Task Priority Control for Aerial Manipulation

Angel Santamaria-Navarro, Vincenzo Lippiello and Juan Andrade-Cetto

Abstract—**This paper presents a task oriented control strategy for aerial vehicles equipped with a manipulator. A camera is attached to the end-effector of the manipulator to perform a primary task consisting on visual servoing towards a desired target. Over-actuation of the whole quadrotor-arm system is exploited to achieve secondary velocity tasks. One subtask is proposed to horizontally stabilize the platform during flight by aligning the arm center of gravity with the quadrotor gravitational vector. The arm singularities and manipulability are addressed by another subtask that leads the arm to a preferable configuration, and also takes into account the arm joint limits. The performance of the whole visual servo and secondary tasks control scheme is shown in a Robot Operating System (ROS) implementation.**

I. INTRODUCTION

Search and rescue missions can greatly benefit from the use of robotic systems. Aerial robots can provide essential support to human task forces in situation assessment and surveillance. In the case of small size Unmanned Aerial Vehicles (UAVs), their maneuverability, low cost and reduced size, provide enhanced accessibility to hazardous environments [1, 2, 3]. Visual feedback plays an important role not only for the mission goal but for the control of the platform [4].

We concentrate on the use of quadrotors for search and rescue operations. Quadrotors are equipped with four aligned coplanar propellers. Due to their symmetric design, motion control is achieved by altering the rotation rate of one or more of these propellers, thereby changing its torque load and thrust lift characteristics. With this actuation technique, a quadrotor becomes an underactuated vehicle with only 4 DOF. This underactuation carries a limitation when an inspection goal should be done by rigidly attaching the camera to the quadrotor. To address underactuation, recent advances in UAV size-to-payload and manipulator weight-to-payload ratios suggest the possibility of attaching a manipulator arm to the base of the robot [5].

In this work, we simulate the attachment of a serial arm to the quadrotor with a camera at its end effector (see Fig. 1(a)). Providing extra degrees of freedom to the camera allows to efficiently maneuver the platform during inspection tasks whilst maintaining the target on sight. This DOF redundancy is exploited not only to achieve a desired visual servo task,

A. Santamaria-Navarro 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, cetto}@iri.upc.edu

V. Lippiello is with Università degli Studi di Napoli Federico II. Stanza 3.12, edificio 3/A, Via Claudio 21, 80125 Napoli, Italy, e-mail: vincenzo.lippiello@unina.it
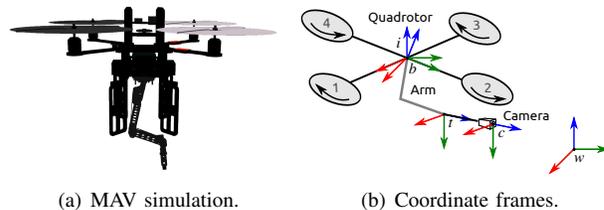
(a) MAV simulation. (b) Coordinate frames.

Fig. 1. Quadrotor-arm system.

but to do so whilst attaining secondary goals during the mission. In this paper, we use uncalibrated image-based visual servo [6] for the main control task, and secondary tasks that help keeping the platform stable.

Flying with a suspended load is a challenging task because the load significantly changes the flight characteristics of the aerial vehicle, and the stability of the vehicle-load system must be preserved. Therefore, it is essential that the flying robot has the ability to minimize the effects of the arm on the flying system during the assigned maneuvers [7], e.g. the change of the center of mass during flight can be solved designing a low-level attitude controller such as a Cartesian impedance controller [8], or an adaptive controller [9]. To avoid this undesired behavior, the redundancy of the system in the form of extra DOFs could be exploited to develop a secondary stabilizing task after the primary servoing task. Other secondary tasks that can also be performed within a hierarchical framework, could be designed such as to optimize some given quality indices, e.g. manipulability, joint limits, etc., [10, 11].

The remainder of this article is structured as follows. In the next section a brief description of the quadrotor-arm robot model is given. The task priority control of the over-actuated system is described in Sec. 3. Sec. 4 shows the feasibility of the proposed control strategy through extensive simulation. Finally, conclusions are given in Sec. 5.

II. ROBOT KINEMATICS

Consider the quadrotor-arm system equipped with a camera mounted on the arm's end-effector and the coordinate frames shown in Fig. 1(b). The goal is to servo the camera to a desired target, say for instance, a fiducial mark on an object to be manipulated. We assume a visual servo approach that provides camera velocities to reach this task [6].

Without loss of generality, we consider the world frame w to be located at the target. With this, the position of the target with respect to the camera in c can be computed integrating the camera velocities obtained from the visual servo.

