Robust Control of UAV with Disturbances and Uncertainty Estimation

Domenico Bianchi [1,2,*] , Stefano Di Gennaro [1,2] , Mario Di Ferdinando [1,2] and Cuauhtémoc Acosta Lùa [2,3]

1   Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Via Vetoio, Loc. Coppito, 67100 L'Aquila, Italy
2   Center of Excellence DEWS, University of L'Aquila, Via Vetoio, Loc. Coppito, 67100 L'Aquila, Italy
3   Centro Universitario de la Ciénega, Universidad de Guadalajara, Av. Universidad No. 1115, Col. Lindavista, Ocotlán 47820, Jalisco, Mexico
*   Correspondence: domenico.bianchi@univaq.it

Abstract: In this work, a nonlinear estimator-based robust controller is designed for the position and yaw control of a quadrotor with uncertainty estimation. This controller ensures the tracking of desired references in the presence of parameters variation and external disturbances, making use of high-order sliding mode (HOSM) estimators to estimate these perturbations that can be canceled by the control, thus improving the dynamic behavior of the controlled system. Its performance is evaluated making use of a Simcenter Amesim quadrotor based on physical models generated from experimental data in a co-simulation framework with Matlab–Simulink used to implement the designed controller with FPGA implementation. A challenging and generic maneuver with time-varying wind disturbances and uncertainty model parameters is considered.

Keywords: UAV control; robust control; estimators

1. Introduction

In recent years UAVs have attracted attention due to the vast range of applications that can be addressed, and they have been widely utilized by researchers, law enforcement and security, delivery companies, farmers, etc. Furthermore, UAVs are evolving from being only important for the sensing action [1] to being fundamental also for the actuator action. In the upcoming years, there will therefore be a remarkable development of drones for functional applications such as payload delivery services to transport deformable linear objects such as hoses and goods to carry organs for organ transplants, which helped reduce car journeys to just a few minutes of flight. Furthermore, a recent study showed that the delivery of UAVs could even help reduce greenhouse gas emissions caused by the freight industry [2]. While the application interest of UAVs is growing, on the other hand, there is a need to make the system more and more efficient. The main limitation that occurs in all applications, however, is the limited amount of energy that batteries or other energy sources provide. In fact, in the drones on the market, the average flight time is between 15 and 45 min, which is a time that highly limits the real application. Improvement and optimization of the source energy, lightening of the drone, optimization of the path and improvement of the performance in the presence of disturbances that modify its behavior and significantly decrease performance from an energy point of view are some of the actions investigated to solve the problem. The execution of trajectories that allow for lower consumption is important. In fact, one of the factors that often influences the consumption is linked to disturbances, such as wind, which leads to a departure from the ideal trajectory, and it is then necessary to compensate through additional time and higher battery consumption. In the same way, the model on which the control is based is an approximation, the parameters are time-varying, and it is therefore necessary to intervene also on the parametric uncertainties. It is a challenge that must be faced from different points of view and that requires a

multidisciplinary effort in order to achieve the goal. Motivated by the European project COMP4DRONES (see [3]), it was necessary to study methodologies for controlling the flight of a drone in precision agriculture, to monitor the health status of plants, and possibly spray fertilizer to the plants with time limits, trying to minimize the energy and to follow desired trajectory in presence of disturbances and parametric uncertainties. This represents a complex problem with different objectives. In [4], a robust flight controller based on linear active disturbance rejection control is proposed for the stability control of an aerial robot quadrotor under wind gusts. The nonlinear dynamical model of the quadrotor, considering the wind disturbance, is firstly established through Newton–Euler method. In this control scheme, the linear extended state observer serves as a compensator which can effectively reject the wind gusts. In this case, the disturbance due to the wind is therefore compensated, but a perfect knowledge of the model and its parameters is assumed. In [5], a robust dynamic sliding mode control algorithm using a nonlinear disturbance observer for system dynamics is investigated. The proposed method is applied to provide a rapid adaptation and strictly robust performance for the attitude and altitude control of UAVs. The procedure of the proposed method consists of two stages. First, a nonlinear disturbance observer is applied to estimate the exogenous perturbation. Second, a robust dynamic sliding mode controller integrated with the estimated values of disturbances is presented by a combination of a PID sliding surface and super twisting technique to compensate for the effect of these perturbations on the system. Again, external noise is considered but not model-related uncertainties. Furthermore, the control is tested on an ideal plant. In [6], a novel extension of the classic Nonlinear Dynamic Inversion (NLDI) control architecture for wind disturbance rejection is designed by using adaptive artificial neural networks. In [7], an adaptive controller for a quadrotor UAV for carrying unknown payloads while tracking any trajectory is proposed. The proposed adaptive controller is robust to modeling uncertainties and does not require any a priori knowledge of the bounds of the uncertainties. The controller is also robust to time-varying delays without any constraint on the derivative of the time delay. The simulations are conducted on the simple basic model and do not take into account possible disturbances, such as wind, which influence performance in a non-negligible way. In [8], the deep reinforcement learning method, and robust deep deterministic policy gradient, are proposed for developing a controller that allows robust flying of an unmanned aerial vehicle in dynamic uncertain environments. While a very interesting work, it is tested in a simplified scenario, and the wind and the parametric uncertainties of the model are not directly considered. Furthermore, there is no formal proof of the stability of the controlled system.

The contribution of the paper is dual: on the one hand, it designs a controller that is robust both to the uncertainties of the model and to external disturbances such as wind or aerodynamic factors using HOSM estimators; on the other side, this controller is tested in a co-simulation environment between Matlab–Simulink and Simcenter Amesim on FPGA. In fact, to our best knowledge, results concerning the quadrotor tracking problem in the case of variations of all the parameters emerging in the dynamics and in the presence of environmental disturbances and/or unmodeled dynamics influencing dynamics of the UAV are not available in the research literature. Moreover, after preliminary tests on Matlab–Simulink implementation, that have not been included in this paper, the proposed controller is tested on the quadrotor model provided by the Simcenter Amesim developed by Siemens, based on physical models, in a co-simulation environment with FPGA implementation. As a matter of fact, Simcenter Amesim is proven to accurately represent the dynamic behavior of the drone, it was extensively validated with experimental tests, and it is used by important aerospace enterprises worldwide. In fact, the Simcenter Amesim drone is characterized by components with physical models, unmodeled dynamics, time-varying parameters, disturbances, and parametric uncertainties that make the problem complicated and validate the research conducted meaningfully. The work is organized as follows. In Section 2, the quadrotor mathematical model is presented. The estimators of disturbances and a nonlinear controller to ensure robustness and asymptotic convergence are designed in

Section 3. In Section 4, the performance of the controller and estimators are presented with respect to Matlab and Simcenter Amesim co-simulation environment implementation with FPGA implementation. Some considerations and future research developments conclude the work in Section 5.

2. Mathematical Model of A Quadrotor

The quadrotor considered in this work consists of a rigid frame equipped with four rotors (see [9,10] for a detailed description). The rotors generate the propeller force $F_i = b \cdot \omega_{p,i}^2$, which is proportional to the propeller angular velocity $\omega_{p,i}$, $i = 1, 2, 3, 4$. Propellers 1 and 3 rotate counterclockwise; propellers 2 and 4 rotate clockwise. Quadrotor orientation with Euler angles is used. Let us indicate with $RC(O, e_1, e_2, e_3)$, $R\Gamma(\Omega, \varepsilon_1, \varepsilon_2, \varepsilon_3)$ the frames fixed with the Earth and the quadrotor, respectively, and Ω is coincident with the center of mass of the quadrotor (see Figure 1).
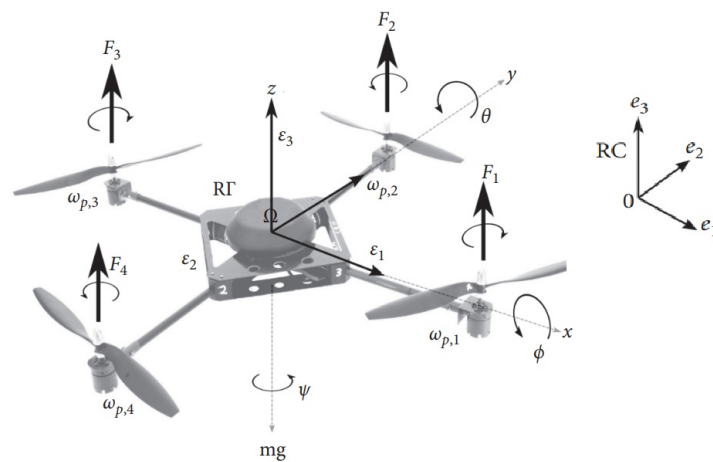

Figure 1. Quadrotor orientation using Euler angles.

The absolute position of the quadrotor in RC is described by $p = (x, y, z)^T$, whereas its attitude is described by the Euler angles $\alpha = (\phi, \theta, \psi)^T$, with $\phi \in [-\pi/2, \pi/2)$, $\theta \in (-\pi/2, \pi/2)$, $\psi \in [-\pi, \pi)$ the roll, pitch, yaw angles, respectively. The sequence 3–2–1 is here considered [11]. Moreover, $v = (v_x, v_y, v_z)^T$, $\omega = (\omega_1, \omega_2, \omega_3)^T$ are the linear and angular velocities of the center of mass of the quadrotor, expressed in RC and in $R\Gamma$, respectively. The translation dynamics, expressed in RC, and rotation dynamics, expressed in $R\Gamma$, of the quadrotor are

$$
\begin{aligned}
\dot{p} &= v \\
\dot{v} &= \frac{1}{m}\left(f_p + f_g + f_d\right) \\
\dot{\alpha} &= M(\alpha)\omega \\
\dot{\omega} &= J^{-1}\left(-\tilde{\omega} J\omega + \tau_p + \tau_{gy} + \tau_d\right)
\end{aligned}
\tag{1}
$$

where m is the mass of the quadrotor, $f_p = \mathcal{R}(\alpha)\varphi_p$ is the force in RC excerted by the propellers (φ_p is the same force, but expressed in $R\Gamma$), J (positive definite symmetric matrix in $\mathbb{R}^{3\times3}$, expressed in $R\Gamma$) is the inertia matrix of the quadrotor, and

$$
\tilde{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}
\tag{2}
$$

is the so–called dyadic representation of ω. Furthermore,

$$\varphi_p = \begin{pmatrix} 0 \\ 0 \\ u_p \end{pmatrix}, \quad u_p = \sum_{i=1}^{4} F_i, \quad f_g = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{3}$$

$$\tau_p = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} \ell\, b(\omega_{p,2}^2 - \omega_{p,4}^2) \\ \ell\, b(\omega_{p,3}^2 - \omega_{p,1}^2) \\ c(\omega_{p,1}^2 - \omega_{p,2}^2 + \omega_{p,3}^2 - \omega_{p,4}^2) \end{pmatrix} \tag{4}$$

are the input forces and moments produced by the propellers (expressed in $R\Gamma$), where ℓ is the distance between the center of mass CG to the rotor shaft, and b ($[b]=\frac{N\,s^2}{rad^2}$) and c ($[c]=\frac{N\,s^2\,m}{rad^2}$) are, respectively, the so–called thrust and drag factor. Obviously,

