2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Image-based control for dynamically cross-coupled aerial manipulation

Rafik Mebarki, Vincenzo Lippiello, and Bruno Siciliano

Abstract— **Manipulation tasks carried out with aerial platforms composed of a UAV and a robotic arm involve cross-coupled dynamics between these subsystems. This paper proposes a new controller for this class of aerial robotic systems that allows regulating on velocity commands generated by an outer image-based visual-servo scheme. The controller, that considers the full dynamics of the system, is designed based on the integral backstepping approach. Visual feedback provided by an onboard camera is employed into a new visual servo scheme to simultaneously generate velocity commands for the UAV and the manipulator so that a visual target is reached. The control system takes into account the under-actuation related to rotary-wing vehicles, while at the same time it exploits the functionality system redundancy to achieve the task. Simulation results validate the proposed control system, as well as its robustness to large modeling error and measurement noise.**

I. INTRODUCTION

Robotic systems are being brought to operate on remote regions of interest for applications ranging from sensing to manipulation. Certain regions, as for instance an electrical tower on a top of a mountain, can be difficult to access by land. Aerial robotics systems (ARS) are able to address this issue with the advantage that no scaffolding nor crane tower would be required [1]. Rotary-wing unmanned aerial vehicles (UAVs) present the advantage of hovering capabilities and thus allow a potential physical interaction with the environment. While most of the reported research works have focused on using UAVs for sensing [2], only recently few are investigating aerial manipulation [3].

In [4] a controller is developed so that a quadrotor can exert a desired force on the environment. The effects of an extra mass when carried by a micro helicopter have been investigated in [5]. In [2], a control system for autonomous load transportation with three small-scale helicopters has been proposed. The cooperative 3D manipulation of an object cable-towed to three UAVs is presented in [6] such that micro quadrotors have been employed for indoor experiments with the guidance of VICON capture system. In [7] a motion planner for 6 degrees-of-freedom (DoFs) object manipulation is shown. Note nevertheless that none of these aforementioned works considers direct manipulation.

The research leading to these results has been supported by the 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 solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

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@unina.it, vincenzo.lippiello@unina.it and bruno.siciliano@unina.it

In [3] an ARS composed of a quadrotor and a parallel manipulator is proposed for wall inspection. An impedance-based controller is designed to deliver the necessary torques to the quadrotor in order to position a carried probe at a desired position, while at the same time exerting a force against the wall. The adoption of an ARS composed of a rotary-wing UAV carrying a 7 DoFs manipulator for peg-in-hole insertion tasks has been proposed in [8], but only the kinematic interaction was considered. In [9] the authors deal with the cross-dynamics effect within an ARS composed of a helicopter carrying two symmetrical manipulators. A low-level controller is proposed to stabilise the UAV attitude when the manipulators are moving.

In the present paper we deal with an ARS composed of a rotary-wing UAV, namely a quadrotor, endowed with an n DoFs robotic manipulator (see Fig. 1). Our contributions are twofold. We first present a nonlinear low-level motion controller for the ARS that considers the full internal cross-dynamics. The controller more precisely endows the ARS with the capability of following velocity commands: three translational and yaw (heading) velocities for the UAV, and a joint rate for each of the manipulator n DoFs. It is designed based on the integral backstepping approach [10]. Then we present a high-level image-based visual servoing scheme that endows the ARS with the capability of reaching visual targets, i.e. pre-grasping phase. As can be seen later, the scheme is based on a new visual error definition, such that it allows considering simultaneous control of both the UAV and manipulator for task achievement. The visual control scheme operates at the kinematic level such that is generates velocity commands to both the UAV and the manipulator simultaneously. It takes into account the peculiarity of under-actuation of rotorcrafts and, on the other hand, it exploits the redundancy afforded by the whole system. In addition, avoidance of the manipulator joint limits is guaranteed. The generated velocity commands are then achieved by the ARS thanks to the low-level controller we introduced above. To the authors knowledge this is the first time that the simultaneous control of a quadrotor and the manipulator it carries is addressed taking into account the internal cross-dynamics.

II. DYNAMIC MODEL

Let $\boldsymbol{p}_b$ and $\boldsymbol{\phi}_b$ be the UAV's position and orientation Euler angles in fixed Cartesian base frame $\{R_b\}$, respectively. Let $\boldsymbol{q}$ be the joint-angles vector of the n DoFs robot manipulator attached to the vehicle. Denoting $\boldsymbol{\xi}$ as a

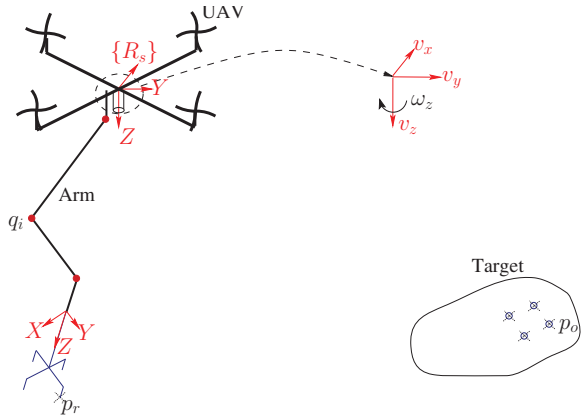

Fig. 1. Quadrotor equipped with a robot manipulator holding an assembly part. The objective is to automatically position the latter on the visual target.

generalised system state vector, such that

$$\boldsymbol{\xi} = \left(\boldsymbol{p}_b^\top \;\; \boldsymbol{\phi}_b^\top \;\; \boldsymbol{q}^\top\right)^\top \in \mathbb{R}^{6+n}, \tag{1}$$

the equations of motions can be described by the following analytical model [11], that takes into account the dynamic cross-coupling between the UAV and the manipulator:

$$\boldsymbol{B}(\boldsymbol{\xi})\,\ddot{\boldsymbol{\xi}} + \boldsymbol{C}(\boldsymbol{\xi},\dot{\boldsymbol{\xi}})\,\dot{\boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi}) = \bar{\boldsymbol{u}} \in \mathbb{R}^{6+n}, \tag{2}$$

where $\bar{\boldsymbol{u}}$ represents the generalised control input [11]. It encloses total thrust $u \in \mathbb{R}$ and torque $\bar{\Gamma} \in \mathbb{R}^3$ input commands to the UAV, and torque $\boldsymbol{\tau}_a \in \mathbb{R}^n$ input command to the manipulator. Matrices $\boldsymbol{B}$, and $\boldsymbol{C} \in \mathbb{R}^{(6+n)\times(6+n)}$ represent the generalised inertia and Coriolis matrices, respectively, while $\boldsymbol{g} \in \mathbb{R}^{6+n}$ the generalised gravitational vector.

