

DSCC2013-3760

EXPLOITING IMAGE MOMENTS FOR AERIAL MANIPULATION CONTROL

Rafik Mebarki

PRISMA Lab, DIETI

University of Naples Federico II

Via Claudio 21, 80125 Naples, Italy

Email: rafik.mebarki@unina.it

Vincenzo Lippiello

Bruno Siciliano

PRISMA Lab, DIETI

University of Naples Federico II

Via Claudio 21, 80125 Naples, Italy

Email: vincenzo.lippiello@unina.it

bruno.siciliano@unina.it

ABSTRACT

This paper proposes a new visual servo control scheme that endows flying manipulators with the capability of positioning with respect to visual targets. A camera attached to the UAV provides real-time images of the scene. We consider the approaching part of an aerial assembling task, where the manipulator carries a structure to be plugged into the visual target. In order to augment the system capabilities regarding the 3D interaction with the target, we propose to use image moments. The developed controller generates desired velocities to both the UAV and the manipulator, simultaneously. While taking into account the under-actuation specific to rotary-wing vehicles, it makes use of the system redundancy to realize potential sub-tasks. The joints limits avoidance is also guaranteed. The presented developments are validated by means of computer simulations.

1 INTRODUCTION

The last decade is witnessing an increasing interest in Unmanned Aerial Vehicles (UAVs). Such systems afford remotely accessing regions of interest and thus envisaging numerous applications, as for instance operating in hazardous and difficult-to-access environments. Particularly, rotary-wing vehicles possess the capability of holding stationary and therefore passing through cumbersome spaces. This makes them relevant candidates for applications such as sensing and surveillance, search and rescue, load transportation [1, 2], probe-based inspection [3, 4], and, freshly, aerial manipulation [5], to name but a few.

The environment needs to be sensed so the control system can deliver the necessary guidance for the vehicle to achieve the task. In addition, for the vehicle to be autonomous, sensors are

mounted on-board. Commonly adopted sensing modalities are the Global Positioning System (GPS) and vision. Besides of being limited to collect position measures, GPS suffers from signal loss at low altitudes and in urban or cluttered areas [6, 7], low resolution, and is not passive. Vision, on the other hand, provides rich information with pixel order resolution, is passive and cheap [8, 9].

The present work considers aerial manipulation guided with vision. The robotic system consists of a rotary-wing UAV carrying a robotic arm, which at its turn actuates a structure or an assembly part. The objective consists in automatically positioning the structure on a desired visual target, such that the camera attached to the UAV observes the scene (see Fig. 1). The presented robotic system takes advantage of the UAV's mobility to reach the target from far location, while the manipulator dexterity can be exploited at the vicinity. In this paper, we present a new image-based visual servo scheme to achieve the positioning task automatically. It simultaneously controls the UAV and the manipulator at the velocity level, while at the same time it fully exploits the highlighted system's mobility and dexterity.

Just like mobile manipulation is broadening robotics operational space, aerial manipulation indeed is free from ground contact and thus is less constrained, and hence allows envisaging countless applications. Yet, having been extensively investigated, the field of mobile manipulation can inspire the younger field of aerial manipulation. In [10], a visual servo scheme is proposed to automatically position a camera actuated by a mobile manipulator in such a way that observed feature points converge to desired locations in the image. The manipulator joint limits are considered by using the Jacobian null space, but the case of large initial errors is not specifically addressed.

Somewhat analogous to the present work, [11] considers the simultaneous control (at the velocity level) of a mobile platform and the manipulator for an object grasping task. An eye-in-hand camera held by the arm provides the feedback information. Joints limits avoidance is guaranteed by adopting a weighted pseudo inverse [12] that blocks the joint reaching its limits.

Similarly to our configuration, in [13] a camera is mounted directly on the mobile platform. The grasping task is decomposed into two steps, which involve different visual servoing schemes insuring their accomplishment. During the first step, the mobile platform is solely controlled and dragged to the vicinity of the target. The corresponding visual servo scheme is augmented with reinforcement learning in order to keep the target in the camera field of view. Then, the manipulator is solely visually controlled for grasping.

In [14], a humanoid robot is controlled for ball catching, where the task is decomposed into different steps with different priorities. The subtasks then need to be realized successively following the framework presented in [15, 16]. The humanoid is first assigned to move through a line that passes near the target. Once the humanoid is in target vicinity, its hand is visually controlled to catch the ball such that a camera mounted on its head relays the feedback images.

In the present work, similarly to [13, 14], a new camera configuration class can be defined. In fact, in the field of visual servoing two main classes are so far considered: *eye-to-hand* and most notoriously *eye-in-hand* configurations. In the former, the camera is observing the robot to control, while in the latter configuration the camera is actuated by the robot. However, in the present configuration the camera is mounted on the UAV, and in fact it observes the manipulator as well. As such, inspired by the above taxonomy of visual servoing schemes, we propose to define this new configuration as “*onboard-eye-to-hand*” to mean that the camera is on-board the UAV (onboard-eye) while observing the robot manipulator (eye-to-hand).

In this paper we present a new visual servo scheme (see Section 2) relevant for this new configuration class defined above. In order to improve the system capability of interacting with the target, we adopt image moments [17]. Thanks to their intuitive and geometrical meaning, they can allow for instance performing motion through a line roughly orthogonal to the target during the last approaching phase towards the target (pre-manipulation). This allows avoiding undesirable/unpredictable Cartesian motions that could not be borne in practice, that is a common drawback of many image control schemes (see Section 3). Both the UAV and manipulator are simultaneously controlled at the velocity level. The proposed controller takes advantage of the whole system redundancy, while at the same time it considers the peculiarity of the under-actuation related to rotary-wing UAVs. By including a weighting matrix into the control law, both the mobility of the UAV and the dexterity of the manipulator are exploited. Manipulator joints limits avoidance is guaranteed in addition (see Section 4). Finally, in Section 5 results achieved from computer simulations show the validity of the proposed method.

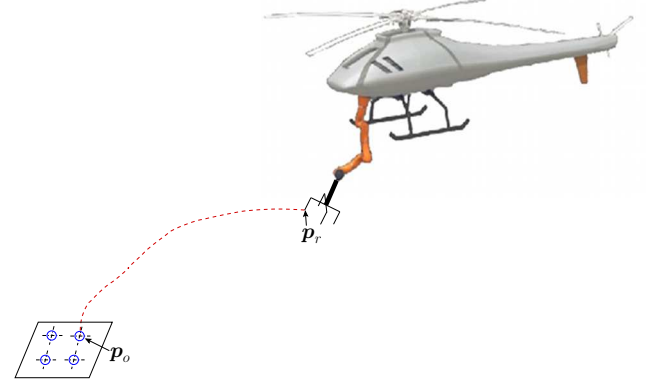


Figure 1. HELICOPTER EQUIPPED WITH A ROBOT MANIPULATOR HOLDING AN ASSEMBLY PART (STRUCTURE). THE OBJECTIVE IS TO AUTOMATICALLY POSITION THE LATTER ON THE VISUAL TARGET. THE PICTURE OF THE HELICOPTER/ARM SYSTEM IS REFERRED TO THE EUROPEAN ARCAS PROJECT (WWW.ARCAS-PROJECT.EU).

