

Article

Robust Backstepping Control Applied to UAVs for Pest Recognition in Maize Crops

Liliam Rodríguez-Guerrero [†], Alejandro Benítez-Morales ^{*,†}, Omar-Jacobo Santos-Sánchez ^{*,†}, Orlando García-Pérez [†], Hugo Romero-Trejo [†], Mario-Oscar Ordaz-Oliver [†] and Jesús-Patricio Ordaz-Oliver 

Electronic and Control Academic Group, Academic Area of Computation and Electronic, Autonomous University of Hidalgo State (UAEH), Pachuca de Soto 42084, Mexico

* Correspondence: abenitez@uaeh.edu.mx (A.B.-M.); omarj@uaeh.edu.mx (O.-J.S.-S.);

Tel.: +52-771-71-72000 (ext. 6734) (A.B.-M. & O.-J.S.-S.)

† These authors contributed equally to this work.

Abstract: In this paper, a robust control technique is developed to achieve the quadrotor stabilization against unmodeled matching vanishing dynamics. The synthesis of the proposed robust control is based on the Lyapunov approach and the backstepping method allowing to construct an iterative control algorithm. To compare the performance of the proposed controller, a Proportional Derivative (PD) controller is used to obtain experimental results in an outdoor environment. To compare the closed-loop system responses with both controllers, the Integral Absolute Error is computed and several tests are conducted to calculate the error standard deviation. Ultimately, employing the robust backstepping control approach in pest recognition in maize crops, which is a specific task of precision agriculture, demonstrates its effectiveness in improving the trajectory tracking of the vehicle while it captures images of the crops.

Keywords: outdoor environment; quadrotor; robust backstepping control; vanishing unstructured dynamics; precision agriculture



Citation: Rodríguez-Guerrero, L.; Benítez-Morales, A.; Santos-Sánchez, O.-J.; García-Pérez, O.; Romero-Trejo, H.; Ordaz-Oliver, M.-O.; Ordaz-Oliver, J.-P. Robust Backstepping Control Applied to UAVs for Pest Recognition in Maize Crops. *Appl. Sci.* **2022**, *12*, 9075. <https://doi.org/10.3390/app12189075>

Academic Editor: Yosoon Choi

Received: 6 August 2022

Accepted: 7 September 2022

Published: 9 September 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, the use of Unmanned Aerial Vehicles (UAVs) is growing, and multirotor systems are expected to replace manned aircraft in a variety of activities [1]. Examples of such activities include search and rescue missions, data collection, precision agriculture, and payload delivery [2–5]. Quadcopters can rapidly change direction and perform vertical takeoffs and landings, thus making them increasingly popular.

Small-scale UAVs typically suffer from the effects of outside wind gusts and unmodeled dynamics, effects that involve changes in mass and inertial moments causing parametric uncertainties. These effects also result from the coupling effect between the forces and moments produced by their actuators, which are commonly neglected in the practical designs [6]. Since an advanced control system is based on the nominal model, the uncertain parameters and unmodeled dynamics adversely affect its performance. Developing robust controllers to guarantee UAV stabilization has been a key topic that some authors have handled in different ways. For instance, on the basis of sliding mode algorithms and backstepping methods, [7] offers a robust backstepping sliding mode controller for attitude and position control; accordingly, smooth bounded disturbances and non-vanishing perturbations were considered for a quadrotor vehicle, and simulation results were presented. In [8], the attitude control of an eight-rotor UAV in the presence of model uncertainties and external disturbances was suggested; hence, a robust backstepping sliding mode controller with adaptive radial basis function neural network was employed, and simulation results were reported. According to [9], the adaptive backstepping sliding mode tracking control method for underactuated unmanned surface vehicle was employed to compensate for its

model uncertainty, time-varying disturbances and input saturation, and a simulation study was introduced. Attitude and position tracking were performed using the backstepping technique and sliding mode control in [10]; in this case, the controller adapts to the mass changes to control the UAV, and the controller performance was evaluated through simulations. Furthermore, attitude regulation and translational movement of an aircraft using a backstepping approach was done in [11], and altitude control was performed with sliding mode control considering uncertainties. As a result of the presence of sinusoidal functions in the roll and pitch subsystems, the virtual input is always bounded, allowing the control input produced with backstepping to be smoother than the input provided by the sliding mode controller, exhibiting no implementation issues such as chattering; however, the controller robustness problem was not addressed. In [12], a backstepping control strategy was employed to decrease the wind disturbances that can impact the accuracy of image acquisition in the course of drone flight in an outdoor environment; in particular, the specific task involved estimating nitrogen in a rice crop using the aerial imagery.

As a result of adaptive backstepping control, several estimations of the non-vanishing external disturbance were employed for the proposed controller in [13]; in this case, the experimental results were obtained in an indoor environment using a VICON Bonita motion capture system and MATLAB software. Similarly, in [14], the same motion capture system was used to implement feedback linearization and an integral-backstepping-like controller to address the perturbation problems that appear in a quadrotor; in this sense, simulations and indoor environment experimental results were presented with slowly-varying wind conditions. To regulate quadrotors in the presence of constant and time-varying disturbances, a nonlinear controller based on backstepping was designed, and simulation results were presented in [15]; the non-vanishing disturbances were estimated by a nonlinear observer. A nonlinear robust and adaptive backstepping control strategy was proposed in [16] to solve the trajectory tracking problem of hexacopter UAVs. The nominal backstepping control approach was designed as the main controller, and simulations were performed considering non-vanishing bounded disturbances. In [17], the robust position and attitude tracking control problem of a quadrotor subject to nonlinearities, input coupling, aerodynamic uncertainties and external wind disturbances was presented, the control scheme was validated through simulations and experimental validation on a Quanser's 3-DOF Hover setup. In [18], a robust landing algorithm onto a heaving platform, using an autonomous quadcopter DJI-F450, was presented. This algorithm addressed the altitude flight under the ground effect and external disturbances. In [19], a nonlinear robust Fast Terminal Sliding Mode Controller was designed to control and to stabilize a reconfigurable UAV in the presence of uncertain and variable parameters. The proposed controller was evaluated through a flight scenario. In [20], a hybrid control architecture that combines Deep Reinforcement Learning and Robust Linear Quadratic Regulator for vision-based lateral control of an autonomous vehicle was presented. The proposal was validated via simulation results. The robust control Lyapunov function approach is another control approach that has been used to tackle the rejecting bounded-matched disturbance problems in a Planar Vertical Take-Off and Landing (PVTOL) [21], presenting experimental results. Furthermore, using the command filtered backstepping approach with a parameter scheduling algorithm, an experimental flying test of a quadrotor with a nonlinear controller in an indoor environment was described in [22]; however, the robust design was not included in the controller synthesis. Simulation routines of a robust control based on a backstepping method was presented in [23] to drive the position and attitude of a unmanned mini aerial rotorcraft vehicle subjected to bounded uncertainties and bounded disturbances.

According to the specialized literature cited above, most of the robust backstepping or robust control algorithms were tested in simulations: [7–10,15,16,19,20,23]; advanced control in an outdoor environment: for nitrogen estimation with relative high-cost equipment was presented in [12], and robust landing algorithm onto a heaving platform in [18]; in an indoor environment in UAVs [13,14,17] and in a PVTOL [21]. In our opinion, a robust

backstepping control algorithm has to be sufficiently simple to implement it on an autopilot for flying tests in a more realistic environment (outdoors and trajectory tracking in precision agriculture). In this sense, the embedded computing resources required by autonomous unmanned aircraft systems represented a challenge [24]. The backstepping algorithm allows to synthesize a recursive stabilizing controllers for nonlinear systems and it could be modified to obtain robust controllers. Moreover, unmodeled dynamics, external disturbances, actuators, and signal conditioner nonlinearities are latent in the real-time control loop. Under all these conditions, it is essential to assess the controller's performance [25] by experimental evaluation of the control algorithms in meaningful environments to validate their robustness in a closed loop with a real plant.

In this contribution, based on the Lyapunov approach, a robust nonlinear controller solves the trajectory tracking problem for a quadrotor. Our proposal uses the backstepping method [26], allowing to construct an iterative control algorithm that rejects the effects of the unstructured dynamics in the quadrotor nonlinear model. For this purpose, the quadrotor model is divided into four subsystems related to the altitude, the yaw angle, the pitch- x and the roll- y , as in [11], and it considers the coupled nonlinear dynamics in the actuators. While these disturbances are unknown, they are assumed as bounded matched vanishing dynamics. The robustness is incorporated in the proposed controller to stabilize the drone in the presence of non-modeled dynamics, which improve the image capture system on board of the UAV as it flies over corn crops to detect dry leaves, providing clues to identify fungus such as *Phyllachora maydis*, *Monographella maydis*, and *Coniothyrium phyllachorae* on the corn leaves. These fungi are the cause of the tar spot in the corn [27], which causes crop losses.

Regarding the proposal mission in this paper, growing loss owing to disease is one of the most representative problem in agriculture. Crop growth inspecting and early identification of pest in the crops is still an important issue. In this regard, farmers are investing great efforts to conserve crops; however, they are mostly failing because they are not correctly monitoring the crops when they have been infested. Additionally, plagues in the crops are also difficult to detect due to it is not uniformly distributed. Hence, UAVs have a key role in crop disease surveillance and early detection [28–30]. This research attempts to provide experimental evidence of the implementation feasibility of the developed robust backstepping control in a PixHawk autopilot used in a pest detection task. An RGB GoPro Hero8 Black camera is mounted on the vehicle, and the video is processed off-line in the MATLAB software. This image processing provides the approximated location of the possibly affected crops. The overall system (UAV and camera) represents a relative low-cost (less than USD 450) alternative in precision agriculture.

