Wind Gust Estimation for Precise Quasi-Hovering Control of Quadrotor Aircraft

Sheikh Izzal Azid[a], Krishneel Kumar[b], Maurizio Cirrincione[a,c] and Adriano Fagiolini[d]

[a] School of Engineering and Physics, The University of the South Pacific, Fiji
[b] Electrical Engineering and Complex Systems, University of Genova, Italy
[c] The University of Technology of Belfort-Montbéliard, France
[d] Mobile and Intelligent Robots @ Panormous Laboratory (MIRPALab), Department of Engineering, University of Palermo, Italy

ARTICLE INFO

Keywords:
Robotics
Disturbance Observers
Unknown Input-State Observers
Robust Control
Quadrotors
ROS/Gazebo

ABSTRACT

This paper focuses on the control of quadrotor vehicles without wind sensors that are required to accurately track low-speed trajectories in the presence of moderate yet unknown wind gusts. By modeling the wind disturbance as exogenous inputs, and assuming that compensation of its effects can be achieved through quasi-static vehicle motions, this paper proposes an innovative estimation and control scheme comprising a linear dynamic filter for the estimation of such unknown inputs and requiring only position and attitude information. The filter is built upon results from Unknown Input Observer theory and allows estimation of wind and vehicle state without measurement of the wind itself. A simple feedback control law can be used to compensate for the offset position error induced by the disturbance. The proposed filter is independent of the recovery control scheme used to nullify the tracking error, as long as the corresponding applied rotor speeds are available. The solution is first checked in simulation environment by using the Robot Operating System middleware and the Gazebo simulator and then experimentally validated with a quadcopter system flying with real wind sources.

1. Introduction

Unmanned Aerial Vehicles (UAV) have been drawing increasing attention for more than two decades in various application fields, ranging from the industry to the military and from the service to the entertainment (see e.g. Bouabdallah et al. (2004); Mannucci et al. (2017); Seminara and Fontanelli (2017); Tomic et al. (2012); Michael et al. (2014); Mahony et al. (2012); Beard and McLain (2012)). Due to their ability to reach places, particularly those that are hardly accessible by land vehicles, UAVs are convenient tools for monitoring areas where natural disasters have just occurred. This is of high interest in the Pacific region, where remote neighborhoods need to be rapidly checked after cyclones or floods (Lee et al., 2016). As an example, the U.N. Food and Agriculture Organization (FAO) has launched in the Philippines a drone initiative to assess where agricultural land is at most risk of natural disasters and how to rapidly evaluate damages after they occur. It is strongly believed that the adoption of UAV platforms can significantly enhance risk and damage assessments, but also revolutionize the way to prepare for and respond to disasters. Indeed, beyond being viable for quick surveillance of the surrounding, it has already been shown that quadrotors can construct shelters (Lindsey et al., 2011), to rapidly provide protection before the occurrence of a natural disaster. Among such flying platforms, *quadrotors* (Fig. 1) have become the most popular for their flexibility, due to very low moments of inertia, greater stability, hovering capability, as well as fewer take-off requirements (Nonami et al., 2010). So far, they have already reached payload and flight endurance capacities for various applications, including environmental exploration and mapping (Michael et al., 2014; Saeedi et al., 2016), marine, riverine (Nuske et al., 2015), and agricultural monitoring (Ball et al., 2017; Zhang and Kovacs, 2012), object transportation and manipulation (Palunko et al., 2012; Tognon et al., 2018), to name a few. Reliable and fully autonomous navigation within indoor scenarios can also be obtained for certain tasks and environments, by extending techniques previously developed for ground vehicles (see e.g. Grzonka et al. (2012)).

Modeling and estimation of quadrotor systems have been well understood (Mahony et al., 2012), also including the effects of blade flapping. Solving the underlying control problems has however been a formidable challenge, even for basic operational scenarios, due to their severe underactuation, i.e. a quadrotor possesses six degrees of freedom (three translational and three rotational), but only four independent inputs (rotor speeds). To obtain accurate path tracking and autonomous flight capacity, combined control of rotational and translational motions is required, which results in highly nonlinear modeling (Sabatino, 2015). Nonetheless, nonlinear as the control problem may be, linear control strategies, particularly those with PIDs have proved quite effective with resulting good flying qualities (Bergamasco and Lovera, 2014). Online self-tuning PIDs also have shown good performance (Yang et al., 2013). However, PID control strategies rely on the linearized model of the quadrotor and are therefore highly dependent on the operational conditions.

*Corresponding author

sheikh.azid@usp.ac.fj (S.I. Azid); krishneel.kumar@edu.unige.it (K. Kumar); maurizio.cirrincione@usp.ac.fj (M. Cirrincione); fagiolini@unipa.it (A. Fagiolini)

ORCID(s): 0000-0003-1151-0973 (S.I. Azid); 0000-0002-9449-0244 (K. Kumar); 0000-0003-1886-1514 (M. Cirrincione); 0000-0001-9943-1975 (A. Fagiolini)

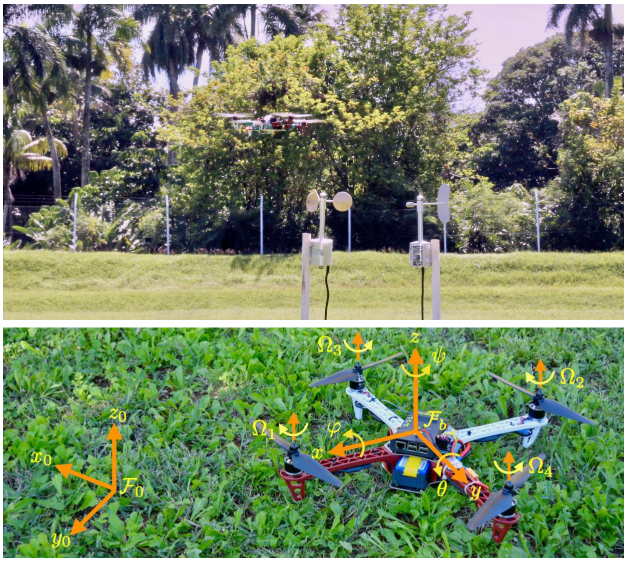

Figure 1: An Erlecopter aircraft and the anemometer, used to collect ground truth information about the wind speed (then converted to wind force), during testing at the University of South Pacific (Fiji) campus (above). The aircraft with associated coordinate frame (below).

To meet the progressively stringent specifications imposed by newly considered tasks of increasing complexity, techniques for high precision identification of linear models (Bergamasco and Lovera, 2014) and nonlinear control strategies have been developed. These are based on sliding mode control (Rios et al., 2018), Model Predictive Control (MPC) (Kamel et al., 2017), Lyapunov techniques (Yesildirek and Imran, 2014), neural networks (Pi et al., 2020), fractional attitude (Izaguirre-Espinosa et al., 2016), dynamic inversion with zero-dynamics stabilization (Das et al., 2009b), backstepping (Das et al., 2009a), robust and adaptive control (L'Afflitto et al., 2018), robust and optimal control (Satici et al., 2013), and combination of various control design strategies with state estimators (Nekoukar and Mahdian Dehkordi, 2021). Properties such as the differential flatness of the quadrotor model have also been exploited to obtain the generation of real-time trajectory (Van Nieuwstadt and Murray, 1998).

Apart from the above issues due to the nonlinearity of the system model, another difficulty lies in the presence of wind gusts, which can adversely affect outdoor mission performance as for precise attitude and position. This is a particularly challenging problem for lightweight quadrotors in harsh windy environments. Few, although important, works have addressed the trajectory recovery problem in this context (Guo et al., 2018; Jackson et al., 2008; Cabecinhas et al., 2015; Liu et al., 2016; Wang et al., 2016). Most of them considered only the quadrotor's attitude while taking into account possible actuator faults (Guo et al., 2018; Lyu et al., 2019; Sun et al., 2020) and input delays (Liu et al., 2016). Very recently disturbance observer-based sliding model controllers have been proposed to mitigate the effect of external inputs such as wind gusts (Taha et al., 2018; Fethalla et al., 2018; Ahmed and Chen, 2018). To the best of the authors' knowledge, only the work in (Cabecinhas et al., 2015) derives a Lyapunov-based global state stabilizer, thus including position, to steer a quadrotor along a predefined path, in presence of constant and known wind disturbance. Optimal trajectory generation has been addressed in (Guerrero et al., 2013) under the same assumption. However, since such hypotheses are approximate, the resulting accuracy is substantially low, while real-time wind reconstruction would allow better performance via e.g. wind-aware receding horizon approaches.

In this respect, a low-cost and lightweight alternative to using onboard anemometer sensors (Bruschi et al., 2016) is that of exploiting state observers, which would provide information about the current position and orientation of the aircraft, based on the input and output histories. The original ideas about state observers for linear systems date back to the sixties and were developed by Luenberger (Luenberger, 1966) for deterministic settings, while the most famous state observer for stochastic scenarios is the well-known Kalman filter. In the context of quadrotor aircraft, a very interesting solution has been studied in Burri et al. (2015) that uses only IMU and barometric pressure data but provides no information about the wind components. Another very useful approach has been proposed in Abeywardena et al. (2014), which combines accelerometer data with vision-based pose estimates, also to compute two components of the wind velocity vector. A more recent approach relies on the data-driven wind estimation to distinguish aerodynamic and contact forces acting on the quadrotor (Tomić et al., 2020). Other types of observers have been developed including Extended Kalman Filters (EKF), Extended State Observer (ESO), and Unknown Input Observers (UIO). Performance comparisons between these observers have been done in other domains and date back over some decades (Hostetter and Meditch, 1973; Wang et al., 1975; Kudva et al., 1980; Bhattacharyya, 1978; Miller and Mukundan, 1982; Kobayashi and Nakamizo, 1982; Yang et al., 2011; Kiyak et al., 2008; Al-Bayati and Skaf, 2010; Yang and Huang, 2009). Based on these papers, Kalman filter design still has issues in reconstructing unknown inputs and often requires assumptions on the disturbance signals or all system inputs and outputs to be measurable. In general, a UIO performs at least as much as the corresponding EKF and ESO, but in many other situations (cf. e.g. Al-Bayati and Skaf (2010)), a UIO combines easy design with superior performance.

This paper presents a complementary yet innovative approach where wind disturbance is modeled as unknown exogenous inputs. In particular, this work presents a new estimation and control scheme including a linear dynamic filter for the estimation of such unknown inputs and requiring only position and attitude information. The filter is built upon results from UIO theory (Sundaram and Hadjicostis, 2007, 2008; Sundaram, 2012) and allows estimation of the vehicle state and the actual wind signals without any direct measurement of the second ones. A UIO has been recently experimentally used for dynamic compensation of unknown load torques (Mudaliar et al., 2018). In the present case, by assuming that compensation of the wind effects can be achieved through quasi-static vehicle motions, a simple feedback control law is then described to compensate for the offset position error induced by the disturbance itself. One important feature of the proposed solution is that the UIO filter is independent of the recovery control scheme, used to nullify the tracking error, as long as the corresponding applied rotor speeds are available. The proposed solution is verified through Simulink/Gazebo simulations, through the exploitation of standard models for constant and large time-varying winds, and then experimentally validated within real wind gust scenarios by using an Erlecopter quadrotor system exploiting the Robot Operating System (ROS) middleware.

Contribution: The novelties and indeed the appealing features of the developed scheme are its simplicity, low computation cost, ability to obtain a fast response to wind gusts, and implementability on virtually all aircraft systems, as a stand-alone solution or an extension plugin for existing controllers. More in detail, the low computation demand is inherited by the simplicity of the linear UIO, that requires no additional sensors and that robustly estimates the overall effects of wind disturbance and other model uncertainty. The promptness and efficacy of the estimator, along with the position recovery scheme, are shown to outperform existing solutions based on EKF, model predictive control, and robust control. In this respect, it should also be noted that traditional robust control involves complex controllers and is unable to react fast enough in the presence of strong disturbances (Sariyildiz et al., 2019; Sariyildiz and Ohnishi, 2013), or may require the application of a signal that is unfeasible or too conservative; in contrast, the present approach compensates for the exact amount of disturbance which is estimated online. Finally, by being independent of the type of control law used to determine the rotor speeds of the aircraft, the developed scheme can provide existing controllers with the additional capacity to better deal with disturbances. This fact is shown in the paper, both in simulation and via experiments, on a platform using an open-source, the standard Ardupilot controller, which communicates with the estimator via a ROS middleware layer.

The paper is organized as follows. Section 2 introduces the nonlinear quadrotor model and its linearized dynamics around hovering configurations. Section 3 describes the theory of state and input observers for dynamical systems with partially unknown inputs. Section 4 illustrates the application of the above theory for deriving a wind and state estimator for quasi-hovering quadrotors and also presents a linear pose recovery scheme. The final Section 5 shows the simulation and experimental setups and discusses the results verifying the validity of the approach.

2. Quadrotor Mathematical Model

A quadrotor aircraft consists of a planar cross-shaped rigid chassis, actuated by four independent rotors that are mounted at the ends of the arms of the chassis itself. As indicated in Fig. 1, where Ω_i, for $i = 1, 2, 3, 4$, denotes the i-th rotor speed, the two opposite pairs of rotors are pitched in reverse directions, so that the drag forces and thrust forces, generated by all rotors spinning at the same speed (i.e. all $\Omega_i = \Omega_0$), produce null net torques on the aircraft. The overall lifting of the aircraft is obtained through suitably determining the value of Ω_0, while the stabilization of the center-of-mass position is achieved by varying the speeds of the independent rotors.

