Hovering stabilization of the DarkO tail-sitter drone with constant wind

Florian Sansou, Fabrice Demourant, Gautier Hattenberger, Thomas Loquen, Luca Zaccarian

Hovering stabilization of the DarkO tail-sitter drone with constant wind

Florian Sansou[1], Fabrice Demourant[2,1], Gautier Hattenberger[1], Thomas Loquen[2,1], Luca Zaccarian[3]

***Abstract*— We present two mathematical models of the DarkO tail-sitter convertible UAV developed and 3D printed at the École Nationale de l'Aviation Civile (ENAC), in Toulouse (France). During a hover flight, the UAV is vertical, which offers a large wing area facing the wind. Thus, aerodynamic disturbances have a strong influence on the nonlinear UAV dynamics. Our models capture this behavior and allow us to characterize relevant equilibria, in the presence of a constant wind, and the corresponding wind-dependent linearized dynamics. Using a parametric family of models, we design an optimality-based robust static output feedback controller that uses two degrees of freedom on the orientation to stabilize the UAV in a hovering condition in spite of an unknown constant wind. This control law has been implemented in the Paparazzi autopilot software to obtain experimental results that validate our theory.**

***Index Terms*— Unmanned aerial vehicles, Nonlinear dynamical systems, Linearization, Application of robust control**

I. Introduction

Convertible drones have the ability to take off or land vertically and fly like an airplane. They are promising architectures providing energy efficient flight capabilities for strong endurance, as compared with classical coplanar UAVs, like quadcopters. This high degree of autonomy is highly desirable in visual detection applications (e.g., surveillance, aerial photography, etc.), but also for environmental exploration and physical interaction.

Several organizations and researchers have worked on dual-rotor tail-sitter aerial vehicles. One example is a dual-rotor tail-sitter named "T-Wing" [1], [2], another tail-sitter named "MavIon" [3], or the "JLion" and "KH-Lion" [4]. These UAVs share a similar architecture based on a wing supporting two motors on the leading edges and blowing two elevons located on the trailing edge. This architecture provides greater robustness than tilt rotors, which require more moving parts, making them more fragile, and a powerful actuator to turn the engine and propeller assembly. Nevertheless, the electromechanical design and the ensuing control system of dual rotor tail-sitters is still an active research area. As the studies show [5], [6], the inherent complexity of these architectures requires modelling work due to the many non-linearities and couplings involved, especially in terms of modelling the aerodynamic effects. In this context, the aerodynamic interference between the fixed wing and rotors has been modeled in [7]–[9], and the propeller forces and moments generated at high angles of attack is addressed in [10]. However, these models are complex and only partially usable for control design.

Another important point is the representation of the drone attitude. It is possible to represent its orientation with Euler angles [11]–[13], which allows for an intuitive understanding, but a singularity appears in certain flight phases. Given the high degree of maneuverability, it is preferable to represent the attitude with a unit quaternion, eliminating any singularity [14]. Numerous publications model the aerodynamic effects as a function of the angle of attack and sideslip generated by the propellers [15], [16]. It is possible to choose another model for the aerodynamic interactions between the engines, wings, and elevons, as presented in [17]. The modelling technique in [17] allows having a global model covering the whole flight envelope, thanks to the so-called Φ-theory approach. Although the Φ-theory approach fails to predict the abrupt drop in the lift force with an increasing angle of attack (AoA) (which is caused by turbulent airflow) [18] it allows representing the drone accurately enough to capture the behavior during aggressive maneuvers.

Typical tailsitter UAV control methods are based on the separation of flight modes: hovering and forward flight. The critical part is in the transition between the two flight modes, where the drone has to remain balanced by transitioning through a succession of equilibria. A PID feedback is used in [19], [20] to stabilize the attitude in hovering mode, then switch to another controller for transitioning and finally to a third forward flight controller. However, the wind effect is not directly addressed and is seen as an external disturbance. Another strategy for controlling UAVs is based on two decoupled speed and attitude controllers [21], which can be extended by a cascaded hierarchical controller [22], [23] for increased robustness, but due to the lack of a hierarchical actuator structure, these controllers are not associated with a stability proof for the full model. Also, [24] proposes a nonlinear controller with a command allocation method, enabling the drone to be controlled over the entire flight envelope, but once again their method does not come with a proof. Finally, [25] proposed an $\mathcal{L}_1$ adaptive control scheme taking into account saturation in a linearised setting. However, the impact of wind on the dynamics is not addressed directly in

Research supported in part by ONERA-ISAE-ENAC Research Federation (FONISEN - PCGN) and by Occitanie region.

[1] Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France, (e-mail: firstname.lastname@enac.fr).

[2] DTIS, ONERA, Université de Toulouse, 31000, Toulouse, France (e-mail: firstname.lastname@onera.fr).

[3] Department of Industrial Engineering, University of Trento, Italy, and LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, (e-mail: zaccarian@laas.fr).

any of these schemes.

A model-free control technique has been used in [26], which allows avoiding the cumbersome modelling effort, but is prone to the well-known improvements stemming from an accurate model-aware solution. When the model is taken into account, one may rely on nonlinear dynamic inversion (NDI) and incremental nonlinear dynamic inversion (INDI) techniques (see, e.g., [18], [27], [28]). One of the limitations of this type of control is the lack of a proof of convergence.

The wind effect is rarely addressed in the above-mentioned solutions. The main problem lies in the complexity of wind measurements, wich is adressed in [29] by employing a wind estimation method for tail-sitter vehicles in any flight phase, and also in [30] where a disturbance observer is proposed to enhance the hovering accuracy in the presence of external disturbances such as crosswinds. Wind estimation is well illustrated in [31], which describes a test campaign conducted in two locations: an indoor wind generator in a controlled environment and an outdoor setting where the wind conditions are unknown.

In this context, the contributions of this paper extend our preliminary works [32]–[34] and can be summarized as follows. First, we derive two models: an accurate non linear model partially presented in the Master thesis [32] and exploiting the modeling approach of [17], [26], and an input-affine model generalize the one discussed in [33], [34] for the case with zero wind. Secondly, for the case of constant wind, we charaterize a set of equilibria that generalize the zero-wind findings of [32], [33] and also explain the undesirable experimental results reported in [34]. Then, for any such equilibrium, we explicitly compute the linearized dynamics of a suitably rotated motion, whose essential bahavior is parametrized by a two parameters, associated with the horizontal and vertical wind intensity. Based on this parametric linearized dynamics, we then design a robust static output feedback controller combined with a specific integral action, capable of stabilizing the hovering condition in the presence of an unknown constant wind. Our solution extends the longitudinal-only (3 DOF) scheme of [34] to the full 6 DOF pose, where the UAV naturally turns so as to face the wind and attains the hovering condition.

The paper is organized as follows. In Section II, we discuss the models, desirable equilibria, and the ensuing linearizations. Then in Section III we describe the control law and two tuning methods illustrating via simulation the importance of our parametrized linearizations. Finally, in Section IV, we validate the controller with experimental indoors flight tests of the DarkO prototype in front of an open-air wind tunnel.

Notation. Given two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we often denote their juxtaposition as $(\boldsymbol{x}_1, \boldsymbol{x}_2) := [\boldsymbol{x}_1^\top \ \boldsymbol{x}_2^\top]^\top$. Given any vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^3$, the skew-symmetric matrix $[\boldsymbol{u}]_\times \in \mathbb{R}^{3\times 3}$ satisfies $[\boldsymbol{u}]_\times \boldsymbol{v} = \boldsymbol{u} \times \boldsymbol{v}$. The symbol $\mathbb{I}_n$ denotes the identity matrix of dimension n and $\mathrm{diag}(x_1, \cdots, x_p)$ denotes a diagonal matrix whose diagonal elements are $x_1, \cdots, x_p$. We introduce the classical notation for a transfer matrix $s \mapsto T_{v\to z}(s)$ from a certain input v to a certain output z.

II. Model of the DarkO UAV

The DarkO UAV, designed and developed at the Ecole Nationale de l'Aviation Civile (ENAC) in Toulouse (France), is a clear example of a convertible UAV with a so-called tail-sitter architecture. DarkO is assembled from multiple 3D printed Onyx parts (a highly robust material comprising omnidirectional carbon fibres). All surfaces are interlocked on a single axis, so that the drone can be easily disassembled for parts replacement or to gain access to the on-board electronics. The on-board autopilot is an Apogee [1] board manufactured at ENAC, see Fig. 1. The autopilot provides the option of

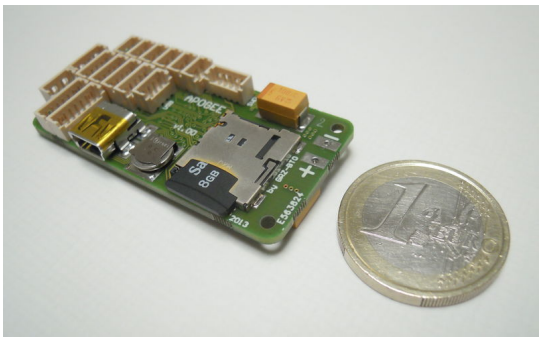

Fig. 1. Apogee v1.00 top view.

recording the on-board data on an SD memory card, at the control frequency of 500 Hz, thus allowing for effective post-processing of acquired data. The communication protocol used between the autopilot and the Electronic Speed Controllers (ESCs) is Dshot 600. The ESCs are AIKON AK32 35A flash with an AM32 firmware. The ground-to-board communication is performed via a bidirectional channel based on XBee-PRO S1 modules. DarkO's actuators consist in two propellers

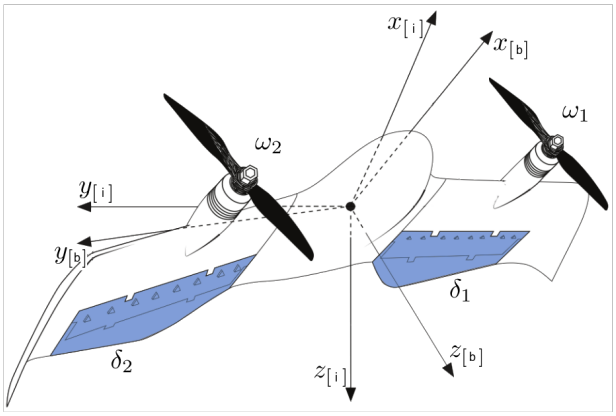

Fig. 2. DarkO body frame with a schematic representation of the actuators (image taken from [34]).

(T-Motor T5147) symmetrically placed at the front of the wing (shown in **black** in Fig. 2) powered by two T-Motor F30 2300kv electric motors and two elevons, placed at the

[1] https://wiki.paparazziuav.org/wiki/Apogee/v1.00

back of the wing (shown in blue in Fig. 2) and acting as control surfaces. The elevons are driven by two MKS DS65K servomotors. Fig. 2 shows the DarkO model, together with a north, east, down (NED) inertial reference frame (or world frame) "i" linked to the Earth's surface, and a body reference frame "b" attached to the drone, with $x_{[\mathrm{b}]}$ corresponding to the longitudinal axis (the propellers axes lay in the $z_{[\mathrm{b}]} = 0$ plane), $y_{[\mathrm{b}]}$ the pitch axis (the direction of the wings), $z_{[\mathrm{b}]}$ the axis to create an orthonormal frame. Using the same notation as in [17], the left and right propeller/elevon are denoted by using subscripts $i = 1$ (left) and $i = 2$ (right). The sign convention will be defined positive for the elevons positions δ_1, δ_2 when they create a pitch-up moment with the propellers rotating in opposite directions with angular speeds $\omega_1 > 0$ and $\omega_2 < 0$, respectively.

Parameter or coefficient	Value	Units
m (drone mass)	0.519	kg
b (wingspan)	0.542	m
c (aerodynamic cord)	0.13	m
$\boldsymbol{B} = \mathrm{diag}(b, c, b)$	$\mathrm{diag}(0.542, 0.13, 0.542)$	m
S (wing area)	0.026936	m^2
S_{wet} (wet area)	0.0180	m^2
S_{p} (propeller area)	0.0127	m^2
$\boldsymbol{J} = \mathrm{diag}(J_x, J_y, J_z)$	$\mathrm{diag}(0.0067, 0.0012, 0.0082)$	$\mathrm{kg\,m}^2$
k_{f} (propeller thrust)	1.7800e-8	kg m
k_{m}(propeller torque)	2.1065e-10	$\mathrm{kg\,m}^2$
p_x (propeller x location)	0.065	m
p_y (propeller y location)	0.162	m
a_y (lift y position)	0.1504	m
ξ_{f} (elevons lift)	0.2	–
ξ_{m} (elevons torque)	1.4	–
ρ (air density)	1.225	$\mathrm{kg\,m}^{-3}$
C_{d} (drag)	0.1644	–
C_y (lateral)	0	–
C_ℓ (lift)	5.4001	–
Δ_{r} (UAV centering)	-0.0145	m

TABLE I
IDENTIFIED NUMERICAL PARAMETERS OF THE DARKO MODEL.

A. Full nonlinear model

Exploiting the modelling method reported in [17] and [26], an accurate model of the DarkO dynamics describes the position $\boldsymbol{p} \in \mathbb{R}^3$ of the origin of the body frame and its velocity $\boldsymbol{v} = \dot{\boldsymbol{p}} \in \mathbb{R}^3$, in addition to its orientation, well represented by a quaternion $\boldsymbol{q} \in \mathbb{S}^3 := \{\boldsymbol{q} \in \mathbb{R}^4 : |\boldsymbol{q}| = 1\}$ and its angular velocity $\boldsymbol{\omega}_{\mathrm{b}}$ represented in the body frame, which satisfy $\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{smallmatrix}\right]$, where $\otimes$ denotes the quaternion product (see [17], [26] or the tutorial [35] for the details). Selecting the overall state as $\boldsymbol{x} := (\boldsymbol{p},\ \boldsymbol{v},\ \boldsymbol{q},\ \boldsymbol{\omega}_{\mathrm{b}})$, the mathematical model derived in [17], depends on a set of parameters listed in Table I, where we also report the value obtained from a system identification procedure reported in the MS Thesis [32]. The dynamical model can be written as

$$\begin{cases} m\dot{\boldsymbol{v}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q})\boldsymbol{F}_{\mathrm{b}}, & \text{(1a)} \\ \boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{b}} = -[\boldsymbol{\omega}_{\mathrm{b}}]_\times \boldsymbol{J}\boldsymbol{\omega}_{\mathrm{b}} + \boldsymbol{M}_{\mathrm{b}}, & \text{(1b)} \end{cases}$$

where $\boldsymbol{g} := \left[\begin{smallmatrix} 0 & 0 & 9.81 \end{smallmatrix}\right]^\top$ denotes the gravity vector, $m \in \mathbb{R}$ is the mass, $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the diagonal moment of inertia (see Table I) and, partitioning the quaternion $\boldsymbol{q} \in \mathbb{S}^3$ as $\boldsymbol{q} := \left[\begin{smallmatrix} \eta & \boldsymbol{\epsilon}^\top \end{smallmatrix}\right]^\top$, the corresponding rotation matrix $\boldsymbol{R}(\boldsymbol{q}) \in SO(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times3} :\ \boldsymbol{R}^\top\boldsymbol{R} = \mathbb{I}_3, \det(\boldsymbol{R}) = 1\}$ is defined as (see [35])

$$\boldsymbol{R}(\boldsymbol{q}) := \mathbb{I}_3 + 2\eta\,[\boldsymbol{\epsilon}]_\times + 2\,[\boldsymbol{\epsilon}]_\times^2\,. \tag{2}$$

According to [17] the force and moment vectors $\boldsymbol{F}_{\mathrm{b}}$ and $\boldsymbol{M}_{\mathrm{b}}$ in (1) depend on (i) the state $\boldsymbol{x}$, (ii) the disturbance $\boldsymbol{w} \in \mathbb{R}^3$, representing the wind speed in the world frame, and (iii) the actuators commands (see Figure 2), comprising the two propellers' rotational speeds $\omega_1, \omega_2 \in \mathbb{R}$ and the two elevons' deflections $\delta_1, \delta_2 \in \mathbb{R}$. Let us first consider the actuators commands' effect. Each propeller generates a thrust $\boldsymbol{T}_i$ oriented in the x direction of the body frame and a moment $\boldsymbol{N}_i$ about the same axis:

$$\boldsymbol{T}_i := \begin{bmatrix} \tau_i \\ 0 \\ 0 \end{bmatrix} := \begin{bmatrix} k_{\mathrm{f}}\omega_i^2 \\ 0 \\ 0 \end{bmatrix},\ \boldsymbol{N}_i := (-1)^i \frac{k_{\mathrm{m}}}{k_{\mathrm{f}}}\boldsymbol{T}_i, \quad i = 1, 2. \tag{3}$$

Each elevon's position $\delta_i \in \mathbb{R}$ is assigned by a servo-motor that imposes an efficiency level (in terms of airstream deflection) quantified by two skew-symmetric matrices:

$$\boldsymbol{\Delta}_i^{\mathrm{f}} := \begin{bmatrix} 0 & 0 & \xi_{\mathrm{f}}\delta_i \\ 0 & 0 & 0 \\ -\xi_{\mathrm{f}}\delta_i & 0 & 0 \end{bmatrix},\ \boldsymbol{\Delta}_i^{\mathrm{m}} := \begin{bmatrix} 0 & 0 & \xi_{\mathrm{m}}\delta_i \\ 0 & 0 & 0 \\ -\xi_{\mathrm{m}}\delta_i & 0 & 0 \end{bmatrix}, \tag{4}$$

$i = 1, 2$. The constant parameters k_{f}, k_{m}, ξ_{f}, ξ_{m} appearing in (3) and (4) are listed in Table I.

