Chapter
Visual Servoing of Aerial Manipulators

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

Abstract This chapter describes the classical techniques to control an aerial manipulator by means of visual information and presents an uncalibrated image-based visual servo method to drive the aerial vehicle. The proposed technique has the advantage that it contains mild assumptions about the principal point and skew values of the camera, and it does not require prior knowledge of the focal length, in contrast to traditional image-based approaches.

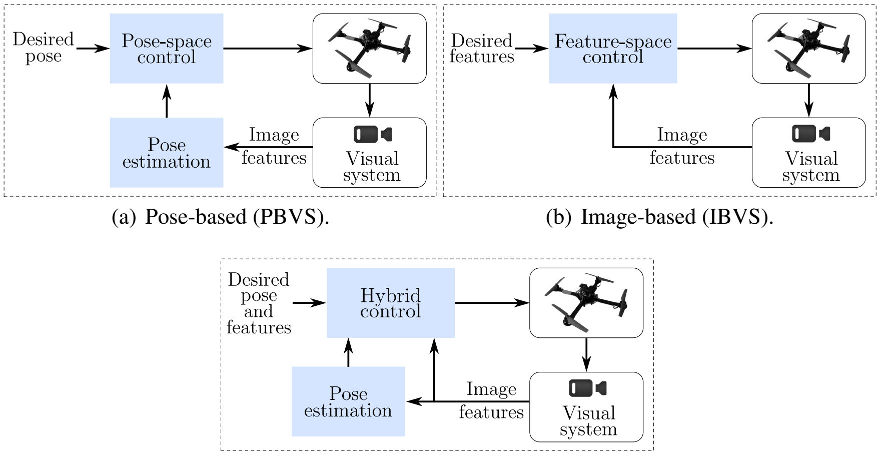
1 Introduction

AROMAs require more precise localization capabilities than traditional UAVs, specially during manipulation phases (*i.e.*, mission stages where the vehicle operates close to the target). For indoor AROMAs, accurate localization is usually obtained from infrared multi-camera systems, like Optitrack[1] or Vicon[2]. However, these devices are not suited for outdoor environments and require external infrastructure. To get rid of this dependency, and considering AROMA restrictions in terms of payload, size and power consumption, it is preferable to embark all perception hardware onboard, thus a reasonable choice is to drive the vehicle using information provided by onboard cameras.

Visual servo (VS), also known as vision-based robot control, is a technique which uses feedback information extracted from one or multiple cameras to control the motion of a robot. Regarding VS algorithms, vision-based robot control systems are usually classified in three groups [5, 14]: Pose-based visual servo (PBVS), image-based visual servo (IBVS) and hybrid control systems (HVS). These visual-control schemes are summarized in Figure 1.

[1] `www.optitrack.com`

[2] `www.vicon.com`



(a) Pose-based (PBVS).

(b) Image-based (IBVS).

(c) Hybrid (HVS).

Fig. 1 Visual servo schemes.

In PBVS, the geometric model of the target is used in conjunction with image features to estimate the target pose with respect to the camera frame ([26]). The control law is designed to reduce such pose error. As a PBVS example, [16] selects an optimal set of visual data at each sampling time and uses it for pose estimation. The dependency on 3D model information and well calibrated cameras makes in general all PBVS methods less attractive than image-based architectures. Moreover, they have the disadvantage that features could easily be lost in the image during the servo loop.

In IBVS on the other hand, both the control objective and the control law are directly expressed in the image space ([9]), minimizing the error between observed and desired image feature coordinates and formulating an image Jacobian to translate that error to Cartesian camera velocities ([11, 8, 3]). As a consequence, IBVS schemes do not need any a priori knowledge of the 3D structure of the observed scene. In addition, IBVS is more robust than PBVS with respect to uncertainties and disturbances affecting the model of the robot, as well as the calibration of the camera [13]. Nevertheless, in most IBVS methods, error convergence to zero can typically be guaranteed only in a neighborhood of the desired configuration [6].