The quadrotor high-level controller commands velocities $(v_{qx}, v_{qy}, v_{qz}, \omega_{qz})$ in the so-called inertial frame i (both frames i and w have their x and y axes in parallel planes

but rotated about the yaw axis) and the low-level attitude controller moves the quadrotor body frame b to reach the desired velocities in i (there exist only roll and pitch rotations between i and b).

With the arm base frame coincident with the quadrotor body frame b, the relation between the quadrotor inertial frame and the target frame is given by the concatenation of the kinematic transforms $\mathbf{T}_w^i = \mathbf{T}_b^i \, \mathbf{T}_t^b \, \mathbf{T}_c^t \, (\mathbf{T}_c^w)^{-1}$.

We are in the position now to define a joint quadrotor-arm Jacobian that relates the local translational and angular velocities of the platform acting on the inertial frame and those of the n arm joints, $\mathbf{v}_{qa} = (v_{qx}, v_{qy}, v_{qz}, \omega_{qx}, \omega_{qy}, \omega_{qz}, \dot{q}_{a1}, \ldots, \dot{q}_{an})^T$, to the desired camera velocities as computed from the visual servo

$$\mathbf{v}_c = \mathbf{J}_{qa} \, \mathbf{v}_{qa} \,, \tag{1}$$

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 quadrotor movement and the arm kinematics (superscripts indicate the reference frame to make it clear to the reader) as $\mathbf{v}_c^c = \mathbf{v}_q^c + \mathbf{v}_a^c$ where $\mathbf{v}_a^c$ is obtained with the arm Jacobian $\mathbf{J}_a$

$$\mathbf{v}_a^c = \begin{bmatrix} \mathbf{R}_b^c & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b^c \end{bmatrix} \mathbf{J}_a \, \dot{\mathbf{q}}_a = \overline{\mathbf{R}}_b^c \mathbf{J}_a \, \dot{\mathbf{q}}_a \,, \tag{2}$$

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

$$\mathbf{v}_q^c = \overline{\mathbf{R}}_b^c \begin{bmatrix} \mathbf{v}_q^b + \boldsymbol{\omega}_q^b \times \mathbf{r}_c^b \\ \boldsymbol{\omega}_q^b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^c & \mathbf{R}_b^c \, [\mathbf{r}_c^b]_\times^T \\ \mathbf{0} & \mathbf{R}_b^c \end{bmatrix} \mathbf{v}_q^b \,. \tag{3}$$

The term $\mathbf{r}_c^b(\mathbf{q}_a)$ indicates the vector between the b and c frames, i.e. the direct arm kinematics.

Finally, the velocity vector of the quadrotor in the body frame, $\mathbf{v}_q^b$, can be obtained using the quadrotor Jacobian $\mathbf{J}_q$ formed by the rotation $\mathbf{R}(\phi, \theta)$ and the transfer matrix $\mathbf{T}(\phi, \theta)$ between the quadrotor inertial and body frames

$$\mathbf{v}_q^b = \mathbf{J}_q \, \mathbf{v}_q^i = \begin{bmatrix} \mathbf{R} & 0_{3\times3} \\ 0_{3\times3} & \mathbf{T} \end{bmatrix} \mathbf{v}_q^i \,, \tag{4}$$

with

$$\mathbf{R} = \begin{bmatrix} c_\theta & s_\theta \, s_\phi & s_\theta \, c_\phi \\ 0 & c_\phi & -s_\phi \\ -s_\theta & c_\theta \, s_\phi & c_\theta c_\phi \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 1 & s_\phi \, t_\theta & c_\phi \, t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}, \tag{5}$$

and the notation $s_x = \sin(x)$, $c_x = \cos(x)$, $t_x = \tan(x)$.

Camera velocities are obtained using a visual servo front end. Specifically we use [6], but any other method could be used instead [12, 13, 14],

$$\mathbf{v}_c = -\lambda \mathbf{J}_{vs}^+ \mathbf{e} \,, \tag{6}$$

where λ is a gain, $\mathbf{J}_{vs}^+$ corresponds to the pseudo-inverse of the so-called interaction matrix, and $\mathbf{e}$ is the target error in the camera frame. Combining Eqs. 6 and 1, we get

$$\mathbf{J}_{qa} \mathbf{v}_{qa} = -\lambda \mathbf{J}_{vs}^+ \mathbf{e} \,. \tag{7}$$

Unfortunately, the quadrotor is an underactuated vehicle [15] with only 4 DOF. Its pitch and roll are internally controlled by the attitude subsystem and we cannot directly

actuate them. So, to remove these variables from the control command, their contribution to the visual servo error can be isolated from that of the other control variables 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 [16].

