

Research Article

A Novel Family of Exact Nonlinear Cascade Control Design Solutions for a Class of UAV Systems

Kristian Maya-Gress ¹, **Jorge Álvarez** ², **Raúl Villafuerte-Segura** ¹,
Hugo Romero-Trejo ¹ and **Miguel Bernal** ²

¹Research Center on Information Technology and Systems, Hidalgo State University, Pachuca-Hidalgo C. P. 42184, Mexico

²Department of Electrical and Electronics Engineering of the Sonora Institute of Technology, 5 de Febrero 818 Sur, Cd. Obregon C. P. 85000, Mexico

Correspondence should be addressed to Miguel Bernal; miguel.bernal@itson.edu.mx

Received 13 July 2021; Revised 7 September 2021; Accepted 13 November 2021; Published 29 November 2021

Academic Editor: Antonios Tsourdos

Copyright © 2021 Kristian Maya-Gress et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this work, a novel family of exact nonlinear control laws is developed for trajectory tracking of unmanned aerial vehicles. The proposed methodology exploits the cascade structure of the dynamic equations of most of these systems. In a first step, the vehicle position in Cartesian coordinates is controlled by means of fictitious inputs corresponding to the angular coordinates, which are fixed to a combination of computed torque and proportional-derivative elements. In a second step, the angular coordinates are controlled as to drive them to the desired fictitious inputs necessary for the first part, resulting in a double-integrator 3-input cascade control scheme. The proposal is put at test in two examples: 4-rotor and 8-rotor aircrafts. Numerical simulations of both plants illustrate the effectiveness of the proposed method, while real-time results of the first one confirm its applicability.

1. Introduction

Unmanned aerial vehicles (UAVs) have become a topic of interest in many works due to the fact that they are capable of operating in degraded environments which might be dangerous for humans. These vehicles are designed to fly with high agility and rapid maneuvering, even under wind gusts. For all these qualities, the UAVs have a wide variety of applications such as military [1], 3D mapping and aerial photography [2, 3], and inspection of places that are not easily accessible or are too dangerous for humans [4], among others [5, 6].

There is a great variety of UAVs that can be classified by a broad number of performance characteristics such as their structure, weight, endurance, range, speed, and so on [7]. In this work, we will focus on plants with a cascade structure that allows for a family of novel cascade nonlinear control laws to be applied, e.g., the quadrotor and the 8-rotor aircrafts. These plants consist in a structure of symmetrical links and 4 or 8 rotors at its ends, respectively. In the case of

the quadrotor, given that the front and the rear motors rotate counter-clockwise while the other two rotate clockwise, gyroscopic effects and aerodynamic torques tend to cancel during flight. A similar situation is produced in the 8-rotor aircraft. Both plants include the rotation motion of three axes and the linear motion of the center of gravity along three axes. Thus, there are six motions: back/front, left/right, up/down, yaw, pitch, and roll. The quadrotor has 4 control inputs and 6 degrees of freedom; its dynamics has the characteristics of nonlinearity and strong coupling [8]. On the other hand, the 8-rotor has 6 control inputs and 6 degrees of freedom [9].

In the literature, there are many works about control laws applied to UAVs. Most of them do not require precise or complete knowledge of the dynamic equations of the plant. Indeed, one of the most basic control structures, the family of proportional-integral-derivative (PID) controllers, which requires tuning of its gains, can be applied successfully to these plants as they are capable of bringing the error signals to zero in trajectory tracking [10–12]. Other reports

incorporate fuzzy logic in combination with PID controllers, thus easing the tuning of the gains and increasing the adaptability of the control law to changing plant or environment parameters; they usually have better performance than classic PID [13–16]. Dead-zone and actuator faults have also been studied in this context [17]. Recent works present solutions based on neural networks for the autonomous landing of a quadrotor both on fixed and moving targets for maritime search-and-rescue applications [18]. In [19], a neural network adaptive scheme is combined with sliding mode control, which preserves the advantages of both methods, namely, adaptability and robustness against coupled perturbations.

On the other hand, there are also control laws that require knowledge of the plant, such as linear quadratic regulator (LQR) techniques that can be found in [20], where a linearization of the plant is necessary to design the controller; nonlinear dynamic models based on quaternions for attitude and LQR control are presented in [21]; in [22], a controller design using the backstepping approach has been applied to the state-space model of a quadrotor, while in [23], the model is used in the Lagrange–Euler form; in [24], a backstepping control and nonlinear disturbance observer have been developed; the observer is constructed separately from the controller to estimate the external disturbances and compensate for the negative effects of the disturbances such as wind gusts.

Yet, another category of controllers uses partial information of the model, e.g., sliding modes in [25–27], which besides being robust and simple, produces high-frequency switching of the control signals (chattering phenomenon); integral predictive and nonlinear robust \mathcal{H}_∞ strategy [28], where the integral of the position error is considered, allowing the achievement of a null steady-state error when sustained disturbances are acting on the system; and a hybrid finite-time control approach [29] for trajectory tracking with unknown dynamics and disturbances. Furthermore, many recent works have shown different approaches for trajectory tracking of the quadrotor such as finite-time output feedback schemes [30], adaptive control [31, 32], supertwisting [33], active disturbance rejection [34], and neural networks with backstepping [35].

1.1. Contribution. The contribution of this article is to provide a family of exact nonlinear cascade control schemes for UAV plants such as the quadrotor and the 8-rotor aircraft, which, due to their characteristics, have a cascade interconnection of groups of states. In contrast with other schemes in the literature, the control laws are exact, not approximate, and allow a variety of options to be easily incorporated via linear matrix inequalities (LMIs) [36], which are efficiently solved via convex optimization software [37].

1.2. Organization. The remainder of this study is organized as follows: Section 2 defines the family of systems to be studied, the basics on the computed torque technique, the mathematical models of the UAVs under study, and the

difficulties for implementing a computed torque-like control scheme: a problem statement is made; in Section 3, the main results concerning the family of exact nonlinear cascade controllers are presented; in Section 4, numerical simulations are provided proving the flexibility and efficiency of the control proposal; real-time results are shown in Section 5; and finally, conclusions are gathered in Section 6 along with a discussion on future work.

2. Preliminaries

This section describes the mathematical structure expected for the plants under consideration, the specific dynamic models of the two UAVs that will be used for illustration of the proposal, basics on the computed torque technique, and the problem statement of this work.

2.1. Cascade Plants. The plants under consideration must have one or more nested cascade structures of the following form:

$$\begin{aligned}\eta(t) &= f(\xi, u_1), \\ \xi(t) &= u_2,\end{aligned}\tag{1}$$

where $\eta \in \mathbb{R}^{n-p}$ and $\xi \in \mathbb{R}^p$ are associated with the plant states; the control inputs are $u_1 \in \mathbb{R}^{m-p}$ and $u_2 \in \mathbb{R}^p$; $f: \mathbb{R}^p \times \mathbb{R}^{m-p} \rightarrow \mathbb{R}^{n-p}$ is a sufficiently smooth nonlinear function in a domain D that contains the origin $(\eta, \xi) = (0, 0)$ with $f(0, 0) = 0$. As it will be seen later, UAVs belong to the former class as η is usually associated with Cartesian coordinates, ξ stands for the angular coordinates, u_1 and u_2 are usually the result of a simplification of the original inputs (i.e., from the voltage/speed inputs to the minimum number of torque ones), and $f(\cdot, \cdot)$ is clearly a nonlinear input distribution vector from the point of view of ξ and u_1 ; this point of view will be exploited later.

Most UAVs require trajectory tracking in the Cartesian coordinate space; therefore, following the analogy established between the model (1) and a UAV model, it is clear that trajectory tracking reduces to find inputs such that $\eta(t)$ is asymptotically driven to a desired trajectory $\eta_d(t)$. Note that, despite the cascade structure, backstepping is not directly applicable as the double-integrator and strong coupling of states in the first equation, precludes the designer from following the usual path. Indeed, most UAVs do not have polynomial coupling of states, but trigonometric ones. This means that backstepping cannot be performed as normally done because the required fictitious inputs cannot be straightforwardly determined [38].

Moreover, backstepping asymptotic stability of the origin is guaranteed via the direct Lyapunov method. But trajectory tracking requires writing the dynamic equations of the tracking error system in a form suitable for Lyapunov analysis, i.e., given the tracking error $e(t) = \eta_d(t) - \eta(t)$, being able to write

$$\begin{aligned}e(t) &= \eta_d(t) - \eta(t) = \eta_d(t) - f(\xi, u_1) \\ &= F(\eta, \dot{\eta}, \eta_d, \dot{\eta}_d, \xi, u_1)e(t),\end{aligned}\tag{2}$$

where, obviously, the form of $F(\cdot)$ might be very hard or impossible to obtain. The methodology to be presented in this study will circumvent this problem; once this is done, if a system can be put in a series of cascade connections such as the one presented above, the technique can be recursively applied.

2.2. UAV Dynamic Models. Since the UAVs employed for illustration of our proposal have similar underlying physical principles, we will focus on the development of the quadrotor model. This is standard material which, nevertheless, may help the reader understanding the origin and meaning of certain terms in the model.

