A General Control Architecture for Visual Servoing and Physical Interaction Tasks for Fully-actuated Aerial Vehicles

Gianluca Corsini, Martin Jacquet, Antonio Enrique Jimenez-Cano, Amr Afifi, Daniel Sidobre, Antonio Franchi

A General Control Architecture for Visual Servoing and Physical Interaction Tasks for Fully-actuated Aerial Vehicles

Gianluca Corsini[1⋆], Martin Jacquet[1⋆], Antonio Enrique Jimenez-Cano[1], Amr Afifi[2], Daniel Sidobre[1], Antonio Franchi[2,1]

***Abstract*— In this paper, we present a general control architecture that allows fully-actuated aerial robots to autonomously accomplish tasks that require both perception and physical interaction with the external environment. We integrate the novel *Flying End-Effector* paradigm and a Hybrid Visual Servoing (HVS) scheme to design a general control architecture for fully-actuated aerial robots. Thanks to the proposed solution, a fully-actuated aerial robot can autonomously accomplish tasks that require both perception and physical interaction without resorting to any external force/torque sensor. The control architecture is entirely described, features a wrench observer and an admittance filter, and is subsequently validated on real experiments. The code for the proposed control architecture is provided open-source.**

SUPPLEMENTARY MATERIAL

The code corresponding to the presented experiments and the material to run some simulations are available at: `https://redmine.laas.fr/projects/visual-physical-control-architecture`

I. INTRODUCTION

In the last decades, Unmanned Aerial Vehicles (UAVs) have gained a lot of popularity, not only in the research community but also among industries, thanks to their maneuverability and agility, their versatility and the possibility to be assembled with off-the-shelf, low-cost, and easily gatherable components. Usually, most of the applications where UAVs have been deployed involve contact-less missions, e.g. environmental monitoring, photography and mapping, search-and-rescue operations.

Given the recent advancements in control, sensing, and actuation of UAVs, the robotic community shifted its interest to contact-based applications, like payload transportation [1], object grasping [2], pipe and surface inspection [3]. Therefore, to enhance aerial physical interaction, aerial robots have been equipped with either a rigid tool [4], or n-Degree-of-Freedoms (DoFs) robotic arms [5]–[7]. Typically, aerial platforms are under-actuated and the use of a robotic arm aims at overcoming such under-actuation and increasing the dexterity of the platform. However, this solution is not free of drawbacks, e.g., the weight of the attached manipulator arm decreases the available payload and reduces the flight time, increasing at the same time the overall mechanical complexity. Moreover, due to the dynamical/inertial coupling between the arm and the aerial robot, the control problem becomes more complex [8].

On the other hand, the use of a rigid tool fixedly attached to the airframe of under-actuated platforms leads to the impossibility to have full 6D control over the dynamics of the end-effector. In fact, under-actuated UAVs equipped with collinear propellers can exert forces only along one direction due to the dynamical coupling between the rotational and translation dynamics. This forces them to re-orient their body to continuously steer the actuation force towards the desired direction, which consequently limits their dexterity and introduces stability issues [9].

To overcome the aforementioned drawbacks, a novel solution is offered by *The Flying End-Effector* paradigm, presented in [8]. The authors propose to adopt robots equipped with non-collinear fixedly-tilted propellers (i.e., each one having a different orientation), which leads to a decoupled rotational and translation dynamics. Therefore, these robots are shown to be natural candidates for physical interaction with the environment, thanks to their intrinsic ability to exert forces and moments in any direction independently. This has largely encouraged the research community to adopt fully-actuated aerial robots in heterogeneous contact-based applications [4], [10]. In order to physically interact with the environment, these robots have been equipped with perceptive sensors, like cameras and infrared sensors, and consequently, several perception-based control strategies and algorithms have been proposed [11], [12]. Among all the sensor possibilities, monocular and stereo vision cameras have become the de facto standard in the onboard sensing equipment of every aerial robot, thanks to their lightweight, affordable price, compact dimensions, and the availability of numerous open-source video-processing software packages. The wide use of these sensors encouraged the development of different camera-based control strategies, which exploit the image feedback to drive the robot towards the desired goal and allow them to accomplish the mission.

One classical approach is *Visual Servoing*, which can be divided into two main classes [13]: *Image-Based Visual Servoing* (shortly IBVS) and *Position-Based Visual Servoing* (PBVS). IBVS uses features directly available from the image, thus it minimizes the tracking error in the image

⋆The two authors contributed equally to the work presented in this paper.

[1]LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS, Toulouse, France, `gianluca.corsini@laas.fr`, `martin.jacquet@laas.fr`, `antonio-enrique.jimenez-cano@laas.fr`, `daniel.sidobre@laas.fr`, `antonio.franchi@laas.fr`

[2]Robotics and Mechatronics lab, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede, The Netherlands `a.n.m.g.afifi@utwente.nl`, `a.franchi@utwente.nl`

This work was partially funded by the ANR, under the Projects ANR-18-CE33-0001 'The flying coworker' and ANR-17-CE33-0007 'MuRoPhen', and European Commission project H2020 AERIAL-CORE (EC 871479).

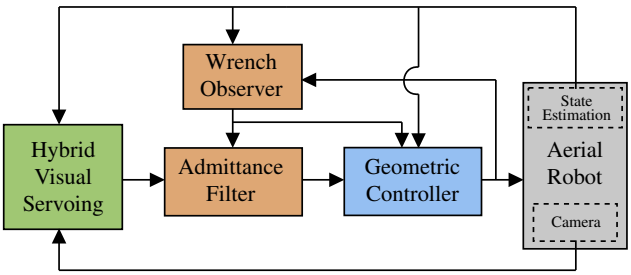

Fig. 1: Generic control architecture for vision-based physical interaction with fully-actuated platforms. In green the vision-based control, in orange the control in charge of the physical interaction, in blue the geometric controller, and in gray the aerial robotic platform.

plane of the camera, while PBVS estimates the 3D pose of the features in the cartesian camera frame and minimizes the tracking error with respect to (w.r.t.) a reference position. While IBVS uses the image features directly, it is considered a more difficult technique due to the complexity of the kinematic relationship between the image features and the motion of the camera. Conversely, PBVS is sensitive to image measurement and kinematic model errors [14]. It also requires a broad knowledge of the camera calibration, the full pose estimation of the tracked feature, and hence it implies the knowledge of its 3D model.

An intermediate solution is offered by *Hybrid Visual Servoing* (HVS) schemes [15], [16], which use features both from the image space and the cartesian space. Those schemes do not require either a 3D model of the tracked object or the precise camera intrinsic and extrinsic calibrations. In addition, they rely on the estimation of the camera displacement to reach the desired view of the object, which is more convenient and robust than the classical IBVS.