To deal with PBVS and IBVS shortcomings, Hybrid methods, also called 2-1/2-D visual servo [18], combine IBVS and PBVS to estimate partial camera displacements at each iteration of the control law, minimizing a functional of both the pose error and the image feature error. [15] present a method to position a AROMA gripper using PBVS and IBVS techniques set in a hierarchical task composition control law.

In all image-based and hybrid approaches however, the resulting image Jacobian or interaction matrix, which relates the camera velocity to the image feature velocities, depends on a priori knowledge of the intrinsic camera parameters. To do away

Table 1 Symbols used in the development of the uncalibrated image-based visual servo method.

Definition	**Symbol**
Target *j-th* control point in camera coordinates	$\mathbf{c}_j$
j-th target control point velocities in the image plane	$[\dot{u}_j, \dot{v}_j]^\top$
Calibration-free image Jacobian for the *j-th* control point	$\mathbf{J}_j$
Camera translational velocity	$\dot{\mathbf{p}}_c$
Camera rotational velocity	ω_c
Image Jacobian	$\mathbf{J}_{vs}$
Visual servo task	σ_S
Jacobian with columns from controllable robot DoFs	$\mathbf{J}_S$
Jacobian with columns from non-controllable robot DoFs	$\bar{\mathbf{J}}_S$
Non-controllable platform DoFs: $\varpi = [\omega_x, \omega_y]^\top$	ϖ

with the depth dependence, one could optimize for the parameters in the image Jacobian whilst the error in the image plane is being minimized, for instance using Gauss-Newton to minimize the squared image error and non-linear least squares optimization for the image Jacobian [22]; using weighted recursive least squares (RLS), not to obtain the true parameters, but instead an approximation that still guarantees asymptotic stability of the control law in the sense of Lyapunov [12]; or using k-nearest neighbor regression to store previously estimated local models or previous movements, and estimating the Jacobian using local least squares (LLS) [10]. To provide robustness to outliers in the computation of the Jacobian, [25] proposes the use of an M-estimator.

Classical PBVS and IBVS approaches are well know by the community. Instead, in this chapter we present the uncalibrated image-based method (UIBVS) described in [24, 23]: an approach to image-based visual servo in which the computation of the image Jacobian makes mild assumptions about the camera parameters (principal point and skew values) without requiring prior knowledge of the camera focal length, in contrast to traditional IBVS (this focal length is iteratively estimated within the control loop). Independence of focal length true value makes the system robust to noise and to unexpected large variations of this parameter (*e.g.*, poor initialization or an unaccounted zoom change).

This Chapter is structured as follows. Section 2 describes how to obtain an image Jacobian that does not depend on camera intrisic parameters and the control law of the UIBVS approach. This control law is then included in a hierarchical scheme described in Section 3. Section 4 presents experimental validations through simulation (4.1) and real robot (4.2) case studies of the described uncalibrated visual servo approach. Final remarks are presented in Section 5.

2 Uncalibrated Image-based Visual Servoing (UIBVS)

Drawing inspiration on the UPnP algorithm [21], we describe in the following subsection a method to solve for the camera pose and focal length using a reference system attached to the target object. Then, the method is extended to compute a calibration-free image Jacobian for our servo task and to compute the desired control law for our AROMA [24, 23]. The main symbols of this mathematical development are depicted in Table 1.

3D target features are parameterized with their barycentric coordinates, and the basis of these coordinates is used to define a set of j control points $\mathbf{c}_j$. Computing the pose of the object with respect to the camera resorts to computing the location of these control points with respect to the camera frame. A least squares solution for the control point coordinates albeit scale, is given by the eigenvector ν of the null eigenvalue of a linear system made up of all 2D to 3D perspective projection relations between the target points. Given the fact that distances between control points must be preserved, these distance constraints can be used in a second least squares computation to solve for scale and focal length, formulating a system of the form $\mathbf{Lb} = \mathbf{d}$, where $\mathbf{b} = [\beta^2, \alpha^2\beta^2]^\top$, $\mathbf{L}$ is a 6×2 matrix built from the elements of ν, and $\mathbf{d}$ is the 6-vector of squared distances between the control points.