III. LOW-LEVEL CONTROL

The objective is to design a controller so as to endow the system with the capability of following velocity references. More precisely, let ${}^b\boldsymbol{v} = \dot{\boldsymbol{p}}_b$ be the vehicle translational velocity, ${}^b\omega_z$ its yaw (heading) velocity also in $\{R_b\}$, and $\dot{\boldsymbol{q}}$ the time variation of $\boldsymbol{q}$. The objective is to analytically derive u, $\bar{\Gamma}$, and $\boldsymbol{\tau}_a$ such that ${}^b\boldsymbol{v}$, ${}^b\omega_z$, and $\boldsymbol{q}$ regulate to their respective desired commands ${}^b\boldsymbol{v}^*$, ${}^b\omega_z^*$, and $\boldsymbol{q}^*$. To achieve this we propose in the following a nonlinear controller leveraging the integral backstepping approach. Note that backstepping techniques have already been considered for image-based control of quadrotor UAVs, e.g. in [12][13]. In the present work, ${}^b\boldsymbol{v}^*$, ${}^b\omega_z^*$, and $\boldsymbol{q}^*$ are generated by a high-level visual servoing scheme. This scheme updates at the onboard camera's frequency, typically 25 Hz, while the nonlinear controller is more faster and generates torque commands roughly at 1 KHz. In addition, the velocity commands are expected to be continuous and smooth in such a way the system slowly approaches and enter in contact with the assembly part. Thus, at the nonlinear controller pace, these velocity commands would appear as very slowly varying. Accordingly, the assumption $\frac{d}{dt}\,{}^b\boldsymbol{v}^* \approx \boldsymbol{0}$ holds. We first formulate the generalised input command involved in (2) as follows:

$$\bar{\boldsymbol{u}} = \left((u\;{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1)^\top \quad \bar{\Gamma}^\top \quad \boldsymbol{\tau}_a^\top\right)^\top, \tag{3}$$

where ${}^b\boldsymbol{R}_u$ is the rotation matrix defining the orientation of UAV body frame $\{R_u\}$ with respect to $\{R_b\}$ and $\boldsymbol{e}_1 = (0\;\;0\;\;1)^\top \in \mathbb{R}^3$. First, let the regulation error on the translational velocity be $\boldsymbol{\delta}_1 = {}^b\boldsymbol{v} - {}^b\boldsymbol{v}^* \in \mathbb{R}^3$, and consider the following Lyapunov storage function:

$$V_1 = \frac{1}{2}|\boldsymbol{\delta}_1|^2 + \frac{1}{2}\,\alpha_1\,|\boldsymbol{\sigma}_1|^2 \quad \text{where} \quad \dot{\boldsymbol{\sigma}}_1 = \boldsymbol{\delta}_1, \tag{4}$$

with α_1 a constant positive gain, and where $|\cdot|$ denotes the Euclidean norm. Element $\boldsymbol{\sigma}_1$ corresponds to the integral of $\boldsymbol{\delta}_1$. It is inserted within V_1 in order to compensate for modeling errors and perturbations that can be caused by wind gust, battery depletion, temperature change, etc.

Let us take back relationship (2) describing the system dynamics, and recast the involved matrices as follows:

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} & \boldsymbol{B}_{13} \\ & \overline{\boldsymbol{B}} & \end{pmatrix}, \;\; \boldsymbol{C}\,\dot{\boldsymbol{\xi}} = \begin{pmatrix} \boldsymbol{c}_1 \\ \overline{\boldsymbol{c}} \end{pmatrix}, \;\; \boldsymbol{g} = \begin{pmatrix} \boldsymbol{g}_1 \\ \overline{\boldsymbol{g}} \end{pmatrix}, \tag{5}$$

such that $(\boldsymbol{B}_{11}, \boldsymbol{B}_{12}) \in \mathbb{R}^{3\times3}$, $\boldsymbol{B}_{13} \in \mathbb{R}^{3\times n}$, $\boldsymbol{c}_1 \in \mathbb{R}^3$, and $\boldsymbol{g}_1 \in \mathbb{R}^3$. Accordingly, the first three rows of (2) can be written as follows:

$$\boldsymbol{B}_{11}\,{}^b\dot{\boldsymbol{v}} + \boldsymbol{B}_{12}\,\ddot{\boldsymbol{\phi}}_b + \boldsymbol{B}_{13}\,\ddot{\boldsymbol{q}} + \boldsymbol{c}_1 + \boldsymbol{g}_1 = u\;{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1. \tag{6}$$

Taking this into account, time variation of error $\boldsymbol{\delta}_1$ writes

$$\begin{aligned} \dot{\boldsymbol{\delta}}_1 &= {}^b\dot{\boldsymbol{v}} = \boldsymbol{B}_{11}^{-1}\boldsymbol{B}_{11}\,{}^b\dot{\boldsymbol{v}} \\ &= \boldsymbol{B}_{11}^{-1}\left(u\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1 - \boldsymbol{c}_1 - \boldsymbol{g}_1 - \boldsymbol{\varepsilon}_1\right), \end{aligned} \tag{7}$$

where $\boldsymbol{\varepsilon}_1 = \boldsymbol{B}_{12}\,\ddot{\boldsymbol{\phi}}_b + \boldsymbol{B}_{13}\,\ddot{\boldsymbol{q}} \in \mathbb{R}^3$. Note that $\boldsymbol{B}_{11}$ is invertible since it simply corresponds to the total mass of the system (see [11]). Substituting equation (7) into the time derivative of (4) yields

$$\begin{aligned} \dot{V}_1 &= \boldsymbol{\delta}_1^\top\,\dot{\boldsymbol{\delta}}_1 + \alpha\,\boldsymbol{\sigma}_1^\top\,\dot{\boldsymbol{\sigma}}_1 \\ &= \boldsymbol{\delta}_1^\top\,\boldsymbol{B}_{11}^{-1}\left(u\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1 - \boldsymbol{c}_1 - \boldsymbol{g}_1 - \boldsymbol{\varepsilon}_1\right) + \alpha_1\boldsymbol{\sigma}_1^\top\,\boldsymbol{\delta}_1. \end{aligned} \tag{8}$$

We consider $\boldsymbol{\varepsilon}_1$ as an unknown force perturbing the system. It would be compensated for thanks to integral error $\boldsymbol{\sigma}_1$ as described in Appendix. Therefore, by setting

$$u\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1 := \boldsymbol{B}_{11}\left(-k_1\boldsymbol{\delta}_1 - \alpha_1\boldsymbol{\sigma}_1\right) + \boldsymbol{c}_1 + \boldsymbol{g}_1 \tag{9}$$

where k_1 is a positive control gain, (8) becomes:

$$\dot{V}_1 = -k_1|\boldsymbol{\delta}_1|^2 - \delta_1^\top\boldsymbol{B}_{11}\boldsymbol{\varepsilon}_1. \tag{10}$$

Notice that control command (9) cannot be directly achieved. Indeed, it represents a set of three desired values, where only the thrust u, which is a scalar, can be directly accessed and controlled. A second control error thus needs to be introduced. In order to achieve (9), we set

$$\boldsymbol{\delta}_2 = \boldsymbol{\delta}_1 + \frac{\alpha_1}{k_1}\boldsymbol{\sigma}_1 + \frac{1}{k_1}\,\boldsymbol{B}_{11}^{-1}\,u\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1 + \boldsymbol{a}_2 \quad \in \mathbb{R}^3, \tag{11}$$

with $\boldsymbol{a}_2 = -\frac{1}{k_1}\,\boldsymbol{B}_{11}^{-1}\left(\boldsymbol{c}_1 + \boldsymbol{g}_1\right) \in \mathbb{R}^3$. In order to nullify both $\boldsymbol{\delta}_2$ and $\boldsymbol{\delta}_1$, the following Lyapunov storage function is considered:

$$V_2 = V_1 + \frac{1}{2}|\boldsymbol{\delta}_2|^2 + \frac{1}{2}\alpha_2\,|\boldsymbol{\sigma}_2|^2, \;\text{ with } \dot{\boldsymbol{\sigma}}_2 = \boldsymbol{\delta}_2 \tag{12}$$

where α_2 is a positive gain. Vector $\boldsymbol{\sigma}_2$ corresponds to the integral of error $\boldsymbol{\delta}_2$. Similarly as for V_1, it is employed to compensate for modeling errors and eventual perturbations.