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \ell & 0 & -\ell \\ -\ell & 0 & \ell & 0 \\ c & -c & c & -c \end{pmatrix}^{-1} \begin{pmatrix} u_p \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix}, \quad \begin{pmatrix} \omega_{p,1}^2 \\ \omega_{p,2}^2 \\ \omega_{p,3}^2 \\ \omega_{p,4}^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{4b} & 0 & \frac{-1}{2b\ell} & \frac{1}{4c} \\ \frac{1}{4b} & \frac{1}{2b\ell} & 0 & \frac{-1}{4c} \\ \frac{1}{4b} & 0 & \frac{1}{2b\ell} & \frac{1}{4c} \\ \frac{1}{4b} & \frac{-1}{2b\ell} & 0 & \frac{-1}{4c} \end{pmatrix} \begin{pmatrix} u_p \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix}, \tag{5}$$

$$\omega_{p,i} = \sqrt{\frac{F_i}{b}}, \quad 0 \le F_i \le F_{i,max}, \quad 0 \le \omega_{p,i} \le \omega_{p,i,max}, \quad i = 1,2,3,4$$

where $F_{i,max}$ and $\omega_{p,i,max}$ are, respectively, the maximum forces and the angular velocities related to each propellers due to physical limitations. Moreover, f_g in (3) is the force due to the gravity, which is expressed in RC. The vectors expressed in $R\Gamma$ are transformed into vectors in RC by the rotation matrix

$$\mathcal{R}_{ib} = \mathcal{R}(\alpha) = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \tag{6}$$

where $c_\star = \cos(\star)$, $s_\star = \sin(\star)$, $\star = \phi, \theta, \psi$. The angular velocity dynamics are expressed using the matrix

$$M(\alpha) = \begin{pmatrix} 1 & s_\phi \mathrm{tg}_\theta & c_\phi \mathrm{tg}_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi \mathrm{sc}_\theta & c_\phi \mathrm{sc}_\theta \end{pmatrix}$$

with $\mathrm{tg}_\star = \tan(\star)$, $\mathrm{sc}_\star = \sec(\star)$, $\star = \phi, \theta, \psi$. Under the assumption of small angles ϕ (roll) and θ (pitch), which is acceptable in the case of a quadrotor performing non-aggressive maneuvers, this matrix can be approximated by the identity matrix, i.e., $M(\alpha) \simeq I_{3x3}$ [12]. The rolling torque τ_1 is produced by the forces F_2 and F_4. Similarly, the pitching torque τ_2 is produced by the forces F_1 and F_3. Due to the Newton's third law, the propellers produce a yawing torque τ_3 on the body of the quadrotor in the opposite direction of the propeller rotation. Moreover, $\tau_{gy} = \sum_{i=1}^{4}(-1)^i J_{p,i} \omega_{p,i} \tilde{\omega} \epsilon_3$ is the gyroscopic torque due to the propeller rotations, with $J_{p,i}$, $i = 1,2,3,4$, the inertia moment of the i_{th} motor plus propeller, with respect to its rotation axis. Finally, f_d and τ_d are the forces and torques due to the external disturbances, for example due to aerodynamic forces. Considering the

nominal values m°, $J^\circ_{p,i} = J^\circ_p$, $i = 1, 2, 3, 4$, $J^\circ = \text{diag}\{J^\circ_1, J^\circ_2, J^\circ_3\}$, f°_p, τ°_p, system (1) can be rewritten as

$$\dot{p} = v$$

$$\dot{v} = \frac{1}{m^\circ}f^\circ_p + \frac{1}{m}f_g + d$$

$$\dot{\alpha} = \omega \tag{7}$$

$$\dot{\omega} = -J^{\circ-1}\tilde{\omega}J^\circ\omega + J^{\circ-1}\tau^\circ_p + J^{\circ-1}\tau^\circ_{gy} + \delta$$

where

$$d = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} = \left(\frac{1}{m} - \frac{1}{m^\circ}\right)f_p + \frac{1}{m}f_d + \frac{1}{m^\circ}\left(f_p - f^\circ_p\right)$$

$$\delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} = -\left(J^{-1}\tilde{\omega}J\omega - J^{\circ-1}\tilde{\omega}J^\circ\omega\right) + \left(J^{-1} - J^{\circ-1}\right)\tau_p + \left(J^{-1}\tau_{gy} - J^{\circ-1}\tau^\circ_{gy}\right) + J^{-1}\tau_d + \tag{8}$$

$$+ J^{\circ-1}(\tau_p - \tau^\circ_p)$$

are the perturbations acting on the drone, due to parametric uncertainties and external disturbances, and

$$\tau^\circ_{gy} = \sum_{i=1}^{4}(-1)^i J^\circ_p \omega_{p,i}\tilde{\omega}\epsilon_3 = J^\circ_p \omega_p \begin{pmatrix} \omega_2 \\ -\omega_1 \\ 0 \end{pmatrix} \tag{9}$$

is the nominal gyroscopic torque due to the propeller rotations, with $\omega_p = -\omega_{p,1} + \omega_{p,2} - \omega_{p,3} + \omega_{p,4}$ the so-called rotor relative speed. With these positions, mathematical model (7) of the quadrotor can be expressed as

$$\dot{x} = v_x, \quad \dot{y} = v_y, \quad \dot{z} = v_z$$

$$\dot{v}_x = \left(c_\phi s_\theta c_\psi + s_\phi s_\psi\right)\frac{u^\circ_p}{m^\circ} + d_x$$

$$\dot{v}_y = \left(c_\phi s_\theta s_\psi - s_\phi c_\psi\right)\frac{u^\circ_p}{m^\circ} + d_y$$

$$\dot{v}_z = -g + c_\phi c_\theta \frac{u^\circ_p}{m^\circ} + d_z$$

$$\dot{\phi} = \omega_1, \quad \dot{\theta} = \omega_2, \quad \dot{\psi} = \omega_3 \tag{10}$$

$$\dot{\omega}_1 = \frac{J^\circ_2 - J^\circ_3}{J^\circ_1}\omega_2\omega_3 + \frac{J^\circ_p}{J^\circ_1}\omega_p\omega_2 + \frac{1}{J^\circ_1}\tau^\circ_1 + \delta_1$$

$$\dot{\omega}_2 = \frac{J^\circ_3 - J^\circ_1}{J^\circ_2}\omega_1\omega_3 - \frac{J^\circ_p}{J^\circ_2}\omega_p\omega_1 + \frac{1}{J^\circ_2}\tau^\circ_2 + \delta_2$$

$$\dot{\omega}_3 = \frac{J^\circ_1 - J^\circ_2}{J^\circ_3}\omega_1\omega_2 + \frac{1}{J^\circ_3}\tau^\circ_3 + \delta_3.$$

Remark 1. *Within the vectors d and δ, there are the variations of mass m, moments of inertia, thrust coefficient b and drag coefficient c linked to the propulsion force F_p, as well as possible external perturbations and unmodeled dynamics that are considered in F_d and τ_d, so as to incorporate the parametric variations, the unmodeled dynamics and external perturbations (as wind) that act both at the level of linear and angular velocity level (see Section 4.2).*

Assumption 1. *f_p and τ_d, and their derivatives, are bounded.*

3. Design of Nonlinear Controller

The output vector to be controlled is composed by the 3D position and yaw rate and the control aim is to design a controller, in the presence of disturbances and parameters

uncertainty, to ensure the asymptotic converge of the variables $\chi = (x, y, z, \psi)$ to some reference trajectories $\chi_r = (x_r, y_r, z_r, \psi_r)$. To this aim, let us consider the kinematic errors

$$
e_\chi = \chi - \chi_r = \begin{pmatrix} x - x_r \\ y - y_r \\ z - z_r \\ \psi - \psi_r \end{pmatrix} = \begin{pmatrix} e_x \\ e_y \\ e_z \\ e_\psi \end{pmatrix}
$$

whose dynamics are

$$
\dot{e}_\chi = \begin{pmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \\ \dot{e}_\psi \end{pmatrix} = \begin{pmatrix} v_x - \dot{x}_r \\ v_y - \dot{y}_r \\ v_z - \dot{z}_r \\ \omega_3 - \dot{\psi}_r \end{pmatrix} = \begin{pmatrix} e_{v_x} \\ e_{v_y} \\ e_{v_z} \\ e_{\omega_3} \end{pmatrix}
$$

and with

$$
\ddot{e}_\chi = \begin{pmatrix} \ddot{e}_x \\ \ddot{e}_y \\ \ddot{e}_z \\ \ddot{e}_\psi \end{pmatrix} = \begin{pmatrix} \dot{v}_x - \ddot{x}_r \\ \dot{v}_y - \ddot{y}_r \\ \dot{v}_z - \ddot{z}_r \\ \dot{\omega}_3 - \ddot{\psi}_r \end{pmatrix} = \begin{pmatrix} \dot{e}_{v_x} \\ \dot{e}_{v_y} \\ \dot{e}_{v_z} \\ \dot{e}_{\omega_1} \end{pmatrix} = \begin{pmatrix} \left(c_\phi s_\theta c_\psi + s_\phi s_\psi\right)\dfrac{u_p^\circ}{m^\circ} + d_x - \ddot{x}_r \\ \left(c_\phi s_\theta s_\psi - s_\phi c_\psi\right)\dfrac{u_p^\circ}{m^\circ} + d_y - \ddot{y}_r \\ -g + c_\phi c_\theta \dfrac{u_p^\circ}{m^\circ} + d_z - \ddot{z}_r \\ \dfrac{J_1^\circ - J_2^\circ}{J_3^\circ}\omega_1\omega_2 + \dfrac{1}{J_3^\circ}\tau_3^\circ + \delta_3 - \ddot{\psi}_r \end{pmatrix} =
$$

$$
= \begin{pmatrix} \left(c_\phi s_{\theta_r} c_\psi + s_{\phi_r} s_\psi\right)\dfrac{u_p^\circ}{m^\circ} + d_x - \ddot{x}_r \\ \left(c_\phi s_{\theta_r} s_\psi - s_{\phi_r} c_\psi\right)\dfrac{u_p^\circ}{m^\circ} + d_y - \ddot{y}_r \\ -g + c_\phi c_\theta \dfrac{u_p^\circ}{m^\circ} + d_z - \ddot{z}_r \\ \dfrac{J_1^\circ - J_2^\circ}{J_3^\circ}\omega_1\omega_2 + \dfrac{1}{J_3^\circ}\tau_3^\circ + \delta_3 - \ddot{\psi}_r \end{pmatrix} + \begin{pmatrix} \left((s_\theta - s_{\theta_r})c_\phi c_\psi + (s_\phi - s_{\phi_r})s_\psi\right)\dfrac{u_p^\circ}{m^\circ} \\ \left((s_\theta - s_{\theta_r})c_\phi s_\psi - (s_\phi - s_{\phi_r})c_\psi\right)\dfrac{u_p^\circ}{m^\circ} \\ 0 \\ 0 \end{pmatrix}.
$$