As the camera moves, the velocity of each target control point $\mathbf{c}_j$ in camera coordinates can be related to the camera spatial velocity $(\dot{\mathbf{p}}_c, \boldsymbol{\omega}_c)$ with $\dot{\mathbf{c}}_j = -\dot{\mathbf{p}}_c - \boldsymbol{\omega}_c \times \mathbf{c}_j$. Which combined with the solution of the previous least squares, together with some simplifications and rearrangements, we can obtain the control point velocities in the image plane

$$\dot{u}_j = \frac{-\dot{p}_x - \alpha\beta\nu_z\omega_y + \beta\nu_y\omega_z}{\beta\nu_z} - \frac{\nu_x(-\dot{p}_z - \beta\nu_y\omega_x + \beta\nu_x\omega_y)}{\alpha\beta\nu_z^2} \tag{1a}$$

$$\dot{v}_j = \frac{-\dot{p}_y - \alpha\beta\nu_z\omega_x + \beta\nu_x\omega_z}{\beta\nu_z} - \frac{\nu_y(-\dot{p}_z - \beta\nu_y\omega_x + \beta\nu_x\omega_y)}{\alpha\beta\nu_z^2}, \tag{1b}$$

where ν_x, ν_y, and ν_z are the x, y, and z components of eigenvector ν related to the control point $\mathbf{c}_j$, and which can be rewritten as $\dot{\mathbf{s}}_j = \mathbf{J}_j \mathbf{v}_c$, with $\dot{\mathbf{s}}_j = [\dot{u}_j, \dot{v}_j]^\top$, the image velocities of control point j, and $\mathbf{v}_c = [\dot{\mathbf{p}}_c^\top, \boldsymbol{\omega}_c^\top]^\top$. $\mathbf{J}_j$ is our desired calibration-free image Jacobian for the j-th control point, and takes the form

$$\mathbf{J}_j = \begin{bmatrix} \frac{-1}{\beta\nu_z} & 0 & \frac{\nu_x}{\alpha\beta\nu_z^2} & \frac{\nu_x\nu_y}{\alpha\nu_z^2} & \frac{-\nu_x^2-\alpha^2\nu_z^2}{\alpha\nu_z^2} & \frac{\nu_y}{\nu_z} \\ 0 & \frac{-1}{\beta\nu_z} & \frac{\nu_y}{\alpha\beta\nu_z^2} & \frac{\nu_y^2+\alpha^2\nu_z^2}{\alpha\nu_z^2} & \frac{-\nu_x\nu_y}{\alpha\nu_z^2} & \frac{-\nu_x}{\nu_z} \end{bmatrix}. \tag{2}$$

Stacking these together, we get the image Jacobian for all control points $\mathbf{J}_{vs} = \left[\mathbf{J}_1 \ \dots \ \mathbf{J}_4\right]^\top$.

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

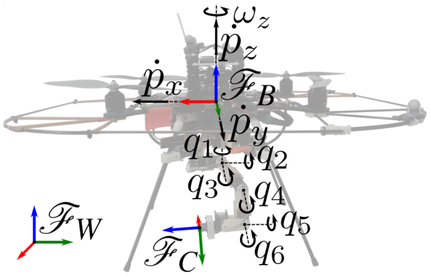


Fig. 2 The AROMA used in the experiments is composed of a 4 DoF quadrotor, commanded at high-level by 3 linear and an angular velocities ($\dot{p}_x$, $\dot{p}_y$, $\dot{p}_z$ and ω_z), and a 6 DoF robotic arm with joints q_i, $i = 1...6$; and world, body and camera reference frames indicated by $\mathscr{F}_W$, $\mathscr{F}_B$ and $\mathscr{F}_C$, respectively.

$\dot{\mathbf{e}} = \dot{\mathbf{s}}$, and, for a desired exponential decoupled error decrease $\dot{\mathbf{e}} = -\Lambda_S \mathbf{e}$, we have a desired camera velocity $\mathbf{v}_c = -\Lambda_S \mathbf{J}_{vs}^{+}\mathbf{e}$, where Λ_S is a 6×6 positive definite gain matrix and $\mathbf{J}_{vs}^{+} = (\mathbf{J}_{vs}^{\top}\mathbf{J}_{vs})^{-1}\mathbf{J}_{vs}^{\top}$ is the left Moore-Penrose pseudoinverse of $\mathbf{J}_{vs}$.

