Visual Coordinated Landing of a UAV on a Mobile Robot Manipulator

Vincenzo Lippiello, Rafik Mebarki, and Fabio Ruggiero
PRISMA Lab – Department of Electrical Engineering and Information Technology
University of Naples Federico II
via Claudio 21, 80125, Naples, Italy
Email: vincenzo.lippiello@unina.it, rafik.mebarki@unina.it, fabio.ruggiero@unina.it

***Abstract*—In this paper an image-based visual-servoing control law for the coordinated landing of a VToL UAV on an actuated landing platform is proposed. The landing platform is carried by a mobile manipulator composed of a mobile platform and a robotic arm. The UAV is endowed with a downward camera employed for the coordination of the landing manoeuvres. The presence of redundancy in the system at the task level, the intrinsic redundancy of the mobile manipulator, and the under-actuation of the UAV are explicitly addressed into the proposed control law. The effectiveness of the proposed approach is tested though simulations.**

I. INTRODUCTION

The adoption of *Vertical Takeoff and Landing* (VToL) *Unmanned Aerial Vehicle* (UAV) in the field of service and rescue robotics has seen a healthy increase in the last years. Thanks to the improved autonomy and reliability of the modern UAV, tasks like surveillance, industrial plant and civil structures inspection, mapping, and more, can now be executed with the help of these new machines.

New more challenging scenarios are currently investigated in the research community. The AIRobots project (www.airobots.eu) [1] has successfully addressed inspection tasks with contact (probing) in real industrial scenarios. The Sherpa project (www.sherpa-project.eu) [2] targets the surveillance and rescuing in hostile environments, like those in which civil protection, alpine rescuers and forest guards operate. In this project, a mobile manipulator acts as landing station for the UAVs involved in the mission.

A promising solution for the autonomous and coordinated landing of a UAV on a mobile platform is to use vision to acquire information in real-time. Camera sensors are cheap, lightweight, low-consumption, and passive systems, that can be easily integrated on both the UAV and the rover.

The autonomous landing of a VToL UAV on moving platforms by means of visual information has been extensively investigated in this last decade. A vision-based algorithm to control a VToL UAV while landing on a moving platform is proposed in [3]. Specifically, an *image-based visual servoing* (IBVS) control scheme [4] has been employed to track the platform in two-dimensional image space such that it generates a velocity reference command to an adaptive sliding mode controller. Compared to vision-based control algorithms, that reconstruct a 3D representation of the environment, IBVS is computationally cheaper and less sensitive to depth estimation.

Optical flow measurements have also been deeply used for the stabilization and landing of UAV. By exploiting the average optical flow, a nonlinear controller for hovering flight and landing of a VToL UAV on a moving platform is proposed in [5]. A similar approach, that makes use of a reconstructed stereo optical flow, has been employed in [6] for the autonomous inspection of planar walls.

The design of an internal-model-based error-feedback dynamic regulator, robust with respect to uncertainties on the mechanical parameters, is proposed in [7] for the autonomous landing of a VToL vehicle on a ship whose deck oscillates in the vertical direction.A similar approach based on the vertical motion prediction is proposed in [8].

In [9], [10] the design and implementation of vision-based landing algorithms for UAVs is proposed, where the landing technique is integrated with algorithms of visual target acquisition and navigation. Similarly, the control of a VToL UAV for hovering and landing on a planar target is addressed in [11], based on the homography matrix computed from the on-board video camera and rate-gyros measurements.

In [12], [13] the redundancy of a UAV endowed with a robotic arm is exploited for an aerial manipulation task. However, the redundancy is not exploited to modulate the mobility request for the UAV platform and the robotic arm.

In this paper we propose an IBVS controller for the coordinated landing of a VToL UAV on a landing platform actuated by a mobile manipulator. The data provided by a camera and an *inertial measurement unit* (IMU) onboard to the UAV, as well as the odometry of the mobile manipulator, are employed. The presence of task redundancy in the system and of under-actuation in the UAV are addressed in the control law design by exploiting the different moving capabilities of the UAV and of the mobile platform. More specifically, large movements should be requested to the UAV when it is far from the landing platform, while the dexterity of the robotic

The research leading to these results has been supported by the SHERPA collaborative project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-600958. The authors are solely responsible for its content. It does not represent the opinion of the European Community, that is not responsible for any use that might be made of the information contained therein.

978-1-4799-0880-6/13/$31.00

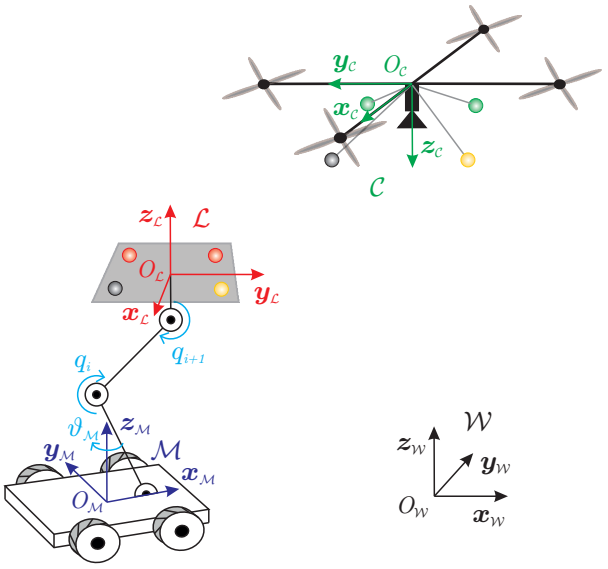

Fig. 1. Reference frames.

arm should be employed during the last part of the landing manoeuvre to ensure a safe and precise contact. A number of case studies are considered in simulation to verify the effectiveness of the proposed approach.

II. MODELING

In this paper, we consider a UAV endowed with a downward-looking camera and a mobile robot manipulator equipped with a landing platform attached at the end-effector of a n-joints robotic arm (see Fig. 1). The goal is to automatically coordinate the motions of the UAV and of the mobile manipulator in order to achieve a safe and efficient landing. The desired behavior of the coordinated control is to fully exploit the specific peculiarities of both platforms, i.e. the large motion capability of the UAV and the dexterity of the robotic mobile arm within its workspace. Via a suitable combination of these features the autonomous landing can be achieved also in unfavorable environmental condition, e.g. in presence of wind.