Contribution. In this paper, we combine an HVS scheme with an admittance filter for compliancy at the end-effector and a wrench observer for estimating the contact forces and moments exchanged during the interaction phase. We propose an autonomous general control scheme for fully-actuated UAVs in physical interaction tasks. In particular, we apply this control strategy to a fully-actuated UAV performing a pick-and-place operation, where several bricks (differently located and oriented in the workspace) must be picked and placed at another location. We show through real experiments how our approach can efficiently and autonomously lead the robot to complete the task without any human intervention.

The paper outline is structured as follows. First, in Sec. II, the generic model of a fully-actuated platform is presented. Secondly, the control strategy is detailed in Sec. III, where the geometric control and the HVS scheme are described, and the physical interaction estimation and the compliance are discussed. Following, in Sec. IV, our experimental framework is presented, and the results obtained in the conducted experiments are shown. Finally, in Sec. V, final conclusions are drawn.

II. Modeling

The fully-actuated UAV is modeled as a Generically Tilted Multi-Rotor (GTMR), i.e., as a rigid body of mass $m \in \mathbb{R}^+$, positive-definite inertia $\mathbf{J} \in \mathbb{R}^{3\times3}$, actuated by $n \in \mathbb{N}^+$ propellers. In order to be fully-actuated, a GTMR must have at least $n = 6$ propellers, whose axes can be arbitrary oriented w.r.t. the UAV frame but cannot be collinear [17].

The world inertial frame and the rigid-body frame (whose origin coincides with the center of mass (C.o.M.) of the multi-rotor) are denoted by $\mathcal{F}_W = O_W, \{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$, $\mathcal{F}_B = O_B, \{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$, respectively. The position of O_B – the origin of $\mathcal{F}_B$ – in $\mathcal{F}_W$ is denoted by ${}^W\mathbf{p}_B$, while ${}^W\mathbf{R}_B$ is the rotation matrix that represents the orientation of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$; similarly for all the other frame pairs. Two other frames of interest, the camera and end-effector frames, are respectively denoted by $\mathcal{F}_C$ and $\mathcal{F}_E$. Finally, we denote ${}^B\boldsymbol{\omega}_B$ the angular velocity of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$, expressed in $\mathcal{F}_B$. The orientation kinematics is then expressed as

$$ {}^W\dot{\mathbf{R}}_B = {}^W\mathbf{R}_B [{}^W\boldsymbol{\omega}_B]_\times, \tag{1} $$

where ${}^W\dot{\mathbf{R}}_B$ is the time-derivative of ${}^W\mathbf{R}_B$, $[\cdot]_\times$ denotes the skew operator, and ${}^W\boldsymbol{\omega}_B$ is the angular velocity of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$, expressed in $\mathcal{F}_W$. The translational kinematics is expressed as

$$ {}^W\mathbf{v}_B = {}^W\dot{\mathbf{p}}_B, \tag{2} $$

where ${}^W\mathbf{v}_B$ is the velocity of O_B expressed in $\mathcal{F}_W$.

The dynamic equation of the system is written as

$$ \begin{bmatrix} m\,{}^W\ddot{\mathbf{p}}_B \\ \mathbf{J}\,{}^B\dot{\boldsymbol{\omega}}_B \end{bmatrix} = \begin{bmatrix} -mg\mathbf{z}_W \\ -{}^B\boldsymbol{\omega}_B \times \mathbf{J}\,{}^B\boldsymbol{\omega}_B \end{bmatrix} + \begin{bmatrix} {}^W\mathbf{R}_B & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{f}_B \\ {}^B\boldsymbol{\tau}_B \end{bmatrix} + \begin{bmatrix} {}^W\mathbf{R}_E & \mathbf{O}_3 \\ [{}^B\mathbf{p}_E]_\times\,{}^B\mathbf{R}_E & {}^B\mathbf{R}_E \end{bmatrix} \begin{bmatrix} {}^E\mathbf{f}_E \\ {}^E\boldsymbol{\tau}_E \end{bmatrix}, \tag{3} $$

where ${}^W\ddot{\mathbf{p}}_B$ and ${}^B\dot{\boldsymbol{\omega}}_B$ are the linear and angular accelerations of the platform, respectively in $\mathcal{F}_W$ and $\mathcal{F}_B$, $g \in \mathbb{R}$ is the gravity, $\mathbf{O}_3 \in \mathbb{R}^{3\times3}$ and $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ are the zero and identity matrices, ${}^W\mathbf{R}_E$ and ${}^B\mathbf{R}_E$ are the orientation of $\mathcal{F}_E$ w.r.t. $\mathcal{F}_W$ and $\mathcal{F}_B$, while ${}^B\mathbf{p}_E$ the position of O_E expressed in $\mathcal{F}_B$. The vectors ${}^E\mathbf{f}_E \in \mathbb{R}^3$ and ${}^E\boldsymbol{\tau}_E \in \mathbb{R}^3$ are the external forces and torques applied to the end-effector and expressed in $\mathcal{F}_E$, while ${}^B\mathbf{f}_B = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^\top \in \mathbb{R}^3$ and ${}^B\boldsymbol{\tau}_B = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^\top \in \mathbb{R}^3$ are the forces and moments applied to the C.o.M. of the platform expressed in $\mathcal{F}_B$. The body wrench (${}^B\mathbf{f}_B$ and ${}^B\boldsymbol{\tau}_B$) is mapped to the vector of the thrusts exerted by the n propellers ($\boldsymbol{\gamma} \in \mathbb{R}^n$) by the *force/torque allocation matrix* $\mathbf{G} \in \mathbb{R}^{6\times n}$, as shown in [18], following the relation

$$ \begin{bmatrix} {}^B\mathbf{f}_B \\ {}^B\boldsymbol{\tau}_B \end{bmatrix} = \mathbf{G}\boldsymbol{\gamma}. \tag{4} $$

We note that to allow full actuation, the matrix $\mathbf{G}$ has to be full ranked, and this is achieved, as mentioned before, by having non-collinear propellers, to exert a total force not collinear with $\mathbf{z}_B$ (but, rather, in a conic or cylindrical shape

around this axis [17]). Moreover, fully-actuated platforms cannot usually exert an arbitrary large force in any direction. Indeed, they are commonly characterized by a principal axis along which most of the thrust can be produced, while only a fraction of it can be exerted along the body lateral directions (i.e., along $\mathbf{x}_B$ and $\mathbf{y}_B$). Therefore, fully-actuated UAVs are also defined as *Laterally-Bounded*, and thus the following constraint must be satisfied:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} \in \mathscr{U}_{x,y} \subset \mathbb{R}^2, \tag{5}$$

where the authors in [17] provide several definitions of $\mathscr{U}_{x,y}$ according to different types of UAVs.

Furthermore, the camera to observe the surrounding environment and the end-effector to perform physical interaction are both modeled as punctual devices rigidly attached to the body. Hence their respective transformations $({}^B\mathbf{p}_C, {}^B\mathbf{R}_C)$ and $({}^B\mathbf{p}_E, {}^B\mathbf{R}_E)$ are assumed known and constant over time.