With the above actuation quantities, we may rearrange the dynamics given in [17, eqns (97), (98)] (see also [32]) and express $\boldsymbol{F}_{\mathrm{b}}$ and $\boldsymbol{M}_{\mathrm{b}}$ in (1) as

$$\begin{aligned} \boldsymbol{F}_{\mathrm{b}} :=\ & \boldsymbol{T}_1 + \boldsymbol{T}_2 + \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\boldsymbol{\Phi}^{(\mathrm{fv})}\Big((\boldsymbol{\Delta}_1^{\mathrm{f}} - \mathbb{I}_3)\boldsymbol{T}_1 + (\boldsymbol{\Delta}_2^{\mathrm{f}} - \mathbb{I}_3)\boldsymbol{T}_2\Big) \\ & + \frac{1}{4}\rho S\boldsymbol{\Phi}^{(\mathrm{fv})}\Big(\boldsymbol{\Delta}_1^{\mathrm{f}} + \boldsymbol{\Delta}_2^{\mathrm{f}} - 2\mathbb{I}_3\Big)\|\boldsymbol{v}_{\mathrm{b}}\|\boldsymbol{v}_{\mathrm{b}} \\ & + \frac{1}{4}\rho S\boldsymbol{\Phi}^{(\mathrm{mv})}\Big(\boldsymbol{\Delta}_1^{\mathrm{f}} + \boldsymbol{\Delta}_2^{\mathrm{f}} - 2\mathbb{I}_3\Big)\boldsymbol{B}\|\boldsymbol{v}_{\mathrm{b}}\|\boldsymbol{\omega}_{\mathrm{b}}, \end{aligned} \tag{5}$$

$$\begin{aligned} \boldsymbol{M}_{\mathrm{b}} :=\ & \boldsymbol{N}_1 + \boldsymbol{N}_2 + \left[\begin{smallmatrix} p_x \\ p_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{T}_1 + \left[\begin{smallmatrix} p_x \\ -p_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{T}_2 \\ & - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\left(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}(\boldsymbol{\Delta}_1^{\mathrm{m}} - \mathbb{I}_3) + \left[\begin{smallmatrix} 0 \\ a_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{\Phi}^{(\mathrm{fv})}(\boldsymbol{\Delta}_1^{\mathrm{m}} + \mathbb{I}_3)\right)\boldsymbol{T}_1 \\ & - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\left(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}(\boldsymbol{\Delta}_2^{\mathrm{m}} - \mathbb{I}_3) + \left[\begin{smallmatrix} 0 \\ -a_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{\Phi}^{(\mathrm{fv})}(\boldsymbol{\Delta}_2^{\mathrm{m}} + \mathbb{I}_3)\right)\boldsymbol{T}_2 \\ & + \frac{1}{4}\rho S\Bigg(\Big(\left[\begin{smallmatrix} 0 \\ a_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{\Phi}^{(\mathrm{fv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Big)\boldsymbol{\Delta}_1^{\mathrm{m}} \\ & + \Big(\left[\begin{smallmatrix} 0 \\ -a_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{\Phi}^{(\mathrm{fv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Big)\boldsymbol{\Delta}_2^{\mathrm{m}} - 2\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Bigg)\|\boldsymbol{v}_{\mathrm{b}}\|\boldsymbol{v}_{\mathrm{b}} \\ & + \frac{1}{4}\rho S\Bigg(\Big(\left[\begin{smallmatrix} 0 \\ a_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{\Phi}^{(\mathrm{mv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{m}\omega)}\Big)\boldsymbol{\Delta}_1^{\mathrm{m}} \\ & + \Big(\left[\begin{smallmatrix} 0 \\ -a_y \\ 0 \end{smallmatrix}\right]_\times \boldsymbol{\Phi}^{(\mathrm{mv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{m}\omega)}\Big)\boldsymbol{\Delta}_2^{\mathrm{m}} - 2\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{m}\omega)}\Bigg)\boldsymbol{B}\|\boldsymbol{v}_{\mathrm{b}}\|\boldsymbol{\omega}_{\mathrm{b}}, \end{aligned} \tag{6}$$

where $\boldsymbol{v}_{\mathrm{b}} := \boldsymbol{R}^\top(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{w})$ represents the air speed seen by the drone, represented in the body frame. In [17], the scalars $\|\boldsymbol{v}_{\mathrm{b}}\|$ appearing in the expressions of $\boldsymbol{F}_{\mathrm{b}}$ and $\boldsymbol{M}_{\mathrm{b}}$ are replaced by the scalar $\eta = \sqrt{\|\boldsymbol{v}_{\mathrm{b}}\|^2 + \mu c^2\|\boldsymbol{\omega}_{\mathrm{b}}\|^2}$, with $\mu \in \mathbb{R}$ being a

parameter related to the model identification, but in the case of DarkO [32], the identification provides $\mu = 0$, therefore we present a simplified description here. The constant aerodynamic coefficients' matrix $\boldsymbol{\Phi} := \begin{bmatrix} \boldsymbol{\Phi}^{(\mathrm{fv})} & \boldsymbol{\Phi}^{(\mathrm{mv})\top} \\ \boldsymbol{\Phi}^{(\mathrm{mv})} & \boldsymbol{\Phi}^{(\mathrm{m}\omega)} \end{bmatrix} \in \mathbb{R}^{6\times 6}$, is defined in [26, eqs. (6)–(9)] as $\boldsymbol{\Phi}^{(\mathrm{fv})} := \mathrm{diag}(C_\mathrm{d}, C_y, C_\ell)$ and

$$\left[\begin{array}{c|c} \boldsymbol{\Phi}^{(\mathrm{mv})} & \boldsymbol{\Phi}^{(\mathrm{m}\omega)} \end{array}\right] := \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0.1396 & 0 & 0.0573 \\ 0 & 0 & -\frac{\Delta_\mathrm{r}}{c} C_\ell & 0 & 0.6358 & 0 \\ 0 & 0 & 0 & 0.0405 & 0 & 0.0019 \end{array}\right],$$

the numerical values of the constants being reported in Table I (these numerical values were not reported in [17] and [26] and are given here to allow reproducing our simulation results). The numerical values in Table I have been obtained by a model identification campaign [32]. In particular, coefficient k_f was identified from equation (3), which links the motor rotation speed ω_i with the generated traction, the minimum and maximum rotation speed and the time constant of the motor actuation chain (see the experiments reported in Fig. 3). The diagonal elements of the inertia $\boldsymbol{J}$ were measured using a bifilar pendulum system. This method is widely used in the drone field [36], and is based on the period of oscillation about each one of the three axes $(x_{[\mathrm{b}]}, y_{[\mathrm{b}]}, z_{[\mathrm{b}]})$ of the drone suspended by two wires, which forms a torsion pendulum as shown in Fig. 4. It is interesting to note that the surface area blown by the propellers represents 67 percent of the drone's total surface area.

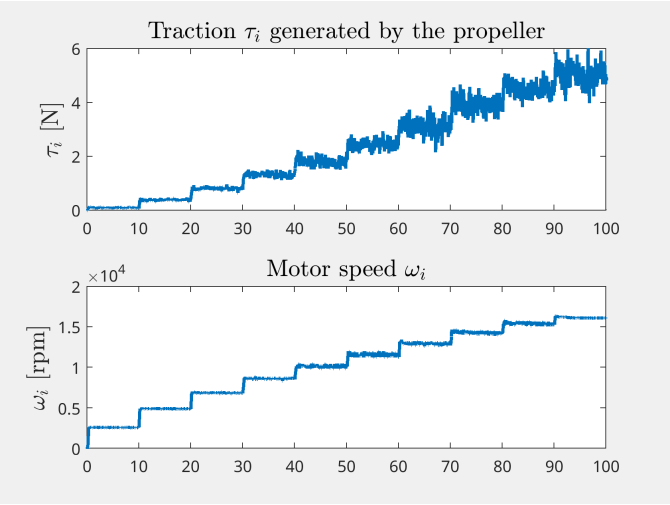

Fig. 3. Input-output response of the Esc-Motor-Propeller assembly.

B. Simplified nonlinear low-speed model

Since we focus in this paper on the hovering phase, where the speed is small, we may simplify the DarkO full nonlinear model (1) by neglecting the aerodynamic effects, namely all the terms that are quadratic in the speed $\boldsymbol{v}_\mathrm{b}$ and $\boldsymbol{\omega}_\mathrm{b}$ in (5) and (6). Neglecting those terms and defining the input vector

$$\boldsymbol{u} := \begin{bmatrix} \tau_1 & \tau_2 & \delta_1 & \delta_2 \end{bmatrix}^\top, \tag{7}$$

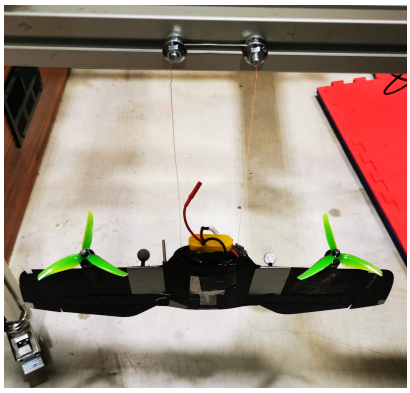

Fig. 4. Bifilar pendulum mounting for the identication of $\boldsymbol{J}$.

allows us to write an approximate low-speed model retaining the most important nonlinear wind effects

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{8a}$$

$$m\dot{\boldsymbol{v}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q})\Big(\boldsymbol{M}_\mathrm{f}(\boldsymbol{u}) + \boldsymbol{D}_\mathrm{f}(\boldsymbol{u})\|\boldsymbol{w}\|\boldsymbol{R}^\top(\boldsymbol{q})(\boldsymbol{v}-\boldsymbol{w})\Big), \tag{8b}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_\mathrm{b} \end{bmatrix}, \tag{8c}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_\mathrm{b} = -[\boldsymbol{\omega}_\mathrm{b}]_\times \boldsymbol{J}\boldsymbol{\omega}_\mathrm{b} + \boldsymbol{M}_\mathrm{m}(\boldsymbol{u}) + \boldsymbol{D}_\mathrm{m}(\boldsymbol{u})\|\boldsymbol{w}\|\boldsymbol{R}^\top(\boldsymbol{q})(\boldsymbol{v}-\boldsymbol{w}), \tag{8d}$$

where vectors $\boldsymbol{M}_\mathrm{f}(\boldsymbol{u})$ and $\boldsymbol{M}_\mathrm{m}(\boldsymbol{u})$, and matrices $\boldsymbol{D}_\mathrm{f}(\boldsymbol{u})$ and $\boldsymbol{D}_\mathrm{m}(\boldsymbol{u})$ stem from setting to zero the angular velocity-dependent terms in (5) and (6). After some manipulations, they can be written as

$$\begin{aligned} \boldsymbol{M}_\mathrm{f}(\boldsymbol{u}) &:= \boldsymbol{T}_1 + \boldsymbol{T}_2 + \frac{S_\mathrm{wet}}{4S_\mathrm{p}}\boldsymbol{\Phi}^{(\mathrm{fv})}\Big((\boldsymbol{\Delta}_1^\mathrm{f} - \mathbb{I}_3)\boldsymbol{T}_1 + (\boldsymbol{\Delta}_2^\mathrm{f} - \mathbb{I}_3)\boldsymbol{T}_2\Big) \\ &= \begin{bmatrix} \left(1 - \frac{S_\mathrm{wet}}{4S_\mathrm{p}} C_\mathrm{d}\right)(\tau_1 + \tau_2) \\ 0 \\ -\frac{S_\mathrm{wet}}{4S_\mathrm{p}} C_\ell \xi_\mathrm{f}(\delta_1\tau_1 + \delta_2\tau_2) \end{bmatrix} \end{aligned} \tag{9}$$

$$\begin{aligned} \boldsymbol{M}_\mathrm{m}(\boldsymbol{u}) &:= \boldsymbol{N}_1 + \boldsymbol{N}_2 + \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}_\times \boldsymbol{T}_1 + \begin{bmatrix} p_x \\ -p_y \\ 0 \end{bmatrix}_\times \boldsymbol{T}_2 \\ &\quad - \frac{S_\mathrm{wet}}{4S_\mathrm{p}}\left(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}(\boldsymbol{\Delta}_1^\mathrm{m} - \mathbb{I}_3) + \begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})}(\mathbb{I}_3 + \boldsymbol{\Delta}_1^\mathrm{m})\right)\boldsymbol{T}_1 \\ &\quad - \frac{S_\mathrm{wet}}{4S_\mathrm{p}}\left(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}(\boldsymbol{\Delta}_2^\mathrm{m} - \mathbb{I}_3) + \begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})}(\mathbb{I}_3 + \boldsymbol{\Delta}_2^\mathrm{m})\right)\boldsymbol{T}_2 \\ &= \begin{bmatrix} \frac{k_\mathrm{m}}{k_\mathrm{f}}(\tau_1 - \tau_2) + \frac{S_\mathrm{wet}}{4S_\mathrm{p}} a_y C_\ell \xi_\mathrm{f}(\delta_1\tau_1 - \delta_2\tau_2) \\ \frac{S_\mathrm{wet}}{4S_\mathrm{p}} \Delta_\mathrm{r} C_\ell \xi_\mathrm{m}(\delta_1\tau_1 + \delta_2\tau_2) \\ \left(p_y + \frac{S_\mathrm{wet}}{4S_\mathrm{p}} a_y C_\mathrm{d}\right)(\tau_1 - \tau_2) \end{bmatrix} \end{aligned} \tag{10}$$

$$\begin{aligned} \boldsymbol{D}_\mathrm{f}(\boldsymbol{u}) &:= \frac{1}{4}\rho S \boldsymbol{\Phi}^{(\mathrm{fv})}\Big(\boldsymbol{\Delta}_1^\mathrm{f} + \boldsymbol{\Delta}_2^\mathrm{f} - 2\mathbb{I}_3\Big) \\ &= \frac{1}{4}\rho S \begin{bmatrix} -2C_\mathrm{d} & 0 & C_\mathrm{d}\xi_\mathrm{f}(\delta_1 + \delta_2) \\ 0 & 0 & 0 \\ -C_\ell \xi_\mathrm{f}(\delta_1 + \delta_2) & 0 & -2C_\ell \end{bmatrix} \end{aligned} \tag{11}$$

$$\begin{aligned} \boldsymbol{D}_\mathrm{m}(\boldsymbol{u}) &:= \frac{1}{4}\rho S \Bigg(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\right)\boldsymbol{\Delta}_1^\mathrm{m} \\ &\quad + \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\right)\boldsymbol{\Delta}_2^\mathrm{m} - 2\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Bigg) \end{aligned} \tag{12}$$

$$= \frac{1}{4}\rho S \begin{bmatrix} -a_y C_{\mathrm{d}} \xi_{\mathrm{m}}(\delta_1 - \delta_2) & 0 & 0 \\ \Delta_{\mathrm{r}} C_\ell \xi_{\mathrm{m}}(\delta_1 + \delta_2) & 0 & 2\Delta_{\mathrm{r}} C_\ell \\ 0 & 0 & -a_y C_\ell \xi_{\mathrm{m}}(\delta_1 - \delta_2) \end{bmatrix}$$

where we may appreciate a fundamentally nonlinear effect of a nonzero-wind, which is nonlinear with $\boldsymbol{q}$, $\|\boldsymbol{v}_{\mathrm{b}}\|$ and $\boldsymbol{w}$. As in [26, eqn. (10)] and according to Diederich's formula, we obtain $C_\ell = C_{\mathrm{d}} + \frac{\pi AR}{1+\sqrt{1+\left(\frac{AR}{2}\right)^2}}$ were $AR = \frac{b^2}{S}$ is the wing aspect ratio.

Remark 1: Note that a coupling term $(\delta_1\tau_1 + \delta_2\tau_2)$ of the actuators effects can be observed in the expressions of the matrices $\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u})$ and $\boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u})$. In Remark 4, we will use a nonlinear transformation of the control input vector motivated by this coupling. ∘

For models (1) and (8), we characterize below certain relevant hovering equilibria induced by input $\boldsymbol{u}$ under the action of a constant wind, $\boldsymbol{w}_{\mathrm{eq}} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \in \mathbb{R}^3$ (expressed in the world frame) such that $\begin{bmatrix} w_x \\ w_y \end{bmatrix} \neq 0$, so that there is some nonzero horizontal component of the wind. To begin with, we define the quaternion $\boldsymbol{q}_{\mathrm{eq}\psi}$ associated with a horizontal rotation $\psi = \arctan(w_x, w_y)$ of the inertial reference frame towards the (nonzero) horizontal wind direction:

$$\boldsymbol{q}_{\mathrm{eq}\psi} := \begin{bmatrix} \cos(\frac{\psi}{2}) & 0 & 0 & \sin(\frac{\psi}{2}) \end{bmatrix}^\top. \tag{13}$$

Then, for each reference position $\boldsymbol{p}_{\mathrm{eq}} \in \mathbb{R}^3$, a set of possible input-state equilibrium pairs $(\boldsymbol{u}_{\mathrm{eq}}, \boldsymbol{x}_{\mathrm{eq}}) = (\boldsymbol{u}_{\mathrm{eq}}, \boldsymbol{p}_{\mathrm{eq}}, \boldsymbol{v}_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{eq}}, \boldsymbol{\omega}_{\mathrm{b,eq}})$ is given by

$$\boldsymbol{u}_{\mathrm{eq}} = \begin{bmatrix} \tau & \tau & \delta & \delta \end{bmatrix}^\top \tag{14a}$$

$$\boldsymbol{q}_{\mathrm{eq}} = \boldsymbol{q}_{\mathrm{eq}\psi} \otimes \boldsymbol{q}_{\mathrm{eq}\theta} \tag{14b}$$

$$\boldsymbol{\omega}_{\mathrm{b,eq}} = 0, \quad \boldsymbol{v}_{\mathrm{eq}} = 0, \tag{14c}$$

where

$$\boldsymbol{q}_{\mathrm{eq}\theta} := \begin{bmatrix} \cos(\frac{\theta}{2}) & 0 & \sin(\frac{\theta}{2}) & 0 \end{bmatrix}^\top \tag{15}$$

and the pitch angle θ, the propeller thrust τ, and the elevons deflection δ can be constructed as specified in Algorithm 1. The fact that (13)–(15) with Algorithm 1 provide a suitable equilibrium is stated and proven next, in our first main result.