Let $\mathcal{C} : O_\mathcal{C} - (\boldsymbol{x}_\mathcal{C}\ \boldsymbol{y}_\mathcal{C}\ \boldsymbol{z}_\mathcal{C})$ be the camera reference frame that, without loss of generality, is coincident with the UAV body frame. The position and orientation of the UAV with respect to the world reference frame $\mathcal{W} : O_\mathcal{W} - (\boldsymbol{x}_\mathcal{W}\ \boldsymbol{y}_\mathcal{W}\ \boldsymbol{z}_\mathcal{W})$ are represented by the vector $\boldsymbol{o}_\mathcal{C}^\mathcal{W} = \begin{bmatrix} x_\mathcal{C}^\mathcal{W} & y_\mathcal{C}^\mathcal{W} & z_\mathcal{C}^\mathcal{W} \end{bmatrix}^\top$ and the rotation matrix $\boldsymbol{R}_\mathcal{C}^\mathcal{W} \in \mathrm{SO}(3)$, respectively. The axis $\boldsymbol{z}_\mathcal{W}$ is assumed upward as shown in Fig. 1. In the remainder of the paper we drop the apex $\mathcal{W}$. The vector $\boldsymbol{v}_\mathcal{C} = \begin{bmatrix} \dot{\boldsymbol{o}}_\mathcal{C} & \boldsymbol{\omega}_\mathcal{C} \end{bmatrix}^\top$ represents the velocity of the camera frame (i.e. of the UAV) with respect to $\mathcal{W}$, where $\dot{\boldsymbol{o}}_\mathcal{C}$ and $\boldsymbol{\omega}_\mathcal{C}$ are the linear and angular velocity components, respectively.

Two different reference frames are considered to describe the motion of the mobile manipulator, namely the mobile platform reference frame $\mathcal{M} : O_\mathcal{M} - (\boldsymbol{x}_\mathcal{M}\ \boldsymbol{y}_\mathcal{M}\ \boldsymbol{z}_\mathcal{M})$, which is fixed with the moving platform, with the z-axis upward and the x-axis that points to the front, and the landing platform reference frame $\mathcal{L} : O_\mathcal{L} - (\boldsymbol{x}_\mathcal{L}\ \boldsymbol{y}_\mathcal{L}\ \boldsymbol{z}_\mathcal{L})$, which is fixed with the robotic arm end-effector with the $\boldsymbol{z}_\mathcal{L}$ axis upward and orthogonal to the landing plane. The vector $\boldsymbol{o}_\mathcal{L}$ and the rotation matrix $\boldsymbol{R}_\mathcal{L} \in \mathrm{SO}(3)$ represent the position and the orientation of the landing platform with respect to $\mathcal{W}$, respectively. Analogously, the vector $\boldsymbol{o}_\mathcal{M}$ represents the position of the mobile platform, while the rotation matrix $\boldsymbol{R}_\mathcal{M}(\vartheta_\mathcal{M}) \in \mathrm{SO}(3)$ its orientation, where $\vartheta_\mathcal{M}$ is the angle of rotation of $\mathcal{M}$ around $\boldsymbol{z}_\mathcal{W}$. By assuming that the ground is planar (i.e. $\boldsymbol{z}_\mathcal{M} = \boldsymbol{z}_\mathcal{W}$), the first two components of $\boldsymbol{o}_\mathcal{M}$, $x_\mathcal{M}$ and $y_\mathcal{M}$ are sufficient to describe the position of the mobile platform on the ground, as well as the angle $\vartheta_\mathcal{M}$ is enough to define its orientation. The velocity of the landing platform is represented by $\boldsymbol{v}_\mathcal{L} = \begin{bmatrix} \dot{\boldsymbol{o}}_\mathcal{L}^\top & \boldsymbol{\omega}_\mathcal{L}^\top \end{bmatrix}^\top$, where $\dot{\boldsymbol{o}}_\mathcal{L}$ and $\boldsymbol{\omega}_\mathcal{L}$ are the linear and angular velocity components, respectively.

The pose of the mobile manipulator is described by the joint generalized vector $\boldsymbol{q} = [x_\mathcal{M}\ y_\mathcal{M}\ \vartheta_\mathcal{M}\ q_1\ \dots\ q_n]^\top$, where q_i is the position of the i-th joint of the robotic arm.

A. Differential camera model

The landing platform is endowed with a number of image feature points (e.g. light spots or text markers) observed by the UAV's camera, which are represented as colored spheres in Fig. 1. As well known, $m = 3$ unaligned points are sufficient to track the pose of a rigid object. However, as typically done in the field of visual servoing, a redundant number of m unaligned image features are considered (e.g. $m = 4$). By employing the *pinhole* camera model [14], the i-th observed image point at position $\boldsymbol{p}_i^\mathcal{C} = \begin{bmatrix} x_i^\mathcal{C} & y_i^\mathcal{C} & z_i^\mathcal{C} \end{bmatrix}^\top$ with respect to $\mathcal{C}$, with $i = 1, \dots, m$, is projected onto the normalized image plane of the camera in a 2D image point $\boldsymbol{s}_i$ with the following *perspective law*

$$\boldsymbol{s}_i = \begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \frac{1}{z_i^\mathcal{C}} \begin{bmatrix} x_i^\mathcal{C} \\ y_i^\mathcal{C} \end{bmatrix}. \tag{1}$$

By stacking all the visual measurement, the $(2m \times 1)$ image features vector $\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_1^\top, \dots, \boldsymbol{s}_m^\top \end{bmatrix}^\top$ is obtained.

The motion of the image features in the image plane depends on the relative motion of the camera, i.e. UAV, with respect to the landing platform. By supposing that the feature points are fixed with respect to $\mathcal{L}$ at known positions $\boldsymbol{p}_i^\mathcal{L}$, one can write $\boldsymbol{p}_i^\mathcal{C} = \boldsymbol{o}_\mathcal{L}^\mathcal{C} + \boldsymbol{R}_\mathcal{L}^\mathcal{C} \boldsymbol{p}_i^\mathcal{L}$, where $\boldsymbol{o}_\mathcal{L}^\mathcal{C}$ and $\boldsymbol{R}_\mathcal{L}^\mathcal{C}$ are the position and orientation matrix of $\mathcal{L}$ with respect to $\mathcal{C}$, respectively. Hence, by differentiating $\boldsymbol{s}$ the following image differential kinematics is achieved

$$\dot{\boldsymbol{s}} = \boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{o}_\mathcal{L}^\mathcal{C}, \boldsymbol{R}_\mathcal{L}^\mathcal{C})\, \boldsymbol{\nu}_\mathcal{L}^\mathcal{C}, \tag{2}$$

where $\boldsymbol{J}_s$ is a $(2m \times 6)$ matrix termed *image Jacobian*, and

$$\boldsymbol{\nu}_\mathcal{L}^\mathcal{C} = \begin{bmatrix} \dot{\boldsymbol{o}}_\mathcal{L}^\mathcal{C} \\ \boldsymbol{R}_\mathcal{C}^\top (\boldsymbol{\omega}_\mathcal{L} - \boldsymbol{\omega}_\mathcal{C}) \end{bmatrix} \tag{3}$$

is the relative velocity of the landing platform with respect to the camera, represented with respect to $\mathcal{C}$.