Setting

$$
\begin{pmatrix} s_{\phi_r} \\ s_{\theta_r} \end{pmatrix} = \frac{1}{\sigma}\begin{pmatrix} c_\phi c_\theta s_\psi & -c_\phi c_\theta c_\psi \\ c_\theta c_\psi & c_\theta s_\psi \end{pmatrix}\begin{pmatrix} \ddot{x}_r - \hat{d}_x + w_x \\ \ddot{y}_r - \hat{d}_y + w_y \end{pmatrix}
$$

$$
u_p^\circ = \frac{m^\circ}{c_\phi c_\theta}\left(\ddot{z}_r + g - \hat{d}_z + w_z\right)
$$

$$
\tau_3^\circ = J_3^\circ\left(\ddot{\psi}_r - \frac{J_1^\circ - J_2^\circ}{J_3^\circ}\omega_1\omega_2 - \hat{\delta}_3 + w_\psi\right) \tag{11}
$$

$$
w_j = -k_{j,0}I_{e_j} - k_{j,1}e_j - k_{j,2}e_{v_j}, \qquad j = x, y, z
$$

$$
w_\psi = -k_{\psi,0}I_{e_\psi} - k_{\psi,1}e_\psi - k_{\psi,2}e_{\omega_3}
$$

$$
\dot{I}_{e_j} = e_j, \qquad j = x, y, z, \psi
$$

where $\sigma = \ddot{z}_r + g - \hat{d}_z + w_z$, $k_{j,0}, k_{j,2} > 0$ and $k_{j,1} > k_{j,0}/k_{j,2}$, $j = x, y, z, \psi$, one gets

$$
\ddot{e}_\chi = \begin{pmatrix} w_x \\ w_y \\ w_z \\ w_\psi \end{pmatrix} + \begin{pmatrix} \mu_x \dfrac{u_p^\circ}{m^\circ} \\ \mu_y \dfrac{u_p^\circ}{m^\circ} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} d_x - \hat{d}_x \\ d_y - \hat{d}_y \\ d_z - \hat{d}_z \\ \delta_3 - \hat{\delta}_3 \end{pmatrix},
$$

where

$$\mu_x = [(s_\theta - s_{\theta_r})c_\phi c_\psi + (s_\phi - s_{\phi_r})s_\psi],$$

$$\mu_y = [(s_\theta - s_{\theta_r})c_\phi s_\psi - (s_\phi - s_{\phi_r})c_\psi].$$

As far as the errors $s_\phi - s_{\phi_r}$, $s_\theta - s_{\theta_r}$ are concerned, let us consider the roll error $e_\phi = \phi - \phi_r$ and the pitch error $e_\theta = \theta - \theta_r$, with $\phi_r = \arcsin s_{\phi_r}$, $\theta_r = \arcsin s_{\theta_r}$. Their dynamics are

$$\dot{e}_\phi = \omega_1 - \dot{\phi}_r = e_{\omega_1}, \quad \dot{e}_\theta = \omega_2 - \dot{\theta}_r = e_{\omega_2}$$

$$\dot{e}_{\omega_1} = \frac{J_2^\circ - J_3^\circ}{J_1^\circ}\omega_2\omega_3 + \frac{J_p^\circ}{J_1^\circ}\omega_p\omega_2 + \frac{1}{J_1^\circ}\tau_1^\circ + \delta_1 - \ddot{\phi}_r$$

$$\dot{e}_{\omega_2} = \frac{J_3^\circ - J_1^\circ}{J_2^\circ}\omega_1\omega_3 - \frac{J_p^\circ}{J_2^\circ}\omega_p\omega_1 + \frac{1}{J_2^\circ}\tau_2^\circ + \delta_2 - \ddot{\theta}_r$$

and the controls

$$
\begin{aligned}
\tau_1^\circ &= J_1^\circ\ddot{\phi}_r - (J_2^\circ - J_3^\circ)\omega_2\omega_3 - J_p^\circ\omega_p\omega_2 - J_1^\circ\hat{\delta}_1 + J_1^\circ w_\phi \\
\tau_2^\circ &= J_2^\circ\ddot{\theta}_r - (J_3^\circ - J_1^\circ)\omega_1\omega_3 + J_p^\circ\omega_p\omega_1 - J_2^\circ\hat{\delta}_2 + J_2^\circ w_\theta \\
w_\phi &= -k_{\phi,0}I_{e_\phi} - k_{\phi,1}e_\phi - k_{\phi,2}e_{\omega_1} \\
w_\theta &= -k_{\theta,0}I_{e_\theta} - k_{\theta,1}e_\theta - k_{\theta,2}e_{\omega_2}
\end{aligned}
\tag{12}
$$

with $k_{j,0}, k_{j,2} > 0$, $k_{j,1} > k_{j,2}/k_{j,0}$, $j = \phi, \theta$. Therefore, one obtains

$$
\begin{pmatrix} \ddot{e}_x \\ \ddot{e}_y \\ \ddot{e}_z \end{pmatrix} = \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix} + \begin{pmatrix} \mu_x\dfrac{u_p^\circ}{m^\circ} \\ \mu_y\dfrac{u_p^\circ}{m^\circ} \\ 0 \end{pmatrix} + \begin{pmatrix} d_x - \hat{d}_x \\ d_y - \hat{d}_y \\ d_z - \hat{d}_z \end{pmatrix}
\tag{13a}
$$

$$
\begin{pmatrix} \ddot{e}_\phi \\ \ddot{e}_\theta \\ \ddot{e}_\psi \end{pmatrix} = \begin{pmatrix} w_\phi \\ w_\theta \\ w_\psi \end{pmatrix} + \begin{pmatrix} \delta_1 - \hat{\delta}_1 \\ \delta_2 - \hat{\delta}_2 \\ \delta_3 - \hat{\delta}_3 \end{pmatrix}
\tag{13b}
$$

Remark 2. *For the computation of the first and second derivatives of the signals ϕ and ψ, since it is not possible to have a direct analytical expression, they will be implemented using the high-order sliding mode differentiator (HOSMD) designed in [13].*

Disturbances Estimator

To achieve robustness of the closed-loop system with respect to external disturbances, a robust disturbance estimator is designed. In what follows, given the measured variables

the HOSM estimator wants to converge in finite time to disturbances values, finally, based on [14], one considers,

$$
\begin{aligned}
\dot{v}_x &= \left(c_\phi s_\theta c_\psi + s_\phi s_\psi \right) \frac{u_p^\circ}{m^\circ} - \lambda_{x,1} \lfloor v_x - v_x \rceil^{1/2} + \hat{d}_x \\
\dot{v}_y &= \left(c_\phi s_\theta s_\psi - s_\phi c_\psi \right) \frac{u_p^\circ}{m^\circ} - \lambda_{y,1} \lfloor v_y - v_y \rceil^{1/2} + \hat{d}_y \\
\dot{v}_z &= -g + c_\phi c_\theta \frac{u_p^\circ}{m^\circ} - \lambda_{z,1} \lfloor v_z - v_z \rceil^{1/2} + \hat{d}_z \\
\dot{v}_1 &= \frac{J_2^\circ - J_3^\circ}{J_1^\circ} \omega_2 \omega_3 + \frac{J_p^\circ}{J_1^\circ} \omega_p \omega_2 + \frac{1}{J_1^\circ} \tau_1^\circ - \lambda_{\phi,1} \lfloor v_1 - \omega_1 \rceil^{1/2} + \hat{\delta}_1 \\
\dot{v}_2 &= \frac{J_3^\circ - J_1^\circ}{J_2^\circ} \omega_1 \omega_3 - \frac{J_p^\circ}{J_2^\circ} \omega_p \omega_1 + \frac{1}{J_2^\circ} \tau_2^\circ - \lambda_{\theta,1} \lfloor v_2 - \omega_2 \rceil^{1/2} + \hat{\delta}_2 \\
\dot{v}_3 &= \frac{J_1^\circ - J_2^\circ}{J_3^\circ} \omega_1 \omega_2 + \frac{1}{J_3^\circ} \tau_3^\circ - \lambda_{\psi,1} \lfloor v_3 - \omega_3 \rceil^{1/2} + \hat{\delta}_3 \\
\dot{\hat{d}}_j &= -\lambda_{k,2} \lfloor v_j - v_j \rceil^0 \\
\dot{\hat{\delta}}_i &= -\lambda_{k,2} \lfloor v_i - \omega_i \rceil^0
\end{aligned}
\tag{14}
$$

which ensure the convergence in finite time of the estimation errors $d_j - \hat{d}_j$, $j = x, y, z$, and $\delta_i - \hat{\delta}_i$, $i = 1, 2, 3$, for some $\lambda_{k,1}, \lambda_{k,2}$ with $k = x, y, z, \phi, \theta, \psi$, where $\lfloor \sigma \rceil^{1/2} = |\sigma|^{1/2} \mathrm{sign}(\sigma)$, $\lfloor \sigma \rceil^0 = \mathrm{sign}(\sigma)$ for any scalar signal σ.