Rearranging terms

$$\mathbf{J}_{qa1} \dot{\mathbf{q}} = \underbrace{-\lambda \mathbf{J}_{vs}^+ \mathbf{e} - \mathbf{J}_{qa2} \begin{bmatrix} \omega_{qx} \\ \omega_{qy} \end{bmatrix}}_{\dot{\mathbf{q}}_1}, \tag{8}$$

where $\mathbf{J}_{qa2}$ is the Jacobian formed by the columns of $\mathbf{J}_{qa}$ corresponding to ω_{qx} and ω_{qy}, and $\mathbf{J}_{qa1}$ is the Jacobian formed by all other columns of $\mathbf{J}_{qa}$, corresponding to the actuated variables $\dot{\mathbf{q}} = [v_{qx}, v_{qy}, v_{qz}, \dot{q}_{a1}, \ldots, \dot{q}_{an}]^T$.

With this, $\dot{\mathbf{q}}_1$ becomes our primary task velocity corresponding to the visual servo.

$$\dot{\mathbf{q}} = \mathbf{J}_{qa1}^\# \, \dot{\mathbf{q}}_1, \tag{9}$$

where $\mathbf{J}_{qa1}^\# = \mathbf{W}^{-1} \mathbf{J}_{qa1}^T \, (\mathbf{J}_{qa1} \, \mathbf{W}^{-1} \, \mathbf{J}_{qa1}^T)^{-1}$ is the weighted generalized inverse of the matrix $\mathbf{J}_{qa1}$ where the weight matrix $\mathbf{W}$ affects the motion distribution over the controlled variables considering the different moving capabilities of the robotic arm and the quadrotor. More specifically, large movements of the flying platform should be achieved by the quadrotor leaving more precise motion to the robotic arm due to its dexterity, i.e. regulating their actuation as a function of the distance d of the platform to the target.

To achieve this behavior, we define a time-varying diagonal weight-matrix as proposed in [17]

$$\mathbf{W}(d) = \text{diag}((1 - \alpha) \, \mathbf{I}_4, \alpha \, \mathbf{I}_n), \tag{10}$$

with n the arm's DOF and

$$\alpha(d) = \frac{1 + \underline{\alpha}}{2} + \frac{1 - \underline{\alpha}}{2} \tanh\left(2\pi \frac{d - \delta_W}{\Delta_W - \delta_W} - \pi\right), \tag{11}$$

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 $\mathbf{W}$ 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 the precise arm movements. Note that since the quadrotor is underactuated, we have to choose $\underline{\alpha}$ in such a way that the arm can still compensate the roll and pitch movements produced by the quadrotor during flight not to loose the target from the image plane.

III. TASK PRIORITY CONTROL

The redundancy obtained with the arm's extra degrees of freedom can be exploited to achieve additional tasks acting on the null space of the quadrotor-arm Jacobian [18], while preserving the primary task in Eq. 9:

$$\dot{\mathbf{q}} = \mathbf{J}_{qa1}^\# \, \dot{\mathbf{q}}_1 + \mathbf{N}_{qa1} \, \dot{\mathbf{q}}_0 \,, \tag{12}$$

where $\mathbf{N}_{qa1} = (\mathbf{I} - \mathbf{J}_{qa1}^+ \mathbf{J}_{qa1})$ is the null space projector for the main task. With this, the secondary task velocity $\dot{\mathbf{q}}_0$ will be used to reconfigure the robot structure without changing

both the position and orientation of the end-effector (usually referred to as *internal motion*).

One possible way to specify the secondary task is to choose the velocity vector $\dot{\mathbf{q}}_0$ as the gradient of a scalar objective function to achieve some kind of optimization [11, 19]. With a more general approach, let $\boldsymbol{\sigma} = \mathbf{f}(\mathbf{q}) \in \mathbb{R}^m$ be the variables of a secondary task to be controlled, the following differential relationship holds:

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_\sigma(\mathbf{q})\dot{\mathbf{q}}, \tag{13}$$

where $\mathbf{J}_\sigma(\mathbf{q}) \in \mathbb{R}^{m \times (4+n)}$ is the configuration-dependent task Jacobian. Hence, by inverting Eq. 13 and by considering a regulation problem of $\boldsymbol{\sigma}$ to the desired value $\boldsymbol{\sigma}^*$, the following general solution can be employed

$$\dot{\mathbf{q}} = \mathbf{J}_{qa1}^{\#} \dot{\mathbf{q}}_1 + \mathbf{N}_{qa1} \mathbf{J}_\sigma^+ \boldsymbol{\Lambda}_\sigma \tilde{\boldsymbol{\sigma}}, \tag{14}$$

where $\boldsymbol{\Lambda}_\sigma \in \mathbb{R}^{m \times m}$ is a positive-definite matrix of gains, and $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \boldsymbol{\sigma}$ is the task error.