From (11), time variation $\dot{\boldsymbol{\delta}}_2$ of $\boldsymbol{\delta}_2$ is derived as follows:

$$\dot{\boldsymbol{\delta}}_2 = \dot{\boldsymbol{\delta}}_1 + \frac{\alpha_1}{k_1}\dot{\boldsymbol{\sigma}}_1 + \frac{1}{k_1}\,\boldsymbol{B}_{11}^{-1}\,\frac{d}{dt}\left(u\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1\right) + \dot{\boldsymbol{a}}_2. \quad (13)$$

We assume $\dot{\boldsymbol{a}}_2 = \boldsymbol{\varepsilon}_2$ as an unknown force disturbing the system which would be compensated for thanks to integral term $\boldsymbol{\sigma}_2$. Using the classical kinematic relationship [14] ${}^b\dot{\boldsymbol{R}}_u = {}^b\boldsymbol{R}_u\,[{}^u\boldsymbol{\omega}_u]_\times$, where ${}^u\omega_u = \left(\overline{\omega}_x\ \overline{\omega}_y\ \overline{\omega}_z\right)^\top \in \mathbb{R}^3$ corresponds to the UAV angular velocity expressed in its body frame $\{R_u\}$, and $[\boldsymbol{a}]_\times$ denotes the skew-symmetric matrix associated with vector $\boldsymbol{a}$, we have

$$\begin{aligned}\dot{\boldsymbol{\delta}}_2 &= k_1(\boldsymbol{\delta}_2 - \boldsymbol{\delta}_1) - \alpha_1\boldsymbol{\sigma}_1 + \tfrac{\alpha_1}{k_1}\boldsymbol{\delta}_1 + \boldsymbol{\varepsilon}_2 + \tfrac{1}{k_1}\,\boldsymbol{B}_{11}^{-1}\overline{\boldsymbol{F}} \\ &\text{with}\quad \overline{\boldsymbol{F}} = {}^b\boldsymbol{R}_u\left(u\,\overline{\omega}_y \quad -u\,\overline{\omega}_x \quad \dot{u}\right)^\top \in \mathbb{R}^3.\end{aligned} \quad (14)$$

Vector $\overline{\boldsymbol{F}}$ is considered as the control input to nullify $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$. This holds since $\dot{u}$ can be directly specified, while the attitude rate $(\bar{\omega}_x,\ \bar{\omega}_y)$ can be achieved with an inner-loop controller, after assuming a time-scale separation. This is more detailed thereafter in the next section. We thus set the desired value of $\overline{\boldsymbol{F}}$ as follows:

$$\begin{aligned}\overline{\boldsymbol{F}}^* &:= k_1\boldsymbol{B}_{11}\left(\alpha_1\boldsymbol{\sigma}_1 - (k_1+k_2)\boldsymbol{\delta}_2 - \tfrac{\alpha_1}{k_1}\,\boldsymbol{\delta}_1 - \alpha_2\boldsymbol{\sigma}_2\right) \\ &\text{with}\quad \overline{\boldsymbol{F}}^* = {}^b\boldsymbol{R}_u\left(u\,\overline{\omega}_y^* \quad -u\,\overline{\omega}_x^* \quad \dot{u}^*\right)^\top,\end{aligned} \quad (15)$$

such that k_2 is a positive control gain. As for $\dot{u}^*$, it corresponds to the desired value of thrust time variation $\dot{u}$, where $\overline{\omega}_x^*$ and $\overline{\omega}_y^*$ are the desired values of the UAV roll and pitch angular velocities $\overline{\omega}_x$ and $\overline{\omega}_y$, respectively. Notice that when $\overline{\boldsymbol{F}} = \overline{\boldsymbol{F}}^*$ one would have

$$\begin{aligned}\dot{V}_2 &= -k_1|\boldsymbol{\delta}_1|^2 - k_2|\boldsymbol{\delta}_2|^2 - \boldsymbol{\delta}_1^\top\boldsymbol{\varepsilon}_1 - \boldsymbol{\delta}_2^\top\boldsymbol{\varepsilon}_2 \\ &\leqslant -(k_1-\tfrac{1}{2})|\boldsymbol{\delta}_1|^2 - (k_2-\tfrac{1}{2})|\boldsymbol{\delta}_2|^2 + \tfrac{1}{2}|\boldsymbol{\varepsilon}_1|^2 + \tfrac{1}{2}|\boldsymbol{\varepsilon}_2|^2\end{aligned} \quad (16)$$

where the remaining error (offset) terms $|\boldsymbol{\varepsilon}_1|$ and $|\boldsymbol{\varepsilon}_2|$ are expected to be compensated for thanks to the integral actions $\alpha_1\,\boldsymbol{\sigma}_1$ and $\alpha_2\,\boldsymbol{\sigma}_2$ involved in $\overline{\boldsymbol{F}}^*$ above. More details can be found in Appendix.

A. Inner-loop control

Control law (15) defines an outer loop that assumes the UAV roll and pitch angular velocities reaches their desired values in relatively short period of time compared to the frequency at which this outer loop runs, i.e. the camera frame rate. Such an assumption referred to as *time-scale* separation is commonly adopted for rotary-wing vehicles control [12]. In the following, we derive the low-level controller to achieve these angular velocities as well as the regulation of the manipulator joint velocities.

Let us recall that the yaw velocity also needs to be regulated, as described above. In a general visual servoing framework [15], the yaw velocity ω_z is expressed in the Cartesian frame of the camera attached to the vehicle. Let $\{R_s\}$ denote this sensor frame. In our configuration, UAV body frame $\{R_u\}$ is set such that in the hovering case (horizontal) it points upward, while $\{R_s\}$ has its z-axis pointing downward. The vehicle angular velocity expressed in $\{R_s\}$ as $\boldsymbol{\omega} = (\omega_x\ \omega_y\ \omega_z)^\top$ writes as function of its value ${}^u\boldsymbol{\omega}_u$ in vehicle body frame $\{R_u\}$ as follows:

$$\boldsymbol{\omega} = {}^s\boldsymbol{\omega}_s = {}^s\boldsymbol{R}_u\,{}^u\boldsymbol{\omega}_u = \left(-\overline{\omega}_x \quad \overline{\omega}_y \quad -\overline{\omega}_z\right)^\top. \quad (17)$$

