

Control of vertical take off, dynamic flight and landing of hybrid drones for airborne wind energy systems

Davide Todeschini^{*}, Lorenzo Fagiano^{*}, Claudio Micheli[†] and Aldo Cattano[†]

Abstract

A control design approach for a hybrid multi-copter/box-wing drone is presented. The drone is designed to be used in an airborne wind energy system, where tethered aircrafts are used to convert wind energy into electricity. It features four propellers and multiple aerodynamic control surfaces, and can operate either as multi-copter or as airplane. An untethered system is considered in this work, and the goal is to achieve fully autonomous take-off, transition to dynamic flight, and landing. A model-based, hierarchical feedback controller is proposed, with linear inner control loops to stabilize the drone's attitude, and an outer nonlinear loop to obtain the desired flight trajectory. A switching strategy is employed to transition from hovering mode (i.e. multi-copter) to dynamic flight mode (i.e. airplane), and vice-versa. Simulation results with a realistic system model indicate that the controller can achieve good performance and robustness in all flight conditions, notwithstanding its simplicity and ease of implementation.

I. INTRODUCTION

Airborne Wind Energy Systems (AWES) employ tethered aircrafts to convert wind energy into electricity, according to one of several possible concepts under development, see e.g. [1], [2], [3]. The main advantages of AWES are low predicted capital costs, thanks to the absence of bulky and expensive components and to the relatively simple system construction, and the possibility to reach high altitudes (300-600 m above ground for most concepts), where wind speed is generally stronger and less variable. The main disadvantage is the high complexity of these systems, that rely completely on active control and automation to operate safely and reliably. Modeling, identification and control aspects have been investigated quite extensively in the literature for systems employing soft wings [4], [5], [6], [7], [8], [9] and, to a smaller extent, rigid ones [10], [11], [12], [13]. In most contributions, the power generation phase is considered, where the aircraft flies fast in so-called crosswind conditions and the tether is taut. Fewer contributions consider instead the take-off and/or landing phases, which are however crucial for the operation of the final system [14], [15], [11]. In this respect, systems employing an aircraft with several onboard actuators and vertical take-off and landing (VTOL) capabilities are being considered by several developers, because of their higher control authority in presence of little wind or strong turbulence. The prototype studied in this paper is a hybrid one; it can both hover and fly like a plane. For take-off and landing phases, it uses multiple rotors to hover ("hovering mode"). For the power generation phase, it exploits aerodynamic forces generated by the wings to fly like a conventional plane ("dynamic flight mode"). A crucial aspect from the point of view of control design is how to transition from one mode to the other, and vice-versa.

The original contribution of this paper with respect to the mentioned literature is to propose a control design approach for one of such hybrid drones, in particular the one developed by Skypull SA [16], see Fig. 1. The drone features a box-wing design with a propeller at each corner. At this stage of our research, we consider the untethered system, and the control objective is to complete autonomous cycles of vertical take-off, horizontal flight, and vertical landing, with two transitions, i.e. from hovering to dynamic flight and back. We adopt a hierarchical control strategy to solve this problem, and test it with a realistic nonlinear model of the system at hand. Experimental implementation and testing of the approach is currently ongoing. This step is instrumental to develop a control strategy for fully automated tethered operation. In the Unmanned Aerial Vehicle (UAV) control literature, the problem of designing control systems for conventional hybrid VTOL aerial vehicles (e.g. tailsitters) has been already considered, see [17], [18]. Thanks to its box-wing design, the UAV considered is different from those studied so far in the literature, since it can steer in dynamic flight with yaw motion instead of roll, exploiting the lift of the lateral wings. This aspect introduces a further element of novelty in our work.

II. SYSTEM DESCRIPTION AND CONTROL OBJECTIVES

We consider an UAV with a box-wing architecture and a propeller at each corner, see Figs. 1,2. Due to the distribution of masses and the positioning of batteries in the lower corners of the frame, the geometric center and the center of gravity are different (see Fig. 2). This allows one to identify a lower wing, the one that is closer to the center of gravity, and an

^{*} Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy. E-mail: {davide.todeschini | lorenzo.fagiano}@polimi.it

[†] Skypull SA, Via alla Stampa 49, 6967 Lugano, Switzerland. E-mail: {claudio.micheli | aldo.cattano}@skypull.com

Corresponding author: Davide Todeschini

This is the preprint of paper D. Todeschini, L. Fagiano, C. Micheli, A. Cattano, "Control of vertical take off, dynamic flight and landing of hybrid drones for airborne wind energy systems", accepted for publication in the Proceedings of the 2019 American Control Conference, Philadelphia, USA, July 10-12 2019. © 2019 IEEE



Fig. 1. Example of prototypes of the considered hybrid drones during untethered, manually piloted tests in the Swiss Alps.

upper one (i.e. the opposite one), and consequently a left and right wing by looking at the drone from behind. On each lateral wing there are two discrete control surfaces, while the upper and lower wings feature three discrete control surfaces each. These surfaces can be actuated either symmetrically, on a single wing or on two opposite wings, leading to a symmetric change of aerodynamic coefficients (hence of lift and drag forces), or in opposite directions. In the second case, the effect is to generate turning moments. All the possible combinations of movements of control surfaces lead to a rather high control authority when operating in dynamic flight mode. In addition, the four propellers act as in standard quad-copters in hovering mode, and they can further contribute to turning moments also in dynamic flight mode, besides providing the forward thrust to keep cruise velocity. In this paper, we assume that the control system can inject turning moments around the three body axes x_B, y_B, z_B depicted in Fig. 2, plus a thrust force in the z_B direction. Therefore, we have four control inputs (three turning moments and the thrust force). The translation from turning moments to individual surface/propeller commands is managed by a low-level logic that aims to provide the wanted action with minimal energy consumption. When operating in tethered mode in presence of strong-enough wind, the propellers are not used to generate thrust force, rather the forward apparent speed is provided by the well-known phenomena of crosswind kite power [19],[3].

The UAV operates in the following main modes:

- **Hovering.** It is the typical flight condition of quad-rotors. The propellers sustain the weight of the vehicle and control the attitude.
- **Dynamic flight.** It is the typical flight condition of an airplane. The lift generated by wings sustains the UAV. In this condition, the propellers are used mainly to achieve a desired speed, while the attitude is regulated with the aerodynamic control surfaces. The cruise velocity of the drone considered here is around 16 m/s, higher speed can be used to ascend, and lower one to descend.
- **Transition from hovering to flight.** In this phase, the UAV must quickly speed up and rotate to achieve the attitude and reference speed that can sustain its weight during the flight mode.
- **Transition from flight to hovering.** In this phase, the UAV must slow down and rotate with propellers pointing up to achieve a stationary condition.