The configuration of the quadrotor is composed by a rigid cross frame and four rotors as shown in Figure 1. The quadrotor is an underactuated and nonlinear coupled system with six degrees of freedom; its mathematical model will be obtained using the well-known Lagrangian method. To this end, let us define $q = (\eta, \xi) \in \mathbb{R}^6$ as the generalized coordinates vector for the quadrotor, where $\eta = (x, y, z) \in \mathbb{R}^3$ is the position of the center of mass of the quadrotor relative to the frame \mathcal{R}_o (Figure 1), and $\xi = (\psi, \theta, \phi) \in \mathbb{R}^3$ are the Euler angles (yaw, pitch, and roll) that describe the orientation of the aircraft. The kinetic energy of both translational and rotational motions are expressed, respectively, as

$$\begin{aligned} T_{\text{tra}} &= \frac{m}{2} \dot{\eta}^T \dot{\eta}, \\ T_{\text{rot}} &= \frac{1}{2} \dot{\xi}^T J \dot{\xi}, \end{aligned} \quad (3)$$

where m denotes the mass of the aircraft, and $J = W_n^T \mathcal{F} W_n$ is the inertia matrix for the rotational kinetic energy, with

$$\begin{aligned} W_n &= \begin{bmatrix} -\sin \theta & 0 & 1 \\ \sin \phi \cos \theta & \cos \phi & 0 \\ \cos \phi \cos \theta & -\sin \phi & 0 \end{bmatrix}, \\ \mathcal{F} &= \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \end{aligned} \quad (4)$$

The only force that contributes to the potential energy is the force due to gravitational acceleration, which is expressed as $U = mgz$, where z is the altitude of the quadrotor, and g is the gravitational constant. Thus, the Lagrangian is given by

$$\mathcal{L}(q, \dot{q}) = T_{\text{tra}} + T_{\text{rot}} - U = \frac{m}{2} \dot{\eta}^T \dot{\eta} + \frac{1}{2} \dot{\xi}^T J \dot{\xi} - mgz. \quad (5)$$

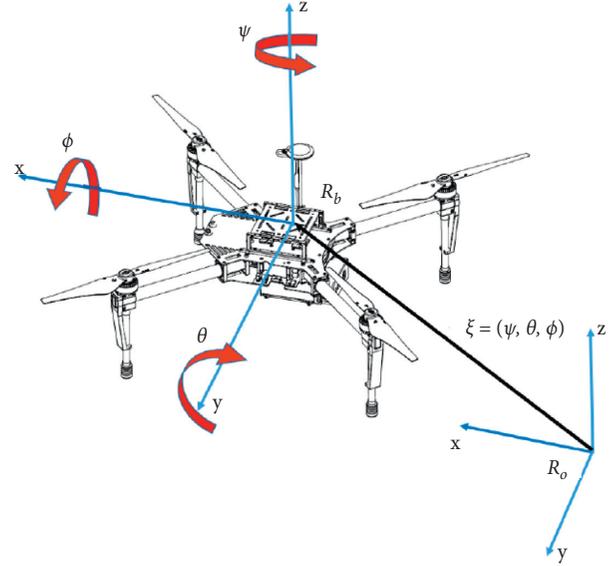


FIGURE 1: Scheme of the quadrotor system.

Considering the Euler-Lagrange equations and taking into account the vector of generalized external forces, the dynamic model of the quadrotor is defined as

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = F, \quad (6)$$

where $F = [F_\eta^T \ \tau^T]^T$ is the vector of generalized forces. F_η groups the forces produced by the control inputs that produce a translational movement, and τ is the vector of generalized moments produced by the control inputs to perform rotational movements. The small body forces are ignored due to their being much smaller compared to the forces produced by the main control inputs. Then, the forces applied to the quadrotor with respect to the reference \mathcal{R}_b are given by

$$\begin{aligned} F_b &= \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}, \\ \tau &= \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix}, \end{aligned} \quad (7)$$

where $u = f_1 + f_2 + f_3 + f_4$, $f_i = k_i w_i^2$ for $i = 1, 2, 3, 4$, and k_i , a positive constant parameter depending on the density of air, and w_i is the angular speed of the i^{th} motor. This set of forces expressed in \mathcal{R}_o is transformed into

$$F_\eta = \mathbf{R} F_b, \quad \text{with } \mathbf{R} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi & \sin \psi \sin \phi + \sin \theta \cos \phi \cos \psi \\ \sin \psi \cos \theta & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \theta \sin \psi \cos \phi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}, \quad (8)$$

where \mathbf{R} is the transformation matrix representing the orientation of the quadrotor from \mathcal{R}_b to \mathcal{R}_o . Furthermore, the vectors of generalized moments acting on variables ξ are as follows:

$$\begin{aligned}\tau_\psi &= \sum_1^4 k_{Mi} w_i^2, \\ \tau_\theta &= (f_2 - f_4)l_c, \\ \tau_\phi &= (f_3 - f_1)l_c,\end{aligned}\quad (9)$$

where k_{Mi} is the moment constant of the i^{th} motor, and l_c is the distance between the axes of the motors and the center of gravity. Since the Lagrangian (5) does not contain any cross-term combining η and ξ , the Euler–Lagrange equation (6) can be split into two subsystems, one for the translational dynamics η and the other for the rotational dynamics ξ as follows:

$$\begin{aligned}m\eta + [0 \ 0 \ mg]^T &= F_\eta, \\ J\dot{\xi} + C(\xi, \dot{\xi})\dot{\xi} &= \tau,\end{aligned}\quad (10)$$

where $C(\xi, \dot{\xi})$ is the Coriolis matrix. In order to simplify the analysis, the following change of the input variables is proposed:

$$\tau = C(\xi, \dot{\xi})\dot{\xi} + J\tilde{\tau}.\quad (11)$$

Here, $\tilde{\tau} = [\tilde{\tau}_\psi \ \tilde{\tau}_\theta \ \tilde{\tau}_\phi]^T$ are the new inputs ([8] for details). Then, $\xi = \tilde{\tau}$ and equation (10) are rewritten as

$$\begin{aligned}x &= \frac{(\sin \phi \sin \psi + \sin \theta \cos \phi \cos \psi)u}{m}, \\ y &= \frac{(\sin \theta \sin \psi \cos \phi - \sin \phi \cos \psi)u}{m}, \\ \ddot{z} &= \frac{(\cos \theta \cos \phi)u}{m} - g, \\ \psi &= \tilde{\tau}_\psi, \\ \ddot{\theta} &= \tilde{\tau}_\theta, \\ \phi &= \tilde{\tau}_\phi,\end{aligned}\quad (12)$$

where the horizontal plane coordinates are represented by (x, y) , the vertical position is represented by z , ψ is the yaw angle around the z axis, θ is the pitch angle around the y axis, and ϕ is the roll angle around the x axis. The system (12) has four control inputs u , $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$, and $\tilde{\tau}_\phi$.

2.3. Computed Torque. Computed torque control is a technique used for trajectory tracking in rigid robotic manipulators consisting of open kinematic chains [39]. This sort of models is amenable to the Lagrange–Euler form:

$$M(q)\dot{q}(t) + V(q, \dot{q}) + F(q, \dot{q}) + G(q) = \tau(t),\quad (13)$$

where $q \in \mathbb{R}^n$ is a vector gathering the n joint variables of the kinematic chain, $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $V(q, \dot{q}) \in \mathbb{R}^n$ is the Coriolis vector, $F(q, \dot{q}) \in \mathbb{R}^n$ accounts for the friction forces (whether viscous, dry, or others), $G(q) \in \mathbb{R}^n$ is the gravity vector, and $\tau(t) \in \mathbb{R}^n$ is the generalized force vector. Traditional computed torque control considers the plant as fully actuated, i.e., all entries in $\tau(t)$ are available for control purposes or it has one available actuator per joint. A desired trajectory $q_d(t)$ can be asymptotically tracked by $q(t)$ if the following control law is used:

$$\tau(t) = M(q)(\dot{q}_d(t) - u(t)) + V(q, \dot{q}) + F(q, \dot{q}) + G(q),\quad (14)$$

where $u(t)$ is responsible for stabilizing the tracking error system:

$$\begin{bmatrix} \dot{e}(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u(t),\quad (15)$$

with $e(t) = q_d(t) - q(t)$. Control law (14) is known as the inner-loop control feedback; it is based on exact feedback linearization [40]. Control law $u(t)$ is known as the outer-loop control feedback; it stabilizes the error system (15), a task usually achieved by linear state feedback or PD control:

$$u(t) = -K_p e(t) + K_v \dot{e}(t),\quad (16)$$

where for simplicity, gains K_p and K_v are the diagonal constant matrices with positive entries.

Clearly, computed torque is critically dependent on the precise knowledge of the model and does not take advantage of the nonlinear structure of the plant, let alone the cascade form of UAV models.

2.4. Problem Statement. For a UAV model of the form (1) to perform trajectory tracking of a Cartesian path $\eta_d(t) = [x_d(t) \ y_d(t) \ z_d(t)]$ by means of its inputs u_1 and u_2 , i.e., $\lim_{t \rightarrow \infty} e(t) = 0$ with $e(t) = \eta_d(t) - \eta(t)$, $\eta(t) = [x(t) \ y(t) \ z(t)]$, $\xi = [\psi \ \theta \ \phi]$, backstepping is not a straightforward option. Indeed, the double-integrator and the trigonometric functions in ξ impede its application.

Moreover, computed torque cannot be applied either because, following its notation, the joint vector $q = [x \ y \ z \ \psi \ \theta \ \phi]^T$ and the generalized force vector $\tau = [u \ \tilde{\tau}_\psi \ \tilde{\tau}_\theta \ \tilde{\tau}_\phi]^T$ do not share dimensions, i.e., there is not enough number of actuators as required by the computed torque technique.

In the next section, a novel family of exact nonlinear control laws is developed for trajectory tracking of UAVs; it will exploit the cascade structure without recurring to backstepping and will be able to deal with the underactuated characteristics that the computed torque technique is unable to cope. In contrast with other proposals, no approximations are employed; all the nonlinearities are taken into account for control purposes.

Notation: in matrix expressions, 0 and I stand for a zero and identity matrix, respectively, whose dimensions can be inferred from the context; $>$ and $<$ stand for positive and

negative-definiteness; the symbol $(*)$ denotes the transpose of the expression on the left, i.e., $A + (*) = A + A^T$.

3. Main Results

In this section, a family of nonlinear cascade controllers for trajectory tracking of UAVs with cascade model structure is proposed. To begin with, we reproduce the general mathematical model of the UAVs under consideration:

$$\begin{aligned} \eta(t) &= f(\xi, u_1), \\ \xi(t) &= u_2, \end{aligned} \quad (17)$$

where $\eta \in \mathbb{R}^3$ and $\xi \in \mathbb{R}^3$ are associated with Cartesian and angular coordinates, respectively; the simplified control inputs are $u_1 \in \mathbb{R}^m$ and $u_2 \in \mathbb{R}^3$; $f(\cdot, \cdot): \mathbb{R}^3 \times \mathbb{R}^m \rightarrow \mathbb{R}^3$ is a sufficiently smooth nonlinear function in a domain D that contains the origin $(\eta, \xi) = (0, 0)$ with $f(0, 0) = 0$.