where ${}^s\boldsymbol{R}_u = \mathrm{diag}(-1,1,-1)$ is the rotation matrix from $\{R_s\}$ to $\{R_u\}$. It is clear that we have $\boldsymbol{\omega} = (-\overline{\omega}_x\ \overline{\omega}_y\ \omega_z)^\top$. Finally, the desired value $\boldsymbol{\omega}^*$ of the vehicle angular velocity in sensor frame is expressed in terms of $\overline{\omega}_x^*$, $\overline{\omega}_y^*$, and ω_z^*. Let us recall that the objective of the inner-loop is to regulate both $\boldsymbol{\omega}$ and $\dot{\boldsymbol{q}}$ to reach $\boldsymbol{\omega}^*$ and $\dot{\boldsymbol{q}}^*$. To this aim we rewrite model (2) as follows:

$$\ddot{\boldsymbol{\xi}} + \boldsymbol{h} = \boldsymbol{B}^{-1}\,\bar{\boldsymbol{u}}, \quad (18)$$

with $\boldsymbol{h} = \boldsymbol{B}^{-1}\,\boldsymbol{C}\,\dot{\boldsymbol{\xi}} + \boldsymbol{B}^{-1}\,\boldsymbol{g} \ \in \mathbb{R}^{n+6}$. Let

$$\boldsymbol{B}^{-1} = \begin{pmatrix}\overline{\boldsymbol{B}}_{11} & \overline{\boldsymbol{B}}_{12} & \overline{\boldsymbol{B}}_{13} \\ \overline{\boldsymbol{B}}_{21} & \overline{\boldsymbol{B}}_{22} & \overline{\boldsymbol{B}}_{23} \\ \overline{\boldsymbol{B}}_{31} & \overline{\boldsymbol{B}}_{32} & \overline{\boldsymbol{B}}_{33}\end{pmatrix}, \quad \text{and} \quad \boldsymbol{h} = \begin{pmatrix}\boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \boldsymbol{h}_3\end{pmatrix}. \quad (19)$$

Considering the last two block equations of (18), one has

$$\begin{cases}\ddot{\boldsymbol{\phi}}_b + \boldsymbol{h}_2 &= u\,\overline{\boldsymbol{B}}_{21}\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1 + \overline{\boldsymbol{B}}_{22}\overline{\Gamma} + \overline{\boldsymbol{B}}_{23}\boldsymbol{\tau}_a \\ \ddot{\boldsymbol{q}} + \boldsymbol{h}_3 &= u\,\overline{\boldsymbol{B}}_{31}\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1 + \overline{\boldsymbol{B}}_{32}\overline{\Gamma} + \overline{\boldsymbol{B}}_{33}\boldsymbol{\tau}_a.\end{cases} \quad (20)$$

Time variation $\dot{\boldsymbol{\phi}}_b$ of Euler angles vector can be expressed as a function of the vehicle angular velocity as [14] $\dot{\boldsymbol{\phi}}_b = \boldsymbol{T}^{-1}\,{}^b\boldsymbol{R}_s\,\boldsymbol{\omega}$. Time differentiating yields

$$\ddot{\boldsymbol{\phi}}_b = -\boldsymbol{T}^{-1}\dot{\boldsymbol{T}}\boldsymbol{T}^{-1}\,{}^b\boldsymbol{R}_s\,\boldsymbol{\omega} + \boldsymbol{T}^{-1}\,{}^b\boldsymbol{R}_s\,\dot{\boldsymbol{\omega}}. \quad (21)$$

Then, substituting (21) into (20) yields

$$\begin{cases}\boldsymbol{T}^{-1}\,{}^b\boldsymbol{R}_s\,\dot{\boldsymbol{\omega}} + \overline{\boldsymbol{h}}_2 = \overline{\boldsymbol{B}}_{22}\,\overline{\Gamma} + \overline{\boldsymbol{B}}_{23}\,\boldsymbol{\tau}_a \\ \ddot{\boldsymbol{q}} + \overline{\boldsymbol{h}_3} = \overline{\boldsymbol{B}}_{32}\,\overline{\Gamma} + \overline{\boldsymbol{B}}_{33}\,\boldsymbol{\tau}_a\,,\end{cases} \quad (22)$$

where

$$\begin{cases}\overline{\boldsymbol{h}}_2 &= \boldsymbol{h}_2 - \boldsymbol{T}^{-1}\dot{\boldsymbol{T}}\boldsymbol{T}^{-1}\,{}^b\boldsymbol{R}_s\,\boldsymbol{\omega} - u\,\overline{\boldsymbol{B}}_{21}\,{}^b\boldsymbol{R}_u\boldsymbol{e}_1 \\ \overline{\boldsymbol{h}}_3 &= \boldsymbol{h}_3 - u\,\overline{\boldsymbol{B}}_{31}\,{}^b\boldsymbol{R}_u\,\boldsymbol{e}_1.\end{cases} \quad (23)$$

Finally, (22) can be recast as follows:

$$\boldsymbol{G}\dot{\boldsymbol{\eta}} + \boldsymbol{b} = \boldsymbol{A}\boldsymbol{\tau} \quad (24)$$

with

$$\begin{cases}\boldsymbol{\eta} = \begin{pmatrix}\boldsymbol{\omega} \\ \dot{\boldsymbol{q}}\end{pmatrix} \in \mathbb{R}^{3+n} \quad \boldsymbol{\tau} = \begin{pmatrix}\overline{\Gamma} \\ \boldsymbol{\tau}_a\end{pmatrix} \in \mathbb{R}^{3+n} \quad \boldsymbol{b} = \begin{pmatrix}\overline{\boldsymbol{h}}_2 \\ \overline{\boldsymbol{h}}_3\end{pmatrix} \\ \boldsymbol{G} = \begin{pmatrix}\boldsymbol{T}^{-1}\,{}^b\boldsymbol{R}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_n\end{pmatrix}, \quad \boldsymbol{A} = \begin{pmatrix}\overline{\boldsymbol{B}}_{22} & \overline{\boldsymbol{B}}_{23} \\ \overline{\boldsymbol{B}}_{33} & \overline{\boldsymbol{B}}_{33}\end{pmatrix}\end{cases} \quad (25)$$

where $\boldsymbol{I}_n$ denotes the identity matrix of size n. Notice that $\boldsymbol{G}$ is always invertible (except when the UAV roll or pitch attitude reaches $\pm\frac{\pi}{2}$, a rare case that might happen only if the UAV destabilizes). The objective is to achieve the desired value $\boldsymbol{\eta}^* = (\boldsymbol{\omega}^{*\top}\ \dot{\boldsymbol{q}}^{*\top})^\top$ of $\boldsymbol{\eta}$. Thus, consider control error

$\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}^* \in \mathbb{R}^{3+n}$, while $\boldsymbol{\sigma}_r$ is its integral. Accordingly, based on the following Lyapunov function:

$$V_r = \frac{1}{2}|\tilde{\boldsymbol{\eta}}|^2 + \frac{1}{2}\,\boldsymbol{\alpha}_r\,|\boldsymbol{\sigma}_r|^2, \tag{26}$$

where $\boldsymbol{\alpha}_r$ is a $(3+n)$ diagonal positive gain matrix, we derive this command

$$\boldsymbol{\tau} = \boldsymbol{A}^{-1}\,\boldsymbol{b} + \boldsymbol{A}^{-1}\,\boldsymbol{G}\,\left(-\boldsymbol{K}_r\tilde{\boldsymbol{\eta}} - \boldsymbol{\alpha}_r\,\boldsymbol{\sigma}_r\right), \tag{27}$$

with $\boldsymbol{K}_r \in \mathbb{R}^{(n+3)\times(n+3)}$ a positive control gain (diagonal) matrix. We obtain $\dot{V}_r = -\tilde{\boldsymbol{\eta}}^\top\,\boldsymbol{K}_r\,\tilde{\boldsymbol{\eta}}$ definite negative. Notice that the integral term $\boldsymbol{\alpha}_r\,\boldsymbol{\sigma}_r$ is employed to compensate for modeling errors and eventual unknown perturbing forces.

