

# Visual Servoing with Exploitation of Redundancy: An Experimental Study

Alessandro De Luca   Massimo Ferri   Giuseppe Oriolo   Paolo Robuffo Giordano  
Dipartimento di Informatica e Sistemistica  
Università di Roma “La Sapienza”  
Via Ariosto 25, 00185 Roma, Italy  
{deluca,oriolo,robuffo}@dis.uniroma1.it   massimo.ferri@gmail.com

**Abstract**— Within the standard IBVS framework for control of generic robotic systems, a suitable exploitation of redundancy w.r.t. the given visual task can significantly improve the overall task execution. Indeed, redundancy can be used to avoid occlusions, joint limits, or to realize tasks that would be ill-conditioned if addressed altogether. In this respect, we propose an experimental evaluation of the performance of two redundancy resolution schemes, namely Task Priority and Task Sequencing, when adopted to realize IBVS tasks on a mobile robot equipped with a pan-tilt camera onboard.

## I. INTRODUCTION

The introduction of visual information for pose control of robot systems has been an active topic in the last decades [1]–[4]. Increase of robot operation flexibility, improvement of position accuracy, robustness against sensor noise and calibration uncertainties, and reactivity to environmental changes are among the major benefits. This is especially true for the class of mobile robots, where the elaboration of visual cues is often crucial for localization and navigation. Another interesting use of visual feedback is the approach known as *visual servoing*. In this framework, the robotic task is directly specified in terms of some image features extracted from a target object. These features are then used to control the robot/camera motion through the scene until the final robot pose is reached. Two main approaches have been proposed in the past years to deal with this kind of tasks, namely position-based visual servoing (PBVS) and image-based visual servoing (IBVS) schemes, but recently a number of hybrid approaches has also been explored. A thorough presentation and discussion of the different methods can be found in [4]–[6].

In this paper, we consider the IBVS approach for control of generic robotic systems, including fixed-base manipulators, wheeled mobile robots (omnidirectional or subject to nonholonomic constraints), and mobile manipulators (a combination of the two above). IBVS methods compute the error signal directly in terms of the features extracted from the image, whose motion on the image plane is mapped to the linear/angular velocity of the camera via an interaction matrix. As a consequence, there is no need of a 3D model of the target, and convergence is generally robust w.r.t. disturbances and uncertainties in the camera/robot model [7]. Moreover, direct control of the feature motion on the image plane allows the implementation of strategies aimed at keeping the target

always in the field of view of the camera [8]. The camera can be mounted on the end-effector of the fixed-base or mobile manipulator (eye-in-hand), or carried on the main body of a mobile platform (possibly, with additional camera motion dofs).

In view of its structure, IBVS naturally lends itself to be interpreted as a standard robotic task, where the controlled variables are the visual features extracted from the image plane. It is then possible to address, within the IBVS framework, all the classical task realization issues, such as set-point regulation, trajectory tracking, kinematic inversion, and, in particular, redundancy resolution [9]. The latter topic can be especially relevant when dealing with visual tasks: for instance, the redundant velocity commands of the system can be exploited to avoid, during the motion, joint limits and occlusions that would cause failure of the task [10]–[12]. Another interesting benefit of exploiting redundancy is to improve the realization of tasks that would be close to singularity if addressed altogether. This can be achieved, for instance, by suitably splitting the main task and by assigning different priorities to each subtask through a Task Priority (TP) strategy [13]. Furthermore, even in non-redundant cases, the main task can be temporally decomposed in distinct phases so that an ‘artificial’ redundancy is introduced during task execution — an approach known as Task Sequencing (TS) [14].

The aim of this work is to overview the theoretical steps (see [14] for a full treatment) and to provide experimental validation of IBVS when redundancy is explicitly exploited during the robot motion. To this end, we present the experimental realization of IBVS regulation tasks on a mobile robot equipped with a pan-tilt camera, by focusing on the benefits obtained through the use of the TP and TS approaches. The rest of the paper is organized as follows: Sect. II reviews the IBVS framework from a task-oriented point of view, while Sect. III addresses the mentioned redundancy resolution techniques. Finally, in Sect. IV the obtained experimental results are presented and discussed.

## II. VISUAL SERVOING TASKS

### A. Pin-hole camera model

With reference to Fig. 1, consider an inertial world reference frame  $\mathcal{F}_O : \{O; \vec{X}_O, \vec{Y}_O, \vec{Z}_O\}$  and a pin-hole camera associated to the moving frame  $\mathcal{F}_C : \{O_C; \vec{X}_C, \vec{Y}_C, \vec{Z}_C\}$ ,

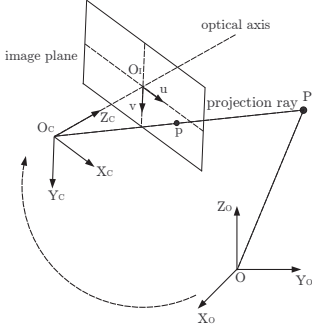


Fig. 1: World and camera frame definition.

with  $\vec{Z}_C$  coincident with the camera optical axis. The image plane, perpendicular to the optical axis, lies at a distance  $\lambda$  (the focal length) from  $O_C$ , and is endowed with a 2D reference frame  $\mathcal{F}_I : \{O_I; \vec{u}, \vec{v}\}$  with axes parallel to  $\vec{X}_C$  and  $\vec{Y}_C$ , respectively. Furthermore, let vector  $[v_C^T \ \omega_C^T]^T \in \mathbb{R}^6$  represent the linear/angular velocity of  $\mathcal{F}_C$  w.r.t.  $\mathcal{F}_O$  expressed in  $\mathcal{F}_C$ .