2 KINEMATICS MODELING

Consider a helicopter carrying an n degree-of-freedom (DOF) robot manipulator, that is actuating a structure (see Fig. 1). An onboard camera attached to the vehicle provides real-time images of the target object. The desired task consists in automatically positioning the actuated structure on the target location. More precisely, let p_r 's and p_o 's be set of (fiducial) points of the carried platform and of the object, respectively (see Figs. 1 and 2). The objective is to position each point p_r on its corresponding target p_o in an autonomous way by using only the real-time images of the target to guide the robotic system. This can be achieved by means of a new controller we describe in the following.

2.1 TARGET IMAGE KINEMATICS

Let $\{R_s\}$ be the camera reference frame, with its X and Y axes within the image plane. Without loss of generality, we assume that it also corresponds to the helicopter body frame (see Fig. 2). Let $s_o = (x_o \ y_o)^T \in \mathbb{R}^2$ be the perspective image coordinates of point p_o , whose coordinates in $\{R_s\}$ are ${}^s t_o = (X_o \ Y_o \ Z_o)^T \in \mathbb{R}^3$ (see Fig. 2). The UAV velocity with respect to a fixed inertial frame and expressed in camera sensor frame $\{R_s\}$ is denoted $\mathbf{v} = (\mathbf{v}^T \ \boldsymbol{\omega}^T)^T \in \mathbb{R}^6$, where $\mathbf{v} = (v_x \ v_y \ v_z)^T$ and $\boldsymbol{\omega} = (\omega_x \ \omega_y \ \omega_z)^T \in \mathbb{R}^3$ represent the translational and rotational velocities, respectively (see Fig. 2). Assuming the target motionless, time variation $\dot{s}_o$ of s_o can be expressed as a function the vehicle velocity as follows:

$$\dot{s}_o = L_o \mathbf{v}, \quad (1)$$

where $L_o \in \mathbb{R}^{2 \times 6}$ is the so-called interaction matrix [18]

$$L_o = \begin{pmatrix} -\frac{1}{Z_o} & 0 & x_o/Z_o & x_o y_o & -1 - x_o^2 & y_o \\ 0 & -\frac{1}{Z_o} & y_o/Z_o & 1 + y_o^2 & -x_o y_o & -x_o \end{pmatrix}. \quad (2)$$

Notice that L_o depends on coordinates (x_o, y_o) available from the observed image, and also on depth information Z_o . In the field of *visual servoing*, Z_o is usually fixed to the value at the desired configuration.

2.2 CARRIED POINTS KINEMATICS

In this work only the target is assumed to be observed by the camera. Let ${}^s t_r = (X_r \ Y_r \ Z_r)^\top \in \mathbb{R}^3$ be the translational vector from $\{R_s\}$ origin to the point p_r expressed in $\{R_s\}$. Let $s_r = (x_r \ y_r)^\top \in \mathbb{R}^2$ be virtual image coordinates of p_r . Since we are considering a perspective projection, s_r can be expressed as a function of ${}^s t_r$ as follows:

$$x_r = \frac{X_r}{Z_r}, \quad y_r = \frac{Y_r}{Z_r}. \quad (3)$$

Time differentiating (3) yields

$$\dot{s}_r = \bar{L}_r {}^s \dot{t}_r, \quad \bar{L}_r = \begin{pmatrix} 1/Z_r & 0 & -x_c/Z_r \\ 0 & 1/Z_r & -y_c/Z_r \end{pmatrix} \in \mathbb{R}^{2 \times 3} \quad (4)$$

such that $\dot{s}_r$ and ${}^s \dot{t}_r$ are the time variations of s_r and ${}^s t_r$, respectively.

Let $\{R_u\}$ be the robot manipulator base frame. Without loss of generality, we assume its origin coincident with that of $\{R_s\}$. Rotation matrix ${}^s R_u$ defines the orientation of the latter with respect to the former. It is constant and a priori known. Also, let $\{R_e\}$ be the end-effector reference frame. Thus, ${}^s t_r$ can be expressed as follows:

$${}^s t_r = {}^s R_e {}^e t_r + {}^s t_e = {}^s R_u ({}^u R_e {}^e t_r + {}^u t_e), \quad (5)$$

where ${}^e t_r$ corresponds to the 3D coordinates of p_r in $\{R_e\}$, which is constant and a priori-known (see Fig. 2). The pair $({}^s t_e, {}^s R_e)$ represents the 3D pose (position and orientation, respectively) of $\{R_e\}$ with respect to $\{R_s\}$, while $({}^u t_e, {}^u R_e)$ the pose of the end-effector attached frame with respect to manipulator base frame $\{R_u\}$. Accordingly, time differentiating (5) yields

$${}^s \dot{t}_r = {}^s R_u J_r {}^u \dot{P}_e, \quad J_r = (I_3 - [{}^u R_e {}^e t_r]_\times) \in \mathbb{R}^{3 \times 6}, \quad (6)$$

such that $[\cdot]_\times$ denotes the skew-symmetric matrix, and where I_3 corresponds to the identity matrix of size 3. More particularly,

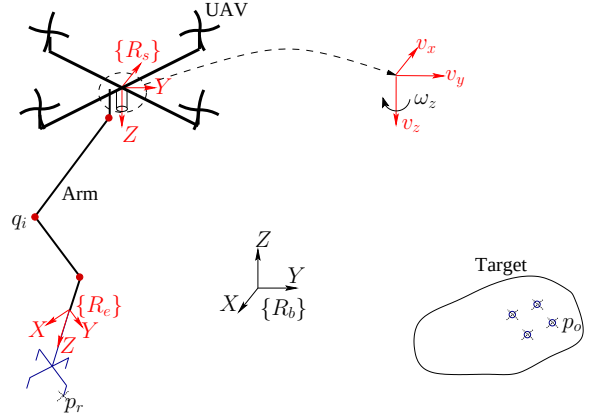


Figure 2. SKETCH OF THE ROBOTIC SYSTEM AND TARGET, AS WELL AS INVOLVED CARTESIAN FRAMES AND VARIABLES.

${}^u \dot{P}_e = ({}^u \dot{t}_e^\top \ {}^u \omega_e^\top) \in \mathbb{R}^6$, such that ${}^u \dot{t}_e$ and ${}^u \omega_e$ represent the translational and rotational velocity of $\{R_e\}$ with respect to $\{R_u\}$, respectively. This velocity can in fact be expressed as a function of joints velocity $\dot{q} \in \mathbb{R}^n$ of the manipulator through its Jacobian matrix J_e by ${}^u \dot{P}_e = J_e(q) \dot{q}$. Finally, plugging the latter into (6), and the result into (4), we obtain

$$\dot{s}_r = L_r \dot{q}, \quad L_r = \bar{L}_r {}^s R_u J_r J_e(q) \in \mathbb{R}^{2 \times n}. \quad (7)$$

Notice that L_r is a function of the virtual image coordinates (x_r, y_r) , depth Z_r , orientation ${}^u R_e$ of the end-effector with respect to the robot base frame, manipulator joint vector q , and known constants ${}^e t_r$ and ${}^s R_u$. To summarize, L_r depends only on the manipulator odometry.

