

# Modeling and Control of Sampled-Data Image-Based Visual Servoing With Three-Dimensional Features

Marco Costanzo<sup>✉</sup>, Member, IEEE, Giuseppe De Maria<sup>✉</sup>, <sup>✉</sup>Ciro Natale<sup>✉</sup>, and Antonio Russo<sup>✉</sup>

**Abstract**—Image-based visual servoing (IBVS) with 3-D features refers to the use of 3-D feature vectors in the visual control. The availability of low-cost and lightweight RGB-D cameras makes it natural to use 3-D point coordinates of the acquired image to construct the feature vector. In this article, by using 3-D features, we propose, first, a novel sampled-data model of the feature dynamics, which, in contrast to the usual forward Euler approximation, retains the rigid motion constraint. The need to introduce the sampled-data model arises from the fact that, due to the limited camera frame rate and actuation delays, the effects of a finite sampling time of the visual control system cannot be neglected. The stability analysis of the resulting discrete-time control system is carried out, showing that the desired equilibrium point of the visual control system is *almost globally asymptotically stable*. Finally, a new IBVS control algorithm is designed by resorting to the Lyapunov direct method. It explicitly takes into account the camera velocity limits while ensuring stability at the same time. Furthermore, despite large sampling times, it guarantees the absence of hidden oscillations and a smooth approach of the camera to the prescribed target configuration. The experiments are carried out in an emulated in-store logistic scenario by performing pick&place and object handover tasks, testifying to the effectiveness of the approach.

**Index Terms**—Sampled-data control, stability of nonlinear systems, visual servo control.

## I. INTRODUCTION

**V**ISUAL control schemes use visual feedback from a camera to navigate a mobile robot or to position the robot end-effector of a manipulator. This objective is achieved by extracting features from the camera image. Features used in visual servoing can be points, lines, and contours (see [1], [2], [3], and references therein) or 3-D features generated by combining 2-D and 3-D information [4], [5]. If these are used to estimate the camera pose directly in the 3-D space, the control strategy is called position-based visual servoing (PBVS), and it aims at minimizing the error between the current and target poses. On the other hand, the so-called image-based visual servoing (IBVS) aims at directly bringing the current image features to the target ones. Reference

tutorials for such approaches are [6], [7], and [8]. The IBVS approach is usually preferred to the PBVS one, owing to its robustness against camera calibration errors. IBVS controllers require the extraction of visual features from the camera image and their matching with the features of the given target image. Then, the tracking module provides the visual controller with the feature vector frame by frame. To avoid the bottleneck of feature extraction and matching steps, the so-called dense approaches have been proposed in the literature. For instance, the photometric approach [9] is based on the luminance of all the pixels in the figure and requires stable lighting conditions. Dense depth map strategies [10], [11] require accurate camera calibration to provide satisfactory accuracy in the camera positioning.

For large camera transformations, the feature-based visual control results in inaccurate matching. To overcome such a difficulty, data-driven visual servoing approaches have been proposed recently. Deep neural network-based methods [12], [13], [14] are adopted to estimate the relative pose between the current and desired images and to perform the camera positioning. The mentioned methods generalize to novel and dynamic scenes [15], but the camera positioning accuracy is still unsatisfactory, especially in dynamic scenes [16].

In this article, with reference to applications where pick&place and handover of objects are dominant, as in intralogistic scenarios, we will adopt the 3-D IBVS control scheme because it guarantees high camera positioning accuracy in the short range of interaction among the actors of a shared task and the object to handle, and last but not least, it is robust against the changing of the scene over time if the visibility constraint is satisfied.

The feature-based visual control strategies proposed in the literature (refer to [17] for an up-to-date textbook on visual servoing) are based on the continuous-time model describing the relationship between the feature rate and the camera velocity screw. The control law aims at giving the feature error a decoupled and monotonically decreasing behavior by acting on the camera velocity as input to the robot controller. An interesting first discussion on stability and convergence problems of such control strategies can be found in [18]. In [19], with reference to the IBVS with 3-D features, the equilibrium points of the 3-D IBVS closed-loop system are identified, and the instability of the undesired equilibrium points is proven by exploiting the isomorphism between  $SE(3)$  and the subset of the feature space to which the feature vector

Manuscript received 19 July 2022; revised 8 May 2023 and 9 May 2023; accepted 1 July 2023. Date of publication 11 July 2023; date of current version 29 December 2023. Recommended by Associate Editor J. B. Hoagg. (Corresponding author: *Ciro Natale*.)

The authors are with the Dipartimento di Ingegneria, Università degli Studi della Campania Luigi Vanvitelli, 81031 Aversa, Italy (e-mail: marco.costanzo@unicampania.it; giuseppe.demaria@unicampania.it; ciro.natale@unicampania.it; antonio.russo1@unicampania.it).

Digital Object Identifier 10.1109/TCST.2023.3292311

must belong to comply with the rigid motion constraint. This result has been extended in [20], where the basin of attraction of such undesired equilibria has been characterized for the classical continuous-time visual servoing scheme.

Due to the limited camera frame rate, actuation delays, and the time needed for computations, the sampling time of the visual control system is not negligible, and its intrinsic sampled-data nature should be taken into account to pursue the objective of improving the visual servoing quality and robotic system overall performance, by ensuring stability at the same time, as it is required in specific scenarios. For instance, in human–robot collaborative scenarios, the collaboration acceptability and the optimal sharing of the task require that the robot is able to perform actions at a speed comparable to that of the human partner. Costanzo et al. [21], [22] present the shelf replenishment task performed by the robot autonomously or in a collaborative way, respectively, by using standard algorithms and libraries for object detection [23] and visual servoing [24].

A first discrete-time analysis and design of a visual control scheme aimed at regulating the relative pose between a robotic camera and a rigid object can be found in [25]. The authors use the forward Euler finite-difference approximation to formulate a discrete-time approximation of the visual servoing system model. Other discrete-time design approaches refer to multirate visual control [26], or they resort to model predictive control (MPC) strategies. MPC techniques are used with the aim to improve visual servoing performance and to take into account work space and robotic system constraints in the control law design. In [27], three sets of constraints are considered, namely: 1) robot mechanical limitations, i.e., work space limits and joint speed saturation; 2) visibility constraint, to ensure that the visual measurements stay in the image plane; and 3) control constraints, such as actuator limitations. Recently, a lightweight MPC architecture using a slim neural network, which can be trained on the fly, is proposed in [14] to solve the IBVS objective of minimizing the error between the current and target images. In [28], the model predictive path integral (MPPI), developed for autonomous robot navigation tasks, is proposed for coping with the PBVS control goal. The cited discrete-time designs are based on the approximation of the feature dynamic model derived via the forward Euler method. However, the obtained discrete-time model does not retain the rigid motion constraint, which is the base of the continuous-time model. Furthermore, the discrete-time IBVS model obtained via the forward Euler method approximates the continuous-time model with a relatively good degree of accuracy only in a neighborhood of the desired equilibrium point. This, in turn, implies that the stability analysis can be performed only locally.

In this article, we first propose an exact nonlinear sampled-data model of the feature dynamics by suitably exploiting the properties of the interaction matrix. Then, we inspect the stability properties of the error dynamics equilibrium points. Extending the results presented in [19] and [20] to the sampled-data case, we prove that, for a suitable choice of the sampling time and control parameters, the sampled-data IBVS closed-loop system exhibits a desired

equilibrium point being *almost globally asymptotically stable* [29], implying that the only trajectories not converging to it are those belonging to zero Lebesgue measure sets. Based on the results on the stability of the sampled-data visual control system specifically developed in this article, we propose a novel control algorithm aimed at improving the accuracy and execution speed of the visual servoing task. The algorithm is based on the discrete-time Lyapunov analysis, and it allows the designer to guarantee end-effector Cartesian velocity constraints, depending on the robot configuration and joint velocity limits. Moreover, the algorithm enforces the constraint of monotonic convergence of the feature error over time, i.e., it does not exhibit any hidden oscillations, and it guarantees a prescribed smooth approach of the camera toward the target configuration. These control objectives are achieved through adaptation of the control gain obtained as the solution of a constrained optimization problem.

The proposed IBVS control schemes are experimentally evaluated in a lab-scale emulated in-store logistic scenario, where pick&place operations for shelf replenishment are executed by a robot in collaboration with a human partner, by utilizing object handover and dexterous handling maneuvers [30].

## II. IBVS MODEL

In this section, to make the notation used in this article self-consistent, we recall notations and the continuous-time dynamics of the IBVS taken from [20], necessary to derive the corresponding exact nonlinear sampled-data model.

### A. Continuous-Time Model

The IBVS model is based on the data provided by RGB-D cameras, consisting of RGB images and per-pixel depth information. Let  $\mathbf{p}_i$  be the  $i$ th point on the image, captured by the camera, having pixel coordinates  $(u_i, v_i)^\top$ , and let  $z_i$  be the corresponding depth. Denote with  $\mathbf{K}$  the camera calibration matrix

$$\mathbf{K} = \begin{bmatrix} F_x & 0 & C_x \\ 0 & F_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $C_x$  and  $C_y$  represent the center of the camera in pixel coordinates, and by denoting with  $f$  the focal distance and with  $l_x$  and  $l_y$  the width and the height of the pixel,  $F_x = f/l_x$  and  $F_y = f/l_y$ . Then, the point  $\mathbf{p}_i$  in 3-D coordinates with respect to the camera frame is expressed as

$$\mathbf{p}_i = \mathbf{K}^{-1} \begin{bmatrix} u_i z_i \\ v_i z_i \\ z_i \end{bmatrix}. \quad (2)$$

We assume that the 3-D feature vectors of the captured RGB image, indicated hereafter with  $\mathbf{s}_i$ , are composed of the 3-D coordinates of the points  $\mathbf{p}_i$ , i.e.,  $\mathbf{s}_i = \mathbf{p}_i$ . The IBVS approach aims at designing a closed-loop control scheme that regulates the current image feature set to a given target set of an image acquired beforehand. Let  $\mathbf{s}_i, \mathbf{s}_i^* \in \mathbb{R}^3, i = 1, \dots, n$ , be the

$i$ th current and target feature vectors, respectively; then, the vectors  $s, s^* \in \mathbb{R}^{3n}$

$$s = [s_1^\top \ \cdots \ s_n^\top]^\top \quad (3)$$

$$s^* = [s_1^{*\top} \ \cdots \ s_n^{*\top}]^\top \quad (4)$$

represent the vectors of  $n$  matched image features.

Denote with  $v = [v^\top \ \omega^\top]^\top \in \mathbb{R}^6$  the body velocity screw of the camera, where  $v \in \mathbb{R}^3$  is the linear velocity and  $\omega \in \mathbb{R}^3$  is the angular velocity. Then, the link between the rate of  $s(t)$  and  $v(t)$  is described by the following relationship:

$$\dot{s}(t) = L(s)v(t), \quad s(0) = s_0 \quad (5)$$

where  $L(s) \in \mathbb{R}^{3n \times 6}$  is the so-called *interaction matrix*, and due to the particular feature set selected, it takes the structure

$$L(s) = \begin{bmatrix} -I_3 & \cdots & -I_3 \\ S^\top(s_1) & \cdots & S^\top(s_n) \end{bmatrix}^\top \quad (6)$$

where  $I_3$  indicates the  $3 \times 3$  identity matrix and  $S(s_i)$  is the skew operator applied to the vector  $s_i$  such that  $S(s_i)\omega = s_i \times \omega$ .

Assuming that there exist at least three distinct nonaligned features, the matrix  $L(s)$  has full rank, i.e.,  $\text{rank}(L(s)) = 6$ .

By defining the error feature vector  $e \in \mathbb{R}^{3n}$  as

$$e(t) = s(t) - s^* \quad (7)$$

then the distance between the current feature and the reference one is represented by  $\|e(t)\|$ . Moreover, denoting with  $L_e = L(e + s^*)$ , the dynamics of the current feature deviation from the target one can be easily derived from (5) and (7) as

$$\dot{e}(t) = L_e v(t). \quad (8)$$

Hereafter, we will drop the dependence on the time for the sake of notation simplicity.

The basic approach to IBVS control, aimed at guaranteeing a decoupled and exponential decrease of the error, is to design the velocity screw as [7]

$$v = -\lambda L_e^\dagger e, \quad \lambda > 0 \quad (9)$$

where  $L_e^\dagger \in \mathbb{R}^{6 \times 3n}$  is the Moore–Penrose pseudoinverse of  $L_e$  and  $\lambda$  is a positive constant or a positive time-varying gain. Therefore, the visual servoing error dynamics is

$$\dot{e} = -\lambda L_e L_e^\dagger e, \quad e(0) = e_0. \quad (10)$$

The matrix  $L_e L_e^\dagger \in \mathbb{R}^{3n \times 3n}$  has rank 6; thus,  $L_e L_e^\dagger$  has a nontrivial null space and the configurations of the feature vector  $s$  such that  $e \in \ker(L_e^\dagger) = \ker(L_e^\top)$  are those in which the velocity controller output is zero and the error gets stuck in an undesired equilibrium point with  $s \neq s^*$ .

The configurations of the feature vector  $s$  and, consequently, of the error  $e$  are subject to the rigid motion constraint, that is,

$$\exists R(r, \theta), \quad p : s = \bar{R}s^* + \bar{p} \quad (11)$$

$$\bar{R} = I_n \otimes R(r, \theta), \quad \bar{p} = \mathbf{1}_n \otimes p \quad (12)$$

where the symbol  $\otimes$  indicates the Kronecker product, and  $I_n$  and  $\mathbf{1}_n$  are the  $n \times n$  identity matrix and the vector

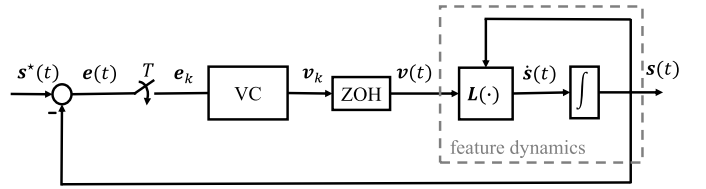


Fig. 1. Block scheme of the controlled sampled-data IBVS system. The block VC represents the visual controller.