To characterize the aircraft dynamics, it is necessary to describe how its position and orientation change due to rotor actuation, gravity, and external disturbance (such as wind gusts). Since aircraft position is measured by GPS sensors, returning global positioning information, its center-of-mass coordinates, x, y, and z, are expressed in an inertial Earth frame $\mathcal{F}_0$. This choice also makes the specification of desired motion paths naturally expressed from the end-user viewpoint. On the other hand, since the orientation is obtained from onboard sensors, the attitude is conveniently described in the body frame $\mathcal{F}_b$ by a set of Euler angles (φ, θ, ψ). Assuming that $\mathcal{F}_b$ is aligned with the aircraft principal inertia axes and choosing the ZXY convention, the vehicle orientation is obtained by first rotating $\mathcal{F}_0$ of ψ (yaw) radians about the z-axis, then rotating of φ (roll) radians about the resulting x-axis, and finally of θ (pitch) radians about the resulting y-axis. The overall rotation matrix R expressing a vector v_b represented in the $\mathcal{F}_b$ in the $\mathcal{F}_0$ frame, via the relationship $v_0 = R\, v_b$, is given by the composition $R = (R_z(\psi) R_x(\varphi) R_y(\theta))^T$, where

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & s_\varphi \\ 0 & -s_\varphi & c_\varphi \end{pmatrix}, R_y(\theta) = \begin{pmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{pmatrix},$$

$$R_z(\psi) = \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where the common shortcuts $s_* = \sin(*)$ and $c_* = \cos(*)$ have been used. Direct computation yields

$$R = \begin{pmatrix} c_\theta c_\psi - s_\varphi s_\theta s_\psi & -c_\varphi s_\psi & s_\theta c_\psi + s_\varphi c_\theta s_\psi \\ c_\theta c_\psi + s_\varphi s_\theta s_\psi & c_\varphi s_\psi & s_\theta c_\psi - s_\varphi c_\theta s_\psi \\ -c_\varphi s_\theta & s_\theta & c_\varphi c_\theta \end{pmatrix}.$$

The forces acting on the quadrotor center of mass are the gravity, always oriented along the negative direction of the z-axis of $\mathcal{F}_0$, the total thrust F applied by the four rotors and aligned with the positive z-axis of $\mathcal{F}_b$, and the wind force $W = (W_x, W_y, W_z)^T$, whose components are expressed in $\mathcal{F}_0$ by convention. Each W_i, for $i \in \{x, y, z\}$ indicate forces rather than speeds. Having denoted with m the vehicle mass and with g the gravity acceleration, Newton's equations for the translational motion of the center of mass are:

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = -m \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + R \begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix} + W,$$

which can be expanded as

$$\begin{pmatrix} m\,\ddot{x} \\ m\,\ddot{y} \\ m\,\ddot{z} \end{pmatrix} = \begin{pmatrix} (s_\theta c_\psi + s_\varphi c_\theta s_\psi)F + W_x \\ (s_\theta s_\psi - s_\varphi c_\theta c_\psi)F + W_y \\ c_\varphi c_\theta F - m\,g + W_z \end{pmatrix}. \tag{1}$$

Moreover, the vehicle angular velocity vector $\eta = (p, q, r)^T$ can be referred to the Euler angles through the following dynamical relationship, for the ZXY convention, as

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_y(\theta)\begin{pmatrix} \dot{\varphi} \\ 0 \\ 0 \end{pmatrix} + R_y(\theta)R_x(\varphi)\begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}.$$

By inversion the previous relationship yields:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} c_\theta & 0 & s_\theta \\ t_\varphi s_\theta & 1 & -t_\varphi c_\theta \\ -s_\theta/c_\varphi & 0 & c_\theta/c_\varphi \end{pmatrix}\begin{pmatrix} p \\ q \\ r \end{pmatrix}, \tag{2}$$

where $t_\varphi = \tan(\varphi)$. Furthermore, due to the lean and thin structure of the vehicle, it is reasonable to assume that the wind torque is negligible, which implies that the only torque vector $T = (\tau_\varphi, \tau_\theta, \tau_\psi)^T$ acting on the vehicle itself is composed of the torques applied by the spinning rotors. The angular motion of the aircraft is described by the Euler's equations that reads $I\dot{\eta} + \eta \times I\,\eta = T$, where $\times$ represents the cross product between two vectors. Such equation can be written in terms of the elements of η as

$$I\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} + \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix} I \begin{pmatrix} p \\ q \\ r \end{pmatrix} = T\,.$$

Moreover, referring again to Fig. 1, the off-diagonal entries of the inertia matrix I are zero due to the symmetry of the quadrotor and the alignment of the axes of $\mathcal{F}_B$ with the vehicle principal inertia axes. Therefore, it holds $I = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$, where I_{xx}, I_{yy}, and I_{zz} indicate the inertia component around the axes of $\mathcal{F}_b$. Direct computation yields:

$$\begin{pmatrix} I_{xx}\,\dot{p} \\ I_{yy}\,\dot{q} \\ I_{zz}\,\dot{r} \end{pmatrix} = \begin{pmatrix} \tau_\varphi - (I_{zz} - I_{yy})\,q\,r \\ \tau_\theta - (I_{xx} - I_{zz})\,p\,r \\ \tau_\psi - (I_{yy} - I_{xx})\,p\,q \end{pmatrix}. \tag{3}$$

Eq. 1, 2, and 3 represent one possible nonlinear dynamic model of a quadrotor aircraft. Other nonlinear models, that are useful when, unlike here, the center of mass has to be expressed in the body frame $\mathcal{F}_b$, can be found e.g. in Beard and McLain (2012).

It is also known that the i-th rotor applies a force perpendicular to the plane of rotation of its blade (being thus aligned with the positive z-axis of the frame $\mathcal{F}_b$), and proportional to the square of its rotation speed, i.e. $k_F\,\Omega_i^2$, where k_F is the thrust constant. This force also generates a torque around the orthogonal axis that is represented by the opposite arm of the vehicle chassis, and its value is $l\,k_F\,\Omega_i^2$, where l the arm length. Finally, the i-th rotor produces a torque, due to the air drag force, that is opposite to its rotation axis and whose absolute value is again proportional to its rotation speed, i.e. $k_M\,\Omega_i^2$, where k_M is the drag constant. Moreover, there are two possible major configurations for the quadcopter: the so-called plus (+) and the cross (x) configurations. While the control of the yaw is identical for both configurations, roll and pitch control for the cross configuration uses all four rotors. Hence the cross configured quadcopter is considered to have more stability when compared to the plus configuration, which is more suitable for acrobatic maneuver (Norouzi Ghazbi et al., 2016). For the quadrotor in this article, the plus configuration model has been chosen to show the robustness of the control technique even in a lesser stable quadrotor configuration. Thus, the overall thrust F and the components of the torque vector T are related to the rotor speeds by the relationship:

$$\begin{pmatrix} F \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} k_F & k_F & k_F & k_F \\ 0 & 0 & -l\,k_F & l\,k_F \\ -l\,k_F & l\,k_F & 0 & 0 \\ -k_M & -k_M & k_M & k_M \end{pmatrix}\begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix}. \tag{4}$$

It is finally worth noticing that the adopted nonlinear model is derived by following a standard modeling approach (Mahony et al., 2012), whose adequacy to provide accurate quadrotor models with respect to reality is well known in the literature and thus is not studied here. Moreover, by including the unknown wind input vector $(W_x, W_y, W_z)^T$, the model is enabled to explicitly capture wind disturbance components, acting on the center of mass of the aircraft; possible terms affecting its orientation are assumed to be negligible and are then incorporated into the model uncertainty. This assumption derives from the fact that large wind gusts, such as those occurring during a storm, are characterized by wavelengths much longer than the quadrotor length. This implies that the wind force acting at each point of the quadrotor surface is practically uniform, and thus that the wind itself produces no meaningful net torque on it.

3. Observers for Dynamic Systems with Partially Unknown Inputs

The theory on delayed UIOs has been developed in Sundaram and Hadjicostis (2007) for a class of discrete-time linear systems affected by fully-unknown inputs. Under the same assumptions of the above reference, the result is here extended to the case where some inputs are in fact available. To this purpose, consider a discrete-time system

$$\begin{array}{rcl} X_{k+1} & = & A\,X_k + B_U\,U_k + B_N\,N_k\,, \\ Y_k & = & C\,X_k + D_U\,U_k + D_N\,N_k\,, \end{array} \tag{5}$$

where k is a time step, $X_k \in \mathbb{R}^n$ is a state vector, $U_k \in \mathbb{R}^{m_U}$ is a known input signal, $N_k \in \mathbb{R}^{m_N}$ is an unknown input disturbance, $Y_k \in \mathbb{R}^p$ is an output vector, and A, B_U, B_N, C, D_U and D_N are matrices of suitable dimensions. Given a positive time delay L, one can define, for $k \geq L$, the output history vector $\mathbb{Y}_k^L = (Y_{k-L}^T, Y_{k-L+1}^T, \cdots, Y_k^T)^T$ piling the latest $L+1$ outputs up to the current time. Then, a UIO is a discrete-time linear filter of the form

$$\begin{array}{rcl} \hat{X}_{k-L+1} &=& E\,\hat{X}_{k-L} + F\,\mathbb{Y}_k^L + B_U\,U_{k-L}\,, \\ \hat{N}_{k-L} &=& h(\hat{X}_{k-L}, U_{k-L}, Y_{k-L})\,, \end{array} \tag{6}$$

which can provide, for $k \geq L$, the *delayed estimates*, $\hat{X}_{k-L}$ and $\hat{N}_{k-L}$, of the system state and unknown input. That is, by the above iterative rule in Eq. 6, the UIO provides, at the time step $k = L$, the estimates $\hat{X}_0$ and $\hat{N}_0$, at the time step $k = L+1$, the estimates $\hat{X}_1$ and $\hat{N}_1$, and so on. Matrices E, F and the function $h(\cdot)$ are to be determined so that to guarantee the convergence of the estimates as it will be shown below.

The existence and the construction of such a UIO depend on conditions on the system matrices as it will be proven below. Recall first the definitions of *L-step observability matrix* $O^L = (C^T, (CA)^T, \cdots (CA^{L-1})^T)^T$ and L-step *invertibility matrix* $\mathbb{H}^L = \text{Invert}(A, B, C, D, L)$ of a linear system described by the matrices A, B, C, and D, that is

$$\mathbb{H}^L = \begin{pmatrix} D & 0 & 0 & 0 & \cdots & 0 \\ CB & D & 0 & 0 & \cdots & 0 \\ CAB & CB & D & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{L-1}B & CA^{L-2}B & \cdots & \cdots & CB & D \end{pmatrix}.$$

It is now possible to directly derive from Sundaram and Hadjicostis (2007) the following first main result:

Theorem 1 (UIO Existence for Partially Known Inputs). *Given a dynamic system as in Eq. 5 and having defined the matrix*

$$\mathbb{H}^L = \begin{pmatrix} \mathbb{H}_U^L & \mathbb{H}_N^L \end{pmatrix},$$

where

$$\begin{array}{l} \mathbb{H}_U^L = \mathit{Invert}(A, B_U, C, D_U, L)\,, \\ \mathbb{H}_N^L = \mathit{Invert}(A, B_N, C, D_N, L)\,, \end{array}$$

a UIO filter of the form of Eq. 6 exists, if F can be chosen so that the two following conditions hold:

$$\begin{array}{c} F\,\mathbb{H}^L = \left(0_{n\times m_U}, B_N, 0_{n\times L m_N}\right)\,, \\ E = A - F\,O^L \text{ is Schur.} \end{array} \tag{7}$$

Proof 1. *To begin with, consider the state estimation error e_k, resulting, at the present time step k, from the difference between the delayed state estimate $\hat{X}_{k-L}$, provided by the UIO at that instant, and the past real state value X_{k-L}, i.e. $e_k = \hat{X}_{k-L} - X_{k-L}$. The UIO design has to guarantee that such error be convergent with time. To this aim, it is instrumental to derive the real system's delayed dynamics by shifting backward Eq. 5 of L time steps. This yields:*

$$X_{k-L+1} \;=\; A\,X_{k-L} + B_U\,U_{k-L} + B_N\,N_{k-L}\,,$$

Then, for the dynamics of e_k the following can be straightforwardly obtained:

$$\begin{array}{rcl} e_{k+1} &=& \hat{X}_{k-L+1} - X_{k-L+1} = \\ &=& E\,\hat{X}_{k-L} - A\,X_{k-L} + F\,\mathbb{Y}_k^L - B_N\,N_{k-L} = \\ &=& E\,e_k + (E - A)\,X_{k-L} + F\,\mathbb{Y}_k^L - B_N\,N_{k-L}\,, \end{array}$$

where the term $E\,X_{k-L}$ has been added and subtracted in the last line of the above equation. To decouple the error dynamics from the system inputs, state, and output, and to force its behavior to be described by the dynamic equation $e_{k+1} = E\,e_k$, there must exist matrices E and F, so that E is Schur and the following condition is satisfied:

$$(E - A)\,X_{k-L} + F\,\mathbb{Y}_k^L - B_N\,N_{k-L} = 0\,. \tag{8}$$

To explore the possibility to satisfy Eq. 8, consider first how to factorize the expressions of all components of the output history vector $\mathbb{Y}_k^L$. More precisely, all such entries can be iteratively found and are given by

$$\begin{array}{rcl} Y_{k-L+h} &=& CA^h\,X_{k-L} + \sum_{i=0}^{h-1} CA^i B_U\,U_{k-L+i} + \\ &+& \sum_{i=0}^{h-1} CA^i B_N\,N_{k-L+i} + \\ &+& D_U\,U_{k-L+h} + D_N\,N_{k-L+h}\,. \end{array}$$

for $h = 0, 1, \cdots, L$. Rearranging the obtained expressions in matrix form allows finding the following relationship:

$$\mathbb{Y}_k^L = O^L X_{k-L} + \mathbb{H}_U^L\,\mathbb{U}_k^L + \mathbb{H}_N^L\,\mathbb{N}_k^L\,, \tag{9}$$

which connects the L-step known and unknown input history vectors $\mathbb{U}_k^L$ and $\mathbb{N}_k^L$, respectively, given by

$$\begin{array}{l} \mathbb{U}_k^L = (U_{k-L}^T, U_{k-L+1}^T, \cdots, U_k^T)^T\,, \\ \mathbb{N}_k^L = (N_{k-L}^T, N_{k-L+1}^T, \cdots, N_k^T)^T\,, \end{array}$$

with the output history vector $\mathbb{Y}_k^L$, through the invertibility matrices $\mathbb{H}_U^L$ and $\mathbb{H}_N^L$. Furthermore, by substituting Eq. 9 in Eq. 8, the following relationship is obtained

$$(E - A + F\,O^L)\,X_{k-L} + F\,\Gamma^L \begin{pmatrix} \mathbb{U}_k^L \\ \mathbb{N}_k^L \end{pmatrix} = 0\,, \tag{10}$$

where

$$\begin{array}{rcl} \Gamma^L &=& \begin{pmatrix} \mathbb{H}_U^L & \mathbb{H}_N^L - (B_N, 0, \cdots, 0) \end{pmatrix} = \\ &=& \mathbb{H}^L - (0, B_N, 0, \cdots, 0)\,. \end{array}$$

Now, by requiring that Eq. 10 holds for every X_{k-L}, $\mathbb{U}_k^L$, and $\mathbb{N}_k^L$, it directly follows that the matrices E and F must satisfy the conditions of the theorem.