Theorem 1: For any constant wind, $\boldsymbol{w} = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^\top \in \mathbb{R}^3$ having a nonzero horizontal component $\begin{bmatrix} w_x \\ w_y \end{bmatrix}$, equations (13)–(15) with θ, τ and δ selected according to Algorithm 1 characterize an equilibrium pair $(\boldsymbol{u}_{\mathrm{eq}}, \boldsymbol{x}_{\mathrm{eq}})$ for the nonlinear dynamics (1) and (8).

Proof: As a first step, let us note that, with the expression of $\boldsymbol{R}$ (2) and the expression of ψ in step 1 of Algorithm 1, we may define the rotated equilibrium disturbance $\boldsymbol{w}_{\mathrm{r,eq}} := \boldsymbol{R}_\psi^\top \boldsymbol{w}_{\mathrm{eq}} := \boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{eq}\psi})\boldsymbol{w}_{\mathrm{eq}}$ (see (21) in Algorithm 1), which corresponds to the rotation required to align the $x_{[\mathrm{b}]}$ axis of the body frame with the direction of the wind. Once the drone is facing the wind, it sees a zero y component and can adjust its pitch angle θ so as to generate the necessary thrust and lift to compensate the wind effects in the longitudinal and vertical direction (the lateral effect is zero due to the specific ψ orientation).

With this ψ rotation, it is possible to express the wind $\boldsymbol{w}$ in the body frame as

$$\boldsymbol{w}_{\mathrm{eq}}^{\mathrm{b}} := \begin{bmatrix} w_x^{\mathrm{b}} \\ 0 \\ w_z^{\mathrm{b}} \end{bmatrix} = \boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{eq}\theta})\boldsymbol{w}_{\mathrm{r,eq}} \tag{16}$$

$$= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}^\top \begin{bmatrix} w_{\mathrm{r}x} \\ 0 \\ w_{\mathrm{r}z} \end{bmatrix} = \begin{bmatrix} w_{\mathrm{r}x}\cos\theta - w_{\mathrm{r}z}\sin\theta \\ 0 \\ w_{\mathrm{r}x}\sin\theta + w_{\mathrm{r}z}\cos\theta \end{bmatrix}$$

We emphasize that w_x^{b} is always negative and never zero, because the drone is oriented in the direction of the wind thanks to the rotation induced by $\boldsymbol{q}_{\mathrm{eq}\psi}$, and because of the assumption $\begin{bmatrix} w_x \\ w_y \end{bmatrix} \neq 0$.

Continuing, equation (8a) shows that $\boldsymbol{v}_{\mathrm{eq}} = 0$ is necessary for an equilibrium. By pre-multiplying (8b) by $\boldsymbol{R}(\boldsymbol{q}_{\mathrm{eq}})$ given in (16), we express it in the body frame. Since, we apply the same input $\tau_1 = \tau_2 = \tau$ to the two propellers and the same input to the elevons $\delta_1 = \delta_2 = \delta$, for both models (1) and (8), the force balance in the $x_{[\mathrm{b}]}$-axis direction gives

$$(2 - \frac{S_{\mathrm{wet}}}{2S_{\mathrm{p}}}C_{\mathrm{d}})\tau - \frac{1}{2}\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\mathrm{d}}\left(w_x^{\mathrm{b}} - \xi_{\mathrm{f}}\delta w_z^{\mathrm{b}}\right) - mg\sin(\theta) = 0 \tag{17}$$

and the force balance in the $z_{[\mathrm{b}]}$-axis direction gives

$$-\frac{S_{\mathrm{wet}}}{2S_{\mathrm{p}}}\xi_{\mathrm{f}}C_\ell\tau\delta - \frac{1}{2}\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell\left(w_z^{\mathrm{b}} + \xi_{\mathrm{f}}\delta w_x^{\mathrm{b}}\right) + mg\cos(\theta) = 0 \tag{18}$$

Similarly, from (1b) and (8d), the moment balance about the $y_{[\mathrm{b}]}$-axis provides

$$\frac{S_{\mathrm{wet}}}{2S_{\mathrm{p}}}\Delta_{\mathrm{r}}\xi_{\mathrm{m}}C_\ell\tau\delta + \frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell\left(w_z^{\mathrm{b}} + \xi_{\mathrm{m}}\delta w_x^{\mathrm{b}}\right) = 0. \tag{19}$$

To compute a solution (θ,τ,δ) of the three balance equations (17)–(19), let us add (18) multiplied by $\Delta_{\mathrm{r}}\xi_{\mathrm{m}}$, to (19) multiplied by ξ_{f}, so as to cancel out the first term, and obtain

$$\Delta_{\mathrm{r}}\xi_{\mathrm{m}}\left(-\frac{1}{2}\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(w_z^{\mathrm{b}} + \xi_{\mathrm{f}}\delta w_x^{\mathrm{b}}) + mg\cos(\theta)\right) + \xi_{\mathrm{f}}\left(\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(w_z^{\mathrm{b}} + \xi_{\mathrm{m}}\delta w_x^{\mathrm{b}})\right) = 0,$$

which is equivalent to

$$\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_z^{\mathrm{b}} + \Delta_{\mathrm{r}}\xi_{\mathrm{m}}mg\cos(\theta) = 0,$$

where (w_x^{b},w_z^{b}) are the first and third components of $\boldsymbol{w}^{\mathrm{b}}$ in (16). Then, using (16) and rearanging, we get

$$-\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}x}\sin\theta + \left(-\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}z} + \Delta_{\mathrm{r}}\xi_{\mathrm{m}}mg\right)\cos\theta = 0,$$

which is satisfied by

$$\theta = -\tan^{-1}\left(\frac{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}z} - 2\xi_{\mathrm{m}}mg}{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}x}}\right). \tag{20}$$

This last expression coincides with selection (22) in Algorithm 1 after some manipulations.

Based on (22), we may now compute the equilibrium input by substituting (17) into (18). After some simplifications, the necessary propeller traction τ to maintain the equilibrium position corresponds to expression (23). Finally, with the value of τ in (23), we may obtain the required elevon deflection δ from equation (18), which provides the value (24). ■

Algorithm 1 Design of the equilibrium parameters in (14).

Input: Wind vector $\boldsymbol{w}_{\text{eq}} = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^\top$

Output: Parameters ψ, θ, τ, δ in (14)

1: Determine the angle $\psi = \text{atan2}(w_x, w_y)$ so as to determine $\boldsymbol{q}_{\text{eq}\psi}$ from (13)

2: Determine the rotated disturbance $\boldsymbol{w}_{\text{r}}$ with zero y component, using $\boldsymbol{R}_\psi := \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$, as follows

$$\boldsymbol{w}_{\text{r,eq}} := \begin{bmatrix} w_{\text{r}x} \\ 0 \\ w_{\text{r}z} \end{bmatrix} := \boldsymbol{R}^\top(\boldsymbol{q}_{\text{eq}\psi})\boldsymbol{w}_{\text{eq}} = \boldsymbol{R}_\psi^\top \boldsymbol{w}_{\text{eq}} \tag{21}$$

3: Select the pitch angle θ so as to determine $\boldsymbol{q}_{\text{eq}\theta}$ from (14b):

$$\theta = -\tan^{-1}\left(\frac{w_{\text{r}z}}{w_{\text{r}x}} + \frac{2mg}{\rho S\|\boldsymbol{w}_{\text{eq}}\|C_\ell(1-\frac{\xi_{\text{f}}}{\xi_{\text{m}}})w_{\text{r}x}}\right) \tag{22}$$

4: For notational convenience, define the scalars

$$\left[\begin{array}{c|c} a & b \\ \hline c & d \end{array}\right] := \left[\begin{array}{c|c} \dfrac{2S_{\text{wet}}C_\ell mg\sin\theta\xi_{\text{f}}}{-4SS_{\text{p}}C_\ell\rho\|\boldsymbol{w}_{\text{eq}}\|w_x^{\text{b}}\xi_{\text{f}}} & \dfrac{2S_{\text{wet}}C_{\text{d}}C_\ell\rho\|\boldsymbol{w}_{\text{eq}}\|w_x^{\text{b}}}{\frac{b\xi_{\text{f}}}{2}} \end{array}\right]$$

and based on scalars (a, b, c, d), select the propeller traction τ in (14a) as

$$\tau = \frac{S_{\text{p}}}{2S_{\text{wet}}C_\ell\xi_{\text{f}}(4S_{\text{p}} - S_{\text{wet}}C_{\text{d}})}\left(a+b+c+d + \left[(a+b+c-d)^2 - 4(d^2+ac-bd)\right.\right. \tag{23}$$
$$\left.\left. - \frac{4{w_z^{\text{b}}}^2 d}{{w_x^{\text{b}}}^2}(d+c) + \frac{4w_z^{\text{b}}ad\cos\theta}{w_x^{\text{b}}C_\ell\sin\theta}\left(C_{\text{d}} - \frac{4S_{\text{p}}}{S_{\text{wet}}}\right)\right]^{\frac{1}{2}}\right),$$

where

$$\begin{bmatrix} w_x^{\text{b}} \\ w_z^{\text{b}} \end{bmatrix} = \begin{bmatrix} w_{\text{r}x}\cos\theta - w_{\text{r}z}\sin\theta \\ w_{\text{r}x}\sin\theta + w_{\text{r}z}\cos\theta \end{bmatrix}.$$

5: Select the elevon deflection δ as follows

$$\delta = \frac{2mg\sin\theta}{\rho S\|\boldsymbol{w}_{\text{eq}}\|C_{\text{d}}\xi_{\text{f}}w_z^{\text{b}}} + \frac{w_x^{\text{b}}}{\xi_{\text{f}}w_z^{\text{b}}} - \frac{(4-\frac{S_{\text{wet}}}{S_{\text{p}}}C_{\text{d}})}{\rho S\|\boldsymbol{w}_{\text{eq}}\|C_{\text{d}}\xi_{\text{f}}w_z^{\text{b}}}\tau. \tag{24}$$

Return: ψ, θ, τ, δ

With the analytical expression (14) of the drone's equilibrium for different wind conditions $\boldsymbol{w}$, we report in Fig. 5 the corresponding values of θ, δ, τ for horizontal wind speed values ranging from 0 to $-20\,\text{m}\,\text{s}^{-1}$ and for vertical wind speed values ranging from -6 to $6\,\text{m}\,\text{s}^{-1}$. The angle of incidence θ decreases from 90° to $-4.65°$. The traction τ attains its minimum at $w_{rx} = -12.8\,\text{m}\,\text{s}^{-1}$, which corresponds to a flight condition that minimizes the power consumption.

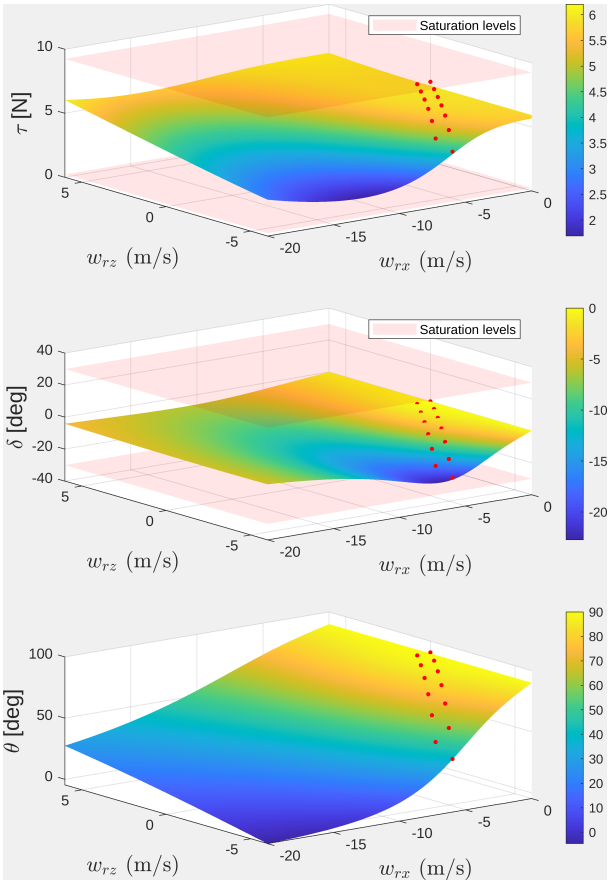

Fig. 5. Parameters ($\boldsymbol{\theta}$, $\boldsymbol{\delta}$, $\boldsymbol{\tau}$) of the equilibrium point (surface) established in Theorem 1 and Algorithm 1 for constant horizontal and vertical wind ($\boldsymbol{w_{rx}}$,$\boldsymbol{w_{rz}}$), and actuators saturation levels (pink).

Remark 2: (About the values of θ and τ in Algorithm 1) It is interesting to note that for each pair of wind components ($w_{\text{r}z}$,$w_{\text{r}x}$) the corresponding equilibrium orientation (14b), (22) is independent of the input $\boldsymbol{u}_{\text{eq}}$. Moreover, it should be emphasized that throughout the reasonable wind values of Fig. 5, equation (23) corresponds to the positive root of a second order polynomial with the other root being always negative, and leading to an unreasonable negative thrust condition. ○

Remark 3: (Plant input saturation) The DarkO actuators have dynamics that limit their actions both in terms of amplitude and rate. For electric motors generating traction by propellers, there are two causes of saturation. A high-speed saturation related to the motor's maximum voltage, and a low-speed saturation related to the motor's minimum coil switching speed to maintain rotation. In addition, these saturations enable us to obtain a realistic finite-energy model. It corresponds to the following constraint $\omega_i \in [2500,\ 16000]\ rpm = [262,\ 1675]\ \text{rad}\,\text{s}^{-1}$, $i = 1, 2$. In terms of dynamics, we have represented the motor actuation chain (consisting of the ESC, the motor and the propeller) by a first order filter having a time constant equal to $0.0125\,\text{s}$, which provides a fairly aggressive actuation system.

The saturation levels characterizing the elevons stem from the servo motors type of actuation. The elevons input is limited in displacement by the shape of the UAV and the

physical limits of the servo motors, which translate into a saturation of the control surface deflection $\delta_i \in [-30\ ;30]^\circ$, $i = 1, 2$. Perhaps the most relevant saturation here is the actuator bandwidth (due to the servo-motor actuation), which is modelled by a first order filter with time constant $0.05\,\mathrm{s}$. These saturation effects are taken into account in the control design and are explicitly considered in our simulation results. ◦

Remark 4: (Special case with zero-wind) We already characterized in our preliminary work [34] the equilibria with zero-wind $\boldsymbol{w}_{\mathrm{eq}} = 0$. In particular, with the input transformation $\boldsymbol{u}_{\mathrm{nowind}} := \begin{bmatrix}\tau_1 & \tau_2 & \delta_1\tau_1 & \delta_2\tau_2\end{bmatrix}^\top$, the following model is derived from (8) by setting $\boldsymbol{w} = 0$,

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \qquad m\dot{\boldsymbol{v}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q})\boldsymbol{F}\boldsymbol{u}_{\mathrm{nowind}}, \quad (25a)$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix}0\\ \boldsymbol{\omega}_{\mathrm{b}}\end{bmatrix} \qquad \boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{b}} = -[\boldsymbol{\omega}_{\mathrm{b}}]_\times J\boldsymbol{\omega}_{\mathrm{b}} + \boldsymbol{M}\boldsymbol{u}_{\mathrm{nowind}}, \quad (25b)$$

with matrices

$$\left[\begin{array}{c|c}\boldsymbol{F} & \boldsymbol{M}\end{array}\right] := \left[\begin{array}{cccc|cccc} a_{\mathrm{f}} & a_{\mathrm{f}} & 0 & 0 & a_{\mathrm{m}} & -a_{\mathrm{m}} & b_{\mathrm{m}} & -b_{\mathrm{m}} \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{\mathrm{m}} & c_{\mathrm{m}} \\ 0 & 0 & b_{\mathrm{f}} & b_{\mathrm{f}} & d_{\mathrm{m}} & -d_{\mathrm{m}} & 0 & 0 \end{array}\right]$$

and scalars

$$\left[\begin{array}{c|c} a_{\mathrm{f}} & b_{\mathrm{f}} \\ \hline a_{\mathrm{m}} & b_{\mathrm{m}} \\ \hline c_{\mathrm{m}} & d_{\mathrm{m}} \end{array}\right] = \left[\begin{array}{c|c} 1 - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}C_{\mathrm{d}} & -\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}C_\ell\xi_{\mathrm{f}} \\ \hline \frac{k_{\mathrm{m}}}{k_{\mathrm{f}}} & \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}a_y C_\ell \xi_{\mathrm{f}} \\ \hline \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\Delta_{\mathrm{r}} C_\ell \xi_{\mathrm{m}} & p_y + \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}a_y C_{\mathrm{d}} \end{array}\right].$$