Considering the high redundancy of the quadrotor-arm system, multiple secondary tasks can be arranged in hierarchy. As proposed in [17], the secondary objective function can be defined as a weighted sum of different objective sub-functions, with the advantage that the weights can be modeled time-varying, i.e. the effect of the secondary task can be changed depending on the phase of the flight. However, the use of some of the sub-functions at the same time can produce undesired behaviours on the arm due to opposite effects of the sub-tasks. To deal with that and to avoid conservative stability conditions [20], the *augmented inverse-based projections method* is here considered [10]. In detail, the generic task is not projected onto the null space of the high hierarchy task, but onto the null space of the task achieved by considering the augmented Jacobian of all the higher hierarchy tasks.

In this work we consider two sub-tasks: 1) center of gravity control, 2) joint-limits avoidance control. By denoting with $\mathbf{J}_G$ and $\mathbf{J}_L$ the Jacobian matrices for the center of gravity and for the joint-limits avoidance control, respectively, where the priority of the task follows the previous enumerating order, the desired system velocity can be rewritten as follows,

$$\dot{\mathbf{q}} = \mathbf{J}_{qa1}^{\#} \dot{\mathbf{q}}_1 + \mathbf{N}_{qa1} \mathbf{J}_G^+ \tilde{\sigma}_G + \mathbf{N}_{qa1|G} \mathbf{J}_L^+ \tilde{\sigma}_L, \tag{15}$$

with $\mathbf{N}_{qa1|G}$, the joint projector of the primary task and of the center of gravity secondary task, which is defined as

$$\mathbf{N}_{qa1|G} = (\mathbf{I} - \mathbf{J}_{qa1|G}^+ \mathbf{J}_{qa1|G}), \tag{16}$$

and $\mathbf{J}_{qa1|G}$ represents the augmented Jacobian $[\mathbf{J}_{qa1} \mathbf{J}_G]^T$.

We now define the scalar objective functions for each of the secondary tasks in the hierarchy.

A. Center of Gravity

If the arm and quadrotor center of gravity (CoG) are not vertically aligned, the motion of the arm produces an undesired static 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 arm CoG with respect to the z axis of the i frame, which can be written as

$$\sigma_G = \lambda_G \, (\mathbf{p}_{Gxy}^i)^T \, \mathbf{p}_{Gxy}^i, \tag{17}$$

where λ_G is a suitable positive gain and with

$$\mathbf{p}_{Gxy}^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}_b^i \, \mathbf{p}_G^b, \tag{18}$$

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

$$\mathbf{p}_G^b = \frac{\sum_{i=1}^n m_i \, \mathbf{p}_{Gi}^b}{\sum_{i=1}^n m_i}, \tag{19}$$

where m_i and $\mathbf{p}_{Gi}^b$ represent the i-th link mass and the position of its CoG, respectively.

As proposed in [21], we can define the CoG of a partial chain of links, with respect to the body frame, from the link j to the end-effector as

$$\mathbf{p}_{Gj}^{*b} = \mathbf{R}_j^b \frac{\sum_{i=j}^n m_i \, \mathbf{p}_{Gi}^b}{\sum_{i=j}^n m_i}, \tag{20}$$

where $\mathbf{R}_j^b$ is the existing rotation between the link j and the quadrotor body frame. Notice that all these quantities are functions of the current joint configuration $\mathbf{q}_a$.

The differential relationship between the CoG, $\mathbf{p}_G$, and the arm joint values is $\dot{\mathbf{p}}_G^b = \mathbf{J}_G^b \, \dot{\mathbf{q}}_a$, where $\mathbf{J}_G^b \in \mathbb{R}^{3 \times n}$ is the CoG Jacobian, expressed in the quadrotor body frame, defined as $\mathbf{J}_G^b = \frac{\partial \mathbf{p}_G^b}{\partial \mathbf{q}_a} = (\mathbf{J}_{G1}^b ... \mathbf{J}_{Gn}^b)$, with $\mathbf{J}_{Gi}^b$ the individual joint i Jacobian formulated from the partial CoG

$$\mathbf{J}_{Gj}^b = \frac{\sum_{i=j}^n m_i}{\sum_{i=0}^n m_i} \left(\mathbf{z}_j \times \mathbf{p}_{Gj}^{*b} \right). \tag{21}$$

Notice how the resultant linear velocity is scaled by the mass of the partial CoG in Eq. 21 because the CoG is the average of the multi-mass system and high velocities on smaller masses play a lesser role on the total velocity of the CoG.

