IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Stability and Convergence Analysis of 3D Feature-Based Visual Servoing

Marco Costanzo, Giuseppe De Maria, Ciro Natale, and Antonio Russo

***Abstract*—Visual control based on image features has received much attention for its inherent robustness against camera calibration errors, modelling uncertainties and the capability to keep the object in the Field of View (FoV) of the camera. Nevertheless, some drawbacks related to the basin of convergence and the existence of local minima, which make the camera to get stuck in undesired equilibrium points, are still worth being investigated. Nowadays, the availability of cheap and lightweight RGB-D cameras makes the use of three-dimensional features natural. By using an RGB-D camera in an eye-in-hand configuration, this letter proposes an in-depth stability and convergence analysis of 3D feature-based visual servoing. It will be proved that the visual control system is *almost globally asymptotically stable* in the sense that the only trajectories not converging to the desired equilibrium point are those belonging to a zero Lebesgue measure set in the feature space. Moreover, a sufficient condition guaranteeing that the feature trajectories remain in the camera FoV is derived and an algorithm to prevent feature loss caused by violation of the camera FoV constraint is proposed.**

***Index Terms*—Visual Servoing, Visual Control, Sensor-based Control.**

I. Introduction

VISUAL control refers to the use of vision data acquired by a camera to provide feedback position control for the robot end effector. The objective of navigating and positioning the robot to a desired location, with a target configuration, is achieved by the vision controller based on features extracted from the image. Hand-crafted features as points, lines, and contours, employed for visual servoing, (see [1], [2] and [3] and references therein), or three dimensional features, obtained by combining 2D and 3D information [4], [5], are extracted from the image. In Position-Based Visual Servoing (PBVS) these features are used to estimate the camera pose directly in the 3D space and the control algorithm aims at minimizing the mismatch between the current pose and the target one. In the other classical approach, referred to as Image-Based Visual Servoing (IBVS), the control aims at minimizing the error between the current image features and the target ones directly in the feature space. Reference tutorials for such approaches are [6], [7], [8].

Historically, the IBVS control technique has received more attention than the PBVS one, due to its robustness against camera calibration errors [9]. The main advantage of the IBVS control scheme is that it does not require the 3D pose reconstruction, but it straightforwardly utilizes the visual features extracted from the current image, acquired by the camera. These features are matched with the ones of the target image acquired beforehand, and, then, the tracking module provides, frame by frame, the feature vector as input to the visual controller. Since the image features are directly controlled, it is likely to occur that the IBVS controller will be able to keep them in the camera Field of View (FoV) [5]. In spite of the appealing peculiarities mentioned above, IBVS suffers from the drawback of the existence of singular configurations in the work space or of local minima in the feature error space when the full six degrees of freedom are controlled. Such drawback directly arises from the nature of the interaction matrix describing the relationship between the rate of variation of the feature vector and the camera velocity. Therefore, only local asymptotic stability of the desired equilibrium point can be ensured. The local stability nature of the desired equilibrium point turns out to be critical when the initial and the desired camera positions are distant. To overcome the drawback of local minima, in [10] the authors introduce an approach which couples path planning in the image space and image-based control. They design a target reference trajectory in the image space under the constraint that the object remains in the camera FoV.

On the other side, considering the position-based visual control approach, while on the one hand it does not suffer from the problems of the existence of local minima or singularities of the corresponding interaction matrix, on the other hand it is sensitive to camera calibration errors, it requires the 3D pose reconstruction and it may suffer from the problem of the features going out of the camera FoV.

To take advantage of the positive aspects of IBVS and PBVS, a mixed 2D-3D approach has been proposed in [11] consisting in combining visual features, obtained directly from the image, with features expressed in the Euclidean space. The stability analysis of such a scheme in presence of camera calibration errors has been proposed in [12]. Another possible strategy to combine 2D and 3D information is proposed in [5] by introducing 3D visual features. While 3D visual features were originally proposed in [4] by introducing the notion of 3D logical vision sensor, in [5] analytical results are presented for the camera velocity screw, which highlights interesting aspects of visual servoing (VS) with 3D features. The use of 3D visual features is motivated by the argument that IBVS, using image point coordinates, requires depth estimation for each feature point to compute the interaction matrix. It is argued

Manuscript received: July, 28, 2022; Accepted September, 20, 2022.

This paper was recommended for publication by Editor Eric Marchand upon evaluation of the Associate Editor and Reviewers' comments.

Marco Costanzo, Giuseppe De Maria, Ciro Natale and Antonio Russo are with Dipartimento di Ingegneria, Università degli Studi della Campania "Luigi Vanvitelli", 81031 Aversa (CE), Italy. *Corresponding author: Marco Costanzo, Email:* marco.costanzo@unicampania.it

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

that by estimating the depth at each control iteration, instead of using the depth at the equilibrium, as some approaches proposed in the literature, a better behaviour of VS could be obtained. In [5] the authors use a stereo camera in an *eye-in-hand* configuration and the depth is estimated from pixel disparity [13]. As the feature vector is linearly dependent on the 3D point coordinates, the object moves along a straight path in the camera frame, and it is likely to remain in the camera FoV during the visual servoing task. The stability of the VS control scheme with 3D features is analyzed in [14], by using the isomorphism between $\mathrm{SE}(3)$ and the subset of the feature space to which the feature vector must belong to comply with the rigid motion constraint.

In this letter, with reference to the 3D feature-based VS, we first inspect the stability properties of the error dynamics equilibrium points. Starting from the results in [14], we prove that the closed-loop system exhibits a desired equilibrium point being *almost globally asymptotically stable*, since the only trajectories not converging to it are those belonging to zero Lebesgue measure sets [15]. Moreover, by deriving the expression of the camera velocity screw, we highlight anomalous behavior of the feature error dynamics that advises against starting from such initial conditions. Finally, starting from the work in [5] we derive a sufficient condition in order for the feature trajectories to remain in the camera FoV during the task and we propose an algorithm to prevent feature loss caused by violation of the camera FoV constraint.

II. VS Model and stability properties

In this section the VS dynamic model is introduced and its stability properties are investigated.

A. The model

The VS model is based on the data provided by an RGB-D camera, which consist in RGB images along with per-pixel depth information. Let $\boldsymbol{p}_i$ be a 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 $\boldsymbol{K}$ the camera calibration matrix

$$\boldsymbol{K} = \begin{bmatrix} F_x & 0 & C_x \\ 0 & F_y & C_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

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

$$\boldsymbol{p}_i = \boldsymbol{K}^{-1} \begin{bmatrix} u_i z_i & v_i z_i & z_i \end{bmatrix}^\top. \tag{2}$$

We assume that the feature vectors of the captured RGB-D image, indicated hereafter with $\boldsymbol{s}_i$, are composed of the 3D coordinates of the points $\boldsymbol{p}_i$, i.e., $\boldsymbol{s}_i = \boldsymbol{p}_i$. The VS 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 $\boldsymbol{s}_i, \boldsymbol{s}_i^\star \in \mathbb{R}^3, i = 1, \ldots, n$ be the i-th current and target feature vectors, respectively, then $\boldsymbol{s}, \boldsymbol{s}^\star \in \mathbb{R}^{3n}$

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_1^\top & \ldots & \boldsymbol{s}_n^\top \end{bmatrix}^\top, \tag{3}$$

$$\boldsymbol{s}^\star = \begin{bmatrix} \boldsymbol{s}_1^{\star\top} & \ldots & \boldsymbol{s}_n^{\star\top} \end{bmatrix}^\top, \tag{4}$$

represent the vectors of n matched image features.