This special case of (8) coincides with the model in [34, eqn. (1)]. All the possible equilibrium pairs $(\boldsymbol{u}_{\mathrm{nowind}}, \boldsymbol{x}) = (\boldsymbol{u}_{\mathrm{nowind,eq}}, \boldsymbol{x}_{\mathrm{eq}})$ are parametrized by an arbitrary angle along the $z_{[\mathrm{i}]}$ axis, $\beta \in \left[-\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}\right]$ as follows

$$\boldsymbol{u}_{\mathrm{nowind,eq}} = \frac{mg}{(1 - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}C_{\mathrm{d}})}[1\ 1\ 0\ 0]^\top \quad (26a)$$

$$\boldsymbol{q}_{\mathrm{eq}} = [\eta_{\mathrm{eq}}\ \boldsymbol{\epsilon}_{\mathrm{eq}}^\top]^\top = \left[\sqrt{\tfrac{1}{2}-\beta}\ \ \beta\ \ \tfrac{2\beta^2-1}{2\sqrt{\frac{1}{2}-\beta}}\ \ \beta\right]^\top. \quad (26b)$$

As compared to the wind-induced equilibrium pairs characterized in Theorem 1, we see that, in the presence of nonzero-wind, the additionnal degree of freedom β (characterizing a rotation about the vertical axis of the inertial frame) only makes sense when the UAV does not need a nonzero θ in (14). In fact, a nonzero θ is required for compensating a nonzero wind effect, and in this case the drone must face the wind, thereby requiring a specific heading, associated with angle ψ in (13). ◦

C. Linearized dynamics with constant wind

For each one of the equilibria characterized in Theorem 1, we derive here some linearized equations of motion with respect to the simplified nonlinear low-speed model (8). A direct approach would lead to linearized equations that depend on the ψ angle characterized in step 1 of Algorithm 1. Instead, we define here the incremental coordinates in a suitably rotated inertial reference frame, so that the linearized dynamics is independent of the ψ angle . More specifically, for each equilibrium wind condition $\boldsymbol{w}_{\mathrm{eq}}$ and the ensuing equilibrium $(\boldsymbol{u}_{\mathrm{eq}}, \boldsymbol{p}_{\mathrm{eq}}, \boldsymbol{v}_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{eq}}, \boldsymbol{\omega}_{\mathrm{b,eq}})$ characterized in (13)–(15), denoting the scalar and vector components of the quaternion in (14b) as $\boldsymbol{q}_{\mathrm{eq}} = (\eta_{\mathrm{eq}}, \boldsymbol{\epsilon}_{\mathrm{eq}})$, and based on the rotation matrix $\boldsymbol{R}_\psi := \boldsymbol{R}(\boldsymbol{q}_{\mathrm{eq}\psi})$ introduced at the beginning of the proof of Theorem 1, we study here the approximate linear dynamics of the rotated incremental input-state vector:

$$\tilde{\boldsymbol{x}} := (\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{\epsilon}}, \tilde{\boldsymbol{\omega}}_{\mathrm{b}}) = \left(\boldsymbol{R}_\psi^\top(\boldsymbol{p} - \boldsymbol{p}_{\mathrm{eq}}), \boldsymbol{R}_\psi^\top\boldsymbol{v}, \boldsymbol{R}_\psi^\top(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\mathrm{eq}}), \boldsymbol{\omega}_{\mathrm{b}}\right),$$
$$\tilde{\boldsymbol{u}} := \boldsymbol{u} - \boldsymbol{u}_{\mathrm{eq}}, \quad \tilde{\boldsymbol{w}} := \boldsymbol{R}_\psi^\top(\boldsymbol{w} - \boldsymbol{w}_{\mathrm{eq}}). \quad (27)$$

Note that the rotation in (27) enjoys the useful property that $\boldsymbol{R}_\psi^\top\boldsymbol{\epsilon}_{\mathrm{eq}} = \begin{bmatrix}0 & \sin(\frac{\theta}{2}) & 0\end{bmatrix}^\top$, a fact that greatly simplifies the linearized motion.

Exploiting the fact that the translational and rotational speeds $(\boldsymbol{v}_{\mathrm{eq}}, \boldsymbol{\omega}_{\mathrm{b,eq}})$ must be zero at the equilibrium (see (14)), we prove below that the approximate linearized dynamics for the state in (27) is given by

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}_w\tilde{\boldsymbol{x}} + \boldsymbol{G}_w\tilde{\boldsymbol{u}} + \boldsymbol{E}_w\tilde{\boldsymbol{w}} \quad (28)$$
$$= \begin{bmatrix}\mathbb{0}_3 & \mathbb{I}_3 & \mathbb{0}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \boldsymbol{A}_{vv} & \boldsymbol{A}_{v\epsilon} & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{\epsilon\omega} \\ \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{\omega\epsilon} & \mathbb{0}_3 \end{bmatrix}\tilde{\boldsymbol{x}} + \begin{bmatrix}\mathbb{0}_{3\times4} \\ \boldsymbol{G}_v \\ \mathbb{0}_{3\times4} \\ \boldsymbol{G}_\omega\end{bmatrix}\tilde{\boldsymbol{u}} + \begin{bmatrix}\mathbb{0}_{3\times3} \\ \boldsymbol{E}_v \\ \mathbb{0}_{3\times3} \\ \boldsymbol{E}_\omega\end{bmatrix}\tilde{\boldsymbol{w}},$$

with matrices $\boldsymbol{A}_{vv}$, $\boldsymbol{A}_{v\epsilon}$, $\boldsymbol{A}_{\epsilon\omega_{\mathrm{b}}}$, $\boldsymbol{A}_{\omega\epsilon}$, $\boldsymbol{G}_v$, $\boldsymbol{G}_\omega$ $\boldsymbol{E}_v$, $\boldsymbol{E}_\omega$ constructed by following Algorithm 2.

Theorem 2: For any constant wind, $\boldsymbol{w} = \begin{bmatrix}w_x & w_y & w_z\end{bmatrix}^\top \in \mathbb{R}^3$ having a nonzero horizontal component $\begin{bmatrix}w_x \\ w_y\end{bmatrix}$, and the ensuing equilibrium pair $(\boldsymbol{u}_{\mathrm{eq}}, \boldsymbol{x}_{\mathrm{eq}})$ of dynamics (8), as characterized in (13)-(15), the linearized dynamics of the incremental input-state vector (27) is given by (28) with the matrices constructed as in Algorithm 2.

Proof: To begin with, by exploiting the rotation matrix $\boldsymbol{R}_\psi := \boldsymbol{R}(\boldsymbol{q}_{\mathrm{eq}\psi})$ used in (27), we transform the nonlinear dynamics (8) into rotated coordinates

$$(\boldsymbol{p}_{\mathrm{r}}, \boldsymbol{v}_{\mathrm{r}}, \boldsymbol{q}_{\mathrm{r}}) := \left(\boldsymbol{R}_\psi^\top\boldsymbol{p}, \boldsymbol{R}_\psi^\top\boldsymbol{v}, \boldsymbol{q}_{\mathrm{eq}\psi}^{-1} \otimes \boldsymbol{q}\right), \ \boldsymbol{w}_{\mathrm{r}} := \boldsymbol{R}_\psi^\top\boldsymbol{w} \quad (29)$$

while $\boldsymbol{\omega}_{\mathrm{b}}$ remains unchanged because it is expressed in the body frame. A few observations allow simplifying the transformed dynamics (8): (i) first, we have $\boldsymbol{R}_\psi^\top m\boldsymbol{g} = m\boldsymbol{g}$ because the rotation of ψ is about the $z_{[\mathrm{i}]}$ axis; (ii) secondly, since $\boldsymbol{q}_{\mathrm{r}} = \boldsymbol{q}_{\mathrm{eq}\psi}^{-1} \otimes \boldsymbol{q}$, then $\boldsymbol{R}_\psi^\top\boldsymbol{R}(\boldsymbol{q}) = \boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}})$; (iii) since $\boldsymbol{v}_{\mathrm{b}} := \boldsymbol{R}^\top(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{w})$ (as defined after equation (6)), then $\|\boldsymbol{v}_{\mathrm{b}}\| = \|\boldsymbol{v} - \boldsymbol{w}\| - \|\boldsymbol{v}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r}}\|$ (iv) finally, $\boldsymbol{R}^\top(\boldsymbol{q})\boldsymbol{w} = \boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})\boldsymbol{R}_\psi^\top\boldsymbol{R}_\psi\boldsymbol{w}_{\mathrm{r}} = \boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})\boldsymbol{w}_{\mathrm{r}}$. Based on the observations above, we can derive the rotated version of equations (8) as

$$\dot{\boldsymbol{p}}_{\mathrm{r}} = \boldsymbol{v}_{\mathrm{r}}, \quad (30a)$$

$$m\dot{\boldsymbol{v}}_{\mathrm{r}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}})\left(\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r}})\right), \quad (30b)$$

$$\dot{\boldsymbol{q}}_{\mathrm{r}} = \left(\frac{1}{2}\boldsymbol{q}_{\mathrm{r}} \otimes \begin{bmatrix}0\\ \boldsymbol{\omega}_{\mathrm{b}}\end{bmatrix}\right), \quad (30c)$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{b}} = -[\boldsymbol{\omega}_{\mathrm{b}}]_\times\boldsymbol{J}\boldsymbol{\omega}_{\mathrm{b}} + \boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r}}) \quad (30d)$$

With these new coordinates, the incremental input-state vectors in (27) can be expressed as

$$\tilde{\boldsymbol{x}} = \left(\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{R}_\psi^\top\boldsymbol{p}_{\mathrm{eq}}, \boldsymbol{v}_{\mathrm{r}}, \boldsymbol{\epsilon}_{\mathrm{r}} - \boldsymbol{R}_\psi^\top\boldsymbol{\epsilon}_{\mathrm{eq}}, \boldsymbol{\omega}_{\mathrm{b}}\right),$$
$$\tilde{\boldsymbol{u}} := \boldsymbol{u} - \boldsymbol{u}_{\mathrm{eq}}, \quad \tilde{\boldsymbol{w}} := \boldsymbol{w}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r,eq}} \quad (31)$$

where $\boldsymbol{w}_{\mathrm{r,eq}} = \boldsymbol{R}_\psi^\top \boldsymbol{w}_{\mathrm{eq}} = \left[\begin{smallmatrix} w_{rx} \\ 0 \\ w_{rz} \end{smallmatrix}\right]$, already defined in (21), and $\boldsymbol{R}_\psi^\top \boldsymbol{\epsilon}_{\mathrm{eq}} = \left[\begin{smallmatrix} 0 & \sin(\frac{\theta}{2}) & 0 \end{smallmatrix}\right]^\top$ have both a convenient sparse structure.

By focusing on the rotated dynamics (30) and the expression (31) of the incremental variables, the proof of the theorem amounts to showing that the linearization of (30) about the rotated equilibrium

$$\begin{aligned} \boldsymbol{x}_{\mathrm{r,eq}} &= \left(\boldsymbol{p}_{\mathrm{r,eq}}, \boldsymbol{v}_{\mathrm{r,eq}}, \boldsymbol{\epsilon}_{\mathrm{r,eq}}, \boldsymbol{\omega}_{\mathrm{br,eq}}\right) \\ &= \left(\boldsymbol{R}_\psi^\top \boldsymbol{p}_{\mathrm{eq}}, \left[\begin{smallmatrix} 0 \\ 0 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ \sin(\frac{\theta}{2}) \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ 0 \\ 0 \end{smallmatrix}\right]\right), \ \boldsymbol{w}_{\mathrm{r,eq}} = \left[\begin{smallmatrix} w_{rx} \\ 0 \\ w_{rz} \end{smallmatrix}\right] \end{aligned} \tag{32}$$

coincides with equation (28) and the expressions in Algorithm 2. To this end, inspired by [37, Proof of Lemma 1], to linearize the dynamics of the quaternion $\boldsymbol{q}_{\mathrm{r}} = \left[\eta_{\mathrm{r}} \ \boldsymbol{\epsilon}_{\mathrm{r}}^\top\right]^\top$ evolving in $\mathbb{S}^3$, we replace η_{r} by its positive value induced by the unit norm of the quaternion. Thus, $\eta_{\mathrm{r}} = (1 - \boldsymbol{\epsilon}_{\mathrm{r}}^\top \boldsymbol{\epsilon}_{\mathrm{r}})^{\frac{1}{2}}$.

Let us now first focus on matrix $\boldsymbol{A}_w$ in (28). Its first block row is evidently $\left[\mathbb{0}_3 \ \mathbb{I}_3 \ \mathbb{0}_3 \ \mathbb{0}_3\right]$, due to linearity of equation (30a). For its second block row we focus on equation (30b) and start by characterizing $\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r,eq}})$, whose structure is relatively simple due to the sparsity of $\boldsymbol{\epsilon}_{\mathrm{r,eq}}$. In particular, we recall from (16) that, using the expression of $\boldsymbol{R}$ in (2), we may write

$$\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r,eq}}) = \boldsymbol{R}_\theta := \begin{bmatrix} 1 - 2\bar{\epsilon}_2^2 & 0 & 2\bar{\epsilon}_2\bar{\eta} \\ 0 & 1 & 0 \\ -2\bar{\epsilon}_2\bar{\eta} & 0 & 1 - 2\bar{\epsilon}_2^2 \end{bmatrix} = \left[\begin{smallmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{smallmatrix}\right],$$

where $\bar{\epsilon}_2 = \sin\frac{\theta}{2}$ denotes the second element of $\boldsymbol{\epsilon}_{\mathrm{r,eq}}$ as per (32) and $\bar{\eta} = \sqrt{1 - \bar{\epsilon}_2^2} = \cos\frac{\theta}{2}$.

With this expression of $\boldsymbol{R}_\theta$, we may derive the following expression from (30b), by using the shortcut notation $\cdot\big|_{\mathrm{eq}}$ to characterize the evaluation of a (matrix or vector) function at the equilibrium (32),

$$\begin{aligned} \boldsymbol{A}_{vv} &:= \frac{\partial}{\partial \boldsymbol{v}} \left(\frac{1}{m}\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}}) \left(\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r}})\right)\right)\Big|_{\mathrm{eq}} \\ &= \frac{\partial}{\partial \boldsymbol{v}} \left(\frac{1}{m}\boldsymbol{R}_\theta \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\boldsymbol{R}_\theta^\top \boldsymbol{v}_{\mathrm{r}}\right)\Big|_{\mathrm{eq}}, \end{aligned} \tag{33}$$

which, also considering the identity $\boldsymbol{D}_{\mathrm{f,eq}} = \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})$, is easily shown to coincide with matrix $\boldsymbol{A}_{vv}$ given in (46) in Algorithm 2.

We now focus on the entry $\boldsymbol{A}_{v\epsilon}$ of $\boldsymbol{A}_w$, which should be computed, starting from (30b) in parallel ways to (33), as

$$\boldsymbol{A}_{v\epsilon} := \frac{\partial}{\partial \boldsymbol{\epsilon}} \left(\frac{1}{m}\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}}) \left(\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})\boldsymbol{w}_{\mathrm{r}}\right)\right)\Big|_{\mathrm{eq}}. \tag{34}$$

For evaluating the right-hand side of (34), starting from the expression of $\boldsymbol{R}(\boldsymbol{q}) = \boldsymbol{R}(\left[\begin{smallmatrix} \eta \\ \epsilon \end{smallmatrix}\right])$ in (2), after substituting $\eta = \sqrt{1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} \neq 0$ (we recall that for all of the characterized equilibria we have $\eta \neq 0$), we may compute the generic derivative

$$\begin{aligned} \partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}, \mathfrak{v}) &:= \frac{\partial}{\partial \boldsymbol{\epsilon}} \boldsymbol{R}\left(\left[\begin{smallmatrix} \sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} \\ \boldsymbol{\epsilon} \end{smallmatrix}\right]\right)\mathfrak{v} \\ &= 2\eta\,[\mathfrak{v}]_\times \left(\frac{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top}{1 - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} - \mathbb{I}_3\right) - 4\mathfrak{v}\boldsymbol{\epsilon}^\top + 2\boldsymbol{\epsilon}\mathfrak{v}^\top + 2\boldsymbol{\epsilon}^\top\mathfrak{v}\mathbb{I}_3, \end{aligned} \tag{35}$$

which also implies

$$\frac{\partial}{\partial \boldsymbol{\epsilon}} \boldsymbol{R}^\top(\left[\begin{smallmatrix} \eta \\ \epsilon \end{smallmatrix}\right])\mathfrak{v} = \frac{\partial}{\partial \boldsymbol{\epsilon}} \boldsymbol{R}\left(\left[\begin{smallmatrix} \sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} \\ -\boldsymbol{\epsilon} \end{smallmatrix}\right]\right)\mathfrak{v} = \partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(-\boldsymbol{\epsilon}, \mathfrak{v}). \tag{36}$$

For evaluating (34), it will be useful to derive the following simplified form

$$\begin{aligned} &\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}\left(\left[\begin{smallmatrix} 0 \\ \epsilon_2 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} \mathfrak{v}_1 \\ 0 \\ \mathfrak{v}_3 \end{smallmatrix}\right]\right) \\ &= 2\begin{bmatrix} 0 & \left(\bar{\eta} - \frac{\bar{\epsilon}_2^2}{\bar{\eta}}\right)\mathfrak{v}_3 & 0 \\ -\bar{\eta}\mathfrak{v}_3 & 0 & \bar{\eta}\mathfrak{v}_1 \\ 0 & \left(\frac{\bar{\epsilon}_2^2}{\bar{\eta}} - \bar{\eta}\right)\mathfrak{v}_1 & 0 \end{bmatrix} + 2\bar{\epsilon}_2 \left[\begin{smallmatrix} 0 & -2\mathfrak{v}_1 & 0 \\ \mathfrak{v}_1 & 0 & \mathfrak{v}_3 \\ 0 & -2\mathfrak{v}_3 & 0 \end{smallmatrix}\right]. \end{aligned} \tag{37}$$

We can define two forces (f_{d}, f_ℓ) acting on the drone at the equilibrium, expressed in the body frame, which depend on the wind $\boldsymbol{w}$ and on the two identical elevon inputs δ. These two forces are the drag and lift generated by the airflow over the wing. They are the result of the development of the expression $\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{v}_{\mathrm{b}}\|\boldsymbol{v}_{\mathrm{b}}$ from (30b) with $\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})$ as in (11):

$$\left[\begin{smallmatrix} f_{\mathrm{d}} \\ 0 \\ f_\ell \end{smallmatrix}\right] := -\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\boldsymbol{R}_\theta^\top \boldsymbol{w}_{\mathrm{r,eq}}, \tag{38}$$

which, after some calculations, can be shown to coincide with the selections in (44), given in Algorithm 2.

