2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Toward image-based visual servoing for cooperative aerial manipulation

Rafik Mebarki, Vincenzo Lippiello, and Bruno Siciliano

Abstract— **This paper proposes a new visual controller to endow flying manipulators with the capability of cooperatively and automatically positioning an assembly part on a visual target. We consider two VToL UAVs each equipped with a robot manipulator and a video camera. The two manipulators are rigidly connected to an assembly part, consisting of bar in this case study, that needs to be automatically placed on its target pose. The control system we propose uses the images relayed from the two onboard cameras to simultaneously control the motions of the UAVs and the manipulators to achieve the task: positioning the bar as well as respecting the closed-chain constraint. The controller evolves in the image space, thus inheriting robustness with respect to calibration errors. It takes into account the under-actuation inherent to rotary-wing aerial vehicles as well as the redundancy of the whole system. Furthermore, by leveraging image moments the 3D relative motion between the carried assembly part and the environment is controlled, thus yielding to natural and practically-relevant movements. Numerical simulations have been carried out to verify the validity of the proposed approach.**

I. Introduction

Robotic manipulation has expanded these last decades from fixed to ground-mobile configuration, and more recently to aerial manipulation. An aerial vehicle can access regions not easily or possibly envisaged by land or water. In addition it yields shorter deployment time since it does not necessitate any road preparation, construction, nor extra structures as scaffolding for instance [1] to reach elevated regions. More particularly, vertical take-off and landing (VToL) Unmanned Aerial Vehicles (UAVs) feature the hovering capability, which is crucial for robotic manipulation from the air, since it enables minimum steadiness and motionless for the carried manipulator, and thus allows envisaging controlled physical interaction with the environment. Nevertheless, most of the research works on UAVs have been limited to sensing, while physical interaction with the environment has only recently been considered [2][3].

In [2], three small-scale helicopters autonomously transport a shared slung mass in an outdoor experiment under GPS guidance, while in [4] three micro quadrotors are employed inside an arena endowed with a motion capture system. Work [5] proposes a motion planner for manipulating 6 degree-of-freedom (DoF) object cable-towed to three quadrotors. [6] proposes a controller to enable a UAV quadrotor exerting a desired force on the environment. In [7], the effect of an extra mass on a UAV is investigated.

Works [8] [9] propose a quadrotor-mounted parallel manipulator actuating a probe for wall inspection. [10] considers aerial peg-in-hole insertion tasks by means of a serial link manipulator carried by a rotary-wing aircraft. A kinematic controller is presented for the purpose, yet the vehicle base and the manipulator are controlled separately. Works [11] [12] deal with the cross-dynamic effects between the UAV's base and the attached manipulators. In [13], a 7 DoF anthropomorphic manipulator is attached to a small-scale helicopter for visual grasping. A camera is attached directly to the craft's base. The base and the manipulator are controlled by means of position-based visual servoing [14][15] in a GPS-covered environment.

The contributions of the present paper are twofold. First, we tackle the problem of cooperative aerial manipulation, such that the scenario of two aerial robotic systems (ARSs) holding rigidly an assembly part is considered. We consider the pre-grasping phase. The closed-chain constraint (CCC) is derived in an amenable form which then enables addressing the second and main objective of the paper: cooperative image-based aerial manipulation. The velocity references generated for each ARS on the basis of the image information are projected onto the subspace subject to the CCC. The second contribution thus builds upon our previous works [16][17] on image-based aerial manipulation to extend them to cooperative manipulation. An onboard camera is attached to each of the vehicle bases and the robotic task consists in automatically placing the carried assembly structure on its target location using only onboard vision as a means for guidance. More precisely, the objective is to position two parts of the assembly structure, e.g. two clips or two hooks. Each onboard camera relays images of one of the two parts it observes. Doing so relaxes the field of view constraint and allows considering even long bars. Indeed, it would be more challenging to cover with a single camera the whole or main portions of a long bar. This paper proposes a new image-based kinematic controller to achieve the described cooperative task. It is based on a visual error definition different from classical visual servoing's [14][15]. The definition we propose enables controlling the craft bases and the manipulators simultaneously. The visual controller generates desired velocities to each of the rotary-wing UAV bases and manipulators. While it takes into account the peculiarity of under-actuation inherent to rotary-wing aircrafts, it exploits the redundancy afforded by the system. It takes advantage of the mobility afforded by the UAV base when far from the target, while it exploits the dexterity of the manipulator once

The research leading to these results has been supported by ARCAS collaborative project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-287617.

The authors are with PRISMA Lab, Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Naples, Italy, {rafik.mebarki, vincenzo.lippiello, and bruno.siciliano}@unina.it.

in the vicinity. More importantly, the control scheme guaranties the CCC, thus generates motions that are physically the least stressing (theoretically) to the system. Furthermore, since image-based approaches lack the feature of controlling directly robot-target relative motion, we propose to address this issue by means of image moments. Such information indeed relay intuitive and geometrical meaning, such that their low-orders describe for instance the area and center of gravity of an observed entity. This is of importance, since they can be exploited to enforce, implicitly, a desired 3D interaction with the environment. For example, to yield the assembly part's motion perpendicular to the surface on which it is to be positioned, during the approaching and final phase of a docking application. It is worth pointing out that low-level interaction control is not the subject of this paper. In this paper we propose a cooperative visual servo control useful to generate the reference for a low-level force or impedance control scheme [18]. Impedance-based approaches [19], as internal impedance [20] or object impedance [21] paradigms, can be adopted and added to the proposed visual controller in a roughly straight forward manner.

II. Modeling

The robotic system consists of two quadrotors, each equipped with a serial-link robot manipulator (see Fig. 1). The quadrotors are not necessarily of the same mass, payload, or size. The two manipulators posses, respectively, n_1 and n_2 controllable DoFs. From now on, for brevity of notation, any entity with subscrip k is associated with the k-th ARS, such that $k = 1, 2$. Each manipulator is attached rigidly to the corresponding vehicle base. The manipulator Cartesian base frame is denoted with $\{b_k\}$, while $\{u_k\}$ represents the body-frame of the vehicle base.

