Uncalibrated Visual Servo for Unmanned Aerial Manipulation

Angel Santamaria-Navarro, Patrick Grosch, Vincenzo Lippiello, Joan Solà and Juan Andrade-Cetto

Abstract—**This paper addresses the problem of autonomous servoing an unmanned redundant aerial manipulator using computer vision. The over-actuation of the system is exploited by means of a hierarchical control law which allows to prioritize several tasks during flight. We propose a safety related primary task to avoid possible collisions. As a secondary task we present an uncalibrated image-based visual servo strategy to drive the arm end-effector to a desired position and orientation using a camera attached to it. In contrast to previous visual-servo approaches, a known value of camera focal length is not strictly required. To further improve flight behavior we hierarchically add one task to reduce dynamic effects by vertically aligning the arm center of gravity to the multirotor gravitational vector, and another one that keeps the arm close to a desired configuration of high manipulability and avoiding arm joint limits. The performance of the hierarchical control law, with and without activation of each of the tasks, is shown in simulations and in real experiments confirming the viability of such prioritized control scheme for aerial manipulation.**

I. Introduction

Unmanned aerial vehicles (UAVs), and in particular multirotor systems, have substantially gained popularity in recent years, motivated by their significant increase in maneuverability, together with a decrease in weight and cost [1]. Until recently, UAVs were not usually required to interact physically with the environment, however this trend is set to change. Some examples are the ARCAS, AEROARMS and AEROWORKS EU funded projects with the aim to develop UAV systems with advanced manipulation capabilities for autonomous industrial inspection and repair tasks, such as the UAM manipulator Kinton from the ARCAS project shown in Fig. 1. Physical interaction with the environment calls for positioning accuracy at the centimeter level, which in GPS denied environments is often difficult to achieve. For indoor UAV systems, accurate localization is usually obtained from infrared multi-camera devices, like Vicon or Optitrack. However, these devices are not suited for outdoor environments and other means should be used, such as visual servoing.

Vision-based robot control systems are usually classified in three groups: position-based visual servo (PBVS), image-based visual servo (IBVS), and hybrid control systems [2], [3]. In PBVS, the geometric model of the target is used in conjunction with image features to estimate the pose of the target with respect to the camera frame. The control law is designed to reduce such pose error in pose space and, in consequence, the target could be easily lost in the image during the servo loop. In IBVS on the other hand, both the error and control law are expressed in the image space, minimizing the error between observed and desired image feature coordinates. As a consequence, IBVS schemes do not need any a priori knowledge of the 3D structure of the observed scene. In addition, IBVS is more robust than PBVS with respect to uncertainties and disturbances affecting the model of the robot, as well as the calibration of the camera [4], [5]. Hybrid methods, also called 2-1/2-D visual servo [6], combine IBVS and PBVS to estimate partial camera displacements at each iteration of the control law minimizing a functional of both.

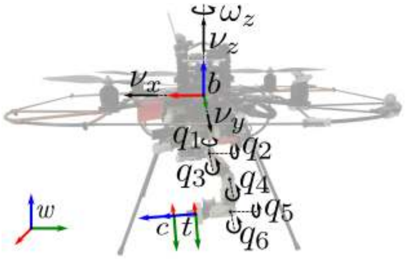


Fig. 1: The UAM used in the experiments is composed of a 4 DoF quadrotor, commanded at high-level by 3 linear and an angular velocities (ν_x, ν_y, ν_z and ω_z), and a 6 DoF robotic arm with joints q_j, $j = 1...6$; and world, camera, tool and body reference frames indicated by the letters w, c, t and b, respectively.

In all image-based and hybrid approaches the resulting image Jacobian or interaction matrix, which relates the camera velocity to the image feature velocities, depends on a priori knowledge of the intrinsic camera parameters. Although image-based methods, and in extension some hybrid approaches, have shown some robustness in these parameters, they usually break down at error levels larger than 10% [5]. In contrast, our method indirectly estimates the focal length online which, as shown in the experiments section, allows to withstand calibration errors up to 20%.

To do away with this dependence, one could optimize for the parameters in the image Jacobian whilst the error in the image plane is being minimized. This is done for instance, using Gauss-Newton to minimize the squared image error and non-linear least squares optimization for the image Jacobian [7]; using weighted recursive least squares, not to obtain the true parameters, but instead an approximation that

A. Santamaria-Navarro, P. Grosch, J. Solà and J. Andrade-Cetto are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, Barcelona 08028, Spain, e-mail: {asantamaria, pgrosch, jsola, cetto}@iri.upc.edu

V. Lippiello is with Università degli Studi di Napoli Federico II. Via Claudio 21, 80125 Napoli, Italy, e-mail: lippiello@unina.it

This work has been funded by the EU project AEROARMS H2020-ICT-2014-1-644271 and by the Spanish Ministry of Economy and Competitiveness project ROBINSTRUCT TIN2014-58178-R.

The paper has supplementary multimedia material available at http://www.angelsantamaria.eu/multimedia still guarantees asymptotic stability of the control law in the sense of Lyapunov [8], [9]; using k-nearest neighbor regression to store previously estimated local models or previous movements, and estimating the Jacobian using local least squares [10], or building a secant model using population of the previous iterates [11]. To provide robustness to outliers in the computation of the Jacobian, [12] proposes the use of an M-estimator.

In this paper we extend our prior work on uncalibrated image-based visual servo (UIBVS) [13], which was demonstrated only in simulation, to a real implementation for the case of aerial manipulation. UIBVS contains mild assumptions about the principal point and skew values of the camera, and does not require prior knowledge of the focal length. Instead, in our method, the camera focal length is iteratively estimated within the control loop. Independence of focal length true value makes the system robust to noise and to unexpected large variations of this parameter (*e.g.*, poor initialization or an unaccounted zoom change).

Multirotors, and in particular quadrotors such as the one used in this work, are underactuated platforms. That is, they can change their torque load and thrust/lift by altering the velocity of the propellers, with only four degrees-of-freedom (DoF), one for the thrust and three torques. But, as shown in this paper, the attachment of a manipulator arm to the base of the robot can be seen as a strategy to alleviate underactuation allowing unmanned aerial manipulators (UAM) to perform complex tasks.

In [14] a vision-based method to guide a UAM with a three DoF arm is described. To cope with underactuation of the aerial platform, roll and pitch motion compensation is moved to the image processing part, requiring projective transformations. Therefore, errors computing arm kinematics are to be coupled with the image-based control law and the scale (*i.e.* camera-object distance) cannot be directly measured.

Flying with a suspended load is a challenging task and it is essential to have the ability to minimize the undesired effects of the arm in the flying system [15]. Among these effects, there is the change of the center of mass during flight, that can be solved designing a low-level attitude controller such as a Cartesian impedance controller [16], or an adaptive controller. Moreover, a desired end-effector pose might require a non-horizontal robot configuration that the low level controller would try to compensate, changing in turn the arm end-effector position. In this way, [17] designs a controller exploiting the whole system model. However, flight stability is preserved by restricting the arm movements to those not jeopardizing UAM integrity. To cope with these problems, parallel robots are analyzed in [18] and [19]. The main advantages they offer are related with the torque reduction in the platform base. However, they are limited in workspace and are difficult to handle due to their highly nonlinear motion models.

The redundancy of quadrotor-arm systems in the form of extra DoF could be exploited to develop a low priority stabilizing task or to optimize some given quality indices, *e.g.* manipulability, joint limits, etc., [20], [21]. In [22] is presented an image-based control law explicitly taking into account the system redundancy and underactuation of the vehicle base. The camera is attached on the aerial platform and the positions of both arm end-effector and target are projected onto the image plane in order to perform an image-based error decrease, which creates a dependency on the precision of the odometry estimator that is rarely achieved in a real scenario without motion capture systems. Moreover, the proposed control scheme is only validated in simulation.

In this work, we exploit the DoF redundancy of the overall system not only to achieve the desired visual servo task, but to do so whilst attaining also other tasks during the mission. We presented in [23] a close approach consisting on a hybrid servoing scheme. In contrast to [23] which uses a combination of classical PBVS and IBVS, in this article we present a fully vision-based self-calibrated scheme that can handle poorly calibrated cameras. Moreover, we attach a light-weight serial arm to a quadrotor with a camera at its end-effector, see Fig. 1, instead of allocating it in the platform frame.

We present a new safety task intended for collision avoidance, designed with the highest priority. Our servo task is considered second in the hierarchy with two low priority tasks, one to vertically align the arm and platform centers of gravity and another to avoid arm joint limits. In contrast to [23] we combine the tasks hierarchically in a less restrictive manner, minimizing secondary task reconstruction only for those components not in conflict with the primary task. This strategy is known to achieve possibly less accurate secondary task reconstruction but with the advantage of decoupling algorithmic singularities between tasks [24].

Although hierarchical task composition techniques are well known for redundant manipulators, its use on aerial manipulation is novel. Specifically, the underactuation of the flying vehicle has critical effects on mission achievement and here we show how the non-controllable DoF must be considered in the task designs. While the control law presented in [23] requires orthogonal tasks to guarantee stability of the system, in our case only independence of non-controllable DoF is required. We validate the use of this task hierarchy in simulations and in extensive real experiments, using our UIBVS scheme to track the target, and also with the aid of an external positioning system.

To summarize, the main contributions of the paper are twofold. On the one hand, we demonstrate now in real experiments (on-board, and in real time) the proposed uncalibrated image-based servo law which was previously only shown in simulation in [13]. The second contribution is the proposal of a hierarchical control law that exploits the extra degrees of freedom of the UAV-arm system which, in contrast to our previous solution [23], uses a less restrictive control law that only actuates on the components of the secondary tasks that do not conflict directly with tasks higher up in the hierarchy.