Consider a desired trajectory in Cartesian coordinates $\eta_d \in \mathbb{R}^3$ such that $\eta_d(t) = [x_d(t) \ y_d(t) \ z_d(t)]^T$, where $x_d(t)$, $y_d(t)$, and $z_d(t)$ are the desired trajectories in the direction of the x , y , and z axis, respectively; let the corresponding tracking error be $e_\eta(t) = \eta_d(t) - \eta(t)$; thus, taking into account (17), we have

$$e_\eta(t) = \eta_d(t) - \eta(t) = \eta_d(t) - f(\xi, u_1), \quad (18)$$

$$\xi(t) = u_2. \quad (19)$$

Thanks to the cascade structure above, the following assumption can be made:

Assumption 1. There exists ξ_d and u_1 such that

$$\eta_d - f(\xi, u_1) = -K_{p\eta}e_\eta - K_{v\eta}\dot{e}_\eta, \quad (20)$$

where $K_{p\eta} \in \mathbb{R}^{3 \times 3}$ and $K_{v\eta} \in \mathbb{R}^{3 \times 3}$ are the matrices with proportional and derivative gains, respectively, to be defined (usually diagonal with positive entries); otherwise, claim the approach fails.

Despite its complexity, note that the existence of ξ_d and u_1 holding Assumption 1 is not only guaranteed for most of the UAV models but also a family of solutions. Indeed, as variables in ξ are angle coordinates, they come into trigonometric expressions which naturally lead to multiple solutions, if any.

Once ξ_d and u_1 holding Assumption 1 are found, the next problem is to drive ξ towards ξ_d , i.e., to drive the tracking error $e_\xi(t) = \xi_d(t) - \xi(t)$ to 0 as $t \rightarrow \infty$. To this end, computed torque techniques come at the hand, i.e., from (19), we have

$$u_2 = \xi_d + K_{p\xi}e_\xi + K_{v\xi}\dot{e}_\xi, \quad (21)$$

where $K_{p\xi} \in \mathbb{R}^{3 \times 3}$ and $K_{v\xi} \in \mathbb{R}^{3 \times 3}$ are the matrices with proportional and derivative gains, respectively, to be defined (usually diagonal with positive entries).

The whole tracking error system, once Assumption 1 and computed torque control law (21) are employed, is given by

$$\frac{d}{dt} \begin{bmatrix} e_\eta \\ \dot{e}_\eta \\ e_\xi \\ \dot{e}_\xi \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ -K_{p\eta} & -K_{v\eta} & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_{p\xi} & -K_{v\xi} \end{bmatrix} \begin{bmatrix} e_\eta \\ \dot{e}_\eta \\ e_\xi \\ \dot{e}_\xi \end{bmatrix}. \quad (22)$$

The main result of this work can now be stated by defining the overall tracking error signal as $e = [e_\eta^T \ \dot{e}_\eta^T \ e_\xi^T \ \dot{e}_\xi^T]^T$.

Theorem 1. *The origin $e = 0$ of the tracking error system (22) is asymptotically stable if there exists matrices $X_1 \in \mathbb{R}^{6 \times 6}$, $X_2 \in \mathbb{R}^{6 \times 6}$, $X_1 = X_1^T$, $X_2 = X_2^T$, $M_{p\eta} \in \mathbb{R}^{3 \times 3}$, $M_{v\eta} \in \mathbb{R}^{3 \times 3}$, $M_{p\xi} \in \mathbb{R}^{3 \times 3}$, and $M_{v\xi} \in \mathbb{R}^{3 \times 3}$, such that the following LMIs hold:*

$$\begin{aligned} X_1 &> 0, \\ X_2 &> 0, \\ \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} X_1 + \begin{bmatrix} 0 \\ I \end{bmatrix} [M_{p\eta} \ M_{v\eta}] &< 0, \\ \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} X_2 + \begin{bmatrix} 0 \\ I \end{bmatrix} [M_{p\xi} \ M_{v\xi}] &< 0. \end{aligned} \quad (23)$$

In that case, the gains are given by

$$\begin{aligned} [-K_{p\eta} \ -K_{v\eta}] &= [M_{p\eta} \ M_{v\eta}] X_1^{-1}, \\ [-K_{p\xi} \ -K_{v\xi}] &= [M_{p\xi} \ M_{v\xi}] X_2^{-1}. \end{aligned} \quad (24)$$

Proof. The closed-loop tracking error system (22) can be rewritten as two independent ones, namely,

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} &= \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} [-K_{p\eta} \ -K_{v\eta}] \right) \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix}, \\ \frac{d}{dt} \begin{bmatrix} e_\xi \\ \dot{e}_\xi \end{bmatrix} &= \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} [-K_{p\xi} \ -K_{v\xi}] \right) \begin{bmatrix} e_\xi \\ \dot{e}_\xi \end{bmatrix}. \end{aligned} \quad (25)$$

Let us focus on the first one as the proof follows the same lines for both of them. Consider the quadratic Lyapunov function candidate $V = [e_\eta^T \ \dot{e}_\eta^T] P_1 [e_\eta^T \ \dot{e}_\eta^T]^T$, where $P_1 = X_1^{-1} > 0$; its time derivative is

$$\begin{aligned} \dot{V} &= \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix}^T P_1 \begin{bmatrix} \dot{e}_\eta \\ e_\eta \end{bmatrix} + \begin{bmatrix} \dot{e}_\eta \\ e_\eta \end{bmatrix}^T P_1 \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} \\ &= \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix}^T \left(P_1 \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} [-K_{p\eta} \ -K_{v\eta}] \right) + (*) \right) \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix}. \end{aligned} \quad (26)$$

Condition $\dot{V} < 0$ for $(e_\eta, \dot{e}_\eta) \neq (0, 0)$ is satisfied if

$$P_1 \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} -K_{p\eta} & -K_{v\eta} \end{bmatrix} \right) + (*) < 0, \quad (27)$$

which pre and postmultiplying by $X_1 = P_1^{-1}$ is equivalent to the third LMI in (23), provided that $\begin{bmatrix} -K_{p\eta} & -K_{v\eta} \end{bmatrix} X_1 = \begin{bmatrix} M_{p\eta} & M_{v\eta} \end{bmatrix}$, thus proving that $V(t)$ is a valid Lyapunov function establishing asymptotic stability of $\begin{bmatrix} e_\eta^T & \dot{e}_\eta^T \end{bmatrix}^T = 0$.

The same analysis establishes asymptotic stability of $\begin{bmatrix} e_\xi^T & \dot{e}_\xi^T \end{bmatrix}^T = 0$, thus concluding the proof. \square

Remark 1. As pointed out before, for any UAV model belonging to the family of systems under consideration, there is always a variety of solutions leading to a linear tracking error system; in this sense, the technique mimics computed torque control [39]. Since some states are employed as fictitious control signals in order to drive another set of states to a desired reference, the technique mimics backstepping [38]. Nevertheless, the proposal does not belong to any of the former approaches.

Remark 2. Notice that, in contrast with computed torque techniques, the gains $K_{p\eta}$, $K_{v\eta}$, $K_{p\xi}$, and $K_{v\xi}$ are not asked to be diagonal, which provides greater flexibility to the controller design. Moreover, these gains are determined via LMIs, which are solved in polynomial time by commercially available software. The LMI formulation has additional advantages as performance specifications such as decay rate, input/output constraints, or \mathcal{H}_∞ disturbance attenuation can be straightforwardly added [36].

4. Simulation Results

In this section, two examples illustrating our proposal are presented: the first one concerns a quadrotor performing trajectory tracking; the second one is an 8-rotor UAV. In both cases, a member of a family of exact nonlinear cascade control laws for trajectory tracking is used.

Example 1. Consider the quadrotor system (12), reproduced for convenience with a split of the state vector corresponding to (17), where the Cartesian coordinates are governed by

$$\eta \equiv \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{(\sin \phi \sin \psi + \sin \theta \cos \phi \cos \psi)u}{m} \\ \frac{(\sin \theta \sin \psi \cos \phi - \sin \phi \cos \psi)u}{m} \\ \frac{(\cos \theta \cos \phi)u}{m} - g \end{bmatrix}, \quad (28)$$

and angular coordinates are driven by

$$\xi \equiv \begin{bmatrix} \psi \\ \dot{\theta} \\ \phi \end{bmatrix} = \begin{bmatrix} \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\phi \end{bmatrix}, \quad (29)$$

where $u_1 = u$ and $u_2 = [\tilde{\tau}_\psi \ \tilde{\tau}_\theta \ \tilde{\tau}_\phi]^T$ in the notation of (17).

Focusing on the dynamic equation (28) governing the position in Cartesian coordinates, it is clear that, with the exception of u , angular variables ψ , θ , and ϕ can be seen as virtual inputs; their ideal values being those necessary to drive the Cartesian coordinates as desired for trajectory purposes; let us name these ideal signals as ψ_d , θ_d , and ϕ_d , which results in (28) being rewritten as the steady-state dynamic equations:

$$\begin{aligned} x &= \frac{(\sin \phi_d \sin \psi_d + \sin \theta_d \cos \phi_d \cos \psi_d)u}{m}, \\ y &= \frac{(\sin \theta_d \sin \psi_d \cos \phi_d - \sin \phi_d \cos \psi_d)u}{m}, \\ z &= \frac{(\cos \theta_d \cos \phi_d)u}{m} - g, \end{aligned} \quad (30)$$

where it is supposed that the angles are already in their sought behaviour.