Denote with $\boldsymbol{v} = \begin{bmatrix} \mathbf{v}^\top & \boldsymbol{\omega}^\top \end{bmatrix}^\top \in \mathbb{R}^6$ the body velocity screw [16] of the camera, where $\mathbf{v}$ is the linear velocity and $\boldsymbol{\omega}$ is the angular velocity. The link between the rate of $\boldsymbol{s}(t)$ and the velocity screw $\boldsymbol{v}(t)$ is described by the following relationship

$$\dot{\boldsymbol{s}}(t) = \boldsymbol{L}(\boldsymbol{s})\boldsymbol{v}(t), \quad \boldsymbol{s}(0) = \boldsymbol{s}_0, \tag{5}$$

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

$$\boldsymbol{L}(\boldsymbol{s}) = \begin{bmatrix} -\boldsymbol{I}_3 & \ldots & -\boldsymbol{I}_3 \\ \boldsymbol{S}^\top(\boldsymbol{s}_1) & \ldots & \boldsymbol{S}^\top(\boldsymbol{s}_n) \end{bmatrix}^\top, \tag{6}$$

where $\boldsymbol{I}_3$ indicates the 3×3 identity matrix and $\boldsymbol{S}(\boldsymbol{s}_i)$ is the skew symmetric operator applied to the vector $\boldsymbol{s}_i$, such that $\boldsymbol{S}(\boldsymbol{s}_i)\boldsymbol{\omega} = \boldsymbol{s}_i \times \boldsymbol{\omega}$.

Assuming that there exist at least 3 distinct non-aligned features, the matrix $\boldsymbol{L}(\boldsymbol{s})$ has full rank, i.e., $\mathrm{rank}(\boldsymbol{L}(\boldsymbol{s})) = 6$.

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

$$\boldsymbol{e}(t) = \boldsymbol{s}(t) - \boldsymbol{s}^\star, \tag{7}$$

then the distance between the current feature and the target one is represented by $\|\boldsymbol{e}(t)\|$.

Denoting with $\boldsymbol{L}_e = \boldsymbol{L}(\boldsymbol{e} + \boldsymbol{s}^\star)$, the dynamics of the current feature error can be easily derived from (5) and (7) as

$$\dot{\boldsymbol{e}}(t) = \boldsymbol{L}_{\boldsymbol{e}}\boldsymbol{v}(t), \quad \boldsymbol{e}(0) = \boldsymbol{e}_0. \tag{8}$$

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

The VS control scheme aims at regulating the feature error $\boldsymbol{e}$ to zero. The classical approach is to design a velocity control law which aims at guaranteeing a decoupled and exponential decrease of the error, that means assuming as reference model for the error dynamics the decoupled linear system $\dot{\boldsymbol{\epsilon}} = -\lambda\boldsymbol{\epsilon}$ [7].

B. Stability properties

The control input for the feature error system (8) is the camera velocity screw $\boldsymbol{v}(t)$ in the camera reference frame. Hereafter, we assume that: the robot accepts Cartesian velocity commands and can be considered as an ideal Cartesian motion device; the camera calibration is perfect and the depth measurement uncertainty can be neglected. The particular reference model selected for the error dynamics makes it straightforward to derive the control $\boldsymbol{v}$ to the robot as

$$\boldsymbol{v} = -\lambda\boldsymbol{L}_{\boldsymbol{e}}^\dagger\boldsymbol{e}, \tag{9}$$

where $\boldsymbol{L}_e^\dagger \in \mathbb{R}^{6 \times 3n}$ is the Moore–Penrose pseudoinverse of $\boldsymbol{L}_e$, and λ is a positive constant gain. Therefore, the visual servoing error dynamics is

$$\dot{\boldsymbol{e}} = -\lambda\boldsymbol{L}_e\boldsymbol{L}_e^\dagger\boldsymbol{e}, \quad \boldsymbol{e}(0) = \boldsymbol{e}_0. \tag{10}$$

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

Furthermore, the configurations of the feature vector $\boldsymbol{s}$ and, consequently, of the error $\boldsymbol{e}$ are subject to the rigid motion constraint, that is

$$\exists \boldsymbol{R}(\boldsymbol{r},\theta), \boldsymbol{p} : \boldsymbol{s} = \bar{\boldsymbol{R}}\boldsymbol{s}^\star + \bar{\boldsymbol{p}}, \tag{11}$$

$$\bar{\boldsymbol{R}} = \boldsymbol{I}_n \otimes \boldsymbol{R}(\boldsymbol{r},\theta),\ \bar{\boldsymbol{p}} = \mathbf{1}_n \otimes \boldsymbol{p} \tag{12}$$

where the symbol $\otimes$ indicates the Kronecker product, $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\mathbf{1}_n = \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}^\top \in \mathbb{R}^n$. The rotational matrix $\boldsymbol{R}(\boldsymbol{r},\theta)$, where $\boldsymbol{r}$ and θ are the corresponding unit axis and the rotation angle, and the translation vector $\boldsymbol{p}$ describe the displacement from the target feature vector to the current one.

In view of the rigid motion constraint expressed in (11), the visual servoing dynamics can be characterized in SE(3) by means of the variables θ, $\boldsymbol{r}$, and $\boldsymbol{p}$. In [14], the authors proved that there exist only three feasible configurations $\hat{\boldsymbol{s}}_x,\ x = a, b, c$, of the feature vector $\boldsymbol{s}$, given the target $\boldsymbol{s}^\star$, which result in undesired equilibrium configurations with $\boldsymbol{e} \in \ker(\boldsymbol{L}_e^\top)$ and $\boldsymbol{s} \neq \boldsymbol{s}^\star$. The three configurations are characterized by $(\theta, \boldsymbol{p}, \boldsymbol{r}) = (\hat{\theta}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{r}})$ that satisfy the following conditions

$$\hat{\theta} = \pm\pi, \tag{13a}$$

$$\hat{\boldsymbol{p}} = \left(\boldsymbol{I}_3 - \hat{\boldsymbol{R}}\right)\boldsymbol{s}_\Sigma^\star, \tag{13b}$$

$$\hat{\boldsymbol{r}} = \text{eigvect}\,\widehat{\boldsymbol{M}}, \tag{13c}$$

where

$$\hat{\boldsymbol{R}} = \boldsymbol{R}(\hat{\boldsymbol{r}},\hat{\theta}), \tag{14}$$

$$\widehat{\boldsymbol{M}} = \sum_{i=1}^{n}\left(\boldsymbol{s}_i^\star\boldsymbol{s}_i^{\star\top}\right) - n\boldsymbol{s}_\Sigma^\star\boldsymbol{s}_\Sigma^{\star\top} \tag{15}$$

$$\boldsymbol{s}_\Sigma^\star = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{s}_i^\star. \tag{16}$$

Even though in [14] the authors proved that such equilibrium points are unstable, the behaviour of the feature error dynamics close to these points has not been fully investigated, also in later literature, to the best of our knowledge. As this behaviour affects the performances of the robotic system, we provide in the following a deep insight of the properties of the undesired equilibria.

It is noteworthy to outline that $\boldsymbol{L}_e\boldsymbol{L}_e^\dagger$ in (10) is the projection matrix on the $\text{Im}(\boldsymbol{L}_e)$. We denote with

- $\boldsymbol{e}_I$ the projection of $\boldsymbol{e}$ on the $\text{Im}(\boldsymbol{L}_e)$, that is $\boldsymbol{e}_I = \boldsymbol{L}_e\boldsymbol{L}_e^\dagger\boldsymbol{e}$;
- $\boldsymbol{e}_N$ the projection of $\boldsymbol{e}$ on the $\ker(\boldsymbol{L}_e^\top)$, that is $\boldsymbol{e}_N = (\boldsymbol{I}_{3n} - \boldsymbol{L}_e\boldsymbol{L}_e^\dagger)\boldsymbol{e}$.

Then, the following lemma holds.