The main overall control objective is to obtain a fully autonomous flight cycle from take-off to landing, going through a dynamic flight phase via suitable transitions. Being capable to carry out such a task is instrumental to achieve fully autonomous operation with the tether attached and in presence of wind, so to enable energy generation according to a pumping principle [3], [20]. Specifically, the control system shall start the UAV from ground and take-off in hovering mode, reach a desired position and attitude and start the transition to dynamic flight. In this phase, the controller must properly track a periodic reference path (we consider a figure-of-eight pattern here, since it is the one also adopted in tethered flight

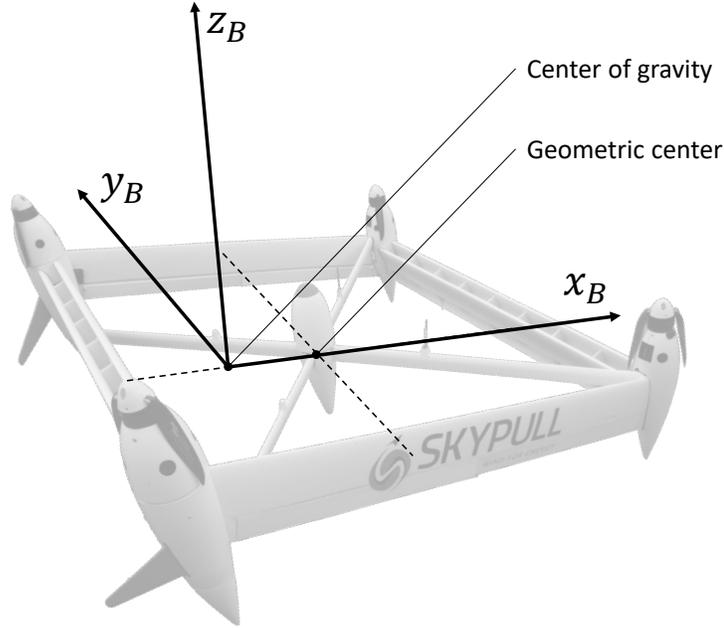


Fig. 2. Position of the center of gravity and geometric center of the drone, and adopted body reference frame B .

for power generation). After completing a few periodic cycles, the control system shall transition from dynamic flight to hovering again, managing the speed decrease and attitude change, and then reach the initial position. All these maneuvers have to be carried out safely and with good precision even in presence of wind disturbances.

We present next a nonlinear dynamical model featuring six degrees of freedom (dof) for the drone, whose aerodynamic coefficients have been identified by a combination of Computational Fluid Dynamics (CFD) simulations and experimental tests with manual operation. We employ this model as testing platform in this paper, to study performance and robustness of the control systems both in nominal conditions and in perturbed ones, in particular in presence of wind of varying intensity. Experimental tests are ongoing at the time of submission of this manuscript.

III. SYSTEM MODEL

The considered model is derived from traditional 6-dof aircraft model equations [21], with modifications due to the architecture of the UAV and the presence of the four propellers. In particular, the various application points of the aerodynamic force and moment vectors generated by the four wings have been taken into account. Moreover, the angle of attack of one wing corresponds to the side-slip angle for the perpendicular one, thus resulting in a rather complex computation of the overall forces and moments acting at each instant on the drone. As a matter of fact, the overall aerodynamic force and moment vectors are nonlinear mappings whose inputs are the UAV angle of attack and side-slip, the apparent wind speed vector, and the actuators' position. Such nonlinear functions are included in the model but not explicitly provided here for confidentiality reasons.

We introduce three right-handed reference frames:

- $F (x_F, y_F, z_F)$ is the inertial reference frame, fixed to a point on ground, with component z_F pointing up and x_F along a chosen direction (usually the prevalent wind direction at the considered location).
- $B (x_B, y_B, z_B)$ is the body reference frame, fixed to the UAV and with origin in its center of gravity (Fig. 2). The z_B axis is aligned with propellers' axes, while x_b points towards the upper wing. This reference system is aligned with the inertial one, when the UAV is stationary in hovering mode.
- $A_w (x_{A_w}, y_{A_w}, z_{A_w})$ is the apparent wind reference system, with x_{A_w} aligned with the apparent wind direction, y_{A_w} parallel to the upper wing, and z_{A_w} pointing up.

The relative orientation between frames F and B can be expressed by a quaternion $\vec{q}(t) = [q_1 \ q_2 \ q_3 \ q_4]^T$, where T denotes the matrix transpose operator and $t \in \mathbb{R}$ is the continuous time variable. For the sake of notational simplicity, in the remainder we omit the explicit dependency of the various quantities on t when this is clear from the context, and we point out the constant (i.e. not time-dependent) quantities and parameters.

A vector in the F system can be expressed in the B system by means of the following rotation matrix:

$$H_{BI} = 2 \begin{bmatrix} (q_1^2 + q_2^2) - 1 & (q_2q_3 + q_1q_4) & (q_2q_4 - q_1q_3) \\ (q_2q_3 - q_1q_4) & (q_1^2 + q_3^2) - 1 & (q_3q_4 + q_1q_2) \\ (q_2q_4 + q_1q_3) & (q_3q_4 - q_1q_2) & (q_1^2 + q_4^2) - 1 \end{bmatrix} \quad (1)$$

In the same way, a vector expressed in A_w reference can be expressed in B system by the following rotation matrix:

$$H_{WB} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ \sin \alpha \sin \beta & \cos \beta & \cos \alpha \sin \beta \\ \sin \alpha \cos \beta & -\sin \beta & \cos \alpha \cos \beta \end{bmatrix} \quad (2)$$

where α is the drone's angle of attack and β is the side-slip angle [21]. Due to the unbalance in mass distribution mentioned in Section II, the constant inertia matrix computed with respect to the considered body reference frame has non-zero terms out of the diagonal:

$$I = \begin{bmatrix} I_{xx} & 0 & I_{zx} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix} \quad (3)$$

The external forces and moments acting on the UAV considered in the model are:

- gravitational force
- propellers' forces and moments
- aerodynamic force and moments

The gravitational force can be defined in inertial frame as $\vec{F}_{g(F)} = m [0 \ 0 \ g]^T$, where the constants m and g are, respectively, the drone's mass and the gravity acceleration.