The remainder of this article is structured as follows. The next section presents our uncalibrated approach to visual servo. Section III describes the kinematics of our UAM and Section IV contains the proposed task priority controller and task definitions. Simulations and experimental results are presented in Section V. Finally, conclusions are given in Section VII.

II. Uncalibrated image-based visual servoing

Drawing inspiration on the UPnP algorithm [25], we describe in the following subsection a method to solve for the camera pose and focal length using a reference system attached to the target object. The method is extended in Sec. II-B to compute a calibration-free image Jacobian for our servo task, and in Sec. II-C to compute the desired control law.

A. Uncalibrated PnP

3D target features are parameterized with their barycentric coordinates, and the basis of these coordinates is used to define a set of control points. Computing the pose of the object with respect to the camera resorts to computing the location of these control points with respect to the camera frame. A least squares solution for the control point coordinates albeit scale, is given by the null eigenvector of a linear system made up of all 2D to 3D perspective projection relations between the target points. Given the fact that distances between control points must be preserved, these distance constraints can be used in a second least squares computation to solve for scale and focal length. More explicitly, the perspective projection equations for each target feature become

$$\sum_{j=1}^{4}\left(a_{ij}x_j + a_{ij}(u_0 - u_i)\frac{z_j}{\alpha}\right) = 0 \tag{1a}$$

$$\sum_{j=1}^{4}\left(a_{ij}y_j + a_{ij}(v_0 - v_i)\frac{z_j}{\alpha}\right) = 0, \tag{1b}$$

where $\mathbf{s}_i = [u_i, v_i]^\top$ are the image coordinates of the target feature i, and $\mathbf{c}_j = [x_j, y_j, z_j]^\top$ are the 3D coordinates of the j-th control point in the camera frame. The terms a_{ij} are the barycentric coordinates of the i-th target feature which are constant regardless of the location of the camera reference frame, and α is our unknown focal length.

These equations can be jointly expressed for n 2D-3D correspondences as a linear system

$$\mathbf{M}\mathbf{x} = \mathbf{0}\,, \tag{2}$$

where $\mathbf{M}$ is a $2n \times 12$ matrix made of the coefficients a_{ij}, the 2D points $\mathbf{s}_i$ and the principal point, and $\mathbf{x}$ is our vector of 12 unknowns containing both the 3D coordinates of the control points in the camera reference frame and the camera focal length, dividing the z terms $\mathbf{x} = [x_1, y_1, z_1/\alpha, ..., x_4, y_4, z_4/\alpha]^\top$. Its solution lies in the null space of $\mathbf{M}$, and can be computed as a scaled product of the null eigenvector of $\mathbf{M}^\top\mathbf{M}$ via singular value decomposition

$$\mathbf{x} = \beta\mathbf{v}\,, \tag{3}$$

the scale β becoming a new unknown. In the noise-free case, $\mathbf{M}^\top\mathbf{M}$ is only rank deficient by one, but when image noise is severe it might loose rank, and a more accurate solution can be found as a linear combination of the basis of its null space. In this work we are not interested on recovering accurate camera pose, but on minimizing the projection error within a servo task. It is sufficient for our purposes to consider only the least squares approximation; that is, to compute the solution only using the eigenvector associated to the smallest eigenvalue.

To solve for β we add constraints that preserve the distance between control points of the form $||c_j - c_{j'}||^2 = d^2_{jj'}$, where $d_{jj'}$ is the known distance between control points c_j and $c_{j'}$ in the world coordinate system. Substituting $\mathbf{x}$ in these six distance constraints, we obtain a system of the form $\mathbf{L}\mathbf{b} = \mathbf{d}$, where $\mathbf{b} = [\beta^2, \alpha^2\beta^2]^\top$, $\mathbf{L}$ is a 6×2 matrix built from the known elements of $\mathbf{v}$, and $\mathbf{d}$ is the 6-vector of squared distances between the control points. We solve this overdetermined linearized system using least squares and estimate the magnitudes of α and β by back substitution

$$\alpha = \sqrt{\frac{|b_2|}{|b_1|}}\,, \quad \beta = \sqrt{b_1}\,. \tag{4}$$

B. Calibration-free Image Jacobian

As the camera moves, the velocity of each target control point $\mathbf{c}_j$ in camera coordinates can be related to the camera spatial velocity $(\mathbf{t}, \boldsymbol{\Omega})$ with $\dot{\mathbf{c}}_j = -\mathbf{t} - \boldsymbol{\Omega} \times \mathbf{c}_j$. Which combined with Eq. 3, we obtain

$$\begin{bmatrix}\dot{x}_j \\ \dot{y}_j \\ \dot{z}_j\end{bmatrix} = \begin{bmatrix}-t_x - \omega_y\,\alpha\,\beta v_z + \omega_z\,\beta v_y \\ -t_y - \omega_z\,\beta v_x + \omega_x\,\alpha\,\beta v_z \\ -t_z - \omega_x\,\beta v_y + \omega_y\,\beta v_x\end{bmatrix}\,, \tag{5}$$

where v_x, v_y, and v_z are the x, y, and z components of eigenvector $\mathbf{v}$ related to the control point $\mathbf{c}_j$, and whose image projection and its time derivative are given by

$$\begin{bmatrix}u_j \\ v_j\end{bmatrix} = \begin{bmatrix}\alpha\frac{x_j}{z_j} + u_0 \\ \alpha\frac{y_j}{z_j} + v_0\end{bmatrix}\,, \quad \begin{bmatrix}\dot{u}_j \\ \dot{v}_j\end{bmatrix} = \alpha\begin{bmatrix}\frac{\dot{x}_j}{z_j} - \frac{x_j\dot{z}_j}{z_j^2} \\ \frac{\dot{y}_j}{z_j} - \frac{y_j\dot{z}_j}{z_j^2}\end{bmatrix}\,. \tag{6}$$

Substituting Eqs. 3 and 5 in Eq. 6 we have

$$\dot{u}_j = \frac{-t_x - \alpha\beta v_z\omega_y + \beta v_y\omega_z}{\beta v_z} - \frac{v_x(-t_z - \beta v_y\omega_x + \beta v_x\omega_y)}{\alpha\beta v_z^2} \tag{7a}$$

$$\dot{v}_j = \frac{-t_y - \alpha\beta v_z\omega_x + \beta v_x\omega_z}{\beta v_z} - \frac{v_y(-t_z - \beta v_y\omega_x + \beta v_x\omega_y)}{\alpha\beta v_z^2}\,, \tag{7b}$$

which can be rewritten as $\dot{\mathbf{s}}_j = \mathbf{J}_j\,\mathbf{v}_c$, with $\dot{\mathbf{s}}_j = [\dot{u}_j, \dot{v}_j]^\top$, the image velocities of control point j, and $\mathbf{v}_c = [\mathbf{t}^\top, \boldsymbol{\Omega}^\top]^\top$. $\mathbf{J}_j$ is our desired calibration-free image Jacobian for the j-th control point, and takes the form

$$\mathbf{J}_j = \begin{bmatrix}\frac{-1}{\beta v_z} & 0 & \frac{v_x}{\alpha\beta v_z^2} & \frac{v_x v_y}{\alpha v_z^2} & \frac{-v_x^2-\alpha^2 v_z^2}{\alpha v_z^2} & \frac{v_y}{v_z} \\ 0 & \frac{-1}{\beta v_z} & \frac{v_y}{\alpha\beta v_z^2} & \frac{v_y^2+\alpha^2 v_z^2}{\alpha v_z^2} & \frac{-v_x v_y}{\alpha v_z^2} & \frac{-v_x}{v_z}\end{bmatrix}\,. \tag{8}$$

Stacking these together, we get the image Jacobian for all control points $\mathbf{J}_{vs} = \begin{bmatrix}\mathbf{J}_1 & \ldots & \mathbf{J}_4\end{bmatrix}^\top$.

C. Control Law

The aim of our image-based control scheme is to minimize the error $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*$, where $\mathbf{s}(t)$ are the current image coordinates of the set of target features, and $\mathbf{s}^*$ are their final desired positions in the image plane, computed with our initial value for α. If we select $\mathbf{s}$ to be the projection of the control points $\mathbf{c}$, and disregarding the time variation of α, and consequently of $\mathbf{s}^*$, the derivative of the error becomes $\dot{\mathbf{e}} = \dot{\mathbf{s}}$, and, for a desired exponential decoupled error decrease

$\dot{\mathbf{e}} = -\mathbf{\Lambda}_S \mathbf{e}$, we have a desired camera velocity

$$\mathbf{v}_c = -\mathbf{\Lambda}_S \, \mathbf{J}_{vs}^{+} \, \mathbf{e} \tag{9}$$

where $\mathbf{\Lambda}_S$ is a 6×6 positive definite gain matrix and $\mathbf{J}_{vs}^{+} = (\mathbf{J}_{vs}^{\top} \, \mathbf{J}_{vs})^{-1} \, \mathbf{J}_{vs}^{\top}$ is the left Moore-Penrose pseudoinverse of $\mathbf{J}_{vs}$.

III. Robot model

A. Coordinate Systems

Consider the quadrotor-arm system equipped with a camera mounted at the end-effector's arm as shown in Fig. 1. Without loss of generality, we consider the world frame (w) to be located at the target. With this, the position of the camera (c) with respect to the target frame, expressed as a homogeneous transform $\mathbf{T}_c^w$, can be computed integrating the camera velocities obtained from the uncalibrated visual servo approach presented in the previous section.