Lemma 1. *Let $\mathcal{D} \subset \mathbb{R}^{3n}$ be an open neighborhood of the equilibrium point $\boldsymbol{e} = \mathbf{0}$, which does not contain any point satisfying conditions* (13)*, then*

$$\lim_{t\to\infty}\frac{\|\boldsymbol{e}(t)\|}{\|\boldsymbol{e}_I(t)\|} = 1, \tag{17}$$

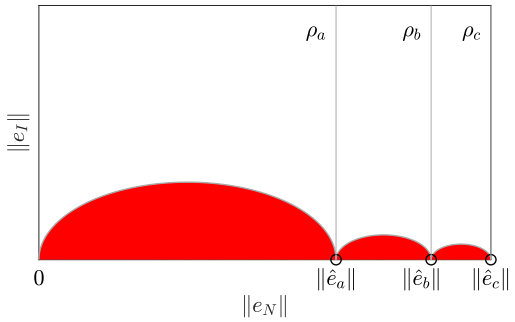

Fig. 1. Plane $(\|\boldsymbol{e}_N\|, \|\boldsymbol{e}_I\|)$, the undesired equilibria are reported on the horizontal axis. The points in the red area do not satisfy the rigid motion constraint.

for any trajectory starting with $\boldsymbol{e}_0 \in \mathcal{D}$.

Proof. By exploiting the Euler-Rodrigues rotation formula [17] $\boldsymbol{R} = e^{\boldsymbol{S}(\theta\boldsymbol{r})} = \boldsymbol{I}_3 + \sin\theta\boldsymbol{S}(\boldsymbol{r}) + (1-\cos\theta)\boldsymbol{S}^2(\boldsymbol{r})$ in a neighbourhood of the origin (i.e., $\boldsymbol{s} = \boldsymbol{s}^\star$), the rotation matrix $\boldsymbol{R}$ in (11) can be written as $\boldsymbol{R} = \boldsymbol{I}_3 + \boldsymbol{S}(\boldsymbol{r})\,\mathrm{d}\theta$, hence equation (11) becomes

$$\begin{aligned}\boldsymbol{s} &= (\boldsymbol{I}_{3n} + \bar{\boldsymbol{S}}(\boldsymbol{r})\,\mathrm{d}\theta)\boldsymbol{s}^\star + \mathrm{d}\bar{\boldsymbol{p}} = \boldsymbol{s}^\star - \left(-\mathrm{d}\bar{\boldsymbol{p}} + \bar{\boldsymbol{S}}(\boldsymbol{s}^\star)\boldsymbol{r}\,\mathrm{d}\theta\right)\\ &= \boldsymbol{s}^\star - \boldsymbol{L}(\boldsymbol{s}^\star)\begin{bmatrix}\mathrm{d}\boldsymbol{p}\\ \boldsymbol{r}\,\mathrm{d}\theta\end{bmatrix} = \boldsymbol{s}^\star - \boldsymbol{L}(\boldsymbol{s}^\star)\boldsymbol{v}\,\mathrm{d}t.\end{aligned} \tag{18}$$

Thus, it holds that $\boldsymbol{e} = -\boldsymbol{L}(\boldsymbol{s}^\star)\boldsymbol{v}\,\mathrm{d}t$ is a vector in the range of the matrix $\boldsymbol{L}(\boldsymbol{s}^\star)$ with zero component along $\ker(\boldsymbol{L}_e^\top)$. Then, since $\boldsymbol{e} = \mathbf{0}$ is the only asymptotically stable equilibrium point in $\mathcal{D}$ [7], (17) holds. □

The rigid motion constraint (11) implies that the feature vector $\boldsymbol{s}$ must belong to the subset $\mathcal{R}^\star \subset \mathbb{R}^{3n}$ (isomorphic to SE(3)) defined, for a given $\boldsymbol{s}^\star$, as the set of all the feature vectors $\boldsymbol{s}$ satisfying equation (11). This argument is synthetically illustrated in Fig. 1 depicting the plane $(\|\boldsymbol{e}_N\|, \|\boldsymbol{e}_I\|)$. The figure has been constructed by generating random values of $\boldsymbol{r} \in \mathbb{R}^3$ on the unit sphere, $\theta \in [-\pi, \pi]$ and $\boldsymbol{p} \in \mathbb{R}^3$ in a unit cube, thus computing the corresponding error vector $\boldsymbol{e}$ and the norm of the projections $\|\boldsymbol{e}_N\|$, $\|\boldsymbol{e}_I\|$. The points in the red areas are those never generated through this procedure as they do not satisfy the constraint (11). The boundaries of such areas result to be tangent to the $\|\boldsymbol{e}_I\|$ axis and to the vertical lines ρ_x, passing through $\|\boldsymbol{e}_N\| = \|\hat{\boldsymbol{e}}_x\|$, being $\hat{\boldsymbol{e}}_x = \hat{\boldsymbol{s}}_x - \boldsymbol{s}^\star$, with $x = a, b, c$. This is not surprising in view of Lemma 1, which holds also in a neighborhood of $\hat{\boldsymbol{e}}_x$ as can be easily verified by applying the translation $\boldsymbol{s} - \hat{\boldsymbol{s}}_x$ and assuming that trajectories converging to $\hat{\boldsymbol{e}}_x$ exist.

As the following analysis equivalently holds for all of the undesired equilibria, the x-subscript has been dropped hereafter for the sake of notation simplicity.

Before proving the existence of trajectories converging to points $\hat{\boldsymbol{e}}$, we show that any translational perturbation $\Delta\bar{\boldsymbol{p}} = \mathbf{1}_n \otimes \Delta\boldsymbol{p}$ on the equilibrium feature configuration $\hat{\boldsymbol{s}}$ yields a feature vector $\boldsymbol{s} = \hat{\boldsymbol{s}} + \Delta\bar{\boldsymbol{p}}$ such that $\boldsymbol{s} \in \mathcal{N}_s$, where $\mathcal{N}_s = \left\{\boldsymbol{s} \in \mathbb{R}^{3n} | \boldsymbol{s} = \hat{\boldsymbol{s}} + \Delta\bar{\boldsymbol{p}}\right\}$ is the set of vectors $\boldsymbol{s}$ such that $\boldsymbol{e}_N = \hat{\boldsymbol{e}}$, or, equivalently, the point $(\|\boldsymbol{e}_N\|, \|\boldsymbol{e}_I\|)$ belongs to the line ρ in Fig. 1. Denoting with $\hat{\bar{\boldsymbol{p}}} = \mathbf{1}_n \otimes \hat{\boldsymbol{p}}$, we present the following proposition.

Proposition 1. *Consider the undesired equilibrium configuration $\hat{\boldsymbol{s}}$ characterized by conditions* (13a)–(13c)*. Let $\tilde{\boldsymbol{s}}$ be a translational displacement from $\hat{\boldsymbol{s}}$ obtained as $\tilde{\boldsymbol{s}} = \hat{\boldsymbol{s}} + \Delta\bar{\boldsymbol{p}}$ where $\Delta\bar{\boldsymbol{p}} = \bar{\boldsymbol{p}} - \bar{\hat{\boldsymbol{p}}}$. Then, the resulting error variable $\tilde{\boldsymbol{e}} = \tilde{\boldsymbol{s}} - \boldsymbol{s}^\star$ is such that $\tilde{\boldsymbol{e}}_N = \hat{\boldsymbol{e}}$.*

Proof. Since $\tilde{\boldsymbol{e}}_N = (\boldsymbol{I}_{3n} - \boldsymbol{L}_{\tilde{\boldsymbol{e}}}\boldsymbol{L}_{\tilde{\boldsymbol{e}}}^\dagger)(\tilde{\boldsymbol{s}} - \boldsymbol{s}^\star)$, then, replacing the definition of $\tilde{\boldsymbol{s}}$, one has

$$\tilde{\boldsymbol{e}}_N = (\boldsymbol{I}_{3n} - \boldsymbol{L}_{\tilde{\boldsymbol{e}}}\boldsymbol{L}_{\tilde{\boldsymbol{e}}}^\dagger)(\bar{\boldsymbol{p}} - \bar{\hat{\boldsymbol{p}}} + \hat{\boldsymbol{e}}).$$