IV. Visual servo controller

Consider an aerial robotic manipulator with a camera attached to the UAV (see Fig. 1). Such a configuration is referred to as "*onboard-eye-to-hand*" [16]. As sketched in Fig. 1, the manipulator actuates an assembly part that needs to be positioned on a visual target automatically. To achieve this, we propose in [16] an image-based servo scheme that simultaneously controls the UAV and the manipulator at the velocity level, such that a novel feedback error is proposed. Let the target and the assembly part be characterised by a set of m 3D points each. Let $\boldsymbol{p}_{oi}$ and $\boldsymbol{p}_{ri}$ be the i-th point of the target and assembly part, respectively. Let $\boldsymbol{s}_{oi}$ be the image coordinates of $\boldsymbol{p}_{oi}$, while $\boldsymbol{s}_{ri}$ be the *pseudo* image coordinates[1] of $\boldsymbol{p}_{ri}$. Thus consider feedback error $\boldsymbol{e}$ characterised as follows:

$$\boldsymbol{e} = [\boldsymbol{e}_1^\top\ \boldsymbol{e}_2^\top \ldots\ \boldsymbol{e}_m^\top]^\top,\ \text{such that}\ \boldsymbol{e}_i = \boldsymbol{s}_{ri} - \boldsymbol{s}_{oi}. \tag{28}$$

It is worth noting that although $\boldsymbol{e}_i$, and hence $\boldsymbol{e}$, bares resemblance to classical visual servoing error definition, there is in fact a large difference. Definition (28) implies that both target image $\boldsymbol{s}_{oi}$ and carried assembly image $\boldsymbol{s}_{ri}$ vary to finally superimpose on each other. In contrast, in classical visual servoing [15][17][18] only the image features of the observed target vary to reach a pre-defined (learned) position. Note in addition that with this new definition there is no need to learn the desired image features. Since only four DoFs of the quadrotor can be servoed, we express $\dot{\boldsymbol{e}}$ as a function of a new system generalised-velocity $\dot{\bar{\boldsymbol{\xi}}}$ as follows [16][2]:

$$\dot{\boldsymbol{e}} = \boldsymbol{J}\,\dot{\bar{\boldsymbol{\xi}}} - \bar{\boldsymbol{L}}_\omega\,\bar{\boldsymbol{\omega}},\ \dot{\bar{\boldsymbol{\xi}}} = (\dot{\boldsymbol{q}}^\top\ \boldsymbol{v}^\top\ \omega_z)^\top \in \mathbb{R}^{n+4} \tag{29}$$

where $\boldsymbol{v}$ is the UAV velocity expressed in camera frame $\{R_s\}$[3]. Matrix $\boldsymbol{J}$ represents the system Jacobian, which is a function of image coordinates $\boldsymbol{s}_{oi|_{i=1,..,m}}$ of the visual target points, their corresponding scale factors (image depth), and the manipulator odometry. As for matrix $\bar{\boldsymbol{L}}_\omega$, it is a function of image coordinates $\boldsymbol{s}_{oi}$'s, where $\bar{\boldsymbol{\omega}} = (\omega_x,\ \omega_y)^\top$ represents the rate of the UAV attitude (roll and pitch) expressed in the camera frame. To nullify $\boldsymbol{e}$, and thence automatically position the assembly part on the visual target, we can use [16]

$$\dot{\bar{\boldsymbol{\xi}}} := \boldsymbol{J}^+(\boldsymbol{L}_{\bar{\omega}}\,\bar{\boldsymbol{\omega}} - \lambda\,\boldsymbol{e}) + (\boldsymbol{I}_{n+4} - \boldsymbol{J}^+\boldsymbol{J})\,\dot{\bar{\boldsymbol{\xi}}}_N, \tag{30}$$

where λ is a positive control gain, and $\dot{\bar{\boldsymbol{\xi}}}_N$ corresponds to desired motions to achieve secondary tasks. The above controller generates desired value $\dot{\bar{\boldsymbol{\xi}}}^*$ of $\dot{\bar{\boldsymbol{\xi}}}$ given by $\dot{\bar{\boldsymbol{\xi}}}^* = (\dot{\boldsymbol{q}}^{*\top}\ \boldsymbol{v}^{*\top}\ \omega_z^*)^\top \in \mathbb{R}^{n+4}$, which consists of desired translational velocity and yaw angular velocity reference signals to the UAV, and desired joint velocities for the manipulator. These desired values are then inputs to a low-level controller. We use the controller described in the previous section. Matrix $\boldsymbol{J}^+$ is a weighted pseudo inverse of Jacobian $\boldsymbol{J}$ given by $\boldsymbol{J}^+ = \boldsymbol{W}\boldsymbol{J}^\top(\boldsymbol{J}\boldsymbol{W}\boldsymbol{J}^\top)^{-1}$. Diagonal weighting matrix $\boldsymbol{W}$ is inserted for two main reasons. It blocks a joint that reaches its limits, thus guaranteeing joint-limits avoidance. Secondly, il allows automatically exploiting only the UAV mobility when far from the target, since otherwise the manipulator would outstretch without any benefit to the task. In fact it allows exploiting the dexterity of the manipulator only when close to the target. Details on the design of $\boldsymbol{W}$ can be found in [16].

V. Results

The control methods developed in this paper have been implemented in C++ programming language. The present section reports computer simulation results showing the performance of the low-level controller to achieve desired velocity for the ARS, in addition of the high-level visual servo scheme for visually-guided aerial manipulation. It is assumed that both the UAV and manipulator update at a rate of 1 KHz, and that the onboard camera updates at a rate of 25 Hz as typically afforded by the state-of-the-art hardware. The robotic arm has $n = 5$ anthropomorphic joints.

A. Velocity control

The effectiveness of the proposed low-level controller, defined by (15) and (27), is verified in the following. To do so, sinusoidal reference velocities are sent to the ARS system. The gains for the outer-loop controller are set to $k_1 = 3$, $\alpha_1 = 5$, $k_2 = 10$, $\alpha_2 = 5$. As for the inner-loop control, $\boldsymbol{K}_r = 10\,\mathrm{diag}(\boldsymbol{I}_3, 100\,\boldsymbol{I}_5)$, and integral gain matrix to $\boldsymbol{\alpha}_r = 5\,\mathrm{diag}(1,1,1,500,500,2500,2500,500)$. We first consider an ideal scenario, where neither modeling errors nor external forces perturb the system. The corresponding results are shown in Fig. 2. In Figs. 2(a), 2(b), and 2(c) we can see that the commands ${}^b\boldsymbol{v}^*$ for the UAV translational velocity, ω_z^* for the yaw velocity, and $\dot{\boldsymbol{q}}^*$ for the manipulator joint rates, respectively, are well achieved. Now the case of modeling errors, and external forces perturbing the system is considered. To do so, the parameters involved in dynamic model (2) are corrupted with roughly 30% of modeling errors. In addition, an extra mass 100 times larger than the last link of the manipulator is added, to the end-effector more precisely, at time t =9 s. This is aimed at mimicking the arm grabbing an object of roughly 150 g weight. The

[1]Term "*pseudo*" is employed since $\boldsymbol{s}_{ri}$ is emulated and computed from the manipulator odometry and not necessarily from the image.

