Vision-based Air-to-Air Autonomous Landing of underactuated VTOL UAVs

Gabriele Roggi[1], Giovanni Gozzini[1], Davide Invernizzi[1], *Member, IEEE*, Marco Lovera[1], *Member, IEEE*

***Abstract*—This paper addresses the problem of Air-to-Air Autonomous Landing (AAAL) of a small underactuated Unmanned Aerial Vehicle (UAV) on a larger one (Carrier) in a non-cooperative manner using vision-based state estimation. A Kalman-filter based state estimator reconstructs the state of the Carrier relying on a camera mounted on the small UAV (Follower). Then, a three-layer hierarchical architecture is proposed. A hybrid automaton based on a quasi-time optimal approach is used at the position layer to ensure a safe and fast landing. An adaptive observer is developed at the velocity layer to track robustly the reference velocity coming from the position layer and to compensate for the lack of information about the Carrier acceleration. An attitude planner and a geometric stabilizer are used at the innermost layer. The proposed control architecture outperforms the one developed in a previous work by reducing the time to land up to 59% and increasing the accuracy up to 47%. Experimental tests have been performed to assess the performance of the proposed algorithm in representative scenarios.**

***Index Terms*—UAVs, Autonomous landing, Adaptive control, Computer vision.**

I. Introduction

In order to extend small UAVs mission duration, a possible solution regards the use of a Carrier UAV with smaller UAVs that can take-off from and land on it. This problem is not only technologically complex but it is also risky and dangerous and therefore requires a careful design. The Air-to-Air Autonomous Landing (AAAL) problem poses several design challenges: tracking and landing on a fast-moving Carrier; handling the aerodynamic interplay between the Carrier and the Follower; counteracting external disturbances, such as wind gusts, during the landing phase; using a non-cooperative Carrier, to name the most relevant ones. In this paper, the design of a procedure enabling AAAL of underactuated VTOL UAVs in a non-cooperative scenario is investigated.

In the literature many examples of tracking and landing of VTOL UAVs on moving platforms exist, especially onto ground vehicles, *e.g.*, [1]–[13]. One of the first examples is shown in [1]. In this work, an autonomous landing has been accomplished through Image-Based Visual Servoing (IBVS), *i.e.*, using controller acting directly in the image space. Position-Based Visual Servoing (PBVS) has been instead employed with the same purpose in [11]. However, visual servoing represents a valid option only to some extent because it requires the landing platform to be visible throughout the entire duration of the task [4].

In order to deal with missing visual information, model-based approaches have been proposed to predict the motion of the landing target [2], [3]. Within this framework, we distinguish between cooperative and non-cooperative approaches. Among the non-cooperative approaches, [4] proposed using an Extended Kalman Filter (EKF) to deal with missing visual detections and to estimate the full state of the platform, while [6] used an Unscented Kalman Filter (UKF).

To the best of the Authors' knowledge, the work presented in this paper represents one of the few examples of a multirotor UAV landing on a flying platform. Exceptions regard [14], [15], although in these works the target platform is assumed in near hover conditions. A different application with respect to landing is explored in [16], where the interaction of two flying vehicles is considered: the target speed is limited to $1\,\mathrm{m/s}$ during tracking and the experimental results show that the interception occurs with the target in hovering conditions.

While in our previous work [17] we solved the AAAL with a moving Carrier cooperatively, using a motion capture system to provide the Carrier state to the Follower, in this paper we address the more challenging non-cooperative case in which the Carrier state must be reconstructed onboard the Follower by using a dedicated vision-based relative navigation system. Following a widely adopted paradigm in autonomous systems, navigation and control tasks are kept separate. A Kalman Filter (KF) is designed to estimate the Carrier state using information coming from a visual sensor mounted on the Follower. Leveraging the cascade structure of the UAV dynamics, a three-layer architecture is employed to simplify the design of the autonomous landing strategy and of the control law itself. The hierarchical approach allows us to use the hybrid Quasi Time-Optimal (QTO) design presented in [17] as the outer loop controller, which proved effective in solving the landing problem when considering only the UAV kinematics and in the cooperative scenario. At the linear velocity loop level, an adaptive observer-based control law, that outperforms the solution proposed in [17], is employed to compensate for unmodeled dynamics and for the lack of information about the Carrier acceleration, which is required to achieve perfect tracking. An attitude planner and a geometric stabilizer [18], [19] are used at the innermost level to handle the underactuated nature of vectored-thrust UAVs, such as multirotors, which can deliver the control force only along the positive direction orthogonal to the plane of the rotors. The stability of the overall system is then proven using recent

[1]G. Roggi, G. Gozzini, D. Invernizzi and M. Lovera are with Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy {gabriele.roggi,giovanni.gozzini, davide.invernizzi,marco.lovera}@polimi.it.

This work was partially supported by the Italian Ministry for Research in the framework of the 2020 Program for Research Projects of National Interest (PRIN), under Grant 2020RTWES4.

input-to-state stability arguments [20].

The proposed AAAL solution is designed for constant speed trajectories of the Carrier, which can be exactly tracked based on the properties of the control law and of the navigation filter. However, the proposed design has also been tested and proven to handle cases with non-null acceleration, such as circular trajectories. Performance improvement when considering non-trivial trajectories of the Carrier could be achieved by embedding more complex motion models in the navigation filter. Our design can also handle and counteract disturbances through the adaptive part of the control law. In cases of significant disturbances (*e.g.*, wind gusts), the landing logic is such that synchronization mode engages, and the Follower autonomously attempts landing once the disturbance subsides. The only scenario not addressed in this work is when the Follower camera loses sight of the Carrier for an extended period, affecting the prediction model in the Kalman Filter. Finally, an experimental campaign has been carried out using two multirotor UAVs to validate the proposed landing procedure in representative scenarios: a landing with the Carrier following first a straight line (design condition) and then a circular trajectory.

Notation. The canonical basis in $\mathbb{R}^n$ is denoted as $e_i := \begin{bmatrix}0 \dots 0\ 1\ 0 \dots 0\end{bmatrix}^\top$ (1 in the i-th entry, 0 elsewhere) for $i \in \{1, ..., n\}$ and the identity matrix in $\mathbb{R}^{n\times n}$ is denoted as $I_n := [e_1 \cdots e_i \cdots e_n]$. Given $x = [x_1 \cdots x_n]^\top \in \mathbb{R}^n$, $\|x\| := \sqrt{x_1^2 + \ldots + x_n^2}$ is the Euclidean norm while $\|x\|_\infty := \max_{1\le i\le n}(|x_i|)$ is the infinity norm. The set $\mathrm{SO}(3) := \{R \in \mathbb{R}^{3\times 3} : R^T R = I_3, \det(R) = 1\}$ denotes the third-order Special Orthogonal group. The normalized distance with respect to I_3, induced by the Frobenius norm, is denoted as $\|R\|_{\mathrm{SO}(3)} := \frac{1}{8}\|R - I_3\|_F = \sqrt{\frac{1}{4}\mathrm{tr}(I_3 - R)} \in [0, 1]$. Given $\omega \in \mathbb{R}^3$, the map $S(\cdot) : \mathbb{R}^3 \to \mathfrak{so}(3) := \{\Omega \in \mathbb{R}^{3\times 3} : \Omega = -\Omega^T\}$ is such that $S(\omega)y = \omega \times y\ \forall y \in \mathbb{R}^3$, where $\times$ represents the cross product in $\mathbb{R}^3$. The inverse of the $S(\cdot)$ map is the *vee* map, denoted as $S(\cdot)^{-1} : \mathfrak{so}(3) \to \mathbb{R}^3$. Given a differentiable function $f(x) : \mathbb{R}^n \mapsto \mathbb{R}^m$, the Jacobian is denoted by $\mathrm{D}_x f$. We use standard definitions of class-$\mathcal{K}, \mathcal{K}_\infty, \mathcal{KL}$ comparison functions from [21].

II. Problem Description and Autonomous System Architecture

The automatic landing problem considered in this work involves a Follower UAV, equipped with a camera, and a Carrier UAV, carrying a visual fiducial marker.

Several reference frames are introduced to describe the geometry of the problem (Fig. 1): $\mathcal{I} = (O_I, \{i_1, i_2, i_3\})$ is the inertial frame, $\mathcal{B} = (O_B, \{b_1, b_2, b_3\})$ is the Follower body-fixed frame, $\mathcal{C} = (O_C, \{c_1, c_2, c_3\})$ is the camera frame and $\mathcal{T} = (O_T, \{t_1, t_2, t_3\})$ is the visual marker fixed frame.

Following [22], we make the assumption that the angular velocity dynamics is controlled at a sufficiently fast rate. We then consider the simplified dynamics

$$\dot{R}_f = R_f S(\omega_c) \tag{1}$$

$$\dot{x}_f = v_f \tag{2}$$

$$m\dot{v}_f = -mge_3 + t_c R_f e_3 + f_e, \tag{3}$$

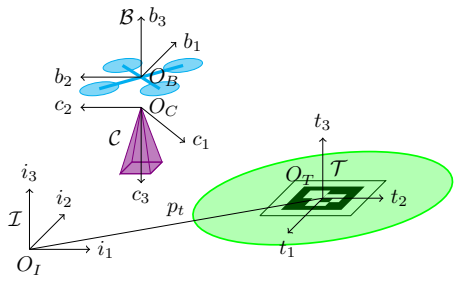

Fig. 1. Reference systems definition.

and focus on the design of a suitable control thrust $t_c \in \mathbb{R}_{>0}$ and angular velocity $\omega_c \in \mathbb{R}^3$ to solve the autonomous landing problem. The state of the Follower has been identified in (1)-(3) with the tuple (x_f, R_f, v_f), where $x_f \in \mathbb{R}^3$ is the position of the center of mass with respect to the inertial frame $\mathcal{I}$, $R_f \in \mathrm{SO}(3)$ is the rotation matrix describing the attitude of the body-fixed frame $\mathcal{B}$ with respect to $\mathcal{I}$, while $v_f \in \mathbb{R}^3 \in \mathbb{R}^3$ is the translational velocity, resolved in $\mathcal{I}$. Finally, f_e is a force (expressed in $\mathcal{I}$) accounting for possible exogenous disturbances, such as aerodynamic drag and wind gusts.