A quadrotor is at the high level of control an underactuated vehicle with only 4 DoF, namely the linear velocities plus the yaw angular velocity $(\nu_{qx}, \nu_{qy}, \nu_{qz}, \omega_{qz})$ acting on the body frame. And at the low level, the attitude controller stabilizes horizontally the quadrotor body. Now, let $\mathbf{q}_a = \left[q_1, \ldots, q_m\right]^{\top}$ be the joint vector of the robotic arm attached to the UAM. With the arm base frame coincident with the quadrotor body frame, the relation between the quadrotor body and camera frames is $\mathbf{T}_c^b = \mathbf{T}_t^b(\mathbf{q}_a) \, \mathbf{T}_c^t$, with $\mathbf{T}_t^b(\mathbf{q}_a)$ the arm kinematics and $\mathbf{T}_c^t$ the tool-camera transform. Moreover, the pose of the quadrotor with respect to the target is determined by the transform $\mathbf{T}_w^b = \mathbf{T}_c^b \, (\mathbf{T}_c^w)^{-1}$.

B. Robot Kinematics

We are in the position now to define a joint quadrotor-arm Jacobian that relates the local translational and angular velocities of the platform and those of the m arm joints, $\mathbf{v}_{qa} = (\nu_{qx}, \nu_{qy}, \nu_{qz}, \omega_{qx}, \omega_{qy}, \omega_{qz}, \dot{q}_1, \ldots, \dot{q}_m)$, to the desired camera velocities computed from the visual servo

$$\mathbf{v}_c = \mathbf{J}_{qa} \, \mathbf{v}_{qa}. \tag{10}$$

with $\mathbf{J}_{qa}$ the Jacobian matrix of the whole robot.

This velocity vector in the camera frame, can be expressed as a sum of the velocities added by the arm kinematics and the quadrotor movement $\mathbf{v}_c = \mathbf{v}_a^c + \mathbf{v}_q^c$ (superscripts indicate the reference frame to make it clear to the reader), where $\mathbf{v}_a^c$ is obtained with the arm Jacobian

$$\mathbf{v}_a^c = \begin{bmatrix} \mathbf{R}_b^c & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b^c \end{bmatrix} \mathbf{J}_a \, \dot{\mathbf{q}}_a = \overline{\mathbf{R}}_b^c \, \mathbf{J}_a \, \dot{\mathbf{q}}_a, \tag{11}$$

with $\mathbf{R}_b^c$ the rotation matrix of the body frame with respect to the camera frame, and where $\mathbf{v}_q^c$ corresponds to the velocity of the quadrotor expressed in the camera frame

$$\mathbf{v}_q^c = \overline{\mathbf{R}}_b^c \begin{bmatrix} \boldsymbol{\nu}_q^b + \boldsymbol{\omega}_q^b \times \mathbf{r}_c^b \\ \boldsymbol{\omega}_q^b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^c & -\mathbf{R}_b^c \left[\mathbf{r}_c^b\right]_{\times} \\ \mathbf{0} & \mathbf{R}_b^c \end{bmatrix} \mathbf{v}_q^b, \tag{12}$$

with $\mathbf{r}_c^b(\mathbf{q}_a)$ the distance vector between the body and camera frames, and $\mathbf{v}_q^b = [\nu_{qx}, \nu_{qy}, \nu_{qz}, \omega_{qx}, \omega_{qy}, \omega_{qz}]^{\top}$ the velocity vector of the quadrotor in the body frame.

Combining Eqs. 9 and 10 we can relate the desired high-level control velocities with our visual servo task, which we term now $\boldsymbol{\sigma}_S$

$$\mathbf{J}_{qa} \mathbf{v}_{qa} = -\mathbf{\Lambda}_S \underbrace{\mathbf{J}_{vs}^{+} \mathbf{e}}_{\boldsymbol{\sigma}_S} . \tag{13}$$

Unfortunately as said before, the quadrotor is an underactuated vehicle. So, to remove the non-controllable variables from the control command, their contribution to the image error can be isolated from that of the controllable ones by extracting the columns of $\mathbf{J}_{qa}$ and the rows of $\mathbf{v}_{qa}$ corresponding to ω_{qx} and ω_{qy}, reading out these values from the platform gyroscopes, and subtracting them from the camera velocity [26]

$$\mathbf{J}_S \dot{\mathbf{q}} + \overline{\mathbf{J}}_S \boldsymbol{\varpi} = -\mathbf{\Lambda}_S \boldsymbol{\sigma}_S \, , \tag{14}$$

where $\boldsymbol{\varpi} = [\omega_{qx}, \omega_{qy}]^{\top}$, $\overline{\mathbf{J}}_S$ is the Jacobian formed by the columns of $\mathbf{J}_{qa}$ corresponding to ω_{qx} and ω_{qy}, and $\mathbf{J}_S$ is the Jacobian formed by all other columns of $\mathbf{J}_{qa}$, corresponding to the actuated variables $\dot{\mathbf{q}} = [\nu_{qx}, \nu_{qy}, \nu_{qz}, \omega_{qz}, \dot{q}_1, \ldots, \dot{q}_m]^{\top}$.

Rearranging terms

$$\mathbf{J}_S \dot{\mathbf{q}} = \underbrace{-\mathbf{\Lambda}_S \boldsymbol{\sigma}_S - \overline{\mathbf{J}}_S \, \boldsymbol{\varpi}}_{\boldsymbol{\xi}} \tag{15}$$

and with this, our main task velocity corresponding to the visual servo is

$$\dot{\mathbf{q}} = \mathbf{J}_S^{+} \, \boldsymbol{\xi} \, , \tag{16}$$

where, with 6 linearly independent rows and $4 + m > 6$ columns, $\mathbf{J}_S^{+}$ is computed with the right Moore-Penrose pseudoinverse $\mathbf{J}_S^{\top} \, (\mathbf{J}_S \, \mathbf{J}_S^{\top})^{-1}$.

C. Motion Distribution

In order to penalize the motion of the quadrotor vs the arm to account for their different motion capabilities, we can define a weighted norm of the whole velocity vector $\|\dot{\mathbf{q}}\|_W = \sqrt{\dot{\mathbf{q}}^{\top} \mathbf{W} \dot{\mathbf{q}}}$ as in [27], and use a weighted task Jacobian to solve for the weighted controls

$$\dot{\mathbf{q}}_W = \mathbf{W}^{-1/2} \, (\mathbf{J}_S \, \mathbf{W}^{-1/2})^{+} \, \boldsymbol{\xi} = \mathbf{J}_S^{\#} \, \boldsymbol{\xi} \, , \tag{17}$$

with

$$\mathbf{J}_S^{\#} = \mathbf{W}^{-1} \, \mathbf{J}_S^{\top} \, (\mathbf{J}_S \, \mathbf{W}^{-1} \, \mathbf{J}_S^{\top})^{-1} \tag{18}$$

the weighted generalized Moore-Penrose pseudoinverse of the servoing Jacobian. With this, large movements should be achieved by the quadrotor whereas the precise movements should be devoted to the robotic arm due to its dexterity when the platform is close to the target. To achieve this behavior, we define a time-varying diagonal weight-matrix, as proposed in [28], $\mathbf{W}(d) = \mathrm{diag}((1-\gamma) \, \mathbf{I}_4, \gamma \, \mathbf{I}_n)$, with $n = 4 + m$ the whole UAM DoF (4 for the quadrotor and m for the arm) and

$$\gamma(d) = \frac{1 + \underline{\gamma}}{2} + \frac{1 - \underline{\gamma}}{2} \tanh\left(2\,\pi \, \frac{d - \delta_W}{\Delta_W - \delta_W} - \pi\right), \tag{19}$$

where $\gamma \in [\underline{\gamma}, 1]$, and δ_W and Δ_W, $\Delta_W > \delta_W$, are the distance thresholds corresponding to $\gamma \cong 1$ and $\gamma \cong \underline{\gamma}$, respectively. The blocks of $\mathbf{W}$ weight differently the velocity components of the arm and the quadrotor by increasing the velocity of the quadrotor when the distance to the target $d > \Delta_W$, while for distances $d < \delta_W$ the quadrotor is slowed down and the arm is commanded to accommodate for the precise movements.

IV. Task Priority Control

A. Hierarchical Task Composition

Even though the quadrotor itself is underactuated (4 DoF), by attaching a robotic arm with more than 2 DoF we can attain over-actuation ($n = 4 + m$). In our case, $m = 6$. Exploiting this redundancy, we can achieve additional tasks acting on the null space of the quadrotor-arm Jacobian [29], while preserving the primary task. These tasks can be used to reconfigure the robot structure without changing the position and orientation of the arm end-effector. This is usually referred to as *internal motion* of the arm. One possible way to specify a secondary task is to choose its velocity vector as the gradient of a scalar objective function to optimize [20], [30]. Multiple secondary tasks can be arranged in hierarchy and, to avoid conservative stability conditions [31], the *augmented inverse-based projections* method is here considered [21]. In this method, lower priority tasks are not only projected onto the null space of the task up in the hierarchy, but onto the null space of an augmented Jacobian with all higher priority tasks.