Finally, the corresponding task Jacobian from the derivative of Eq. 17 becomes

$$\mathbf{J}_G = \begin{bmatrix} \mathbf{0}_{1 \times 4} & 2 \lambda_G \, (\mathbf{p}_{Gxy}^i)^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}_b^i \mathbf{J}_G^b \end{bmatrix}. \tag{22}$$

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.

B. Joint Limits

To avoid arm joint limits we can drive the arm joints toward a desired value $\mathbf{q}_a^*$ that can be chosen far from an undeliverable configuration and/or close one characterized by a high manipulability index or suitable with respect to the assigned task. Hence our cost function for this task is a weighted squared sum of the joint angle differences from the desired values over the joint limit ranges

$$\sigma_L = \sum_{i=1}^n \lambda_L \left(\frac{q_{ai} - q_{ai}^*}{\overline{q}_{ai} - \underline{q}_{ai}} \right)^2, \tag{23}$$

(a) Initial situation   (b) Initial target reprojection   (c) Final situation   (d) Final target reprojection
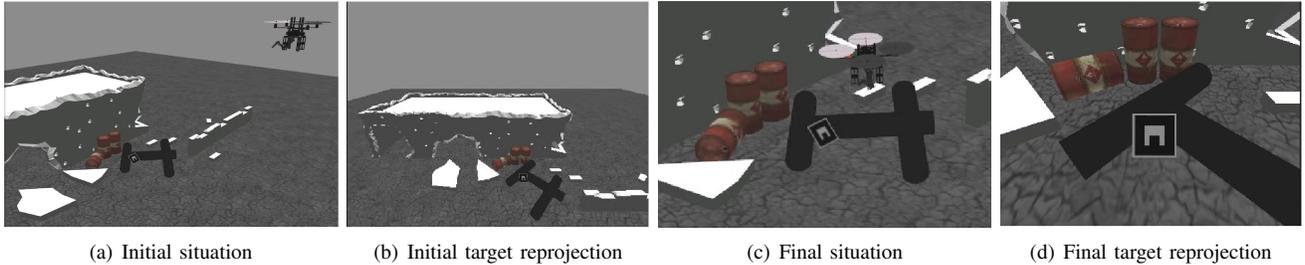
Fig. 2. ROS simulation of a visual servo quadrotor-arm system.

Joint	θ	d	a	α
1	q_1	-0.002	0.004	$-\pi/2$
2	q_2	0	0.149	0
3	$q_3 + \pi$	0.004	0.085	π
4	$q_4 + \pi/2$	0	0	$-\pi/2$
5	$q_5 - \pi/2$	0	0	0

TABLE I

DENAVIT-HARTENBERG PARAMETERS OF THE 5DOF ARM.

where $\overline{q}_{ai}$ and $\underline{q}_{ai}$ are the high and low joint limit values, respectively, for the i-th link . Rearranging Eq. 23, the proposed task function becomes

$$\sigma_L = (\mathbf{q}_a - \mathbf{q}_a^*)^T \mathbf{\Lambda}_L (\mathbf{q}_a - \mathbf{q}_a^*), \qquad (24)$$

where $\mathbf{\Lambda}_L$ is a diagonal matrix whose diagonal elements are equal to the inverse of the squared joint limit ranges

$$\mathbf{\Lambda}_L = \lambda_L \operatorname{diag}\{(\overline{q}_{a1} - \underline{q}_{a1})^{-2}, \ldots, (\overline{q}_{an} - \underline{q}_{an})^{-2}\}, \quad (25)$$

and λ_L is a suitable positive gain. The desired task variable is $\sigma_L^* = 0$ (i.e. $\tilde{\sigma}_L = -\sigma_L$), and the corresponding task Jacobian is

$$\mathbf{J}_L = \begin{bmatrix} \mathbf{0}_{1 \times 4} & -2\lambda_L (\mathbf{\Lambda}_L (\mathbf{q}_a - \mathbf{q}_a^*))^T \end{bmatrix}. \qquad (26)$$

A reasonable choice of $\mathbf{q}_a^*$ is to consider the configuration of maximum manipulability [22], which could be evaluated with the Jacobian from Eq. 1 with $w = \sqrt{|\mathbf{J}_{qa} \mathbf{J}_{qa}^T|}$. We have searched for such configuration discretizing all possible arm joint positions and quadrotor inclinations (with ϕ_q and θ_q between the ranges of $[-pi/2, pi/2]$). Unfortunately, in our particular application, the configuration of largest manipulability leads to a configuration with structural self occlusion of the robot body onto the camera frame. In our simulations we have chosen instead desired configurations where the camera maximizes its field of view, i.e. below the quadrotor in either the front or rear parts of the robot.