A. Closed-chain constraint

Let E_k denote the end-effector (EE) of the manipulator, and let $\{E_k\}$ be the Cartesian frame attached to it (see Fig. 1). Let then ${}^{E_1}\boldsymbol{t}_{E_2}$ represent the 3D position of E_2 in frame $\{E_1\}$, while ${}^{E_1}\boldsymbol{R}_{E_2}$ is the rotation matrix from $\{E_1\}$ to $\{E_2\}$. Since the two manipulators are holding the assembly structure rigidly, the CCC can be formulated as follows:

$$ {}^{E1}\boldsymbol{R}_{E2} = \text{constant} \quad \text{and} \quad {}^{E1}\boldsymbol{t}_{E2} = \text{constant}. \tag{1} $$

Constraint (1) expresses the fact that no relative displacement nor rotation (twist) is allowed between the two frames $\{E_1\}$ and $\{E_2\}$. We can reformulate it in terms of time variations ${}^{E_1}\dot{\boldsymbol{t}}_{E_2}$ and ${}^{E_1}\dot{\boldsymbol{R}}_{E_2}$ of ${}^{E_1}\boldsymbol{t}_{E_2}$ and ${}^{E_1}\boldsymbol{R}_{E_2}$, respectively, in the following equivalent constraint:

$$ \begin{cases} {}^{E1}\dot{\boldsymbol{R}}_{E2} &= \boldsymbol{0}_{3\times 3} \\ {}^{E1}\dot{\boldsymbol{t}}_{E2} &= \boldsymbol{0}_3, \end{cases} \tag{2} $$

where $\boldsymbol{0}_{3\times 3}$ denotes the null matrix of size 3×3, while $\boldsymbol{0}_3$ the 3-element null vector. It would be more appropriate to work on the dynamic relationship (2) instead of the static (1), as described in the sequel. Contraint (2) will be expressed in terms of the velocity of each of the involved robotic elements.

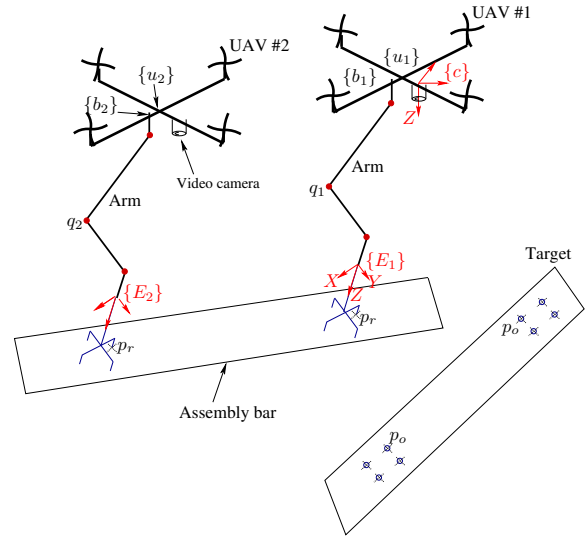

Fig. 1. Sketch of the robotic system and target, as well as involved Cartesian frames and variables.

B. Rotational part of the CCC

Let $\boldsymbol{\omega}_k = {}^{u_k}\boldsymbol{\omega}_{u_k} = (\omega_{kx}\ \omega_{ky}\ \omega_{kz}) \in \mathbb{R}^3$ be the angular velocity of the k-th vehicle base, i.e. the angular velocity of $\{u_k\}$ expressed in $\{u_k\}$. Notice that that ω_{kx} and ω_{ky} represent the attitude (roll and pitch) rate, while ω_{kz} is the heading (yaw) rate. Let also ${}^{E1}\boldsymbol{R}_{b_k}$ and ${}^{E1}\boldsymbol{R}_{u_k}$ be the rotation matrices describing the orientations from frame $\{E_1\}$ to frame $\{b_k\}$, and from $\{E_1\}$ to $\{u_k\}$, respectively. The rotational part of the CCC given by the first relationship of (2) can be recast as follows:

$$ {}^{E1}\boldsymbol{R}_{b_1}\, {}^{b1}\boldsymbol{\omega}_{E_1} + {}^{E1}\boldsymbol{R}_{u_1}\, \boldsymbol{\omega}_1 - {}^{E1}\boldsymbol{R}_{u_2}\, \boldsymbol{\omega}_2 - {}^{E1}\boldsymbol{R}_{b_2}\, {}^{b2}\boldsymbol{\omega}_{E_2} = \boldsymbol{0}_3, \tag{3} $$

such that ${}^{b_1}\boldsymbol{\omega}_{E_1}$ is the rotational velocity of the manipulator EE's frame $\{E_1\}$ with respect to its base frame $\{b_1\}$. It is expressed in $\{b_1\}$. Likewise, ${}^{b_2}\boldsymbol{\omega}_{E_2}$ is the angular velocity of $\{E_2\}$ with respect to $\{b_2\}$ and expressed in $\{b_2\}$. Let $\boldsymbol{q}_k$ be the n_k-elements joint-angles vector of the k-th manipulator, while $\dot{\boldsymbol{q}}_k$ is its time variation. Let also ${}^{b_k}\boldsymbol{t}_{E_k}$ be the translational vector from b_k to E_k, while ${}^{b_k}\dot{\boldsymbol{t}}_{E_k}$ is its time variation. Both velocities ${}^{b_k}\dot{\boldsymbol{t}}_{E_k}$ and ${}^{b_k}\boldsymbol{\omega}_{E_k}$ can be expressed in terms of $\dot{\boldsymbol{q}}_k$ via the manipulator geometrical Jacobian, $\boldsymbol{J}_{E_k}$, as follows:

$$ \begin{pmatrix} {}^{b_1}\dot{\boldsymbol{t}}_{E1} \\ {}^{b_1}\boldsymbol{\omega}_{E1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{J}_{E1t} \\ \boldsymbol{J}_{E1r} \end{pmatrix} \dot{\boldsymbol{q}}_1 \quad \text{and} \quad \begin{pmatrix} {}^{b_2}\dot{\boldsymbol{t}}_{E2} \\ {}^{b_2}\boldsymbol{\omega}_{E2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{J}_{E2t} \\ \boldsymbol{J}_{E2r} \end{pmatrix} \dot{\boldsymbol{q}}_2, \tag{4} $$

where $\boldsymbol{J}_{E_k} = \left(\boldsymbol{J}_{Ekt}^\top\ \boldsymbol{J}_{Ekr}^\top\right)^\top \in \mathbb{R}^{6\times 6}$, such that $\boldsymbol{J}_{Ekt} \in \mathbb{R}^{3\times 3}$ relates the translational part, while $\boldsymbol{J}_{Ekr} \in \mathbb{R}^{3\times 3}$ relates the rotational part. Replacing (4) in (3) leads to the following relationship:

$$ \boldsymbol{M}_{11}\, \dot{\boldsymbol{\xi}}_1 + \boldsymbol{M}_{12}\, \dot{\boldsymbol{\xi}}_2 = \overline{\boldsymbol{R}}_1\, \overline{\boldsymbol{\omega}}_1 - \overline{\boldsymbol{R}}_2\, \overline{\boldsymbol{\omega}}_2, \tag{5} $$

where

$$ \dot{\boldsymbol{\xi}}_k = \left(\dot{\boldsymbol{q}}_k^\top \quad \boldsymbol{v}_k^\top \quad \omega_{kz}\right)^\top \in \mathbb{R}^{n_k+4}. \tag{6} $$