Since the control objective is that the system tracks a reference in Cartesian coordinates, the error signals are defined as $e_x = x_d - x$, $e_y = y_d - y$, and $e_z = z_d - z$, where x_d , y_d , and z_d are the desired trajectories in x , y , and z coordinates, respectively. Thus, taking into account the dynamic equations above, the error system (18) is given by

$$\begin{aligned} e_x &= x_d - x = x_d - \frac{\sin \phi_d \sin \psi_d + \sin \theta_d \cos \phi_d \cos \psi_d}{m} u, \\ e_y &= y_d - y = y_d - \frac{\sin \theta_d \sin \psi_d \cos \phi_d - \sin \phi_d \cos \psi_d}{m} u, \\ e_z &= z_d - z = z_d - \frac{\cos \phi_d \cos \theta_d}{m} u + g. \end{aligned} \quad (31)$$

Following the methodology, equation (20) in Assumption 1 is expressed as

$$\begin{aligned} & \begin{bmatrix} x_d - \frac{\sin \phi_d \sin \psi_d + \sin \theta_d \cos \phi_d \cos \psi_d}{m} u \\ y_d - \frac{\sin \theta_d \sin \psi_d \cos \phi_d - \sin \phi_d \cos \psi_d}{m} u \\ z_d - \frac{\cos \phi_d \cos \theta_d}{m} u + g \end{bmatrix} \\ &= -K_{p\eta} \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} - K_{v\eta} \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \end{bmatrix}, \end{aligned} \quad (32)$$

where $K_{p\eta} \in \mathbb{R}^{3 \times 3}$ and $K_{v\eta} \in \mathbb{R}^{3 \times 3}$ are the constant gains to be found. If the above equations hold, the tracking $x \rightarrow x_d$, $y \rightarrow y_d$, and $z \rightarrow z_d$ will be performed; they are 3 equations with 4 variables, namely, u , ψ_d , θ_d , and ϕ_d . Due to their nonlinear trigonometric nature, there is an infinite number of solutions, each of them yielding a family of controllers that guarantee the tracking error goes to zero. In order to obtain a particular solution, we simplify the analysis considering that the gains $K_{p\eta}$ and $K_{v\eta}$ have a diagonal form, i.e.,

$$K_{p\eta} = \begin{bmatrix} K_{px} & 0 & 0 \\ 0 & K_{py} & 0 \\ 0 & 0 & K_{pz} \end{bmatrix}, \quad (33)$$

$$K_{v\eta} = \begin{bmatrix} K_{vx} & 0 & 0 \\ 0 & K_{vy} & 0 \\ 0 & 0 & K_{vz} \end{bmatrix}.$$

Thus, from the last equation in (32), it is clear that the choice

$$u = \frac{m}{\cos \phi_d \cos \theta_d} (z_d + g + K_{pz} e_z + K_{vz} \dot{e}_z), \quad (34)$$

will drive the error signal e_z to zero, i.e., $z \rightarrow z_d$. Substituting (34) in the first two equations in (32), we have

$$\frac{\tan \phi_d \sin \psi_d}{\cos \theta_d} + \tan \theta_d \cos \psi_d = \frac{x_d + K_{px} e_x + K_{vx} \dot{e}_x}{z_d + g + K_{pz} e_z + K_{vz} \dot{e}_z},$$

$$\sin \psi_d \tan \theta_d - \frac{\cos \psi_d \tan \phi_d}{\cos \theta_d} = \frac{y_d + K_{py} e_y + K_{vy} \dot{e}_y}{z_d + g + K_{pz} e_z + K_{vz} \dot{e}_z}. \quad (35)$$

There are many ways to find a set of solutions to the equations above. One choice is to solve for θ_d and ϕ_d , which will depend on ψ_d , the other variables in the expression being known. Nevertheless, regardless of the value of ψ_d , the equalities will be fulfilled; therefore, it is enough to drive ψ to $\psi_d = 0$. Now that the control signal u and the virtual inputs ψ_d , θ_d , and ϕ_d are known, time derivatives $\dot{\psi}$, $\dot{\theta}_d$, $\dot{\phi}_d$, $\dot{\psi}$, $\dot{\theta}_d$, and $\dot{\phi}_d$ can be calculated in order to use them in the computed torque control (21). The latter consists in designing $\tilde{\tau}$ such that the orientation angles go to their corresponding desired signals, that is, $\psi \rightarrow \psi_d$, $\theta \rightarrow \theta_d$, and $\phi \rightarrow \phi_d$ as $t \rightarrow \infty$. Thus, considering that gains $K_{p\xi}$ and $K_{v\xi}$ in (21) have the following structure,

$$K_{p\xi} = \begin{bmatrix} K_{p\psi} & 0 & 0 \\ 0 & K_{p\theta} & 0 \\ 0 & 0 & K_{p\phi} \end{bmatrix}, \quad (36)$$

$$K_{v\xi} = \begin{bmatrix} K_{v\psi} & 0 & 0 \\ 0 & K_{v\theta} & 0 \\ 0 & 0 & K_{v\phi} \end{bmatrix},$$

the control law (21) is given by

$$\begin{aligned} \tilde{\tau}_\psi &= \psi_d + K_{p\psi} e_\psi + K_{v\psi} \dot{e}_\psi, \\ \tilde{\tau}_\theta &= \theta_d + K_{p\theta} e_\theta + K_{v\theta} \dot{e}_\theta, \\ \tilde{\tau}_\phi &= \phi_d + K_{p\phi} e_\phi + K_{v\phi} \dot{e}_\phi, \end{aligned} \quad (37)$$

where $K_{p\psi}$, $K_{p\theta}$, $K_{p\phi}$, $K_{v\psi}$, $K_{v\theta}$, and $K_{v\phi}$ are the constant gains to be found.

The LMIs in Theorem 1 can now be programmed; they yield the following feasible solution:

$$X_1 = \begin{bmatrix} 0.2639 & 0 & 0 & -0.4342 & 0 & 0 \\ 0 & 0.2639 & 0 & 0 & -0.4342 & 0 \\ 0 & 0 & 0.4827 & 0 & 0 & -0.4972 \\ -0.4342 & 0 & 0 & 0.9186 & 0 & 0 \\ 0 & -0.4342 & 0 & 0 & 0.9186 & 0 \\ 0 & 0 & -0.4972 & 0 & 0 & 0.5507 \end{bmatrix},$$

$$X_2 = \begin{bmatrix} 0.2302 & 0 & 0 & -0.438 & 0 & 0 \\ 0 & 0.2302 & 0 & 0 & -0.438 & 0 \\ 0 & 0 & 0.2302 & 0 & 0 & -0.438 \\ -0.438 & 0 & 0 & 0.8885 & 0 & 0 \\ 0 & -0.438 & 0 & 0 & 0.8885 & 0 \\ 0 & 0 & -0.438 & 0 & 0 & 0.8885 \end{bmatrix},$$

$$K_{p\eta} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 30 \end{bmatrix},$$

$$K_{v\eta} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 28 \end{bmatrix},$$

$$K_{p\xi} = \begin{bmatrix} 80 & 0 & 0 \\ 0 & 80 & 0 \\ 0 & 0 & 80 \end{bmatrix},$$

$$K_{v\xi} = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 40 \end{bmatrix}. \quad (38)$$

Figure 2 shows the simulation results of the position and control signals where the quadrotor follows the desired trajectory $x_d = 1 + \cos(t)$, $y_d = -\sin(t)$, and $z_d = 1 + 0.2 \sin(2t)$. The initial conditions for simulation are $x(0) = 0$, $y(0) = -1$, $z(0) = 2$, $\dot{x}(0) = \dot{y}(0) = \dot{z}(0) = 0$, and $\psi(0) = \theta(0) = \phi(0) = \dot{\psi}(0) = \dot{\theta}(0) = \dot{\phi}(0) = 0$. High-frequency signals are customary during the transient of UAV control due to the very fast dynamics of these plants; this is the case of the control signals on the right.

Example 2. Consider the 8-rotor aircraft system whose mathematical model is given by

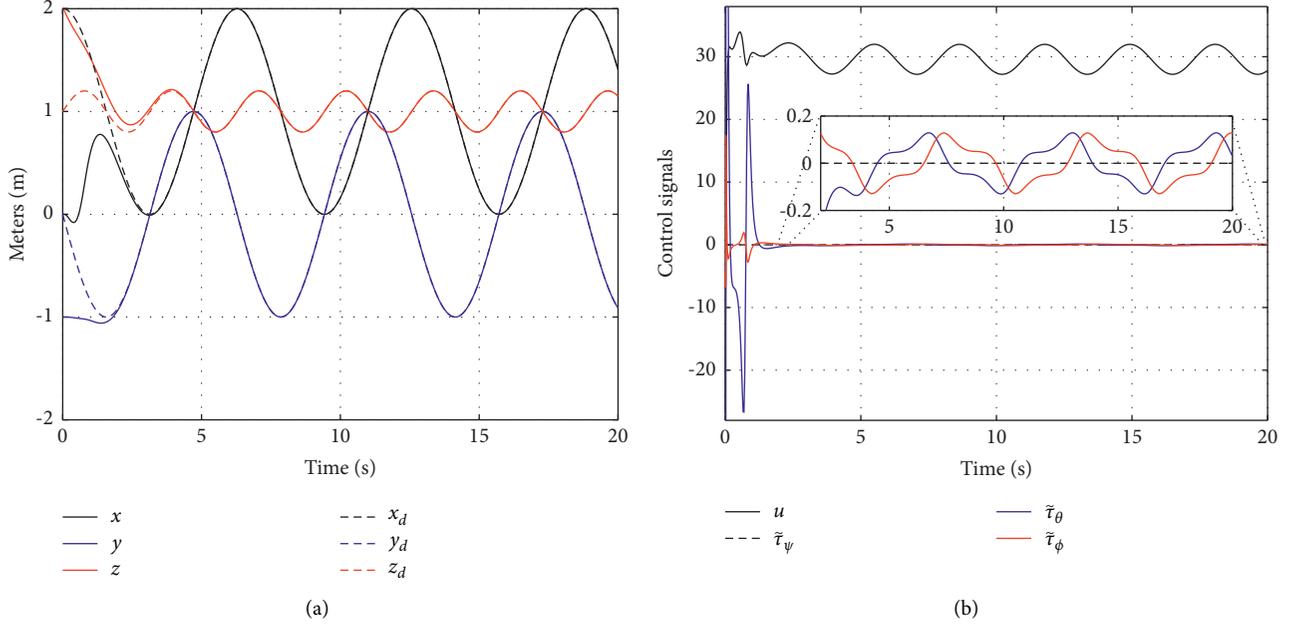


FIGURE 2: Quadrotor simulation results for trajectory tracking: position (a) and control (b).

$$\begin{aligned}
 mx &= u_x \cos \theta \cos \psi - u_y (\cos \phi \sin \psi - \cos \psi \sin \theta \sin \phi) + u_z (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta), \\
 my &= u_x \cos \theta \sin \psi + u_y (\cos \phi \cos \psi + \sin \psi \sin \theta \sin \phi) - u_z (\sin \phi \cos \psi - \cos \phi \sin \psi \sin \theta), \\
 mz &= -u_x \sin \theta + u_y \cos \theta \sin \phi + u_z \cos \theta \cos \phi - mg, \\
 \psi &= \tilde{\tau}_\psi, \\
 \dot{\theta} &= \tilde{\tau}_\theta, \\
 \phi &= \tilde{\tau}_\phi,
 \end{aligned} \tag{39}$$

where (x, y, z) represent the position of the aircraft in Cartesian coordinates, (ψ, θ, ϕ) are the orientation angles, m is the mass of the aircraft, and g is the gravitational constant. In contrast with the quadrotor, this model has six control inputs u_x , u_y , u_z , $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$, and $\tilde{\tau}_\phi$ (see [9] for modelling details).