IV. SIMULATIONS

By attaching a 5DOF arm to the quadrotor we end up with a 9 DOF overactuated system. The robotic arm was designed with a joint setting to compensate the possible noise existing in the quadrotor positioning, i.e. to maintain the target in a fixed position with respect to the camera frame whilst the quadrotor is hovering still but subject to external perturbations. We present now simulations in ROS for a dynamical model (i.e. a modification of the Hector quadrotor stack [23] with the Asctec Pelican parameters) using the Gazebo simulator. The arm Denavit-Hartenberg parameters are given in Table I.

The quadrotor-arm system is teleoperated to an initial position as shown in Fig. 2 (a-b). At that point, the visual servo method is switched on, together with the secondary tasks.

Fig. 2 (c-d) show the final robot configuration and how the arm changed its joint values to reach the desired camera location. To differentiate the proposed system versus pure servoing to a landing target, the marker for the manipulation task is oriented in a pose that would not be reachable without compromising flight stability, had the system not included an arm.

We first simulate a servo scheme to show the effect of the proposed secondary tasks. Fig. 3 shows the camera pose error during the servoing task (i.e. $e_i = \|c_i^* - c_i\|_2$ with c_i^* and c_i the individual desired and current pose terms $i = [x, y, z, \phi, \theta, \psi]$). Note the difference in time scale. Whilst both servo schemes reach the target, the use of the secondary tasks produces a more efficient controller. This is due to the undesired torque added to the quadrotor when the weight distribution of the arm is not aligned along the quadrotor CoG. By the addition of the secondary tasks, this torque is reduced as shown in Fig. 3(c) during the servoing task. So, flight stability can be enhanced by the correct parametrization of the sub-tasks and therefore the time to reach the target.

A series of simulations were carried out in order to select an appropriate value for λ_G in Eq. 17. The experiment was repeated for varying values of λ_G, and the distance from the arm CoG to the vertical axis was plotted in Fig. 4(a)[1]. Notice how the error decreases rapidly for values of λ_G larger than 0.1. Some overshooting is observed however in these cases due to high velocities acquired by some of the joints. Even when we have added saturation on the joint actuators to avoid velocities larger than $1 rad/s$ in the arm joints, we set $\lambda_G = 0.1$. For the chosen parameter value, now the distance from the arm CoG to the quadrotor vertical axis is smoothly reduced when the secondary task is activated, as shown in Fig. 4(b).

In order to select the value of λ_L for the third task, we have first chosen a desired arm configuration consistent with the arm initial conditions and the other first two tasks. This is, we initialize the system and let it run with the other two tasks active, and once it stabilizes, we recorded the final configuration into $\mathbf{q}_a^*$. Then, we run the experiment including the third

---

[1]In the plot, simulations end at different times because the stopping condition was not set in terms of time of execution but in percentage of error decreased.
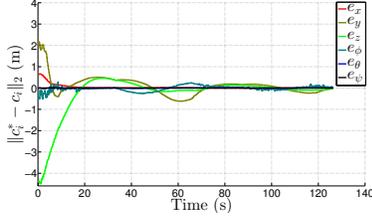
(a) Camera position and orientation error enforcing only the primary task.
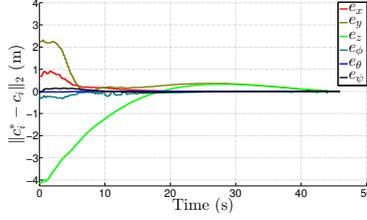
(b) Camera position and orientation error enforcing primary and secondary tasks.
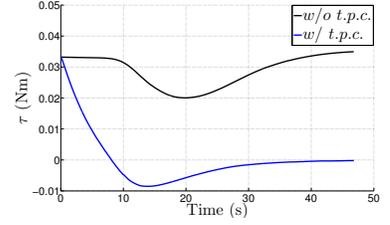
(c) Arm torque with and without enforcing the secondary tasks.

Fig. 3.   Visual servoing task with and without task priority control.
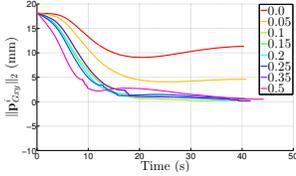
(a) Distance from the arm CoG to the quadrotor vertical axis for different values of λ_G.
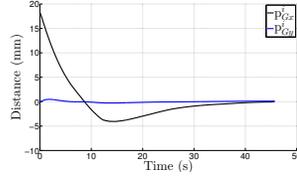
(b) Distance along each axis for a chosen value of $\lambda_G = 0.1$

Fig. 4.   CoG alignement.
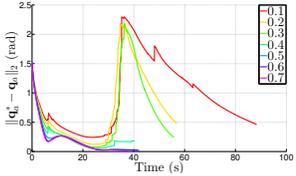
(a) Total error between the current and the desired arm joint configurations for different values of λ_L.
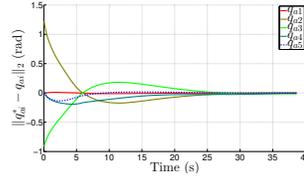
(b) Individual joint errors for a fixed value of $\lambda_L = 0.7$.