Vector $\boldsymbol{v}_k \in \mathbb{R}^3$ denotes the linear velocity of the k-th UAV base with respect to a fixed Cartesian frame. As for $\dot{\boldsymbol{\xi}}_k$, it represents the generalised velocity of the k-th ARS. As for the involved matrices, they have been derived as follows:

$$\begin{aligned}
&\boldsymbol{M}_{11} = \begin{pmatrix} -{}^{E1}\boldsymbol{R}_{b_1}\,\boldsymbol{J}_{E1r} & \boldsymbol{0}_{3\times3} & -\boldsymbol{r}_1 \end{pmatrix} \in \mathbb{R}^{3\times(n_1+4)} \\
&\boldsymbol{M}_{12} = \begin{pmatrix} {}^{E1}\boldsymbol{R}_{b_2}\,\boldsymbol{J}_{E2r} & \boldsymbol{0}_{3\times3} & \boldsymbol{r}_2 \end{pmatrix} \in \mathbb{R}^{3\times(n_2+4)} \\
&{}^{E1}\boldsymbol{R}_{u_1} \equiv \begin{pmatrix} \overline{\boldsymbol{R}}_1 & \boldsymbol{r}_1 \end{pmatrix} \in SO(3);\ \boldsymbol{r}_1 \in \mathbb{R}^{3\times1} \\
&{}^{E1}\boldsymbol{R}_{u_2} \equiv \begin{pmatrix} \overline{\boldsymbol{R}}_2 & \boldsymbol{r}_1 \end{pmatrix} \in SO(3);\ \boldsymbol{r}_2 \in \mathbb{R}^{3\times1} \\
&\boldsymbol{\omega}_1 \equiv (\overline{\boldsymbol{\omega}}_1^\top\ \omega_{1z})^\top \quad \text{and} \quad \boldsymbol{\omega}_2 \equiv (\overline{\boldsymbol{\omega}}_2^\top\ \omega_{2z})^\top \in \mathbb{R}^3.
\end{aligned} \tag{7}$$

C. Translational part of the CCC

Let ${}^{u_1}\boldsymbol{t}_{u_2} \in \mathbb{R}^3$ be the translational vector expressing the 3D coordinates of second quadrotor body-frame $\{u_2\}$ with respect to first quadrotor body-frame $\{u_1\}$. Also, ${}^{u_2}\boldsymbol{t}_{E2}$ represents the 3D coordinates of the end-effector of the second manipulator with respect to $\{u_2\}$. Thus, the 3D coordinates of $\{E_2\}$ with respect to $\{E_1\}$ write:

$${}^{E1}\boldsymbol{t}_{E2} = {}^{E1}\boldsymbol{t}_{u_1} + {}^{E1}\boldsymbol{R}_{u_1}\,{}^{u_1}\boldsymbol{t}_{u_2} + {}^{E1}\boldsymbol{R}_{u_2}\,{}^{u_2}\boldsymbol{t}_{E2}. \tag{8}$$

We express time variation ${}^{E1}\dot{\boldsymbol{t}}_{E2}$ as a function of the manipulator joint velocity vector $\dot{\boldsymbol{q}}_k$, and the quadrotor linear and angular velocities $\boldsymbol{v}_k$ and $\boldsymbol{\omega}_k$ as follows:

$$\begin{aligned}
{}^{E1}\dot{\boldsymbol{t}}_{E2} = {}&{}^{E1}\boldsymbol{R}_{u_2}\,\boldsymbol{v}_2 - {}^{E1}\boldsymbol{R}_{u_1}\,\boldsymbol{v}_1 + {}^{E1}\boldsymbol{R}_{b_2}\,\boldsymbol{J}_{E2t}\,\dot{\boldsymbol{q}}_2 \\
&- \boldsymbol{L}_1\,\dot{\boldsymbol{q}}_1 + \boldsymbol{A}_1\,{}^{b_1}\boldsymbol{\omega}_{E1} + \boldsymbol{B}_1\,\boldsymbol{\omega}_1 + \boldsymbol{B}_2\,\boldsymbol{\omega}_2,
\end{aligned} \tag{9}$$

where matrices $\boldsymbol{L}_1$, $\boldsymbol{A}_1$, $\boldsymbol{B}_2$, and $\boldsymbol{B}_3 \in \mathbb{R}^{3\times3}$ have been obtained as follows:

$$\begin{aligned}
\boldsymbol{L}_1 &= {}^{E1}\boldsymbol{R}_{u_1}[{}^{u_1}\boldsymbol{t}_{E1}]_\times\,{}^{u_1}\boldsymbol{R}_{b_1}\,\boldsymbol{J}_{E1r} + {}^{E1}\boldsymbol{R}_{b_1}\,\boldsymbol{J}_{E1t} \\
\boldsymbol{A}_1 &= {}^{E1}\boldsymbol{R}_{b_1}\,[{}^{b_1}\boldsymbol{R}_{u_2}\,{}^{u_2}\boldsymbol{t}_{E2}]_\times + [{}^{E1}\boldsymbol{R}_{u_1}\,{}^{u_1}\boldsymbol{t}_{u_2}]_\times\,{}^{E1}\boldsymbol{R}_{b_1} \\
\boldsymbol{B}_1 &= {}^{E1}\boldsymbol{R}_{u_1}\,[{}^{u_1}\boldsymbol{t}_{u_2}]_\times + {}^{E1}\boldsymbol{R}_{u_1}\,[{}^{u_1}\boldsymbol{R}_{u_2}\,{}^{u_2}\boldsymbol{t}_{E2}]_\times \\
\boldsymbol{B}_2 &= -\,{}^{E1}\boldsymbol{R}_{u_2}\,[{}^{u_2}\boldsymbol{t}_{E2}]_\times,
\end{aligned} \tag{10}$$

such that $[\cdot]_\times$ denotes the skew-symmetric matrix. Subsequently, (9) can be recast in the following compact form after taking into account (6) and the last two relationships of (7):

$$\boldsymbol{M}_{21}\,\dot{\boldsymbol{\xi}}_1 + \boldsymbol{M}_{22}\,\dot{\boldsymbol{\xi}}_2 = -\overline{\boldsymbol{B}}_1\,\overline{\boldsymbol{\omega}}_1 - \overline{\boldsymbol{B}}_2\,\overline{\boldsymbol{\omega}}_2, \tag{11}$$