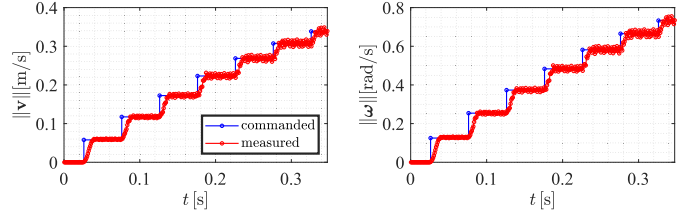


Fig. 2. Test to show that the ideal Cartesian motion device assumption is reasonable. Left: norm of the translational velocity. Right: norm of the rotational velocity.

$[1 \ 1 \ \cdots \ 1]^\top \in \mathbb{R}^n$ , respectively. Furthermore,  $r$  and  $\theta$  are the unit axis and the rotation angle of the rotation matrix  $R$ , which, together with the translation vector  $p$ , describes the displacement from the target feature vector to the current one.

### B. Sampled-Data Model

The camera frame rate and the sampling rate that can be achieved by the visual perception pipeline are usually much lower than the sampling rate of the low-level robot control loop. With standard hardware and taking into account any possible latency of the robot control interface, the sampling time of the visual control loop can be as high as 100 ms. Therefore, the sampled-data nature of the control loop should be explicitly taken into account in the design to allow the maximum possible performance of the robotic system while, at the same time, ensuring the stability and precision of the closed-loop system. Fig. 1 shows a block scheme of the visual control system, where the feature dynamic system is commanded with a sampled-data control input through a zero-order-hold (ZOH) block. We here assume that the robot accepts Cartesian velocity commands, and it can be considered an ideal Cartesian motion device. To prove that this assumption is reasonable, the following experiment has been carried out. Considering the same setup that will be used in Section V, the robot is real-time controlled via velocity commands at 1 kHz, and the velocity commands are updated at 20 Hz. The robot is commanded with sinusoidal translational and rotational velocity commands, sampled at 50 ms, with an amplitude of 0.4 m/s and 0.8 rad/s, respectively, and a frequency of 0.5 Hz. The left and right plots of Fig. 2 show the norm of the translational and rotational velocities, respectively. The blue piecewise constant line is the command, while the measured velocity is reported in red. The results clearly show that the measured velocity reaches the set point value much faster than the command sampling rate.

The first step toward the design of the proposed IBVS control algorithm is to derive the sampled-data model of the feature dynamics without resorting to the forward Euler

approximation adopted in the previous literature. Taking into account the structure of  $\mathbf{L}(s)$  in (6) and the properties of the skew symmetric matrix  $\mathbf{S}(s_i)$ , i.e.,  $\mathbf{S}(s_i)\boldsymbol{\omega} = -\mathbf{S}(\boldsymbol{\omega})s_i = \mathbf{S}^\top(\boldsymbol{\omega})s_i$ , then the term at the right-hand side of (5) can be written as  $\mathbf{L}(s)\mathbf{v} = \bar{\mathbf{S}}^\top(\boldsymbol{\omega})\mathbf{s} - \bar{\mathbf{v}}$ , and the feature dynamics in (5) can be expressed as

$$\dot{\mathbf{s}} = \bar{\mathbf{S}}^\top(\boldsymbol{\omega})\mathbf{s} - \bar{\mathbf{v}} \quad (13)$$

where  $\bar{\mathbf{S}}(\boldsymbol{\omega}) \in \mathbb{R}^{3n \times 3n}$  and  $\bar{\mathbf{v}} \in \mathbb{R}^{3n}$  have the following expressions:

$$\bar{\mathbf{S}}(\boldsymbol{\omega}) = \mathbf{I}_n \otimes \mathbf{S}(\boldsymbol{\omega}), \quad \bar{\mathbf{v}} = \mathbf{1}_n \otimes \mathbf{v}. \quad (14)$$

Denoting with  $T$  the control sampling time, the velocity control  $\mathbf{v}$  is kept constant at the value  $\mathbf{v}(kT) = \mathbf{v}_k = [\mathbf{v}_k^\top \boldsymbol{\omega}_k^\top]^\top$  in the time interval  $[kT, (k+1)T)$ . Therefore,  $\bar{\mathbf{S}}(\boldsymbol{\omega}_k)$  is instantaneously updated at each step  $k$  and is constant within this time interval. This makes the system in (13) a linear piecewise constant system, and the feature evolution in this time interval can be computed as

$$\begin{aligned} s(kT+t) &= e^{-\bar{\mathbf{S}}(\boldsymbol{\omega}_k)t} s(kT) + \\ &\quad - \int_{kT}^{kT+t} e^{-\bar{\mathbf{S}}(\boldsymbol{\omega}_k)(kT+t-\tau)} d\tau \bar{\mathbf{v}}_k \quad \forall t \in [0, T). \end{aligned} \quad (15)$$

Assuming that  $t = T$  and by a suitable change of variable, the feature dynamics in the sampling instants are governed by

$$s_{k+1} = e^{-\bar{\mathbf{S}}(\boldsymbol{\omega}_k)T} s_k - \int_0^T e^{-\bar{\mathbf{S}}(\boldsymbol{\omega}_k)\sigma} d\sigma \bar{\mathbf{v}}_k \quad \forall k \in \mathbb{N}_0 \quad (16)$$

where  $s_k$  stands for  $s(kT)$ .

By expanding the exponential matrices in the previous equation in power series, it holds that

$$\int_0^T e^{-\bar{\mathbf{S}}(\boldsymbol{\omega}_k)\sigma} d\sigma = T \sum_{i=0}^{\infty} \frac{(\bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k))^i T^i}{(i+1)!} \quad (17)$$

and the discrete-time feature dynamics results in

$$\begin{aligned} s_{k+1} &= s_k + \sum_{i=1}^{\infty} \frac{(\bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k))^i T^i}{i!} s_k + \\ &\quad - T \sum_{i=0}^{\infty} \frac{(\bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k))^i T^i}{(i+1)!} \bar{\mathbf{v}}_k. \end{aligned} \quad (18)$$

By rearranging the previous equation, we obtain

$$\begin{aligned} s_{k+1} &= s_k + T \sum_{i=1}^{\infty} \frac{(\bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k))^{i-1} T^{i-1}}{i!} \bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k) s_k + \\ &\quad - T \sum_{i=0}^{\infty} \frac{(\bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k))^i T^i}{(i+1)!} \bar{\mathbf{v}}_k \end{aligned} \quad (19)$$

and by changing the index in the first summation, we have

$$s_{k+1} = s_k + T \sum_{i=0}^{\infty} \frac{(\bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k))^i T^i}{(i+1)!} \left( \bar{\mathbf{S}}^\top(\boldsymbol{\omega}_k) s_k - \bar{\mathbf{v}}_k \right). \quad (20)$$

Finally, by following the same arguments used to derive (13), in view of (17), (20) can be written in a more compact form as

$$s_{k+1} = s_k + \mathbf{P}_k \mathbf{L}(s_k) \mathbf{v}_k \quad (21)$$

with  $\mathbf{P}_k \in \mathbb{R}^{3n \times 3n}$  defined as

$$\mathbf{P}_k = \mathbf{P}(\boldsymbol{\omega}_k) = \int_0^T e^{-\bar{\mathbf{S}}(\boldsymbol{\omega}_k)\sigma} d\sigma. \quad (22)$$

Equation (21) represents the sampled-data feature dynamic model, which describes exactly the behavior of the IBVS system at the sampling instants.

Now, let us denote the sampled-data feature error vector  $\mathbf{e} \in \mathbb{R}^{3n}$  as

$$\mathbf{e}_k = s_k - s^*. \quad (23)$$

Then, it is straightforward to derive the feature error dynamics as

$$\mathbf{e}_{k+1} = \mathbf{e}_k + \mathbf{P}_k \mathbf{L}_{\mathbf{e}_k} \mathbf{v}_k \quad (24)$$

where  $\mathbf{L}_{\mathbf{e}_k} = \mathbf{L}(\mathbf{e}_k + s^*)$  and  $\mathbf{v}_k = [\mathbf{v}_k^\top \boldsymbol{\omega}_k^\top]^\top \in \mathbb{R}^6$ . Let the velocity control law in (24) be defined as

$$\mathbf{v}_k = -\lambda \mathbf{L}_{\mathbf{e}_k}^\dagger \mathbf{e}_k \quad (25)$$

with, again,  $\lambda$  being either a positive constant or a positive time-varying scalar. Then, the expression of the sampled-data closed-loop feature error dynamics in (24) becomes

$$\mathbf{e}_{k+1} = (\mathbf{I}_{3n} - \lambda \mathbf{P}_k \mathbf{L}_{\mathbf{e}_k} \mathbf{L}_{\mathbf{e}_k}^\dagger) \mathbf{e}_k, \quad \mathbf{e}_0 = \mathbf{e}(0). \quad (26)$$

### III. STABILITY PROPERTIES

#### A. Geometric Characterization

The dynamics in (26) presents two families of equilibrium points: the desired equilibrium  $\mathbf{e}_k = \mathbf{0}$  and the set of undesired equilibrium points such that  $\mathbf{e}_k \in \ker(\mathbf{L}_{\mathbf{e}_k}^\top)$ , that is, the set of  $\mathbf{e}_k$  such that  $\mathbf{L}_{\mathbf{e}_k} \mathbf{L}_{\mathbf{e}_k}^\dagger \mathbf{e}_k = \mathbf{0}$ . In [19], it is proven that, given the target  $s^*$ , there exist only three feasible configurations  $\hat{s}_x$ ,  $x = a, b, c$ , of the feature vector  $s$  corresponding to the undesired equilibrium configurations with  $\mathbf{e} \in \ker(\mathbf{L}_e^\top)$ ,  $s \neq s^*$  and satisfying the rigid motion constraint (11). The three configurations are characterized by  $(\theta, \mathbf{p}, \mathbf{r}) = (\hat{\theta}, \hat{\mathbf{p}}, \hat{\mathbf{r}})$  that satisfy the following conditions:

$$\hat{\theta} = \pm \pi \quad (27a)$$

$$\hat{\mathbf{p}} = (\mathbf{I}_3 - \mathbf{R}) s_\Sigma^* \quad (27b)$$

$$\hat{\mathbf{r}} = \text{eigvect} \left( \sum_{i=1}^n (s_i^* s_i^{*\top}) - n s_\Sigma^* s_\Sigma^{*\top} \right) \quad (27c)$$

where

$$s_\Sigma^* = \frac{1}{n} \sum_{i=1}^n s_i^*. \quad (28)$$

By virtue of the rigid motion constraint expressed in (11), the visual servoing dynamics can be characterized in SE(3) by means of the variables  $\theta$ ,  $\mathbf{r}$ , and  $\mathbf{p}$ . In the SE(3) space, the desired equilibrium configuration  $s = s^*$  corresponds to  $\theta = 0$ ,  $\mathbf{r} \in \mathbb{R}^3$  (that is,  $\mathbf{R}(\mathbf{r}, \theta) = \mathbf{I}_3$ ), and  $\mathbf{p} = \mathbf{0}$ . The undesired

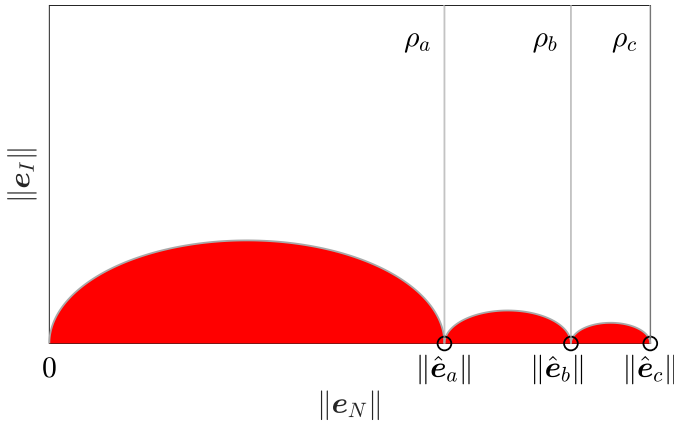


Fig. 3. Plane  $(\|e_N\|, \|e_I\|)$ ; the undesired equilibria are reported on the horizontal axis. The points in the red area do not satisfy the rigid motion constraint.

equilibrium configurations, instead, are those characterized by  $\theta$ ,  $\mathbf{r}$ , and  $\mathbf{p}$  satisfying conditions (27).

Following the same idea used for the continuous-time analysis in [20], we denote with  $e_I$  the projection of  $e$  on the  $\text{Im}(\mathbf{L}_e)$ , that is,  $e_I = \mathbf{L}_e \mathbf{L}_e^\dagger e$ , and  $e_N$  the projection of  $e$  on the  $\text{ker}(\mathbf{L}_e^\top)$ , that is,  $e_N = (\mathbf{I}_{3n} - \mathbf{L}_e \mathbf{L}_e^\dagger) e$ .

Note that, even if the feature vector  $s \in \mathbb{R}^{3n}$ , the rigid motion constraint (11) imposes the features to belong to the subset  $\mathcal{R}^* \subset \mathbb{R}^{3n}$  [isomorphic to  $\text{SE}(3)$ ] defined as the set of all the feature vectors  $s$  satisfying (11). Fig. 3 graphically illustrates such constraint in the plane  $(\|e_N\|, \|e_I\|)$ . The figure has been numerically constructed with the procedure described in [20]. The red areas correspond to points that do not satisfy the constraint (11); thus, any point of the system trajectories reported in the plane  $(\|e_N\|, \|e_I\|)$  cannot belong to such regions.