The contributions of the article are as follows:

- A new robust backstepping approach-based control algorithm that considers matched vanishing disturbances is proposed. The proposed controller uses a virtual bounded input: the function $\sin(\cdot)$, which produces bounded control signals, and it is appropriate to the physical constrains of the UAV.
- Experimental results in trajectory tracking using a UAV in an outdoor environment are reported. A specific precision agriculture task is involved using a commercial camera system and MATLAB software.
- The robustness provided to the proposed controller allows to reduce the capture of distorted crop images and it represents better performance when our proposal is compared to a PD controller, although a gimbal or additional software are not used.
- This article tries to fill the gap between the technological development with advanced control theory.

Accordingly, the paper is organized as follows: In Section 2, the problem formulation and the main result are presented. The quadrotor model is shown in Section 3, and the application of the control strategy is developed in Section 4. The description of the experimental platform and the experimental results are given in Section 5. Section 6 demonstrates how pests are detected in the field. Finally, concluding comments are enunciated in Section 7.

2. Problem Formulation and Main Results

In this part, the problem is stated and the theoretical main result is presented.

2.1. System Description

The following perturbed nonlinear system is addressed:

$$\dot{\vartheta} = f_0(\vartheta) + f_1(\vartheta)\zeta \tag{1}$$

$$\dot{\zeta} = u + \delta(t, \vartheta), \tag{2}$$

where $[\vartheta^T, \zeta]^T \in \mathbb{R}^{n+1}$ is the state, and $u \in \mathbb{R}$ is the control input. It is considered that the next assumptions are fulfilled:

1. The known functions $f_0 : D \rightarrow \mathbb{R}^n$ and $f_1 : D \rightarrow \mathbb{R}^n$ are continuously differentiable in a domain $D \subset \mathbb{R}^n$ that contains the origin $\vartheta = 0$ and $f_0(0) = 0$.
2. The Equation (1) of the system can be stabilized with a state feedback $\zeta = \mu(\vartheta)$, with $\mu(0) = 0$, then, there is a Lyapunov function that satisfies the following equation:

$$\frac{\partial V(\vartheta)}{\partial \vartheta} [f_0(\vartheta) + f_1(\vartheta)\mu(\vartheta)] \leq -W(\vartheta),$$

where $W(\vartheta)$ is a positive definite function, $\forall \vartheta \in D$.

3. Function $\delta(t, \vartheta)$ is a bounded matched vanishing perturbation, i.e.,

$$\delta(t, 0) = 0, |\delta(t, \vartheta)| \leq \Delta, \text{ and } \Delta > 0. \tag{3}$$

Problem statement: Design a control u that guarantees the closed-loop robust stability of the origin ($\vartheta = 0, \zeta = 0$) of system (1) and (2) in the presence of unstructured matched disturbance $\delta(t, \vartheta)$. Then, compute the control laws for every subsystem of the quadrotor, guaranteeing its robust stabilization.

2.2. Main Result

The synthesis of the proposed control algorithm is stated as follows.

Theorem 1. Consider the system (1) and (2), under the Assumptions 1–3, for $k > 0$, and for a Lyapunov function of the form $V_1(\vartheta, \bar{z}) = V(\vartheta) + \frac{1}{2}\bar{z}^2$, with $\bar{z} = \zeta - \mu(\vartheta)$, then, the control law

$$u = \frac{\partial \mu(\vartheta)}{\partial \vartheta} [f_0(\vartheta) + f_1(\vartheta)\zeta] - \frac{\partial V(\vartheta)}{\partial \vartheta} f_1(\vartheta) - k\bar{z} - \text{sgn}(\bar{z})\Delta \tag{4}$$

robustly stabilizes the system.

Proof of Theorem 1. The proof is inspired on the methodology given in [26]. Adding and subtracting $f_1(\vartheta)\mu(\vartheta)$ on the right-hand side of Equation (1), it yields to

$$\begin{aligned} \dot{\vartheta} &= [f_0(\vartheta) + f_1(\vartheta)\mu(\vartheta)] + f_1(\vartheta)[\zeta - \mu(\vartheta)] \\ \dot{\zeta} &= u + \delta(t, \vartheta). \end{aligned} \tag{5}$$

Define $\bar{f}_0(\vartheta) = f_0(\vartheta) + f_1(\vartheta)\mu(\vartheta)$, $\bar{z} = \zeta - \mu(\vartheta)$, then its derivative is given by $\dot{\bar{z}} = \dot{\zeta} - \dot{\mu}(\vartheta) = (u + \delta(t, \vartheta)) - \dot{\mu}(\vartheta)$, and consider the change of variable

$$\bar{v} = u - \dot{\mu}(\vartheta), \tag{6}$$

where

$$\dot{\mu}(\vartheta) = \frac{\partial \mu(\vartheta)}{\partial \vartheta} \dot{\vartheta} = \frac{\partial \mu(\vartheta)}{\partial \vartheta} [f_0(\vartheta) + f_1(\vartheta)\zeta]. \tag{7}$$

Now, the system (5) is expressed in the classical backstepping form

$$\begin{aligned} \dot{\vartheta} &= \bar{f}_0(\vartheta) + f_1(\vartheta)\bar{z} \\ \dot{\bar{z}} &= \bar{v} + \delta(t, \vartheta), \end{aligned} \tag{8}$$

when $\bar{z} = 0$, it guarantees that the system (8) has an equilibrium point at the origin (according to Assumption 2).

A positive definite Lyapunov candidate function of the form $V_1(\vartheta, \bar{z}) = V(\vartheta) + \frac{1}{2}\bar{z}^2$ is proposed, and its derivative along the trajectories of system (8), according to Assumption 2, is given by

$$\begin{aligned} \dot{V}_1(\vartheta, \bar{z})|_{(8)} &= \frac{\partial V(\vartheta)}{\partial \vartheta} \bar{f}_0(\vartheta) + \frac{\partial V(\vartheta)}{\partial \vartheta} f_1(\vartheta)\bar{z} + \bar{z}\dot{\bar{z}} \\ &\leq -W(\vartheta) + \frac{\partial V(\vartheta)}{\partial \vartheta} f_1(\vartheta)\bar{z} + \bar{z}\dot{\bar{z}}, \end{aligned}$$

where $\dot{\bar{z}} = \bar{v} + \delta(t, \vartheta)$, set

$$\bar{v} = -k\bar{z} - \frac{\partial V(\vartheta)}{\partial \vartheta} f_1(\vartheta) - \text{sgn}(\bar{z})\Delta, \tag{9}$$

with $k > 0$; substituting into the previous equation, it yields

$$\dot{V}_1(\vartheta, \bar{z})|_{(8)} \leq -W(\vartheta) - k\bar{z}^2 + \bar{z}\delta(t, \vartheta) - \bar{z}\text{sgn}(\bar{z})\Delta. \tag{10}$$

Majorizing the term

$$\bar{z}\delta(t, \vartheta) \leq |\bar{z}\delta(t, \vartheta)| \leq |\bar{z}||\delta(t, \vartheta)| \leq |\bar{z}|\Delta,$$

and $|\bar{z}| \triangleq \bar{z}\text{sgn}(\bar{z})$; so, $|\bar{z}|\Delta = \bar{z}\text{sgn}(\bar{z})\Delta$; substituting into Equation (10), it yields

$$\dot{V}_1(\vartheta, \bar{z})|_{(8)} \leq -W(\vartheta) - k\bar{z}^2 < 0,$$

so, the origin of the system (8) is robustly stable; since $\mu(0) = 0$, the origin of system (1) and (2) is robustly stable too. The input $u = \bar{v} + \dot{\mu}(\vartheta)$ is obtained from Equation (6), and replacing Equations (7) and (9) into u , the result follows in Equation (4). \square

The control action (4) is used to solve the trajectory tracking problem of a quadrotor whose model is shown in the following section.

3. Quadrotor Model

The quadrotor model (see Figure 1) is obtained by representing a solid body moving in 3D space and being subjected to one force and three moments [31]. The generalized coordinates of the four rotor helicopter $q = (\gamma, \lambda)$, where $\gamma = (x, y, z) \in \mathbb{R}^3$ represents the relative position of the center of mass of the quadrotor with respect to an inertial reference, and $\lambda = (\phi, \theta, \psi) \in \mathbb{R}^3$ are the Euler angles representing the orientation of the quadrotor, known as roll-pitch-yaw, respectively. The inertial moments in the flying robot are defined by I_x, I_y, I_z ordered with respect to x, y and z axes, while m denotes its mass, l is the arm length, and τ_ϕ, τ_θ and τ_ψ are the input signals to be applied to the motors.

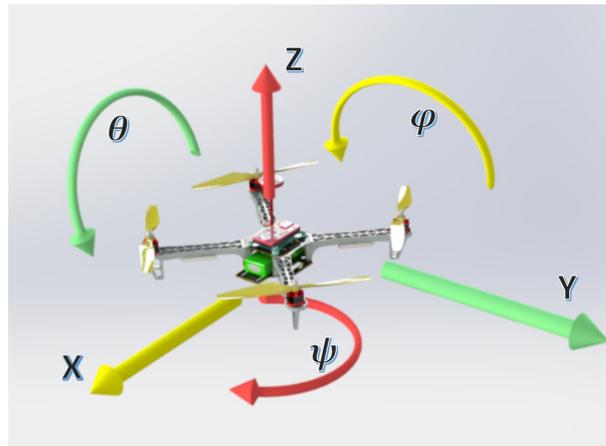


Figure 1. Quadcopter diagram.