In Section III-B we showed how to compute a visual servo control law that takes into account the uncontrollable state variables. This is not however our main task. We decide to locate higher up in the hierarchy an obstacle avoidance task needed to guarantee system integrity. In a more general sense, we can define any such primary task as a configuration dependent task $\boldsymbol{\sigma}_0 = \mathbf{f}_0(\mathbf{x})$. Differentiating it with respect to $\mathbf{x}$, and separating the uncontrollable state variables as in Eq. 14 we have

$$\dot{\boldsymbol{\sigma}}_0 = \frac{\partial \mathbf{f}_0(\mathbf{x})}{\partial \mathbf{x}}\,\dot{\mathbf{x}} = \mathbf{J}_0\,\dot{\mathbf{q}}_0 + \overline{\mathbf{J}}_0\boldsymbol{\varpi}\,, \tag{20}$$

which again, considering as in Eq. 16 a main task error $\widetilde{\boldsymbol{\sigma}}_0 = \boldsymbol{\sigma}_0^* - \boldsymbol{\sigma}_0$, to regulate $\boldsymbol{\sigma}_0$ to a desired value $\boldsymbol{\sigma}_0^*$, the control law for the main task becomes

$$\dot{\mathbf{q}}_0 = \mathbf{J}_0^+(\boldsymbol{\Lambda}_0\widetilde{\boldsymbol{\sigma}}_0 - \overline{\mathbf{J}}_0\boldsymbol{\varpi})\,, \tag{21}$$

where as with Eq. 15 and 16, $\boldsymbol{\Lambda}_0$ is a positive definite gain matrix and $\mathbf{J}_0^+$ is the generalized inverse of $\mathbf{J}_0$.

Consider now a secondary lower priority task $\boldsymbol{\sigma}_1 = \mathbf{f}_1(\mathbf{x})$ such that

$$\dot{\boldsymbol{\sigma}}_1 = \mathbf{J}_1\,\dot{\mathbf{q}}_1 + \overline{\mathbf{J}}_1\boldsymbol{\varpi}\,, \tag{22}$$

with $\dot{\mathbf{q}}_1 = \mathbf{J}_1^+(\boldsymbol{\Lambda}_1\widetilde{\boldsymbol{\sigma}}_1 - \overline{\mathbf{J}}_1\boldsymbol{\varpi})$ and a task composition strategy that minimizes secondary task velocity reconstruction only for those components in Eq. 22 that do not conflict with the primary task [24], namely

$$\dot{\mathbf{q}} = \mathbf{J}_0^+\,\boldsymbol{\Lambda}_0\,\widetilde{\boldsymbol{\sigma}}_0 + \mathbf{N}_0\mathbf{J}_1^+\,\boldsymbol{\Lambda}_1\,\widetilde{\boldsymbol{\sigma}}_1\, - \overline{\mathbf{J}}_{0|1}\,\boldsymbol{\varpi}\,, \tag{23}$$

where $\mathbf{N}_0 = (\mathbf{I}_n - \mathbf{J}_0^+\,\mathbf{J}_0)$ is the null space projector of the primary task and $\overline{\mathbf{J}}_{0|1} = \mathbf{J}_0^+\overline{\mathbf{J}}_0 + \mathbf{N}_0\mathbf{J}_1^+\overline{\mathbf{J}}_1$ is the Jacobian matrix that allows for the compensation of the variation of the uncontrollable states $\boldsymbol{\varpi}$.

This strategy, in contrast to the more restrictive one we presented in [23] might achieve larger constraint-task reconstruction errors than the full least squares secondary task solution in [23] but with the advantage that algorithmic singularities arising from conflicting tasks are decoupled from the singularities of the secondary tasks.

The addition of more tasks in cascade is possible as long as there exist remaining DoF from the concatenation of tasks higher up in the hierarchy. The generalization of Eq. 23 to the case of η prioritized subtasks is

$$\dot{\mathbf{q}} = \mathbf{J}_0^+\boldsymbol{\Lambda}_0\widetilde{\boldsymbol{\sigma}}_0 + \sum_{i=1}^{\eta}\mathbf{N}_{0|\ldots|i-1}\mathbf{J}_i^+\boldsymbol{\Lambda}_i\widetilde{\boldsymbol{\sigma}}_i - \overline{\mathbf{J}}_{0|\ldots|\eta}\,\boldsymbol{\varpi} \tag{24}$$

with the recursively-defined compensating matrix

$$\overline{\mathbf{J}}_{0|\ldots|\eta} = \mathbf{N}_{0|\ldots|i-1}\mathbf{J}_i^+\,\overline{\mathbf{J}}_i\, + \,(\mathbf{I} - \mathbf{N}_{0|\ldots|i-1}^+\mathbf{J}_i^+\overline{\mathbf{J}}_i)\overline{\mathbf{J}}_{0|\ldots|i-1}\,, \tag{25}$$

where $\mathbf{N}_{0|\ldots|i}$ is the projector onto the null space of the augmented Jacobian $\mathbf{J}_{0|\ldots|i}$ for the i-th subtask, with $i = 0, \ldots, \eta - 1$, and are respectively defined as follows

$$\mathbf{N}_{0|\ldots|i} = (\mathbf{I} - \mathbf{J}_{0|\ldots|i}^+\,\mathbf{J}_{0|\ldots|i}) \tag{26}$$

$$\mathbf{J}_{0|\ldots|i} = [\mathbf{J}_0^\top\,\ldots\,\mathbf{J}_i^\top]^\top\,. \tag{27}$$

B. Stability analysis

To assess the stability of each i-th individual task, we use Lyapunov analysis by considering the positive definite candidate Lyapunov function $\mathcal{L} = \frac{1}{2}\,\|\boldsymbol{\sigma}_i(t)\|^2$ and its derivative $\dot{\mathcal{L}} = \boldsymbol{\sigma}_i^T\,\dot{\boldsymbol{\sigma}}_i$. Then, for the primary task we can substitute Eq. 21 into Eq. 20, giving $\dot{\boldsymbol{\sigma}}_0 = \boldsymbol{\Lambda}_0\widetilde{\boldsymbol{\sigma}}_0$, which for a defined main task error $\widetilde{\boldsymbol{\sigma}}_0 = \boldsymbol{\sigma}_0^* - \boldsymbol{\sigma}_0$ and $\boldsymbol{\sigma}_0^* = \mathbf{0}$, the asymptotic stability is proven with $\dot{\mathcal{L}} = -\boldsymbol{\sigma}_0^T\Lambda_0\boldsymbol{\sigma}_0$.

Similarly, substituting Eq. 23 into Eq. 22, and considering a task error $\widetilde{\boldsymbol{\sigma}}_1 = \boldsymbol{\sigma}_1^* - \boldsymbol{\sigma}_1$, with $\boldsymbol{\sigma}_1^* = \mathbf{0}$, the following dynamics for the secondary task is achieved

$$\dot{\boldsymbol{\sigma}}_1 = -\mathbf{J}_1\mathbf{J}_0^+\boldsymbol{\Lambda}_0\boldsymbol{\sigma}_0 - \boldsymbol{\Lambda}_1\,\boldsymbol{\sigma}_1 + (\mathbf{J}_1\mathbf{J}_0^+\overline{\mathbf{J}}_0)\boldsymbol{\varpi}\,, \tag{28}$$

where we used the property $\mathbf{J}_1\mathbf{N}_0\mathbf{J}_1^+ = \mathbf{I}$. Notice how exponential stability of the secondary task in Eq. 28 can only be guaranteed when the tasks are independent for the uncontrollable states $\boldsymbol{\varpi}$ (*i.e.* $\mathbf{J}_1\mathbf{J}_0^+\overline{\mathbf{J}}_0 = \mathbf{0}$), hence $\dot{\mathcal{L}} = -\boldsymbol{\sigma}_1^T\mathbf{J}_1\mathbf{J}_0^+\boldsymbol{\Lambda}_0\boldsymbol{\sigma}_0 - \boldsymbol{\sigma}_1^T\Lambda_1\boldsymbol{\sigma}_1$, which is a less stringent condition than whole task orthogonality $\mathbf{J}_1\mathbf{J}_0^+ = \mathbf{0}$ that was needed in [23].

Finally the dynamics of the system can be written as

$$\begin{bmatrix}\dot{\boldsymbol{\sigma}}_0\\ \dot{\boldsymbol{\sigma}}_1\end{bmatrix} = \begin{bmatrix}-\boldsymbol{\Lambda}_0 & \mathbf{O}\\ -\mathbf{J}_1\mathbf{J}_0^+\boldsymbol{\Lambda}_0 & -\boldsymbol{\Lambda}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{\sigma}_0\\ \boldsymbol{\sigma}_1\end{bmatrix}\,, \tag{29}$$

which is characterized by a Hurwitz matrix as in [23] that guarantees the exponential stability of the system. Notice how the secondary task does not affect the dynamics of the main task thanks to the null space projector, hence the stability of the main task is again achieved.

The previous stability analysis can be straightforwardly extended to the general case of η subtasks.