III. Control Architecture

Numerous applications in aerial robotics require visual awareness and physical interaction with the environment, even though different tasks or scenarios will require different types of motions. Applications of such nature will only differ in the choice of visual features needed for the vision-based control and the algorithmic procedure to fulfill the objective (e.g., the state machine used as the autonomous supervisor). However, the UAV geometric control, the vision-based motion generation, and the compliancy between the end-effector and the environment can be abstracted to a generic framework. In particular, this versatility is allowed by the use of a fully-actuated aerial platform that allows control of the 6D pose or the 6D interaction wrench when in contact.

Fig. 1 presents the general control architecture for such vision-based physical interaction tasks using a flying end-effector. A reference trajectory is produced by the HVS to drive the UAV toward the objective. This trajectory, before reaching the pose controller, is filtered by an admittance filter, which also receives the observed external wrench at the end-effector. Finally, the filtered reference trajectory is given to the geometric controller, along with the external wrench to be compensated. In the following subsections, we detail the formulation of each block presented in Fig. 1.

A. Geometric Controller

The controller used to stabilize the dynamics of the fully-actuated robot is derived from our previous work [17]. We refer the interested reader to the original work for its complete derivation and theoretical proof, while in this section, we report the main results of that control scheme.

This controller ensures, in nominal conditions, the tracking of a 6D pose (position ${}^W\mathbf{p}_B^r$ plus orientation ${}^W\mathbf{R}_B^r$) reference trajectory for a generic fully-actuated platform. As shown in Fig. 2, this controller exploits a cascade structure. Indeed, it first computes the necessary thrust force ${}^W\mathbf{f}_B^r \in \mathbb{R}^3$ to track the desired position reference (composed by the position,

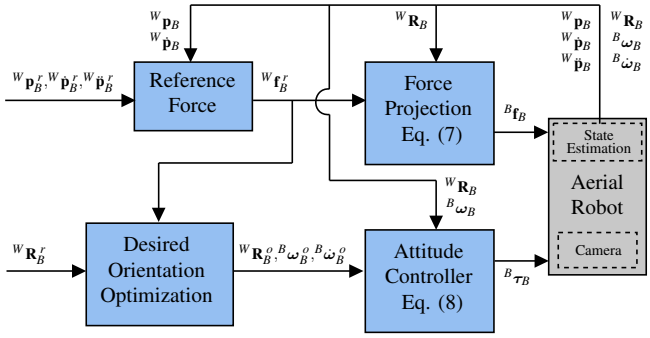

Fig. 2: Schematic representation of the geometric controller adopted to stabilize the translational and rotational dynamics of the fully-actuated UAV.

linear velocity and acceleration ${}^W\mathbf{p}_B^r$, ${}^W\dot{\mathbf{p}}_B^r$, and ${}^W\ddot{\mathbf{p}}_B^r$), by exploiting the translational part of the dynamic model Eq. (3) as

$${}^W\mathbf{f}_B^r = m\,{}^W\ddot{\mathbf{p}}_B^r + mg\mathbf{z}_W - \mathbf{K}_\mathbf{p}\mathbf{e}_\mathbf{p} - \mathbf{K}_\mathbf{v}\mathbf{e}_\mathbf{v} - \mathbf{K}_i \int_0^t \mathbf{e}_\mathbf{p}(\tau)\mathrm{d}\tau, \tag{6}$$

where the reference positional kinematical quantities ${}^W\mathbf{p}_B^r, {}^W\dot{\mathbf{p}}_B^r, {}^W\ddot{\mathbf{p}}_B^r$ are used with the feedback variables ${}^W\mathbf{p}_B, {}^W\dot{\mathbf{p}}_B$ to compute the kinematic error quantities $\mathbf{e}_\mathbf{p} = {}^W\mathbf{p}_B - {}^W\mathbf{p}_B^r$ and $\mathbf{e}_\mathbf{v} = {}^W\dot{\mathbf{p}}_B - {}^W\dot{\mathbf{p}}_B^r$. Matrices $\mathbf{K}_\mathbf{p} \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_\mathbf{v} \in \mathbb{R}^{3\times3}$ are diagonal positive-definite gain matrices. Besides, compared to [17], we add an integral action on the translational error, with its diagonal positive-definite gain matrix $\mathbf{K}_i \in \mathbb{R}^{3\times3}$, and similarly also to the orientation part, as discussed later. However, this computed force vector ${}^W\mathbf{f}_B^r$ is the force, expressed in $\mathscr{F}_W$, that ideally one would like to apply to the aerial vehicle C.o.M. if the system would not be subject to any lateral input bound and be completely fully-actuated. Since this force is related to the body force ${}^B\mathbf{f}_B$ (generated by the robot's actuators) through the orientation of the platform ${}^W\mathbf{R}_B$, it can be proven that there is a finite non-empty set of orientations that allow tracking that force without violating the actuators' lateral bounds. However, this set may or may not contain the reference orientation ${}^W\mathbf{R}_B^r$ that one would like the aerial robot to track. Therefore, at each time t, the controller selects an orientation ${}^W\mathbf{R}_B^o \in SO(3)$ that 1) belongs to the set of orientations allowing to track the computed reference force ${}^W\mathbf{f}_B^r$ (which in turn allows following the reference position trajectory), and 2) minimizes a certain cost function w.r.t. the input reference orientation. In our previous work, it is proven that if the reference position ${}^W\mathbf{p}_B^r$ leads to a feasible ${}^W\mathbf{f}_B^r$ w.r.t. the actuators' constraints, then the selected orientation ${}^W\mathbf{R}_B^o$ will exponentially converge to the reference orientation ${}^W\mathbf{R}_B^r$. If not, the closest desired orientation to the reference one will be selected, which allows to track the reference position trajectory (thus the reference force) and satisfies the input bounds. In the optimization step where the new orientation is computed, also a new angular velocity ${}^B\boldsymbol{\omega}_B^o$ and acceleration ${}^B\dot{\boldsymbol{\omega}}_B^o$ can be computed by, for instance, adding a regularization term in the cost function, as discussed in [17].