In order to achieve the desired task, we propose in [19] to nullify the following error:

$$e_c = s_r - s_o \in \mathbb{R}^2 \quad (8)$$

for each pair $(p_{ri}, p_{oi})_{i=1, \dots, m}$, where m is the number of target points. This error characterizes a new *image-based* visual servoing paradigm. Indeed, similarly to classical visual servoing where s_o is the real image information of the target, while s_r on the other hand represents virtual image features of the actuated structure. Moreover, both these two informations vary in the image to finally superimpose for the error to be null. So far in the field of visual servoing, only the features of the target vary in order to superimpose on constant prior learnt image features. Notice that with this new paradigm we present, there is no need to learn the target image. We refer to this paradigm as SVS, standing for *Self Visual Servoing* to mean that the visual servoing scheme “inherently seeks” (self) for the superimposition of the image features of the observed target with those of the actuated object to position.

Differently from [19], the current objective is that the structure (or assembly part) approaches the target with a desired be-

havior, e.g. the structure moves perpendicularly to the target before it reaches it. However it does not seem evident to achieve this goal by directly employing image coordinates errors, as (8). To deal with this and thus achieve our objectives, we propose the adoption of image moments [17] because of their geometrical interpretation.

3 IMAGE MOMENTS

Low order image moments provide physical interpretation, as the area, center of gravity, and the orientation of an observed object in the image. This shows to be of great interest since it allows intuitively selecting a set of visual features yielding the robotic system decoupled. Therefore the objective stated above can for instance be achieved by forcing the area error to be the last converging to zero. This is more detailed in Section 3.3.

In [20] [21] image moments are employed as feedback visual features for *eye-in-hand* configuration. The interaction matrix (Jacobian) relating the time variation of such features to the robotic system velocity is analytical derived. It is assumed that the surface of the target object is a priori known. That constraint is relaxed in the present paper, where we calculate the exact interaction matrix.

3.1 DEFINITION

Let S be the image section of an observed object (see Fig. 3). The corresponding image moment of order $p + q$ is given by [17]

$$m_{pq} = \iint_S x^p y^q dx dy \quad (9)$$

In [22], a relationship to calculate the image moments for polygonal shapes using their contour points is provided. Let $s_{i|i=0,\bar{m}} = (x_i, y_i)$ be image points characterizing the contour of S (see Fig. 3), where (x_i, y_i) are their image coordinates. The points are arranged in the counter-clockwise sense, where $s_m = s_0$ to close the contour. Moment m_{pq} is thus expressed [22]

$$m_{pq} = \sum_{i=1}^m (y_i - y_{i-1}) \sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} x_i^k x_{i-1}^{p+1-k} y_i^j y_{i-1}^{q-j} \quad (10)$$

$$\text{with } a_{kj} = \frac{1}{(p+q+2)(p+1)} \frac{\binom{p+1}{k} \binom{q}{j}}{\binom{p+q+1}{k+j}},$$

where $\binom{\cdot}{\cdot}$ denotes the binomial coefficient. For calculus convenience we re-write (10) as

$$m_{pq} = \sum_{i=1}^m (y_i - y_{i-1}) \sigma_{i,pq}, \quad (11)$$

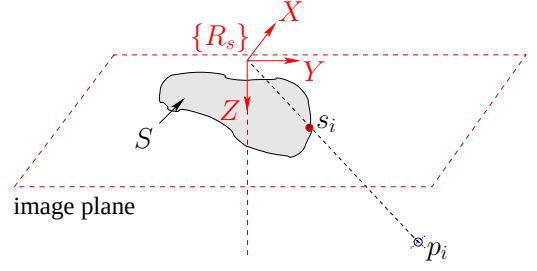


Figure 3. OBSERVED CAMERA IMAGE. BOTH POINT s_i AND SECTION S LIE IN THE IMAGE PLANE.

with

$$\begin{cases} \sigma_{i,pq} = \sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} f_x f_y \\ f_x = x_i^k x_{i-1}^{p+1-k}, \quad f_y = y_i^j y_{i-1}^{q-j} \end{cases} \quad (12)$$

In the following, time variation $\dot{m}_{pq}$ of moment m_{pq} is expressed as a function of the robotic system velocity. Hence this will allow us to design a controller at the velocity-level with image moments as feedback visual features.

3.2 INTERACTION MATRIX RELATIVE TO IMAGE MOMENTS

Time differentiating (11) yields

$$\dot{m}_{pq} = \sum_{i=1}^m [(\dot{y}_i - \dot{y}_{i-1}) \sigma_{i,pq} + (y_i - y_{i-1}) \dot{\sigma}_{i,pq}], \quad (13)$$

with

$$\begin{cases} \dot{\sigma}_{i,pq} = \sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} (\dot{f}_x f_y + f_x \dot{f}_y) \\ \dot{f}_x = k x_i^{k-1} x_{i-1}^{p+1-k} \dot{x}_i + (p+1-k) x_i^k x_{i-1}^{p-k} \dot{x}_{i-1} \\ \dot{f}_y = j y_i^{j-1} y_{i-1}^{q-j} \dot{y}_i + (q-j) y_i^j y_{i-1}^{q-j-1} \dot{y}_{i-1} \end{cases} \quad (14)$$

Note that $z^\beta = 0$ for $\beta < 0$ is considered. Taking back the expression of s_j introduced above in Section 3.1, and using results (1) and (7), we write for conciseness

$$\dot{s}_j = \begin{pmatrix} \dot{x}_j \\ \dot{y}_j \end{pmatrix} = \begin{pmatrix} L_{x_j} \\ L_{y_j} \end{pmatrix} \zeta, \quad (15)$$

where L_{x_j} and $L_{y_j} \in \mathbb{R}^{1 \times \hat{n}}$ and $\zeta \in \mathbb{R}^{\hat{n}}$, such that

$$(\dot{\zeta}, \dot{n}) = \begin{cases} (\dot{q}, n) & \text{for } s = s_r \\ (\mathbf{v}, 6) & \text{for } s = s_o \end{cases} \quad (16)$$