Finally, the existence proof concludes by showing that a function $h(\cdot)$ can be found, which uses the information of the differences between the one-step forward projection of the delayed estimated state and of its past value, and between the delayed expected output and the real past one. More precisely, it results:

$$\begin{array}{rcl} X_{k-L+1} - \hat{X}_{k-L+1} &=& A\,X_{k-L} + B_U\,U_{k-L} + \\ &+& B_N\,N_{k-L} - \hat{X}_{k-L+1}\,, \\ Y_{k-L} - \hat{Y}_{k-L} &=& Y_{k-L} - C\hat{X}_{k-L} - D_U\,U_{k-L} + \\ &-& D_N\,N_{k-L}\,. \end{array}$$

As soon as the delayed system state estimate converges, i.e. $\hat{X}_{k-L} \rightarrow X_{k-L}$ (and hence also the delayed output estimate

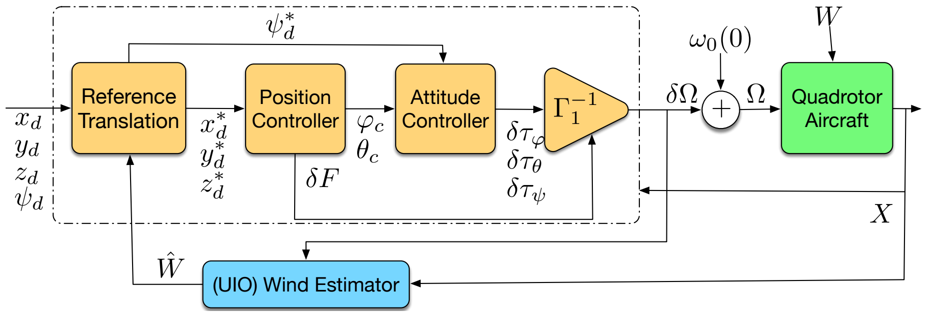

Figure 2: Architecture of the proposed controller with cascaded position and attitude regulation and wind estimation and compensation. The measured state X and rotor speed variations $\delta\Omega$ are used to obtain wind estimates $\hat{W}$, which are mapped to the proper translated references, x_d^*, y_d^*, z_d^*, and ψ_d^*, through Eq. 33 and 36; these position references are then used to derive commanded roll φ_c and pitch θ_c, which, along with the yaw reference are converted to the required thrust variation δF and torque variations, $\delta\tau_\varphi$, $\delta\tau_\theta$, $\delta\tau_\psi$; finally these quantities are converted to rotor speed variations and, by adding $\omega_0(0)$, to the required absolute rotor speeds Ω.

does it, i.e. $\hat{Y}_{k-L} \to Y_{k-L}$), the left-hand side of the above expressions vanishes and the unknown input N_{k-L} can be reconstructed via the equation:

$$N_{k-L} = G \begin{pmatrix} \hat{X}_{k-L+1} - A\,X_{k-L} - B_U\,U_{k-L} \\ Y_{k-L} - C\hat{X}_{k-L} - D_U\,U_{k-L} \end{pmatrix} \tag{11}$$

where $G = (B_N^T, D_N^T)^{T^\dagger}$ is the input decoupling matrix and † is the pseudo-inverse of a matrix. ■

Remark 1. *If $(B_N^T, D_N^T)^T$ is full-column rank, Eq. 7 in Th. 1 is solvable if, and only if, $rank(\mathbb{H}^L) = rank(\mathbb{H}_N^L) + m_N$. Moreover, E can be made Schur by the hypothesis of the pair (C, A) being observable.* □

4. Application to Wind Estimation and Compensation for Quasi-Hovering Quadrotors

A quadrotor aircraft is in *quasi-hovering* if it is floating over a spot or slowly moving with small roll and pitch angles. Around this configuration, its dynamics can be described with enough accuracy by a linear approximated model, derived from Eqs. 1, 2, and 3 (Burri et al., 2015; Michael et al., 2010). In the following, it is assumed that a misalignment due to a wind disturbance can be recovered while maintaining the quasi-hovering hypothesis. Under this hypothesis, this paper proposes the reference compensation and the wind estimation scheme depicted in Fig. 2.

4.1. Approximate Linear Models

Consider that the aircraft is required to reach a desired position (x_d, y_d, z_d) with a desired yaw angle ψ_d, while a constant wind vector $\bar{W} = (\bar{W}_x, \bar{W}_y, \bar{W}_z)^T$ is present. At the corresponding hovering equilibrium, by assuming that roll and pitch angles are small, i.e. $\varphi, \theta \approx 0$, the total thrust $\bar{F}$ and $\bar{W}$ are related by the following expression obtained from Eq. 1:

$$\bar{W}_x = 0\,, \quad \bar{W}_y = 0\,, \quad \bar{F} = m\,g - \bar{W}_z\,,$$

while the torques along all axes must be null, i.e.

$$\bar{\tau}_\varphi = \bar{\tau}_\theta = \bar{\tau}_\psi = 0\,.$$

as obtained from Eq. 2 and Eq. 3. Through Eq. 4 the following conditions are obtained:

$$\begin{array}{rcl} k_F\,\sum_i^4 \bar{\Omega}_i^2 & = & mg - \bar{W}_z\,, \\ -l\,k_F\,\bar{\Omega}_3^2 + l\,k_F\,\bar{\Omega}_4^2 & = & 0\,, \\ -l\,k_F\,\bar{\Omega}_1^2 + l\,k_F\,\bar{\Omega}_2^2 & = & 0\,, \\ -k_M\,\bar{\Omega}_1^2 - k_M\,\bar{\Omega}_2^2 + k_M\,\bar{\Omega}_3^2 + k_M\,\bar{\Omega}_4^2 & = & 0\,, \end{array}$$

which finally results in the fact that all required rotor speeds must be the same and given by

$$\bar{\Omega}_i = \omega_0(\bar{W}_z) = \frac{1}{2}\sqrt{\frac{mg - \bar{W}_z}{k_F}}\,, \quad \text{for all } i\,. \tag{12}$$

It is worth noticing that each $\bar{\Omega}_i$ is nonlinearly dependent on the unknown z-component of the wind, $\bar{W}_z$. However, when such a component is small compared to the gravity force, Eq. 12 can be approximated with a first-order Taylor expansion as follows:

$$\begin{aligned} \omega_0(\bar{W}_z) \quad &\approx \quad \omega_0(0) + \frac{\partial\omega_0(\bar{W}_z)}{\partial\bar{W}_z}(0)\,\bar{W}_z = \\ &= \quad \omega_0(0) - \frac{1}{4\sqrt{k_F mg}}\bar{W}_z\,, \end{aligned} \tag{13}$$

where the unknown input $\bar{W}_z$ has been conveniently separated from the nominal hovering speed value $\omega_0(0) = 1/2\sqrt{mg/k_F}$.

The linear approximated model of the aircraft center of mass motion can be obtained as a first-order truncation of the Taylor expansion of Eq. 1, being

$$\begin{pmatrix} m\,\ddot{x} \\ m\,\ddot{y} \\ m\,\ddot{z} \end{pmatrix} = \begin{pmatrix} s_{\psi_d}\,\bar{F}\,\varphi + c_{\psi_d}\,\bar{F}\,\theta + W_x \\ -c_{\psi_d}\,\bar{F}\,\varphi + s_{\psi_d}\,\bar{F}\,\theta + W_y \\ \delta F + W_z \end{pmatrix}, \tag{14}$$

where $\bar{F} = mg - \bar{W}_z$ is the force required at the equilibrium and $\delta F = F - \bar{F} = F - mg + \bar{W}_z$ is its variation. The first two equations of Eq. 14 linearly depend on φ, θ, which will be used as internal control variables, and the disturbance input W_z. Moreover, a linear approximation of δF can be obtained by expanding to the first order the first row of Eq. 4, thus leading to

$$\delta F = 2\, k_F\, \omega_0(\bar{W}_z) \left(\delta\Omega_1 + \delta\Omega_2 + \delta\Omega_3 + \delta\Omega_4\right)\,, \quad (15)$$

where the rotor speed variations are defined as

$$\delta\Omega_i = \Omega_i - \omega_0(0) \;\text{ for all } i\,.$$

Plugging then Eq. 13 into Eq. 15 gives

$$\delta F = \sqrt{k_F mg}\, \textstyle\sum_i \delta\Omega_i - \dfrac{\bar{W}_z}{2}\sqrt{\dfrac{k_F}{mg}}\, \textstyle\sum_i \delta\Omega_i\,,$$

which finally allows the last dynamic equation of Eq. 14 to be re-written as

$$m\, \ddot{z} = \sqrt{k_F mg}\left(\delta\Omega_1 + \delta\Omega_2 + \delta\Omega_3 + \delta\Omega_4\right) + W^*_{z,z}\,, \quad (16)$$

where

$$W^*_{z,z} = \bar{W}_z \left(1 - \alpha \left(\delta\Omega_1 + \delta\Omega_2 + \delta\Omega_3 + \delta\Omega_4\right)\right)\,, \quad (17)$$

with $\alpha = \frac{1}{2}\sqrt{\frac{k_F}{mg}}$. Furthermore, assuming again that roll and pitch remain small ($\varphi \approx 0$ and $\theta \approx 0$) during the entire operation of the aircraft, Eq. 2 is reduced to $\dot{\varphi} = p$, $\dot{\theta} = q$ and $\dot{\psi} = r$, which plugged into Eq. 3, along with the last three relationships of Eq. 4, yields

$$\begin{pmatrix} I_{xx}\, \ddot{\varphi} \\ I_{yy}\, \ddot{\theta} \\ I_{zz}\, \ddot{\psi} \end{pmatrix} = \begin{pmatrix} l\, k_F \left(\Omega_4^2 - \Omega_3^2\right) - a\, \dot{\theta}\, \dot{\psi} \\ l\, k_F \left(\Omega_2^2 - \Omega_1^2\right) - b\, \dot{\varphi}\, \dot{\psi} \\ k_M \left(\Omega_3^2 + \Omega_4^2 - \Omega_1^2 - \Omega_2^2\right) - c\, \dot{\varphi}\, \dot{\theta} \end{pmatrix},$$

with $a = I_{zz} - I_{yy}$, $b = I_{xx} - I_{zz}$ and $c = I_{yy} - I_{xx}$. The linearized model for the aircraft attitude motion is thus obtained, through a first-order Taylor expansion, where Eq. 13 is again used, giving the following expression:

$$\begin{pmatrix} I_{xx}\, \ddot{\varphi} \\ I_{yy}\, \ddot{\theta} \\ I_{zz}\, \ddot{\psi} \end{pmatrix} = \begin{pmatrix} l\sqrt{k_F mg}\left(\delta\Omega_4 - \delta\Omega_3\right) + W^*_{z,\varphi} \\ l\sqrt{k_F mg}\left(\delta\Omega_2 - \delta\Omega_1\right) + W^*_{z,\theta} \\ \beta\left(\delta\Omega_4 + \delta\Omega_3 - \delta\Omega_1 - \delta\Omega_2\right) + W^*_{z,\psi} \end{pmatrix}, \quad (18)$$

where $\beta = k_M\sqrt{mg/k_F}$ and

$$W^*_{z,\varphi} = -\frac{l\,\bar{W}_z}{2}\sqrt{\frac{k_F}{mg}}\left(\delta\Omega_4 - \delta\Omega_3\right)\,, \quad (19)$$

$$W^*_{z,\theta} = -\frac{l\,\bar{W}_z}{2}\sqrt{\frac{k_F}{mg}}\left(\delta\Omega_2 - \delta\Omega_1\right)\,, \quad (20)$$

$$W^*_{z,\psi} = -\frac{k_M \bar{W}_z}{2\sqrt{k_F mg}}\left(\delta\Omega_4 + \delta\Omega_3 - \delta\Omega_1 - \delta\Omega_2\right)\,, \quad (21)$$

are the *cross-coupled wind disturbance* components.

Remark 2. *The linear approximation of the aircraft model allows all unknown wind force components to be gathered into suitable disturbance terms.* □

Remark 3. *The x and y wind force components affect only the aircraft linear motion under the assumption of small angles, while component along z also has an impact on the angular degrees of freedom.* □

The observation of Remark 3 justifies the discussion of the following two cases of increasing complexity: a horizontal wind blowing along the xy-plane (leading to only two unknown signals) and general wind involving six unknown disturbances. For both cases, the control vector is defined as

$$U = \left(\Omega_1 - \omega_0(0), \Omega_2 - \omega_0(0), \Omega_3 - \omega_0(0), \Omega_4 - \omega_0(0)\right)^T\,,$$

while the derivation of the state forms proceeds separately as follows.

Finally, the above-obtained linearized model is based on the so-called small-angle assumption, which holds when the aircraft is close to the hovering configuration. Its adoption introduces approximations whose accuracy diminishes with the increase of the attitude angles and hence, contributes to the overall model uncertainty generated also by all neglected nonlinear terms. It should be noticed here that, in the presence of a strong wind, it may be necessary for any control algorithm to violate the small-angle assumption. Indeed, to accurately hover on a spot or track a trajectory, any adopted controller needs to tilt the aircraft so that the projections of the total thrust vector, along the x, y, and z axes, balance the wind force components, which eventually may lead the aircraft to attain roll and pitch angles that are far from zero. The choice of adopting the linearized model is hence not connected to seeking an extremely accurate model, yet a simplified one allowing, as it will be seen in the following sections, an input-state estimator and a controller that are simple and implementable even on low-cost setups. However, the robustness of the obtained solution with respect to model uncertainties has to be checked *a posteriori*, as it will be done afterward in the paper.