Fig. 5.   Error between desired and the current arm joint values.

	Time to target (s)	
Method	μ_t	$std(t)$
Weighted sum	85.3979	46.5853
Hierarchical	50.9624	21.4597

TABLE II
TIME TO COMPLETION STATISTICS FOR MULTIPLE REALIZATIONS OF THE EXPERIMENT UNDER VARYING INITIAL CONDITIONS FOR THE TWO METHODS: WEIGHTED SUM CONTROL, AND HIERARCHICAL CONTROL.

task, and for varying values of σ_L. The results are plotted in Fig. 5(a)[1]. Notice how for values of $\lambda_L < 0.4$, the third task cannot be achieved. This is because the secondary task has preference in driving the arm to a singular configuration with the arm either completely contracted or stretched (which also fulfill the CoG condition). We shall prevent the system from reaching such singular configurations, thus a larger value of σ_L is preferred. Given that this task is exerted at the lowest level in the hierarchy, the system is less sensitive to overshooting in this case. We have chosen a value of $\lambda_L = 0.7$ (see Fig. 5(b)).

Fig. 6 shows the resulting contributed velocities from each task to the total camera velocity, expressed in the camera frame, during servoing when the hierarchical task priority control is enabled. Notice how in all cases, the velocities are decreased towards zero while the individual goal of each task is achieved.

The control law proposed in [17] 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 can not satisfy accurately none of the subtasks requirements, e.g. when one task tries to reach a singular arm configuration with the arm CoG vertical aligned with the quadrotor gravitational vector, whilst another task is driving the robot away from such singularity and hence, from CoG alignment. The resultant

velocity from the sum of the two would drive the camera to a pose that does not satisfy either of the two tasks and still the weighted sum would be reduced to zero. In contrast, the control law presented here takes into account the priority of each task. That is, the desired arm configuration of the third task will be only fulfilled whilst the arm and quadrotor CoGs are aligned vertically. This secondary task in turn will also be consistent with the higher priority task corresponding to the visual servo.

We have performed an extensive experimentation with the two strategies for varying initial conditions and final desired configurations. The results are shown in Fig. 7 and also in Table 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. Under these conditions, the time to completion varied significantly between using and not using the hierarchical scheme proposed. We can conclude that the proposed method reaches task completion sooner mainly because it prioritizes quadrotor stability through CoG alignment over the third task, thus avoiding antagonistic task behaviors.

V. CONCLUSIONS

We have presented a task priority control scheme for aerial vehicles equipped with a manipulator for aerial surveillance suitable for search and rescue missions. A serial arm is attached to the base of a quadrotor, and a camera is fixed at its end-effector. A primary task is designed to respond to visual servo control commands. A secondary task is developed to address the torques exerted on the quadrotor through the actuation of the arm when the overall CoG of the system is modified. A third task is provided to avoid arm singularities and to increase manipulability by setting the hand to a desired configuration. The presented control law takes into account the hierarchy of the tasks by projecting each one into the Jacobian null space of the previous one. The technique is demonstrated in a ROS implementation.