After that, the following control laws are used to obtain

the inputs to apply at the C.o.M. of the platform

$$
\begin{aligned}
{}^{B}\mathbf{f}_B =& \text{sat}_{\mathscr{U}_{x,y}}\left(\left({}^{W}\mathbf{f}_B^{r\top}\,{}^{W}\mathbf{R}_B\mathbf{x}_B\right)\mathbf{x}_B+\left({}^{W}\mathbf{f}_B^{r\top}\,{}^{W}\mathbf{R}_B\mathbf{y}_B\right)\mathbf{y}_B\right)\\
&+\left({}^{W}\mathbf{f}_B^{r\top}\,{}^{W}\mathbf{R}_B\mathbf{z}_B\right)\mathbf{z}_B, \qquad (7)
\end{aligned}
$$

$$
\begin{aligned}
{}^{B}\boldsymbol{\tau}_B =& {}^{B}\boldsymbol{\omega}_B\times\mathbf{J}\,{}^{B}\boldsymbol{\omega}_B-\mathbf{K}_{\mathbf{R}}\mathbf{e}_{\mathbf{R}}-\mathbf{K}_{\boldsymbol{\omega}}\mathbf{e}_{\boldsymbol{\omega}}-\mathbf{K}_I\mathbf{e}_I\\
&-\mathbf{J}\left([{}^{B}\boldsymbol{\omega}_B]_\times\,{}^{W}\mathbf{R}_B^{\top}\,{}^{W}\mathbf{R}_B^{o}\,{}^{B}\boldsymbol{\omega}_B^{o}-{}^{W}\mathbf{R}_B^{\top}\,{}^{W}\mathbf{R}_B^{o}\,{}^{B}\dot{\boldsymbol{\omega}}_B^{o}\right), \qquad (8)
\end{aligned}
$$

where $\text{sat}_{\mathscr{U}_{x,y}}(\cdot)$ is a saturation operator which guarantees that the output vector of the lateral forces belongs to $\mathscr{U}_{x,y}$, $\mathbf{e}_{\mathbf{R}}=\frac{1}{2}\left({}^{W}\mathbf{R}_B^{o\top}\,{}^{W}\mathbf{R}_B-{}^{W}\mathbf{R}_B^{\top}\,{}^{W}\mathbf{R}_B^{o}\right)^{\vee}$ is the rotational error, $(\cdot)^{\vee}$ is the inverse map of the skew operator $[\cdot]_\times$, and $\mathbf{e}_{\boldsymbol{\omega}}={}^{B}\boldsymbol{\omega}_B-{}^{W}\mathbf{R}_B{}^{\top}\,{}^{W}\mathbf{R}_B^{o}\,{}^{B}\boldsymbol{\omega}_B^{o}$ is the angular velocity error. The matrices $\mathbf{K}_{\mathbf{R}}\in\mathbb{R}^{3\times3}$, $\mathbf{K}_{\boldsymbol{\omega}}\in\mathbb{R}^{3\times3}$ and $\mathbf{K}_I\in\mathbb{R}^{3\times3}$ are again diagonal positive-definite gain matrices. For the integral error $\mathbf{e}_I$, we take inspiration from [19], which defines it as

$$
\mathbf{e}_I=\int_0^t\mathbf{e}_{\boldsymbol{\omega}}(\tau)+c_2\mathbf{e}_{\mathbf{R}}(\tau)\mathrm{d}\tau,\quad c_2\in\mathbb{R}^{+}. \qquad (9)
$$

B. Vision-based Control

In order to autonomously generate the motion toward the object, we propose to use a visual servoing scheme. The concept has already been recalled in Sec. I. In this section, we will detail the equations of the HVS scheme used to generate the velocity commands of the platform, after [20].

First, the visual feature vector $\mathbf{s}\in\mathbb{R}^6$ and its reference $\mathbf{s}^r\in\mathbb{R}^6$ are chosen such that the tracking error can be defined as

$$
\mathbf{e}=\mathbf{s}-\mathbf{s}^r. \qquad (10)
$$

In a classical HVS scheme, the feature vector is defined as

$$
\mathbf{s}=(\mathbf{x},\ \log Z,\ \theta\mathbf{u})\in\mathbb{R}^6, \qquad (11)
$$

where $\mathbf{x}\in\mathbb{R}^2$ is the (x,y) position of the target and can be defined either in camera frame, normalized camera coordinates, or pixel coordinates, while $Z\in\mathbb{R}^{+}$ is the position of the feature along the principal axis of the camera $\mathbf{z}_C$. $\theta\mathbf{u}\in\mathbb{R}^3$ is the angle-axis representation of the orientation error. In the following, we present the equations for $\mathbf{x}$ being the normalized coordinates of the detected feature. The reference vector $\mathbf{s}^r$ has to be chosen in order to align the end-effector with its target, hence is task-specific.

The velocity control in $\mathscr{F}_C$ is designed to nullify $\mathbf{e}$. We typically impose an exponentially decreasing rate on $\mathbf{e}$

$$
\dot{\mathbf{e}}=-\lambda\mathbf{e},\quad \lambda\in\mathbb{R}^{+}. \qquad (12)
$$

Thus, the interaction matrices $\mathbf{L}_{\mathbf{v}}\in\mathbb{R}^{3\times3}$, $\mathbf{L}_{\boldsymbol{\omega}}\in\mathbb{R}^{3\times3}$ and $\mathbf{L}_{\theta\mathbf{u}}\in\mathbb{R}^{3\times3}$ are defined such that

$$
\dot{\mathbf{e}}=\begin{bmatrix}\mathbf{L}_{\mathbf{v}} & \mathbf{L}_{\boldsymbol{\omega}}\\ \mathbf{O}_3 & \mathbf{L}_{\theta\mathbf{u}}\end{bmatrix}\begin{bmatrix}{}^{C}\mathbf{v}_C\\ {}^{C}\boldsymbol{\omega}_C\end{bmatrix}, \qquad (13)
$$

where ${}^{C}\mathbf{v}_C$ and ${}^{C}\boldsymbol{\omega}_C$ are the desired linear and angular velocities for the camera, expressed in $\mathscr{F}_C$.

Then, the angular velocity control scheme is defined, as in [13]. The orientation interaction matrix $\mathbf{L}_{\theta\mathbf{u}}$ is defined as

$$
\mathbf{L}_{\theta\mathbf{u}}=\mathbf{I}_3-\frac{\theta}{2}[\mathbf{u}]_\times+\left(1-\frac{\sin\theta}{\text{sinc}^2\frac{\theta}{2}}\right)[\mathbf{u}]_\times^2, \qquad (14)
$$

where sinc is the sinus cardinal operator. We note here that since the determinant of the above matrix is given by

$$
\det(\mathbf{L}_{\theta\mathbf{u}})=\frac{1}{\text{sinc}^2\frac{\theta}{2}}, \qquad (15)
$$

$\mathbf{L}_{\theta\mathbf{u}}$ is singular only for $\theta=2k\pi,\ k\neq0$, which is out of the potential workspace, since $\theta\in[0,\pi]$.

Putting together Eq. (12), (13) and (15), we have

$$
{}^{C}\boldsymbol{\omega}_C=-\lambda\mathbf{L}_{\theta\mathbf{u}}^{-1}\theta\mathbf{u}. \qquad (16)
$$

We can now define the linear velocity control scheme, following [20]. We define $\mathbf{L}_{\mathbf{v}}$ and $\mathbf{L}_{\boldsymbol{\omega}}$ as

$$
\mathbf{L}_{\mathbf{v}}=\frac{1}{\rho_Z Z^r}\begin{bmatrix}-1 & 0 & x\\ 0 & -1 & y\\ 0 & 0 & -1\end{bmatrix}, \qquad (17)
$$

$$
\mathbf{L}_{\boldsymbol{\omega}}=\begin{bmatrix}xy & -(1+x^2) & y\\ 1+y^2 & -xy & -x\\ -y & x & 0\end{bmatrix}, \qquad (18)
$$

where $\rho_Z=Z/Z^r$, $Z^r\in\mathbb{R}^{+}$ being the reference for Z, and $x\in\mathbb{R}$, $y\in\mathbb{R}$ are the normalized coordinates of the detected feature. As noted in [20], ρ_Z can be obtained from a partial pose estimation scheme. It makes the HVS scheme more generic than PBVS since the estimation process is much lighter. We also note that $\mathbf{L}_{\mathbf{v}}$ is singular only when $Z\to\infty$, making the inversion always feasible.