Adopting (15), $\dot{f}_x$ and $\dot{f}_y$ of (14) are expressed as follows:

$$\begin{cases} \dot{f}_x = kx_i^{k-1}x_{i-1}^{p+1-k}L_{x_i}\dot{\zeta} + (p+1-k)x_i^kx_{i-1}^{p-k}L_{x_{i-1}}\dot{\zeta} \\ \dot{f}_y = jy_i^{j-1}y_{i-1}^{q-j}L_{y_i}\dot{\zeta} + (q-j)y_i^jy_{i-1}^{q-j-1}L_{y_{i-1}}\dot{\zeta} \end{cases}, \quad (17)$$

and thus we get

$$\dot{f}_x = L_{f_x}\dot{\zeta}, \quad \dot{f}_y = L_{f_y}\dot{\zeta}, \quad (18)$$

with

$$\begin{cases} L_{f_x} = kx_i^{k-1}x_{i-1}^{p+1-k}L_{x_i} + (p+1-k)x_i^kx_{i-1}^{p-k}L_{x_{i-1}} \\ L_{f_y} = jy_i^{j-1}y_{i-1}^{q-j}L_{y_i} + (q-j)y_i^jy_{i-1}^{q-j-1}L_{y_{i-1}} \end{cases}. \quad (19)$$

Similarly $\dot{\sigma}_{i,pq}$ of (14) is derived as follows:

$$\begin{aligned} \dot{\sigma}_{i,pq} &= \sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} (f_y L_{f_x} \dot{\zeta} + f_x L_{f_y} \dot{\zeta}) \\ &= \sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} (f_y L_{f_x} + f_x L_{f_y}) \dot{\zeta} \\ &= \left[\sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} (f_y L_{f_x} + f_x L_{f_y}) \right] \dot{\zeta} \\ &= L_{\sigma_{i,pq}} \dot{\zeta}, \end{aligned} \quad (20)$$

such that

$$L_{\sigma_{i,pq}} = \sum_{k=0}^{p+1} \sum_{j=0}^q a_{kj} (f_y L_{f_x} + f_x L_{f_y}) \in \mathbb{R}^{1 \times \hat{n}}. \quad (21)$$

Finally, coming back to the expression of $\dot{m}_{pq}$ given by (13), plugging those of $\dot{\sigma}_{i,pq}$ and $\dot{y}_j$ given by (20) and (15), respectively, we obtain

$$\begin{aligned} \dot{m}_{pq} &= \sum_{i=1}^m \left[\sigma_{i,pq} (L_{y_i} - L_{y_{i-1}}) \dot{\zeta} + (y_i - y_{i-1}) L_{\sigma_{i,pq}} \dot{\zeta} \right] \\ &= \left\{ \sum_{i=1}^n [\sigma_{i,pq} (L_{y_i} - L_{y_{i-1}}) + (y_i - y_{i-1}) L_{\sigma_{i,pq}}] \right\} \dot{\zeta} \\ &= L_{m_{pq}} \dot{\zeta}, \end{aligned} \quad (22)$$

with

$$L_{m_{pq}} = \sum_{i=1}^m [\sigma_{i,pq} (L_{y_i} - L_{y_{i-1}}) + (y_i - y_{i-1}) L_{\sigma_{i,pq}}] \in \mathbb{R}^{1 \times \hat{n}}. \quad (23)$$

Notice that $L_{m_{pq}}$ is obtained for both the target image and the actuated structure by replacing (15) with (1) and (7), respectively.

3.3 VISUAL FEATURES BASED ON IMAGE MOMENTS

As highlighted earlier, image moments afford visual features with physical interpretation. Accordingly, in order to yield a decoupled robotic system, the following set of visual features is proposed

$$s_m = (\sqrt{a} \ x_g \ y_g \ \eta_{11} \ I_1 \ \sqrt{I_2})^\top \in \mathbb{R}^6, \quad (24)$$

where a is the area of the observed section S in the image, (x_g, y_g) its gravity center, η_{11} second order moment invariant to translation and scale, and (I_1, I_2) are moments invariants to both area, translation, and orientation. They are given as a function of image moment m_{pq} as follows [17]

$$\begin{cases} a = m_{00}, \quad x_g = m_{10}/a, \quad y_g = m_{01}/a \\ I_1 = \eta_{20} + \eta_{02}, \quad I_2 = 4\eta_{11}^2 + (\eta_{20} - \eta_{02})^2 \end{cases}, \quad (25)$$

with μ_{pq} the centered image moments of order $p+q$ and η_{pq} the moments invariants to both translations and scale. They are calculated as follows [17]:

$$\begin{cases} \mu_{pq} = \iint_S (x - x_g)^p (y - y_g)^q dx dy \\ \eta_{pq} = \mu_{pq} / a^{(1 + \frac{p+q}{2})} \end{cases}. \quad (26)$$

As one can notice, μ_{pq} can easily be expressed as a function of image moments m_{ij} 's up to the $(p+q)$ -th order. Notice that in (24) $\sqrt{a}$ and $\sqrt{I_2}$ are employed instead of a and I_2 , respectively, in order to have unit consistency. Finally, we can also notice that s as defined by (24) endows decoupling properties. Indeed, area a and thus $\sqrt{a}$ are mainly affected by motions orthogonal to the camera plane, while x_g, y_g are affected by the motion parallel to the image plane. As for η_{11} it is mainly related to the section's main angle α , that strongly depends on the rotation around the image plane. Finally, I_1 and I_2 are invariant to both scale, translation, and rotation in the image. Thus, they should be affected mainly by the remaining motions: the rotations around image X- and Y-axis. Remark that in fact α has been replaced by η_{11} since we noticed that the former induces shakiness in the system behavior.

4 CONTROL LAW

Let S_o be the image section whose contour is formed by the image points $so_{i=1,\dots,m}$ of the target. Feature s_{mo} corresponds to the moments vector computed on S_o according to (24). Similarly, section S_r 's contour is formed by virtual points $sr_{i=1,\dots,m}$, where s_{mr} is the corresponding moments-feature vector. Accordingly, consider the following visual error

$$e_m = s_{mr} - s_{mo} \in \mathbb{R}^6. \quad (27)$$

Achieving the desired task comes to nullify e_m . A suitable controller is now derived to this aim. Time variations $\dot{s}_{mr}$ and $\dot{s}_{mo}$ of s_{mr} and s_{mo} , respectively, are derived by time differentiating (24) according to (25) and (26). Substituting with result (22) yields

$$\begin{aligned}\dot{s}_{mr} &= L_{mr} \dot{q}, \quad L_{mc} \in \mathbb{R}^{6 \times n} \\ \dot{s}_{mo} &= L_{mo} \mathbf{v}, \quad L_{mo} \in \mathbb{R}^{6 \times 6}.\end{aligned}\quad (28)$$

Then, time variation $\dot{e}_m$ of e_m given by (27) is obtained by plugging the above relationship, which yields

$$\dot{e}_m = \dot{s}_{mr} - \dot{s}_{mo} = L_{mr} \dot{q} - L_{mo} \mathbf{v}. \quad (29)$$