Denoting with  $\rho_x$  the vertical lines passing through  $\|e_N\| = \|\hat{e}_x\|$ , being  $\hat{e}_x = \hat{s}_x - s^*$ , with  $x = a, b, c$ , the three undesired configurations, in [20], we have proved the following results.

- 1) The boundaries of the red areas in Fig. 3 are tangent to the vertical lines  $\rho_x$  in the points  $(\|e_x\|, 0)$ .
- 2) As a consequence of result 1, it is

$$\lim_{\|e\| \rightarrow 0} \frac{\|e\|}{\|e_I\|} = 1. \quad (29)$$

- 3) In view of [20, Proposition 1], for all three undesired equilibria  $\hat{s}_x$  (the subscript  $x$  will be dropped to simplify the rest of the treatment), the following property holds. Any point  $s$  obtained as a pure translation of  $\hat{s}$  belongs to the manifold  $\mathcal{N} = \{s \mid s = \hat{s} + \Delta \bar{\mathbf{p}}\}$ , where  $\Delta \bar{\mathbf{p}} = \bar{\mathbf{p}} - \hat{\mathbf{p}}$  and  $\hat{\mathbf{p}} = \mathbf{1}_n \otimes \hat{\mathbf{p}}$ . Such a set for any given  $s^*$  corresponds to the  $\rho$  lines in the plane  $(\|e_N\|, \|e_I\|)$  in Fig. 3. In other words, considering a pure translation starting from  $\hat{s}$ , the feature error projection  $e_N$  remains constant and equal to  $\hat{e}$ .

### B. Dynamics Characterization

In our former paper [20], we proved that the trajectories of the continuous-time closed-loop system (10) starting from  $e_0$  such that  $s_0 \in \mathcal{N}$  converge to the undesired equilibrium  $\hat{e}$

for any value of the gain  $\lambda$ . Here, we are ready to prove that such behavior is exhibited also by the sampled-data closed-loop system (26) but with a limit on the gain  $\lambda$  depending on the sampling time. Note that the proof of the next proposition is fairly more articulated due to the presence of the matrix  $\mathbf{P}(\omega_k)$  in the sampled-data system dynamics.

Given the above reasoning, we are now ready to analyze the behavior of the system (26) when  $s_0 \in \mathcal{N}$ .

*Proposition 1:* With reference to the system (26), the only trajectories  $e_k$  that asymptotically converge to  $\hat{e}$  are those starting from  $e_0$  such that  $s_0 \in \mathcal{N}$  if and only if  $0 < \lambda T < 2$ .

*Proof:* Recalling result 3, any  $s_0 \in \mathcal{N}$  can be generated as a pure translation of  $\hat{s}$ , that is,  $s_0 = \hat{s} + \Delta \bar{\mathbf{p}}_0$ , with  $\Delta \bar{\mathbf{p}}_0 = \bar{\mathbf{p}}_0 - \hat{\mathbf{p}}$ . Thus, for any given  $s^*$ ,  $e_0 = \hat{e} + \Delta \bar{\mathbf{p}}_0$ . Then, given the dynamics in (26), for  $k = 0$ , it holds that

$$e_1 = (\mathbf{I}_{3n} - \lambda \mathbf{P}_0 \mathbf{L}_{e_0} \mathbf{L}_{e_0}^\dagger) (\hat{e} + \Delta \bar{\mathbf{p}}_0). \quad (30)$$

Since  $\hat{e} \in \text{ker}(\mathbf{L}_e^\top)$  and its components  $\hat{e}_i \in \mathbb{R}^3$  are such that

$$\begin{aligned} \sum_{i=1}^n \hat{e}_i &= \sum_{i=1}^n \left( (\hat{\mathbf{R}} - \mathbf{I}) s_i^* + \hat{\mathbf{p}} \right) \\ &= 2\mathbf{S}^2(\hat{\mathbf{r}}) \sum_{i=1}^n s_i^* - 2n\mathbf{S}^2(\hat{\mathbf{r}}) s_\Sigma^* = \mathbf{0} \end{aligned} \quad (31)$$

it holds that

$$\mathbf{L}_{e_0}^\top \hat{e} = \left[ \mathbf{S}^\top (\Delta \bar{\mathbf{p}}_0) \sum_{i=1}^n \hat{e}_i \right] = \mathbf{0}. \quad (32)$$

Then,

$$e_1 = \hat{e} + (\mathbf{I}_{3n} - \lambda \mathbf{P}_0 \mathbf{L}_{e_0} \mathbf{L}_{e_0}^\dagger) \Delta \bar{\mathbf{p}}_0 \quad (33)$$

which, considering that  $\Delta \bar{\mathbf{p}}_0 \in \text{Im}(\mathbf{L}_{e_0})$ , becomes

$$e_1 = \hat{e} + (\mathbf{I}_{3n} - \lambda \mathbf{P}_0) \Delta \bar{\mathbf{p}}_0. \quad (34)$$

Since  $s_0 \in \mathcal{N}$ , then  $\omega_0 = \mathbf{0}$ , and from (22),  $\mathbf{P}_0 = T \mathbf{I}_{3n}$ , providing

$$\begin{aligned} e_1 &= \hat{e} + (1 - \lambda T) \Delta \bar{\mathbf{p}}_0 \\ &= \hat{e} + \Delta \bar{\mathbf{p}}_1 \end{aligned}$$

with  $\Delta \bar{\mathbf{p}}_1 = (1 - \lambda T) \Delta \bar{\mathbf{p}}_0$ , resulting in the fact that  $e_1$  represents a mere translation with respect to  $\hat{e}$ . This, in turn, implies that no rotation is generated. Hence, by iterating the above reasoning, it is possible to write the dynamics of trajectories  $e_k$  with  $e_0$  such that  $s_0 \in \mathcal{N}$  as

$$e_{k+1} = \hat{e} + (1 - \lambda T) \Delta \bar{\mathbf{p}}_k \quad (35)$$

or, equivalently,

$$e_{k+1} - \hat{e} = (1 - \lambda T) \Delta \bar{\mathbf{p}}_k. \quad (36)$$

Thus,

$$\lim_{k \rightarrow \infty} \|e_k - \hat{e}\| = 0 \quad (37)$$

since  $\Delta \bar{\mathbf{p}}_{k+1} = (1 - \lambda T) \Delta \bar{\mathbf{p}}_k$  is an LTI discrete-time system, which is stable for  $0 < \lambda T < 2$ .

Thus, the feature trajectories starting from the set  $\mathcal{N}$  converge to the undesired equilibrium points  $\hat{s}$  without leaving  $\mathcal{N}$ , which means that the error converges to  $\hat{e}$ .

To prove that they are the only ones, consider a slight perturbation on  $e_0$  obtained by perturbing  $\theta = \hat{\theta} + \delta\theta$  and  $r = \hat{r} + \delta r$ . Since, on the manifold  $\mathcal{N}$ , the angular velocity  $\omega = \mathbf{0}$ , then a slight perturbation from  $\mathcal{N}$  generates a low angular velocity.

The matrix  $P_k = P(\omega_k)$  in the system dynamics (26), by virtue of the Euler–Rodrigues rotation formula  $R = e^{S(\theta r)} = I_3 + \sin(\theta)S(r) + (1 - \cos(\theta))S^2(r)$  [31], can be written as

$$P(\omega_k) = T \left[ I_{3n} + \frac{1 - \cos(\|\omega_k\|T)}{\|\omega_k\|T} \bar{S}^\top \left( \frac{\omega_k}{\|\omega_k\|} \right) + \frac{\|\omega_k\|T - \sin(\|\omega_k\|T)}{\|\omega_k\|T} \bar{S}^2 \left( \frac{\omega_k}{\|\omega_k\|} \right) \right]. \quad (38)$$

For very low values of  $\|\omega_k\|$ , it holds

$$P(\omega_k) \approx T I_{3n} + \frac{1}{2} T^2 \bar{S}^\top(\omega_k). \quad (39)$$

Now, consider the Lyapunov candidate function

$$V_k = (e_k - \hat{e})^\top (e_k - \hat{e}) \quad (40)$$

whose first-order variation along the error dynamics (26)  $\Delta V_k = V_{k+1} - V_k$  can be written, recalling that  $e_{Ik} = L_{e_k} L_{e_k}^\dagger e_k$  and  $e_{Nk} = e_k - e_{Ik}$ , as

$$\Delta V_k = 2\lambda e_{Ik}^\top P_k^\top (\hat{e} - e_{Nk}) - 2\lambda e_{Ik}^\top P_k^\top e_{Ik} + \lambda^2 e_{Ik}^\top P_k^\top P_k e_{Ik}. \quad (41)$$

Substituting (39) into (41), it holds that

$$\Delta V_k \approx 2\lambda T e_{Ik}^\top (\hat{e} - e_{Nk}) + 2\lambda T^2 e_{Ik}^\top \bar{S}^\top(\omega_k) (\hat{e} - e_{Nk}) - \lambda T e_{Ik}^\top ((2 - \lambda T) I_{3n} + \frac{1}{4} \lambda T^3 \bar{S}^2(\omega_k)) e_{Ik}. \quad (42)$$

Note that the approximation (42) of (41) is obtained by neglecting terms of the order  $\|\omega_k\|^2$ ; therefore, there exist a neighborhood of  $\omega_k = \mathbf{0}$  where they have the same sign. Furthermore, in view of (39), system (26) becomes

$$e_{k+1} = e_k - \lambda T \left( I_{3n} - \frac{1}{2} T \bar{S}^\top(\omega_k) \right) e_{Ik}. \quad (43)$$

As long as  $\omega_k$  is negligible, considering  $e_k = e_{Ik} + e_{Nk}$ , the solution of the system can be approximated as

$$e_{Nk} = e_{N0} \quad (44)$$

$$e_{Ik} = (1 - \lambda T)^k e_{I0} \quad (45)$$

implying that  $e_{Ik}$  decreases. By evaluating the single terms at the right-hand side of (42), we note that, being  $s_0$  close to the manifold  $\mathcal{N}$ , the vectors  $e_{Nk} = e_{N0}$  and  $\hat{e}$  are almost aligned and such that  $\|\hat{e}\| \approx \|e_{Nk}\|$ . Thus, the difference  $\hat{e} - e_{Nk}$  can be considered orthogonal to both vectors, i.e., it is almost aligned with  $e_{Ik}$ , implying that the first term in (42) is positive definite. Regarding the second term in (42), we cannot evaluate its sign, but we can evaluate its norm. In fact, comparing the first and second terms, it can be observed that the norm of the second term is definitely smaller than the norm of the first one due to the presence of terms  $\bar{S}(\omega_k)$  (with  $\omega_k$  having negligible

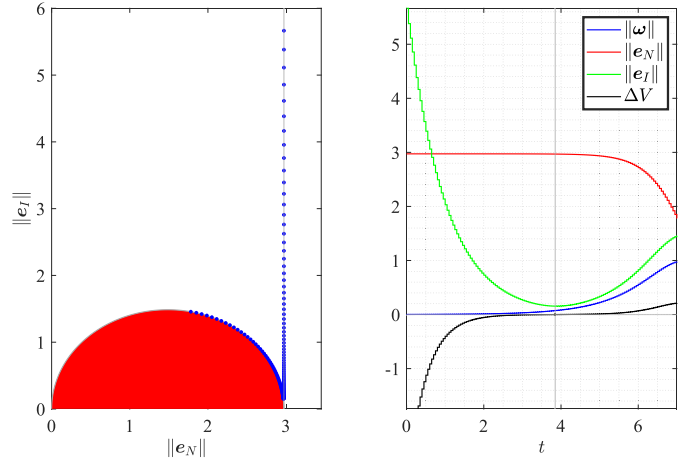


Fig. 4. Simulation results with  $s_0$  close to the manifold  $\mathcal{N}$ . Left: trajectory in the plane  $(\|e_N\|, \|e_I\|)$ . Right:  $\|\omega\|$ ,  $\|e_N\|$ , and  $\Delta V$ .

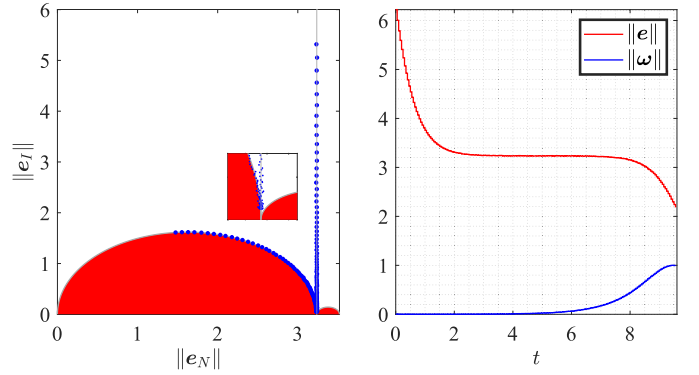


Fig. 5. Simulation results with camera measurement noise and  $s_0$  on the manifold  $\mathcal{N}$ . Left: trajectory in the plane  $(\|e_N\|, \|e_I\|)$ . Right: error norm.

norm) and  $T^2$  (being  $T$  smaller than 1). Thus, the second term can be neglected compared to the first one. Regarding the third and last terms, the term characterized by  $T^3 \bar{S}^2(\omega_k)$  is positive semidefinite, and it can be neglected due to its small magnitude. Therefore, the first-order variation  $\Delta V_k$  in (42) can be further approximated as

$$\Delta V_k \approx 2\lambda T e_{Ik}^\top (\hat{e} - e_{Nk}) - \lambda T (2 - \lambda T) \|e_{Ik}\|^2 \quad (46)$$

where the first term is linear with respect to  $e_{Ik}$ , while the second term is quadratic. As long as the norm of  $\omega_k$  is negligible, the solution of  $e_{Ik}$  evolves as in (45), and there exist a time instant  $k$  such that the linear term definitely dominates the quadratic one, thus yielding a positive first-order variation of the Lyapunov function. This implies that  $\|e_k - \hat{e}\|$  increases, and the trajectory moves away from the manifold  $\mathcal{N}$ , thus proving that any trajectory starting outside  $\mathcal{N}$  cannot converge to the undesired equilibria. ■

The arguments on the feature error dynamics character, discussed in the proof of Proposition 1, are elucidated by means of the simulation results illustrated in Figs. 4 and 5. The results in Fig. 4 are obtained by starting the feature dynamics with an initial condition close to the manifold  $\mathcal{N}$ , that is,  $\theta_0 = 179.9^\circ$ ,  $r_0 = \hat{r}$ , and  $p_0 = [0.3, 0, 0]^\top$ . The left plot in Fig. 4 shows the error norm trajectory in the plane  $(\|e_N\|, \|e_I\|)$ . It can be observed that the trajectory

initially evolves due to the translational velocity control only; in fact, the angular velocity  $\|\omega\|$  in the right plot is negligible. In correspondence with the sign variation of  $\Delta V$ , in the right plot,  $\|\omega_k\|$  begins to grow up; at the same time,  $\|e_N\|$  starts to change noticeably, and  $\|e_I\|$  begins to increase. For this reason, the error trajectory in the left plot goes away from the  $\rho$  line, which means that the feature dynamics moves away from the manifold  $\mathcal{N}$ .