The pin-hole camera projects a 3D point  $P$  in  $\mathcal{F}_C$  with homogeneous coordinates  $\bar{P} = [X \ Y \ Z \ 1]^T$  into a 2D point  $p$  with homogeneous normalized coordinates  $\bar{p} = [p_u \ p_v \ 1]^T = [X/Z \ Y/Z \ 1]^T$ . The image plane measurement (in pixels) of point  $p$  is given by  $\tilde{p} = [\tilde{p}_u \ \tilde{p}_v \ 1]^T = A\bar{p}$ , where  $A$  is a nonsingular matrix containing the camera intrinsic parameters, i.e.,

$$A = \begin{bmatrix} \lambda k_u & -\lambda k_u / \tan \delta & u_0 \\ 0 & \lambda k_v / \sin \delta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

with  $[u_0 \ v_0]^T$  being the coordinates of the principal point (in pixels),  $\lambda$  the focal length (in meters),  $k_u$  and  $k_v$  the magnifications in the  $\vec{u}$  and  $\vec{v}$  directions (in pixel/meters), and  $\delta$  the angle between these axes. In order to simplify the notation, in the following we will assume that any quantity is expressed in the normalized space. This is equivalent to assume a calibrated camera, i.e., full knowledge of the calibration matrix  $A$ .

Given a vector of visual features  $f = [f_1 \dots f_s]^T \in \mathbb{R}^s$ , the velocity twist  $(v_C, \omega_C)$  of the camera is mapped into  $\dot{f}$  by a  $s \times 6$  matrix  $J_v(f, \chi)$  called the *interaction matrix*

$$\dot{f} = J_v(f, \chi) \begin{bmatrix} v_C \\ \omega_C \end{bmatrix}, \quad (2)$$

where  $\chi$  is a vector representing 3D information associated to  $f$ . It is possible to determine the interaction matrix for many features of interest, see [3] for the case of points, lines, planes, circles, and [15], [16] for the set of image moments.

### B. Visual task Jacobian

Let  $q \in \mathbb{R}^n$  be the configuration vector of the overall robot system,  $u \in \mathbb{R}^p$  the available velocity commands, and  $f \in \mathbb{R}^s$  the feature vector of interest for the chosen visual task. In kinematically unconstrained systems, like standard fixed-base manipulators, the joint velocity  $\dot{q}$  can

be arbitrarily specified at any arm configuration, hence we can set  $\dot{q} = u$  and  $p = n$ . On the other hand, when a (partially) nonholonomic system is considered, as in the case of a nonholonomic mobile manipulator (NMM), the number  $p$  of available velocity commands is less than the number  $n$  of generalized coordinates, implying that  $\dot{q} \neq u$ . This is particularly relevant in our case since the robot used in the experiments falls into this class, being in particular  $\dot{q} = G(q)u$ , with  $G$  having more rows than columns— see Sect. IV-A for further details. Specific kinematic modeling of such systems can be found in [17] where we show that, for any given task, a suitable Jacobian can be derived and used in all classical problems originally addressed for fixed-base manipulators.

For any robotic system equipped with a camera, the differential mapping between  $\dot{f}$  and  $u$  can be obtained by dividing the problem in two steps, i.e., by deriving the  $s \times p$  image Jacobian  $J_{\text{img}}$ , at a given configuration, as the product of two matrices

$$\dot{f} = J_{\text{img}}(f, \chi, q)u = J_v(f, \chi)J_c(q)u, \quad (3)$$

where

- the  $6 \times p$  matrix  $J_c$  gives the linear and angular velocity pair  $(v_C, \omega_C)$  of the camera mounted on the end-effector (expressed in the camera frame), in response to a velocity command  $u$ ;
- the  $s \times 6$  interaction matrix  $J_v$ , defined in (2), describes how features move on the image plane as a consequence of the camera motion.

Note that evaluation of matrix  $J_v(f, \chi)$  requires knowledge of vector  $\chi(t)$  which contains unmeasurable, in general time-varying, 3D quantities like, e.g., the depths  $Z_i$  of point features  $p_i$ . While a typical solution is to approximate  $\chi(t)$  with the constant value  $\chi_d$  relative to the desired robot final pose, another possibility is to replace  $\chi(t)$  with an estimate  $\hat{\chi}(t)$  obtained on-line during the robot motion, as explained in the next subsection.

### C. Depth estimation for IBVS

The 3D information represented by  $\chi$  largely depends on the particular feature extracted from the considered geometric structure. In the case of a point  $P = [X \ Y \ Z]^T$  (the most simple shape),  $\chi$  reduces to the unknown depth  $Z$  while, for more complex shapes, additional quantities are required, like radii of spheres, plane orientations of planar shapes, and so on. In all cases, however, depth is always present in  $\chi$  even if not in an explicit way as in the case of a point feature.