All the joints of the manipulator can be directly controlled in order to achieve the task. Thus, $\dot{q}$ is considered as a control input for our visual control scheme. As for the UAV, it is in fact an under-actuated system, where only four of its DOFs can be independently controlled. Therefore, only $\mathbf{v} = (v_x \ v_y \ v_z)^\top$ and ω_z are added to the visual control inputs. Accordingly, matrix L_{mo} is expressed in this block form $L_{mo} = [L_{mo_v} \ L_{mo_{\omega_x}} \ L_{mo_{\omega_y}} \ L_{mo_{\omega_z}}]$, where each block element is related to $\mathbf{v}$, ω_x , ω_y , and ω_z , respectively. Therefore, (29) is recast as follows:

$$\begin{aligned}\dot{e}_m &= (L_{mr} - L_{mo_v} - L_{mo_{\omega_z}}) \begin{pmatrix} \dot{q} \\ \mathbf{v} \\ \omega_z \end{pmatrix} - (L_{mo_{\omega_x}} \ L_{mo_{\omega_y}}) \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} \\ &= J_m \dot{\xi} - \bar{L}_{m\omega} \bar{\omega},\end{aligned}\quad (30)$$

with $\dot{\xi} = (\dot{q} \ \mathbf{v} \ \omega_z)^\top \in \mathbb{R}^{n+4}$ representing the generalized velocity, $\bar{\omega} = (\omega_x \ \omega_y)^\top \in \mathbb{R}^2$ and $\bar{L}_{m\omega} = (L_{mo_{\omega_x}} \ L_{mo_{\omega_y}}) \in \mathbb{R}^{6 \times 2}$. Vector ξ can thus be exploited as a control input nullifying e_m .

4.1 MAIN TASK: NULLIFYING THE IMAGE MOMENTS ERROR

From (30), the following control law is derived in order to yield e_m exponentially converges to zero according to $\dot{e}_m = -\lambda_1 e_m$ with λ_1 a positive control gain:

$$\dot{\xi} := J_m^+ (-\lambda_1 e_m + \bar{L}_{m\omega} \bar{\omega}) + (I_{n+4} - J_m^+ J_m) \dot{\xi}_2. \quad (31)$$

J_m^+ is a pseudo-inverse of J_m , and $\dot{\xi}_2$ corresponds to system's motions afforded by its redundancy to achieve a secondary task. Notice that $\dot{\xi}_2$ does not affect the main task e_m since it is projected into null space $(I_{n+4} - J_m^+ J_m)$ of Jacobian J_m , where I_{n+4} is the identity matrix of size $n+4$. The described robotic system is intrinsically endowed with the mobility afforded by the UAV, and the manipulability and dexterity afforded by the manipulator. In order to exploit these capabilities, we employ a weighted pseudo-inverse. Matrix J_m^+ thus takes on this form

$$J_m^+ = W J_m^\top (J_m W J_m^\top)^{-1}, \quad (32)$$

where $W \in \mathbb{R}^{(n+4) \times (n+4)}$ represents the weighting matrix. When the UAV is close enough to the target, it gives more weight and thus priority to the manipulator to achieve task e_m . On the other hand, when the UAV is far from the target, more weight is assigned to it. Inspired by [11, 23], we propose the following diagonal weighting matrix

$$W_{ii} = \begin{cases} \tanh \left(\frac{\epsilon_h}{(|\frac{\partial H(q)}{\partial q_i}| + \epsilon_h)} \right) \frac{\lambda_w \epsilon_e}{(e_{\parallel} + \epsilon_e)} & i = \overline{1, n} \\ e_{\parallel} & i = \overline{n+1, n+4}, \end{cases} \quad (33)$$

where W_{ii} is the i -th diagonal element of W , and $e_{\parallel}$ denotes the absolute value of image area error, i.e. the first element of e_m . $H(q)$ is a continuous function with its minimum at the optimal-chosen joints configuration, and it increases to infinity when reaching the joints limits. It is given by [23]

$$H(q) = \frac{1}{4} \sum_{i=1}^n \frac{(q_{iM} - q_{im})^2}{(q_{iM} - q_i)(q_i - q_{im})}, \quad (34)$$

where q_{im} and q_{iM} represent the minimum and maximum limits for i -th joint q_i of q , respectively. We can first notice that when the UAV is far from the target, error $e_{\parallel}$ is large and thus the UAV is prioritized accordingly. The manipulator in contrast is penalized with the inverse $\frac{\epsilon_e}{e_{\parallel} + \epsilon_e}$. Parameter ϵ_e is added to the denominator in order to avoid infinity when visual error is reaching zero. On the other hand, we can note that $\epsilon_h / (|\frac{\partial H(q)}{\partial q_i}| + \epsilon_h)$ blocks a joint reaching its limits since in this case it decreases towards zero. It however increases as soon as the joint moves away from the limit. We employed $\tanh(\cdot)$ function for saturation since we noticed that $H(q)$ can vary to largely different values. The joint is therefore prioritized accordingly. Finally, to summarize, W yields the system exploiting the mobility of the UAV when far from the target, while in contrast the manipulator dexterity is made use when close to the target.

Let us come back to moments-based feedback s_m as defined by (24), and also the features employed for instance in [20]. Assume the targeted object forms a rectangle lying in a plane. We noticed in fact that the system converged well in case the final configuration implies the plane of the camera image parallel to that of the target. In the scenario of a final orientation angle between the two planes, although error e_m converged to zero, we noticed an offset in the 3D space. The issue consists in the fact that for two different images, we obtained equal values of image moments features, i.e. redundancy in the image space with respect to image moments. In order to overcome this problem, we propose to use $\dot{\xi}_2$ involved in (31) to nullify the error on image coordinates, namely e_c introduced by (8). This allows us to resolve the ambiguity caused by image redundancy without altering the objective motivating the use of image moments. In

fact, since $\dot{\xi}_2$ lies in the null space of Jacobian J_m related to image moments, the latter and thus the main task are not affected by it. Therefore, the desired behavior that image moments confer to the system is preserved.

4.2 SECONDARY TASK: NULLIFY THE ERROR ON IMAGE POINTS COORDINATES

Let us take back image points coordinates error $e_{ci} \in \mathbb{R}^2$ given by (8), but with including index i to refer to each of the m pairs $(p_{ri}, p_{oi})_{i=1, \dots, m}$. Time variation $\dot{e}_{ci}$ of e_{ci} can be expressed as a function of generalized velocity $\dot{\xi}$ as follows [19]

$$\dot{e}_{ci} = J_{ci} \dot{\xi} - \bar{L}_{\omega i} \bar{\omega}, \quad (35)$$

with

$$\begin{aligned} J_{ci} &= (L_r \quad -L_{ov} \quad -L_{\omega \omega_x}) \in \mathbb{R}^{2 \times (n+4)} \\ \bar{L}_{\omega i} &= (L_{\omega \omega_x} \quad L_{\omega \omega_y}) \in \mathbb{R}^{2 \times 2}, \end{aligned} \quad (36)$$

such that $L_o = (L_{ov} \quad L_{\omega \omega_x} \quad L_{\omega \omega_y} \quad L_{\omega \omega_z}) \in (\mathbb{R}^{2 \times 3} \quad \mathbb{R}^{2 \times 1} \quad \mathbb{R}^{2 \times 1} \quad \mathbb{R}^{2 \times 1})$. We recall that L_o and L_r are given by (2) and (7), respectively.