We can think of two avenues for further research. On the
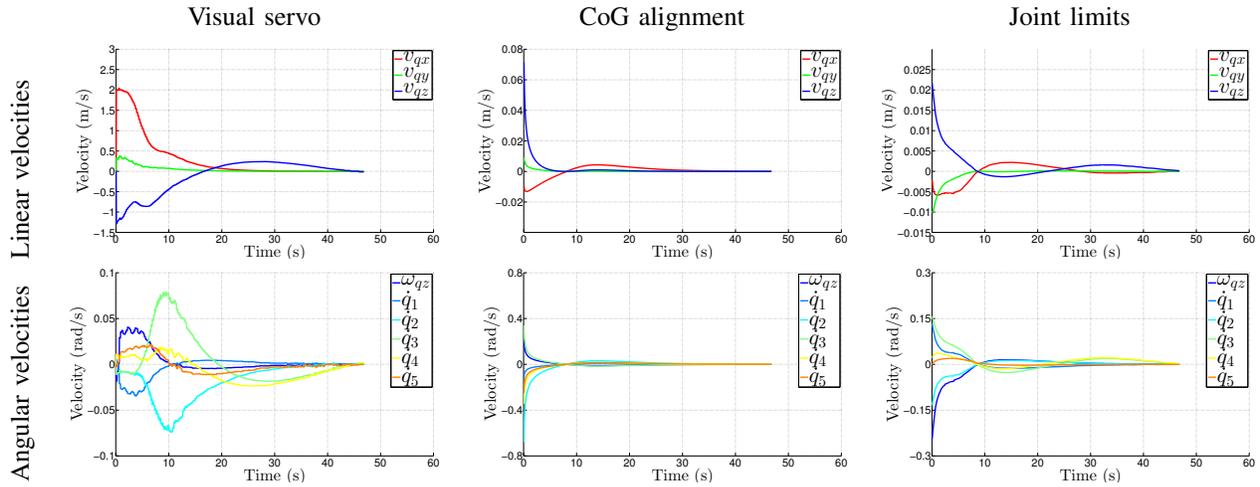
Fig. 6. Actuator velocities corresponding to three hierarchical subtasks, visual servoing, CoG alignement, and joint limits.
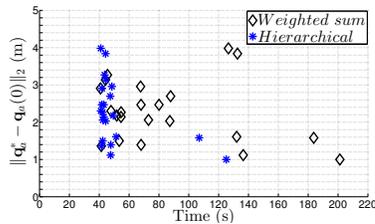
Fig. 7. Comparison between weighted sum and hierarchical control laws during servoing tasks, considering the time to reach the target (horizontal axis), with a final linear and angular euclidean distances lower than $5cm$ and $0.026rad$ respectively, and for varying initial conditions (vertical axis).

one hand, the approach is demonstrated in a simulated case study, thus the next natural step is to test it in a real robot setting. Secondly, the true dynamic behavior of the UAV has not been taken into account in the design of the control law, and specifically, in the low-level attitude controller. To avoid some of these effects we have added the reduction of the distance between the center of gravity of the arm and the quadrotor gravitational vector, but some other effects could also be considered. For instance, the inertia added by the arm movement could be taken into account in designing the low level attitude controller of the quadrotor. We leave this problem for future research.

REFERENCES

[1] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. Cooper, M. Quigley, J. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *J. Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.

[2] P. Doherty and P. Rudol, *A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4830.

[3] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," in *Proc. Int. Conf. Em. Security Tech.*, Alcal de Henares, Sep. 2010, pp. 142–147.

[4] 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," *Robotics Autom. Mag.*, vol. 19, no. 3, pp. 46–56, 2012.

[5] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle," *J. Int. Robotic Syst.*, vol. 69, no. 1-4, pp. 227–240, 2013.

[6] 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.

[7] C. Korpela, M. Orsag, T. Danko, B. Kobe, C. McNeil, R. Pisch, and P. Oh, "Flight stability in aerial redundant manipulators," in *Proc. IEEE Int. Conf. Robotics Autom.*, Saint Paul, May. 2012, pp. 3529–3530.

[8] 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.

[9] G. Antonelli, E. Cataldi, P. Giordano, S. Chiaverini, and A. Franchi, "Experimental validation of a new adaptive control scheme for quadrotors MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Nov. 2013, pp. 2439–2444.

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

[11] 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.

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

[13] ——, "Visual servo control. II. Advanced approaches," *Robotics Autom. Mag.*, vol. 14, no. 1, pp. 109–118, 2007.

[14] 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.

[15] T. Hamel and R. Mahony, "Image based visual servo control for a class of aerial robotic systems," *Automatica*, vol. 43, no. 11, pp. 1975–1983, 2007.

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

[17] V. Lippiello, R. Mebarki, and F. Ruggiero, "Visual coordinated landing of a UAV on a mobile robot manipulator," in *Proc. 11th IEEE Int. Symp. Safe. Sec. Resc. Robotics.*, Linkoping, Oct. 2013, pp. 1–7.

[18] 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.

[19] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990.

[20] 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.

[21] R. Boulic, R. Mas, and D. Thalmann, "A robust approach for the control of the center of mass with inverse kinetics," *Comp. and Graph.*, vol. 20, no. 5, pp. 693–701, 1996.

[22] T. Yoshikawa, "Manipulability and redundancy control of robotic mechanisms," in *Proc. IEEE Int. Conf. Robotics Autom.*, vol. 2, St. Louis, Mar. 1985, pp. 1004–1009.

[23] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, "Comprehensive simulation of quadrotor UAVs using ROS and Gazebo," in *Proc. 3rd Int. Conf. Simul. Model. Program. Auton. Robots*, Tsukuba, Nov. 2012, pp. 400–411.