Now consider the quadrotor-arm system equipped with a camera mounted at the end-effector's arm as shown in Fig. 2, with the world frame ($\mathscr{F}_W$) to be located at the target. We can define a joint quadrotor-arm Jacobian that relates the local translational and angular velocities of the platform and those of the i arm joints, $\mathbf{v}_{qa} = (\dot{p}_x, \dot{p}_y, \dot{p}_z, \omega_x, \omega_y, \omega_z, \dot{q}_1, \ldots, \dot{q}_M)$, to the desired camera velocities computed from the visual servo $\mathbf{v}_c = \mathbf{J}_{qa}\mathbf{v}_{qa}$, with $\mathbf{J}_{qa}$ the Jacobian matrix of the whole robot. Then, we can relate the desired high-level control velocities with our visual servo task, which we term now σ_S

$$\mathbf{J}_{qa}\mathbf{v}_{qa} = -\Lambda_S \underbrace{\mathbf{J}_{vs}^{+}\mathbf{e}}_{\sigma_S} . \tag{3}$$

Unfortunately as said before, the quadrotor is an underactuated vehicle. So, to remove the non-controllable variables from the control command, their contribution to the image error can be isolated from that of the controllable ones by extracting the columns of $\mathbf{J}_{qa}$ and the rows of $\mathbf{v}_{qa}$ corresponding to ω_x and ω_y, reading out these values from the platform gyroscopes, and subtracting them from the camera velocity ([17]) as $\mathbf{J}_S\dot{\mathbf{q}} + \bar{\mathbf{J}}_S\boldsymbol{\varpi} = -\Lambda_S\sigma_S$, where $\boldsymbol{\varpi} = [\omega_x, \omega_y]^{\top}$, $\bar{\mathbf{J}}_S$ is the Jacobian formed by the columns of $\mathbf{J}_{qa}$ corresponding to ω_x and ω_y, and $\mathbf{J}_S$ is the Jacobian formed by all other columns of $\mathbf{J}_{qa}$, corresponding to the actuated variables $\dot{\mathbf{q}} = [\dot{p}_x, \dot{p}_y, \dot{p}_z, \omega_z, \dot{q}_1, \ldots, \dot{q}_M]^{\top}$. Rearranging terms

$$\mathbf{J}_S\dot{\mathbf{q}} = \underbrace{-\Lambda_S\sigma_S - \bar{\mathbf{J}}_S\boldsymbol{\varpi}}_{\boldsymbol{\xi}} \tag{4}$$

Table 2 Symbols used in the development of the hierarchical task control.

Definition	**Symbol**
Jacobian matrix of the *i-th* task	$\mathbf{J}_i$
Augmented Jacobian matrix for the *i-th* subtask	$\mathbf{J}_{0\|\ldots\|i}$
Projection matrix into the null space of task Jacobian $\mathbf{J}_i$	$\mathbf{N}_i$
Augmented projection matrix into the null space of task augmented Jacobian $\mathbf{J}_{0\|\ldots\|i}$	$\mathbf{N}_{0\|\ldots\|i}$

and with this, our main task velocity corresponding to the visual servo is $\dot{\mathbf{q}} = \mathbf{J}_S^+ \boldsymbol{\xi}$, where, with 6 linearly independent rows and $4+M > 6$ columns, $\mathbf{J}_S^+$ is computed with the right Moore-Penrose pseudoinverse $\mathbf{J}_S^\top (\mathbf{J}_S \mathbf{J}_S^\top)^{-1}$.

In order to penalize the motion of the quadrotor vs the arm to account for their different motion capabilities, we can define a weighted norm of the whole velocity vector $\|\dot{\mathbf{q}}\|_W = \sqrt{\dot{\mathbf{q}}^\top \mathbf{W} \dot{\mathbf{q}}}$ as in [4], and use a weighted task Jacobian to solve for the weighted controls as in [23].