4.1.1. Horizontal Wind Scenario

When the wind gust is assumed to be horizontally blowing, i.e. when $\bar{W}_z = 0$ for all t, it also holds

$$W^*_{z,z} = W^*_{z,\varphi} = W^*_{z,\theta} = W^*_{z,\psi} = 0\,,$$

for all t, and the unknown input signal vector reduces to

$$N = (W_x, W_y)^T\,.$$

The corresponding equilibrium point is characterized by $\bar{F} = mg$, $x = x_d$, $y = y_d$, $z = z_d$, $\varphi = \theta = 0$, $\psi = \psi_d$, while the desired values for the speed variables u, v, w, p, q, and r are null. Having defined the aircraft state as

$$X = \left(x - x_d, y - y_d, z - z_d, v, u, w, \varphi, \theta, \psi - \psi_d, p, q, r\right)^T\,,$$

one obtains, from the first two equations of Eq. 14, from Eq. 16, and from Eq. 18, the following linear state form

$$\begin{aligned} \dot{X} &= A^* X + B_U^* U + B_N^{(1)} N , \\ Y &= C X + D_U U + D_N^{(1)} N , \end{aligned} \tag{22}$$

where

$$A^* = \begin{pmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \begin{pmatrix} g\, s_{\psi_d} & g\, c_{\psi_d} & 0 \\ -g\, c_{\psi_d} & g\, s_{\psi_d} & 0 \\ 0 & 0 & 0 \end{pmatrix} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{pmatrix} ,$$

$$B_U^* = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & b_z & 0 & 0 & 0 & 0 & b_\theta & -b_\psi \\ 0 & 0 & 0 & 0 & 0 & b_z & 0 & 0 & 0 & 0 & -b_\theta & -b_\psi \\ 0 & 0 & 0 & 0 & 0 & b_z & 0 & 0 & 0 & -b_\varphi & 0 & b_\psi \\ 0 & 0 & 0 & 0 & 0 & b_z & 0 & 0 & 0 & b_\varphi & 0 & b_\psi \end{pmatrix}^T ,$$

$$B_N^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T ,$$

with $b_z = \sqrt{k_F g/m}$, $b_\varphi = l\sqrt{k_F m g}/I_{xx}$, $b_\theta = l\sqrt{k_F m g}/I_{yy}$, $b_\psi = k_M \sqrt{mg/k_F}/I_{zz}$, and D_U and $D_N^{(1)}$ are null matrices of proper dimensions. Moreover, assuming that linear and angular positions can be measured through GPS and IMU sensors, respectively, the output matrix is

$$C = \begin{pmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \end{pmatrix} . \tag{23}$$

Finally, to apply the UIO theory described in Section 3, it is necessary to obtain a discrete-time model for the aircraft dynamics as in the form of Eq. 5. As usual, this result can be achieved by sampling all continuous-time variables, and thereby defining the discrete-time quantities

$$\begin{aligned} X_k = X(Tk) ,\ U_k = U(Tk) , \\ N_k = N(Tk) ,\ Y_k = Y(Tk) , \end{aligned} \tag{24}$$

where T is a suitable sample time and k is the discrete time step, and by using e.g. the following Euler discretization

$$A = I_{12\times12} + T A^* ,\quad B_U = T B_U^* ,\quad B_N = T B_N^{(1)} . \tag{25}$$

4.1.2. General Wind Scenario

In case of wind blowing from a generic direction, and thus also including a possibly non-null vertical component, $\bar{W}_z \neq 0$, the complete unknown input signal vector becomes

$$N = (W_x, W_y, W_{z,z}^*, W_{z,\varphi}^*, W_{z,\theta}^*, W_{z,\psi}^*)^T ,$$

and the aircraft dynamic model reads

$$\begin{aligned} \dot{X} &= A^* X + B_U^* U + B_N^{(2)} N , \\ Y &= C X + D_U U + D_N^{(2)} N , \end{aligned} \tag{26}$$

where

$$B_N^{(2)} = \begin{pmatrix} 0_{3\times3} & \frac{1}{m} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \begin{pmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{pmatrix} \end{pmatrix}^T ,$$

where $D_N^{(2)}$ is a null matrix of proper dimensions; in this way the discrete-time quantities defined as in Eq. 24 evolve according to the dynamic and input matrices

$$A = I_{12\times12} + T A^* ,\quad B_U = T B_U^* ,\quad B_N = T B_N^{(2)} , \tag{27}$$

the output matrix C defined as in Eq. 24, and the null matrices D_U and $D_N^{(2)}$, to which the UIO theory presented above can be applied.

4.2. Realization of the UIO filters

The design of the UIO filters for both wind scenarios can be obtained with the application of the theory presented in Section 3, while their construction can be carried out by following similar steps as in Sundaram and Hadjicostis (2007).

4.2.1. Horizontal Wind Scenario

For the dynamic model in Eq. 22, the feasibility rank condition (see Remark 1) is satisfied with delay $L = 2$, thereby ensuring the existence of a UIO filter using, at every instant, two consecutive output values. More precisely, the direct computation of the UIO matrices yields the dynamic matrix

$$E = 0_{12\times12} ,$$

the output injection matrix

$$F = \begin{pmatrix} \Delta & 0_{3\times3} & F_{1,3} & 0_{3\times3} & \Delta & 0_{3\times3} \\ \frac{3\Delta}{2T} & 0_{3\times3} & F_{2,3} & 0_{3\times3} & F_{2,5} & 0_{3\times3} \\ 0_{3\times3} & \frac{I_{3\times3}}{3} & 0_{3\times3} & \frac{I_{3\times3}}{3} & 0_{3\times3} & \frac{I_{3\times3}}{3} \\ 0_{3\times3} & -\frac{I_{3\times3}}{2T} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \frac{I_{3\times3}}{2T} \end{pmatrix} ,$$

where $\Delta \in \mathbb{R}^{3\times3}$ is a matrix with all null entries except for $\Delta(3,3) = 1/3$, $F_{1,3} = I_{3\times3} - 2\Delta$, $F_{2,3} = \frac{3\Delta - I_{3\times3}}{T}$, $F_{2,5} = \frac{2I_{3\times3} - 3\Delta}{2T}$, and finally the input decoupling matrix of Eq. 11 given by

$$G = \begin{pmatrix} 0_{2\times3} & \frac{m}{T} I_{2\times2} & 0_{2\times13} \end{pmatrix} .$$

4.2.2. General Wind Scenario

By repeating for the dynamic model in Eq. 26 the feasibility rank condition (see again Remark 1), it is apparent that the required delay is $L = 2$, also in this case. The obtained UIO filter dynamic matrix E is null, while F and G are reported in the Appendix.

Finally, by collecting Eq. 17, 19, 20, and 21, as in

$$\left(W_{z,z}^*, W_{z,\varphi}^*, W_{z,\theta}^*, W_{z,\psi}^* \right)^T = M \bar{W}_z ,$$

with

$$M = \begin{pmatrix} \left(1 - \frac{1}{2}\sqrt{\frac{k_F}{mg}}\left(\delta\Omega_1 + \delta\Omega_2 + \delta\Omega_3 + \delta\Omega_4\right)\right) \\ -\frac{l}{2}\sqrt{\frac{k_F}{mg}}\left(\delta\Omega_4 - \delta\Omega_3\right) \\ -\frac{l}{2}\sqrt{\frac{k_F}{mg}}\left(\delta\Omega_2 - \delta\Omega_1\right) \\ -\frac{k_M}{2\sqrt{k_F mg}}\left(\delta\Omega_4 + \delta\Omega_3 - \delta\Omega_1 - \delta\Omega_2\right) \end{pmatrix},$$

the third component of the wind, $\bar{W}_z$, can be reconstructed as

$$\hat{W}_z = \left(M^T M\right)^{-1} M^T \left(W^*_{z,z}, W^*_{z,\varphi}, W^*_{z,\theta}, W^*_{z,\psi}\right) . \quad (28)$$

Remark 4. *The approach proposed in Sundaram and Hadjicostis (2007) requires that all four rotor speeds and all three wind-force components be unavailable, leading to a total of seven unknown inputs. The corresponding rank test as in Remark 1, with $m_N = 7$, fails to find an integer delay $L \in [1, 12]$, thus indicating that a UIO filter where all inputs are considered unknown cannot be constructed. This depends on the fact that, at hovering, the throttle force and the z component of the wind are identically aligned (and thus indistinguishable) In this case, a UIO filter such as the one proposed, using the values of the rotor speeds, is instead required.*

Remark 5. *Since the general wind scenario includes the more specific one with only horizontal wind, the UIO filter derived for the 3D case can effectively produce a correct estimate even when $W_z = 0$ for all t. However, when it is known that the wind blows practically horizontally, it is convenient to use the more compact UIO filter obtained in 4.2.1, which avoids the calculation of Eq. 28 and involves smaller matrices F and G. Furthermore, obtaining the 2D-scenario filter from the 3D-case one is not immediate, since the dependency on the system parameters is lost during the derivation of the larger matrices F and G.*

4.3. Position and Attitude Control and Recovery

A common strategy to cope with the aircraft underactuation is by using the four-rotor inputs to directly control the z position of the center of mass and the vehicle attitude (φ, θ, ψ), and then to exploit roll and pitch as internal inputs, which drive the remaining longitudinal x and lateral y positions (see e.g. Mahony et al. (2012)). As it is known, this approach gives rise to a cascaded proportional-derivative controller, in which a first *position controller* generates the required thrust F and the commanded roll and pitch signals, φ_c and θ_c, while a second *attitude controller* determines the required torques τ_φ, τ_θ, and τ_ψ. Thrust and torques are then converted to rotor speed commands by inverting Eq. 4. This paper adopts and extends this approach here by assuming that the wind force is constant or slowly varying, i.e.

$$W_x(t) = \bar{W}_x\,,\ W_y(t) = \bar{W}_y\,,\ W_z(t) = \bar{W}_z\,.$$

A possible way to derive such a regulator is presented below.

4.3.1. *Horizontal Wind Scenario*

In the case of a horizontally blowing wind, the third and the fifth equations of the dynamic model in Eq. 22 read

$$\dot{z} = w\,,\quad \dot{w} = b_z(\delta\Omega_1 + \delta\Omega_2 + \delta\Omega_3 + \delta\Omega_4)\,, \quad (29)$$

and its last six equations read

$$\begin{gathered} \dot{\varphi} = p\,,\ \dot{\theta} = q\,,\ \dot{\psi} = r\,, \\ \dot{p} = b_\varphi(\delta\Omega_4 - \delta\Omega_3)\,,\ \dot{q} = b_\theta(\delta\Omega_1 - \delta\Omega_2)\,, \\ \dot{r} = b_\psi(\delta\Omega_3 + \delta\Omega_4 - \delta\Omega_1 - \delta\Omega_2)\,, \end{gathered} \quad (30)$$

which can be rewritten as

$$\begin{pmatrix} \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \Gamma_1 \begin{pmatrix} \delta\Omega_1 \\ \delta\Omega_2 \\ \delta\Omega_3 \\ \delta\Omega_4 \end{pmatrix},$$

where

$$\Gamma_1 = \begin{pmatrix} b_z & b_z & b_z & b_z \\ 0 & 0 & -b_\varphi & b_\varphi \\ b_\theta & -b_\theta & 0 & 0 \\ -b_\psi & -b_\psi & b_\psi & b_\psi \end{pmatrix}.$$

By forcing the above equation to have the following convergent dynamics

$$\begin{pmatrix} \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} -k_{z,2}\,\dot{z} - k_{z,1}(z - z_d) \\ -k_{\varphi,2}\,\dot{\varphi} - k_{\varphi,1}(\varphi - \varphi_c) \\ -k_{\theta,2}\,\dot{\theta} - k_{\theta,1}(\theta - \theta_c) \\ -k_{\psi,2}\,\dot{\psi} - k_{\psi,1}(\psi - \psi_d) \end{pmatrix} \overset{\text{def}}{=} \begin{pmatrix} \delta F \\ \delta\tau_\varphi \\ \delta\tau_\theta \\ \delta\tau_\psi \end{pmatrix},$$

with $k_{z,i}, k_{\varphi,i}, k_{\theta,i}, k_{\psi,i} > 0$, for $i = \{1, 2\}$, one obtains that the first part of the controller can be described by the formula:

$$\left(\delta\Omega_1, \delta\Omega_2, \delta\Omega_3, \delta\Omega_4\right)^T = \Gamma_1^{-1}\left(\delta F, \delta\tau_\varphi, \delta\tau_\theta, \delta\tau_\psi\right)^T\,, \quad (31)$$

where φ_c and θ_c are command signals that are assumed to be small, slowly changing, and that have still to be chosen. Moreover, the remaining equations of (22) are given by

$$\begin{gathered} \dot{x} = u\,,\ \dot{y} = v\,, \\ \dot{u} = g\, s_{\psi_d}\, \varphi + g\, c_{\psi_d}\, \theta + \bar{W}_x/m\,, \\ \dot{v} = -g\, c_{\psi_d}\, \varphi + g\, s_{\psi_d}\, \theta + \bar{W}_y/m\,, \end{gathered}$$

which can be rewritten as

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \Gamma_2 \begin{pmatrix} \varphi \\ \theta \end{pmatrix} + \frac{1}{m}\begin{pmatrix} \bar{W}_x \\ \bar{W}_y \end{pmatrix},$$

with

$$\Gamma_2 = g\begin{pmatrix} s_{\psi_d} & c_{\psi_d} \\ -c_{\psi_d} & s_{\psi_d} \end{pmatrix}.$$

Imposing for the above equation the convergence behavior described by

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} -k_{x,2}\,\dot{x} - k_{x,1}(x - x_d) \\ -k_{y,2}\,\dot{y} - k_{y,1}(y - y_d) \end{pmatrix},$$

with $k_{x,i}, k_{y,i} > 0$, for $i = \{1, 2\}$, one can obtain the *best-effort* roll and pitch command signals

$$\begin{pmatrix} \varphi_c \\ \theta_c \end{pmatrix} = \Gamma_2^{-1} \begin{pmatrix} -k_{x,2}\,\dot{x} - k_{x,1}(x - x_d) - \frac{1}{m}\hat{W}_x \\ -k_{y,2}\,\dot{y} - k_{y,1}(y - y_d) - \frac{1}{m}\hat{W}_y \end{pmatrix},$$

where $\hat{W}_x$ and $\hat{W}_y$ are the available estimates of the wind gust force, which completes the structure of the cascaded controller. Equivalently, one can combine the compensation term of the wind with the desired x and y positions, by rewriting the above expression as follows

$$\begin{pmatrix} \varphi_c \\ \theta_c \end{pmatrix} = \Gamma_2^{-1} \begin{pmatrix} -k_{x,2}\,\dot{x} - k_{x,1}(x - x_d^*) \\ -k_{y,2}\,\dot{y} - k_{y,1}(y - y_d^*) \end{pmatrix}, \quad (32)$$

with

$$x_d^* = x_d - \frac{1}{k_{x,1}m}\hat{W}_x\,, \quad y_d^* = y_d - \frac{1}{k_{y,1}m}\hat{W}_y\,. \quad (33)$$

Finally, since all involved dynamic equations are of second order, the simplest choice for the parameters of the controllers is

$$\begin{aligned} k_{\varphi,1} = k_{\theta,1} = k_{\psi,1} = \lambda_a^2\,, \quad k_{\varphi,2} = k_{\theta,2} = k_{\psi,2} = 2\lambda_a\,, \\ k_{x,1} = k_{y,1} = k_{z,1} = \lambda_p^2\,, \quad k_{x,2} = k_{y,2} = k_{z,2} = 2\lambda_p\,, \end{aligned} \quad (34)$$

where $-\lambda_a < 0$ and $-\lambda_p < 0$ are the desired eigenvalues characterizing the aircraft attitude and position dynamics, respectively. In order to enforce the underlying condition that φ_c and θ_c be slowly varying with respect to the attitude regulation dynamics, it must also be $\lambda_a \gg \lambda_p$. Fig. 2 shows the position and attitude controller in the general system architecture.

4.3.2. General Wind Scenario

In the presence of wind also with a non-null vertical component, it follows from the linearized model in Eq. 26, that the second equation of (29) and the fourth, fifth, and sixth equations of (30) become

$$\begin{aligned} \dot{w} &= b_z(\delta\Omega_1 + \delta\Omega_2 + \delta\Omega_3 + \delta\Omega_4) + \tfrac{1}{m}W_{z,z}^*\,, \\ \dot{p} &= b_\varphi(\delta\Omega_4 - \delta\Omega_3)\tfrac{1}{I_{xx}}W_{z,\varphi}^*\,, \\ \dot{q} &= b_\theta(\delta\Omega_1 - \delta\Omega_2) + \tfrac{1}{I_{yy}}W_{z,\theta}^*\,, \\ \dot{r} &= b_\psi(\delta\Omega_3 + \delta\Omega_4 - \delta\Omega_1 - \delta\Omega_2) + \tfrac{1}{I_{zz}}W_{z,\psi}^*\,. \end{aligned}$$

With straightforward calculations, like the horizontal wind case, it can be found that Eq. 31 generalizes to

$$\begin{pmatrix} \delta\Omega_1 \\ \delta\Omega_2 \\ \delta\Omega_3 \\ \delta\Omega_4 \end{pmatrix} = \Gamma_1^{-1} \begin{pmatrix} -k_{z,2}\,\dot{z} - k_{z,1}(z - z_d^*) \\ -k_{\varphi,2}\,\dot{\varphi} - k_{\varphi,1}(\varphi - \varphi_c) - \frac{1}{I_{xx}}\hat{W}_{z,\varphi}^* \\ -k_{\theta,2}\,\dot{\theta} - k_{\theta,1}(\theta - \theta_c) - \frac{1}{I_{yy}}\hat{W}_{z,\theta}^* \\ -k_{\psi,2}\,\dot{\psi} - k_{\psi,1}(\psi - \psi_d^*) \end{pmatrix}, \quad (35)$$

Table 1
Erlecopter's parameters.