The analysis in Proposition 1 and the simulation have been carried out so far, assuming the absence of any camera measurement noise. It is clear that, in real cases, in the presence of noise, the feature trajectory cannot lie on the zero Lebesgue measurement manifold  $\mathcal{N}$ , even if the initial condition belongs to it. Anyway, the importance of Proposition 1 is to highlight a pathology on the trajectory convergence character when  $\theta_0 \approx \hat{\theta}$  and  $r_0 \approx \hat{r}$  due to the very low norm of the camera angular velocity in the initial part of the feature trajectory. Fig. 5 shows the simulation results carried out assuming  $\theta_0 = \hat{\theta}$ ,  $r_0 = \hat{r}$ , and a random uniform uncertainty of 1% on the depth measurement. As illustrated in the left plot, in the first part of the trajectory, the camera moves only due to a translational velocity control; thus, it translates only and reaches the target at around  $t = 3$  s, but  $\theta$  is still close to  $\hat{\theta}$ . Between 3 and 6 s, the angular velocity is almost zero, and the error norm is practically constant (see the right plot in Fig. 5). At  $t = 6$  s, the angular velocity starts increasing, and the error trajectory moves away from the manifold  $\mathcal{N}$ . This behavior entails a drawback: the end-effector rotates only after getting very close to the target, e.g., in a task of object pick&place, this implies that the robot may collide with the object or the desk on which it lies, and/or depending on the robot configuration, a collision between the camera and the robot arm may occur. The above analysis suggests that initial conditions of the feature points close to the manifold  $\mathcal{N}$  should be avoided in order to prevent the generation of sole translational trajectories followed by sole rotations; the latter performed when the camera is very close to the target configuration. By simulation, we will show later that the basin of convergence of the trajectories that do not present the described pathology is very large.

We now discuss the stability properties of the desired equilibrium  $e = \mathbf{0}$ .

*Proposition 2:* Consider the sampled-data closed-loop system (26) with  $\lambda = \lambda_k$ . Denote with  $\omega_k^1 = -[\mathbf{0}_3 \ I_3] L_{e_k}^\dagger e_k$  and  $\alpha_k = \|e_k\|/\|e_{Ik}\|$ . If  $\lambda_k$  is selected such that

$$0 < \lambda_k T < \frac{2}{1 + \alpha_k \|\omega_k^1\|} \quad \forall k \in \mathbb{N}_0 \quad (47)$$

then the desired equilibrium point  $e = \mathbf{0}$  of system (26) is *almost globally asymptotically stable*.

*Proof:* Since the skew matrix is normal, i.e.,  $S^\top(\omega_k)S(\omega_k) = S(\omega_k)S^\top(\omega_k)$ , by resorting to [32, Th. 2.5.3], the skew matrix  $S^\top(\omega_k)$  can be factorized as

$$S^\top(\omega_k) = U_k \begin{bmatrix} 0 & 0 & 0 \\ 0 & -j\|\omega_k\| & 0 \\ 0 & 0 & j\|\omega_k\| \end{bmatrix} U_k^* \quad (48)$$

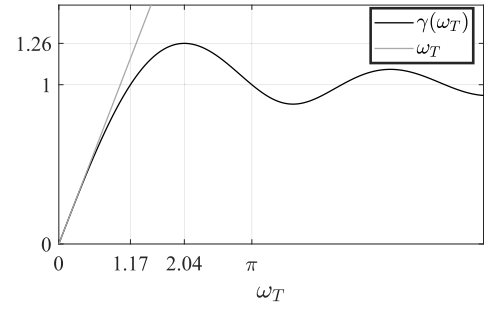


Fig. 6. Plot of the function  $\gamma(\omega_T)$  in (56).

where  $U_k$  is unitary and  $U_k^*$  is the conjugate transpose of  $U_k$ . Since the diagonal matrix at the right-hand side of (48) is a  $3 \times 3$  singular diagonal matrix with the first element of the diagonal equal to zero, then the following integral can be written as:

$$\begin{aligned} & \int_0^T e^{S^\top(\omega_k)\sigma} d\sigma \\ &= T U_k \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{j\omega_T k} \operatorname{sinc}(\omega_T k) & 0 \\ 0 & 0 & e^{-j\omega_T k} \operatorname{sinc}(\omega_T k) \end{bmatrix} U_k^* \\ &= T U_k \Lambda_k U_k^* \end{aligned} \quad (49)$$

where  $\omega_T k = \|\omega_k\|(T/2)$ . Thus, the matrix  $P_k$  can be expressed as

$$P_k = \int_0^T e^{-\bar{S}(\omega_k)\sigma} d\sigma = T \bar{U}_k \bar{\Lambda}_k \bar{U}_k^* \quad (50)$$

where  $\bar{\Lambda}_k, \bar{U}_k \in \mathbb{C}^{3n \times 3n}$  are defined as

$$\bar{\Lambda}_k = I_n \otimes \Lambda_k, \quad \bar{U}_k = I_n \otimes U_k. \quad (51)$$

By selecting as Lyapunov function candidate  $V(e_k) = e_k^\top e_k$ , its first-order variation  $\Delta V_k = e_{k+1}^\top e_{k+1} - e_k^\top e_k$  along the error dynamics (26) can be written as

$$\begin{aligned} \Delta V_k &= -2\lambda_k e_k^\top P_k L_{e_k} L_{e_k}^\dagger e_k \\ &\quad + \lambda_k^2 e_k^\top L_{e_k} L_{e_k}^\dagger P_k^\top P_k L_{e_k} L_{e_k}^\dagger e_k. \end{aligned} \quad (52)$$

Denoting with  $B_k$  the matrix  $P_k - T I_{3n}$ , we can write

$$B_k = T \bar{U}_k (\bar{\Lambda}_k - I_{3n}) \bar{U}_k^*. \quad (53)$$

Then,  $\Delta V_k$  in (52) can be rewritten as

$$\begin{aligned} \Delta V_k &= -2\lambda_k T e_k^\top L_{e_k} L_{e_k}^\dagger e_k - 2\lambda_k e_k^\top B_k L_{e_k} L_{e_k}^\dagger e_k \\ &\quad + \lambda_k^2 e_k^\top L_{e_k} L_{e_k}^\dagger P_k^\top P_k L_{e_k} L_{e_k}^\dagger e_k. \end{aligned} \quad (54)$$

Recalling that  $L_{e_k} L_{e_k}^\dagger$  is the projector matrix on the  $\operatorname{Im}(L_{e_k})$  and  $e_{Ik} = L_{e_k} L_{e_k}^\dagger e_k$ , then  $\Delta V_k$  can be upper bounded as follows:

$$\begin{aligned} \Delta V_k &\leq -2\lambda_k T \|e_{Ik}\|^2 + 2\lambda_k \|B_k\| \|e_{Ik}\| \|e_k\| \\ &\quad + \lambda_k^2 e_{Ik}^\top P_k^\top P_k e_{Ik}. \end{aligned} \quad (55)$$

As the matrix  $P_k^\top P_k = T^2 \bar{U}_k \bar{\Lambda}_k^* \bar{\Lambda}_k \bar{U}_k^*$ , with  $\bar{U}_k$  being a unitary matrix and  $\|\bar{\Lambda}_k^* \bar{\Lambda}_k\| = 1$ , as can be easily verified,

then we have  $\|\mathbf{P}_k^\top \mathbf{P}_k\| \leq T^2$ . In view of (53), the norm of the matrix  $\mathbf{B}_k$  is such that

$$\frac{\|\mathbf{B}_k\|}{T} \leq \left| e^{i\omega_{Tk}} \text{sinc}(\omega_{Tk}) - 1 \right| = \gamma(\omega_{Tk}). \quad (56)$$

The graph of the function  $\gamma(\omega_T)$  is reported in Fig. 6, where the plot clearly shows that  $\gamma(\omega_{Tk}) \leq \omega_{Tk}$ .

In view of the above arguments and the definition of  $\alpha_k$ , the upper bound for  $\Delta V_k$  can be rewritten as

$$\Delta V_k \leq -\lambda_k T (2\|\mathbf{e}_{Ik}\| - \lambda_k T \alpha_k \|\boldsymbol{\omega}_k^1\| \|\mathbf{e}_{Ik}\| + \lambda_k T \|\mathbf{e}_{Ik}\|) \|\mathbf{e}_{Ik}\|. \quad (57)$$

Since  $\mathbf{e}_N \in \ker(\mathbf{L}_e^\top)$ , then  $\boldsymbol{\omega}_k^1 = -[\mathbf{0}_3 \ \mathbf{I}_3] \mathbf{L}_{e_k}^\dagger \mathbf{e}_{Ik}$ , and by virtue of (29), the three terms in parenthesis tend to zero with the same rate. This, in turn, implies that, for  $\|\mathbf{e}_{Ik}\| \neq 0$ , it is always possible to select  $\lambda_k T$  such that  $\Delta V_k \leq 0$ . Hence, for such selection of  $\lambda_k T$ , the rate of variation of the Lyapunov function is negative definite almost everywhere (since, apart from  $\|\mathbf{e}_k\| = \mathbf{0}$ ,  $\Delta V_k = 0$  only in the undesired equilibrium points). Since the only trajectories converging to the undesired equilibria are those belonging to the zero Lebesgue measure sets introduced in Proposition 1, by virtue of (29), *almost global asymptotic stability* is achieved for the desired equilibrium point if, for each  $k$ ,  $\lambda_k$  is such that

$$0 < \lambda_k T < \frac{2}{1 + \alpha_k \|\boldsymbol{\omega}_k^1\|} \quad \forall k \in \mathbb{N} \quad (58)$$

implying that  $\Delta V_k < 0$ ,  $\forall k$ . ■

#### IV. SAMPLED-DATA VISUAL SERVOING CONTROL

In this section, a novel visual servoing control algorithm is presented to pursue the fulfillment of the following requirements.

- 1) The feature error dynamics should exhibit a monotonic decrease over the continuous time  $t$ .
- 2) The camera velocity control should comply with the joint velocity limits.
- 3) The camera velocity control should guarantee a prescribed smooth approach of the camera toward the prescribed target configuration.
- 4) The visual control should adapt to the specific robotic system without the need of experimentally tuning the control parameters.

The first three requirements aim at fully exploiting the robot's dynamic performance, while the last requirement ensures the portability of the visual controller.

The control algorithm proposed in the following proposition requires the definition of two scalar functions, i.e.,  $\pi(\cdot, \cdot)$  called *cost function* and  $l(\cdot)$  called *landing function*. The first one is defined as

$$\pi(\mathbf{e}_k, \lambda_k T) = -\lambda_k T \left( 2l(\mathbf{e}_k) - \alpha_k \lambda_k T \|\boldsymbol{\omega}_k^1\| - \lambda_k T \right) \quad (59)$$

which, as a function of  $\lambda_k T$ , being  $\lambda_k > 0$ , is a concave up parabola passing through the origin and the point  $\lambda_k T = 2l(\mathbf{e}_k)/(1 + \alpha_k \|\boldsymbol{\omega}_k^1\|) = \lambda_k^0 T$ . The *landing function*  $l(\mathbf{e}_k)$ , which is designed to allow a smooth approach to the feature target configuration, is defined as follows. Denote with  $e_{Mk}$

the mean square error  $e_{Mk} = \|\mathbf{e}_k\|/\sqrt{n}$ , with  $e_L$  and  $e_H$  two positive scalars such that  $e_L < e_H$ , let  $\mu \in (0, 1]$ , and then

$$l(\mathbf{e}_k) = \begin{cases} \mu, & \text{if } e_{Mk} \leq e_L \\ 1, & \text{if } e_{Mk} \geq e_H \\ \mu + (1 - \mu)q\left(\frac{e_{Mk} - e_L}{e_H - e_L}\right), & \text{otherwise} \end{cases} \quad (60)$$

where  $q(\cdot)$  is a fifth-order polynomial whose coefficients are determined such that the first- and second-order derivatives are zero at the extremes of the interval  $[0, 1]$  and  $q(0) = 0$ ,  $q(1) = 1$ . Note that  $0 < l(\mathbf{e}_k) \leq 1$ , and  $\mu = 1$  implies that  $l(\mathbf{e}_k) = 1$ . Finally, denote with  $v_k^M$  and  $\omega_k^M$  the maximum allowed norm of the linear and angular velocities at the  $k$ th step, respectively.

*Remark 1:* The term ‘‘landing function’’ for  $l(\mathbf{e}_k)$  has been chosen because an appropriate setting of the values  $e_L$  and  $e_H$ , and the parameter  $\mu$  allows to design the deceleration profile for a smooth approach to the target configuration. Guidelines to select these values will be provided in Section V.