Since $\hat{\boldsymbol{e}}$ is, by definition, such that the vector $\hat{\boldsymbol{e}} \in \ker(\boldsymbol{L}_{\tilde{\boldsymbol{e}}}^\top)$ then

$$\tilde{\boldsymbol{e}}_N = \hat{\boldsymbol{e}} + (\boldsymbol{I}_{3n} - \boldsymbol{L}_{\tilde{\boldsymbol{e}}}\boldsymbol{L}_{\tilde{\boldsymbol{e}}}^\dagger)(\bar{\boldsymbol{p}} - \bar{\hat{\boldsymbol{p}}}). \tag{19}$$

Finally, the term $\bar{\boldsymbol{p}} - \bar{\hat{\boldsymbol{p}}}$ can be written as

$$\bar{\boldsymbol{p}} - \bar{\hat{\boldsymbol{p}}} = \boldsymbol{L}_{\tilde{\boldsymbol{e}}} \begin{bmatrix} \hat{\boldsymbol{p}} - \boldsymbol{p} \\ \boldsymbol{0}_3 \end{bmatrix}.$$

Thus, $(\bar{\boldsymbol{p}} - \bar{\hat{\boldsymbol{p}}}) \in \mathrm{Im}(\boldsymbol{L}_{\tilde{\boldsymbol{e}}})$ that implies $\tilde{\boldsymbol{e}}_N = \hat{\boldsymbol{e}}$. □

Remark 1. *The result presented in Proposition 1 states that any point on the manifold $\mathcal{N}_s$ is reachable from the undesired equilibrium point $\hat{\boldsymbol{s}}$ by means of a pure translation. The manifold $\mathcal{N}_e$ in the feature error space corresponding to $\mathcal{N}_s$ can be denoted as $\mathcal{N}_e = \left\{\boldsymbol{e} \in \mathbb{R}^{3n}, \boldsymbol{e} = \boldsymbol{e}_I + \boldsymbol{e}_N | \boldsymbol{e}_N = \hat{\boldsymbol{e}}\right\}$.*

Given the above reasoning, we are now ready to analyze the behaviour of the system (10) in the neighbourhood of $\mathcal{N}_e$.

Proposition 2. *With reference to the system* (10)*, the only trajectories $\boldsymbol{e} = \boldsymbol{e}(t)$ that asymptotically converge to $\hat{\boldsymbol{e}}$ are those starting from $\boldsymbol{e}_0 \in \mathcal{N}_e$.*

Proof. From Proposition 1, any $\boldsymbol{s}_0 \in \mathcal{N}_s$ can be generated as a pure translation from $\hat{\boldsymbol{s}}$, that is $\boldsymbol{s}_0 = \hat{\boldsymbol{s}} + \Delta\bar{\boldsymbol{p}}_0$, with $\Delta\bar{\boldsymbol{p}}_0 = \bar{\boldsymbol{p}}_0 - \bar{\hat{\boldsymbol{p}}}$. Thus, for any given $\boldsymbol{s}^\star$, $\boldsymbol{e}_0 = \hat{\boldsymbol{e}} + \Delta\bar{\boldsymbol{p}}_0 \in \mathcal{N}_e$. Since, $\boldsymbol{e}_N = \hat{\boldsymbol{e}}$ for any point $\boldsymbol{e}$ in $\mathcal{N}_e$, then it results $\boldsymbol{e}_{I0} = \Delta\bar{\boldsymbol{p}}_0$.

Given the feature error dynamics in (10), it holds that

$$\dot{\boldsymbol{e}} = -\lambda\boldsymbol{L}_{\boldsymbol{e}}\boldsymbol{L}_{\boldsymbol{e}}^\dagger\boldsymbol{e} = -\lambda\boldsymbol{e}_I, \quad \boldsymbol{e}(0) = \hat{\boldsymbol{e}} + \Delta\bar{\boldsymbol{p}}_0. \tag{20}$$

Recalling that $\boldsymbol{e} = \boldsymbol{e}_I + \boldsymbol{e}_N$, we can write (20) as

$$\dot{\boldsymbol{e}}_I = -\lambda\boldsymbol{e}_I - \dot{\boldsymbol{e}}_N, \quad \boldsymbol{e}_I(0) = \Delta\bar{\boldsymbol{p}}_0. \tag{21}$$

Since by virtue of Proposition 1, and as highlighted in Remark 1, the feature error trajectories $\boldsymbol{e} = \boldsymbol{e}(t)$ on the manifold $\mathcal{N}_e$ can be generated as pure translations, i.e. $\boldsymbol{e}_N = \hat{\boldsymbol{e}}$, then the solution of the system in (21) is

$$\begin{aligned} \boldsymbol{e}_N &= \hat{\boldsymbol{e}}, \\ \boldsymbol{e}_I &= e^{-\lambda t}\Delta\bar{\boldsymbol{p}}_0. \end{aligned} \tag{22}$$

This implies that, even though the undesired equilibrium points are unstable, as proven in [14], trajectories converging to them exist and $\mathcal{N}_e$ results to be an invariant manifold.

Now we have to prove that they are the only ones. Consider the Lyapunov candidate function

$$V(\boldsymbol{e}) = \frac{1}{2}(\boldsymbol{e} - \hat{\boldsymbol{e}})^\top(\boldsymbol{e} - \hat{\boldsymbol{e}}), \tag{23}$$

note that, since $\boldsymbol{e} = \boldsymbol{e}_I + \boldsymbol{e}_N$, on the manifold $\mathcal{N}_e$, i.e., $\boldsymbol{e}_N = \hat{\boldsymbol{e}}$, the Lyapunov surfaces lie on $\mathcal{N}_e$ and (23) reduces to

$$V(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}_I^\top\boldsymbol{e}_I, \tag{24}$$

then, its time derivative along the error dynamics (10) is

$$\dot{V}(\boldsymbol{e}) = -\lambda\boldsymbol{e}_I^\top\boldsymbol{e}_I, \tag{25}$$

as expected from the result in (22). Outside the manifold $\mathcal{N}_e$, i.e. $\boldsymbol{e}_N \neq \hat{\boldsymbol{e}}$, we have to consider the Lyapunov function in (23) and its time derivative along the error dynamics in (10) results in

$$\dot{V}(\boldsymbol{e}) = -\lambda\boldsymbol{e}_I^\top\boldsymbol{e}_I + \lambda\hat{\boldsymbol{e}}^\top\boldsymbol{e}_I, \tag{26}$$

where the first term is quadratic with respect to $\boldsymbol{e}_I$ and negative definite, while the second one depends linearly on $\boldsymbol{e}_I$ and its sign is not defined. This means that if the trajectory starts at $t = 0$ such that $\|\boldsymbol{e}_{I0}\|^2 > \hat{\boldsymbol{e}}^\top\boldsymbol{e}_{I0}$, $\dot{V}(\boldsymbol{e})$ is negative and it remains negative as long as the above inequality holds. Therefore, as the norm $\|\boldsymbol{e} - \hat{\boldsymbol{e}}\|$ decreases, also $\|\boldsymbol{e}_I\|$ reduces since for $\boldsymbol{e} = \hat{\boldsymbol{e}}$ it is $\boldsymbol{e}_I = \boldsymbol{0}$. Thus, we can state that there exists a time instant $\bar{t}$ such that $\dot{V}(\boldsymbol{e})$ becomes definitely positive for any $t > \bar{t}$. This in turn implies that the norm $\|\boldsymbol{e} - \hat{\boldsymbol{e}}\|$ increases. This proves the statement. □

It is now possible to formulate the following proposition.