3 Hierarchical task control

Even though the quadrotor itself is underactuated (4 DoF), by attaching a robotic arm with more than 2 DoF we can attain over-actuation ($D = 4+M$). In our case, $M = 6$. Exploiting this redundancy, we can achieve additional tasks acting on the null space of the quadrotor-arm Jacobian [20], while preserving the primary task. These tasks can be used to reconfigure the robot structure without changing the position and orientation of the arm end-effector. This is usually referred to as *internal motion* of the arm. One possible way to specify a secondary task is to choose its velocity vector as the gradient of a scalar objective function to optimize [7, 19]. Multiple secondary tasks can be arranged in hierarchy and, to avoid conservative stability conditions [1], the *augmented inverse-based projections* method is here considered [2]. In this method, lower priority tasks are not only projected onto the null space of the task up in the hierarchy, but onto the null space of an augmented Jacobian with all higher priority tasks. The addition of tasks in cascade is possible as long as there exist remaining DoF from the concatenation of tasks higher up in the hierarchy. The main symbols of this mathematical development are depicted in Table 2.

This hierarchical control law can be described in a generalized form for the case of η prioritized subtasks as follows (we refer the reader to [23] for exhaustive details on the hierarchical control law development)

$$\dot{\mathbf{q}} = \mathbf{J}_0^+ \Lambda_0 \widetilde{\sigma}_0 + \sum_{i=1}^{\eta} \mathbf{N}_{0|\ldots|i-1} \mathbf{J}_i^+ \Lambda_i \widetilde{\sigma}_i - \bar{\mathbf{J}}_{0|\ldots|\eta}\, \varpi \tag{5}$$

with the recursively-defined compensating matrix

$$\bar{\mathbf{J}}_{0|\ldots|\eta} = \mathbf{N}_{0|\ldots|i-1}\mathbf{J}_i^{+}\bar{\mathbf{J}}_i + (\mathbf{I} - \mathbf{N}^{+}_{0|\ldots|i-1}\mathbf{J}_i^{+}\bar{\mathbf{J}}_i)\bar{\mathbf{J}}_{0|\ldots|i-1}\,, \tag{6}$$

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

$$\mathbf{N}_{0|\ldots|i} = (\mathbf{I} - \mathbf{J}^{+}_{0|\ldots|i}\mathbf{J}_{0|\ldots|i}) \tag{7}$$

$$\mathbf{J}_{0|\ldots|i} = [\mathbf{J}_0^{\top}\ \ldots\ \mathbf{J}_i^{\top}]^{\top}\,. \tag{8}$$

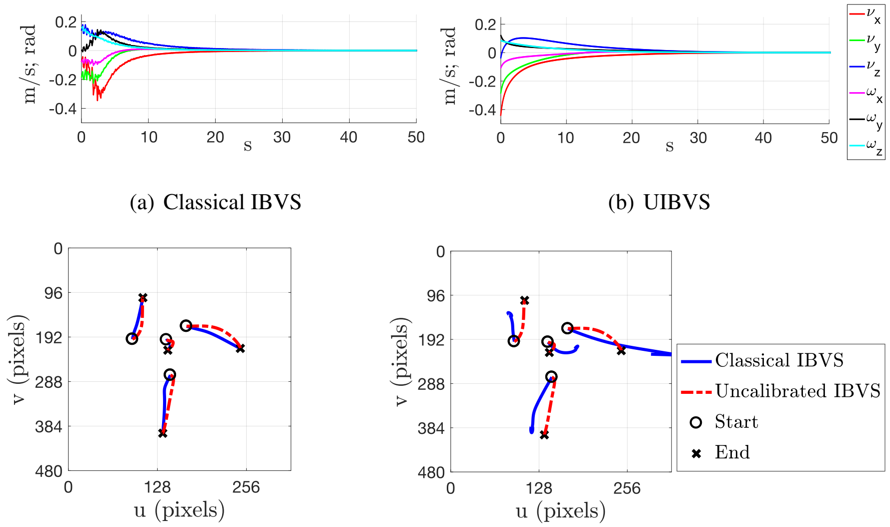
This strategy, in contrast to the more restrictive one we presented in [15] might achieve larger constraint-task reconstruction errors than the full least squares secondary task solution in [15] but with the advantage that algorithmic singularities arising from conflicting tasks are decoupled from the singularities of the secondary tasks.