The Carrier UAV is considered as a moving disk described by the following time-varying set (see Fig. 1)

$$\Omega_t(t) = \left\{ y \in \mathbb{R}^3 : e_3^\top (y - x_t(t)) = 0, \|y - x_t(t)\| \le r_t, \right\}, \tag{4}$$

$t \ge 0$, where $x_t \in \mathbb{R}^3$ is the position of the disk center, resolved in $\mathcal{I}$, $r_t \in \mathbb{R}_{>0}$ is the disk radius, while $y \in \mathbb{R}^3$ is a vector describing the position of a generic point inside the moving disk. The above definition also implies the attitude of the Carrier be constantly aligned with the gravity direction, which is the most desirable condition to land. However, the landing strategy proposed in this work will be robust to small attitude motion of the Carrier.

Before stating the autonomous landing problem addressed in this work, we make the following assumptions.

Assumption 1: 1) Images coming from the camera mounted on the Follower provide information about the position of the visual marker frame $\mathcal{T}$ attached to Carrier in the camera frame $\mathcal{C}$, resolved in $\mathcal{C}$; 2) at the initial time, the Follower is above the Carrier and during all the landing operations the visual marker is in the Field of View (FoV) of the camera sensor; 3) the state of the Follower (x_f,v_f,R_f) is known at all times; 4) the velocity $v_t := \dot{x}_t(t)$, acceleration $a_t(t) = \dot{v}_t(t)$ and jerk $\dot{a}_t(t)$ profiles of the Carrier are uniformly bounded.

Note that in Assumption 1 the camera is required to always see the visual marker on the Carrier. In practice, such assumption can be relaxed: by predicting the Carrier motion using a suitably designed navigation filter, missing visual information can be handled for a reasonable amount of time. Based on the second point, we are considering the final phase of the landing, when the Follower is sufficiently close to the Carrier.

Remark 1: No knowledge about the trajectory of the Carrier will be required for the proposed control design to work: the Carrier motion can be any arbitrary trajectory that has uniformly bounded derivatives, as formally required in Assumption 1. Hence, the controller of the Carrier must be capable of stabilizing the UAV about its predefined trajectory,

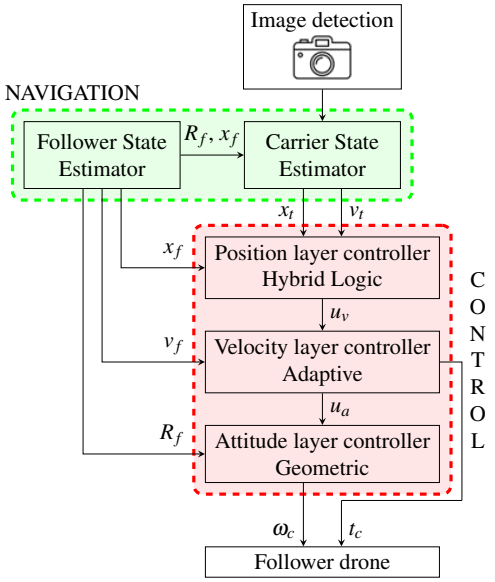

Fig. 2. Block diagram of the autonomous system for the landing problem.

even when the aerodynamic interaction with the Follower becomes significant during the landing phase. Such basic assumption is verified in practice since, by design, the Carrier UAV is a much larger platform than the Follower UAV and the disturbances created during landing can be considered as perturbations within the rejection capabilities of the Carrier control system. These perturbations are taken into account by tuning robustly the Carrier controller. ⌟

The considered air-to-air landing problem can be formalized as follows.

Problem 1: Consider the model of the Follower UAV in (2)-(3) with the (unknown) Carrier motion described by (4). Under Assumption 1, find a control law for (t_c, ω_c) such that the state of the Follower (x_f, v_f) converges *safely* to (Ω_t, v_t), with Ω_t defined in (4), in finite time.

The adverb "*safely*" in Problem 1 encodes the requirement that the Follower has to land from above the Carrier and in a sufficiently slow manner, which allows minimizing perturbation effects between the two UAVs.

The landing problem formalized above is addressed in this work by considering separately navigation and control tasks, as customary in autonomous systems [23] (see Fig. 2).

A navigation function, described in details in Section IV, has been developed to compute state estimates of the Carrier and of the Follower. The proposed navigation function provides estimates of the state of the Carrier using a KF that combines the estimates of the state of the Follower, coming from the onboard state estimator, and the measurement of the relative position of the Carrier with respect to the Follower, coming from an onboard dedicated vision-based algorithm that elaborates the image taken by the camera mounted on the Follower.

Using the Follower and Carrier state estimates provided by the navigation function, a controller has been designed, as described in Section III, to ensure that the Follower tracks the Carrier safely and eventually lands on it. The proposed controller is based on a hierarchical architecture that exploits the cascade structure of the Follower dynamics (1)-(3) to split the control design challenges at different dynamical levels when attemping to solve Problem 1.

III. Control law design

We present in this section the solution to the control task, for which we assume that state estimates of the Carrier and Follower UAVs coming from the navigation function are available at all times. The following three-layer architecture is proposed:

- at the outer-loop level, the velocity is considered as a virtual input to make the position of the Follower track the position of the Carrier;
- the virtual velocity is used as a reference signal for the inner-loop position controller, which assigns the control force;
- a planner computes a virtual attitude that has the third axis aligned with the desired control force;
- the attitude controller computes the angular velocity required to track the planned attitude.

After presenting the structure of the hierarchical architecture, we develop the control laws within each layer. In presenting the high-level control law (corresponding to the position kinematics), we just recall the main steps of the design since the hybrid QTO logic proposed in [17] is exploited with minor adjustments.

A. Hierarchical control design for underactuated UAVs

As the landing problem requires tracking of the Carrier trajectory, we first introduce the tracking error

$$e_p := x_f - x_t \tag{5}$$

the dynamics of which is given by

$$\dot{e}_p = \dot{x}_f - \dot{x}_t = v_f - v_t. \tag{6}$$

Defining u_v as a virtual control input for (6) and $e_v := v_f - u_v$ as the corresponding velocity error, we can rewrite (6) as

$$\dot{e}_p = v_f - v_t \pm u_v = u_v - v_t + e_v. \tag{7}$$

The time-derivative of the velocity error as defined above reads:

$$\dot{e}_v = \dot{v}_f - \dot{u}_v = -ge_3 + a_c R_f e_3 + a_e - \dot{u}_v \tag{8}$$

where $a_c := \frac{t_c}{m}$ is the control acceleration and $a_e := \frac{f_e}{m}$ is the acceleration associated with exogenous disturbances. Note that at this dynamical level, the input $a_c \in \mathbb{R}_{>0}$ appears explicitly but cannot be used directly to control the velocity error dynamics because it does not span $\mathbb{R}^3$. Hence, we first sum and subtract a virtual input $u_a \in \mathbb{R}^3$ in (8)

$$\dot{e}_v = -(ge_3 + \dot{u}_v) + a_c R_f e_3 + a_e \pm u_a. \tag{9}$$

Then, following consolidated hierarchical approaches for underactuated VTOL UAVs, we define a planned attitude $R_p = [\,p_1\ p_2\ p_3\,] \in \mathrm{SO}(3)$ such that $p_3 := \frac{u_a}{|u_a|}$ at all times as[1]:

$$R_p := \left[\, \tfrac{p_3 \times h_d}{|p_3 \times h_d|} \times p_3 \quad \tfrac{p_3 \times h_d}{|p_3 \times h_d|} \quad p_3 \,\right], \tag{10}$$

where $h_d \in \mathbb{R}^3$ is a unit vector, with $h_d^\top e_3 = 0$, specifying a desired heading direction, which defines a secondary control objective with respect to the problem at hand. Setting $a_c = |u_a|$ and then summing and subtracting $|u_a| R_p e_3$ in (3), we obtain the perturbed velocity dynamics

$$\dot{e}_v = -(g e_3 + \dot{u}_v) + u_a + a_e + \Delta a(R_p, R_e, u_a), \tag{11}$$

where $\Delta a := (R_p R_e R_p^\top - I_3) u_a$ represents the mismatch between the delivered and the virtual acceleration and which depends upon the attitude error $R_e := R_p^\top R_f$. Using (1), the dynamics of R_e is given by

$$\dot{R}_e = R_e S(\omega_c - R_e^\top \omega_p), \tag{12}$$

with $\omega_p = S^{-1}(R_p^\top \dot{R}_p)$ being the planned angular velocity that can be computed analytically by time-differentiation of (10), given a sufficiently smooth virtual input u_a and Carrier trajectory as per Assumption 1 (we assume constant heading direction set-point h_d in this work, given that it does not affect the landing procedure).

The following sections discuss in details the derivation of the inputs $(u_v, u_a, t_c, \omega_c)$ to ensure that Problem 1 is solved.