Each propeller generates a thrust force, $\vec{F}_{p(B)j}$, and a drag torque in z_B direction, $T_{p(B)j}$, $j = 1 \dots, 4$, expressed in body frame as:

$$\begin{aligned} \vec{F}_{p(B)j} &= [0 \ 0 \ b_j \omega_j^2]^T \\ T_{p(B)j} &= [0 \ 0 \ c_j \omega_j^2]^T \end{aligned} \quad (4)$$

Where ω_j is the rotational speed the j^{th} propeller, and b_j, c_j are constant parameters. The propellers' forces can be linearly combined to obtain the total thrust in z_B direction, denoted by U_1 , and rotational moments around x_B (ΔU_{p2}), y_B (ΔU_{p3}) and z_B (ΔU_{p4}):

$$\begin{bmatrix} U_1 \\ \Delta U_{p2} \\ \Delta U_{p3} \\ \Delta U_{p4} \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_1 d_{xB1} & -b_2 d_{xB2} & -b_3 d_{xB3} & b_4 d_{xB4} \\ b_1 d_{yB1} & -b_2 d_{yB1} & b_3 d_{yB3} & -b_4 d_{yB4} \\ -c_1 & -c_2 & c_3 & c_4 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (5)$$

where d_{xBi}, d_{yBi} , $i = 1, \dots, 4$ are the constant position coordinates of each propeller with respect to the origin of B .

The box-wing structure generates aerodynamic forces, which in apparent wind reference system (A_w) can be expressed as:

$$\vec{F}_{aero(A_w)} = [F_D \ F_S \ F_L]^T \quad (6)$$

where F_D, F_S, F_L are the drag component, lateral component, and lift component, respectively. These are computed as:

$$\begin{aligned} F_L &= \frac{1}{2} \rho S c_L(\alpha, \beta) W_a^2 \\ F_D &= \frac{1}{2} \rho S c_D(\alpha, \beta) W_a^2 \\ F_S &= \frac{1}{2} \rho S c_S(\alpha, \beta) W_a^2 \end{aligned} \quad (7)$$

Where ρ is the air density, S is the wing area (both constant parameters), W_a is the magnitude of the apparent wind seen by the UAV, c_L, c_D and c_S are respectively the lift, drag and side forces' coefficients, which depend on (α, β) . As mentioned above, these lumped parameters take into account the specific geometry of the wing. Similarly, the overall aerodynamic moment is denoted as \vec{M}_{aero} and it is computed in frame A_w as a quadratic function of W_a and a nonlinear function of (α, β) , featuring suitable constant aerodynamic coefficients. The details are omitted here for brevity. The apparent wind vector \vec{W}_a can be computed in the inertial frame as:

$$\vec{W}_a = \vec{W}_{(F)} - \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{z}_F \end{bmatrix}$$

where $\vec{W}_{(F)}$ is the absolute wind vector expressed in the inertial frame. The system model consists of 13 ordinary differential equations (ODEs):

$$\begin{aligned}
F_{x_B} &= m(\dot{U} - RV + QW) \\
F_{y_B} &= m(\dot{V} - PW + UR) \\
F_{z_B} &= m(\dot{W} - QU + PV) \\
M_{x_B} &= \dot{P}I_{xx} - (\dot{R} + PQ)I_{zx} + RQ(I_{zz} - I_{yy}) + Q\sum_{j=1}^4 h_{z,j} \\
M_{y_B} &= \dot{Q}I_{yy} + (R^2 + P^2)I_{zx} + PR(I_{xx} - I_{zz}) - P\sum_{j=1}^4 h_{z,j} \\
M_{z_B} &= \dot{R}I_{zz} - (\dot{P} + QR)I_{zx} + PQ(I_{yy} - I_{xx})
\end{aligned}
\tag{8}$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{z} \end{bmatrix} = H_{BI} \begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

Where: U , V and W are the drone's velocity vector components in the body frame B ; P , Q and R are the rotational speeds around the axes of B ; F_{x_B} , F_{y_B} and F_{z_B} are the components in the B frame of the vector sum of all external forces acting on the UAV; M_{x_B} , M_{y_B} and M_{z_B} are the components in the B frame of the vector sum of all external moments applied to the UAV. Finally, the constant parameters $h_{z,j}$, $j = 1, \dots, 4$ are the moments of inertia of motors/propellers in z_B direction. The control inputs are the total thrust force, U_1 , and the total moments provided by the propellers and by the aerodynamic control surfaces, denoted as U_2 , U_3 and U_4 for rotations around axes x_B , y_B , and z_B , respectively. These variables do not appear explicitly in (8), since they contribute, respectively, to F_{z_B} , M_{x_B} , M_{y_B} and M_{z_B} . In particular, the turning moments are the sum of ΔU_{p2} , ΔU_{p3} , ΔU_{p4} provided by the propellers (see (5)), exploited mainly in hovering mode, with the contributions of the aerodynamic surfaces, used prevalently in dynamic flight mode. As mentioned above, the actual control allocation among discrete aerodynamic surfaces is carried out by a low level control allocation scheme, not treated in this paper.

As shown in the following, we employed simplified physical equations to design our control strategy, which we then tested on the full nonlinear model (8). The latter captures rather accurately all the relevant effects occurring in the real device.