Among the various approximations, one interesting possibility is to obtain an estimation  $\hat{\chi}(t)$  to be used in place of  $\chi(t)$  during the servoing. In [18], we proposed an on-line observer able to estimate the unknown depth  $Z$  of a point feature  $p$  during the camera motion. In particular, by taking  $\hat{x} = [\hat{p}_u \ \hat{p}_v \ 1/\hat{Z}]^T$  as the current estimation of the partially unknown state  $x = [p_u \ p_v \ 1/Z]^T$  and  $y = [p_u \ p_v]^T$  as the measured output on the image plane, the update law takes

the form of a nonlinear observer

$$\dot{\hat{x}} = \alpha(\hat{x}, y) \begin{bmatrix} v_C \\ \omega_C \end{bmatrix} + \beta(\hat{x}, y, v_C, \omega_C), \quad (4)$$

where suitable  $\alpha(\cdot)$  and  $\beta(\cdot)$  guarantee that  $\lim_{t \rightarrow \infty} \|x(t) - \hat{x}(t)\| = 0$  exponentially, as long as

- 1) the camera is moving with a non-zero linear velocity  $v_C$ ;
- 2) the camera is not translating along the projection ray of point  $p$ .

Therefore, thanks to this result, we will assume that the actual value of  $Z$  of each point feature is reconstructed on-line and made available during the servoing task.

### III. KINEMATIC CONTROL SCHEMES UNDER REDUNDANCY

Assume a reference signal  $f_d$  is specified for the task variables  $f$ . If  $f_d = f_d(t)$ , one has a *tracking* problem, whereas a *regulation* problem is assigned when  $f_d$  is constant. In the following, only regulation tasks will be considered. Furthermore, we shall use  $u$  as the control input for the robot, assuming that low-level (direct) controllers are in charge of imposing the corresponding velocities to the system, complying at a fast scale with the neglected dynamics — a classical approach known as *kinematic control*.

Consider first the case  $p = s$ , in which the image Jacobian in (3) is square. Outside singularities, we can realize the given task by choosing<sup>1</sup>

$$u = J_{\text{img}}^{-1} K (f_d - f) = J_{\text{img}}^{-1} K e, \quad K > 0, \quad (5)$$

which yields a closed-loop exponential convergence to zero for each component of task error  $e$ , i.e.,  $\dot{e} = -K e$ .

When  $p > s$ , so that  $J_{\text{img}}$  has more columns than rows, the robot is said *redundant* w.r.t. the given task. In the literature, a number of techniques has been proposed to exploit redundancy during the task execution [9]. In the following, we will briefly summarize the Projected Gradient (PG), Task Priority (TP), and Task Sequencing (TS) algorithms. More details on this topic can also be found in [14], [17] where the use of such technique for the cases of NMMs performing IBVS tasks are discussed in detail.

In the classical PG method, robot commands are computed as

$$u = J_{\text{img}}^\dagger K e + (I - J_{\text{img}}^\dagger J_{\text{img}}) u_0 \quad (6)$$

where  $J_{\text{img}}^\dagger$  is the unique pseudoinverse of the image Jacobian,  $I - J_{\text{img}}^\dagger J_{\text{img}}$  is the orthogonal (and symmetric) projection operator into the null-space of  $J_{\text{img}}$ , and  $u_0 \in \mathbb{R}^p$  is an arbitrary vector usually chosen so as to optimize a given criterion  $H(q)$ . In general,  $J_{\text{img}}^\dagger$  can be computed from the Singular Value Decomposition of  $J_{\text{img}}$ . If  $\text{rank } J_{\text{img}} = s$ , it is  $J_{\text{img}}^\dagger = J_{\text{img}}^T (J_{\text{img}} J_{\text{img}}^T)^{-1}$  [19]. Note that, as before, (6) yields an exponentially convergent closed-loop behavior for  $e$ , with the additional presence of an internal motion in the

robot structure which is unobservable at the output level, i.e., in the task space.

In the neighborhood of singular points of the image Jacobian, the use of the PG method (6) may result in very high velocity commands which cannot be realized by the low-level actuator controllers. One way to deal with this problem is to use the *Task Priority* (TP) technique<sup>2</sup> [13]. Reorder the task vector  $f$  into  $\mu$  subtasks  $(f_1, \dots, f_\mu)$ , each of dimension  $s_i$  ( $i = 1, \dots, \mu$ , with  $\sum s_i = s$ ), and consider  $f_i$  as a task with higher priority than  $f_j$  if  $i < j$ . In the case of  $\mu = 2$  subtasks, we have

$$\begin{bmatrix} \dot{f}_1 \\ \dot{f}_2 \end{bmatrix} = \begin{bmatrix} J_{\text{img}_1} \\ J_{\text{img}_2} \end{bmatrix} u. \quad (7)$$

The robot commands are then generated as

$$u = J_{\text{img}_1}^\dagger K_1 e_1 + (I - J_{\text{img}_1}^\dagger J_{\text{img}_1}) J_{\text{img}_2}^T K_2 e_2, \quad (8)$$

where  $K_1 > 0$  and  $K_2 > 0$  are suitable gains, and  $e_1$  and  $e_2$  the subtask errors on  $f_1$  and  $f_2$ .

An alternative way to drive all task variables to their set-points is to follow a *Task Sequencing* (TS) approach. The idea is to process the  $\mu$  subtasks one at a time. The  $s_1$  subtask variables  $r_1$  are regulated first, then the  $s_2$  subtask variables  $r_2$  are driven toward their desired value without changing  $r_1$ , then the  $s_3$  subtask variables  $r_3$  are brought to their desired value without changing  $r_1$  and  $r_2$ , and so on. With this approach, an ‘artificial’ redundancy is introduced in the kinematic control process: in particular, the degree of redundancy during the execution of the sequence will be  $p - s_1$  in the first phase,  $p - (s_1 + s_2)$  in the second phase, and so on. In the last phase, the redundancy degree will be back to its (possibly zero) original value  $p - s \geq 0$ .