Let $e_c = (e_{c1}^\top \quad e_{c2}^\top \quad \dots \quad e_{cm}^\top)^\top \in \mathbb{R}^{2m}$ be the error on image coordinates for the whole m pair of points. The objective is that e_c converges to zero, which consists the secondary task. Using (35), time variation $\dot{e}_c$ of e_c is

$$\dot{e}_c = J_c \dot{\xi} - \bar{L}_\omega \bar{\omega}, \quad (37)$$

with $J_c = [J_{c1}^\top \quad J_{c2}^\top \quad \dots \quad J_{cm}^\top]^\top \in \mathbb{R}^{2m \times (n+4)}$, and $\bar{L}_\omega = [\bar{L}_{\omega 1}^\top \quad \bar{L}_{\omega 2}^\top \quad \dots \quad \bar{L}_{\omega m}^\top]^\top \in \mathbb{R}^{2m \times 2}$. The desired value of generalized velocity $\dot{\xi}$ can now be derived from (37) to nullify e_c . Since the latter is considered as the secondary task, we have

$$\dot{\xi}_2 := J_c^+ (-\lambda_2 e_c + \bar{L}_\omega \bar{\omega}) + (I_{n+4} - J_c^+ J_c) \dot{\xi}_3, \quad (38)$$

where λ_3 is a positive control gain, J_c^+ is the weighted pseudo-inverse of J_c expressed similarly as (32), and $\dot{\xi}_3$ corresponds to a generalized velocity to achieve a tertiary task with the available system DOFs.

4.3 THIRD TASK: ALIGNING THE MANIPULATOR WITH THE UAV'S CENTER OF GRAVITY

The objective is to limit the momentum of the manipulator on the UAV, which thence might be destabilized. Let $\mathcal{D}$ be the straight line in the gravity direction passing through the UAV center of mass. It is clear that when the end-effector is on $\mathcal{D}$ the momentum is decreased. Let ${}^{b,s}t_e$ be the translational vector from $\{R_s\}$'s origin to end-effector point e expressed in base

frame $\{R_b\}$ (whose Z -axis is assumed parallel to the gravity direction)

$${}^{b,s}t_e = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = {}^b R_s {}^s t_e = {}^b R_s {}^s R_u {}^u t_e \quad (39)$$

where we recall that ${}^u t_e$ is the translational vector from the arm base frame's origin to the end-effector. In order to have e on $\mathcal{D}$, the following constraint needs to be satisfied:

$$(t_x, t_y) := 0. \quad (40)$$

Time differentiating (39), we have

$${}^{b,s}\dot{t}_e = {}^b \dot{R}_s {}^s R_u {}^u t_e + {}^b R_s {}^s \dot{R}_u {}^u t_e + {}^b R_s {}^s R_u {}^u \dot{t}_e. \quad (41)$$

Using the classical relationship ${}^b \dot{R}_s = {}^b R_s [\omega]_\times$, recalling that ${}^s R_u$ is constant, and that ${}^u \dot{t}_e = J_{te} \dot{q}$, yields¹

$${}^{b,s}\dot{t}_e = {}^b R_s [\omega]_\times {}^s R_u {}^u t_e + {}^b R_s {}^s R_u J_{te} \dot{q}. \quad (42)$$

Let $\bar{t} = (t_x, t_y)^\top \in \mathbb{R}^2$, then we have

$$\bar{t} = I_{2 \times 3} {}^{b,s}t_e, \quad \text{with} \quad I_{2 \times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \quad (43)$$

Thus, by using (42) it follows that

$$\begin{aligned} \bar{t} &= I_{2 \times 3} {}^b R_s [\omega]_\times {}^s R_u {}^u t_e + I_{2 \times 3} {}^b R_s {}^s R_u J_{te} \dot{q} \\ &= \bar{t}_0 + J_t \dot{q}, \end{aligned} \quad (44)$$

with $\bar{t}_0 = I_{2 \times 3} {}^b R_s [\omega]_\times {}^s R_u {}^u t_e \in \mathbb{R}^2$ and $J_t = I_{2 \times 3} {}^b R_s {}^s R_u J_{te} \in \mathbb{R}^{2 \times n}$. Desired value $\bar{q}_g$ of joint velocity $\dot{q}$ in order $\bar{t}$ converges to zero can be derived as follows²

$$\dot{q}_g := J_t^+ (-\lambda_3 \bar{t} - \bar{t}_0) + (I_n - J_t^+ J_t) \dot{q}_l, \quad (45)$$

where λ_3 is a positive control gain, and $\dot{q}_l \in \mathbb{R}^n$ represents joints velocity not affecting task $\bar{t}$. Therefore, desired value $\dot{\xi}_3$ of $\dot{\xi}$ to achieve the tertiary task is

$$\dot{\xi}_3 = (\dot{q}_g^\top \quad 0_4^\top)^\top \in \mathbb{R}^{n+4}, \quad (46)$$

with 0_4 the (4×1) null vector.

¹Matrix J_{te} corresponds to the first blok of J_e relating the translational motions.

²In order to have consistency with the paper developments and to include other potential subtasks, relationship (44) can be expressed as a function of $\dot{\xi}$ instead of $\dot{q}$ as $\bar{t} = \bar{t}_0 + \bar{J}_t \dot{\xi}$ with $\bar{J}_t = (J_t \quad 0_{2 \times 4})$.