C. Task Order

In this paper we consider the following ordered tasks: a primary safety task (I) considering potential collisions (inflation radius); a secondary task performing visual servoing (S), and lower in the hierarchy, the alignment of the center of gravity of the UAM (G), and a technique to stay away from the arm's joint limits (L). By denoting with $\mathbf{J}_I$, $\mathbf{J}_S$, $\mathbf{J}_G$ and $\mathbf{J}_L$ the

Jacobian matrices of the above-mentioned tasks, the desired system velocity can be written as follows

$$\dot{\mathbf{q}} = \mathbf{J}_I^{\#}\,\widetilde{\boldsymbol{\sigma}}_I + \mathbf{N}_I\,\mathbf{J}_S^{\#}\boldsymbol{\Lambda}_S\widetilde{\boldsymbol{\sigma}}_S + \mathbf{N}_{I|S}\,\mathbf{J}_G^{+}\,\widetilde{\sigma}_G \\ +\mathbf{N}_{I|S|G}\,\mathbf{J}_L^{+}\,\widetilde{\boldsymbol{\sigma}}_L \,-\overline{\mathbf{J}}_{I|S|G|L}\boldsymbol{\varpi}, \tag{30}$$

where $\mathbf{N}_I$, $\mathbf{N}_{I|S}$, $\mathbf{N}_{I|S|G}$ are the projectors of the safety, the visual servoing and of the center of gravity tasks, which are defined as

$$\mathbf{N}_I = (\mathbf{I} - \mathbf{J}_I^{\#}\,\mathbf{J}_I) \tag{31a}$$

$$\mathbf{N}_{I|S} = (\mathbf{I} - \mathbf{J}_{I|S}^{+}\,\mathbf{J}_{I|S}) \tag{31b}$$

$$\mathbf{N}_{I|S|G} = (\mathbf{I} - \mathbf{J}_{I|S|G}^{+}\,\mathbf{J}_{I|S|G})\,, \tag{31c}$$

with $\mathbf{J}_{I|S}$ and $\mathbf{J}_{I|S|G}$ the augmented Jacobians computed as in Eq. 27. Notice that the safety task Jacobian pseudo-inverse $\mathbf{J}_I^{\#}$ is also weighted and how in the null space projectors $\mathbf{N}_{I|S}$ and $\mathbf{N}_{I|S|G}$ from Eq. 31, the involved pseudo-inverses do not need to be weighted because the center of gravity alignment and the joint limits avoidance involve only arm movements and also should be accomplished during the flight. As described in the following subsections, to compute the recursively-defined compensation matrix $\overline{\mathbf{J}}_{I|S|G|L}$ from Eq. 30 in our particular case, neither the inflation radius nor the gravity vector alignment or joint limits avoidance tasks involve the uncontrolled variables (*i.e.* $\overline{\mathbf{J}}_I = \overline{\mathbf{J}}_G = \overline{\mathbf{J}}_L = \mathbf{0}$), thus we end up with $\overline{\mathbf{J}}_{I|S|G|L} = \mathbf{N}_I\overline{\mathbf{J}}_{S|S}$, which in fact, as the Jacobian of the safety measure is weighted with an activation matrix (*i.e.* when the safety task is not active $\mathbf{J}_I = \mathbf{0}$), it only involves the colmuns of the visual servo Jacobian corresponding to the uncontrolled variables ($\overline{\mathbf{J}}_S$). We now give more detailed descriptions of task Jacobians and task errors involved.

D. Collision Avoidance

The most important task during a mission is to preserve flight safety. When a rotor operates near an obstacle, different aerodynamic effects are revealed, such as the so called "ground" or "ceiling" effects, that can lead to an accident. Hence, to avoid them, we propose a task with the highest priority to maintain a safety distance to obstacles by defining a safety sphere around the flying platform, and comparing the Euclidean distance to the obstacle ($\mathbf{d}_o$) with the sphere radius (r_I). Then our task function to minimize becomes

$$\sigma_I = \lambda_I\,(r_I - ||\mathbf{d}_o||)\,, \tag{32}$$

with λ_I a suitable positive gain. Note that this corresponds to a proportional control law although integral or derivative errors could also be considered. The desired task variable is $\sigma_I^* = 0$ (i.e. $\widetilde{\sigma}_I = -\sigma_I$), while the corresponding task Jacobian is

$$\mathbf{J}_I = \begin{bmatrix} -2\,\mathbf{d}_o^{\top} & \mathbf{0}_{1\times(m+1)} \end{bmatrix}\,. \tag{33}$$

As the hierarchy is based on the dimension of the null space of the higher priority task Jacobians, we define a generalized pseudoinverse similarly to Eq. 18 with a diagonal activation matrix, $\mathbf{J}_I^{\#} = \mathbf{H}^{-1}\,\mathbf{J}_I^{\top}\,(\mathbf{J}_I\,\mathbf{H}^{-1}\,\mathbf{J}_I^{\top})^{-1}$. With this activation matrix, we can prevent potential collisions by cancelling the motion of the flying platform only in those directions susceptible to collide. This is usually called joint clamping (JC). The elements in $\mathbf{H}$ are $\mathbf{H} = \mathrm{diag}(h_x, h_y, h_z, \mathbf{0}_{1\times(m+1)})$, and will block only those quadrotor DoF where the inflation radius is not respected defining

$$h_i = \begin{cases} 1, & \text{if } d_{oi} < (r_I - ||\mathbf{d}_{oj}||)\ \forall j \neq i. \\ 0, & \text{otherwise.} \end{cases} \tag{34}$$

This control law clamps any motion that violates the minimum distance to the obstacle.

E. Center of Gravity

If the arm and quadrotor center of gravity (CoG) are not vertically aligned, the motion of the arm produces an undesired torque on the quadrotor base that perturbs the system attitude and position. This effect can be mitigated by minimizing the distance between the arm CoG and the vertical line of the quadrotor gravity vector.

The task function we introduce is the square distance of the robotic arm's CoG with respect to the z axis of the $i-th$ frame, which can be written as

$$\sigma_G = \lambda_G\,(\mathbf{p}_{Gxy}^{i})^{\top}\,\mathbf{p}_{Gxy}^{i}, \tag{35}$$

where λ_G is a suitable positive gain and

$$\mathbf{p}_{Gxy}^{i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}_b\,\mathbf{p}_G^{b}, \tag{36}$$

where $\mathbf{R}_b$ is the rotation matrix of the body frame b with respect to the world horizontal frame, and the desired task variable is $\sigma_G^* = \mathbf{0}$ (i.e. $\widetilde{\sigma}_G = -\sigma_G$). The position of the arm CoG $\mathbf{p}_G^b$ is a function of the arm joint configuration defined as

$$\mathbf{p}_G^b = \frac{\sum_{i=1}^{\nu} m_i\,\mathbf{p}_{Gi}^b}{\sum_{i=1}^{\nu}}, \tag{37}$$

where m_i and $\mathbf{p}_{Gi}^b$ are the mass and the position of the CoG of link i. We can compute the arm CoG with respect to the body frame for the sequence of links j to the end-effector with

$$\mathbf{p}_{Gj}^{*b} = \mathbf{R}_j^b\,\frac{\sum_{i=j}^{\nu} m_i\,\mathbf{p}_{Gi}^b}{\sum_{i=j}^{\nu}}, \tag{38}$$

where $\mathbf{R}_j^b$ is the rotation between link j and the body frame. Notice that all these quantities are a function of the current joint configuration $\mathbf{q}_a$. Then, differential relationship between the CoG and the arm joint configuration is $\dot{\mathbf{p}}_G^b = \mathbf{J}_G^b\,\dot{\mathbf{q}}_a$, where $\mathbf{J}_G^b \in \mathbb{R}^{3\times m}$ is the CoG Jacobian, expressed in the quadrotor body frame,

$$\mathbf{J}_G^b = \frac{\partial \mathbf{p}_G^b}{\partial \mathbf{q}_a} = \left(\mathbf{J}_{G1}^b ... \mathbf{J}_{G\nu}^b\right), \tag{39}$$

with $\mathbf{J}_{Gi}^b$ the individual joint i Jacobian

$$\mathbf{J}_{Gj}^b = \frac{\sum_{i=j}^{\nu} m_i}{\sum_{i=0}^{\nu}}\,\left(\mathbf{z}_j \times \mathbf{p}_{Gj}^{*b}\right). \tag{40}$$

Notice how the resultant linear velocity is scaled by the mass of the partial CoG in Eq. 40 because the CoG is the average of the multi-mass system with the consequence that high velocities on smaller masses play a smaller role on the total velocity of the CoG.

Finally, the corresponding task Jacobian from the derivative of Eq. 35 is defined as

$$\mathbf{J}_G = \begin{bmatrix}\mathbf{0}_{1\times 4} & 2(\mathbf{p}^i_{Gxy})^\top \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\mathbf{R}^i_b\,\mathbf{J}^b_G\end{bmatrix}. \qquad (41)$$

With this choice, the CoG of the arm is controlled to be aligned with the CoG of the vehicle along the direction of the gravitational force.

F. Joint Limits Avoidance

Joint limits avoidance can be achieved by driving the arm joints toward a desired value $\mathbf{q}^*_a$ that can be chosen far from an unrealizable configuration and/or close to one characterized by a high manipulability index or suitable with respect to the assigned task. The sum of normalized distances of the position of the i-th joint to its desired configuration is given by

$$\sum_{i=1}^{m}\left(\frac{q_{ai}-q^*_{ai}}{\overline{q}_{ai}-\underline{q}_{ai}}\right)^2. \qquad (42)$$

So our task function is selected as the squared distance of the whole arm joint configuration with respect to the desired one

$$\sigma_L = (\mathbf{q}_a - \mathbf{q}^*_a)^\top\,\mathbf{\Lambda}_L\,(\mathbf{q}_a - \mathbf{q}^*_a), \qquad (43)$$

where $\overline{\mathbf{q}}_a = \left[\overline{q}_{a1},\ldots,\overline{q}_{am}\right]^\top$ and $\underline{\mathbf{q}}_a = \left[\underline{q}_{a1},\ldots,\underline{q}_{am}\right]^\top$ are the high and low joint-limit vectors respectively, and $\mathbf{\Lambda}_L$ is a diagonal matrix whose diagonal elements are equal to the inverse of the squared joint limit ranges

$$\mathbf{\Lambda}_L = \mathrm{diag}((\overline{q}_{a1}-\underline{q}_{a1})^{-2},\ldots,(\overline{q}_{am}-\underline{q}_{am})^{-2}). \qquad (44)$$

The desired task variable is $\sigma^*_L = 0$ (i.e. $\widetilde{\sigma}_L = -\sigma_L$), while the corresponding task Jacobian is

$$\mathbf{J}_L = \left[\mathbf{0}_{1\times 4} \quad -2\left(\mathbf{\Lambda}_L\left(\mathbf{q}_a - \mathbf{q}^*_a\right)\right)^\top\right]. \qquad (45)$$