4 Experimental results

In this section, we take advantage of the hierarchical control law presented in [23], where the visual servo control law that takes into account the uncontrollable state variables is not the main task. Specifically, here we consider the following ordered tasks: a primary safety task (I) considering potential collisions (inflation radius); a secondary task performing visual servoing (S), and lower in the hierarchy, the alignment of the center of gravity of the AROMA (G), and a technique to stay away from the arm's joint limits (L). We refer the reader to [23] for more details on this control law, the extra task definitions and to the corresponding stability analysis.

4.1 Validation through simulation

To validate the proposed visual servo scheme Fig. 3 shows the effects of noise in the camera focal length during a simulated servo task. Specifically, camera velocities for a servoing task with a focal length of 10 mm, and subject to white noise variations with 1 mm standard deviation are shown in Fig. 3(a) using a focal-length dependent Jacobian, and results with our proposed calibration-free scheme are shown in Fig. 3(b). Even when the servo task can be successfully completed in both cases in approximately the same amount of time, the proposed method provides a much smoother tracking of the camera to such variations in camera length. Fig. 3(c) and 3(d) shows the image plane trajectories of the control points for a similar experiment. Under equal noise-free simulation conditions (plot 3(c)) both methods have comparable asymptotic convergence. But, for an initialization error in the focal length of 20%, the classical approach is unable to reach the desired configuration, in contrast to the proposed approach (plot 3(d)). An unexpected variation



(a) Classical IBVS

(b) UIBVS

(c) Noise-free

(d) Initialization error of 20% in camera focal length.

Fig. 3 Effects of noise in both IBVS and UIBVS during a simulated servo task. Figures a) and b) correspond to camera velocities subject to white noise with 1 mm standard deviation in the focal length. Figures c) and d) are control point trajectories in the image plane.

of focal length is assumed by the classical image-based servo method as camera motion along the z axis and to recover from this, the control law induces undesirable changes in the robot velocity commands.

4.2 Real AROMA experiments

To address the dynamical effects of the overall system our cascaded architecture considers two different control loops at very high frequency (1KHz), one for the arm and one for the attitude of the UAV; and a hierarchical task controller running at much lower frequency (camera frame rate), hence avoiding dynamic coupling between them. Moreover, we impose bounds on the maximum velocities and accelerations for the arm joints that in practice result also in small torques in the arm base, which the task controller is able to adequately compensate. Furthermore, instead of developing a robust 3D feature detector and tracker, we use a planar detector of a target with a known geometry to retrieve the target frame, to which we add virtual features and then compute their location with respect to the target frame, as well as their basis, i.e., the control points. At each iteration, the marker is detected in the scene and the projection of the control points is computed. Those 2D-3D feature relationships represent the input to our visual servoing algorithm

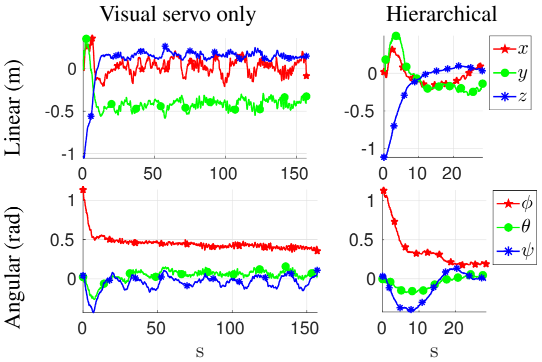
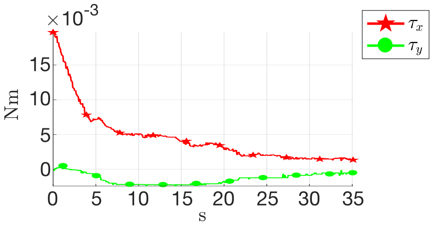


Fig. 4 Camera pose error during visual servoing. Comparison of using or not the hierarchical task priority control law with all the subtasks proposed.