Parameters	Value	Unit
Aircraft's Mass, m	1.12	[Kg]
Inertia along x, I_{xx}	$3.48 \cdot 10^{-2}$	[Kg·m^2]
Inertia along y, I_{yy}	$4.59 \cdot 10^{-2}$	[Kg·m^2]
Inertia along z, I_{zz}	$9.77 \cdot 10^{-2}$	[Kg·m^2]
Axle length, l	0.141	[m]
Thrust constant, k_F	$8.55 \cdot 10^{-4}$	[N·m/rad^2]
Drag constant, k_M	$1.60 \cdot 10^{-2}$	[N·m/rad^2]

where

$$z_d^* = z_d - \frac{1}{k_{z,1}m}\hat{W}_{z,z}^*\,, \quad \psi_d^* = \psi_d - \frac{1}{k_{\psi,1}I_{zz}}\hat{W}_{z,\psi}^*\,, \quad (36)$$

and $\hat{W}_{z,z}^*$, $\hat{W}_{z,\varphi}^*$, $\hat{W}_{z,\theta}^*$, $\hat{W}_{z,\psi}^*$ are the estimated cross-coupled wind components. The remainder of the controller regulating the x and y positions maintains the same structure as above.

Remark 6. *It should be noticed that a transversal wind, i.e. a wind blowing in any direction, mainly occurs only when the quadrotor hovers close to a hill or a very irregular ground. Horizontal components are typically much more intense and frequent than the vertical ones. Moreover, the assumption of a practically horizontal wind gust, in which the vertical component is negligible, allows obtaining a simpler and smaller state-input observer (UIO), which can be of benefit for low-cost aircraft systems with less computational power.*

5. Simulation and Experimental Validation

This section presents the validation of the proposed approach using an Erlecopter Aircraft platform. Tests and results are presented by increasing complexity, starting from Matlab/Simulink simulation, to ROS/Gazebo emulation, and finally to real experiments.

5.1. Simulink-based Validation

As a first validation step, the quadrotor linearized model derived in Section 4-A1), the UIO filter obtained in Section 4-B1), and the pose recovery scheme of Section 4-C) have been implemented in the Matlab/Simulink environment. The purpose of this first step is to show the "core" behavior of the proposed method in ideal conditions, without the presence of model uncertainties due to nonlinearities, delays due to the dynamics of the propellers, and noise measurement. The parameters of the Erlecopter which have been identified are available from data-sheets and are reported in Table 1. The aircraft dynamic model has been simulated in continuous time, while the UIO filter has been designed with a first-order Euler time-discretization of the model as shown in Eq. 24 and 25 and afterwards implemented at 100 Hz (thus with $T = 10^{-2}$ s). The controlled system has been tested for its ability to execute two typical tasks with horizontal wind

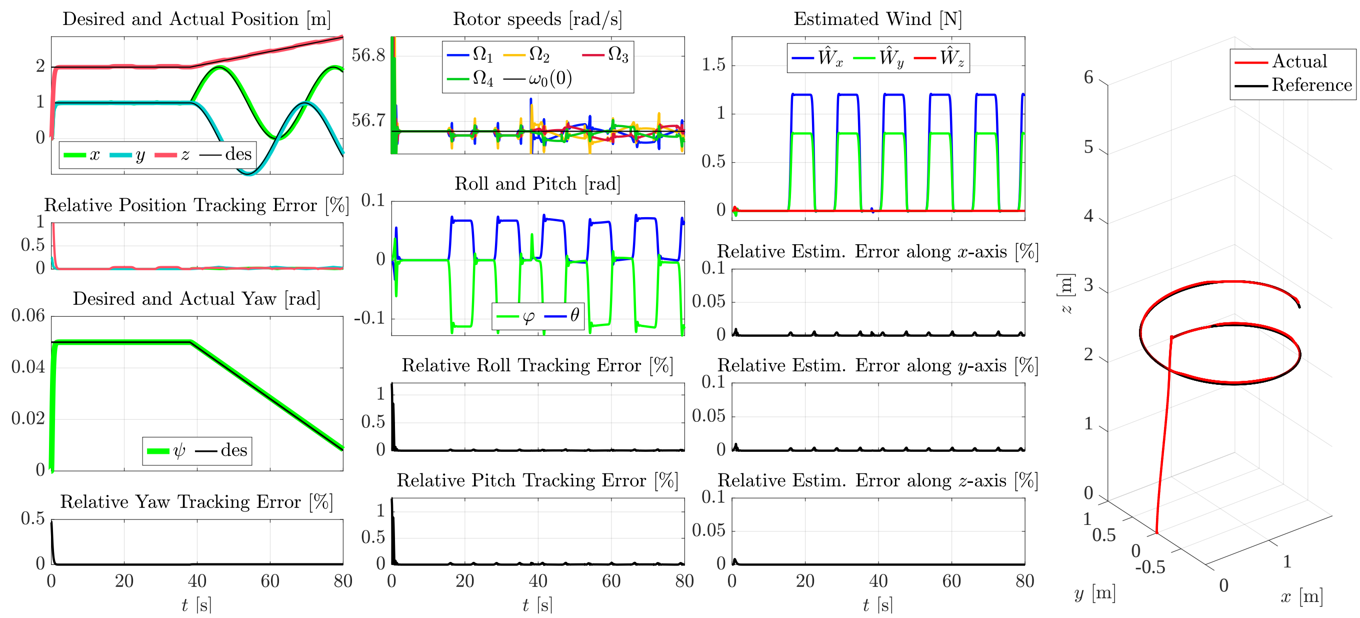

Figure 3: Results from a Matlab/Simulink simulation run with horizontal wind gusts. The controlled quadrotor quickly reaches a desired position of $x_d = 1$ m, $y_d = 1$ m, $z_d = 2$ m and $\psi_d = 5 \cdot 10^{-2}$ rad. As the wind-gust series starts at $t = 15$ s, the UIO rapidly reconstructs the wind-force component signals and provides their values to the pose recovery controller. The controller commands rotor speed signals that ensure a sufficient total thrust and lead to suitable roll and pitch angles. From $t = 38$ s, the quadrotor starts tracking a spiral trajectory, while the wind-gust series continues to be present. It can be observed that the small-value hypothesis for the roll and pitch angles is always satisfied. Gust amplitudes are $W_{x,m} = 1.2$ N and $W_{y,m} = 0.8$ N, gust lengths are $d_{x,m} = d_{y,m} = 0.5$ m, and gust frequencies are $f_x = f_y = 0.5$ rad/s. State and unknown wind estimates are very accurate.

gust bursts: 1) hovering maintenance and 2) spiral trajectory tracking. The first task aims to evaluate the controlled aircraft's resilience to largely changing winds when trying to keep a constant position, like during a photography operation or on-the-spot object manipulation. The second task intends to assess its capacity to follow curved paths while ascending or descending along the z-axis, which is useful when the quadrotor has to circle a target point while landing or taking off. The general case with a generic wind is omitted here for lack of space and will be directly shown in the next step.

Wind gusts are modeled according to the Military Specification MIL-F-8785C Military Specification (1980), which is commonly used to assess airplane responses to large wind disturbances. The mathematical representation of a wind gust force along each component W_i, with $i \in \{x, y\}$, is the following:

$$W_i = \begin{cases} 0 & \sigma_i < 0\,, \\ \frac{W_{i,m}}{2}\left(1-\cos\left(\frac{\pi\sigma_i}{d_{i,m}}\right)\right) & 0 \le \sigma_i \le d_{i,m}\,, \\ W_{i,m} & \sigma_i > d_{i,m}\,, \end{cases}$$

where $W_{i,m}$ is the gust amplitude, $d_{i,m}$ is the distance from the start of the gust to the point at which the gust velocity reaches a maximum, and σ_i is the distance the aircraft has penetrated the gust, which is dynamically updated in accordance with the motion of the aircraft itself. For a more in-depth description and motivation of the model, the reader can also see the reference Etele (2006). To obtain wind gust bursts, the amplitude $W_{i,m}$ is chosen as a time-parameterized sinusoid function with unitary amplitude and frequency f_i, i.e.

$$W_{i,m}(t) = \sin(2\pi f_i\, t)\,.$$

The same wind gust model is also used in the next step validation with ROS/Gazebo, which will be presented in the next subsection.

Fig. 3 reports the results obtained and shows the effectiveness of the proposed estimation and control methods. It can be seen that the controlled quadrotor quickly reaches a desired position in the first phase. Then, as the wind-gust series starts, the UIO rapidly reconstructs the wind-force component signals and provides their values to the pose recovery controller. The controller commands rotor speed signals that ensure a sufficient total thrust and lead to suitable roll and pitch angles. Indeed, the internal commands φ_c and θ_c reach steady-state values satisfying the relationship in Eq 33. The controller can instantaneously compensate for the effect of the wind, to maintain the quadrotor position constant. In the last part of the simulation, the quadrotor starts tracking a spiral trajectory, while the wind-gust series continues to be present. It can be observed that the small-value hypothesis for the roll and pitch angles is always satisfied during the simulation.

5.2. ROS/Gazebo-based Software-in-the-Loop (SITL) Validation

As a second validation step, the proposed solution has been tested within a ROS/Gazebo-based framework at least for the following three reasons: 1) to test and assess the algorithm's robustness to model and measurement uncertainties, 2) to show the independence of the proposed estimator with respect to the controller used, provided that the information about the rotor speeds is available, and 3) to obtain a SITL validation of the proposed software implementation, later enabling us to proceed with its direct upload on the real Erlecopter platform without the need to change or re-tune any control parameter.

To ensure complete software compatibility with the Erlecopter Brain 3 interface, ROS Indigo distribution and Gazebo 4 have been used. In particular, the architecture implemented is as follows: 1) ROS provides a middleware layer for the SITL emulation, 2) the Gazebo software provides a reliable simulation platform for the aircraft physics, including wind gust forces, and 3) a Matlab/Simulink scheme implements the proposed UIO filter. Gazebo receives the rotor speeds from subscribed ROS topics and publishes the aircraft pose (position and attitude) to ROS, while the Matlab/Simulink node subscribes and receives these pose topics.

The Erlecopter dynamics simulated in Gazebo is equivalent to the full nonlinear model described in Sun et al. (2020) (cf. Eq. 1-6 in the reference), where only the aerodynamic yaw damping coefficient γ is neglected, i.e. $\gamma = 0$. The model also includes the four propellers' dynamics and saturation and measurement noise.

The first simulation has been done in a *hybrid modality*, by using the proposed UIO filter, designed for a horizontal wind scenario, in combination with the Ardupilot controller. This aimed at showing one of the appealing features of the proposed method, which is the possibility to be easily plugged in within existing commercial solutions, thereby enabling a fast technological transfer. The simulation has required the Erlecopter system to execute a takeoff-and-hovering task with the presence of a constant horizontal wind force of amplitude $N = (\bar{W}_x, \bar{W}_y) = (1.2, 0.8)$ N. Fig. 4 shows the obtained results when, starting from $X(0) = 0_{12\times1}$, the drone is required to move to $x_d = 0$ m, $y_d = 0$ m, $z_d = 9$ m, and $\psi_d = -1.15$ rad. As it can be seen, the UIO filter can accurately reconstruct the two wind components. As the Ardupilot controller is used as a black box, the desired positions and yaw angle are corrected to the values, $(x_d^*, y_d^*, z_d^*)^T$ and yaw angle ψ_d^* from Eq. 33 and 36, in order to compensate for the effect of the estimated wind. The figure shows that the controller asymptotically tracks the original desired pose. A small difference in the rotor speeds at the hovering configuration can be observed, with $\Omega_1 \simeq \Omega_2 > \Omega_3 \simeq \Omega_4$, which causes a final very small yet non-zero yaw torque, and is due to the slow adjustment of the yaw angle commanded by the Ardupilot when close to the desired one ψ_d. This final slow variation of the yaw can be seen in the left-bottom graph of the picture. Notwithstanding that the controller is encapsulated within the Erlecopter Brain Interface, the UIO filter can effectively retrieve the wind components within few seconds and before the aircraft reaches the desired steady-state

The second simulation aims at showing the results obtained by using the proposed full solution (estimator + linear regulator) with the most general scenario, in which the aircraft is required to track a spiral trajectory while generic-direction wind gusts are present. Fig. 5 presents the outcomes of the SITL validation with a wind gust of amplitude $N = (\bar{W}_x, \bar{W}_y, \bar{W}_z) = (-0.5, 0.8, 1.2)$ N. The UIO filter is the one obtained as in Section 4-B2), with the linear relation of Eq. 28 used to compensate for $\bar{W}_z$. As can be seen from the plots, the wind force is estimated with a relative error of about 0.1%.

It is finally worth comparing the estimation and tracking performance of the proposed method with those obtained by the hybrid UIO+Ardupilot solution, and the recently proposed MPC-based solution described in Kamel et al. (2017). Fig. 6 compares typical behaviors of the controlled Erlecopter, obtained via ROS/Gazebo SITL simulations when the aircraft is required to hover and is subject to a horizontal yet constant wind. In the simulation, the Ardupilot is provided with desired positions, corrected according to the UIO information as in Eq. 33 and 36. Since the comparison is performed with the proposed linear controller, the linear MPC from Kamel et al. (2017) is selected, with default controller values. In the simulation with the proposed approach, the proportional and derivative gains have been chosen as $\lambda_p = 0.18$ and $\lambda_a = 1.8$, to be compliant with the ones of the MPC. Remarkably, all three solutions are capable of dealing with the presence of noise. However, in the two simulations with the implemented UIO, the wind estimation is faster and the achieved accuracy is much higher (the average relative estimation error is below 0.1% along every direction) than the ones obtained with the MPC approach (where the average relative estimation error is around 3.2% for the x and y directions and 41% for the z component). As for the tracking skills, the proposed solution performs better than the MPC, which is, in turn, more effective than the Ardupilot. However, it is also noteworthy that a larger steady-state error of the yaw angle occurs when the MPC is adopted.