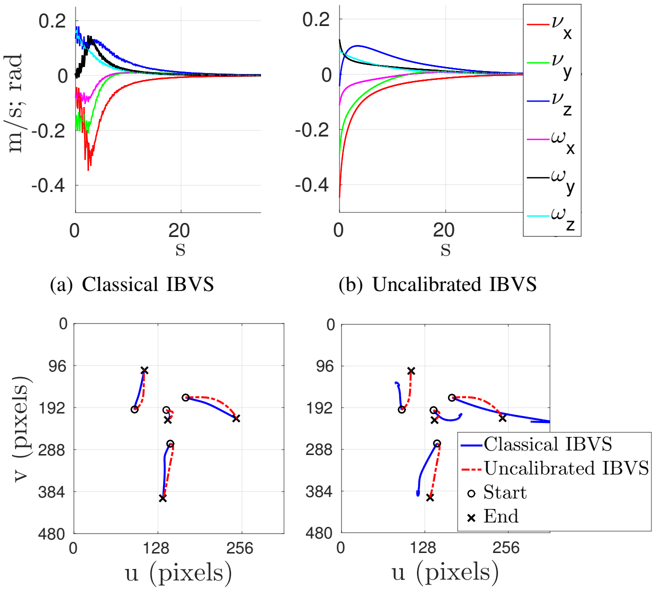
One common choice of $\mathbf{q}^*_a$ for the joint limit avoidance is the middle of the joint limit ranges (if this configuration is far from kinematic singularities), $\mathbf{q}^*_a = \underline{\mathbf{q}}_a + \frac{1}{2}(\overline{\mathbf{q}}_a - \underline{\mathbf{q}}_a)$.

V. Simulation and Validation

A. Visual Servo

To validate the proposed visual servo scheme we first present results of a comparison between the classical and the uncalibrated image-based visual-servo approaches using Matlab-Simulink. As inputs, we have a random set of target features and initial and desired final camera positions with respect to the target. For these comparisons we show simulation runs for 50 sec with time steps of 0.1 sec where the camera is assumed to be fully controllable (*i.e.* 6 DoF), and the controller used is a simple proportional controller with gain $\mathbf{\Lambda}_S = 0.125\,\mathbf{I}$. The value was chosen empirically to be able to compare the time both algorithms take to reduce both image and Cartesian errors.

Both methods turned out to be robust to noise levels of 1 to 3 pixels in the image coordinates. This noise typically accounts for the unmodelled inaccuracies of the image feature detector and equally affects both the IBVS and UIBVS methods. But the interesting results were obtained when noise was added to the focal length, which can be caused by mechanical vibrations of the optics, bad initial calibration values, or unaccounted zoom changes, to which only UIBVS is adequately responsive.



(a) Classical IBVS (b) Uncalibrated IBVS

(c) Noise-free conditions (d) Initialization error of 20% in camera focal length.

Fig. 2: Effects of noise in both IBVS and UIBVS during a simulated servo task. Figures a) and b) correspond to camera velocities subject to white noise with 1 mm standard deviation in the focal length. Figures c) and d) are control point trajectories in the image plane.

Fig. 2 shows the effects of noise in the camera focal length during a simulated servo task. Specifically, camera velocities for a servoing task with a focal length of 10 mm, and subject to white noise variations with 1 mm standard deviation are shown in Fig. 2(a) using a focal-length dependent Jacobian, and results with our proposed calibration-free scheme are shown in Fig. 2(b). Even when the servo task can be successfully completed in both cases in approximately the same amount of time, the proposed method provides a much smoother tracking of the camera to such variations in camera length. Fig. 2(c) and 2(d) shows the image plane trajectories of the control points for a similar experiment. Under equal noise-free simulation conditions (plot 2(c)) both methods have comparable asymptotic convergence. But, for an initialization error in the focal length of 20%, the classical approach is unable to reach the desired configuration, in contrast to the proposed approach (plot 2(d)). An unexpected variation of focal length is assumed by the classical image-based servo method as camera motion along the z axis and to recover from this, the control law induces undesirable changes in the robot velocity commands.

B. UAM system

To demonstrate the proposed hierarchical task composition we designed and built a lightweight robotic arm with a joint setting to compensate the possible noise existing in the quadrotor positioning while hovering, and to avoid self collisions during take off and landing maneuvers. However, its design was a trade-off between payload and accuracy, leading to a weight of 200g including batteries and an end-effector positioning precision of approximately 0.2rad and 10mm of angular and linear error norms, with the arm at its highest

Joint	θ (rad)	d (m)	α (rad)	a (m)
1	q_1	0	0	0
2	$q_2 - \pi/2$	0	-$\pi/2$	0
3	$q_3 - \pi/2$	0	-$\pi/2$	0
4	q_4	0	0	0.065
5	$q_5 + \pi/2$	0.065	0	0.065
6	q_6	0	$\pi/2$	0

TABLE I: Denavit-Hartenberg parameters of our 6 DoF arm.

motor torque capacity. The arm is shown in Fig 1 and its Denavit-Hartenberg parameters are given in Tbl. I. The arm base is $10mm$ below the body frame along the z axis and the camera is displaced $20mm$ from the end-effector along the z axis (defining $\mathbf{T}_c^t$).

To address the dynamical effects of the overall system our cascaded architecture considers two different control loops at very high frequency (1KHz), one for the arm and one for the attitude of the UAV; and a hierarchical task controller running at much lower frequency (camera frame rate), hence avoiding dynamic coupling between them. Moreover, we impose bounds on the maximum velocities and accelerations for the arm joints that in practice result also in small torques in the arm base, which the task controller is able to adequately compensate. We have added a comment regarding this issue in the experiments section.

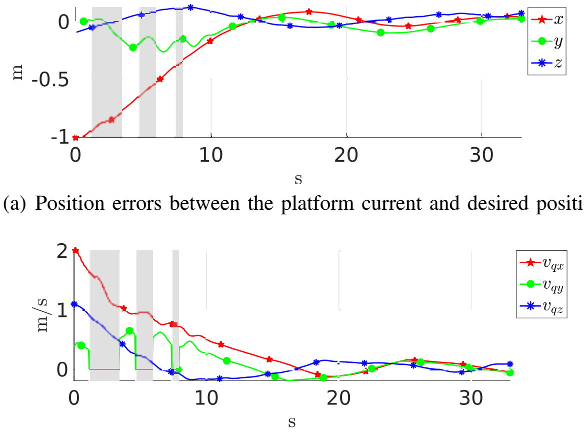
Similarly to [32], we present now simulations in ROS for our dynamical model (*i.e.,* a modification of the Hector quadrotor stack www.wiki.ros.org/hector_quadrotor, with the Asctec Pelican parameters) using the Gazebo simulator and the real parameters of our robotic arm. The mission consists on three phases. First the UAM is driven autonomously (taking-off and following waypoints) to a point where a main target is in the camera field of view. Secondly, the hierarchical control law is switched on to perform the servoing until a certain error in camera pose is reached by the end-effector. Finally when the servoing phase is accomplished, the UAM is autonomously commanded to land.

The visual servoing scheme presented consists on two least squares minimizations. First we solve for the control point coordinates in camera frame albeit scale in Eq. 2. Then we use the inter-distance constraints to solve for scale and focal length. As explained in Sec. II-A, we assume a set of randomly selected 3D feature points on the target and their 2D projections. Instead of developing a robust 3D feature detector and tracker, we use a planar detector of a target with a known geometry to retrieve the target frame, to which we add virtual features and then compute their location with respect to the target frame, as well as their basis, i.e., the control points. At each iteration, the marker is detected in the scene and the projection of the control points is computed. Those 2D-3D feature relationships represent the input to our visual servoing algorithm (Eq. 2). Notice how our scheme can be applied also to other sensory setups that can detect the object frame from other sources and not just visual features. In the accompanying video we show real experiments using the marker detector, as well as with an Optitrack system to detect the target.

1) Primary task: Among all other tasks, the one with the highest priority must be the safety task, not to compromise the platform integrity. Fig. 3 shows an example of how this task works. We start the servoing in a point free of collisions. The inflation radius is set to $0.5m$ and at the middle of the expected trajectory we add an obstacle $0.2m$ to the left of the quadrotor. Notice how in Fig. 3 the safety task becomes active (vertical grey areas in all plots) on the quadrotor y axis to move it away from collision. This DoF is used to keep the platform integrity and cannot be used in other tasks with lower priorities including the main visual servo task. When the obstacle does not violate the inflation radius, the safety task becomes deactivated and the other subtasks can regain access to the previously blocked DoF. Fig. 3(a) shows how the servoing task is elusive during the first 10 seconds of the simulation when the obstacle is present, but is accomplished afterwards when the obstacle is no longer an impediment to the secondary task. The activation and deactivation of this task can induce some chattering phenomena. Although this is not explicitly considered in the formulation, one can define a hysteresis scheme for the desired task variable σ_I^* or exchange the role of the safety measure to a lower priority when the inflation radius is not violated and move it back to the highest priority when required. In this last case also a smoothing procedure must be considered to adequately switch priorities.

	Tasks			Control law	
Time to target (s)	S	S+G	S+G+L	Weighted sum	Hierarchical
μ_t	42.143	29.973	29.036	85.3979	50.9624
$std(t)$	17.361	12.833	11.857	46.5853	21.4597

TABLE II: Time to completion statistics for multiple realizations of a simulated experiment considering different subtask arrangements and comparing two control law definitions.

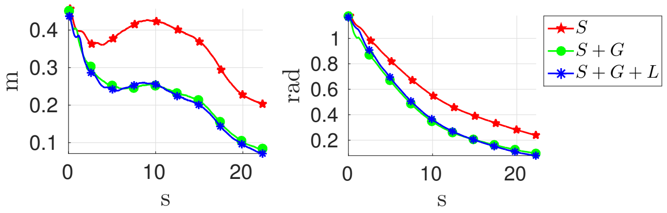


(a) Position errors between the platform current and desired positions.