Theorem 1. *Given the reference χ_r, bounded along with its derivatives $\dot{\chi}_r$, $\ddot{\chi}_r$, the controller (11), (12), with the gains $\lambda_{\eta,1}, \lambda_{\eta,2} > 0$ with $\eta = x, y, z, \phi, \theta, \psi$, ensures that the error e_χ tends to zero asymptotically and the perturbation estimations $\hat{d}_\eta$, in (14) tends to d_η in finite time, with $\eta = x, y, z, \phi, \theta, \psi$.*

Proof. Since the subsystems (13a) and (13b), with $\begin{pmatrix} d_x - \hat{d}_x & d_y - \hat{d}_y & d_z - \hat{d}_z & \delta_1 - \hat{\delta}_1 & \delta_2 - \hat{\delta}_2 & \delta_3 - \hat{\delta}_3 \end{pmatrix}^T$ as input, is input-to-state stable (in fact, the dynamics matrix of each subsystem is Hurwitz, and the ISS Lyapunov function is quadratic) and the system, for $\eta = x, y, z, \phi, \theta, \psi$,

$$
\begin{aligned}
\xi_\eta &= v_\eta - v_\eta, \quad \tilde{d}_\eta = d_\eta - \hat{d}_\eta, \\
\dot{\xi}_\eta &= -\lambda_{\eta,1} \lfloor \xi_\eta \rceil^{1/2} + \tilde{d}_\eta \\
\dot{\tilde{d}}_\eta &= -\lambda_{\eta,2} \lfloor \xi_\eta \rceil^0
\end{aligned}
$$

converges to the origin in finite time, and the systems (13a) and (13b) is globally uniformly exponentially stable. In fact, considering the part relating to the position on the x-axis χ_x (the demonstrations are analogous for the other variables taken into consideration), it is possible to consider the following Lyapunov function

$$
V_{e,x} = \frac{1}{2} \chi_x^T P_{e,x} \chi_x, \quad \chi_x = \begin{pmatrix} \chi_{x,1} \\ \chi_{x,2} \end{pmatrix} = \begin{pmatrix} \lfloor v_x - v_x \rceil^{1/2} \\ d_x - \hat{d}_x \end{pmatrix}
$$

with

$$
P_{e,x} = P_{e,x}^T = \begin{pmatrix} \lambda_{x,1}^2 + 4\lambda_{x,2} & -\lambda_{x,1} \\ -\lambda_{x,1} & 2 \end{pmatrix} > 0.
$$

since $\lambda_{x,2} > 0$. Notice that $\lambda_{min}^{P_{e,x}}\|\chi_x\|_2^2/2 \leq V_{e,x} \leq \lambda_{max}^{P_{e,x}}\|\chi_x\|_2^2/2$, with $\lambda_{min}^{P_{e,x}}$, $\lambda_{max}^{P_{e,x}}$ the minimum and maximum eigenvalues of $P_{e,x}$. To show the convergence, one computes the derivative of the Lyapunov candidate

$$\dot{V}_{e,x} = -\frac{1}{2\chi_{x,1}}\chi_x^T P_{e,x} A_x \chi_x = -\frac{\lambda_{x,1}}{2\chi_{x,1}}\chi_x^T Q_{e,x}\chi_x$$

where

$$Q_{e,x} = \begin{pmatrix} (\lambda_{x,1}^2 + 2\lambda_{x,2}) & -\lambda_{x,1} \\ -\lambda_{x,1} & 1 \end{pmatrix} > 0$$

$$P_{e,x} A_x = \begin{pmatrix} \lambda_{x,1}^2 + 4\lambda_{x,2} & -\lambda_{x,1} \\ -\lambda_{x,1} & 2 \end{pmatrix}\begin{pmatrix} \lambda_{x,1} & -1 \\ 2\lambda_{x,2} & 0 \end{pmatrix} = \lambda_{x,1} Q_{e,x} > 0$$

since $\lambda_{x,1}, \lambda_{x,2} > 0$. Since $\|\chi_x\|_2^2 = |\xi_x| + \tilde{d}_x^2$, clearly $|\xi_x|^{1/2} \leq \|\chi_x\|_2 \leq \sqrt{2}\, V_{e,x}^{1/2}/\sqrt{\lambda_{min}^{P_{e,x}}}$, so that

$$-\frac{1}{|\xi_x|^{1/2}} \leq -\frac{\sqrt{\lambda_{min}^{P_{e,x}}}}{\sqrt{2}}\frac{1}{V_{e,x}^{1/2}}$$

and

$$\dot{V}_{e,x} = -\frac{\lambda_{x,1}}{2|\xi_x|^{1/2}}\chi_x^T Q_{e,x}\chi_x \leq -\lambda_{min}^{Q_{e,x}}\|\chi_x\|_2^2 \leq -\frac{\lambda_{min}^{Q_{e,x}}}{\lambda_{max}^{P_{e,x}}}2V_{e,x} = -\gamma_{min}^{e,x}V_{e,x}^{1/2}$$

$$\gamma_{min}^{e,x} = \lambda_{x,1}\frac{\sqrt{\lambda_{min}^{P_{e,x}}}}{\sqrt{2}\,\lambda_{max}^{P_{e,x}}}\lambda_{min}^{Q_{e,x}}$$

where

$$\lambda_{min}^{Q_{e,x}} = \frac{\lambda_{x,1}^2 + 2\lambda_{x,2} + 1}{2} - \frac{1}{2}\sqrt{(\lambda_{x,1}^2 + 2\lambda_{x,2} + 1)^2 - 8\lambda_{x,2}}.$$

This implies that ξ_x, $\tilde{d}_x$ converge to zero in finite time for $\lambda_{x,1}, \lambda_{x,2} > 0$, since

$$\frac{\sqrt{\lambda_{min}^{P_{e,x}}}}{\sqrt{2}}\|\chi_x(t)\|_2 \leq V_{e,x}^{1/2}(t_0) - \frac{\gamma_{min}^{e,x}}{2}(t - t_0) \leq \frac{\sqrt{\lambda_{max}^{P_{e,x}}}}{\sqrt{2}}\|\chi_x(t_0)\|_2 - \frac{\gamma_{min}^{e,x}}{2}(t - t_0) = 0$$

for $\bar{t} = t_0 + \sqrt{2\lambda_{max}^{P_{e,x}}}\|\chi_x(t_0)\|_2/\gamma_{min}^{e,x}$. □

The control scheme is divided into an inner and outer loop design. The inner loop design is responsible for attitude and altitude control, while the outer loop is responsible for controlling lateral and longitudinal motion. The desired trajectories for translation and heading movements are provided externally, while the roll and pitch trajectories are generated from the outer loop. The functional scheme of the proposed controller is the one shown in Figure 2.
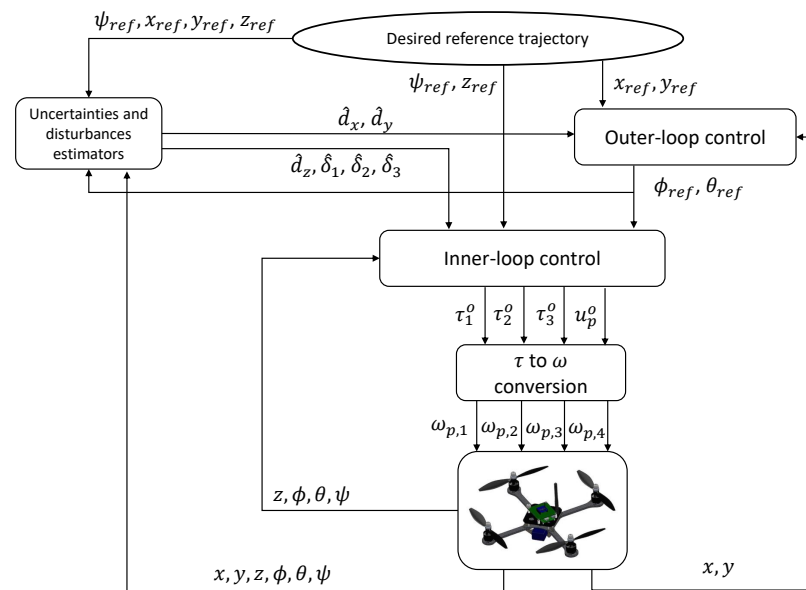
Figure 2. Functional block diagram of the controller.

4. Simulation and Results Analysis

The performance of the proposed controller is discussed in this section. First of all, the simulation environment used is briefly described in Section 4.1; then, the main wind models used in aerospace research are briefly recalled in Section 4.2, and specifically, the one used in the simulations of this paper is described. Finally, the results obtained by the robust controller in a generic flight mission are shown and compared in Section 4.3 with the non-robust controller in the presence of disturbances and parameters uncertainty present in the physical model of the chosen simulation environment.

4.1. Co-Simulation Environment with Matlab-Simulink and Simcenter Amesim with FPGA Implementation

The software Simcenter Amesim (see [15]), a multi-physics system simulation tool developed by Siemens Digital Industries Software, is used to model the different subsystems of the UAV: batteries, propulsion chain and flight dynamics. In general, the tool provides off-the-shelf components available in libraries, covering several physical (fluids, mechanical, electrical, thermal...) and application (aerospace, automotive, gas turbines...) domains. Those subsystem plant models have been favorably compared against experimental data provided by the manufacturers first. The use of this tool was therefore aimed at validating the controller and the estimators, which in the first instance were developed and tested only in the Matlab–Simulink environment. The overall plant model was then integrated in a co-simulation framework capable of modeling drone's navigation sensors, mission environment and algorithms to simulate the drone's behavior under different scenarios. The model present in Simcenter Amesim is complex with respect to the reference mathematical model considered for the design of the controller presented in Section 2; therefore, it contains both unmodeled dynamics and parameters that are time-varying and is therefore a valid test bed to evaluate controller's performance. In the present research, the UAV model considered is seen as a black box model, and the parameters used in the model-based controller have been preliminary identified with Least Square Method. For example, at the modeling level, the aerodynamic coefficients such as the thrust and drag coefficient play an important role in the design of the controller. In fact, several studies have shown that they are variable based on the operating conditions of the drone. In both the physical models used in the advanced, realistic simulator used in this paper (see [15]) and in experimental studies, they are determined by multivariable functions (see [16,17]) such as orientation,

angular and total velocity as well as the geometric characteristics themselves. For these reasons, they are parameters that can be identified in a static way, and this is the practice that can be followed, but they are actually time-varying during the simulation. For this reason, it is necessary to take them into account in the design phase of the controller in order to make it robust, and in our case, all these variations in propulsion level are also considered in the disturbance term. The drone simulation parameters used in the model-based design controller are presented in Table 1.