The following equations describe the movement of the quadcopter affected by collective throttle u_s and torques τ_ϕ , τ_θ and τ_ψ [32]

$$\begin{aligned}
 \ddot{x} &= -\frac{1}{m} \sin \theta u_s \\
 \ddot{y} &= \frac{u_s}{m} \cos \theta \sin \phi \\
 \ddot{z} &= \frac{u_s}{m} \cos \theta \cos \phi - g \\
 \ddot{\phi} &= \dot{\theta} \dot{\psi} \frac{I_y - I_z}{I_x} + \frac{1}{I_x} \tau_\phi \\
 \ddot{\theta} &= \dot{\phi} \dot{\psi} \frac{I_x - I_z}{I_y} + \frac{1}{I_y} \tau_\theta \\
 \ddot{\psi} &= \dot{\phi} \dot{\theta} \frac{I_y - I_x}{I_z} + \frac{1}{I_z} \tau_\psi.
 \end{aligned} \tag{11}$$

The following assumptions, proposed in [11], are considered:

1. The UAV is considered as a rigid and symmetrical body.
2. The center of gravity (CoG) of the quadrotor is in its origin.
3. The blades are rigid bodies with fixed angle pitch.
4. In the nominal model of the quadrotor, the aerodynamics effects have not been considered.
5. The motor dynamic could be modeled as a first-order transfer function, and it is sufficient to reproduce the dynamics between the propeller's speed set-point and its true speed. As the time constant of this transfer function is small, we can consider that the rotor dynamic is approximately equal to one [33]. This assumption is used to suppose additional dynamics (or uncertain parameters) that represent matched vanishing perturbations in the actuators.
6. The attitude angles, pitch, roll and yaw are restricted in the interval $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

Remark 1. Additionally to Assumption 5, the effects of the propellers are neglected [34]. However, a more accurate trust model of the vehicle is given in [34], which can be expressed as

$$\begin{aligned}
 T_i &= c_1 \omega_i^2 \left(c_2 \left(1 + \frac{3}{2} \mu_i^2(t) \right) - \lambda_i(t) \right) \\
 &= c_1 \omega_i^2 c_2 + \underbrace{c_1 \omega_i^2 \left[c_2 \frac{3}{2} \mu_i^2(t) - \lambda_i(t) \right]}_{\delta},
 \end{aligned}$$

where T_i is the trust of motor i^{th} , ω_i is the angular velocity of the motor i^{th} , $\mu_i(t)$ and $\lambda_i(t)$ are the advance and inflow ratios respectively (nonlinear functions), c_1 and c_2 are positive constants. So, δ could be the second term of T_i , among other unmodelled dynamics.

With the nominal model given by (11), the robust backstepping controller is synthesized in the next section.

4. Control Strategy Applied to the Quadrotor

In this section, the quadrotor mathematical model (11) is divided into subsystems as was done in [11,35]: the altitude subsystem z , the yaw subsystem ψ , the pitch subsystem $x - \theta$ and the roll subsystem $y - \phi$. Then, the robust stabilizing controllers for each subsystem are obtained using the result of Theorem 1 while considering the presence of matched vanishing disturbances $\delta(t, \vartheta)$.

4.1. Altitude Controller

The subsystem z that describes the UAV dynamic of altitude is represented by

$$\ddot{z} = \frac{\cos \theta \cos \phi}{m} u_s - g + \delta(\vartheta, t), \tag{12}$$

where $|\delta(t, \vartheta)| \leq \Delta_1$, Δ_1 is the upper bound of the matched disturbances. Defining the state variables $x_1 = z$ and $x_2 = \dot{z}$, and

$$u_s = \frac{m}{\cos \theta \cos \phi} [g + u_z], \tag{13}$$

and according to the backstepping methodology [26], $x_2 = \mu(x_1) = -\alpha_6 x_1$, for $\alpha_6 > 0$, the subsystem (12) is rewritten as follows:

$$\begin{aligned} \dot{x}_1 &= -\alpha_6 x_1 \\ \dot{x}_2 &= u_z + \delta(\vartheta, t). \end{aligned} \tag{14}$$

The Lyapunov function is $V_z(x_1) = \frac{1}{2} x_1^2$, and following the result of Theorem 1, with $\zeta = x_2$, $f_0(\vartheta) = 0$, $\vartheta = x_1$, $f_1(\vartheta) = 1$, and $\bar{z} = x_2 + \alpha_6 x_1$, the stabilizing control for subsystem (14) takes the following form:

$$u_z = -(1 + k_z \alpha_6) x_1 - (\alpha_6 + k_z) x_2 - \text{sgn}(x_2 + \alpha_6 x_1) \Delta_1.$$

4.2. Yaw Controller

The dynamic of the ψ angle is given by

$$\ddot{\psi} = \dot{\phi} \dot{c}_1 + c_2 \tau_\psi + \delta(\vartheta, t), \tag{15}$$

where $c_1 = \frac{I_y - I_x}{I_z}$, $c_2 = \frac{1}{I_z}$, $|\delta(t, \vartheta)| \leq \Delta_2$; I_x, I_y, I_z are parameters of the inertia matrix. Defining the state variables $x_3 = \psi$, $x_4 = \dot{\psi}$, and

$$\tau_\psi = \frac{1}{c_2} (-\dot{\phi} \dot{c}_1 + u_\psi), \tag{16}$$

then Equation (15) becomes as follows:

$$\ddot{\psi} = u_\psi + \delta(\vartheta, t).$$

Following a similar procedure as in the altitude control, $x_4 = \mu(x_3) = -\alpha_7 x_3$, for $\alpha_7 > 0$, the subsystem (15) yields to

$$\begin{aligned} \dot{x}_3 &= -\alpha_7 x_3 \\ \dot{x}_4 &= u_\psi + \delta(\vartheta, t). \end{aligned} \tag{17}$$

Let $V_\psi(x_3) = \frac{1}{2} x_3^2$ the Lyapunov function and defining $\zeta = x_4$, $\vartheta = x_3$, $f_0(\vartheta) = 0$, $f_1(\vartheta) = 1$, $\bar{z} = x_4 + \alpha_7 x_3$, the control law for the subsystem (17) is given by

$$u_\psi = -(1 + k_\psi \alpha_7) x_3 - (\alpha_7 + k_\psi) x_4 - \text{sgn}(x_4 + \alpha_7 x_3) \Delta_2. \tag{18}$$

4.3. Controller for Subsystem $x - \theta$

Dynamics related with translational motion along x -axis and the pitch angle θ around the y -axis are described by

$$\begin{aligned} \dot{x} &= -\frac{1}{m} \sin \theta u_s \\ \ddot{\theta} &= \dot{\phi} \psi \frac{I_x - I_z}{I_y} + \frac{1}{I_y} \tau_\theta + \delta(\vartheta, t), \end{aligned} \tag{19}$$

where $c_3 = \frac{I_x - I_z}{I_y}$, $u_s \neq 0$, and $c_4 = \frac{1}{I_y}$. Let $x_5 = x$, $x_6 = \dot{x}$, $x_7 = \theta$ and $x_8 = \dot{\theta}$. Then, the state space representation is given by

$$\begin{aligned} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\frac{1}{m} \sin x_7 u_s \\ \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \dot{\phi} \psi c_3 + c_4 \tau_\theta + \delta(\vartheta, t). \end{aligned} \tag{20}$$

Considering the first and second equations of the state space representation (20) and defining the virtual input as follows:

$$u_{1x} = \sin x_7 = -\frac{m}{u_s} u_x, \tag{21}$$

then, according to the backstepping methodology [26] $x_6 = \mu(x_5) = -\alpha_5 x_5$, for $\alpha_5 > 0$. Notice that the selected virtual input $\sin(x_7)$ produces bounded control signals and it is appropriate to the physical constrains of the UAV: $x_7 \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. These equations can be rewritten as follows:

$$\begin{aligned} \dot{x}_5 &= -\alpha_5 x_5 \\ \dot{x}_6 &= u_x, \end{aligned}$$

where $u_x = -(1 + \alpha_5 k_5)x_5 - (\alpha_5 + k_5)x_6$ is the control input, computed with the Lyapunov function $V_1(x_5) = \frac{1}{2}x_5^2$. Replacing u_x into Equation (21), it yields to

$$u_{1x} = \frac{m}{u_s} (d_1 x_5 + d_2 x_6), \tag{22}$$

where $d_1 = 1 + \alpha_5 k_5$ and $d_2 = \alpha_5 + k_5$.

Next, the iterative methodology of backstepping is applied; taking the first three equations of the state space representation (20), the subsystem is rewritten as

$$\begin{aligned} \begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} &= \begin{bmatrix} x_6 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{u_s}{m} \end{bmatrix} \sin x_7 \\ \dot{x}_7 &= x_8 = u_{2x}, \end{aligned} \tag{23}$$

In this case, $\vartheta = (x_5, x_6)^T$, and $\zeta = x_7$. The modified backstepping structure, proposed by [11], is applied for system (23) considering the Lyapunov function

$$V_2 = V_2(x_5, x_6) = \frac{1}{2}x_5^2 + \frac{\beta_1}{2}(x_6 + \alpha_5 x_5)^2.$$

The virtual input $\mu_1 = \mu_1(x_5, x_6) = \frac{m}{u_s} (d_1 x_5 + d_2 x_6)$ is defined as u_{1x} , given in Equation (22); substituting Equation (13) in μ_1

$$\mu_1(x_5, x_6) = \frac{\cos x_7 \cos x_{11}}{p} (d_1 x_5 + d_2 x_6).$$

where $p = g + u_z$. Applying Proposition 1 of [11], the control u_{2x} has the following form:

$$u_{2x} = \frac{\frac{\partial \mu_1}{\partial \vartheta} [f_0(\vartheta) + f_1(\vartheta) \sin \zeta] - \frac{\partial V_2}{\partial \vartheta} f_1(\vartheta) - k_6 (\sin x_7 - \mu_1)}{\cos \zeta}, \tag{24}$$

where

$$\begin{aligned} \frac{\partial(\mu_1(x_5, x_6))}{\partial x_5} &= \frac{\cos x_7 \cos x_{11} d_1}{p}, \\ \frac{\partial(\mu_1(x_5, x_6))}{\partial x_6} &= \frac{\cos x_7 \cos x_{11} d_2}{p}, \\ \frac{\partial(V_2(x_5, x_6))}{\partial x_6} &= \beta_1 \alpha_5 x_5 + \beta_1 x_6, \end{aligned}$$