B. Position layer design: Quasi-Time Optimal stabilizer and hybrid logic (design of u_v)

At the highest level, the control design developed in [17] is employed. In deriving the control law for u_v, the position tracking error dynamics with $e_v \equiv 0$ is considered, namely:

$$\dot{e}_p = u_v - v_t. \tag{13}$$

For reasons related to the landing logic, e_p is split in a planar and in a vertical component, denoted as $e_p^\perp := [\,e_{p_1}\ e_{p_2}\,]^\top$ and e_{p_3}, respectively. The following control law taken from [17] is considered:

$$u_v = -\gamma_q(e_p) + v_t, \tag{14}$$

where $\gamma_0(e_p) := \begin{bmatrix} \sigma_{v_s}^{\perp}(k_\perp e_p^\perp) \\ \sigma_{v_s}^{v_s}(k_3(e_{p_3} - h_s)) \end{bmatrix}$, $\gamma_1(e_p) := \begin{bmatrix} \sigma_{v_s}^{\perp}(k_\perp e_p^\perp) \\ \sigma_{v_a}^{v_s}(k_3 e_{p_3}) \end{bmatrix}$ and

$$\sigma_{v_s}^{\perp}(k_\perp e_p^\perp) := \begin{cases} v_s \tanh\left(\frac{k_\perp}{v_s}\|e_p^\perp\|\right) \begin{bmatrix} \frac{e_{p_1}}{\|e_p^\perp\|} \\ \frac{e_{p_2}}{\|e_p^\perp\|} \end{bmatrix} & \text{if } e_p^\top \neq 0 \\ 0 & \text{else} \end{cases} \tag{15}$$

$$\sigma_{v_a}^{v_s}(k_3 e_{p_3}) := \frac{v_s \exp\left(k_3 \frac{v_a + v_s}{v_a v_s} e_{p_3}\right) - v_s}{\exp\left(k_3 \frac{v_a + v_s}{v_a v_s} e_{p_3}\right) + v_s / v_a} \tag{16}$$

[1]The operating conditions considered for the landing are far away from the singularity arising when $u_a = 0$ (p_3 undefined), which corresponds to free fall flight, and do not impact our (local) results (Theorem 2). A discussion on possible remedies for this issue can be found in [22].

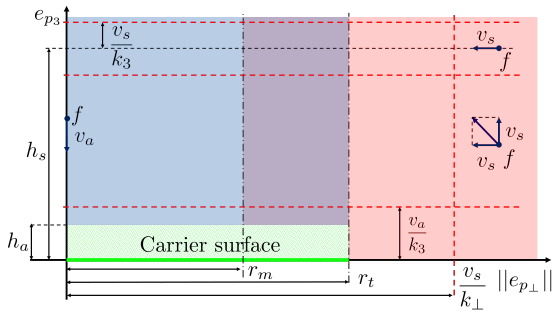

Fig. 3. Landing logic and commanded velocity of the Follower (u_v). The coloured regions represent the different domains of the hybrid automaton which give for each mode the set where the continuous state evolves: synchronization mode in red, approach mode in blue and land mode in green. The overlapping between synchronization and approach modes (violet) is due to the hysteresis mechanism implemented to avoid chattering. The red dashed lines identify the values at which the Follower switch from a saturated velocity regime to the unsaturated one and vice versa. For further details about this figure we address the reader to [17].

are smooth saturation functions[2] with saturation levels $v_a, v_s \in \mathbb{R}_{>0}$, and $k_\perp, k_3$ are scalar positive gains. As shown in [17], the control law (14) ensures Global Asymptotic Stability (GAS) of the equilibrium point $e_p = 0$ for the system (13).

The landing logic can be described by a hybrid automaton, in which three different modes are defined using a logic state $q \in Q := \{0, 1, 2\}$ (see Fig. 3):

- During the Synchronize Mode ($q = 0$), the Follower has to get close to a position at a certain height above the Carrier in a sufficiently fast way. The control law

$$u_v = \begin{bmatrix} -\sigma_{v_s}^{\perp}(k_\perp e_p^\perp) + v_{t_\perp} \\ -\sigma_{v_s}^{v_s}(k_3(e_{p_3} - h_s)) + v_{t_3} \end{bmatrix} \tag{17}$$

  is implemented to ensure tracking of a point at a height $\mathbb{R}_{>0} \ni h_s \gg h_a$ above the Carrier, namely, $x_s := x_t + [\,0\ 0\ h_s\,]^\top$ which represents a safe point to start the landing.
  Remark 2: The scalar h_s is a control variable that specifies the relative altitude above the Carrier that the Follower must reach during the synchronization phase. When the Follower is in the approach mode and exits for any reason the landing domain (blue-violet area in Fig. 3), the hybrid logic activates the synchronization mode and the Follower is commanded to increase its relative altitude to get back to the safety point with relative vertical distance equal to h_s. To reduce the landing time, a slightly modified strategy has been implemented in the experiments: h_s is used as an additional state of the logic which is updated by setting it equal to the relative vertical distance that has been achieved right before exiting the approach domain. ⌟
  The mode is active outside a cylinder of radius $r_m < r_t$ about the vertical axis passing through x_t (red+violet area in Fig. 3).

[2]The two functions correspond to sigmoid approximations of $\mathrm{sat}_{v_s}^{\perp}(k_\perp e_p^\perp) = \min\left(k_\perp, v_s / \|e_p^\perp\|\right) e_p^\perp$ and $\mathrm{sat}_{v_a}^{v_s}(k_3 e_{p_3}) = \min(\max(k_3 e_{p_3}, -v_a), v_s)$. Using smooth saturation functions is required here because the inner loop controller presented in Section III-C requires the time derivative of u_v.

- In the Approach Mode ($q=1$), the Follower starts a sufficiently slow and controlled descent towards the Carrier. Specifically, the control law
$$u_v = \begin{bmatrix} -\sigma_{v_s}^{\perp}(k_{\perp} e_p^{\perp}) + v_{t_{\perp}} \\ -\sigma_{v_a}^{v_s}(k_3 e_{p_3}) + v_{t_3} \end{bmatrix} \tag{18}$$
is implemented to ensure tracking of the Carrier point x_t, with the approaching vertical speed bounded by v_a. The mode is active inside a cylinder of radius r_t about the vertical axis passing through x_t (blue+violet area in Fig. 3).
- The Land Mode ($q=2$) is triggered when the Follower is inside the approach cylinder below height h_a above the Carrier landing surface (green area in Fig. 3). When the landing command is activated, the UAV is disarmed.

The violet area where the approach and synchronization modes are active simultaneously introduces a hysteresis mechanism that avoids chattering. The reader is referred to [17] for the specific definition of the elements of the hybrid automaton (edges, guard conditions, reset maps, *etc.*) and the proof that the proposed logic solves the landing problem when considering the kinematic model alone.

C. Velocity layer design: adaptive controller (design of u_a, t_c)

The velocity controller is in charge of tracking the virtual velocity u_v commanded by the position controller layer using the virtual acceleration input u_a. In deriving the control law, we assume $\Delta a \equiv 0$ and substitute the expression of u_v in (14) inside (11):

$$\begin{aligned} \dot{e}_v &= -g e_3 + u_a + a_e + \mathrm{D}_{e_p}\gamma_q(e_p)\dot{e}_p - \dot{v}_t \\ &= -g e_3 + u_a + a_e + \mathrm{D}_{e_p}\gamma_q(e_p)(e_v - \gamma_q(e_p)) - \dot{v}_t \end{aligned} \tag{19}$$

where $u_a := \frac{u}{m}$ is the control acceleration and a_e was defined in (8). As already mentioned, the Carrier acceleration $\dot{v}_t$ is not in general available (the navigation function provides only position and velocity estimates) and therefore is treated here as an exogenous term. The velocity error dynamics thus becomes:

$$\dot{e}_v = -g e_3 + \mathrm{D}_{e_p}\gamma_q(e_p)(e_v - \gamma_q(e_p)) + u_a + a_d \tag{20}$$

where $a_d := a_e - \dot{v}_t$ is a disturbance acceleration. To ensure stabilization of $e_v = 0$ in the presence of the disturbances, we propose the following observer-based adaptive control law

$$\dot{\hat{e}}_v = -\gamma_v(e_v) + L(e_v - \hat{e}_v) \tag{21}$$
$$\dot{\hat{a}}_d = \mathrm{proj}(\hat{a}_d, -\Gamma_d(\hat{e}_v - 2e_v)) \tag{22}$$
$$u_a = g e_3 - \mathrm{D}_{e_p}\gamma_q(e_p)(e_v - \gamma_q(e_p)) + \gamma_v(e_v) - \hat{a}_d, \tag{23}$$

where $(\hat{e}_v, \hat{a}_d) \in \mathbb{R}^3 \times \mathbb{R}^3$ is the state of the controller, $\gamma_v(e_v) := \sigma_M(K_v e_v)$ is a saturated velocity stabilizer, $L \in \mathbb{R}^{3\times 3}$, $\Gamma_d \in \mathbb{R}^{3\times 3}$, $K_v \in \mathbb{R}^{3\times 3}$ are diagonal positive definite matrices, and $M \in \mathbb{R}_{>0}$ are saturation levels for each axis. Finally, given $y \in \mathbb{R}$, the projection operator is defined as:

$$\mathrm{proj}(\theta, y) := \begin{cases} y - f(\theta)\frac{\nabla f(\theta)(\nabla f(\theta))^{\top}}{\|\nabla f(\theta)\|^2} y, & \text{if } [f(\theta) > 0 \wedge y \nabla f(\theta) > 0] \\ y, & \text{otherwise} \end{cases} \tag{24}$$

where $\theta \in \mathbb{R}$ is the parameter estimate and $f(\theta) := \frac{\|\theta\|^2 - \theta_M^2}{2\varepsilon\theta_M^2 + \varepsilon^2}$, with $\varepsilon > 0$ a tunable parameter and θ_M the maximum absolute value of θ. When applied to a vector, the operator is assumed to act component-wise and to keep the corresponding estimate bounded in the interval $\pm\theta_M$ (see [24, Chapter 11] for more information about the projection operator).