From the two forces (f_{d}, f_ℓ) in (38), it is possible to determine their partial derivatives with respect to the quaternion component $\bar{\epsilon}_2$ representing the drone's pitch. Using (36), we obtain

$$\left[\begin{smallmatrix} \frac{\partial f_{\mathrm{d}}}{\partial \epsilon_2} \\ 0 \\ \frac{\partial f_\ell}{\partial \epsilon_2} \end{smallmatrix}\right] = -\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(-\boldsymbol{\epsilon}, \boldsymbol{w}_{\mathrm{r,eq}}) \left[\begin{smallmatrix} 0 \\ 1 \\ 0 \end{smallmatrix}\right], \tag{39}$$

which, after some calculations, also considering the identity $\boldsymbol{D}_{\mathrm{f,eq}} = \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})$, can be shown to coincide with the selections in (44), given in Algorithm 2.

Following parallel derivations, the force f_{m} generated by the motors, linked to the propeller traction and the drag generated by the airflow over the wing, and the force f_{e} generated by the elevons, linked to the airflow created by the propellers are obtained from (9) as

$$\left[\begin{smallmatrix} f_{\mathrm{m}} \\ 0 \\ f_{\mathrm{e}} \end{smallmatrix}\right] := \boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}}), \tag{40}$$

which, after some calculations, can be shown to coincide with the selections in (45), given in Algorithm 2.

Using the definitions (35), (36), together with the sparse expressions (37), (38), (40), and their equivalent forms reported in (44), (45) given in Algorithm 2, we finally may compute from (34)

$$\begin{aligned} \boldsymbol{A}_{v\epsilon} := \frac{1}{m}\big(&\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}, \boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})) - \partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}, \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\boldsymbol{R}_\theta^\top \boldsymbol{w}_{\mathrm{eq}}) \\ &- \boldsymbol{R}_\theta \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r,eq}}\|\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(-\boldsymbol{\epsilon}, \boldsymbol{w}_{\mathrm{r,eq}}) \left[\begin{smallmatrix} 0 \\ 1 \\ 0 \end{smallmatrix}\right]\big)\Big|_{\mathrm{eq}}. \end{aligned}$$

which provides expression (47) in Algorithm 2 after some straightforward calculations also exploiting $\boldsymbol{D}_{\mathrm{f,eq}} = \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})$.

We now focus on the entry $\boldsymbol{A}_{\epsilon\omega}$ of $\boldsymbol{A}_w$, and we recall that, due to the properties of the quaternion product (see, e.g., [35]), $\left[\begin{smallmatrix} \eta \\ \epsilon \end{smallmatrix}\right] \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{smallmatrix}\right] = \left[\begin{smallmatrix} -\boldsymbol{\epsilon}^\top \\ \eta\mathbb{I}_3 + [\boldsymbol{\epsilon}]_\times \end{smallmatrix}\right] \boldsymbol{\omega}_{\mathrm{b}}$. From the two lower terms of the

matrix at the right-hand side of this last equation, when developing (30c) and computing $\boldsymbol{A}_{\epsilon\omega} := \frac{\partial}{\partial \boldsymbol{\omega}_\mathrm{b}} \left(\frac{1}{2} \boldsymbol{q}_\mathrm{r} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_\mathrm{b} \end{bmatrix} \right) \Big|_\mathrm{eq}$, we obtain the two terms in expression (47) given in Algorithm 2.

We now focus on the entry $\boldsymbol{A}_{\omega\epsilon}$ of $\boldsymbol{A}_w$, which should be computed starting from (30d). Since only the last term of the right hand-side depends on $\boldsymbol{\epsilon}$ (through $\boldsymbol{q}_\mathrm{r}$), we obtain

$$\boldsymbol{A}_{\omega\epsilon} := \boldsymbol{J}^{-1} \boldsymbol{D}_\mathrm{m}(\boldsymbol{u}_\mathrm{eq}) \|\boldsymbol{w}_\mathrm{r}\| \frac{\partial}{\partial \boldsymbol{\epsilon}} \left(\boldsymbol{R}^\top(\boldsymbol{q}_\mathrm{r})(\boldsymbol{v}_\mathrm{r} - \boldsymbol{w}_\mathrm{r}) \right) \Big|_\mathrm{eq}. \quad (41)$$

To compute the explicit expression of (41), we exploit again (36) and (37), and use the expression of $\boldsymbol{D}_\mathrm{m}$ in (12), together with the identities $\overline{\eta}^2 - \overline{\epsilon}_2^2 = \cos\theta$ and $2\overline{\eta}\overline{\epsilon}_2^2 = \sin\theta$, which provide, after some simplifications, the expression (48), given in Algoirithm 2.

Let us now move on to deriving the entries of matrix $\boldsymbol{G}_w$ in (28), whose components can be derived from (30b) and (30d). Recalling from (7) the four entries of $\boldsymbol{u}$, and also based on the structure of $\boldsymbol{M}_\mathrm{f}$, $\boldsymbol{D}_\mathrm{f}$, in (9), (11), an explicit form for

$$\boldsymbol{G}_v := \frac{1}{m} \boldsymbol{R}_\theta \frac{\partial}{\partial \boldsymbol{u}} \left(\boldsymbol{M}_\mathrm{f}(\boldsymbol{u}) - \boldsymbol{D}_\mathrm{f}(\boldsymbol{u}) \|\boldsymbol{w}_\mathrm{r}\| \boldsymbol{w}_\mathrm{eq}^\mathrm{b} \right) \Big|_\mathrm{eq}, \quad (42)$$

can be computed as in (49), after some straightforward factorizations.

Similarly, based on the matrices $\boldsymbol{M}_\mathrm{m}$, $\boldsymbol{D}_\mathrm{m}$ in in (10), (12), we may compute

$$\boldsymbol{G}_\omega := \boldsymbol{J}^{-1} \frac{\partial}{\partial \boldsymbol{u}} \left(\boldsymbol{M}_\mathrm{m}(\boldsymbol{u}) - \boldsymbol{D}_\mathrm{m}(\boldsymbol{u}) \|\boldsymbol{w}_\mathrm{r}\| \boldsymbol{w}_\mathrm{eq}^\mathrm{b} \right) \Big|_\mathrm{eq} \quad (43)$$

as in (50), after some straightforward factorizations.

Let us finally determine the expression of $\boldsymbol{E}_v$ in (28) as follows. First note that we may write $\|\boldsymbol{w}_\mathrm{r}\| \boldsymbol{w}_\mathrm{r} = \boldsymbol{w}_\mathrm{r} \sqrt{\boldsymbol{w}_\mathrm{r}^\top \boldsymbol{w}_\mathrm{r}}$, so that

$$\frac{\partial}{\partial \boldsymbol{w}_\mathrm{r}} \|\boldsymbol{w}_\mathrm{r}\| \boldsymbol{w}_\mathrm{r} = \|\boldsymbol{w}_\mathrm{r}\| \mathbb{I}_3 + \frac{\boldsymbol{w}_\mathrm{r} \boldsymbol{w}_\mathrm{r}^\top}{\|\boldsymbol{w}_\mathrm{r}\|} = \|\boldsymbol{w}_\mathrm{r}\| \left(\mathbb{I}_3 + \frac{\boldsymbol{w}_\mathrm{r} \boldsymbol{w}_\mathrm{r}^\top}{\boldsymbol{w}_\mathrm{r}^\top \boldsymbol{w}_\mathrm{r}} \right).$$

Then, starting from (30b) and (30d) and following similar computations to the previous cases, also using the expression of $\boldsymbol{w}_\mathrm{r}$ in (29), we obtain expression (51) (reported in Algorithm 2), for $\boldsymbol{E}_v := -\frac{1}{m} \boldsymbol{R}_\theta \frac{\partial}{\partial \boldsymbol{w}_\mathrm{r}} \left(\boldsymbol{D}_\mathrm{f}(\boldsymbol{u}) \|\boldsymbol{w}_\mathrm{r}\| \boldsymbol{w}_\mathrm{r} \right) \Big|_\mathrm{eq}$ and $\boldsymbol{E}_w := -\boldsymbol{J}^{-1} \frac{\partial}{\partial \boldsymbol{w}_\mathrm{r}} \left(\boldsymbol{D}_\mathrm{m}(\boldsymbol{u}) \|\boldsymbol{w}_\mathrm{r}\| \boldsymbol{w}_\mathrm{r} \right) \Big|_\mathrm{eq}$, where we recall that $\boldsymbol{D}_\mathrm{m,eq} = \boldsymbol{D}_\mathrm{m}(\boldsymbol{u}_\mathrm{eq})$. ■

Remark 5: By inspecting the construction in Algorithm 2 and recalling that $\|\boldsymbol{w}_\mathrm{eq}\| = \sqrt{w_{rz}^2 + w_{rx}^2}$, it is apparent that the matrices characterizing the linearized dynamics in (52) depend on the two key parameters (w_{rz}, w_{rx}) characterizing the vertical and horizontal speed of the wind. This parametric dependence will be used in Section III for suitably tuning the feedback controller parameters. ○

Remark 6: Consider again the case of zero-wind discussed in Remark 4. In this case, using the expressions of $\boldsymbol{u}_\mathrm{nowind}$ and $\boldsymbol{u}_\mathrm{nowind,eq}$ defined in equation (26a) and recalling the input transformation $\boldsymbol{u}_\mathrm{nowind} := \begin{bmatrix} \tau_1 & \tau_2 & \delta_1 \tau_1 & \delta_2 \tau_2 \end{bmatrix}^\top$, the zero-wind linearized dynamics is

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}_0 \tilde{\boldsymbol{x}} + \boldsymbol{G}_0 (\boldsymbol{u}_\mathrm{nowind} - \boldsymbol{u}_\mathrm{nowind,eq}), \quad (52)$$

Algorithm 2 Design of the linearization matrices in (28)

Input: Wind vector $\boldsymbol{w}_\mathrm{eq} = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^\top$ and equilibrium $(\boldsymbol{u}_\mathrm{eq}, \boldsymbol{x}_\mathrm{eq})$ from (14) and Algorithm 1.

Output: Matrices $\boldsymbol{A}_w$, $\boldsymbol{G}_w$, $\boldsymbol{E}_w$ in (28)

1: Select parameters ψ, θ, τ, δ in (14) from Algorithm 1 and $\overline{\epsilon}_2 = \sin\frac{\theta}{2}$, $\overline{\eta} = \cos\frac{\theta}{2}$.

2: With the quantities in (16), (11), (12), define:

$$\boldsymbol{R}_\psi := \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R}_\theta := \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

$$\begin{bmatrix} w_{rx} \\ 0 \\ w_{rz} \end{bmatrix} := \boldsymbol{R}_\psi^\top \boldsymbol{w}_\mathrm{eq}, \quad \begin{bmatrix} w_x^\mathrm{b} \\ w_z^\mathrm{b} \end{bmatrix} := \begin{bmatrix} w_{rx}\cos\theta - w_{rz}\sin\theta \\ w_{rz}\cos\theta + w_{rx}\sin\theta \end{bmatrix}$$

$$\left[\boldsymbol{D}_\mathrm{f,eq} \,\middle|\, \boldsymbol{D}_\mathrm{m,eq} \right] := \frac{\rho S}{2} \left[\begin{array}{ccc|ccc} -C_\mathrm{d} & 0 & C_\mathrm{d}\xi_\mathrm{f}\delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta_\mathrm{r} C_\ell \xi_\mathrm{m} \delta & 0 & 2\Delta_\mathrm{r} C_\ell \\ -C_\ell \xi_\mathrm{f} \delta & 0 & -C_\ell & 0 & 0 & 0 \end{array} \right]$$

3: Define the drag and lift forces, and their derivatives with respect to ϵ_2 (defined in Step 1), as

$$\begin{bmatrix} f_\mathrm{d} & \frac{\partial f_\mathrm{d}}{\partial \overline{\epsilon}_2} \\ 0 & 0 \\ f_\ell & \frac{\partial f_\ell}{\partial \overline{\epsilon}_2} \end{bmatrix} := -\|\boldsymbol{w}_\mathrm{eq}\| \boldsymbol{D}_\mathrm{f,eq} \begin{bmatrix} w_x^\mathrm{b} & \left(4\overline{\eta} - \frac{2\overline{\epsilon}_2^2}{\overline{\eta}} \right) w_{rz} - 8\overline{\epsilon}_2 w_{rx} \\ 0 & 0 \\ w_z^\mathrm{b} & \left(4\overline{\eta} - \frac{2\overline{\epsilon}_2^2}{\overline{\eta}} \right) w_{rx} - 8\overline{\epsilon}_2 w_{rz} \end{bmatrix}, \quad (44)$$

4: Define the motor and elevon forces as

$$\begin{bmatrix} f_\mathrm{m} \\ f_\mathrm{e} \end{bmatrix} := \begin{bmatrix} \left(\frac{S_\mathrm{wet} C_\mathrm{d}}{2 S_\mathrm{p}} - 2 \right) \tau \\ -\frac{S_\mathrm{wet} \tau \delta \xi_\mathrm{f} C_\ell}{2 S_\mathrm{p}} \end{bmatrix} \quad (45)$$

5: Select the entries of matrix $\boldsymbol{A}_w$ in (28) as:

$$\boldsymbol{A}_{vv} = \frac{\|\boldsymbol{w}_\mathrm{eq}\|}{m} \boldsymbol{R}_\theta \boldsymbol{D}_\mathrm{f,eq} \boldsymbol{R}_\theta^\top \quad (46)$$

$$\begin{bmatrix} \boldsymbol{A}_{v\epsilon}^{1,2} \\ \boldsymbol{A}_{v\epsilon}^{2,1} \\ \boldsymbol{A}_{v\epsilon}^{2,3} \\ \boldsymbol{A}_{v\epsilon}^{3,2} \end{bmatrix} := \begin{bmatrix} 2\overline{\eta} - \frac{\overline{\epsilon}_2^2}{\overline{\eta}} & 4\overline{\epsilon}_2 & 2\overline{\epsilon}_2^2 - 1 & 2\overline{\epsilon}_2 \overline{\eta} \\ -2\overline{\eta} & -2\overline{\epsilon}_2 & 0 & 0 \\ 2\overline{\epsilon}_2 & -2\overline{\eta} & 0 & 0 \\ -4\overline{\epsilon}_2 & 2\overline{\eta} - \frac{\overline{\epsilon}_2^2}{\overline{\eta}} & -2\overline{\epsilon}_2 \overline{\eta} & 1 - 2\overline{\epsilon}_2^2 \end{bmatrix} \begin{bmatrix} f_\mathrm{e} + f_\ell \\ f_\mathrm{m} + f_\mathrm{d} \\ \frac{\partial f_\mathrm{d}}{\partial \overline{\epsilon}_2} \\ \frac{\partial f_\ell}{\partial \overline{\epsilon}_2} \end{bmatrix}$$

$$\boldsymbol{A}_{v\epsilon} = \frac{1}{m} \begin{bmatrix} 0 & \boldsymbol{A}_{v\epsilon}^{1,2} & 0 \\ \boldsymbol{A}_{v\epsilon}^{2,1} & 0 & \boldsymbol{A}_{v\epsilon}^{2,3} \\ 0 & \boldsymbol{A}_{v\epsilon}^{3,2} & 0 \end{bmatrix}, \boldsymbol{A}_{\epsilon\omega} = \frac{\overline{\eta}}{2} \mathbb{I}_3 + \frac{\overline{\epsilon}_2}{2} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (47)$$

$$\boldsymbol{A}_{\omega\epsilon} = \frac{\rho S C_\ell \Delta_\mathrm{r} \|\boldsymbol{w}_\mathrm{eq}\| (w_x^\mathrm{b} - \xi_\mathrm{m} \delta w_z^\mathrm{b})}{J_y \overline{\eta}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (48)$$