So, the virtual input u_{2x} is

$$\begin{aligned} u_{2x} &= \left(\frac{k_6 d_1 \cos x_{11}}{p} + \frac{\beta_1 \alpha_5 p}{\cos^2 x_7 \cos x_{11}} \right) x_5 \\ &+ \left(\frac{d_1 \cos x_{11} + k_6 d_2 \cos x_{11}}{p} + \frac{\beta_1 p}{\cos^2 x_7 \cos x_{11}} \right) x_6 - (d_2 + k_6) \tan x_7 \end{aligned} \tag{25}$$

Finally, the whole state space representation (20) is considered with $\vartheta = (x_5, x_6, x_7)^T$, $\zeta = x_8$, and for

$$\tau_\vartheta = \frac{1}{c_4} [-\dot{\phi} \psi c_3 + \tau_{\theta a}], \tag{26}$$

this subsystem can be rewritten as

$$\begin{aligned} \begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} &= \begin{bmatrix} x_6 \\ -\frac{p \sin x_7}{\cos x_7 \cos x_{11}} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_8, \\ \dot{x}_8 &= \tau_{\theta a} + \delta(\vartheta, t) = u_{3x} + \delta(\vartheta, t). \end{aligned} \tag{27}$$

The proposed Lyapunov function for subsystem (27) is

$$V_3 = V_3(x_5, x_6, x_7) = \frac{1}{2} x_5^2 + \frac{\beta_1}{2} (x_6 + \alpha_5 x_5)^2 + \frac{\beta_2}{2} \left(\sin x_7 - \frac{\cos x_7 \cos x_{11}}{p} (d_1 x_5 + d_2 x_6) \right)^2,$$

and the virtual input μ_2 is associated with u_{2x} , given in (25), as

$$\mu_2 = \left(\frac{k_6 d_1 \cos x_{11}}{p} + \frac{\beta_1 \alpha_5 p}{\cos^2 x_7 \cos x_{11}} \right) x_5 + \left(\frac{(d_1 + k_6 d_2) \cos x_{11}}{p} + \frac{\beta_1 p}{\cos^2 x_7 \cos x_{11}} \right) x_6 - (d_2 + k_6) \tan x_7.$$

According to Theorem 1, the controller is given by

$$u_{3x} = -k_7 z_3 - \frac{\partial V_3(\vartheta)}{\partial \vartheta} f_1(\vartheta) - \text{sgn}(z_3) \Delta_3 + \frac{\partial \mu_2(\vartheta)}{\partial \vartheta} [f_0(\vartheta) + f_1(\vartheta) \zeta],$$

where $z_3 = x_8 - \mu_2$, $k_7 > 0$, $\vartheta = (x_5, x_6, x_7)^T$, $f_1(\vartheta) = [0, 0, 1]^T$, and

$$\begin{aligned} \frac{\partial \mu_2}{\partial x_5} &= \left(\frac{k_6 d_1 \cos x_{11}}{p} + \frac{\beta_1 \alpha_5 p}{\cos^2 x_7 \cos x_{11}} \right) \\ \frac{\partial \mu_2}{\partial x_6} &= \left(\frac{d_1 \cos x_{11}}{p} + \frac{k_6 d_2 \cos x_{11}}{p} + \frac{\beta_1 p}{\cos^2 x_7 \cos x_{11}} \right) \\ \frac{\partial \mu_2}{\partial x_7} &= \left(\frac{2\beta_1 \alpha_5 p \sin x_7}{\cos^3 x_7 \cos x_{11}} \right) x_5 + \left(\frac{2\beta_1 p \sin x_7}{\cos^3 x_7 \cos x_{11}} \right) x_6 - (d_2 + k_6) \sec^2 x_7 \\ \frac{\partial V_3}{\partial x_7} &= \beta_2 \sin x_7 \cos x_7 - \left[\frac{\beta_2 \cos x_{11} (d_1 x_5 + d_2 x_6)}{p} \right] (2 \cos^2 x_7 - 1). \end{aligned}$$

Therefore, the input u_{3x} is

$$u_{3x} = \frac{\partial \mu_2}{\partial x_5} x_6 - \frac{\partial \mu_2}{\partial x_6} \frac{p \sin x_7}{\cos x_7 \cos x_{11}} + \frac{\partial \mu_2}{\partial x_7} x_8 - \frac{\partial V_3}{\partial x_7} - \text{sgn}(x_8 - \mu_2) \Delta_3 - k_7 (x_8 - \mu_2).$$

4.4. Controller for Subsystem $y - \phi$

The translational and rotational dynamics experienced by the flying robot along the y -axis and the roll angle ϕ respectively are represented by

$$\begin{aligned} \dot{y} &= \frac{u_s}{m} \cos \theta \sin \phi \\ \dot{\phi} &= \theta \dot{\psi} c_5 + c_6 \tau_\phi + \delta(t, \vartheta), \end{aligned} \tag{28}$$

where $c_5 = \frac{I_y - I_z}{I_x}$, $c_6 = \frac{1}{I_x}$. Defining the variables $x_9 = y$, $x_{10} = \dot{y}$, $x_{11} = \phi$, $x_{12} = \dot{\phi}$, the state space representation is given by

$$\begin{aligned} \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= \frac{\cos x_{11}}{m} \sin x_{11} u_s \\ \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= \theta \dot{\psi} c_5 + c_6 \tau_\phi + \delta(t, \vartheta). \end{aligned} \tag{29}$$

Considering the first two equations of (29), the virtual input

$$u_y = \sin x_{11} = \frac{m}{u_s \cos x_{11}} u_1, \tag{30}$$

and defining $x_{10} = \mu(x_9) = -\alpha_1 x_9$, for $\alpha_1 > 0$; the subsystem is rewritten as $\dot{x}_9 = -\alpha_1 x_9$ and $x_{10} = u_1$. The Lyapunov function is $V_1(x_9) = \frac{1}{2} x_9^2$, and according to the classic backstepping methodology, the input $u_1 = -(k_1 \alpha_1 + 1)x_9 - (k_1 + \alpha_1)x_{10}$ is substituted into Equation (30), which yields

$$u_y = \frac{m}{u_s \cos x_{11}} [-b_1 x_9 - b_2 x_{10}], \tag{31}$$

where $b_1 = k_1 \alpha_1 + 1$, $b_2 = k_1 + \alpha_1$; substituting (13) into (31)

$$u_y = \frac{\cos x_{11}}{p} (-b_1 x_9 - b_2 x_{10}).$$

In the next step, the third equation of the state space representation (29) is added, and the subsystem is rewritten as in [11], with $\vartheta = (x_9, x_{10})^T$, and $\zeta = x_{11}$, as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}_9 \\ \dot{x}_{10} \end{bmatrix} &= \begin{bmatrix} x_{10} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{p}{\cos x_{11}} \end{bmatrix} \sin x_{11} \\ \dot{x}_{11} &= x_{12} = u_{y2}. \end{aligned} \tag{32}$$

Following a similar procedure applied to the previous subsystems, with $V_2(x_9, x_{10}) = \frac{1}{2} x_9^2 + \frac{\beta_3}{2} (x_{10} + \alpha_1 x_9)^2$, and $\mu_1 = \frac{\cos x_{11}}{p} (-b_1 x_9 - b_2 x_{10})$, a control is obtained as the given in the Equation (24), and it has the following structure:

$$u_{y2} = - \left[\frac{k_2 b_1}{p} + \frac{\beta_3 \alpha_1 p}{\cos^2 x_{11}} \right] x_9 - \left[\frac{b_1 + k_2 b_2}{p} + \frac{\beta_3 p}{\cos^2 x_{11}} \right] x_{10} - (b_2 + k_2) \tan x_{11}. \tag{33}$$

In the last step, subsystem (29) is fully considered, with $\vartheta = (x_9, x_{10}, x_{11})^T$, $\zeta = x_{12}$, and defining

$$\tau_\phi = \frac{1}{c_6} [-\theta \dot{\psi} c_5 + \tau_1], \tag{34}$$

it can be rewritten as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \end{bmatrix} &= \begin{bmatrix} x_{10} \\ \frac{p \sin x_{11}}{\cos x_{11}} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_{12} \\ \dot{x}_{12} &= \tau_1 + \delta(t, \vartheta), \end{aligned} \tag{35}$$

where $\tau_1 = u_{y3}$, μ_2 is associated with u_{y2} , given in Equation (33), and the Lyapunov function is defined by

$$V_3(x_9, x_{10}, x_{11}) = \frac{1}{2}x_9^2 + \frac{\beta_3}{2}(x_{10} + \alpha_1x_9)^2 + \frac{\beta_4}{2} \left[\sin x_{11} + \frac{(b_1x_9 + b_2x_{10}) \cos x_{11}}{p} \right]^2.$$

From Theorem 1, the input u_{y3} is computed as

$$u_{y3} = \frac{\partial \mu_2}{\partial x_9} x_{10} + \frac{\partial \mu_2}{\partial x_{10}} \frac{p \sin x_{11}}{\cos x_{11}} + \frac{\partial \mu_2}{\partial x_{11}} x_{12} - \frac{\partial V_3}{\partial x_{11}} - k_3(x_{12} - \mu_2) - \text{sgn}(x_{12} - \mu_2)\Delta_4.$$

where

$$\begin{aligned} \frac{\partial \mu_2}{\partial x_9} &= -\left(\frac{k_2b_1}{p} + \frac{\beta_3\alpha_1p}{\cos^2 x_{11}} \right) \\ \frac{\partial \mu_2}{\partial x_{10}} &= -\left(\frac{b_1+k_2b_2}{p} + \frac{\beta_3p}{\cos^2 x_{11}} \right) \\ \frac{\partial \mu_2}{\partial x_{11}} &= -\left(\frac{(k_2+b_2)}{\cos^2 x_{11}} + \frac{(\alpha_1x_9+x_{10})2\beta_3p \sin x_7}{\cos^3 x_7} \right) \\ \frac{\partial V_3}{\partial x_{11}} &= \left[\frac{\beta_4}{p} (b_1x_9 + b_2x_{10}) \right] (2 \cos^2 x_{11} - 1) \\ &+ \left(\beta_4 - \frac{\beta_4(b_1^2x_9^2+b_2^2x_{10}^2)}{p^2} - \frac{2\beta_4b_1b_2x_9x_{10}}{m^2p^2} \right) \sin x_{11} \cos x_{11}. \end{aligned}$$

To assess the effectiveness of the synthesized controllers (13), (16), (26) and (34) in the trajectory tracking task of a quadrotor UAV, experimental tests are conducted as described in the next section.