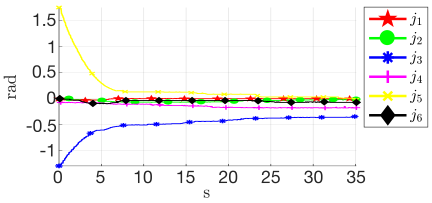


Fig. 5 Torque effects in the quadrotor body applying G task.

Fig. 6 Arm joints positioning error applying L task.

We conducted a series of experiments with missions consisting on autonomously taking off and flying to a location in which the target appears in the field of view of the camera, turning on then the hierarchical task controller to servo the system towards a desired camera pose, and finally autonomously landing the system. The real experiments were conducted with our robot Kinton (Fig. 2), based on an Asctec Pelican quadrotor, and equipped with an onboard embedded PC (1.6GHz CPU) and a variety of sensors including an IMU and a barometer. All our algorithms are running onboard in real time with a camera frame rate at 20Hz.

Fig. 4 shows a comparison of task execution with and without activation of the task hierarchy. The left frames show linear and angular pose errors when only the servo task is active. In this experiment, task completion is considered to be reached at an Euclidean position error of 0.15m and 0.2rad; otherwise task abortion is executed if after 3 min of flight time the target is not reached. We are aware that better performance is possible with a more elaborate tuning of the different control gains, however, this level of precision is good enough to show that the task composition scheme allowed the system to quickly reach its target, whereas without the hierarchical task composition in place, the task could not be accomplished.

Arm CoG alignment is crucial to improve flight behavior (*i.e.*, G task). With it the approaching maneuver is softer allowing us not only to easily reach the desired servoing error but also reducing aggressive maneuvers to compensate the arm pay-

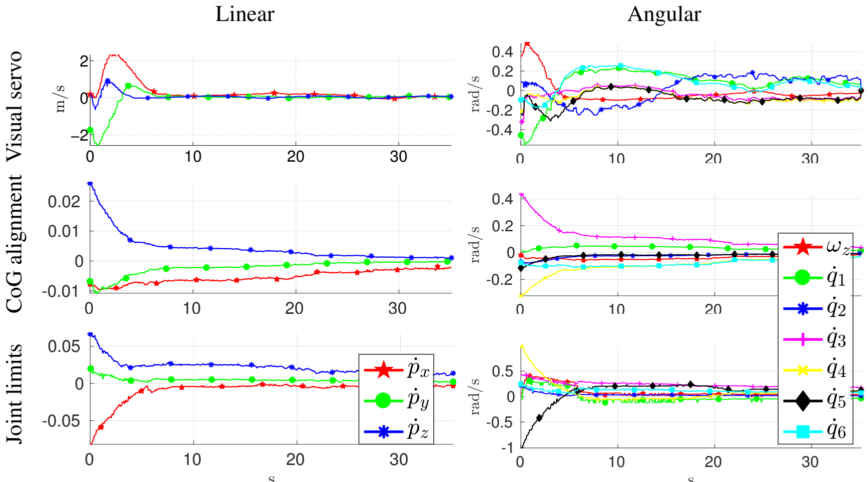


Fig. 7 Actuator velocities in a real experiment corresponding to the individual contributions of each subtask: visual servoing, CoG alignment and joint limits.

load, thus reducing energy consumption which is a very important issue for this type of aerial platforms. Fig. 5 shows the effect of this alignment in terms of arm torque reduction.

The last task is designed to favor a desired arm configuration and it can be used to push the joints away from singularities and potentially increase maneuverability. Fig. 6 shows the error between the current and desired joint positions when the task is included in the hierarchy at the lowest priority level.

Finally, to evaluate the contribution of each control variable to the execution of the different tasks we present plots of the whole set of velocity profiles applied to the AROMA actuators (*i.e.*, 3 quadrotor linear velocities and 6 arm angular velocities plus quadrotor yaw velocity) in Fig. 7. The fact that all tasks report velocity values for the control variables indicate the availability of DoF for their execution from their higher priority tasks. The visual servo mission task requires 6 DoF, and the secondary and comfort tasks with lower priority can take advantage of the remaining 4 DoF. The gravitational vector alignment task and the joint limits avoidance task require 1 DoF each being scalar cost functions to minimize (see [23]).