where

$$\begin{aligned}
\boldsymbol{M}_{21} &= \begin{pmatrix} \left(\boldsymbol{A}_1\,\boldsymbol{J}_{E1r} - \boldsymbol{L}_1\right) & -\,{}^{E1}\boldsymbol{R}_{u_1} & \boldsymbol{b}_1 \end{pmatrix} \in \mathbb{R}^{3\times(n_1+4)} \\
\boldsymbol{M}_{22} &= \begin{pmatrix} {}^{E1}\boldsymbol{R}_{b_2}\,\boldsymbol{J}_{E2t} & {}^{E1}\boldsymbol{R}_{u_2} & \boldsymbol{b}_2 \end{pmatrix} \in \mathbb{R}^{3\times(n_2+4)} \\
\boldsymbol{B}_k &\equiv \begin{bmatrix} \overline{\boldsymbol{B}}_k & \boldsymbol{b}_k \end{bmatrix} \quad \overline{\boldsymbol{B}}_k \in \mathbb{R}^{3\times2}, \quad \boldsymbol{b}_k \in \mathbb{R}^{3\times1}.
\end{aligned}$$

D. Kinematic and compact form of the CCC

Let $\dot{\boldsymbol{\xi}} = \begin{pmatrix} \dot{\boldsymbol{\xi}}_1^\top & \dot{\boldsymbol{\xi}}_2^\top \end{pmatrix}^\top \in \mathbb{R}^{n'}$, such that $n' = n_1 + n_2 + 8$, be the generalised velocity describing the two ARSs. Stacking the derived relationships (5) and (11), we can finally formulate the CCC as follows:

$$\boldsymbol{M}\dot{\boldsymbol{\xi}} = \boldsymbol{G}\,\overline{\boldsymbol{\omega}}, \tag{12}$$

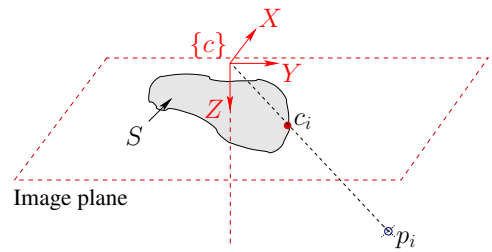

Fig. 2. Image section. Both point $\boldsymbol{c}_i$ and section S lie in the image plane.

with $\boldsymbol{M} \in \mathbb{R}^{6\times n'}$, $\boldsymbol{G} \in \mathbb{R}^{6\times4}$, and $\overline{\boldsymbol{\omega}} = \begin{pmatrix} \overline{\boldsymbol{\omega}}_1^\top & \overline{\boldsymbol{\omega}}_2^\top \end{pmatrix}^\top \in \mathbb{R}^4$. Relationship (12) features an amenable form for control. Indeed, its left side encloses only controllable system velocities, $\dot{\boldsymbol{\xi}}$, while the right side contains only roll and pitch attitude rates $\overline{\boldsymbol{\omega}}$ of the two quadrotors. The latter being under-actuated systems provide only four controllable DoFs, namely the linear velocity $\boldsymbol{v}_k$ and the heading rate ω_{kz}.

III. Image-Based Cooperative Visual Servoing

The focus of this paper is to enable coordinated automatic positioning of a rigidly carried assembly structure on a visual target, using the guidance from the onboard cameras. In the following, we present the higher-level image-based visual servo scheme to achieve the objective. Let $\dot{\boldsymbol{\xi}}_T \in \mathbb{R}^{n'}$ be a desired value of system velocity $\dot{\boldsymbol{\xi}}$ generated by the image-based controller to achieve a cooperative task, as detailed afterwards. These motions do not necessarily respect the CCC, and thus do not guarantee any acceptable cooperative action. To overcome this, we propose that reference velocity $\dot{\boldsymbol{\xi}}_c$ to be assigned to the two ARSs (i.e. that $\dot{\boldsymbol{\xi}}$ needs to track) be as follows:

$$\dot{\boldsymbol{\xi}}_c = \boldsymbol{M}^+\boldsymbol{G}\,\overline{\boldsymbol{\omega}} + (\boldsymbol{I}_{n'} - \boldsymbol{M}^+\,\boldsymbol{M})\,\dot{\boldsymbol{\xi}}_T \tag{13}$$

where $\boldsymbol{I}_x$ denotes the identity matrix of size x and $\boldsymbol{M}^+$ is a pseudo inverse of matrix $\boldsymbol{M}$. In contrast to $\dot{\boldsymbol{\xi}}_T$, $\dot{\boldsymbol{\xi}}_c$ guaranties the CCC described by (12), since $\dot{\boldsymbol{\xi}}_T$ is projected onto null space $(\boldsymbol{I}_{n'} - \boldsymbol{M}^+\,\boldsymbol{M})$ of $\boldsymbol{M}$. Remark that in addition to $\dot{\boldsymbol{\xi}}_T$, relationship (13) requires knowledge of standard measures. Attitude rate $\overline{\boldsymbol{\omega}}$ can be provided by the Inertial Measurement Unit's (IMU) gyroscope onboard each of the two vehicles. The same sensors are used to measure the involved attitudes (i.e. rotation matrices). The encoders of the two manipulators provide angle measures to build up $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, as well as $\boldsymbol{J}_{E1}$ and $\boldsymbol{J}_{E2}$. As for ${}^{u_1}\boldsymbol{t}_{u_2}$, it can be extracted either from the positions of the two UAVs with respect to a fixed frame using GPS measures for instance, or from the CAD model of the carried assembly structure since rigid manipulation is considered. In the last scenario, Eq. (8) can be employed by treating ${}^{E1}\boldsymbol{t}_{E2}$ as known.

Now, we describe how can $\dot{\boldsymbol{\xi}}_T$ be obtained. The robotic task consists in docking an assembly structure into its target location, such that both elements are characterised by a set of landmarks (see Fig. 1). Without loss of generality, each landmark $\boldsymbol{p}_r$ of the structure is associated with its counterpart $\boldsymbol{p}_o$ of the target. The robotic task thus can be expressed as making each pair $(\boldsymbol{p}_r, \boldsymbol{p}_o)$ overlap. In other

words, the visual control scheme generates motions such that $\boldsymbol{p}_r$'s converge to their respective $\boldsymbol{p}_o$'s. An onboard camera is attached to the base of each vehicle (see Fig. 1). Such camera configuration, referred to as *onboard-eye-to-hand* in [16], differs from the configurations considered in classical visual servoing, namely *eye-in-hand* or *eye-to-hand* configurations. Indeed, visual servoing [14][15] approaches consider either the scenario where only the camera motions need to be controlled, or where the camera is fixed observing objects (robots) to be controlled. In the *onboard-eye-to-hand* configuration considered in the present paper, however, the camera motion as well as the motion of the elements it observes need to be controlled. Indeed, the camera motions correspond to those of the vehicle base, since the camera is directly attached to the base. As for the observed element, it corresponds to the manipulator end-effector. Classical visual servoing thus can not be directly applied in our scenario, but instead new schemes need to be established.