This method applies to robots that are either redundant or non-redundant w.r.t. the task. For instance, consider the non-redundant case  $p = s$  and a decomposition of task  $f$  into  $\mu = 2$  subtasks  $f_1, f_2$ . The associated two-phase command sequence is

$$\begin{cases} u_I = J_{\text{img}_1}^\dagger K_1 e_1 + (I - J_{\text{img}_1}^\dagger J_{\text{img}_1}) u_0, & t \in [0, T_1] \\ u_{II} = (I - J_{\text{img}_1}^\dagger J_{\text{img}_1}) J_{\text{img}_2}^T K_2 e_2 + J_{\text{img}_1}^\dagger K_1 e_1, & t \in [T_1, T_2], \end{cases} \quad (9)$$

where  $u_0$  can be chosen for optimization purposes as in (6). The switching times  $T_i$  ( $i = 1, 2$ ) denote the completion of subtask  $i$ , i.e., such that a suitable termination condition is reached.

## IV. EXPERIMENTAL STUDY

### A. Robot and visual task description

The experiments considered in this paper have been conducted on the MagellanPro, a unicycle-like WMR equipped with a pan-tilt camera (Fig. 2(a)). The results can also be appreciated in the video clip attached to the paper. By considering the pan-tilt unit equivalent to a (polar) 2R manipulator (Fig. 2(b)), this robotic system falls into the class on NMMs.

<sup>1</sup>Dependence of Jacobian matrices on  $q$  is dropped from now on for compactness.

<sup>2</sup>We use here the result in [13] which is a modified version of the original task priority method (see [9]).

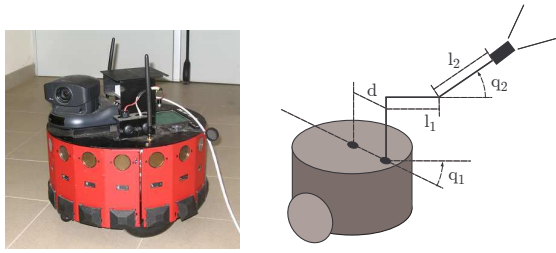


Fig. 2: The robot used in our experiments. Left: MagellanPro with the onboard pan-tilt camera. Right: schematic view of the robot.

Therefore, its kinematic model, and in particular matrix  $J_c$  in (3), can be obtained through the techniques developed in [14], [17]. To this end, let the configuration vector  $q$  be partitioned as  $q = [q_p \ q_m]^T \in \mathbb{R}^n$ , where  $q_p = [x \ y \ \theta]^T \in \mathbb{R}^3$  represents the mobile platform configuration (position and orientation), and  $q_m = [q_1 \ q_2]^T \in \mathbb{R}^2$  the ‘manipulator’ joint variables (i.e., pan and tilt of the camera). Hence, in this case  $n = 5$  generalized coordinates are needed to describe the configuration of the robot. Let  $u = [u_p \ u_m]^T \in \mathbb{R}^p$  be the induced partition of the robot velocity commands and assume  $\dot{q}_m = u_m$ , i.e., that any pan/tilt velocity can be arbitrarily specified. On the other hand, the kinematic model of the nonholonomic mobile platform can be expressed as the driftless system

$$\dot{q}_p = G(q_p)u_p = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (10)$$

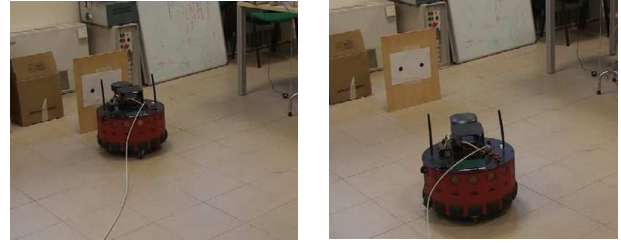
with  $u_p = [v \ \omega]^T$  being the linear/angular velocity of the platform. Because of the platform restricted instantaneous mobility, the total number of velocity commands is  $p = 4$ , less than the number of generalized coordinates ( $n = 5$ ).

For NMM systems with eye-in-hand cameras, the relation between linear/angular camera velocity and the commands can be written as

$$\begin{bmatrix} v_C \\ \omega_C \end{bmatrix} = J_p(q)\dot{q}_p + J_m(q)\dot{q}_m = \begin{bmatrix} J_p(q)G(q_p) & J_m(q) \end{bmatrix} \begin{bmatrix} u_p \\ u_m \end{bmatrix} = J_c(q)u. \quad (11)$$

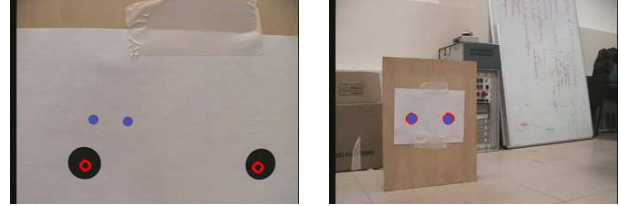
However, it is possible to prove that matrix  $J_c$  is independent of the platform absolute position/orientation  $q_p$ , depending only on manipulator variables  $q_m$  [14]. As a result, the computation of  $J_c(q_m)$  is not affected by the mobile base absolute localization, which is typically obtained from noisy and possibly unreliable data processing algorithms (such as dead-reckoning). For our robot, it is

$$J_c(q_m) = \begin{bmatrix} s_1 & -dc_1 - l_1 - l_2c_2 & -l_1 - l_2c_2 & 0 \\ c_1s_2 & ds_1s_2 & 0 & -l_2 \\ c_1c_2 & ds_1c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -c_2 & -c_2 & 0 \\ 0 & s_2 & s_2 & 0 \end{bmatrix}, \quad (12)$$



(a) Initial external view.