5. Experimental Results

5.1. Experimental Platform Description

The quadrotor is constructed on a plastic frame F450. An autopilot Pixhawk, version 2.4.8, is on board the F450 structure. The Pixhawk exhibits the following performance: a primary clock at 168 MHz, and with a 32-bit processor STM32F427 Cortex M4 core with a floating processor unit. The autopilot has two accelerometers, a gyroscope, and two magnetometers for attitude measurements of the UAV. The Pixhawk is equipped with radio signal input ports compatible with Futaba 10J radio. Furthermore, it has eight pulse width modulation main outputs that are used to control the motors. To pinpoint the longitudinal positions, an external module Ublox NEO-M8N GPS (accuracy of 0.6–0.9 m) is mounted. The yaw angle is estimated via an internal magnetometer located in the GPS. As power supply of the UAV, Lipo technology batteries are employed (capacity of 5200 mAh, discharge rate of 15 C). The quadrotor has four propellers of model 1045, four 2212 920 KV brushless motors and four electronic speed controllers of model SIMONK30A. The trajectory tracking reference was programmed into the Pixhawk (Figure 2).



Figure 2. Quadrotor prototype.

5.2. Experimental Results Applying Synthesized Controllers

In this subsection, for comparison purposes, experimental tests in outdoor environment of UAV are presented for the proposed backstepping controller and a PD controller. For the trajectory tracking task on the spaces $(x; y; z)$ and (ϕ, θ, ψ) , the UAV follows three paths: 1. The manual takeoff of the UAV, until the desired altitude is reached, and this altitude is defined by the pilot. 2. The quadrotor tracks parametric circle equations given by

$$\begin{aligned} x_{ref} &= \left(-5 \cos\left(\frac{\pi}{180}t\right) + 5\right)m, \\ y_{ref} &= -5 \sin\left(\frac{\pi}{180}t\right)m, \\ z_{ref} &= 2.3m. \end{aligned}$$

3. At the end, when the trajectory is finished, the pilot regains control of the quadrotor and lands it. The programmed sampling time (T) is $T = 0.01$ s, it was used in all experimental tests. The quadrotor parameters are as follows: mass $m = 1.3$ kg, distance from the motors to the centre of gravity $l = 0.3$ m, constant of gravity $g = 9.81$ m/s² and inertia moments $I_x = I_y = 0.01567$ kgm², $I_z = 0.028346$ kgm². The gains of controllers (13), (16), (26) and (34) are heuristically proposed in Table 1. The delta values ($\Delta_i, i = 1, 2, 3, 4$) are the maximum values for the matched disturbances that could be adjusted heuristically such that the closed-loop performance is satisfactory.

Table 1. Controller gains.

Subsystems			
ψ	z	$x - \theta$	$y - \phi$
$k_\psi = 24.2$	$k_z = 7.39$	$k_5 = 0.1$	$k_1 = 0.4$
$\alpha_7 = 5.7$	$\alpha_6 = 12.3$	$k_6 = 0.13$	$k_2 = .27$
$\Delta_1 = 3.2$	$\Delta_2 = 2.5$	$k_7 = 7.3$	$k_3 = 5.3$
		$\Delta_3 = 5$	$\Delta_4 = 2$
		$\alpha_5 = 12$	$\alpha_1 = 0.009$
		$\beta_1 = 0.1$	$\beta_3 = 5.2$
		$\beta_2 = 0.25$	$\beta_4 = 3.64$

To compare the behavior of the proposed controller, a PD controller, which is tuned as is proposed in [35], is employed, obtaining the gains by proposing the following temporal parameters: the maximum overshoot and the settling time. Translational and attitude initial values of the drone are near to the origin.

Figure 3 displays the translational positions recorded during the experiment following the desired trajectory. The figure is divided in three figures: 1. The first shows the trajectory of the vehicle on the X axis, 2. The second displays the trajectory along the Y axis, 3. The last one illustrates the trajectory tracking task on the Z axis. As shown in Figure 3, the trajectory tracking task in (X, Y, Z) space, in the blue line, displays a reliable performance when it is compared with the reference values in the red line. The position performance of the drone in the (X, Y) space is displayed in Figures 4 and 5. It is noteworthy that these experimental results were obtained in an outdoor environment.

Figure 6 shows the relative small errors during the trajectory tracking. During the trajectory tracking task, the references are subject to changes in the X, Y axes, then the errors may increase or decrease. Nevertheless, for the altitude, the Set Point is constant, then the error is relatively small. Furthermore, the measurements of the position, provided by the GPS device, present a relatively larger deviations (around 1.4 m to 1.6 m) than other devices employed for indoor environment. This explains the errors in the trajectory-tracking displacement task.

The four control input signals for roll, pitch, altitude and yaw, respectively, are presented in Figure 7. These control inputs are obtained with the proposed backstepping controllers and are sent to the motors in real-time during the tests for desired trajectory tracking.

Figure 8 shows the recorded translational positions, following the desired trajectory, for PD controllers. This figure is divided into three subfigures, representing the trajectory performance along the X, Y and Z axes.

Figure 9 shows the UAV's trajectory performance in three-dimensional space, whereas Figure 10 displays it in two-dimensional space. Figures 11 and 12 show the errors during the trajectory tracking and the control signals for PD controllers, respectively.

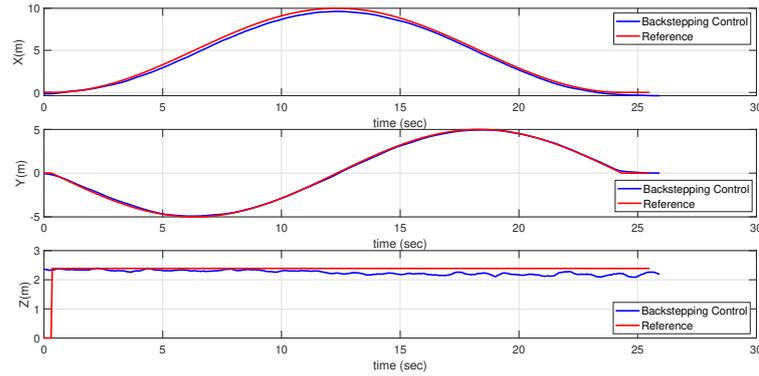


Figure 3. Positions in the X, Y and Z axes; (Backstepping controllers).

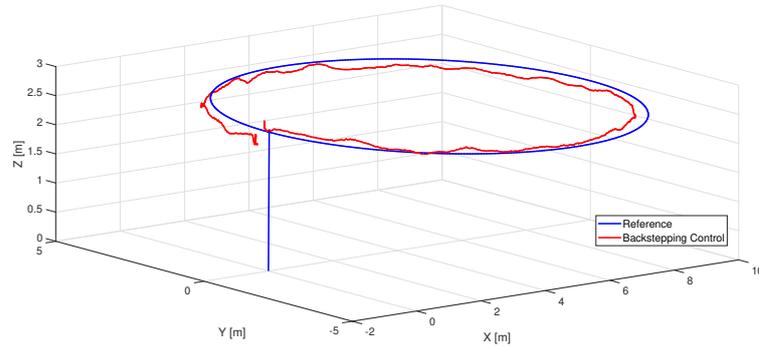


Figure 4. 3D view trajectory tracking; (Backstepping controllers).

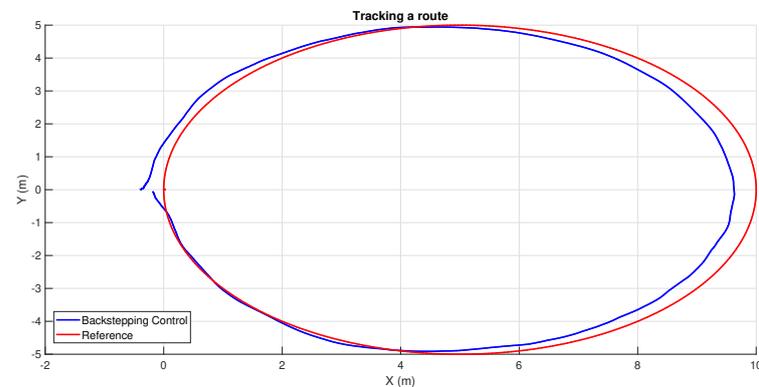


Figure 5. Trajectory tracking; (Backstepping controllers).