Let $\boldsymbol{c}_o = (x_o\ y_o)^\top \in \mathbb{R}^2$ be the image coordinates of point $\boldsymbol{p}_o$ lying on the target. The latter is assumed motionless, or at least varying with relatively far low rate compared to the visual control frequency. Let $\boldsymbol{c}_r = (x_r\ y_r)^\top \in \mathbb{R}^2$ be instead the *pseudo* image coordinates of point $\boldsymbol{p}_r$ attached to the structure, that can be either calculated and emulated from the manipulator odometry or directly extracted from the image [16]. We proposed in [16] a feedback image-based visual error characterised as follows:

$$\boldsymbol{e} = \boldsymbol{c}_r - \boldsymbol{c}_o. \tag{14}$$

Although this definition seems equivalent to what proposed in classical visual servoing, it in fact subtly characterises a different new paradigm. Indeed, in classical visual servoing the visual error corresponds to the difference between the current observed features and constant desired ones. The latter are usually learnt after positioning the robotic system at the desired configuration. Thereby only the observed features vary in the image to reach a pre-defined position. As for the new definition (14), both features vary to superimpose in the image. Moreover, there is no need for the preliminary learning phase.

While yielding the system robust with respect to odometry modelling and camera calibration errors, the image-based control paradigm however inherently lacks the ability to control directly system 3D motions. Only the motions in the 2D image plane are controlled, which might lead to undesirable, perhaps impracticable, robot path in the real world. Nevertheless, relevant combinations of image coordinates can lead implicitly to desired 3D motions. Image moments information [22] provide intuitive and geometrical meaning of an observed entity in the image. More particularly, their low-order values describe the area and center of gravity. This shows to be of interest for robotics control, such that they can be adopted as feedback in an image-based visual servo scheme so as to yield amenable system 3D motion [23] [24]. In the present docking scenario, it is desired that during the approaching phase the assembly structure be aligned with the target. This would avoid lateral motions between the assembly structure and the taget, motions that can be difficult to control or unborn in practice. In terms of image moments, the desired task can be formulated as keeping the error on the center of gravity null, while the error on area converging to zero. Note that instead of superimposing a set of points on their respective counter parts, as considered in [16], the task reformulates instead as superimposing the respective sections these points form.

1) Definition: Let $\mathcal{S}$ be a section of an observed object in the image and $\boldsymbol{c} = (x, y)$ be a point of $\mathcal{S}$ (see Fig. 2). The $(p+q)$-th order image moment m_{pq} associated with $\mathcal{S}$ writes [22]:

$$m_{pq} = \iint_S x^p\, y^q \,\mathrm{d}x\,\mathrm{d}y. \tag{15}$$

The area a and center of gravity $(\overline{x}, \overline{y})$ of $\mathcal{S}$ in the image write in terms of image moments up to the first order as follows:

$$a = m_{00}, \quad \overline{x} = m_{10}/m_{00} \quad \text{and} \quad \overline{y} = m_{01}/m_{00}. \tag{16}$$

2) Kinematics: In order to use image moment m_{pq} as control feedback, its time variation needs first to be expressed in terms of the robotic system velocity. In [23], a relationship is derived for the case of *eye-in-hand* camera configuration. It however does not hold for the *onboard-eye-to-hand* configuration characterising our system. We instead exploit and start from the subsequent formula [25], defining the image moments for polygonal shapes. It can be adopted in fact for any closed-shape forms. Let $\boldsymbol{c}_i = (x_i, y_i)$ be the i-th of m points describing the contour of $\mathcal{S}$, such that $\boldsymbol{c}_0 = \boldsymbol{c}_m$ to close the contour. Image moment m_{pq} can be expressed in terms of the contour points as follows [25]:

$$m_{pq} = \sum_{i=1}^{m} (y_i - y_{i-1})\, \sigma_{i,pq}, \text{ with } \sigma_{i,pq} = \sum_{k=0}^{p+1} \sum_{j=0}^{q} a_{kj}\, f_x\, f_y$$
$$\begin{cases} f_x = x_i^k\, x_{i-1}^{p+1-k}, \quad f_y = y_i^j\, y_{i-1}^{q-j} \\ a_{kj} = \frac{1}{(p+q+2)(p+1)} \binom{p+1}{k} \binom{q}{j} \Big/ \binom{p+q+1}{k+j}, \end{cases} \tag{17}$$

such that $\binom{\cdot}{\cdot}$ denotes the binomial coefficient. Note that $z^\beta = 0$ for $\beta < 0$ is considered in (17). Time differentiating, then substituting $\dot{x}_i$ and $\dot{y}_i$ with their expressions function of the system generalised velocity, as detailed in [17], it follows that time variation of moment m_{pq} can be obtained in this linear form [17]:

$$\dot{m}_{pq} = \boldsymbol{L}_{m_{pq}}\, \dot{\boldsymbol{\zeta}}, \tag{18}$$

such that $\dot{\boldsymbol{\zeta}}$ corresponds to either the UAV velocity or the manipulator joints-vector rate. Let ${}^r m_{pq}$ be the $(p+q)$-th order image moment of the section lying on the assembly part. More precisely, the contour of that section is characterised by $\boldsymbol{p}_r$'s (on Fig. 1 $\boldsymbol{p}_r$'s delimit a rectangle, that needs to superimpose on the rectangle delimited by $\boldsymbol{p}_o$'s). Likewise, ${}^o m_{pq}$ is associated to the target section, whose contour is characterised by $\boldsymbol{p}_o$'s. Thus the time variation of ${}^r m_{pq}$ and ${}^o m_{pq}$ write according to (18) as follows:

$${}^r\dot{m}_{pq} = \boldsymbol{L}_{m_{pq},r}\, \dot{\boldsymbol{q}}, \quad \text{and} \quad {}^o\dot{m}_{pq} = \boldsymbol{L}_{m_{pq},o}\, \mathbf{v}. \tag{19}$$

where $\mathbf{v} = (\boldsymbol{v}^\top \ \boldsymbol{\omega}^\top)^\top \in \mathbb{R}^6$.