Proposition 3. *The origin of the system* (10) *is almost globally asymptotically stable for any $\lambda > 0$.*

Proof. The proof follows straightforwardly from Propositions 2, since the only trajectories not converging to $\boldsymbol{e} = \boldsymbol{0}$ are those belonging to zero Lebesgue measure sets $\mathcal{N}_e \subset \mathbb{R}^{3n}$. □

C. Convergence analysis

Now we analyze the convergence character of the feature trajectories as it is strongly related to the performance of the robotic system during the visual servoing task. Specifically, it can be proved that the trajectories starting with $\theta(0)$ and $\boldsymbol{r}(0)$, close to satisfy (13a) and (13c), respectively, converge to $\boldsymbol{e} = \boldsymbol{0}$ with an initial low rotational speed, and, thus, a long transient. This behaviour can be observed by computing the camera angular velocity $\boldsymbol{\omega}$ when $\theta = \hat{\theta} + \delta\theta$, i.e., when θ is close to $\pm\pi$, and $\boldsymbol{r} = \hat{\boldsymbol{r}}$. By considering the explicit form of $\boldsymbol{L}_{\boldsymbol{e}}^\dagger$ expressed in terms of camera pose derived in [5] and written in the case $\boldsymbol{s}_i = \boldsymbol{p}_i$, one has

$$\boldsymbol{L}_{\boldsymbol{e}}^\dagger = \begin{bmatrix} \cdots & -\frac{1}{n}\boldsymbol{I}_3 + \boldsymbol{S}(\boldsymbol{p})\boldsymbol{R}\boldsymbol{G}^{-1}\boldsymbol{S}(\boldsymbol{s}_i^\star - \boldsymbol{s}_\Sigma^\star)\boldsymbol{R}^\top & \cdots \\ & \boldsymbol{R}\boldsymbol{G}^{-1}\boldsymbol{S}(\boldsymbol{s}_i^\star - \boldsymbol{s}_\Sigma^\star)\boldsymbol{R}^\top & \end{bmatrix}, \tag{27}$$

where $\boldsymbol{s}_\Sigma^\star$ is defined in (16) and $\boldsymbol{G} = \sum_{i=1}^n \boldsymbol{S}^2(\boldsymbol{s}_i^\star - \boldsymbol{s}_\Sigma^\star)$. Substituting (27) in (9), the components of the control input can be explicitly written as

$$\boldsymbol{\omega} = -\lambda\boldsymbol{R}\boldsymbol{G}^{-1}\sum_{i=1}^n \boldsymbol{S}(\boldsymbol{s}_i^\star - \boldsymbol{s}_\Sigma^\star)(\boldsymbol{I}_3 - \boldsymbol{R}^\top)(\boldsymbol{s}_i^\star - \boldsymbol{s}_\Sigma^\star), \tag{28}$$

$$\mathbf{v} = \lambda\boldsymbol{p} + \boldsymbol{S}(\boldsymbol{p})\boldsymbol{\omega}. \tag{29}$$

Recalling the Euler-Rodrigues rotation formula and the equivalence $\boldsymbol{S}^2(\hat{\boldsymbol{r}}) = \hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^\top - \boldsymbol{I}_3$ (see [14]), it is easy to prove that any rotation matrix $\boldsymbol{R}$ can be replaced, in a neighborhood of, e.g., $\theta = \pi$, by $\tilde{\boldsymbol{R}}$, where

$$\tilde{\boldsymbol{R}} = (\pi - \theta)\boldsymbol{S}(\hat{\boldsymbol{r}}) + \hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^\top. \tag{30}$$

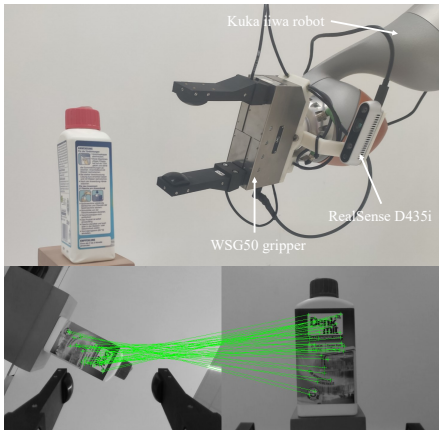

Fig. 2. Top: Experimental setup and sample initial robot configuration with $\boldsymbol{r}(0) = [0\,0\,1]^\top$ and $\theta(0) = 120°$; Bottom: 37 matched features.

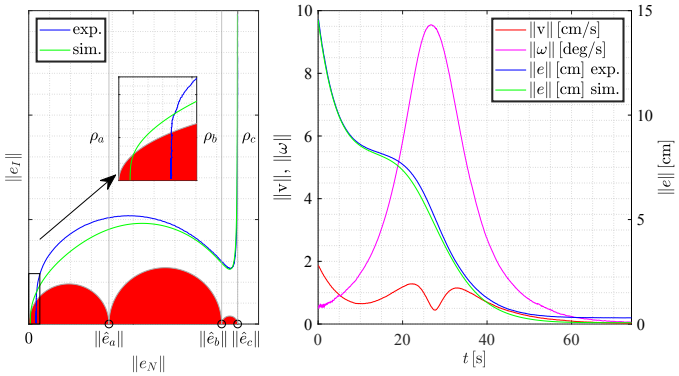

Fig. 3. Experiment with the initial angle $\theta(0) = 178°$. Left plot: the error trajectory on the plane $(\|\boldsymbol{e}_N\|, \|\boldsymbol{e}_I\|)$ for the experiment (blue curve) and a simulation (green line). Right plot: corresponding linear and angular velocities (red and magenta curves, respectively - left axis) and error norm (blue curve right axis).

Then, replacing (30) in (28), the angular velocity $\boldsymbol{\omega}$ can be approximated as

$$\begin{aligned}\boldsymbol{\omega} &= \lambda(\pi-\theta)\tilde{\boldsymbol{R}}\boldsymbol{G}^{-1}\sum_{i=1}^{n}\boldsymbol{S}(\boldsymbol{s}_i^\star-\boldsymbol{s}_\Sigma^\star)\boldsymbol{S}^\top(\hat{\boldsymbol{r}})(\boldsymbol{s}_i^\star-\boldsymbol{s}_\Sigma^\star)\\ &\quad+2\lambda\tilde{\boldsymbol{R}}\boldsymbol{G}^{-1}\sum_{i=1}^{n}\boldsymbol{S}(\boldsymbol{s}_i^\star-\boldsymbol{s}_\Sigma^\star)\hat{\boldsymbol{r}}\hat{\boldsymbol{r}}^\top(\boldsymbol{s}_i^\star-\boldsymbol{s}_\Sigma^\star)\\ &= \lambda(\pi-\theta)\hat{\boldsymbol{r}}+2\lambda\tilde{\boldsymbol{R}}\boldsymbol{G}^{-1}\boldsymbol{S}^\top(\hat{\boldsymbol{r}})\sum_{i=1}^{n}(\boldsymbol{s}_i^\star-\boldsymbol{s}_\Sigma^\star)\hat{\boldsymbol{r}}^\top(\boldsymbol{s}_i^\star-\boldsymbol{s}_\Sigma^\star)\end{aligned} \tag{31}$$

Finally, by means of simple manipulation of the term inside the summation operator, (31) can be equivalently rewritten as

$$\boldsymbol{\omega} = \lambda(\pi-\theta)\hat{\boldsymbol{r}}+2\lambda\tilde{\boldsymbol{R}}\boldsymbol{G}^{-1}\boldsymbol{S}^\top(\hat{\boldsymbol{r}})\widehat{\boldsymbol{M}}\hat{\boldsymbol{r}}. \tag{32}$$

where $\widehat{\boldsymbol{M}}$ is defined as in (15). Thus, noting that $\hat{\boldsymbol{r}}$ is eigenvector of $\widehat{\boldsymbol{M}}$ and replacing θ with $\hat{\theta}+\delta\theta$ it holds that

$$\boldsymbol{\omega} = \lambda\delta\theta\hat{\boldsymbol{r}}. \tag{33}$$

Thus, the smaller $\delta\theta$, the smaller the angular velocity $\boldsymbol{\omega}$.