The mapping between the image features velocity, the absolute camera velocity, and the absolute landing platform velocity are retrieved by differentiating $\boldsymbol{o}_\mathcal{L}^\mathcal{C} = \boldsymbol{R}_\mathcal{C}^\top (\boldsymbol{o}_\mathcal{L} - \boldsymbol{o}_\mathcal{C})$:

$$\dot{\boldsymbol{o}}_\mathcal{L}^\mathcal{C} = \boldsymbol{R}_\mathcal{C}^\top (\dot{\boldsymbol{o}}_\mathcal{L} - \dot{\boldsymbol{o}}_\mathcal{C}) + \boldsymbol{S}(\boldsymbol{o}_\mathcal{L}^\mathcal{C}) \boldsymbol{R}_\mathcal{C}^\top \boldsymbol{\omega}_\mathcal{C}, \tag{4}$$

where $\boldsymbol{S}(\cdot)$ is the skew matrix. Folding (4) in (3) yields

$$\boldsymbol{\nu}_\mathcal{L}^\mathcal{C} = \boldsymbol{v}_\mathcal{L}^\mathcal{C} + \boldsymbol{\Gamma}(\boldsymbol{o}_\mathcal{L}^\mathcal{C}) \boldsymbol{v}_\mathcal{C}^\mathcal{C}, \quad \text{with} \quad \boldsymbol{\Gamma}(\cdot) = \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{S}(\cdot) \\ \boldsymbol{O} & -\boldsymbol{I} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{I}$ and $\boldsymbol{O}$ indicate the identity and the null matrix, respectively. Hence, (2) can be rewritten as follows

$$\dot{\boldsymbol{s}} = \boldsymbol{J}_s \boldsymbol{v}_{\mathcal{L}}^{\mathcal{C}} + \boldsymbol{L}_s \boldsymbol{v}_{\mathcal{C}}^{\mathcal{C}}, \tag{6}$$

where the $(2m \times 6)$ matrix $\boldsymbol{L}_s = \boldsymbol{J}_s(\boldsymbol{s}, \boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}, \boldsymbol{R}_{\mathcal{L}}^{\mathcal{C}}) \boldsymbol{\Gamma}(\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}})$ is called *interaction matrix*, which represents the linear mapping between the absolute velocity of the camera and the corresponding velocity of the image features when the landing platform is fixed with respect to $\mathcal{W}$ ($\boldsymbol{v}_{\mathcal{L}}^{\mathcal{C}} = \boldsymbol{0}$).

The analytic expression of the interaction matrix of a point image feature $\boldsymbol{p}_i^{\mathcal{C}}$ is (see [14])

$$\boldsymbol{L}_{s,i}(\boldsymbol{s}_i, z_i^{\mathcal{C}}) = \begin{bmatrix} -\frac{1}{z_i^{\mathcal{C}}} & 0 & \frac{X_i}{z_i^{\mathcal{C}}} & X_i Y_i & -1 - X_i^2 & Y_i \\ 0 & -\frac{1}{z_i^{\mathcal{C}}} & \frac{Y_i}{z_i^{\mathcal{C}}} & 1 + Y_i^2 & -X_i y_o & -X_i \end{bmatrix}. \tag{7}$$

The interaction matrix of a set of m feature points is a $(2m \times 6)$ matrix obtained by stacking the interaction matrix of each point as follows

$$\boldsymbol{L}_s(\boldsymbol{s}, \boldsymbol{z}^{\mathcal{C}}) = \begin{bmatrix} \boldsymbol{L}_{s,1}(\boldsymbol{s}_1, z_1^{\mathcal{C}}) \\ \vdots \\ \boldsymbol{L}_{s,m}(\boldsymbol{s}_m, z_m^{\mathcal{C}}) \end{bmatrix}, \tag{8}$$

with $\boldsymbol{z}^{\mathcal{C}} = \begin{bmatrix} z_1^{\mathcal{C}} & \dots & z_m^{\mathcal{C}} \end{bmatrix}^\top$. The analytic expression of the image Jacobian can be calculated from the interaction matrix through $\boldsymbol{J}_s = \boldsymbol{L}_s(\boldsymbol{s}, \boldsymbol{z}^{\mathcal{C}}) \boldsymbol{\Gamma}(-\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}})$, where the property $\boldsymbol{\Gamma}^{-1}(\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}) = \boldsymbol{\Gamma}(-\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}})$ has been employed.

III. CONTROL LAW

The main objective of the proposed controller is to move the UAV and the mobile manipulator in a coordinated way to make possible the landing of the the UAV on the platform carried by the robotic arm. Let $\boldsymbol{s}_d$ be the constant $(2m \times 1)$ vector describing the desired configuration of the image features in the image plane when the UAV is landed in a suitable way. Hence, the image error

$$\boldsymbol{e}_s = \boldsymbol{s}_d - \boldsymbol{s} \tag{9}$$

can be evaluated in the image plane through the camera image measurements. By differentiating (9) and taking into account (6), one has

$$\dot{\boldsymbol{e}}_s = -\dot{\boldsymbol{s}} = -\boldsymbol{L}_s \boldsymbol{\Gamma}(-\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}) \bar{\boldsymbol{R}}_{\mathcal{C}}^\top \boldsymbol{v}_{\mathcal{L}} - \boldsymbol{L}_s \bar{\boldsymbol{R}}_{\mathcal{C}}^\top \boldsymbol{v}_{\mathcal{C}}, \tag{10}$$

with $\bar{\boldsymbol{R}}_{\mathcal{C}} = \mathrm{diag}(\boldsymbol{R}_{\mathcal{C}}, \boldsymbol{R}_{\mathcal{C}})$ a (6×6) diagonal blocks matrix provided by the IMU onboard the UAV. Finally, by using the differential kinematics of the mobile robotic arm, the velocity of the landing platform $\boldsymbol{v}_{\mathcal{L}}$ can be expressed as

$$\boldsymbol{v}_{\mathcal{L}} = \boldsymbol{J}_q(\boldsymbol{q}) \dot{\boldsymbol{q}}, \tag{11}$$

where $\boldsymbol{J}_q$ is the Jacobian matrix of the mobile system [14].