(b) Visual servo task velocities

Fig. 3: Example of the safety task in action, with the inflation radius set to $0.5m$, when an obstacle appears $0.2m$ to the left of the expected trajectory (quadrotor y body axis). In grey are the zones where the safety task is activated, thus deactivating the involved DoF for all other tasks, including the visual servo mission task.

2) Task Composition: We now compare the effect of using the remaining subtasks in the hierarchy by launching several missions considering the following schemes: only visual servo (S), visual servo and arm CoG alignment (S + G); and the two plus also the joint-limit avoidance (S + G + L). The resulting statistics are presented in Tbl. II and in Fig. 4, which shows the root mean square error (RMSE) between the current and desired camera poses w.r.t. to mission time for the different task compositions.



(a) Linear RMSE.

(b) Angular RMSE.

Fig. 4: Root mean square error (RMSE) for multiple simulations considering different subtask arrangements.

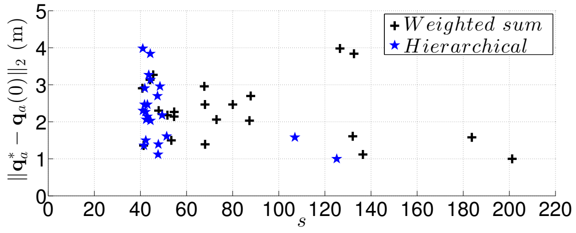


Fig. 5: Comparison between weighted sum and hierarchical task composition, considering the time to reach the target (horizontal axis), and for randomly varying initial conditions (vertical axis).

When only the visual servoing (S) is executed, the time to reach the target is significantly higher than those cases in which the arm CoG is vertically aligned (S+G and S+G+L). This is due to the undesired torque added to the quadrotor when the arm weight distribution is not aligned with the quadrotor CoG. By the addition of the CoG alignment task, this torque is reduced during the servoing task. However, if only the CoG is aligned, the arm can still reach undesired configurations, close to singularities or joint limits. The slight improvement in RMSE between S+G+L and S+G are because the arm is fully extended in the S+G case, increasing the vertical distance between the arm CoG and the platform base, leading to larger inertial effects than the S+G+L where a retracted configuration was set as the last task in the hierarchy.

The control law proposed in [28], in contrast to the hierarchical approach described previously, contains a unique secondary task corresponding to a weighted sum of subtasks, which can be problematic when the tasks are antagonistic. Depending on the weights assigned, the resulting velocities could not satisfy accurately some of the subtasks requirements. To show the advantage of the hierarchical task composition vs the weighted sum method we performed extensive simulations with the two strategies for varying initial conditions and final desired configurations. The results are shown in Fig. 5 and in Tbl. II. In all cases, simulations were ceased once a distance to the target smaller than 5cm was reached with an orientation closer than 0.026 radians. The main observed result is that both strategies were equally capable of reaching the target with the desired accuracy level, and that the hierarchical task composition method consistently did so in about 50 seconds, independent of the initial configuration; whereas the weighted sum method required on average 85 seconds to achieve the task.

VI. Real robot experiments

We conducted a series of experiments with missions similar to those shown in simulations, *i.e.* autonomously taking off and flying to a location in which the target appears in the field of view of the camera, turning on then the hierarchical task controller to servo the system towards a desired camera pose, and finally autonomously landing the system. The real experiments were conducted with our robot Kinton (Fig. 1), based on an Asctec Pelican quadrotor, and equipped with an onboard embedded PC (1.6GHz CPU) and a variety of sensors including an IMU and a barometer. All our algorithms are running onboard in real time with a camera frame rate at 20Hz.

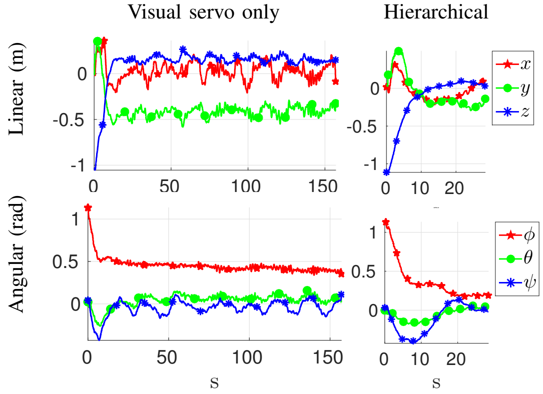


Fig. 6: Camera pose error during visual servoing. Comparison of using or not the hierarchical task priority control law with all the subtasks proposed.

Fig. 6 shows a comparison of task execution with and without activation of the task hierarchy. The left frames show linear and angular pose errors when only the servo task is active. In this experiment, task completion is considered to be reached at an Euclidean position error of 0.15m and 0.2rad; otherwise task abortion is executed if after 3 min of flight time the target is not reached. We are aware that better performance is possible with a more elaborate tuning of the different control gains, however, this level of precision is good enough to show that the task composition scheme allowed the system to quickly reach its target, whereas without the hierarchical task composition in place, the task could not be accomplished.

Arm CoG alignment is crucial to improve flight behavior. With it the approaching maneuver is softer allowing us not only to easily reach the desired servoing error but also reducing aggressive maneuvers to compensate the arm payload, thus reducing energy consumption which is a very important issue for this type of aerial platforms. Fig. 7 shows the effect of this alignment in terms of arm torque reduction.

The last task is designed to favor a desired arm configuration and it can be used to push the joints away from singularities and potentially increase maneuverability. Fig. 8 shows the error between the current and desired joint positions when the task is included in the hierarchy at the lowest priority level.

Finally, to evaluate the contribution of each control variable to the execution of the different tasks we present plots of the whole set of velocity profiles applied to the UAM actuators (*i.e.* 3 quadrotor linear velocities and 6 arm angular velocities plus quadrotor yaw velocity) in Fig. 9. Note how for the main mission task, the visual servo task, the quadrotor linear velocities play an important role during the first 5 seconds of the experiment, when the UAM is far from the target, with the arm joints accommodating later on for the fine positioning of the camera thanks to the time-varying weighted motion

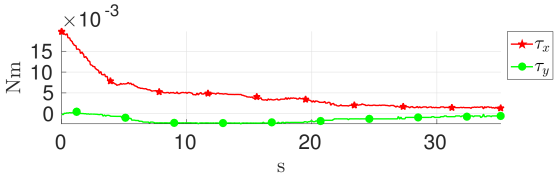


Fig. 7: Torque effects in the quadrotor body applying G task.

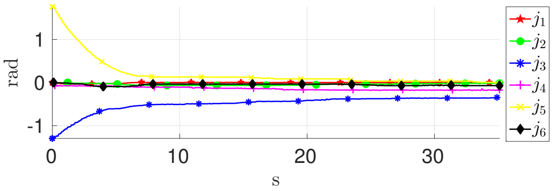


Fig. 8: Arm joints positioning error applying L task.

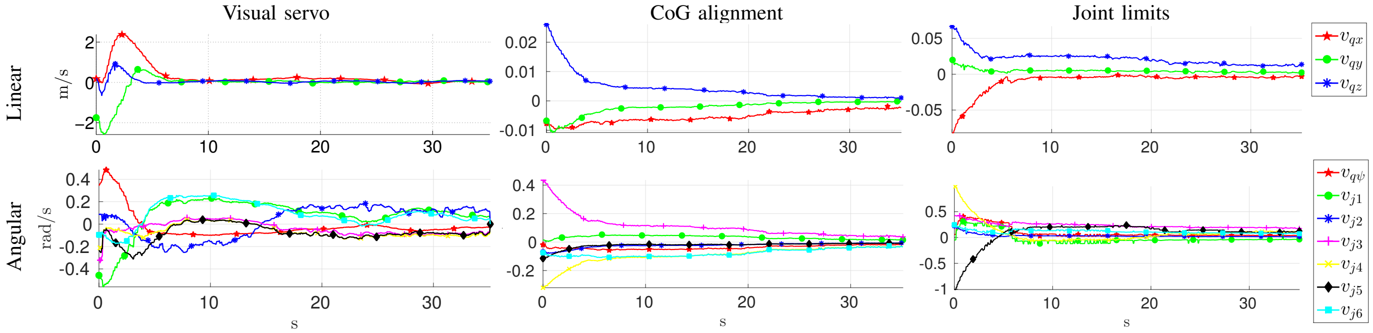


Fig. 9: Actuator velocities in a real experiment corresponding to the individual contributions of each subtask: visual servoing, CoG alignment and joint limits.

distribution presented in Sec.III-C.

The fact that all tasks report velocity values for the control variables indicate the availability of DoF for their execution from their higher priority tasks. Indeed, the dimension of the associated space to each null space projector are $dim(N_I) = 10$, $dim(N_{I|S}) = 4$ and $dim(N_{I|S|G}) = 3$. These dimensions indicate how tasks with lower priority than the inflation radius task can actuate on all 10 DoF of the robot when the inflation radius is not violated. The visual servo mission task requires 6 DoF, and the secondary and comfort tasks with lower priority can take advantage of the remaining 4 DoF. The gravitational vector alignment task and the joint limits avoidance task require 1 DoF each being scalar cost functions to minimize (see Eq. 35 and 43). These results have been experimentally confirmed, computing the number of singular values associated to each of the null space projectors in the task hierarchy. Although the dimension of these associated spaces give an idea of the available DoF for each task in the hierarchy, it does not imply that the subtask can always be fulfilled. An empirical study of this consideration led us to the task order priority presented in Sec. IV-C, and a thorough analytical study of these spaces by means of their basis (*i.e.* singular vectors associated to the non-zero singular values) and their implications to guarantee subtask completion is left as future work.