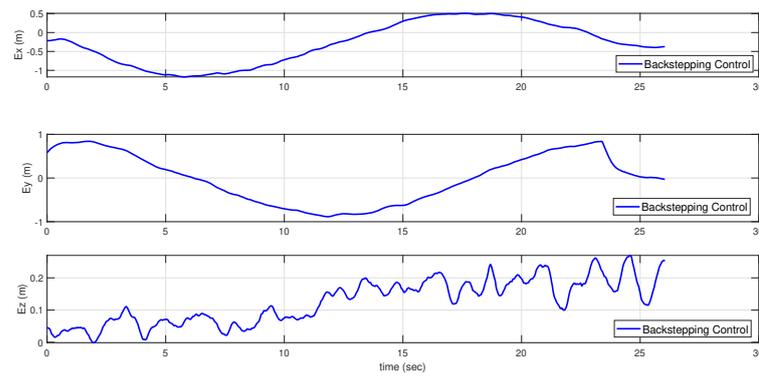


Figure 6. Error signals; (Backstepping controllers).

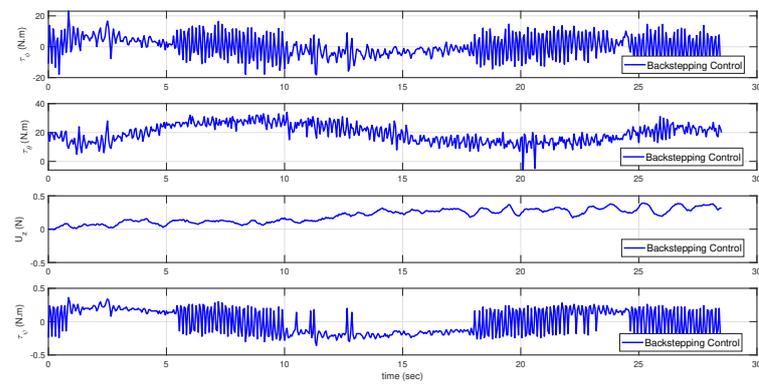


Figure 7. Control signals; (Backstepping controllers).

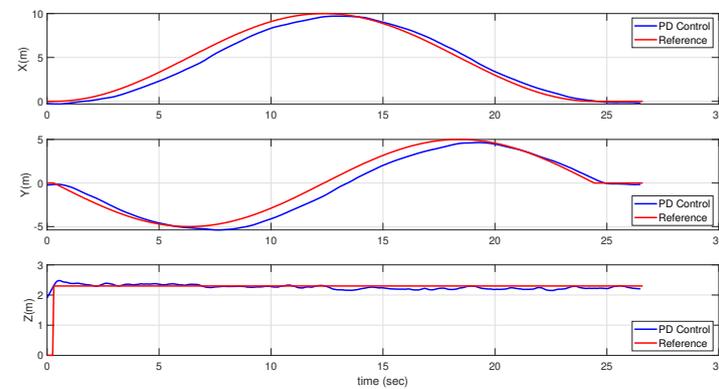


Figure 8. Translational positions in the (X, Y, Z) space (PD controllers).

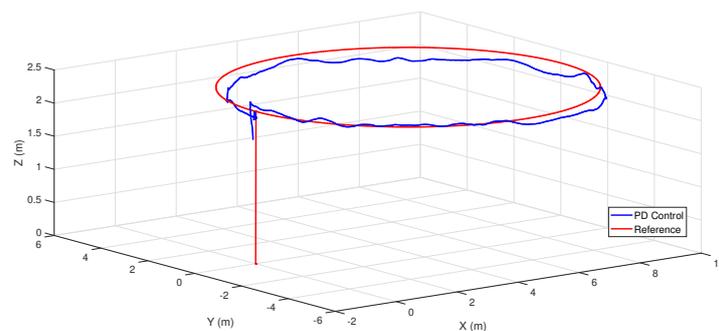


Figure 9. 3D view trajectory tracking (PD controllers).

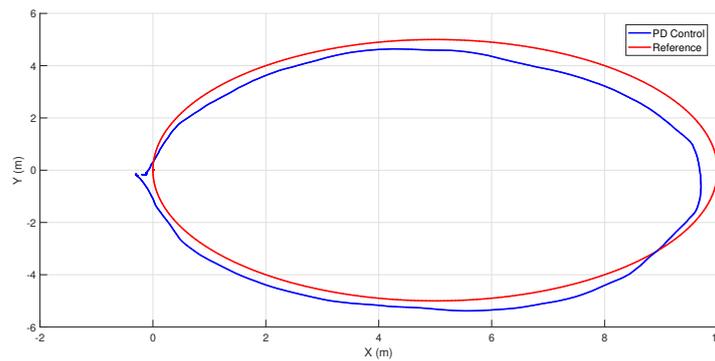


Figure 10. Trajectory tracking (PD controllers).

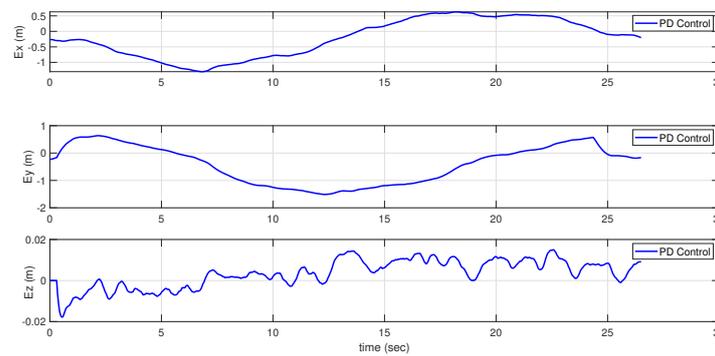


Figure 11. Error signals (PD controllers).

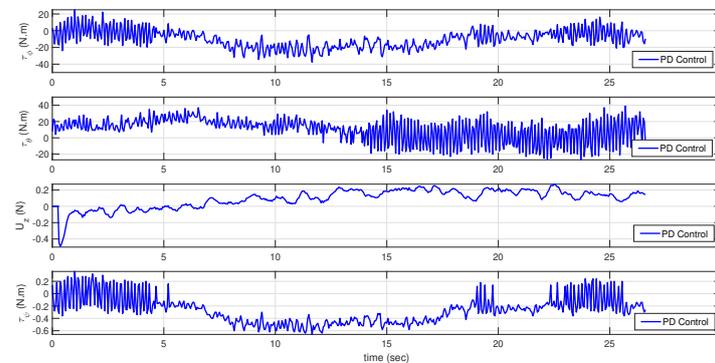


Figure 12. Control signals (PD controllers).

To compare the performances of both controllers, the Integral Absolute Error (IAE) is computed for the robust backstepping (BS) and PD, and these values are reported in Table 2.

Table 2. Comparative analysis of the performance IAE for the trajectory tracking phase.

Trajectory Tracking				
Performance Index	BS	PD	σ_{BS}	σ_{PD}
IAEx	674.0325	732.8948	0.3195	0.6095
IAEy	646.48	849.0646	0.5691	0.6949
IAEz	59.96	85.4140	0.06747	0.0678

Table 2 shows that the robust backstepping controller outperforms the PD controller in the trajectory-tracking phase of the quadrotor in the presence of disturbances; additionally, the standard deviation is calculated to analyze the error dispersion, and the mean value of

the 10 conducted experimental tests is calculated and compared with the results obtained using the PD control.

Additionally, Table 3 shows the mean of the integral of the absolute value of the signal control, defined as

$$J_{control} = \sum_{i=0}^N |control(i)|,$$

where N is the number of samples. The index $J_{control}$ is related with the energy consumption of the actuators.

Table 3. Comparison of absolute control values during the trajectory tracking phase.

Performance Index	Trajectory Tracking	
	BS	PD
J_{τ_ϕ}	6051.2	6662
J_{τ_θ}	7337	8661
J_{u_s}	519.647	662.5175
J_{τ_ψ}	802.8	848.2

According to the results showed on Table 3, the energy consumption required by the Robust Backstepping Controller (RBC) is lower than the PD controller. It implies that the control signal of the RBC is smaller and softer than the control signal of PD, which allows the captured images to be less blurred and distorted.

6. Detection of Pest in Corn Leaves

In this section, the vision system and the offline images processing are presented, aiming to locate the tar spot complex [27] in the maize fields based on the images taken when the quadcopter flies over the corn crop.

6.1. Vision System

The modules of the vision system include the camera and the image processing (a PC and MATLAB software). The natural light illuminates the plants to be inspected. The GoPro HERO8 Black camera [36] is mounted under the UAV frame (see Figure 13).



Figure 13. UAV with the GoPro HERO8 Black camera.

While the quadrotor flies over the crops, the camera lens captures and sends images to the camera sensor in the form of light. This sensor converts this light to a digital image that is then stored in the camera's memory for later analysis. For the image processing, the "Color Thresholding" toolbox is used (included in MATLAB Software), in which the interesting areas in the analyzed images are selected based on the color spaces: Red, Green, Blue (RGB); Hue, Saturation, Value (HSV); Luma, Blue and Red chromaticity (YCbCr) and

Luminosity, Red/Green and Yellow/Blue chromaticity coordinates (CIE-Lab). This is done by applying segmentation methods on the images; in this regard, image segmentation is a processing technique that refers to the extraction of useful information from a frame to facilitate observation and analysis since the rest of the image content is not useful for the purpose, i.e., if within an image the red color is selected, everything that is not red is discriminated by the segmentation method. This toolbox is selected because it allows comparing the intensity level, pixel by pixel, with a certain threshold previously defined by the user. To establish this threshold, it is necessary to analyze the interest region to be isolated, finding a characteristic and exclusive color level.

6.2. Image Processing

In computer vision, segmentation processes are of vital importance when it comes to detecting objects in unstructured environments.