6: Select the entries of matrix $\boldsymbol{G}_w$ in (28) as:

$$\boldsymbol{G}_v = \frac{1}{m} \boldsymbol{R}_\theta \left[\boldsymbol{G}_{v\tau} \,\middle|\, \boldsymbol{G}_{v\delta} \right], \; \boldsymbol{G}_{v\tau} := \begin{bmatrix} 1 - \frac{S_\mathrm{wet} C_\mathrm{d}}{4 S_\mathrm{p}} \\ 0 \\ -\frac{S_\mathrm{wet} C_\ell \xi_\mathrm{f} \delta}{2 S_\mathrm{p}} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top$$

$$\boldsymbol{G}_{v\delta} := \begin{bmatrix} -\frac{1}{4} \rho S C_\mathrm{d} \xi_\mathrm{f} \|\boldsymbol{w}_\mathrm{eq}\| w_z^\mathrm{b} \\ 0 \\ -\frac{S_\mathrm{wet} C_\ell \xi_\mathrm{f} \tau}{2 S_\mathrm{p}} + \frac{1}{4} \rho S C_\ell \xi_\mathrm{f} \|\boldsymbol{w}_\mathrm{eq}\| w_x^\mathrm{b} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top \quad (49)$$

$$\boldsymbol{G}_\omega = \boldsymbol{J}^{-1} \left[\boldsymbol{G}_{\omega\tau} \;\; \boldsymbol{G}_{\omega\delta} \right], \boldsymbol{G}_{\omega\delta} := \frac{S_\mathrm{wet} C_\ell \tau}{4 S_\mathrm{p}} \begin{bmatrix} a_y \xi_\mathrm{f} & -a_y \xi_\mathrm{f} \\ \Delta_\mathrm{r} \xi_\mathrm{m} & \Delta_\mathrm{r} \xi_\mathrm{m} \\ 0 & 0 \end{bmatrix} + \frac{\rho S \|\boldsymbol{w}_\mathrm{eq}\| \xi_\mathrm{m}}{4} \begin{bmatrix} a_y C_\mathrm{d} w_x^\mathrm{b} & -a_y C_\mathrm{d} w_x^\mathrm{b} \\ \Delta_\mathrm{r} C_\ell w_x^\mathrm{b} & \Delta_\mathrm{r} C_\ell w_x^\mathrm{b} \\ a_y C_\ell w_z^\mathrm{b} & -a_y C_\ell w_z^\mathrm{b} \end{bmatrix} \quad (50)$$

$$\boldsymbol{G}_{\omega\tau} := \begin{bmatrix} \frac{k_\mathrm{m}}{k_\mathrm{f}} + \frac{S_\mathrm{wet}}{4 S_\mathrm{p}} a_y \xi_\mathrm{f} C_\ell \delta \\ 0 \\ p_y + \frac{S_\mathrm{wet}}{4 S_\mathrm{p}} a_y C_\mathrm{d} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^\top + \begin{bmatrix} 0 \\ \frac{S_\mathrm{wet}}{4 S_\mathrm{p}} \Delta_\mathrm{r} \xi_\mathrm{m} C_\ell \delta \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top$$

7: Select the entries of matrix $\boldsymbol{E}_w$ in (28) as:

$$\begin{bmatrix} \boldsymbol{E}_v \\ \boldsymbol{E}_\omega \end{bmatrix} = - \begin{bmatrix} \boldsymbol{A}_{vv} \\ \boldsymbol{J} \|\boldsymbol{w}_\mathrm{eq}\| \boldsymbol{D}_\mathrm{m,eq} \boldsymbol{R}_\theta^\top \end{bmatrix} \left(\mathbb{I}_3 + \frac{\boldsymbol{R}_\psi^\top \boldsymbol{w}_\mathrm{eq} \boldsymbol{w}_\mathrm{eq}^\top \boldsymbol{R}_\psi}{\boldsymbol{w}_\mathrm{eq}^\top \boldsymbol{w}_\mathrm{eq}} \right) \quad (51)$$

Return: $\boldsymbol{A}_w$, $\boldsymbol{G}_w$, $\boldsymbol{E}_w$

where the expression of $\boldsymbol{A}_0$ is

$$\boldsymbol{A}_0 = \boldsymbol{A}_w\Big|_{\boldsymbol{w}=0} = \begin{bmatrix} \mathbb{0}_3 & \mathbb{I}_3 & \mathbb{0}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{v\epsilon} & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{\epsilon\omega} \\ \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 \end{bmatrix}, \qquad (53)$$

with the following selections

$$\boldsymbol{A}_{\epsilon\omega} = \frac{\sqrt{2}}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{A}_{v\epsilon} = \sqrt{2}\begin{bmatrix} 0 & -2g & 0 \\ g & 0 & g \\ 0 & -2g & 0 \end{bmatrix},$$

while the expression of $\boldsymbol{G}_0$ is

$$\boldsymbol{G}_0 := \begin{bmatrix} \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} \\ 0 & 0 & a_\mathrm{g} & a_\mathrm{g} \\ 0 & 0 & 0 & 0 \\ b_\mathrm{g} & b_\mathrm{g} & 0 & 0 \\ \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} \\ c_\mathrm{g} & -c_\mathrm{g} & d_\mathrm{g} & -d_\mathrm{g} \\ 0 & 0 & e_\mathrm{g} & e_\mathrm{g} \\ f_\mathrm{g} & -f_\mathrm{g} & 0 & 0 \end{bmatrix},$$

with

$$\left[\begin{array}{c|c} a_\mathrm{g} & b_\mathrm{g} \\ \hline c_\mathrm{g} & d_\mathrm{g} \\ \hline e_\mathrm{g} & f_\mathrm{g} \end{array}\right] = \left[\begin{array}{c|c} -\frac{S_\mathrm{wet}}{4mS_\mathrm{p}}C_\ell\xi_\mathrm{f} & \frac{1}{m}(1-\frac{S_\mathrm{wet}}{2S_\mathrm{p}}C_\mathrm{d}) \\ \hline \frac{k_\mathrm{m}}{J_x k_\mathrm{f}} & \frac{S_\mathrm{wet}a_y}{4J_x S_\mathrm{p}}C_\ell\xi_\mathrm{f} \\ \hline \frac{S_\mathrm{wet}\Delta_\mathrm{r}}{4J_y S_\mathrm{p}}C_\ell\xi_\mathrm{m} & \frac{1}{J_z}(p_y+\frac{S_\mathrm{wet}}{4S_\mathrm{p}}a_y C_\mathrm{d}) \end{array}\right].$$

We emphasize that these equations coincide with the expressions given in our preliminary work [33, eqn. (22)]. ○

III. Integral-based linear control

A. Description of the control scheme

A careful inspection of the control and the disturbance input matrices $\boldsymbol{G}_w$ and $\boldsymbol{E}_w$ in model (28) (see the output of Algorithm 2) suggests an effective control architecture to reject a constant wind disturbance $\boldsymbol{w}$. Indeed, the ailerons and the propellers can be used symmetrically to generate respectively a moment about the $y_{[\mathrm{b}]}$ axis, verifying equation (19) and a force along the $x_{[\mathrm{b}]}$ axis, verifying equation (17), thus compensating for the disturbance effect. Nevertheless, there is still a force along the $z_{[\mathrm{b}]}$ axis to be compensated for by verifying equation (18), and an integral action can asymptotically converge to the desired force, even with a non-measured wind disturbance $\boldsymbol{w}$. We may thus stabilize the UAV at a hovering equilibrium as characterized in Theorem 1. Since we don't measure the wind $\boldsymbol{w}$, the values of ψ and θ in Algorithm 1 are unknown. The proposed controller, shown in Fig. 6, uses integral action to obtain these two unknown angles. Its feedback loop involves the following error variables output, which should converge to zero in any hovering position:

$$\boldsymbol{e}_p = \boldsymbol{r}_p - \boldsymbol{p},\ \boldsymbol{e}_{v\epsilon\omega} = -\begin{bmatrix} \mathbb{I}_3 & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times2} & \mathbb{0}_3 \\ \mathbb{0}_{1\times3} & 1 & \mathbb{0}_{1\times2} & \mathbb{0}_{1\times3} \\ \mathbb{0}_3 & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times2} & \mathbb{I}_3 \end{bmatrix}\begin{bmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\epsilon} \\ \tilde{\boldsymbol{\omega}}_\mathrm{b} \end{bmatrix}, \qquad (54)$$

where $\boldsymbol{r}_p \in \mathbb{R}^3$ is the constant position reference comprising a target position for the translational motion (note that $\boldsymbol{r}_p$ is the reference input to the control scheme).

The error variables in (54) can be represented as in the block diagram of Fig. 6 by defining the output $\boldsymbol{y} \in \mathbb{R}^{10}$ of the linearized plant dynamics (28), having the incremental state vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^{10\times1}$, as follows

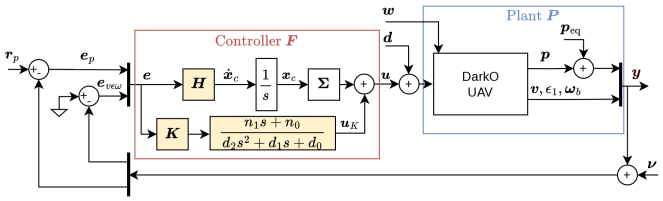

Fig. 6. Proposed integral-based controller with the wind perturbation $\boldsymbol{w}$, a plant-input perturbation $\boldsymbol{d}$ and a plant-output perturbation $\boldsymbol{\nu}$.

$$\boldsymbol{y} = \boldsymbol{C}\tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{p}_\mathrm{eq} \\ \mathbb{0}_{7\times1} \end{bmatrix}, \quad \boldsymbol{C} := \begin{bmatrix} \mathbb{I}_6 & \mathbb{0}_{6\times1} & \mathbb{0}_{6\times2} & \mathbb{0}_{6\times3} \\ \mathbb{0}_{1\times6} & 1 & \mathbb{0}_{1\times2} & \mathbb{0}_{1\times3} \\ \mathbb{0}_{3\times6} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times2} & \mathbb{I}_3 \end{bmatrix}, \qquad (55)$$

where the output matrix $\boldsymbol{C} \in \mathbb{R}^{10\times12}$ removes the $\tilde{\epsilon}_2$ and $\tilde{\epsilon}_3$ components from the state vector $\tilde{\boldsymbol{x}}$.

As shown in Fig. 6, the controller dynamic equations are based on the measured error $\boldsymbol{e}$ as follows

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_p^\top & \boldsymbol{e}_{v\epsilon\omega}^\top \end{bmatrix}^\top, \quad \dot{\boldsymbol{x}}_c = \boldsymbol{H}\boldsymbol{e}, \quad \boldsymbol{u} = \boldsymbol{\Sigma}\boldsymbol{x}_c + \boldsymbol{u}_K,$$
$$\boldsymbol{\Sigma} := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^\top, \quad \boldsymbol{u}_K = \frac{n_1 s + n_0}{d_2 s^2 + d_1 s + d_0}\boldsymbol{K}\boldsymbol{e}, \qquad (56)$$

where $\boldsymbol{x}_c \in \mathbb{R}^2$ is the integral action state; $\boldsymbol{\Sigma}$ is an input allocation matrix that allows assigning the first component of the integrator state to the propellers action and the second component to the elevons action. Scalars n_1, n_0, d_2, d_1, d_0 are respectively the numerator and denominator coefficients of a filter used to avoid a direct input-output transmission that would amplify high-frequency measurement noise. This filter induces a strictly proper controller, for increased robustness to additive uncertainties. We define the controller $\boldsymbol{F}$ having dimensions 4×10 having transfer matrix $\boldsymbol{F}(s) = T_{\boldsymbol{e}\to\boldsymbol{u}}(s)$ as described in (56) and interconected as in Fig. 6. The plant $\boldsymbol{P}$ having dimensions 10×4 represents the linearized DarkO dynamics. The output of the plant $\boldsymbol{y} \in \mathbb{R}^{10\times1}$ is used as the input of controller $\boldsymbol{F}$.

In view of the symmetries of the actuators on the UAV, we have constrained the structure of matrix $\boldsymbol{K}$ in (56), associated with the controller's proportional action, in order to use the actuators in a physically meaningful way as follows:

$$\boldsymbol{K}_\mathrm{struct} = \begin{bmatrix} k_1 & -k_2 & k_3 & k_4 & -k_5 & k_6 & -k_7 & k_8 & k_9 & -k_{10} \\ k_1 & k_2 & k_3 & k_4 & k_5 & k_6 & k_7 & -k_8 & -k_9 & k_{10} \\ -k_{11} & -k_{12} & k_{13} & -k_{14} & -k_{15} & -k_{16} & k_{17} & -k_{18} & k_{19} & -k_{20} \\ -k_{11} & k_{12} & k_{13} & -k_{14} & k_{15} & k_{16} & -k_{17} & k_{18} & k_{19} & k_{20} \end{bmatrix}. \qquad (57)$$

In particular, a position error on the $z_{[\mathrm{i}]}$ axis of the NED world frame (see Fig. 2) results in a symmetric use of the two propellers that generates a force along the $x_{[\mathrm{b}]}$ axis of the UAV. The symmetric use of the two motors is reflected by the same-sign in coefficients k_3 and k_6 on columns 3 and 6 of $\boldsymbol{K}$, corresponding respectively to the position and velocity errors on the $z_{[\mathrm{i}]}$ axis. Similarly, a position or speed error along the drone's lateral axis $y_{[\mathrm{b}]}$ will be compensated for by an antisymmetric use of the motors, as reflected by the coefficients k_2 and k_5 and their opposite signs on columns 2 and 5 of $\boldsymbol{K}$. An angular velocity error about the $x_{[\mathrm{b}]}$ axis must

be compensated for by an antisymmetric use of the elevons, as reflected by coefficient k_{18} having opposite signs on column 8 of $\boldsymbol{K}$. Parallel arguments explain the remaining coefficients of matrix $\boldsymbol{K}$ in (57). An advantage of the structure in (57) is the reduction of the number of variables to be optimized, from 40 to 20 scalar gains.

The closed loop shown in Fig. 6, is an output feedback with 10 outputs, consisting of the three linear positions, the three linear velocities, one of the three attitude angles (ϵ_1) and the three angular velocities. This structure can be seen as a MIMO proportional-integral feedback. The parameters to be tuned in controller $\boldsymbol{F}$ (56) are the proportional gain $\boldsymbol{K} \in \mathbb{R}^{4\times 10}$ in (57), the integral gain $\boldsymbol{H} \in \mathbb{R}^{2\times 10}$ and the filter parameters n_1, n_0, d_2, d_1, d_0, as highlighted in yellow in Fig. 6. A suitable tuning method should ensure desirable disturbance rejection and satisfactory robustness to unmodeled dynamics. These two goals lead to a trade-off because disturbance rejection requires an aggressive tuning while robustness properties are ensured by a frequency roll-off strategy. We discuss next two optimization-based tuning methods. The first one is issued from the ideas proposed in [34], which did not need the linearized dynamics of Theorems 1 and 2, and is summarized in Section III-B. It is a multi-objective synthesis with H_∞ constraints based on the zero-wind model discussed in Remarks 4 and 6 and derived in [34]. We will show that this first method fails to stabilize the drone in certain wind ranges, due to the lack of knowledge of the dynamics characterized in Theorems 1 ans 2. The second tuning method, presented in Sec.III-C, is an iterative multi-objective synthesis with H_∞ constraints, based on a collection of models associated with different wind conditions and derived based on Theorems 1 and 2, through Algorithms 1 and 2. In our numerical validation,

Measurement	Value	Units
$\boldsymbol{p}$	2.5×10^{-4}	m
$\tilde{\boldsymbol{v}}$	1.2×10^{-3}	$\mathrm{m\,s^{-1}}$
$\tilde{\boldsymbol{\epsilon}}$	4.7×10^{-4}	
$\tilde{\boldsymbol{\omega}}_\mathrm{b}$	2.7×10^{-3}	$\mathrm{rad\,s^{-1}}$

TABLE II
STANDARD DEVIATION OF THE MODELED SENSOR NOISE ADDED TO THE SIMULATED MEASUREMENTS.

reported in Sections III-B and III-C (see in particular Fig. 7 and Fig. 9), measurement noise is added to the output to produce practically reasonable numerical results. The standard deviations of the adopted noise levels are reported in Table II. Moreover, in addition to reporting the simulation results of the linear feedback of Fig. 6 with the linearized model (28), in Sections III-B and III-C, we also simulate the closed loop by replacing the linearized plant $\boldsymbol{P}$ with the nonlinear model (1) including many real-world effects. When replacing the linearized plant with the nonlinear dynamics (1), whose state is $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}_\mathrm{b}) \in \mathbb{R}^{13}$, we replace the linear output $\boldsymbol{y}$ with the following surrogate nonlinear version

$$\boldsymbol{y}_\mathrm{NL} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ \epsilon_1 \\ \boldsymbol{\omega}_\mathrm{b} \end{bmatrix} = \begin{bmatrix} \mathbb{I}_6 & \mathbb{0}_{6\times 1} & \mathbb{0}_{6\times 1} & \mathbb{0}_{6\times 2} & \mathbb{0}_3 \\ \mathbb{0}_{1\times 3} & 0 & 1 & \mathbb{0}_{1\times 2} & \mathbb{0}_{1\times 3} \\ \mathbb{0}_3 & \mathbb{0}_{3\times 1} & \mathbb{0}_{3\times 1} & \mathbb{0}_{3\times 2} & \mathbb{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_\psi^\top \boldsymbol{p} \\ \boldsymbol{R}_\psi^\top \boldsymbol{v} \\ \boldsymbol{q}_{\mathrm{eq}\psi}^{-1} \otimes \boldsymbol{q} \\ \boldsymbol{\omega}_\mathrm{b} \end{bmatrix}. \quad (58)$$

In the next sections we denote the modulus margin of a transfer matrix $s \mapsto T_{v\to z}$ as $\Delta_m(T_{v\to z}) = \min_{\omega\in R} \sigma_{\min}(T_{v\to z}(j\omega))$.