VII. Conclusions

We have presented an uncalibrated image-based visual servo scheme for manipulation UAVs. Target features are parametrized with their barycentric coordinates, and the basis of these coordinates is used to define a set of control points. A method is given to recover the coordinates of these control points and also of the camera focal length. With these, a new image Jacobian is derived which is guaranteed by construction to be of full rank. This guarantees asymptotic stability of the control law regardless of the target point selection, as long as planar configurations are avoided.

A serial arm is attached to the base of a quadrotor, and a camera is fixed at its end-effector. We have presented a control law to achieve not only the visual servoing but also other tasks taking into account their specific priorities. A primary task is designed to safeguard the platform integrity. The secondary task corresponds to the uncalibrated visual servo, and lower priority tasks are designed to alleviate quadrotor stabilization issues. This hierarchical strategy might not achieve the optimum constraint-task reconstruction errors, but instead the algorithmic singularities arising from conflicting tasks are decoupled from the singularities of the secondary tasks. Moreover, the presented control law only requires independent tasks for the uncontrollable variables to guarantee exponentially stability of the system.

The technique is demonstrated using Matlab and ROS in both simulation and a real UAM. All our code is available for download at www.ros.org/wiki/kinton_apps and a video of the method at work is also available in the same page, and included as suplemental material.

We can think of two avenues for further research. On the one hand, the activation and deactivation of the safety task as well as a dynamic exchange of task priority roles can induce some chattering phenomena, which can be avoided by introducing a hysteresis scheme. Secondly, the dimensionality of the subspace associated to each null space projector is a necessary condition to be considered when designing subtasks, however it might not be sufficient to guarantee the fulfilment of the subtask and a thorough analytical study of these spaces can be required.

Acknowledgments

The authors would like to thank S. Hernandez and F. Herrero for their support during the experiments.

References

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, “Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue,” *IEEE Robotics Autom. Mag.*, vol. 19, no. 3, pp. 46–56, 2012.

[2] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *IEEE Robotics Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[3] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, “Comparison of basic visual servoing methods,” *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 5, pp. 967–983, 2011.

[4] S. Hutchinson, G. D. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robotics Autom.*, vol. 12, no. 5, pp. 651–670, 1996.

[5] E. Malis and P. Rives, “Robustness of image-based visual servoing with respect to depth distribution errors,” in *Proc. IEEE Int. Conf. Robotics Autom.*, Taipei, Sep. 2003, pp. 1056–1061.

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

[7] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, “A dynamic jacobian estimation method for uncalibrated visual servoing,” in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, Atlanta, Sep. 1999, pp. 944–949.

[8] K. Hosoda and M. Asada, “Versatile visual servoing without knowledge of true Jacobian,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Munich, Sep. 1994, pp. 186–193.

[9] Q. Fu, Z. Zhang, and J. Shi, “Uncalibrated visual servoing using more precise model,” in *Proc. IEEE Conf. Robotics, Aut. Mechatron.*, Chengdu, Sep. 2008, pp. 916–921.

[10] A. Farahmand, A. Shademan, and M. Jagersand, “Global visual-motor estimation for uncalibrated visual servoing,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, Nov. 2007, pp. 1969–1974.

[11] M. Bonković, A. Hace, and K. Jezernik, “Population-based uncalibrated visual servoing,” *IEEE/ASME Trans. Mechatron.*, vol. 13, no. 3, pp. 393–397, 2008.

[12] A. Shademan, A. Farahmand, and M. Jagersand, “Robust Jacobian estimation for uncalibrated visual servoing,” in *Proc. IEEE Int. Conf. Robotics Autom.*, Anchorage, May 2010, pp. 5564–5569.

[13] A. Santamaria-Navarro and J. Andrade-Cetto, “Uncalibrated image-based visual servoing,” in *Proc. IEEE Int. Conf. Robotics Autom.*, Karlsruhe, May 2013, pp. 5247–5252.

[14] S. Kim, H. Seo, S. Choi, and H. J. Kim, “Vision-guided aerial manipulation using a multirotor with a robotic arm,” *IEEE/ASME Trans. Mechatron.*, 2016, to appear.

[15] I. Palunko, P. Cruz, and R. Fierro, “Agile load transportation: Safe and efficient load manipulation with aerial robots,” *IEEE Robotics Autom. Mag.*, vol. 19, no. 3, pp. 69–79, 2012.

[16] V. Lippiello and F. Ruggiero, “Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Oct. 2012, pp. 3768–3773.

[17] M. Orsag, C. Korpela, M. Pekala, and P. Oh, “Stability control in aerial manipulation,” in *Proc. American Control Conf.*, Washington, DC, Jun. 2013, pp. 5581–5586.

[18] T. Danko, K. Chaney, and P. Oh, “A parallel manipulator for mobile manipulating UAVs,” in *Proc. IEEE Int. Conf. Tech. Pract. Rob. App.*, Woburn, MA, Apr. 2015, pp. 1–6.

[19] J. Scholten, M. Fumagalli, S. Stramigioli, and R. Carloni, “Interaction control of an uav endowed with a manipulator,” in *Proc. IEEE Int. Conf. Robotics Autom.*, Karlsruhe, May 2013, pp. 4910–4915.

[20] S. Chiaverini, “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators,” *IEEE Trans. Robotics Autom.*, vol. 13, no. 3, pp. 398–410, 1997.

[21] P. Baerlocher and R. Boulic, “Task-priority formulations for the kinematic control of highly redundant articulated structures,” *J. Intell. Robotic Syst.*, vol. 1, pp. 323–329, 1998.

[22] R. Mebarki and V. Lippiello, “Image-based control for aerial manipulation,” *Asian J. Control*, vol. 16, no. 3, pp. 646–656, 2014.

[23] V. Lipiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. Trujillo, Y. R. Esteves, and A. Viguria, “Hybrid visual servoing with hierarchical task composition for aerial manipulation,” *IEEE Robotics Autom. Lett.*, vol. 1, no. 1, pp. 259–266, 2016, presented at ICRA’16.

[24] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin: Springer-Verlag, 2008.

[25] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer, “Exhaustive linearization for robust camera pose and focal length estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2386–2400, 2013.

[26] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics Autom. Mag.*, vol. 19, no. 3, pp. 20–32, 2012.

[27] T. F. Chan and R. Dubey, “A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators,” *IEEE Trans. Robotics Autom.*, vol. 11, no. 2, pp. 286–292, Apr 1995.

[28] V. Lippiello, R. Mebarki, and F. Ruggiero, “Visual coordinated landing of a UAV on a mobile robot manipulator,” in *Proc. 11th IEEE Int. Sym. Saf. Secur. Rescue Robotics*, Linkoping, Oct. 2013, pp. 1–7.

[29] Y. Nakamura, H.Hanafusa, and T. Yoshikawa, “Task-priority based redundancy control of robot manipulators,” *Int. J. Robotics Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[30] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, 1990.

[31] G. Antonelli, “Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems,” *IEEE Trans. Robotics*, vol. 25, no. 5, pp. 985–994, 2009.

[32] A. Santamaria-Navarro, V. Lipiello, and J. Andrade-Cetto, “Task priority control for aerial manipulation,” in *Proc. 12th IEEE Int. Sym. Saf. Secur. Rescue Robotics*, Toyako-cho, Oct. 2014, pp. 1–6.

A. Santamaria-Navarro was born in Moià, Barcelona on October 11, 1982. He received the B.Sc. degrees in Telecomunications and Industrial Management and the M.Sc. degree in Automatic Control and Robotics, from the Universitat Politècnica de Catalunya (UPC), Barcelona, in 2004 and 2012 respectively. He is a PhD candidate at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain. His current research interests include computer vision, mobile robotics and unmanned aerial vehicles (UAVs).

P. Grosch received the B.Sc. degree in Mechanical Engineering from Universidad de los Andes, Colombia, in 1993 and the PhD degree in Automatic Control, Robotics and Computer Vision from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2016. He is currently a postdoctoral researcher at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain. His current research interests include robotics related mechanical design.

V. Lipiello (M’12) was born in Naples, Italy, on June 19, 1975. He received the Laurea degree in Electronic Engineering and the PhD degree in Information Engineering from the University of Naples, Naples, in 2000 and 2004, respectively. He is an Assistant Professor of Automatic Control in the Department of Computer and Systems Engineering, University of Naples. Dr. Lippiello is the Chair of the International Federation of Automatic Control Technical Committee on Robotics. His research interests include visual servoing, hybrid visual/force control, adaptive control, grasping, and manipulation.

J. Solà received the B.Sc. degree from UPC, in 1995, the M.Sc. degree from the École Doctorale Systèmes de Toulouse in 2003, and the PhD degree from INPT, LAAS-CNRS, in 2007. Prior to joining IRI, he was a postdoc at SRI International and at LAAS-CNRS. He is now a Ramón y Cajal Postdoctoral Fellow at the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain. In the private sector, he has developed power converters for renewable energy systems at Ecotècnia s.c.c.l., Barcelona, now Alstom Wind; and also Litihium-Ion batteries and fault-tolerant power systems for manned deep-water submarines at Ictineu Submarins SL, Barcelona. His research interests are in sensor fusion, computer vision for robotics and SLAM.

J. Andrade-Cetto (S’94-M’95) received the BSEE degree from CETYS Universidad, Mexico, in 1993, the MSEE degree from Purdue University, USA, in 1995, and the PhD degree in Systems Engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, in 2003. He received the EURON Georges Giralt Best PhD Award in 2005. He is Associate Researcher of the Spanish Scientific Research Council and Director of the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona. His research interests include state estimation and computer vision for robotics.