Remark 3: The observer dynamics (21) is obtained by replicating the velocity error dynamics (20) with the estimate $\hat{a}_d$ in place of the unknown term a_d, namely,

$$\dot{\hat{e}}_v = -g e_3 + \mathrm{D}_{e_p}\gamma_q(e_p)(e_v - \gamma_q(e_p)) + u_a + \hat{a}_d \tag{25}$$

and then by adding an output injection term $L(e_v - \hat{e}_v)$ to improve convergence speed and smoothen the oscillations in the estimated parameter signal. Choosing L close to Γ_d guarantees a good trade-off between speed of convergence and oscillations of the estimate. When substituting the control law (23) into (25), one obtains the observer dynamics reported in (21). ⌟

By defining the error $\tilde{e}_v := \hat{e}_v - e_v$, the closed-loop dynamics obtained combining (8) with the controller (21)-(23) is described by the following perturbed nonlinear system:

$$\dot{e}_v = -\gamma_v(e_v) - \tilde{a}_d \tag{26}$$
$$\dot{\tilde{e}}_v = -L\tilde{e}_v + \tilde{a}_d \tag{27}$$
$$\dot{\tilde{a}}_d = \mathrm{proj}(\hat{a}_d, -\Gamma_d(\tilde{e}_v - e_v)), \tag{28}$$

where $\tilde{a}_d := \hat{a}_d - a_d$ is the parameter estimation error.

The following theorem, the proof of which is reported in the Appendix, establishes the stability properties guaranteed by the inner-loop controller.

Proposition 1: Consider the closed-loop dynamics given in (26)-(28). When $\dot{a}_d = 0$, the equilibrium point $(e_v, \tilde{e}_v, \hat{a}_d) = (0, 0, a_d)$ is Globally Asymptotically Stable (GAS). For any bounded $\dot{a}_d$, the closed-loop solutions of the system are Uniformly Ultimately Bounded (UUB) for any initial condition in the set $\mathcal{R}_v := \{(e_v, \tilde{e}_v, \hat{a}_d) \in \mathbb{R}^9 : \|\hat{a}_d\| \le a_M\}$, where a_M is the upper bound on the disturbance acceleration.

Remark 4: The adaptive controller (21)-(23) allows to effectively counteract disturbances and to mitigate the lack of acceleration information from the navigation function by collapsing their effect in the disturbance acceleration a_d. Based on Proposition 1, constant wind gusts are exactly rejected, ensuring a safer landing. ⌟

D. Attitude layer design: geometric controller

Given that the attitude error kinematics (12) is fully actuated by ω_c, the following rotation-matrix based law is employed:

$$\omega_c := -\gamma_R(R_e) + R_e^{\top}\omega_p, \tag{29}$$

where $\gamma_R(R_e) := \frac{1}{2}S^{-1}(K_R R_e - R_e^{\top} K_R)$ and K_R is a gain matrix. By substituting (29) into (12), the error attitude kinematics is described by the autonomous system

$$\dot{R}_e = -R_e S(\gamma_R(R_e)). \tag{30}$$

Proposition 2: Consider the closed-loop attitude kinematics in (30). For any symmetric matrix $K_R \in \mathbb{R}^{3\times 3}$ such that $\mathrm{tr}(K_R) I_3 - K_R$ is positive definite, then the equilibrium point

$R_e = I_3$ is almost globally asymptotically stable. Moreover, given any $0 < \ell < \underline{\ell}_R := \lambda_m(\mathrm{tr}(K_R)I_3 - K_R)$, for any initial condition in the set $\mathcal{S}_R := \left\{R \in \mathrm{SO}(3) : \frac{1}{2}\mathrm{tr}(K_R(I_3 - R)) \leq \ell\right\}$, the solution satisfies $\|R_e(t)\|_{\mathrm{SO}(3)} \leq \sqrt{\frac{\ell}{\underline{\ell}_R}} \quad \forall t \geq t_0$ and $\lim_{t\to\infty} \|R_e(t)\|_{\mathrm{SO}(3)} = 0$.

The proof of Proposition 2 is omitted as it follows from existing results in the literature [18], [19]. For the special case of $K_R = k_r I_3$, the basin of attraction can be as large as $\|R_e\|_{\mathrm{SO}(3)} < 1$, which corresponds to all but 180 deg rotations. The proposed stabilizer possesses also robustness properties with respect to angular velocity perturbations [19].

E. Stability analysis of the hierarchical controller

The stability of the overall control system will be assessed within each operating mode. We first establish a stability result which is valid for the case of fully actuated UAVs (for which attitude and position can be controlled independently), characterized by the following error dynamics:

$$\dot{e}_p = -\gamma_q(e_p) + e_v \tag{31}$$
$$\dot{e}_v = -\gamma_v(e_v) - \tilde{a}_d \tag{32}$$
$$\dot{\tilde{e}}_v = -L\tilde{e}_v + \tilde{a}_d \tag{33}$$
$$\dot{\hat{a}}_d = \mathrm{proj}(\hat{a}_d, -\Gamma_d(\tilde{e}_v - e_v)), \tag{34}$$

which is in the form of a cascade system, wherein (31) is the lower subsystem, while (32)-(34) is the upper subsystem, which is perturbed by $\dot{a}_d$. The stabilizing properties of the proposed control design are stated in the following theorem, the proof of which can be found in the Appendix.

Theorem 1: Consider the closed-loop dynamics (31)-(34). If $\dot{a}_d = 0$, the equilibrium point $(e_p, e_v, \tilde{e}_v, \hat{a}_d) = (0,0,0,a_d)$ is GAS. If $\dot{a}_d \neq 0$, then there exist control parameters such that the closed-loop solutions starting from $\mathbb{R}^3 \times \mathcal{R}_v$ are UUB.

For underactuated UAVs, the following error dynamics must be instead considered:

$$\dot{R}_e = -R_e S(\gamma_R(R_e)) \tag{35}$$
$$\dot{e}_p = -\gamma_q(e_p) + e_v \tag{36}$$
$$\dot{e}_v = -\gamma_v(e_v) - \tilde{a}_d + \Delta a(R_p, R_e, u_a) \tag{37}$$
$$\dot{\tilde{e}}_v = -L\tilde{e}_v + \tilde{a}_d - \Delta a(R_p, R_e, u_a) \tag{38}$$
$$\dot{\hat{a}}_d = \mathrm{proj}(\hat{a}_d, -\Gamma_d(\tilde{e}_v - e_v)), \tag{39}$$

In this case, the following result holds, as proven in the Appendix.

Theorem 2: Consider the closed-loop dynamics (35)-(39). If $\dot{a}_d = 0$, the equilibrium point $(R_e, e_p, e_v, \tilde{e}_v, \hat{a}_d) = (I_3, 0, 0, 0, a_d)$ is locally asymptotically stable. If $\dot{a}_d \neq 0$, then there exist control parameters such that the closed-loop solutions are UUB.

Remark 5: To understand how the landing logic, designed on the kinematic model, works when applied to the dynamic model, let us point out that while the above stability analysis has been carried out on the three dimensional (3D) model, for the sake of compactness, the choice of the feedback errors in Section III-B allows decoupling the kinematics in a planar 2D and a vertical 1D motion. The stabilizer of planar motion (17)-(18) is the same in both the synchronization and approach mode. Therefore, the planar error dynamics only flows as long as the land mode is not activated and one can conclude that $\|e_p^{\perp}\| < r_m$ after a finite time, provided that the ultimate bound of the solutions is strictly less than r_m. As such, after possibly multiple switches between the synchronize and approach mode, the Follower will then stay in the approach mode until landing. ⌟

IV. Vision-based Carrier state estimation

Vision is used to estimate the position and velocity of the Carrier in an inertial frame. In order to simplify the detection task, an ArUco marker has been attached to the Carrier and employed as visual marker. The approach can be generalized to different tags and detection algorithms. ArUco marker detection and pose reconstruction is carried out on the images coming from the camera equipped on the Follower, exploiting the OpenCV library [25].

The position x_t and the velocity v_t of the Carrier in the inertial frame are estimated through a discrete-time KF. The vision-based state estimation algorithm is presented in discrete-time, consistently with the reference literature on the topic. Let the state vector be defined as $x = \begin{bmatrix} x_t & v_t \end{bmatrix}^\top$. A kinematic model is used for the Carrier motion, corresponding to the following discrete-time model:

$$x_k = Fx_{k-1} + w_{k-1} \tag{40}$$

with $F = \begin{bmatrix} I_3 & I_3\Delta t \\ 0_3 & I_3 \end{bmatrix}$, where Δt is the sampling period of the filter. The process noise w_k is assumed to be a zero-mean Gaussian white noise with covariance matrix Q.

The correction phase is performed each time a measurement, *i.e.*, the 3D position of the moving Carrier, is provided by the ArUco detection and pose reconstruction algorithms. Note that, even using a monocular camera, we are able to solve the ambiguity of scale by knowing the ArUco marker size.

The measurement model can be written as:

$$y = R_c(R_f^\top(x_t - x_f)) + p_c, \tag{41}$$

where p_c and R_c are the known fixed translation and rotation of the camera frame $\mathcal{C}$ relative to the Follower body frame $\mathcal{B}$.