3) Image moments-based visual error: The objective of adopting image moments in this paper is to realise image-based visual servoing while at the same time mastering the 3D interaction. More precisely, to endow the robotic system with the capability of making the assembly structure perpendicularly approaches the target so as to have a practically-feasible 3D motion. To do so, let the visual feature vector be[1]

$$\boldsymbol{\eta} = \begin{pmatrix} \sqrt{a} & \overline{x} & \overline{y} \end{pmatrix}^\top \in \mathbb{R}^3, \tag{20}$$

and the visual feedback error be

$$\boldsymbol{e}_m = \boldsymbol{\eta}_r - \boldsymbol{\eta}_o, \tag{21}$$

such that $\boldsymbol{\eta}_r$ and $\boldsymbol{\eta}_o$ correspond to $\boldsymbol{\eta}$ calculated on the assembly structure and the target, respectively. The system thus can be assigned to nullify first the error about the center of gravity (the last two elements of $\boldsymbol{e}_m$). This will yield the structure and the target aligned. Then, while that error is kept null the error on the area is decreased to zero (the first element of $\boldsymbol{e}_m$). This finally allows approaching the target perpendicularly. Remark that only three independent visual features have been selected. This in fact shows to be enough to position the assembly structure into a desired pose (position and orientation) when employing a dual ARS with a rigid grasp. While it is clear that only six independent features are required to set the rigid object into a desired pose, each ARS will contribute with three features to position the object, yielding indeed a total of to six features. Time differentiating (20), taking into account (16), and substituting with (19) yields

$$\dot{\boldsymbol{\eta}}_r = \boldsymbol{L}_{m,r}\, \dot{\boldsymbol{q}} \quad \text{and} \quad \dot{\boldsymbol{\eta}}_o = \boldsymbol{L}_{m,o}\, \mathbf{v}. \tag{22}$$

The above developments were presented for only one ARS. Now let us take back subscript k to denote the k-th ARS. The visual error defined by (21) now rewrites

$$\boldsymbol{e}_{m,k} = \boldsymbol{\eta}_{r_k} - \boldsymbol{\eta}_{o_k}. \tag{23}$$

Then, time variation $\dot{\boldsymbol{e}}_{m,k}$ of visual error $\boldsymbol{e}_{m,k}$ writes as a function of generalised velocity $\dot{\boldsymbol{\xi}}_k$ as follows from (23) and (22):

$$\dot{\boldsymbol{e}}_{m,k} = \boldsymbol{J}_{m,k}\, \dot{\boldsymbol{\xi}}_k - \overline{\boldsymbol{L}}_{m,\omega_k}\, \overline{\boldsymbol{\omega}}_k, \tag{24}$$

where recall that $\overline{\boldsymbol{\omega}}_k = (\omega_{kx} \ \omega_{ky})^\top \in \mathbb{R}^2$.

4) Moments-based visual servoing: Nullifying visual error $\boldsymbol{e}_{m,k}$ can be achieved by applying the following velocity to the k-th ARS:

$$\dot{\boldsymbol{\xi}}_k = \boldsymbol{J}^+_{m,k} \left(\overline{\boldsymbol{L}}_{m,\omega_k} \overline{\boldsymbol{\omega}}_k - \lambda_{m,k}\, \boldsymbol{e}_{m,k} \right) + \boldsymbol{P}_{mk}\, \dot{\boldsymbol{\xi}}_{N_k}, \tag{25}$$

where $\boldsymbol{P}_{mk} = \boldsymbol{I}_{4+n_k} - \boldsymbol{J}^+_{m,k} \boldsymbol{J}_{m,k}$ is a null space projector of $\boldsymbol{J}_{m,k}$ to consider secondary tasks $\dot{\boldsymbol{\xi}}_{N_k}$. Matrix $\boldsymbol{J}^+_{m,k}$ is a weighted pseudo inverse of Jacobian $\boldsymbol{J}_{m,k}$ that can be formulated:

$$\boldsymbol{J}^+_k = \boldsymbol{W}_k\, \boldsymbol{J}^\top_k\, (\boldsymbol{J}_k\, \boldsymbol{W}_k\, \boldsymbol{J}^\top_k)^{-1}, \tag{26}$$

[1]The square root of a is considered so as to have the elements of $\boldsymbol{\eta}$ of the same unit, thus homogeneity.

where $\boldsymbol{W}_k$ is a diagonal weighting matrix to give more priorities to certain DoFs in nullifying $\boldsymbol{e}_k$. For instance, if the ARS is in a far location and $\boldsymbol{W}_k$ is set to identity, both UAV base and manipulator would be equivalently involved to nullify $\boldsymbol{e}_k$. The manipulator would outstretch, reaching its joint limits. Weighting matrix $\boldsymbol{W}_k$ is inserted to avoid such scenarios, and in fact it is used to exploit the ARS' base and manipulator complementarily. It would enable the ARS reaching far targets by forcing the system using only the UAV base, by forcing to zero the weights associated with the manipulator. On the other hand, once in the vicinity of the target, the manipulator can be yield the main actor to achieve the task, by setting to zero only the weights associated to the UAV base. Parameter $\lambda_{m,k}$ is a positive control gain.

Finally, for the robotic system to automatically position the assembly structure, cooperatively, with, more particularly, following only perpendicular motions during the approaching phase (monitored 3D interaction), the control system first calculates $\dot{\boldsymbol{\xi}}_1$ and $\dot{\boldsymbol{\xi}}_2$ according to (25), stacks them inside task velocity $\dot{\boldsymbol{\xi}}_T = \begin{pmatrix} \dot{\boldsymbol{\xi}}^\top_1 & \dot{\boldsymbol{\xi}}^\top_2 \end{pmatrix}^\top \in \mathbb{R}^{n'}$, which then is plugged into (13) to guarantee the rigid CCC.

IV. Results

The proposed approach has been tested via numerical simulations, as reported in the present section. The robotic system consists of a dual flying manipulator holding rigidly a bar (see Fig. 1). Each ARS consists of a quadrotor to which is attached a 5 DoF (n_1=n_2=5) anthropomorphic manipulator. A video camera is also attached to each quadrotor. A total of 8 landmarks (denoted $\boldsymbol{p}_r$'s in the above developments) characterise the bar: 4 landmarks from each side observed by one of the onboard cameras as depicted in Fig. 1. Likewise, the target on which the bar needs to be plugged on consists of 2 sets of 4 markers each (denoted $\boldsymbol{p}_o$'s). Each set, whether lying on the actuated bar or on the target, characterises a rectangle of 10x20 cm segment lengths. The bar is of 2 m length. The proposed method as well as the mathematical models emulating the two ARSs holding rigidly the bar have been implemented in C++ programming language. Each ARS is emulated with a separate C++ code, such that data transmission is managed with ROS[2]. In our simulations, the controller runs onboard only the second vehicle so as to avoid duplication, thus saving the computational resources of the first vehicle for other eventual demands and applications. The first ARS sends in real-time its state information and the images from its onboard camera to the second ARS through a ROS topic. Those data along with the second ARS's state information and the images from its onboard camera are used by the controller that delivers reference velocities $\dot{\boldsymbol{\xi}}_{c1}$ and $\dot{\boldsymbol{\xi}}_{c2}$ to the two ARSs. Command $\dot{\boldsymbol{\xi}}_{c1}$ is finally sent back to the first ARS via another ROS topic. The onboard cameras separately stream images at the cadence of 25 Hz, while the two ARSs update at 1 KHz, as typically afforded by the state-of-the-art technology.