IV. CONTROL DESIGN

A. Euler angles for feedback control

To formulate the feedback control algorithms in an intuitive way, we resort to two definitions of Euler angles, one for the hovering mode and one for the dynamic flight. These angles are defined as functions of the elements of the rotation matrix H_{BI} introduced in (1):

- Euler angles for hovering mode $(\varphi_h, \theta_h, \psi_h)$, defined as $Z - Y - X$ right handed rotation sequence from the inertial frame F to body frame B :

$$\begin{bmatrix} \varphi_h \\ \theta_h \\ \psi_h \end{bmatrix} = \begin{bmatrix} \arctan \frac{H_{BI}(2,3)}{H_{BI}(3,3)} \\ \arcsin -H_{BI}(1,3) \\ \arctan \frac{H_{BI}(1,2)}{H_{BI}(1,1)} \end{bmatrix}
\tag{9}$$

- Euler angles for dynamic flight mode $(\varphi_f, \theta_f, \psi_f)$, defined as $X - Y - Z$ right handed rotation sequence from F to B . With this convention, when the drone flies in dynamic mode at zero pitch and roll the Z_B axis is parallel to the ground and X_B axis points up.

$$\begin{bmatrix} \varphi_f \\ \theta_f \\ \psi_f \end{bmatrix} = \begin{bmatrix} \arctan \frac{H_{BI}(3,2)}{H_{BI}(3,1)} \\ \arcsin -H_{BI}(3,3) \\ \arctan \frac{H_{BI}(2,3)}{H_{BI}(1,3)} \end{bmatrix}
\tag{10}$$

In practice, the set of angles for hovering mode use a typical quad-copter convention, while the set of angles for dynamic flight mode uses an aircraft convention. Depending on the orientation of the UAV, the controller will employ one or the other set to track its attitude. The use of two different sets of Euler angles also allows us to avoid the well-known gimbal lock problem, occurring at pitch angle $\theta = \frac{\pi}{2}$. Indeed, such a pitch value is normally reached in either one of the two Euler triplets introduced above in the various flight modes, but never in both triplets at the same time.

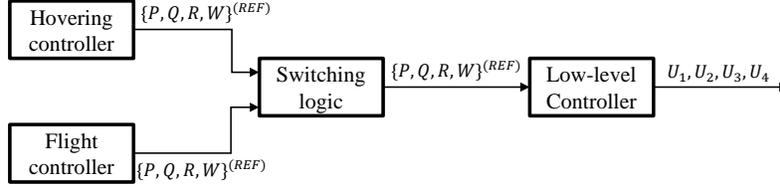


Fig. 3. Control scheme

B. Controller's structure

We adopt the hierarchical structure shown in Fig. 3, with a common low-level attitude controller and different high-level controllers for the various operating modes. In Section IV-C we present the low-level controller, followed by the high-level ones for hovering and dynamic flight (Sections IV-D and IV-E, respectively), and the one for transitions' management (Section IV-F).

We assume that the full system state is available for feedback control. This is reasonable, since the drone is equipped with inertial measurement units integrating GPS to measure absolute position, velocity, attitude and angular rates. Finally, all the gains of the feedback controllers in the remainder are computed via pole-placement, unless otherwise noted.

C. Low level attitude controller

To design the low-level controller, we consider the three simplified equations of the rotational motion in (8). Neglecting gyroscopic effects and all external moments except those induced by the control inputs U_2, U_3 and U_4 , and solving the system for the derivatives of angular speed, we obtain:

$$\begin{aligned}\dot{P} &= \frac{U_2 I_{zz} + U_4 I_{zx}}{I_{xx} I_{zz} - I_{zx}^2} \\ \dot{Q} &= \frac{U_3}{I_{yy}} \\ \dot{R} &= \frac{U_4 I_{xx} + U_2 I_{zx}}{I_{xx} I_{zz} - I_{zx}^2}\end{aligned}\quad (11)$$

We introduce two fictitious inputs to decouple the dynamics:

$$\begin{aligned}U_P &= U_2 I_{zz} + U_4 I_{zx} \\ U_R &= U_4 I_{xx} + U_2 I_{zx}\end{aligned}\quad (12)$$

and then we design the controller based on the following system:

$$\begin{aligned}\dot{P} &= \frac{U_P}{I_{xx} I_{zz} - I_{zx}^2} \\ \dot{Q} &= \frac{U_3}{I_{yy}} \\ \dot{R} &= \frac{U_R}{I_{xx} I_{zz} - I_{zx}^2}\end{aligned}\quad (13)$$

The attitude controller takes the form:

$$\begin{bmatrix} U_P \\ U_2 \\ U_R \end{bmatrix}^{(ref)} = \begin{bmatrix} k_P & 0 & 0 \\ 0 & k_Q & 0 \\ 0 & 0 & k_R \end{bmatrix} \begin{bmatrix} P^{(ref)} - P \\ Q^{(ref)} - Q \\ R^{(ref)} - R \end{bmatrix}\quad (14)$$

with constant gains k_P, k_Q, k_R . The reference angular rates $P^{(ref)}, Q^{(ref)}$ and $R^{(ref)}$ are computed by the outer control loops described in the next sections, see Fig. 3.

After computing the fictitious inputs, we derive the actual ones as:

$$\begin{bmatrix} U_2 \\ U_4 \end{bmatrix}^{(ref)} = \begin{bmatrix} I_{zz} & I_{zx} \\ I_{zx} & I_{xx} \end{bmatrix}^{-1} \begin{bmatrix} U_P \\ U_R \end{bmatrix}^{(ref)}\quad (15)$$

D. Hovering controller

The hovering controller computes the reference rotation rates sent to the attitude controller when the drone is in hovering condition. This controller is based again on a hierarchical approach: the inner loop tracks the reference attitude and altitude, a mid level controller tracks the reference speed in x_F and y_F , and an outer loop tracks the position of the drone in the (x_F, y_F) plane.

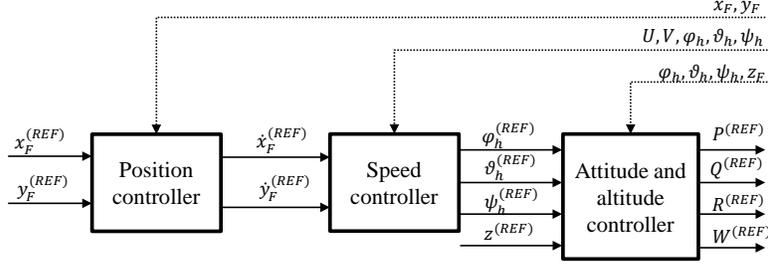


Fig. 4. Hovering control scheme

The attitude controller exploits the hovering Euler angles as feedback variables:

$$\begin{aligned}\dot{\phi}_h^{(ref)} &= k_\phi(\phi_h^{(ref)} - \phi_h) \\ \dot{\theta}_h^{(ref)} &= k_\theta(\theta_h^{(ref)} - \theta_h) \\ \dot{\psi}_h^{(ref)} &= k_\psi(\psi_h^{(ref)} - \psi_h)\end{aligned}\quad (16)$$

with constant gains k_ϕ, k_θ, k_ψ . The nonlinear link between Euler angular rates and rotational rates around the axes of frame B is taken into account by the following matrix, which transforms the reference commands of the hovering controller into reference inputs for the low level controller described in Section IV-C:

$$\begin{bmatrix} P^{(ref)} \\ Q^{(ref)} \\ R^{(ref)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta_h \\ 0 & \cos \phi_h & \cos \theta_h \sin \phi_h \\ 0 & -\sin \phi_h & \cos \theta_h \cos \phi_h \end{bmatrix} \begin{bmatrix} \dot{\phi}_h^{(ref)} \\ \dot{\theta}_h^{(ref)} \\ \dot{\psi}_h^{(ref)} \end{bmatrix}\quad (17)$$