[2]Notice the difference between $\dot{\bar{\boldsymbol{\xi}}}$, that represents the control inputs to the visual-servo scheme, and system state $\dot{\boldsymbol{\xi}}$ involved in (2).

[3]We have $\boldsymbol{v} = {}^b\boldsymbol{R}_s^\top\,{}^b\boldsymbol{v}$, where ${}^b\boldsymbol{R}_s$ represents the rotation matrix from $\{R_b\}$ to $\{R_s\}$ and is obtained from the UAV attitude angles provided by the onboard IMU.

corresponding results are shown in Fig. 3. We can see the effect of this extra mass on velocity v_z, while the thrust compensates for it. The extra mass also affects the tracking of the joint rates as can be seen in Fig. 3(c). Nevertheless, the controller compensates well for this perturbation, and the tracking is resumed. This is afforded by the inserted integral terms in the control scheme. These results validate the robustness of the developed control system to both modeling errors and external forces.

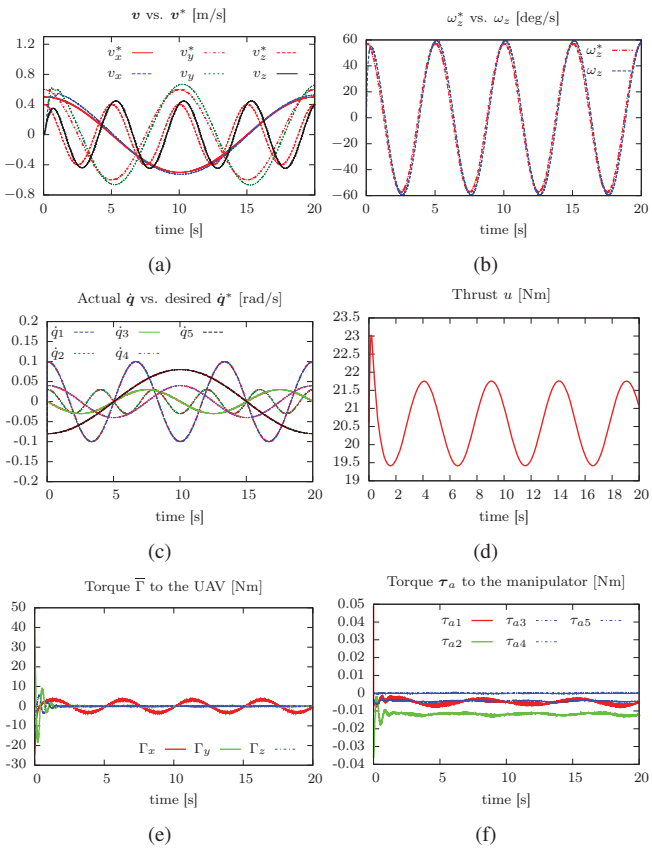

Fig. 2. Control of the ARS' velocity in an ideal case. The UAV and manipulator are simultaneously servoed.

B. Visual servoing

The robotic task now consists in automatically positioning an assembly part carried by the ARS on a visual target (Fig. 1). The target is characterised by four points that form a square of 20 cm segment length. Note that the square is tilted by an angle of 30 deg from the horizontal plane, in order to make the task even more complex. The control scheme proposed in Section IV is applied, such that the low-level control of the system is realised with the law derived in Section III. More precisely, the high-level image-based controller generates velocity references according to (30), references to be achieved by the low-level controller that outputs thrust and torque commands to the UAV, and torque commands to the manipulator according to (15) and (27). We first consider an ideal scenario where neither noise nor modeling errors occur in the system. This allows to

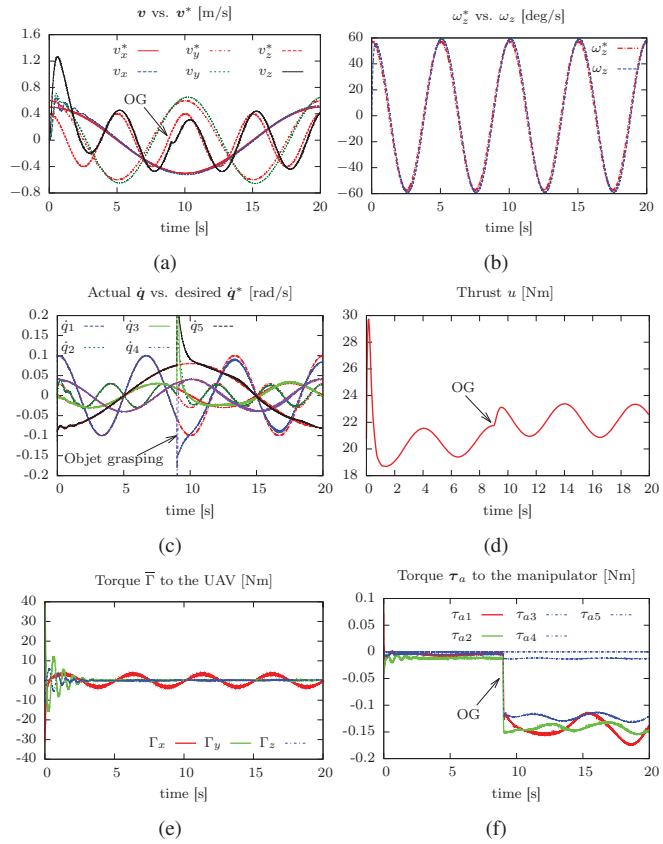

Fig. 3. Velocity control in a scenario of roughly 30% modeling errors, with an external force at time 9 s. This force mimics object grasping. Term "OG" is an abbreviation of "Object Grasped".

verify the validity of the derived controller and the effect it induces on the robotic system. Since the convergence rate is proportional to $\lambda \boldsymbol{e}$ as can be noticed from (30), we select gain λ as $\lambda = \max\left[\lambda_m,\ \lambda_M \tanh\left(\frac{\beta}{\|\boldsymbol{e}\|}\right)\right]$. This allows λ to compensate for the otherwise low convergence rate when $\boldsymbol{e}$ is small, such that it takes maximum value λ_M. In contrast, it decreases to λ_m when $\boldsymbol{e}$ is large. Parameter β defines the transition between λ_m and λ_M according to the error norm. In this simulation $\lambda_M = 1$, $\lambda_m = 0.01$, and $\beta = 0.1$. As for the gains of the low-level controller defined by (15) and (27), they are set as follows: $k_1 = 1$, $\alpha_1 = 1$, $k_2 = 10$, $\alpha_2 = 3$, $k_r = 10\,\mathrm{diag}(1, 1, 1, 10, 20, 10, 10, 10)$, $\alpha_r = \mathrm{diag}(\boldsymbol{I}_3,\ 10\boldsymbol{I}_5)$. The corresponding simulation results are shown in Fig. 4. Visual error $\boldsymbol{e}$ converges to zero as can be seen from Fig. 4(a), and the assembly part is well positioned on the target [Fig. 4(i)] where the corresponding error is shown in Fig. 4(f). The latter plots in fact the positioning error between 3D points $\boldsymbol{p}_{ri}$'s of the carried assembly part and points $\boldsymbol{p}_{oi}$'s of the target. As can be seen, this error goes to zero which means that the assembly part is well positioned on the visual target. The evolution of image points $\boldsymbol{s}_r$ and $\boldsymbol{s}_o$ are shown in Fig. 4(e), such that we can see both these two features varying to finally converge on each other. Such a behaviour characterises our new control paradigm which is, as highlighted earlier, quite different from