[2]http://www.ros.org/

Note that since the convergence rate is proportional to the visual error norm as can be deduced from control law (25), the system would be increasingly slow when close to the target. To overcome this, we propose to employ a variable control gain:

$$\lambda_m = \begin{cases} \underline{\lambda} & \text{if} \quad |\boldsymbol{e}_m| > \overline{e}_m \\ \min\left(\lambda_d \tanh\left(\frac{\beta}{|\boldsymbol{e}_{m,1}|+|\boldsymbol{e}_{m,2}|}\right), \overline{\lambda}\right) & \text{otherwise,} \end{cases} \tag{27}$$

where $|\cdot|$ denotes the Euclidean norm. When the norm of one of the visual errors is above threshold $\overline{e}_m$, the control gain takes minimum value $\underline{\lambda}$. If however the error norms are under that threshold, then the control gain increases inversely to that norm. Parameters λ_d and β set the profile according to which the control gain evolves, with $\overline{\lambda}$ the maximum value it can take. Finally, weighting matrix $\boldsymbol{W}_k$ is inserted to leverage the mobility of the UAV base and the dexterity of the manipulator complementarily. To do so, its elements are set according to norm $|\boldsymbol{e}_m|$:

$$\boldsymbol{W}_k = \text{diag}(\gamma \boldsymbol{I}_5, \eta \boldsymbol{I}_4) \quad \text{such that}$$
$$\begin{cases} \gamma = 0.1, \quad \eta = 0.9 \quad \text{if} \quad |\boldsymbol{e}_{m,k}| > \overline{e}_m \\ \gamma = 0.9, \quad \eta = 0.1 \quad \text{otherwise.} \end{cases} \tag{28}$$

When the error is above threshold $\overline{e}_m$, the UAV base's weights are predominant, which forces the controller to mainly using it. In contrast, when close to the target the manipulator's weights are larger, and thus it is the main actor for achieving the task.

The target location on which the bar is to be positioned is centerred with respect to the world frame. It is represented by two 10x20 cm rectangles. At the initial configuration, just before visual servoing is launched, the first ARS' UAV base is located at (2, -2, 2) m from the world frame, while the first UAV at (4, 0, 2) m. The two manipulators start from the same configuration, yielding the carried bar to be initially horizontal. In order to make the task even more complex, the female target-bar is tilted by 30 deg rotation around each of the three axes of the world frame, thus requiring both the mobility of the UAV and the dexterity of the manipulator to position the bar. More particularly, by tilting the desired location the second ARS becomes farther from the target than the first ARS is, as sketched in Fig. 1. This allows to test the effectiveness of the proposed image-based cooperative control scheme, since achieving the task necessitates not simply the same amount of motion from the two ARSs. Corresponding results[3] are shown in Fig. 3. We can see from Fig. 3(a) that visual errors $\boldsymbol{e}_{m,1}$ and $\boldsymbol{e}_{m,2}$ automatically converge to zero, which means that the two sections (two rectangular tags) of the carried bar have been successfully superimposed on their associated ones of the target. This can be verified indeed from Fig. 3(b) plotting the norm of the 3D positioning errors, i.e. the error between the 3D coordinates of $\boldsymbol{p}_r$'s and $\boldsymbol{p}_o$'s in a fixed Cartesian world

[3]An animation video accompanies the paper.

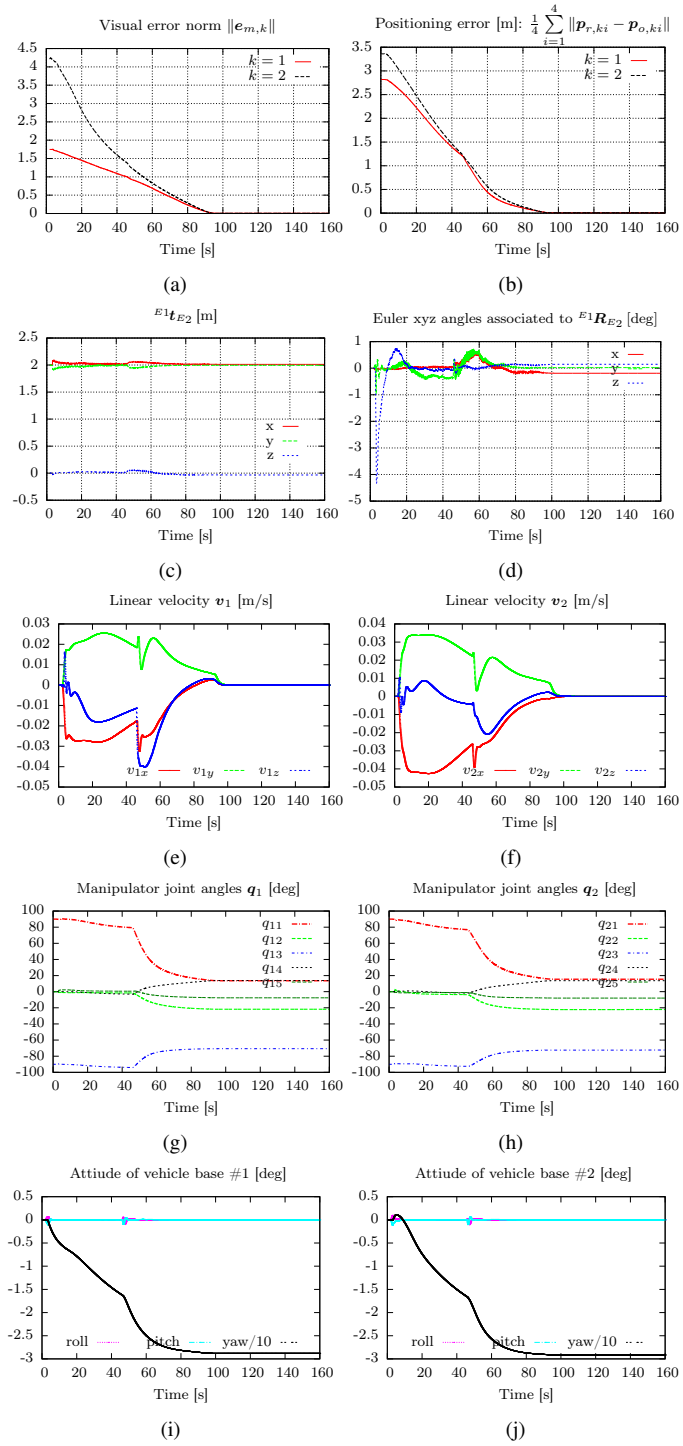

Fig. 3. Image-based cooperative aerial manipulation. A dual flying manipulator holding rigidly an assembly structure (a bar in this case, as depicted in Fig. 1) receives commands according to (25) and (13) in order to automatically position the bar on its visual target.