As can be seen from equation (41), the position x_f and attitude R_f of the Follower in the inertial frame are needed (Assumption 1). They are directly retrieved from the Follower state estimator. Being the measurement model linear in the state, we can evaluate it at instant k as:

$$y_k = H_k x_k + v_k, \tag{42}$$

where the measurement noise v_k is a Gaussian random vector with zero mean and covariance R, uncorrelated with the process noise w_k, and $H_k = \left.\frac{\partial y}{\partial x}\right|_k$ is the measurement model Jacobian evaluated at instant k.

Remark 6: The prediction model (40) is capable of handling exactly constant speed motions for the Carrier (that is a typical design condition for landing). The algorithm provides satisfactory results also in case of trajectories with time-varying speed (see the circle trajectory case study) through a proper tuning. Nonetheless, it could also be possible to

substitute the model with a more detailed one, if additional information about the motion of the Carrier were available. ⌟

The filter prediction step is written as:

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ \tag{43}$$

$$P_k^- = FP_{k-1}^+F^\top + Q \tag{44}$$

where P^- and P^+ are the *a priori* and *a posteriori* state estimation error covariance matrices respectively, and $\hat{x}^-$ and $\hat{x}^+$ are the *a priori* and *a posteriori* estimates of the state.

The filter correction step can be summarized with the following equations:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k z_k \tag{45}$$

$$P_k^+ = (I_6 - K_k H_k)P_k^-. \tag{46}$$

where $z_k = y_k - H_k\hat{x}_k^-$, $Z_k = H_kP_k^-H_k^\top + R$ and $K_k = P_k^-H_k^\top Z_k^{-1}$. Note that after the computation of z_k and of its covariance Z_k, in order to be robust to possible measurement outliers, we perform a χ^2-test based on the Mahalanobis distance of the innovation, as shown in [26]. Inliers are validated by checking the Normalized Estimation Error Squared (NEES), namely:

$$z^\top Z^{-1} z \leq \chi^2_{th}, \tag{47}$$

with χ^2_{th} equal to the 0.95 probability quantile of the χ^2 distribution. If the measurement passes the test, we proceed by computing the Kalman gain K and by updating the filter state and covariance.

Remark 7: Formally studying the stability properties of the complete closed-loop system (navigation and control) is out of the scope of this work. Indeed, establishing stability properties of vision-based state estimators is a challenging task on its own [27]. We rely here on a separation principle assumption, which is commonly accepted when dealing with complex autonomous systems. Of note, the running frequency of the state estimator is high such that the filter output can be considered as continuous when addressing the stability analysis of the closed-loop system, that is designed to have a much slower dynamics. Assuming that the errors associated with navigation estimates are bounded and sufficiently small, the complete system will remain stable, given that the control systems is able to tolerate small perturbations, as established in Theorem 1 and 2. Note that the estimation errors can actually be significant in practice, especially in case of dynamic trajectories with time-varying speed profiles (see Fig. 5 in the experiment section). Nonetheless, the proposed control law is capable of handling them well, ensuring a safe landing. ⌟

V. Experimental results

A. Experimental setup

Flight tests are carried out inside the Flying Arena for Rotorcraft Technologies (FlyART) of Politecnico di Milano which is an indoor facility equipped with an Optitrack Motion Capture system (Mo-Cap). The UAV used as Follower, codename ANT-X, is a quadcopter, while the Carrier one, codename CARRIER-1, is an octocopter with a flat landing surface with an ArUco marker attached (Fig. 4). In all experiments and simulations presented next we considered initial conditions

Fig. 4. CARRIER-1 (Carrier) and ANT-X (Follower) UAVs.

in which the marker on the Carrier surface is always in the camera FoV; this can be considered the end goal of an exploration phase (see [12]). The Mo-Cap system, composed by 12 cameras, detects reflectors mounted on the UAVs and provides both ground truth data and position and yaw measurements for the Follower state estimation filters. The Follower carries a monocular camera to detect the ArUco marker and estimates the state of the Carrier using on-board computational capabilities. The companion computer is a small open source ARM board (NanoPi NEO Air[3]). On this board, the virtual velocity input is also computed according to the outer-loop control law and communicated to the PX4 autopilot using the Robotic Operating System (ROS) as middleware. The inner-loop control law has been integrated in the PX4 firmware using the ANT-X rapid prototyping system for multirotor control [28].

Before performing experiments, simulations have been conducted using the Software-in-the-Loop (SITL) PX4 autopilot firmware environment, which employs the Gazebo simulator and ROS. A video of the simulation is available online[4].

B. Preliminary test results

Before performing vision-based landing experiments, preliminary tests have been conducted focusing on the quality of the estimation with different values of camera parameters. In particular with the Carrier still on the ground and the Follower moving slowly above it[5], we have tried to quantify the mean μ and the standard deviation σ of the error between the true and the estimated position of the Carrier at different values of camera resolution (QQVGA, QVGA, VGA)[6]. The obtained results are shown in Table I. Given the trade-off between accuracy and CPU load, the QVGA resolution has been chosen for all the tests.

Another test has been performed with the Follower in synchronization mode at an altitude of 2.5 m and estimating the Carrier that is moving along a circular trajectory with 1 m radius and angular frequency $0.2\,\mathrm{rad/s}$ at a constant altitude of 1 m. While the error in the position estimate is small, the main difference is in the velocity estimate, which has a significant delay (almost 1 s), as can be seen in Fig. 5.

[3]https://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Air

[4]Visit https://www.youtube.com/watch?v=LsINkVjS6R0 or the ASCL website http://ascl.daer.polimi.it.

[5]With the marker always in the camera FoV.

[6]Higher resolutions have not been considered because of the CPU load.

[7]Rate of the algorithm that detects the ArUco marker.

TABLE I
MEAN AND STANDARD DEVIATION OF THE ERROR BETWEEN THE TRUE AND THE ESTIMATED POSITION OF THE CARRIER.

	QQVGA	QVGA	VGA
Image rate [Hz]	50	30	15
ArUco rate[7] [Hz]	6	4	4
μ_x [m]	0.071	0.042	0.053
σ_x [m]	0.035	0.042	0.039
μ_y [m]	0.043	0.014	0.019
σ_y [m]	0.039	0.035	0.042
μ_z [m]	-0.042	0.045	0.097
σ_z [m]	0.041	0.077	0.063
CPU load	78%	85%	88%

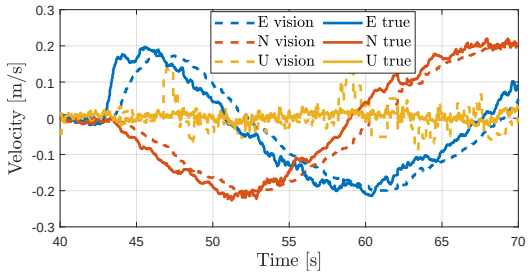

Fig. 5. True velocity of the Carrier and estimate obtained from the KF described in Section IV using the vision measurement.

C. Comparison with our previous design [17]

An experimental campaign has been conducted to compare the adaptive controller ("New") and the one proposed in [17] ("Old"). The latter was a simple QTO controller for the position layer, tuned properly to obtain a safe and fast landing as described precisely in [17], together with a PID controller on the velocity layer tuned with structured H_∞. In particular, the AAAL problem has been performed exploiting the external Mo-Cap system to obtain the state (position and velocity) estimates of both the two UAVs instead of using the monocular camera onboard the Follower. This has been done to eliminate the delay in the estimate of the velocity in order to be able to compare the two controllers in the same conditions with a repeatable experiment. The experiment has been conducted with the Carrier moving along a circular trajectory of 1 m radius centered in $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$ m with respect to the ENU (East-North-Up) inertial frame of the flying arena and the Follower starting at $\begin{bmatrix}0 & 0.75 & 2.5\end{bmatrix}^\top$ m. The test has been repeated five times for each of the two different values of velocity of the Carrier and has shown that the adaptive controller outperforms the one in [17] as can be seen in Table II where the mean values of the five tests of the time to land (TTL), the final in-plane position error $\|e_p^\perp(t_f)\|$, the in-plane position error mean $\overline{\|e_p^\perp\|}$ and standard deviation $\sigma_{p_\perp}$ from the beginning of the landing algorithm are reported. In Fig. 6 the in-plane relative position time history of one of the experiments can be seen using the two controllers with an angular frequency for the Carrier trajectory of $0.4\,\mathrm{rad/s}$.

D. Vision-based landing results

The landing has been performed in two conditions: with the Carrier moving along a circular trajectory of 1 m radius and angular frequency $0.25\,\mathrm{rad/s}$ and with the Carrier moving along a linear trajectory with velocity up to $0.4\,\mathrm{m/s}$. The controller gains used in the experiments are: $k_\perp = 1.4$, $k_3 = 1$, $K_v = 3I_3$, $v_a = 0.3\mathrm{m/s}$, $v_s = 1\mathrm{m/s}$, $M = 4.9\mathrm{m/s}^2$, $K_R = \mathrm{diag}(2.8, 2.8, 17.2)$, $h_s(0) = 1.5\mathrm{m}$, $\Gamma_d = 15I_3$ and $L = 10I_3$. The parameters used in the KF are $Q = \begin{bmatrix}0_3 & 0_3\\ 0_3 & q_v\end{bmatrix}$ with $q_v = 10^{-4}I_3\mathrm{m}^2$, and $R = 10^{-2}I_3\mathrm{m}^2$, selected trading-off noise in the estimate and phase shift with offline simulations of the KF that uses experimental data of the relative pose provided by ArUco detection algorithm recorded during a synchronization test.