In order to deal with the under-actuation of the VToL UAV as considered in this paper, the interaction matrix can be divided by columns considering the translational and rotational velocity contributions, respectively, as follows: $\boldsymbol{L}_s \bar{\boldsymbol{R}}_{\mathcal{C}}^\top = \begin{bmatrix} \boldsymbol{L}_P \boldsymbol{R}_{\mathcal{C}}^\top & \boldsymbol{L}_O \boldsymbol{R}_{\mathcal{C}}^\top \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{L}}_P & \bar{\boldsymbol{L}}_O \end{bmatrix}$, with $\bar{\boldsymbol{L}}_P$ and $\bar{\boldsymbol{L}}_O$ are $(2m \times 3)$ matrices. The matrix $\bar{\boldsymbol{L}}_O$ can be further split: $\bar{\boldsymbol{L}}_O = \begin{bmatrix} \bar{\boldsymbol{L}}_{O,xy} & \bar{\boldsymbol{l}}_{O,z} \end{bmatrix}$, where $\bar{\boldsymbol{L}}_{O,xy}$ is a $(2m \times 2)$ matrix and $\bar{\boldsymbol{l}}_{O,z}$ is a $(2m)$ column vector. Hence, (10) can be rewritten as follows

$$\begin{aligned} \dot{\boldsymbol{e}}_s &= \begin{bmatrix} -\bar{\boldsymbol{L}}_P & \boldsymbol{L}_P \boldsymbol{S}(-\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}) \boldsymbol{R}_{\mathcal{C}}^\top - \bar{\boldsymbol{L}}_O \end{bmatrix} \boldsymbol{J}_q \dot{\boldsymbol{q}} \\ &\quad + \begin{bmatrix} \bar{\boldsymbol{L}}_P & \bar{\boldsymbol{L}}_{O,xy} & \bar{\boldsymbol{l}}_{O,z} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{o}}_{\mathcal{C}}^\top & \boldsymbol{\omega}_{\mathcal{C},xy}^\top & \omega_{\mathcal{C},z}^\top \end{bmatrix}^\top \\ &= \begin{bmatrix} \boldsymbol{L}_q & \bar{\boldsymbol{L}}_P & \bar{\boldsymbol{l}}_{O,z} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}^\top & \dot{\boldsymbol{o}}_{\mathcal{C}}^\top & \omega_{\mathcal{C},z}^\top \end{bmatrix}^\top \\ &\quad + \bar{\boldsymbol{L}}_{O,xy} \boldsymbol{\omega}_{\mathcal{C},xy} = \bar{\boldsymbol{L}} \boldsymbol{\nu} + \bar{\boldsymbol{L}}_{O,xy} \boldsymbol{\omega}_{\mathcal{C},xy}, \end{aligned} \tag{12}$$

where $\boldsymbol{L}_q = \begin{bmatrix} -\bar{\boldsymbol{L}}_P & \boldsymbol{L}_P \boldsymbol{S}(-\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}) \boldsymbol{R}_{\mathcal{C}}^\top - \bar{\boldsymbol{L}}_O \end{bmatrix} \boldsymbol{J}_q$, $\bar{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{L}_q & \bar{\boldsymbol{L}}_P & \bar{\boldsymbol{l}}_{O,z} \end{bmatrix}$, $\boldsymbol{\nu} = \begin{bmatrix} \dot{\boldsymbol{q}}^\top & \dot{\boldsymbol{o}}_{\mathcal{C}}^\top & \omega_{\mathcal{C},z} \end{bmatrix}^\top$ is the system velocity control input, and $\boldsymbol{\omega}_{\mathcal{C}} = \begin{bmatrix} \boldsymbol{\omega}_{\mathcal{C},xy}^\top & \omega_{\mathcal{C},z} \end{bmatrix}^\top$.

The proposed control law is

$$\boldsymbol{\nu} = -\bar{\boldsymbol{L}}^{\#} \left(\lambda \boldsymbol{e}_s + \bar{\boldsymbol{L}}_{O,xy} \boldsymbol{\omega}_{\mathcal{C},xy} \right) + \left(\boldsymbol{I} - \bar{\boldsymbol{L}}^{\#} \bar{\boldsymbol{L}} \right) \boldsymbol{\nu}^*, \tag{13}$$

where $\bar{\boldsymbol{L}}^{\#} = \boldsymbol{W}^{-1} \bar{\boldsymbol{L}}^\top (\bar{\boldsymbol{L}} \boldsymbol{W}^{-1} \bar{\boldsymbol{L}}^\top)^{-1}$ is the weighted generalized inverse of the matrix $\bar{\boldsymbol{L}}$, with $\boldsymbol{W}$ a (10×10) weight matrix, $\lambda > 0$ is the control gain, and $\boldsymbol{\nu}^*$ is a velocity control input to control the internal motion of the robotic manipulator in presence of redundancy (*secondary tasks*). By substituting (13) into (12) the error dynamics become $\dot{\boldsymbol{e}}_s + \lambda \boldsymbol{e}_s = \boldsymbol{0}$ that ensures the exponential convergence of the error towards $\boldsymbol{0}$, thus the image feature vector $\boldsymbol{s}$ converges to the desired value $\boldsymbol{s}_d$ (*primary task*). Notice that $\boldsymbol{\omega}_{\mathcal{C}}$ is provided by the IMU, while the relative position $\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}$ of $\mathcal{L}$ with respect to $\mathcal{C}$, that is required for the evaluation of $\boldsymbol{z}^{\mathcal{C}}$, and then for the computation of $\bar{\boldsymbol{L}}$, is estimated with the iterative algorithm proposed in Section III-C.

The system considered in this paper, which is composed of a UAV and a mobile manipulator, is functionally redundant with respect to the desired landing task[1]. This means that a wide range of possible landing trajectories could be achieved during the execution of the primary task. With the proposed control law the redundancy can be fully exploited through the weight matrix $\boldsymbol{W}$ in such a way to achieve the desired behavior for the primary task.

A. Primary-task control

The behavior of the primary task is determined by (13) and it depends on both the gain λ and on the weight matrix $\boldsymbol{W}$. In fact, the gain factor λ determines the convergence rate of the error. For visual servoing applications values of $\lambda \in [0.1, 1]$ are common, and the fine tuning mainly depends on the available camera frame rate (typical values range from 15 to 30 fps). However, for the case study considered, if the initial distance from the UAV and the landing platform is large, the adoption of a small value for λ is preferable to avoid possible saturations of the actuators and local instability of the system. On the other hand, small values of λ could yield to very slow convergence of the error, and then a long landing time. Moreover, the poor capability of the system to reject disturbances could be critical when the norm of the error become small, i.e. when the UAV is close to the landing platform, which is the most critical step of

[1]The independent control inputs are 10 while just 6 could guarantee the fulfilment of the landing task.

the landing manoeuvre. The introduction of simple saturations on the control velocities could only partially solve the problem.