Furthermore, as observed above, the robustness of the proposed solution, that is based on the linearized model, has to be check with respect to the possible violation of the small-angle assumption. To this purpose, as a final evaluation step within the ROS/Gazebo environment, it is worth assessing the applicability of our proposed estimation and control solution with sudden and large step-wise winds, which represents a condition even more challenging than that of a real smoothed wind gust. According to the Beaufort wind force scale, adopted in many countries, a storm is characterized by a wind-gust series with horizontal speed components $v_x, v_y \in [24.5, 28.4]$ m/s and wavelength $\sigma \in [9, 12.5]$ m. The corresponding wind forces W_x and W_y affecting the quadrotor depend on the air density ρ and the quadrotor sections S_x, S_y exposed to the wind. Using the formula reported below in Eq. 37, the worst-case speed of 28.4 m/s ap-

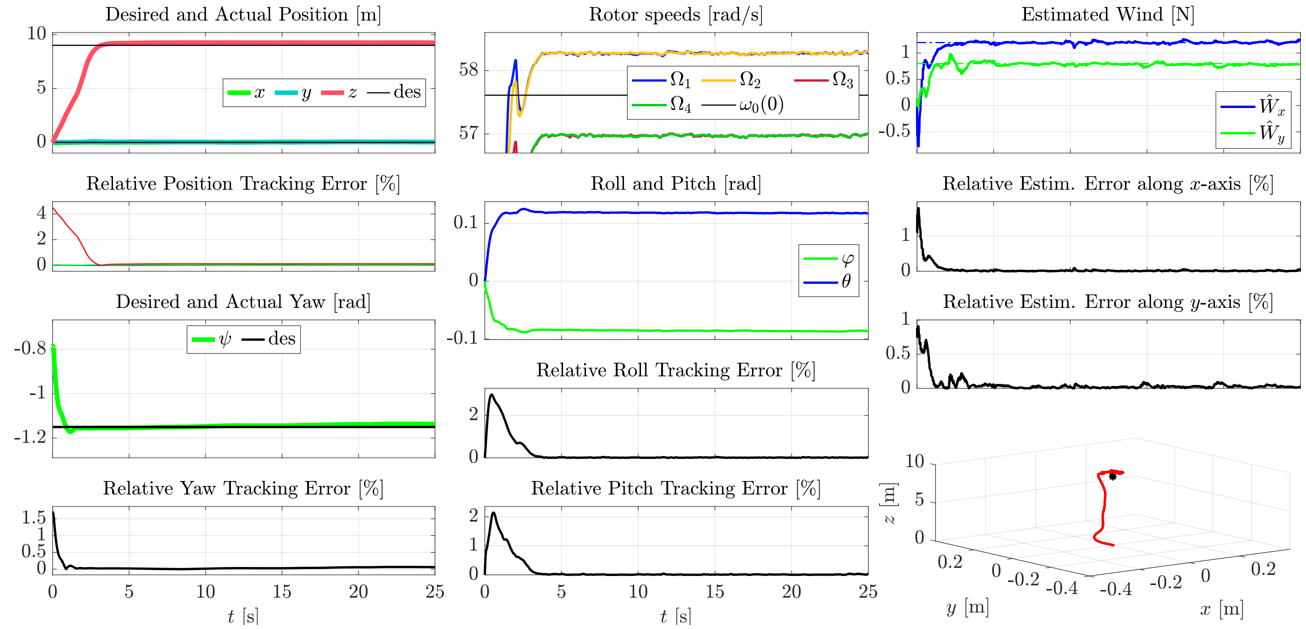

Figure 4: ROS/Gazebo-based validation of the hybrid solution with the proposed UIO filter, designed for a horizontal wind scenario, used in combination with the Ardupilot controller. The Erlecopter is required to takeoff up to $x_d = 0$ m, $y_d = 0$ m, $z_d = 9$ m, with required yaw of $\psi_d = -1.15$ rad, while the wind amplitude is $N = \left(\bar{W}_x, \bar{W}_y\right) = (1.2, 0.8)$ N.

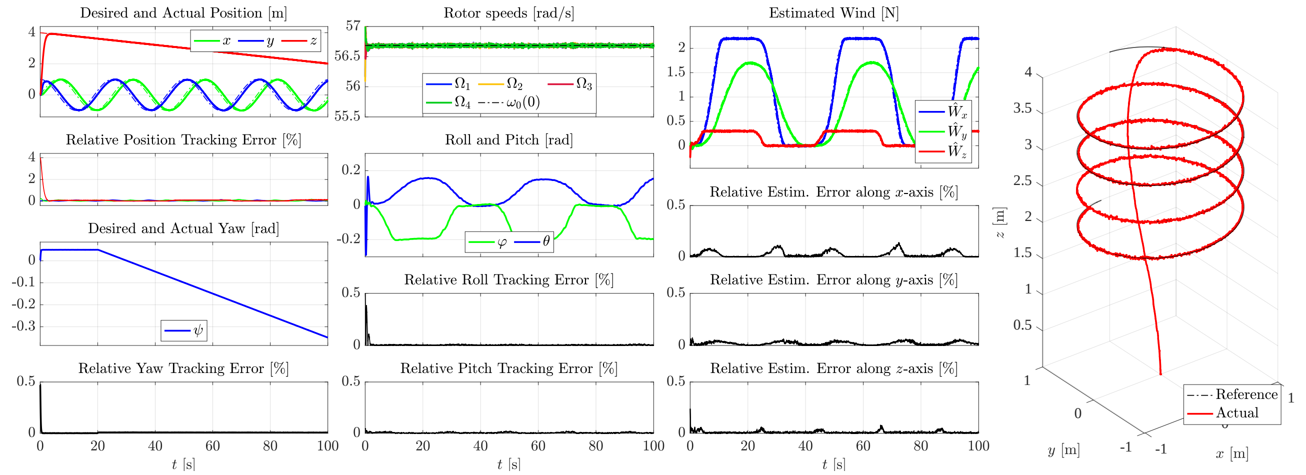

Figure 5: ROS/Gazebo SITL validation with a general wind gust of force components $N = (\bar{W}_x, \bar{W}_y, \bar{W}_z) = (-0.5, 0.8, 1.2)$ N. The UIO filter can estimate the wind with a high accuracy and the linear controller allows accurate tracking of a spiral trajectory.

proximately corresponds to a wind force of 9.7618 N. Fig. 7 illustrates the behavior of the controlled Erlecopter, closed in the loop with the proposed UIO filter and linear controller when a step wind force of $N = (11, 10, 7)^T$ N is instantaneously applied. It can be observed that even though our solution has been designed under the assumption of small roll φ and pitch θ angles, it can successfully handle also the large and sudden wind gust case. Steady-state roll and pitch angles are about 0.86 rad (49.27 degrees) and -0.94 rad (-53.85 degrees), respectively, which are clearly far from the small-angle hypothesis. An exact characterization of the region of attraction of the controlled system, attained by using our linear controller, is a task depending on the aircraft's mass, inertia, actuator saturation, etc, and is out of the scope of the present work. Most importantly, the simulation shows that such a region is not small and that the proposed solution has a large validity.

5.3. Experimental Validation

As the final step in the validation process, experiments on a real Erlecopter platform have been carried out within a scenario with a time-varying horizontal wind. The experimental setup includes an Erlecopter platform and a base station, which are connected to the local network. From a hardware perspective, the electronics of the aircraft consist of a Raspberry Pi 3 Model B computer board, four brushless DC motors, and four Electronic Speed Controllers (ESC) for regulating the propeller speeds. It also includes

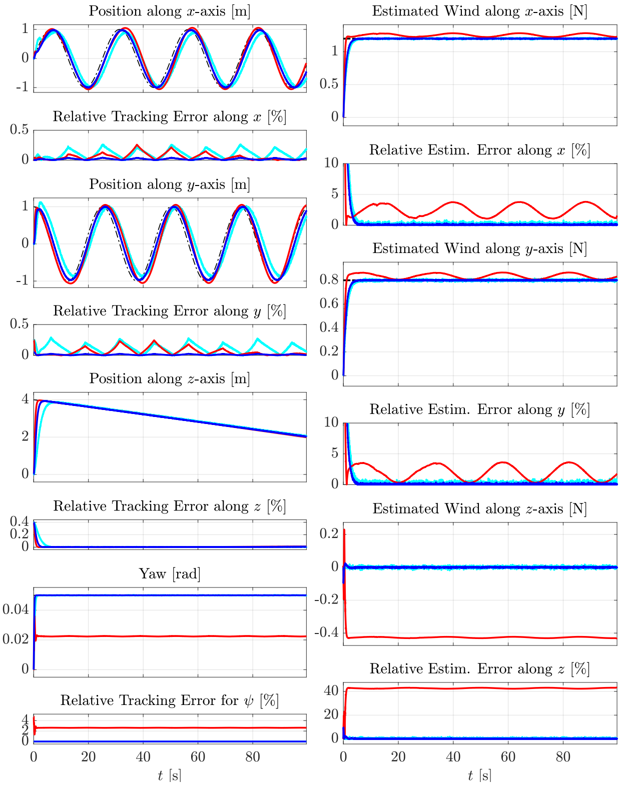

Figure 6: Estimation and tracking performance comparison obtained via the proposed method (blue), the hybrid solution of the UIO filter and the Ardupilot controller (cyan), and the MPC proposed in Kamel et al. (2017) (red), when the Erlecopter is required to hover at one point while affected by a horizontal wind force of constant amplitude $N = (1.2, 0.8, 0)$ N.

a WIFI dongle to communicate via TCP/IP with the base station, from which the quadrotor system receives the desired position (x_d, y_d, z_d, ψ_d), a built-in uBlox Neo-8M GPS sensor for the measurement of the center of mass position, (x, y, z), and an Inertial Measurement Unit (IMU) with a compass sensor for the measurement of the aircraft attitude (φ, θ, ψ). From a software viewpoint, the aircraft computer mounts a Raspbian OS running the following relevant processes: 1) an onboard ROS Indigo distribution acting as a server node and providing the ROS publish-subscribe abstraction layer for the other processes; 2) two publisher ROS nodes reading data from the GPS and IMU+Compass sensors, respectively, 3) a Matlab/Simulink-compiled ROS-node implementing the proposed UIO, already used during the ROS/Gazebo validation, which is subscribed to the desired aircraft pose specified by the base station and to the commanded rotor speeds, and which publishes the estimated wind force components, $(\hat{W}_x, \hat{W}_y, \hat{W}_z)$, and the desired pose, $(x^*_d, y^*_d, z^*_d, \psi^*_d)$ according to Eq. 33 and 36, and 4) an Erle-Brain Ardupilot node which is subscribed to correct the desired pose and publishes the commanded rotor speeds, $(\Omega_1, \Omega_2, \Omega_3, \Omega_4)$.

Moreover, the base station consists of a two-dimensional anemometer sensor and a PC. In particular, ground-truth information about the norm and direction of the horizontal wind speed is measured by an R30M 3-cup rotor sensor and a wind vane, respectively (see Fig. 1). Such information is sampled by a Campbell Scientific CR1000 datalogger and finally transferred to the PC via an RS232 connection port. The measured wind speed components, v_x and v_y, are converted to the corresponding wind force components, W_x and W_y, according to the known formula:

$$W_i = \rho \, S_x \, v_i^2 \,, \quad \text{for } i \in \{x, y\} \,, \tag{37}$$

where $\rho \approx 1.225$ Kg/m^3 is the air density at sea level, and the quadrotor lateral sections, for small roll and pitch, can be approximated as $S_x = S_y = 9.88 \cdot 10^{-3}$ m^2. The PC mounts a Linux OS and runs the following two relevant processes: 1) a ROS node acting as a remote client of the ROS server onboard the aircraft; an APM Ardupilot Mission Planner node, receiving the desired position and heading information specified by the end-user via its Graphical User Interface (GUI), and communicating it to the ROS server via MAVlink; 2) a subscriber-only ROS node saving synchronized data about the ground truth and the wind estimated by the UIO node. A block scheme depicting the above-described data flow is reported in Fig. 8.

The experiment aims at validating the effectiveness of the proposed approach in reality, within an outdoor scenario with a horizontal wind acting on the aircraft. Specifically, the scope is to highlight the differences from the simulated results and to assess the accuracy of the obtained wind estimation and attained pose. The experiment also aims at verifying the real feasibility of the hybrid solution, where the proposed UIO is used in combination with the existing Ardupilot controller. During the test, the quadrotor is required to take off and hover at around 0.5 meter-height and then reach the desired position $x_d = y_d = 0$, $z_d = 3$ m and $\psi_d = 3$ rad. The anemometer sensors are placed at around 1-meter distance and the same height as the desired hovering position. The results of the experiment are illustrated in Fig. 9 and Fig. 10. The small difference in the rotor speeds, which appears in pairs, is coherent with that observed during the ROS/Gazebo-based simulation of Fig. 4, in which the Erlecopter is also hovering. This confirms that the CAD and dynamic models used for the Erlecopter are very accurate. The UIO filter can successfully estimate the horizontal wind force components with an average relative estimation error of about 1.12% and 0.78% along the x and y-axis. The rotor speeds commanded by the Ardupilot controller are also coherent with the simulations. The obtained average relative tracking errors along x, y, z, and ψ are 0.26%, 0.27%, 0.18%, and 0.17%, respectively, thus showing good accuracy of the method. It is very important to notice that the actual feasibility of this hybrid solution proves a possible rapid technology transfer to existing comment platforms.

While a real experiment with a wind force signal belonging to the storm category is not easy to be performed, the control signals commanded by the proposed approach are feasible even with a low-cost aircraft system, such as the Erlecopter. To show this, consider a wind with a speed of 28.4

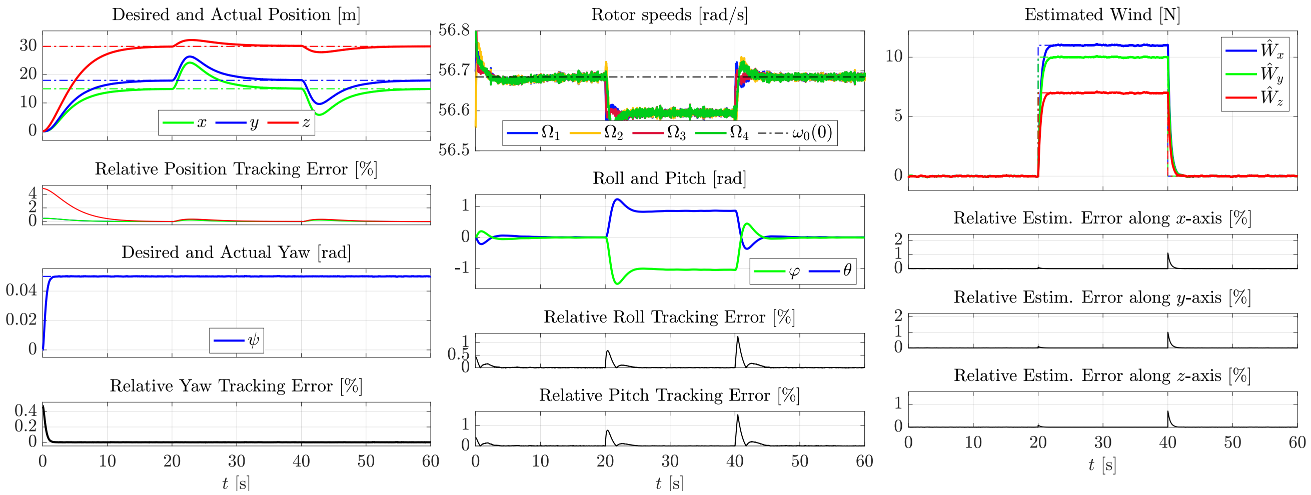

Figure 7: ROS/Gazebo-based testing of the controlled Erlecopter, closed in the loop with the proposed proposed UIO and linear controller, under the occurrence of a sudden and large step-wise wind. A step-wise wind force signal of storm category with magnitude $N = (11, 10, 7)^T$ N is successfully reconstructed and its effect compensated. Controller gains are $\lambda_a = 0.4$ and $\lambda_p = 4$. Steady-state roll and pitch angles are about 0.86 rad (49.27 degrees) and -0.94 rad (-53.85 degrees), respectively, which are clearly far from the small angle hypothesis. This reveals a large applicability of the proposed method.