TABLE II
TIME TO LAND (TTL), FINAL IN-PLANE POSITION ERROR $\|e_p^\perp(t_f)\|$, IN-PLANE POSITION ERROR MEAN $\overline{\|e_p^\perp\|}$ AND STANDARD DEVIATION $\sigma_{p_\perp}$ FROM THE BEGINNING OF THE LANDING ALGORITHM.

	$\omega = 0.25\,\mathrm{rad/s}$		$\omega = 0.4\,\mathrm{rad/s}$	
	Old	New	Old	New
TTL [s]	7.66	5.82	14.75	6.05
$\|e_p^\perp(t_f)\|$ [m]	0.0725	0.0548	0.0926	0.0126
$\overline{\|e_p^\perp\|}$ [m]	0.0812	0.0724	0.1030	0.0544
$\sigma_{p_\perp}$ [m]	0.0480	0.0546	0.0500	0.0559

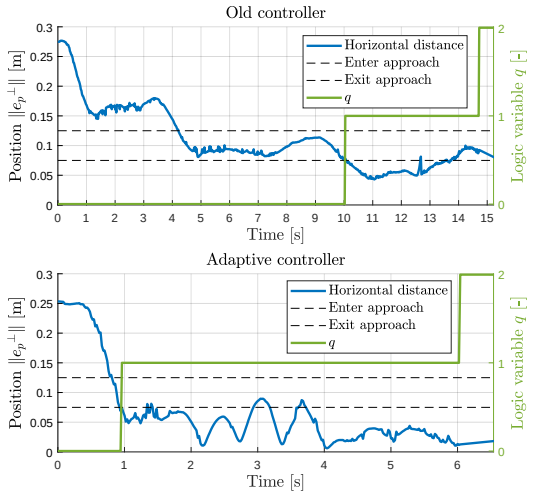

Fig. 6. In-plane relative position $\|e_p^\perp\|$ time history from the beginning of the Carrier trajectory to 0.5 seconds after landing for the new and the old controller during the experiment with angular frequency $0.4\,\mathrm{rad/s}$.

For the circular trajectory case, the Follower starts at $\begin{bmatrix}0 & 0.75 & 2.5\end{bmatrix}^\top$ m while the circular trajectory is centered in $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$ m with respect to the inertial frame (with ENU convention) of the flying arena. In Fig. 7, the in-plane position trajectories of the two UAVs are shown together with the Carrier desired trajectory. Markers are used to identify the switches of the logic variable q. In correspondence of circle markers and cross markers, the synchronization mode ($q = 0$) and the approach mode ($q = 1$) are activated, respectively. Instead, the star marker indicates when the disarm command is sent to the Follower ($q = 2$). As can be observed, at the beginning the synchronization mode is active and the Follower tracks the Carrier keeping the vertical distance constant. When the in-plane error $p_\perp$ is less then $0.01 = r_m < r_t = 0.015\mathrm{m}$, the logic variable switches to 1 and the Follower starts the approach phase. Right after entering, the Follower exits the approach mode because of the delay in the estimate of the

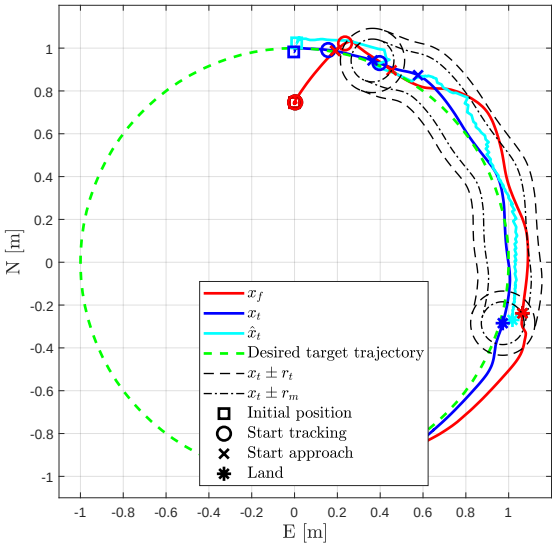

Fig. 7. In-plane position of the Follower (red), desired (green), estimated (cyan) and real (blue) Carrier position from the beginning of the Carrier trajectory to 3 seconds after landing. Comparing the desired Carrier trajectory (green) with the actual Carrier position (blue), it can be appreciated how well the Carrier is able to follow the desired circular trajectory even during the approach phase. This has been achieved by tuning robustly the Carrier controller in order to better reject disturbances.

velocity of the Carrier and therefore the logic state is switched back to the synchronization mode again. When the in-plane distance is again lower than r_m, q jumps to 1. Finally it keeps in the C_1 domain until the landing mode is activated ($q=2$) at height $h_a = 0.15$m. In Fig. 8 and Fig. 9, the true and estimated relative in-plane and vertical distances are shown together with the time evolution of the logic variable.

Remark 8: The relative distance obtained from the vision system underestimates the true one. This could lead to dangerous situations, *e.g.*, landing outside the Carrier surface. For this reason, we have chosen conservative bounds r_m and r_t with respect to the ones in [17] considering the values for the mean of the error reported in Table I. In this way, even if at the end of the approach phase the true in-plane relative distance is slightly outside the boundary of the domain, landing can be considered safe. ⌟

Remark 9: During the final part of the descent, at a certain height above the Carrier, the tag can no longer be seen by the camera. This height depends on the marker size and on the camera FoV. Thus, a proper selection of these parameters is needed. It is worth noting that the proposed approach can cope with missing visual information for an amount of time dependent on the accuracy of the motion model of the Carrier. ⌟

The performance of the proposed design is clearly limited by the delay of the estimated Carrier velocity (Fig. 5), that is prominently related to the model used for the prediction step, which is kept simple (constant velocity), being a reasonable design condition for the landing. As can be seen in Fig. 10 when a linear trajectory with velocity $v_t = 0.4\,\mathrm{m/s}$ is considered for the Carrier, the results improve in terms of estimation performance because the real motion of the Carrier and the model used in the estimator match. In this experiment

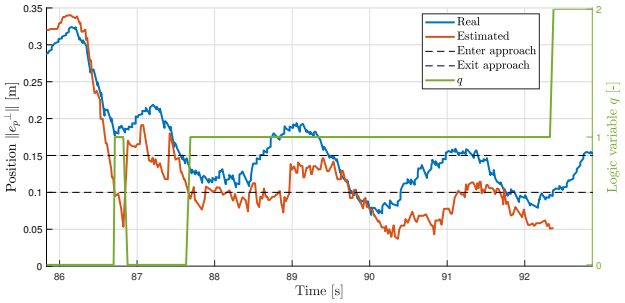

Fig. 8. True and estimated in-plane relative position $\|e_p^\perp\|$ time history from the beginning of the Carrier trajectory to 0.5 seconds after landing. The initial peak in the difference between estimated and real relative positions can be explained considering that aggressive rotational and translational movements of the Follower can cause degradation in the quality of images, and, thus, of measurements.

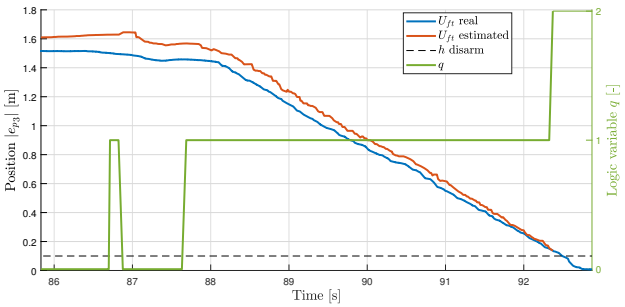

Fig. 9. True and estimated relative vertical position $|e_{p_3}|$ time history from the beginning of the Carrier trajectory to 0.5 seconds after landing.

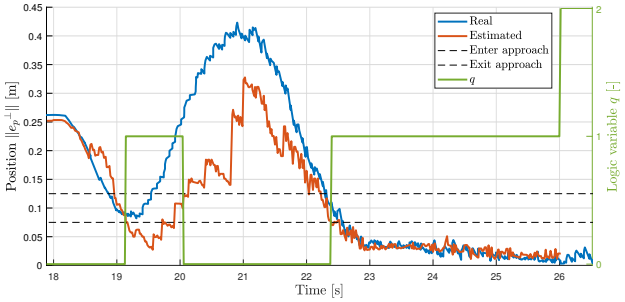

Fig. 10. True and estimated in-plane relative position $\|e_p^\perp\|$ time history during Carrier linear trajectory.

the bounds for the in-plane error were further reduced to $0.0075 = r_m < r_t = 0.0125$m. A video of the experiments is available online.[8]

VI. Conclusion

In this paper we tackled the problem of vision-based AAAL of UAVs. A KF state estimation approach has been used to determine the position and velocity of the Carrier UAV in a non-cooperative manner. The availability of this information has been exploited to perform a safe landing combining a hybrid QTO approach and an observer-based adaptive law that outperforms the performance of the controller presented in our previous work. Finally, the proposed strategy has been validated through simulations and an experimental campaign involving two multirotor UAVs.

[8]Visit https://www.youtube.com/watch?v=aPF_--zHy2w or the ASCL website http://ascl.daer.polimi.it.