B. Zero-wind H_∞-based controller tuning

For tuning the controller based on the zero wind, we use the linear plant model detailed in Remark 6, $\boldsymbol{P}(s) = T_{\boldsymbol{u}\to\boldsymbol{y}}(s)$, obtained from equations (52) and (55) as

$$\boldsymbol{P}(s) = \boldsymbol{C}(s\mathbb{I}_{12} - \boldsymbol{A}_0)^{-1}\boldsymbol{G}_0.$$

With reference to Fig. 6, we introduce transfer matrices that correspond to robustness objectives: the output sensitivity function as $T_{\nu\to e} = (\mathbb{I}_{10} + \boldsymbol{PF})^{-1}$ having dimensions 10×10, so that, $\|T_{\nu\to e}\|_\infty = \Delta_m(T_{\nu\to e})^{-1}$ and the input sensitivity function $T_{d\to u} = (\mathbb{I}_4 + \boldsymbol{FP})^{-1}$ having dimensions 4×4, so that $\|T_{d\to u}\|_\infty = \Delta_m(T_{d\to u})^{-1}$. Consequently, the minimization of the H_∞-norm of $T_{\nu\to e}$ or $T_{d\to u}$, corresponds to increasing the input and output modulus margins. Since plant $\boldsymbol{P}$ is MIMO, we give importance to both the input and the output sensitivity functions which do not coincide,

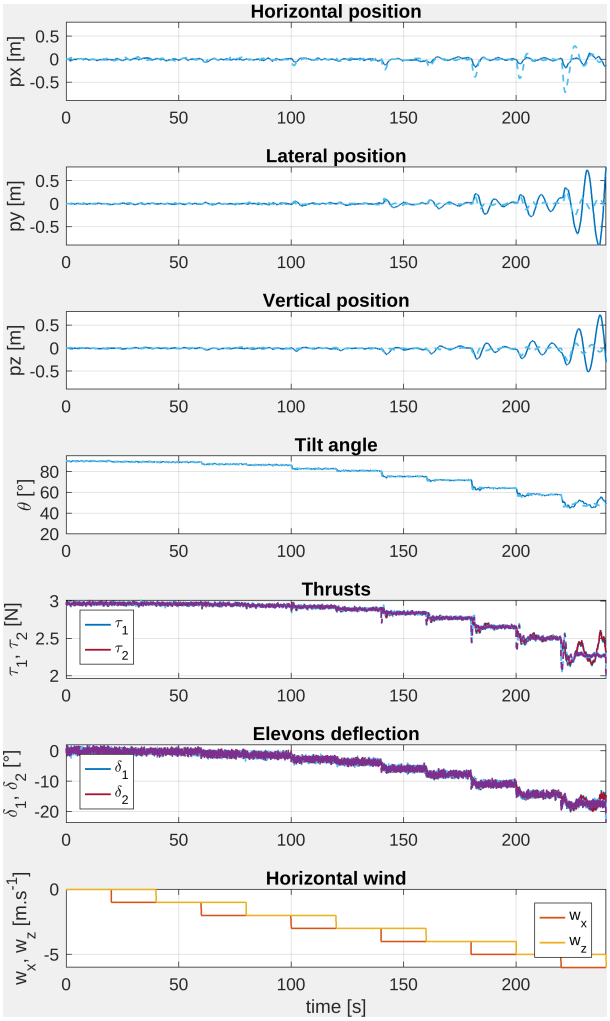

Fig. 7. Simulation of the non-linear model (1) (solid line) and the linearized model (28) (dashed line) with increasing constant wind steps with the controller tuned using the zero-wind optimization of Section III-B.

because $\boldsymbol{P}$ and $\boldsymbol{F}$ do not commute. We also define the transfer matrix $T_{\nu\to u}$ having dimensions 4×10 linked to the impact of the measurement noise ν on the control input $\boldsymbol{u}$, and $T_{d\to y}$ having dimensions 10×4 representing the impact of the input disturbance $\boldsymbol{d}$ on the plant output $\boldsymbol{y}$. We solve the same problem as in our preliminary work [34, eqn. (13)] using the `Systune` software [38], however we use the control diagram presented in section III-A which includes a filter on the proportional action and a different number of outputs. We also include in the plant $\boldsymbol{P}$ the linear actuators dynamics discussed in Remark 3.

Successive steps of increasing horizontal and vertical wind intensity (ranging from zero to $-6\,\mathrm{m\,s^{-1}}$) are applied, as shown in the lower plot of Fig. 7. The selected wind pairs (w_x, w_z) are represented by red dots on the surfaces in Fig. 5, where we can see that the equilibrium $(\boldsymbol{u}_{\text{eq}}, \boldsymbol{x}_{\text{eq}})$ is reached without saturating the actuators. We only focus on the negative part of the vertical wind speed because it is the most limiting one. Indeed, the drone is lifted by the rising vertical wind (whose sign is negative in the NED frame), so it needs less traction on the propellers to compensate for the gravity. The motors generate less airflow over the elevons, which reduces their efficiency, leads to saturation, and destabilizes the drone. The aim of the control system is to keep the UAV at the hovering position (defined as $\boldsymbol{r}_p = [0,0,0]^\top$), despite the increasing horizontal and vertical wind w_x and w_z. Fig. 7 both linear simulations with the linearized plant dynamics (28) (dashed) and nonlinear simulations with the accurate model (1) (solid). Both the linear and nonlinear simulations consistently show that the controller performs well at low wind speed (in fact, the tuning is performed based on the zero-wind model). However, when the wind speed w_x and w_z exceed $-5\,\mathrm{m\,s^{-1}}$, the hovering position becomes unstable and the drone oscillates and diverges. The tilt angles θ are used to represent the attitude to give a better insight of the vehicle behavior, however the nonlinear simulation of the nonlinear dynamics (1) is carried out with unit quaternions. The instability observed in the simulation results of Fig. 7 confirms the experimental instabilities reported in [34] where we used this same tuning method, and confirms the importance of Theorems 1 and 2 in Section II, for an appropriate tuning of the controller gains, which is performed in the next section.

C. Multimodel H_∞-based controller tuning

The simulation results obtained with the zero-wind tuning method (see Fig. 7) together with the experimental instabilities observed in [34] confirm the need for a controller gain tuning procedure exploiting the parametrized non-zero wind linearizations of Theorems 1 and 2. Focusing again on the control scheme of Fig. 6, we now explicitly consider the (linearized) wind effect on the plant, and we consider the linearized plant dynamics (28) with output (55) and with the selections in Algorithm 2 as

$$\begin{aligned}\boldsymbol{P}_w(s) &= \begin{bmatrix}\boldsymbol{P}_u(s;w) & \boldsymbol{P}_w(s;w)\end{bmatrix} \\ &:= \boldsymbol{C}(s\mathbb{I}_{12} - \boldsymbol{A}_w)^{-1}\begin{bmatrix}\boldsymbol{G}_w & \boldsymbol{E}_w\end{bmatrix},\end{aligned} \tag{59}$$

whose input is the concatenation of the control input $\boldsymbol{u}$ and the wind disturbance input $\boldsymbol{w}$. As the model depends on the wind speed $\boldsymbol{w}$, we introduce a new transfer matrix $T_{w\to y}$ having dimensions 10×3, which corresponds to the transfer matrix between the wind input $\boldsymbol{w}$ and the plant output $\boldsymbol{y}$, quantifying the effect of the wind disturbance on the UAV feedback loop. With the set of transfers matrices defined in Sec. III-B and the new transfer matrix $T_{w\to y}$, we use the algorithmic approach in [38], [39], named "`systune`", which uses non-smooth optimization techniques to deal with non-convex tuning problems, such as our structured control architecture where we optimize the gain matrices $\boldsymbol{K}$, $\boldsymbol{H}$ and the filter parameters n_1, n_0, d_2, d_1, d_0 (in yellow on the figure 6). As reported in [39, eq. (2)], we solve a multi-objective optimization problem, by exploiting the Matlab implementation well explained in [39, §3]. In particular, based on a set $\mathcal{W}$ comprising a finite collection of pairs (w_x, w_z), with $w_x \in [0,\ 8]\ \mathrm{m\,s^{-1}}$ and $w_z \in [\text{-}4,\ 4]\ \mathrm{m\,s^{-1}}$, we consider the ensuing set of linearized plants (59) and solve the following convex optimization, where scalars W_1, W_2, W_3, W_4 and W_5 are weighting factors to be tuned to obtain a satisfactory trade-off between robustness (associated with W_2, W_3 and W_4) and performance (associated with W_1 and W_5)

$$\gamma^\star = \min_{\boldsymbol{F}} \max_{w\in\mathcal{W}} \left\| \begin{matrix} \|W_1 T_{\nu\to e}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_2 T_{d\to u}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_3 T_{\nu\to u}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_4 T_{d\to y}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_5 T_{w\to y}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \end{matrix} \right\|_\infty, \text{ subject to } \tag{60}$$
$$\boldsymbol{F} \text{ stabilizes internally } \mathcal{F}_\ell(\boldsymbol{P}_w, \boldsymbol{F}), \forall w \in \mathcal{W},$$

where $\mathcal{F}_\ell(\boldsymbol{P}_w, \boldsymbol{F})$ denotes the linear feedback interconnection of Fig. 6 for a specific value of w (this is consistent with the classical robust control notation [38], [39]). Notice that, with reference to [39, eq. (2)], we only specify soft constraints and we do not specify any hard constraint.

The optimization problem (60) becomes increasingly cumbersome, from a computational viewpoint, as we increase the cardinality of the set of wind conditions considered in $\mathcal{W}$. In fact, a brute force approach including a fine grid of points in $\mathcal{W}$ leads to a computationally intractable optimization. Instead, we follow here the iterative procedure overviewed in Algorithm 3, where $\mathcal{W}$ is initially selected as a sparse grid comprising $3 \times 3 = 9$ points (step 1) and then a synthesis step (step 2) is repeatedly followed by a (computationally simple) analysis step (step 3) where controller $\boldsymbol{F}$ is fixed. Step 3 identifies the violating points by using a finer validation grid $\mathcal{W}_{\text{v}}$ and adds them to the optimization set $\mathcal{W}$. The algorithm terminates after some iterations, when no points of the validation grid violate the constraints.

Executing Algorithm 3 for the DarkO models of Theorems 1 and 2 with the selection of the positive weighting scalars W_1–W_5 reported in Table III-C, returned the following selection

Weighting scalars	W_1	W_2	W_3	W_4	W_5
Values	18	16	11	26	5

TABLE III
VALUES OF THE POSITIVE WEIGHTING SCALARS $\boldsymbol{W_1}$–$\boldsymbol{W_5}$ USED IN THE EXECUTION OF ALGORITHM 3.

Algorithm 3 Iterative multimodel controller gain tuning.

Input: $\boldsymbol{A}_w$, $\boldsymbol{G}_w$, $\boldsymbol{E}_w$ the output matrices of Algorithm 2 and the positive weighting scalars W_1–W_5

Output: $\boldsymbol{K}$, $\boldsymbol{H}$ and the filter gains

1: (Initialization) Initialize $\mathcal{W}$ as a grid comprising all the pairs $w_x \in \{0,\ -4,\ -8\}$ and $w_z \in \{-4,\ 0,\ 4\}$

2: (Synthesis) Solve the optimization (60) with the software `systune`

3: (Analysis) Define a validation grid $\mathcal{W}_\text{v}$ by discretizing the interval $(w_x, w_y) \in [0, 8] \times [-4, 4]$ with a discretization step of 1 and using the controller $\boldsymbol{F}$ obtained from the previous step, compute, for each $w_\text{v} \in \mathcal{W}_\text{v}$,

$$\gamma_\text{v} = \left\| \begin{matrix} \|W_1 T_{\nu\to e}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_2 T_{d\to u}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_3 T_{\nu\to u}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_4 T_{d\to y}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_5 T_{w\to y}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \end{matrix} \right\|_\infty , \tag{61}$$

and augment $\mathcal{W}$ with the corresponding point if $\gamma_\text{v} > 1$ or γ_v is undefined (namely if $\boldsymbol{F}$ is not internally stabilizing).

4: (Termination) If $\mathcal{W}$ has not been augmented at the previous step, then move to step 5, otherwise move to step 2.

5: **Return:** $\boldsymbol{K}$, $\boldsymbol{H}$ and filter parameters n_1, n_0, d_2, d_1, d_0

after 2 iterations:

$$\left[\, \boldsymbol{K}^\top \,\middle|\, \boldsymbol{H}^\top \,\right] = \left[\begin{array}{cccc|cc} -3.86 & -3.86 & 0.79 & 0.79 & 0.02 & 0.48 \\ 1.43 & -1.43 & 1.71 & -1.71 & -0.47 & -1.63 \\ 4.06 & 4.06 & -2.07 & -2.07 & -0.45 & 0.52 \\ -6.86 & -6.86 & -11.60 & -11.60 & -0.14 & 1.40 \\ -10.75 & 10.75 & -1.89 & 1.89 & 3.35 & 5.69 \\ 27.20 & 27.20 & -4.29 & 4.29 & -1.84 & 3.79 \\ -12.32 & 12.32 & -3.46 & 3.46 & 3.72 & 6.81 \\ -5.84 & 5.84 & -2.29 & 2.29 & 1.58 & 3.13 \\ -5.19 & 5.19 & 5.79 & 5.79 & 2.86 & -1.54 \\ -6.52 & 6.52 & 0.08 & -0.08 & 0.08 & 2.82 \end{array}\right],$$

$$\left[\begin{array}{c|c} n_1 & n_0 \\ \hline d_2 & d_1 \\ \hline d_0 & \end{array}\right] = \left[\begin{array}{c|c} -429 & -389 \\ \hline 1 & 6475 \\ \hline 4905 & \end{array}\right], \tag{62}$$

For the first iteration of Algorithm 3, after a candidate controller $\boldsymbol{F}$ has been evaluated at step 2, Fig. 8 shows in blue the bode diagrams of the maximum singular values of $T_{\nu\to e}$, $T_{d\to u}$, $T_{\nu\to u}$, $T_{d\to y}$, and $T_{w\to y}$ (associated with the value of γ_v) reported in (61) at the analysis step 3, to be compared to the inverse of the five weights W_1–W_5, represented by the green horizontal lines. The diagrams in red correspond to the points that violate the constraints and that are added to the set $\mathcal{W}$ for the next iteration. The few diagrams in magenta, instead, correspond to the 9 points considered in $\mathcal{W}$ for the first iteration of the synthesis step 2. The red diagrams in Fig. 8 clearly illustrate that the iterative algorithm manages to detect the critical values of wind speed (w_x, w_z) to be added to the optimization set $\mathcal{W}$.

The singular values of the output and the input sensitivity function (respectively $T_{r\to e}$ and $T_{d\to u}$) are shown in Fig. 8 top line. The graph in the third line represents the singular value of the transfer between the wind disturbance $\boldsymbol{w}$ and the drone output $\boldsymbol{y}$. The singular value tangent to the constraint is that for the highest wind condition a.g. $(w_x, w_z) = (-8, -4)\ \mathrm{m\,s^{-1}}$.

With the tuning reported in (62), as obtained with Algorithm 3, we report in Fig. 9 parallel simulation results to

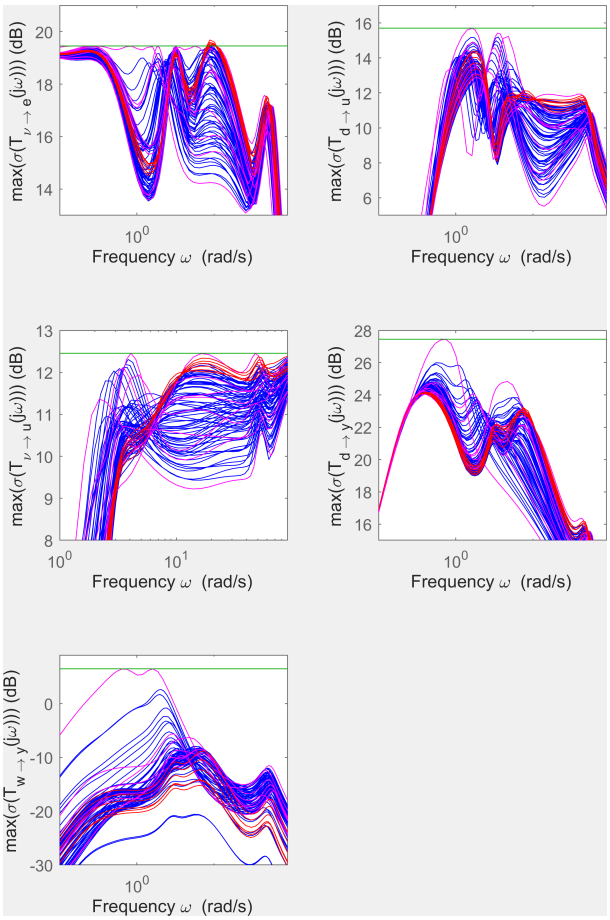

Fig. 8. Diagrams of the singular values of the transfer functions in (61) at the first iteration of Algorithm 3.

those already shown in Fig. 7 for the zero-wind tuning method discussed in Section III-B. Once again we simulate both the nonlinear plant (1) (solid lines) and the linearized plant (28) (dashed line). As compared to Fig. 7, the simulations of Fig. 9 show that the controller tuning based on Theorems 1 and 2 solves the instability issues and manages to stabilize the hovering condition in all of the considered wind scenarios. We also note from Fig. 9 shows a more aggressive action, indeed the control input u (both thrust and deflections) is more affected by the measurement noise. The effectiveness of the control scheme tuned on the basis of Algorithm 3 is also confirmed by the experimental results reported in the next section.