A time varying gain λ depending on the estimated distance $d = \|\tilde{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}\|$ between the UAV and the landing platform is used to introduce a tradeoff between the above requirements:

$$\lambda(d) = \frac{\overline{\lambda} + \underline{\lambda}}{2} - \frac{\overline{\lambda} - \underline{\lambda}}{2} \tanh\left(2\pi \frac{d - \delta_\lambda}{\Delta_\lambda - \delta_\lambda} - \pi\right), \quad (14)$$

where $\underline{\lambda}$ ($\overline{\lambda}$) is the minimum (maximum) desired value of λ, with $\overline{\lambda} > \underline{\lambda} > 0$, and Δ_λ and δ_λ are the corresponding threshold values of the distance, such that $\Delta_\lambda > \delta_\lambda$. With this choice λ ranges from $\underline{\lambda}$ for distances bigger than Δ_λ and progressively increases to $\overline{\lambda}$ for distances less than δ_λ.

On the other hand, the weight matrix $\boldsymbol{W}$ affects the motion distribution over the controlled variables. Different choices of $\boldsymbol{W}$ can lead to large motion of the UAV versus small movements of the mobile platform, and viceversa. The desired behavior requires that the UAV and the mobile robotic arm are differently involved in reason of the current state of the task: the UAV should be strongly involved when the relative distance with respect to the landing platform is large, while the robotic arm should be called in action allowing a safe landing manoeuvre thanks to its rapidity for limited motion. To achieve this behavior, the following time-varying diagonal weight-matrix is proposed

$$\boldsymbol{W}(d) = \mathrm{diag}\left(\alpha \boldsymbol{I}_{3+n}, (1-\alpha)\boldsymbol{I}_4\right), \quad (15)$$

with

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

where $\alpha \in [\underline{\alpha}, 1]$, and δ_W and Δ_W ($\Delta_W > \delta_W$) are the distance thresholds corresponding to $\alpha \cong 1$ and $\alpha \cong \underline{\alpha}$, respectively. The blocks of $\boldsymbol{W}$ weight differently the velocity components of the mobile robotic arm and the UAV by increasing the velocity of the UAV when the distance $d > \Delta_W$, while for distances $d < \delta_W$ the UAV is slowed down and the arm is commanded to accommodate the landing.

B. Secondary tasks

The secondary task which is achievable with the control law (13) can be exploited to control the internal motion that can be generated due to the redundancy of the system. In particular, due to the under-actuation of the UAV, the intrinsic redundancy can be added only by the robotic arm. If this is the case, the redundant degrees of freedom can be exploited to accomplish secondary tasks, e.g. arm joint limits avoidance, obstacle avoidance, manipulability, etc.

A possible choice for the velocity $\boldsymbol{\nu}^*$ is the following

$$\boldsymbol{\nu}^* = \begin{bmatrix} k_0 \dfrac{\partial w(\boldsymbol{q})}{\partial \boldsymbol{q}} & \boldsymbol{0}^\top \end{bmatrix}^\top, \quad (16)$$

where $k_0 > 0$ and $w(\boldsymbol{q})$ is an objective function of the generalized joint vector $\boldsymbol{q}$. Since the velocity $\boldsymbol{\nu}^*$ moves along the direction of the gradient of the object function, it attempts to maximize it locally compatible with the primary task.

The proposed objective function is a time-varying weighted sum of different objective sub-functions as follows

$$w(\boldsymbol{q}) = k_1 w_1(\boldsymbol{q}) + k_2 w_2(\boldsymbol{q}) + k_3 w_3(\boldsymbol{q}), \quad (17)$$

where gains $k_1, k_2, k_3 > 0$ determine the priority of one subtask either another depending on the current state of the system. In particular, the first term generates an internal velocity that try to keep the joint positions as close as possible to the center of the mechanical limits:

$$w_1(\boldsymbol{q}) = -\frac{1}{2n} \sum_{i=1}^{n} \left(\frac{q_i - q_i^*}{\overline{q_i} - \underline{q_i}}\right)^2, \quad (18)$$

where $\overline{q_i}$ ($\underline{q_i}$) is the maximum (minimum) joint limit, and $\boldsymbol{q}^* = \begin{bmatrix} q_1^* & \dots & q_n^* \end{bmatrix}^\top$ is a desired joint configuration (e.g. the middle value of the joint ranges). The second term is employed to avoid singular configuration:

$$w_2(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}_q \boldsymbol{J}_q^\top)}, \quad (19)$$

which vanishes at a singular configuration. The last term is exploited to locally avoid obstacles along the path of the mobile platform:

$$w_3(\boldsymbol{q}) = \min_{\boldsymbol{p}_{\mathcal{M}}, \boldsymbol{p}_o} \|\boldsymbol{p}_{\mathcal{M}} - \boldsymbol{p}_o\|, \quad (20)$$

where $\boldsymbol{p}_{\mathcal{M}}$ is a point of the mobile manipulator, and $\boldsymbol{p}_0$ is a point of the obstacles. Notice that the considered coordinated landing task leave always some degrees of redundancy in the system because the landing position in the world frame is not affected by the primarily task. Hence, this redundant degrees of freedom can fruitfully employed to avoid obstacles or chose a suitable landing area.

C. Iterative estimation algorithm of the relative pose $\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}$

In this section an iterative algorithm for the estimation of the relative pose $\boldsymbol{o}_{\mathcal{L}}^{\mathcal{C}}$, which is required for the control law (13), is proposed. Let $\hat{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}$ be the current estimate of the relative position of $\mathcal{L}$ with respect to $\mathcal{C}$, and $\hat{\boldsymbol{s}}$ the corresponding image feature vector computed as the back-projection of the 3D target points, which depends on $\hat{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}$ and $\boldsymbol{p}_i^{\mathcal{L}}$. By taking into account (2), an estimation of the relative velocity between $\mathcal{L}$ and $\mathcal{C}$ can be achieved as follows

$$\dot{\hat{\boldsymbol{o}}}_{\mathcal{L}}^{\mathcal{C}} = \boldsymbol{J}_{s,P}^{\#}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}, \boldsymbol{R}_{\mathcal{L}}^{\mathcal{C}})\, \dot{\hat{\boldsymbol{s}}}, \quad (21)$$

where $\boldsymbol{J}_{s,P}$ corresponds to the first three rows of $\boldsymbol{J}_s$. The estimation error in the image plane can be defined as

$$\boldsymbol{e}_{\hat{s}} = \boldsymbol{s} - \hat{\boldsymbol{s}}(\hat{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}). \quad (22)$$