Table 1. Nominal quadrotor parameters.

m	Mass of the airframe	5 kg
l	Distance of CoG to the rotor shaft	0.3 m
J_1	Inertia in the x-axis	$0.011521 \text{ kg} \cdot \text{m}^2$
J_2	Inertia in the y-axis	$0.0362132 \text{ kg} \cdot \text{m}^2$
J_3	Inertia in the z-axis	$0.029142 \cdot \text{m}^2$
J_p	Inertia of the propellers	$0.0003 \cdot \text{m}^2$
g	Gravity acceleration	$9.81 \text{ m} \cdot \text{s}^{-2}$
b	Trust factor	$4.5625 \times 10^{-5} \text{ N} \cdot \text{s}^2 \cdot \text{rad}^{-2}$
c	Drag factor	$1.375 \times 10^{-5} \text{ N} \cdot \text{s}^2 \cdot \text{rad}^{-2} \cdot \text{m}$

In addition to the use of the physical model of the Siemens drone with identical performance to the real one, a perfect candidate for this application is the FPGA implementation, since it can provide strict real–time execution, high power efficiency, small size, and high reliability, at the cost of a reduced dynamic range. Specifically, the Intel Cyclone 10 LP FPGA was available and corresponded to what was needed; then, it has been used as a reference, and it comes on a minimal board, which is useful for actual power consumption tests and fast prototyping, allowing installation on small UAVs for real-world tests. Without going into the details of the FPGA implementation, Figure 3 shows the structure of the implemented controller, which is only part of a complete system architecture. During the implementation on FPGA, the typical problems related to the operations of trigonometric function, their inverse, reciprocal or division, square root, computation of the derivatives, discrete approximation, variables normalization, as well as to the fixed-point arithmetic of the controller and input variables have been addressed. The simulations presented were carried out in a co-simulation environment between Matlab and Simcenter Amesim in order to have a complex model on which to test the controller and were implemented in FPGA at the end.
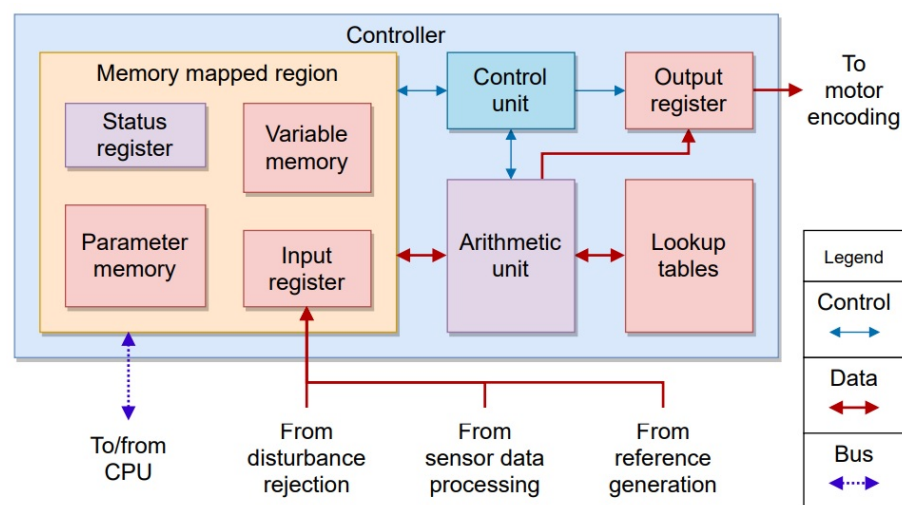
Figure 3. Some detail of the FPGA control architecture.

4.2. Environmental Disturbances Acting on the Quadrotor: Wind Modeling

The categories, features, and mathematical models of the wind, which have a big influence on UAV in the low-altitude condition, can be classified in four groups: constant wind, turbulent flow, many kinds of wind shear, and the propeller vortex (see [18]). The attitude, speed, and position of UAV are sensitive to wind disturbance, and consequently, the wind is seen as a parameter that has the bigger influence on efficiency. The main features of each wind category are shortly evoked below:

- Constant wind deals with the average wind speed in a specified environment. Constant wind does not exist in nature, because it is only the reference value of the wind speed in a given environment. To perform a more realistic simulation test, other types of wind need to be considered. Generally through statistical data, it is possible to obtain its value in the different temporal or spatial conditions. The numerous relevant websites that log wind data are supportive in this regard. Usually, constant wind is used in simulation tests of UAVs at first, but it is unable to reproduce the flight environment correctly.

- Turbulent flow is a continuous random change, which is always conducted by constant wind. The source of turbulent flow is coupled to many elements, such as wind shear, heat exchange, topographical factors, etc. (see [19]). Stochastic Process Theory is commonly used to describe atmospheric turbulence in engineering applications. Turbulent flow models incorporate the Dryden model and the Von Karman model, both of which depend on a vast amount of measurements and statistics (see [20]). The difference is that Dryden's model finds the correlation function of turbulence before giving rise to the spectral function, while Von Karman, on the contrary, first establishes the spectral function and then deduces the correlation function of turbulence. From the aforementioned literature, it emerges that between the two models, there is no relevant difference for the application use in the simulation phase; therefore, both can be used within engineering problems designed to simulate the wind.

- Wind shear The American National Research Council defines the degree of wind shear as the ratio of the difference between two wind vectors at two points and the distance between those two points. There are many sources of wind shear. In UAV flight, common low-altitude wind shear contains frontal wind shear, topography-induced wind shear, microdownburst and night jet wind shear. The frequency of the wind shear is also not as large as the turbulent flow, it changes every few seconds as its maximum frequency. The speed of the wind shear is a function of the spatial position, and its model is relatively simple. In the simulation test, the abrupt change in constant wind speed is used to indicate the occurrence of wind shear. Wind shear is a discrete or deterministic wind speed, which often occurs in a very short time,

and it is a strong atmospheric perturbation. The wind shear model can be divided into several categories based on its profile geometry, including the rectangular model, the trapezoidal model, and the '1-consine' model.

- Propeller vortex is related to large manned aircraft. There are wake vortex caused by aircraft wings that significantly affect other aircraft. As for UAVs, due to their slower flight speed, the propeller vortex will no longer play an important role, while here, another vortex caused by the propellers could be considered, just as the name suggests, but it can be kept in consideration in the modeling phase of the system to be controlled.

The Principle of Wind Affects the UAV

All kind of wind modify the speed of the UAV, as a result altering the angle of attack and the angle of drift, which influences the values of the aerodynamic derivatives and eventually causes disturbances to the outcome force and torque of the UAV. In general, the most common approach to relate the consequences of winds on UAVs is from the point of view of speed. The wind models cited above explain only the distribution of velocity in space and time. Unfortunately, under most conditions, the induced speed of a certain wind field is not uniform, such as the wind gust field and the vortex field of the propeller, which denotes that the speed is not a constant value and changes together with the different parts of the UAV. To figure out this problem, in [21], we proposed an averaging way to calculate the actual wind speed and the wind gradient in order to derive the relationships between the variation of the air speed and the actual wind as well as between the variation of the body angular velocities and wind gradient. On the other hand, from the point of view of force, the effects of the non-uniform part of the flow field are considered an external force on the UAV. It is a well-known fact that interaction forces are the essential reason for the change in the kinetic states of objects, and the force caused by air is called resistance. In the simulation results presented, the Dryden wind turbulence model is used. The Dryden model considers the linear and angular velocity components of continuous gusts as spatially varying stochastic processes and specifies each component's power spectral density. The Dryden wind turbulence model is characterized by rational power spectral densities, so exact filters can be designed that take white noise inputs and output stochastic processes with the Dryden gusts' power spectral densities. The trends of the wind disturbances considered in the subsequent simulation are shown in Figure 4, dividing the contributions with respect to the three reference axes by linear and angular speed. For more mathematical details on the considered model, it is possible to refer to [22]. Table 2 shows the main parameters and the values used for the generation of disturbances with reference to the Simulink block called "Dryden Wind Turbulence Model (Continuous)". The gains value of the HOSM controller and of the disturbances estimators used in the simulations shown are reported in the Table 3. We want to emphasize that controller parameters give a formal solution of the stated control problem. Nevertheless, in practice, one often requires to calibrate the convergence rate, either to slow it down (relaxing the burden on the actuators) or to accelerate it in order to meet some system requirements. Consider, in this context, redundantly augmenting the magnitude parameter does not accelerate the convergence but rather only increases the chattering, while its reduction may lead to a loss of convergence.

Table 2. Dryden Wind Turbulence Model Parameters.

Model Type	Continuos Dryden (+q −r): Use continuous representation of Dryden velocity spectra with positive vertical and negative lateral angular rates spectra.
Altitude	z
UAV Velocity	$\sqrt{v_x^2 + v_y^2 + v_z^2}$
Measured wind speed speed	15 m/s. Measured wind speed at a height of 6 m
Wind direction	0 Measured wind direction at a height of 6 m (degrees clockwise from north) to aid in transforming the low-altitude turbulence model into a body coordinates.
Turbulence intensity	10^{-2} -Light Probability of the turbulence intensity being exceeded
Wingspan	10 m required in the turbulence calculation on the angular rates.
Band limited noise sample time	0.01 s Noise sample time at which the unit variance white noise signal is generated
Noise seeds (ug vg wg pg)	(23,341 23,342 23,343 23,344) Random noise seeds, specified as a four-element vector one for each of the three velocity components and one for the roll rate.

Table 3. Parameters of HOSM Controller and Estimators.

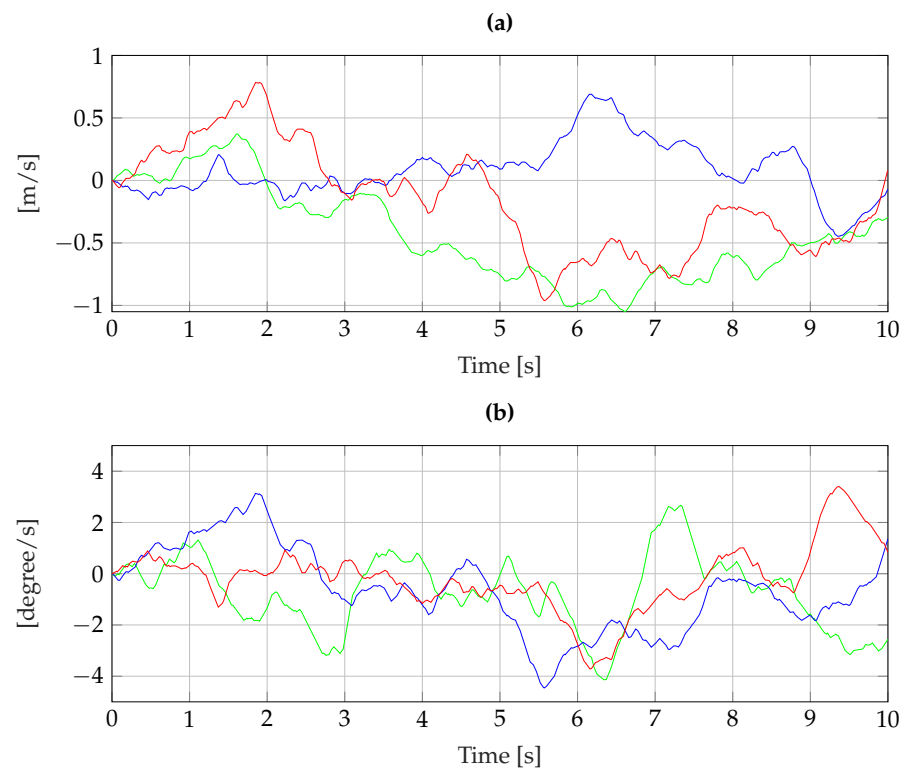|					
$k_{x,0} = 3$	$k_{x,1} = 10$	$k_{x,2} = 3$	$k_{y,0} = 3$	$k_{y,1} = 10$	$k_{y,2} = 2$
$k_{z,0} = 3$	$k_{z,1} = 10$	$k_{z,2} = 2$	$k_{\phi,0} = 1.5$	$k_{\phi,1} = 5$	$k_{\phi,2} = 1$
$k_{\theta,0} = 1.5$	$k_{\theta,1} = 5$	$k_{\theta,2} = 1$	$k_{\psi,0} = 2.7$	$k_{\psi,1} = 9$	$k_{\psi,2} = 1.8$
$\lambda_{x,1} = 10$	$\lambda_{x,2} = 3$	$\lambda_{y,1} = 10$	$\lambda_{y,2} = 3$	$\lambda_{z,1} = 10$	$\lambda_{z,2} = 3$
$\lambda_{\phi,1} = 2$	$\lambda_{\phi,2} = 1$	$\lambda_{\theta,1} = 2$	$\lambda_{\theta,2} = 1$	$\lambda_{\psi,1} = 2$	$\lambda_{\psi,2} = 1$

(a)

(b)

Figure 4. Disturbances for linear and angular velocity: (**a**) d_x (green), d_y (blue), d_z (red), (**b**) δ_1 (green), δ_2 (blue), δ_3 (red).