(b) Final external view.



(c) Initial camera view.

(d) Final desired camera view.

Fig. 3: Initial and final robot views for the TP experiment.

where  $c_i$  and  $s_i$  stand for  $\cos q_i$  and  $\sin q_i$ , and the geometric quantities ( $d = 0.135$  m,  $l_1 = 0.011$  m,  $l_2 = 0.0233$  m) are defined in Fig. 2(b).

As visual task, we considered the regulation of two point features  $p_1, p_2$  on the image plane. Thus, using normalized image coordinates,

$$f = [p_{u_1} \ p_{v_1} \ p_{u_2} \ p_{v_2}]^T \in \mathbb{R}^s, \quad s = 4, \quad (13)$$

and

$$J_v(f, \chi) = \begin{bmatrix} J_P(p_1, Z_1) \\ J_P(p_2, Z_2) \end{bmatrix}, \quad (14)$$

where the  $2 \times 6$  interaction matrix  $J_P$  for a single feature point is

$$J_P = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{p_u}{Z} & p_u p_v & -(1 + p_u^2) & p_v \\ 0 & -\frac{1}{Z} & \frac{p_v}{Z} & 1 + p_v^2 & -p_u p_v & -p_u \end{bmatrix}.$$

The image Jacobian  $J_{\text{img}}$  in (3) can then be evaluated from (12) and (14). Note that the dimension of the visual task defined in (13) matches the number of available robot commands ( $s = p = 4$ ), so that the obtained image Jacobian  $J_{\text{img}}$  is a square  $4 \times 4$  matrix.

### B. Experiments with Task Priority

Although the task could be realized by direct inversion of  $J_{\text{img}}$  as in (5), lack of redundancy may cause poor results when  $J_{\text{img}}$  is close to ill-conditioning. Consider, for example, the situation shown in Fig. 3(a)–(d) corresponding to the following initial conditions:

$$\begin{cases} f(t_0) = [-0.4319 \ 0.3803 \ 0.6580 \ 0.3893]^T \text{ m} \\ Z_1(t_0) = Z_2(t_0) = 0.21 \text{ m} \\ q_1(t_0) = 0.23 \text{ rad} \\ q_2(t_0) = -0.07 \text{ rad.} \end{cases}$$

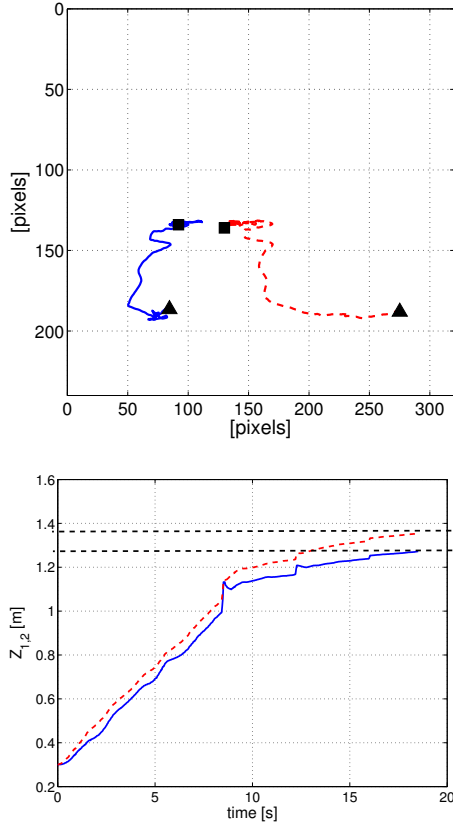


Fig. 4: TP Experiment. Top: motion, from  $\blacktriangle$  to  $\blacksquare$ , of the two feature points on the image plane ( $p_1$  is the blue solid line and  $p_2$  the red dashed line). Bottom: behavior of the estimated depths  $\hat{Z}_1(t)$  and  $\hat{Z}_2(t)$  over time. The dashed horizontal lines represent the final ground truth values of the depths.

In this configuration, matrix  $J_{\text{img}}$  is very close to being singular and its inversion would generate very large velocity commands for the robot. However, if the whole task is split as in (7) with  $f_1 = p_1$  and  $f_2 = p_2$ , the two  $2 \times 4$  sub-Jacobians  $J_{\text{img}_1}$  and  $J_{\text{img}_2}$  relative to the individual feature points are well conditioned. Indeed, we have  $\sigma(J_{\text{img}}) \sim 0.15$ ,  $\sigma(J_{\text{img}_1}) \sim 227$  and  $\sigma(J_{\text{img}_2}) \sim 246$ , where  $\sigma(A)$  denotes the smallest singular value of matrix  $A$ . Therefore, it is interesting to test the performance of the TP method for the realization of task (13).