Following the methodology shown in Section 3, the error signals of interest are defined as $e_x = x_d - x$, $e_y = y_d - y$, and $e_z = z_d - z$. Dynamic equations of the error system (18)

can be straightforwardly written (omitted for brevity). In contrast with the previous example where variables ϕ , θ , and ψ were considered as auxiliary/virtual control inputs, in the case of the 8-rotor aircraft, there are two more authentic control inputs; therefore, it is not necessary to take other signals as virtual control ones. In this case, considering the gains structure (33), Assumption 1 is given by the following equations, and it is fulfilled solving for $u_1 = [u_x \ u_y \ u_z]^T$ regardless of $\xi = [\psi \ \theta \ \phi]^T$,

$$\begin{aligned}
 e_x &= x_d - \frac{u_x}{m} \cos \theta \cos \psi + \frac{u_y}{m} (\cos \phi \sin \psi - \cos \psi \sin \theta \sin \phi) - \frac{u_z}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \\
 &= -K_{px} e_x - K_{vx} \dot{e}_x, \\
 e_y &= y_d - \frac{u_x}{m} \cos \theta \sin \psi - \frac{u_y}{m} (\cos \phi \cos \psi + \sin \psi \sin \theta \sin \phi) + \frac{u_z}{m} (\sin \phi \cos \psi - \cos \phi \sin \psi \sin \theta) \\
 &= -K_{py} e_y - K_{vy} \dot{e}_y, \\
 e_z &= z_d + \frac{u_x}{m} \sin \theta - \frac{u_y}{m} \cos \theta \sin \phi - \frac{u_z}{m} \cos \theta \cos \phi + g = -K_{pz} e_z - K_{vz} \dot{e}_z.
 \end{aligned} \tag{40}$$

If equalities above are satisfied, the tracking errors are guaranteed to go to zero as time tends to infinity; thus, the

control signals u_x , u_y , and u_z can be solved. By way of illustration, the shortest of them is provided, namely, $u_y = u_{ym}/u_{yd}$, where

$$\begin{aligned} u_{ym} = & -(2(y_d \cos \phi \cos \psi - x_d \cos \phi \sin \psi + z_d \cos \theta \sin \phi + x_d \cos \psi \sin \phi \sin \theta \\ & + y_d \sin \phi \sin \psi \sin \theta + K_{vy} \dot{e}_y \cos \phi \cos \psi + K_{py} e_y \cos \phi \cos \psi - K_{vx} \dot{e}_x \cos \phi \sin \psi \\ & - K_{px} e_x \cos \phi \sin \psi + K_{vz} \dot{e}_z \cos \theta \sin \phi + K_{pz} e_z \cos \theta \sin \phi + gm \cos \theta \sin \phi \\ & + K_{vx} \dot{e}_x \cos \psi \sin \phi \sin \theta + K_{px} e_x \cos \psi \sin \phi \sin \theta + K_{vy} \dot{e}_y \sin \phi \sin \psi \sin \theta \\ & + K_{py} e_y \sin \phi \sin \psi \sin \theta)), \\ u_{yd} = & 4\cos^2 \phi \cos^2 \psi - 4\cos^2 \phi - 4\cos^2 \psi - 2\cos^2 \theta - \sin(2\theta) + 2\cos^2 \phi \cos^2 \theta + 4\cos^2 \psi \cos^2 \theta \\ & + 2\cos^2 \phi \cos \theta \sin \theta - 4\cos^2 \phi \cos^2 \psi \cos^2 \theta + 4 \cos \phi \cos \psi \sin \phi \sin \psi \sin \theta + 2. \end{aligned} \quad (41)$$

Once the control signals u_x , u_y , and u_z have been found, the remaining ones $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$, and $\tilde{\tau}_\phi$ are proposed with a simple PD form, such that $\psi \rightarrow 0$, $\theta \rightarrow 0$, and $\phi \rightarrow 0$ as $t \rightarrow \infty$. Thus,

$$\begin{aligned} \tilde{\tau}_\psi &= -K_{p\psi} \psi - K_{v\psi} \dot{\psi}, \\ \tilde{\tau}_\theta &= -K_{p\theta} \theta - K_{v\theta} \dot{\theta}, \\ \tilde{\tau}_\phi &= -K_{p\phi} \phi - K_{v\phi} \dot{\phi}. \end{aligned} \quad (42)$$

Figure 3 shows the simulation results for initial conditions $x(0) = 0$, $y(0) = -1$, $z(0) = 2$, $\dot{x}(0) = \dot{y}(0) = \dot{z}(0) = 0$, and $\psi(0) = \theta(0) = \phi(0) = \dot{\psi}(0) = \dot{\theta}(0) = \dot{\phi}(0) = 0$. It can be seen in 3(a) that the position of the aircraft follows the desired trajectory $x_d = 1 + \cos(t)$, $y_d = -\sin(t)$, and $z_d = 1 + 0.2 \sin(t)$. The gains used are $K_{px} = K_{py} = K_{pz} = 20$, $K_{vx} = K_{vy} = K_{vz} = 10$, $K_{p\psi} = K_{p\theta} = K_{p\phi} = 80$, and $K_{v\psi} = K_{v\theta} = K_{v\phi} = 40$. Control signals are shown 3(b). Since in this case, it was not necessary to take signals as virtual control inputs, the initial conditions for the angles are

zero, and the control signals $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$, and $\tilde{\tau}_\phi$ are zero during simulation.

The previous solution does not exploit the cascade structure of the system; in this sense, it is a 0-level cascade one. A 1-level one is, nevertheless, possible; such control law does not only serve artificial purposes but also can help standing against actuator fails. To illustrate this alternative, the design can take up from the error equations in (40); any variable in these expressions can be considered as a control signal, which results in a cascade structure and in turn represents a family of solutions. Indeed, considering the control input u_z as

$$u_z = \frac{mg + m\theta + u_x \sin \theta - u_y \sin \phi \sin \theta}{\cos \phi \cos \theta}, \quad (43)$$

the dynamics of the z position can be rewritten as $z = \theta$. Thus, θ can be seen as a fictitious control input, thus resulting in a cascade control. Renaming θ as θ_d and substituting (43), Assumption 1 yields

$$\begin{aligned} x_d - \frac{u_x}{m} \cos \theta_d \cos \psi + \frac{u_y}{m} (\cos \phi \sin \psi - \cos \psi \sin \theta_d \sin \phi) \\ - \frac{(gm + m\theta_d + u_x \sin \theta_d - u_y \sin \phi \sin \theta_d)(\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta_d)}{m \cos \phi \cos \theta_d} &= -K_{px} e_x - K_{vx} \dot{e}_x, \\ y_d - \frac{u_x}{m} \cos \theta_d \sin \psi - \frac{u_y}{m} (\cos \phi \cos \psi + \sin \psi \sin \theta_d \sin \phi) \\ + \frac{(gm + m\theta_d + u_x \sin \theta_d - u_y \sin \phi \sin \theta_d)(\sin \phi \cos \psi - \cos \phi \sin \psi \sin \theta_d)}{m \cos \phi \cos \theta_d} &= -K_{py} e_y - K_{vy} \dot{e}_y, \\ z_d - \theta_d &= -K_{pz} e_z - K_{vz} \dot{e}_z, \end{aligned} \quad (44)$$

which can be solved for u_x , u_y , and θ_d , such that Assumption 1 holds.

Once the control signals u_x , u_y , u_z , and the virtual input θ_d are determined, the time derivatives $\dot{\theta}_d$ and $\ddot{\theta}_d$ can be calculated to be used in the computed torque control (21),

which consists in designing $\tilde{\tau}$, such that the orientation angles go to their corresponding desired signals, i.e., $\theta \rightarrow \theta_d$ as $t \rightarrow \infty$. Thus, considering that the gains $K_{p\xi}$ and $K_{v\xi}$ have the structure (36), the control law (21) is given by

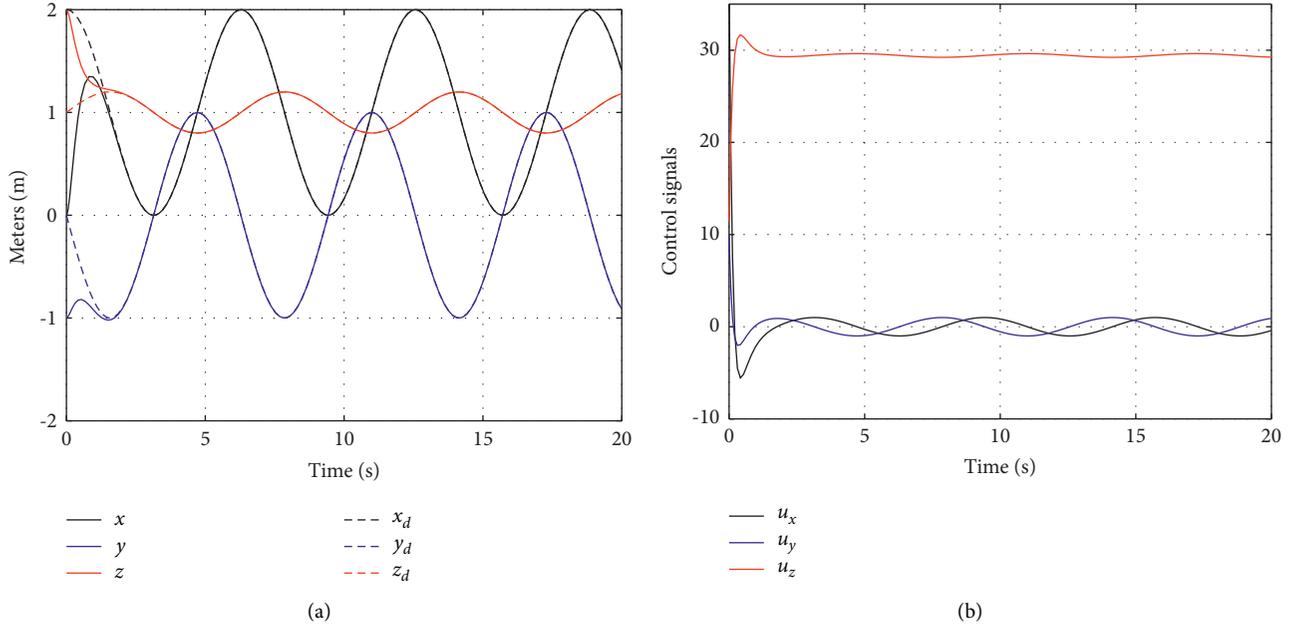


FIGURE 3: Octarotor simulation results for trajectory tracking: position (a) and control (b).

$$\begin{aligned}
 \bar{\tau}_\psi &= \psi_d + K_{p\psi}e_\psi + K_{v\psi}\dot{e}_\psi, \\
 \bar{\tau}_\theta &= \ddot{\theta}_d + K_{p\theta}e_\theta + K_{v\theta}\dot{e}_\theta, \\
 \bar{\tau}_\phi &= \phi_d + K_{p\phi}e_\phi + K_{v\phi}\dot{e}_\phi,
 \end{aligned} \tag{45}$$

where $K_{p\psi}$, $K_{p\theta}$, $K_{p\phi}$, $K_{v\psi}$, $K_{v\theta}$, and $K_{v\phi}$ are the constant gains to be found.