This behaviour has been shown in a series of experiments performed with a Kuka LBR iiwa 7 robot equipped with an Intel RealSense D435i RGB-D camera mounted in an *eye-in-hand* configuration. Initially, the 3D features are extracted both from the camera in the initial configuration and the target image. Then, the collected features are matched and, during the task execution, the Kanade–Lucas–Tomasi (KLT) tracking algorithm [18] keeps track of the current features collected from the current camera image. The matching and tracking algorithms are provided by the ViSP library [19]. Figure 2 shows the setup with an initial angle $\theta(0) = 120°$, with the bottom figure showing the image acquired in the initial condition and the target one as well as the 37 matched features. In the proposed experiments the robot initial pose is selected such that $\boldsymbol{r}(0) = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top$, satisfying condition (13c) and $\boldsymbol{p}(0) = \begin{bmatrix}0 & 0 & -0.12\end{bmatrix}^\top$ m. In the first experiment the initial angle has been selected as $\theta(0) = 178°$ and the results are shown in Fig. 3. The left plot in Fig. 3 synthetically represents the error trajectory on the plane $(\|\boldsymbol{e}_N\|, \|\boldsymbol{e}_I\|)$. The blue and green lines represent the experiment and the simulation results, respectively, both initialized with the same initial condition $\boldsymbol{e}(0)$. Note that, initially, both experimental and simulated trajectories are very close to the manifold $\mathcal{N}_e$ (this is equivalently represented in the plane in Fig. 3-left as the couple $(\|e_N\|, \|e_I\|)$ being close to ρ_c line) and $\boldsymbol{e}_N$ is almost constant. This effect is highlighted in the velocity plot (Fig.3 right plot, left axis), where the initial linear velocity is maximum (2 cm/s) and the rotational one is relatively low (0.5 deg/s), thus the camera motion is nearly a pure translation. This corresponds, in the left plot, to the first part of the trajectory in the plane $(\|e_N\|, \|e_I\|)$ lying close to the ρ_c line. As the angular velocity increases the trajectory moves away from the ρ_c line and converges towards $\|\boldsymbol{e}\| = \boldsymbol{0}$. As a consequence, the error norm dynamics is not exponential (Fig. 3 right plot, right axis) but it presents a change of concavity in a point of flex with a small derivative. This results in a very small camera velocity in the time interval $[8.8,\ 20]$ s followed by an increase in the angular velocity. Note that the curves resulting from simulation and experiment are very close at the beginning but neither of them converge exactly to zero. Specifically, they converge in the red area where the rigid motion constraint (11) is not satisfied. This behavior is caused by the unavoidable errors of the matching and tracking algorithms that generate features that do not perfectly satisfy the rigid motion constraint. Specifically, the simulation curve is only affected by matching errors in the initial condition $\boldsymbol{e}(0)$, while the experimental curve is also affected by the tracking error that may accumulate during the visual servoing task (see the zoomed frame in Fig. 3-right).

Figure 4 shows the same experiment repeated, firstly, with different initial rotation angles, i.e., $\theta(0) = 170°$ and $160°$, and, then, by changing both the rotation axis and angle, i.e., $\boldsymbol{r}(0) = [0.2, -0.05, -0.98]^\top$ and $\theta(0) = 122°$. It is clear that, as the initial angle is farther from the critical condition (13a), the described effect is less evident. In particular, when $\theta(0) = 160°$ the point of flex is not visible anymore and when $\theta(0) = 122°$ the trajectory starts far from the $\mathcal{N}_e$ manifold and rapidly converges to $\boldsymbol{e} = \boldsymbol{0}$. Finally, also the

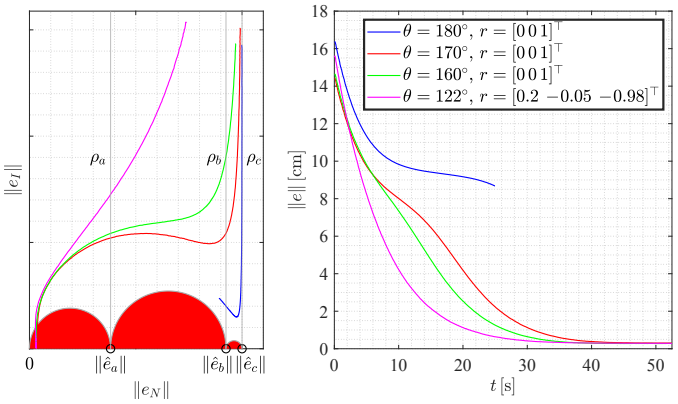

Fig. 4. Experiments repeated with $\boldsymbol{r} = [0, 0, 1]^\top$ and initial rotation angles $\theta(0)$ equal to 170° (red), 160° (green), 180° (blue), and 122° with $\boldsymbol{r} = [0.2\ -0.05\ -0.98]^\top$ (magenta).

special case where $\theta(0) = 180°$ has been tested. As expected, even if the initial error state belongs to the manifold $\mathcal{N}_e$, due to the camera noise, the trajectory does not belong to it but evolves remaining close to $\mathcal{N}_e$. This particular trajectory results in almost decoupled translational and rotational phases. In fact, the robot initially executes a mere translation to approach the object and then operates the required rotation. As a consequence, collision among the robot end effector and the object occurs during the rotation, thus requiring to stop the experiment (see Fig. 4, where the blue curve is interrupted at $t = 25\,\mathrm{s}$ due to the collision).

III. Keeping the Features Within the Camera FoV

A. Sufficient condition for guaranteed FoV constraint

In Section II, discussing stability and convergence issues, we have assumed that the set of the n matched features $(\boldsymbol{s}_i, \boldsymbol{s}_i^\star)$, $i = 1, \ldots, n$, does not decrease its dimension during the visual servoing task, i.e. the visual control has the capability to keep the current features $\boldsymbol{s}_i$ inside the camera FoV. This requirement is equivalent to the condition

$$(u_i(t), v_i(t)) \in \mathcal{V} \quad \forall i,\ \forall t \geq 0, \tag{34}$$

$$\mathcal{V} = \{(u, v) \in \mathbb{R}^2 \colon u \in [0,\ u_{\max}), v \in [0,\ v_{\max})\}, \tag{35}$$

where (u_i, v_i) are the pixel coordinates corresponding to the feature $\boldsymbol{s}_i$ and the rectangle $\mathcal{V}$ represents the camera FoV.

In [5] it has been proved that, during the visual task, the feature centroid evolves in $\mathbb{R}^3$ as a straight line, stating that *most likely* the features remain in the camera FoV during the whole task. However, there exist initial conditions close to the boundary of $\mathcal{V}$ such that (34) does not hold. By resorting to simple geometric reasoning, it is possible to find a sufficient condition in order for the feature trajectories to satisfy (34). Figure 5 shows a 3D representation of the feature perspective projection on the camera plane. Let us define a 3D sphere $\mathcal{B}(\boldsymbol{s}_\Sigma, \rho_s)$ centered in the feature centroid $\boldsymbol{s}_\Sigma = \begin{bmatrix} x_s & y_s & z_s \end{bmatrix}^\top = \frac{1}{n}\sum_{i=1}^n \boldsymbol{s}_i$ with radius

$$\rho_s = \max_i \|\boldsymbol{s}_i - \boldsymbol{s}_\Sigma\|. \tag{36}$$

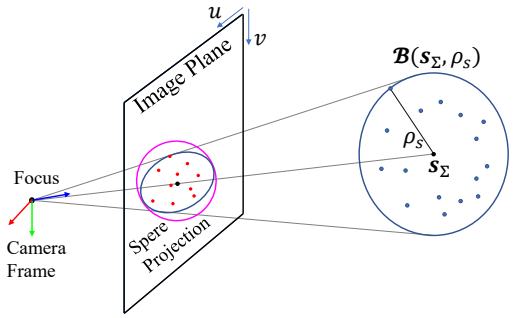

Fig. 5. The sphere wrapping all the 3D features (blue) projected as ellipse in the image plane. The smallest circumference wrapping the red ellipse is depicted in magenta on the image plane. The blue dots represent the 3D feature points, while the red ones represent the corresponding projected 2D features.