4.3. Simulation Results

In the test mission shown, we consider a rest-to-rest maneuver of the quadrotor with an initial state $[x\ y\ z\ \psi] = [0\ 0\ 0\ 0]$ and a final state $[x\ y\ z\ \psi] = [4\ 5\ 15\ \pi/4]$. In fact, we assumed zero linear and angular velocity of the UAV at the boundary states, with zero body orientation at time $t = 0$. The reference generator of the trajectories to be tracked is generated in such a way to have minimum accelerations and with a time suitable for carrying out the mission (see [23] for more details). Figures 5–12 are representative of the simulation considered in the cosimulation environment with FPGA implementation described in (Section 4.1): they compare and show the tracked characteristic state variables between the robust controller with disturbances compensation and the non-robust controller and the errors, the control inputs, the disturbances d_x, d_y, d_z, δ_3 and their estimation $\hat{d}_x$, $\hat{d}_y$, $\hat{d}_z$, $\hat{\delta}_3$; the effectiveness of the robust controller compared to the non-robust one and good estimation of the disturbances is evident. In fact, if the wind disturbance is not compensated, it takes the drone away from the trajectory to follow and leads it to dangerous states in terms of dynamics and greater energy consumption due to the removal from the desired trajectory. In particular, Figures 5–7 show the trends of the longitudinal (x), latitudinal (y), and altitudinal (z) positions, respectively, and their errors: in order to make evident the tracking difference between the robust and non-robust controller, their trends have been separated as well as their errors. The tracking errors of the non-robust controller are always large due to disturbances and parametric variations, but in the final part of the simulation, it cannot even keep the drone stable. With the same philosophy at the graphic level, Figure 8 shows the trend of yaw rate for the two controllers: the results are similar; therefore, with better performance than the robust controller, it should be emphasized that unlike the previous figures in this case, probably due to the wind and the unknown time-varying dynamics of some coefficients present in the simulator, the error reaches values of 12 degrees while keeping it always bounded.

Figure 9 shows the control inputs of robust control expressed as angular velocities of the four propellers. Modern-day ESC protocols can communicate at speeds of 37.5 kHz or

greater (in general, it is between 30 and 60 kHz because of the relatively low inductance of the high-speed motors and the potential interference to sensor boards), see [24]. The frequency of the actuation signals of the robust controller is 100 Hz and thus abundantly meets the requirements of modern commercial drones. The control inputs in the robust case have more rapid variations, while in the non-robust case, they assume a smooth trend: this is due to the terms of the estimators trying to compensate for what they see as possible parametric disturbance or uncertainty. Figure 10 shows the three-dimensional spatial trajectories of the two controlled drones in order to be able to visually quantify the error that occurs when the controller is not made robust with respect to disturbances. It is evident that the error that non-robust control makes at the 3D trajectory level is high, and therefore, there is an increase in consumption and the risk of drone instability. Finally, the disturbances d_x-d_y-d_z-δ_3 and their estimates used in robust control, together with their respective errors, are shown in Figures 11 and 12. The estimates are good for all disturbances: we note the presence of noise and a slightly larger error in δ_3, which is an event that causes errors of some degree in the tracking of the yaw rate, error due to the complex simulator considering that it has variable parameters over time, dynamics not considered in our model, and physical models based on experimental data; in fact, in the simulations not shown due to the lack of space made with the plant in Matlab, this does not happen.

To compare the performance of two proposed controllers in the considered mission, the Mean Absolute Error (MAE), Mean Square Error (MSE), Integral Squared Error (ISE), Integral Absolute Error (IAE), Integral Time-Weighted Absolute Error (ITAE) and Integral Time Square Error (ITSE) measures have been considered as comparative indices for the tracking along x, y, z and ψ. These performance indices are defined, for the variable x (the argument is analogous for the other variables) with respect to the reference x_{ref} to be tracked, during the N sampling instants, as follows

$$\text{MAE} = \frac{\sum_{i=1}^{N}[x_{ref,i} - x_i]}{N}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_{ref,i} - x_i)^2$$

$$\text{ISE} = \int (x_{ref} - x)^2 dt$$

$$\text{IAE} = \int |x_{ref} - x| \, dt$$

$$\text{ITAE} = \int (t \, |x_{ref} - x|) dt$$

where $x_{ref,i}$ and x_i are, respectively, the values of the variable x_{ref} and x at sampling time i, and ISE, IAE and ITAE integrate over time. The results are reported in Table 4, which shows the difference for each performance index of the tracked variables between the non-robust and robust controller.
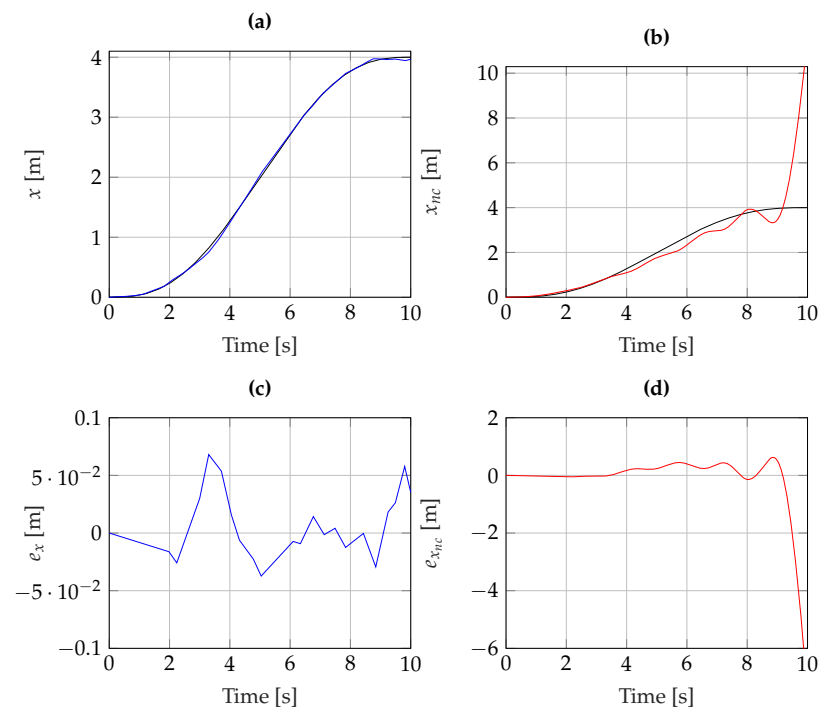
Figure 5. (**a**) Longitudinal motion of the UAV: the reference x_{ref} (black), x with robust control (blue); (**b**) Longitudinal motion of the UAV: the reference x_{ref} (black), x_{nc} with non-robust control (red); (**c**) Longitudinal error motion of the UAV with robust control e_x (blue); (**d**) Longitudinal error motion of the UAV with non-robust control $e_{x_{nc}}$ (red).
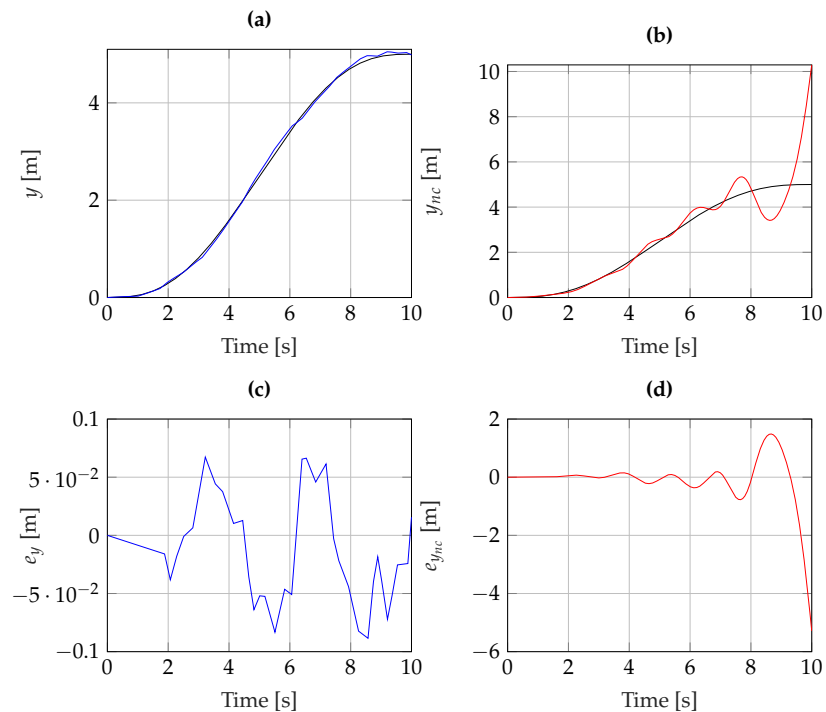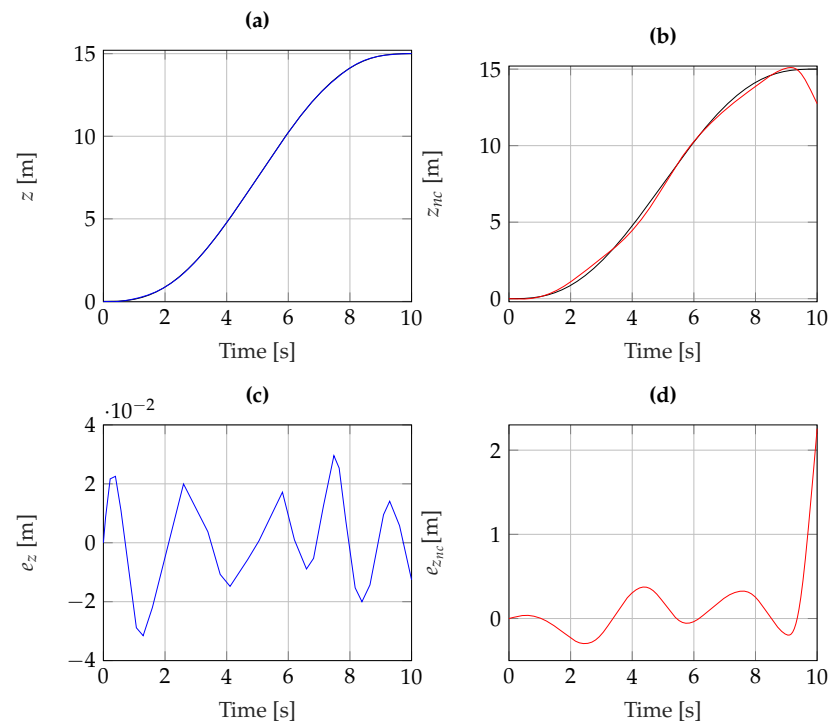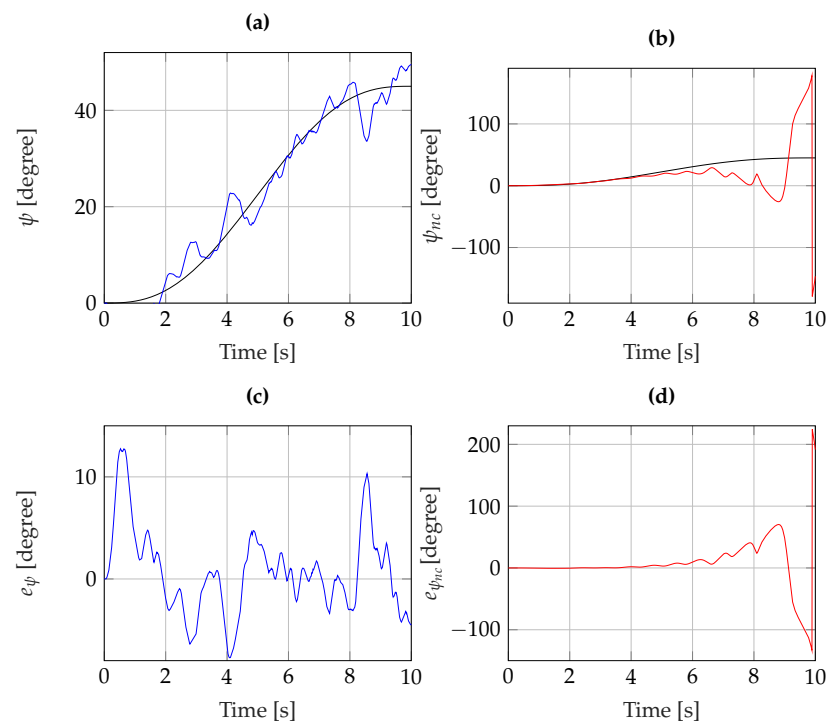