Notice that, for the purpose of numerical integration, the feature vector $\boldsymbol{s}$ is constant while the current estimate $\hat{\boldsymbol{s}}$ depends on the current integration time. Therefore, computing the time derivative of (22) yields

$$\dot{\boldsymbol{e}}_{\hat{s}} = -\dot{\hat{\boldsymbol{s}}} = -\boldsymbol{J}_{s,P}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}, \boldsymbol{R}_{\mathcal{L}}^{\mathcal{C}}) \dot{\hat{\boldsymbol{o}}}_{\mathcal{L}}^{\mathcal{C}}. \quad (23)$$

Assuming that the matrix $\boldsymbol{J}_{s,P}$ is non singular, the choice

$$\dot{\boldsymbol{e}}_{\hat{s}} = -\dot{\hat{\boldsymbol{s}}} = -\boldsymbol{J}_{s,P}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{o}}_{\mathcal{L}}^{\mathcal{C}}, \boldsymbol{R}_{\mathcal{L}}^{\mathcal{C}}) \boldsymbol{K}_s \dot{\hat{\boldsymbol{o}}}_{\mathcal{L}}^{\mathcal{C}}, \quad (24)$$

with $\boldsymbol{K}_s$ (diagonal) positive definite, leads to the linear system $\dot{\boldsymbol{e}}_{\hat{s}} + \boldsymbol{K}_s \boldsymbol{e}_{\hat{s}} = \boldsymbol{0}$, so guaranteing the asymptotical convergence of the estimation error to zero with a rate that depends on the

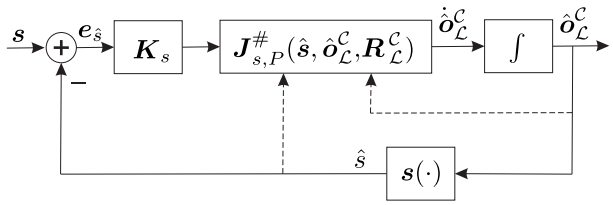

Fig. 2. Relative pose estimation algorithm.

eigenvalues of $\boldsymbol{K}_s$. The convergence to zero of the error $\boldsymbol{e}_{\hat{s}}$ ensures the asymptotic convergence of the estimate $\hat{\boldsymbol{o}}^{\mathcal{C}}_{\mathcal{L}}$ to the true value $\boldsymbol{o}^{\mathcal{C}}_{\mathcal{L}}$.

The block scheme of the proposed relative pose estimation algorithm is shown in Fig. 2, where $\boldsymbol{s}(\cdot)$ denotes the function computing the image feature vector of the "virtual" image corresponding to the current estimate $\hat{\boldsymbol{o}}^{\mathcal{C}}_{\mathcal{L}}$.

Notice that the proposed relative position estimation method is as efficient in terms of accuracy, speed of convergence and computational load, as the initial estimate $\hat{\boldsymbol{o}}^{\mathcal{C}}_{\mathcal{L}}$ is close to the true value. In the proposed context the estimate on an image taken at time $\bar{t}$ is computed assuming as initial value the estimate computed on the image taken at time $\bar{t} - T$, being T the sampling time of the image. Typically, an enough accuracy in the estimation is achieved with very few iterations.Moreover, the image feature vector $\boldsymbol{s}$ employed as reference input for this algorithm is the same already extracted for the computation of the control law, so no significant increase of complexity is requested for the execution in real-time of this iterative algorithm.

IV. SIMULATION

The proposed visual coordinated landing algorithm is tested in simulation by considering the fully dynamics of both the UAV and the mobile manipulator in a MATLAB/SIMULINK environment. Figure 3 shows some screen-shots extracted from the MATLAB animation of one of the considered case study where also the presence of an obstacle has been considered (see the following subsections).

A. Low-level velocity controllers

A Cartesian-velocity control has been adopted for the UAV [15], [6], [16], [17], and for the mobile manipulator at joint level [14]. The kinematic and dynamic parameters of the mobile manipulator are inspired from a KUKA YOUBOT, while the UAV model simulates an ASCTEC PELICAN.

Moreover, as for a real application where the safety of the system is considered, the magnitude of the Cartesian velocity control-input for the UAV has been saturated to 0.5 m/s, as well as the maximum roll and pitch angles to 20 deg.

B. Secondary tasks: implementation details

The subtasks of Section III-B have been implemented by taking into account some peculiarities of the setup. In particular, in the first subtask (18) the desired joint configuration $\boldsymbol{q}^*$ is chosen in such a way to keep the landing plane horizontal and sufficiently far from the arm joint limits.

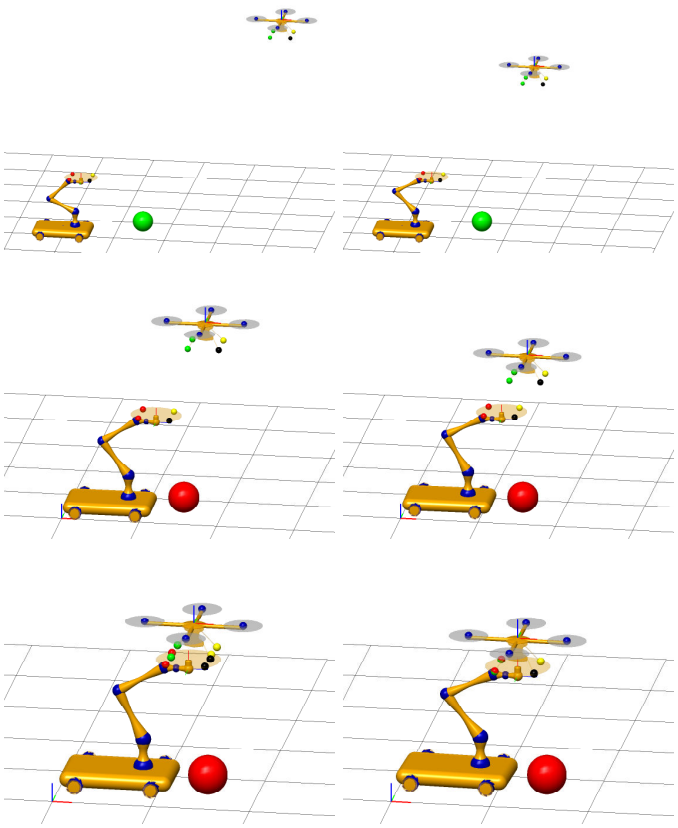

Fig. 3. Screen-shots of the animation corresponding to the case study D. The green ball represents the obstacles when it is at a safety distance, while it is depicted in red when it becomes at a dangerous distance and the obstacle-avoidance secondary task is activated.