Note that, by virtue of the rigid motion constraint, the radius ρ_s remains constant replacing $\boldsymbol{s}_i$ with $\boldsymbol{s}_i^\star$ and $\boldsymbol{s}_\Sigma$ with $\boldsymbol{s}_\Sigma^\star$.

The perspective projection of the sphere on the image plane is an ellipse in which all the projected features (u_i, v_i) are contained. If the ellipse is fully contained in $\mathcal{V}$, then the features are in the camera FoV. As shown in [5], during the closed-loop visual servoing phase, the centroid position dynamics can be expressed as

$$\dot{\boldsymbol{s}}_\Sigma = -\lambda(\boldsymbol{s}_\Sigma - \boldsymbol{s}_\Sigma^\star), \tag{37}$$

describing a straight line with respect to the camera frame. Thus, the sphere center motion can be parametrized as

$$\boldsymbol{s}_\Sigma(\gamma) = \gamma \boldsymbol{s}_\Sigma^\star + (1 - \gamma)\boldsymbol{s}_{\Sigma 0}, \quad \gamma \in [0,\ 1], \tag{38}$$

where $\boldsymbol{s}_{\Sigma 0}$ is the feature centroid at the beginning of the task and γ specifies a point on the 3D line segment. The projected ellipse can be obtained from the spheroid perspective projection formula [20] adapted to the special case of sphere projection. It is convenient to describe this ellipse in the image plane using metric units in place of pixel units. After deriving the ellipse in metric units, it is then possible to convert its equation in pixel units. Therefore, without loss of generality, in what follows we will consider the pixel width and height being unitary, that is $l_x = l_y = 1$, and the center of the camera in pixel coordinates corresponding to $C_x = 0$, $C_y = 0$.

From [20], the coordinates of the center (x_e, y_e), the semi-major axis length a_e and the semi-minor axis length b_e of the ellipse in Fig. 5 are expressed as

$$\begin{bmatrix} x_e & y_e \end{bmatrix}^\top = f\frac{z_s}{z_s^2 - \rho_s^2}\begin{bmatrix} x_s & y_s \end{bmatrix}^\top, \tag{39}$$

$$a_e = f\rho_s\sqrt{\frac{1}{z_s^2 - \rho_s^2}\left(1 + \frac{x_s^2 + y_s^2}{z_s^2 - \rho_s^2}\right)}, \tag{40}$$

$$b_e = f\rho_s\sqrt{\frac{1}{z_s^2 - \rho_s^2}}. \tag{41}$$

Note that the ellipse parameters depend on γ since $\boldsymbol{s}_\Sigma$ moves on the line segment as in (38). Moreover, parameters (39)-(41) do not fully describe the ellipse as they do not include its orientation. However, the knowledge of the ellipse orientation in the image plane can be avoided by wrapping the projected

TABLE I
INTEL REALSENSE D435I RGB-D INTRINSIC CAMERA PARAMETERS.

f [mm]	l_x [μm]	l_y [μm]	c_x	c_y	$u_{\max}$	$v_{\max}$
1.88	3.0732	3.0731	314.7	246.49	640	480

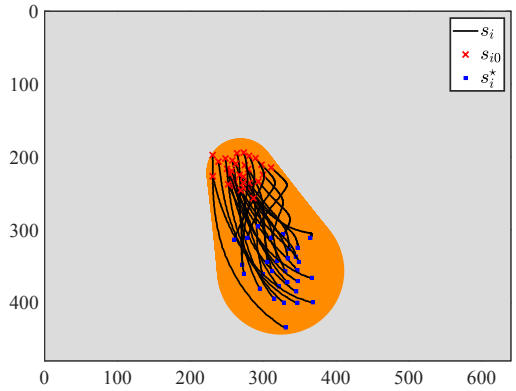

Fig. 6. Image plane obtained by an experiment. Orange area represents the motion of the projected 3D sphere always enclosing the feature trajectories.

ellipse within a circumference having radius a_e and equation $(x-x_e)^2+(y-y_e)^2=a_e^2$ (see the magenta line in Fig. 5). Finally, the wrapping circumference equation can be translated in pixel units as

$$\frac{(u-(x_e/l_x+C_x))^2}{a_e^2/l_x^2}+\frac{(v-(y_e/l_y+C_y))^2}{a_e^2/l_y^2}=1, \quad (42)$$

which, since $l_x \neq l_y$, describes an ellipse in the image plane.

The above analysis guarantees that, given the feature centroids at $t=0$ $\boldsymbol{s}_{\Sigma 0}$ and $\boldsymbol{s}_\Sigma^\star$, if for all $\gamma \in [0,1]$ the resulting 2D ellipse in pixel coordinates having equation (42) lies inside the rectangle $\mathcal{V}$, then the feature points are ensured to remain within the camera FoV for all the visual servoing task. Nevertheless, by virtue of (37), the dynamics of the coordinates of the centroid are decoupled and they converge to the desired centroid coordinates with the same time constant, thus the projected ellipsoid describes a convex hull. Now, a sufficient condition on $\boldsymbol{s}_0$ ensuring that condition (34) is satisfied can be stated as follows.

Let the desired centroid $\boldsymbol{s}_\Sigma^\star$ generate an ellipse that lies within $\mathcal{V}$, compute a_e, x_e and y_e for $\boldsymbol{s}_\Sigma = \boldsymbol{s}_{\Sigma 0}$. If the resulting ellipse lies within $\mathcal{V}$, then the feature trajectories are guaranteed to remain within the FoV during the whole task.

The importance of such condition lays from the fact that the feature trajectories in the image plane can be predicted only by running a simulation beforehand. However, since it is possible to easily construct the convex hull starting from $\boldsymbol{s}_{\Sigma 0}$ and $\boldsymbol{s}_\Sigma^\star$, it is possible to predict if the feature trajectory are guaranteed to belong to the camera FoV. An example is given in Figure 6 for the experiment with $\theta(0)=122°$ described in Section II. The image plane is expressed in pixel coordinates and the camera parameters are selected as in Table I. The orange area represents the motion of the projected 3D sphere on the image plane, obtained by the ellipse in (42) varying $\gamma \in [0,\,1]$ in (38). The plot shows the trajectory of 30 features, where the crosses represent the initial features $\boldsymbol{s}_0$ and the squares the desired ones $\boldsymbol{s}^\star$. Note that the feature trajectory lies within the orange area and the above condition is not violated.

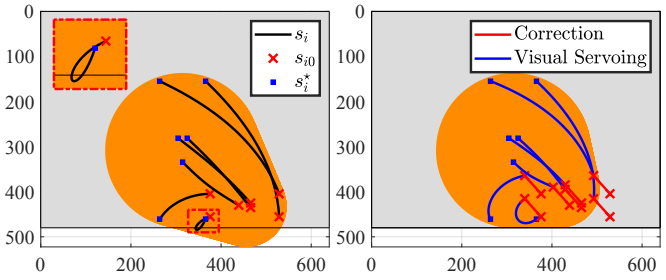

Fig. 7. Simulation: camera FoV violation (left) and proposed solution (right).

B. Algorithm to prevent feature loss

Violating the sufficient condition given in the previous subsection could potentially yield features to leave the camera FoV (35), which would result in features loss. As a consequence, this may cause discontinuities in the error variable, which generate undesired spikes in the velocity control signal.