The design of the visual servoing controller is then presented in the following proposition.

*Proposition 3:* Consider the sampled-data closed-loop system (26) with  $\lambda = \lambda_k$ . Assuming that  $s_0 \notin \mathcal{N}$ , if the gain  $\lambda_k$  is determined, for each sampling time interval, by solving

$$P1: \min_{\lambda_k T} \pi(\mathbf{e}_k, \lambda_k T) \quad (61a)$$

$$\text{s.t. } 0 < \lambda_k T < \frac{2l(\mathbf{e}_k)}{1 + \alpha_k \|\boldsymbol{\omega}_k^1\|} \quad (61b)$$

$$\|\boldsymbol{\omega}_k\| \leq \omega_k^M, \quad \|\mathbf{v}_k\| \leq v_k^M \quad (61c)$$

then the following holds.

- 1) The equilibrium point  $\mathbf{e} = \mathbf{0}$  of the sampled-data closed-loop system is *almost globally asymptotically stable*.
- 2) The feature error norm  $\|\mathbf{e}(t)\|$  exhibits a monotonic decrease over the time  $t$ .

*Proof:* In Proposition 2 we have proven that the sampled-data closed loop system in (26) is *almost globally asymptotically stable* if  $\lambda_k T$  is selected in the range defined in (47). The proof has been performed via the Lyapunov direct method by selecting the Lyapunov function candidate  $V(\mathbf{e}_k) = \mathbf{e}_k^\top \mathbf{e}_k$ . An upper bound of its first-order variation  $\Delta V_k$  can be written by rearranging (57) as

$$\Delta V_k \leq -\lambda_k T (2 - \lambda_k T \alpha_k \|\boldsymbol{\omega}_k^1\| - \lambda_k T) \|\mathbf{e}_{Ik}\|^2 \quad (62)$$

implying that

$$\Delta V_k \leq \pi(\mathbf{e}_k, \lambda_k T) \|\mathbf{e}_{Ik}\|^2 \quad (63)$$

since  $l(\mathbf{e}_k)$  belongs to the range  $(0, 1]$ . Then, the equilibrium point  $\mathbf{e} = \mathbf{0}$  is *almost globally asymptotically stable* if  $\lambda_k T$  is in the range (61b).

Now, we have to prove that the proposed gain adaptation algorithm allows us to achieve the objective 2).

By multiplying and dividing the last term in (54) by  $T^2$  and by dividing the left-hand and the right-hand side by  $\|\mathbf{e}_{Ik}\|^2$ , we obtain

$$\frac{\Delta V_k}{\|\mathbf{e}_{Ik}\|^2} = \lambda_k^2 T^2 a_k - 2\lambda_k T b_k = f_k(\lambda_k T) \quad (64)$$



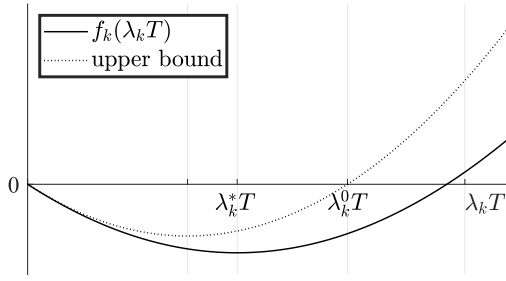


Fig. 7. Plot of  $f_k(\lambda_k T)$  and its upper bound in (57) normalized with respect to  $\|e_I\|^2$ .

where  $a_k$ , by virtue of (50), is given as follows:

$$a_k = e_{Ik}^\top \bar{U}_k \bar{\Lambda}_k^* \bar{\Lambda}_k \bar{U}_k^* e_{Ik} / \|e_{Ik}\|^2 > 0 \quad (65)$$

$$b_k = (1 + e_k^\top \bar{U}_k (\bar{\Lambda} - I_{3n}) \bar{U}_k^* e_{Ik} / \|e_{Ik}\|^2) > 0. \quad (66)$$

Let us define the function  $g_k(\lambda_k t) = \|e(kT + t)\|^2 - \|e(kT)\|^2$ ,  $t \in [0, T]$ , which has the following properties:

$$\begin{aligned} g_k(0) &= 0 \\ g_k(\lambda_k T) &= f_k(\lambda_k T) \|e_{Ik}\|^2 \quad \forall \lambda_k T > 0. \end{aligned} \quad (67)$$

The function  $g_k(\lambda_k T)$  is upper bounded by the right-hand side of (63), which is a parabola with branches pointing up and negative definite for  $\lambda_k T$  belonging to the open interval in (47).

The function  $f_k(\lambda_k T)$  and its upper bound are represented in Fig. 7.

Define  $\lambda_k^* T$  as

$$\lambda_k^* T = \arg \min_{\lambda_k T} f_k(\lambda_k T). \quad (68)$$

Then, for each  $k$ ,  $f_k(\lambda_k T)$  is monotonically decreasing in the interval  $[0, \lambda_k^* T]$ .

Now, we claim the following.

- 1)  $g_k(\lambda_k^* t)$  is monotonically decreasing in the interval  $[0, \lambda_k^* T]$ .
- 2) For all  $\bar{\lambda}_k > \lambda_k^*$ ,  $g_k(\bar{\lambda}_k t)$  is not monotonically decreasing in the interval  $[0, \bar{\lambda}_k T]$ .

*Proof of Claim 1:*  $\forall \lambda_k^* t_a, \lambda_k^* t_b \in [0, \lambda_k^* T] : t_a < t_b$  and  $\lambda_{ka} T, \lambda_{kb} T$

$$\lambda_{ka} T = \lambda_k^* t_a, \quad \lambda_{kb} T = \lambda_k^* t_b \quad (69)$$

as a consequence of the monotonic decrease of  $f_k(\lambda_k T)$  in  $[0, \lambda_k^* T]$ , we have that

$$\lambda_{ka} T < \lambda_{kb} T \implies f_k(\lambda_{ka} T) > f_k(\lambda_{kb} T) \quad (70)$$

and, by virtue of properties (67), we have

$$g_k(\lambda_k^* t_a) > g_k(\lambda_k^* t_b) \quad (71)$$

which ensures the monotonic decrease of  $g_k(\lambda_k^* t)$  in  $[0, \lambda_k^* T]$  and then of  $\|e(kT + t)\|$  with  $t \in [0, T]$ ,  $\forall k$ .

*Proof of Claim 2:*  $\forall \bar{\lambda}_k > \lambda_k^*$ ,  $\exists t' = T \lambda_k^* / \bar{\lambda}_k$ ,  $t'' = T : t' < t''$ , for which it results

$$g_k(\bar{\lambda}_k t') = g_k(\lambda_k^* T) = f_k(\lambda_k^* T) \|e_{Ik}\|^2 \quad (72)$$

$$g_k(\bar{\lambda}_k t'') = g_k(\bar{\lambda}_k T) = f_k(\bar{\lambda}_k T) \|e_{Ik}\|^2. \quad (73)$$

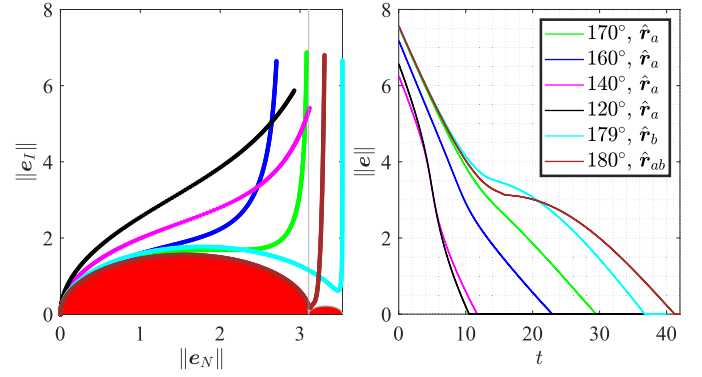


Fig. 8. Simulation results for values of the initial angle error  $\theta_0 = \{180^\circ, 179^\circ, 170^\circ, 160^\circ, 150^\circ, 140^\circ, 120^\circ\}$ . Left: error trajectory in the plane  $(\|e_N\|, \|e_I\|)$ . Right: error norm.

As  $f_k(\lambda_k^* T) < f_k(\bar{\lambda}_k T)$ , this implies that

$$g_k(\bar{\lambda}_k t'') > g_k(\bar{\lambda}_k t'). \quad (74)$$

Then,  $g_k(\lambda_k t)$  is not monotonically decreasing in  $[0, \bar{\lambda}_k T]$   $\forall k$ , and the sampled-data visual servoing system (26) exhibits hidden oscillations.

Since the function  $\pi(e_k, \lambda_k T)$  has a minimum in correspondence of  $\tilde{\lambda}_k$  with  $\tilde{\lambda}_k < \lambda_k^*$ , the solution of problem P1 ensures the absence of hidden oscillations.

If  $\lambda_k = \tilde{\lambda}_k$  is such that the constraint (61c) is not satisfied, the minimization algorithm will select  $\lambda_k < \tilde{\lambda}_k$ ; as the camera velocity depends linearly from  $\lambda_k$ , this again ensures that  $g_k(\lambda_k t)$  is monotonically decreasing for  $t \in [0, T]$ ,  $\forall k \in \mathbb{N}_0$ . ■

Note that Problem P1 in (61) has a simple closed-form solution in  $\lambda_k T$ . In fact, this is a scalar minimization problem of the parabola defined by the function  $\pi$ . The unconstrained minimum of the parabola is  $\lambda_k T = l(e_k) / (1 + \alpha_k \|\omega_k^1\|)$ , which satisfies constraint (61b). Moreover, the constraints in (61c) are linear inequality constraint in  $\lambda_k T$  and can be written as  $\lambda_k T < T \omega_k^M / \|\mathbf{0}_3 \quad \mathbf{I}_3\| L_{e_k}^\dagger e_k\|$  and  $\lambda_k T < T v_k^M / \|\mathbf{I}_3 \quad \mathbf{0}_3\| L_{e_k}^\dagger e_k\|$ . Thus, the closed-form solution of P1 can be written as

$$\lambda_k T = \min \left( \frac{l(e_k)}{(1 + \alpha_k \|\omega_k^1\|)}, \frac{T \omega_k^M}{\|\mathbf{0}_3 \quad \mathbf{I}_3\| L_{e_k}^\dagger e_k\|}, \frac{T v_k^M}{\|\mathbf{I}_3 \quad \mathbf{0}_3\| L_{e_k}^\dagger e_k\|} \right). \quad (75)$$

After defining the novel visual servoing control algorithm guaranteeing *almost global asymptotic stability* of the closed-loop system and the absence of hidden oscillations, it is of interest to investigate for which cases of the initial orientation error the pathological behavior described in Section III actually affects the behavior of the closed-loop system causing a longer converging time (see Figs. 4 and 5). This has been verified through a simulation campaign, as presented in Figs. 8 and 9, carried out to show the behavior of the controlled system for different values of the initial orientation error  $\theta_0$ . All the simulations have been performed by adopting the control algorithm proposed in Proposition 3 with  $p(0) = [0.3 \ 0 \ 0]^\top$  m,  $T = 0.05$  s,  $v_k^M =$

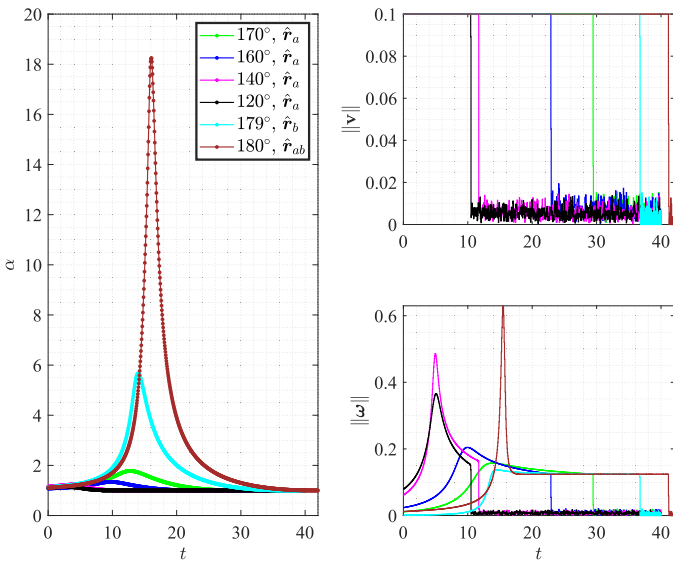


Fig. 9. Simulation results for values of the initial angle error  $\theta_0 = \{180^\circ, 179^\circ, 170^\circ, 160^\circ, 150^\circ, 140^\circ, 120^\circ\}$ . Left: value of  $\alpha$ . Right: translational and rotational commanded velocities.