The speed controller is designed based on simplified equations, valid under a small angle approximation, linking the acceleration components \dot{U}, \dot{V} to the roll and pitch angles, via the thrust force U_1 :

$$\begin{aligned}\dot{U} &= \frac{U_1}{m} \theta_h \\ \dot{V} &= \frac{U_1}{m} \phi_h\end{aligned}\quad (18)$$

Here, the input U_1 is considered to be roughly constant, this is a reasonable approximation when hovering at constant (possibly zero) vertical speed. The speed controller is thus given by:

$$\begin{aligned}\theta_h^{(ref)} &= k_U(U^{(ref)} - U) \\ \phi_h^{(ref)} &= k_V(V^{(ref)} - V)\end{aligned}\quad (19)$$

with constant gains k_U, k_V .

The planar position controller is defined in the inertial frame F :

$$\begin{aligned}\dot{x}_F^{(ref)} &= k_x(x_F^{(ref)} - x_F) \\ \dot{y}_F^{(ref)} &= k_y(y_F^{(ref)} - y_F)\end{aligned},\quad (20)$$

with constant gains k_x, k_y . Once the reference speed components $\dot{x}_F^{(ref)}, \dot{y}_F^{(ref)}$ in the inertial frame are computed, they have to be rotated to the body reference frame, to make them compatible with the mid-level controller:

$$\begin{bmatrix} U^{(ref)} \\ V^{(ref)} \end{bmatrix} = \begin{bmatrix} \cos \psi_h & \sin \psi_h \\ -\sin \psi_h & \cos \psi_h \end{bmatrix} \begin{bmatrix} \dot{x}_F^{(ref)} \\ \dot{y}_F^{(ref)} \end{bmatrix}\quad (21)$$

Finally, the elevation over ground is regulated by an altitude controller. The latter is designed exploiting the simplified equation $\ddot{z}_F = \frac{U_1}{m} - g$, which is valid in hovering mode when the z_B axis is roughly aligned with the inertial z_F axis:

$$\begin{aligned}W^{(ref)} &= k_z(z_F^{(ref)} - z_F) \\ U_1 &= k_W(W^{(ref)} - W) + mg\end{aligned}\quad (22)$$

with constant gains k_z, k_W . The pole-placement computation of each feedback controller has been carried out according to the desired time constant and damping of each control loop, and using faster time constants for the inner loops.

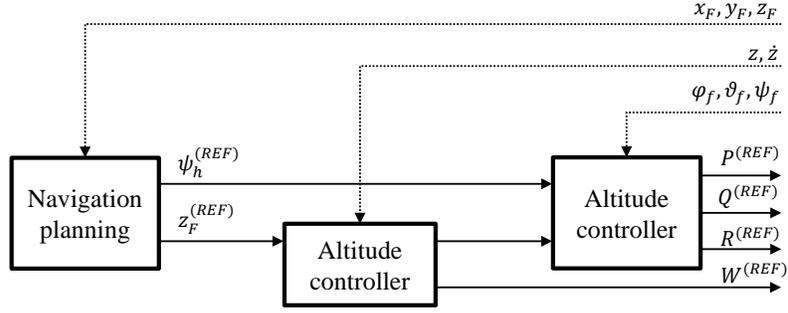


Fig. 5. Dynamic flight control scheme

E. Flight controller

The controller for dynamic flight features a hierarchical logic, with an inner loop responsible for attitude tracking, a middle loop responsible for altitude tracking, and an outer loop for inertial navigation planning. This controller is designed using the Euler angles for dynamic flight, $(\varphi_f, \theta_f, \psi_f)$. Due to the system layout and intended use of the UAV in an airborne wind energy system, the steering mechanism is a yawing motion instead of a roll one, commonly used in traditional airplanes. The controller's equations for the inner loop read:

$$\begin{aligned}\dot{\varphi}_f^{(ref)} &= k_\varphi(\varphi_f^{(ref)} - \varphi_f) \\ \dot{\theta}_f^{(ref)} &= k_\theta(\theta_f^{(ref)} - \theta_f) \\ \dot{\psi}_f^{(ref)} &= k_\psi(\psi_f^{(ref)} - \psi_f)\end{aligned}\quad (23)$$

where the reference roll angle $(\varphi_f^{(ref)})$ is set to zero, while the reference pitch angle $(\theta_f^{(ref)})$ and yaw angle $(\psi_f^{(ref)})$ are computed, respectively, by the altitude controller and the navigation planner, described below. Once the reference Euler angular rates are computed, again a non-linear transformation has to be applied in order to make them compatible with the inputs of the low-level controller described in Section IV-C:

$$\begin{bmatrix} P^{(ref)} \\ Q^{(ref)} \\ R^{(ref)} \end{bmatrix} = \begin{bmatrix} 0 & -\sin \varphi_f & \cos \theta_f \cos \varphi_f \\ 0 & \cos \varphi_f & -\cos \theta_f \sin \varphi_f \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_f^{(ref)} \\ \dot{\theta}_f^{(ref)} \\ \dot{\psi}_f^{(ref)} \end{bmatrix}\quad (24)$$

The altitude controller exploits two different inputs to track the desired altitude: the pitch angle and the front speed. As a first step, a stable horizontal flight condition is computed, and the system is linearized around it. Let $\bar{\theta}_f, \bar{W}_f$ be the constant values of pitch and speed along z_B -axis when flying in such a condition (remember that in dynamic flight, the z_B -axis is roughly parallel to ground when flying horizontally), and let \bar{U}_1 be the corresponding value of propellers' thrust at the equilibrium. Then, the altitude controller is designed based on the linearized equation (8) with LQR method:

$$\begin{bmatrix} \delta \theta_f^{(ref)} \\ \delta W^{(ref)} \end{bmatrix} = K_{LQR} \begin{bmatrix} z_F^{(ref)} - z \\ \dot{z}_F^{(ref)} - \dot{z}_F \end{bmatrix}\quad (25)$$