In this article, the segmentation process employs the method based on regions [37] due to the specific task (detecting color levels on affected leaves). Methods based on regions aim to determine the areas of an image that have homogeneous properties, and the border of these areas will delimit some objects from others. Regarding the segmentation methods, Figure 14 displays the process of application followed in this work.

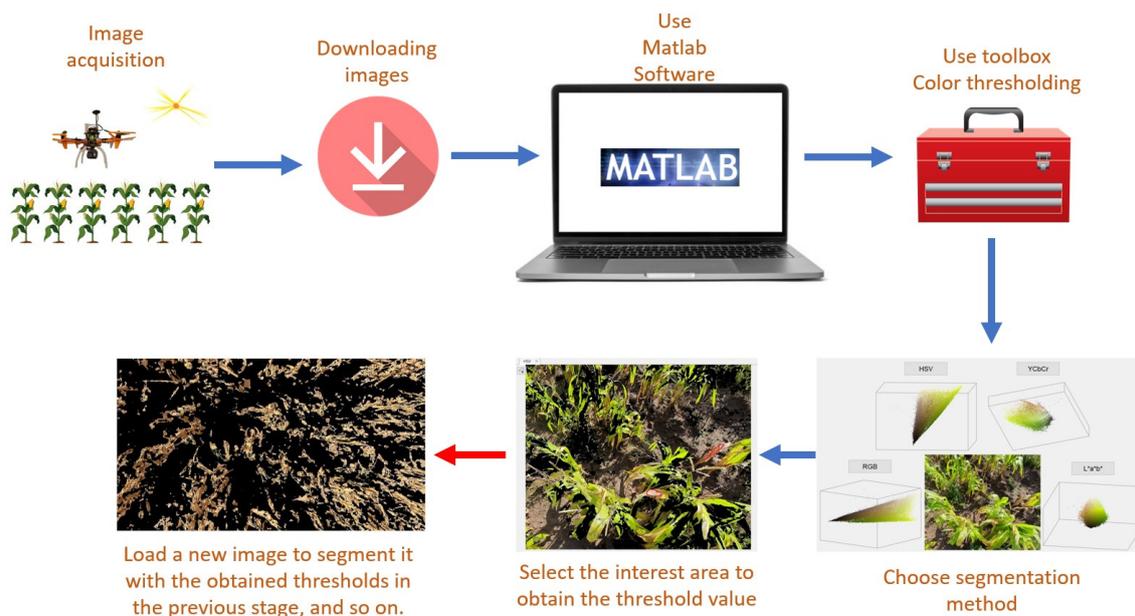


Figure 14. Scheme for the application of segmentation.

Segmentation Method

The application process of the segmentation method starts with the images of the corn fields (Figure 15) where the dry leaves of the plants are detected, as this is a symptom that the plant could be ill. For this purpose, first, based on the photograph, the pixels of interest are selected to find the thresholds. Once the pixel values are obtained, a new image is loaded to search for dry leaves in the field.

Figure 16 shows the processed image using different segmentation methods where the dry leaves of the plant are highlighted. It should be mentioned that in the processed image, the areas that are not of interest are colored in black. For the RGB segmentation method, the threshold values are as follows: $R > 126$, $G > 104$, $B < 80$. As for the HSV method, the ranges are as follows: $0.096 < H < 0.153$, $0.243 < S < 0.944$, $0.222 < V < 0.944$. In the YCbCr method, the following values are obtained: $70 < Y < 225$, $72 < Cb < 109$, $136 < Cr < 163$. Finally, from the application of the CIE-Lab segmentation, the values are as follows: $38.49 < L < 100$, $-10.353 < a < 19.622$, $19.622 < b < 55.531$.



Figure 15. Image captured by the camera.

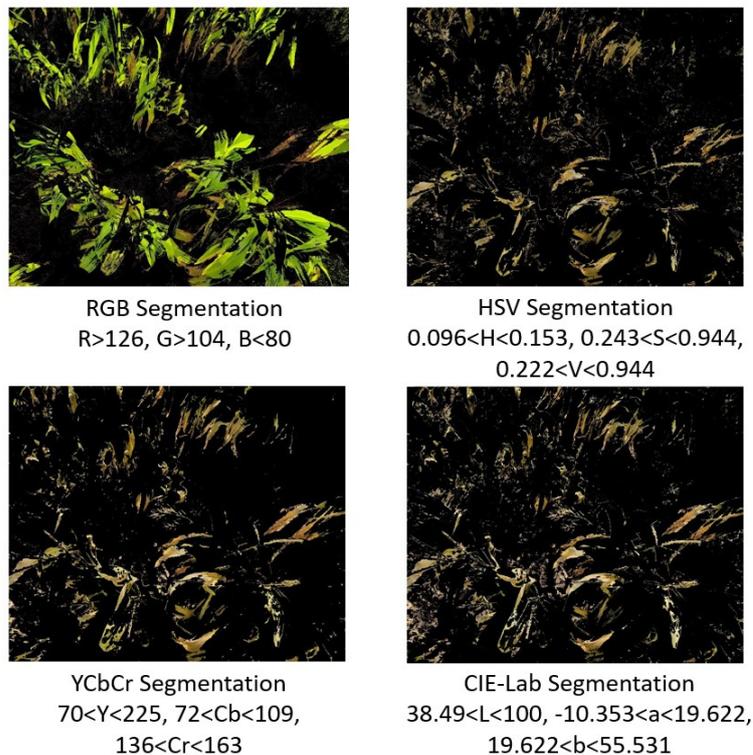


Figure 16. Segmentation methods with thresholds.

With the obtained thresholds, from the segmentation methods, a MATLAB code runs to analyze the images obtained from the flights made over the corn fields. First, an image is selected and then subjected to the thresholds of each method to identify dry leaves in the field.

Figure 17 displays the segmented images using the RGB, HSV, YCbCr and CIE-Lab methods. From the images obtained after applying the segmentation methods with the thresholds proposed in the previous section, it can be seen that using RGB segmentation to detect dry leaves is not very convenient as only some points were highlighted, while in the HSV segmentation method, a greater detection of dry leaves can be noted in the image of the crop; however, these highlighted points look like brush strokes, and the plant is not well distinguished. In the YCbCr segmentation, the dry leaves can be better identified in the plot; however, when it is compared with the original image, there are parts of the dry leaves of the plant that despite being dry, do not appear in the processed image. Finally, the

best detection is obtained when the CIE-Lab segmentation method is applied to the image, although it also detects a little of the soil, which does not represent a relevant issue.

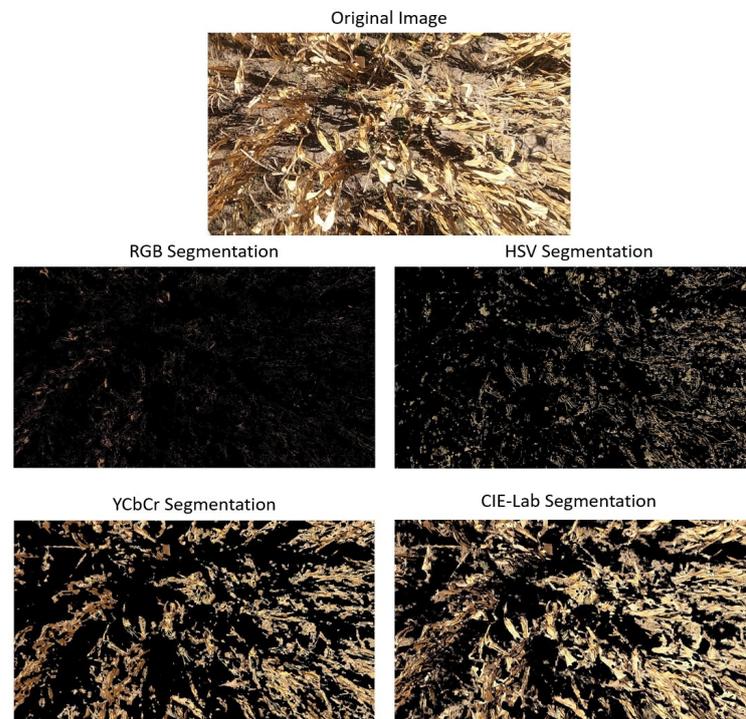


Figure 17. Segmentation methods applied.

The robustness provided to the proposed controller is necessary to improve the closed loop performance of the vehicle in outdoor environment. This improved performance is a crucial issue to the image capturing while the trajectory tracking task is executed by the UAV, allowing reduce distorted capture of the crops. In fact, when a PD controller is used to control the vehicle in order to capture the crop images, from ten experiments only six were satisfactory to stabilize the drone with the mounted camera. In contrast, for the robust backstepping control approach, all the tests were satisfactory, despite a gimbal or extra software not being used.

Furthermore, each processed image has an (x, y) coordinate, which is obtained with the real-time clock ('Time stamp' function) of the Pixhawk autopilot. Then, when an image with affected leaves is detected, the user obtains the relative position (x, y) of the chosen reference frame, and then the user can apply a corrective action, programming a trajectory whose final point corresponds to these coordinates.

7. Conclusions

The quadrotor trajectory tracking problem, in outdoor environment, is addressed by the proposed robust control scheme. The four controllers are synthesized using the robust backstepping approach, it uses virtual bounded inputs (the function $\sin(\cdot)$), which produce bounded control signals, and it is appropriate to the physical constrains of the UAV. The convergence for altitude, translational and rotational UAV variables is guaranteed when the vehicle is subjected to vanishing disturbances. The proposed control algorithm achieves a higher performance in closed loop when is compared to a linearized model-based PD controller. This improved performance is a crucial issue to the image capturing while a trajectory tracking task is executed by the UAV, allowing to reduce the wrong capture of the crops, despite a gimbal or extra software not being used. In fact, the experimental results validate and confirm the aforementioned higher performance level of the proposed control algorithm. Moreover, the proposal presents a potential advantage in precision agriculture as it can prevent crop losses by identifying crop areas with dry leaves. In contrast to the

most of the reported works, our proposal was tested in a precision agriculture task on a real maize crop. Future work includes onboard image processing and to implement new image segmentation techniques.