Putting together Eq. (12), (13), (17) and (18), we obtain

$$
{}^{C}\mathbf{v}_C=-\mathbf{L}_{\mathbf{v}}^{-1}\left(\lambda\begin{bmatrix}x\\ y\\ \log Z\end{bmatrix}+\mathbf{L}_{\boldsymbol{\omega}}\,{}^{C}\boldsymbol{\omega}_C\right). \qquad (19)
$$

We can now define the desired linear and angular velocities of the UAV, which are sent to the geometric controller defined in Sec. III-A, using the equations

$$
{}^{W}\boldsymbol{\omega}_B={}^{W}\boldsymbol{\omega}_C={}^{W}\mathbf{R}_C\,{}^{C}\boldsymbol{\omega}_C, \qquad (20)
$$

$$
{}^{W}\mathbf{v}_B={}^{W}\mathbf{R}_C\,{}^{C}\mathbf{v}_C-{}^{W}\mathbf{R}_B[{}^{W}\boldsymbol{\omega}_B]_\times\,{}^{B}\mathbf{p}_C. \qquad (21)
$$

C. Control of Physical Interaction

Here, we discuss how our control architecture takes into consideration the physical interaction. First, we present how we estimate and compensate the external wrench arising during physical interaction at the end-effector, and secondly, how this is used to let the platform be compliant.

1) External Wrench Observer: To estimate the external wrench that is applied to the end-effector, we adopt the observer proposed in [21]. Taking inspiration from ground manipulators, they propose a *hybrid* wrench observer tailored

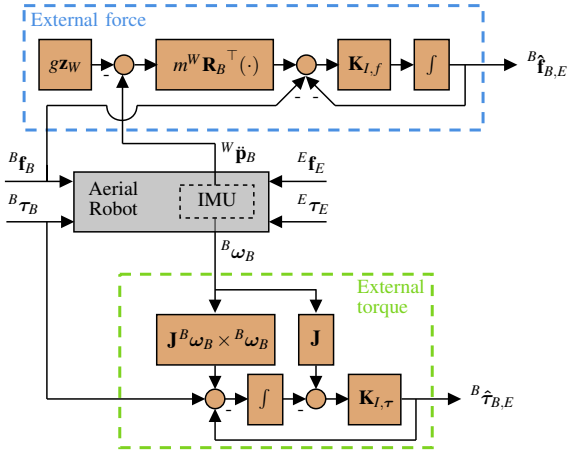

Fig. 3: Internal structure of the hybrid external wrench estimator.

for flying robots, which is a combination of an acceleration-based estimator for the external forces and a momentum-based one for the external torques. This hybrid estimator is well suited for flying robots since it exploits only proprioceptive sensors, which are usually available on board, as an Inertial Measurement System (IMU). The estimator, whose structure is detailed in Fig. 3, computes the external wrench in $\mathscr{F}_B$, composed by the forces ${}^B\hat{\mathbf{f}}_{B,E} \in \mathbb{R}^3$ and torques ${}^B\hat{\boldsymbol{\tau}}_{B,E} \in \mathbb{R}^3$, as

$$\begin{cases} {}^B\hat{\mathbf{f}}_{B,E} = \displaystyle\int_0^t \mathbf{K}_{I,\mathbf{f}}\left(m\,{}^B\mathbf{a}_B - {}^B\mathbf{f}_B - {}^B\hat{\mathbf{f}}_{B,E}\right)\mathrm{d}\tau \\ {}^B\hat{\boldsymbol{\tau}}_{B,E} = \mathbf{K}_{I,\boldsymbol{\tau}}\left(\mathbf{J}\,{}^B\boldsymbol{\omega}_B - \displaystyle\int_0^t \left({}^B\boldsymbol{\tau}_B + \mathbf{J}\,{}^B\boldsymbol{\omega}_B \times {}^B\boldsymbol{\omega}_B - {}^B\hat{\boldsymbol{\tau}}_{B,E}\right)\mathrm{d}\tau\right) \end{cases} \tag{22}$$

where $\mathbf{K}_{I,\mathbf{f}} \in \mathbb{R}^{3\times3}$, $\mathbf{K}_{I,\boldsymbol{\tau}} \in \mathbb{R}^{3\times3}$ are respectively the estimator gains for the external forces and moments, and ${}^B\mathbf{a}_B = {}^W\mathbf{R}_B{}^\top({}^W\ddot{\mathbf{p}}_B - g\mathbf{z}_W)$ is the robot acceleration in $\mathscr{F}_B$. This latter quantity is provided by the onboard accelerometer, while ${}^B\boldsymbol{\omega}_B$ is provided by the gyroscope. Note that the notation ${}^B\hat{\mathbf{f}}_{B,E}$ and ${}^B\hat{\boldsymbol{\tau}}_{B,E}$ denotes the estimated value, respectively, of the forces and torques applied to the robot's C.o.M. (i.e. to O_B) generated by the external wrench applied to the end-effector, expressed in $\mathscr{F}_B$.

Once the external wrench is computed, it can be compensated by the geometric controller of Sec. III-A, by adding it at the right-hand side of Eq. (6) and (8), which can be rewritten as

$${}^W\mathbf{f}_B^r = {}^W\bar{\mathbf{f}}_B^r - {}^W\mathbf{R}_B\,{}^B\hat{\mathbf{f}}_{B,E} \tag{23}$$

$${}^B\boldsymbol{\tau}_B = {}^B\bar{\boldsymbol{\tau}}_B - {}^W\mathbf{R}_B\,{}^B\hat{\boldsymbol{\tau}}_{B,E}. \tag{24}$$

where ${}^W\bar{\mathbf{f}}_B^r$ and ${}^B\bar{\boldsymbol{\tau}}_B$ are the nominal values of the forces and torques computed according to Eq. (6) and Eq. (8), respectively.