APPENDIX

Proof of Theorem 1. In case $\dot{a}_d = 0$, then system (26), (27) becomes an autonomous one. Define $x_v := (e_v, \tilde{e}_v, \hat{a}_d)$, then the singleton $\mathcal{A}_v := \{x_v \in \mathcal{R}_v : x_v = (0,0,a_d)\}$ is the unique equilibrium point. Consider the Lyapunov function

$$V(e_v, \tilde{e}_v, \hat{a}_d) := \tfrac{1}{2}\|e_v\|^2 + \tfrac{1}{2}\|\tilde{e}_v\|^2 + \tfrac{1}{2}\tilde{a}_d^\top \Gamma_d^{-1}\tilde{a}_d, \tag{48}$$

which satisfies $\underline{\alpha}(|x_v|_{\mathcal{A}_v}) \leq V(x_v) \leq \overline{\alpha}(|x_v|_{\mathcal{A}_v})\ \ \forall x_v \in \mathbb{R}^9$, where $\underline{\alpha}(|x_v|_{\mathcal{A}_v}) := \min\left(1, \lambda_M^{-1}(\Gamma_d)\right)|x_v|^2_{\mathcal{A}_v}$ and $\overline{\alpha}(|x_v|_{\mathcal{A}_v}) := \max\left(1, \lambda_m^{-1}(\Gamma_d)\right)|x_v|^2_{\mathcal{A}_v}$ are (quadratic) class$-\mathcal{K}_\infty$ functions and $|x_v|_{\mathcal{A}_v} := \|(e_v, \tilde{e}_v, \tilde{a}_d)\|$. Since the projection operator satisfies $(\hat{a}_d - a_d)^\top(\text{proj}(\hat{a}_d, -y) + y) \leq 0\ \ \forall \|a_d\| \leq a_M, \forall y \in \mathbb{R}^3$, the Lie derivative of V along (26)-(28) reads

$$\begin{aligned}\dot{V} &= e_v^\top(-\gamma_v(e_v) - \tilde{a}_d) + \tilde{e}_v^\top(-L\tilde{e}_v + \tilde{a}_d)\\ &+ \tilde{a}_d^\top\Gamma_d^{-1}\,\text{proj}(\hat{a}_d, -\Gamma_d(\tilde{e}_v - e_v)) \leq -e_v^\top\gamma_v(e_v) - \tilde{e}_v^\top L\tilde{e}_v\end{aligned} \tag{49}$$

$\forall x_v \in \mathcal{R}_v$. Given that $\dot{V} \leq 0\ \ \forall x_v \in \mathbb{R}^9$, one can conclude (global) stability of $x_v = 0$ by [21, Theorem 4.2]. As $\dot{V}$ is only negative semi-definite and we are dealing with an autonomous system, we can prove global attractivity of $x_v = 0$ (and therefore GAS) by relying on standard arguments based on LaSalle's invariance principle.

We then prove the UUB property in case $\dot{a}_d \neq 0$. Consider again the quadratic Lyapunov candidate (48). For $\dot{a}_d \neq 0$, the Lie derivative of V along (26)-(28) reads $\dot{V} = -e_v^\top\gamma_v(e_v) - \tilde{e}_v^\top L\tilde{e}_v + \tilde{a}_d^\top\Gamma_d^{-1}(\Gamma_d(\tilde{e}_v - e_v) + \text{proj}(\hat{a}_d, -\Gamma_d(\tilde{e}_v - e_v)) - \dot{a}_d)$. Thanks to the properties of the projection operator, which guarantees that $\|\hat{a}_d(t)\| \leq a_M + \varepsilon\ \ \forall \hat{a}_d(t_0) \leq a_M + \varepsilon$, $\dot{V}$ can be bounded as $\dot{V} \leq -e_v^\top\gamma_v(e_v) - \tilde{e}_v^\top L\tilde{e}_v + \tilde{a}_d^\top\Gamma_d^{-1}\dot{a}_d \leq -\alpha(\|\bar{x}_v\|) + \delta_1$, where $\bar{x}_v := [\,e_v^\top\ \tilde{e}_v^\top\,]^\top$, $\alpha(\cdot)$ is a class-$\mathcal{K}_\infty$ function and $\delta_1 := \frac{\bar{a}_M}{\lambda_m(\Gamma_d)}\dot{a}_M$, $\bar{a}_M := 2a_M + \varepsilon$. Then, $\dot{V} \leq -(1-\lambda)\alpha(\|\bar{x}_v\|)\ \ \forall \|\bar{x}_v\| \geq \alpha^{-1}\left(\frac{\delta_1}{\lambda}\right) =: \delta_2$ for any $\lambda \in (0,1)$. Picking $\delta := \sqrt{\bar{a}_M^2 + \delta_2^2}$, the sublevel set $\Omega_\delta := \{x_v \in \mathcal{R}_v : V(x_v) \leq \bar{\alpha}(\delta)\}$ is uniformly attractive from $\mathcal{R}_v$, proving UUB of solutions: there exists a time interval $\Delta t \geq 0$, independent of t_0, such that $\|x_v(t)\| \leq \underline{\alpha}^{-1}(\bar{\alpha}(\delta)) = \sqrt{\frac{\max\left(1,\lambda_m^{-1}(\Gamma_d)\right)}{\min\left(1,\lambda_M^{-1}(\Gamma_d)\right)}}\delta\ \ \forall t \geq t_0 + \Delta t$ for some $\lambda \in (0,1)$.

Proof of Theorem 1. The proof makes use of the results of [20], [29]: we show that the lower subsystem is strongly iISS with respect to e_v and then use the main result of [20] to conclude AS of the cascade connection.

Lemma 1: System (31), with $\gamma_q(e_p)$ defined as below (14), is strongly iISS with respect to e_v, *i.e.*, there exists $r_p > 0$, a class-$\mathcal{KL}_\infty$ function $\beta_p(\cdot,\cdot)$ and class-$\mathcal{K}_\infty$ functions $\mu_1(\cdot), \mu_2(\cdot), \mu_p(\cdot)$ such that $\forall t_0 \geq 0$, $\forall t \geq t_0$

$$\begin{aligned}&\|e_v\|_\infty < r_p \implies \|e_p(t)\| \leq \beta_p(\|e_p(t_0)\|, t - t_0) + \mu_p(\|e_v\|)\\ &\|e_p(t)\| \leq \beta_p(\|e_p(t_0)\|, t - t_0) + \mu_1\Big(\int_{t_0}^{t}\mu_2(\|e_v(s)\|)ds\Big).\end{aligned} \tag{50}$$

Proof of Lemma 1. Consider the Lyapunov function

$$V_p(e_p) := \int_0^{\|e_p^\perp\|} v_s \tanh\left(\frac{k_\perp}{v_s}s\right)ds + \int_0^{e_{p3}}\gamma_{q,3}(s)ds. \tag{51}$$

The Lie derivative of V_p along (31) is $\dot{V}_p(e_p) = -\gamma_q(e_p)^\top\gamma_q(e_p) + \gamma_q(e_p)^\top e_v$ $= -\|\gamma_q(e_p)\|^2 + \|\gamma_q(e_p)\|\|e_v\| \leq -\alpha_p(\|e_p\|) + b_p\|e_v\|$, $\forall e_p^\perp \neq 0$, where $\alpha(\cdot)$ is a class-$\mathcal{K}$ function with $\alpha(\infty) = \liminf_{\|e_p\|\to\infty}\|\gamma_q(e_p)\|^2 = v_a^2$, while $b_p := \sup_{e_p\in\mathbb{R}^3}(\|\gamma_q(e_p)\|) = \sqrt{2}v_s < \infty$ (assuming $v_a < v_s$). Then, leveraging [29, Theorem 1], the claim of the Lemma is proven with input threshold $r_p = \frac{v_a^2}{\sqrt{2}v_s}$. □

Combining strong iISS of the lower subsystem proved in Lemma 1 and GAS of the perturbing subsystem proved in Theorem 1, then the origin of the complete system is GAS, thanks to [20, Corollary 2].

In case the perturbing subsystem is only UUB, the cascade will be at most UUB, provided that the ultimate bound of the upper subsystem solutions is below a given value. The proof can be done by following the ideas in [20]. Specifically, given any $x_v(t_0) \in \mathcal{R}_v$ and bounded $a_d(t)$ with bounded derivative, the solution $x_v(t; x_v(t_0), t_0)$ to (32)-(34) is UUB, in particular, we have $\|e_v(t)\| \leq \sqrt{2\max\left(1,\lambda_m^{-1}(\Gamma_d)\right)}\alpha^{-1}\left(\frac{\delta}{\lambda}\right) =: b_v$ $\forall t \geq t_0 + \Delta t =: t_1$. By iISS of (31), the solution $e_p(t; e_p(t_0), t_0)$ will exist forward in time. Picking saturation levels v_a, v_s such that $r_p > b_v$, where r_p is the input threshold in (50), we have $\|e_p(t)\| \leq \beta_p(\|e_p(t_1)\|, t - t_1) + \mu_p(\|x_v\|_{[t_1,\infty)})\quad \forall t \geq t_1$ by small input ISS, which proves UUB of the overall cascade solutions.