classical visual servoing. Also, we can see from Fig. 4(c) that the elements of weighting matrix $\boldsymbol{W}$ related to the UAV are higher at the beginning, but then decrease while those related to the manipulator start increasing. This, as desired, means that since at the beginning the ARS is far from the target (≈ 5 m), the control system exploits first the UAV mobility to reach the target and, once in the vicinity, the dexterity of the manipulator is exploited to finally reach (grasp) the target. These results clearly show the effectiveness of this novel visual servo scheme along with the low-level controller to track its generated reference velocities.

Now a more realistic scenario is considered, such that large modeling errors and measurement noises are introduced and added to all the measurements used in the control system. The attitude of the UAV and its angular velocity, both provided by the onboard IMU, are corrupted with additive white Gaussian noises (WGN)'s of 2 deg and 5 deg/s standard deviation (SD), respectively. Likewise the measure of the translational velocity is corrupted with a WGN of 5 cm/s. Also, the image coordinates of the visual features are corrupted with WGN's of 1 cm SD, while their image depths are corrupted with 5 cm SD. Notice that the latter noise is roughly large, since the image depth is typically a more complex value to extract from a 2D image. It is also important to note that the control system is provided with only an approximate of the dynamic model, such that large modeling errors of roughly 30% the nominal value are introduced. This allows us to test, in addition to sensor noises, the robustness of the control system with respect to dynamic and kinematic parameter errors. Corresponding results are shown in Fig. 5. We can deduce that the assembly part is well positioned on the target since the positioning error converges to zero. In addition, the behaviour of the system is smooth, considering the large modeling errors and noises corrupting the controller. This can be seen from the plotted thrust, UAV attitude, and the manipulator joint angles. These results clearly show the robustness of the proposed control system to both dynamic and kinematic modeling errors as well as to large measurements noise.

VI. CONCLUSIONS

A new image-based visual-servo scheme that endows a flying manipulator with the capability of positioning with respect to visual targets has been proposed. The control scheme takes into account the full cross-coupled dynamics of the ARS, while a nonlinear integral backstepping control ensures low-level servoing. By proposing a new visual error definition, the high-level controller can command both the UAV and manipulator simultaneously. Results from simulations provide a proof of concept of the approach, such that even an approximate of the dynamic model shows to be sufficient for the control. The control system is also robust to large measurements noise.

APPENDIX

In the following section, an analysis on the convergence of $\boldsymbol{\delta}_1$ is provided. First, it can be concluded from (27) that $\dot{\boldsymbol{q}}$ converges to constant $\dot{\boldsymbol{q}}^*$. This means that $\ddot{\boldsymbol{q}}$ becomes null. Taking this into account, and replacing (11) into (7) we have

$$\dot{\boldsymbol{\delta}}_1 = -k_1(\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2) - \alpha_1 \boldsymbol{\sigma}_1 - \boldsymbol{B}_{11}^{-1}\boldsymbol{B}_{12}\,\ddot{\boldsymbol{\phi}}_b. \tag{31}$$

Since the inner-loop is faster we can assume $\overline{\boldsymbol{F}}^*$ given by (15) as achieved. Using this in (14) yields

$$\dot{\boldsymbol{\delta}}_2 = -k_2\,\boldsymbol{\delta}_2 - k_1\,\boldsymbol{\delta}_1 + \boldsymbol{\varepsilon}_2 - \alpha_2\,\boldsymbol{\sigma}_2. \tag{32}$$

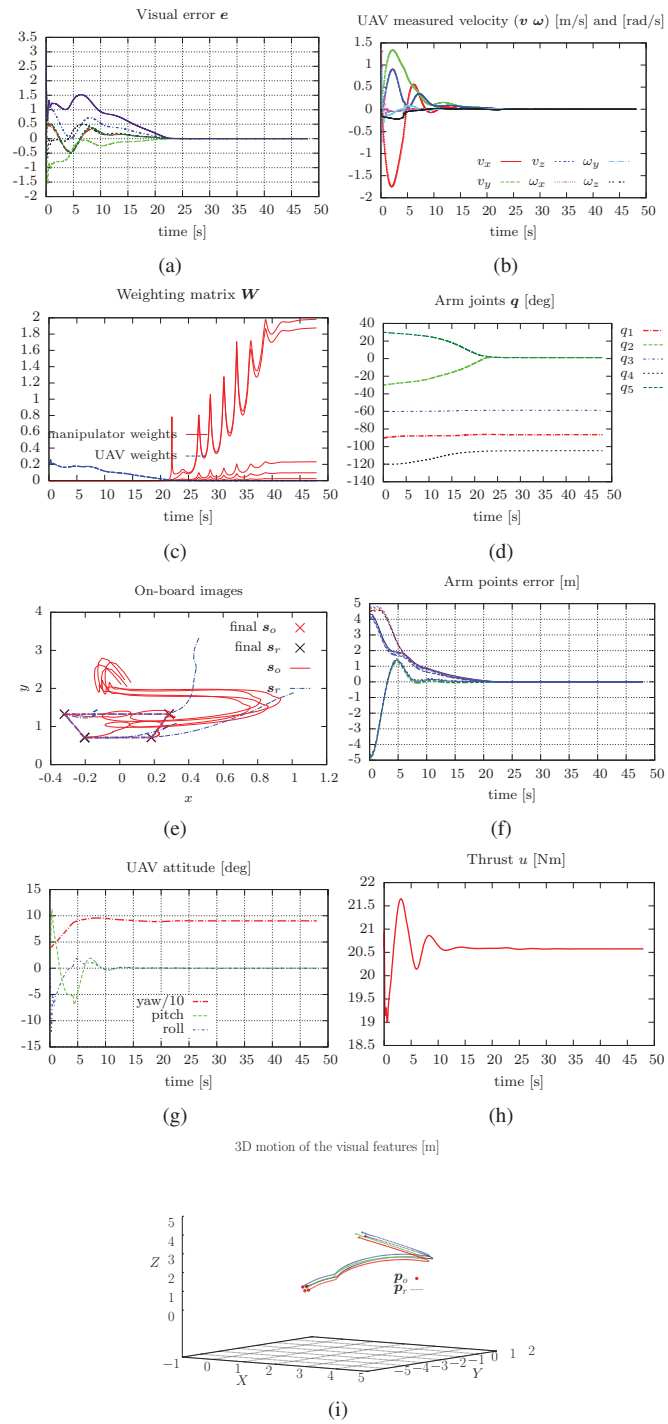

Fig. 4. Image-based control of the aerial robotic manipulator carrying an assembly part to automatically position on a visual target. The scenario is ideal in the sense that no modeling errors nor measurement noises exists.