0.1 m/s, and  $\omega_k^M = 0.75$  rad/s for all  $k$ . The response of the closed-loop system has been evaluated considering six different values of the initial axis-angle error  $\theta_0$  and  $\mathbf{r}(0)$ , i.e.,  $\theta_0 = 180^\circ, 179^\circ, 170^\circ, 160^\circ, 150^\circ$ , and  $120^\circ$  and  $\mathbf{r}(0) = \hat{\mathbf{r}}_a, \hat{\mathbf{r}}_b, \hat{\mathbf{r}}_{ab}$  with  $\hat{\mathbf{r}}_a$  and  $\hat{\mathbf{r}}_b$  the rotation axis corresponding to the undesired equilibria  $\hat{\mathbf{e}}_a$  and  $\hat{\mathbf{e}}_b$ , respectively, and  $\hat{\mathbf{r}}_{ab}$  is a rotation axis obtained as mean between  $\hat{\mathbf{r}}_a$  and  $\hat{\mathbf{r}}_b$ . These values are combined, as shown in the legends of Figs. 8 and 9. It addition, it is assumed that the depth measurements are affected by white noise corresponding to 1% of the current depth value. Fig. 8 displays the error convergence for each value of the initial angle error, with the left plot showing the plane  $(\|e_N\|, \|e_I\|)$ , while  $\|e\|$  is shown on the right plot. As expected, in all the considered cases, the error asymptotically converges to zero, but the initial angle plays a major role in the determination of the actual error trajectory. Consider first the cases where  $\mathbf{r}(0) = \hat{\mathbf{r}}_a$ . When  $\theta_0 = 170^\circ$ , the camera starts with a lower rotational velocity [see Fig. 9 (right bottom)]. In fact, until  $t = 5$  s, the camera motion is predominantly a translation, and the error trajectory decreases mostly along  $e_I$  [see again Fig. 8 (left)]. As the initial angle  $\theta_0$  decreases, this behavior is less noticeable, and when  $\theta_0 = 120^\circ$ , the error trajectory evolves from the very beginning along both  $e_I$  and  $e_N$ , and the camera rotates and translates at the same time. As a result, the overall error convergence is faster, as shown in Fig. 8 (right). Therefore, comparing the convergence times obtained for different values of the initial rotation errors, it can be concluded that, for  $\theta_0 < 140^\circ$ , the effect of the pathological behavior is almost negligible. Hence, only trajectories with large initial rotation errors are truly affected by the latency generated by small angular velocities. Unsurprisingly, such behavior is evident also considering a different initial rotation axis. When  $\mathbf{r}(0) = \hat{\mathbf{r}}_b$  and  $\theta = 179^\circ$ , the trajectory seems to evolve toward the undesired equilibrium, but, similar to the simulation in Fig. 4, it slowly converges to the origin. The same pathological behavior appears also when the angle is the

critical one  $\theta(0) = 180^\circ$ , but the rotation axis is neither of the three equilibrium ones (brown curves) showing that the angle plays a major role in the determination of such behavior. Note that, despite the brown and the cyan curves in Fig. 8 across the  $\rho_a$  line in the  $(\|e_N\|, \|e_I\|)$  plane, this does not mean that the trajectory crosses the manifold  $\mathcal{N}_e$  since the represented plane is just a convenient 2-D visual representation of the 3n-dimensional state space.

Finally, Fig. 9 shows the value  $\alpha = \|e\|/\|e_I\|$  used to define the upper bound of the gain in Proposition 2. From (29), we know that, near the origin,  $\alpha = 1$ . The plot shows that  $\alpha$  remains finite and close to 1 in all the simulated cases, and it reaches a maximum value of  $\alpha = 1.75$  in the  $\theta_0 = 170^\circ$  case.

## V. EXPERIMENTS

### A. Experimental Setup

The visual servoing algorithm has been tested in a collaborative in-store logistic scenario for shelf replenishment tasks. A Kuka LBR iiwa 7 robot has to pick various objects and place them on the shelves. The objects could be picked either from a desk or exchanged with the clerk by means of a handover maneuver. The end-effector is equipped with an Intel RealSense D435i RGB-D camera mounted in an *eye-in-hand* configuration. The grasping device is a WSG50 gripper by Weiss Robotics equipped with the SUNTouch force/tactile sensors [33]. The small clearance between the shelves limits the robot's collision-free workspace. To successfully execute the placing task, the robot dexterity is enhanced by the in-hand manipulation capabilities allowed by the slipping control algorithm described in [34] based on the dynamic friction model and the nonlinear observer proposed in [35]. To perform controlled sliding, the robot is required to grasp the object in prescribed grasping points with a maximum positioning error of 2 mm [21]. The visual servoing algorithm is considered correctly executed if the root mean square error  $e_M$  goes below 2 mm. A video of the experiments described in this section is available at <https://youtu.be/unNwifaBD4A>.

The algorithm runs on an ROS network. The iiwa robot is controlled via the fast robot interface (FRI) with a PC running Ubuntu with the PREEMPT\_RT real-time patch, ROS Melodic, Intel i7-8700K CPU @ 3.70 GHz, and 64 GB of RAM. The robot control rate is set to 1 kHz. The joint reference position commands are filtered via an additional first-order filter with a cutoff frequency of 100 Hz.

The RealSense camera is connected to the ROS network via the REALSENSE-ROS wrapper, the camera resolution is set to  $640 \times 480$  pixels, and the camera rate is set to the upper limit of 30 Hz. The default ROS wrapper publishes the depth image data with a depth resolution of 1 mm, whereas, by setting the camera driver to a depth resolution of 0.1 mm, we obtain an accuracy of 1% of the depth measurement in the range of 6 m, which is adequate for a collaborative task.

Since the FRI interface imposes a maximum joint displacement of 0.001 rad between two consecutive commands and the robot control rate is 1 kHz, the joint velocity limit  $\dot{q}_M$  results to be 1 rad/s. This has to be translated into Cartesian

velocity limits. The relationship between the Cartesian and joint velocity commands is given by

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q})\mathbf{v} \quad (76)$$

where  $\mathbf{q} \in \mathbb{R}^7$  is the joint position and  $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times 7}$  is the robot Jacobian. From (76), the squared norm of the joint velocity is

$$\dot{\mathbf{q}}^\top \dot{\mathbf{q}} = \mathbf{v}^\top \mathbf{J}^{\dagger\top}(\mathbf{q}) \mathbf{J}^\dagger(\mathbf{q}) \mathbf{v} \quad (77)$$

which is a quadratic form depending on the robot Jacobian. From (77), it is straightforward to find the time-varying Cartesian velocity limits that keep the joint speed below the constraint value  $\dot{q}_M$

$$\mathbf{v}^M = \left\| \frac{\dot{q}_M}{\sigma_1(\mathbf{J}(\mathbf{q}))} \boldsymbol{\sigma}_{v11}(\mathbf{J}(\mathbf{q})) \right\| \quad (78)$$

$$\boldsymbol{\omega}^M = \left\| \frac{\dot{q}_M}{\sigma_1(\mathbf{J}(\mathbf{q}))} \boldsymbol{\sigma}_{v21}(\mathbf{J}(\mathbf{q})) \right\| \quad (79)$$

where  $\sigma_1(\mathbf{J}(\mathbf{q}))$  is the largest singular value of the robot Jacobian and the vectors  $\boldsymbol{\sigma}_{v11}(\mathbf{J}(\mathbf{q}))$  and  $\boldsymbol{\sigma}_{v21}(\mathbf{J}(\mathbf{q})) \in \mathbb{R}^3$  contain the first and last three elements of the corresponding singular vector, respectively.

### B. Algorithm Implementation Details

We assume that the robot already knows which object has to grasp and the desired grasping pose that corresponds to the target image. To acquire the target image, we manually brought the robot to the grasp location such that the fingertips touch the object at the prescribed grasping points.

The 3-D feature points vector  $s_i$  and the corresponding target  $s_i^*$ ,  $i \in 1, \dots, n$ , are extracted and matched at the beginning ( $t = 0$ ) via the keypoint matching algorithm available in the ViSP library [24] (vpKeyPoint class) by using the first frame acquired by the camera and the target image. After the initial matching, the features are tracked only on the current image by using the keypoint tracking algorithm available in the same library (vpKltOpencv class).

### C. Simulation and Real Experiment Comparison

The sampled-data model of the visual servoing system and the proposed controller are available in a Simulink library at the following URL [https://github.com/Vanvitelli-Robotics/MATLAB\\_visual\\_servo](https://github.com/Vanvitelli-Robotics/MATLAB_visual_servo). All the simulated plots in this article are generated through this library.

This section compares the experimental visual servoing task with the simulated one to highlight that assuming the robot as a pure Cartesian motion device is reasonable.

The robot is commanded to grasp an object from the desk by using the algorithm proposed in Proposition 3 to synthesize the control action. The initial conditions are selected as  $\theta(0) = 17^\circ$ ,  $\mathbf{r}(0) = [0.0415 \ -0.9989 \ -0.0208]^\top$ , and  $\mathbf{p}(0) = [0.0647 \ 0.0146 \ 0.1145]^\top$  m. The landing function in (60)  $l(e_k)$  is designed taking into account the capability of the robot to decelerate. By assuming a uniformly decelerated motion at the end of the visual servoing task, an estimation of the deceleration space is  $3v^{M^2}/2a_M$ , where  $a_M$  is the maximum

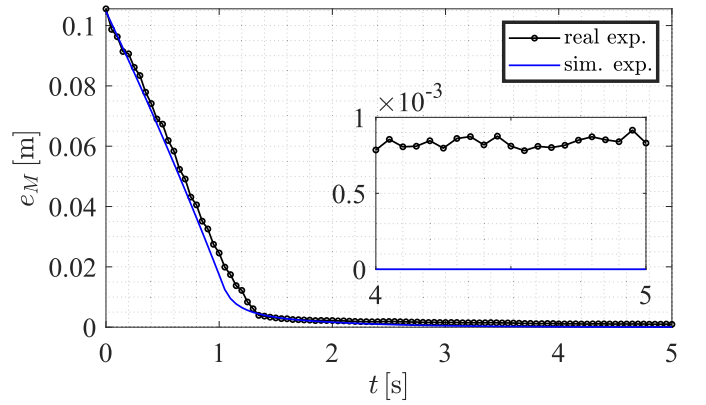


Fig. 10. Comparison of  $e_M$  between simulation and real experiment. Black line: real experiment. Blue line: simulation. The inner plot is a zoomed-in view that shows the difference in the neighborhood of  $e_M = 0$ .

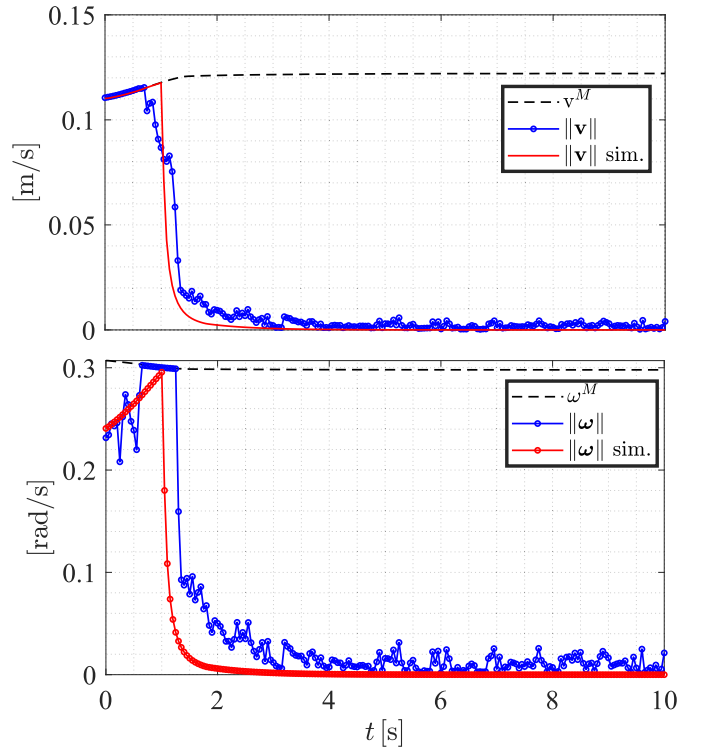


Fig. 11. Comparison between real (blue) and simulated (red) experiment. Top plot: camera control translational velocity norm. Bottom plot: camera control rotational velocity norm.

Cartesian acceleration that the robot can generate. Assuming that  $a_M$  is about  $5 \text{ m/s}^2$ , then  $e_H$  is set to 5 mm, while  $e_L = 2 \text{ mm}$  since it corresponds to the accepted bound on the steady-state error. Concerning  $\mu$ , when the root mean square error  $e_M$  is below  $e_L$ , assuming a pure translational motion, the translational velocity is upper bounded as  $\|\mathbf{v}\| \leq \mu e_L \sqrt{n}/T$ . Therefore,  $\mu = 0.25/\sqrt{n}$  has been selected in order to have a maximum velocity of 1 cm/s inside the bound defined by  $e_L$ .

The initial matched features  $s(0)$  and the corresponding target  $s^*$  are used to initialize the simulator too. Moreover, the same time-varying Cartesian velocity limits  $\mathbf{v}_k^M$  and  $\boldsymbol{\omega}_k^M$  of the real experiment [computed as in (78) and (79)] are used in the simulator.

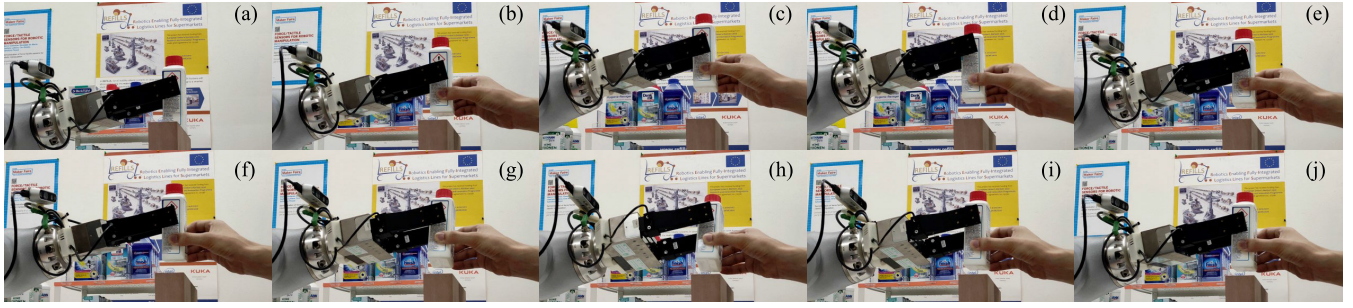


Fig. 12. Snapshots of the tracking experiment. Snapshot (a) shows the relative camera/object desired pose; snapshots (b)–(f) show the object being translated upwards; snapshots (g)–(j) show the object being rotated.