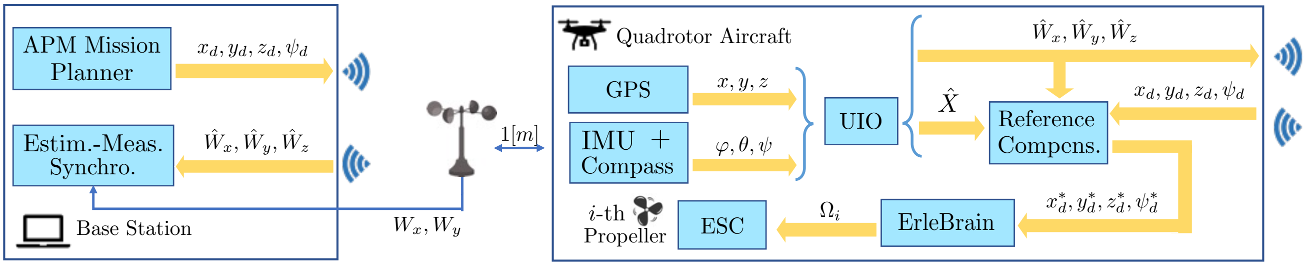

Figure 8: Data-flow in the experimental setup on the aircraft system and the base station. ROS nodes and topics are represented by blocks and arrows, respectively. Ground-truth information about the norm and direction of the horizontal wind-speed is measured by a 3-cup rotor sensor and a wind vane, placed at around 1-meter distance and the same height of the desired hovering position.

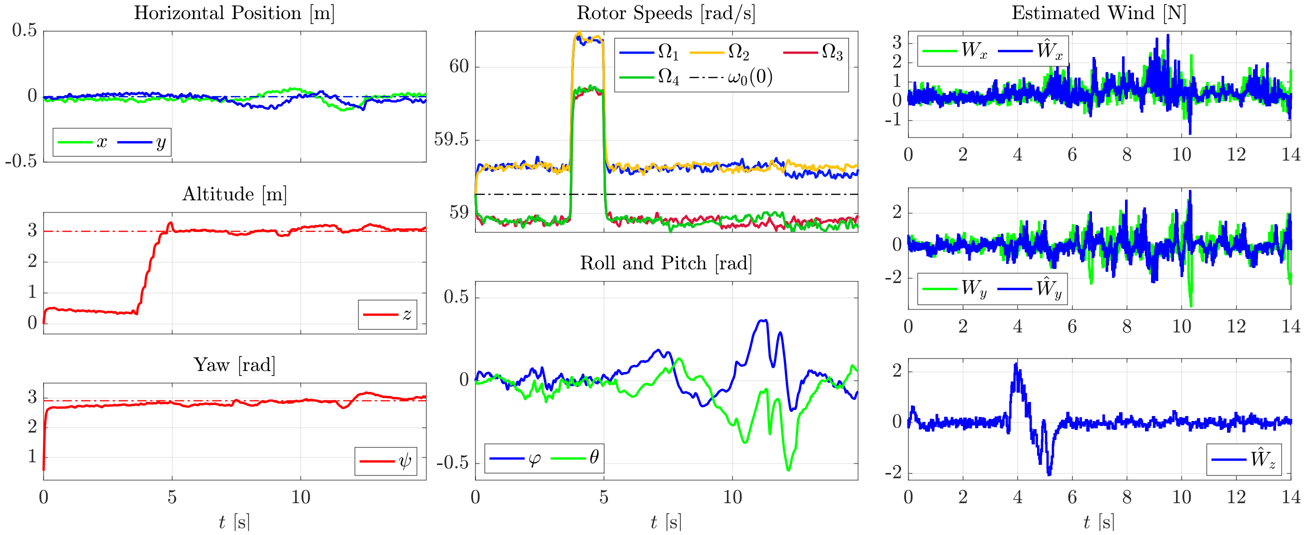

Figure 9: Experimental results during a point-to-point motion task where the aircraft autonomously hovers at 0.5 meters, then quickly reaches 3 meters, and finally continues to hover. Apart from the very fast translation phase (for $t \in [3.6, 5.1]$ s), during which the quasi-hovering assumption is temporarily violated and the UIO filter detects a virtual upward vertical force, the filter itself correctly estimates the wind components as expected and the proposed controller maintains the desired position.

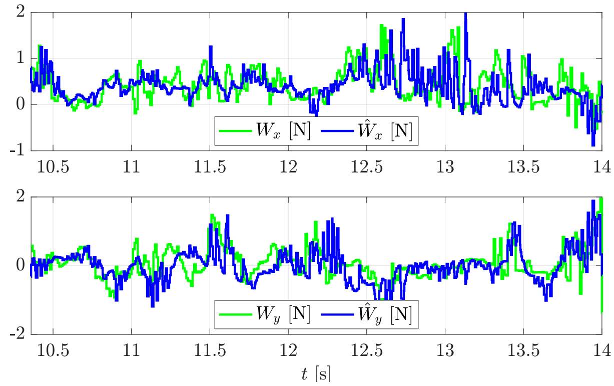

Figure 10: Experimental results during a hovering task. The UIO filter shows a sufficient level of accuracy in the wind estimation.

m/s, blowing for simplicity along the positive direction of x-axis in $\mathcal{F}_0$. Given the formula in Eq. 37, the corresponding wind force $\bar{W}_x$ affecting the Erlecopter is about 9.76 N. At the equilibrium, the resultant of the gravity force (oriented towards the negative direction of the z-axis of $\mathcal{F}_0$), the thrust force F (oriented towards the positive direction of the z-axis of $\mathcal{F}_b$), and the wind force, having an amplitude of $\bar{W}_x$ and pointing by assumption towards the positive direction of the x-axis of $\mathcal{F}_0$ must be null. Assuming without loss of generality $\psi = 0$, it must hold $\bar{\varphi} = 0$ rad, $\bar{\theta} = \arctan(\bar{W}_x/(mg))$, and $\bar{F} = \sqrt{(mg)^2 + \bar{W}_x^2}$. Given the aircraft mass $m = 1.12$ Kg, the steady-state pitch and thrust values are $\bar{\theta} = 0.726$ rad and $\bar{F} = 14.70$ N. Moreover, since at the equilibrium all rotors must be spinning at the same velocity (to have zero torques), the total thrust $\bar{F}$ is shared among the four rotors, each of which has to contribute with $\bar{F}/4 = 3.7$ N. For the thrust constant value of $k_F = 8.5 \cdot 10^{-4}$ N/rad^2, this means that each rotor spins at 65.6 rad/s. In this respect, the Erlecopter system is provided with four 5-V DC brushless motors with propellers, which can spin at 920 rpm/V, which is about 96 rad/s per volt. Thus, the above-required speed attained in simulation is also feasible in reality, thereby confirming that a platform such as the Erlecopter can compensate for these strong winds.

6. Conclusion

This paper presents a novel approach for wind gust estimation and quasi-hovering position control of a quadrotor aircraft. The proposed solution consists of a linear dynamic filter for the estimation of the unknown wind which is acting as an exogenous disturbance input and requires only position and attitude information. The control scheme for both horizontal and general wind gusts is described. The desired position is modified to consider estimated wind gusts. This is performed by forcing the linearized dynamics to a given dynamics and deriving the novel desired position that takes into account the wind. Finally, this novel reference position feeds a position controller that provides the desired attitude of the quadrotor. One appealing feature of the described unknown input observer filter is that it is independent of the recovery control scheme used to nullify the tracking error, as long as the applied rotor speeds are available. The application of the proposed solution has been proved first in simulated environments, using Simulink to implement the controller, the Gazebo software to obtain physically accurate motion simulation of the aircraft, and the Robot Operating System middleware to allow their intercommunication; then it has been also validated in experimental setups which proved the effectiveness and high accuracy of the method.

Future works will consider the dynamic motion control for an aircraft that is required to execute fast translations while strong wind gusts are blowing. Under these circumstances, it is expected that the linearized model does not describe accurately enough the behavior of the aircraft, and a nonlinear model, including actuator delays and saturation, must be used instead. This would require the exploitation of more computationally involved approaches for deriving reliable nonlinear unknown state-input observers. Another feature to be introduced when dealing with such a more complex scenario is a wind prediction scheme, which may require learning the actual wind model from estimates obtained online from the unknown input observer. Obviously, such desired estimation and control solutions for more dynamic motion tasks will require more computation power and thus also larger energy storage than that normally found in low-cost commercial aircraft.

A. Appendix

This section reports the details about the derivation of the two UIO filters:

1) Invertibility Condition: When designing the UIO as described in Sundaram and Hadjicostis (2007), the rank of $(B_N^T, D_N^T)^T$ must be full and L has to be determined as the smallest delay such that $\mathrm{rank}(\mathbb{H}^L) = \mathrm{rank}(\mathbb{H}_N^L) + m_N$ is satisfied. For the case of this paper, L =2 as it holds $\mathrm{rank}(\mathbb{H}^2) - \mathrm{rank}(\mathbb{H}^1) = 3$ or explicitly

$$\mathrm{rank}\begin{pmatrix} D_N & 0 & 0 \\ CB_D & D_N & 0 \\ CAB_N & CB_N & D_N \end{pmatrix} - \mathrm{rank}\begin{pmatrix} D_N & 0 \\ CB_N & D_N \end{pmatrix} = 3\,.$$

2) Matrices E and F: To find matrices E and F, it is essential to determine a matrix $\bar{N}$ whose rows form a basis of the left nullspace of $\mathbb{H}^1$. In this case, $\bar{N} = I_{12\times 12}$ and thus matrix N has the form $N = W\begin{pmatrix} I_P & 0 \\ 0 & \bar{N} \end{pmatrix}$, where $p = 6$. Furthermore, W has to satisfy the condition

$$\begin{pmatrix} 0 \\ I \end{pmatrix} = N \begin{pmatrix} D_N \\ CB_N \\ CAB_N \end{pmatrix} = W \begin{pmatrix} I_P & 0 \\ 0 & \bar{N} \end{pmatrix} \begin{pmatrix} D_N \\ CB_N \\ CAB_N \end{pmatrix}$$
$$= W \underbrace{\begin{pmatrix} 0_{12\times 3} \\ (T_s/m) I_{3\times 3} \\ 0_{3\times 3} \end{pmatrix}}_{H}\,.$$

A choice for matrix W such that its last $m_N = 3$ rows are a left inverse of matrix H and its first $2n - m_N = 21$ rows form a basis of the left nullspace of H is

$$W = \begin{pmatrix} I_{12\times12} & 0_{12\times3} & 0_{12\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & m/T_{3\times3} & 0_{3\times3} \end{pmatrix}.$$

From the above equation, it follows:

$$N = \begin{pmatrix} I_{12\times12} & 0_{12\times3} & 0_{12\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & m/T_{3\times3} & 0_{3\times3} \end{pmatrix} \begin{pmatrix} I_p & 0 \\ 0 & \bar{N} \end{pmatrix}.$$

As a next step, the following decomposition is done

$$\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = N \begin{pmatrix} C \\ CA_d \\ CA_d^2 \end{pmatrix},$$

where S_2 is made of the last $m_N = 3$ rows of the above matrix and S_1 is the remainder ones. Finally $\hat{F}_1$ is determined so that $E = A - BS_2 - \hat{F}_1\, S_1$ is Schur. As the quadrotor eigenvalues are unstable, it is necessary to find $\hat{F}_1$, e.g. by using the Matlab command $place((A - BS_2)', S_1, p)'$, in order to allocate all eigenvalues at the locations specified by a vector p. Finally, the input-state observer dynamic matrix is

$$E = \begin{pmatrix} 0_{9\times9} & 0_{9\times3} \\ 0_{3\times9} & E_{22} \end{pmatrix},\ E_{22} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

the output injection matrix F is, as per Eq. 8, given by

$$F = \begin{pmatrix} 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \frac{I_{3\times3}}{3} & 0_{3\times3} & \frac{I_{3\times3}}{3} \\ 0_{3\times3} & 0_{3\times3} & \frac{I_{3\times3}}{10T} & 0_{3\times3} \\ 0_{3\times3} & \frac{I_{3\times3}}{20T} & 0_{3\times3} & \frac{I_{3\times3}}{20T} \end{pmatrix},$$

and the input decoupling matrix of Eq. 11 is

$$G = \begin{pmatrix} 0_{2\times3} & \frac{m}{T} I_{2\times2} & 0_{2\times13} \end{pmatrix}.$$

Acknowledgments

Authors wish to thank Dr. Eng. Maja Trumić for insightful advice and useful help.

References

Abeywardena, D., Wang, Z., Dissanayake, G., Waslander, S.L., Kodagoda, S., 2014. Model-aided state estimation for quadrotor micro air vehicles amidst wind disturbances, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 4813–4818.

Ahmed, N., Chen, M., 2018. Sliding mode control for quadrotor with disturbance observer. Advances in Mechanical Engineering 10.

Al-Bayati, A.H., Skaf, Z., 2010. A comparative study of linear observers applied to a dc servo motor, in: Proceedings of the 2010 International Conference on Modelling, Identification and Control, IEEE. pp. 785–790.

Ball, D., Ross, P., English, A., Milani, P., Richards, D., Bate, A., Upcroft, B., Wyeth, G., Corke, P., 2017. Vision-based robotics for broad-acre agriculture. IEEE Robotics & Automation Mag. 1070.

Beard, R.W., McLain, T.W., 2012. Small Unmanned Aircraft: Theory and Practice. Princeton University Press, Princeton, NJ, USA.