The value $\theta_f^{(ref)} = \bar{\theta}_f + \delta \theta_f^{(ref)}$ is then fed to the inner loop (23), while the value $\delta W^{(ref)}$ is used to compute the control input U_1 :

$$\begin{aligned}W^{(ref)} &= \bar{W}_f + \delta W^{(ref)} \\ U_1 &= k_{W_f}(W^{(ref)} - W)\end{aligned}\quad (26)$$

Finally, the navigation planner is designed to obtain a figure-eight pattern parallel to ground, exploiting a switching strategy among target points with planar coordinates $(\tilde{x}_F^{(ref)}, \tilde{y}_F^{(ref)})$, as done e.g. in [22], [11]. It thus provides a constant value of $z_F^{(ref)}$ as reference to the altitude controller. Regarding the reference yaw angle $\psi_f^{(ref)}$, this is computed on the basis of the currently active target point:

$$\psi_f^{(ref)} = \begin{cases} \text{atan}_2(y_F - \tilde{y}_F^{(ref)}, x_F - \tilde{x}_F^{(ref)}) & \text{if } \psi_f \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \text{atan}_2(\tilde{y}_F^{(ref)} - y_F, \tilde{x}_F^{(ref)} - x_F) & \text{otherwise} \end{cases}\quad (27)$$

In practice, this formula corresponds to setting as reference course the segment connecting the current position to that of the active target point and guarantees continuity of the reference signal when crossing the values $|\psi_f| = \pi/2$. The yaw

references are filtered with a low pass filter with cut off frequency f_c that has to be tuned to perform smooth turns. To obtain the wanted patterns, a periodic sequence of target points is defined by the control designer, and a switching condition based on proximity of the drone to the active point is adopted, see e.g. [22].

F. Transition management

The system must carry out two different transitions, from hovering to flight and back. The first one starts with constant references to the inner loop of the hovering controller:

$$\begin{bmatrix} \varphi_h^{(ref)} \\ \theta_h^{(ref)} \\ \psi_h^{(ref)} \\ W^{(ref)} \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{\pi}{2} \\ \tilde{\psi} \\ \tilde{W} \end{bmatrix}$$

where $\tilde{\psi}$ is chosen according to the transition direction and \tilde{W} is the front speed value in stable flight. When $\theta_h \leq \underline{\theta}_h$ the controller is switched from hovering to flight to track a given path. The value of $\underline{\theta}_h$ is a tuning parameter chosen by the control designer, corresponding typically to an intermediate pitch configuration between hovering and dynamic flight. The second transition is more critical than the first one, because of the high initial speed of the drone as initial condition. The transition starts again with constant references issued to the inner loop of flight controller:

$$\begin{bmatrix} \varphi_f^{(ref)} \\ \theta_f^{(ref)} \\ \psi_f^{(ref)} \\ W^{(ref)} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \\ 0 \end{bmatrix}$$

Under these reference inputs, the UAV starts to pitch up and when $\theta_f \geq \bar{\theta}_f$, the outer loop controller is switched to hovering mode. In this phase, the position loop of the controller is disabled for a short time period during which the controller stabilizes the drone attitude and speed, without tracking a reference position. When the drone reaches a steady hovering state, the position controller is enabled again, and the drone reaches the reference position.

Finally, as a technical implementation aspect, the switching between all the different outer loop controllers described so far is smoothed by the following exponential law:

$$R_{Low-level} = R_{new} + (\hat{R}_{old} - \hat{R}_{new})e^{-(t-\hat{t})/\tau}$$

where $R_{Low-level}$ is a generic reference variable given to the low-level controller, R_{new} is the reference issued by the the new controller selected, \hat{R}_{new} and \hat{R}_{old} are the new and the old reference commands at the switching instant \hat{t} , finally the time constant τ is a tuning parameter. In this way, the reference discontinuities occurring at the switching instants are filtered, but such a filtering action does not affect the reaction times of the controller when remaining in the same operating mode.

V. SIMULATIONS RESULTS

For the simulations, a drone with 11 kg of mass and 1.17 m of wingspan has been taken into account. The controller operates at 100 Hz. Attitude controllers have been tuned to have 20-Hz-bandwidth in close loop, speed controllers 1 Hz, and position ones 0.2 Hz. Furthermore, we employed the values $\underline{\theta}_h = -\frac{\pi}{3}$, $\bar{\theta}_f = \frac{\pi}{3}$ in the switching conditions to transition from hovering to flight and vice-versa. Full flight simulations are reported, from take-off to landing. Fig. 6 and 7 present the different flight phases and the courses of the relevant quantities. Fig. 7 also highlights the two considered sets of Euler angles and their use in the different phases: in hovering phase and in the transition to dynamic flight, Euler angles in hovering mode have been adopted, while in the other cases the controller adopts the flight convention. Fig. 8 shows the speed of the drone along z_B axis: when hovering, it initially increases to track the reference altitude and converges to zero when approaching such an altitude. Then, during the transition to dynamic flight, the speed rapidly increases to achieve sufficient aerodynamic lift. In the flight phase, the speed is maintained roughly constant. During the transition from flight to hovering, the speed along z_B decreases rapidly as the aircraft pitches up, in order to approach a stable hovering condition. When this condition is reached, the drone starts to descend with a saturated negative speed, until the reference arrival point is reached. Fig. 8 shows also that altitude is properly tracked both in hovering and in flight mode, and that the phase transitions are smooth. To test the robustness of the controller, simulations have been carried out with 3-m/s wind gusts in all directions. An example of such tests is presented in Fig. 9, showing that the controller properly tracks the reference path notwithstanding the disturbance. Finally, Fig. 10 shows the planar path comparison between simulations with and without wind: the path is slightly different but satisfactory in both cases.