An example is given in Fig. 7-left, where the sufficient condition fails (note that the initial 2D ellipse in pixel coordinates does not lie completely inside the rectangle $\mathcal{V}$) and one of the features leaves the camera FoV. An algorithm to avoid the feature loss is described in the following. The idea is to exploit (38) in order to guarantee that the camera FoV constraint is satisfied. Specifically, the proposed algorithm can be synthesized in the following steps:

- find γ', while keeping the z_s coordinate fixed to its initial value, such that the centroid defined as $\boldsymbol{s}'_\Sigma = \begin{bmatrix} x_s(\gamma') & y_s(\gamma') & z_s(0) \end{bmatrix}^\top$ generates a projected ellipse that lies in $\mathcal{V}$.
- move the camera, while keeping fixed the camera orientation, i.e., $\boldsymbol{\omega} = \mathbf{0}$, such that the current centroid moves toward $\boldsymbol{s}'_\Sigma$ by applying the following control law $\mathbf{v} = \lambda(\boldsymbol{s}_\Sigma - \boldsymbol{s}'_\Sigma)$, $\boldsymbol{\omega} = \mathbf{0}$.
- apply the visual servoing control law (9) as soon as the features centroid $\boldsymbol{s}_\Sigma$ is sufficiently close to the desired centroid configuration $\boldsymbol{s}'_\Sigma$.

The proposed solution has been applied to the feature configuration in Fig 7-left and the resulting trajectories, obtained in simulation, are shown in Fig 7-right. The red lines represent the feature trajectory in the first phase, that is when the control law $\mathbf{v} = \lambda(\boldsymbol{s}_\Sigma - \boldsymbol{s}'_\Sigma)$ is applied. Note that, since the centroid is moving on a straight line and the camera rotation is fixed ($\boldsymbol{\omega} = \mathbf{0}$), all the features also follow a straight line, thus they cannot violate the camera FoV constraint during this phase. As soon as the sufficient condition is satisfied, the classical visual servoing control (9) is applied and the feature trajectories cannot leave the orange area anymore and, in turn, the camera FoV. This second phase is represented by the blue lines in Fig. 7-right.

The proposed algorithm has been also tested through experiments and the resulting feature trajectories are shown in Fig. 8. Specifically, the initial set of features is such that the generated 2D ellipse does not entirely lie in $\mathcal{V}$ (Fig. 8-left). Application

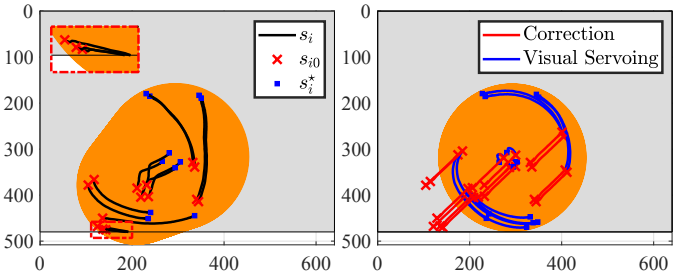

Fig. 8. Experiment: camera FoV violation (left) and proposed solution (right).

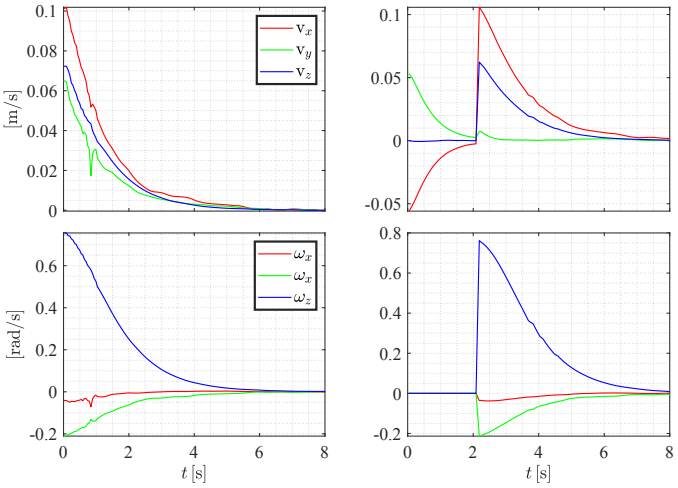

Fig. 9. Control velocities generated without (left) and with (right) the proposed algorithm. Linear velocities are shown in the top plots, angular velocities are shown in the bottom ones. In the right plots, the corrective phase is active up to $t = 2\,\text{s}$.

of the visual control law (9) causes three features to leave $\mathcal{V}$ (see the zoomed plot in Fig. 8-left), thus generating spikes in the velocity control signal (Fig. 9-left at $t = 0.8\,\text{s}$). In order to avoid this undesired phenomenon, the proposed algorithm has been applied and results are shown in Fig. 8-right, where the control law $\mathbf{v} = \lambda(\boldsymbol{s}_\Sigma - \boldsymbol{s}'_\Sigma)$, $\boldsymbol{\omega} = \mathbf{0}$ is applied to move the centroid towards $\boldsymbol{s}'_\Sigma$ such that the 2D ellipse lies within $\mathcal{V}$ (see the trajectories in red color). This results into the linear and angular velocities shown in Fig. 9-right up to $t = 2\,\text{s}$. Then, the classical visual control (9) is applied (generating the velocities in Fig. 9-right for $t > 2\,\text{s}$), and no features escape the camera FoV (see the trajectories in blue color in Fig. 8-right). As a result, no velocity spikes are generated. The accompanying video shows the visual servoing task when features are lost and when feature loss is prevented by the proposed algorithm.

IV. Conclusion

In this letter the stability and convergence analysis of a 3D feature-based VS scheme has been thoroughly analyzed extending existing results. Specifically, it has been proved that the desired equilibrium point of the visual control system is *almost globally asymptotically stable* in the sense that the only trajectories not converging to the desired equilibrium point are those belonging to a zero Lebesgue measure set in the feature space. Furthermore, the behaviour of the error state trajectories starting close to this zero Lebesgue measure set is described both theoretically and with experiments.

Using the coordinates of the 2D image points and the corresponding depths as visual features would be an interesting approach to investigate to avoid propagation of the depth noise on the estimation of the first two coordinates.

Finally, analysis of the camera FoV constraint has been carried out providing a sufficient condition guaranteeing that the feature trajectories remain in the camera FoV during the visual servoing task. When such condition is not satisfied, an algorithm is proposed to avoid feature loss and to ensure that the feature trajectories remain inside the camera FoV.

References

[1] P. I. Corke, "Visual control of robot manipulators – a review," in *Visual Servoing*. World Scientific, Oct. 1993, pp. 1–31.
[2] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 679–699, 2002.
[3] E. Malis, G. Chesi, and R. Cipolla, "2 $\frac{1}{2}$d visual servoing with respect to planar contours having complex and unknown shapes," *The International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 841–853, 2003.
[4] P. Martinet, J. Gallice, and K. Djamel, "Vision Based Control Law using 3D Visual Features," in *World Automation Congress, WAC'96, Robotics and Manufacturing Systems*, 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," *The International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 821–839, 2003.
[6] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
[7] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
[8] ——, "Visual servo control, Part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.
[9] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Experimental Robotics III*, T. Yoshikawa and F. Miyazaki, Eds. Springer Berlin Heidelberg, 1994, pp. 182–192.
[10] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.
[11] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.
[12] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, 2002.
[13] F. Berry and E. Cervera P. Martinet, "Image based stereo visual servoing: 2D vs 3D features." in *Proceedings of the 15th IFAC World Congress on Automatic Control, IFAC'02*. Spain: IFAC, 2002.
[14] F. Schramm, G. Morel, A. Micaelli, and A. Lottin, "Extended-2d visual servoing," in *IEEE International Conference on Robotics and Automation, 2004. Proc. ICRA '04. 2004*, vol. 1, 2004, pp. 267–273.
[15] A. Rantzer, "A dual to lyapunov's stability theorem," *Systems & Control Letters*, vol. 42, no. 3, pp. 161–168, 2001.
[16] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. USA: Cambridge University Press, 2017.
[17] J. S. Dai, "Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections," *Mechanism and Machine Theory*, vol. 92, pp. 144–152, 2015.
[18] C. Tomasi and T. Kanade, "Detection and tracking of point features," International Journal of Computer Vision, Tech. Rep., 1991.
[19] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.
[20] D. S. Wokes and P. L. Palmer, "Perspective reconstruction of a spheroid from an image plane ellipse," *International Journal of Computer Vision*, vol. 90, no. 3, pp. 369–379, Jul. 2010.