2) Admittance Filter: For letting the robot be compliant during the physical interaction, we resort to an admittance filter, which is a well-known technique in the literature [22]. The admittance filter takes as input the desired trajectory of the end-effector, expressed in $\mathscr{F}_W$, and composed by the end-effector position, attitude, as well as their first and second-order derivatives (top-scripted $\cdot^d$). The filter produces a new reference trajectory ($\cdot^r$), which is compliant w.r.t. the applied external wrench. Since it is possible to relate the kinematic quantities of the $\mathscr{F}_E$ to the ones of the $\mathscr{F}_B$ and vice versa, and similarly transpose the external wrench effect in one of the two frames, the admittance filter can be either applied to the end-effector level or in the body C.o.M., in both cases making the platform's end-effector compliant. Since the HVS scheme is generating the desired velocities of the $\mathscr{F}_B$ expressed in $\mathscr{F}_W$, we write the admittance filter w.r.t. the robot C.o.M, which therefore takes the following form

$$\mathbf{M}_B \begin{bmatrix} \ddot{\mathbf{e}}_{\mathbf{p},B} \\ \dot{\mathbf{e}}_{\boldsymbol{\omega},B} \end{bmatrix} + \mathbf{D}_B \begin{bmatrix} \dot{\mathbf{e}}_{\mathbf{p},B} \\ \mathbf{e}_{\boldsymbol{\omega},B} \end{bmatrix} + \mathbf{K}_B \begin{bmatrix} \mathbf{e}_{\mathbf{p},B} \\ \mathbf{e}_{\mathbf{R},B} \end{bmatrix} = \begin{bmatrix} {}^W\hat{\mathbf{f}}_{B,E} \\ {}^W\hat{\boldsymbol{\tau}}_{B,E} \end{bmatrix}, \tag{25}$$

where $\mathbf{e}_{\mathbf{p},B} = {}^W\mathbf{p}_B^r - {}^W\mathbf{p}_B^d$ and $\boldsymbol{e}_{\boldsymbol{\omega},B} = {}^W\boldsymbol{\omega}_B^r - {}^W\boldsymbol{\omega}_B^d$ are the translational and angular errors, and matrices $\mathbf{M}_B \in \mathbb{R}^{6\times6}$, $\mathbf{D}_B \in \mathbb{R}^{6\times6}$, $\mathbf{K}_B \in \mathbb{R}^{6\times6}$ are the designed mechanical inertia, damping and stiffness, respectively. Finally, ${}^W\hat{\mathbf{f}}_{B,E}$ and ${}^W\hat{\boldsymbol{\tau}}_{B,E}$ constitute the estimated external forces and torques computed by the wrench observer through Eq. (22), but rotated to $\mathscr{F}_W$. The orientation error $\mathbf{e}_{\mathbf{R},B}$ is defined as

$$\mathbf{e}_{\mathbf{R},B} = \frac{1}{2}\left({}^W\mathbf{R}_B^r\,{}^W\mathbf{R}_B^{d\top} - {}^W\mathbf{R}_B^d\,{}^W\mathbf{R}_B^{r\top}\right)^\vee, \tag{26}$$

similary as in Eq. (8).

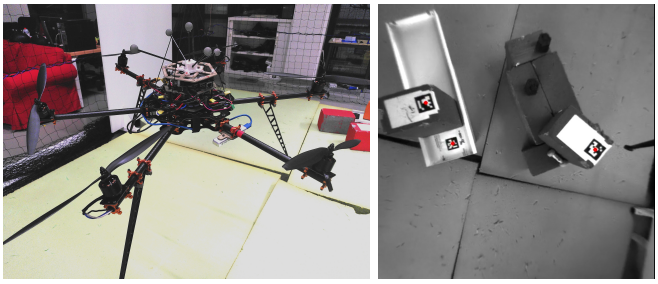

Fig. 4: On the left, a picture of the fully-actuated UAV used in the experimental validation, while on the right a picture taken from the onboard camera about the feature detection.

IV. Experimental Validation

In this section, we first describe the validation scenario used as a specific implementation of the previously introduced generic control scheme. Then, we present the experimental setup employed and the experimental results.

A. Validation scenario: Autonomous pick and place

In order to validate our control scheme, we perform an autonomous pick-and-place operation with a laterally-bounded fully-actuated UAV. This section presents the implementation and the various assumptions made for this specific use case.

The UAV that we use is a tilted-propeller hexa-rotor. It is equipped with a monocular camera and an end-effector capable of tightly gripping an object, e.g., a simple brick, in order to move it in the workspace. The wrench applied by the brick at the end-effector is compensated by the controller

using the aforementioned strategy, so no prior knowledge of the brick mass or inertia is needed.

To ensure a safe motion in the workspace, we assume without any loss of generality that the motion is done at a constant altitude z. We also impose, to exploit the full actuation of the platform and reduce the uncertainty in the orientation measurements, a constant roll and pitch for the platform. This is achieved by enforcing the associated vector components to be zero in the HVS outputs (Eq. (20) and (21)). Once the end-effector is aligned with the object, the UAV descends until sufficient force feedback is measured by the wrench observer, ensuring contact for picking. Similarly, the placing operation is autonomously performed by visually servoing to a certain location and by employing contact feedback.

Finally, in order to ensure the feasibility of the task when the features are not in the field of view (f.o.v.) of the camera, a position-based searching policy is implemented to scan the (pre-defined) area until the brick is found. Then, the HVS controller is re-enabled. Similarly, the placing location is not known a priori and, if not already visible inside the f.o.v. of the camera, is searched and detected by adopting the same area-scanning routine.

B. Experimental Setup

The UAV - whose picture is provided on the left of Fig. 4 - is a custom tilted-propeller hexa-rotor called the *FiberTHex* designed in the LAAS-CNRS, where the experiments are performed. It is sized to have a comfortable payload while providing enough lateral thrust to ensure rapid motion. The platform diameter is about $80\,[\text{cm}]$ and is actuated by 6 evenly-spaced $13\,[\text{in}]$ propellers.

The UAV is equipped with an onboard Intel NUC, with an Intel Core i7-8565U and 8GB of DDR3 RAM, in order to be able to run the image processing algorithms. It runs Ubuntu 18.04, and the software architecture is implemented in C++, using GenoM [23], which is a middleware agnostic component generator that can then be compiled for a given middleware, e.g., ROS. We make use of the TeleKyb3 software for the state estimation as well as the low-level control, available on the OpenRobots platform[1]. The UAV state feedback is provided by an onboard IMU ($1\,[\text{kHz}]$) and an external motion capture system ($100\,[\text{Hz}]$). Sensor fusion of the available sensor data is realized by using an Unscented Kalman Filter, which provides the full state feedback at $1\,[\text{kHz}]$.

The onboard camera is an Intel Realsense T265, chosen for its lightweight and commodity, but its odometry feedback is not used. The feature detection for the camera is performed using AruCo fiducial markers [24], as shown on the right of Fig. 4, for the sake of simplifying and abstracting the detection process. However, many UAV-oriented detection algorithms exist in the literature; for instance, deep-learning-based algorithms, as presented in [25], [26].