The secondary task addressing the singularity of the mobile manipulation system has been simplified by tanking into account the presence of the previous secondary task that already try to keep the arm in a suitable configuration. However, by analyzing the arm kinematics, a geometric locus corresponding to a singularity is the vertical axis passing at the base of the arm –commonly called *shoulder singularity* for anthropomorphic robot. For this reason the secondary task has been simplified by trying to keep the tool center point (TCP) as far as possible from this axis:

$$w_2(\boldsymbol{q}) = \left\| \boldsymbol{o}^{\mathcal{M}}_{\mathcal{L},xy} - \boldsymbol{o}^{\mathcal{M}}_{0,xy} \right\|, \tag{25}$$

where $\boldsymbol{o}^{\mathcal{M}}_0$ is the position of the robotic arm base with respect to $\mathcal{M}$[2], while the subs xy indicates that only the x and y components of the vector are considered.

Finally, the last secondary task has been simplified in our case studies where only one spherical obstacle has been considered. In details, only the positions of the mobile and landing platform have been considered yielding

$$w_3(\boldsymbol{q}) = \min\left(\|\boldsymbol{o}_{\mathcal{M}} - \boldsymbol{p}_o\|, \|\boldsymbol{o}_{\mathcal{L}} - \boldsymbol{p}_o\| \right), \tag{26}$$

where $\boldsymbol{p}_o$ corresponds to the center of the obstacle. The corresponding gain has been modified by introducing an activation

[2]Notice that, in general, the reference frame $\mathcal{M}$ attached to the mobile platform could be different with respect to the arm base frame.

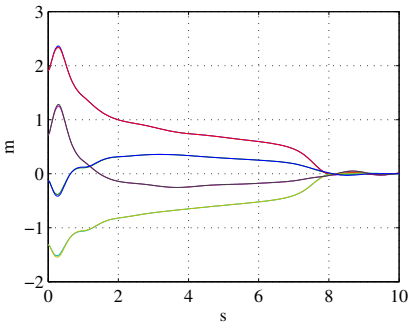

Fig. 4. Image error in the case study A.

threshold as follows

$$k_3(w_3) = \begin{cases} 0 & \text{if } w_3 > \delta_o \\ k_3'(\delta_o - w_3) & \text{if } w_3 \leq \delta_o \end{cases}, \qquad (27)$$

where δ_o is the activation distance, and $k_3' > 0$.

C. Case studies

The following case studies are considered:

A) No weight matrix, no subtask ($\boldsymbol{W} = \boldsymbol{I}$, $k_0 = 0$).
B) No secondary tasks ($k_0 = 0$).
C) Secondary tasks without obstacle avoidance ($k_3 = 0$).
D) Secondary tasks with obstacle avoidance.

Notice that the presence of an obstacle (a sphere with a radius of 10 cm) is explicitly considered only in case D.

The parameters employed in the simulation are the following: $\overline{\lambda} = 0.1$, $\underline{\lambda} = 2$, $\delta_\lambda = 0.1$ m, $\Delta_\lambda = 0.5$ m, $\underline{\alpha} = 0.02$, $\delta_W = 0.5$ m, $\Delta_W = 2$ m. For the secondary tasks the following parameters have been employed: $k_0 = 5$, $k_1 = 4$, $k_2 = 1$, $k_3 = 100$, $\boldsymbol{p}_o = [0.75 \;\; 0.25 \;\; 0.1]^\top$m, $\delta_o = 0.5$ m, and $k_3' = 20$. The initial configuration of the system is $x_{\mathcal{M}} = y_{\mathcal{M}} = \vartheta_{\mathcal{M}} = 0$, $\boldsymbol{o}_{\mathcal{C}} = [2 \;\; 1 \;\; 2]^\top$m, and $\boldsymbol{R}_{\mathcal{C}} = \boldsymbol{I}$.

The camera is calibrated and runs at 25 fps; the image features are the corners of a 20 cm square centered in $\mathcal{L}$.

The trajectory of the image error $\boldsymbol{e}_s$ in the normalized image plane is shown in Fig. 4 for case A (similar results are achieved for the other cases). The convergence to zero highlights the fulfillments of the main task.

Figure 5 shows the Cartesian trajectories of the reference frames of the mobile platform, the landing platform, and the UAV. In case A, the absence of the weighted matrix make free the system to follow the minimum energy solution that requires a similar motion to both the involved systems, the UAV and the mobile manipulator. This determines an undesired motion of the mobile platform also when the UAV is far away. In the other cases, the control of the main task behavior guarantees the involvement of the two systems in the desired way, i.e. requiring the motion of the mobile arm mainly for the last part of the landing manoeuvre. However, in case B the arm performs the landing manoeuvre with wide joint motions leading the arm in a poor dexterous configuration. This undesired behavior is corrected with the activation of the

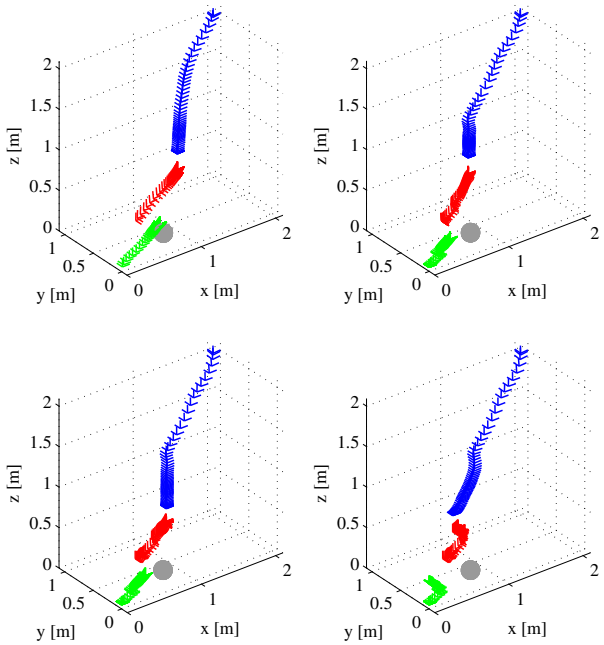

Fig. 5. Cartesian trajectories of the mobile base frame (green), TCP frame (red), and UAV frame (blue). Case study A: top-left, B: top-right, C: bottom-left, and D: bottom-right. The grey sphere represents the obstacle.

secondary task, as in cases C and D. However, in the former the presence of the obstacles is not yet considered and the mobile platform collides during the last part of the landing. In the latter, the landing manoeuvre is perfectly achieved by also avoiding the obstacle.

The accomplishment of the secondary tasks in all case studies is quantified in Fig. 6. In details, the norm of the distance in the arm joint space between the desired configuration and the actual one is shown in Fig. 6(a). In cases A and B, this subtask is ignored but the results are poor only for the case B because the weighted solution without the secondary tasks generates large movements of the arm ignoring the dexterity. Obviously, in case D a little degradation is seen when the object is under the activation distance because the obstacle avoidance subtask use part of the degrees of freedom needed by all the subtasks.