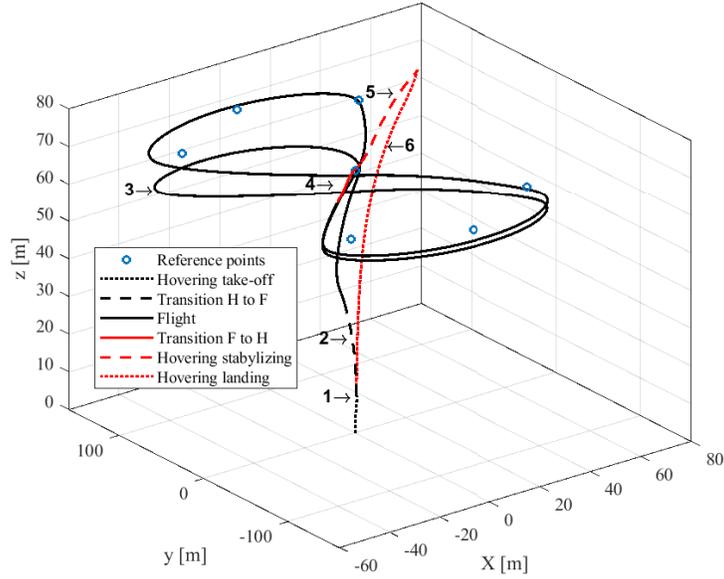


Fig. 6. UAV complete path: 1-Hovering take-off; 2-Transition from hovering to dynamic flight; 3-Dynamic flight mode; 4-Transition from dynamic flight to hovering; 5-Hovering stabilizer phase; 6-Hovering landing. Target points used by the navigation loop are shown as 'o'.

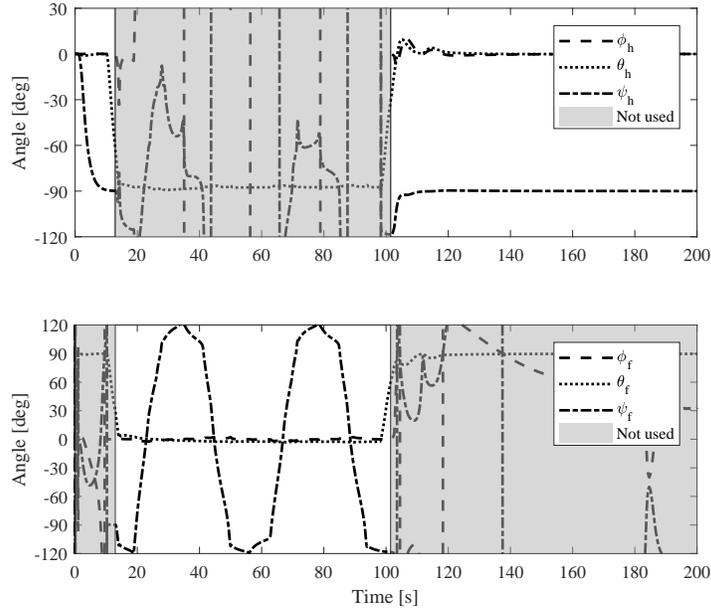


Fig. 7. Euler angles during the simulation. Upper plot: Hovering Euler angles. Lower plot: dynamic flight Euler angles. The shaded areas indicate that those variables are not currently being used for feedback control.

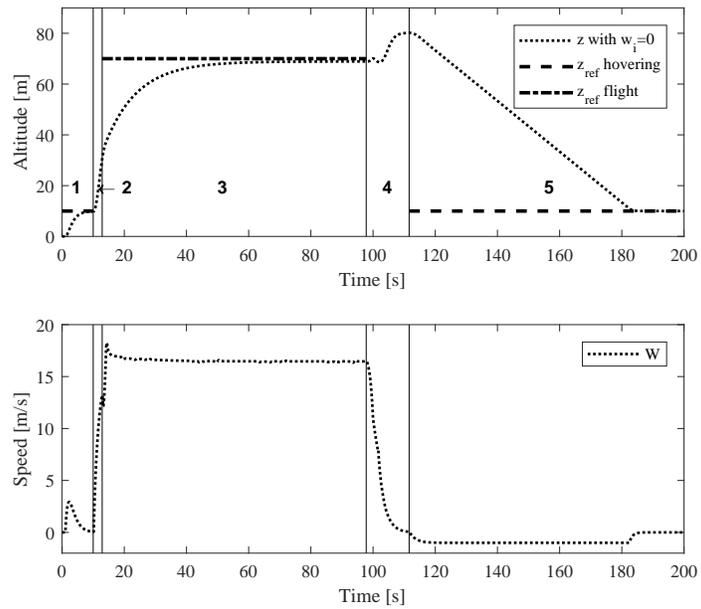


Fig. 8. On top: reference and actual altitude z_F during a simulated cycle from take-off to landing. Phase 1 and 5 hovering condition; phase 3 refers to dynamic flight; phase 2 and 4 are transition phases. Bottom: Drone speed in z_B direction during the different phases

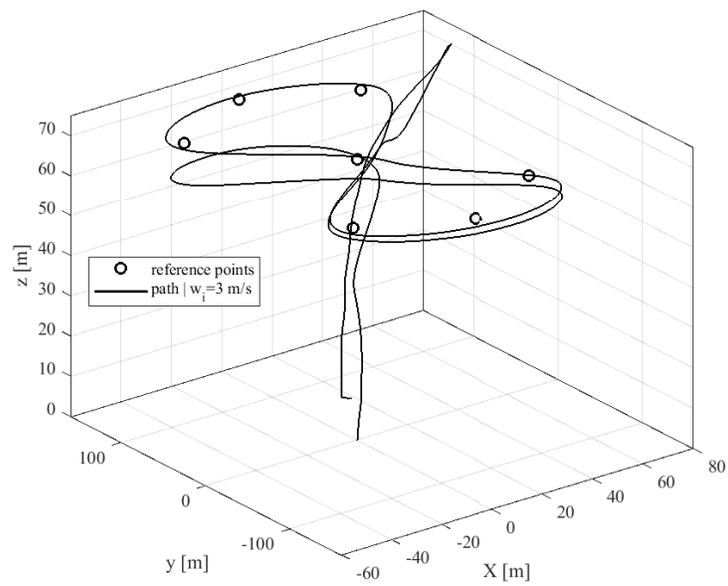


Fig. 9. UAV complete path with 3 m/s wind disturbance.