[1] `https://git.openrobots.org/projects/telekyb3`

	Full Scheme	No W.O. No A.F.
$\mu_d\,[\text{m}]$	0.0512	0.1059
$\sigma_d\,[\text{m}]$	0.0281	0.0498
$\mu_{e_\eta}\,[\text{Deg}]$	[0.2634 0.2875 2.8201]	[0.3059 0.2578 2.7623]
$\sigma_{e_\eta}\,[\text{Deg}]$	[0.2175 0.2235 2.1948]	[0.2063 0.1951 2.0716]

TABLE I: Mean ($\mu_{(\cdot)}$) and standard deviation ($\sigma_{(\cdot)}$) of the errors in position (d) and attitude ($\boldsymbol{\eta} = [\phi, \theta, \psi]$) for both experiments.

C. Experimental Results

In this section, we present the results of two conducted experiments, each one requiring the robot to find a brick, pick and place it at another location. In the former experiment, we employ the full control architecture presented in this paper, while in the latter, we disable the modules that handle the physical interaction, precisely the wrench observer of Sec. III-C.1 and the admittance filter of Sec. III-C.2. These two experiments aim to demonstrate the validity and effectiveness of our proposed control architecture compared to classical control approaches, which do not take into consideration the physical interaction and rely solely on the disturbance-rejection capabilities of the controller. Moreover, in both experiments, we disable the integral action in the position part (Eq. (6)) of the geometric controller of Sec. III-A, while we keep it in the attitude (Eq. (8)), in order to further highlight how the proposed control method can achieve accurate tracking of the reference motion.

In Tab. I, we report the metrics computed from the two experiments. They are the mean and standard deviation of the euclidean distance d between the samples of the reference trajectory and the current robot's state, and the means and standard deviations of the orientation errors between the reference and the current robot's attitude (expressed as Euler angles Roll-Pitch-Yaw $\boldsymbol{\eta} = [\,\phi\ \theta\ \psi\,]^\top$). These metrics show how the adoption of the full control architecture we presented (column *Full scheme*) achieves better reference-tracking accuracy compared to an HVS scheme solely combined with a standard controller unaware of the physical interaction (column *No W.O. No A.F.*). Indeed, during the second experiment, a larger mean and standard deviation of the position error are obtained on average. On the contrary, almost identical attitude-tracking performances are obtained, as we require the platform to maintain a flat orientation w.r.t. the ground, to exploit its full actuation. Instead, the nature of the larger error in the position tracking, obtained during the second experiment, can be better appreciated by inspecting Fig. 5.

The top plot of Fig. 5 shows the reference trajectory of the robot over its current position state for the first experiment, while the bottom one shows the same quantities for the second experiment. In the region highlighted in light blue, it is possible to note how relevant is the adoption of a control method aware of the physical interaction and capable of compensating for it. Indeed, during the second experiment, a classical controller (bottom plot) provides a noticeably larger deviation from the reference z. On the other hand, the tracking performances on the y and x components are less affected. Since our validation scenario consists of a

pick-and-place operation, it is clear that the most stressed motion component is the vertical one, perpendicular to the ground, along which most of the interaction forces arise due to the picked payload, while it naturally leads to smaller forces along the lateral directions. Therefore, we expect that another physical interaction task, involving the exchange of significant forces along the directions parallel to the ground, would bring the classical controller to deviate more also from the x and y reference trajectories, thus further increasing the position error metric of Tab. I.

In Fig. 6, it is possible to appreciate the altitude of the robot (top), and the magnitude of the force exchanged along $\mathbf{z}_B$ between the robot and the brick (bottom), only during the first experiment. In the regions highlighted in green, the picking phase occurs: the robot's altitude z decreases to the brick level (denoted by z_{pick}), while the vertical component of the estimated contact force F_z arises. The z restarts to increase after the contact is detected, i.e., when the monitored contact force exceeds the threshold of 2 [N], after which F_z decreases. On the other hand, in the regions highlighted in red, the placing phase takes place. Contrary to the picking phase, the robot's z firstly decreases to reach the placing location (denoted by z_{place}), followed by an increase of the interaction force. Then, after the contact force has reached the defined threshold and the brick has been placed, the robot moves upwards again. However, due to the altitude of the placing location, the wind induced by the propellers' thrust is causing some *ground effect*, which exerts a vertical force on the UAV. Indeed, we see in Fig. 6, that the estimated vertical force F_z increases before the contact is established, hence the placing contact phase is much shorter than the picking one. In-between these two phases, while the robot moves away from the picking and towards the placing position, F_z oscillates around the constant value of about 4 [N], which corresponds to the picked object's weight.

Finally, in Fig. 7, the displacement of the detected brick in the image plane of the camera is drawn as a continuous black line. Every 1 [s], the current position of the brick is drawn as a red dot, as well its current orientation is synthetically displayed through a purple segment. The adopted HVS scheme leads the robot to accurately move the equipped camera on top of the detected brick and to align with it. Indeed, as the reference position of the brick (denoted by the yellow squared marker) is approached, the camera orientation converges to the desired one (the latter displayed as a green segment).

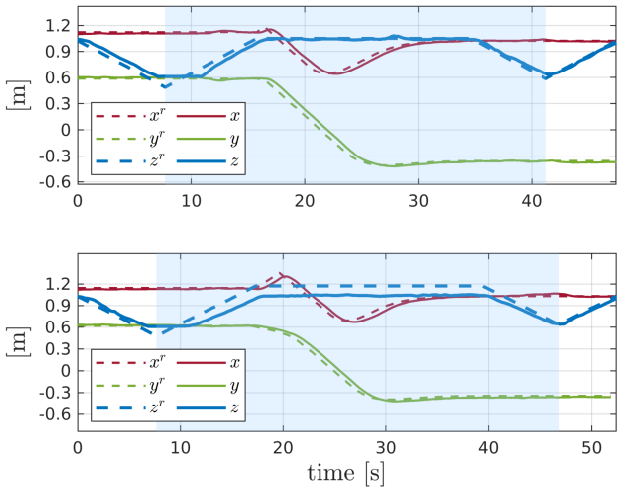

Fig. 5: The position tracking during the two experiments. In dashed lines the reference position trajectories, while in continuous lines the current position states of the robot. The blue area shows a phase of interest, as detailed in Sec. IV-C.

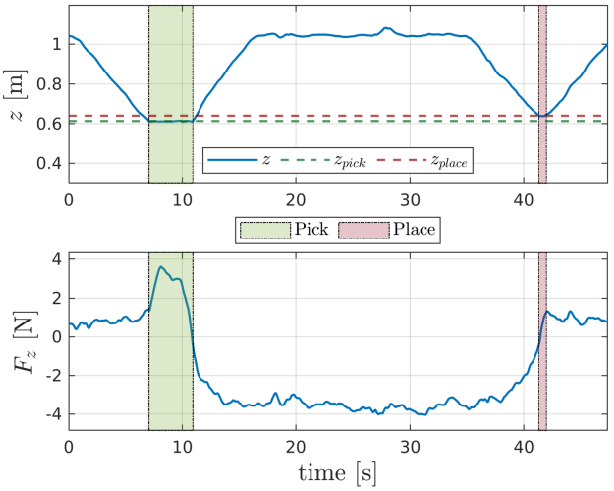

Fig. 6: In the first plot, the altitude of the robot in $\mathcal{F}_W$, while in the second, the estimated contact force ${}^B\hat{\mathbf{f}}_{B,E}$ along $\mathbf{z}_B$.