Figures 4(a)–(b), 5(a)–(b) and 6 show the results of the servoing realized with the TP control law (8) where we set  $K_1 = 2$ ,  $K_2 = 0.001$ ,  $f_d = [-0.3889 \ 0.0801 \ -0.1716 \ 0.0915]^T$ . We also used  $v_{\text{max}} = 0.09$  m/s,  $\omega_{\text{max}} = \dot{q}_{1\text{max}} = \dot{q}_{2\text{max}} = 0.14$  rad/s as maximum allowed velocity commands. The task  $f$  is correctly executed (Fig. 4(a)), with simultaneous motion commanded to the platform (Fig. 5(a)) and the pan-tilt unit (Fig. 5(b)), despite the ill-conditioning of  $J_{\text{img}}$ . Figure 6 reports the time behavior of  $\sigma(J_{\text{img}})$ ,  $\sigma(J_{\text{img}_1})$  and  $\sigma(J_{\text{img}_2})$ : matrix  $J_{\text{img}}$  remains always close to singularity, while  $J_{\text{img}_1}$  and  $J_{\text{img}_2}$  are well conditioned during the whole motion. Furthermore, Fig. 4(b)

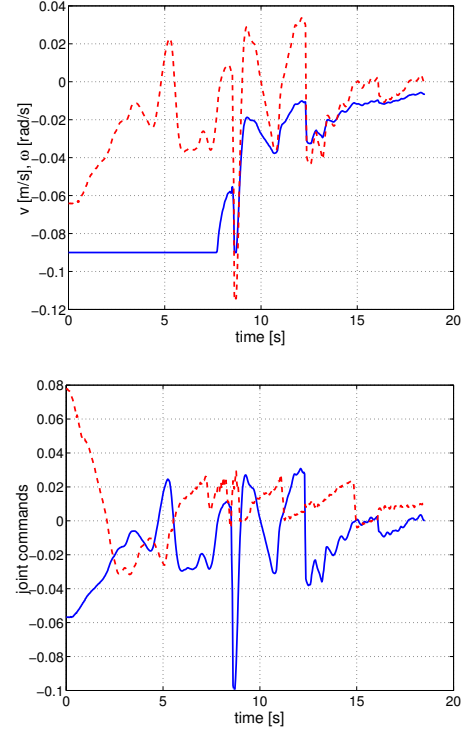


Fig. 5: TP Experiment. Top: platform linear velocity  $v$  (solid blue line) and angular velocity  $\omega$  (dashed red line). Bottom: pan velocity  $\dot{q}_1$  (solid blue line) and tilt velocity  $\dot{q}_2$  (dashed red line).

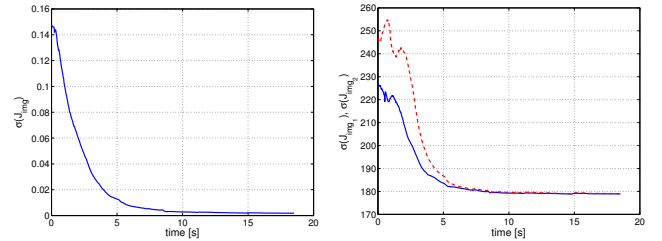


Fig. 6: Singularity analysis during the visual task. Top: time evolution of  $\sigma(J_{\text{img}})$ . Bottom: time evolution of  $\sigma(J_{\text{img}_1})$  (solid blue line) and  $\sigma(J_{\text{img}_2})$  (dashed red line). Note the different scales in the two plots.

shows the behavior of the two feature depths estimated via the observer (4), initialized with  $\hat{Z}_1(t_0) = \hat{Z}_2(t_0) = 0.3$  m. There is a good convergence towards the final real depth values represented by two dashed horizontal lines. Finally note that the task is executed in about 18 sec with a backward motion and a small clockwise rotation of the platform. The slightly erratic commands are a consequence of the slow sampling rate of the control architecture (30 Hz) and the presence of unmodeled disturbances due to the gaps between the floor tiles.

### C. Experiments with Task Sequencing

The TS method was tested by decomposing the visual task (13) in two phases, being  $f_1 = p_1$  the variables regulated

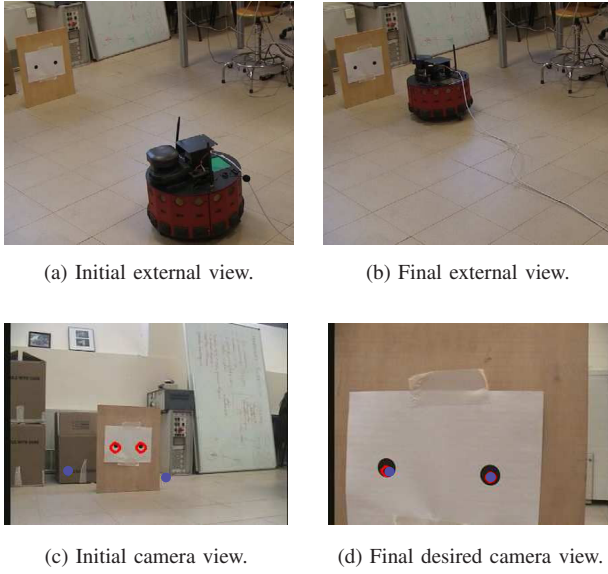


Fig. 7: Initial and final robot views for the TS experiment.