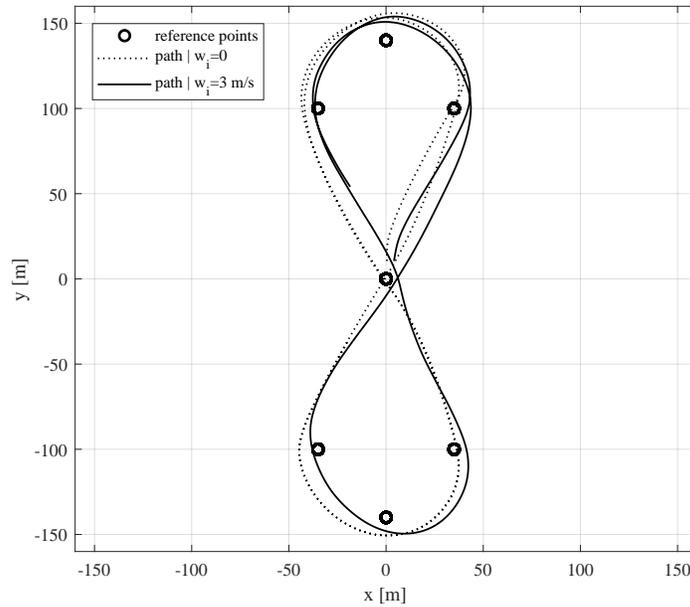


Fig. 10. UAV dynamic flight path comparison with (solid line) and without (dotted) wind disturbances.

VI. CONCLUSIONS AND FUTURE RESEARCH

A full flight controller for a drone to be used in an Airborne Wind Energy system with pumping operation has been presented. The controller has a hierarchical structure and employs mainly linear, static feedback loops. The nonlinear dynamics are dealt with by nonlinear transformations based on kinematics (hence not affected by parametric uncertainty), exploiting the measure of the full system state. The simulation results with a realistic model show that the controller achieves good performance in all flight situations even in presence of wind gusts, notwithstanding its simplicity. These results can be used as starting point to design a controller for the drone also during tethered flight in the various phases of airborne wind energy conversion. The presented controller is currently being employed in experimental tests with a small-scale prototype.

REFERENCES

- [1] U. Ahrens, M. Diehl, and R. Schmehl, Eds., *Airborne Wind Energy*. Berlin Heidelberg: Springer, 2013.
- [2] R. Schmehl, Ed., *Airborne Wind Energy - Advances in Technology Development and Research*. Singapore: Springer, 2018.
- [3] L. Fagiano and M. Milanese, "Airborne wind energy: an overview," in *American Control Conference 2012*, Montreal, Canada, 2012, pp. 3132–3143.
- [4] L. Fagiano, A. U. Zraggen, M. Morari, and M. Khammash, "Automatic crosswind flight of tethered wings for airborne wind energy: modeling, control design and experimental results," *IEEE Transactions on Control System Technology*, vol. 22, no. 4, pp. 1433–1447, 2014.
- [5] M. Erhard and H. Strauch, "Flight control of tethered kites in autonomous pumping cycles for airborne wind energy," *Control Engineering Practice*, vol. 40, pp. 13–26, 2015.
- [6] U. Fechner, R. van der Vlugt, E. Schreuder, and R. Schmehl, "Dynamic model of a pumping kite power system," *Renewable Energy*, 2015.
- [7] A. Zraggen, L. Fagiano, and M. Morari, "Automatic retraction and full-cycle operation for a class of airborne wind energy generators," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 594–698, 2016.
- [8] E. Schmidt, M. D. Lellis, R. Saraiva, and A. Trofino, "State estimation of a tethered airfoil for monitoring, control and optimization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 246 – 13 251, 2017, 20th IFAC World Congress.
- [9] R. van der Vlugt, A. Bley, M. Noom, and R. Schmehl, "Quasi-steady model of a pumping kite power system," *Renewable Energy*, 2018.
- [10] R. Ruiterkamp and S. Sieberling, "Description and preliminary test results of a six degrees of freedom rigid wing pumping system," in *Airborne Wind Energy*, U. Ahrens, M. Diehl, and R. Schmehl, Eds. Berlin Heidelberg: Springer, 2013, ch. 26, pp. 443–458.
- [11] L. Fagiano, E. Nguyen-Van, F. Rager, S. Schnez, and C. Ohler, "Autonomous take off and flight of a tethered aircraft for airborne wind energy," *IEEE transactions on control systems technology*, vol. 26, no. 1, pp. 151–166, 2018.
- [12] G. Licitra, A. Brger, P. Williams, R. Ruiterkamp, and M. Diehl, "Optimal input design for autonomous aircraft," *Control Engineering Practice*, vol. 77, pp. 15 – 27, 2018.
- [13] R. Luchsinger, D. Aregger, F. B. D. Costa, C. Galliot, F. Gohl, J. Heilmann, H. Hesse, C. Houle, T. A. Wood, and R. S. Smith, *Pumping Cycle Kite Power with Twings*, in *Schmehl R. (eds). Airborne Wind Energy. Green Energy and Technology*. Singapore: Springer, 2018.
- [14] E. Bontekoe, "Up! - how to launch and retrieve a tethered aircraft," Master's thesis, TU Delft, August 2010, accessed in September 2018 at <http://repository.tudelft.nl/>.
- [15] L. Fagiano and S. Schnez, "On the take-off of airborne wind energy systems based on rigid wings," *Renewable Energy*, vol. 107, pp. 473–488, 2015.
- [16] Skypull SA. [Online]. Available: <https://www.skypull.technology/>
- [17] J. Zhou, X. Lyu, Z. Li, S. Shen, and F. Zhang, "A unified control method for quadrotor tail-sitter UAVs in all flight modes: Hover, transition, and level flight," in *International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, September 2017, 2017, pp. 4835–4841.
- [18] M. Kokume and K. Uchiyama, "Control architecture for transition from level flight to hover of a fixed-wing UAV," in *37th Annual Conference of the IEEE Industrial Electronics Society*, Melbourne, Australia, November 2011, 2011, pp. 522–527.

- [19] M. L. Loyd, "Crosswind kite power," *Journal of Energy*, vol. 4, no. 3, pp. 106–111, 1980.
- [20] A. Cherubini, A. Papini, R. Vertechy, and M. Fontana, "Airborne wind energy systems: A review of the technologies," *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1461–1476, 2015.
- [21] B. Etkin, *Dynamics of Atmospheric Flight*. Dover, 2005.
- [22] L. Fagiano, A. Zraggen, M. Morari, and M. Khammash, "Automatic crosswind flight of tethered wings for airborne wind energy: modeling, control design and experimental results," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1433–1447, 2014.