LMIs in Theorem 1 have been found feasible with the following solution:

$$\begin{aligned}
 X_1 &= \begin{bmatrix} 0.2126 & 0 & 0 & -0.3695 & 0 & 0 \\ 0 & 0.3192 & 0 & 0 & -0.5168 & 0 \\ 0 & 0 & 0.3192 & 0 & 0 & -0.5168 \\ -0.3695 & 0 & 0 & 1.2294 & 0 & 0 \\ 0 & -0.5168 & 0 & 0 & 1.0959 & 0 \\ 0 & 0 & -0.5168 & 0 & 0 & 1.0959 \end{bmatrix}, \\
 X_2 &= \begin{bmatrix} 0.2609 & 0 & 0 & -0.4959 & 0 & 0 \\ 0 & 0.2609 & 0 & 0 & -0.4959 & 0 \\ 0 & 0 & 0.2609 & 0 & 0 & -0.4959 \\ -0.4959 & 0 & 0 & 1.0065 & 0 & 0 \\ 0 & -0.4959 & 0 & 0 & 1.0065 & 0 \\ 0 & 0 & -0.4959 & 0 & 0 & 1.0065 \end{bmatrix}, \\
 K_{p\eta} &= \begin{bmatrix} 15 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}, \\
 K_{v\eta} &= \begin{bmatrix} 5 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \\
 K_{p\xi} &= \begin{bmatrix} 80 & 0 & 0 \\ 0 & 80 & 0 \\ 0 & 0 & 80 \end{bmatrix}, \\
 K_{v\xi} &= \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 40 \end{bmatrix}.
 \end{aligned} \tag{46}$$

Figure 4(a) shows the simulation results of the position where the aircraft follows the desired trajectory $x_d = 1 + \cos(t)$, $y_d = 0.5 - \sin(t)$, and $z_d = 1 + 0.2 \sin(t) + 0.2 \cos$

$(2t)$. Initial conditions are $x(0) = 0$, $y(0) = -1$, $z(0) = 2$, $\dot{x}(0) = \dot{y}(0) = \dot{z}(0) = 0$, and $\psi(0) = \theta(0) = \phi(0) = \dot{\psi}(0) = \dot{\theta}(0) = \dot{\phi}(0) = 0$. Figure 4(b) shows the control signal.

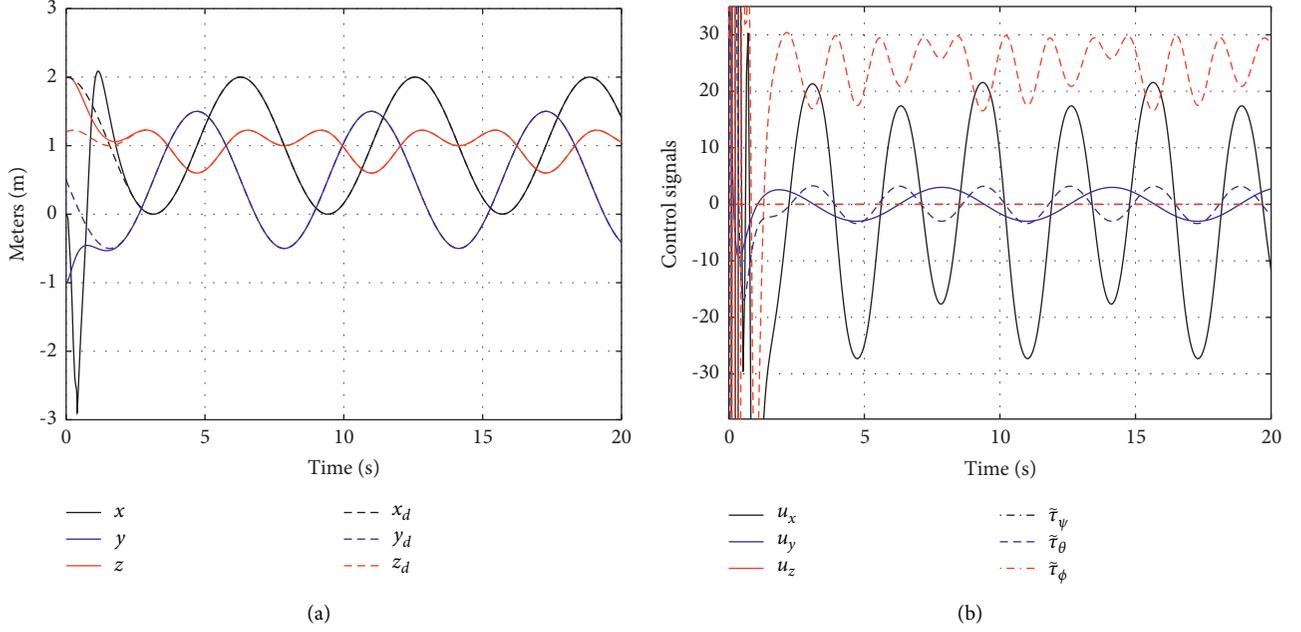


FIGURE 4: Octarotor simulation results for trajectory tracking: position (a) and control (b).

5. Real-Time Results

To corroborate the effectiveness of the proposed results in the previous section, its implementation on an experimental platform is presented; such implementation has been made in the National Laboratory of Autonomous Vehicles and Exoskeletons from the Autonomous University of Hidalgo State. The system used for the experiments is from the DJI Matrice 100 quadrotor (Figure 5). Some details about the performance of this platform are given in Table 1, while the specifications of the structure are given in Table 2 [41].

The mathematical model of the experimental platform above is (10). To implement the controller (11), (34), and (37), first a simulation was conducted using the DJI flight simulator provided by the manufacturer, later to program on the experimental platform Android Studio and DJI Mobile SDK (software development kit).

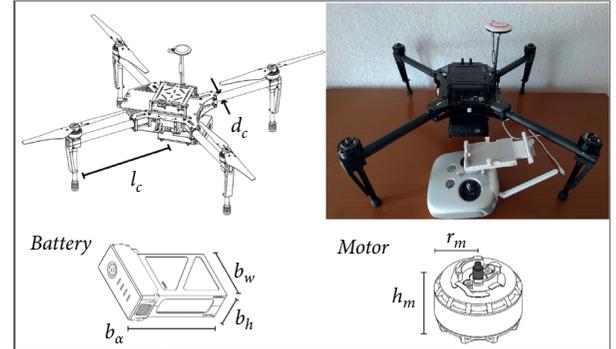


FIGURE 5: DJI Matrice 100 quadrotor.

Using the data given in the previous tables, the values of the Coriolis C and inertia J matrices are

$$C(\xi, \dot{\xi}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \quad (47)$$

$$J = \begin{bmatrix} I_{xx} & 0 & -I_{xx} \sin(\theta) \\ 0 & I_{yy} \cos^2(\phi) + I_{zz} \sin^2(\phi) & Mn_1 \\ -I_{xx} \sin(\theta) & Mn_2 & Mn_3 \end{bmatrix},$$

where $I_{xx} = 0.023442$, $I_{yy} = 0.025871$, and $I_{zz} = 0.0264413$, $c_{11} = 0$,

TABLE 1: Performance of experimental platform.

Hovering accuracy (<i>P</i> -mode with GPS)	Vertical: 0.5 m, horizontal: 2.5 m
Max. angular velocity	Pitch: 300°/s, yaw: 150°/s
Max. tilt angle	35°
Max. speed of ascent	5 m/s
Max. speed of descent	4 m/s
Max. wind resistance	10 m/s
Max. speed	22 m/s (ATTI mode, no payload, no wind) 17 m/s (GPS mode, no payload, no wind)
Hovering time (with TB47D battery)	No payload: 22 min; 500 g Payload: 17 min; 1 kg payload: 13 min

TABLE 2: Structure specifications of experimental platform.

Length of the frame arm (<i>l</i>)	0.65 m
Distance to center of gravity (<i>l_c</i>)	<i>l</i> /2
Mass of the frame arm (<i>m_{bc}</i>)	0.05 kg
Radius of the frame arm (<i>r_{bc} = d_c/2</i>)	0.01 m
Mass of the motor (<i>m_m</i>)	0.106 kg
Radius of the motor (<i>r_m</i>)	0.025 m
Height of the motor (<i>h_m</i>)	0.03 m
Mass of the battery (<i>m_b</i>)	0.6 kg
Width of the battery (<i>b_w</i>)	0.08 m
Height of the battery (<i>b_h</i>)	0.04 m
Length of the battery (<i>b_a</i>)	0.135 m
Net weight (<i>m</i>)	2.335 kg