in the first phase and  $f_2 = p_2$  in the second phase — see (9). This choice allows to gain two degrees of redundancy during the first phase that can be exploited to meet additional requirements. In our case, we chose to use redundancy in order to keep the target as much as possible in front of the robot. This was obtained by minimizing the cost function  $H(q) = \frac{1}{2}q_1^2$ , i.e., the pan angle distance from the platform forward direction. Special care is required when specifying vector  $u_0$  in (9) with the aim of optimizing a given criterion  $H(q)$ . Indeed, while for standard manipulators one can set  $u_0 = \pm\alpha\nabla_q H(q)$ , where  $\alpha > 0$  is a suitable stepsize, in the NMM case the choice that locally realizes the best improvement of  $H(q)$  is

$$u_0 = \pm\alpha \begin{bmatrix} G^T(q_p) & 0 \\ 0 & I \end{bmatrix} \nabla_q H(q). \quad (15)$$

Note that, dependence of  $u_0$  from  $q_p$  in (15) is avoided in our case since, for the chosen  $H(q)$ , it is  $\nabla_{q_p} H(q) \equiv 0$ . The first phase switching time  $T_1$  is set according to the value of  $q_1$ , so that the switch occurs when  $|q_1(t)| < 0.1 \text{ rad}^3$ .

Initial and final conditions of the robot are shown in Figs. 7(a)–(d). As can be seen from Fig. 7(a), the initial pan angle was intentionally set at about  $-90$  deg (with the platform counter-rotated w.r.t. the target by a similar amount) in order to fully appreciate the null-space motion during the first phase. The initial conditions of the experiment are

$$\begin{cases} f(t_0) = [-0.1944 \ 0.1087 \ -0.0343 \ 0.1087]^T \text{ m} \\ Z_1(t_0) = Z_2(t_0) = 1.7 \text{ m} \\ q_1(t_0) = -1.68 \text{ rad} \\ q_2(t_0) = 0 \text{ rad}, \end{cases}$$

and we set in (9)  $K_1 = 1.5$ ,  $K_2 = 0.001$ ,  $\alpha = 1.5$ , and  $f_d = [-0.509 \ 0.2631 \ 0.1258 \ 0.3031]^T$ . The maximum command values were chosen as in the previous case. Results

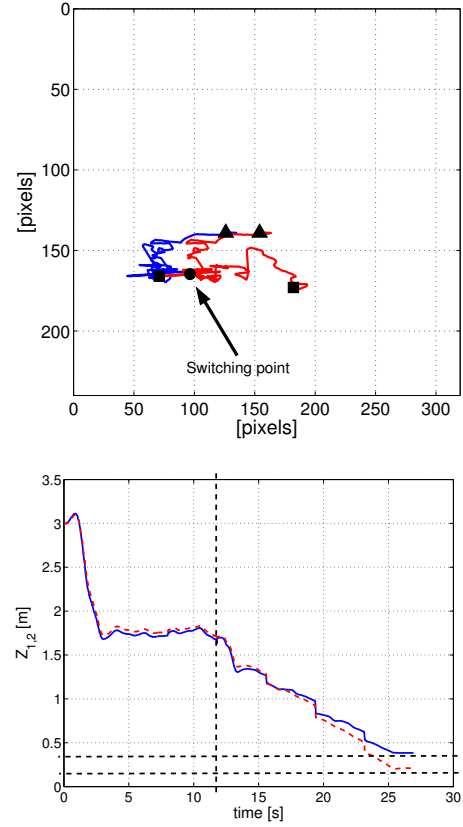


Fig. 8: TS Experiment. Top: motion, from  $\blacktriangle$  to  $\blacksquare$ , of the two feature points on the image plane ( $p_1$  is the blue solid line and  $p_2$  the red dashed line). The black circle indicates the switching point between the two phases. Bottom: behavior of the estimated depths  $\hat{Z}_1(t)$  and  $\hat{Z}_2(t)$  over time. The dashed horizontal lines represent the final ground truth values of the depths.

of the experiment are shown in Figs. 8(a)–(b), 9(a)–(b) and 10. During the first phase, angle  $q_1$  is brought back to zero (Fig. 10) and feature point  $p_1$  is kept close to its desired value, but no direct control is applied to the motion of  $p_2$  (Fig. 8(a)). After the switching instant (represented in the plots by a dashed vertical line), the desired final position of feature point  $p_2$  is recovered while keeping  $p_1$  fixed to its reached desired location. Figures 9(a)–(b) show the velocity commands sent to the robot during the task execution. It is evident there the discontinuity due to the switch between the two phases. In particular, during the first phase the platform moves backwards and rotates clockwise in order to compensate for the counterclockwise pan motion; in the second phase, the platform mainly moves forward while slightly rotating to keep feature  $p_1$  fixed on the image plane. Finally, in Fig. 8(b) the behavior of the estimated depths is reported. The observer was initialized with  $\hat{Z}_1(t_0) = \hat{Z}_2(t_0) = 3 \text{ m}$  with an error of about 1.3 m from the actual initial depths (measured independently and not used in the experiment). Despite this rough approximation, the observer is able to recover the true values of  $Z_{1,2}$  in about 3 sec of

<sup>3</sup>A termination condition on  $e_1$  could also be added.