IV. Experimental flight with open wind tunnel

DarkO's experimental flight took place in a dedicated space (see Fig. 10) with an Optitrack localization system based on a NED convention as per Figure 2. We used an open-vein wind generator to obtain wind steps that we measured with a hot-wire probe (the vertical bar in Fig. 10). Although this wind information is recorded on board the drone to synchronize the data, we do not use this measurement in the control law. The measurement frequency of this wind probe is only 0.5 Hz, so we only have one measurement every two seconds. The state estimation is carried out using an inertial navigation system

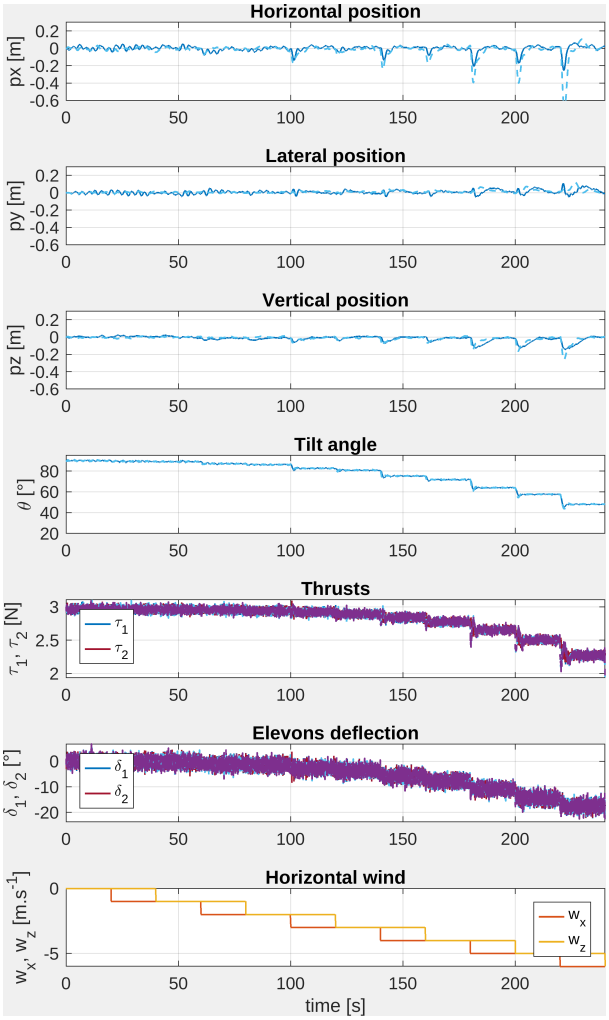

Fig. 9. Simulation of the non-linear model (1) (solid line) and the linearized model (28) (dashed line) with increasing constant wind steps with the controller tuned using the multimodel optimization of Algorithm 3 in Section III-C.

to merge the Inertial Measurement Unit (IMU) + Optitrack sensor data in order to obtain an accurate estimation of the output $\boldsymbol{y}$ in Fig. 6. However, the drone's angular velocity $\boldsymbol{\omega}_\mathrm{b}$ is measured based on the IMU's gyrometer, which provides noisy measurements, therefore we added a second order Butterworth low-pass filter with cut-off frequency of 20 Hz to smoothen out the output $\boldsymbol{\omega}_\mathrm{b}$. The Butterworth filter is considered in the linearized dynamics when optimizing the controller gains following Algorithm 3.

We also used the ESCs associated with the performance shown in Figure 3 for the propellers actuation. The two ESCs were flashed with the open-source code available in the GitHub repository AM32-MultiRotor-ESC-firmware[2]. The advantage of this firmware, as compared with the commercial code, is that it exploits a low-level PID feedback of the speed of rotation of the motor, which is calculated at the same speed as the motor phase commutation. We adapted the speed loop code in the firmware, following the approach of [40], featuring

[2]https://github.com/FlorianSan/AM32-MultiRotor-ESC-firmware

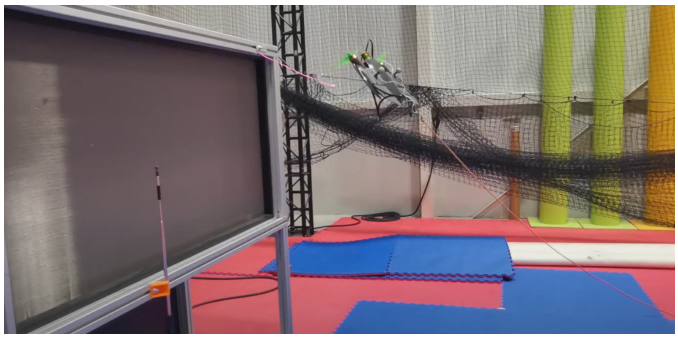

Fig. 10. DarkO's experimental flight in front of the open wind tunnel.

an adaptive bias and adaptive gain algorithm (ABAG). In this way, we compensate the battery discharge effects and obtain an accurate realization of the commanded speed. Before this modification, the integral action of the stabilizing feedback of Fig. 6 compensated for the motor speed loss caused by the battery voltage reduction during flight. This integral compensation was indirectly generated by the altitude loss of the UAV caused by the reduced traction. The advantages of the ABAG solution are high responsiveness and adaptability, as the propeller dimensions can be changed without needing to modify the actuation gains.

We carried out a flight experiment where DarkO was manually put into a stabilized hovering mode in front of the wind tunnel, then we switched on the control law of Algorithm 3. As the drone had to be stabilized at least $30\,\mathrm{cm}$ away from the wind tunnel, a manual command was gradually applied to avoid overshooting, which could damage the wind tunnel. Once DarkO was close enough to the setpoint $\boldsymbol{r}_p$ of Fig. 6, we switched on the proposed controller, obtaining the results in Fig. 11. During the follow-up experimentation phase, as shown in the lower plot of Fig. 11, we stepwise increased the wind speed, waiting 20 seconds between each pair of consecutive steps, up to a final wind speed of $7\,\mathrm{m\,s^{-1}}$.

Figs 11 and 12 show that the drone maintains its position despite the increasing wind speed. We can note a few important points, in agreement with the simulations: the motor traction decreases when increasing the wind speed. The control scheme takes advantage of the lift generated by the wind to support the drone, so that less energy is needed to stabilize the hovering position. The drone maintains its tilt angle at a value that is unknown a priori to the control law and naturally stems from the integral action that asymptotically attains the required value of the drone's pitch angle θ. To stabilize the position, the UAV uses the elevons to cancel the pitch moment generated by the shape of the wing, subjected to a horizontal wind, without reaching the saturation limits. We also notice a slight asymmetry of the effectiveness of the actuators, which is effectively compensated by the proportional action of the control scheme.

V. Conclusions

We presented two models of the DarkO tailsitter UAV, and characterized a set of equilibrium points for different wind

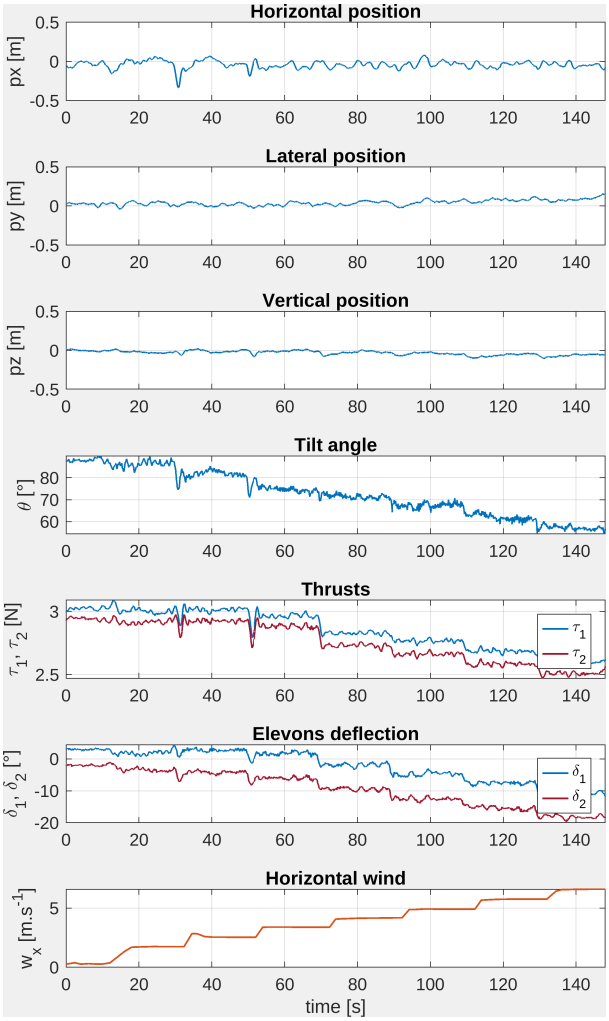

Fig. 11. Experiment of the DarkO UAV in front of the wind tunnel with increasing constant wind levels (lower plot).

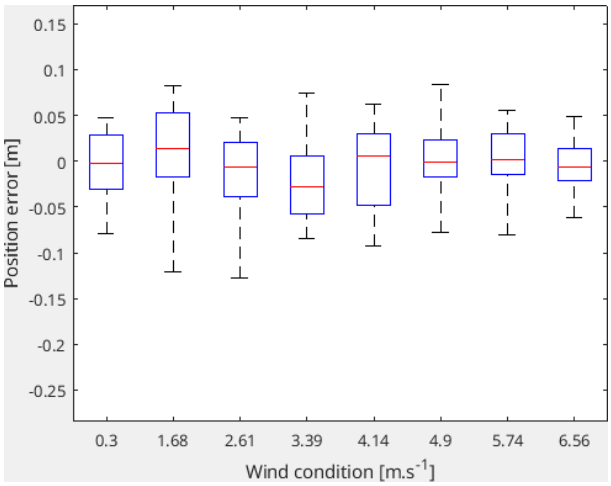

Fig. 12. Statistical visualization of the hovering performance.

conditions. Our analysis showed that all equilibrium points were achievable within the saturation limits of the actuators. For each one of these equilibria, a suitable rotated linearization has been characterized, thereby obtaining a parametric set of models depending on two parameters (horizontal and vertical wind speed). A control scheme was then proposed, for hovering stabilization in the presence of an unknown constant wind, embedding an integral action and not requiring the wind speed measurement. The parametric linearized models were then shown to be a key instrument to perform an optimized tuning of the controller parameters. After studying the simulation results, we carried out an experimental flight campaign in a controlled environment to validate our control solution. Future work will involve outdoor testing in an uncontrolled environment, followed by the development of a controller to stabilize the forward flight.

REFERENCES

[1] H. Stone and K. C. Wong, "Preliminary design of a tandem-wing tail-sitter UAV using multi-disciplinary design optimisation," in *International Aerospace Congress*, 2002.

[2] R. H. Stone, P. Anderson, C. Hutchison, A. Tsai, P. Gibbens, and K. C. Wong, "Flight testing of the t-wing tail-sitter unmanned air vehicle," *Journal of Aircraft*, vol. 45, no. 2, pp. 673–685, 2008.

[3] L. Lustosa, F. Defaÿ, and J. Moschetta, "Longitudinal study of a tilt-body vehicle: modeling, control and stability analysis," in *Proc. of International Conference on Unmanned Aircraft Systems*, Denver, Colorado, US, June 2015, pp. 816–824.

[4] Y. Ke, K. Wang, K. Gong, S. Lai, and B. M. Chen, "Model based robust forward transition control for tail-sitter hybrid unmanned aerial vehicles," in *2017 13th IEEE International Conference on Control and Automation (ICCA)*, 2017, pp. 828–833.

[5] A. S. Saeed, A. B. Younes, C. Cai, and G. Cai, "A survey of hybrid Unmanned Aerial Vehicles," *Progress in Aerospace Sciences*, vol. 98, pp. 91–105, Apr. 2018.

[6] G. J. J. Ducard and M. Allenspach, "Review of designs and flight control techniques of hybrid and convertible VTOL UAVs," *Aerospace Science and Technology*, vol. 118, p. 107035, Nov. 2021.

[7] G. Droandi, A. Zanotti, G. Gibertini, D. Grassi, and G. Campanardi, "Experimental investigation of the rotor-wing aerodynamic interaction in a tiltwing aircraft in hover," *The Aeronautical Journal*, vol. 119, no. 1215, p. 591–612, 2015.

[8] B. M. Simmons and P. C. Murphy, "Aero-propulsive modeling for tilt-wing, distributed propulsion aircraft using wind tunnel data," *Journal of Aircraft*, vol. 59, no. 5, pp. 1162–1178, 2022.

[9] P. Aref, M. Ghoreyshi, A. Jirasek, M. J. Satchell, and K. Bergeron, "Computational study of propeller–wing aerodynamic interaction," *Aerospace*, vol. 5, no. 3, 2018.

[10] L. F. Fernandez, M. Bronz, N. Bartoli, and T. Lefebvre, "Assessment of methods for propeller performance calculation at high incidence angles," in *AIAA SCITECH 2023 Forum*, 2023.

[11] J. Escareno, S. Salazar, and R. Lozano, "Modelling and control of a convertible VTOL aircraft," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 69–74.

[12] J. A. Guerrero, R. Lozano, G. Romero, D. Lara-Alabazares, and K. C. Wong, "Robust control design based on sliding mode control for hover flight of a mini tail-sitter unmanned aerial vehicle," in *2009 35th Annual Conference of IEEE Industrial Electronics*, 2009, pp. 2342–2347.

[13] S. Zhang, Q. Fei, J. Liang, and Q. Geng, "Modeling and control for longitudinal attitude of a twin-rotor tail-sitter unmanned aerial vehicle," in *2017 13th IEEE International Conference on Control and Automation (ICCA)*, 2017, pp. 816–821.

[14] Y. Ke and B. M. Chen, "Full envelope dynamics modeling and simulation for tail-sitter hybrid UAVs," in *2017 36th Chinese Control Conference (CCC)*, 2017, pp. 2242–2247.

[15] J. Escareno, R. Stone, A. Sanchez, and R. Lozano, "Modeling and control strategy for the transition of a convertible tail-sitter UAV," in *European Control Conference*, 2007, pp. 3385–3390.

[16] R. Chiappinelli and M. Nahon, "Modeling and control of a tailsitter UAV," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 400–409.

[17] L. R. Lustosa, F. Defaÿ, and J.-M. Moschetta, "Global singularity-free aerodynamic model for algorithmic flight control of tail sitters," *Journal*

of Guidance, Control, and Dynamics, vol. 42, no. 2, pp. 303–316, Feb. 2019.

[18] E. Tal and S. Karaman, "Global incremental flight control for agile maneuvering of a tailsitter flying wing," *arXiv preprint arXiv:2207.13218*, 2022.

[19] J. Liang, Q. Fei, B. Wang, and Q. Geng, "Tailsitter VTOL flying wing aircraft attitude control," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 439–443.

[20] J. Escareño, R. Stone, A. Sanchez, and R. Lozano, "Modeling and control strategy for the transition of a convertible tail-sitter UAV," in *2007 European Control Conference (ECC)*, 2007, pp. 3385–3390.

[21] D. A. Ta, I. Fantoni, and R. Lozano, "Modeling and control of a convertible mini-UAV," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1492–1497, 2011, 18th IFAC World Congress.

[22] E. Bulka and M. Nahon, "Autonomous control of agile fixed-wing UAVs performing aerobatic maneuvers," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 104–113.

[23] Y. Yang, J. Zhu, X. Zhang, and X. Wang, "Active disturbance rejection control of a flying-wing tailsitter in hover flight," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6390–6396.

[24] S. Verling, B. Weibel, M. Boosfeld, K. Alexis, M. Burri, and R. Siegwart, "Full Attitude Control of a VTOL tailsitter UAV," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3006–3012.

[25] J. Zhong, B. Song, Y. Li, and J. Xuan, "l_1 adaptive control of a dual-rotor tail-sitter unmanned aerial vehicle with input constraints during hover flight," *IEEE Access*, vol. 7, pp. 51 312–51 328, 2019.

[26] J. M. Olszanecki Barth, J.-P. Condomines, M. Bronz, J.-M. Moschetta, C. Join, and M. Fliess, "Model-free control algorithms for micro air vehicles with transitioning flight capabilities," *International Journal of Micro Air Vehicles*, vol. 12, pp. 1–22, Apr. 2020.

[27] E. J. J. Smeur, M. Bronz, and G. C. H. E. de Croon, "Incremental control and guidance of hybrid aircraft applied to the Cyclone tailsitter UAV," *Journal of Guidance, Control, and Dynamics*, Sep. 2019.

[28] N. Silva, J. Fontes, R. Inoue, and K. Branco, "Dynamic inversion and gain-scheduling control for an autonomous aerial vehicle with multiple flight stages," *Journal of Control, Automation and Electrical Systems*, vol. 29, no. 3, pp. 328–339, 2018.

[29] Y. Demitrit, S. Verling, T. Stastny, A. Melzer, and R. Siegwart, "Model-based wind estimation for a hovering VTOL tailsitter UAV," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3945–3952.

[30] X. Lyu, J. Zhou, H. Gu, Z. Li, S. Shen, and F. Zhang, "Disturbance Observer Based Hovering Control of Quadrotor Tail-Sitter VTOL UAVs Using H_∞ Synthesis," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2910–2917, 2018.

[31] G. Hattenberger, M. Bronz, and J.-P. Condomines, "Estimating wind using a quadrotor," *International Journal of Micro Air Vehicles*, vol. 14, p. 17568293211070824, 2022.

[32] F. Sansou, "Commande hybride d'un drone convertible pour des déplacements sous optimaux," *Master Thesis. ENAC Toulouse*, 2022.

[33] F. Sansou and L. Zaccarian, "On local-global hysteresis-based hovering stabilization of the DarkO convertible UAV," in *2022 European Control Conference (ECC)*, 2022, pp. 40–45.

[34] F. Sansou, F. Demourant, G. Hattenberger, T. Loquen, and L. Zaccarian, "Open wind tunnel experiments of the DarkO tail-sitter longitudinal stabilization with constant wind," *IFAC-PapersOnLine*, vol. 55, no. 22, pp. 1–6, 2022, 22nd IFAC Symposium on Automatic Control in Aerospace ACA 2022.

[35] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE, Control Systems*, vol. 33, pp. 61–75, 02 2013.

[36] M. Jardin and E. Mueller, "Optimized measurements of UAV mass moment of inertia with a bifilar pendulum," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2012.

[37] J.-F. Tregouet, D. Arzelier, D. Peaucelle, C. Pittet, and L. Zaccarian, "Reaction Wheels Desaturation Using Magnetorquers and Static Input Allocation," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 525 – 539, Feb. 2015.

[38] P. Apkarian and D. Noll, "Nonsmooth H∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[39] P. Apkarian, "Tuning controllers against multiple design requirements," in *2013 American Control Conference*, 2013, pp. 3888–3893.

[40] A. Franchi and A. Mallet, "Adaptive Closed-loop Speed Control of BLDC Motors with Applications to Multi-rotor Aerial Vehicles," in *IEEE International Conference on Robotics and Automation*, Singapour, Singapore, May 2017.