002.
[42] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 4, pp. 1400–1406, 2013.
[43] K. Ogata, *Modern Control Engineering*. Mineola, NY: Prentice Hall, 2010.
[44] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *Int. J. Control*, vol. 76, no. 9–10, pp. 924–941, 2003.
[45] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton, NJ: Princeton Univ. Press, 2008.
[46] A. Filippov, *Differential Equations with Discontinuous Right-Hand Sides*. Dordrecht, The Netherlands: Kluwer Academic, 1988.
[47] L. Friedman and A. Levant, "Higher order sliding modes," in *Sliding Mode Control in Engineering*, W. Perruquetti and J. P. Barbot, Eds. Basel, Switzerland: Marcel Dekker, 2002, ch. 3, pp. 53–102.
[48] J.-J. Slotine and S. S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators," *Int. J. Control*, vol. 38, no. 2, pp. 465–492, 1983.
[49] J. A. Burton and A. S. I. Zinober, "Continuous approximation of variable structure control," *Int. J. Syst. Sci.*, vol. 17, no. 6, pp. 875–885, 1986.
[50] L. Friedman and A. Levant, "Introduction to high-order sliding modes," in *Sliding Mode Control in Engineering*, W. Perruquetti and J. P. Barbot, Eds. Basel, Switzerland: Marcel Dekker, 2002, ch. 1, pp. 1–28.
[51] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993.
[52] H. Nemati, M. Bando, and S. Hokamoto, "Chattering attenuation sliding mode approach for nonlinear systems," *Asian J. Control*, vol. 19, no. 4, pp. 1519–1531, 2017.
[53] S. T. Venkataraman and S. Gulati, "Control of nonlinear systems using terminal sliding modes," *Trans. ASME J. Dyn. Syst. Meas. Control*, vol. 115, no. 3, pp. 554–560, 1993.
[54] M. K. Shaik and J. F. Whidborne, "Robust sliding mode control of a quadrotor," in *Proc. United Kingdom Automatic* Control *Council Int. Conf. Control*, 2016, pp. 1–6.
[55] A. G. Loukianov, "Robust block decomposition sliding mode control design," *Math. Probl. Eng.*, vol. 8, no. 4–5, pp. 349–365, 2002.
[56] F. Chen, K. Zhang, Z. Wang, G. Tao, and B. Jiang, "Trajectory tracking of a quadrotor with unknown parameters and its fault-tolerant control via sliding mode fault observer," *J. Syst. Control Eng.*, vol. 229, no. 4, pp. 279–292, 2015.
[57] H. Wang, X. Ye, Y. Tian, G. Zheng, and N. Christov, "Model-free-based terminal SMC of quadrotor attitude and position," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2519–2528, 2016.
[58] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Trans.*, vol. 53, no. 4, pp. 1350–1356, 2014.
[59] L. Luque-Vega, B. Castillo-Toledo, and A. G. Loukianov, "Robust block second order sliding mode control for a quadrotor," *J. Franklin Inst.*, vol. 349, no. 2, pp. 719–739, 2012.
[60] B. Sumantri, N. Uchiyama, and S. Sano, "Second order sliding mode control for a quad-rotor helicopter with a nonlinear sliding surface," in *Proc. IEEE Conf. Control Applications*, 2014, pp. 742–746.
[61] A. Rezoug, M. Hamerlain, Z. Achour, and M. Tadjine, "Applied of an adaptive higher order sliding mode controller to quadrotor trajectory tracking," in *Proc. IEEE Int. Conf. Control System Computing and Engineering*, 2015, pp. 353–358.
[62] A. Benallegue, A. Mokhtari, and L. Fridman, "High-order sliding-mode observer for a quadrotor UAV," *Int. J. Robust Nonlinear Control*, vol. 18, no. 4–5, pp. 427–440, 2008.
[63] A. Poznyak, "Stochastic sliding mode control: What is this?" in *Proc. Int. Workshop Variable Structure Systems*, 2016, pp. 328–333.