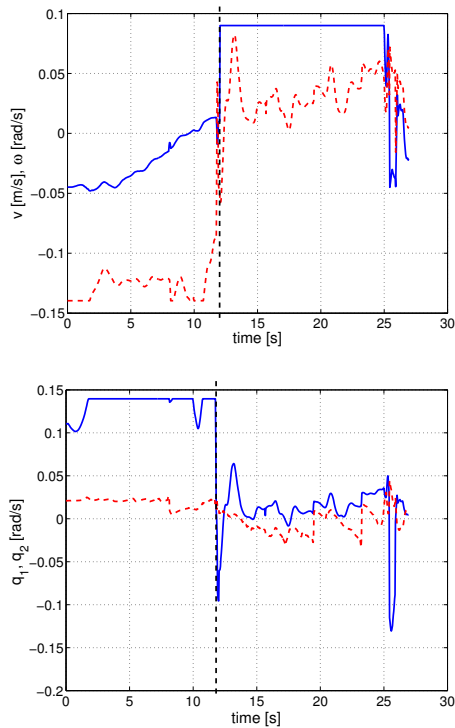


Fig. 9: TS Experiment. Top: platform linear velocity  $v$  (solid blue line) and angular velocity  $\omega$  (dashed red line). Bottom: pan velocity  $\dot{q}_1$  (solid blue line) and tilt velocity  $\dot{q}_2$  (dashed red line).

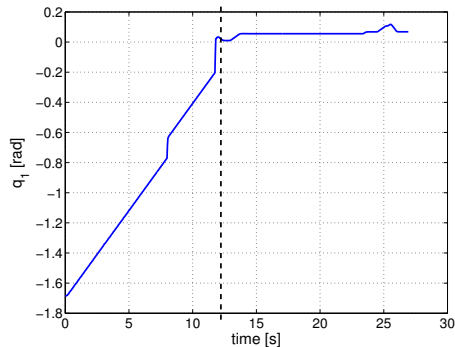


Fig. 10: Behavior of  $q_1(t)$  over time. The vertical dashed line indicates the switching point.

motion (the fast initial transient on Fig. 8(b)), and yields, at the end of the task, depth estimates close to the real final depth values (the dashed horizontal lines).

## V. CONCLUSIONS

In this work we analyzed how redundancy can be exploited to improve the fulfillment of standard IBVS tasks for eye-in-hand robotic systems. In particular, redundancy can help in realizing tasks that would be close to singularity if addressed altogether, or can be exploited to meet secondary requirements during task execution. In this respect, we presented the experimental validation of two redundancy resolution schemes, Task Priority and Task Sequencing, applied to a unicycle-like mobile platform carrying onboard a pan-tilt unit. The obtained results showed the effectiveness of both

techniques against noise and unmodeled effects present in realistic conditions, as well as the ability to correctly execute regulation tasks for which the inversion scheme (5) would be ill-conditioned. In particular, with the TP method it was possible to simultaneously regulate all the task variables despite the rank deficiency of the Jacobian  $J_{img}$ . However, no additional requirements could be specified for the motion execution. On the other hand, the TS approach allowed such a possibility thanks to the ‘artificial’ redundancy introduced in the first phase, at the expense of a longer execution time for completing the original task. Future work will consider the use of more general visual features (e.g., moments) and the exploitation of redundancy for visual tracking of a moving target, such as a ball rolling on the plane.

## REFERENCES

- [1] L. Weiss, A. C. Sanderson, and C. P. Neuman, “Dynamic sensor-based control of robots with visual feedback,” *IEEE J. on Robotics and Automation*, vol. RA-3, no. 5, 1987.
- [2] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, “Relative end-effector control using cartesian position based visual servoing,” *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.
- [3] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.
- [4] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [5] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *IEEE Robotics & Automation Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
- [6] —, “Visual servo control. II. Advanced approaches,” *IEEE Robotics & Automation Mag.*, vol. 14, no. 1, pp. 109–118, 2006.
- [7] B. Espiau, “Effect of camera calibration errors on visual servoing in robotics,” pp. 182–192, 1993.
- [8] P. I. Corke and S. A. Hutchinson, “A new partitioned approach to image-based visual servo control,” *IEEE Trans. on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.
- [9] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, 1991.
- [10] F. Chaumette and E. Marchand, “A redundancy-based iterative approach for avoiding joint limits: application to visual servoing,” *IEEE Trans. on Robotics and Automation*, vol. 17, no. 5, pp. 719–730, 2001.
- [11] N. Mansard and F. Chaumette, “Directional redundancy: a new approach of the redundancy formalism,” *Proc. 2005 IEEE Conf. on Decision and Control*, pp. 5366–5371, 2005.
- [12] —, “Task sequencing for high-level sensor-based control,” *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.
- [13] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, “Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy,” *Int. J. of Robotics Research*, vol. 10, pp. 410–425, 1991.
- [14] A. De Luca, G. Oriolo, and P. Robuffo Giordano, “Image-based visual servoing schemes for nonholonomic mobile manipulators,” *Robotica*, vol. 25, no. 2, pp. 131–145, 2007.
- [15] F. Chaumette, “Image moments: A general and useful set of features for visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 4, pp. 713–723, 2004.
- [16] O. Tahri and F. Chaumette, “Point-based and region-based image moments for visual servoing of planar objects,” *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.
- [17] A. De Luca, G. Oriolo, and P. Robuffo Giordano, “Kinematic modeling and redundancy resolution for nonholonomic mobile robots,” pp. 1867–1873, 2006.
- [18] —, “On-line estimation of feature depth for image-based visual servoing schemes,” *Proc. 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2823–2828, 2007.
- [19] A. A. Maciejewski and C. A. Klein, “The singular value decomposition: Computation and applications to robotics,” *Int. J. of Robotics Research*, vol. 8, no. 6, pp. 63–79, 1989.