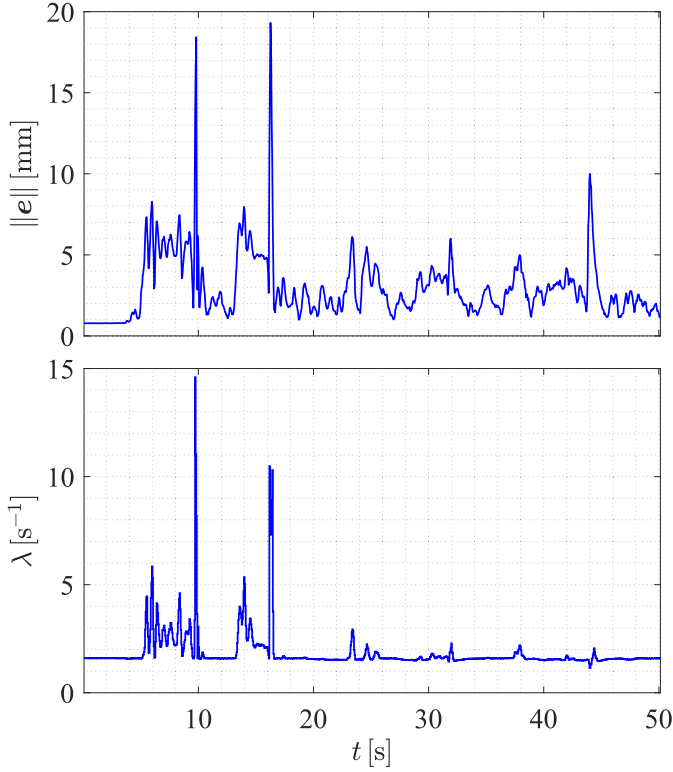


Fig. 13. Tracking experiment. Top plot: root mean square error  $e_M$ . Bottom plot: control gain computed by the proposed algorithm.

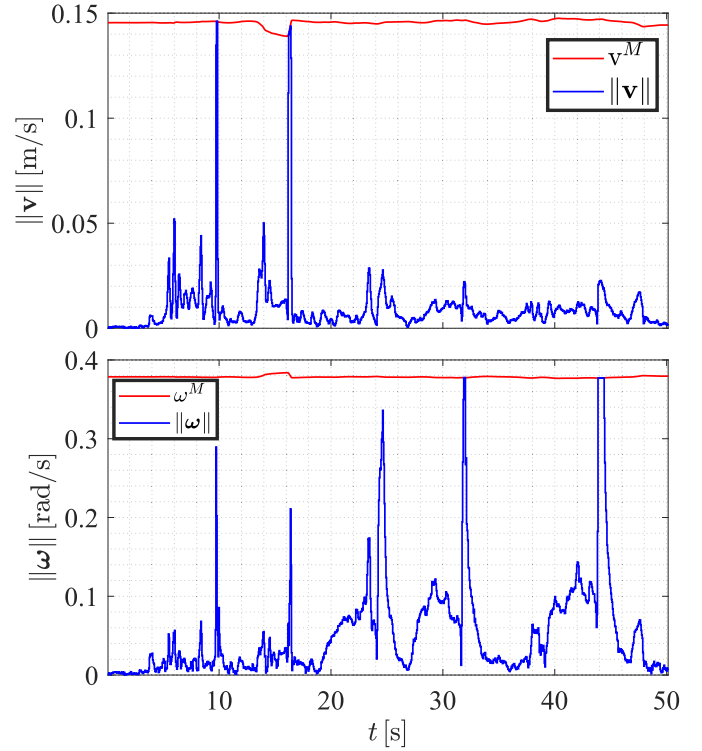


Fig. 14. Tracking experiment. Translational (top) and rotational (bottom) control velocities norm and the respective maximum values  $v^M$  and  $\omega^M$ .

Fig. 10 shows the results of the comparison. The blue and black lines represent the root mean square error  $e_M$  resulting from the simulation and the real experiment, respectively.

Remarkably, the blue and black lines have very similar behavior. The simulated  $e_M$  converges to zero asymptotically, while the one of the real experiment is ultimately bounded in a neighborhood of  $e = \mathbf{0}$  whose radius is less than 1 mm in spite of the feature error measurement uncertainty and the limited bandwidth of the robot control interface.

Fig. 11 (top) and (bottom) shows the norm of the translational and rotational controller output velocities, respectively, both for the real and simulated experiments and the velocity limits  $v_k^M$  and  $\omega_k^M$ . Note that the proposed visual servoing algorithm is able to not only ensure convergence of the feature error very close to zero but also push the robot up to the maximum of its performances.

#### D. Tracking Experiment

This section shows a tracking experiment in a collaborative scenario. The robot is commanded to align the camera to a target object, while a human partner moves the object. Note that, in this context, the word “tracking” refers to the task of following a moving object, while the reference features  $s^*$  are kept constant for the whole experiment. Thus, the object motion is a disturbance that the controller has to compensate for. This experiment is aimed at showing that, with the proposed control algorithm, the robot is capable of following the velocity imposed by the human partner up to its velocity limits. Fig. 12 shows the snapshots of the experiments, the first image shows the equilibrium configuration, and the robot has to track the object and maintain the same relative camera-object pose during the whole experiment. Fig. 13 shows the resulting mean square error  $e_M$  and the computed gain  $\lambda$ , while Fig. 14 shows the norm of the

translational and rotational control input and the respective limits.

In the beginning, between  $t = 5$  s and  $t = 9$  s, the human partner slowly rises the object up [see Fig. 12(2)], and the robot is able to follow it within its velocity limits; at  $t = 9$  s, the human partner stops, and the robot reaches again the desired relative pose [see Fig. 12(3)]. At  $t = 10$  s, the human partner rapidly moves the object down [see Fig. 12(4)]; this time, the robot is pushed at the edge of its translational velocity limits [see Fig. 14 (up)], and correspondingly, the error has a peak. At the same time, the gain  $\lambda$  [see Fig. 13 (bottom)], automatically computed by the control algorithm, increases to push the velocity up to the robot limits. Nevertheless, the robot is still able to counteract this disturbance, and at  $t = 11$  s, the robot reaches again the equilibrium relative pose [see Fig. 12(5)].

The same motion has been repeated in the orthogonal direction. The human partner first slowly moves the object toward the robot and then moves it back again with a fast motion. The results are very similar to the previous ones; during the slow motion, between  $t = 13$  s and  $t = 16$  s, the robot is able to follow the object by keeping the camera velocity within its limits. Once again, at  $t = 16$  s, when the human partner rapidly moves the object back, the robot is pushed to its translational velocity limits [see Fig. 14 (up)]. Finally, at the end of this phase ( $t = 18$  s), the robot reaches the equilibrium pose again, as shown in [see Fig. 12(6)].

The remaining part of the experiment is devoted to rotational motions. The human partner slowly rotates the object [see Fig. 12(7)] to reach the configuration in Fig. 12(8); then, with a fast rotational motion [see Fig. 12(9)], the human brings the object in its original orientation [see Fig. 12(10)]. Three repetitions of this rotational perturbation are performed and shown in the graphs between  $t = 19$  s and  $t = 50$  s. The behavior is qualitatively similar to the translational case, and the rotational velocity control plot reaches lower values (about 0.1 rad/s) during the slow human motions followed by higher peaks during the fast motions. Note how, during the last two fast rotational motions ( $t = 32$  s and  $t = 44$  s), the robot is pushed to its rotational velocity limits as the  $\|\omega\|$  signal in Fig. 14 (bottom) saturates to  $\omega^M$ .

The whole experiment followed by an additional phase where the human operator moves randomly in all the rotational and translational directions is available in the accompanying video.

### E. Pick&Place Experiments

This section shows pick&place experimental tasks in a collaborative scenario.

The object to pick is placed on the picking desk [see Fig. 15 (top left)], and the robot is commanded to pick and place it on a selected shelf [see Fig. 15 (top right)]. Two experiments are executed; in the first one, the robot performs the pick&place task autonomously. In the second one, the object position is perturbed by a human operator during the visual servoing subtask [see Fig. 15 (bottom)].

Fig. 16 shows the results of the two experiments. The top plot reports the root mean square error  $e_M$ , while the bottom

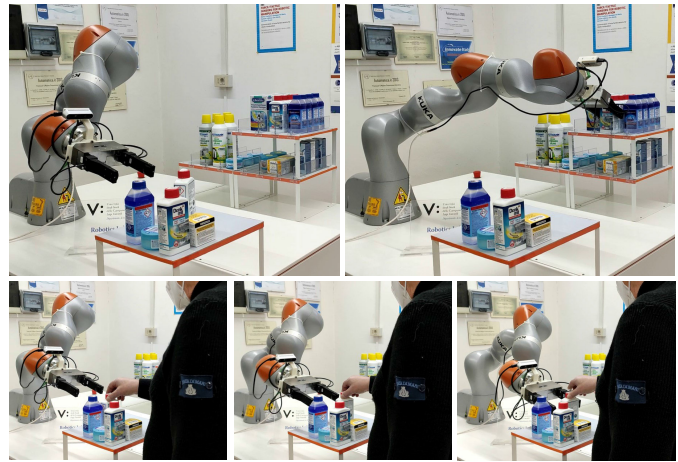


Fig. 15. Pick&place experiments. Top left: initial configuration. Top right: final placing configuration. Bottom: snapshots of a disturbance induced by a human operator who moves the object to be picked.

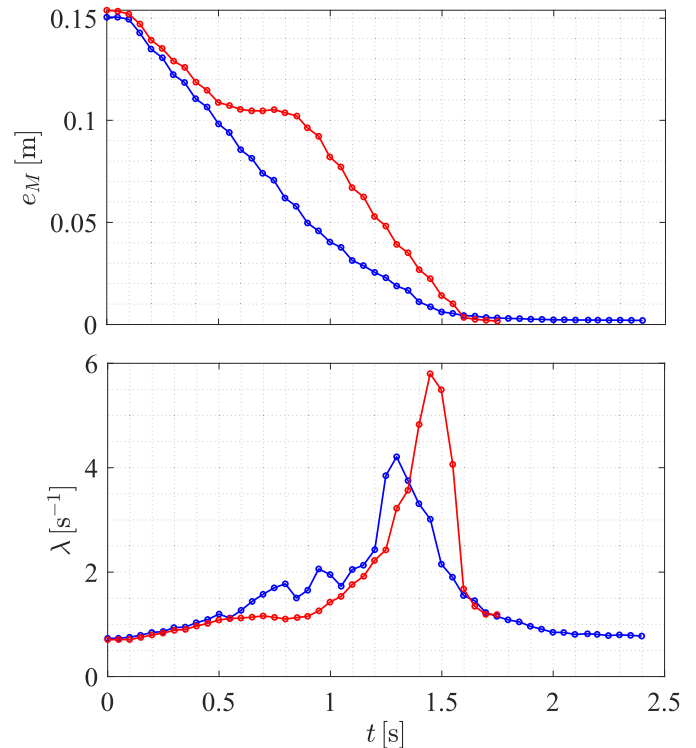


Fig. 16. Pick&place experiment. Top plot: root mean square errors: unperturbed experiment (blue curve) and perturbed one (red curve).

one reports the control gain  $\lambda$ . The blue lines represent the results obtained when the object pose is not perturbed by the human partner. The red lines represent the perturbed case. At about 0.5 s, the human partner changes the position of the object to be grasped to avoid the collision between the robot and another object present on the picking desk. The object position perturbation lasts 0.5 s and ends at  $t = 1$  s. During this time interval, the robot follows the moving object, and the root mean square error is almost constant. When the perturbation ends, the error decreases and converges toward zero.

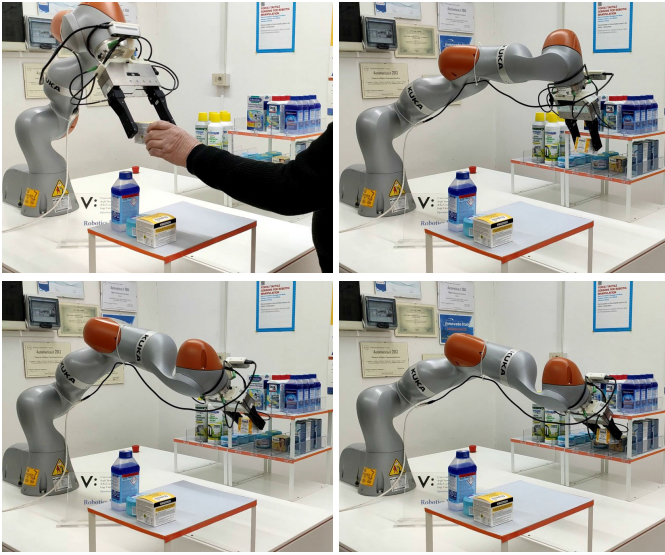


Fig. 17. Handover experiment performed with Object B. Top left: handover phase. Top right: preplacing configuration. Bottom left: after the change of the relative gripper pose orientation by means of the slipping control algorithm (the so-called pivoting maneuver). Bottom right: actual placing phase.

When the root mean square error  $e_M$  reaches the prescribed value of 2 mm, the visual servoing subtask is considered completed, and the robot grasps the object, moves it to reach the target location, and places it on the destination shelf to complete the pick&place task. The placing trajectory, after the visual servoing phase, has been planned offline by using the motion planners available in MOVEit [36].

#### F. Pick&Place Experiments With Human–Robot Handover

The human partner hands the object, picked from the desk, over to the robot. The human operator handles the object in the camera’s field of view, and the robot reaches the grasping location by means of the visual controller. Since the object is never perfectly still, the object pose is subject to continuous perturbations due to the unavoidable shaking of the human hand. A plastic bottle and a carton box are picked and placed in sequence.