Bergamasco, M., Lovera, M., 2014. Identification of linear models for the dynamics of a hovering quadrotor. IEEE Transactions on Control Systems Technology 22, 1696–1707.

Bhattacharyya, S., 1978. Observer design for linear systems with unknown inputs. IEEE transactions on Automatic Control 23, 483–484.

Bouabdallah, S., Murrieri, P., Siegwart, R., 2004. Design and control of an indoor micro quadrotor, in: IEEE International Conference on Robotics and Automation, IEEE. pp. 4393–4398.

Bruschi, P., Piotto, M., Dell'Agnello, F., Ware, J., Roy, N., 2016. Wind speed and direction detection by means of solid-state anemometers embedded on small quadcopters. Procedia Engineering 168, 802–805.

Burri, M., Dätwiler, M., Achtelik, M.W., Siegwart, R., 2015. Robust state estimation for micro aerial vehicles based on system dynamics, in: Intl. Conference on Robotics and Automation, IEEE. pp. 5278–5283.

Cabecinhas, D., Cunha, R., Silvestre, C., 2015. A globally stabilizing path following controller for rotorcraft with wind disturbance rejection. IEEE Trans. on Control Systems Tech. 23, 708–714.

Das, A., Lewis, F., Subbarao, K., 2009a. Backstepping approach for controlling a quadrotor using lagrange form dynamics. Journal of Intelligent and Robotic Systems 56, 127–151. doi:10.1007/s10846-009-9331-0.

Das, A., Subbarao, K., Lewis, F., 2009b. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. IET control theory & applications 3, 303–314.

Etele, J., 2006. Overview of wind gust modelling with application to autonomous low-level uav control. Mechanical and Aerospace Engineering Department, Carelton University, Ottawa, Canada .

Fethalla, N., Saad, M., Michalska, H., Ghommam, J., 2018. Robust observer-based dynamic sliding mode controller for a quadrotor uav. IEEE Access 6, 45846–45859.

Grzonka, S., Grisetti, G., Burgard, W., 2012. A fully autonomous indoor quadrotor. IEEE Transactions on Robotics 28, 90–100.

Guerrero, J.A., Escareño, J.A., Bestaoui, Y., 2013. Quad-rotor mav trajectory planning in wind fields, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 778–783.

Guo, Y., Jiang, B., Zhang, Y., 2018. A novel robust attitude control for quadrotor aircraft subject to actuator faults and wind gusts. IEEE/CAA Journal of Automatica Sinica 5, 292–300. doi:10.1109/JAS.2017.7510679.

Hostetter, G., Meditch, J., 1973. Observing systems with unmmeasurable inputs. IEEE Transactions on Automatic Control 18, 307–308.

Izaguirre-Espinosa, C., Muñoz-Vázquez, A.J., Sánchez-Orta, A., Parra-Vega, V., Castillo, P., 2016. Attitude control of quadrotors based on fractional sliding modes: theory and experiments. IET Control Theory & Applications 10, 825–832.

Jackson, S., Tisdale, J., Kamgarpour, M., Basso, B., Hedrick, J.K., 2008. Tracking controllers for small uavs with wind disturbances: Theory and flight results, in: Conf. on Dec. and Cont., IEEE. pp. 564–569.

Kamel, M., Stastny, T., Alexis, K., Siegwart, R., 2017. Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system, in: Koubaa, A. (Ed.), Robot Operating System (ROS) The Complete Reference, Volume 2. Springer.

Kiyak, E., Çetin, Ö., Kahvecioğlu, A., 2008. Aircraft sensor fault detection based on unknown input observers. Aircraft Engineering and Aerospace Technology 80, 545–548.

Kobayashi, N., Nakamizo, T., 1982. An observer design for linear systems with unknown inputs. International Journal of Control 35, 605–619.

Kudva, P., Viswanadham, N., Ramakrishna, A., 1980. Observers for linear systems with unknown inputs. IEEE Transactions on Automatic Control 25, 113–115.

L'Afflitto, A., Anderson, R.B., Mohammadi, K., 2018. An introduction to nonlinear robust control for unmanned quadrotor aircraft: How to design control algorithms for quadrotors using sliding mode control and adaptive control techniques [focus on education]. IEEE Control Systems Magazine 38, 102–121.

Lee, S., Har, D., Kum, D., 2016. Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion, in: Computer Science and Engineering (APWC on CSE), 2016 3rd Asia-Pacific World Congress on, IEEE. pp. 84–89.

Lindsey, Q., Mellinger, D., Kumar, V., 2011. Construction of cubic structures with quadrotor teams. Proc. Robotics: Science & Systems VII .

Liu, H., Liu, D., Zuo, Z., 2016. Robust attitude control for uncertain quadrotors with input time delays, in: European Control Conference, IEEE. pp. 2312–2315.

Luenberger, D., 1966. Observers for multivariable systems. IEEE Transactions on Automatic Control 11, 190–197.

Lyu, P., Liu, S., Lai, J., Liu, J., 2019. An analytical fault diagnosis method for yaw estimation of quadrotors. Control Engineering Practice 86, 118–128. URL: https://www.sciencedirect.com/science/article/pii/S0967066118306038, doi:https://doi.org/10.1016/j.conengprac.2019.03.010.

Mahony, R., Kumar, V., Corke, P., 2012. Multirotor aerial vehicles. IEEE Robotics and Automation magazine 20.

Mannucci, A., Nardi, S., Pallottino, L., 2017. Autonomous 3d exploration of large areas: a cooperative frontier-based approach, in: International Conference on Modelling and Simulation for Autonomous Systems, Springer. pp. 18–39.

Michael, N., Mellinger, D., Lindsey, Q., Kumar, V., 2010. The graps multiple micro-uav testbed. IEEE Robotics & Automation Magazine 17, 56–65.

Michael, N., Shen, S., Mohta, K., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., et al., 2014. Collaborative mapping of an earthquake damaged building via ground and aerial robots, in: Field and Service Robotics, Springer. pp. 33–47.

Military Specification, 1980. Flying qualities of piloted airplanes. United States Department of Defense 2, 3.

Miller, R., Mukundan, R., 1982. On designing reduced-order observers for linear time-invariant systems subject to unknown inputs. International Journal of control 35, 183–188.

Mudaliar, H.K., Kumar, D.M., Azid, S., Cirrincione, M., Fagiolini, A., Pucci, M., 2018. Dynamical compensation of the load torque in a high-performance electrical drive with an induction motor, in: Intl. Conf. on Electrical Machines and Systems, Jeju (Korea).

Nekoukar, V., Mahdian Dehkordi, N., 2021. Robust path tracking of a quadrotor using adaptive fuzzy terminal sliding mode control. Control Engineering Practice 110, 104763. URL: https://www.sciencedirect.com/science/article/pii/S096706612100040X, doi:https://doi.org/10.1016/j.conengprac.2021.104763.

Nonami, K., Kendoul, F., Suzuki, S., Wang, W., Nakazawa, D., 2010. Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles. 1st ed., Springer.

Norouzi Ghazbi, S., Aghli, Y., Alimohammadi, M., Akbari, A.A., 2016. Quadrotors unmanned aerial vehicles: A review. International Journal on Smart Sensing and Intelligent Systems 9, 309–333. doi:10.21307/ijssis-2017-872.

Nuske, S., Choudhury, S., Jain, S., Chambers, A., Yoder, L., Scherer, S., Chamberlain, L., Cover, H., Singh, S., 2015. Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers. Journal of Field Robotics 32, 1141–1162.

Palunko, I., Cruz, P., Fierro, R., 2012. Agile load transportation: Safe and efficient load manipulation with aerial robots. IEEE robotics & automation magazine 19, 69–79.

Pi, C.H., Hu, K.C., Cheng, S., Wu, I.C., 2020. Low-level autonomous control and tracking of quadrotor using reinforcement learning. Control Engineering Practice 95, 104222. URL: https://www.sciencedirect.com/science/article/pii/S0967066119301923, doi:https://doi.org/10.1016/j.conengprac.2019.104222.

Rios, H., Falcon, R., Gonzalez, O.A., Dzul, A., 2018. Continuous sliding-modes control strategies for quad-rotor robust tracking: Real-time application. IEEE Transactions on Industrial Electronics .

Sabatino, F., 2015. Quadrotor control: modeling, nonlinearcontrol design, and simulation.

Saeedi, S., Nagaty, A., Thibault, C., Trentini, M., Li, H., 2016. 3d mapping and navigation for autonomous quadrotor aircraft, in: IEEE 29th Canadian Conf. on Electrical and Computer Engineering.

Sariyildiz, E., Mutlu, R., Zhang, C., 2019. Active disturbance rejection based robust trajectory tracking controller design in state space. Journal of Dynamic Systems, Measurement, and Control 141. doi:10.1115/1.4042878.

Sariyildiz, E., Ohnishi, K., 2013. Analysis the robustness of control systems based on disturbance observer. International Journal of Control 86, 1733–1743. doi:10.1080/00207179.2013.795663.

Satici, A.C., Poonawala, H., Spong, M.W., 2013. Robust optimal control of quadrotor uavs. IEEE Access 1, 79–93.

Seminara, G., Fontanelli, D., 2017. First responders robotic network for disaster management, in: International Conference on Modelling and Simulation for Autonomous Systems, Springer. pp. 350–373.

Sun, S., Wang, X., Chu, Q., d. Visser, C., 2020. Incremental nonlinear fault-tolerant control of a quadrotor with complete loss of two opposing rotors. IEEE Transactions on Robotics (Early Access) , 1–15.

Sundaram, S., 2012. Fault-tolerant and secure control systems. University of Waterloo, Lecture Notes .

Sundaram, S., Hadjicostis, C.N., 2007. Delayed observers for linear systems with unknown inputs. IEEE Transactions on Automatic Control 52, 334–339. doi:10.1109/TAC.2006.890371.

Sundaram, S., Hadjicostis, C.N., 2008. Partial state observers for linear systems with unknown inputs. Automatica 44, 3126–3132.

Taha, W., Al-Durra, A., Errouissi, R., Al-Wahedi, K., 2018. Nonlinear disturbance observer-based control for quadrotor uav, in: IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, IEEE. pp. 2589–2595.

Tognon, M., Gabellieri, C., Pallottino, L., Franchi, A., 2018. Aerial co-manipulation with cables: The role of internal force for equilibria, stability, and passivity. IEEE Robotics and Automation Letters 3, 2577–2583.

Tomić, T., Lutz, P., Schmid, K., Mathers, A., Haddadin, S., 2020. Simultaneous contact and aerodynamic force estimation (s-cafe) for aerial robots. The International Journal of Robotics Research 39, 688–728.

Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I.L., Ruess, F., Suppa, M., Burschka, D., 2012. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. IEEE Robotics & Aut. Mag. 19, 46–56.

Van Nieuwstadt, M.J., Murray, R.M., 1998. Real-time trajectory generation for differentially flat systems. Intl. Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal 8, 995–1020.

Wang, C., Song, B., Huang, P., Tang, C., 2016. Trajectory tracking control for quadrotor robot subject to payload variation and wind gust disturbance. J. of Intell. & Robotic Syst. 83, 315–333.

Wang, S.H., Wang, E., Dorato, P., 1975. Observing the states of systems with unmeasurable disturbances. IEEE transactions on Automatic Control 20, 716–717.

Yang, J., Cai, Z., Lin, Q., Wang, Y., 2013. Self-tuning pid control design for quadrotor uav based on adaptive pole placement control, in: Chinese Automation Congress (CAC), 2013, IEEE. pp. 233–237.

Yang, X., Huang, Y., 2009. Capabilities of extended state observer for estimating uncertainties, in: 2009 American control conference, IEEE. pp. 3700–3705.

Yang, Z., Rasmussen, K.B., Kieu, A.T., Izadi-Zamanabadi, R., 2011. Fault detection and isolation for a supermarket refrigeration system–part two: Unknown-input-observer method and its extension. IFAC Proceedings Volumes 44, 4238–4243.

Yesildirek, A., Imran, B., 2014. Nonlinear control of quadrotor using multi lyapunov functions, in: American Control Conference, IEEE. pp. 3844–3849.

Zhang, C., Kovacs, J.M., 2012. The application of small unmanned aerial systems for precision agriculture: a review. Precision agriculture 13, 693–712.

Sheikh Izzal Azid is a Lecturer in the School of Engineering and Physics at the University of the South Pacific. His Ph.D. thesis is in the area of

Aerial Robotics and Robust Control from the University of the South Pacific in collaboration with the University of Palermo, Italy. He attained MSc in Engineering in 2010 in the area of Automation and Intelligent systems. He is a recipient of the Erasmus+ staff exchange program and he was attached at the Technical University of Sofia. His research interest includes Robotics, Controls, UAVs, Autonomous systems, State Observers, and Microprocessor Applications. He is a member of IEEE and Vice President of IEEE VTS New Zealand North. He has also worked as a consultant for several European Projects on Science, Technology, and Innovation in the Pacific.

Krishneel Kumar is a Ph.D. student in Science and Technologies for Electrical Engineering and Complex Systems for Mobility at the University of Genova, Italy. He obtained his Master of Science in Engineering in 2019 in the field of robotics and control at the University of the South Pacific He also attained the Bachelor of Engineering degree in Electrical and Electronics in 2017, where he was awarded a gold medal.

Maurizio Cirrincione (born 2 May 1961, Italian citizen), IEEE Senior Member (2010) received the Laurea's degree from the Politecnico di Turino, Turin, Italy, in 1991, and the Ph.D. degree from the University of Palermo, Italy, in 1996, both in electrical engineering. From 1996 to 2005, he was a Researcher with the ISSIA-CNR (Institute on Intelligent Systems for Automation) in Palermo. Since 2005 he has been a Full Professor of Control Systems at the UTBM in Belfort, France, and a member of the IRTES Laboratory and the FCLab (Fuel Cell laboratory) in Belfort, France, and since 2016 he has been 1st Class Full Professor. Currently, he is the Head of the School of Information Technology, Engineering, Mathematics and Physics of the University of the South Pacific in Suva, Fiji Islands. He was awarded the 1997 "E.R.Caianiello" prize in 1997 for the best Italian Ph.D. thesis on neural networks. He is the author of over 170 papers, most of which are on high impact factor journals, and of 2 books.

Adriano Fagiolini is an Assistant Professor at the University of Palermo, Italy. He received an M.S. degree in Computer Science Engineering in 2004 and a Ph.D. degree in Robotics and Automation in 2009 from the University of Pisa. He has been a Visiting Researcher at the Department of Energy, IUT Longwy Université de Lorraine (France) in 2019, and at the Department of Mechanical Engineering, University of California at Riverside, in 2015 and 2017. He enrolled in the Summer Student Programme at the European Center for Nuclear Research (CERN), Geneva, in 2002. In 2008, he led the team of the University of Pisa during the first European Space Agency's Lunar Robotics Challenge, which resulted in a second-place prize for the team. He was one of the recipients of the IEEE ICRA Best Manipulation Paper Award in 2005.