Proof of Theorem 2. The proof of AS for $\dot{a}_d = 0$ follows from [30, Theorem 84] using cascade arguments, given that $R_e = I_3$ is LAS for the upper subsystem (35) and $(e_p, e_v, \tilde{e}_v, \hat{a}_d) = (0,0,0,0,a_d)$ is LAS for the lower subsystem (36)-(39). In case $\dot{a}_d \neq 0$, we could rely on local ISS arguments coming again from local AS properties of the unforced systems. Nonetheless, we quantitatively characterize this result by computing the input restriction and ultimate bound of solutions of (36)-(39) then use Proposition 2 to conclude UUB of the cascade. To this aim, we refer again to the Lyapunov function (48) and compute its Lie derivative along (35)-(39):

$$\begin{aligned}\dot{V} &= -e_v^\top\gamma_v(e_v) - \tilde{e}_v^\top L\tilde{e}_v + \tilde{a}_d^\top\Gamma_d^{-1}(\Gamma_d(\tilde{e}_v - e_v)\\ &+ \text{proj}(\hat{a}_d, -\Gamma_d(\tilde{e}_v - e_v)) - \dot{a}_d) + (e_v - \tilde{e}_v)^\top\Delta a(R_p, R_e, u_a).\end{aligned} \tag{52}$$

Since $\|u_a\| \leq c_1 + c_2\|e_v\|$, $\dot{V} \leq -e_v^\top\gamma_v(e_v) - \tilde{e}_v^\top L\tilde{e}_v + \delta_1 + \|\bar{x}_v\|\|R_e\|_{\text{SO}(3)}\|u_a\| \leq -\alpha(\|\bar{x}_v\|) + \delta_1 + 2\sqrt{2}(c_1\|\bar{x}_v\| + c_2\|\bar{x}_v\|^2)\|R_e\|_{\text{SO}(3)}$. In a bounded domain $\|\bar{x}_v\| < s$, $\|\hat{a}_d\| \leq a_M$ the above bound can be written as $\dot{V} \leq -\alpha(\|\bar{x}_v\|) + \lambda_s(\|R_e\|_{\text{SO}(3)})$, where $\lambda_s(\|R_e\|_{\text{SO}(3)}) := \delta_1 + 2\sqrt{2}(c_1 s + c_2 s^2)\|R_e\|_{\text{SO}(3)}$. Hence, we have $\dot{V} \leq -(1-\lambda)\alpha(\|\bar{x}_v\|)\ \ \forall \|\bar{x}_v\| \geq \rho(\|R_e\|_{\text{SO}(3)}) := \alpha^{-1}\left(\frac{\lambda_s(\|R_e\|_{\text{SO}(3)})}{\lambda}\right)$, where $\rho(\cdot)$ is strictly increasing with $\rho(0) = \delta_1$, and $\lambda \in (0,1)$. Then, there exist a large enough s and a small enough attitude error $\sup_{t_0\leq\tau\leq t}\|R_e(\tau)\|_{\text{SO}(3)} < r_a$ such that $\sqrt{\rho(\sup_{t_0\leq\tau\leq t}\|R_e(\tau)\|_{\text{SO}(3)})\|)^2 + \bar{a}_M^2} < \overline{\alpha}^{-1}(\underline{\alpha}(s))$. Therefore, the solutions starting from $\|x_v(t_0)\| \leq \sqrt{\frac{\min\left(1,\lambda_M^{-1}(\Gamma_d)\right)}{\max\left(1,\lambda_m^{-1}(\Gamma_d)\right)}}s$ are UUB with ultimate bound $\|x_v(t)\| \leq \sqrt{\frac{\max\left(1,\lambda_m^{-1}(\Gamma_d)\right)}{\min\left(1,\lambda_M^{-1}(\Gamma_d)\right)}}\alpha^{-1}\left(\frac{\rho(\sup_{t_0\leq\tau\leq t}\|R_e(\tau)\|_{\text{SO}(3)})}{\lambda}\right)$, $\sup_{t_0\leq\tau\leq t}\|R_e(\tau)\|_{\text{SO}(3)} < r_a$. Using the results of Proposition 2, the proof is concluded.

REFERENCES

[1] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, Minnesota, USA, 2012, pp. 971–976.

[2] P. Vlantis, P. Marantos, C. Bechlioulis, and K. Kyriakopoulos, "Quadrotor landing on an inclined platform of a moving ground vehicle," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, pp. 2202–2207.

[3] J. Kim, Y. Jung, D. D. Lee, and D. H. Shim, "Landing control on a mobile platform for multi-copters using an omnidirectional image sensor," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 529–541, 2016.

[4] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Shanghai, China, 2017, pp. 200–207.

[5] A. Borowczyk, N. Tien, A. Nguyen, D. Nguyen, D. Saussie, and J. Le Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *IFAC Papers Online*, vol. 50, no. 1, pp. 10 488–10 494, 2017.

[6] T. Báča, P. Štěpán, V. Spurný, D. Hert, R. Pěnička, M. Saska, J. Thomas, G. Loianno, and V. Kumar, "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 5, pp. 874–891, 2019.

[7] B. Hu and S. Mishra, "Time-optimal trajectory generation for landing a quadrotor onto a moving platform," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 585–596, 2019.

[8] K. Guo, P. Tang, H. Wang, D. Lin, and X. Cui, "Autonomous landing of a quadrotor on a moving platform via model predictive control," *MDPI Aerospace*, vol. 9, no. 1, 2022.

[9] G. Niu, Q. Yang, Y. Gao, and M. Pun, "Vision-based autonomous landing for unmanned aerial and ground vehicles cooperative systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6234–6241, 2022.

[10] J. Lim, T. Lee, S. Pyo, J. Lee, J. Kim, and J. Lee, "Hemispherical InfraRed (IR) marker for reliable detection for autonomous landing on a moving ground vehicle from various altitude angles," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 485–492, 2022.

[11] J. Lin, Y. Wang, Z. Miao, H. Zhong, and R. Fierro, "Low-complexity control for vision-based landing of quadrotor UAV on unknown moving platform," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5348–5358, 2022.

[12] N. Xuan-Mung, N. P. Nguyen, T. Nguyen, D. B. Pham, M. T. Vu, H. L. N. N. Thanh, and S. K. Hong, "Quadcopter precision landing on moving targets via disturbance observer-based controller and autonomous landing planner," *IEEE Access*, vol. 10, pp. 83 580–83 590, 2022.

[13] K. Xia, M. Shin, W. Chung, M. Kim, S. Lee, and H. Son, "Landing a quadrotor UAV on a moving platform with sway motion using robust control," *Control Engineering Practice*, vol. 128, p. 105288, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066122001423

[14] N. J. Sanket, C. D. Singh, C. M. Parameshwara, C. Fermüller, G. C. H. E. de Croon, and Y. Aloimonos, "EVPropNet: Detecting drones by finding propellers for mid-air landing and following," in *2021 Robotics. Science and Systems (RSS)*, Virtual, 2021.

[15] K. P. Jain and M. W. Mueller, "Flying batteries: In-flight battery switching to increase multirotor flight time," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3510–3516.

[16] Z. W. Lee, W. H. Chin, and H. W. Ho, "Air-to-air micro air vehicle interceptor with an embedded mechanism and deep learning," *Aerospace Science and Technology*, vol. 135, p. 108192, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963823000895

[17] G. Gozzini, D. Invernizzi, S. Panza, M. Giurato, and M. Lovera, "Air-to-air automatic landing of unmanned aerial vehicles: A quasi time-optimal hybrid strategy," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 692–697, 2020.

[18] T. Fernando, J. Chandiramani, T. Lee, and H. Gutierrezee, "Robust adaptive geometric tracking controls on SO(3) with an application to the attitude dynamics of a quadrotor UAV," in *50th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2011, pp. 7380–7385.

[19] D. Invernizzi, M. Lovera, and L. Zaccarian, "Hierarchical dynamic control for robust attitude tracking," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6171–6176, 2020, 21st IFAC World Congress.

[20] A. Chaillet, D. Angeli, and H. Ito, "Strong iISS is preserved under cascade interconnection," *Automatica*, vol. 50, no. 9, pp. 2424–2427, 2014.

[21] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[22] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1837–1853, 2009.

[23] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, "Fast, autonomous flight in GPS-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.

[24] E. Lavretsky and K. A. Wise, "Robust and adaptive control: With aerospace applications," *Advanced textbooks in control and signal processing. London and New York: Springer*, 2013.

[25] "OpenCV," http://https://opencv.org/, Access date: 10/11/2021.

[26] A. Santamaria-Navarro, G. Loianno, J. Solà, V. Kumar, and J. Andrade-Cetto, "Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors," *Autonomous robots*, vol. 42, no. 6, pp. 1263–1280, 2018.

[27] M. Wang, S. Berkane, and A. Tayebi, "Nonlinear observers design for vision-aided inertial navigation systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1853–1868, 2022.

[28] ANT-X website, https://antx.it/.

[29] A. Chaillet, D. Angeli, and H. Ito, "Combining iISS and ISS With Respect to Small Inputs: The Strong iISS Property," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2518–2524, 2014.

[30] M. Vidyasagar, *Nonlinear Systems Analysis (2nd Ed.)*. USA: Prentice-Hall, Inc., 1993.

Gabriele Roggi holds a M.Sc. in Aeronautical Engineering and a B.Sc. in Aerospace Engineering both from Politecnico di Milano. He received the Ph.D. degree at the Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano. His research activity involves the design, implementation and experimental validation of guidance, navigation and control algorithms for autonomous multirotor UAVs.

Giovanni Gozzini holds a M.Sc. in Aeronautical Engineering and a B.Sc. in Aerospace Engineering both from Politecnico di Milano. He is currently a Ph.D. student at the Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano. His research activity involves the design, implementation and experimental validation of guidance, navigation and control algorithms for multirotor UAVs and hybrid control with applications to aerospace systems.

Davide Invernizzi (Member, IEEE) graduated in space engineering from the Politecnico di Milano, Milan, Italy, in 2014, and received the Ph.D. degree (cum laude) in aerospace engineering from the Politecnico di Milano in 2018. He is currently an Assistant Professor with the Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano. His research interest includes nonlinear control with applications to aerospace systems.

Marco Lovera (Member, IEEE) received the M.Sc. in electronics engineering at Politecnico di Milano, Milano, Italy, 1993 and the Ph.D. degree in Computer Science and Control Engineering at Politecnico di Milano, Milano, Italy, 1998. After a one-year period in industry he joined the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, in 1999. Since 2015, he is with the Dipartimento di Scienze e Tecnologie Aerospaziali of the Politecnico di Milano, where he leads the Aerospace Systems and Control Laboratory (ASCL).