[64] S.-G. Wang, L. Bai, and M. Chen, "Robust sliding mode control of general time-varying delay stochastic systems with structural uncertainties," *Control Theory Technol.*, vol. 12, no. 4, pp. 357–367, 2014.
[65] Y. Niu, Y. Liu, and T. Jia, "Reliable control of stochastic systems via sliding mode technique," *Optimal Control Applicat. Methods*, vol. 34, no. 6, pp. 712–727, 2013.
[66] G. Kreisselmeier and K. Narendra, "Stable model reference adaptive control in the presence of bounded disturbances," *IEEE Trans. Automat. Contr.*, vol. 27, no. 6, pp. 1169–1175, 1982.
[67] K. Narendra and A. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Contr.*, vol. 32, no. 2, pp. 134–145, 1987.
[68] L. Dugard and I. D. Landau, "Stochastic model reference adaptive controllers," in *Proc. IEEE Conf. Decision and Control*, 1980, pp. 1132–1137.
[69] G. C. Goodwin and D. Q. Mayne, "Continuous-time stochastic model reference adaptive control," *IEEE Trans. Automat. Contr.*, vol. 36, no. 11, pp. 1254–1263, 1991.
[70] Y. D. Landau, *Deterministic and Stochastic Model Reference Adaptive Control*. Dordrecht, The Netherlands: Springer, 1981, pp. 387–419.
[71] A. L'Afflitto, "Differential games, partial-state stabilization, and model reference adaptive control," *J. Franklin Inst.*, vol. 354, no. 1, pp. 456–478, 2017.
[72] D. S. Bernstein, *Matrix Mathematics*, 2nd ed. Princeton, NJ: Princeton Univ. Press, 2009.
[73] M. Schreier, "Modeling and adaptive control of a quadrotor," in *Proc. IEEE Int. Conf. Mechatronics and Automation*, 2012, pp. 383–390.
[74] J. M. Selfridge and G. Tao, "A multivariable adaptive controller for a quadrotor with guaranteed matching conditions," *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 24–33, 2014.
[75] M. Mohammadi and A. M. Shahri, "Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation," *J. Intell. Robot. Syst.*, vol. 72, no. 1, pp. 105–122, 2013.
[76] V. I. Utkin and A. S. Poznyak, *Adaptive Sliding Mode Control*. Berlin, Germany: Springer, 2013, pp. 21–53.
[77] H. Bouadi, S. S. Cunha, A. Drouin, and F. Mora-Camino, "Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking," in *Proc. IEEE Int. Symp. Computational Intelligence and Informatics*, 2011, pp. 449–455.
[78] S. Islam, M. Faraz, R. K. Ashour, G. Cai, J. Dias, and L. Seneviratne, "Adaptive sliding mode control design for quadrotor unmanned aerial vehicle," in *Proc. Int. Conf. Unmanned Aircraft Systems*, 2015, pp. 34–39.
[79] N. Sudhir and A. Swarup, "On adaptive sliding mode control for improved quadrotor tracking," *J. Vibration Control*, pp. 1–12, 2017. doi: 10.1177/1077546317703541.
[80] S. Li, B. Li, and Q. Geng, "Adaptive sliding mode control for quadrotor helicopters,"in *Proc. 33rd Chinese Control Conf.*, 2014, pp. 71–76.
[81] T. X. Dinh and K. K. Ahn, "Adaptive tracking control of a quadrotor unmanned vehicle," *Int. J. Precis. Eng. Manuf.*, vol. 18, no. 2, pp. 163–173, 2017.
[82] Y. Zou, "Nonlinear robust adaptive hierarchical sliding mode control approach for quadrotors," *Int. J. Robust Nonlinear Control*, vol. 27, no. 6, pp. 925–941, 2017.
[83] N. Hovakimyan and C. Cao, $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.
[84] L. Wang, C. He, and P. Zhu, "Adaptive sliding mode control for quadrotor aerial robot with type configuration," *Int. J. Robust Nonlinear Control*, vol. 3, no. 1, pp. 20–26, 2014.