The manipulability increases in cases C and D [see Fig. 6(b)], when the corresponding subtask is active.

Figure 6(c) clearly shows the effectiveness of the obstacle avoidance subtask. In fact, the mobile platform is able to avoid the collision with the obstacle only in case D, as also shown in Fig. 5. Notice that the reference frame of the mobile platform represented in Fig. 5 is centered in the middle of the mobile platform, hence a collision with the obstacle only happen when the relative distance become lower than 0.4 m.

Finally, the performance of the proposed iterative estimation algorithm of the relative pose between the landing platform and the camera is shown in Fig. 7 for all the case studies, with the gain matrix $\boldsymbol{K}_s = 150\boldsymbol{I}$. The number of required iteration ranges between 3 and 5, depending on the relative velocity of the two moving systems, hence without a significant increase of the computational cost.

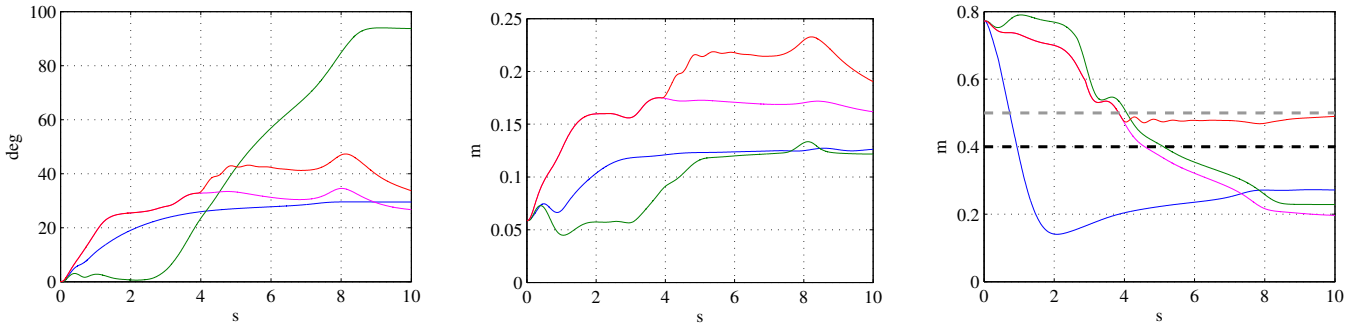

Fig. 6. Secondary tasks fulfillment for case study A in blue, B in green, C in magenta, and D in red. *Left*: Norm of the distance in the arm joint space between the current joint pose and the desired one. *Center*: Distance between the TCP and the shoulder singularity axis. *Right*: Distance between the center of the mobile platform and the obstacle; the black dashed line indicates the threshold under which a collision happen, while the gray dashed line indicates the threshold of activation (δ_o) of the obstacle avoidance secondary task which has been employed in the case study D.

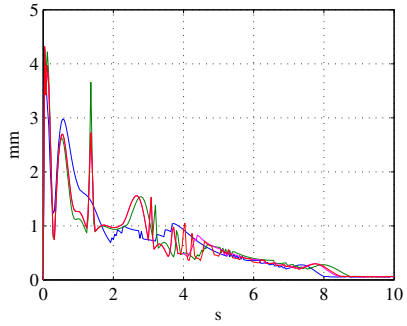

Fig. 7. Norm of the estimation error of the relative pose between the landing platform and the camera $\boldsymbol{o}^{\mathcal{C}}_{\mathcal{L}}$: case study A in blue, B in green, C in magenta, and D in red.

Notice that $\boldsymbol{o}^{\mathcal{C}}_{\mathcal{L}}$ is mainly employed for the interaction matrix estimation, hence the degradation of performance in a real case should not affect the performance of the control.

V. CONCLUSION

An IBVS control for the coordinated landing of an UAV on an actuated landing platform has been proposed. A system composed of a landing platform carried by a mobile manipulator and a VToL UAV endowed with a downward camera has been considered. The presence of redundancy in the system at the task level, the intrinsic redundancy of the mobile manipulator, and the under-actuation of the UAV have been taken into account for the design of the control law. The performance of the proposed approach has been shown in several case studies.

REFERENCES

[1] 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.

[2] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty, A. Kleiner, V. Lippiello, A. Finzi, B. Siciliano, A. Sala, and N. Tomatis, "The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *IEEE Int. Symposium on Safety, Security, and Rescue Robotics*, pp. 1–4, 2012.

[3] D. Lee, T. Ryan, and H. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, pp. 971–976, 2012.

[4] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[5] B. Herisse, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, 2012.

[6] V. Lippiello and B. Siciliano, "Wall inspection control of a VTOL unmanned aerial vehicle based on a stereo optical flow," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4296–4302, 2012.

[7] L. Marconi, A. Isidori, and A. Serrani, "Autonomous vertical landing on an oscillating platform: an internal-model based approach," *Automatica*, vol. 38, pp. 21–32, Jan. 2002.

[8] X. Yang, H. Pota, M. Garratt, and V. Ugrinovskii, "Prediction of vertical motions for landing operations of UAVs," in *47th IEEE Conference on Decision and Control*, pp. 5048–5053, 2008.

[9] S. Saripalli, J. Montgomery, and G. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–380, 2003.

[10] C. Sharp, O. Shakernia, and S. Sastry, "A vision system for landing an unmanned aerial vehicle," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1720–1727 vol.2, 2001.

[11] H. de Plinval, P. Morin, P. Mouyon, and T. Hamel, "Visual servoing for underactuated VTOL UAVs: A linear, homography-based approach," in *IEEE Int. Conf. on Robotics and Automation*, pp. 3004–3010, 2011.

[12] V. Lippiello and F. Ruggiero, "Exploiting redundancy in cartesian impedance control of UAVs equipped with a robotic arm," in *IEEE/RSJ Inter. Conf. on Intelligent Robots and System (IROS)*, 2012.

[13] 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.

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

[15] V. Lippiello, G. Loianno, and B. Siciliano, "MAV indoor navigation based on a closed-form solution for absolute scale velocity estimation using optical flow and inertial data," in *IEEE Conference on Decision and Control and European Control Conference*, pp. 3566–3571, 2011.

[16] V. Lippiello and R. Mebarki, "Closed-form solution for absolute scale velocity estimation using visual and inertial data with a sliding least-squares estimation," in *21st Mediterranean Conference on Control and Automation*, 2013.

[17] Lippiello, Vincenzo, Villani, Luigi, Siciliano, and Bruno, "An open architecture for sensory feedback control of a dual-arm industrial robotic cell," *Industrial Robot: An International Journal*, vol. 34, no. 1, pp. 46–53, 2007.