$$\begin{aligned}
c_{12} &= -I_{xx}\dot{\psi}\cos(\theta) + I_{yy}\left(\dot{\theta}\sin(\phi)\cos(\phi) + \dot{\psi}\cos(\theta)\sin^2(\phi) - \dot{\psi}\cos(\theta)\cos^2(\phi)\right) \\
&\quad - I_{zz}\left(\dot{\psi}\cos(\theta)\sin^2(\phi) - \dot{\psi}\cos(\theta)\cos^2(\phi) + \dot{\theta}\sin\phi\cos\phi\right), \\
c_{13} &= I_{yy}\dot{\psi}\cos^2(\theta)\sin(\phi)\cos(\phi) - I_{zz}\dot{\psi}\cos^2(\theta)\sin(\phi)\cos(\phi), \\
c_{21} &= I_{xx}\dot{\psi}\cos(\theta) + I_{yy}\left(-\dot{\theta}\sin(\phi)\cos(\phi) + \dot{\psi}\cos(\theta)\cos^2(\phi) - \dot{\psi}\cos(\theta)\sin^2(\phi)\right) \\
&\quad + I_{zz}\left(\dot{\psi}\cos(\theta)\sin^2(\phi) - \dot{\psi}\cos(\theta)\cos^2(\phi) + \dot{\theta}\sin(\phi)\cos(\phi)\right), \\
c_{22} &= -I_{yy}\dot{\phi}\sin(\phi)\cos(\phi) + I_{zz}\dot{\phi}\sin(\phi)\cos(\phi), \\
c_{23} &= -I_{xx}\dot{\psi}\sin(\theta)\cos(\theta) + I_{yy}\dot{\psi}\sin(\theta)\cos(\theta)\sin^2(\phi) + I_{zz}\dot{\psi}\sin(\theta)\cos(\theta)\cos^2(\phi), \\
c_{31} &= -I_{xx}\dot{\theta}\cos(\theta) + I_{yy}\dot{\psi}\cos^2(\theta)\sin(\phi)\cos(\phi) - I_{zz}\dot{\psi}\cos^2(\theta)\sin(\phi)\cos(\phi), \\
c_{32} &= I_{xx}\dot{\phi}\sin(\theta)\cos(\theta) - I_{yy}\left(\dot{\theta}\sin(\theta)\sin(\phi)\cos(\phi) + \dot{\phi}\cos(\theta)\sin^2(\phi) - \dot{\phi}\cos(\theta)\cos^2(\phi)\right) \\
&\quad + \dot{\psi}\sin(\theta)\cos(\theta)\sin^2(\phi) + I_{zz}\left(\dot{\phi}\cos(\theta)\sin^2(\phi) - \dot{\phi}\cos(\theta)\cos^2(\phi)\right) \\
&\quad - \dot{\psi}\sin(\theta)\cos(\theta)\cos^2(\phi) + \dot{\theta}\sin(\theta)\sin(\phi)\cos(\phi), \\
c_{33} &= I_{xx}\dot{\theta}\sin(\theta)\cos(\theta) + I_{yy}\left(-\dot{\theta}\sin(\theta)\cos(\theta)\sin^2(\phi) + \dot{\phi}\cos^2(\theta)\sin(\phi)\cos(\phi)\right) \\
&\quad - I_{zz}\left(\dot{\theta}\sin(\theta)\cos(\theta)\cos^2(\phi) + \dot{\phi}\cos^2(\theta)\sin(\phi)\cos(\phi)\right), \\
Mn_1 &= \cos(\theta)\cos(\phi)\sin(\phi)(I_{yy} - I_{zz}), \\
Mn_2 &= \cos(\theta)\cos(\phi)\sin(\phi)(I_{yy} - I_{zz}), \\
Mn_3 &= I_{xx}\sin^2(\theta) + I_{yy}\cos^2(\theta)\sin^2(\phi) + I_{zz}\cos^2(\theta)\cos^2(\phi).
\end{aligned} \tag{48}$$

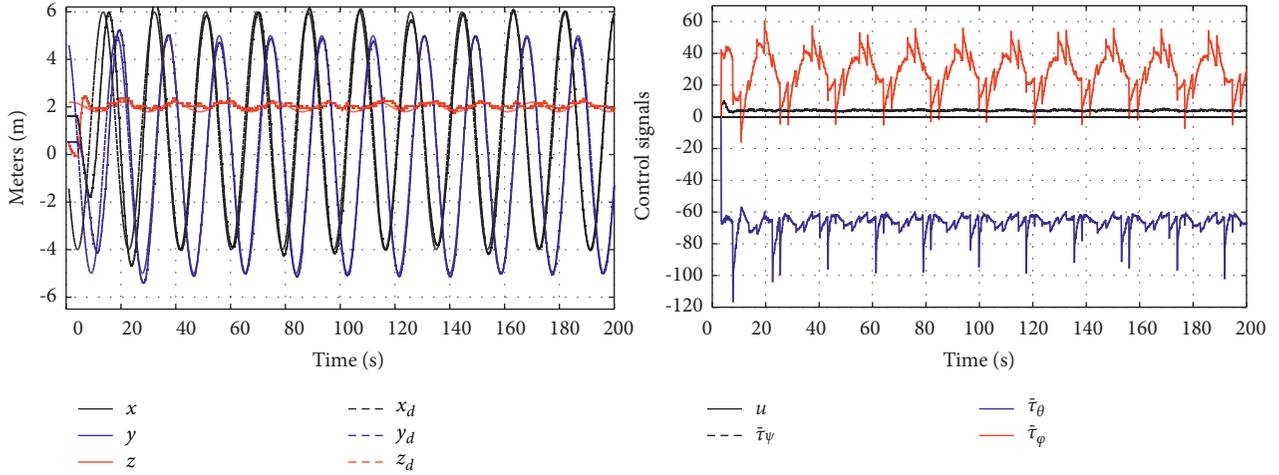


FIGURE 6: Quadrotor simulation results for trajectory tracking (2D).

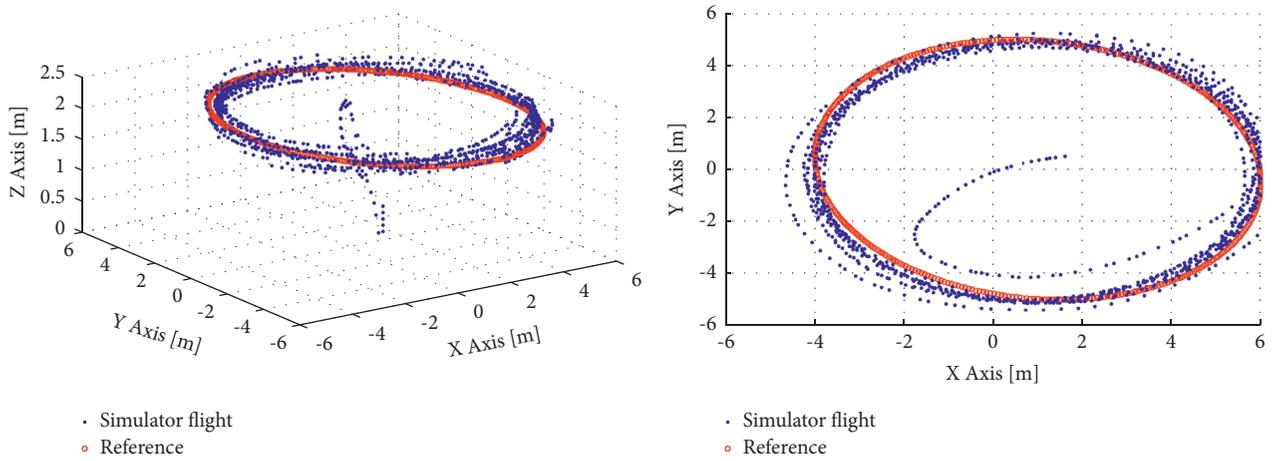


FIGURE 7: Quadrotor simulation results for trajectory tracking (3D).

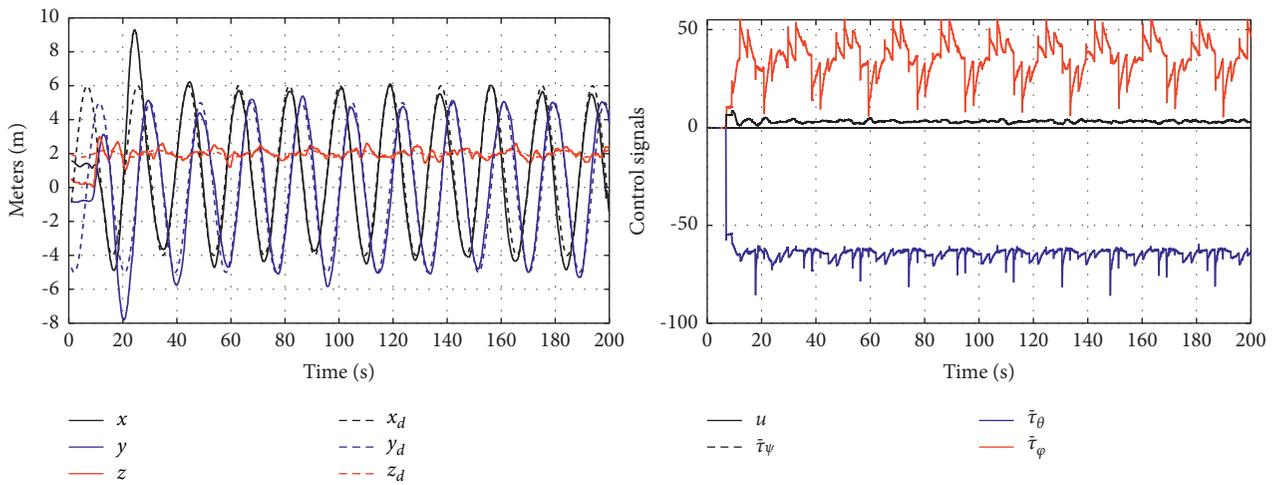


FIGURE 8: Quadrotor real-time results for trajectory tracking (2D).

Here, the desired trajectory of the quadrotor is $x_d = 1 + 5 \cos(t)$, $y_d = 5 \sin(t)$, and $z_d = 2 + 0.2 \sin(t)$. The initial conditions are $x(0) = y(0) = 0$, $z(0) = 1$,

$\dot{x}(0) = \dot{y}(0) = \dot{z}(0) = 0$, and $\psi(0) = \theta(0) = \phi(0) = \dot{\psi}(0) = \dot{\theta}(0) = \dot{\phi}(0) = 0$. For the input signal (24), $K_{px} = K_{py} = 50$, $K_{pz} = 60$, and $K_{vx} = K_{vy} = 10.5$ and $K_{vy} = 28$ are proposed.