V. Conclusion

We present a general control architecture for autonomous physical interaction tasks tailored to fully-actuated UAVs. We employ a 6D physical-interaction paradigm called *The Flying End-Effector*, by adopting, in particular, a position/attitude decoupled controller to exploit the 6D motion capability of the fully-actuated platform. To make the architecture autonomous, the environment is monitored using a monocular camera; thus, we implement a classical vision-based trajectory generator called Hybrid Visual Servoing. Finally, the physical interaction is handled by using an onboard model-based wrench observer, in order to autonomously react to the physical contact, and the generated trajectory is filtered through an admittance filter to achieve compliancy. This control architecture is employed, as a proof of concept, in a vision-driven autonomous pick-and-place scenario. The experimental results are provided to demonstrate the coherence and efficiency of the proposed framework. In particular, we show how the association of these techniques allows for an efficient motion and, at the same time, it ensures better trajectory-tracking accuracy compared to controllers unaware of the physical interaction with the environment.

However, even if such framework aims at being generic, it still requires a lot of tuning, e.g., of the controller's and filters' parameters. Moreover, the task has to be designed to be feasible by the UAV; in particular, the camera and the targets have to be placed and oriented in the workspace such

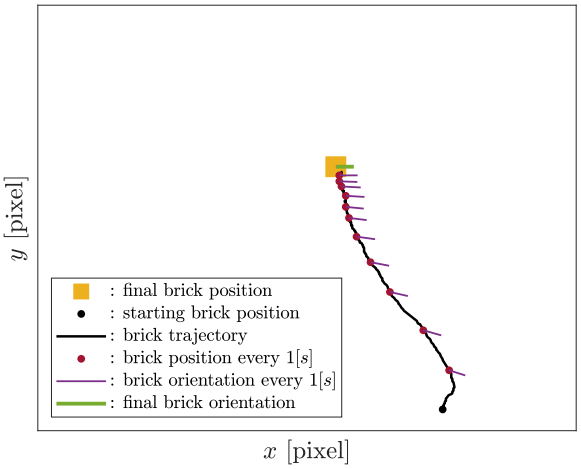

Fig. 7: The displacement and orientation of the detected brick in the image plane of the camera.

that the accomplishment of the task is ensured. Nevertheless, it is interesting to point out that the features provided by this control framework could be essential when considering future scenarios where, for instance, these robots could be requested to interact and cooperate with human operators sharing their same workspace. Examples of such essential features are: 1) full actuation and thus 6*D* dexterity, 2) the capability to be visually driven to the desired goal, 3) being aware of the surrounding environment, and 4) the ability to physically interact with it. Thanks to these attributes, an aerial robot could provide a human worker with the necessary tools at the required time, or it could help him in co-manipulating heavy or bulky equipment.

ACKNOWLEDGMENT

This work has been motivated by preparing for the 2020 edition of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC), an international robotics competition held every two years. The authors thank the organizers for their effort and help during the event.

REFERENCES

[1] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, 2011.

[2] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, 2014.

[3] M. Tognon, H. A. Tello Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés, and A. Franchi, "A truly redundant aerial manipulator system with application to push-and-slide inspection in industrial plants," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1846–1851, 2019.

[4] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "Active interaction force control for contact-based inspection with a fully actuated aerial vehicle," *IEEE Trans. on Robotics*, vol. 37, no. 3, pp. 709–722, 2021.

[5] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Behavioral control of unmanned aerial vehicle manipulator systems," *Autonomous Robots*, vol. 41, no. 5, pp. 1203–1220, 2017.

[6] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 4990–4995.

[7] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi, "Dynamic decentralized control for protocentric aerial manipulators," in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017, pp. 6375–6380.

[8] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6D interaction control with aerial robots: The flying end-effector paradigm," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1045–1062, 2019.

[9] H. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 3458–3464.

[10] G. Jiang, R. M. Voyles, and J. J. Choi, "Precision fully-actuated uav for visual and physical inspection of structures for nuclear decommissioning and search and rescue," in *2018 IEEE Int. Symp. on Safety, Security and Rescue Robotics*, 2018, pp. 1–7.

[11] L. Lin, Y. Yang, H. Cheng, and X. Chen, "Autonomous vision-based aerial grasping for rotorcraft unmanned aerial vehicles," *Sensors*, vol. 19, no. 15, 2019.

[12] R. Rashad, D. Bicego, R. Jiao, S. Sanchez-Escalonilla, and S. Stramigioli, "Towards vision-based impedance control for the contact inspection of unknown generically-shaped surfaces with a fully-actuated uav," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 1605–1612.

[13] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[14] R. Ozawa and F. Chaumette, "Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach," in *2011 IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 5670–5676.

[15] E. Malis, F. Chaumette, and S. Boudet, "2 1/2D visual servoing," *IEEE Trans. on Robotics*, vol. 15, no. 2, pp. 238–250, 1999.

[16] F. Conticelli, B. Allotta, and C. Colombo, "Hybrid visual servoing: A combination of nonlinear control and linear vision," *Robotics and Autonomous Systems*, vol. 29, no. 4, pp. 243–256, 1999.

[17] A. Franchi, R. Carli, D. Bicego, and M. Ryll, "Full-pose tracking control for aerial robotic systems with laterally-bounded input force," *IEEE Trans. on Robotics*, vol. 34, no. 2, pp. 534–541, 2018.

[18] G. Michieletto, M. Ryll, and A. Franchi, "Fundamental actuation properties of multi-rotors: Force-moment decoupling and fail-safe robustness," *IEEE Trans. on Robotics*, vol. 34, no. 3, pp. 702–715, 2018.

[19] F. Goodarzi, D. Lee, and T. Lee, "Geometric nonlinear pid control of a quadrotor uav on se(3)," in *2013 European Control Conference*, 2013, pp. 3845–3850.

[20] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[21] T. Tomić, C. Ott, and S. Haddadin, "External wrench estimation, collision detection, and reflex reaction for flying robots," *IEEE Trans. on Robotics*, vol. 33, no. 6, pp. 1467–1482, 2017.

[22] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[23] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan, and F. Ingrand, "GenoM3: Building middleware-independent robotic components," in *2010 IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 4627–4632.

[24] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[25] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu *et al.*, "Visdrone-DET2018: The vision meets drone object detection in image challenge results," in *2018 IEEE/CVF European Conf. on Computer Vision*, 2018.

[26] Y. Akbari, N. Almaadeed, S. Al-maadeed, and O. Elharrouss, "Applications, databases and open computer vision research from drone videos and images: a survey," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3887–3938, 2021.