frame. Notice from Figs. 3(a) and 3(b) the initial difference in the amount between the errors associated with the two ARSs. The motions performed by the first ARS are shown in Figs. 3(e), 3(i), and 3(g). The former figure shows the linear velocity of the quadrotor, the second figure its attitude, while the latter plots the evolution of the manipulator joint

angles. Also, the motion of the second ARS is shown by 3(f), 3(j), and 3(h). One can see how these motions are smooth and in an acceptable range. Finally, the relative pose (${}^{E1}\boldsymbol{t}_{E2}$ and ${}^{E1}\boldsymbol{R}_{E2}$) between the end-effectors of the two ARSs is depicted in Figs. 3(c) and 3(d). Only very small variations are observed, which means that the rigid CCC is respected. In fact, these slight relative displacements are mainly due to the numerical integration, since only first Euler algorithm has been employed. In a real scenario, a low-level impedance or force control will work to limitate internal forces due to errors of the motion controller. These results thus show that the approach proposed in this paper allows a dual flying manipulator automatically positioning a rigidly-actuated assembly structure on its visual target. Furthermore, by adopting image moments the 3D interaction is controlled: during the last approaching phase the assembly structure is aligned with the target[3].

V. Conclusion

The paper addressed image-based visual control to endow aerial manipulators with the capability of automatically and cooperatively positioning rigidly-grasped assembly structures on visual targets. Two ARSs have been considered, each consisting of a quadrotor equipped with a video camera and carrying a manipulator. We first derived the rigid closed-chain constraint in an amenable form from a kinematic, cooperative, and image-based control point of view. Then, image-based visual servoing for a single ARS has been revised and adapted for multi-vehicles. We proposed to exploit image moments as control feedback to handle the motions of the actuated object, such that for the sake of amenable and compliant 3D motions, it was desired that the assembly structure be aligned with the target during the approaching phase. The obtained results from numerical simulations show the validity of the proposed approach.

References

[1] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, Modeling, Estimation and Control for Aerial Grasping and Manipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2668–2673, 2011.

[2] M. Bernard, K. Kondak, I. Maza, and A. Ollero, "Autonomous transportation and deployment with aerial robots for search and rescue missions," *Journal of Field Robotics*, vol. 28, no. 6, pp. 914–931, 2011.

[3] L. Marconi, F. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Hurzeler, V. Lippiello, R. Naldi, J. Nikolic, B. Siciliano, S. Stramigioli, and E. Zwicker, "Aerial service robotics: The AIRobots perspective," in *2nd Int. Conf. on Applied Robotics for the Power Industry*, pp. 64–69, 2012.

[4] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[5] M. Manubens, D. Devaurs, L. Ros, and J. Cortes, "Motion planning for 6-D manipulation with aerial towed-cable systems," in *Robotics: Science and Systems*, 2013.

[6] S. Bellens, J. D. Schutter, and H. Bruyninckx, "A Hybrid Pose / Wrench Control Framework for Quadrotor Helicopters," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2269–2274, 2012.

[7] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping From the Air: Hovering Capture and Load Stability," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2491–2498, 2011.

[8] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and Control of a Flying Robot for Contact Inspection," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 3532–3537, 2012.

[9] J. L. Scholten, M. Fumagalli, S. Stramigioli, and R. Carloni, "Interaction control of an UAV endowed with a manipulator," in *IEEE Int. Conf. on Robotics and Automation*, pp. 4910–4915, May 2013.

[10] C. Korpela, P. Brahmbhatt, M. Orsag, and P. Oh, "Towards the realization of mobile manipulating unmanned aerial vehicles (MM-UAV): Peg-in-hole insertion tasks," in *IEEE Int. Conf. on Technologies for Practical Robot Applications*, pp. 1–6, 2013.

[11] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Lyapunov based Model Reference Adaptive Control for Aerial Manipulation," in *Int. Conf. on Unmanned Aircraft Systems*, pp. 966–973, 2013.

[12] R. Mebarki, V. Lippiello, and B. Siciliano, "Image-based control for dynamically cross-coupled aerial manipulation," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4827–4833, 2014.

[13] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero, "Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2107–2112, May 2014.

[14] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 651–670, October 1996.

[15] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 82–90, 2006.

[16] R. Mebarki and V. Lippiello, "Image-based control for aerial manipulation," *Asian Journal of Control*, vol. 16, pp. 646–656, May 2014.

[17] R. Mebarki, V. Lippiello, and B. Siciliano, "Exploiting image moments for aerial manipulation control," in *ASME Dynamic Systems and Control Conf.*, Oct. 2013.

[18] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Six-dof impedance control based on angle/axis representations," *IEEE Trans. on Robotics and Automation*, vol. 15, pp. 289–300, April 1999.

[19] N. Hogan, "Impedance control: An approach to manipulation: Parts I–III," *ASME J. Dyn. Syst., Meas., Control.*, vol. 107, pp. 1–24, 1985.

[20] R. Bonitz and T. Hsia, "Internal force-based impedance control for cooperating manipulators," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 1, pp. 78–89, 1996.

[21] S. Schneider and J. Cannon, R.H., "Object impedance control for cooperative manipulation: theory and experimental results," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.

[22] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, February 1962.

[23] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. on Robotics*, vol. 20, pp. 713–723, August 2004.

[24] R. Mebarki, A. Krupa, and F. Chaumette, "2D ultrasound probe complete guidance by visual servoing using image moments," *IEEE Trans. on Robotics*, vol. 26, pp. 296–306, April 2010.

[25] C. Steger, "On the calculation of arbitrary moments of polygons," Technical Report FGBV–96–05, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, Technische Universität München, Oct. 1996.