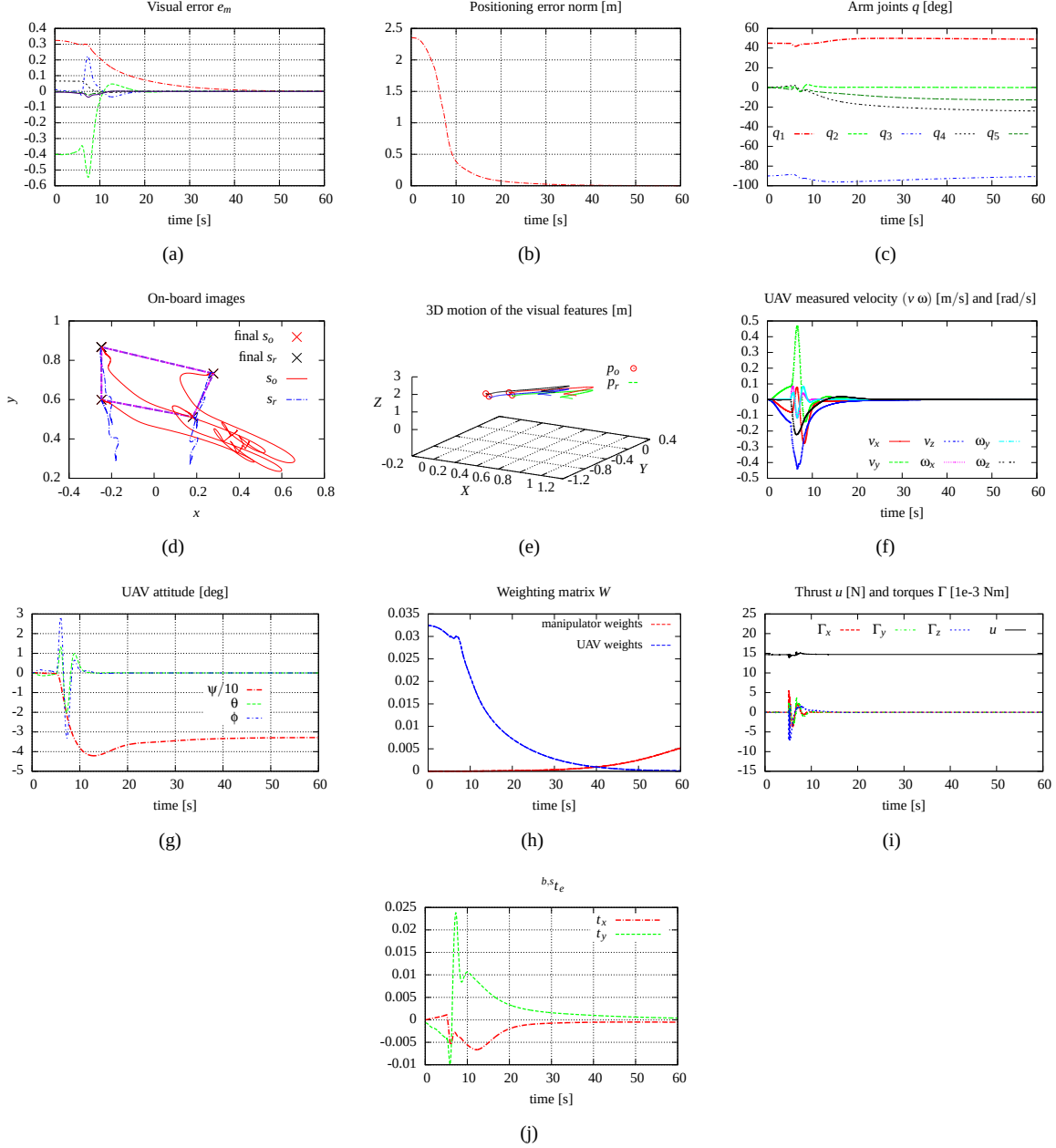


Figure 4. UAV/ARM SYSTEM AUTOMATICALLY POSITIONS AN ASSEMBLY PART (STRUCTURE) ON A VISUAL TARGET: DOCKING.

4.4 FOURTH TASK: KEEPING THE JOINTS IN DESIRED CONFIGURATIONS

Finally, the last subtask consists in keeping the joints in certain desired configurations. For instance, the i -th joint q_i is desired to remain centered between its two limits q_{im} and q_{iM} . To achieve this, we set desired value $\dot{q}_{li}$ of $\dot{q}$'s i -th element to

$$\dot{q}_{li} = -\lambda_4 \left[q_i - \frac{1}{2}(q_{mi} + q_{Mi}) \right], \quad (47)$$

with λ_4 a positive control gain.

5 SIMULATION RESULTS

The methods developed in this paper are implemented in C++ programming language. In this section we present simulation results, where the scenario consists in automatically positioning a structure on its corresponding target observed by an onboard camera. A five DOF ($n = 5$) robot arm manipulator endowed with rotational joints is considered. Each of the structure's part and the target, to which the former should be plugged in, consists in four ($m = 4$) points forming a rectangle, namely p_{ri} and $p_{oi_{i=1..4}}$ (e.g. see Fig. 1). The target rectangle is tilted by $(60^\circ, 60^\circ)$ around the X and Y axes of base frame $\{R_b\}$.

We consider error e_m on image moments as the main task for the robotic system by employing (31). We impose that the structure perpendicularly approaches the target before they superimpose. Thanks to the physical meaning of image moments highlighted earlier in this paper, this can be achieved by imposing area error (first element of e_m) to be the last converging to zero. This behavior can be achieved by setting its gain relatively low. Command (31) generates nine desired values: five for the manipulator joint velocity and four for the UAV translational and yaw velocities. These values are sent at the frequency of the camera streaming rate: 25 Hz, as commonly provided by the current video cameras. For the UAV to achieve each set of these desired velocities between two captures images, i.e. 40 ms, a Cartesian velocity low-level controller generates the necessary thrust and propeller torques. We employ a controller similar to that presented in [24], and assume that the measures of the vehicle translational velocity is provided or estimated using for instance [25–27]. The quadrotor accepts commands at 1 KHz.

As described in Section 4.2, the secondary task consists in nullifying error e_c on image coordinates using (38) in order to overcome the highlighted image redundancy. Moreover, in order to avoid eventual momentum that could be generated by the manipulator, (46) is used. Finally, and in line with the precedent subtask, we constrain the manipulator to have a “V” form by forcing the first and third joint of q to keep on $\frac{\pi}{4}$ and $-\frac{\pi}{2}$ rad, respectively, by using (47).

The involved control gains are empirically set to $\lambda_1 = 0.2\lambda$, $\Lambda = \text{diag}(0.1, 1, 1, 1, 1, 1)$, $\lambda_2 = 0.2\lambda$, $\lambda_3 = 0.5\lambda$, and $\lambda_4 = 0.05\lambda$. Parameter λ is an exponentially raising gain from 0.01 to 1, with value equal to 0.4 at the 200-th iteration. The corresponding results are shown in Fig. 4. We can see that moments-based visual error e_m converges to zero exponentially (Fig. 4(a)). We can also note that the error on area a decreases relatively with a low pace, as expected. Doing so in fact allows approaching the target roughly perpendicularly. Especially, Fig. 4(d) shows the variation in the image of each of the target points and structure points, respectively denoted s_o and s_r . We can see how each of the m pairs (s_{ri}, s_{oi}) vary in the image to finally superimpose. The final polygon in the image corresponds to the two oriented rectangles, which represent the superimposing target and structure images. One can notice that in contrast to the classical or so far adopted visual servoing schemes where only the image of the target varies, the paradigm (SVS) we present in this paper implies the variation of both target and the carried object image features. In addition, no desired image has been learnt.

Figure 4(b) plots the evolution of the error between the actuated structure and the target. More precisely, it shows the norm of the error between, i.e. $\frac{1}{m} \sum_{i=1}^m \|b_{tri} - b_{toi}\|$, where b_{tri} and $b_{toi} \in \mathbb{R}^3$ correspond to the 3D position of points p_{ri} and p_{oi} in base frame $\{R_b\}$, respectively. We can see indeed that this error converges to zero, which means that the structure has been correctly positioned on the target. This can also be concluded from Fig. 4(e), that depicts the motions of points p_{ri} and p_{oi} . The roll and pitch

angles of the quadrotor are depicted in Fig. 4(g).