5 Conclusions

With this control law we can achieve not only the visual servoing but also other tasks taking into account their specific priorities. A primary task is designed to safeguard the platform integrity. The secondary task corresponds to the uncalibrated visual servo, and lower priority tasks are designed to alleviate quadrotor stabilization issues. This hierarchical strategy might not achieve the optimum constraint-task reconstruction errors, but instead the algorithmic singularities arising from conflicting tasks are decoupled from the singularities of the secondary tasks. Moreover, it only requires independent tasks for the uncontrollable variables to guarantee exponentially stability of the system.

References

1. G. Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25(5):985–994, October 2009.
2. P. Baerlocher and R. Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, volume 1, pages 323–329 vol.1, Oct 1998.
3. Odile Bourquardez, Robert Mahony, Nicolas Guenard, Franois Chaumette, Tarek Hamel, and Laurent Eck. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, 25(3):743–749, 2009.
4. Tan Fung Chan and R. V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292, Apr 1995.
5. F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90, Dec 2006.
6. Francois Chaumette. Potential Problems of Stability and Convergence in Image-Based and Position-Based Visual Servoing. In *The confluence of vision and control*, number 237 in Lecture Notes in Control and Information Sciences, pages 66–78. Springer, 1998.
7. S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, Jun 1997.
8. P.I. Corke and S.A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, August 2001.
9. B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.
10. Amir Massoud Farahmand, Azad Shademan, and Martin Jgersand. Global visual-motor estimation for uncalibrated visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1969–1974, San Diego, USA, October 2007.
11. K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura. Manipulator control with image-based visual servo. In *IEEE International Conference on Robotics and Automation*, pages 2267–2271, Sacramento, USA, April 1991.
12. Koh Hosoda and Minoru Asada. Versatile visual servoing without knowledge of true jacobian. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 186–193, Munich, Germany, September 1994.
13. S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, Oct 1996.
14. F. Janabi-Sharifi, L. Deng, and W. J. Wilson. Comparison of basic visual servoing methods. *IEEE/ASME Transactions on Mechatronics*, 16(5):967–983, Oct 2011.
15. Vincenzo Lippiello, Jonathan Cacace, Angel Santamaria-Navarro, Juan Andrade-Cetto, Miguel ngel Trujillo, Yamnia Rodrguez Esteves, and Antidio Viguria. Hybrid visual servoing with hierarchical task composition for aerial manipulation. *IEEE Robotics and Automation Letters*, 1(1):259–266, Jan 2016.
16. Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani. Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration. *IEEE Transactions on Robotics*, 23(1):73–86, February 2007.

17. R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, Sept 2012.
18. E. Malis, F. Chaumette, and S. Boudet. 2 frac12;d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, Apr 1999.
19. Yoshihiko Nakamura. *Advanced robotics - redundancy and optimization.* Addison-Wesley, 1991.
20. Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.
21. A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2387–2400, Oct 2013.
22. Jenelle Armstrong Piepmeier, Gary V. McMurray, and Harvey Lipkin. Uncalibrated dynamic visual servoing. *IEEE Transactions on Robotics and Automation*, 20(1):143–147, 2004.
23. A. Santamaria-Navarro, P. Grosch, V. Lippiello, J. Sol, and J. Andrade-Cetto. Uncalibrated visual servo for unmanned aerial manipulation. *IEEE/ASME Transactions on Mechatronics*, 22(4):1610–1621, Aug 2017.
24. Angel Santamaria-Navarro and Juan Andrade-Cetto. Uncalibrated image-based visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 5247–5252, Karlsruhe, Germany, May 2013.
25. Azad Shademan, Amir-Massoud Farahmand, and Martin Jgersand. Robust jacobian estimation for uncalibrated visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 5564–5569, Anchorage, USA, May 2010.
26. W.J. Wilson, C.C. Williams Hulls, and G.S. Bell. Relative end-effector control using Cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, October 1996.