Figure 6. (**a**) Latitudinal motion of the UAV: the reference y_{ref} (black), y with robust control (blue); (**b**) Latitudinal motion of the UAV: the reference y_{ref} (black), y_{nc} with non-robust control (red); (**c**) Latitudinal error motion of the UAV with robust control e_y (blue); (**d**) Latitudinal error motion of the UAV with non-robust control $e_{y_{nc}}$ (red).

Figure 7. (**a**) Altitudinal motion of the UAV: the reference z_{ref} (black), z with robust control (blue); (**b**) Altitudinal motion of the UAV: the reference z_{ref} (black), z_{nc} with non-robust control (red); (**c**) Altitudinal error motion of the UAV with robust control e_z (blue); (**d**) Altitudinal error motion of the UAV with non-robust control $e_{z_{nc}}$ (red).

Figure 8. (**a**) UAV yaw angle: the reference ψ_{ref} (black), ψ with robust control (blue); (**b**) UAV yaw angle: the reference ψ_{ref} (black), ψ_{nc} with non-robust control (red); (**c**) yaw angle error of the UAV with robust control e_ψ (blue); (**d**) yaw angle error of the UAV with non-robust control $e_{\psi_{nc}}$ (red).

Figure 9. Propulsion angular velocities of robust (blue) and non-robust controller (red): $\omega_{p,1}$ (**a**), $\omega_{p,2}$ (**b**), $\omega_{p,3}$ (**c**), $\omega_{p,4}$ (**d**).
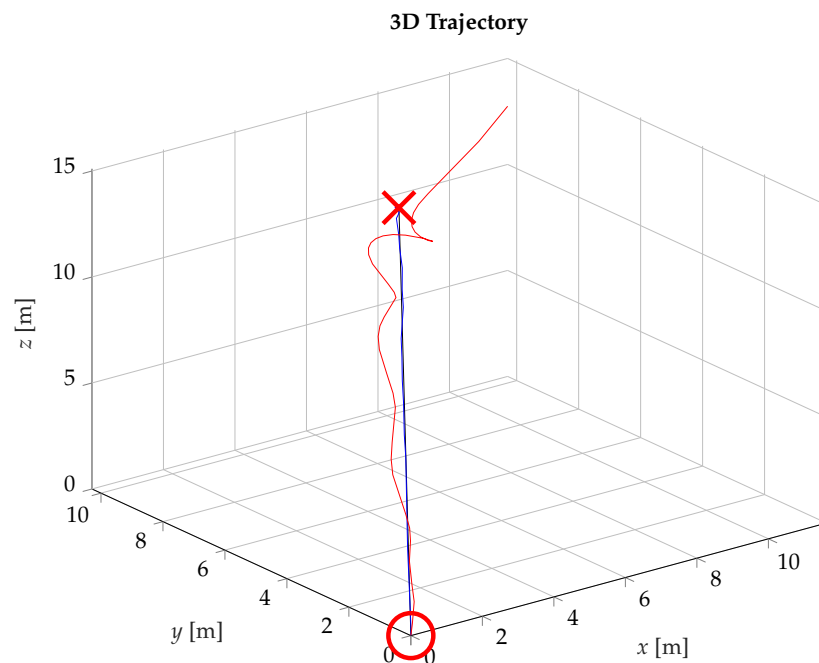

Figure 10. Three-dimensional (3D) drone trajectory: reference (black), robust control (blue), and non-robust control (red). The red circle and the red cross indicate, respectively, the starting and ending point of 3D reference trajectory.
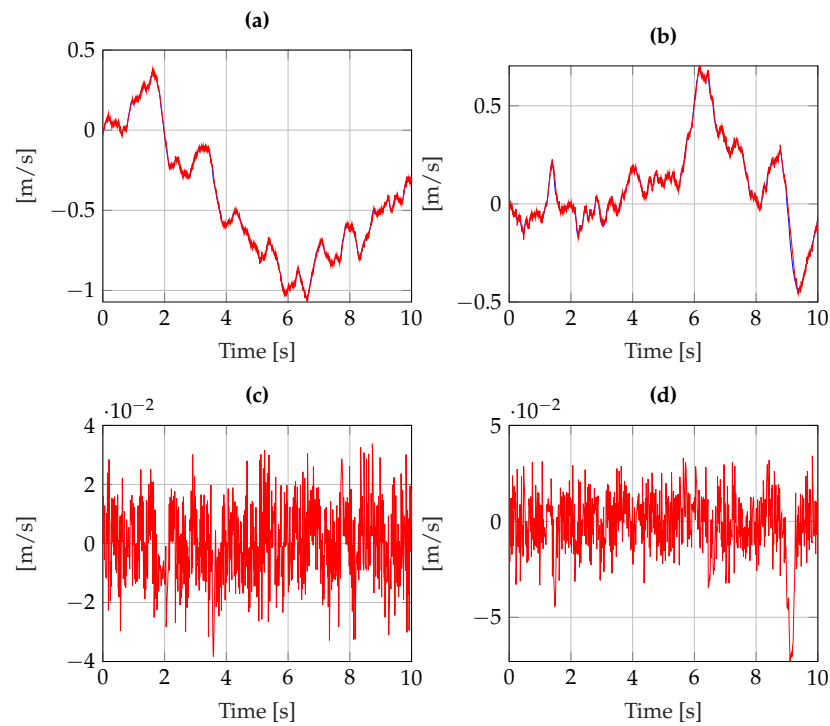
Figure 11. Disturbances estimations: (**a**) d_x (blue) and $\hat{d}_x$ (red), (**b**) d_y (blue) and $\hat{d}_y$ (red), (**c**) $d_x - \hat{d}_x$, (**d**) $d_y - \hat{d}_y$.
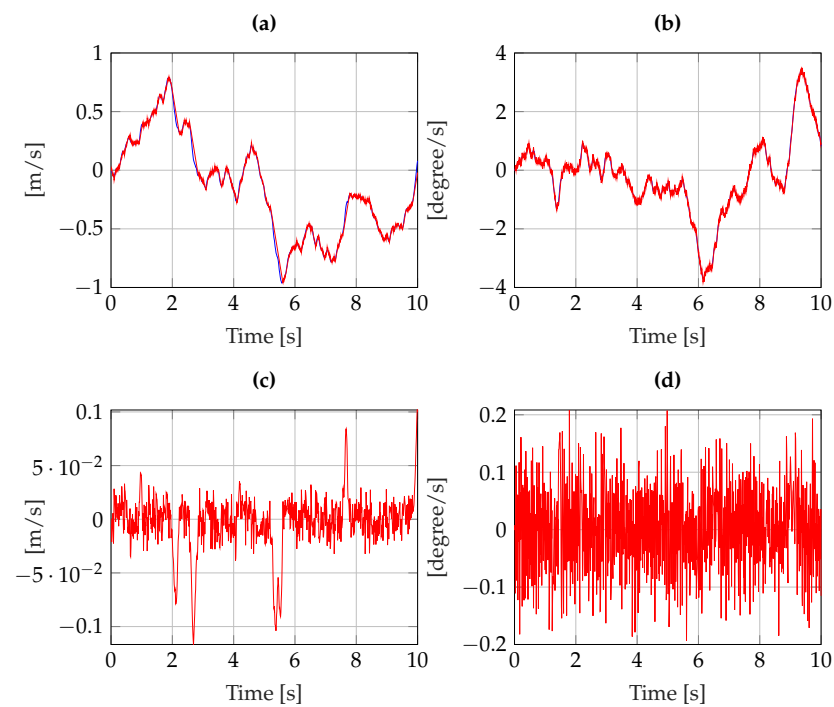

Figure 12. Disturbances estimations: (**a**) d_z (blue) and $\hat{d}_z$ (red), (**b**) δ_3 (blue), and $\hat{\delta}_3$ (red), (**c**) $d_z - \hat{d}_z$, (**d**) $\delta_3 - \hat{\delta}_3$.