The human operator shows the carton box to the robot, who grasps it by using the visual controller [see Fig. 17 (top left)]. Fig. 18 shows the error and the norm of the commanded translational and rotational velocities. Unsurprisingly, the plots are similar to the previous experiments, and once again, the robot is pushed close to its velocity limits. Note that, in less than 1 s, the visual servoing reaches the value  $e_M = 2$  mm, and the remaining time in the plot corresponds to the gripper closing and grasping phase. During the gripper closing, the visual servoing controller is still active to track the object and reject any perturbation given by the human operator (see the accompanying video). This time, the robot has to place the object between the two shelves on the right (see Fig. 17). Before the placing phase, the robot goes in front of the prescribed shelf in a configuration not feasible for the placing operation (top right figure). By using a gripper-pivoting maneuver, the robot is able to change on the fly the relative orientation between the gripper and the object to

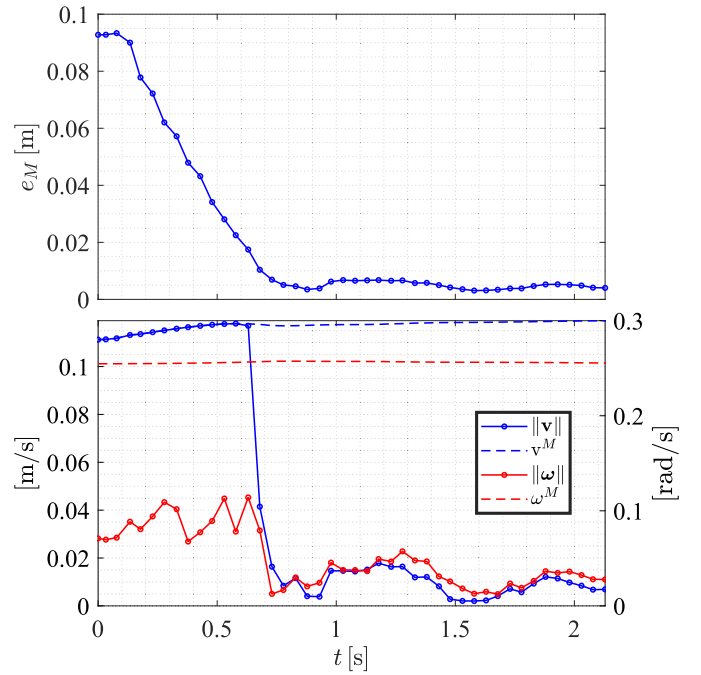


Fig. 18. Handover subtask performed with the carton box. Top: root mean square error  $e_M$ . Bottom: norm of the commanded camera translational velocity (left axis: blue curve) and norm of commanded camera rotational velocity (right axis: red curve). The dashed lines represent the velocity limits.

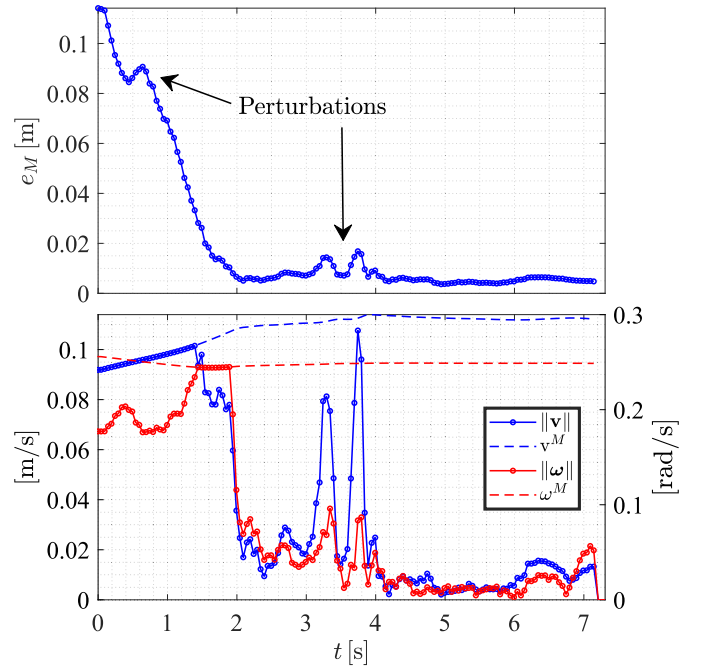


Fig. 19. Handover subtask performed with the plastic bottle. Top: root mean square error  $e_M$ . Bottom: norm of the commanded camera translational velocity (left axis: blue curve) and norm of commanded camera rotational velocity (right axis: red curve). The dashed lines represent the velocity limits.

reach a feasible configuration (bottom left figure; see also the accompanying video). Finally, the robot is able to place the object on the shelf (bottom right figure).

The same experiment is repeated with the plastic bottle. Fig. 19 shows the results of the visual servoing subtask. This time, the human operator intentionally applies larger

perturbations to the object pose. At about  $t = 0.5$  s, a first perturbation is applied, but the robot is already at the edge of the velocity constraint, and the control is not able to counteract the perturbation; thus, the error increases. When the perturbation ends, the error decreases again. This is visible in Fig. 19 (top). Between  $t = 3$  s and  $t = 4$  s, another perturbation is applied when the error is very close to the equilibrium point; in this case, the visual control counteracts this event by increasing the velocity command. This is visible in the two peaks in the bottom plot corresponding to the two peaks that correspond to the perturbations in the top plot.

## VI. CONCLUSION

In this article, we presented a novel IBVS with 3-D features. A sampled-data model is presented to describe the exact closed-loop dynamics in the sampling times while retaining the rigid motion constraint. Useful insights into the well-known undesired equilibria of an IBVS scheme are provided to characterize pathological situations in the application of an IBVS control strategy based on 3-D feature points.

The control design is aimed at pushing the robotic system performances up in terms of accuracy and execution speed. The algorithm is based on an exact sampled-data model of the visual servoing system and a discrete-time Lyapunov analysis. The input to the robot controller has been synthesized by adapting the control gain solving a constrained optimization problem. The constraints refer to the camera Cartesian velocity limits, depending on the robot configuration and joint velocity limits, and the monotonic decreasing in the continuous time of the feature error dynamics. Even though the algorithm assumes that the robot is an ideal Cartesian motion generator, the experimental results in the real environment have shown that such an assumption is entirely reasonable.

Three pick&place experiments, involving a human partner and requiring handover maneuvers and in-hand dexterous handling, have been presented. The accompanying video shows that the robot exhibits a high accuracy in grasping and placing the objects and behaves, in terms of speed of execution, in a way comparable to the human operator, satisfying the requirement for the optimal sharing of a collaborative logistic task.

## REFERENCES

- [1] P. I. Corke, "Visual control of robot manipulators—A review," in *Visual Servoing*. Singapore: World Scientific, Oct. 1993, pp. 1–31.
- [2] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 679–699, Aug. 2002.
- [3] E. Malis, G. Chesi, and R. Cipolla, "2½d visual servoing with respect to planar contours having complex and unknown shapes," *Int. J. Robot. Res.*, vol. 22, nos. 10–11, pp. 841–853, 2003.
- [4] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3D visual features," in *Proc. Committees, Econometrica*, 1996, pp. 497–502.
- [5] E. Cervera, A. P. del Pobil, F. Berry, and P. Martinet, "Improving image-based visual servoing with three-dimensional features," *Int. J. Robot. Res.*, vol. 22, nos. 10–11, pp. 821–839, Oct. 2003.
- [6] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
- [7] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006. [Online]. Available: <https://hal.inria.fr/inria-00350283>
- [8] F. Chaumette and S. Hutchinson, "Visual servo control, part II: Advanced approaches," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Dec. 2007. [Online]. Available: <https://hal.inria.fr/inria-00350638>
- [9] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 828–834, Aug. 2011. [Online]. Available: <https://hal.inria.fr/inria-00629834>
- [10] C. Teulière and E. Marchand, "A dense and direct approach to visual servoing using depth maps," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1242–1249, Oct. 2014.
- [11] J. Issac, M. Wüthrich, C. G. Cifuentes, J. Bohg, S. Trimpe, and S. Schaal, "Depth-based object tracking using a robust Gaussian filter," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 608–615.
- [12] Q. Bateau, E. Marchand, J. Leitner, F. Chaumette, and P. I. Corke, "Training deep neural networks for visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Brisbane, QLD, Australia, May 2018, pp. 3307–3314. [Online]. Available: <https://hal.inria.fr/hal-01716679>
- [13] F. Tokuda, S. Arai, and K. Kosuge, "Convolutional neural network-based visual servoing for eye-to-hand manipulator," *IEEE Access*, vol. 9, pp. 91820–91835, 2021.
- [14] M. N. Qureshi et al., "RTVS: A lightweight differentiable MPC framework for real-time visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3798–3805.
- [15] Y. V. S. Harish, H. Pandya, A. Gaud, S. Terupally, S. Shankar, and K. M. Krishna, "DFVS: Deep flow guided scene agnostic image based visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9000–9006.
- [16] N. A. Piga, "Hybrid architectures for object pose and velocity tracking at the intersection of Kalman filtering and machine learning," Ph.D. dissertation, Dept. Inform., Bioeng., Robot. Syst. Eng., Univ. Genova, Genoa, Italy, 2022.
- [17] P. Corke, *Robotics, Vision and Control*. Berlin, Germany: Springer Nature, 2017.
- [18] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, no. 237, D. Kriegman, G. Hager, and A. Morse, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 66–78. [Online]. Available: <https://hal.inria.fr/inria-00352553>
- [19] F. Schramm, G. Morel, A. Micaelli, and A. Lottin, "Extended-2D visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2004, pp. 267–273.
- [20] M. Costanzo, G. De Maria, C. Natale, and A. Russo, "Stability and convergence analysis of 3D feature-based visual servoing," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12022–12029, Oct. 2022.
- [21] M. Costanzo, G. De Maria, G. Lettera, and C. Natale, "Can robots refill a supermarket shelf?: Motion planning and grasp control," *IEEE Robot. Autom. Mag.*, vol. 28, no. 2, pp. 61–73, Jun. 2021.
- [22] M. Costanzo, G. De Maria, and C. Natale, "Handover control for human–robot and robot–robot collaboration," *Frontiers Robot. AI*, vol. 8, p. 132, May 2021.
- [23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [24] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: A generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.
- [25] F. Coticelli and B. Allotta, "Discrete-time robot visual feedback in 3D positioning tasks with depth adaptation," *IEEE/ASME Trans. Mechatronics*, vol. 6, no. 3, pp. 356–363, Sep. 2001.
- [26] J. E. Solanes, J. Tornero, L. Armesto, and V. Gírbés, "Multi-rate visual servoing based on dual-rate high order holds," in *Towards Autonomous Robotic Systems*. Berlin, Germany: Springer, 2011, pp. 195–206.
- [27] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 933–939, Oct. 2010. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00530203>
- [28] I. S. Mohamed, G. Allibert, and P. Martinet, "Sampling-based MPC for constrained vision based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3753–3758.
- [29] A. Rantzer, "A dual to Lyapunov's stability theorem," *Syst. Control Lett.*, vol. 42, pp. 161–168, Mar. 2001.
- [30] M. Costanzo, G. De Maria, and C. Natale, "Two-fingered in-hand object handling based on force/tactile feedback," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 157–173, Feb. 2020.

- [31] J. S. Dai, "Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections," *Mechanism Mach. Theory*, vol. 92, pp. 144–152, Oct. 2015.
- [32] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, Dec. 1985. [Online]. Available: <https://doi.org/10.1017/cbo9780511810817>
- [33] M. Costanzo, G. De Maria, C. Natale, and S. Pirozzi, "Design and calibration of a force/tactile sensor for dexterous manipulation," *Sensors*, vol. 19, no. 4, p. 966, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/4/966>
- [34] M. Costanzo, "Control of robotic object pivoting based on tactile sensing," *Mechatronics*, vol. 76, Jun. 2021, Art. no. 102545.
- [35] A. Cavallo, M. Costanzo, G. De Maria, and C. Natale, "Modeling and slipping control of a planar slider," *Automatica*, vol. 115, May 2020, Art. no. 108875.
- [36] T. C. David. (2014). *Reducing the Barrier to Entry of Complex Robotic Software: A MoveIt! Case Study*. [Online]. Available: <https://aisberg.unibg.it/handle/10446/87657>



**Giuseppe De Maria** has been a full Professor of automatic control with the Università degli Studi della Campania Luigi Vanvitelli, Aversa, Italy, since 1994. His research interests include active noise and vibration control of flexible structures, robust control of mechanical systems, and industrial and advanced robotics. His current research interests are focused on the development of force and tactile sensors for grasping and in-hand manipulation in professional service robotics.



**Ciro Natale** is currently a Full Professor of robotics and automatic control with the Università degli Studi della Campania Luigi Vanvitelli, Aversa, Italy, where he is also a Coordinator of the Research Doctorate Course in Industrial and Information Engineering. He has authored or coauthored more than 130 journal articles and conference papers, including two monographs. His research interests include modeling and control of industrial manipulators, force and visual control, manipulation based on force-tactile sensing, and safe human–robot interaction.

Dr. Natale has served as an Associate Editor for the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY and the IEEE ROBOTICS AND AUTOMATION LETTERS. He is also an Associate Editor of *Automatica*.



**Marco Costanzo** (Member, IEEE) received the Ph.D. degree in industrial and information engineering from the Università degli Studi della Campania Luigi Vanvitelli, Aversa, Italy, in 2021.

He is currently a Researcher with the Università degli Studi della Campania Luigi Vanvitelli. His research interests range from robotic manipulation based on force/tactile sensing to multimodal sensing for safe human–robot interaction.

Dr. Costanzo serves as an Associate Editor for the IEEE ROBOTICS AND AUTOMATION LETTERS.



**Antonio Russo** received the Ph.D. degree in industrial and information engineering from the Università degli Studi della Campania Luigi Vanvitelli, Aversa, Italy, in 2021.

He is currently a Researcher with the Università degli Studi della Campania Luigi Vanvitelli. His research interests include nonlinear control, variable structure control of the sliding mode type, stability analysis of switched systems, and control of power systems in the framework of more electric aircraft.