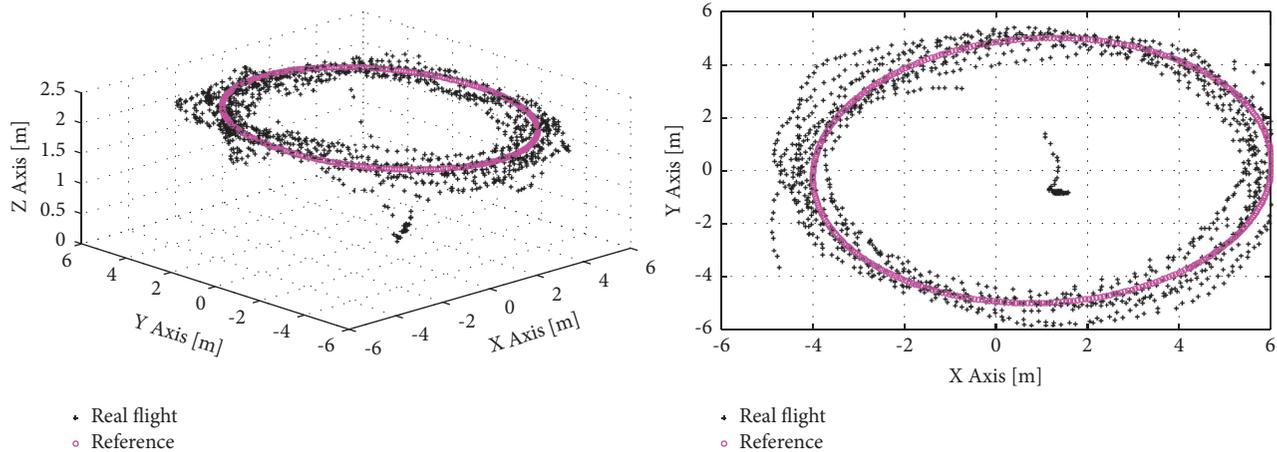


FIGURE 9: Quadrotor real-time results for trajectory tracking (3D).

While, for the equation (26), the gains used are $K_{p\theta} = K_{p\phi} = K_{p\psi} = 80$ and $K_{v\theta} = K_{v\phi} = K_{v\psi} = 40$.

Figures 6 and 7 show the performance of the DJI Matrice 100 quadrotor using the flight simulator provided by the manufacturer. The video with the simulation results can be seen at the following link: <https://youtu.be/Igoobvl61AE>.

Figures 8 and 9 show the performance of the DJI Matrice 100 quadrotor in real-time. The video where these results are shown can be consulted in the following link: <https://youtu.be/oGN5N7M3UZw>.

6. Conclusions

A family of exact nonlinear cascade control laws for trajectory tracking of unmanned aerial vehicles has been presented. It has been shown that the control scheme can be applied to UAVs because it takes advantage of their underactuated characteristics, which produce a cascade model suitable for Lyapunov-based design. The proposal has been proven to produce an infinite number of solutions which guarantee that the position in Cartesian coordinates of the aircraft follows a desired trajectory. The methodology has been illustrated via numerical simulations in 4- and 8-rotor systems. Real-time results have been provided on the quadrotor, which confirm the applicability of the presented approach. Future work is on course for including robustness and more general classes of underactuated systems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research has been supported by the CONACYT scholarship (731421), the Laboratory LANAVEX from the Universidad Politécnica de Pachuca, and the project ITSON-PROFAPI 2021-0027.

References

- [1] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military uav relay networks," *IEEE Communications Letters*, vol. 20, no. 4, pp. 804–807, 2016.
- [2] F. Nex and F. Remondino, "Uav for 3d mapping applications: a review," *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.
- [3] C. Qian, H. Yang, T. Ren, Z. Li, and F. Teng, "Real-time aerial photography and global positioning system for a quadrotor design," *Journal of Jilin University (Information Science Edition)*, vol. 4, 2015.
- [4] I. Sa and P. Corke, "Vertical infrastructure inspection using a quadcopter and shared autonomy control," in *Field and Service Robotics*, pp. 219–232, Springer, New York, NY, USA, 2012.
- [5] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: hovering capture and load stability," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 2491–2498, IEEE, Shanghai, China, May 2011.
- [6] J. J. Potter, C. J. Adams, and W. Singhose, "A planar experimental remote-controlled helicopter with a suspended load," *IEEE/ASME transactions on mechatronics*, vol. 20, no. 5, pp. 2496–2503, 2015.
- [7] M. Arjomandi, S. Agostino, M. Mammone, M. Nelson, and T. Zhou, *Classification of Unmanned Aerial Vehicles. Report for Mechanical Engineering Class*, University of Adelaide, Adelaide, Australia, 2006.
- [8] H. Romero-Trejo, *Modélisation et asservissement visuel d'un mini hélicoptère*, Ph.D. thesis, University of Technology of Compiègne, Compiègne, France, 2008.
- [9] S. Salazar, H. Romero, R. Lozano, and P. Castillo, "Modeling and real-time stabilization of an aircraft having eight rotors," in *Unmanned Aircraft Systems*, pp. 455–470, Springer, New York, NY, USA, 2008.
- [10] B. Erginer and E. Altug, "Modeling and pd control of a quadrotor VTOL vehicle," in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, pp. 894–899, IEEE, Istanbul, Turkey, June 2007.
- [11] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 562–571, 2006.

- [12] K. F. Maya-Gress, R. Villafuerte-Segura, H. Romero-Trejo, and M. Á. Bernal-Reza, "Puesta en operación y modelado de un cuatrorotor Matrice 100," *de DJI. Pádi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 9, no. 17, pp. 67–75, 2021.
- [13] C. Coza and C. J. Macnab, "A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization," in *Proceedings of the NAFIPS 2006-2006 Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 454–458, IEEE, Montreal, Canada, June 2006.
- [14] B. Erginer and E. Altuğ, "Design and implementation of a hybrid fuzzy logic controller for a quadrotor VTOL vehicle," *International Journal of Control, Automation and Systems*, vol. 10, no. 1, pp. 61–70, 2012.
- [15] M. R. Mehranpour, A. M. Shahri, O. Emamgholi, and M. Farrokhi, "A new fuzzy adaptive control for a quadrotor flying robot," in *Proceedings of the 2013 13th Iranian Conference on Fuzzy Systems (IFSC)*, pp. 1–5, IEEE, Qazvin, Iran, August 2013.
- [16] H. Gao, C. Liu, D. Guo, and J. Liu, "Fuzzy adaptive pd control for quadrotor helicopter," in *Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 281–286, IEEE, Shenyang, China, June 2015.
- [17] G. Dong, L. Cao, D. Yao, H. Li, and R. Lu, "Adaptive attitude control for multi-muav systems with output dead-zone and actuator fault," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, pp. 1567–1575, 2020.
- [18] A. Almeshal and M. Alenezi, "A vision-based neural network controller for the autonomous landing of a quadrotor on moving targets," *Robotics*, vol. 7, no. 4, p. 71, 2018.
- [19] H. Razmi and S. Afshinfar, "Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor uav," *Aerospace Science and Technology*, vol. 91, pp. 12–27, 2019.
- [20] E. Okyere, A. Bousbaine, G. T. Poyi, A. K. Joseph, and J. M. Andrade, "LQR controller design for quad-rotor helicopters," *Journal of Engineering*, vol. 2019, no. 17, pp. 4003–4007, 2019.
- [21] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard, "LQR control for a quadrotor using unit quaternions: modeling and simulation," in *Proceedings of the CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*, pp. 172–178, IEEE, Cholula, Puebla, Mexico, March 2013.
- [22] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3255–3260, IEEE, Beijing, China, October 2006.
- [23] A. Das, F. Lewis, and K. Subbarao, "Backstepping approach for controlling a quadrotor using Lagrange form dynamics," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1-2, pp. 127–151, 2009.
- [24] F. Chen, W. Lei, K. Zhang, G. Tao, and B. Jiang, "A novel nonlinear resilient control for a quadrotor uav via backstepping control and nonlinear disturbance observer," *Nonlinear Dynamics*, vol. 85, no. 2, pp. 1281–1295, 2016.
- [25] L. Luque-Vega, B. Castillo-Toledo, and A. G. Loukianov, "Robust block second order sliding mode control for a quadrotor," *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 719–739, 2012.
- [26] K. Runcharoon and V. Srichatrapimuk, "Sliding mode control of quadrotor," in *Proceedings of the 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE)*, pp. 552–557, IEEE, Konya, Turkey, May 2013.
- [27] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Transactions*, vol. 53, no. 4, pp. 1350–1356, 2014.
- [28] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.
- [29] N. Wang, Q. Deng, G. Xie, and X. Pan, "Hybrid finite-time trajectory tracking control of a quadrotor," *ISA Transactions*, vol. 90, pp. 278–286, 2019.
- [30] B. Tian, H. Lu, Z. Zuo, Q. Zong, and Y. Zhang, "Multivariable finite-time output feedback trajectory tracking control of quadrotor helicopters," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 1, pp. 281–295, 2018.
- [31] L. Sun and Z. Zuo, "Nonlinear adaptive trajectory tracking control for a quad-rotor with parametric uncertainty," *Proceedings of the Institution of Mechanical Engineers-Part G: Journal of Aerospace Engineering*, vol. 229, no. 9, pp. 1709–1721, 2015.
- [32] R. Perez-Alcocer and J. Moreno-Valenzuela, "Adaptive control for quadrotor trajectory tracking with accurate parametrization," *IEEE Access*, vol. 7, pp. 53236–53247, 2019.
- [33] W. Wu, X. Jin, and Y. Tang, "Vision-based trajectory tracking control of quadrotors using super twisting sliding mode control," *Cyber-Physical Systems*, vol. 6, no. 4, pp. 207–230, 2020.
- [34] Y. Zhang, Z. Chen, and M. Sun, "Trajectory tracking control for a quadrotor unmanned aerial vehicle based on dynamic surface active disturbance rejection control," *Transactions of the Institute of Measurement and Control*, vol. 42, no. 12, pp. 2198–2205, 2020.
- [35] M. A. Mohd Basri, "Trajectory tracking control of autonomous quadrotor helicopter using robust neural adaptive backstepping approach," *Journal of Aerospace Engineering*, vol. 31, no. 2, Article ID 04017091, 2018.
- [36] S. Boyd, L. E. Ghaoui, E. Feron, and V. Belakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Vol. 15, SIAM: Studies In Applied Mathematics, Philadelphia, PA, USA, 1994.
- [37] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox*, Math Works, Natick, MA, USA, 1995.
- [38] H. K. Khalil, *Nonlinear Control*, Pearson Higher Ed, New York, NY, USA, 2014.
- [39] F. Lewis, D. Dawson, and C. Abdallah, *Robot Manipulator Control: Theory and Practice*, CRC Press, Boca Raton, FL, USA, 2003.
- [40] A. Isidori, *Nonlinear Control Systems*, Springer, London, UK, 3rd edition, 1995.
- [41] DJI, *Description of the Dji Matrice 100 Quadcopter*, DJI, Nanshan, Shenzhen, China, 2019, <https://www.dji.com/mx/matrice100?site=brandsite&from=nav>.