We can see on Fig. 4(h) the evolution of weighting matrix W given by (33). As desired, while the UAV weights decrease when approaching the target, the manipulator’s weights raise thus taking the lead to position the platform.

Concerning the third task, Fig. 4(j) plots the first two elements of b^{st_e} given by (39). We can note indeed that they converge to zero, which means that the manipulator end-effector is aligning with UAV’s center of gravity. Finally, the fourth task is also achieved, where the manipulator’s first and third angle keep on $\frac{\pi}{4}$ and $-\frac{\pi}{2}$ rad, respectively, as can be seen on Fig. 4(c).

6 CONCLUSIONS

This paper presented a new visual servoing paradigm, namely SVS, applied for aerial manipulation. The developed methods endow a UAV equipped with a robot manipulator to automatically position structures on visual targets. The latter are sensed thanks to a video camera attached to the UAV. We have shown that employing image moments allows considering further requirements, that the desired task could benefit from. The proposed developments have been validated with computer simulations.

ACKNOWLEDGMENT

The research leading to these results has been supported by the ARCAS collaborative project, which has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-287617. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

REFERENCES

- [1] Maza, I., Kondak, K., Bernard, M., and Ollero, A., 2010. “Multi-UAV cooperation and control for load transportation and deployment”. *J. Intell. Robot. Syst.*, **57**, pp. 417–449.
- [2] Michael, N., Fink, J., and Kumar, V., 2011. “Cooperative manipulation and transportation with aerial robots”. *Auton. Robots*, **30**, pp. 73–86.
- [3] Marconi, L., Basile, L., Caprari, G., Carloni, R., Chiacchio, P., Huerzeler, C., Lippiello, V., Naldi, R., Janosch, N., Siciliano, B., Stramigioli, S., and Zwicker, E., 2012. “Aerial service robotics: The AIRobots perspective”. In 2nd Int. Conf. on Applied Robotics for the Power Industry.
- [4] Cacace, J., Finzi, A., Lippiello, V., Loianno, G., and Sanzone, D., 2013. “Aerial service vehicles for industrial inspection: Task decomposition and plan execution”. In 26th Int. Conf. on Industrial, Engineering & other Applications of Applied Intelligent Systems.

- [5] Lippiello, V., and Ruggiero, F., 2012. "Exploiting redundancy in cartesian impedance control of UAVs equipped with a robotic arm". In *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pp. 3768 – 3773.
- [6] Wu, A. D., Johnson, E. N., and Proctor, A. A., 2005. "Vision-aided inertial navigation for flight control". In *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, pp. 348–360.
- [7] Sattigeri, R. J., Johnson, E., and Calise, A. J., 2008. "A vision-based target tracking with adaptive target state". In *AIAA Guidance, Navigation, and Control Conference and Exhibit*.
- [8] Shakernia, O., Koo, T., and Sastry, S., 1999. "Landing an unmanned air vehicle: vision based motion estimation and nonlinear control". *Asian Journal of Control*, **1**, pp. 128–145.
- [9] Prasad, J., Calise, A. J., Johnson, E. N., Sattigeri, R., and Moon, J., 2008. "Flight demonstration of an adaptive guidance controller for autonomous formation flight". In *American Helicopter Society 64th, Annual Forum*.
- [10] De luca, A., Oriolo, G., and Robuffo Giordano, P., 2007. "Image-based visual servoing schemes for nonholonomic mobile manipulators". *Robotica*, **25**, Mar., pp. 131–145.
- [11] Pence, W. G., Farelo, F., Alqasemi, R., Sun, Y., and Dubey, R., 2012. "Visual servoing control of a 9-DoF WMRA to perform ADL tasks". In *IEEE Int. Conf. on Robotics and Automation*, pp. 916–922.
- [12] Chan, T. F., and Dubey, R. V., 1995. "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators". *IEEE Trans. on Robotics and Automation*, **11**, pp. 286 – 292.
- [13] Wang, Y., Lang, H., and de Silva, C. W., 2010. "A hybrid visual servo controller for robust grasping by wheeled mobile robots". *IEEE/ASME Trans. on Mechatronics*, **15**, pp. 757–769.
- [14] Mansard, N., Stasse, O., Chaumette, F., and Yokoi, K., 2007. "Visually-guided grasping while walking on a humanoid robot". In *IEEE Int. Conf. on Robotics and Automation*, pp. 3041–3047.
- [15] Siciliano, B., and Slotine, J.-J., 1991. "A general framework for managing multiple tasks in highly redundant robotic systems". In *Fifth International Conf. on Advanced Robotics*, pp. 1211–1216.
- [16] Mansard, N., and Chaumette, F., 2007. "Task sequencing for high level sensor-based control". *IEEE Trans. on Robotics*, **23**, pp. 60–72.
- [17] Hu, M. K., 1962. "Visual pattern recognition by moment invariants". *IRE Transactions on Information Theory*, February, pp. 179–187.
- [18] Chaumette, F., and Hutchinson, S., 2006. "Visual servo control, part I: Basic approaches". *IEEE Robotics and Automation Magazine*, **13**, pp. 82–90.
- [19] Mebarki, R., and Lippiello, V., Submitted. "Image-based control for aerial manipulation". *Asian Journal of Control*.
- [20] Chaumette, F., 2004. "Image moments: A general and useful set of features for visual servoing". *IEEE Trans. on Robotics*, **20**(4), August, pp. 713–723.
- [21] Tahri, O., and Chaumette, F., 2005. "Point-based and region-based image moments for visual servoing of planar objects". *IEEE Trans. on Robotics*, **21**(6), December, pp. 1116–1127.
- [22] Steger, C., 1996. On the calculation of arbitrary moments of polygons. Technical Report FGBV-96-05, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, Technische Universität München, Oct.
- [23] Zghal, H., Dubey, R. V., and Euler, J. A., 1990. "Efficient gradient projection optimization for manipulators with multiple degrees of redundancy". In *IEEE Int. Conf. on Robotics and Automation*, pp. 1006–1011.
- [24] Castillo, P., Dzul, A., and Lozano, R., 2004. "Real-time stabilization and tracking of a four-rotor mini rotorcraft". *IEEE Trans. on Control Systems and Technology*, **12**, pp. 510 – 516.
- [25] Lippiello, V., Loianno, G., and Siciliano, B., 2011. "MAV indoor navigation based on a closed-form solution for absolute scale velocity estimation using optical flow and inertial data". In *IEEE Int. Conf. on Decision and Control/European Control Conf.*, pp. 3566–3571.
- [26] Lippiello, V., and Mebarki, R., 2013. "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*.
- [27] Mebarki, R., and Siciliano, B., To appear. "Velocity-free image-based control of unmanned aerial vehicles". In *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*.