Finally, the control effort for each of the two controllers is also reported in Table 5. It is computed as the time integral of the square of the angular velocity for each propeller. The effort of non-robust control is greater, but above all, it has not fulfilled the mission it was supposed to carry out, and therefore, additional time and energy will be required. In conclusion, Tables 4 and 5 point out, respectively, that the robust controller has better performance in terms of tracking error and lower total actuation effort.

Future work will deal with a comparison with another control strategy such as a neuroadaptive learning algorithm for constrained nonlinear systems with disturbance rejection [25] and asymptotic tracking with novel integral robust schemes for mismatched uncertain nonlinear systems [26].

Table 4. Performance indices for the x, y, z and ψ tracking errors.

Controller	Controlled Variable	MAE	MSE	ISE	IAE	ITAE	ITSE
Non-robust	x	0.4312	1.3198	12.92	4.2777	35.9148	125.34
Robust	x	0.018	0.000063	0.0063	0.1798	0.9743	0.0339
Non-robust	y	0.3867	1.3198	7.0007	3.8442	32.3692	66.1772
Robust	y	0.036	0.0019	0.0196	0.3604	2.2055	0.127
Non-robust	z	0.216	0.1321	1.2969	2.1505	13.9362	11.0271
Robust	z	0.0124	0.000021	0.0021	0.1238	0.5591	0.0089
Non-robust	ψ	19.3308	1492.2	1475.3	1.9254	1655.83	138,009
Robust	ψ	3.0647	17.5	175.7	30.6552	1.3336	629

Table 5. Actuation effort of robust and non-robust controllers as propeller angular velocity square integral along all the mission time.

Controller	Non-Robust	Robust
$\omega_{p,1}$	2.535937464×10^8	2.522437042×10^8
$\omega_{p,2}$	2.534601465×10^8	2.512100166×10^8
$\omega_{p,3}$	2.533127333×10^8	2.514066778×10^8
$\omega_{p,4}$	2.534819281×10^8	2.515695169×10^8
Total	1.013848554×10^9	1.006429915×10^9

5. Conclusions and Future Works

This paper presents a dynamic controller for a quadrotor where the user has partial knowledge of the model and in the presence of environmental disturbances such as wind along the linear and angular velocity. The proposed PI-like controller is able to compensate these issues by means of the use of HOSM estimators. The performance of such a controller has been tested in a co-simulation integrated environment between Matlab-Simulink and Simcenter Amesim with FPGA implementation. The results obtained by the robust controller are better than a basic version of the control both in terms of desired trajectories tracking and of actuation effort.

Future investigations will concern the development of theoretical results concerning the digital implementation of the proposed controller by exploiting the methodologies provided, for instance, in [27,28], experimental simulation and multi-agent (quadrotor) control systems in the presence of delay, quantization, and transmission errors (see [29]).

Author Contributions: Conceptualization, D.B. and S.D.G.; methodology, D.B. and S.D.G.; software, D.B. and M.D.F.; validation, D.B., M.D.F. and C.A.L.; formal analysis, D.B. and S.D.G.; investigation, D.B.; resources, D.B.; data curation, D.B.; writing—original draft preparation, Bianchi D.; writing—review and editing, Bianchi D.; visualization, D.B.; supervision, D.B. and S.D.G.; project

References

1. Lippitt, C.D.; Zhang, S. The impact of small unmanned airborne platforms on passive optical remote sensing: A conceptual perspective. *Int. J. Remote Sens.* **2018**, *39*, 4852–4868. [CrossRef]
2. Stolaroff, J.K.; Samaras, C.; O'Neill, E.R.; Lubers, A.; Mitchell, A.S.; Ceperley, D. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nat. Commun.* **2018**, *9*, 409–422. [CrossRef] [PubMed]
3. ECSEL JU Project "COMP4DRONES", 2019–2022. Available online: https://www.comp4drones.eu/ (accessed on 3 January 2023).
4. Ding, L.;Wang, Z. A Robust Control for an Aerial Robot Quadrotor under Wind Gusts. *J. Robot. Hindawi* **2018**, *2018*, 5607362. [CrossRef]
5. Ha, L.N.N.T.; Hong, S.K. Robust Dynamic Sliding Mode Control-Based PID–Super Twisting Algorithm and Disturbance Observer for Second-Order Nonlinear Systems: Application to UAVs. *Electronics* **2019**, *8*, 760. [CrossRef]
6. Verberne, J.; Moncayo, H. Robust Control Architecture for Wind Rejection in Quadrotors. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 152–161.
7. Sankaranarayanan, V.N.; Satpute, S.; Nikolakopoulos, G. Adaptive Robust Control for Quadrotors with Unknown Time-Varying Delays and Uncertainties in Dynamics. *Drones* **2022**, *6*, 220. [CrossRef]
8. Wan, K.; Gao, X.; Hu, Z.; Wu, G. Robust Motion Control for UAV in Dynamic Uncertain Environments Using Deep Reinforcement Learning. *Remote Sens.* **2020**, *12*, 640. [CrossRef]
9. Lúa, C.A.; García, C.C.V.; Di Gennaro, S.; Castillo-Toledo, B.; Morales, M.E.S. Real-Time Hovering Control of Unmanned Aerial Vehicles. *J. Math. Probl. Eng. Hindawi* **2020**, *2020*, 2314356.
10. Guillén-Bonilla, J.T.; García, C.C.V.; Di Gennaro, S.; Morales, M.E.S.; Lúa, C.A. Vision-Based Nonlinear Control of Quadrotors Using the Photogrammetric Technique. *Math. Probl. Eng.* **2020**, *2020*, 5146291. [CrossRef]
11. Hughes, P.C. *Spacecraft Attitude Dynamics*; Dover Publications, Inc.: Mineola, NY, USA, 1986.
12. Nagaty, A.; Saeedi, S.; Thibault, C.; Seto, M.; Li, H. Control and Navigation Framework for Quadrotor Helicopters. *J. Intell. Robot. Syst.* **2013**, *70*, 1–12. [CrossRef]
13. Shtessel, Y.; Edwards, C.; Fridman, L.; Levant, A. *Sliding Mode Control and Observation*; Birkhauser: New York, NY, USA, 2014.
14. Fridman, L.; Shtessel, Y.; Edwards, C.; Yan, X.-G. Higher-order sliding-mode observer for state estimation and input reconstruction in nonlinear systems. *Int. J. Robust Nonlinear Control.* **2008**, *18*, 399–412. [CrossRef]
15. Cappuzzo, F.; Dezobry, V.; Bianchi, D.; Di Gennaro, S. A Novel Co-simulation framework for Verification and Validation of GNC Algorithms for Autonomous UAV. In Proceedings of the 78th Vertical Flight Society Annual Forum and Technology Display, FORUM 2022, Fort Worth, TX, USA, 10–12 May 2022.
16. Nguyen, D.H.; Liu, Y.; Mori, K. Experimental Study for Aerodynamic Performance of Quadrotor Helicopter. *Trans. Jpn. Soc. Aeronaut. Space Sci.* **2018**, *61*, 29–39. [CrossRef]
17. Six, D.; Briot, S.; Erskine, J.; Chriette, A. Identification of the Propeller Coefficients and Dynamic Parameters of a Hovering Quadrotor From Flight Data. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1063–1070. [CrossRef]
18. Wang, B.H.; Wang, D.B.; Ali, Z.A.; Ting, B.T.; Wang, H. An overview of various kinds of wind effects on unmanned aerial vehicle. *Meas. Control.* **2019**, *52*, 731–739. [CrossRef]
19. Xiao, Y.; Jin, C. *The Flight Principle in the Atmospheric Disturbance*; National Defense Industry Press: Beijing, China, 1993.
20. Beal, T.R. Digital simulation of atmospheric turbulence for Dryden and von Karman models. *J. Guid. Control. Dyn.* **1993**, *16*, 132–138. [CrossRef]
21. Venkataramanan, S.; Dogan, A.; Blake, W. Vortex effect modelling in aircraft formation flight. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Austin, TX, USA, 11–14 August 2003.
22. Hakim, T.M.I.; Arifianto, O. Implementation of Dryden Continuous Turbulence Model into Simulink for LSA-02 Flight Test Simulation. *Iop Conf. Ser. J. Phys. Conf. Ser.* **2018**, *1005*, 012017. [CrossRef]

23. Bianchi, D.; Borri, A.; Di Gennaro, S.; Preziuso, M. UAV trajectory control with rule-based minimum-energy reference generation. *Proc. Eur. Control. Conf. (ECC)* **2022**, 1497–1502. [CrossRef]

24. Mogensen, K.N. Motor-Control considerations for electronic speed control in drones. *Analog. Appl. J. Tex. Instrum.* **2016**. Available online: https://www.ti.com/lit/an/slyt692/slyt692.pdf (accessed on 2 January 2023).

25. Yang, G.; Yao, J.; Dong, Z.; Neuroadaptive learning algorithm for constrained nonlinear systems with disturbance rejection. *Int. J. Robust Nonlinear Control.* **2022**, *32*, 6127–6147. [CrossRef]

26. Yang, G. Asymptotic tracking with novel integral robust schemes for mismatched uncertain nonlinear systems. *Int. J. Robust Nonlinear Control.* **2022**, *33*, 1988–2002. [CrossRef]

27. Di Ferdinando, M.; Pepe, P.; Di Gennaro, S. A new approach to the design of sampled-data dynamic output feedback stabilizers. *IEEE Trans. Autom. Control.* **2022**, *67*, 1038–1045. [CrossRef]

28. Di Ferdinando, M.; Pepe, P.; Di Gennaro, S. On Semi–Global Exponential Stability Under Sampling for Locally Lipschitz Time–Delay Systems. *IEEE Trans. Autom. Control.* **2022**, *99*, 120–131. [CrossRef]

29. Di Ferdinando, M.; Bianchi, D.; Di Gennaro, S.; Pepe, P. On the Robust Quantized Sampled–Data Leaderless Consensus Tracking of Nonlinear Multi–Agent Systems. In Proceedings of the 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 13–15 December 2021; pp. 3263–3268.