Author Contributions: Conceptualization, L.R.-G. and O.-J.S.-S.; methodology, L.R.-G., O.-J.S.-S., H.R.-T. and J.-P.O.-O.; software, A.B.-M. and O.G.-P.; validation, A.B.-M., M.-O.O.-O. and O.G.-P.; formal analysis, L.R.-G. and O.-J.S.-S.; investigation, L.R.-G. and O.-J.S.-S.; writing—original draft preparation, writing—review and editing, all authors; project administration, funding acquisition, L.R.-G. and O.-J.S.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data have been included in this paper.

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

References

1. Thanh, H.L.N.N.; Hong, S.K. Quadcopter robust adaptive second order sliding mode control based on PID sliding surface. *IEEE Access* **2018**, *6*, 66850–66860. [[CrossRef](#)]
2. Ononiwu, G.; Onojo, O.; Ozioko, O.; Nosiri, O. Quadcopter design for payload delivery. *J. Comput. Commun.* **2016**, *4*, 1–12. [[CrossRef](#)]
3. Duggal, V.; Sukhwani, M.; Bipin, K.; Syamasundar Reddy, G.; Madhava Krishna, K. Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5121–5127.
4. Mogili, U.R.; Deepak, B. Review on application of drone systems in precision agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [[CrossRef](#)]
5. Kuantama, E.; Tarca, R.; Dzitac, S.; Dzitac, I.; Vesselenyi, T.; Tarca, I. The design and experimental development of air scanning using a sniffer Quadcopter. *Sensors* **2019**, *19*, 3849. [[CrossRef](#)] [[PubMed](#)]
6. Rudin, K.; Hua, M.D.; Ducard, G.; Bouabdallah, S. A robust attitude controller and its application to quadrotor helicopters. *IFAC Proc. Vol.* **2011**, *44*, 10379–10384. [[CrossRef](#)]
7. Ramirez-Rodriguez, H.; Parra-Vega, V.; Sanchez-Orta, A.; Garcia-Salazar, O. Robust backstepping control based on integral sliding modes for tracking of quadrotors. *J. Intell. Robot. Syst.* **2014**, *73*, 55–66. [[CrossRef](#)]
8. Peng, C.; Bai, Y.; Gong, X.; Gao, Q.; Zhao, C.; Tian, Y. Modeling and robust backstepping sliding mode control with Adaptive RBFNN for a novel coaxial eight-rotor UAV. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 56–64.
9. Zhao, Y.; Sun, X.; Wang, G.; Fan, Y. Adaptive Backstepping Sliding Mode Tracking Control for Underactuated Unmanned Surface Vehicle With Disturbances and Input Saturation. *IEEE Access* **2021**, *9*, 1304–1312. [[CrossRef](#)]
10. Kim, N.S.; Kuc, T.Y. Sliding Mode Backstepping Control for Variable Mass Hexa-Rotor UAV. In Proceedings of the 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea, 13–16 October 2020; pp. 873–878.
11. García, O.; Ordaz, P.; Santos-Sánchez, O.J.; Salazar, S.; Lozano, R. Backstepping and Robust Control for a Quadrotor in Outdoors Environments: An Experimental Approach. *IEEE Access* **2019**, *7*, 40635–40648. [[CrossRef](#)]
12. Colorado, J.D.; Cera-Bornacelli, N.; Caldas, J.S.; Petro, E.; Rebolledo, M.C.; Cuellar, D.; Calderon, F.; Mondragon, I.F.; Jaramillo-Botero, A. Estimation of nitrogen in rice crops from UAV-captured images. *Remote Sens.* **2020**, *12*, 3396. [[CrossRef](#)]
13. Cabecinhas, D.; Cunha, R.; Silvestre, C. A nonlinear quadrotor trajectory tracking controller with disturbance rejection. *Control Eng. Pract.* **2014**, *26*, 1–10. [[CrossRef](#)]
14. Vallejo-Alarcón, M.A.; Castro-Linares, R. Robust backstepping control for highly demanding quadrotor flight. *Control Eng. Appl. Inform.* **2020**, *22*, 51–62.
15. Aboudonia, A.; El-Badawy, A.; Rashad, R. Active anti-disturbance control of a quadrotor unmanned aerial vehicle using the command-filtering backstepping approach. *Nonlinear Dyn.* **2017**, *90*, 581–597. [[CrossRef](#)]
16. Zhang, J.; Gu, D.; Deng, C.; Wen, B. Robust and adaptive backstepping control for hexacopter UAVs. *IEEE Access* **2019**, *7*, 163502–163514. [[CrossRef](#)]
17. Dhadekar, D.D.; Sanghani, P.D.; Mangrulkar, K.; Talole, S. Robust control of quadrotor using uncertainty and disturbance estimation. *J. Intell. Robot. Syst.* **2021**, *101*, 1–21. [[CrossRef](#)]
18. Xuan-Mung, N.; Hong, S.K.; Nguyen, N.P.; Le, T.L. Autonomous quadcopter precision landing onto a heaving platform: New method and experiment. *IEEE Access* **2020**, *8*, 167192–167202. [[CrossRef](#)]
19. Derrouaoui, S.H.; Bouzid, Y.; Guiatni, M. Nonlinear robust control of a new reconfigurable unmanned aerial vehicle. *Robotics* **2021**, *10*, 76. [[CrossRef](#)]

20. de Morais, G.A.; Marcos, L.B.; Bueno, J.N.A.; de Resende, N.F.; Terra, M.H.; Grassi, V., Jr. Vision-based robust control framework based on deep reinforcement learning applied to autonomous ground vehicles. *Control Eng. Pract.* **2020**, *104*, 104630. [[CrossRef](#)]
21. Castillo, P.; Muñoz, L.; Santos, O. Robust control algorithm for a rotorcraft disturbed by crosswind. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 756–763. [[CrossRef](#)]
22. Li, C.; Zhang, Y.; Li, P. Full control of a quadrotor using parameter-scheduled backstepping method: Implementation and experimental tests. *Nonlinear Dyn.* **2017**, *89*, 1259–1278. [[CrossRef](#)]
23. Mokhtari, M.R.; Cherki, B. A new robust control for minirotorcraft unmanned aerial vehicles. *ISA Trans.* **2015**, *56*, 86–101. [[CrossRef](#)] [[PubMed](#)]
24. Mejias, L.; Diguët, J.P.; Dezan, C.; Campbell, D.; Kok, J.; Coppin, G. Embedded computation architectures for autonomy in Unmanned Aircraft Systems (UAS). *Sensors* **2021**, *21*, 1115. [[CrossRef](#)] [[PubMed](#)]
25. López-Labra, H.A.; Santos-Sánchez, O.J.; Rodríguez-Guerrero, L.; Ordaz-Oliver, J.P.; Cuvas-Castillo, C. Experimental results of optimal and robust control for uncertain linear time-delay systems. *J. Optim. Theory Appl.* **2019**, *181*, 1076–1089. [[CrossRef](#)]
26. Khalil, H.K. *Nonlinear Systems*, 3rd ed.; Prentice Hall: Hoboken, NJ, USA, 1996.
27. Hock, J.; Kranz, J.; Renfou, B. Studies on the epidemiology of the tar spot disease complex of maize in Mexico. *Plant Pathol.* **1995**, *44*, 490–502. [[CrossRef](#)]
28. Velusamy, P.; Rajendran, S.; Mahendran, R.K.; Naseer, S.; Shafiq, M.; Choi, J.G. Unmanned Aerial Vehicles (UAV) in precision agriculture: Applications and challenges. *Energies* **2021**, *15*, 217. [[CrossRef](#)]
29. Kitpo, N.; Inoue, M. Early rice disease detection and position mapping system using drone and IoT architecture. In Proceedings of the 2018 12th South East Asian Technical University Consortium (SEATUC), Yogyakarta, Indonesia, 12–13 March 2018; Volume 1, pp. 1–5.
30. Görlich, F.; Marks, E.; Mahlein, A.K.; König, K.; Lottes, P.; Stachniss, C. UAV-based classification of cercospora leaf spot using RGB images. *Drones* **2021**, *5*, 34. [[CrossRef](#)]
31. Castillo, P.; Lozano, R.; Dzul, A. *Modelling and Control of Mini-Flying Machines*, 1st ed.; Springer: London, UK, 2005.
32. Lozano, R. *Unmanned Aerial Vehicles: Embedded Control*, 1st ed.; Wiley-ISTE: London, UK, 2010.
33. Bouabdallah, S.; Siegwart, R. Full control of a quadrotor. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 153–158.
34. Svacha, J.; Mohta, K.; Kumar, V. Improving quadrotor trajectory tracking by compensating for aerodynamic effects. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 860–866.
35. Santos, O.; Romero, H.; Salazar, S.; Garcia, O. Optimized Discrete Control Law for Quadrotor Stabilization: Experimental Results. *J. Intell. Robot. Syst.* **2016**, *84*, 67–81. [[CrossRef](#)]
36. GoPro-Cameras. GoPro Inc. June 2022. Available online: <https://gopro.com/en/us/shop/cameras/hero8-black/CHDHX801-master.html> (accessed on 30 June 2022).
37. Kaganami, H.G.; Beiji, Z. Region-based segmentation versus edge detection. In Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12–14 September 2009; pp. 1217–1221.