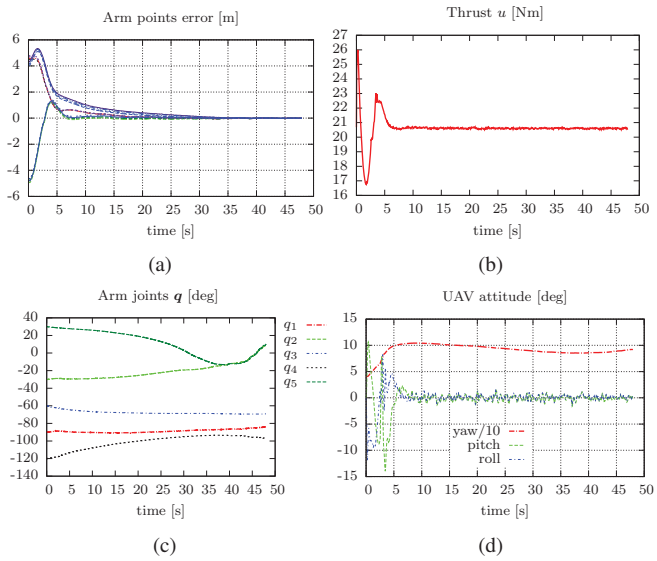

Fig. 5. Image-based control of the ARS in the scenario of large measurement noises and 30% modeling errors on the dynamic model.

It is clear that $\boldsymbol{\delta}_2$ converges to zero thanks to integral term $\boldsymbol{\sigma}_2$. Time derivative of $\overline{\boldsymbol{F}}$ from (14) can write in this form

$$\dot{\overline{\boldsymbol{F}}} = u\boldsymbol{A}\ddot{\phi}_b + \boldsymbol{\varepsilon}_3, \tag{33}$$

with $\boldsymbol{\varepsilon}_3$ a bounded term. Subsequently, since $\overline{\boldsymbol{F}} = \overline{\boldsymbol{F}}^*$, then $\dot{\overline{\boldsymbol{F}}}$ can be obtained from the time derivative of the first relationship's right hand in (15) as follows

$$\begin{aligned}\dot{\overline{\boldsymbol{F}}} = k_1\,\boldsymbol{B}_{11}\Big[&\alpha_1\,\boldsymbol{\delta}_1 - \frac{\alpha_1}{k_1}\,\dot{\boldsymbol{\delta}}_1 - \alpha_2\,\boldsymbol{\delta}_2 \\ &- (k_1+k_2)(-k_2\,\boldsymbol{\delta}_2 - k_1\boldsymbol{\delta}_1 + \boldsymbol{\varepsilon}_2 - \alpha_2\,\boldsymbol{\sigma}_2)\Big].\end{aligned} \tag{34}$$

Notice that due to the fact that $\boldsymbol{\delta}_2$ converges to zero as highlighted above, where perturbation force $\boldsymbol{\varepsilon}_2$ is compensated for by integral $\boldsymbol{\sigma}_2$, we drop both $\boldsymbol{\sigma}_2$ and $\boldsymbol{\delta}_2$ for simplicity of analysis. Then $\ddot{\phi}_b$ can write after plugging (34) into (33)

$$\ddot{\phi}_b = \frac{1}{u}\boldsymbol{A}^+\left(\boldsymbol{B}_{11}\left(k_1^2(k_1+k_2+\frac{\alpha_1}{k_1})\boldsymbol{\delta}_1 - \alpha_1\dot{\boldsymbol{\delta}}_1\right) - \boldsymbol{\varepsilon}_3\right), \tag{35}$$

with $\boldsymbol{A}^+$ a pseudo-inverse of $\boldsymbol{A}$. Finally, substituting $\ddot{\phi}_b$ with its above relationship in (6) yields

$$\begin{aligned}&\left(\boldsymbol{I}_3 - \frac{\alpha_1}{u}\boldsymbol{B}_{12}\,\boldsymbol{A}^+\right)\dot{\boldsymbol{\delta}}_1 + \alpha_1\left(\boldsymbol{I}_3 - k_1\,\boldsymbol{B}_{12}\,\boldsymbol{A}^+\,\boldsymbol{G}\right)\boldsymbol{\sigma}_1 \\ &+ k_1\left(1 + \boldsymbol{B}_{12}\,\boldsymbol{A}^+\,\frac{1}{u}\Big(k_1(k_1+k_2)+\alpha_1\Big)\right)\boldsymbol{\delta}_1 = d,\end{aligned} \tag{36}$$

with d a small term ($d \approx 0$). Recalling that $\dot{\boldsymbol{\delta}}_1 = \ddot{\boldsymbol{\sigma}}_1$, the above relationship takes on this form

$$a\,\ddot{\boldsymbol{\sigma}}_1 + b\,\dot{\boldsymbol{\sigma}}_1 + c\,\boldsymbol{\sigma}_1 = d. \tag{37}$$

Accordingly, the only condition yielding $\boldsymbol{\delta}_1$ converging to zero is that both a and b needs to be positive. This means that the subsequent conditions need to be satisfied:

$$\alpha_1\frac{1}{u}|\boldsymbol{B}_{12}\,\boldsymbol{A}^+| \text{ and } [k_1(k_1+k_2)+\alpha_1]\frac{1}{u}|\boldsymbol{B}_{12}\,\boldsymbol{A}^+| < 1. \tag{38}$$

Notice that at the hovering (stabilisation) we would have $u \approx |\boldsymbol{B}_{11}|\,g$, where g (≈ 9.8) is the gravity term. As for matrix $\boldsymbol{A}$, its Frobenius norm is equal to 2. And since $\boldsymbol{B}_{11}$ is k times larger than $\boldsymbol{B}_{12}$, condition (38) summarizes to the following gain limits $0 < k_1(k_1+k_2)+\alpha_1 < 2\,k\,g$. In the reported simulations we found k of roughly 1e2 order minimum. Accordingly, $k_1(k_1+k_2)+\alpha_1$ needs to have limits of roughly 1e3 order.

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] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] Y. Tan, J. Chang, H. Tan, and J. Hu, "Integral backstepping control and experimental implementation for motion system," in *IEEE Int. Conf. on Control Applications*, pp. 367 – 372, 2000.

[11] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a UAV with a robotic arm," in *10th IFAC Symposium on Robot Control*, pp. 3768–3773, Sept. 2012.

[12] T. Hamel and R. Mahony, "Visual servoing of an under-actuated rigid body system: An image based approach," *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 187–198, 2002.

[13] R. Mebarki and B. Siciliano, "Velocity-free image-based control of unmanned aerial vehicles," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1522–1527, 2013.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 1st ed., 2008.

[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, no. 3, pp. 646–656, 2014.

[17] V. Lippiello and L. Villani, "Managing redundant visual measurements for accurate pose tracking," *Robotica*, vol. 21, no. 5, pp. 511–519, 2003.

[18] V. Lippiello, B. Siciliano, and L. Villani, "Eye-in-hand/eye-to-hand multi-camera visual servoing," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, pp. 5354–5359, 2005.